NETWORKED
& ELECTRONIC
MEDIA

2012 NEM SUMMIT

www.nem-summit.eu

Implementing Future Media Internet Towards New Horizons

October 16-18, 2012 - Istanbul, Turkey

Conference Proceedings

www.nem-initiative.org

ISBN 978-3-00-039710-3

Publisher

Eurescom – the European Institute for Research and Strategic Studies in Telecommunications – GmbH
Wieblinger Weg 19/4 - 69123 Heidelberg - Germany
Phone: +49 6221 989 0 - Fax: +49 6221 989 209 - http://www.eurescom.eu
For publisher: Halid Hrasnica
On behalf of NEM Initiative – http://www.nem-initiative.org

eBook and USB produced by Sigma Orionis

1240, route des dolines - BP287 Valbonne - France
Phone: +33 (0) 493 001 550 - Fax: +33 (0) 493 001 560 - http://www.sigma-orionis.com
For producer: Roger Torrenti
On behalf of NEM Initiative – http://www.nem-initiative.org

Foreword

The NEM Initiative – the European Technology Platform on networked electronic media – with the support of the European Commission, is organizing fifth edition of Networked and Electronic Media Summit (NEM Summit) NEM Summit 2012. The NEM Summit aims to be a major annual conference and exhibition devoted to the field of networked and electronic media and ICT in general. The NEM Summit 2012 is supported by Sigma Orionis and Eurescom GmbH, whereas the main local support is ensured by Turk Telekom.

We are pleased to welcome all participants to the NEM Summit 2012 on October 16-18, 2012 in Istanbul, Turkey. This NEM Summit builds upon the success of the previous NEM Summit held in 2008 &, 2009 (St Malo, France), 2010 (Barcelona, Spain), and 2011 (Torino, Italy).

The NEM Summit 2012 is dedicated to the broad scope of R&D activities ***«Implementing Future Media Internet towards New Horizons»****. It will provide a unique opportunity to network and share information and viewpoints on the status of theoretical and practical work in this area and perspectives for the future. The objective of the Summit is to stimulate research and contribute to the solution of new problems encountered by scientists and engineers working in the fields of:*

- *Electronic media content*
- *Distributed media applications*
- *New Media delivery networks and network services*
- *User devices and terminals*
- *NEM enabling technologies.*

The NEM Summit Programme Committee has received a large number of interesting and high quality papers. After review of the papers by experts from the NEM area, 22 papers have been selected for the following Summit Scientific and Technical tracks and are published in the proceedings:

- ***Digital Media Content***
- ***Connected Media Worlds***
- ***Networked Media Experience***

In addition, five contributions have been selected for presentation at the Summit Application and Experimentation tracks, five keynote speeches, one open round table discussion, as well as several welcome and other addresses are included in the programme of the NEM Summit 2012.

As editors of the NEM Summit 2012 ***«Implementing Future Media Internet towards New Horizons»*** *proceedings, we would like to express our thanks to all persons involved in organisation of the Summit, particularly to the authors, members of the Summit Programme and Organisation Committees, as well as local sponsors and supporters of this great event.*

We wish you a fruitful event!

Editors: Jean-Dominique Meunier, Enis Erkel, Halid Hrasnica, and Roger Torrenti
On behalf of the NEM Initiative – http://www.nem-initiative.org

Table of content

Session 2B «Connected Media Worlds II»

Session 3A «Digital Media Content II»

Session 3B «Application & Experimentation track»

2012 NEM Summit Programme Committee

General NEM Summit 2012 Co-chairs:	Jean-Dominique Meunier (Technicolor)
	Enis Erkel (Turk Telekom)
Programme Committee Co-chairs:	Gozde Bozdagi Akar (Middle East Technical University)
	Thorsten Herfet (Intel)
Programme Committee Coordinator:	Halid Hrasnica (Eurescom)
Programme Committee Board members:	Jovanka Adzic (Telecom Italia)
	Reha Civanlar (Özygin University)
	Haluk Gökmen (Arcelik)
	Rowena Goldman (BBC)
	Jose Manuel Menendez (Universidad Politecnica de Madrid)
	Duygu Öktem (Turk Telekom)
	Julian Sesena (Rose Vision)
	Murat Tekalp (Koc University)
Observer:	Francisco Medeiros (European Commission)
Further Programme Committee members:	Federico Alvarez (Universidad Politecnica de Madrid)
	Jon Arambarri (Virtualware)
	Stefan Arbanowski (Fraunhofer FOKUS)
	Andy Bower (BBC)
	Federica Cena (University of Turin)
	Hadmut Holken (Holken Consultants)
	Richard Jacobs (BT)
	Amela Karahasanovic (SINTEF)
	Joachim Köhler (Fraunhofer IAIS)
	Artur Krukowski (Intracom)
	Goran Petrovic (Saarland University)
	Jukka Salo (Nokia Siemens Networks)
	Gwendal Simon (Telecom Bretagne)
	Alexandru Stan (IN2)
	Georg Thallinger (Joanneum Research)
	Graham Thomas (BBC)
	Lieven Trappeniers (Alcatel-Lucent)
In addition to the PC members paper review process has been supported by:	Fabio Varesano (University of Turin)
	Francesco Osborne (University of Turin)
	Amon Rapp (University of Turin)
	Elisa Chiabrando (University of Turin)
Organisation Committee Co-chairs:	Duygu Oktem (Turk Telekom)
	Roger Torrenti (Sigma Orionis)
Organisational Committee members:	Jovanka Adzic (Telecom Italia)
	Pierre-Yves Danet (Orange)
	Elçin Mol (Turk Telekom)
	Nga Tran (Sigma Orionis)
	Rowena Goldman (BBC)
	Halid Hrasnica (Eurescom)
	Yves-Marie Le Pannerer (Technicolor)
	Francisco Medeiros (European Commission)
	Ezgi Bener (TUBITAK)
	Viorel Peca (European Commission)
	Kemal Uyanik (Boogy)

Keynote speakers

Opening ceremony (17 October, 2012)

«Mobile multimedia meets the cloud»
Authors: Chang Wen Chen, State University of New York at Buffalo

Keynote session (18 October, 2012)

«TV World: 3D, Connected and Smart TVs, Interfaces and Trends»
Authors: Cem Kural, Arçelik

«4D Modelling: From Video to Interactive 3D Digital Media Content»
Authors: Adrian Hilton, University of Surrey

«E-Heritage, Cyber Archaeology and Cloud Museum»
Authors: Katsushi Ikeuchi, University of Tokyo

Closing ceremony (18 October, 2012)

«Bringing the Olympics to audiences in 8k Ultra High Definition»
Authors: John Zubryzycki, BBC R&D

Mobile Multimedia Meet Cloud: Challenges and Future Directions

Chang Wen CHEN

State University of New York at Buffalo, USA

Smart phones and tablets are becoming the most desired platforms for ubiquitous multimedia services.

When this contemporary trend of mobile media meets the increasing availability of public Clouds, a new technical paradigm, Cloud Mobile Media, is now emerging. This new paradigm presents numerous challenges for researchers to develop next generation cloud-driven media services for omnipresent mobile users.

This talk shall identify several major challenges in cloud-centric mobile media in properly discovering and seamlessly transporting the user desired media contents in their most appropriate form between the ubiquitous cloud infrastructures and the heterogeneous mobile devices. In particular, key factors that impact the cloud mobile media services, including service latency, user experience, mobility management, energy efficiency, and content security, will be examined.

This talk shall also outline some future research directions to further advance this emerging cloud mobile media by overcoming technical barriers resulting from the mismatch between resource abundant cloud infrastructures and severely resource limited mobile devices.

Digital Media Content I

Session 1A

Chaired by Jose Manuel Menendez, UPM

Predictable Reliability for Media Streaming over unmanaged Internet

Manuel Gorius[1], Thorsten Herfet[2]
Telecommunications Lab, Saarland University, Saarbruecken, Germany
E-mail: [1]gorius@cs.uni-saarland.de, [2]herfet@cs.uni-saarland.de

Abstract—**Managed Internet provides guaranteed Quality of Service to IP-based multimedia applications. Yet managed flows require support from the underlying infrastructure and rely on individual Service Level Agreements with the corresponding Service Provider. Therefore, QoS guarantees are missing for flows that are delivered beyond the service provider's infrastructure as well as over lossy home network segments. Outside of managed infrastructure, reliability and multiple access are ensured by self-managed end-to-end error and congestion control on transport layer. However, available transport layer protocols optimize their objectives in error and congestion control without respect to the application's individual QoS constraints such that multimedia services suffer from significant degradation. In this paper, we present a novel protocol layer that efficiently supports the reliability required by multimedia services under their individual delay constraint. Specifically, the protocol is designed to provide suchlike applications a self-managed, predictably reliable end-to-end channel on arbitrary Internet paths. The solution is based on the *Predictably Reliable Real-time Transport (PRRT)* protocol that efficiently optimizes proactive and reactive error control under strict time constraints. We evaluate PRRT's QoS provisioning on unmanaged Internet paths under the influences of congestion loss and compare the results with standardized QoS requirements as defined by the ITU-T Y.1541 QoS classes.**

Keywords: Predictable Reliability, IP Media Transport, Managed Internet, Quality of Service

1 INTRODUCTION

Despite the fact that the Internet Protocol has not been designed to deliver real-time media services at a high data rate, it is meanwhile established as an alternative carrier infrastructure for such content. As a result of their increasing popularity, a shift in paradigm towards *everything over IP* [6] becomes evident. The Internet Protocol (IP) is becoming increasingly important in the digital media broadcast, where the delivery of live media content via IP infrastructure is already widely deployed.

The requirements of the media transport differ significantly from those of file transfer. In particular, a continuous packet stream is offered to the network that requires synchronous handling at the end points despite the asynchronous nature of the IP infrastructure. Guidelines on individual requirements of interactive and non-interactive audio and video services have been standardized under the ITU-T Y.1541 QoS (Quality of Service) classes [14] with mainly the following dimensions:

- **Packet Loss Rate:** Compressed audiovisual media have a specific tolerance for packet loss. In case of video transmission the reliability requirements are strict. Therefore, the packet loss requirements are strict (one out of 10^5) in order to limit the number of visual artifacts. Audio codecs, however, can compensate for lost transmission units via interpolation such that a small percentage of packet loss is tolerable.
- **End-to-End Delay:** The delay requirement differs between interactive and non-interactive multimedia applications. While interactivity is considered to be degraded as soon as the bidirectional delay exceeds 150 to 250 milliseconds, non-interactive services might tolerate up to one second [14]. Especially for conversational applications the end-to-end delay variation should be less than 50 milliseconds, since the end devices implement a limited buffer space.
- **Bandwidth:** Multimedia streams have a rigid bandwidth requirement. Depending on their source coding, the bandwidth requirement is constant or variable over time [15]. A continuous rendering of the media stream is not possible if the bandwidth requirement is not fulfilled.

1.1 Problem Statement

The fundamental requirement for the IP media transport can be stated as finding a network path with sufficiently large bandwidth and low latency between arbitrary endpoints and efficiently utilizing the bandwidth available on this path. This requirement conflicts with several aspects of the current Internet architecture. As a result of queue overload and the transport-layer error control, the end-to-end delay of any single packet is hardly predictable such that continuous and timely delivery of media streams cannot be guaranteed. Continuous packet streams suffer especially under the congestion control policy of the Internet's dominant transport layer protocol – the Transmission Control Protocol (TCP), which aggressively fills network queues until it observes their overload from packet loss (TCP-induced packet loss).

Managed resource allocation is today's single option to predictably protect continuous media streams from queueing losses and delays on Internet paths. Consequently, commercial voice and video communications services rely on call bandwidth reservation where the QoS they require is guaranteed via Service Level Agreements (SLA). However, many network segments are not under the network service provider's control. This refers especially to scenarios in which the multimedia traffic is routed beyond the borders of the managed infras-

Corresponding author: Manuel Gorius, Saarland University, Campus C 6 3, 66123 Saarbruecken, Germany, +49 681 302-6544, gorius@cs.uni-saarland.de

tructure or over a wireless home network segment. Without explicit management, packet-switching Internet is a best-effort service implementing reliability on transport layer. Available transport layer protocols have a view over the whole end-to-end path but they do not offer any interface to communicate and negotiate QoS requirements [1]. In addition, adequate path monitoring techniques that are necessary to verify these requirements are missing within those implementations.

1.2 Related Work

Substantial work has been contributed in order to protect multimedia flows on unmanaged Internet paths. Solutions focussed mainly on three areas: partially reliable transport, the prediction of congestion losses as well as the exchange of QoS information across several layers of the Internet stack.

Under the *paradigm of partial reliability*, the Stream Control Transmission Protocol (SCTP) and the Datagram Congestion Control Protocol (DCCP) are recently developed transport layer protocols. SCTP enables the aggregation of several TCP-like connections and reliable datagram transmission. DCCP offers congestion control for unreliable datagram services. Partial reliability extensions (PR-SCTP [17], DCCP-PR [7]) have been specified for those protocols, both leading to similar correction performance by a limited number of packet repetitions in order to achieve limited delay. Being specified to be pure Automatic Repeat reQuest (ARQ) schemes with TCP-like congestion control, the correction performance suffers under tight delay constraints as well as in presence of large propagation delay in the delivery network. The Real-time Transport Protocol (RTP) shifts QoS provisioning to the application layer. Specifically, partial reliability is added by the profiles RTP/AVPF [8] and RTP/FEC [10]. These profiles provide packet repetition and Forward Error Coding (FEC), respectively, under the strict scalability rules of RTP/RTCP. This results in very limited correction performance, which might be adequate for voice transmission but insufficient for video streaming.

The *modeling and prediction of TCP-induced packet loss* has already been addressed in previous work. Bolot et al. performed modeling of congestion losses via the Gilbert-Elliot (GE) model, a two-state Markov chain [2]. The model has been used to adjust FEC for voice services. Roychoudhuri et al. proposed the prediction of congestion based on the measured one way delay (OWD) experienced by real-time services [11]. The authors consider the scheme as a basis for proactive FEC and rate control actions in order to protect real-time voice services from congestion losses. Seferoglu et al. developed a learning algorithm in order to dimension an FEC with respect to the observed propagation delay and its derivative [13]. The authors compare the classification of erasure events and their lengths based on RTT and OWD. Inherent problem of these predictive methods is the high probability of false detection, leading to a high residual packet loss rate.

In addition to *protocol enhancements*, *cross-layer approaches* have been proposed to deal with delay and loss constraints of multimedia applications. Yuksel et al. [19] proposed to inform the IP layer about the link layer loss rate in order to initiate fast failover to alternative routes in case of high loss rates. The scheme is specifically designed to support the unreliable multicast of IPTV services over UDP and RTP. It avoids the application of multicast reliability on transport layer, which is subject to scalability problems. The Autonomic Transport Framework developed by the LAAS Toulouse optimizes available transport and QoS allocation methods to offer partial order and partial reliability [3]. The framework specifies an Implicit Packet Meta Header (IPMH) that enables cross-layer communication of QoS requirements. For instance, a media stream's delay and reliability constraints could be made available to all layers in the multimedia communication stack in order for them to undertake respective actions to meet those constraints. However, this requires support from the underlying infrastructure.

1.3 Contribution

Section 2 of this paper presents a self-managing transport architecture operating under *predictable reliability* as an alternative approach to fulfill QoS requirements of multimedia applications on unmanaged Internet paths. Predictably reliable transport relies on the assumption that such services prefer timeliness over reliability in that they can tolerate a small amount of residual error. The protocol's error control is adaptively optimized under the application's QoS constraints and allows for predictable residual packet loss in order to guarantee timely delivery of real-time services. This optimization is based on stochastic modeling of the network state and the corresponding protocol performance under those constraints. The scheme is implemented into the Predictably Reliable Real-time Transport (PRRT)[1] protocol. In order to obtain predictable error control, the protocol combines proactive and reactive error control in an adaptive Hybrid Error Coding (HEC) approach [4], which leads to near-optimal coding efficiency under variable network conditions, where the channel capacity is dynamic.

In Section 3 we discuss characteristics of the network state under TCP-induced packet loss, in particular examining the packet loss rate and the burstiness of the packet loss process. The measurements show that both packet loss rate and burstiness depend significantly on the source rate of the media stream as well as the network path's Round Trip Time (RTT) as they are observed by the PRRT protocol. We evaluate PRRT's performance under an emulated bottleneck bandwidth that is shared with a variable number of TCP streams. The results show that PRRT implements a predictable trade-off between timeliness and coding overhead while fulfilling delay constraints of reasonably less than one second with an average residual packet loss rate (PLR) of 10^{-5} on Internet paths with moderate RTT.

[1] http://www.nt.uni-saarland.de/en/projects/running-projects/prrt.html

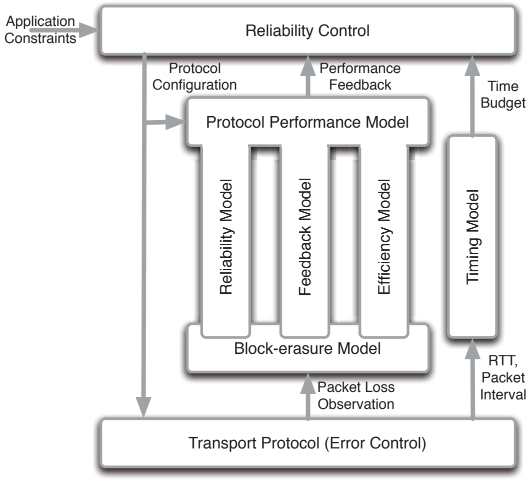


Figure 1: Architecture – Predictable Reliability

2 PREDICTABLE RELIABILITY

Suppose a transport protocol that allows an application to formulate QoS requirements in terms of delay, reliability and bandwidth. Since these parameters are not mutually independent on an unmanaged Internet infrastructure, the protocol has two main features: It establishes a managed pipe through the unmanaged network that fulfills the application's requirements with high probability, while adapting bandwidth allocation and reliability in a self-controlled fashion. In addition, it instantly monitors the path's quality and notifies the application if the service cannot be delivered under the formulated requirements.

Predictably reliable transport assigns a *delivery time budget* to every single packet. The budget is specified between the time the packet enters the protocol stack at the sender and the time it becomes available to the receiver application. The protocol must finalize the repair operations for each single packet within this time budget. Therefore, the predictably reliable protocol applies packet-level HEC that allows to send coded packets proactively as well as upon receiver request (reactively). Under a time constraint, proactive redundancy enables the control over the residual packet loss rate. Therefore, we formulate the parametrization of the predictably reliable protocol as an optimization problem: Spend the application's time budget by sending an optimal ratio of proactive and reactive redundancy such that the desired level of reliability is achieved with minimum redundancy information (coding overhead). We refer to this optimization problem as *predictable reliability* [4].

2.1 Architecture

The overall architecture of predictable reliability is presented in Figure 1. The basis of the predictably reliable transport is a transport protocol with adjustable error control. A network monitoring unit is connected to the protocol in order to record observations from the network path. Based on a suitable block-erasure model, a short-term prediction of the network state is possible. Further, a stochastic model of the protocol's error control evaluates the performance of a chosen protocol configuration under the predicted network state. The joint modeling of network state and protocol performance enables the reliability control unit to optimize protocol parameters under the application's QoS constraints. The optimized protocol parameters are instantly applied to the transport protocol.

Transport Protocol: The transport protocol is the executive unit of the predictably reliable architecture. It implements a scalable (multicast) error control scheme that is optimized under time constraints. The error control is highly dynamic and allows for fine-grained parameter adjustments. In order to support the adaptive error control, the protocol implements receiver feedback within a configurable period and upon packet loss. a detailed *timing model* describes the impact of time-related system parameters, such as the packet interval of the real-time source and the network's round trip time. It models the time that is allocated for the error control scheme and incorporates communication delays between the hosts. The transport protocol observes packet loss and delay on the end-to-end network path.

Block-erasure Model: The packet loss process is potentially observed with a specific burstiness depending on the error source. For instance, queueing losses tend to affect several packets in sequence, whereas a noisy wireless transmission is more likely to produce distributed packet losses. In order to express the burstiness in the packet loss process, a suitable stochastic block- erasure model with memory is being fitted to the pattern of packet losses. The block- erasure model is being updated via statistical evaluation and signal processing on the measurement samples from the protocol's packet loss observations on the network path. The model estimates the network's packet loss probability. The accuracy of the model is expressed as a function of the number of available measurement samples.

Protocol Performance Model: An essential part of the architecture is the protocol performance model that stochastically formulates the error correction performance of the protocol. Given a set of protocol parameters and the current network state, the model simulates the protocol's residual packet loss rate and the required coding overhead. Therefore, the module splits into three components, each contributing a specific performance equation. The block-error distribution provides a probability distribution of the erasure length that is obtained from the block-erasure model. Under consideration of this distribution, the *reliability model* predicts the level of reliability obtained by a given set of coding parameters for the protocol's error control. The model simulates the impact of packet loss on coded source packets. In addition, a *feedback model* determines the effect of missing loss notifications due to packet loss in the return path as well as the result of timer-based feedback suppression in the multicast. The residual packet loss rate of the protocol is formulated as a joint result of both models. The *efficiency model* determines the protocol overhead of a chosen parameter set based on the simulated network state. This includes the redundancy added by the coding scheme as well as the protocol's header overhead. The efficiency model formulates the expected bandwidth requirement of the protocol for a given source rate.

Reliability Control: The reliability control module adaptively configures the protocol's error control and per- forms equation-based congestion control. The module formulates predictable reliability as an optimization problem. It receives the delay and reliability constraints from the application and maximizes the protocol goodput under those constraints. In order to optimize the protocol parameters, the module applies the protocol performance model to the simulated network state represented by the timing model and the block-erasure model and obtains the predicted level of reliability. Based on this information it adapts the protocol parameters dynamically to temporally variable network conditions. The optimization problem is solved by a fast search algorithm that benefits from explicit knowledge about the parameter search space. Adaptation of the protocol parameters is performed periodically within a specific interval.

2.2 Protocol Implementation

The Predictably Reliable Real-time Transport (PRRT) protocol implements the above architecture on transport layer. It is based on a packet-level, adaptive HEC method, which achieves a near-optimal error correction performance on bidirectional packet-erasure channels. With this method, traditional FEC and ARQ error controls are jointly applied, each contributing their individual advantages. Whereas the use of a proactive FEC results in an error control performance independent of the path RTT, the use of ARQ allows to send repair packets upon explicit packet loss notifications. The scheme can be understood as a variant of Type-II Hybrid-ARQ coding [12], which has been found to be optimal in case the capacity of a packet erasure channel is unknown or dynamic.

The protocol first applies a systematic block-erasure code to a block of k source packets [9]. The resulting set of $n-k$ parity packets is divided into subsets corresponding to at most N_C repair cycles. Let $N_P = (N_P[0], N_P[1], ..., N_P[N_C])$ represent the number of parity packets scheduled for transmission in each cycle. The protocol's coding scheme is thus fully defined with two parameters: the FEC block-length k and the parity schedule N_P. Initially, only the k source packets are transmitted, optionally followed by a small number of $N_P[0]$ parity packets sent proactively. The remaining $n - k - N_P[0]$ parity packets are sent reactively upon receiver feedback according to N_P. This strategy is commonly known under the term *incremental redundancy*.

3 PERFORMANCE EVALUATION

3.1 Experimental Setup

In order to evaluate PRRT's performance under congestion loss, it is sent through a network bottleneck along with a different number of TCP sessions. The experimental setup comprises a dumbbell topology with a $50\,Mbps$ bottleneck bandwidth emulated via a Dummynet[2] bridge. The base RTT of the Dummynet bridge is set to $50\,ms$, $100\,ms$ and $150\,ms$ during different tests. The emulator performs drop-tail queueing, whereas the buffer size is set to the bandwidth-delay product of the emulated link. In different experiments, PRRT streams of $5\,Mbps$, $10\,Mbps$ and $20\,Mbps$ source rate are sent through the bottleneck along with several TCP-Cubic [5] sessions established via Iperf[3]. During several experiments, PRRT competes with up to 9 TCP streams while their number and PRRT's source rate are adjusted to achieve an equal share of the bandwidth for all flows. All experiments are terminated after transmitting 10^7 PRRT packets with a payload size of $1316\,byte$. For comparison the experiments are repeated while replacing PRRT with PR-SCTP as a representative for a partially reliable transport protocol.

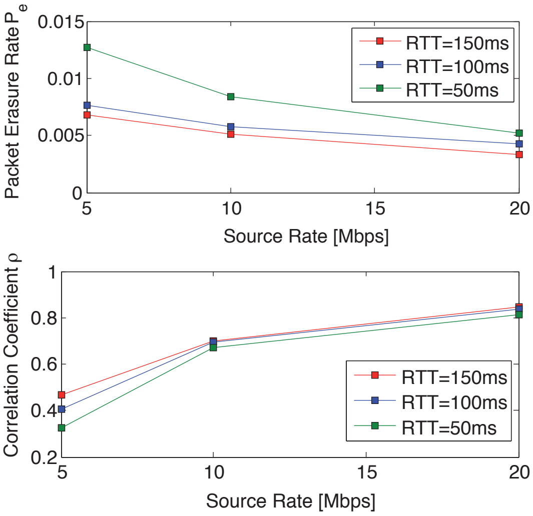


Figure 2: Measured packet erasure rate P_e and correlation coefficient ρ obtained by fitting a simplified GE model to the average erasure length.

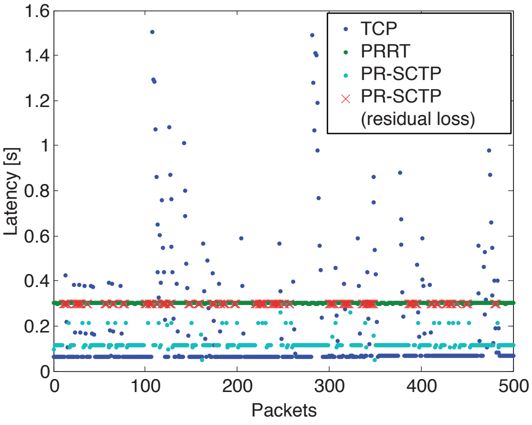


Figure 3: Comparison of the end-to-end latency and correction performance of TCP, PR-SCTP and PRRT. PR-SCTP and PRRT have been configured with a delay constraint of $300\,ms$ under an RTT of $50\,ms$.

[2] http://info.iet.unipi.it/~luigi/dummynet/
[3] http://sourceforge.net/projects/iperf/

3.2 Observed Network State

PRRT measures the network state in terms of the packet loss rate P_e, the burstiness coefficient of the loss process ρ as well as the network path's round trip time RTT. Those parameters characterize the impact of the TCP-induced queue saturation on the continuous media stream and they essentially determine PRRT's adaptive parametrization. However, during the experiments the network state is dynamic such that some characteristic long-term trends are discussed in the following. The observations are represented via their average values obtained throughout the entire experiment.

The erasure probability and the burstiness of the packet loss process observed by the PRRT stream are subject to two parameters (Figure 2): PRRT's source rate as well as the network's RTT. The erasure probability implicitly reflects the frequency of the congestion events, which grows for lower RTTs. The burstiness of the packet loss refers to the impact of a congestion event on the real-time flow, which mainly increases together with the source rate.

We express burstiness or temporal correlation in the packet loss by instantiating the protocol's block-erasure model as a simplified Gilbert-Elliott model, i. e. a two-state, non-hidden Markov chain that has been proposed to express the burstiness of measured packet loss on Internet paths [2], [18]. The model relies on the assumption that periods of packet loss and periods of successful reception have geometrically distributed length. We fit the model via maximum likelihood estimation to the observed packet erasure pattern in order to obtain the model parameters, consisting in packet erasure rate P_e and correlation coefficient ρ. Based on the model parameters we obtain the network's block error distribution, which is applied to optimize protocol parameters based on the stochastic protocol performance model.

TCP sessions that are sharing the same bottleneck tend to acquire *global synchronization* [16], which results in longer periods of network saturation and significant underutilization afterwards, due to their collective window reduction. TCP synchronization is known to increase for a lower number of parallel TCP sessions. For higher media source rates this leads to a larger number of packets being blocked consecutively at the saturated queue.

The GE model compensates for the large average erasure length with an overestimated correlation coefficient ρ. A model of higher order or a queueing model would be more appropriate to express the temporal correlation of queueing losses. However, the overestimation of the correlation coefficient results in the selection of conservative protocol parameters by PRRT's reliability control unit. Therefore, the significant increase in complexity introduced by a more sophisticated network model is not justified for the real-time application.

3.3 QoS Allocation via PRRT

Under the experimental scenarios depicted in Table 1, PRRT's reliability control dynamically finds the optimal block length k and a corresponding repair packet schedule N_P subject to a desired residual erasure rate P_T and a delay constraint D_T [4]. This results in a dynamic code rate obtained by the parameter adaptation itself as well as the incremental sending of repair packets. Pure FEC represents the borderline case of sending all parity proactively, purely reactive repair with a block length of $k = 1$ corresponds to the functionality of an ARQ-based, partially reliable protocol.

Table 1: Residual packet loss rate P_{res} and overhead of PRRT and PR-SCTP under different round trip time RTT, source rate SR, delay constraint D_T and the corresponding ITU-T 1541 QoS classes.

RTT $[ms]$	SR $[Mbps]$	D_T $[ms]$	P_T	P_{res} $[\times 10^{-2}]$	Coding Overhead	QoS Class
PRRT						
50	5	100	10^{-5}	0.0009	0.562	6
	10			0.0001	0.716	
	20			0.0000	0.640	
100	5	150		0.0012	0.620	7
	10			0.0000	0.719	
	20			0.0000	0.705	
150	5	200		0.0012	0.697	7
	10			0.0000	0.721	
	20			0.0007	0.713	
50	5	300	10^{-5}	0.0004	0.025	7
	10			0.0007	0.022	
	20			0.0010	0.015	
100	5	500		0.0000	0.016	4
	10			0.0011	0.015	
	20			0.0010	0.012	
150	5	700		0.0011	0.015	4
	10			0.0000	0.013	
	20			0.0012	0.010	
PR-SCTP						
50	5	300	n. a.	16.1184	0.001	n. a.
	10			2.8925	0.001	
	20			1.3458	0.001	
100	5	500		13.7375	0.002	
	10			6.2276	0.001	
	20			1.2907	0.001	
150	5	700		6.1445	0.003	
	10			2.5706	0.002	
	20			1.6592	0.002	

In order to clarify PRRT's operational mode, we compare its behavior with TCP and PR-SCTP as representatives for totally and partially reliable protocols (Figure 3). While TCP translates packet loss into unbounded delivery delay, PR-SCTP leaves a large amount of residual packet losses because the purely reactive error repair cannot control the residual loss rate under a strict delay constraint. PRRT, however, maintains a constant delivery delay and adjusts the proactive sending of repair packets in order to meet the reliability requirement defined by the application. As evident from Figure 3, each packet experiences constant delivery delay under PRRT since, by allocating a sufficiently large end-to-end delay budget, the protocol makes the underlying network dynamics transparent to the application.

Because of the fact that the continuous media stream cannot immediately back off the sending rate at the TCP-induced congestion event, packet erasures tend to appear in longer sequences due to the temporally correlated queueing losses. Reactive repair packets must not be sent immediately after sensing the packet loss as they would contribute to the queue saturation. Therefore, a small multiple of the RTT should be available for PRRT's time budget D_T in order for the search algorithm to spread the repair packets over several cycles (Table 1). Alternatively, if the RTT is large compared to D_T, the protocol can operate as an adaptive FEC at the price of reasonably larger coding overhead. Whereas an FEC configuration requires 60% to 70% coding overhead to satisfy the reliability requirement of $P_T = 10^{-5}$, hybrid configurations add less than 3% overhead within all considered scenarios.

As a metric for PRRT's QoS allocation we compare the results with the requirements of specific ITU-T Y.1541 QoS classes. Class 6 and class 7 formulate the tightest reliability constraint with a residual packet loss rate of 10^{-5} under an end-to-end delay of $100\,ms$ and $400\,ms$, respectively. Those requirements approach the QoS constraints of IP-based live media broadcast. As evident from Table 1, class 6 can only be satisfied by an adaptive FEC configuration under a low RTT, while the requirements of class 7 are met under a larger RTT. For RTT's around $50\,ms$ even the more efficient hybrid (proactive and reactive) configuration can meet the constraints of class 7 very well. For RTTs of $100\,ms$ and more, however, the delay constraint of $400\,ms$ is too tight for the hybrid error control such that these configurations fulfill just class 4 with a delay constraint of $1\,s$.

Finally, we compare the results of PRRT with the correction performance of PR-SCTP under the same time constraints. Unfortunately, it is hardly possible to implement a fair comparison between both protocols since PR-SCTP as well as TCP significantly suffer under the combination of congestion control via Additive Increase, Multiplicative Decrease (AIMD) and ARQ-only error control. As a result of the incompatibility of both schemes with real-time media transport, a large amount of packets is rejected under the specified delay constraints. In the evaluated scenario, PR-SCTP achieves therefore residual packet loss rates between $1\,\%$ and $16\,\%$ during the competition with TCP at the network bottleneck (Table 1).

4 CONCLUSION

In this paper we present a self-managing protocol architecture that fulfills the QoS requirements of high data rate, real-time media streaming applications on unmanaged Internet paths. Such applications suffer under the impact of TCP-induced congestion losses unless they are delivered over infrastructures with resource reservation. Our protocol implements *predictably reliable error control*, which allows the application to formulate individual QoS constraints to the transport layer.

We discussed general characteristics of TCP-induced packet loss as it is observed by continuous media streams. The protocol architecture optimizes the error control based on the measured network state. As a result, any PRRT-based application experiences constant end-to-end delivery delay and controlled residual packet loss rate under optimized coding overhead. Our experiments show that the protocol can satisfy the demanding constraints of the ITU-T Y.1541 QoS classes.

REFERENCES

[1] C. Aurrecoechea, A. T. Campbell, and L. Hauw. A survey of QoS architectures. *Multimedia Systems*, 6:138–151, May 1998.

[2] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive fec-based error control for internet telephony. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1453 –1460 vol.3, mar 1999.

[3] E. Exposito, M. Gineste, L. Dairaine, and C. Chassot. Building self-optimized communication systems based on applicative cross-layer information. *Computer Standards and Interfaces*, 31:354–361, February 2009.

[4] M. Gorius, Y. Shuai, and T. Herfet. Predictably reliable media transport over wireless home networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC) 2012*, pages 62–67, January 2012.

[5] S. Ha, I. Rhee, and L. Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review*, 42:64–74, July 2008.

[6] S. A. Karim and P. Hovell. Everything over IP - an overview of the strategic change in voice and data networks. *BT Technology Journal*, 17(2):24–30, April 1999.

[7] Y.-C. Lai and C.-N. Lai. DCCP partial reliability extension with sequence number compensation. *Comput. Netw.*, 52:3085–3100, November 2008.

[8] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey. Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF). RFC 4585 (Proposed Standard), July 2006. Updated by RFC 5506.

[9] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *SIGCOMM Computer Communication Review*, 27:24–36, April 1997.

[10] J. Rosenberg and H. Schulzrinne. An RTP Payload Format for Generic Forward Error Correction. RFC 2733 (Proposed Standard), Dec. 1999. Obsoleted by RFC 5109.

[11] L. Roychoudhuri and E. Al-Shaer. Autonomic qos optimization of real-time internet audio using loss prediction and stochastic control. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pages 73 –80, april 2008.

[12] D. Rubenstein, J. Kurose, and D. Towsley. A study of proactive hybrid FEC/ARQ and scalable feedback techniques for reliable, real-time multicast. *International Journal for the Computer and Telecommunications Industry*, 24:563–574, 2001.

[13] H. Seferoglu, A. Markopoulou, U. Kozat, M. Civanlar, and J. Kempf. Dynamic fec algorithms for tfrc flows. *Multimedia, IEEE Transactions on*, 12(8):869 –885, dec. 2010.

[14] N. Seitz. ITU-T QoS standards for IP-based networks. *IEEE Communications Magazine*, 41(6):82–89, June 2003.

[15] D. S. Turaga and T. Chen. Hierarchical modeling of variable bit rate video sources. In *Proceedings of the 11th International Workshop on Packet Video (PV) 2001*, page 2001, April 2001.

[16] K. Vlachos. Burstification effect on the TCP synchronization and congestion window mechanism. In *Fourth International Conference on Broadband Communications, Networks and Systems, (BROADNETS) 2007*, pages 24–28, September 2007.

[17] C. Xu, E. Fallon, Y. Qiao, G.-M. Muntean, X. Li, and A. Hanley. Analysis of real-time multimedia transmission over PR-SCTP with failover detection delay and reliability level differential. *Communication Software and Networks, International Conference on*, 2009.

[18] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modelling of the temporal dependence in packet loss. In *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) 1999*, volume 1, pages 345–352, March 1999.

[19] M. Yuksel, K. K. Ramakrishnan, R. Doverspike, R. Sinha, G. Li, K. Oikonomou, and D. Wang. Cross-layer techniques for failure restoration of IP multicast with applications to IPTV. In *Proceedings of the 2nd international conference on Communication Systems and Networks (COMSNETS) 2010*, pages 223–232, January 2010.

Controlling a Software-Defined Network via Distributed Controllers

Volkan Yazıcı[1], M. Oğuz Sunay[1], Ali Ö. Ercan[1]

[1]Özyeğin University, Istanbul, Turkey

E-mail: [1]{volkan.yazıcı, oguz.sunay, ali.ercan}@ozyegin.edu.tr

Abstract: **In this paper, we propose a distributed OpenFlow controller and an associated coordination framework that achieves scalability and reliability even under heavy data center loads. The proposed framework, which is designed to work with all existing OpenFlow controllers with minimal or no required changes, provides support for dynamic addition and removal of controllers to the cluster without any interruption to the network operation. We demonstrate performance results of the proposed framework implemented over an experimental testbed that uses controllers running Beacon.**

Keywords: software-defined networking, reliability, scalability, high-availability

1 INTRODUCTION

Today's networks have become exceedingly complex, because they implement an ever increasing number of distributed protocols standardized by IETF and the individual packet routing/switching components within these networks use closed and proprietary programs that take these protocols into account. In this environment it is too difficult, if not impossible, for network operators, third parties, including researchers, and even vendors to innovate [1]. To address this problem, in 2011, the Open Networking Foundation (ONF) has been formed with the aim of promoting a new networking paradigm, called *Software-Defined Networking* (SDN). The fundamental idea behind SDN is a network architecture where the control plane is decoupled from the data plane. This abstraction opens up the possibility for a programmable network, where the administrators can customize the network to fit their needs.

The *OpenFlow* communication protocol is one of the enablers of the SDN paradigm [1]. It provides a common set of instructions for the control plane to interact with the data plane realized via packet-forwarding hardware. The control-plane, which is commonly referred to as the controller or the network operating system in SDN, resides on a dedicated server and commands the packet-forwarding hardware through the OpenFlow protocol as illustrated in Figure 1. This abstraction enables the controller to easily enforce flow-based sophisticated traffic management policies (routing, QoS, VLAN tagging, etc.) in the network.

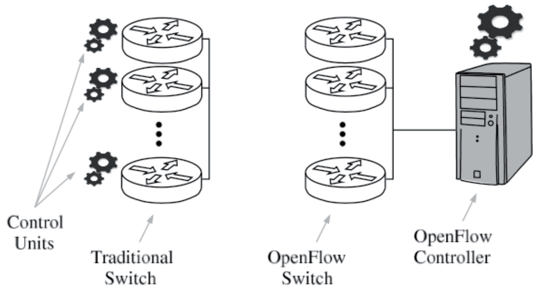


Figure 1: Comparison of traditional and SDN networks

Recent studies conducted on the networks of many real-world data centers showed that such networks necessitate the handling of about 150 million flows per second [2]. On the other hand, today's OpenFlow controllers are known handle at most 6 million flows per second on a high end dedicated server with 4 cores (see Section III.) Therefore, implementation of SDN for one of such data center networks requires a controller running either on an appropriate mainframe computer with sufficiently many cores or a server cluster where each server is composed of limited cores.

Implementation of the controller on a cluster offers a number of benefits. First, this platform is scalable, as an increasing load on the controller is easily handled by introducing new servers to the cluster. Second, the cluster offers more reliability than an implementation on a single mainframe, which presents a single point of failure. For this reason, we propose a cluster based distributed OpenFlow controller framework as illustrated in Figure 2 in this paper.

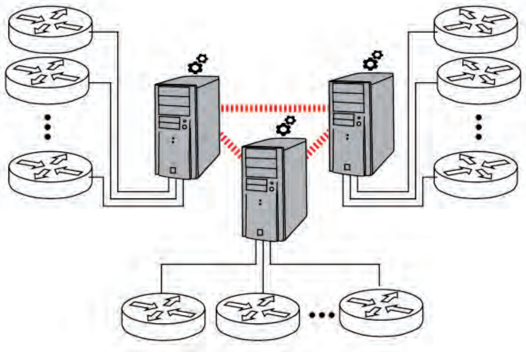


Figure 2: Distributed OpenFlow controller architecture

Corresponding author: Volkan Yazıcı, Özyeğin University, Çekmeköy, İstanbul, Turkey, +90(216)564-9000, volkan.yazici@ozu.edu.tr

It is well-known that coordination is an important pillar of distributed systems. For the proposed cluster based distributed controller architecture, a well-designed coordination framework is necessary to allow for load balancing between the distributed controllers and for replacement of failed controllers by active ones so that scalability and reliability is sustained with zero network down-time.

A number of papers have recently appeared in the literature on distributed implementations of OpenFlow controllers. In [3], the authors present a distributed NOX-based controllers interwork through extended GMPLS protocols. [4] proposes to deploy multiple instances of the same NOX controller on a set of distributed nodes. [5] proposes to use a distributed set of autonomous controllers, but does not provide means for switch migration amongst them. In [6], a new platform is introduced, over which a distributed network control plane may be implemented.

In this paper, we propose a distributed controller framework and an associated coordination framework that achieves scalability and reliability even under heavy data center loads. The proposed architecture provides two novel key features that are not present in previous related work. First, the proposed framework provides support for dynamic addition and removal of controllers to the cluster without any interruption to the network operation. Second, the proposed distributed controller and associated coordination framework is designed to work with all existing OpenFlow controllers with minimal or no required changes.

The remainder of this paper is organized as follows. In Section II, we introduce the coordination framework for the cluster based controller implementation. We present experimental results of the proposed framework in Section III and conclude in Section IV.

2 PROPOSED FRAMEWORK

In this paper, we consider an OpenFlow controller implementation that is cluster based as illustrated in Figure 2. In the implementation, multiple controllers are realized in a cluster, each on a distinct server. The proposed framework may use any of the operating systems from the literature to implement the controllers, with simple modifications, if necessary.

In the proposed framework, controllers in the cluster communicate with each other using the JGroups membership notifications and messaging infrastructure [7]. JGroups is a mature, robust and flexible group communication library used in many data centers for various mission critical applications. In this setting, the controllers elect a master node amongst them which conducts and maintains the global controller-switch mapping in the network. The master node is periodically monitored by all other nodes, and if it is found to be inaccessible, it is immediately replaced by one of the other nodes. Thus, the proposed framework does not expose a single-point-of-failure.

The proposed controller architecture interfaces the switches in the network as well as the applications running above posing as a single, centralized controller. In other words, the switches and the applications are unaware of the switch assignments and re-assignments in the network to the individual controllers in the cluster. This provides a seamless, compatible operation with legacy OpenFlow switches and applications. However, if a switch and/or application is aware of the underlying distributed architecture, the necessary API to utilize its features is also provided.

We now discuss how the master controller is selected, how the switches in the network are mapped to the controllers, how load balancing between the controllers is achieved, and what happens when one of the controllers becomes inaccessible in the proposed framework. Next, we present a network model to simplify the discussion.

2.1 Notation

Let $\mathcal{C}=\{c_1,\ldots,c_n\}$denote the controllers in the cluster and $\mathcal{S}=\{s_1,\ldots,s_m\}$ denote the switches in the network. The network is represented by an undirected graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ denotes the vertices and $\mathcal{E}$ denotes the edges. Vertices are composed of controllers and switches, i.e, $\mathcal{V}=\mathcal{C}\cup\mathcal{S}$ and edges are composed of two-tuples representing the connections between vertices, that is, $\mathcal{E} \subseteq \{(v_k,v_l)|v_k,v_l \in \mathcal{V}\}$, $k,l=1,2,\ldots,n+m$. k and l are omitted in the text from now on for brevity. The switch-controller mapping is given by the set $\mathcal{M}$, which is a subset of $\mathcal{E}$, connecting vertices between $\mathcal{C}$ and $\mathcal{S}$; $\mathcal{M} \subseteq\{(c_k,s_l) \in \mathcal{E} \mid c_k \in \mathcal{C}, s_l \in \mathcal{S}\}$. In $\mathcal{M}$, each switch is constrained to be connected to a single controller. That is, if $s_l \in S$ and $c_p,c_q \in C$, then $(c_p,s_l) \in \mathcal{M}$ and $(c_q,s_l) \in \mathcal{M}$ if and only if p=q.

In an OpenFlow network, controllers and switches are connected via an IP network. In a given network $\mathcal{G}=(\mathcal{V},\mathcal{E})$, an IP network is represented by a graph $\mathcal{N}=(\mathcal{V}_{\mathcal{N}}, \mathcal{E}_{\mathcal{N}})$, where vertices and edges are given by $\mathcal{V}_{\mathcal{N}} \in \mathcal{V}$ and $\mathcal{E}_{\mathcal{N}}=\{(v_k,v_l)|v_k,v_l \in \mathcal{V}_{\mathcal{N}}\}$, respectively. Here, each edge $(v_k,v_l) \in \mathcal{E}_{\mathcal{N}}$ is constrained to have a path from v_k to v_l in $\mathcal{G}$. That is, $(v_k,v_l) \in \mathcal{E}_{\mathcal{N}}$ for $v_k,v_l \in \mathcal{V}_{\mathcal{N}}$ if and only if $\{(v_k, v_{i1}),(v_{i2}, v_{i3}),\ldots,(v_{ir},v_l)\} \subseteq \mathcal{E}$ and $\exists\, v_{i1},v_{i2},\ldots,v_{ir} \in \mathcal{V}$. In an IP network $\mathcal{N}$, each vertex vi $\in \mathcal{V}_{\mathcal{N}}$ is assigned a set of IP addresses denoted by $IP_{\mathcal{N}}\{v_i\}$ (assigning multiple IP addresses to a single interface is possible through IP aliasing). Assigned IP addresses in $\mathcal{N}$ are chosen to be pairwise disjoint, that is, $IP_{\mathcal{N}}\{v_k\}\cap IP_{\mathcal{N}}\{v_l\}= \varnothing$ for $k \neq l$ and $\forall\, v_k,v_l \in \mathcal{V}_{\mathcal{N}}$.

2.2 Local IP Networks

In the proposed framework, the network of controllers and switches is divided into two distinct IP networks: an IP network $\mathcal{A}=(\mathcal{V}_{\mathcal{A}},\mathcal{E}_{\mathcal{A}})$ for the controller-controller communication (i.e., $\mathcal{V}_{\mathcal{A}}=\mathcal{C}$) and an IP network $\mathcal{B}=(\mathcal{V}_{\mathcal{B}},\mathcal{E}_{\mathcal{B}})$ for the controller-switch communication (i.e., $\mathcal{V}_{\mathcal{B}}=\mathcal{C} \cup \mathcal{S}$).

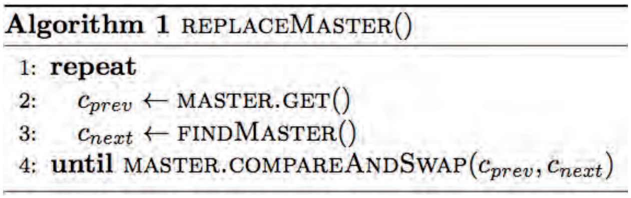
IP network $\mathcal{A}$: Here, each controller $c_i \in \mathcal{V}_{\mathcal{A}}$ is statically assigned a unique IP address C_i, i.e., $IP_{\mathcal{A}}\{c_i\}=\{C_i\}$.

IP network $\mathcal{B}$: Here, a unique IP address S_i is assigned to each switch $s_i \in S$, i.e., $IP_{\mathcal{B}}\{s_i\}=\{S_i\}$. In this network, a pool of IP addresses, P_i describes the controllers. There are as many P_i's as there are switches in this pool. At a given time, each switch, s_i is statically configured to connect to the controller with IP address P_i. The master controller decides on how the switches are mapped to the controllers by partitioning the IP address pool P_i amongst all controllers, including itself. Once a switch s_i is mapped to controller c_m, the IP address P_i is dynamically assigned to that controller by IP aliasing. Thus at any given time, if $P_l \in IP_{\mathcal{B}}\{c_k\}$, controller $c_k \in \mathcal{C}$ is said to be controlling the switch $s_l \in \mathcal{S}$. If, for some reason, the controller-switch mapping changes at some stage, the IP alias P_i is moved to the new controller that starts to control the switch. To avoid multiple controllers trying to control the same switch or a switch not being controlled by any of the controllers, IP addresses are assigned to be mutually exhaustive and pairwise disjoint, i.e., $\{P_i\}=\cup_{ck} IP_{\mathcal{B}}\{c_k\}$ and $IP_{\mathcal{B}}\{c_k\} \cap IP_{\mathcal{B}}\{c_l\}= \varnothing$ for $c_k,c_l \in \mathcal{C}$ and $k \neq l$.

Figure 3a shows a sample OpenFlow network composed of 2 controllers (c_1, c_2) and 5 switches ($s_1,\ldots,s_5$). Here, the initial mapping between the controllers and the switches is given by $\mathcal{M}=\{(c_1,s_1), (c_1,s_2), (c_2,s_3), (c_2,s_4), (c_2,s_5)\}$. Switches s_1 and s_2 are controlled by controller c_1, i.e., $IP_{\mathcal{B}}\{c_1\} = \{P_1,P_2\}$, $IP_B\{s_1\}=\{S_1\}$ and $IP_{\mathcal{B}}\{s_2\}=\{S_2\}$. Similarly, switches s_3,s_4,s_5\$ are controlled by controller c_2 i.e., $IP_{\mathcal{B}}\{c_2\} = \{P_3,P_4,P_5\}$, $IP_{\mathcal{B}}\{s3\}=\{S_3\}$, $IP_{\mathcal{B}}\{s_4\}=\{S_4\}$ and $IP_{\mathcal{B}}\{s_5\}=\{S_5\}$.

2.3 Master Controller Selection

In the proposed framework, each controller in the cluster is equipped with the same algorithm that generates and updates the network mapping. In this setting, a master node is responsible for realizing the controller-switch mapping updates. The master controller is determined using a distributed atomic integer primitive provided by JGroups. This scheme is outlined in Algorithm 1.



Algorithm 1 REPLACEMASTER()

```
1: repeat
2:     $c_{prev}$ ← MASTER.GET()
3:     $c_{next}$ ← FINDMASTER()
4: until MASTER.COMPAREANDSWAP($c_{prev}$, $c_{next}$)
```

There are various approaches to the selection of a master in a cluster environment [8]. These studies generally employ a cost function with a set of user provided constraints over the measurements collected throughout the system. Then, the most effective configuration is selected from all available candidates. In the proposed framework, the processing cost imposed by the cluster is almost negligible on a master controller and is rarely encountered, e.g., while balancing loads or in case of a server failure. In the framework presented herein, when FINDMASTER() is invoked, it is set to return the controller with the smallest system load. To avoid frequent master changes, this algorithm is invoked only when the current master becomes unsuitable based on some user-defined criteria.

In the proposed framework, initially, the controller that first completes the execution of algorithm 1 is established as the master. In this algorithm, when a controller decides that the master node needs to be changed, it finds a suitable node for replacement (line 2 and 3). Then, using the atomic integer primitive COMPAREANDSWAP(), this algorithm repeatedly tries to replace the master node, until it succeeds to do so (line 4) or number of repetitions exceed a certain threshold. Once the master controller is replaced, JGroups ensures that it gets atomically propagated throughout the cluster. Each cluster member checks if it is the master node, if so, it starts executing the regular switch-controller mapping checks and decisions.

Therefore, to avoid making master node a single point of failure, the rest of the controllers in the cluster regularly check the working status of the master node and, in case of a failure, attempt to replace it.

2.4 Mapping

In the proposed framework, the master controller dynamically partitions the IP addresses Pi amongst all controllers. A controller c_k controls a switch s_l by adding IP address P_l into $IP_{\mathcal{B}}\{c_k\}$, i.e., its list of IP aliases in network $\mathcal{B}$. This necessitates generation and updating of the mapping $\mathcal{M}$ regularly. The mapping information is stored locally by all controllers and this database is automatically synchronized amongst all of them via JGroups with every local update.

The mapping $\mathcal{M}$ is generated and updated by the master controller using the GENERATEMAPPING() function. This function takes the two IP networks $\mathcal{A}$, $\mathcal{B}$, current mapping $\mathcal{M}$ (empty set if no current mapping exists) and a set of system statistics parameters, STATS as input parameters. STATS is composed of statistics such as link traffic, controller loads, etc., which are collected and provided by the controller architecture. Additionally, a network administrator might decide to route certain flows over certain machines for security concerns, custom applications might require customized quality-of-service measures, etc. All such constraints need to be taken into account in the GENERATEMAPPING() function. The implementation of the GENERATEMAPPING() function highly depends on the work flow of the underlying network and is out of the scope of this work.

The result of the GENERATEMAPPING function is propagated throughout the controller cluster by the master controller. For this purpose, the master controller invokes the SETMAPPING($\mathcal{A}$,$\mathcal{B}$,$\mathcal{M}_{\text{old}}$,$\mathcal{M}_{\text{new}}$) algorithm, where $\mathcal{M}_{\text{old}}$ and $\mathcal{M}_{\text{new}}$ arguments denote the the old and the new mappings, respectively. In Algorithm 2, SETMAPPING() function is detailed. Here, for every switch the function determines its current (line 2) and next (line 3) controller, implied by the mappings $\mathcal{M}_{\text{old}}$ and $\mathcal{M}_{\text{new}}$, respectively. Next, the function decides on how to realize the operation of migrating a switch from c_{old} to c_{new} using COALESCE() (line 4). Here, if c_{old} is alive, it is selected as the first point of contact for the mapping update message. If not, c_{new} is selected as the first point of contact. Finally, in order to trigger the actual switch migration, the function makes a remote procedure call on the first point of contact, c_r to run the function MOVE() (line 5).

Algorithm 2 SETMAPPING($\mathcal{A}, \mathcal{B}, \mathcal{M}_{old}, \mathcal{M}_{new}$)

```
1: for $\ell \leftarrow 1 \ldots m$ do
2:     $c_{old} \leftarrow \exists c_k$ for $(s_\ell, c_k) \in \mathcal{M}_{old}$
3:     $c_{new} \leftarrow \exists c_k$ for $(s_\ell, c_k) \in \mathcal{M}_{new}$
4:     $r \leftarrow$ COALESCE($c_{old}, c_{new}$)
5:     $c_r$ ! MOVE($\mathcal{A}, \mathcal{B}, s_\ell, c_{old}, c_{new}$)
6: end for
```

Whenever a controller receives a remote call to run the MOVE() function, it is expected to either release a switch for some other controller to subsequently acquire it, or acquire an already released switch. This operation is detailed in Algorithm 3. Here, the function first inquires its own rank in the cluster (controller ID) (line 1) and determines if it is expected to release (line 2) or acquire (line 6) the switch. If it is invoked to release the switch, first it releases the control of the switch (line 3) and updates its list of IP addresses in network $\mathcal{B}$ (line 4). Then, it invokes the controller that will take control of the switch (c_{new}) (line 5) to run the MOVE() function. Otherwise, if MOVE() is invoked at the controller to acquire the switch, it first acquires the control of the switch (line 7) and updates its list of IP addresses in network B (line 8). Finally, the acquiring controller alerts the switch to reset its ARP cache (line 9).

Algorithm 3 MOVE($\mathcal{A}, \mathcal{B}, s_\ell, c_{old}, c_{new}$)

```
1: $r \leftarrow$ CURRENTRANK()
2: if $c_r = c_{old}$ then
3:     RELEASE($P_\ell$)
4:     $\mathrm{IP}_\mathcal{B}(c_r) \leftarrow \mathrm{IP}_\mathcal{B}(c_r) \setminus P_\ell$
5:     $c_{new}$ ! MOVE($s_\ell, c_{old}, c_{new}$)
6: else if $c_r = c_{new}$ then
7:     ACQUIRE($P_\ell$)
8:     $\mathrm{IP}_\mathcal{B}(c_r) \leftarrow \mathrm{IP}_\mathcal{B}(c_r) \cup P_\ell$
9:     ARPPING($s_\ell$)
10: end if
```

2.5 Operation

The master controller regularly observes the network statistics provided by STATS. If a load imbalance (induced by controller/switch addition, controller/switch failure, flow/traffic changes, \etc.) is detected, the master first triggers a GENERATEMAPPING() call. The resulting mapping is then updated for each controller via JGroups and is executed by the SETMAPPING() function. The operation of the proposed framework is examplified in Figures 3b and 3c. Here, assume that the controller c_2 is overloaded. In the new mapping, the switch s3 is reassigned to c_1 by GENERATEMAPPING() to alleviate the load of c_2. The IP address P_3 is first released by c_2 and subsequently acquired by c1 through executions of SETMAPPING() first on c_2 and then on c_3. In the new configuration, c_3 is no longer overloaded and the loads of c_1 and c_2 are more even. The transition is seamless for the switch s_3 since it is still connected to IP address P_3 for controller traffic.

When a new controller is added to the cluster, the rest of the cluster is instantly notified by the JGroups membership notification feature. Consequently, a new GETMAPPING()-SETMAPPING() cycle takes place so that some of the loads of the existing controllers is passed on to the newly added one to provide some desired level of load balancing. Similarly, if a controller fails or is selectively turned off by the system administrator due to low traffic, the rest of the cluster is immediately notified by JGroups and again, a new GETMAPPING()-SETMAPPING() cycle is executed. Newly computed mapping will replace the inaccessible controllers with the working ones. This is exemplified in Figure 3d. Here, c_2 dies. Therefore in the new mapping all switches are assigned to c_1.

2.6 Routing

The use of JGroups facilitates synchronism of the network map $\mathcal{M}$ across all controllers. Therefore, even though the switches in the network are distributed dynamically across the controllers, the framework allows for optimized end-to-end routing of flows over the entire network. The necessary messaging to facilitate the routing operation is beyond the scope of this paper.

3 RESULTS

We have conducted experiments to assess the performance of the distributed controller framework proposed herein. In the experiments, we implemented the framework using Beacon [9] Beacon has a successful track record in performance benchmarks (see Figure 4a) and enables the use of JVM libraries.

To run the experiment, we first enhanced Beacon by adding a new OSGi bundle, called `cluster`. Four controller machines - running Debian GNU/Linux 6.0.4 (i686) on a system with Dual-Core 2.80GHz CPU, 2GB

RAM, and JDK 1.6.0-30 - were configured with the enhanced Beacon. All machines were connected through an unmanaged gigabit switch and cluster communications were forced to run on physically separate NICs on each machine.

We first examine the cluster power up using the experimental setup. We observe that it takes approximately 12 sec. for the cluster OSGi bundle to start up initially when no other controllers are active. If there is at least one active controller in the cluster, the start up time is approximately 3 sec. The 9 sec. difference is due to discovery time during JGroups channel initialization. In the proposed framework, we measure that it takes in the order of under 50 milliseconds for members to get notified by the removal/arrival of a member in the cluster.

In the proposed framework, controllers acquire and release switches via maintenance of IP aliases. Using the experimental setup, we measure the time it takes for one of the controllers in the cluster to acquire and release IP aliases. The results are plotted in Figure 4b. We observe that the time it takes for these operations is slightly convex. However, we note that even with 254 simultaneous IP alias changes, the operation clocks under 5 sec.

Next, we measure the time it takes to migrate a group of switches from one controller to another. The results are plotted in Figure 4c. We note here that the switch migration includes an IP alias location change as well necessary communications over JGroups and kernel calls. In the figure, we observe that even 254 simultaneous switch migrations clock around 8 sec. approximately 4 of which is due to IP alias relocate operations. We note here that under normal network operations, the number of simultaneous switch migrations would be significantly less than 254.

Finally, the experimental setup with 4 controllers and 4 emulated switches is used to assess the performance increase with multiple controllers. Switch emulators are configured to run cbench [10] instances to stress the controller throughput for a period of ten seconds. Each stress test is repeated ten times. The first and the last runs are discarded to remove the effects of warm-up and cool-down times. The average number of controller responses per second per switch when one, two, three or four controllers are used are reported in Figure 4d. When multiple controllers are used, the switches are assigned to the controllers in a load balanced manner. As seen in the figure, the number of responses per second per switch increase super-linearly as more controllers are used. This is because, the possible combinations of interaction between the switches assigned to a controller increases quadratically with the number of switches. Thus, the overhead of coordinating switches per controller increases super-linearly with the number of switches assigned to it. This results in the super-linear increase in the performance as less switches are assigned to a controller.

4 CONCLUSIONS

This paper presents a distributed OpenFlow controller architecture and an associated coordination framework that achieves scalability and reliability even under heavy data center loads via the use of JGroups. The proposed architecture, which is designed to work with all existing OpenFlow controllers with minimal or no required changes, provides support for dynamic addition and removal of controllers to the cluster without any interruption to the network operation. Experimental results confirm that using the proposed framework, migration of switches amongst multiple controllers, addition/removal of controllers to the network is possible. The use of JGroups facilitates synchronism of the network map across all active controllers. As such, this framework allows for optimized end-to-end routing of flows across the network while achieving scalability and reliability of the network controllers.

References

[1] N. McKeown et al., “OpenFlow: Enabling Innovation in Campus Networks,” *SIGCOMM Comput. Commun. Rev*. vol. 38, pp. 69-74, March 2008.

[2] T. Benson, A. Akella and D.A. Maltz, “Network Traffic Characteristics of Data Centers in the Wild,” in *Proc. ACM IMC*, New York, NY, USA, 2010.

[3] R. Martinez et al., “OpenFlow-Based Hybrid Control Plane witin the CTTC ADRENALINE Testbed,” *in Proc. OFELIA Workshop*, Geneva, Switzerland, September 2011.

[4] A. Tootoonchian and Y. Ganjali, “HyperFlow: A Distributed Control Plane for OpenFlow Networks,” *in Proc. INM/WREN*, San Jose, CA, USA, April 2010.

[5] C. Macapuna, C. Rothenberg, and M. Magalhaes, “In-Packet Bloom Filter Based Data Center Networking with Distributed OpenFlow Controllers,” *in Proc. IEEE Globecom*, December 2010.

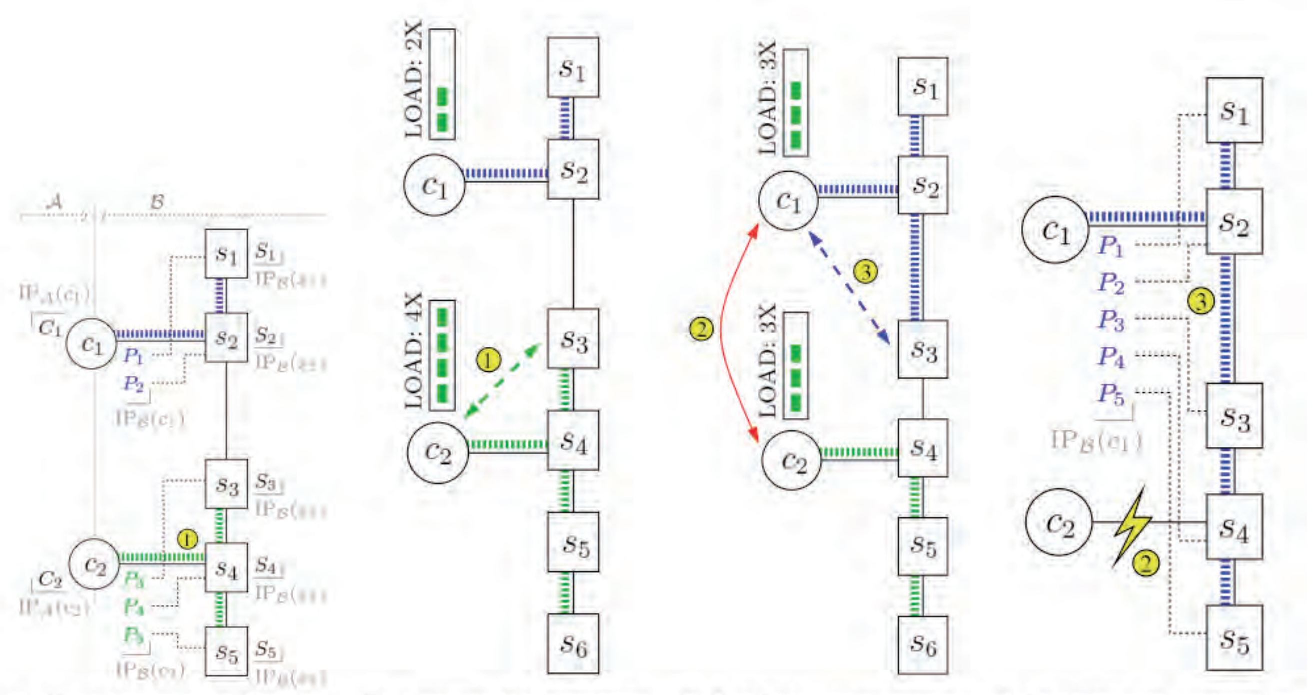
[6] T. Koponen et al., “Onix: A Distributed Control Platform for Large-Scale Production Networks,” *in Proc. USENIX OSDI*, Vancouver, BC, Canada, October 2010.

[7] B. Ban, “Design and Implementation of a Reliable Group Communication Toolkit for Kava,” *http://www.jgroups.org/papers/Coots.ps.gz.*

[8] G. Shau, F. Berman, and R. Wolski, “Master/Slave Computing on the Grid,” *in Proc. IEEE 9th Heterogeneous Computing Workshop*, Washington DC, USA, 2000.

[9] D. Erickson, “Beacon: A Fast, Cross-Platform, Modular, Java-Based OpenFlow Controller,” *http://beaconcontroller.net/,* 2011.

[10] C. Rostos et al. “OFLOPS: An Open Framework for OpenFlow Switch Evaluation,” *in Proc. PAM*, 2012.



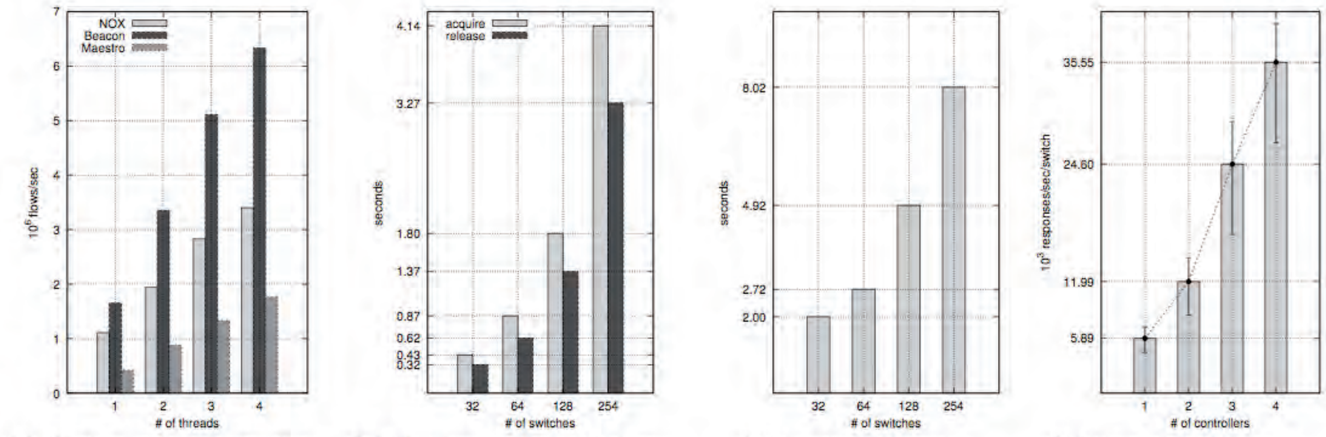
(a) IP addressing in the proposed framework

(b) Unequal loads amongst controllers

(c) Load-balancing amongst controllers

(d) Replacing an inaccessible controller

Figure 3: Controller coordination using the proposed framework



(a) Performance comparison of OpenFlow controllers[2].

(b) IP Alias Acquiry/Release

(c) Switch Migration

(d) Single vs. Multiple

Figure 4: Experimental results for the proposed distributed OpenFlow Controller Framework

OpenQoS: OpenFlow Controller Design and Test Network for Multimedia Delivery with Quality of Service

Hilmi E. Egilmez[1], S. Tahsin Dane[2], Burak Gorkemli[3], A. Murat Tekalp[4]

[1,2,4]Koc University, Istanbul, Turkey; [3]Argela Technologies, Istanbul, Turkey

{[1]hegilmez, [2]sdane, [4]mtekalp}@ku.edu.tr, [3]burak.gorkemli@argela.com.tr

Abstract: **OpenFlow is a Software Defined Networking (SDN) paradigm that decouples control and data forwarding layers of routing. In this paper, we propose OpenQoS, which is a novel OpenFlow controller design for multimedia delivery with end-to-end Quality of Service (QoS) support. Our approach is based on QoS routing where the routes of multimedia traffic are optimized dynamically to fulfill the required QoS. We also discuss possible application areas of our design and measure its performance over a real test network. Our experimental results show that we can guarantee seamless video delivery with little or no video artifacts experienced by the end-users. Unlike current QoS architectures, in OpenQoS the guaranteed service is handled without having adverse effects on other types of traffic in the network.**

Keywords: Software Defined Networking, OpenFlow, QoS Routing, Video Streaming, Multimedia Delivery

1 INTRODUCTION

The Internet design is based on end-to-end arguments [1] where the network support is minimized and the end hosts are responsible for most of the communication tasks. This design has two main advantages: First, it allows a unified best-effort service for any type of data at the networking layer where service definitions are made at the upper layers (hosts). Second, it reduces the overhead and the cost at the networking layer without losing reliability and robustness. This type of architecture fits perfectly to data transmission where the primary requirement is reliability. Yet, in multimedia transmission, timely delivery is preferred over reliability. Multimedia streaming applications have stringent delay requirements which cannot be guaranteed in the best-effort Internet. So, it is desirable that the network infrastructure supports some means to provide Quality of Service (QoS) for multimedia traffic. To this effect, the Internet Engineering Task Force (IETF) has explored several QoS architectures, but none has been truly successful and globally implemented. This is because QoS architectures such as IntServ [2] and Diffserv [3] are built on top of the current Internet's completely distributed hop-by-hop routing architecture, lacking a broader picture of overall network resources. Even though MPLS [4] provides a partial solution via its ultra-fast switching capability, it lacks real-time reconfigurability and adaptivity.

Software Defined Networking (SDN) [5] is a paradigm shift in network architecture where the network control is decoupled from forwarding and is directly programmable. This migration of control provides an abstraction of the underlying network for the applications residing on upper layers, enabling them to treat the network as a logical or virtual entity [6].

Among several attempts, OpenFlow is the first successful implementation [7] of SDN which has recently started being deployed throughout the world and has already attracted many commercial vendors [8]. As proposed in SDN, OpenFlow moves the network control to a central unit, called controller; while the forwarding function remains within the routers, called forwarders (see Fig.1). The OpenFlow controller is the brain of the network where packet forwarding decisions are made on per-flow basis and the network devices are configured accordingly via the OpenFlow protocol, which defines the communication between the controller and the underlying devices. OpenFlow provides complete network visibility, resource monitoring, and network virtualization, allowing sophisticated network management solutions. In this paper, we address a specific networking problem, providing QoS for multimedia delivery, and present an OpenFlow based solution for it.

This paper proposes OpenQoS, a novel controller design that enables QoS for multimedia delivery over OpenFlow networks. In order to support QoS, we group the incoming traffic as data flows and multimedia flows, where the multimedia flows are dynamically placed on QoS guaranteed routes and the data flows remain on their traditional shortest-path. Our approach is different from current QoS architectures, since we employ dynamic routing which is now possible with OpenQoS. OpenQoS is based on our prior work in [9]-[11], where the optimization framework and the results presented in these papers are exploited. Here, we also demonstrate the performance of OpenQoS on a real network with commercial OpenFlow-enabled switches.

The rest of the paper is organized as follows. Section 2 presents the OpenQoS design that supports QoS over OpenFlow networks. The OpenFlow test network, the OpenQoS implementation and possible application areas of OpenQoS are described in Section 3. Section 4 presents the results showing the performance of OpenQoS over the test network. Concluding remarks are given in Section 5.

Corresponding author: Hilmi Enes Egilmez, Koc University, Rumelifeneri Yolu 34450,Sariyer, Istanbul, Phone: +905555666653, E-mail: hegilmez@ku.edu.tr

Table 1: Comparison of QoS architectures

	Flow Support	*Type of Guarantee*	*Complexity*	*Effects on other flows*	*Mechanism*
IntServ	Individual flows	Hard & end-to-end	High	High (due to reservation)	Resource reservation
DiffServ	Multiple flows	Soft & hop-by-hop	Medium	Medium (priority queuing)	Scheduling, priority queuing
OpenQoS	Multiple flows	Soft & end-to-end	Low	Low (only based on routing)	Dynamic QoS routing

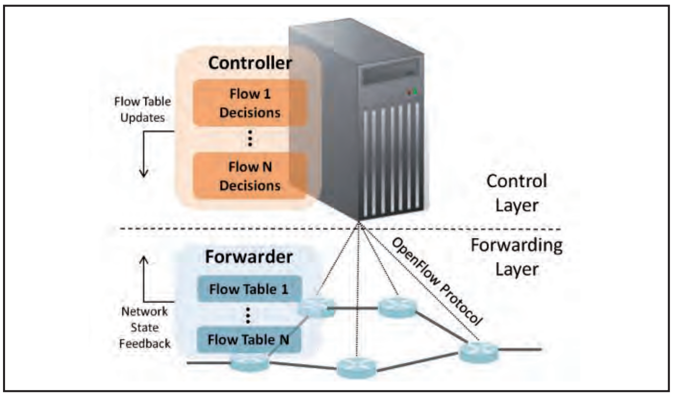


Figure 1: OpenFlow Architecture

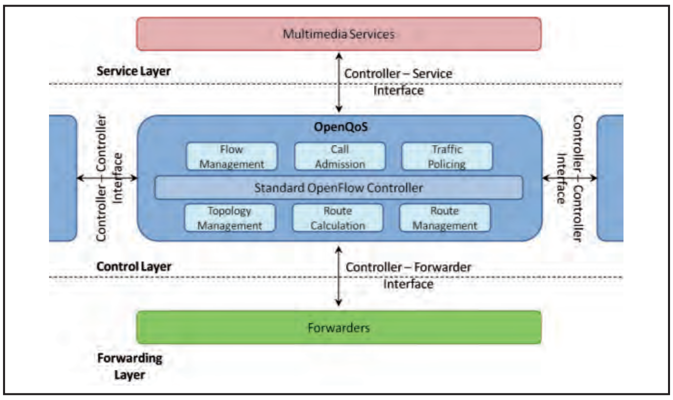


Figure 2: OpenQoS controller design

2 OPENQOS: CONTROLLER DESIGN

In this section we introduce OpenQoS. We first discuss its architecture, and then we present the optimization framework for dynamic QoS routing. The organization of this section is as follows: In Section 2.1, we present the proposed QoS architecture and OpenQoS running on top. Section 2.2 discusses the routing mechanism in OpenQoS. The optimization framework for QoS routing is given in Section 2.3.

2.1 QoS Architectures & OpenQoS Design

There is a continuing debate on how to evolve the Internet in order to provide QoS for multimedia traffic. Currently, there is no QoS architecture that is successful and globally implemented. Some researchers argue that fundamental changes should be done to fully guarantee QoS, while others think slight changes are enough to have soft guarantees which will provide the requested QoS with high probability. So, we can group the QoS architectures into two major categories:

- **Integrated Services (IntServ) like architectures** provide hard QoS guarantees via resource reservation (bandwidth, buffer) techniques. The mechanism is similar to circuit switched networks (e.g. ATM), and data transmission starts after an end-to-end connection is established. The major problem of IntServ based architectures is that they require fundamental changes in the network core.
- **Differentiated Services (DiffServ) like architectures** provide soft QoS guarantees via scheduling (priority queuing). Unlike IntServ, DiffServ requires changes in the edge of the network. Edge routers should have packet classification functionality, and core routers should forward the packets based on their priorities.

In terms of routing, DiffServ still applies the same routing mechanism as the Internet does. On the other hand, IntServ uses the Resource Reservation Protocol (RSVP) [12] which inter-operates with any routing protocol to reserve resources along the calculated path. For each connection, QoS routing is performed only once for connection establishment and that connection remains until teardown. In IntServ, dynamic QoS routing is not applied because, (1) there is no need to change the QoS route since the required QoS has already been guaranteed; (2) it introduces latency due to reconnection establishment.

With OpenQoS, we propose a new prioritization scheme which is based on routing. In order to fulfill the required end-to-end QoS, we propose dynamic QoS routing for QoS flows (multimedia traffic) while other flows (data) remain on their shortest path. Our approach is different from the current QoS architectures since we use neither resource reservation nor priority queuing (*i.e.* rate shaping). The main advantage of not using these methods is that the adverse effects of QoS provisioning on non-QoS flows, such as packet loss and latency, are minimized. Complete comparison between OpenQoS and the current QoS architectures is presented in Table 1.

OpenQoS is an extension of the standard OpenFlow controller which provides multimedia delivery with QoS. As depicted in Fig.2, OpenQoS offers various interfaces and functions to enable QoS. The main interfaces of the controller design are:

- **Controller - Forwarder Interface**: The controller attaches to forwarders with a secure channel using the OpenFlow protocol to share necessary information. The controller is responsible to send flow tables associated with data flows, to request network state information from forwarders for discovering the network topology, and to monitor the network.
- **Controller - Controller Interface**: The single controller architecture does not scale well when the network is large. As the number of the OpenFlow nodes increases, multiple controllers are required. This interface allows controllers to share the necessary information to cooperatively manage the whole network.
- **Controller - Service Interface**: The controller provides an open, secure interface for service providers to set flow

definitions for new data partitions and even to define new routing rules associated with these partitions. It also provides a real-time interface to signal the controller when a new application starts a data flow.

The controller should also manage several key functions:

- **Topology management**: This function is responsible for discovering and maintaining network connectivity through data received from forwarders.
- **Route management**: This function is responsible for determining the availability and packet forwarding performance of routes to aid the route calculation. It requires collecting the up-to-date network state from the forwarders on a synchronous or asynchronous basis.
- **Flow management:** This function is responsible for collecting the flow definitions received from the service provider through the controller-service interface, and efficient flow management by aggregation.
- **Route calculation**: This function is responsible for calculating and determining routes for different types of flows. Several routing algorithms can run in parallel to meet the performance requirements and objectives of different flows. Network topology and route management information are input to this function along with service reservations.
- **Call admission**: This function denies/blocks a request when the requested QoS parameters cannot be satisfied (i.e. there is no feasible route), and informs the controller to take necessary actions.
- **Traffic policing**: This function is responsible for determining whether data flows agree with their requested QoS parameters, and applying the policy rules when they do not (e.g. pre-empting traffic or selective packet dropping).

2.2 Per-Flow Routing in OpenQoS

The current Internet does not allow routing on per-flow basis. When a packet arrives at a router, it checks the packet's source and destination address pair with the entries of the routing table, and forwards it according to predefined rules (e.g. routing protocol) configured by the network operator. On the other hand, OpenFlow provides the flexibility of defining different types of flows to which a set of actions and rules can be associated. For example, one type of flow may be forwarded using the Open Shortest Path First (OSPF) [13] routing protocol and the other flows may follow manually configured routes over the network. So, each flow (*i.e.* packet) can be treated differently at the networking layer.

In OpenFlow, we can define flows in many ways. Flows can contain same type or different types of packets. For example, packets with the TCP port number 80 (reserved for HTTP) can be a flow definition, or packets having RTP header may indicate a flow which carries voice, video or both. In essence, it is possible to set flows as a combination of header fields as illustrated in Fig.3, but the network operator should also take into account the processing power limitations of the network devices (routers or switches). In order to avoid complex flow table lookups, flow definitions should be cleverly set and if possible aggregated [14]. In OpenFlow, network devices store the flows and their associated rules in flow tables which are processed as a pipeline shown in Fig.4. The goal of the pipelined processing is to reduce the packet processing time.

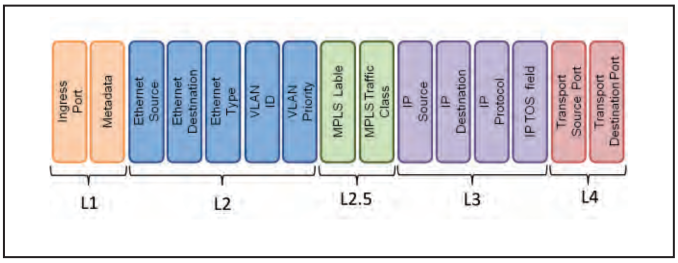


Figure 3: Flow identification fields in OpenFlow

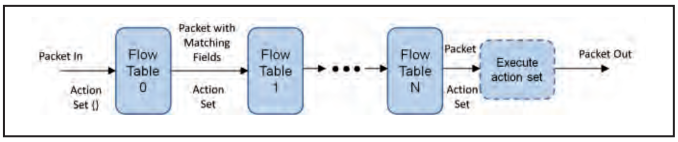


Figure 4: Flow tables and their pipelined processing

OpenQoS exploits OpenFlow's flow-based forwarding paradigm so that we can differentiate data and multimedia traffic. Multimedia flows may be determined by using the following packet header fields or values:

- Traffic class header field in MPLS,
- TOS (Type of Service) field of IPv4,
- Traffic class field in IPv6,
- If multimedia server is known, source IP address,
- Transport source and/or destination port numbers.

It is desirable to define flows according to lower layer (L2, L3) packet headers since the packet parsing complexity is lower compared to processing up to upper layers (L4). Therefore, we propose to define multimedia flows using fields in MPLS which is considered in between data link and network layer (L2.5), and provides ultra-fast switching capability. But, in some cases upper layer header fields may also be required for better packet type discrimination, and OpenFlow allows the flexibility of defining flows using upper layer (L4) header fields.

In order to calculate the QoS routes, it is essential to collect up-to-date global network state information, such as delay, bandwidth, and packet loss rate for each link. The performance of any routing algorithm is directly related to how precise the network state information is. Over large networks, collecting the network state globally may be challenging due to the scale of the network. The problem becomes even more difficult in the Internet because of its completely distributed (hop-by-hop) architecture. OpenFlow eases this task by employing a centralized controller. As illustrated in Fig.1, instead of sharing the state information with all other routers, OpenFlow forwarders directly send their local state information to the controller. Then, the controller collects the forwarders' state information and computes the best feasible routes accordingly.

2.3 Optimization of Dynamic QoS Routing

We pose the dynamic QoS routing as a Constrained Shortest Path (CSP) problem. It is crucial to select a cost metric and constraints where they both characterize the network conditions and support the QoS requirements. In multimedia applications, the typical QoS indicators are packet loss, delay and delay variation (jitter). However, some QoS indicators may differ depending on the type of the application, such as:

- *Interactive multimedia applications* that have strict end-to-end delay requirements (e.g. 150-200 ms for video conferencing). So, the CSP problem constraint should be based on the total delay.
- *Video streaming applications* that require steady network conditions for continuous video playout; however, the initial start-up delay may vary from user to user. This implies that the delay variation is required to be bounded, so the CSP problem constraint should be based on the delay variation.

In our formulation, a network is represented as a directed simple graph $G(N, A)$, where N is the set of nodes and A is the set of all arcs (also called links), so that arc (i, j) is an ordered pair, which is outgoing from node i and incoming to node j. Let R_{st} be the set of all routes (subsets of A) from source node s to destination node t. For any route $r \in R_{st}$ we define cost f_C and delay f_D measures as,

$$f_C(r) = \sum_{(i,j)\in r} c_{ij} \tag{1}$$

and

$$f_D(r) = \sum_{(i,j)\in r} d_{ij} \tag{2}$$

where c_{ij} and d_{ij} are cost and delay coefficients for the arc (i, j), respectively. The CSP problem can then be formally stated as finding

$$r^* = \arg\min_r \{ f_C(r) \mid r \in R_{st},\ f_D(r) \le D_{\max} \} \tag{3}$$

that is, finding a route r which minimizes the cost function $f_C(r)$ subject to the delay $f_D(r)$ to be less than or equal to a specified value $D_{\max}$. We select the cost metric as follows,

$$c_{ij} = g_{ij} + d_{ij},\ \forall (i, j) \in A \tag{4}$$

where g_{ij} denotes the congestion measure for the traffic on link (i, j) and d_{ij} is the delay measure. OpenQoS collects necessary parameters g_{ij} and d_{ij} using the *route management* function.

The CSP problem stated in (3) is known to be NP-complete, so there are heuristic and approximation algorithms in the literature. For the *route calculation* function of OpenQoS, we propose to use the Lagrangian Relaxation Based Aggregated Cost (LARAC) algorithm which is a polynomial-time algorithm that efficiently finds a good route without deviating from the optimal solution in $O([m+n\log n]^2)$ time [14]. When the *route management* function updates the QoS indicating parameters or the *topology management* function detects a topology change, the *route calculation* function runs the LARAC algorithm to solve the CSP problem of (3). Then, controller updates the forwarders' flow tables accordingly. Hence, the QoS routes are dynamically set.

3 OPENQOS TEST NETWORK

3.1 Test Network

We deployed the OpenFlow test network composed of three OpenFlow enabled Pronto 3290 switches, one controller and 3 host computers. As shown in Fig.5, the switches are connected in a triangular shape to ensure path diversity. The video streaming server and the client are connected to different switches, while the traffic loader is connected to same switch that the server connects, inserting cross-traffic into the network. Each switch initiates a secure connection to the controller using the OpenFlow protocol (see dashed lines in Fig.5). The controller runs our OpenQoS implementation which is described in detail in Section 3.2.

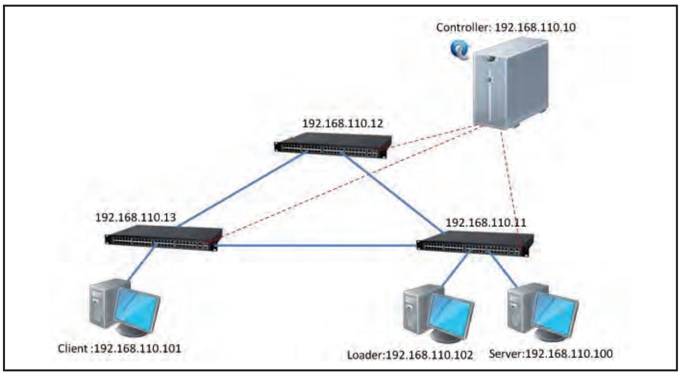


Figure 5: OpenFlow Test Network.

3.2 Implementation of OpenQoS

We implement OpenQoS over a standard OpenFlow controller, Floodlight [16]. There are also several standard controller alternatives such as NOX [17], Beacon [18], Maestro [19] to implement OpenQoS, but currently Floodlight is the most stable controller. Floodlight is an open source controller written in Java. It provides a modular programming environment so that we can easily add new modules on top and decide which existing modules to be run.

In our implementation of OpenQoS, we add two major modules to enable *route management* and *route calculation* functions discussed in Section 2.1. The *topology management* function has already been implemented in Floodlight and we directly used that module in OpenQoS. These functions are essential building blocks of our controller design which makes dynamic QoS routing possible. Yet, the OpenQoS implementation is still incomplete. First, *controller-to-controller, controller-to-service* interfaces must be defined, and then the functions using those interfaces (*flow management*, *call admission*, *traffic policing*) must be implemented. Since we concentrate on QoS in this paper, we left these open issues as future works.

3.2.1 Route Management

The route management module provides one of the key functions in the OpenQoS controller. It collects the up-to-date network state information such as link speed, available bandwidth and packet drop counts from the forwarders. The controller requests various statistics from forwarders by sending `FEATURE_REQUEST` messages, and in return forwarders send FEATURE_REPLY messages containing requested statistics. These messaging mechanisms are described in detail in OpenFlow specification v1.0 [20].

In order to support dynamic QoS, it is essential to keep the network state information up-to-date. The performance of the route calculation depends on the accuracy of the collected data. So, OpenQoS controller periodically collects available bandwidth for each link. The period is set to 1s since in the literature it has been shown that the Internet traffic behaves like independent Poisson distribution in sub-second time scales [21].

After receiving the available bandwidth measures from the forwarders, the *route management* module

- detects whether there is a congestion event in any of the links.
- determines link cost parameters to be used in the optimization problem stated in (3).

Each link can be in two states: *congested* or *non-congested*. We consider that a link is congested if that link is 70% bandwidth utilized. The link costs are determined by using the exact same formula in (4), where the congestion measure is found as,

$$g_{ij} = \begin{cases} \dfrac{\left(T_{ij} - 0.7 \times B_{ij}\right)}{T_{ij}} & , T_{ij} \geq 0.7 \times B_{ij} \\ 0 & , T_{ij} < 0.7 \times B_{ij} \end{cases} \tag{5}$$

where T_{ij} is the total measured traffic amount in bps and B_{ij} is the max achievable bandwidth in bps on link (i, j). Note that, in (5), the *non-congested* links have 0 congestion measure value. The delay parameter d_{ij} in (4) is set to 1 which simply corresponds to hop-count. This is because the current OpenFlow switch implementations do not have any support on collecting delay related statistics (total delay, jitter).

In order to add an event based dynamicity to the QoS routing, the *route manager* signals forwarders when QoS routes need to be rerouted. This signalling can be achieved by deleting a specific flow entry. After a QoS flow entry is deleted, the forwarders cannot match newly coming packets, therefore they ask the controller to define new flow entries which causes multimedia packets to be rerouted. The flow deletion is triggered in two cases: (1) If a link previously *non-congested* is now *congested*, we delete the flow entries matching multimedia (QoS) packets in the flow tables of the forwarders. (2) If a link previously *congested* is *non-congested* in the last 3 periods, we again delete the flows accordingly. We require 3 periods of *non-congested* state to ensure there are no fluctuations in the traffic rate on the links.

3.2.2 Route Calculation

In Floodlight, route calculation is done when a PACKET_IN message arrives to the controller. It calculates the shortest path route and pushes flow definitions to the switches along that path accordingly. On the other hand, OpenQoS first checks if it is a multimedia packet or not, based on pre-defined flow setups described in Section 2.2. Then, the *route calculation* module calculates two paths between the source and destination pair of the incoming packets. One path is the QoS optimized path and the other is the shortest path. Note that the QoS routes are calculated using the LARAC algorithm as described in Section 2.3. Currently, we only detect multimedia packets but it can be easily modified to add new routing policies to new type of services.

3.3 Possible Application Areas of OpenQoS

Video-sharing websites like YouTube, Vimeo or Metacafe stream video to users over the Internet without providing any QoS guarantees. This best effort streaming service is acceptable for short clips where a moderate pre-buffering period provides continuous playback. However, the QoS guarantee on Internet Protocol television (IPTV), which is defined as multimedia services delivered over IP based networks managed to provide the required level of QoS [22], cannot be realized effectively over the best effort Internet. That’s why the common delivery scenario for the IPTV service is over an investment-heavy walled garden network, where the IPTV service provider builds a separate IP network besides the existing Internet access infrastructure. However, the usage of OpenQoS will change this scenario, making the IPTV delivery possible over existing OpenFlow enabled networks, thanks to the dynamic QoS that it provides. OpenQoS eliminates the need to build a separate IPTV infrastructure and increases the utilization of the current network with its adaptive routing logic.

Another application area for OpenQoS is load-balancing. In a classical scenario, a load-balancer device distributes the incoming traffic to a pool of servers, based on a pre-defined algorithm, so as to balance the server load. Hence, the conventional load-balancer chooses the optimum server to direct the traffic to, based on the server state, but it does not consider the state of the network to be used in routing the requests to the server. Here, OpenQoS can be utilized to jointly select the optimum server and the optimum path to the server, balancing the load of the network as well as the server. There are already examples of OpenFlow enabled load-balancers, working to distribute both the server and the network load, like Aster*x [23].

4 RESULTS

To demonstrate the performance of our OpenQoS implementation, we built a video streaming environment over a real OpenFlow test network shown in Fig.5. Throughout the tests, we used the standard MPEG test sequence *into the tree* having 500 frames with the resolution of 1280×720. We looped the raw video sequence reversely once to have 1000 frames lasting about 40s (25fps). We then encoded the sequence in H.264 format at an average bit-rate of 1.8 Mbps (32.51dB PSNR) using the *ffmpeg* encoder (v.0.7.3) [24].

We created a scenario where two copies of the same video (*into the tree*) are streamed from the server residing at 192.168.110.100 to the client with the IP address 192.168.110.101 (see Fig.5). The server uses VLC media player [25] to stream videos using UDP/RTP. One copy of the video is sent to the destination port 5004 while the other copy is sent to port 5005. To show the performance difference in terms of QoS, we matched the multimedia (QoS) flows to the transport port number, 5004. Thus, the video packets destined to port 5004 are identified as being part of a multimedia flow by the OpenQoS controller and routed accordingly, while the other video (destined to port 5005) is considered as a data flow which has no QoS support (i.e. best-effort). In each test, 10 second long cross-traffic is sent from the loader (192.168.110.102) to the client once at a random time. The client runs two VLC players, listening UDP/RTP packets at ports 5004 and 5005, to play and save the received videos. We expect to see distortions in the video received on port 5005 during the cross-traffic while the other video received on port 5004 will be rerouted and affected little or not at all in terms of video quality.

We decode the received videos using *ffmpeg* and measure the peak signal to noise ratio (PSNR) values with respect to the original raw video. The results are given in Figs.6 and 7 which are in terms of received video quality (PSNR) versus time. The vertical dashed lines mark the start time and end time of cross-traffic.

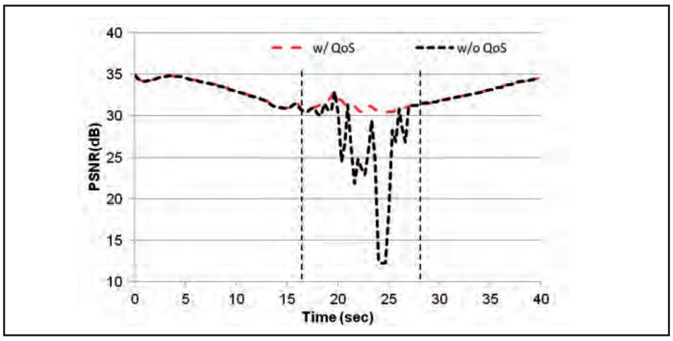


Figure 6: Best case result

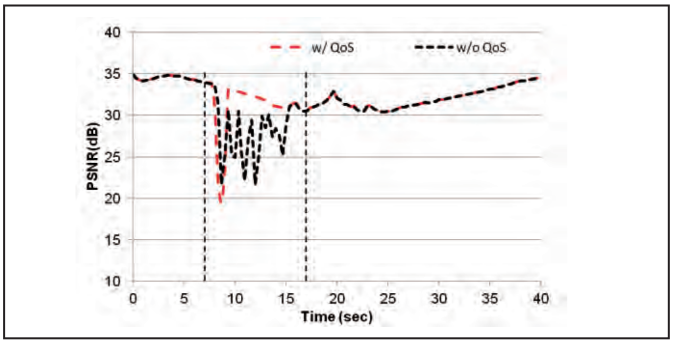


Figure 7: One case result

The best case result is shown in Fig.6 where the video with QoS support (w/ QoS) is not affected from the cross traffic and approaches full video quality, while the video without QoS (w/o QoS) support has a significant amount of quality loss. However, in Fig.7, the video with QoS also has a significant amount of quality loss, but it is recovered in less than 1s. After repeating the scenario 10 times, we observed that the average loss recovery period is 0.85s, and most of the time the user watching the video is not disturbed from the quality loss in such a small interval.

5 CONCLUSION

OpenQoS is a novel approach to stream video over OpenFlow networks with QoS. It is different from the current QoS mechanisms since we propose dynamic QoS routing to fulfill end-to-end QoS support which is possible with OpenFlow's centralized control capabilities over the network. Our experimental results show that OpenQoS can guarantee seamless video delivery with little or no disturbance experienced by the end users. Unlike other QoS architectures, OpenQoS minimizes the adverse effects (such as packet loss and latency) on other types of flows.

References

[1] J. H. Saltzer, D. P. Reed, and D. D. Clark "End-to-end arguments in system design", *ACM Transactions on Computer Systems*, vol.2, no.4, 277-288, Nov. 1984.

[2] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview", RFC 1633, Jun. 1994.

[3] S. Blake, D. Black, et.al., "An architecture for differentiated services", RFC 2475, Dec. 1998.

[4] E. Rosen, and Y. Rekhter, "BGP/MPLS VPNs", RFC 2547, Mar. 1999.

[5] "Open Networking Foundation" [Online]. Available: http:// opennetworking.org

[6] Open Networking Foundation, "Software defined networking: the new norm for networks", ONF whitepaper.

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks", *ACM SIGCOMM Computer Communication Review*, 38(2):69-74, April 2008.

[8] "OpenFlow." [Online]. Available: http://openflowswitch.org

[9] H.E. Egilmez, B. Gorkemli, A.M. Tekalp, and S. Civanlar, "Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing", *IEEE Proc. Intl. Conf. on Image Processing*, September 2011.

[10] H.E. Egilmez, S. Civanlar, and A.M. Tekalp, "A distributed QoS routing architecture for scalable video streaming over multi-domain OpenFlow networks", *IEEE Proc. Intl. Conf. on Image Processing*, 2012.*(to appear)*

[11] S. Civanlar, M. Parlakisik, A.M. Tekalp, B. Gorkemli, B. Kaytaz, and E. Onem, "A QoS-enabled OpenFlow environment for scalable video streaming", *IEEE Globecom 2010 Workshop on Network of the Future (FutureNet-III)*, Miami, USA, 2010.

[12] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol", *IEEE Network*, September 1993.

[13] J. Moy, "Experience with the OSPF protocol", RFC 1246, July 1991.

[14] S. Das, Y. Yiakoumis, G. Parulkar, N. McKeown, et.al., "Application-aware aggregation and traffic engineering in a converged packet-circuit network", In *OFC/NFOEC, 2011 and the National Fiber Optic Engineers Conference*, pp. 1-3, Mar. 2011.

[15] A. Juttner, B. Szviatovszki, I. Mecs, and Z. Rajko, "Lagrange relaxation based method for the QoS routing problem," *IEEE Proc. INFOCOM*, vol. 2, pp. 859–868, Apr. 2001.

[16] "Floodlight" [Online]. Available: http://floodlight.openflowhub.org

[17] "Nox" [Online]. Available: http://noxrepo.org

[18] "Beacon" [Online]. Available: http:// openflow.stanford.edu/display/Beacon/

[19] Z. Cai, A. L. Cox, T. S. Eugene Ng, "Maestro: balancing fairness, latency and throughput in the OpenFlow control plane", Rice University Technical Report TR11-07.

[20] "OpenFlow Switch Specification v1.0" [Online]. Available: http://openflow.org/wp/documents/

[21] V.S. Frost, B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, vol.32, no.3, pp.70-81, Mar. 1994.

[22] *Requirements for the support of IPTV services*, ITU-T Recommendation ITU-T Y.1901, 2009.

[23] N. Handigol, S. Seetharaman, M. Flajslik, and A. Gember, "Aster*x: load-balancing web traffic over wide-area networks", *Design* , 2010.

[24] "ffmpeg" [Online]. Available: http://ffmpeg.org

[25] "VLC media player" [Online]. Available: http://videolan.org/vlc

Web delivery of free-viewpoint video of sport events

Chris Budd[1], Oliver Grau[2], Peter Schübel[3]

[1]University of Surrey, Guildford, UK; [2,3]BBC Research & Development, London, UK;

E-mail: [1]chris.budd@surrey.ac.uk, [2]Oliver.Grau@gmx.net,
[3]peter.schuebel@d3technologies.com

Abstract: **This paper describes capture and web delivery of free-viewpoint video (FVV). FVV allows the viewer to freely change the viewpoint. This is particularly attractive to view and analyse sport incidents. Based on previous work on the capture and replay of sport events for TV programme making we present a FVV player based on the WebGL API, which is part of HTML 5. The player implements a streaming mode over IP and an image-based rendering using view-dependent texture mapping.**

Keywords: 3D Capture of action, view-dependent texture mapping, Web-browser, Image-based Rendering.

1 INTRODUCTION

Free-viewpoint video (FVV) allows the viewer to change the viewpoint freely during the replay of a scene or action [1]. The scene is usually captured with multiple cameras surrounding the action and a 3D representation of that action is computed automatically. FVV has been applied to capture action in a studio, but also to visualise sport incidents for TV broadcast applications [2].

In TV programme making the interactive abilities of FVV are used as a tool by programme makers, the resulting visualisation is then delivered as conventional video. The viewer has in this case no control over the interactive capabilities of FVV.

An example of an interactive experience of sport events delivered via a web-service is Virtual Replay^{TM}. BBC Sport used this system for placing some selected incidents of football games online [3]. Virtual Replay^{TM} is based on Adobe Shockwave^{TM}. The football players are represented by generic 3D person models, i.e. they do not resemble much similarity with the actual players. The position of players is determined by an operator from still images of the game and is therefore only precise within the limitations of these manual placements. The system does allow changing the viewpoint to 'explore' the captured incident, but it does not provide a video playback functionality, i.e. it is limited to a static or 'frozen' scene.

This paper builds upon the automatic capture and processing of multi-camera captured action, as described in [2]. The rendering of this approach produces photo-realistic looking FVV of the captured scene. This is achieved by using view-dependent texture mapping employing all available video images in real-time in the player [4]. In this paper we describe a FVV player implemented using WebGL [5]. WebGL is a new API and is available natively in most HTML5 compatible web browsers, i.e. no plug-in is needed.

The remainder of this paper is structured as follows: The next section gives a very brief summary of our multi-camera capture system including the processing. Section 3 describes the coding and transmission of the FVV data. Section 4 gives a description of the FVV WebGL renderer. The paper finishes with some results and conclusions.

2 CAPTURE OF SPORT EVENTS USING MULTI-CAMERAS

This section gives a brief overview of our 3D reconstruction, for more detail see [2]. Synchronised capture of video content from multiple cameras is achieved with a distributed system based on IT-components [14]. The image and 3D processing consists of the following steps or modules:

1. camera calibration
2. image segmentation
3. 3D reconstruction
4. texture computation / preparation

2.1 Camera setup and calibration

The system is scalable with respect to the number of cameras used. The cameras can be arranged in an inhomogeneous, unstructured setup, allowing for example varying object distances, baselines, focal lengths or sensor sizes and resolutions. Calibration in controlled environments is achieved using a calibration target (a chart or LED object) that is recorded in various poses to cover the whole volume used for reconstruction. Static cameras are calibrated once before a capture session. In uncontrolled environments such as outdoor sports events, image-based techniques, like pitch-based calibration are used. For non-static cameras a live calibration is used with techniques described in [6].

2.2 Image processing and 3D Reconstruction

The 3D reconstruction is a technique known as 'shape-from-silhouette' or visual hull computation and relies on a segmentation of the scene into foreground objects (the silhouettes) and background. The silhouettes can be acquired through chroma keying if a studio with chroma-keying facility is available. Other techniques include difference keying, or a combination of the two, which was successfully applied to a number of sports scenarios [2].

For the 3D reconstruction we use a robust octree-based visual hull algorithm as presented in [7].

3 TRANSMISSION

The goal of this work is to allow home users to access the benefits of FVV without any further requirements, besides an internet connection and a WebGL-compatible browser. Therefore all content needs to be transmitted to the client when it visits the website. This website contains a basic HTML structure, including a HTML5 canvas for the WebGL content and JavaScript code for the rendering loop and the user control interface. The actual code for initialisation, processing and rendering, as well as the required data are served separately.

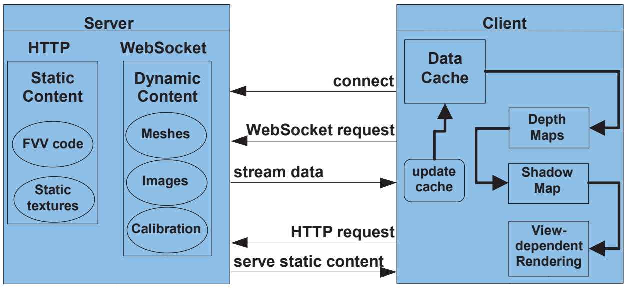


Figure 1: Client-Server setup for FVV transmission.

3.1 Web server setup

The web server provides the clients with static resources, as well as dynamic resources which change depending on the current frame of the FVV. Static resources are provided via the HTTP protocol, whereas for frame-dependent data the new WebSockets protocol [9][10] was chosen. The WebSocket protocol is an independent TCP-based protocol aimed at the requirements of streaming live and interactive content by enabling bidirectional connections. Unlike the HTTP protocol, the current version 1 of the WebSockets protocol does not support data compression (e.g. use of *gzip*). Native protocol compressions could be used as soon as they are included into a new version of the WebSockets standard.

The server machine runs a standard HTTP server and a separate WebSockets server on different ports to deal with all requests. For our implementation the server-side JavaScript environment *Node.js* with its WebSockets module *socket.io* were chosen [12]. Server-side caching is used for all static content, but also for the dynamic content, as this demonstrator provides a short pre-recorded sequence, rather than actual live content. The client consequently caches all received content, so it can play back the sequence in a loop and allows the user to pause, play and rewind the sequence.

3.1.1 Static resources

Static resources need to be sent to the client only once per session. They include the JavaScript code that implements I/O and user control, the WebGL rendering code including GLSL shader code, as well as static background textures. Static resources might still change sporadically, e.g. when a background texture has been modified, or after updates to the rendering code. In this case the client browser will need to reload the website. The client requests static content from the web server using *HTTP GET* requests, which in JavaScript are initiated through *XMLHttpRequest* calls [11]. If it is supported by both server and client, then *gzip* compression is used to minimise the amount of data transferred.

3.1.2 Dynamic resources

Dynamic resources need to be sent to the client continuously, usually at the frame rate of the source video. They include the actual 3D mesh data and texture images for each original camera. They also include per-frame camera calibration data, as the cameras may be moving, rotating or zooming, etc.

The client first initiates a WebSocket connection to the server. It registers different callback functions for images, meshes and camera data, and finally requests the data stream from the server. The server then begins streaming data to the client, which will call the respective callback method for each received piece of data, to update the image texture, mesh or camera calibration data (see Figure 1).

3.2 3D triangular surface models

The 3D surface models of the scene are represented as triangle lists (sometimes called 3D meshes), which include surface normal data. They are streamed as array-style JavaScript objects to allow direct transfer to the GPU. No compression is used, as JavaScript doesn't natively support handling of binary data.

Currently our 3D data is coded on a frame-by-frame basis. This means that each mesh corresponds to a single frame and has a different set of vertices and varying topology. The application of a single texture map throughout the sequence is thus not possible. The lack of temporal consistency of the mesh topology makes it necessary to calculate a new set of texture coordinates for each mesh and each set of camera calibration data. This is done automatically in WebGL using projective texturing.

3.3 Image textures

Projectively texturing each frame requires that for every camera which is to be included in the texturing process the corresponding video frame is transmitted. The client only needs a certain subset of images for any given virtual camera pose (e.g. for the 3 cameras closest to the virtual camera). Although it could request only those required images at any point in time, this would introduce additional latency, due to the required bidirectional communication. As all of the data in this demonstrator is cached first anyway, we currently transmit all camera images for every frame in our experiments.

For multiple HD cameras this results in a lot of data both for transmission and transfer to the GPU. In order to make this feasible the images associated with each frame are down-sampled prior to transmission. To further reduce the amount of transferred data, all images are JPEG compressed on the server side. Due to the lack of support for binary data in JavaScript, the transmission via WebSockets is achieved by applying server-side Base64 encoding on the JPEG data. Although this again increases the data size to be transmitted by roughly a third, this is still outweighed by the gain of the JPEG compression. Another solution to reduce the immense amount of image data would involve transmission of the images as a single video stream. The multiple (down-sampled) camera views could be concatenated into a single HD image and compressed into a single HD video stream. The inter-frame compression of video codecs would thus help to reduce the amount of image data considerably.

As the available internet connection bandwidth increases, the resolution of this video stream could be increased to improve rendering quality. Encoding multiple resolutions of video and choosing a resolution for transmission based on the available bandwidth also provides a means to make FVV available to the widest possible audience. The resolution actually streamed could also depend on the distance of the virtual camera to a certain surface, to optimise the FVV quality. Multiple resolutions further open the door for use on mobile devices which are increasingly providing support for OpenGL ES.

3.4 Camera calibration data

The camera calibration data includes external and internal parameters required for projective texturing and to calculate the blending weights for the FVV. It is parsed from XML files and transmitted as uncompressed JavaScript objects. For static cameras only one set of parameters is transmitted. For cameras that move, rotate or zoom a separate set of parameters is transmitted for every frame.

4 FVV WEBGL RENDERER

The recent release of HTML 5 has lead to the emergence of a number of compatible technologies to improve graphics and interactivity within web pages. WebGL is a JavaScript API developed by the Khronos Group, which exposes a subset of OpenGL functionality from within the web browser. WebGL is based on the OpenGL ES 2.0 specification and render graphics to an HTML 5 canvas [5]. For the purpose of the work presented here WebGL version 1 is used. On the client-side the web browsers Chrome 14 and Firefox 8 and higher versions were successfully tested.

The WebGL rendering pipeline that enables FVV consists of a number of stages: depth map generation, shadow map production and projective texturing based on the multiple camera views.

The texturing of each mesh is created projectively on a frame-by-frame basis, by applying a texture projection matrix that is calculated for every camera. In the current implementation projectively texturing a mesh involves each vertex being back-projected into each of the camera views, to get the actual texture coordinates. Final pixel colours are decided after rasterization in the fragment shader on a per-fragment basis.

4.1 Visibility test

The amount by which a camera is contributing to any given pixel is assessed based on a visibility test and the interpolated surface normal at the fragment in question. A fragment is considered visible from a particular camera when the projection of its corresponding world coordinate into that camera matches the depth at that pixel in the depth map for that camera. The depths are considered to match if they are within a given delta to allow for precision errors. Fragments occluded in each of the camera views will have different depths (see Figure 2). The visibility is a hard binary decision that initialises a per-camera weight to either 1 (visible) or 0 (not visible).

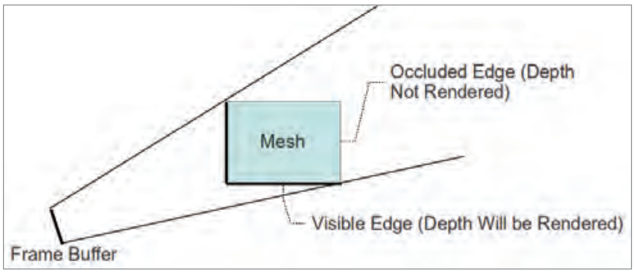

Occluded Edge (Depth Not Rendered)
Mesh
Visible Edge (Depth Will be Rendered)
Frame Buffer

Figure 2: Depth map visibility testing.

4.2 Blending function

The final colour of a pixel is selected based on a maximum of three cameras which have the best view of that fragment. The visibility test already eliminates all cameras that do not see the corresponding world point. The quality of the view is assessed based on a weight that is derived from the viewing angle with which a camera views that fragment. This is quantified by the cosine of the angle between the surface normal at the fragment and a ray cast from the camera to the world coordinate of the fragment (see Figure 3). In practice this weight is calculated using the dot product between these two vectors. Note that the surface normals for each fragment are automatically interpolated from the surrounding vertex normals of the triangle to which the fragment belongs.

A camera view is also discarded if the calculated weight is smaller than a threshold value of 0.25. This value was chosen since it was experimentally demonstrated to result in fewer artefacts from unsuitable camera views.

Up to three cameras can contribute to the final pixel colour of each fragment. If more than three cameras have a weight above the threshold, then the best three are selected. A fragment with multiple cameras, which view the pixel within these criteria, is coloured according to the sum of each contributing camera's colour, weighted according to its normalised weight. The weight value is normalised by dividing it by the sum of the weights of all selected cameras. For scenes with evenly distributed cameras with similar framing, using more than three cameras does not provide significant improvement of the visual quality and therefore this number has been chosen.

Currently no lighting model is applied to the rendered meshes since the camera images are captured under lit conditions. A re-lighting of the model would require the computation of un-lit texture maps (albedo).

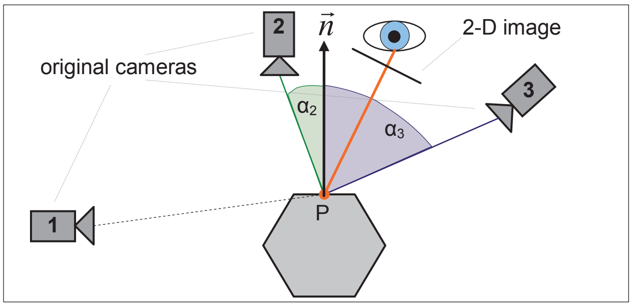


Figure 3: Weighting for view-dependent rendering: camera 1 cannot see the surface point P; camera 2 and 3 will contribute according to the cosine of their respective angle

4.3 Depth map calculation

Visibility testing during the projective texturing process requires additional depth information to assess visibility from each camera view. This depth information would not normally be required when rendering with a single global texture map. The standard approach to generating depth maps involves rendering the scene onto each camera using a perspective projection based on the camera parameters associated with the corresponding camera. Modern graphics cards are capable of performing this task for multiple cameras, whilst maintaining a frame rate high enough to view a smooth video. WebGL presents a few caveats to consider when creating depth maps. At the time of writing, WebGL does not permit access to the depth channel of a frame buffer object from the CPU. Additionally, floating point textures are only supported with the *OES_texture_float* extension [8], which is not yet supported by all browsers. To maximise compatibility, the depth is packed into the 24 bits provided by the RGB colour channels. The alpha channel of the colour attachment is avoided since – at the moment – the WebGL implementations in Firefox and Chrome always pre-multiply the alpha channel in textures. This practical limitation allows only the RGB colour channels to be used to convey data to the graphics card.

There is potential for this computation to be done offline, prior to transmission. Since the video stream being rendered has known camera positions for each frame, the visibility information can be pre-computed and transmitted as a single bit per pixel representing visibility in each camera. With this approach there is also the potential for stream-based compression of this visibility information yielding the smallest possible transmission format. Due to JavaScript's lack of support for binary data this would again require Base64 encoding, which would result in a 33% increase to the compressed data size. This makes on-the-fly depth map computation more attractive.

4.4 Shadow rendering

Shadow rendering is a standard feature in high-end computer graphics, as for example used in offline rendered animations. Shadows can also improve real-time graphics and we implemented a technique based on shadow maps to add shadows to the scene.

The shadow mapping approach is selected over shadow volumes since there is a limit to the amount of computational power available and shadow mapping is less computationally expensive. The same principle applies to the production of shadow maps as the creation of depth maps. Shadow maps are depth maps rendered from the point of view of a light source rather a camera.

To achieve this in WebGL a frame buffer object, with attached colour buffer is created to hold each shadow map. The perspective projection is generated based on the location of the light source and required field of view to produce all visible shadows. By virtue of a custom shader each vertex is back-projected into the appropriate light source and the computed depth is packed into the RGB channels of the provided frame buffer.

All shadow and depth maps are independent of the selected viewpoint and only change on each frame. An additional computational saving is made by only updating shadow maps at the frame rate of the transmitted free-viewpoint video rather than the rendering rate achieved on the local machine. Shadow maps are computed as the meshes are displayed for the first time and on subsequent renderings of that frame they are reused.

Shadows maps are applied as a down-weighting of the colour of each fragment that is in shadow. Fragments are in shadow if the depth of the corresponding world coordinate when projected into a light does not match the depth at that location in the shadow map (to within a given delta).

5 RESULTS

5.1 Studio setup

A studio-based test capture was set up around several basketball players to evaluate the 3D reconstruction pipeline. For this work a sequence with two players was selected. A total of 13 broadcast-quality HD cameras with 1920x1080 pixels and a single SD camera were used. All cameras were based on 3-CCD chip technology, except one HD camera that had a 3-CMOS chip sensor. For this test, all cameras were static and had a frame rate of 25 Hz. A shutter speed of 1/250 s was used to prevent motion blur problems caused by fast movement of the players. Synchronisation of the exposure time intervals was achieved with a common burst signal provided to all cameras through the GenLock input. The distributed real-time capture system described in [14] allows synchronisation to be maintained across the boundary of broadcast and IT equipment.

A camera setup was chosen that would cover the scene from all angles to aid the 3D reconstruction process and provide good textures. The SD camera was positioned directly above the players to improve the shape generation, but was not used for texturing. 10 HD cameras were placed in an oval setup around the studio at a height of 2.5m to cover as much of the scene as possible from 360 degrees. Two HD cameras were placed on tripods at head height, using longer focal lengths than the other cameras, to give higher resolution textures. The last HD camera was fixed high on the ceiling to improve shape and texture when looking down on the scene.

Two of the 14 original camera images can be seen in Figure 4 and Figure 5. These images are taken from the capture servers after applying lens undistortion.

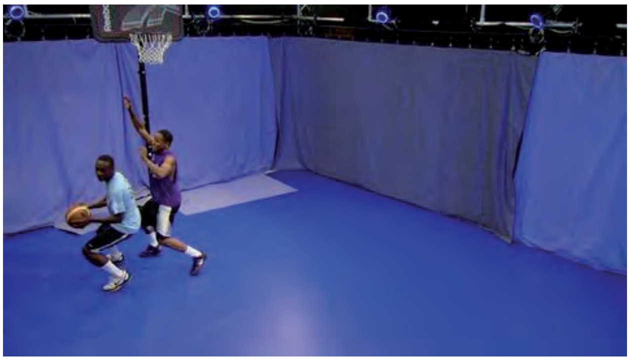


Figure 4: Undistorted image from original camera 7

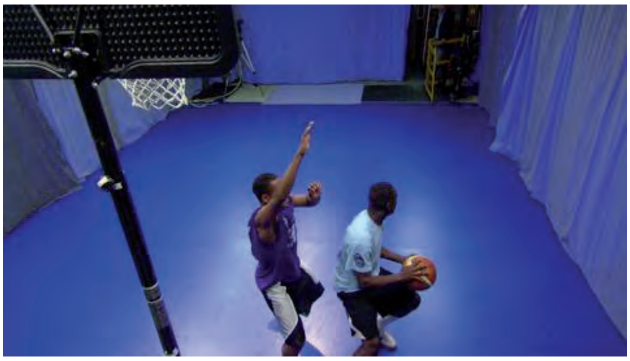


Figure 5: Undistorted image from original camera 10

5.2 Test systems

The proposed proof-of-concept client was tested on an Intel Core i7 920 quad-core processor with a NVIDIA GeForce GTX 260 GPU and 8 GByte of RAM. The server application was running on an Intel Xeon 5140 dual-core processor with 2 GByte of RAM. Both systems are running the GNU/Linux operating system openSUSE 11.4 and are connected via a 1 GBit network. The web browsers were further tested on the same client machine running Windows Vista 64-bit.

5.3 Data transmission

After capture and processing of the video data, all images, meshes and camera files were stored on the web server. For this demonstrator 7 HD cameras were selected. For these cameras, a two second clip (50 frames at 25 Hz) was actually used for transmission between server and client.

The generated meshes have an average complexity of 17000 vertices and 35000 surfaces. The actual size of the transmitted mesh objects varied between 235 and 355 Kbytes. The camera calibration data requires only around 600 byte per camera, totalling about 4 Kbyte.

All HD camera images are in RGB colour space and have an uncompressed size of around 6 MByte. Scaling the images down to a 16th of the original resolution already reduces the image size considerably. JPEG compression of the files results in a further compression factor of around 13. However, the Base64 encoding currently necessary increases the image size again by a factor of around 1.3. The total size of the images of one camera for the 50 frame clip is between 1.6 and 2.2 MByte. This means that for the 7 chosen cameras around 14.4 MByte of images need to be transferred.

In summary, for this 2 second clip the WebSocket server needs to transmit a total of 27.3 Mbyte to the client. On an internet connection with 8 MBit/s bandwidth this would still require over 14 seconds until all the data for 1 second is transmitted. The client can however start replay as soon as a certain part of the sequence has been buffered. For every frame almost 300 KByte of images are sent to the client, which results in about 7.3 MByte/s. An alternative approach where the client requests just the images actually required would only roughly halve this amount of data, because three images will still be required for the view-dependent texturing.

5.4 Visual output

To test the client-side website application, Google's Chrome browser with versions between 14 and 19 and Mozilla Firefox versions from 8 to 12 were used. Both browsers were successfully running the interactive FVV client application, and the graphical output quality was indistinguishable in both browsers. Tests with an alpha-version of the Opera 12 browser showed that it supported the WebGL and WebSocket technologies, but that it failed to access the texture images that were embedded using a data URL scheme [13]. The rendered output therefore showed only untextured black 3D models. Tests with Safari 5.1 on an iMac with a 3.4 GHz i7 Intel CPU, 4 GByte RAM and an AMD Radeon HD 6970M failed due to missing support for some WebGL extensions.

Figure 6 and Figure 7 show the actual FVV running in the Chrome and Firefox web browsers. The basketball players were placed in a simple virtual environment with static textures. The black sphere in the background marks the position of the light that causes the visible shadow on the virtual floor.

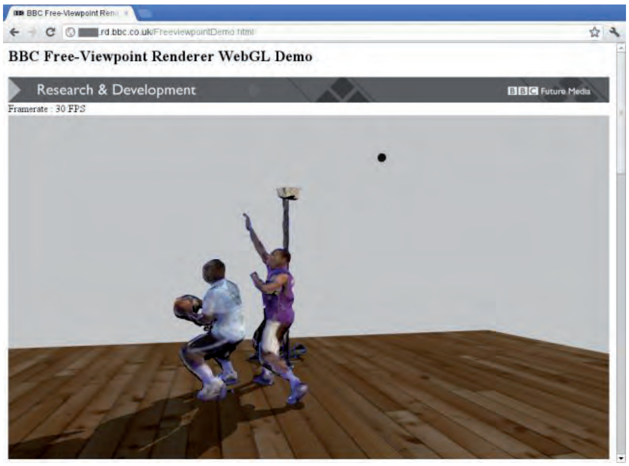

BBC Free-Viewpoint Ren
[redacted].rd.bbc.co.uk/FreeviewpointDemo.html
BBC Free-Viewpoint Renderer WebGL Demo
Research & Development
BBC Future Media
Framerate : 30 FPS

Figure 6: FVV in web browser Chrome 18.0

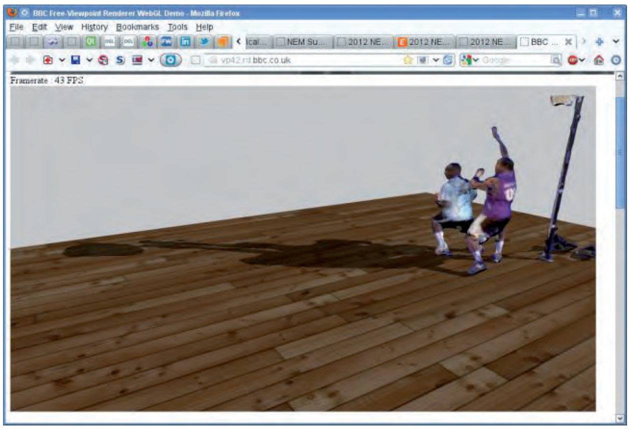

BBC Free-Viewpoint Renderer WebGL Demo - Mozilla Firefox
File Edit View History Bookmarks Tools Help
ical...
NEM Su...
2012 NE...
2012 NE...
2012 NE...
BBC ...
vp42.rd.bbc.co.uk
Google
Framerate : 43 FPS

Figure 7: FVV in web browser Firefox 12.0

5.5 Performance analysis

The rendering performance depended strongly on the chosen virtual viewpoint and on the client. In Chrome the frame rate ranged from 14 to 33 fps, in Firefox from 23 to 47 fps. Under Windows Vista those values were about 20 fps higher. Systems with more modern GPUs can be expected to achieve much higher frame rates. The memory usage in Chrome was 350 MByte, in Firefox 310 MByte and in Opera 530 Mbyte.

The server start-up times depend heavily on the I/O performance of the system, as most of the work is concerned with loading and caching all resources. On the older Xeon server system this takes about 9.6s, on the i7 only 2.5s.

The start-up times on the client were measured for individual start-up stages using the Chrome browser on the specified client test system. The mean of 20 measurements was used, and each measurement was taken after clearing the browser cache and a restart. To load all static resources it took around 154 ms. The time for initialising the WebGL context, GLSL shader compilation, creation of static graphics content and initiating the WebSocket connection took on average around 643ms. The delay between initiating the client connection and the first mesh arriving at the client was 1.41s. The total time between initiating the client connection and processing the last arrived item on the client was 3.71s.

6 CONCLUSIONS

This paper demonstrates a prototype of an interactive FVV application in a HTML5 web browser. With the help of a WebGL-enabled browser and an internet connection users can navigate a virtual environment populated with 3D objects generated from real video data. The visual quality of the rendered 3D objects is increased by using view-dependent texturing. This is implemented by intelligently combining multiple images from the original cameras at the texturing stage, taking into account viewing angles and occlusions. Virtual shadows are added to improve the integration of the reconstructed objects in the virtual environment.

Future work will include research on adding interactivity that goes beyond the choice of viewpoints and enables actual interaction with the reconstructed 3D objects. Research into the reconstruction of temporally consistent 3D models (see [7], [15]) also promises both better quality of meshes and the possibility of inter-frame mesh compression.

This work builds heavily on JavaScript, as well as WebGL and WebSockets, two innovations of the new HTML5 standard. Certain aspects of these new technologies are still in their early stages. Above all, the current limitations of WebGL demanded unintuitive or suboptimal solutions to several problems, especially when compared to the features offered by OpenGL. Recent research has shown development of WebGL extensions which are capable of making use of video textures at a high frame rate. This yields the potential for faster transmission of the image data and faster transfer to the GPU, when compared to the current single image implementation.

Despite those current shortfalls, this work shows that a working FVV application can achieve interactive frame rates. Future improvements on the used technologies and further research into the above mentioned areas will improve the achievable quality and allow larger audiences to be reached. More efficient transmission coding of texture and mesh data and the continuing spread of high bandwidth internet connections will eventually allow FVV for live action content.

References

[1] J. Carranza, C. Theobalt, M Magnor, and H.-P. Seidel, “Free-viewpoint video of human actors,” ACM Trans. on Computer Graphics, vol. 22, no. 3, July 2003.

[2] Oliver Grau et al., “A free-viewpoint system for visualisation of sport scenes,” in Conference Proc. Of International Broadcasting Convention, Sept. 2006.

[3] Virtual Replay on BBC sport page. http://news.bbc.co.uk/sport1/hi/football/fa_cup/virtual_replay/default.stm

[4] J. Starck and J. Kilner and A. Hilton, "A Free-Viewpoint Video Renderer“, Journal of graphics, GPU and game tools, vol. 14, no.3, 2009

[5] “WebGL, A OpenGL API for web applications”, Khronos Group. http://www.khronos.org/webgl/

[6] Graham A. Thomas. “Real-time camera pose estimation for augmenting sports scenes”, Proc. of 3rd European Conf. on Visual Media Production (CVMP2006), London, UK, November 2006.

[7] O. Grau, "Multi-view 4D reconstruction of human action for entertainment applications", book chapter in ‘Visual Analysis of Humans: Looking at People’, Moeslund, Th.B.; Hilton, A.; Krüger, V.; Sigal, L. (Eds.), Springer 2011.

[8] “WebGL OES_texture_float Extension Specification”, Khronos Group. http://www.khronos.org/registry/webgl/extensions/OES_texture_float/

[9] I. Fette and A. Melnikov, “The WebSocket Protocol”, Internet Engineering Task Force, IETF RFC 6455, December 2011. http://tools.ietf.org/html/rfc6455

[10] Ian Hickson, “The WebSocket API”, W3C Candidate Recommendation, December 2011, http://www.w3.org/TR/websockets/

[11] “XMLHttpRequest Level 2”, W3C Working Draft 17 January 2012. http://www.w3.org/TR/XMLHttpRequest

[12] Node.js, Joyent Inc.., http://nodejs.org

[13] L. Masinter, “The data URL scheme”, The Internet Society, IETF RFC 2397, August 1998. http://tools.ietf.org/html/rfc2397

[14] J. Easterbrook, O.Grau, P.Schübel, “A system for distributed multi-camera capture and processing”. Conference on Visual Media Production (CVMP 2010), London, UK, November 2010.

[15] C. Cagniart, E. Boyer, and S. Ilic. 2010. “Probabilistic deformable surface tracking from multiple videos”. Proc. of the 11th European conference on Computer vision, Crete, Greece, 2010.

Connected Media Worlds I

Session 1B

Chaired by Julian Sesena, Rose Vision

GUIDE: Personalisable Multi-modal User Interfaces for Web Applications on TV

C. Jung[1], P. Hamisu[2], Carlos Duarte[3], P. Biswas[4], L. Almeida[5]

[1,2]Fraunhofer IGD, Germany; [3]University of Lisbon, Portugal; [4]University of Cambridge, UK; [5]CCG, Portugal

E-mail: [1,2]{chjung,phamisu}@igd.fraunhofer.de, [3]cad@di.fc.ul.pt, [4]pb400@cam.ac.uk, [5]luis.almeida@ccg.pt

Abstract: **This paper presents a novel software framework to adapt multi-modal user interfaces (UI) on TV platforms, which can help developers to personalize interfaces efficiently. The proposed framework adapts the UI of a web application (HTML5) according to the needs of individual users, based on a previously created user profile. The user profile is executed through a user model that represents capabilities (vision, hearing, motor functions and cognition) and preferences of users, which also makes the framework a comprehensive tool to support accessibility. Profiles can be created by the framework through a dedicated user initialisation procedure, guiding the user through a sequence of interactive tests. Based on created profiles, the framework can automatically select and configure various multi-modal UI technologies (speech, gestures, remote controls, second screen, virtual characters, graphical user interfaces, etc.). Core UI adaptation mechanism comprise fusion of input data (e.g. speech and gestures), contextual reasoning (user-, environmental states), managed user dialogs (disambiguation of low confidence input, contextual help), and configuration of multi-modal output channels (GUI, text-to-speech). In the paper we describe the architecture and its basic functional blocks of our prototype implementation and explain most important adaptation scenarios.**

Keywords: Personalisation, Multi-modal Interaction, Accessibility, User Interface Adaptation, Smart TV, HTML5, Web Applications

1 INTRODUCTION

The new Digital TV is becoming more and more an interactive media terminal in living rooms, offering exciting internet-based services on Smart TV platforms, combined with novel multi-modal interaction capabilities (like speech control, gestures), which go beyond established interaction schemes based on standard remote controls.

With new opportunities new challenges for developers arise. They need to integrate, manage and configure the new technologies, and address a wide range of users with heterogeneous capabilities and preferences. Especially for elderly users (or users with impairments) the new services introduce barriers through increased complexity and limited accessibility support. In order to support users, many design rules have to be followed and contextual support must be provided by a service, to handle erroneous user input and ensure that the user can access the UI even with degradations in vision, hearing, motor functions and/or cognitive capabilities. Commercial systems are often designed to fulfil the interaction needs of a wide majority of the society, but do not reflect individuals. Personalised configuration of the user interface (UI) is of course possible, but has to be applied manually in different contexts of the service (e.g. settings/options menus), and cannot be automatically derived from user requirements. A chosen configuration is often only valid in the scope of a specific service, and not across various services and platforms. Personalised TV configuration and adaptation of the user interface could help to enhance the experience of services, support the user during interaction and also compensate for potential functional limitations of users.

In the following sections we describe how our proposed Framework can (1) integrate and adapt various multimodal user interfaces, (2) adapt and personalise based on user profiles and (3) support integration of (legacy) web applications through a rich browser API.

2 RELATED WORK

Multimodal interfaces aim to provide a more natural and transparent interaction to users. They have been shown to enhance human-computer interaction (HCI) in many ways [1] including increasing user satisfaction, robustness and accuracy of inputs, accessibility, efficiency and reliability. These benefits can be further extended through techniques of interface adaptation, capable of mitigating some limitations of multimodal processing, while increasing the rewards of multimodal interaction. Interface personalisation through adaptation techniques has been mainly explored in the domain of content personalisation and developing intelligent information filtering or recommendation systems based on user profiles. However there are a few significant projects on interface personalisation outside the content personalisation scope and especially for elderly users. These include the SUPPLE project [2] and the AVANTI project [3] for people with disabilities. However, none of these projects has been able to cover a wide range of user characteristics as base for adaptation, and neither has offered a generalized framework for interface personalisation over a broad spectrum of output configuration possibilities. Still, UI personalisation has been found to be better than pre-customised UIs, especially for older adults [4]. The GUIDE project takes a unique approach to inclusive interfaces through a user centred design process involving user modelling. The process involves users in early design process, understands their requirements, formulates it

Corresponding author: Christoph Jung, Fraunhofer IGD, Fraunhoferstrasse 5, 64283 Darmstadt, +49 6151 155-614, christoph.jung@igd.fraunhofer.de

in user models and implements the model into a software framework to personalize interfaces with change of their range of abilities. The development process ensures adaptive interfaces allowing the elderly to easily bridge possible barriers to technology use. Based on the user profiles gathered in requirements engineering, user interfaces can be adapted and personalised in order to meet the individual user's needs.

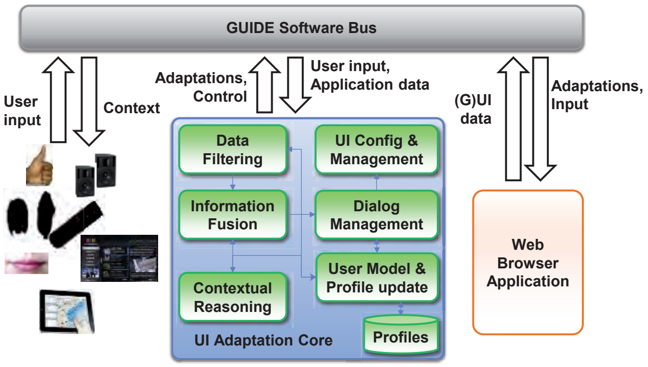


Figure 1: Framework architectural overview

3 ADAPTATION FRAMEWORK

Designing a UI Framework that manages user interaction with TV services in the living room requires building blocks that ensure the user's service experience is adequately addressed and enriched. Our UI Framework builds on an open source bus-based communication model for the deployment of services in an assisted living environment [5], and compliments it with core components realizing advanced methods for guided profile creation, user interface personalisation and user-system-level interaction concepts.

In the following we provide an overview of the basic architecture of our Framework. The Framework abstracts user interface components and web-based applications running in a web browser (see Figure 1 and 2). Applications can communicate with the Framework through a browser plugin (and JavaScript API) being connected to the GUIDE Software Bus (section 3.1). This bus concept connects the application with the central adaptation logic (Core) as well as various UI technology components (section 4).

To address tasks for user interface personalisation and concepts for handling user-system-level dialogs during user interaction with applications and TV services, our Framework defines and implements the following components: *User Model and Profiling* (for user profile data analysis and management as well as generation of user interface requirements for the application), *Fusion* and *Fission* for handling input/output (IO) data streams of different user-profile supported modalities, *Dialog Manager* (for coordinating and managing user-system-level dialogs), *Context Reasoning* modules (assisting in context-aware interaction based on rule-based logic), as well as the *Web Browser Interface* (WBI) for runtime adaptation and updates on the application's Graphical User Interface (GUI).

It should be noted that our UI Framework bus takes into consideration both cross-platform conformance issues as well as the use of standards-based languages [6,7] for data modelling and representation as well as component integration. Third party UI components can be modelled and easily integrated with our Framework through by implementing (native C++) interfaces of the bus API. Once being connected to the bus, components can publish or subscribe for messages or events, which are processed by the Framework or consumed directly by the browser application through a plug-in interface.

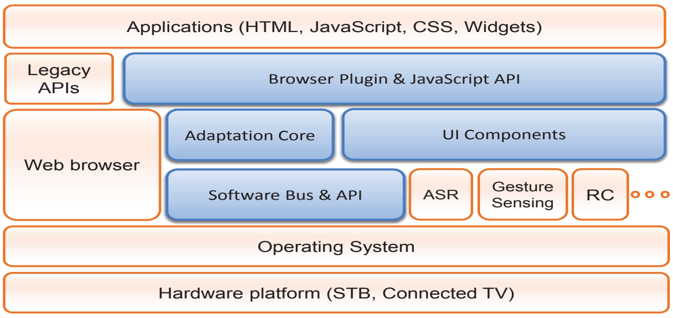

Applications (HTML, JavaScript, CSS, Widgets)
Legacy APIs
Browser Plugin & JavaScript API
Web browser
Adaptation Core
UI Components
Software Bus & API
ASR
Gesture Sensing
RC
Operating System
Hardware platform (STB, Connected TV)

Figure 2: GUIDE Framework layered architecture (Framework components in blue).

3.1 Software bus

The basic communication in our framework relies on a proven software bus architecture [5], realising a transparent inter-process and inter-device communication between components. The bus system defines two main interface roles: an event handlers interface (master) and a participating nodes (slaves) interface that publish or subscribe for messages or events on the bus. Furthermore, to facilitate data handling, we identify groups of components exchanging events and messages via three logical data buses:

Context Bus: This bus is event-based for handling contextual events. Bus publishers post context events on to the bus, while subscribers of context events, and receive notifications whenever such events are published. A context event is formulated using the standards Resource Description Framework (RDF) syntax; which is a statement modelling the subject (a given resource), predicate (the property of that subject or the resource in question) and an object (current value assumed by that property or predicate).

Output Bus: This bus is call-based for handling user interface output data. A user interface request is sent to the bus by a caller and causes an output to the user or a corresponding response/return value from the user, either immediately (synchronous) or delayed (asynchronous) through the bus from a callee.

Service Bus: This bus operates using a similar call-based model like the output bus. However, service calls on the bus have direct bindings to a specific service callee that can process the caller's request. Most often, a service callee is a shared resource that can be called by any component participating in the bus model. In our Framework design, a Virtual Character UI component is connected through the service bus, to receive and perform certain behaviour instructions (e.g. to perform animation or read aloud a text).

3.2 User interface ontology

To be extensible and to integrate additional (future) UI technologies, the Framework provides means to modelling UI capabilities and data exchanged among components. It is crucial to thereby facilitate the integration of third party and legacy user interface components. Our framework follows a semantic modelling approach for the integration the user interface components and the exchange of their associated input data. In a central registry, components can specify shared data and resources based on a standardized Ontological Modelling Language (OWL) [6] and Resource Description Format (RDF) [7].

The Framework UI ontology models various basic domain concepts, like application, UI components, data (like speech, UI representations, Virtual Character behaviours, gestures) and UI semantic mark-up, as well as the relations among them. Thereby common features of UI components belonging to the same category of input types can be refactored and modelled in an ontology shared-data. The ontology can be easily extended by defining new concepts or by deriving from existing ones. This enhances re-use and reduces application and device-dependent solutions.

4 USER INTERFACE (UI) COMPONENTS

In the GUIDE Framework, User Interface Components are multi-modal user interface technologies that implement the software bus API and send/receive event data to/from the Framework core (see Figure 1). The Framework supports various multi-modal user interface technologies, which can connect to the GUIDE core through the GUIDE software bus, thereby transparently bridging process- and machine boundaries. The UI components can describe capabilities and data exchanged by referencing to the common GUIDE UI ontology (section 3.2).

In the following we provide some examples of UI components as they are currently integrated in the Framework:

Remote controls: The framework supports standard remote controls as the well-established paradigm for interaction with TVs. Developers can retrieve data from a device API and send it to the software bus using the GUIDE C++ API, in order to benefit from adaptations. The API supports normal key data as well as gyroscopic sensor information, e.g. to move a cursor on screen. Typical adaptations for RCs comprise recognition/recovery of/from false key input and contextual cursor movement adaptation.

Speech recognition: With automatic speech recognition capabilities, the Framework can recognize voice commands from the user during interaction. Commands can be used anytime by the user, and are processed in combination with other available input data (data fusion). This allows performing true multi-modal commands, like pointing to an element on screen and triggering an activity on that element via voice. GUIDE speech recognition currently uses an industry standard ASR engine (Loquendo), but can work with any other technology. The Framework supports generation of contextual vocabularies (only relevant commands in a given UI context are considered), which improves robustness and recognition efficiency.

Virtual characters: Virtual Characters can be useful to elderly people in a variety of contexts, like for example as anthropomorphic interface agents in TV-based applications [8]. The VC employed in our framework (see Figure 5) supports nonverbal and verbal communication, enhancing the user's interaction with the application. This module enables, amongst other, the transformation of text into speech, the parameterisation of the VC gender, to adjust the volume and speed of speech or even the selection of emotional and body expressions, consistently to the defined user profile. This interactive VC technology is based on large set of pre-rendered high-quality elementary video snippets that can be combined for displaying one specific VC behaviour (verbal and non-verbal expression). This approach achieves very high quality graphical representations and animation for VC while being very efficient. This is especially required on low-end devices such as TV-embedded platforms or set-top boxes. The Framework can overlay a 3D virtual character (VC) to the legacy graphical user interface of the application, to support the user in different contexts. The VC can read aloud selectable items on screen (screen reading), e.g. for blind or physically absent users.

Visual sensors & gesture recognition: If the target platform has visual sensors (cameras) and body tracking logic available (like e.g. in Microsoft Kinect or latest Samsung interactive TV sets), the Framework can also manage it in the course of UI adaptation. It handles cursor data from the user's hand movement or various types of body gestures (head, hand, etc.). Cursor positions can be filtered according to the user's motor functions (tremor, restricted range of motion, etc.). Gestures can be automatically mapped to event handlers of the application, by mapping to semantic annotations provided by the developer (see section 5 for details).

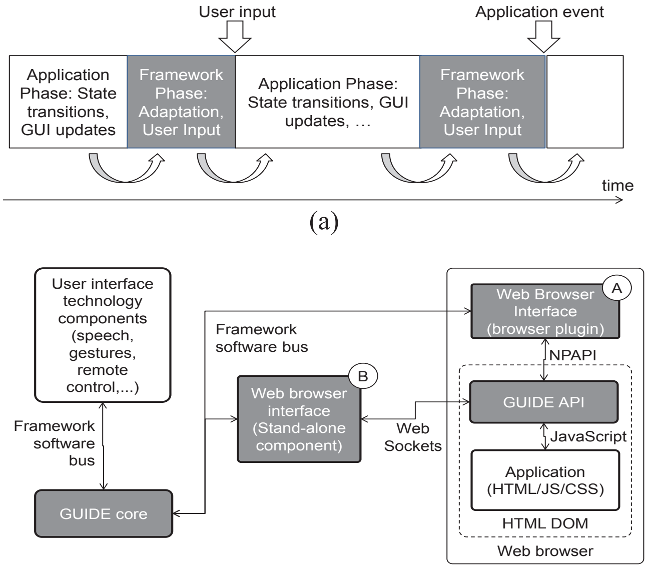
Second screens: Second screens can render specialised views of the application (from a generic UI representation in UIML), in order to implement context sensitive Remote Controls on tablet devices or mobile phones. When the second screen is used as a clone view of the TV, GUIDE can also consider the finger movement and gestures that are performed by the user on the second screen.

5 INTEGRATION IN HTML5 WEB APPLICATIONS

The proposed UI adaptation framework concept in principle supports any application environment or runtime (incl. C/C++ or JAVA), since all UI model parameters can be represented in a generic UI description (UIML). Nevertheless, due to growing acceptance and platform independence, it was decided to implement our first proof of concept for web applications. Web applications comprise HTML(5) with embedded JavaScript, CSS as well as media objects, like images and videos. They are executed in a web browser, either client- or server-side, as classic web pages or local code packages (widgets).

Web Browser Interface (WBI): The WBI (see Figure 3b) is the basic component in our framework that abstracts the application to the framework and vice versa. This is necessary since the framework must be able to work with any concrete application/UI environment.

In this sense the WBI ensures that all UI-related information that is exchanged with the framework is being mapped to the concrete HTML/JS representation in the browser. The WBI can receive events from the framework (like user input, required GUI adaptations, cursor positions, etc.) and forward data from the application to the framework (current UI representation, submission of new user profile data, etc.). The WBI can be operated as a native NPAPI plugin in a web browser, or as a stand-alone component connected via Web Sockets (see Figure 3b).


User input
Application event
Application Phase: State transitions, GUI updates
Framework Phase: Adaptation, User Input
Application Phase: State transitions, GUI updates, …
Framework Phase: Adaptation, User Input
time
User interface technology components (speech, gestures, remote control,...)
Framework software bus
Framework software bus
GUIDE core
Web browser interface (Stand-alone component)
B
Web Sockets
Web Browser Interface (browser plugin)
A
NPAPI
GUIDE API
JavaScript
Application (HTML/JS/CSS)
HTML DOM
Web browser

(a)

(b)

Figure 3: GUIDE Protocol for synchronous operation of framework and application (a), Integration of Web Browser Interface (WBI) and Framework – NPAPI (Variant A) or WebSockets (Variant B) (b)

Integration of applications: Application developers can access the WBI services through a JavaScript API. The application must follow a specific protocol (see Figure 3a). Whenever the application has finished internal state transitions ("Application phase") and requires new user input, it calls the Framework. The WBI now queries the HTML DOM for annotated elements (see next paragraph) and generates a UIML [9] representation from the elements. Now the framework core starts various adaptation processes and concurrently recognizes multi-modal user input ("Framework phase"). In this phase the WBI receives instructions from the core to modify the GUI, e.g. by manipulating elements in the HTML DOM (e.g. increase font size for a vision-impaired user). Once the Framework has recognized relevant user input (which maps to available application input slots), the Core sends this input to the WBI, which in turn maps the input e.g. to a click event that is emitted on the corresponding HTML element. A user can for example select an item on screen using voice, and thereby click the element. It should be noted that this process is absolutely transparent for the application. In addition to automatic adaptations, the application can anytime query the API for user-dependent UI parameters, to apply them manually if required.

UI semantics by annotation: The Framework needs to gain information about the applications UI to generate the universal UIML representation for Core processing. GUIDE provides various ways in its JavaScript API to make the HTML-based UI more expressive in terms of semantics. Developers can use standard mark-up (WAI-ARIA[10], HTML5 and proprietary) to add information to the bare UI elements (like HTML div, img, etc.).

6 USER PROFILES AND PROFILE INITIALISATION

The GUIDE user model, which constitutes the basis for UI adaptation in the GUIDE Framework, maps users' functional parameters to interface parameters. It was developed using a simulator [11] and calibrated through user studies [12]. The simulator consists of detailed models of visual and auditory perception, cognition and motor action. It can show the effects of a particular disease on visual functions and hand strength metrics and in turn their effect on interaction. Using information from more than 100 users from three different countries (Spain, UK and Germany), collected with an extensive survey focusing a wide array of characteristics, we have selected a set of variables that are relevant to the user model and statistically significantly different among clusters ($p<0.01$). We separately clustered these data for visual, cognitive and motor abilities of users using k-means clustering. Following this, we ran the GUIDE simulator, taking the parameters of each cluster centre for configuration, and generated recommendations for each cluster. Individual users were assigned to the recommendations based on their cluster memberships. For the present set of users we identified the following three profiles: Profile A: No adaptation required; Profile B: Mobility Impaired (e.g. increase button spacing); Profile C: Mobility Impaired and Colour Blind (e.g. increase button spacing and change colour contrast). A representative set of clusters is shown in Figure 4 below.

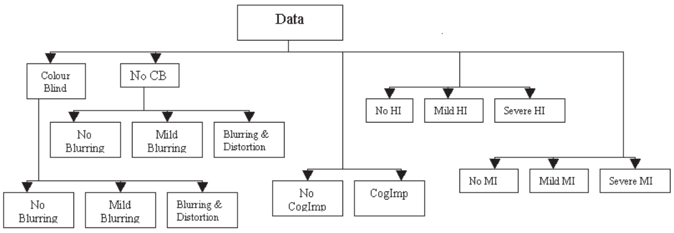
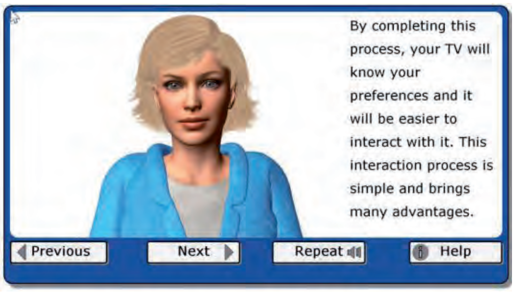


Figure 4: Clusters based on range of abilities of users

The GUIDE User Model predicts three sets of parameters: (1) UI parameters for the Multimodal Fission Module, (2) Adaptation Code for the Input Adaptation Module and (3) Modality Preference for the Multimodal Fusion Module. The rules relating the users' range of abilities with interface parameters were developed by running the simulator in Monte Carlo simulation. The user model predicts: the minimum button spacing required, from the users' motor capabilities and screen size; foreground and background colour schemes from user's colour blindness assessment; required use of pointer correction techniques (applied by the input adaptation module); and the best modality of interaction for a specific user, albeit users are always free to interact through their modalities of choice.

Guided user profile creation: The GUIDE Framework also introduces the concept of the User Initialization Application (UIA, see Figure 4). The UIA is an introductory application that is presented to the user (once) on first usage of the system.

When a new user logs in (or is automatically recognized by sensors), the UIA at first presents a step-by-step introduction of the system, acting as a tutorial on how to use the system and how to benefit from different modalities available. Secondly, the UIA expedites the user profiling procedure by gathering data on the user's capabilities and preferences in a sequence of interactive tests and questions. Since it would be hardly feasible to ask masses of end users to complete an extensive survey before using a product, the UIA presents a much reduced set of questions and tasks to the user in order to allow the User Model to assign the user to one of the previously created profiles. Even though this does not allow for a profile perfectly fitted to the user, it delivers a good starting point for adaptation purposes, because the profiles described in the previous section were created from a large pool of representative users. Additional information collected during system usage is used to refine the user profile (run-time adaptation).



By completing this process, your TV will know your preferences and it will be easier to interact with it. This interaction process is simple and brings many advantages.

Figure 5: User Initialisation Application (UIA) – Guided tutorial and user profile creation.

The tasks and metrics chosen for the UIA are the ones for which the resulting data is the most capable to assign the more appropriate profile to the user profile. They were selected from an analysis of the extensive survey data, taking into account the feasibility of gathering the data. For those instances where it was not feasible to gather the data in a living room environment, alternative sources were selected and combined to estimate the required data. These variables include: **Colour Blindness**: Plates 16 and 17 of Ishihara Test [8] as it may classify among Protanopia, Deuteranopia and any other type of colour blindness; **Dexterity**: We estimated Grip Strength and Active Range of Motion of wrist from age, sex and height of users following earlier Ergonomics research [13] **Tremor**: We conducted earlier a test involving a Tablet device in horizontal position, and estimated tremor from the average number of times users need to touch the screen to select small buttons. Additionally, other tasks were chosen with the purpose of allowing users to personalize the system, while being a hands-on tutorial regarding new modality interaction and feedback configuration. The most relevant ones are the following: **Modality Introduction**: Self-explanatory videos of how to interact with each modality, followed by "do-it-yourself" tasks; **Button and Menu Configuration**: Button size, and font and background colour configuration; **Cursor Configuration**: Cursor size, shape and colour configuration; **Audio Perception**: Hearing capabilities and preferences.

The UIA has a simple user interface (Figure 5), with a different screen for every task and metric identified above. Few buttons are presented per screen (preventing user confusion). Every screen preserves the same navigation model - an area with "next", "previous" and "repeat" buttons, and another visually distinct area for presenting information and requests. For every metric to be measured, tests are presented as simple questions about preferences. Also, for every modality available in the system, a video introducing its use is presented, followed by the possibility for the user to try it out. A virtual character accompanies the user through this process, offering explanations and assisting the user in the personalisation. As the user goes through each task and preference setting, the UIA adapts itself to the preferences already manifested.

7 USER INTERFACE ADAPTATION AND MANAGED INTERACTION

The major added value of the proposed software framework is its adaptive multimodal layer, implemented by the Framework core components. The core consists of two groups of components:

(1) Adaptation: The first set of components implement multimodal adaptation algorithms for processing and managing input and output data as well as to adapt parameters across modalities for a given user profile. These include: the *Input Adaptation* module for filtering and manipulating a sequence of continuous user input data such as cursor positions from a computer mouse; the *Fusion Module* that is responsible for interpreting inputs from different input devices into meaningful commands for a given application; the *Dialog Manager* manages changes in the application state in the framework and supports the output bus in updating and signalling adaption parameter changes on the abstract UI representation property of an output event; and the *Fission Module* that is responsible for preparing and coordinating the multimodal rendering of content to the user.

(2) Context and user management: A second set of components manages context information and user profile data. The *Context Model* stores and manages context events generated by different context sources as well as implements some rule-based logic for reasoning on context data for given situational changes. With the User Initialisation Application, the GUIDE system starts collecting data about users to make possible any kind of adaptation. The User initialization application consists of a list of tests represented in a game-like fashion to users. It collects data on basic visual, cognitive and motor skills of users and also their preference about several interface properties. The data is sent to the *user model* component to extract a basic profile for the user. The user model predicts interface parameters based on range of abilities of users. For example, it predicts minimum button spacing required from the users' motor capabilities (like presence of tremor, spasm or weakness of arms) and screen size and colour contrast based on the presence and type of colour blindness of users.

Basically adaptation at runtime can be triggered by one of two main events: (1) an input event from the user and (2) an output event caused by application state changes (see section 5). The former causes the user model component to perform recalculations on its profile clusters in order to create a matching user profile with the desired preferences; while the latter is managed through an interaction of the Fission and the Dialog Manager on the abstract UI representation property of an output event on the output bus.

7.1 Adaptation scenarios

Based on a given user profile and the adaptation components in the Core, several adaptation scenarios can be supported, involving different UI components as well as combinations of them (in concurrent use):

User input disambiguation: If the user enters erroneous or low confidence data to the Framework, it can be in a first step fused with other modalities (e.g. speech + gestures) to increase accuracy by joint recognition. Further, when data confidence remains low, the Framework can notify the user to repeat the input with a visual message on the screen.

Adaptation of pointer data: Pointer data could be delivered by gyroscopic remote controls or body tracking- and gesture control systems. These UI component technologies can send pointer input data to the Core, which selects an appropriate adaptation strategy for the cursor data. This could be smoothing of cursor positions or “gravity wells” to make the cursor being attracted by UI elements.

Contextual reasoning & support: The Framework retrieves contextual data from UI components, such as the Visual Human Sensing (VHS) module, based on visual sensors. It can identify the user from facial information, and generate information on presence or level of attention. The Context model applies reasoning to such events to derive new context events (of higher semantic level). The Framework Core can also recognize situations where the user is unable to proceed in a certain UI context. Indication for such situations can be repeated erroneous input, or missing expected input by the user. The framework can then provide help to support the user recovering from this situation, either based on the framework managed virtual character, or simply by multi-modal on-screen dialogs (text, audio). Typical recovery strategies comprise interaction tutorials or navigating back to a known start (“home”) place.

Intent recognition: The Framework can identify erroneous or incomplete user input in specific cases, and map it to input which is relevant in the current context (e.g. recognition of repeatedly pressing a wrong key on a remote control, or accidently clicking in a region around a button on the screen).

GUI adaptations: Based on the generic UI representation of the applications UI (from HTML to UIML) and the user profile, the core makes decisions about required visual adaptations. This could be for example a change of font size, colour schemes or reduction of complexity. The required adaptations are propagated to the WBI, which either applies the changes directly in the HTML DOM of the application, or notifies the application to handle the adaptations. This also allows for further more comprehensive manipulations of the GUI, e.g. re-layouting or adding/hiding information.

Virtual characters (VC): The VC employed in our framework supports nonverbal and verbal communication, enhancing the user’s interaction with the application. This module enables, among other, the transformation of text into speech, the parameterization of the VC gender, to adjust the volume and speed of speech or even the selection of emotional and body expressions, consistently to the defined user profile. This interactive VC technology is based on large set of pre-rendered high-quality elementary video snippets that can be combined for displaying a specific VC behaviour (verbal and non-verbal expression). This approach achieves very high quality graphical representations and animations for VC while remaining very efficient. This is especially required on low-end devices such as TV-embedded platforms or set-top boxes.

Second screen devices: The Framework supports second screens (tablet PCs) in different roles. A second screen can either (1) be adapted as a GUI supplement of the main TV application, or (2) can connect to the framework as an auxiliary input device or adaptive remote control. The second case allows for example to render alternative views from the main UI on the TV.

CONCLUSION

In this paper we have presented a novel Framework for multi-modal adaptation of user interfaces for TV-based web applications. Our implementation integrates various kinds of user interface technologies, and provides managed interaction for users based on individual user profiles. We described the overall architecture of the Framework and highlighted the major innovations, like automatic UI adaptation (multi-modal, profile-based), guided profile creation (user initialisation), extensibility (software bus, ontology, UIML), cross-platform and application support (HTML applications, JavaScript API), and described the most important scenarios for adaptation. Future work will comprise extensions to the functionality of Core adaptation mechanisms, with dedicated focus on establishing a managed supportive dialog with the user, which remains transparent to the application.

References

[1] Sharon Oviatt, Trevor Darrell, and Myron Flickner. 2004. Multimodal interfaces that flex, adapt, and persist. Commun. ACM 47, 30-33.

[2] Gajos K., et al. Automatically generating user interfaces adapted to users' motor and vision capabilities. UIST 2007.

[3] Stephanidis, C. et al: Adaptable and Adaptive User Interfaces for Disabled Users in the AVANTI Project. S.Triglia et al. (Eds.): IS&N’98, 153-166, 1998

[4] Kim, J., Pan, Y., McGrath, B. “Personalization in Digital Television: Adaptation of Pre-Customized UI Design”, In EuroITV2005

[5] Tazari, M. An open distributed framework for adaptive user interaction in ambient intelligence, AmI Conference 2010

[6] “OWL Web Ontology Language Reference: Detailed description of the OWL W3C specification”, w3c, 2004

[7] “Resource Description Framework (RDF): Concepts and Abstract Syntax”, W3C Recommendation, 2004

[8] Colour Blindness Tests 2008. Available at: http://www.kcl.ac.uk/teares/gktvc/vc/lt/colourblindness/cblind.htm, 2008

[9] Abrams, Phanouriou, et al., “UIML: An appliance-independent XML user interface language”, 1999

[10] WAI-ARIA-1.0, Accessible Rich Internet Applications, W3C candidate recommendation. 18.01.2011

[11] Biswas P., Langdon P. & Robinson P. (2012) Designing inclusive interfaces through user modelling and simulation, IJHCI, Taylor & Francis, Vol; 28, Issue 1, 2012

[12] Coelho J., Duarte C., Biswas P. and Langdon P., Developing Accessible TV Applications, Proceedings of ASSETS2011

[13] Angst F. et. al.., Prediction of grip and key pinch strength in 978 healthy subjects, BMC Musculoskeletal Disorders 2010.

GrafiTV: Interactive and Personalized Information System over Audiovisual Content

Ismael de Fez[1], Jon Arambarri[2], Pau Arce[1], Francesc Arribas[3], Sergio Barrera[2], Igor Bilbao[4], Eduardo Burgoa[3], Juan Carlos Guerri[1], Patricia Ortiz[4], Edgar Vaz[4], David Zaragoza[5]

[1]Universitat Politècnica de València, Valencia, Spain; [2]VirtualwareLabs, Vizcaya, Spain; [3]Aido, Valencia, Spain; [4]Innovalia Metrology, Vizcaya, Spain; [5]Avanzis, Valencia, Spain;

E-mail: [1]{isdefez, paarvi, jcguerri}@upv.es, [2]{jarambarri, sbarrera}@virtualwaregroup.com, [3]{farribas, eburgoa}@aido.es, [4]graphitv@datapixel.com, [5]dzaragoza@avanzis.com

Abstract: **This paper presents the GrafiTV project, which aims at developing an interactive and personalized information system over future audiovisual content in real time. The system is intended to be used by new connected TVs and mobile devices with multimedia capabilities, such as tablets or smartphones. The system allows the personalized insertion of additional information through graphics, therefore, synchronization on audiovisual content as well as object detection over images in real time is required. This paper presents the system architecture and describes the main technologies used by the platform in the areas of tracking, transmission, synchronization and visualization.**

Keywords: Interactivity, personalization, tracking, synchronization, multimedia system, IP networks, 3D graphics

1 INTRODUCTION

Nowadays there is an increasing need to obtain specific information through the most accessible device at any time and to show that information in the most attractive way. A good usability and operability of the devices is also an important requirement, as well as an appropriate and customizable interface to display the information. Current mobile devices, such as smartphones or tablets allow Internet access through browsers or specific applications. This way, it is possible to obtain immediate information over different topics.

In this sense, the television has experienced a great growth in the last years. Current connected TVs allow Internet connection and have a huge number of applications and contents. Thus, users can interact with the television, which was unthinkable just few years ago.

Some of the most important advantages of television are the large number of people that use it and its simple utilization. This fact has motivated the appearance of widgets, which are applications specifically designed for television. Currently, most new televisions include applications with different functions such as video rental or web page access to display content. Furthermore, television manufacturers have agreements with television channels to offer their video on demand (VoD) service.

Current widgets show their interface and the content requested by the users over the audiovisual content, thus reducing the vision of the live content. Moreover, this additional information appears in a fixed position and is shown according to the scheduling (that is, in a certain time) and the user cannot add or remove this information from the display.

In this sense, the GrafiTV project aims at developing a system that allows to add additional information over the audiovisual content in a personalized way for each spectator.

The platform offers a wide variety of possibilities. One of the most useful scopes is the broadcasting of sports events. By means of the platform, a spectator can view statistical information about a certain sport event and place this information in different positions of the screen. For instance, if a user is watching a basketball match and wants to see the number of personal faults that a certain player has made, the system will show the number of faults around that selected player. Thus, spectators can obtain statistical information in real time without waiting for the content provider to show this information. Figure 1 illustrates an example of the application, where statistical information is shown over each player together with information about a specific player.

Furthermore, the insertion of personalized advertising on the platform is another of its main applications. This way, it is possible to insert personalized advertisements and information according to the spectator habits and preferences while a live sport event is being displayed, without disturbing its viewing. Also, the platform allows to show information about products that appear in series, films or advertisements and, even to access to a specific interface in order to buy the product.

There are different projects that work in this direction. Thus, the CustomTV project allows users to access to different information and multimedia sources simultaneously. In this sense, the objective of the NoTube project is to develop an end-to-end architecture for the personalized creation, distribution and consumption of TV content, where users

Corresponding author: Ismael de Fez, Institute of Telecommunications and Multimedia Applications (Universitat Politècnica de València), Camí de Vera s/n 46022, Valencia (Spain), +34 963879588, isdefez@iteam.upv.es

could control their data. More information about these two European projects (within the 7th Framework Program) can be found in the CORDIS official website [1].

The rest of this paper is organized as follows. Section 2 describes the technologies used. Section 3 explains the platform architecture. Finally, conclusions and future work are shown in Section 4.

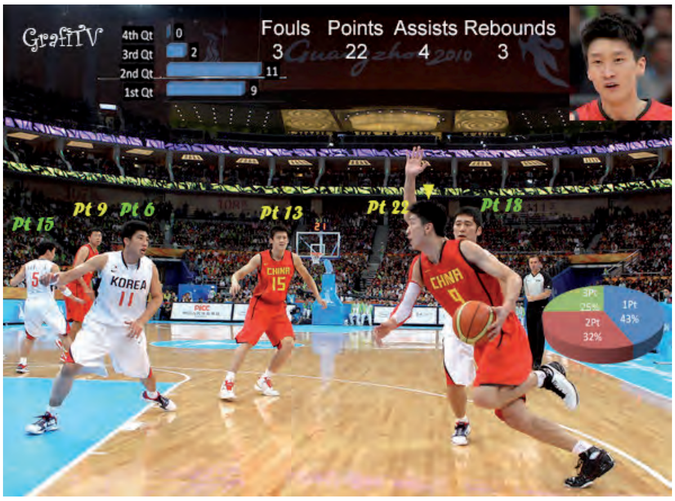


Figure 1: Example of a GrafiTV use case

2 TECHNOLOGIES

GrafiTV combines graphics insertion over audiovisual content, object detection over images in real time and synchronization with the channel emission displayed at any time.

In order to carry out these functionalities, the platform uses different technologies of several fields such as tracking, transmission and synchronization over IP networks as well as content reproduction and display. Next sections explain the main technologies used by the platform.

2.1 Tracking

One of the main interactive visualization problems in real time is the detection and later tracking of the objects by means of a sequence of images.

The choice of a detection algorithm depends on the features of the object to be detected. In case of detecting fixed images, such as the number of a player, the SURF algorithm stands out [2]. It consists of making copies of the image so as to look for the points that are contained in all the copies, guaranteeing the scale invariance. Figure 2 shows an example of pattern detection using the SURF algorithm.

When willing to detect the face of a player, face detection techniques are used [3], which focus on searching different faces on an image and, later on, identifying the specific person. Moreover, features of the object have an important influence (colour, level of details, geometry, light conditions…). Some of the most relevant tracking techniques are the Lucas-Kanade (KLT) method [4] and the movement pattern estimators. KLT method is based on the assumption that the flow is essentially constant in the pixels surrounding the pixel under consideration. Once it is recognized, the movement pattern estimators estimate the possible future movement (if the goal is to follow an object or a person). The performance of these estimators is based on two phases. In the first one (predictive), information learned in the past is used in order to find the position in which is likely to find the object or the person. In the second stage (corrective), a measurement is performed, which is used to correct the predictions made in the previous phase.

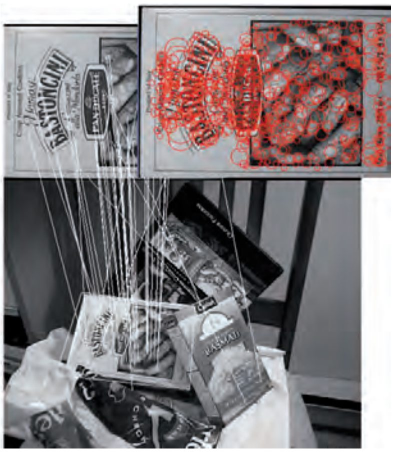


Figure 2: Pattern search with SURF algorithm

Table 1 shows the main current working areas in detection and tracking. As can be seen, different algorithms are used depending on the type of shot: medium, close-up, and long. Therefore, if a player needs to be detected in a medium shot, the Watershed or Mean-Shift algorithms, together with Delaunay triangulation and classifiers, are the most popular techniques. However, if there is a close-up of the target player, the Lucas-Kanade technique and movement pattern estimators are the most appropriate ones, as explained. Finally, as it has been previously stated, facial detection and SURF algorithms are employed in order to track a player face or a player representative object.

Activity name	Group	Main techniques
Players detection	Medium Shot	* Watershed algorithm * Mean Shift algorithm * Delaunay triangulation * Classifiers
Players detection	Close-up	* Lucas-Kanade method * Movement patterns * Estimators
Players tracking	Long Shot	* Facial detection * SURF algorithm

Table 1 Tracking techniques

2.2 Transmission and synchronization

Regarding transmission and synchronization of the multimedia content, we highlight two main approaches to deal with the enriched content. On the one hand, the broadcast approach, with a strong emphasis on the synchronization. On the other hand, the Internet approach, oriented to a flexible framework that allows to offer any kind of service through browsers with scripting capabilities. Since these two approaches finally converge, it is needed to find solutions compatible with both scenarios, in order to take the most of both approaches.

In broadcast technologies, the composition of virtual or synthetic elements with real video is defined in scenes. As far as the description of these scenes concern, there are different standards referred to the spatial and temporal synchronization of the objects synthetically generated and the video flows. Among these standards, DIMS (Dynamic Interactive Multimedia Scenes) [5] defines scenes, formats and interactivity mechanisms with the user. Moreover, MPEG-4 BIFS [6] defines a binary format to transmit additional information together with the audio and video flows. Thus, the receiver is able to display all the information synchronously. Other meaningful standards are SMIL (Synchronized Multimedia Integration Language) [7] and LASER (Lightweight Application Scene Representation) [8]; the latter also belongs to the MPEG-4 standards. BIFS, DIMS and MPEG-LASER consider the use of the DSM-CC (Digital Storage Media Command and Control) protocol for object transportation over MPEG-TS, but also consider the use of IP protocols such as RTP (Real Time Transport Protocol), HTTP (Hypertext Transfer Protocol), and FLUTE (File delivery over Unidirectional Transport), as well as video, audio and data encapsulation in multimedia files. These standards allow the description of scenes with enriched contents, as well as support for spatial-temporal synchronization.

On the other hand, Internet technologies such as HTML5 allow to combine different formats and content sources on a single user interface. In this sense, CE-HTML, the reference industrial standard for connected TVs, uses HTTP for the transport of enriched contents or RUIs (Remote User Interfaces), the term used on that specification. The RUIs overlap the audiovisual content received by means of a DVB modulator integrated in the television. Moreover, the video tag used in HTML5 is agnostic to the transport protocol and the video container used. Each browser will offer support for a certain group of protocols, containers, codecs, formats, etc. HTML5 and CE-HTML have also different functionalities to manage the communication among the components coming from different sources through sockets. Also, HTML5 allows to monitor the events triggered by the player that manages the video tag. Last generation browsers support video elements and advanced graphics formats such as SVG. However, the native support for 3D graphics is still under development.

Nowadays, broadcast transmissions arrive through television arrays and intelligent applications use data networks to obtain additional information. A good synchronization between the broadcast contents and the applications will allow a better display interface and a greater adaptability. That synchronization and a good knowledge of the content by the application will improve the Quality of Experience (QoE) of the user.

2.3 Graphics generation and display

Currently, graphics insertion over television broadcasting is made through characters generators, which insert animation and texts. This insertion of extra information is determined by the broadcaster or content provider, not by the user. Thus, the time and place where the additional information is inserted for being displayed is decided in the production stage. Currently, there are no systems that carry out an automatic synchronization with the audiovisual content.

Among the several technologies used to generate and display 3D graphics and information over audiovisual content in real time, we find WebGL, O3D and XML3D.

WebGL [9] is a standard specification to display 3D graphics in web browsers. Through WebGL it is possible to display hardware accelerated 3D graphics on the browser without installing additional software. It works on any platform that supports OpenGL 2.0 or OpenGL ES 2.0.

On the other hand, O3D [10] is an open source web API used to create interactive 3D graphic applications running in the browser. O3D allows the creation of several 3D graphic applications such as games, virtual worlds or 3D model viewers.

In contrast to the previous solutions, XML3D [11] is a web standard that does not require a deep knowledge about 3D programming. XML3D is an extension to HTML5 that allows interactive 3D graphics and attempts to achieve maximum compatibility with both HTML5 and XHTML. XML3D takes advantage of several W3C standards and recommendations, such as XML (Extensible Markup Language), DOM (Document Object Model) or XBL (XML Binding Language).

2.4 Communication interfaces

The use of a standard mechanism to exchange information is necessary when communication between different technologies occurs. In this sense, Service Oriented Architecture (SOA) [12] is a framework that provides a good scalability and flexibility.

SOA is a software architecture concept that defines the use of services to support the requirement of the user software. SOA provides a methodology and a framework to support the integration and consolidation activities. In a SOA environment, network nodes make their resources available to other participants as independent services, accessing in a standardized way. Most of the SOA definitions identify the use of web services (using SOAP and Web Services

Description Language, WSDL) in their implementation. Nevertheless, it is possible to implement a SOA architecture using any technology based in their services.

Also, SOA architectures are made of application services imperceptibly connected and highly interoperable. This communication is based on a formal definition independent of the underlying platform and the programming language (e.g., WSDL). The interface definition makes the architecture independent of the manufacturer and the developed technology. Thus, the software components developed are very reusable, since the interface is defined following a standard. For instance, a C# service could be used by a Java application.

3 ARCHITECTURE

3.1 Block diagram

Figure 3 depicts the GrafiTV architecture. Three main elements are shown: applications, managers and repositories. The communication between the different elements is made by means of interfaces.

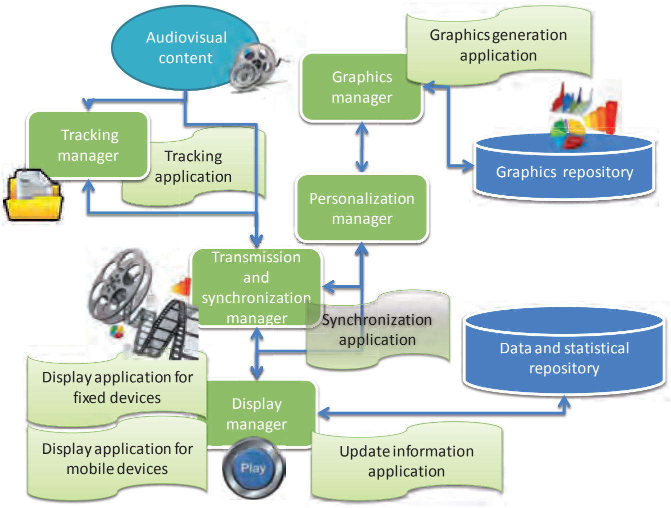

Audiovisual content
Graphics generation application
Graphics manager
Tracking manager
Tracking application
Graphics repository
Personalization manager
Transmission and synchronization manager
Synchronization application
Data and statistical repository
Display application for fixed devices
Display manager
Display application for mobile devices
Play
Update information application

Figure 3: GrafiTV architecture

Users can interact with the platform through the client application. This user interface application allows the user to select the information and the statistics to be displayed, and how these will be shown. The display manager offers a set of templates to the client application and accommodates the raw data information sent together with the media flow.

If the user requires any change about additional information of an event (such as the display mode), the display application communicates with the display manager, which retrieves information from the personalization manager. The latter is the only manager with a direct communication with all the platform managers. The personalization manager processes all the information sent by the user through the display manager and requests and sends information to the other managers.

Depending on the user request, the personalization manager communicates with the appropriate managers to comply with the request and provide the proper information in order to display the multimedia content on the screen.

As aforementioned, the communication between the different modules shown in the architecture is performed using SOA, through web services with HTPP/SOAP protocols. Different service APIs are defined for each module of the platform in WSDL format.

The following sections explain more deeply all the blocks of the platform.

3.2 Graphics generation

The graphics generation application generates the graphics of the GrafiTV system and stores them in the graphics repository, which contains the format of the graphics to show. The application also recommends the suitable graphics for each use case by means of the graphics manager. That manager selects the most appropriate graphic depending on the information to show or generates the graphic if that does not exist. The platform has been specifically designed to support the generation and the display of 3D graphics.

3.3 Tracking

By means of tracking, a real time image processing is carried out, which allows the identification and positioning of objects in the image. This way it is possible to insert the information that user selects on the screen through 3D graphics, which will make it easier for a human viewer to process the information and to know what is going on in real time.

The tracking application carries out the patterns, people or objects detection, identifying them in the video. The application is able to track the movements of the selected element during the video frames. As a result, within each image, the object position is obtained. This positioning information will be useful to locate the information related to the object or person detected so that synthetic objects can be positioned accordingly.

On the other hand, the tracking manager is in charge of knowing the pattern that the tracking application must search. Moreover, the tracking manager provides the tracking application with the images or the video so that the application can look for the information on the video. The tracking manager obtains the positioning information of the desired element in the frame to be shown.

3.4 Transmission and synchronization

The transmission and synchronization application transmits all the information selected by the transmission and synchronization manager to the display manager, taking into account the synchronization of the enriched information with the video. The kind of information sent must be understandable by the display manager, which must be able to show this information on the screen independently of the device where the application is running.

Also, the transmission and synchronization manager synchronizes the additional information together with the audiovisual content displayed using the metadata encapsulated with the video flow.

3.5 Playback, display and information updating

The client application displays all the information received by the display manager according to the selected criteria. It performs user interface functions to display the GrafiTV settings menu and also receives user interactions for the system. The client application contains the player that displays the enriched content that has been generated over the media stream. Since the system must support both fixed and mobile devices, it is necessary to develop different user interfaces, which will be centralized in the display manager. The GrafiTV system employs a browser player for viewing media content through the Internet on connected TVs and mobile devices. Client applications have a direct communication with the updating application to update the information shown on the screen.

In addition, the updating application provides the values of the information to show and informs about changes or updates from the information and statistics repository. In order to have up-to-date information at any time, the updating application connects with the different information providers. In this sense, the statistical information repository contains all the information that can be shown by the system. The repository is updated with information from the event as it happens. The client application gets the information every time it is updated or when it is necessary. The repository of statistical information may have different providers to obtain all the information needed, both real-time and statistical information.

The display manager is responsible for implementing the visualization of the audiovisual content, including the combination of the main media stream with all the synthetic graphics and information requested by the user on the screen.

3.6 Personalization

The personalization manager communicates with all the platform managers to make the GrafiTV system work. The personalization manager processes all the user requests, triggered through the user interface. Furthermore, it is in charge of storing all the preferences from the users of the GrafiTV system. The personalization manager has a direct communication with the different managers:

- With the graphics manager. The personalization manager requests information about new graphics requested by the user. The graphics or objects to be shown will depend on the information requested, the display mode, the application and the device where the application is running. The graphics manager will recommend the appropriate graphics for every case taking into account different parameters such as the screen size, the resolution, the information to display, etc. Thus, a proper configuration will avoid overloading the screen with excessive graphics or with unreadable graphics. The graphics manager is in charge of taking these considerations into account when it generates new graphics or when it demands pre-created objects within the graphics repository.
- With the tracking manager. When there is a need to change any pattern to identify on the video, the tracking manager detects special patterns of the different events in order to locate the information referred to them in a nearby point. The tracking application detects patterns and objects whereas the tracking manager is in charge of choosing the pattern to find depending on the content displayed.
- With the transmission and synchronization manager. The personalization manager informs about all the information that must be sent to the display manager, as well as the graphics and additional data to be sent. Some examples of additional data to be included are the selected display mode, the data to display, the state of the GrafiTV system, the time when the enriched information must be inserted and the position where the enriched information must be displayed. All this information will be processed by the display manager and shown on the screen through the display application.
- With the display manager. When the user requires additional information of a certain event, the personalization manager is in charge of serving the requested information.

4 CONCLUSIONS AND FUTURE WORK

This paper has presented the GrafiTV project, a multimedia system that improves the Quality of Experience of the users since it provides interactivity and personalization services. The system is a step forward on the information society demands since it offers interactive information services, which are synchronously merged with audiovisual content. The platform is compatible with both fixed and mobile devices. Currently, the project is in the development stage, so it needs to be validated and evaluated by end users, which is part of the future work. Thus, one of the future objectives is the development of this service on current smart televisions platforms. The goal is to develop a widget or application integrated in televisions.

On the other hand, another future objective of GrafiTV is to allow users to create additional repositories to those presented in the platform. In these repositories users would include their own information and comments, thus improving the customization of the services.

Acknowledgements

This research has been funded by the Spanish Ministry of Industry, Tourism and Trade under the GrafiTV project (TSI-090100-2011-173).

References

[1] CORDIS: Community Research and Development Information Service, http://cordis.europa.eu, 2012.

[2] W. Kai, C. Bo, M. Lu, and X. Song, "Multi-source remote sensing image registration based on normalized SURF Algorithm," in Proc. Int. Conf. on Computer Science and Electronics Engineering (ICCSEE), Hangzhou, China, March 2012.

[3] P. Viola and M. J. Jones, "Robust real-time face detection," International Journal of Computer Vision, vol. 57, no. 2, May 2004.

[4] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proc. of the 7th Int. Conf. on International Conference (IJCAI), vol. 2, San Francisco, USA, 1981.

[5] 3GPP TS 26.142 v10, "Dynamic and Interactive Multimedia Scenes (DIMS)," March 2011.

[6] ISO/IEC 14496-11, Information technology- coding of audio-visual objects, "Part 11: Scene description and application engine," 2005.

[7] W3C, "Synchronized Multimedia Integration Language (SMIL 3.0), 2008.

[8] ISO/IEC 14496-20, Information technology – coding of audio-visual objects, "Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)," 2009.

[9] C.E. Catalano, M. Mortara, M. Spagnuolo, and B. Falcidieno, "Semantics and 3D media: current issues and perspectives," Computers & Graphics, vol. 35, no. 4, pp. 869-877, August 2011.

[10] O3D Project Page from Google Code, available: http://code.google.com/p/o3d/.

[11] XML3D official website, available: http://www.xml3d.org.

[12] M. Bell, "Introduction to Service-Oriented Modelling", Service-Oriented Modeling: Service Analysis, Design and Architecture, Wiley & Sons, pp. 3, 2008.

SocialSensor: Surfacing Real-Time Trends and Insights from Multiple Social Networks

Sotiris Diplaris[1], Giorgos Petkos[1], Symeon Papadopoulos[1], Yiannis Kompatsiaris[1], Nikos Sarris[2], Carlos Martin[3], Ayse Goker[3], David Corney[3], Joost Geurts[4], Yaning Liu[4], Jean-Charles Point[4]

[1]Information Technologies Institute, Thessaloniki, Greece; [2]Athens Technology Center, Athens, Greece; [3]Dept. of Information Science, City University London, UK; [4]JCP-Consult SAS, Cesson-Sévigné, France

E-mail: [1]{diplaris, gpetkos, papadop, ikom}@iti.gr, [2]n.sarris@atc.gr, [3]{Martin.Carlos.1, Ayse.Goker.1, David.Corney.1}@city.ac.uk, [4]{joost.geurts, Yaning.Liu, pointjc}@jcp-consult.com

Abstract: **This paper presents the first steps towards implementing a vision of a real-time system that aims to incorporate emerging knowledge from social media. In order to achieve this, crawling techniques that are specially designed for addressing the particularities of social web are being developed in terms of the SocialSensor FP7 project. Subsequent steps in the pipeline involve suitable mining to extract relevant information from social media streams. The surfaced information is presented using an interface that is optimized and adapted to the context of the user, taking into account efficient content delivery techniques to optimize quality of experience. First data collection and research approaches are discussed, particularly focusing in social media analytics, multi-modal learning for social event detection and efficient social content delivery in mobile settings. We also discuss how these new methodologies will be incorporated to build novel tools for news and infotainment.**

Keywords: social search, sensor mining, social data analytics, content delivery, news, infotainment

1 INTRODUCTION

The latest developments in the use of the web and mobile devices have transformed the way that media content is created, edited and distributed. Media content is created and published online at unprecedented rates by both regular users and professional organisations. The wide availability of smart phones has enabled the creation and instant sharing of media content at the time and place of an event.

At the same time, social networks have become an integral part of modern life driving more and faster communication than ever before. For politics, business and leisure these new connections are shaping our world.

In this context, the challenge for traditional information providers is to use and embrace these new content authoring and provision methods, and the channels offered by social media and mobile technologies, to their fullest advantage, in both information gathering and information distribution. A key challenge in this respect is to develop appropriate tools for quickly surfacing trends, sentiments and discussions in relevant and useful ways.

To get the most out of the content residing in social networks, a number of challenges are as yet unsolved. These include the following: (a) Verification: ensure that the content posted in social networks is accurate; (b) Filtering: according to particular needs and interests; (c) Sensing: discover trending topics and what is "up and coming" in order to guide further investigation; (d) Analysis: analyze particular trends and tendencies according to specific questions; (e) Visualisation: present search results in an attractive, easy to understand way; (f) Cross-platform issues: enable searches across different Social Media platforms; (g) Speed: time is money, all processes need to happen quickly and efficiently, without being at the expense of accuracy; (h) Usability: tools and interfaces should be intuitive and easy to use.

SocialSensor is a 3-year FP7 European Integrated Project aiming to tackle some of the challenges outlined above and to offer solutions as well as improvements. It is developing a new framework for enabling real-time multimedia indexing and search across multiple social media sources, introducing the concept of Dynamic Social COntainers (DySCOs), a layer of multimedia content organisation with particular emphasis on the real-time, social and contextual nature of content and information consumption. Through the proposed DySCOs-centred media search, SocialSensor will integrate content mining, search and intelligent presentation in a personalised, context and network-aware way, based on aggregation and indexing of user-generated and multimedia web content. It will be a single platform consisting of practical tools that incorporate novel user-centric media recommendation, visualisation, browsing and delivery methods.

The means to improve the user experience on the web goes through the automatic understanding of information streams, through a *sensor mining* procedure. SocialSensor attempts to bring new mining techniques for intelligently merging the content coming from different sources and performing analysis on the aggregate representation, effectively managing the arrival of large, heterogeneous and evolving data.

Such data are captured, represented, indexed and searched from social and web sources to provide relevant, and context-aware results for multimedia and text content in a real-time manner using novel *social search* approaches. The concept of DySCOs is central for social search due to the need for an bridge to organise information between the context-based search needs of information consumers and the indexing and aggregation capabilities of large-scale data stores.

Corresponding author: Sotiris Diplaris, Information Technologies Institute, 6th Km. Charilaou-Thermi road, Thessaloniki, Greece, +302311257778, diplaris@iti.gr

In order to leverage the user experience in a real-time environment, the content delivery and quality of service aspect is also considered in the project framework. The *semantic middleware* of the SocialSensor platform allows ad hoc networked users to seamlessly discover, compose and share semantically-relevant multimedia data and services. For this purpose, it includes components for (a) semantic peer-to-peer selection and composition planning of data services that are relevant to a given query; (b) semantic query answering over continuous streams of potentially inconsistent social data; and (c) intelligent caching, pre-fetching and web-based sharing of data in *ad hoc* user groups.

Finally in the *user modelling* layer, in order to reflect different information needs of users, a user and context model for long-term profiling and short-term activities is created and algorithms and tools for personalised information delivery and recommendations based on user feedback are developed.

The resulting multimedia search system will be showcased and evaluated in two use cases: news and infotainment.

The news use case targets two end user groups: (a) news professionals that are interested in leveraging social media content in their work; and (b) casual online and mobile news readers. The aim is to build applications that will provide these users a new way of discovering and accessing news information hidden in social media. With respect to professional usage, different scenarios will be supported, such as discovery of emerging trends and topics; the aggregation of social media with professional content; the analysis of massive amounts of social data for new insights; and the profiling of news portal users to aid the recommendation of relevant content. Casual news readers will benefit from innovative features, such as real-time discovery of news items, proactive delivery (push) of relevant content to them based on their context, and socialisation of user with other news reader through ad hoc social networking.

The infotainment use case targets individuals attending large events, such as festivals and expos. The aim is to build for these users mobile applications that will help them organize their visits to such events by providing context aware information using an enhanced quality of service network. By leveraging the user's context for search, the physical surroundings of a user acts as a lens on the social media content that relates to her current activities, location, and physical or social ties. SocialSensor will deliver mobile tools supporting diverse usage scenarios, such as context-triggered multimedia search, proximity-based real-time activity recommendation, facilitation of social networking aspects, and real-time interaction with the event acts.

Providing real-time social indexing capabilities for both of these use cases is expected to have a transformational impact on both sectors. The subsequent chapters outline some of the research challenges on which SocialSensor focuses, as well as some early results stemming from the first individually implemented research methodologies and the use cases requirements gathering phase.

2 METHODS

2.1 Search and Indexing with DySCOs

In Figure 1 we depict the generic conceptual architecture of SocialSensor in the highest level of abstraction, connecting the content sources, its components and the users.

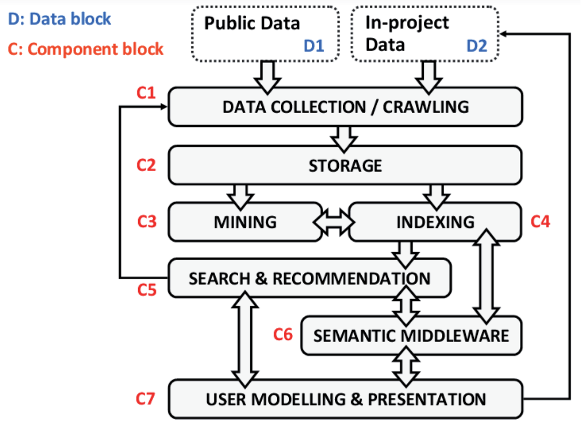


Figure 1: SocialSensor generic conceptual architecture.

Currently, online content is indexed and searched at an atomic level, i.e. each content item is processed and indexed independently of the rest of the collection. SocialSensor will extend this paradigm by performing indexing and search over composite objects relating to a common topic of interest. Such composite objects are called DySCOs. The benefit of using DySCOs over single items is that it will be possible to extract aggregate knowledge and inferences by analyzing them as a collection. In addition, performing the indexing at a collection-level is expected to enable richer representation of contextual information with respect to content, i.e. the indexing mechanism will be able to access contextual information about content items. In this sense DySCOs can be defined as composite topic-centred objects that encode contextual and inferred information about collections of content items (automatically detected to be related to the given topic of interest). DySCOs and their attributes are created as a result of Sensor Mining methods. Indexing of DySCO fields and relations is task for the Social Search component. Transfer, composition, and packaging of DySCOs take place in the Semantic Middleware. Querying and retrieval of DySCOs is taken care of by the Semantic Middleware and Social Search components. DySCO-based recommendation takes place in the context of Social Search and User Modelling. Figure 2 depicts the typical lifecycle of a DySCO. There are two main stages involved: (a) creation and maintenance; and (b) search, delivery and presentation.

2.2 Social data collection

The use of social media in applications such as journalism and infotainment is becoming increasingly important. For example social media has significantly changed the nature of breaking news, putting pressure on editors on what to broadcast or publish, as well as changing the emphasis on being the first to

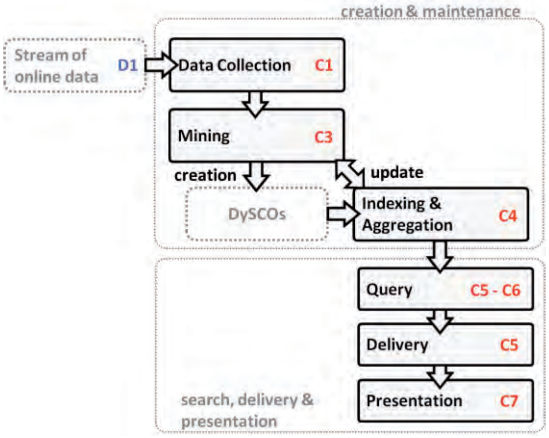


Figure 2: DySCO lifecycle. For each block, its code refers to the conceptual architecture of Figure 1.

break new stories to the importance of verifying content [1]. The Social Search component will use web analytics to examine user behaviour and feed this information to search algorithms that will provide real-time context-aware search to meet user needs (e.g. verifying content). Support for search will be provided by indexing low-level multimedia content and by using social media to establish links between content items. Indexing data extracted from different sources will be aggregated. These components will support multimedia filtering and content (e.g. meeting the requirements of journalists for image and/or video related to the story they are working on).

One of the major political events of 2012 will be the US presidential election on November 6th. We will be collecting and analysing data from online social networks in real-time during the election period. For journalists, tweets may contain important stories or links to important multimedia content, such as videos or images on YouTube, Facebook etc. Although the widespread use of online social networks is relatively new, studies have already demonstrated important analysis of tweets about German elections [2] and the Arab Spring [3], as well the use of Facebook activity to understand previous US primary elections [4].

This task is closely related to several of the challenges presented in the Introduction: filtering by the interests of the journalists and their research areas; sensing trending topics and breaking news from Social Media; the analysis of the evolution of different topics of interest to the users of SocialSensor; and cross-platform issues regarding data collection from different Social Media platforms.

2.3 Multimodal learning in social content

Once social data are crawled and collected, the next step is to perform analysis over them with the goal of *sensing* trending topics and events. SocialSensor deals with data that is inherently *multimodal*. Content originating from social networks, for instance a Flickr image, is often associated to textual, visual, temporal, spatial and social information. Multimodality is a challenging issue in machine learning and often requires the use of data fusion techniques. In order to detect real-world events from social multimedia content, a clustering approach has been examined, in which social content items are clustered and each resulting cluster is treated as a single real-world event. The result of a clustering algorithm on multimodal data depends heavily on the weights put on different modalities, when using either a late or an early fusion approach. For instance, if location data is weighted more heavily for computing overall item distances, the resulting clusters may represent spatial hotspots, instead of real-world events. In order to overcome this issue, we propose an explicit supervisory signal that will guide the clustering process towards forming clusters with the desired semantics [1][5]. The procedure is displayed in Figure 3 and is outlined below. At the outset, a relevant dataset that captures the nature of the desired clustering is collected, e.g. a set of Flickr images that display a set of known real-world events. Distances are computed for each modality and for each pair of data points of the example clustering. The distances for all modalities are compiled in a single vector for each pair of data points and each vector is labelled as positive or negative depending on whether the two corresponding data points belong to the same cluster (event) or not. This dataset is used to train a classifier that predicts the "same cluster" relationship between two data points. In order to cluster a new dataset, the "same cluster" relationships between all pairs of items are computed and these are used to form an "indicator vector" that summarizes the "same cluster" relationship between a single item and all other items. Finally, since it is expected that items that belong to the same cluster will have similar indicator vectors as other items of the same cluster, the indicator vectors are clustered using some generic clustering algorithm (e.g. *k*-means or spectral clustering) in order to obtain the final clustering result.

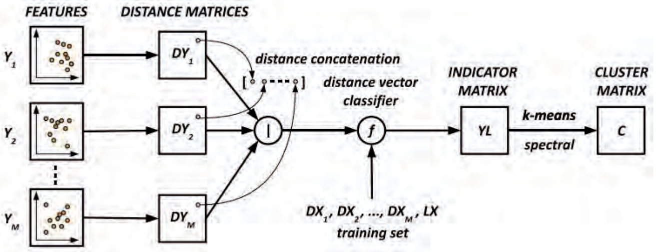


Figure 3: Framework of proposed multimodal clustering using a supervisory signal.

The advantage of this approach is that there is no need to search for explicit fusion weights or to apply special fusion techniques. The dataset used to train the classifier should be sufficiently diverse to enable automatically determining the appropriate fusion strategy.

2.4 Efficient content delivery via CCN

The SocialSensor framework aspires to enable real-time multimedia indexing and search of the social web. Both indexing and search require highly efficient content delivery to live up to its speed objective. However, the architecture of the internet is optimized for point-to-point communication, which is less well suited for content distribution. As a result, content delivery often suffers from long latencies and popular content in particular gets congested due to too many simultaneous requests. A Content Centric Network (CCN) distributes content efficiently by caching content when it traverses the network. If a second request is made for the same content it can be retrieved from an approximate cache (created during the first request) instead of fetching it from the source as happens in the current internet architecture [6]. Instead of locating content through IP addresses, CCN decouples content and location, and implements the mapping between them via routing. A content request is issued by an "interest" packet. This Interest is routed to the "closest" content, and the source responds with a Data packet along the reverse path to the requestor. Each CCN node consists of three components: Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB). A CCN node keeps data in CS, and forwards and stores the interests in FIB and PIT. PIT enables CCN to meet the simultaneous demands, for example, in live streaming. When several clients ask for the same piece of content (i.e. the same interest) at the same time, PIT sends the request for this interest only once, and keeps track of other interest requests. As soon as the data packet that satisfies the interest arrives, it will be sent to all requests in PIT. The CS caches the forwarding content, such as a caching-along-default-path design, to satisfy any future requests. When a client asks for a piece of content, this request propagates through the network until the content is found somewhere. When the content is returned to the requesting client, the content is cached along the past path. Consequently, when a second client that is physically close to any CCN node along the past path, asks for the same piece of content, it can be quickly retrieved from one of the neighbouring caches. CCN can also initially support mobility. A re-location of a CCN mobile user simply re-issues previously sent Interest packets that have not been satisfied yet. Let us assume that a mobile user requests content in a frequent handover scenario. The requested content is cached in a router close to the requesting mobile node. Hence, the content can be retrieved efficiently as soon as the mobile node reconnects to the network. CCN is relatively new network technology that, so far, has been developed mainly in the domain of network research. SocialSensor as such provides an ideal use-case to validate (and adapt) the theory and technology in the context of social networks and media sharing. In addition, the strong focus on context in SocialSensor opens the possibility of improving the quality of service (QoS) for the real-time social web applications such as audio and video streaming in mobile usage scenarios. An implementation of video streaming over CCN network is further described in Section 3.3.

3 EXPERIMENTAL RESULTS

3.1 Social data analytics

As a pilot study, we collected and analysed tweets related to the US Republican party primaries, i.e. elections held in individual states where members of the Republican party vote for their choice of presidential candidate. On Tuesday 6th March 2012, or "Super Tuesday", ten US states held primary elections. We selected tweets containing any of the four candidates' names in "hashtags" or "mentions"; relevant keywords such as #SuperTuesday; or the names of any of the ten states. This crawl produced a set of 284,732 tweets (with 100,832 distinct users tweeting) that matched one or more of these hashtags or mentions during a 72-hour period (from March 5 to March 8) centred on Super Tuesday. These were obtained through the Twitter "spritzer", which is the default access level that is publicly available to third-party developers. We analysed the frequency of mentions of candidates and states across time to consider the temporal dimension of the collection. One aim is to measure the participation of Twitter users before, during and after the primary elections and to determine the most frequently named candidates and states in each period of time. Although this analysis was carried out after the tweets were collected, the intention is to automate the process and provide real-time analysis of future events. There is not space here to provide a full analysis of the results; instead we highlight some interesting observations. Unsurprisingly, the number of relevant tweets posted rose to a peak on the day of the election. To allow a more detailed analysis, we normalised the distribution by calculating what fraction of tweets in each hour were associated with each candidate and with each state. Figure 4 shows the normalised distribution of tweets per candidate and state over time. This example shows the proportion of tweets about one candidate (Newt Gingrich) in three states (Oklahoma, Georgia and Alaska). For example, the central peak of the graph shows that in one hour more than 4% of all tweets in our collection posted in that hour mention both Newt Gingrich and Georgia. The use of such normalised distributions allows us to consider the temporal dimension of tweets and improves the detection of interesting topics (e.g. trending topics or breaking news). For example, on the evening of the election, Sarah Palin (the former governor of Alaska) announced that she had voted for Gingrich some minutes after the polls were closed. Figure 4 shows a corresponding spike in Twitter traffic mentioning

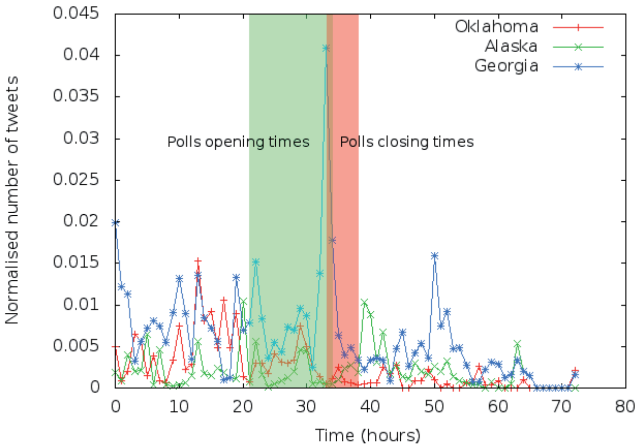


Figure 4: Normalised distribution of the tweets per hour for candidate Newt Gingrich in three different states. The period covers Monday 5th March 10:20 -- Thursday 8th March 12:00 (EST time).

both Gingrich and Alaska in subsequent the hours (39-44[1]). This suggests that important real-world events can be associated with characteristics of Twitter traffic. Our ongoing work is to automate this process, namely identifying spikes in Twitter activity and then providing explanations of these spikes in terms of real-world events. This will include consideration of the manner in which information spreads across the Twitter network [7].

3.2 Evaluation of multimodal learning

The multimodal clustering technique presented in Section 2.4 was tested with data from the MediaEval Social Event Detection 2011 challenge [8]. In particular, 2074 images corresponding to 36 events were used. Ten runs were executed and in each run the photos corresponding to a random subset of the events were used to train the classifier, whereas the rest were used for testing the clustering procedure. The set of features that was used includes upload time, location (available only for one fifth of the data), SIFT features and TF-IDF weights that were computed on the tags of the images. The proposed approach was compared to a multimodal spectral clustering approach that fuses modalities by computing an aggregate affinity matrix as the weighted sum of the affinity matrices that correspond to each modality. A search in the space of possible weights is performed for the competing approach. The results are presented in Table 1. Normalized Mutual Information (NMI) was used as the evaluation measure. The proposed approach significantly outperforms the baseline. Additionally, it does not require explicit selection of fusion weights, as the supervisory signal directly determines the appropriate fusion mechanism.

Table 1: Average and standard deviation of NMI achieved by the two tested methods for the 10 runs and on average.

Run	*Baseline*	*Proposed*
1	0,2996±0,1855	0,6737±0,1892
2	0,2701±0,1560	0,7181±0,1504
3	0,2869±0,1657	0,7051±0,1784
4	0,3051±0,1787	0,7248±0,1231
5	0,2859±0,1534	0,7005±0,1220
6	0,2863±0,1688	0,6843±0,1486
7	0,2992±0,1901	0,6956±0,1576
8	0,2389±0,1264	0,6258±0,1453
9	0,2468±0,1396	0,7067±0,1354
10	0,2500±0,1533	0,6323±0,2137
Avg.	**0,2769±0,1643**	**0,6867±0,1619**

3.3 Implementation of DASH on CCN

Globally, video traffic will be 55% of all consumer internet traffic in 2016, up from 51% in 2011 [14]. As such, it is vulnerable to bottlenecks in the network, which results in poor QoS for users of the SocialSensor application. Dynamic Adaptive Streaming over HTTP (DASH) is an emerging standard for adaptive streaming over HTTP, developed by ISO/IEC [9]. SocialSensor is performing research on DASH and provides a range of tools for DASH (e.g., DASH VLC Plugin, DASHEncoder, libdash, MPEG-DASH MPD Validator, DASH-JS) [10][11]. DASH is primarily designed for HTTP and follows the approach of chunk-based HTTP streaming [12]; however, the modular design of DASH allows HTTP to be substituted by CCN. This optimizes the transportation of content by reducing latency and possible congestion. DASH supports segmented IBMFF (ISO Base Media File Format) [13] which divides the media file into segments. These segments are described in a so-called Media Presentation Description file (MPD). To allow switching between representations of the media content, it has to be transcoded to different bitrates and resolutions. At runtime, the DASH client selects the most appropriate segment given the current user context. Figure 5 presents the architecture of DASH over CCN (components that are adapted with respect to the traditional DASH architecture are highlighted). When the DASH client requires a particular segment, a CCN Interest message is generated by the CCN Access component, which is an instruction to download the respective segment. Finally, the Interest messages are satisfied by video segments that are stored in the network (not necessarily retrieved from the same host). The adaptation of DASH to work on CCN proved to be relatively straightforward, which is a promising result for the adaptation of other components developed in the project. We are currently in the process of testing the effect on the quality of service of DASH over IP and DASH over CCN. In addition, the segmentation of video fragments, which is currently computed *a priori* by a DASH encoder, could also be dynamically adapted to the network conditions using CCN, which is particularly designed for fragmentation. This may reduce the overhead cost necessary to setup a connection.

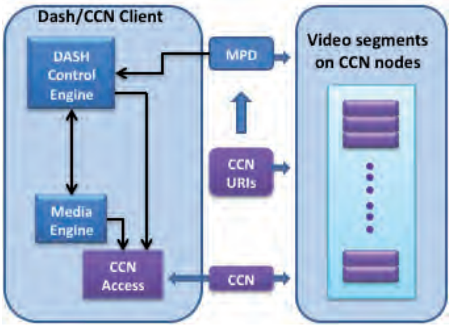


Figure 5: DASH over CCN architecture.

4 ENVISIONED USE CASES

4.1 Verification: The SocialSensor Alethiometer

We are currently experiencing a social web explosion, which is giving the power of speech back to the citizens who had been practically deprived of this since the gradual explosion of the population that made it impossible for news to travel via the old channel of the ‘word-of-mouth’. We now live

[1] 7/3/12 1:00 – 7/3/12 6:00 (EST time) on 7/3/12.

through the new phenomenon of the 'e-word-of-mouth', where information travels at rapid pace and in huge volumes through tweets, posts and blogs. This could be an opportunity for direct access to information coming from first-hand sources, as it was happening centuries ago in small societies. The problem is that the scales are now very different and every side has its say in this "big bang" of gossiping: truth and lies, positive and negative, genuine and fake. As explained in the previous sections, we are trying to discover and organise information hidden in social media as unfolding events. In addition to this however, we will also provide a way for the users to assess the credibility of information found in social media. 'Alethia' is the Greek word for 'truth' and in SocialSensor we are developing the 'Alethiometer': a module attempting to measure the credibility of information coming from any social source. We will deliver this measure for verification through what we call the 'triple-C' structure: we will provide credibility metrics under the following three categories: (a) *Contributor* – We will provide data about the source of information and try to calculate trust, reputation and influence of this source; (b) *Content* – We will try to show whether the content is reliable by checking things like the language used, the history (has it been posted before?) and possible manipulations (has an image been altered?); (c) *Context* – We will try to indicate whether the 'what', 'when' and 'where' can contextualise together. The results from the above steps will be weighted and combined together to provide a metric that will give a sense of truth to the user, while giving access to all the evidence that leads to this result, so that the user can also subjectively decide on how credible every piece of information is. Although this is work in progress, we have mostly identified the methods and metrics that will be used in the implementation phase.

4.2 Enhancing user experience in Infotainment

Recently it has become common for large scale infotainment events, be it film or music festival, large expo, sport or scientific event, to be supported by a mobile application. However, a competitive analysis of such applications showed that they do not currently offer intelligent features, while personalisation is often limited to merely allowing the user to create personalised event schedules. SocialSensor envisions an intelligent real-time system that will incorporate advanced social media search and analysis features, as well as enhanced visualisations and contextual recommendations, in order to deliver relevant and timely content to the event attendant, and to ultimately leverage the user's event experience. To further assess the need for intelligent mobile applications for infotainment events, SocialSensor was engaged to analyse the needs of event attendants. In order to reach a wide range of end-users and gather more informed feedback, an iPhone application [2] incorporating the previously identified basic functionality was designed, developed and released for the 14th Thessaloniki Documentary Festival, during March 2012. The analysis of user responses revealed that intelligent features such as recommendations based on user ratings, connection with social media, personalized schedules, and user group formations for peer-to-peer content sharing are very useful and would enhance the user experience. Also, complex services are being designed in order to satisfy the needs of several event user groups that involve joint actions. Feedback from the users will guide the implementation of the second prototype application, to be made available in the upcoming Thessaloniki International Film Festival in November 2012.

5 CONCLUSION

In this paper we have outlined and highlighted some of the scientific challenges that have to be confronted in order to create a platform for real-time social media content indexing, search and delivery. SocialSensor aspires to provide such tools for professional journalists, casual newsreaders and attendees of large infotainment events by investing in innovative analysis techniques of social sensors, assisted by effective indexing of real-time social media streams. Preliminary social data collection and analytics, and technical attempts to tackle some of the identified issues were also presented along with promising initial research results.

References

[1] N. Newman. "The rise of social media and its impact on mainstream journalism", Reuters Institute for the Study of Journalism, Univ. Oxford, 2009.

[2] A. Tumasjan, T. O Sprenger, P. G Sandner, I. M Welpe. "Predicting Elections with Twitter: What 140 Characters Reveal About Political Sentiment". Proc. 4th Int. AAAI Conf. on Weblogs and Social Media, 178–185, 2010.

[3] A. Younus, M. A. Qureshi, F. F. Asar, M. Azam, M. Saeed, N. Touheed. 2011. "What Do the Average Twitterers Say: A Twitter Model for Public Opinion Analysis in the Face of Major Political Events". In 2011 Int. Conference on Advances in Social Networks Analysis and Mining, 618–623.

[4] C. Williams, G. Gulati. "What Is a Social Network Worth? Facebook and Vote Share in the 2008 Presidential Primaries". In Annual Meeting of the American Political Science Association, 1–17, 2008.

[5] G. Petkos, S. Papadopoulos, Y. Kompatsiaris. "Social Event Detection using Multimodal Clustering and Integrating Supervisory Signals." In Proc. ACM Int. Conference on Multimedia Retrieval (ICMR), Hong Kong, 2012

[6] V. Jacobson, D.K. Smetters, J.D. Thornton,M.F. Plass, N. Briggs, R. Braynard. "Networking named content". Proc. of the 5th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2009), Rome, Italy. NY: ACM, 1-12, 2009.

[7] D. Romero, B. Meeder, J. Kleinberg. "Differences in the Mechanics of Information Diffusion Across Topics: Idioms, Political Hashtags, and Complex Contagion on Twitter." Proc. 20th Intl. WWW Conference, 2011.

[8] S. Papadopoulos, R. Troncy, V. Mezaris, B. Huet, I. Kompatsiaris. "Social Event Detection at MediaEval 2011: Challenges, Dataset and Evaluation". In MediaEval 2011 Workshop, Pisa, Italy, 2011.

[9] ISO/IEC 23009-1:2012, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats"

[10] C. Müller, C. Timmerer, "A VLC Media Player Plugin enabling Dynamic Adaptive Streaming over HTTP", ACM Multimedia, Scottsdale, Arizona, November 28, 2011.

[11] DASH at ITEC/Alpen-Adria-Universität Klagenfurt, http://dash.itec.aau.at (last access: May 2012)

[12] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Standards and Design Principles", ACM Multimedia Systems, San Jose, California, USA, Feb. 133-143, 2011.

[13] ISO/IEC 14496-12:2005, "Information technology — Coding of audio-visual objects — Part 12: ISO base media file format"

[14] CISCO, "Cisco visual networking index: Forecast and methodology, 2011-2016," CISCO, Tech. Rep., May 2012

[2] http://thessfest.socialsensor.eu/app/

Immersive Autostereoscopic Telepresence

Mathias Johanson[1], Kjell Brunnström[2]

[1]Alkit Communications, Mölndal, Sweden; [2]Acreo, Kista, Sweden

E-mail: [1]mathias@alkit.se, [2]kjell.brunnstrom@acreo.se

Abstract: **A major shortcoming of traditional videoconferencing systems is that they present the user with a flat image of the other participants on a screen, while in real life, our binocular visual system gives us a three-dimensional view of the persons we are interacting with. Other common problems impairing the realism and usability of interpersonal video communication include lack of eye contact and a general feeling of a technology-induced barrier between the participants. In this paper, we present the development of a telepresence system based on novel concepts and new technology to give the users a sensation of being immersed in a shared space, while being geographically distributed. Key elements of the system are multiple cameras, autostereoscopic displays and a chroma keying based immersion technique combined with an eye contact mechanism. Our preliminary usage tests with the prototype system indicate that the novel mechanisms have a great potential of improving the feeling of presence in videoconferencing sessions.**

Keywords: Telepresence, videoconferencing, stereoscopy, multimedia, communication.

1 INTRODUCTION

Recent technological advances in video and display technology together with decreasing cost of high bandwidth communication links have improved the opportunities to realize very high quality videoconferencing systems that give the users the impression of being physically present at the same location. Sometimes the term *telepresence* is used to denote high quality videoconferencing systems, where great care is taken not only in the design of the communication system itself, but also with respect to the physical environment of the installations, including displays, furniture, lighting and integration of technical equipment in the room. Several videoconferencing system vendors today can provide systems supporting high definition (HD) video, which together with large size displays can give a reasonable experience of telepresence. However, these systems are often lacking in many respects and do not give the users a particularly strong feeling of being physically present at the same place or immersed in the same environment. One major shortcoming is that the systems present the user with a flat image of the other participants on a screen, while in real life, the binocular human visual system gives us a stereoscopic three-dimensional view of the persons we are interacting with and the environment they are immersed in. Another common problem is difficulties providing eye contact between the users since the cameras are most often placed above the screens. Moreover, while smarty designed, existing systems rarely blend in with the environment in a good enough way to make the technology transparent to the users. This creates a technological barrier between the participants which limits the interactivity, spontaneity and naturalness of interpersonal communication.

Recent technological progress, particularly regarding display technology, now makes it possible to realize the next generation immersive telepresence systems, giving the users a strong feeling of being present at the same place where they can interact freely and effortlessly. In this paper, we present the development of a prototype system which is based on novel use of autostereoscopic display technology and chroma keying based immersion techniques, which we hope will give a hint of what the next generation telepresence systems have to offer.

2 IMMERSIVE TELEPRESENCE

The idea of immersing the users of a communication and collaboration system in a common virtual space first appeared within the virtual reality (VR) research community [1, 2, 3]. The technology used to realize the immersion in the early VR systems was based on that time's state-of-the-art 3D graphics systems in combination with head-mounted displays or large projector-based visualization systems such as the CAVE system [4], and often also some form of tracking mechanism to detect the users' positions. In these first generation immersive VR systems, the users were represented by 3D models, known as avatars, much like today's first-person shooter computer games. These systems were expensive and the hardware for the immersive visualization was bulky and awkward to operate. Moreover, the realism of the virtual spaces left a lot to be desired, due to performance limitations of the 3D graphics systems available at the time, and the fact that the avatars were not lifelike renderings of the users.

To improve the realism, mixed reality [5] systems appeared, combining the use of 3D graphics with live video of the participants. These systems also had severe performance problems, both for video processing (generating 3D textures out of video in real time) and due to bandwidth restrictions in the communication networks of the time. Another obstacle was the problem of combining stereoscopic visualization with videoconferencing, since the available stereo visualization systems all required bulky eyewear, making it impossible to see the users' eyes.

Corresponding author: Mathias Johanson, Alkit Communications, Mölndal, Sweden, +4631675543, mathias@alkit.se

Although these efforts initially showed a lot of promise in terms of supporting high quality immersive interactions between distributed individuals, the complexities of the systems and immaturity of the technology resulted in collaborative VR systems largely being abandoned for most applications. Instead, traditional videoconferencing technology has experienced a tremendous uplift during the last decade, both for high-end professional use and low-end semi-professional or home use. This pretty much appeared to be the end of immersive communication and collaboration systems.

With the gradual improvements of videoconferencing technology and the increasing bandwidth available in communication networks, the systems evolved into high performance collaboration studios. In this context, the term telepresence was popularized a few years ago, denoting videoconferencing installations where great care has been taken with respect to the physical environments, e.g. screens, furniture, lighting, to give the appearance of the users being present in a single virtual room. This brings us once again back to the concept of immersing the users, although with a slightly different technological basis, driven by video communication technology rather than 3D and VR technology. To be truly immersive, stereoscopic visualization techniques, providing true depth perception through stereopsis are required, but as previously discussed, this has hitherto been difficult to combine with videoconferencing, due to the need for specialized eyewear. Notwithstanding the recent improvements of shutter glasses technology and the possibility of using passive stereo with less expensive eyewear, a successful combination of stereoscopic visualization and videoconferencing requires auto-stereoscopic displays, i.e. displays that can realize stereoscopic visualization without the need for glasses. Fortunately, autostereoscopic display technology has improved dramatically lately, and can now support multiple views in high resolution. This is one of the key elements of the prototype system described in this paper.

It is worth pointing out that in the original use of the term "immersion," dating back to the VR and augmented reality era, the users of the communication and interaction system are immersed in a technologically created virtual space, wherein the users are typically represented by synthetic 3D avatars. For immersive telepresence systems of the kind we will explore in this paper, on the other hand, the technological representations of the users (e.g. screens displaying video of users) are immersed in the real world.

3 PROTOTYPE SYSTEM DEVELOPMENT

The goal of the prototype development presented in this paper is to serve as a proof of concept for the next generation immersive autostereoscopic telepresence, and to provide the possibility of performing experiments and user tests with high quality autostereoscopic video communication. Although some experimental stereoscopic telepresence systems have been developed and reported in literature [6, 7], this is still an emerging research area that will need much more experimental work before commercial systems can be expected on the market.

The prototype system was developed by extending an existing videoconferencing and collaboration system called Alkit Confero [8] with support for multiple HD video streams and autostereoscopic visualization.

3.1 Multiple HD video streams

In the prototype system, two video signals are captured from two HD cameras at 1280x720 resolution, and independently encoded in software as two separate H.264 video streams, packetized and multiplexed using the Real-time Transport Protocol (RTP) and the RTP profile for H.264 encoded video [9]. The possibility of using a multiview codec (e.g. H.264/MVC) was considered not to give enough bandwidth reduction to motivate the added complexity. This decision was further substantiated by subjective tests of video quality with and without the use of multiview codecs [10].

The streams are transported over UDP/IP to the destination, where they are demultiplexed, decoded and rendered.

3.2 Autostereoscopic rendering

An autostereoscopic display is a display that can support stereoscopic vision without any eyewear. Most of the stereoscopic display technologies in the consumer electronics segment are based on either active or passive polarizing glasses that filter out the left and right video images for the left and right eye respectively. As previously noted, however, the use of eyewear is seriously prohibitive for telepresence, since it occludes the eyes of the users, effectively preventing eye contact.

Although the quality of state-of-the-art autostereoscopic displays has improved considerably over the last few years, the recent technology development trend has focused on applications such as digital signage, with slightly different requirements compared to the telepresence application we are focusing on here. Most autostereoscopic displays are based on the multiview lenticular lens technology, which in essence means that a sheet of small lenticular lenses are mounted in front of the LCD panel of a display, which refracts the light from the RGB subpixels of the display differently depending on the viewing angle. The fact that the human eyes are horizontally translated in relation to each other makes it possible to render the subpixels of the images in a way that makes the left and right eye see different subsets of the subpixels from a fixed viewpoint. For our telepresence application, the two video streams from the two cameras can thus properly rendered give a stereoscopic impression. However, most commercially available autostereoscopic displays support more views than two, which is the minimum to support stereoscopic perception. For a telepresence application, this at first seems like a complication, since more than two views will require more than two video cameras at the sender side, which is no problem in principle, but will make the system unnecessarily complex, bandwidth demanding and expensive. The decision was therefore taken to stick with a two camera configuration. However, two-view autostereoscopic displays are somewhat hard to find, especially at large sizes. A requirement for the displays of our telepresence system, except for being autostereoscopic, is that they must be big enough to display the upper part of a human body in scale 1:1. This means at least 40" screens, preferably 50". The display finally chosen for our prototype system, a 47" autostereoscopic LCD display from Alioscopy, supports eight views. To be able to display the two video signals stereoscopically on this eight view display, the two video signals are mapped to the two centermost channels of the screen and then the six other views are automatically generated from the two original streams by a signal processing algorithm that shifts the apparent viewpoint of the video by a horizontal translation consistent with the estimated head-motion of the observer. The result of this is a true stereoscopic view when the user has his or her head centered in front of the screen, and an emulated (i.e. computed) 3D view when the head is moved to the sides. This improves the experience of 3D immersion for the user compared to the two-view only situation, since the movement of the head in the latter case does not shift the perspective of the rendered scene, which gives an unnatural sensation. However, the computer generated views can appear a bit distorted, since the changes in perspective due to a head shift is difficult to compute.

The implementation of the rendering of the two incoming video signals is a combination of the signal processing algorithm mapping the two camera views into the eight displayed views with the multiview autostereoscopic rendering algorithm proposed by van Berkel et al. [11]. Since the rendering must be done in real time, the performance aspects of the implementation have to be considered. On a modern CPU, however, our implementation can accomplish the rendering with enough processing power left over for the other CPU demanding parts of the system (mainly video compression, decompression and chroma keying).

3.3 Chroma keying and stereoscopy

As discussed above, the use of an eight view autostereoscopic display prompted the automatic generation of six video views based on the two video streams transmitted from the far end cameras. This algorithmic video signal generation inspired the idea to use a chroma keying technique to combine the live video of the person in front of the cameras with a purely synthetic 3D view of the surrounding room. Chroma keying is a video processing technique used heavily in television and movie studios, whereby a video signal of a person (or any object) in front of a blue (or green) screen is preprocessed so that the blue (or green) background is replaced by another still or moving picture. If done properly, this can give the illusion of a person being immersed in a synthetic environment or located at a different place. By placing a blue screen behind the users of the telepresence system, the background can be substituted at the receiving end by the chroma keying algorithm. Traditionally, chroma keying is performed as a preprocessing step at the sender side, but in the present case it is done at the receiving end, as a post-processing step, after the two video streams are mapped to the eight channels needed for the autostereoscopic rendering. This makes it possible to key in any selected image independently for each rendered view. The procedure is illustrated in Figure 1 below.

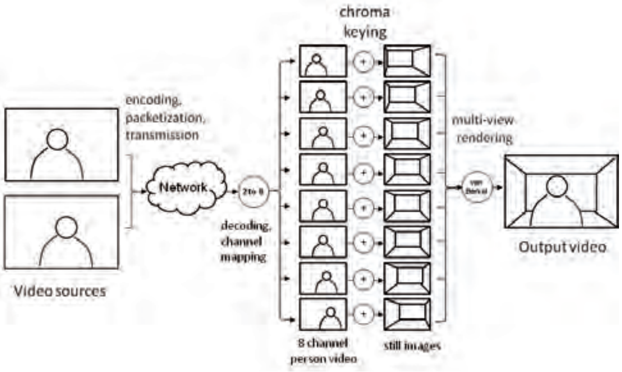


Figure 1: Receiver-side multi-channel chroma keying. The two source video streams from the far end cameras are decoded and mapped into 8 channels representing 8 viewing positions. Chroma keying is then performed on each channel independently, and the resultant signals are rendered on a multi-view autostereoscopic display using the van Berkel rendering algorithm.

The images to be keyed in can be either computer generated (synthetic 3D environment) or based on photographs taken from eight properly positioned camera viewpoints. Regardless of how the images are produced, the synthetic or photo-based environment surrounding the rendition of the person (or persons) should be designed in a way that makes the stereoscopic visual cues coherent. For instance, the depth cues must be consistent, so that objects partially occluded by the rendered person are perceived as being farther away than the person; otherwise conflicting depth cues will effectively ruin the illusion of stereopsis and immersion. Other visual cues, such as relative size of objects must also be considered.

In the case when the multi-view keyed-in background is generated from photographs, an interesting opportunity is to take the background photographs at the same location where the receiving side telepresence system is installed, from eight viewpoints located in a line through the position of the viewer's eyes, with the view directed towards the center of the display. Figure 2 shows the arrangement.

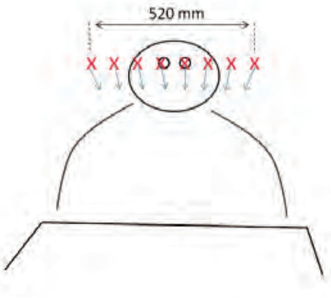


Figure 2: Camera positions for 8-channel viewpoint configuration.

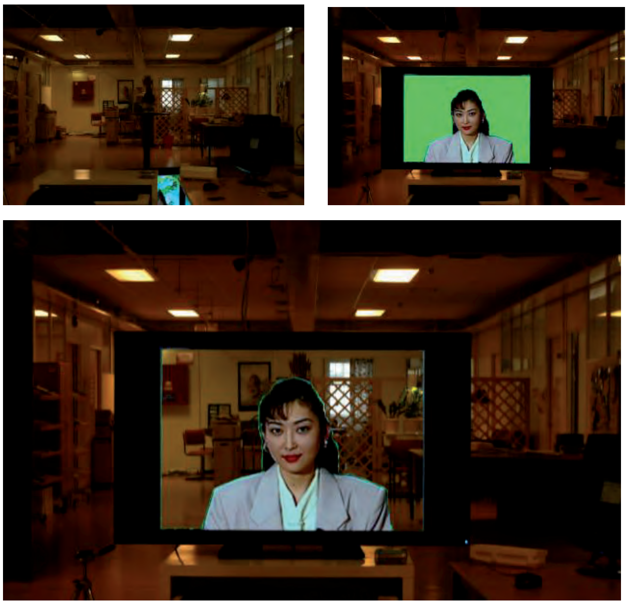


Figure 3: Top left: The room where the prototype system is installed, with the screen temporarily removed when taking the background photographs. Top right: The screen is put back with the video displayed, but the chroma keying mechanism not yet enabled. Bottom: The background photographs are keyed in to produce the illusion of a transparent screen, giving the sensation of the remote person being physically immersed in the room, with consistent depth cues.

When shooting the background photos, the telepresence system (in particular the display) is removed, to expose the background of the room. The camera is positioned at each of the eight viewpoints using a special purpose pod, with the shooting angle adjusted to point exactly toward the center of the screen. This is done manually by looking through the camera viewfinder while displaying a crosshair target centered on the screen. When the eight photographs taken this way are rendered properly on the multiview display, an illusion of the display being transparent can be achieved. In combination with the chroma keying, this can create the sensation of the remote interlocutor being physically immersed in the room. The process is illustrated in the sequence of images in Figure 3.

3.4 Mirror-reflected presentation for improved immersion and eye contact

To make a perfect illusion of the remote user being immersed in the physical room, the bezel (i.e. the rim) of the display needs to be removed somehow. This can be accomplished by placing the display horizontally on a table, and suspending a mirror at a 45 degree angle above the screen reflecting the image to the viewer, as illustrated in Figure 4. By using a semi-transparent (half silvered) mirror, this arrangement can also be used to enable eye contact between the interlocutors. As can be seen in Figure 4, the two cameras are placed directly behind the semi-transparent mirror, at the height where the eyes of the person are rendered, and this avoids the parallax angle between the user's gaze direction and the camera, which is typical in traditional videoconferencing set-ups where the camera is placed on top of the screen.

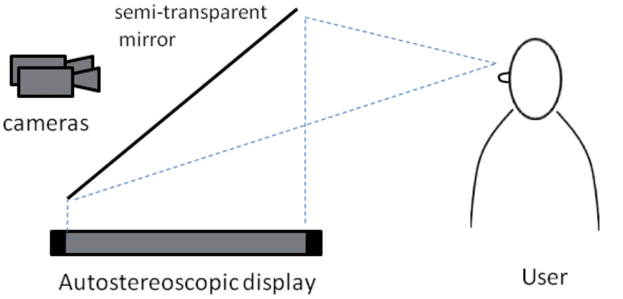


Figure 4: Semi-transparent mirror set-up supporting eye contact while also removing the bezel of the screen improving the illusion of immersion of the remote user in the physical space of the system installation.

To compensate for the mirror-reflected presentation, the video images rendered on the display must be mirrored (i.e. in software). Since this operation can easily be performed by a slight modification of the autostereoscopic rendering algorithm, it does not impose any significant additional processing requirements.

Initial experiments with eye contact in combination with autostereoscopic video rendering have been conducted, showing that the technique is feasible in practice. However, more extensive subjective user tests will be needed to assure that the improved realism and eye contact achievable can motivate the somewhat bulky and cumbersome set-ups with mirrored displays.

4 USAGE EXPERIENCES

Since the prototype system described in this paper is still under development, comprehensive subjective user tests have not yet been conducted, so the experiences from actual usage are still very limited and preliminary. However, a number of general observations regarding the subjective experience of the prototype system can nevertheless be made.

First of all, the experience of seeing stereoscopic video of a person on a display without need for glasses is still something of a novelty for most people, so the first impression of the system is often that of the user being intrigued by the technology, rather than experiencing an immediate sensation of a remote person being physically present (which the user knows *a priori* is not the case). This failure of the technology to become transparent to the user, which as previously mentioned is in fact one of the main goals, we believe to be primarily due to the prototype not being finished, in combination with the fact that a proper subjective test environment has not yet been established. However, it is reasonable to suspect that the system also when perfected will require some time to get used to.

Another general observation regarding the stereoscopic aspect of the system is that the stereopsis is mainly perceived as giving depth to the room around the user, and not so manifestly to the user's face and upper body. This is of course due to the fact that a frontal view of a person's upper body is actually rather flat, unless the person is extending an arm or something like that. Partly this is also due to the fact that six out of the eight views are algorithmically generated from the two real camera views, and the synthetic views tend to appear flatter than the real camera views. On the other hand, due to the multiview rendering, small head movements to some extent give the user the sensation of being able to look around the person on the screen, just like in real life, and this is a powerful cue greatly improving the sensation of telepresence.

One of the main general observations is that stereopsis in itself should not be overestimated as the key to achieving realism and presence in teleconferencing. Rather, it is the combination of a number of visual, aural and other cues that creates a substantially improved feeling of presence compared to traditional systems. True size of participants, high enough video quality, directional audio, lip-sync, stereoscopic rendering, immersion techniques and eye contact all contribute to the complete experience and if one of the mechanisms fail, the feeling of presence quickly diminishes.

When it comes to determining what level of quality is required for the actual video signals, experiments targeting non-interactive stereoscopic 3D video viewing show that the quality experience will be perceived as equally good using H.264/AVC or H.264/MVC with a Quantization Parameter (QP) above 38. There could also be safely done a preprocessing with spatial resolution reduction of four without loss of perceived quality, but frame rate reduction affects the quality negatively. Furthermore, temporarily switching to 2D as concealment strategy is perceptually preferable to using a traditional 2D based concealment strategy, e.g. standard H.264/AVC [12]. These types of experiments give useful information on how to optimize the video communication. There are also applicable results on the impact of transmission delay and audio/video synchronization (see e.g. [13, 14]). However, it is far from enough. To really understand the added value i.e. the degree of presence, which this type of system can offer, carefully designed subjective tests with the system in operation has to be performed.

When it comes to the mechanism devised for achieving eye contact, the technique based on semi-transparent mirrors is well established and used heavily in television studios, albeit with traditional 2D displays. Studies of human sensitivity to eye contact in 2D and 3D show that human perception of eye contact is unaffected by stereoscopic depth [15]. This strengthens the hypothesis that the proposed solution is not only technically feasible, but that it will significantly enhance the subjective experience of users of stereoscopic videoconferencing sessions. However, this claim will require more comprehensive subjective tests to be substantiated.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have presented the design of an immersive autostereoscopic telepresence system aimed at improving the sensation of virtual presence in videoconferencing sessions. The system relies on a novel combination of multiview autostereoscopic displays, chroma keying and mirror-reflected presentation to achieve depth-perception, immersion and eye contact.

Our preliminary usage experiences indicate that the stereopsis, immersion and eye contact mechanisms of the system in combination with other well-known means to enhance the teleconferencing experience can significantly improve the feeling of telepresence.

Our future work is to finish a testbed installation of the prototype system, so that more extensive subjective user tests can be performed in a controlled environment.

Acknowledgment

This work was partially funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems.

References

[1] Rheingold, H. “Virtual Reality,” Summit, New York, 1991.

[2] Bowers, J., Pycock, J., O'Brien, J., “Talk and embodiment in collaborative environments,” in Proc. of ACM CHI'96, ACM Press, 1996.

[3] Benford, S. and Fahlén, L. “A spatial model of interaction in large virtual environments,” ECSCW'93, Milan, September, 1993.

[4] Cruz-Neira, C., Sandin, D. J. and Defanti, T. A. “Surround-screen projection-based virtual reality: the design and implementation of the CAVE,” Communications of the ACM, 1993.

[5] Koleva, B. and Benford, S. “Theory and application of mixed reality boundaries,” Proceedings of UK-VRSIG, 1998.

[6] Schreer, O. et al. “3DPresence - A system concept for multi-user and multi-party immersive 3D videoconferencing,” 5th European Conference on Visual Media Production, Nov. 2008.

[7] Rhee et al. “Low-cost telepresence for collaborative virtual environments,” IEEE Transactions on Visualization and Computer Graphics, vol. 13, issue 1, pp. 156 - 166, Jan. 2007.

[8] Johanson, M. “Multimedia communication, collaboration and conferencing using Alkit Confero,” Alkit technical report, 2004.

[9] Wenger, S. et al. “RTP Payload Format for H.264 Video,” IETF RFC 3984, February 2005.

[10] Wang, K. et al. “Subjective evaluation of HDTV stereoscopic videos in IPTV scenarios using absolute category rating,” Proceedings of SPIE 7863, January 2011.

[11] van Berkel, C., Parker, D. W. and Franklin, A. R. “Multiview 3D-LCD,” Proceedings of SPIE 2653, pp. 32-39, April 1996.

[12] Wang, K., Barkowsky, M., Brunnström, K., Sjöström, M., Cousseau, R., and Le Callet, P., “Perceived 3D TV transmission quality assessment: Multi-laboratory results using Absolute Category Rating on Quality of Experience scale,” IEEE Transactions on Broadcasting (to appear), 2012.

[13] van den Brink, R.F.M. and Ahmed, K., “Test Suite for Full-Service End-to-End Analysis of Access Solutions: Test Objectives,” Multi-Service Access Everywhere (MUSE), IST-6thFP-26442, DTF4.4a, 2007.

[14] van den Brink, R. F. M. and Ahmed, K., “Test Suite for Full-Service End-to-End Analysis of Access Solutions: Test Methods,” Multi-Service Access Everywhere (MUSE), IST-6thFP-26442, DTF4.4b, 2007.

[15] van Eijk, R.L.J., Kuijsters, A., Dijkstra, K. and IJsselsteijn, W.A. “Human sensitivity to eye contact in 2D and 3D videoconferencing,” Proceedings of the 2nd International Workshop on Quality of Multimedia Experience (QoMEX), June 21-23, Trondheim, Norway, IEEE, Piscataway, pp.76 - 81, 2010.

Networked Media Experience

Session 2A

Chaired by Murat Tekalp, Koc University

Audiovisual Network Service Optimization by Quality of Experience Estimation

Mathias Johanson[1], Jonas Jalminger[1], Jukka-Pekka Laulajainen[2], Kaan Bür[3]

[1]Alkit Communications, Mölndal, Sweden; [2]VTT Technical Research Centre of Finland, Oulu, Finland; [3]Lund University, Lund, Sweden

E-mail: [1]{mathias, jonas}@alkit.se, [2]jukka-pekka.laulajainen@vtt.fi, [3]kaan.bur@eit.lth.se

Abstract: **With the growing popularity of audio and video communication services on the Internet, network operators, service providers and application developers are becoming increasingly interested in assuring that their services give the best possible experience to the users. Since real-time audio and video services are very sensitive to packet loss, latency and bandwidth variations, the performance of the network must be monitored in real time so that the service can be adapted to varying network conditions by mechanisms such as rate control, forward error correction and jitter buffer adaptation. However, in order to optimize a service in terms of the user's experience, the subjective effect that various network perturbations have on the user should be taken into consideration in the service adaptation mechanism. In this paper we present a novel approach to performance optimization based on rate adaptation driven by real-time estimation of the subjective Quality of Experience of a videoconferencing service. A proof-of-concept service optimization framework consisting of network monitoring, quality estimation, rate adaptation and service optimization mechanisms is presented and a testbed configuration based on network emulation is described and used for evaluation. Our initial experiments show that the approach is viable in practice and can substantially improve the Quality of Experience of real-time audio-visual services.**

Keywords: Quality of Experience, video communication, congestion control, service optimization.

1 INTRODUCTION

Audio and video communication services are typically very sensitive to variations in bandwidth, packet loss and latency. Consequently, in order for such services to work well in IP-based networks without a guaranteed Quality of Service (QoS), sophisticated end-to-end mechanisms are needed to monitor the network for perturbations and predict how a perturbation will affect the users of the service. Based on the monitoring of the network and the predicted subjective Quality of Experience (QoE), the service can be adapted in real time to maximize the quality experienced by the users.

Whereas adaptive multimedia communication services have been studied for a long time [1, 2, 3], the introduction of a QoE model in the adaptation process, in order to capture the effect of network perturbations on the user, is a novel concept. The rationale is that by considering not only the monitored network parameters (e.g. loss rate, latency), but also the way these parameters affect the user, the service optimization can be performed in a way that is perceptually preferable. In this paper we describe one such effort, wherein a videoconferencing service is extended with QoE prediction mechanisms for conversational audio and video, which is used to optimize the performance of the service based on the observed network conditions.

Different methods and metrics are available for QoE estimation and prediction. One approach explored in this work is called Pseudo-Subjective Quality Assessment (PSQA) [4], which is based on a neural network that has been trained through subjective QoE tests. With this approach, the application (i.e. the videoconferencing tool) measures the network parameters and calls a function that applies the parameters to the neural network, which in response returns an value between 1 and 5, representing the Mean Opinion Score (MOS) of the test panel that rated similar media streams with similar network conditions when the neural network was trained. The knowledge about how human subjects will experience media streams of different quality is thus encoded in the neural network. The advantage of this approach is that it gives a good correlation between subjective experience and network conditions and that it can be used in real time (which of course is necessary for the application considered here). The disadvantage is that it is time consuming and costly to conduct the subjective tests to train the neural network.

As the QoE of media streams is being estimated, the service optimization algorithm decides what can be done to improve the current situation, i.e. to maximize the QoE. If the currently estimated QoE is above a certain level, the proper action might be to do nothing (i.e. good enough quality). If the QoE is below some other (lower) threshold, the only reasonable thing to do might be to recommend the user to try again later (i.e. too bad quality to be even feasible). Between the extremes, the application should optimize the performance by adapting to the network conditions experienced. The most common service optimization mechanisms include codec rate adaptation (i.e. adjusting the sending rate or transcoding rate), adaptive Forward Error Correction (FEC) and stream shaping mechanisms.

Corresponding author: Mathias Johanson, Alkit Communications AB, Sallarängsbacken 2,SE-431 37 Mölndal, Sweden, +4631675543,mathias@alkit.se

To study the opportunities with QoE-driven service optimization, we have developed an experimentation and demonstration testbed, wherein Real-time Transport Protocol (RTP) audio and video streams originating from a videoconferencing system are sent through a network emulator to introduce perturbations in a controlled way. The prototype algorithms for QoE estimation and adaptation integrated in the videoconferencing tool and in the RTP reflector used for multipoint conferencing are then studied for different network conditions, and the performance of the optimization mechanisms is evaluated. Our testbed configuration also includes an external measurement system for verification of the network conditions, i.e. to make sure the network performance measured by the built-in probes, driving the QoE estimation and service optimization mechanisms, are correct and have the desired effect on the network.

Initial experiments show that the mechanisms developed have a considerable positive impact on the perceived quality of videoconferencing sessions in networks with large fluctuations in bandwidth, latency and loss rate.

2 MEASURING QUALITY OF EXPERIENCE

Recently, the term Quality of Experience (QoE) has emerged to complement the traditional concept of Quality of Service (QoS) for assessing the quality of networked services. Whereas the notion of QoS only takes technical parameters into account, such as packet loss rate or latency, the concept of QoE also includes the effect these performance metrics have on the user of the service [5, 6].

When assessing the experienced quality of a service, the available mechanisms can be broadly classified as *objective, subjective* or *hybrid* approaches. Objective methods are based on statistical or mathematical models for calculating how well a signal that is distorted (e.g. by transmission over a noisy communication channel) corresponds to the original. For instance, the Peak Signal to Noise Ratio (PSNR) is a very common objective quality metric for image and video communication services. Subjective methods, on the other hand, rely on having a panel of test subjects rate the perceived quality of media sequences in a controlled environment. Subjective quality is usually quantified by a value between one and five, representing the Mean Opinion Score (MOS) of the test panel. Hybrid methods, incorporating both objective and subjective elements, are typically based on utilizing some form of Machine Learning (ML) technique that is trained using subjective tests.

Quality assessment methods can also be classified according to what kind of *reference* is available when doing the assessment. In Full Reference (FR) models, the original, undistorted media is available for comparison with the transmitted, distorted media. This allows for a detailed, offline analysis of the objective or subjective difference between the original (reference) signal and the recreated far-end signal. In No Reference (NR) models, there is no reference signal available to compare the recreated signal to. Finally, in Reduced Reference (RR) models, partial information about the original signal is available for comparison against the recreated signal.

Although FR metrics typically give the most reliable results, they are inherently incompatible with the application of interest here, since the original media is not available. For such real-time control purposes as we are focusing on in this paper, a NR hybrid approach is the most suitable.

3 SERVICE OPTIMIZATION FRAMEWORK

As discussed above, the adaptive service optimization concept is based on continuously monitoring the network state to predict the QoE, i.e. the quality of the media experienced by the user, and then adapt the application to optimize the QoE. The framework we have developed to test the concept is based on five main components:

- A network monitoring framework, based on probes measuring parameters like throughput, loss rate, latency and jitter
- A QoE estimation framework for audio and video streams
- A service adaptation mechanism, for maximizing the perceived QoE of the application for each state of the network
- A video communication platform based on the software products Alkit Confero (videoconferencing end system) and Alkit Reflex (RTP reflector), wherein the monitoring and service optimization components are integrated
- A network emulator used to introduce network perturbations to observe the performance of the service optimization and QoE estimation mechanisms.

3.1 QoE Estimation and Prediction

For adaptation purposes, an accurate no-reference QoE model is needed in order to obtain estimations in real time. Our QoE estimation framework is based on the use of PSQA, which is a parametric methodology for estimating perceived quality. It works by mapping network- and application-layer parameters having an effect on quality to subjective scores. The parameters used as input to the estimator are:

- Codec, bitrate
- FEC
- Packet loss rate
- Temporal distribution of losses (mean loss burst size)
- One-way delay
- Packet interarrival jitter

The mapping from the parameters to the subjective scores is done by training a Random Neural Network (RNN) to learn the relationship between the input parameters and the subjective quality. A subset of input parameter combinations is carefully selected, a subjective assessment campaign conducted for each of them in an emulated network, and the MOS values recorded for each combination. Once the RNN is trained with the results from the subjective assessment, it can give good estimations not only for the parameter combinations used in the subjective test, but also for other parameter combinations within the range of parameters used in the training. The use of a trained RNN for MOS estimation is computationally trivial and gives very high correlation with subjective scores.

It should be noted that, because the quality value resulting from the RNN corresponds to the actual user experience, the quality of the original signal (i.e. codec, bitrate) has an effect on the MOS values. In practice, MOS values of 5 cannot be achieved and MOS above 4 is considered toll-quality.

3.2 Audiovisual Service Optimization

The service optimization algorithm is implemented in the videoconferencing end system. It works by continuously feeding data from the network monitoring component into the PSQA algorithm, which returns a MOS-like score of the estimated quality. Currently, this is done only for the audio streams received. The estimated audio quality is then used as a trigger for the actual optimization algorithm, which is implemented both in the sender side of the end system and in the RTP packet reflector. This is to allow the same mechanism to be employed both when a reflector is used and when there is no reflector (i.e. point-to-point or multicast sessions.) Many different adaptation events can be envisioned in response to QoE changes. The currently implemented actions are to trigger rate adaptation for the video streams received by the end system and to trigger a modality change from audio/video to audio only (and vice versa). The events are triggered when the score returned by the PSQA algorithm passes two threshold levels.

When the calculated MOS decreases below 4, the bandwidth adaptation mechanism in the reflector (or in the sender if no reflector is used) is triggered. This is done using RTP Control Protocol (RTCP) packets containing the monitored loss rate and throughput values, causing the bandwidth adaptation algorithm to reduce the rate of the video stream. The fact that the QoE estimation of the audio stream is used to trigger an adaptation event related to the video streams might seem strange at first, but the rationale for this is that in most videoconferencing situations, the cause of audio quality degradation is in fact the video consuming too much of the available bandwidth, leading to packet loss both in the audio and video streams. In other words, degradation in audio quality usually indicates degradation in video quality as well. By having the video bandwidth reduced, the audio stream quality is improved. Since the bandwidth of video streams is usually much higher than the audio bandwidth, only a slight reduction in video bandwidth can make a big difference in terms of audio quality. Moreover, video codecs typically provide greater opportunity to trade bandwidth for quality, compared to audio codecs. Video streams are hence more suited for rate adaptation.

If the available bandwidth in a videoconferencing session gets below a certain level, the only viable approach in order to be able to continue the conference at all, is to simply drop the video and continue using audio only. This quite radical measure is triggered when the MOS calculated by the PSQA algorithm goes below 2 (corresponding to more or less unusable quality). This signals to the sender of the media streams (or reflector) to stop sending video. When the estimated quality increases above 3 again, the video is re-enabled, in rate-controlled mode. When the MOS reaches 4, the rate adaptation is disabled after a configurable hold-down time.

3.3 Rate Adaptation

The rate adaptation mechanism is implemented both in the end system and in the RTP reflector. By having the adaptation mechanism in the reflector, the downstream rates from the reflector can be adapted to different bandwidth levels for heterogeneous network configurations. The upstream rate (to the reflector, or when no reflector is used) is controlled by the sender. The rate adaptation algorithm is basically the same in both cases: the frame rate of the video is adjusted to match a bandwidth limit which is determined by RTCP Receiver Reports (RR) of actual throughput, as measured by each receiver. At the reflector, this is done by intelligent frame dropping. At the sender, it is done by configuring the codec's target bitrate.

The components of the rate control mechanism is illustrated schematically in Figure 1 for a hypothetical scenario with one sender (S) and three heterogeneous receivers (R_i) interconnected by a reflector (R). (In a real scenario, all end systems would typically be both senders and receivers.) The rate control components are illustrated by circles in the figure, whereas the RTP stream monitoring components (measuring loss rate and throughput) are represented by triangles.

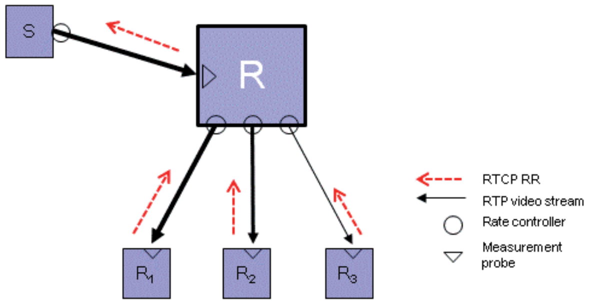


Figure 1: Schematical illustration of the rate control components implemented in the sender (S) and reflector (R). The downstream rates from the reflector are determined from the RTCP Receiver Reports from the receivers (R_i).

The advantage of adjusting the frame rate instead of some other video codec parameter, such as the quantization level which is commonly used for codec rate adaptation, is that it can be done at the reflector without the need for transcoding. For the sender side rate controller, more elaborate codec parameter adjustments could be preferable. The main disadvantage of frame rate adaptation is that the burstiness of the resultant stream increases, which can cause problems in networks with low bandwidth links and small router buffers. In this case, a stream shaping mechanism (token bucket or similar) can be applied after the rate adaptation to smooth out the frames.

The RTP streams received by the end systems are monitored to measure throughput and packet loss rate. These performance metrics are reported back to the reflector in RTCP RR packets. The reflector monitors the reports, and as soon as packet loss is detected, the bandwidth limit of the corresponding RTP stream is adjusted to the actual throughput reported in the RTCP packet. To effectuate the rate control, the reflector selectively drops packets in order to keep the downstream rate below the bandwidth limit that the receiver reported when it first experienced loss. By selectively we mean that the reflector drops full frames, dropping P-frames before I-frames, instead of dropping random packets and the current bandwidth usage is re-evaluated on I-frame boundaries.

The reflector keeps the bandwidth limit until it receives a new RTCP RR packet with updated loss rate and throughput values. If there is still loss, the bandwidth is set to what is reported by the receiver, if there is no loss the bandwidth is kept at the current rate.

When there has been a lossless period for about 10 seconds following a loss event, the bandwidth limit is increased by 5 percent to probe for available bandwidth. After yet another period with no loss, the bandwidth is increased another 5 percent. Hence, the algorithm is driven by the receiver reports and typically the algorithm adapts rapidly to worsening conditions and rather slowly to improved conditions. We have so far seen that a too optimistic approach to increase the rate usually ends up with an overestimation of the available bandwidth, resulting in congestion and a poor performance with respect to perceived user experience.

The upstream adaptation algorithm, i.e. limiting the stream from the sender to the reflector, is more or less done in the same way, but with the reflector measuring throughput and loss rate and sending RTCP RR packets to the rate controller in the sender.

Given the rate control algorithm described above, there is of course a set of parameters that are of interest to experiment with. Firstly, how often should the receiver reports ideally be sent? At least for the case when the receiver detects nonzero packet loss it should be reported as quickly as possible. However, too often would be a waste of bandwidth, without improving responsiveness. Secondly, should the reflector use the reported effective throughput as the limit for the sending rate or should it possibly use a bandwidth slightly below the reported bandwidth? Thirdly, how often should the probing for available bandwidth take place and at what rate should it be increased? We are currently investigating these issues in order to optimize the performance.

Another open research issue is how the available bandwidth should be shared between multiple streams. This question is very interesting since there are many possible ways to allocate the available bandwidth to the streams. One of the more obvious solutions would be to use a bandwidth “fairness” algorithm where each stream's allocated bandwidth is in direct proportion to the original stream's bandwidth. Another approach would be to add information about who is currently speaking and allocate more bandwidth to that stream, since it could somehow be considered more important.

4 TESTBED CONFIGURATION

A testbed for experimentation and demonstration with the QoE-driven rate adaptation has been established, as depicted in Figure 2. In this set-up, one computer, labeled *Demo System 1* in the figure, sends real-time audio and video streams through an RTP reflector to three video receivers labeled *Demo System 2*, *Demo System 3* and *Demo System 4* respectively. A network emulator is positioned between the RTP reflector and two of the video receivers to simulate different network conditions, while one of the receivers (*Demo System 2*) is unaffected by the perturbations introduced. This gives the opportunity to study how the QoE optimization can be performed in the RTP reflector independently for a set of heterogeneous receivers, enabling different quality levels for the receivers based on their downstream bandwidths.

The external monitoring system and QoE estimation visualization, shown in the top part of Figure 2, are used to verify and visualize the actual network conditions and the QoE as estimated based on the monitoring. The external monitoring systems tap into the network at any place of interest, typically before and after the network emulator, as indicated in the figure.

Service optimizations mechanisms implemented in the end systems (demo systems) and in the RTP reflector can be demonstrated by showing the performance with and without the mechanisms, for the emulated network conditions.

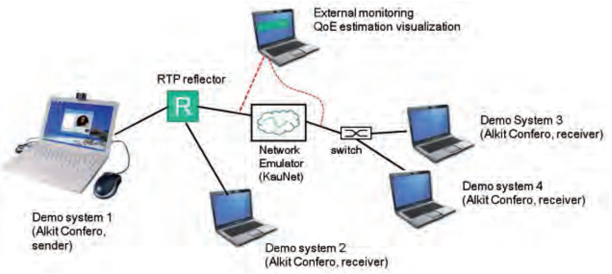


Figure 2: Testbed for QoE-driven service optimization

4.1 Network Emulation

The purpose of using a network emulator in the testbed is to create a variety of network conditions in a controlled manner. Our intention is to analyse the performance of our service optimization framework under controlled conditions. Once the framework is fully deployed, testing the system in a real network is going to be an important next step. However, the operating point of a network at any given time is a non-trivial combination of a number of different factors, making the overall behaviour non-deterministic. In order to fine-tune our video bandwidth adaptation algorithm, we need to observe its behaviour as we change essential network parameters like bandwidth, delay, and packet loss rate in a repeatable way. In other words, a network emulator provides us with the fully deterministic network we need for our tests.

In order to have repeatable network emulation in our testbed, we use KauNet 2.0 [7], a software tool developed at Karlstad University. KauNet actually extends another network emulation software, Dummynet [8], which is a standard tool on FreeBSD and Mac OS X. KauNet provides its users with many configuration possibilities, such as bandwidth, delay, packet loss, packet reordering, bit errors, or any combination of these. Pattern files are generated in advance to define the desired network behaviour on a per-packet or per-millisecond basis. KauNet processes these pattern files to emulate the network conditions. Using the pattern files, KauNet is even capable of emulating temporal changes in the network conditions, such as an increase or decrease in bandwidth or delay over time.

In our testbed, KauNet runs on Linux on a desktop PC with two network interfaces. In the network topology depicted in Figure 2, this node corresponds to a network link between the RTP reflector and the switch connected to two of the receivers in the demo system. This configuration enables KauNet to act as a transparent node between two nodes in the network and to introduce delay, loss, and change of bandwidth as defined by the user.

5 PERFORMANCE EVALUATION

The testbed described in section 4 has been used to verify how our novel QoE-driven audiovisual service optimization framework can improve the quality of video communication and conferencing sessions. For the results presented here, a configuration with a single sender and a single receiver of audio and video streams is chosen for simplicity. In the general case, multiple senders and receivers can be present.

In the performance plot shown in Figure 3, the MOS calculated by the receiving end system is plotted together with the packet loss rate for the audio stream. Figure 4 shows the bandwidth of the audio and video streams as measured by the same receiver. As can be seen in the figures, the media streams are initially received without loss, at their full transmitted rate (about 1.1 Mbps for video and 32 kbps for audio). The estimated audio quality is above 4 (good quality). After about 3 seconds into the experiment, a bandwidth limitation of 500 kbps is introduced by the network emulator. This can be seen to drastically reduce the throughput of video (as expected), while the loss rate increases to about 5%, and in response the estimated audio quality drops below 4. This triggers the video rate adaptation in the reflector, which initially reduces the video bandwidth to around 200 kbps and then stabilizes around 400 kbps, leaving enough bandwidth for the audio stream to recover.

After about 10 seconds, the network emulator is reconfigured with a bandwidth restriction of 50 kbps. This can be clearly seen to increase the packet loss rate dramatically, to over 80% loss, since the rate adaptation mechanism is unable to reduce the video bandwidth enough, resulting in an estimated audio quality measure of below 2. This triggers the modality change event, whereby the video stream is dropped altogether by the reflector, which makes the audio quality recover to the highest level attainable in practice (about 4.2), as the packet loss rate vanishes.

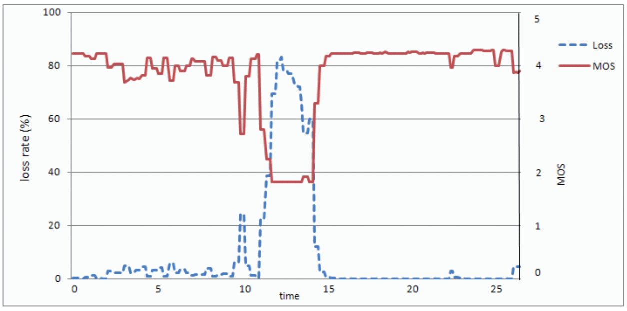


Figure 3: Packet loss rate and MOS

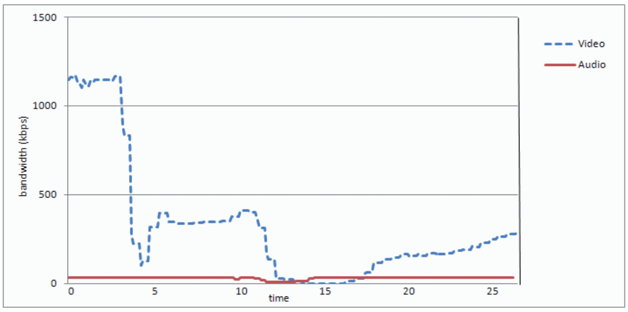


Figure 4: Audio and video bandwidth

Finally, after about 17 seconds, the bandwidth restriction is removed in the emulator, which in combination with the expiration of a hold-down timer causes the video to be re-enabled and the rate adaptation can be seen to start increasing the video bandwidth. The rate increase is done in small increments, to avoid driving the network to congestion when probing for available bandwidth. This will lead to a slow convergence to the optimal bandwidth level, when recovering from a low level (i.e. when going from bad network conditions to good). This part of the rate adaptation algorithm needs to be further developed with a more aggressive decision mechanism, although still careful enough not to congest the network immediately and not cause oscillation between on-off states for video.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a proof-of-concept implementation of QoE-driven audiovisual service optimization. The concept is based on conducting subjective tests with a video communication tool in a controlled network environment, where network perturbations are introduced and user response recorded to train a neural network. The subjective QoE can then be estimated in real time by the video communication service by feeding network monitoring data into the neural network, resulting in a MOS-like score quantifying the QoE.

The QoE estimation of our current prototype uses audio quality estimations to drive the video rate adaptation algorithm of the service and to trigger modality changes from audio/video to audio only. Our initial experiments show that the novel approach with QoE-driven real-time adaptation and quality optimization of audiovisual communication services is feasible in practice and can improve the subjective experience of future systems and services.

As future work, we are going to incorporate video quality estimations into the video rate adaptation system. We also intend to improve the adaptation algorithm according to performance evaluation results that we obtain from discrete event simulations where we can experiment further with various network scenarios complementing the testbed we designed.

Acknowledgment

This work is supported by IPNQSIS, a Celtic-Plus project. The work of the Swedish partners is partially funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems. The work of the Finnish partner is partially funded by Tekes, the Finnish Funding Agency for Technology and Innovation.

References

[1] X. Wang, H. Schulzrinne, "Comparison of adaptive Internet multimedia applications," IEICE Transactions on Communication, Special issue on distributed processing for controlling telecommunications systems, vol. E82-B, no. 6, June 1999.

[2] J. C. Bolot, T. Turletti, "A rate control mechanism for packet video in the Internet," Proceedings of IEEE INFOCOM'94, June 1994.

[3] M. Johanson, "Supporting video-mediated communication over the Internet," PhD Thesis, Chalmers University of Technology, Department of Computer Engineering, ISBN 91-7291-282-0, May 2003.

[4] M. Varela, "Pseudo-Subjective Quality Assessment of Multimedia Streams and its Applications in Control," Ph.D. Thesis, INRIA/IRISA, univ. Rennes I, Rennes, France, Nov. 2005.

[5] K. Kilkki , "Quality of Experience in Communications Ecosystem ," Journal of Universal Computer Science, vol. 14, no. 5 (2008), pp. 615-624, 2008.

[6] P. Reichl, "From Quality-of-Service and Quality-of-Design to Quality-of- Experience: A Holistic View on Future Interactive Telecommunication Services," 15th International Conference on Software, Telecommunications and Computer Networks, Sept. 2007.

[7] J.Garcia, P. Hurtig, A. Brunstrom, "KauNet: A Versatile and Flexible Emulation System," Proceedings of SNCNW 2008 – the 5th Swedish National Computer Networking Workshop, Karlskrona, Sweden, April 2008.

[8] M. Carbone, L. Rizzo, "Dummynet Revisited," SIGCOMM CCR, Vol. 40, No. 2, April 2010.

REDUCED REFERENCE 3D VIDEO QUALITY ASSESSMENT BASED ON CARTOON EFFECT

Gokce Nur[1]*, Gozde Bozdagi Akar*[2]*, and Haluk Gokmen*[3]

[1]Kirikkale University Electrical and Electronics Engineering Department, Kirikkale, Turkey

[2]METU Electrical and Electronics Engineering Department, Ankara, Turkey

[3]Arcelik A.S., Istanbul, Turkey

ABSTRACT

3-Dimensional (3D) services provided to home, offices, etc enables immersive video experience for demanding media customers. In order to provide a better service to these customers, 3D video experience monitored at the user side can be exploited as feedback information to modify the system parameters. However, due to the complex nature of 3D video (e.g., color texture, depth, etc), measuring 3D video experience of users at the receiver side is a challenge. Moreover, the use of Full Reference (FR) metrics for 3D video quality measurement enhances this challenge since both the original and compressed 3D video sequences should present at the receiver side. In this paper, we propose a Reduced Reference (RR) objective metric for the quality evaluation of color texture plus depth 3D video. Moreover, a framework is proposed to enable this metric. The RR metric is motivated using the fact that perceptually significant features of color texture sequence of 3D video can signify the main objects in the sequences. Thus, these features can be utilized as distinguishable information (i.e., side information) in the 3D video quality assessment. To determine the perceptually significant information, the color texture sequences are filtered using cartoon-like effect filter. Structural SIMilarity metric (SSIM) is used to predict the 3D video quality ensured by the degradation in the perceptually important features of the compressed color texture sequences. The performance of the proposed framework assessed using different 3D video sequences presents better results compared to its FR counterparts with reduced side information overhead.

Index Terms— 3D Video Quality, Cartoon Effect, Reduced Reference Metric, SSIM.

1. INTRODUCTION

The advancement of 3-Dimensional (3D) video technologies and future media internet will enable 3D video services to capture the consumer electronics market. In order to provide better 3D video services to demanding 3D video technology and internet users, 3D video experienced at the receiver side can be exploited as feedback information to fine tune service parameters. To be able to modify the system parameters in the best way as possible, 3D video experience should be reliably and efficiently measured.

Reliably assessing 3D video quality can now only be performed using subjective quality evaluation techniques due to the lack of reliable and efficient 3D objective techniques in literature. Nevertheless, subjective quality evaluation techniques are costly in terms of expense and time [1]. Therefore, there is a need for investigating objective quality evaluation techniques to measure 3D video quality in a quick, iterative, and reliable way.

Objective techniques to assess 3D video quality can be divided into three types in the literature as; Full Reference (FR), No Reference (NR), and Reduced Reference (RR) [2], [4] metrics.

Recent studies prove that commonly used FR metrics for measuring 2-Dimensional (2D) video quality (i.e., Peak-Signal-to-Noise-Ratio (PSNR) [5], Structural SIMilarity (SSIM) [6] and Video Quality Metric (VQM) [7]) are efficient for assessing 3D video quality [1]. However, the FR metric type causes operational difficulties for measuring 3D video quality since both original and compressed video sequences should be available at the receiver side. Thus, the NR and RR metric types should be utilized alternatively to the FR metric type for assessing 3D video quality.

No original video sequence is required at the receiver side for the NR metric type. For the RR metric type, side information extracted from the original and/or compressed sequences are exploited while assessing 3D video quality. Therefore, the RR metric type provides more accurate 3D video quality assessment than the NR metric type [2]. Considering this fact, a RR metric is proposed to measure 3D video quality in this paper. Color plus depth map 3D video representation is utilized while developing the proposed RR metric due to its advantages over the other representation types [1].

Although, a number of RR metrics are proposed for evaluating 2D video quality, there is no commonly used RR metric proposed for assessing 3D video quality.

Perceptually important features (e.g., edges, shadows, etc) of color texture sequence of 3D video can identify the main foreground and background objects in the sequence. Therefore, to measure color texture video quality of 3D video, any information degradation in the perceptually important features of a compressed color texture sequence compared to its associated original sequence can be quantified. Thus, perceptually important features in compressed and original color texture sequences are used as side-information while developing the proposed RR metric in this paper. In order to extract perceptually important features from original and compressed color texture sequences, the sequences are filtered using cartoon-like effect filter [8]. Using the cartoon-like effect, perceptually important information (e.g., edges, shadows, etc) are emphasized while simplifying visual content by abstraction. Then, SSIM is used to measure the 3D video quality of the cartoon-like compressed and original color texture sequences. The reason of using SSIM in the proposed RR metric to assess 3D video quality is that SSIM considers the structural distortion of a distorted video compared to the original one [6]. Therefore, it can efficiently measure the distortion in the perceptually important features of a compressed sequence compared to its associated one. A framework is also proposed to perform the proposed RR metric in the paper.

The rest of the paper is organized as follows. The proposed framework to perform the proposed RR metric is discussed in Section 2. In Section 3, the results and discussions are presented. Finally, Section 4 concludes the paper.

2. PROPOSED FRAMEWORK

In order to evaluate the video quality of the compressed video using a RR metric, the framework presented in Fig. 1 is proposed. As can be observed from the figure, the original and compressed color texture sequences are filtered to give cartoon-like effect to these sequences.

While providing cartoon-like effect to the video sequences, low contrast regions (low salient) are simplified while high contrast regions (high salient) are improved. In other words, to give cartoon-like effect to the video sequences, the edges and shadows are highlighted with increasing color contrast. Thus, using the cartoon-like effect filter, meaningful information for the 3D video quality measurement is emphasized and extra information that is not necessary to experience the 3D video quality is omitted.

In order to give cartoon-like effect, after the video sequences are abstracted using bilateral filter, they are stylized considering soft color quantization [8].

As seen from Fig. 1, after the original and compressed sequences are cartoon-like filtered, the degradation in the perceptually important features of these sequences are measured using SSIM. $\text{SSIM}_{\text{cartoon}}$ in Fig. 1 refers to the SSIM measurement for the side information (i.e., the perceptually important features in the cartoon-like original and compressed video sequences).

3. RESULTS AND DISCUSSION

The performance of the proposed framework is assessed in this section. Ten 3D video sequences namely; Akko, Newspapers, Advertisement, Break Dance, Farm, Ice, Eagle, Butterfly, Chess, and Football are used in the performance assessments. Four different bit rates (i.e., 512, 768, 1024, and 1536 kbps) are utilized to encode ten 3D video sequences by the Joint Scalable Video Model (JSVM) reference software version 9.13.1 [9]. 80% of the target bit rate was allocated for the video sequences of these 3D videos, and the remaining bit rate (i.e., 20%) was allocated for the depth map sequences of the 3D videos, as suggested in [1].

In order to investigate the performance of the proposed framework for assessing stereoscopic video quality, first of all, subjective experiments were conducted. Double Stimulus Impairment Scale (DSIS) method was used to for the experiments as described in the ITU-R BT-500.11 standard [10]. In this method, the video sequences are shown in pairs: the first video is the reference and the second video is the compressed one. An assessment scale ranging from 1 to 5 was used to rate the sequences during the experiments. A score of 1 represents the lowest video quality and a score of 5 represents the best video quality for the impaired video sequence compared to the original one.

A 42” Philips multi-view auto-stereoscopic display, which has a resolution of 1920 × 1080 pixels, was used to display the 3D video sequences. 18 viewers (7 females and 11 males) participated in the experiments, which is in compliance with the recommendation in [10]. They were all non-expert viewers, whose ages ranged from 19 to 37. After the experiments, the outliers are eliminated. Thus, the Mean Opinion Scores (MOSs) and confidence intervals [10] are calculated using 16 participants.

As the second step of the performance assessments of the proposed framework, the relationship between the MOS results and the objective 3D video quality measures assessed with the VQM, SSIM, PSNR, and proposed metric is approximated by the symmetrical logistic function as suggested in ITU-R BT.500-11 [10]. The symmetrical logistic function is calculated as follows;

$$s = \frac{1}{1 + e^{(D - D_M)G}} \tag{1}$$

where, s is the normalized opinion score, D is the distortion parameter, and D_M and G are constants.

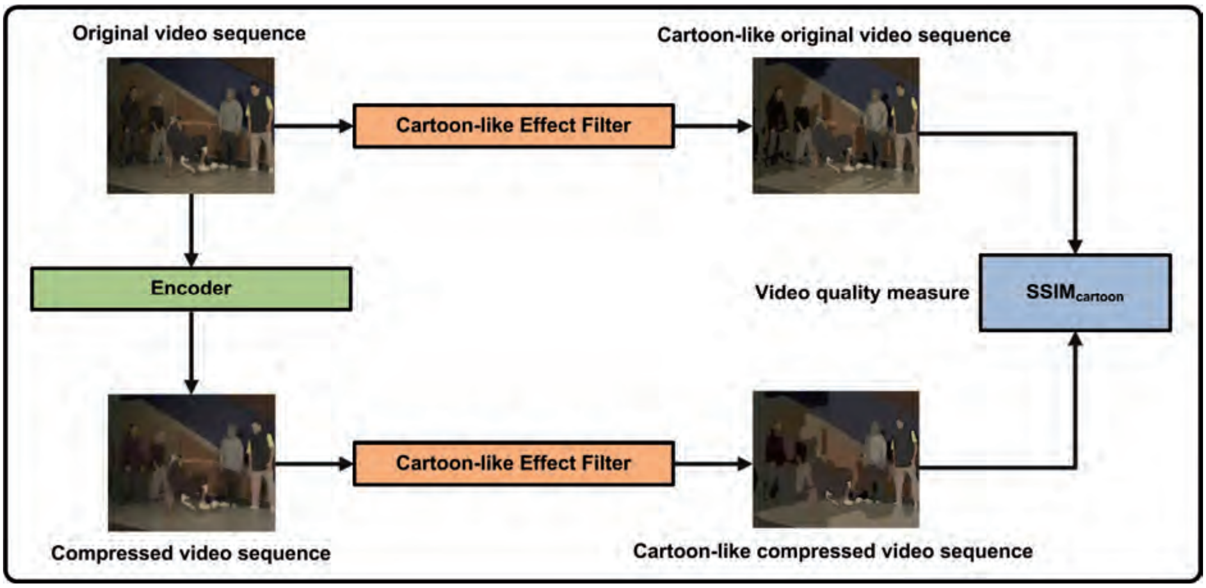
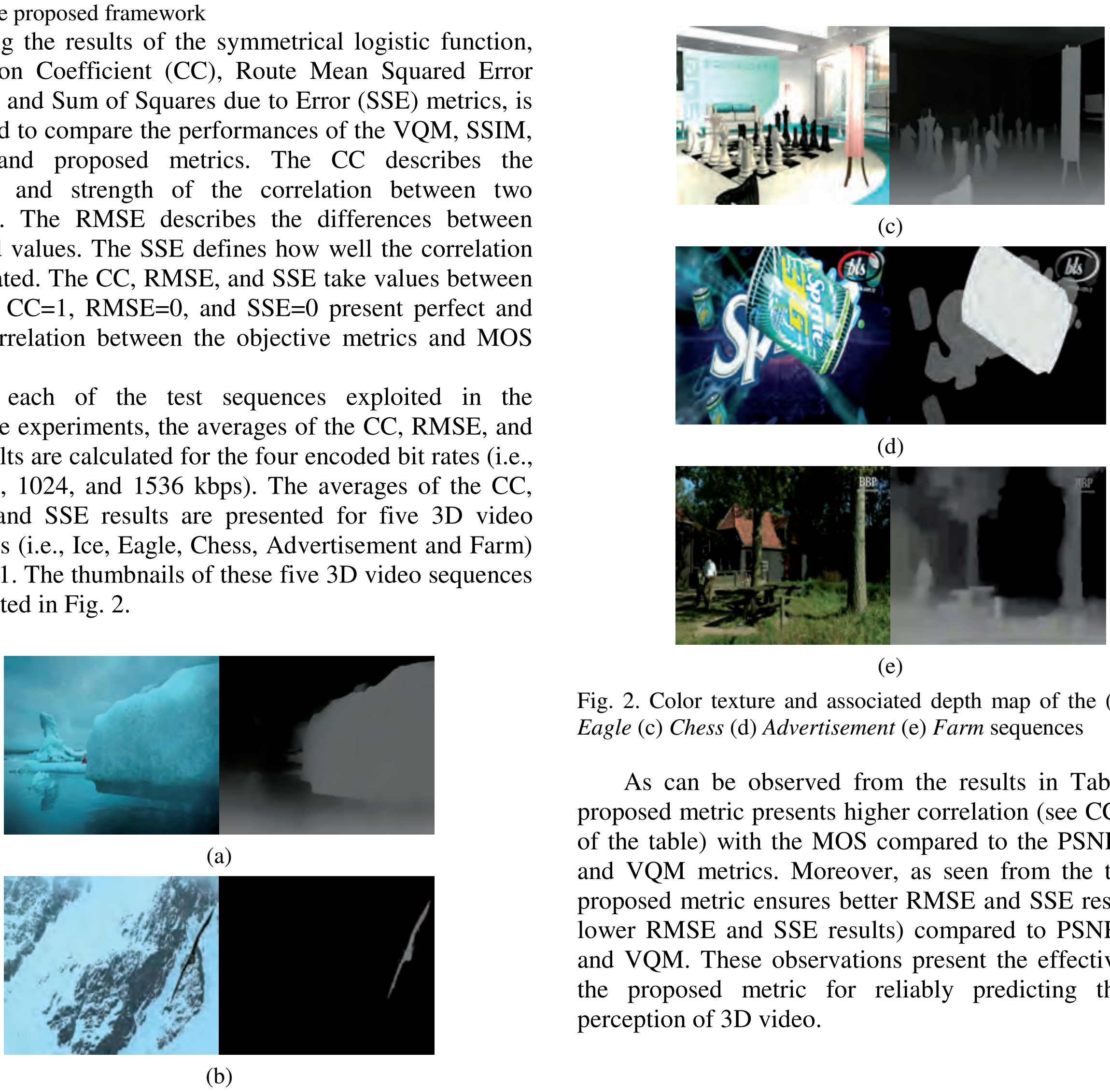

Original video sequence
Cartoon-like original video sequence
Cartoon-like Effect Filter
Encoder
Video quality measure
$SSIM_{cartoon}$
Cartoon-like Effect Filter
Compressed video sequence
Cartoon-like compressed video sequence

Fig. 1. The proposed framework

Using the results of the symmetrical logistic function, Correlation Coefficient (CC), Route Mean Squared Error (RMSE), and Sum of Squares due to Error (SSE) metrics, is calculated to compare the performances of the VQM, SSIM, PSNR, and proposed metrics. The CC describes the direction and strength of the correlation between two variables. The RMSE describes the differences between predicted values. The SSE defines how well the correlation is calculated. The CC, RMSE, and SSE take values between 0 and 1. CC=1, RMSE=0, and SSE=0 present perfect and worst correlation between the objective metrics and MOS results.

For each of the test sequences exploited in the subjective experiments, the averages of the CC, RMSE, and SSE results are calculated for the four encoded bit rates (i.e., 512, 768, 1024, and 1536 kbps). The averages of the CC, RMSE, and SSE results are presented for five 3D video sequences (i.e., Ice, Eagle, Chess, Advertisement and Farm) in Table 1. The thumbnails of these five 3D video sequences are depicted in Fig. 2.



(a)

(b)

(c)

(d)

(e)

Fig. 2. Color texture and associated depth map of the (a) *Ice* (b) *Eagle* (c) *Chess* (d) *Advertisement* (e) *Farm* sequences

As can be observed from the results in Table 1, the proposed metric presents higher correlation (see CC column of the table) with the MOS compared to the PSNR, SSIM, and VQM metrics. Moreover, as seen from the table, the proposed metric ensures better RMSE and SSE results (i.e., lower RMSE and SSE results) compared to PSNR, SSIM, and VQM. These observations present the effectiveness of the proposed metric for reliably predicting the depth perception of 3D video.

Table 1: The Performance of the Proposed Framework for Assessing 3D Video Quality

3D Video Sequence	Metric	3D Video Quality Measure Correlation with the MOS Results		
		CC	RMSE	SSE
Ice	PSNR	0.802	0.062	0.176
	VQM	0.863	0.049	0.168
	SSIM	0.771	0.069	0.187
	Proposed RR Metric	0.927	0.032	0.137
Eagle	PSNR	0.793	0.066	0.184
	VQM	0.857	0.051	0.174
	SSIM	0.764	0.074	0.196
	Proposed RR Metric	0.922	0.038	0.141
Chess	PSNR	0.784	0.069	0.188
	VQM	0.849	0.054	0.179
	SSIM	0.758	0.077	0.207
	Proposed RR Metric	0.913	0.041	0.146
Advertisement	PSNR	0.776	0.072	0.196
	VQM	0.838	0.058	0.182
	SSIM	0.747	0.081	0.213
	Proposed RR Metric	0.905	0.045	0.154
Farm	PSNR	0.769	0.077	0.201
	VQM	0.829	0.062	0.186
	SSIM	0.738	0.083	0.221
	Proposed RR Metric	0.887	0.051	0.157
Average Results of the Ten 3D Video Sequences Used in the Experiments	PSNR	0.787	0.068	0.193
	VQM	0.841	0.057	0.184
	SSIM	0.749	0.079	0.197
	Proposed Metric	0.915	0.046	0.148

4. CONCLUSION

In this paper, a RR metric has been proposed to measure 3D video quality. The color-plus-depth-map representation of the 3D video has been considered in the proposed RR metric. The perceptually important features, which have been determined using cartoon-like effect filter, have been used as side information in the proposed metric. The cartoon-like effect filter provides edge and shadow preserving smoothening to the objects of the color texture sequences. A framework has also been proposed to enable the proposed metric. SSIM has been exploited to compare the degradation in the perceptually important features of the original and compressed color texture sequences. The performance assessment results have presented that the proposed RR metric is quite efficient in terms of measuring the video quality of the color texture sequences. Using the proposed RR metric, further advancement of 3D video services can speed up into the consumer electronics market.
In our future studies, depth map of 3D video will also be incorporated with the proposed framework to accelerate this advancement.

REFERENCES

[1] C.T.E.R. Hewage, S.T. Worrall, S. Dogan, S. Villette, and A.M. Kondoz, "Quality evaluation of color plus depth map-based stereoscopic video," *IEEE J. Selected Topics in Signal Process.*, vol. 3, no. 2, pp. 304-318, Apr. 2009.

[2] P. L. Callet, C. Viard-Gaudin, S. Pechard ,and E. Caillaultn, "No reference and reduced reference video

[3] quality metrics for end to end QoS monitoring," *IEICE Trans. Commun,*, vol. E85_B, no.2 Feb. 2006.

[4] A. K. Moorthy and A. C. Bovik, "Video quality assessment algorithms: what does the future hold?," *Springer Multimedia Tools Appl.*, vol. 51, no. 2, pp. 675-696, 2011.

[5] Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *IET Electronics Letters*, vol. 44, no. 13, pp. 800–801, Jun. 2008.

[6] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Proc. of Signal Processing: Image Com.*, vol. 19, no. 2, pp. 121-132, Feb. 2004.

[7] M.H. Pinson and S .Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcasting*, vol. 50, no. 3, pp. 312-322, Sep. 2004.

[8] H. Winnemoller, S. C. Olsen, B. Gooch, "Real-Time Video Abstraction," *ACM Transactions on Graphics*, vol. 25, Issue 3, pp. 1221 – 1226, Jul. 2006.

[9] JSVM 9.13.1. CVS Server [Online]. Available Telnet: garcon.ient.rwth aachen.de:/cvs/jvt

[10] Methodology for the Subjective Assessment of the Quality of Television Pictures, ITU-R BT.500–11, 2002.

Towards Scalable And Interactive Delivery of Immersive Media

O.A. Niamut[1], J.-F. Macq[2], M.J. Prins[1], R. Van Brandenburg[1], N.Verzijp[2], P. R. Alface[2]

[1]TNO, Delft, The Netherlands; [2]Alcatel-Lucent Bell Labs, Antwerp, Belgium

E-mail: [1]{omar.niamut, martin.prins, ray.vanbrandenburg}@tno.nl, [2]{jean-francois.macq, nico.verzijp, patrice.rondao_alface}@alcatel-lucent.com

Abstract: **Within the EU FP7 project FascinatE, a capture, production and delivery system capable of allowing end-users to interactively view and navigate around an ultra-high resolution video panorama showing a live event is being developed. Within this system, network-based processing is used to repurpose the audiovisual content to suit delivery towards different device types and user selection of regions of interest. In this paper we report on the ongoing developments of the FascinatE delivery network functionality. We present the delivery network architecture and its constituent functional components. The content segmentation procedures at the ingest of audiovisual data are considered and two delivery mechanisms are discussed.**

Keywords: immersive media, high-resolution video, content aware networking, adaptive streaming

1 INTRODUCTION

New kinds of ultra-high resolution sensors and ultra large displays are generally considered to be a logical next step in providing a more immersive visual experience to end users. This notion of immersive media with ultra-high definition TV (UHDTV) and displays, highlighted by the NHK work on 8K Super Hi-Vision video [1] and the Fraunhofer HHI 6K OMNICAM system [2] seems contradictory with the explosive growth of device diversity. That is, having the content available on an increasing number of mobile devices, such as smartphones and tablets, each with its own characteristics, facilitates the user in selecting and controlling content. In contrast, UHDTV still assumes a more or less passive behaviour on the end user's side. The relevance of this contradiction for future ICT research is recognised in the NEM Strategic Research Agenda [3], as it refers to technologies for transport, coding and rendering, e.g. content-centric networks, spatial and ultra-high resolution video and video over the device continuum, as being vital to immersive media reproduction.

Within the EU FP7 project FascinatE [4] a capture, production and delivery system capable of supporting interaction, such as pan/tilt/zoom (PTZ) navigation, with immersive media is being developed by a consortium of 11 European partners from the broadcast, film, telecoms and academic sectors. The FascinatE project aims to develop a system that allows end-users to interactively view and navigate around an ultra-high resolution video panorama showing a live event, with the accompanying audio automatically changing to match the selected view. The output is adapted to the particular kind of device, ranging from a mobile handset to an immersive panoramic display. At the production side, an audio and video capture system is developed that delivers a so-called Layered Scene Representation (LSR), i.e. a multi-resolution, multi-source representation of the audiovisual environment [5]. In addition, content analysis and scripting systems are employed to control the shot framing options presented to the viewer. Intelligent networks with processing components are used to repurpose the content to suit different device types and framing selections, and user terminals supporting innovative gesture-based interaction methods allow viewers to control and display the content suited to their needs.

This paper focuses on the FascinatE delivery network. This network needs to ingest the whole set of audiovisual (A/V) data produced to support immersive and personalized applications. This typically translates into very demanding bandwidth requirements. As an example, the live delivery of the immersive A/V material in an LSR consisting of an OMNICAM and three HD image sequences would require an uncompressed data rate of more than 16 Gbps. In situations where the full LSR is to be received by an end-user terminal, say in the case of a theatre with large-scale immersive rendering conditions, the delivery requires massive end-to-end bandwidth provisioning, even when using mezzanine or broadcast video compression. But FascinatE also aims at delivering immersive video services to terminal devices with lower bandwidth access or less processing power. In particular, a high-end home set-up capable of processing the full LSR for interactive rendering, but with typical residential network access, may be unable to receive the data rate of the complete LSR. Finally in case of low-powered devices, such as mobile phones or tablets, one of the FascinatE goals is to introduce media proxies, capable of performing some or all rendering functionality on behalf of the end-client.

Corresponding author: Omar Niamut, TNO, Brassersplein 2, 2612CT Delft, +31 651916242, omar.niamut@tno.nl

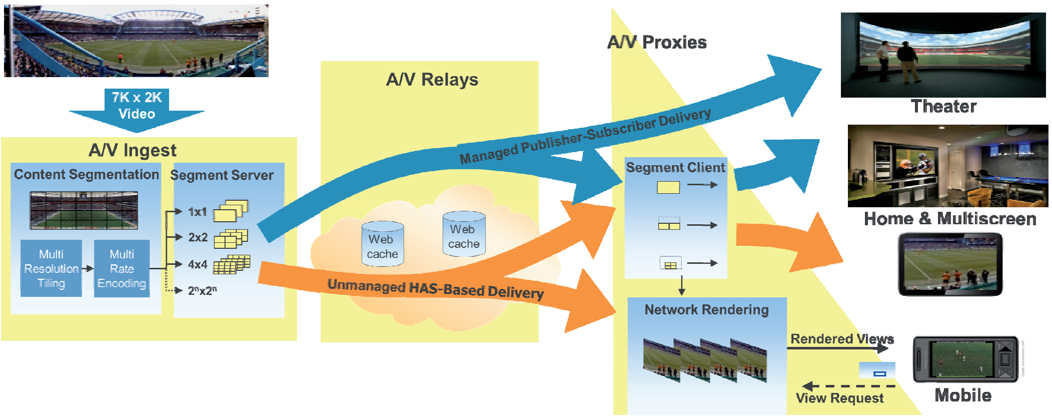


Figure 1 – FascinatE network functionality and delivery mechanisms.

The paper is organized as follows; we first describe the FascinatE network architecture and its constituent functional components in section 2. Then, in section 3, we discuss the ingest of A/V data and content segmentation. The FascinatE delivery network includes two delivery mechanisms, which are described in sections 4 and 5, respectively. Finally, in section 6, we discuss the future work planned in the project.

2 THE FASCINATE DELIVERY NETWORK

FascinatE considers three main use cases, each with its associated target end-device and screen type. First, in the theatre or public screen case, the captured content is transmitted to and displayed on a large panoramic screen, enabling multiple viewers to simultaneously see the content. In contrast, in home viewing situations a limited number of viewers consumes the content via a large TV screen and interacts using gestures, e.g. by selecting players to follow when watching a sports game and zooming in on interesting events. Lastly, for mobile usage, users can employ their individual devices, such as smartphones and tablets, to personalize their views.

In the first case, that is large displays for public viewing, dedicated optical networks such as Cinegrid [6] are already employed for uncompressed UHDTV transmissions. In contrast, for current and near-future home viewing situations based on IPTV or DOCSIS cable networks, we expect bandwidths ranging between 20-100 Mbps; not enough to transmit the full LSR, even in compressed form. Furthermore, in the case of mobile broadband networks, bandwidths of up to 20 Mbps are foreseen. Hence, the role of the FascinatE delivery network is different for each of these cases, as shown in Figure 1. In this paper, we concentrate on a delivery network architecture with functional components that facilitate the transmission of the LSR to devices for home viewing, such as Connected TVs and set-top boxes, and to mobile devices such as smart phones and tablets.

2.1 Related Work

A network-based approach for interaction with immersive media was recently demonstrated by KDDI [7]. The demonstrated prototype allows a user to zoom into a region of interest (ROI) on a mobile device. The ROI parameters are sent to a network proxy, which then crops the transmitted video to reduce the overall video bandwidth. That is, since the user is looking at a specific ROI, only that spatial part of the video can be transmitted without loss of resolution. PTZ interaction with video was previously studied by Mavlankar and Khiem. In [8] a video coding approach is described which allows for extracting ROIs directly from the coded bit stream. In [9] a tiled streaming approach is presented. Within FascinatE, the merits of these approaches are studied and incorporated into the delivery network architecture.

2.2 Delivery Network Architecture

Figure 1 also shows the three high-level active delivery components in the FascinatE delivery network. These components provide the following functionality at specific stages of the delivery, namely, ingest, storage and forwarding, and rendering.

- A/V Ingest: receives as input the full LSR and performs initial view rendering applicable for all or a large fraction of the end-users. This rendering is performed by an instance of the FascinatE Rendering Node (FRN). Furthermore, content is prepared for the actual delivery by a content segmentation operation, resulting in FascinatE media delivery units that we refer to as segments. This operation is described in section 3.
- A/V Proxy: at the other end of the network, this block is responsible for ensuring that the A/V segments required by one user or a local set of end-users are delivered and reassembled according to their interactivity requests. The proxy can also perform in-network A/V processing using an FRN instance to adapt to personalized requests and/or personalized delivery conditions, such as access bandwidth and device capabilities.

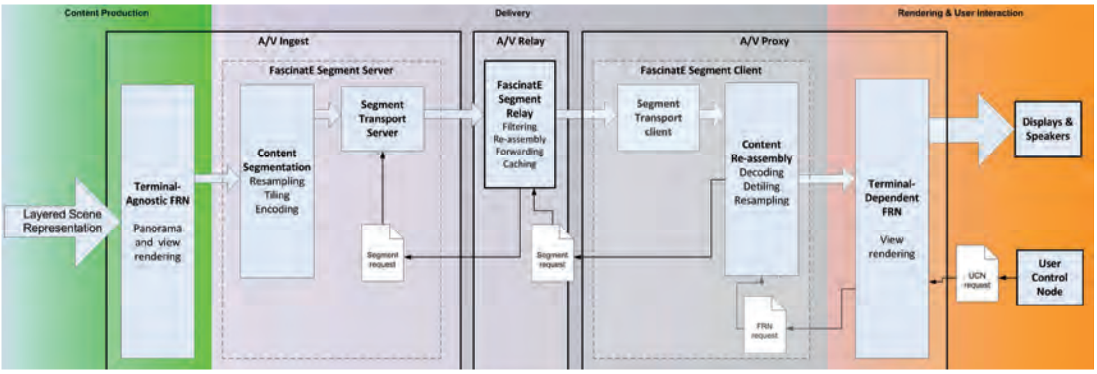


Figure 2 – FascinatE delivery network architecture.

- A/V Relay: in between these two network demarcation points, the transport of A/V segments needs to act as an end-to-end filter that accommodates the network capabilities as well as the aggregated requests of the deployed A/V proxies. This can be ensured by intermediate transport nodes, that can aggregate, cache and/or relay segment requests at the transport protocol control level, and also serve as demarcation points between delivery modes for the downstream A/V flows.

Figure 2 shows a detailed version of the FascinatE delivery network architecture, including the aforementioned functional components and the neighbouring content production and user interaction domains.

2.3 Delivery Mechanisms

The actual transport of A/V segments takes place between Segment Transport Servers and Clients. Two specific delivery mechanisms are developed between the A/V Ingest and the A/V Proxy, catering for different usage scenarios and network deployments. On the one hand, the tiled HTTP Adaptive Streaming (HAS) mechanism is suitable for web-based over-the-top delivery, in e.g. CDN or cloud video deployments. This mechanism is described in section 4. On the other hand, the PUB/SUB mechanism fits the requirements of managed delivery networks such as IPTV over xDSL and cable. This mechanism is described in section 5.

3 CONTENT SEGMENTATION

A key realisation in developing the delivery network architecture is the fact that inside the delivery network, i.e. between A/V ingest and A/V proxy, the adaptive delivery of parts of the content based on the viewing behaviour of the client (or the user) can be supported by spatially segmenting the A/V data into tiles that relate to a specific spatial region of a video frame. In most cases, tiles are grouped for a certain time period, in which case they are called segments. The particular grouping can be dependent on the transport protocol used, but globally, the FascinatE delivery network is aimed at delivery of tiled and segmented content. Regular H.264 video coding can be employed. Trade-offs in the encoding of spatially segmented content are reported in [9].

Content segmentation is required to recast the LSR content into segments that are suitable for network encapsulation and further transport functions. The general concept behind spatial segmentation is to spatially partition each video frame into rectangular pieces called tiles. All frames representing a single area of the video are taken together, encoded and stored as a new independent video stream, or spatial segment. The result is a large number of video files, each representing a specific area of the original video file. Encoding each spatial segment as an independent video stream allows an A/V Proxy to only request a subset of segments, based on the ROI selected by the user for which it performs the spatial recombination. Upon reception of the individual spatial segments, the A/V Proxy can then recombine them with a content reassembling operation and pass the result to the end-user device. In certain cases, a user may also want to see an overview of the entire video. In order to do so, the A/V Proxy would need to receive all spatial segments, resulting in enormous bandwidth requirements. Furthermore, it would need to downscale the resulting video, e.g. in order to be able to present it on a small smartphone display. To solve these inefficiencies we create multiple resolution scales. Each scale is a collection of spatial segments that together encompasses the entire video. However, each scale does so in a different resolution. For example, the top scale might consist of 144 (12x12 tiling) segments, each of resolution 640x360, together encompassing the original OMNICAM resolution of 7680x4320, while the bottom scale might only consist of 4 segments, with a combined resolution of 1280x720. It is even possible to create a scale consisting of only a single segment, with a 640x360 resolution, still showing the entire video but at a much lower quality.

The resulting spatial segmentation system provides an efficient method -in terms of required bandwidth- for receiving parts of an ultra-high resolution video. By only having to receive those areas of a video in which a user is interested, combined with support for a wide variety of display sizes and resolutions, it is possible to exploit next-generation ultra-high resolution camera systems with current generation delivery networks. Note that the combined usage of spatial segments and multiple resolution scales may lead to a significant number of video files.

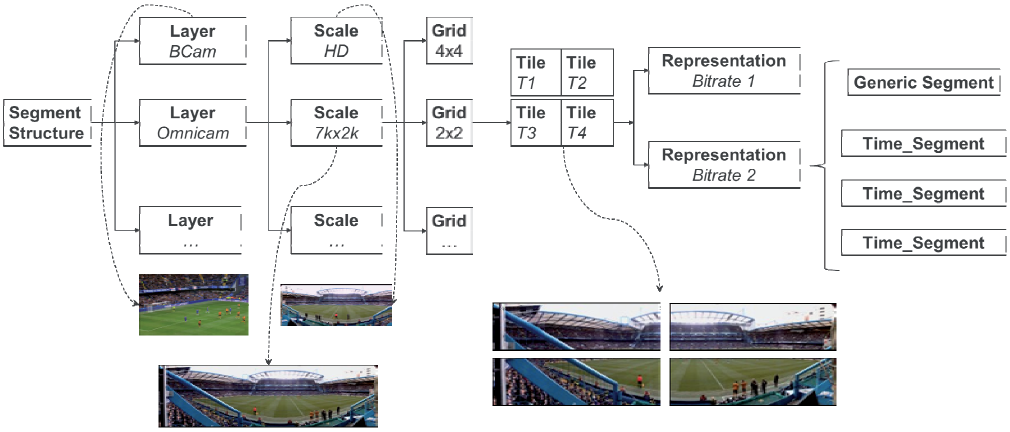


Figure 3 – Content segmentation hierarchy.

Figure 3 shows the hierarchy of segmented content and some example segments. This shows that, starting from an LSR, video content is segmented at the following levels:

- For every layer in the LSR, one or more resolution scales are created;
- For every scale, one or more MxN tiling grids are created, leading to a set of tiled video streams per scale;
- For every tile, one or more representations at different quality settings are created;
- For every representation, the associated tiled video stream can be temporally segmented.

4 WEB-BASED DELIVERY

Web-based or over-the-top delivery refers to open Internet video transport as provisioned by overlay networks such as a Content Delivery Network (CDN) or a cloud video platform. HTTP adaptive streaming (HAS) is emerging as a popular transport protocol for video streaming format. For the interactive online video services considered in FascinatE, we aim at extending HAS with spatial segmentation, so that interaction with high resolution video can be supported by a CDN provider operator, without the need for a complete overhaul of its current CDN deployment. In particular, the prototype described in section 4.3 supports a companion screen scenario, where the final rendering stage is performed on e.g. a Connected TV, whereas the interactive control is done on a thin client, e.g. a tablet device. The developed prototype allows one to freely navigate into the 7K x 2K OMNICAM video using a second screen, as shown in Figure 4.

4.1 HTTP Adaptive Streaming

HTTP adaptive streaming has recently emerged as a standard for video delivery over best-effort networks. HAS enables the delivery of (live) video by means of the HTTP protocol, by providing the video in segments that are independently requested by the client from a web server. A video is temporally split in several video segments, which in itself are standalone video files. These segment files can be delivered separately. When recombined they provide a seamless video stream. A video can be provided in several representations: alternative versions of the same content that differ in resolution, the number of audio channels and/or different bitrate. All representations are temporally aligned such that segments of different representations can be interchanged. An ISO/IEC HAS standard has recently been created by MPEG and is referred to as DASH (Dynamic Adaptive Streaming over HTTP [10]).

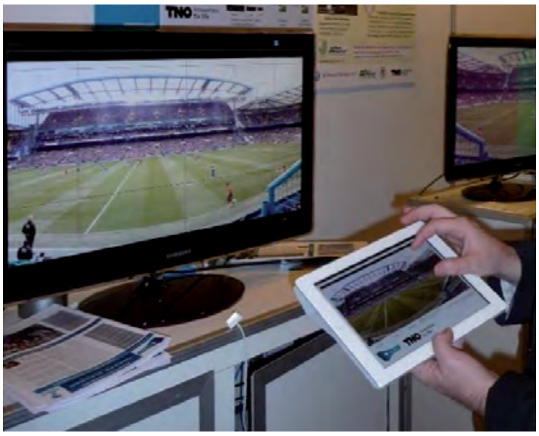


Figure 4 - Prototype allowing for second-screen pan/tilt/zoom interaction with the panoramic video.

4.2 HAS and Tiled Streaming

The aforementioned HAS solutions focus on temporal-segmentation. HTTP adaptive streaming can however also be combined with the spatial content segmentation procedure described in section 3. Each video tile is individually encoded and temporally segmented according to common HAS solutions. An advantage of using HAS for the delivery of spatial tiles is that the inherent time-segmentation makes it relatively easy to resynchronize different spatial tiles. That is, all HAS tiles are temporally aligned such that segments from different tiles can be easily recombined to create the reassembled picture. As long as the time segmentation process makes sure that time-segments between different spatial tiles have exactly the same length, the relative position of a frame within a time segment can be used as a measure for the position of that frame within the overall timeline.

In HAS solutions such as MPEG-DASH, a *manifest file* is used to describe the structure of the segmented content. This manifest is referred to as a Media Presentation Description (MPD). The MPD includes all information that a HTTP client needs to retrieve the media segments corresponding to a media session, such as the Media Presentation, alternative representations of the media, specific groupings of media and segment and media information, e.g. segment length, resolution, audio and video codecs and the container format. The MPD as defined in MPEG DASH can be readily extended with resolution scale and spatial tiling information.

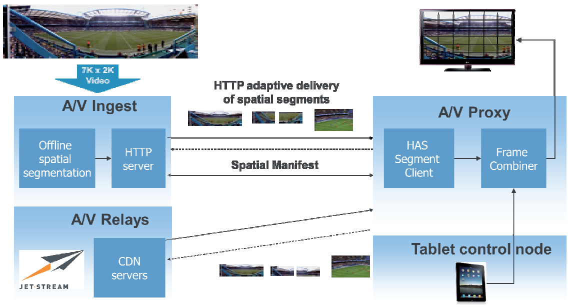


Figure 5 – Tiled HAS proof of concept.

4.3 Proof of Concept

A proof of concept prototype of the tiled HAS delivery mechanism has been developed as part of the overall FascinatE delivery network testbed. This proof of concept is illustrated in Figure 5. It supports HAS-based delivery of H.264 encoded content segments that allows the system to be deployed on current CDNs and cloud infrastructures. This was achieved by developing an integrated set functions for the A/V Ingest and A/V Proxy components based on the tiled HAS mechanism, and by using an actual CDN as the A/V Relay. The main functions at the A/V Ingest, Relay and Proxy are the following:

- At the A/V ingest: the main component is the tiled HAS server that hosts the segmented LSR content.
- At the A/V relay: the main component is a live CDN delivery server, for scalable and distributed delivery of segments.
- At the A/V proxy: the main components are the tiled HAS segment client which requests the segments, and the frame combiner which performs the content reassembly function and adapts the reassembled view to the target device.

Further functionalities incorporated in the prototype are trick play, picture-in-picture through multi-ROI rendering and predetermined ROI selection. Also, additional layers from the LSR, e.g. from broadcast cameras can be made available on the companion screen.

5 MANAGED DELIVERY

Managed delivery refers to a video transport platform that is fully under control of a service provider. This includes controlling how content is represented and transported over the complete end-to-end delivery path, from ingest till client device, as well as some policies regarding the management of network resources and performances. A typical example in managed IPTV services is the allocation of linear TV channels to provisioned multicast trees. For the interactive TV services tackled in FascinatE, we aim at extending the optimized transport approach of managed delivery, so that low-delay interaction and high-quality video can be supported by a network operator at a reasonable cost in terms of resources consumed.

5.1 Publisher/Subscriber mechanism

One of the main challenges for the FascinatE delivery network is that the requirements on the type of transport technology seem contradictory depending on which end of the network one looks at:

- At the delivery network ingress where the whole LSR is ingested, the network elements are responsible for pushing the content through the network, agnostic to the actual user requests.
- At the terminal side, user-specific portions of the layered scene may be requested. Therefore, if the capabilities of the end-to-end network and of the terminal cannot support a plain transmission of the full LSR, the terminal has to send some requests upstream to pull those parts of the LSR which are required for rendering.

To cover these two requirements, we propose to use a message-queue mechanism, which specifies “publisher” and “subscriber” functions that can work asynchronously at each end of the network. This approach fits well for a deployment in a managed network, such as next generation IPTV systems, that would be required to support a large number of end-devices with various bandwidth and processing capabilities. In this Publisher/Subscriber (PUB/SUB) mechanism, the tiled-based representation naturally leads to assigning each publisher to a given spatial tile. The published data is transported over a combination of unicast and multicast channels, organized according to the multi-resolution hierarchy of spatial segments described in section 3.

In addition to a better control of the transport channels, a managed network context also opens the possibility to put more processing functions into the network. In particular, our managed solution supports an end-to-end scenario, where the final rendering stage is also performed in the network, so as to support thin clients. In this case, the thin client does not need to directly subscribe to the segmented data, but directly receives a pre-rendered video stream. This requires the network to include in the A/V Proxy rendering functions are responsible for making the received segments ready for delivery to the end-device. Such a Video Proxy prototype (shown in Figure 6) has been developed so as to allow any thin client device to freely navigate into the 7K x 2K OMNICAM videos.

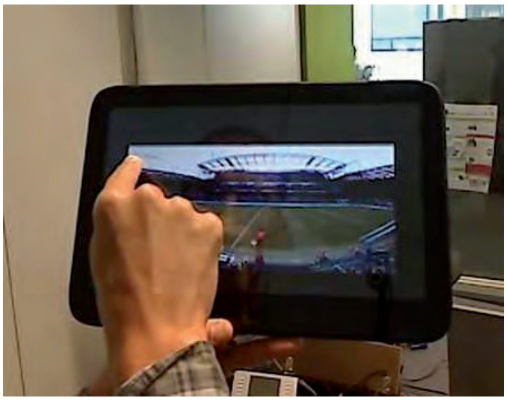


Figure 6 - Continuous interactive video rendering on a Thin Tablet Client.

The end-device only has to send its pan-tilt-zoom navigation commands (e.g. from a touch-based user interface) to the proxy and receives back the requested sequence of views, fully pre-rendered by the network and delivered at a resolution and bandwidth that match the device capabilities. With this approach, high-resolution video content can be watched interactively in a natural manner, even on a low-power and small-display device.

5.2 Proof of Concept

A proof of concept prototype of the PUB/SUB delivery mechanism has been developed as part of the overall FascinatE delivery network testbed. The current set-up is illustrated in Figure 7. It supports live in-network rendering (A/V Proxy) that can serve multiple Video Thin Clients. Two rendering mode are supported. A 2D mode consists in continuously reframing and rescaling the panorama according to the stream of user navigation commands. A 3D mode, relying on GPU acceleration, compensates in addition the geometrical distortion of the cylindrical representation of the OMNICAM content.

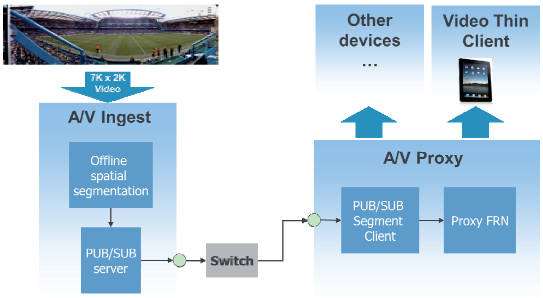


Figure 7 – PUB/Sub proof of concept.

The role of the PUB/SUB mechanism is to connect the full hierarchy of tiled content at the A/V Ingest till the A/V Proxy where only a subset of this content is required on behalf of end clients. The delivery of video tiles is optimized so as to guarantee that each A/V Proxy receives the required subset of LSR data and minimize the bandwidth usage between ingest and proxy. This work is based on our previous work [11] where this joint optimization of video coding and tile selection was studied.

6 FUTURE WORK

In the remainder of the FascinatE project, the delivery network architecture and proof of concept implementations will evolve so as to support live delivery of ultra-high and interactive video services, in manner that can scale to many client with heterogeneous access bandwidth and end-device processing power. Further studies will evaluate the scalability performance of the proposed approaches. Additionally, other FascinatE components will be supported and incorporated, such as scripted view rendering, interactive audio rendering and gesture-based interaction.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 248138. Thanks to SIS Live and the Premier League for their assistance and permission to use the football images. The authors would like to thank the project partners for the insightful discussions and collaboration. Also thanks to CDN provider Jet-Stream for making available their cutting-edge CDN testing environment.

References

[1] M. Maeda, Y. Shishikui, F. Suginoshita, Y. Takiguchi, T. Nakatogawa, M. Kanazawa, K. Mitani, K. Hamasaki, M. Iwaki and Y. Nojiri. "Steps Toward the Practical Use of Super Hi-Vision". NAB2006 Proceedings, Las Vegas, USA, April 2006.

[2] R. Schäfer, P. Kauff, and C. Weissig. "Ultra-high resolution video production and display as basis of a format agnostic production system", IBC2010 Proceedings, Amsterdam, Netherlands, September 2010.

[3] NEM Strategic Research Agenda - Position Paper on Future Research Directions, 2nd edition, September 2011.

[4] O. Schreer, G. Thomas, O.A. Niamut, J-F. Macq, A. Kochale, J-M. Batke, J. Ruiz Hidalgo, R. Oldfield, B. Shirley, G. Thallinger. "Format-agnostic Approach for Production, Delivery and Rendering of Immersive Media", NEM Summit 2011, Torino, Italy, 27th September, 2011.

[5] G.A. Thomas, O. Schreer, B. Shirley, J. Spille. "Combining panoramic image and 3D audio capture with conventional coverage for immersive and interactive content production", IBC 2011, Amsterdam, The Netherlands, 11th September, 2011.

[6] P. Grosso, L. Herr, N. Ohta, P. Hearty and C. de Laat. "Super high definition media over optical networks", Future Generation Computer Systems, Volume 27, Issue 7, Pages 881-990, July 2011.

[7] KDDI R&D Labs, Three Screen Service Platform. http://www.youtube.com/watch?v=urjQjR5VK_Q. Visited: May 10th, 2012.

[8] Mavlankar, A., "Peer-to-Peer Video Streaming with Interactive Regionof- Interest", Ph.D. Dissertation, Department of Electrical Engineering Stanford University, April 2010

[9] Khiem, N., Ravindra, G., Carlier, A., and Ooi., W. 2010. Supporting zoomable video streams with dynamic region-of-interest cropping. In Proceedings of the first annual ACM SIGMM conference on Multimedia systems (MMSys '10). ACM, New York, NY, USA, 259-270.

[10] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP - Standards and Design Principles", MMSys'11, February 23–25, 2011, San Jose, California, USA.

[11] P. R. Alface, J.-F. Macq, and N. Verzijp, "Interactive Omnidirectional Video Delivery: A Bandwidth-Effective Approach", Bell Labs Technical Journal 16(4): 135-147, 2012.

Using the MPEG-7 Audio-Visual Description Profile for 3D Video Content Description

Nicholas Vretos, Nikos Nikolaidis, Ioannis Pitas

Computer Science Department, Aristotle University of Thessaloniki, Thessaloniki, Greece

E-mail: {vretos, nikolaid, pitas}@aiia.csd.auth.gr

Abstract: **In this paper we propose a way of using the Audio-Visual Description Profile (AVDP) of the MPEG-7 standard for stereo video content. Our aim is to provide means of using AVDP in such a way that 3D video content can be correctly and consistently described. Since, AVDP semantics do not include ways for dealing with 3D video content, and thus, a new semantic framework within AVDP is proposed. Finally, we show some examples of xml files that describe stereo video content.**

Keywords: Stereo video, MPEG-7, AVDP.

1 INTRODUCTION

Automatic analysis of videos consists of algorithms for shot boundaries detection, face detection/tracking/recognition, facial expression recognition and others. These algorithms are used in applications such as fast indexing in databases, implementation of better editing tools, as well as better program schedulers in real-time applications such as the TV context. It is very easy to conclude that video data increase exponentially with time and researchers are focusing on finding better ways in classification and retrieval applications for video databases. Moreover, the way of constructing videos has also changed in the last years. The potential of digital videos gives producers better editing tools for a film production. Finally, automatic schedulers for TV programming are essential in the broadcasting business.

The new trend in multimedia is the use of 3D representation. Most of the recent film productions have their 3D versions. Stereo video is an approach of 3D video film making, based on the human eye system [1]. It consists of two different cameras put together side-by-side (most of the time), and therefore, by means of special glasses and viewing software, the viewer is able to perceive a 3D motion picture in front her/his eyes. Analysis of stereoscopic video have the advantage of additional information to improve results of the before mentioned algorithms, and also, derive annotation for 3D specific content such as 3D position of foreground objects, viewing quality of 3D content and others.

For a better manipulation of all the above, MPEG-7 standardizes a set of Descriptors (Ds), Description Schemes (DSs), a description definition language (DDL) and a description encoding [2]-[6]. A considerable amount of research effort, have been invested over the last years to improve MPEG-7 performance in terms of semantic content description [7]-[10]. Nevertheless, 3D content description has not yet been investigated in the MPEG-7 context. Although some description and description schemes have been proposed to model 3D information, they are only explicit descriptors for geometrical information and not for 3D video content.

The AudioVisual Description Profile (AVDP) is very recently adopted as a new profile of the MPEG-7 standard. This profile consists of a subset of the original MPEG-7 standard, and aims in describing the results of most of the known media analysis tasks (e.g. shot detection, face detection/tracking, and others), in a normative manner.

Our aim is to show that AVDP can be used, with new semantics, to describe also 3D media analysis tasks results. An approach of using the descriptors and description schemes of the AVDP is proposed, for most of the known media analysis tasks extended to the 3D context.

The paper is organized as follows: Section 2, an overview of the AVDP will be presented and the way 3D information can be incorporated. In Section 3, we show the details of the 3D AVDP semantics for several known media analysis algorithms. In Section 4, specific 3D issues that need to be integrated in the AVDP are treated. Finally, in Section 5, conclusions and future work is discussed.

2 THE AUDIOVISUAL DESCRIPTION PROFILE (AVDP)

The Audio Visual Description Profile (AVDP) of MPEG-7 provides a standard way to store high and low level information, which is extracted from the content analysis of video. AVDP was designed to benefit both broadcasters and industry in order to create a normative layer between research and end users (i.e. TV broadcasters and media analysis industry). In this paper, we propose to store semantic analysis results to an XML file. This XML file must be compatible to the specifications described in the XML Description Schema (XSD) of the AVDP. We have selected a subset of the description tools available in the AVDP, which cater to our needs, and we have defined a description procedure, using these description tools, in order to store information describing 3D content.

Corresponding author: Nicholas Vretos, Computer Science Department, Aristotle University of Thessaloniki, vretos@aiia.csd.auth.gr

In the following a 3D video segment can mean one of the following:

- a stereoscopic video consisting of two channels (left and right)
- a stereoscopic video consisting of two channels (left and right) and two or four extra channels containing corresponding disparity information (horizontal and vertical)
- a video consisting of a color channel and a depth information channel.

The following list contains most of the tools and the description schemes (DSs) that were used to our end.

- TemporalDecomposition (TD). This description scheme (DS) acts as a tool that performs temporal decomposition of the video in multiple temporal segments, such as Video Segments (VS) or Audio-Visual Segments (AVS).
- MediaSourceDecomposition (MSD): used in the AVDP context to decompose an audiovisual segment (or an entire audiovisual content) into the audio and video channels that it contains.
- VideoSegment (VS): provides a way to describe a video segment of the visual content. The starting time point of the VS and its time span defines each segment.
- SpatioTemporalDecomposition (STD): enables a video segment to be decomposed into parts defined spatiotemporally namely MovingRegions (e.g., in order to store information related to moving objects).
- MovingRegion (MR): used to describe, for example, a moving object by storing the spatiotemporal behavior of the object (spatial coordinates of the bounding box of an object and their change with respect to time).
- SpatialDecomposition (SD): used for the spatial decomposition of a frame. The result is one or more regions of interest (StillRegion) within a frame that may depict an object, face etc.
- StillRegion (SR): used to describe a still object, by defining its spatial span within a frame. A StillRegion can also denote an entire frame.
- StructuredAnnotation: used to annotate concepts, events, human actions etc.
- FreeTextAnnotation. Annotate a video segment with free text.

In order to stay consistent with the AVDP profile, we use different content entities for each video and depth channel. A content entity, as its name indicate, is a container type able to store all lower level descriptors of the AVDP. It is used a the root of the description for a specifc channel. By these means, we treat each channel as a different video. Figure 1 below illustrates this fact.

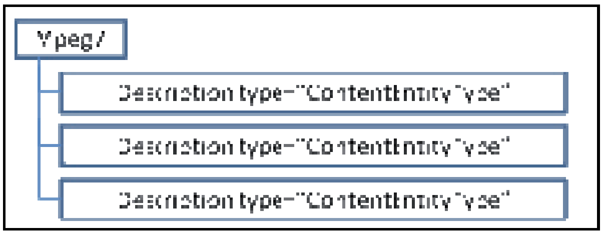
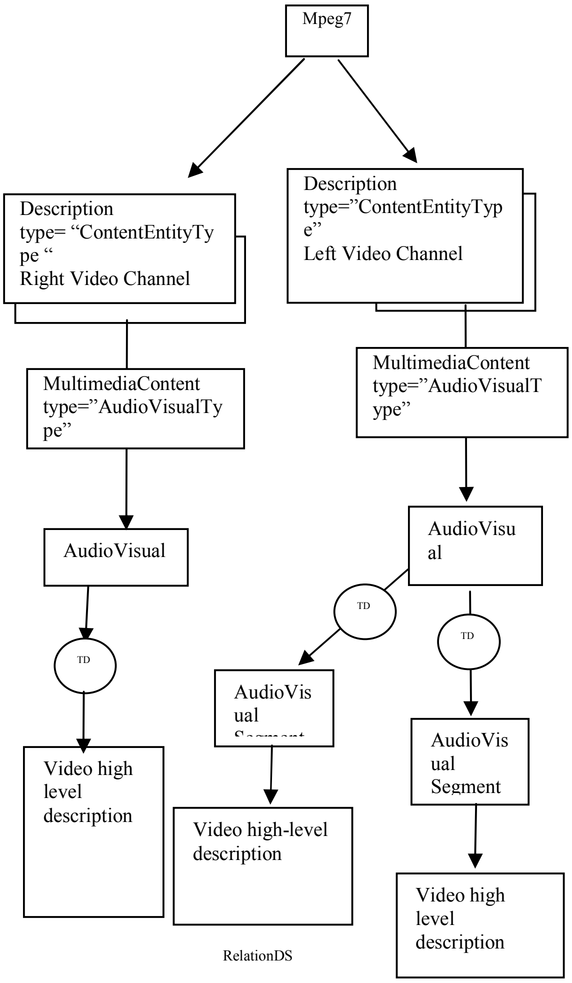

Mpeg7
Description type="ContentEntityType"
Description type="ContentEntityType"
Description type="ContentEntityType"

Figure 1: Each contentEntityType is a different video channel or depth channel.

Therefore, we create for each ContentEntityType, the analysis tree based on the AVDP as show in Figure 2.


Mpeg7
Description type= "ContentEntityType " Right Video Channel
Description type="ContentEntityType" Left Video Channel
MultimediaContent type="AudioVisualType"
MultimediaContent type="AudioVisualType"
AudioVisual
AudioVisual
TD
TD
TD
AudioVisual Segment
AudioVisual Segment
Video high level description
Video high-level description
Video high level description
RelationDS

Relation between object appearing in left and right channels.

Figure 2: AVDP tree for 3D Video description

3 AVDP FOR 3D VIDEO ANALYSIS TASKS

In this section we shall describe 10 different 3D content analysis algorithms and the way their results can be described using the AVDP. The analysis tasks that we deal with, are major research areas in the video and 3D video processing area. The algorithms are:

- Scene boundaries detection
- Shot boundaries detection
- Key Frames and Key Video Segment Extraction
- Object Detection
- Object Tracking
- Human Activity Recognition
- Face Clustering
- Object Clustering
- Facial Expression Recognition

3.1 Scene/Shot Boundaries Detection

A scene/shot boundaries detection algorithm is able to detect boundaries of scenes/shots in multimedia content. The aim is a temporal decomposition of a multimedia content into different scenes/shots. For the scene/shot boundaries detection algorithm a temporal decomposition of an AudioVisualSegment is generated. The description of such an output is simple in terms of AVDP.

Instead, a scene or shot boundaries detection algorithm, may store its results in an AudioVisual Temporal Decomposition (TD). Each resulting AVS will include some temporal information and nothing else. The schematic representation of such a descriptor is shown in Figure 3.

Moreover, with exactly the same approach we can handle the cases of shot transitions, which exceed the simple cut case. Such transitions are the fade-in, fade-out, dissolve and others, which contains more than one frame. In these cases we use the same description scheme and code the transition as a shot (i.e. a Video Segment).

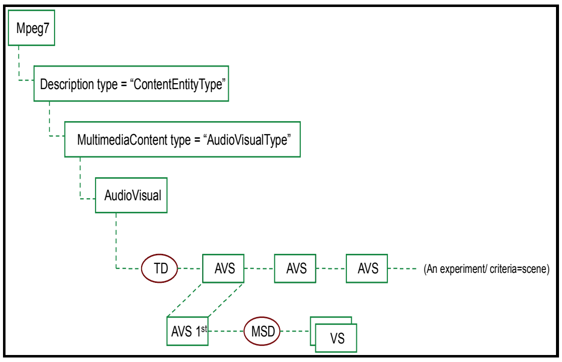

Mpeg7
Description type = "ContentEntityType"
MultimediaContent type = "AudioVisualType"
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=scene)
AVS 1st
MSD
VS

Figure 3: Scene/shot schematic representation

3.2 Key Frames and Key Video Segment Extraction

Key frames extraction refers to the multimedia analysis task, where some characteristic frames are extracted from a video segment (in most cases from a shot). Key video segments extraction is the analysis task where characteristic video segments, namely visual summaries, are extracted from a shot. Key frames and segments can be used for fast browsing and condensed representation of query results in a 3D video asset management environment. The key frames and key video segments extraction algorithm generates a list of video segments of duration of one frame or more frames, respectively. The description of such an output is simple in terms of AVDP. The schematic representation of such a descriptor is shown in Figure 4.

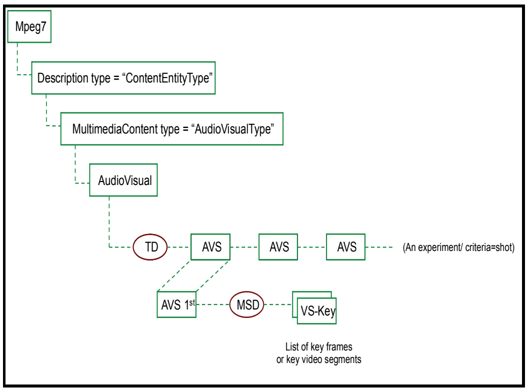

Mpeg7
Description type = "ContentEntityType"
MultimediaContent type = "AudioVisualType"
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=shot)
AVS 1st
MSD
VS-Key
List of key frames
or key video segments

Figure 4: Key Frame and Key Video Segment schematic representation

3.3 Object Detection/Tracking

Object detection is the process of finding a predefined object (e.g. a face, a car, a ball etc) in a 3D video. Usually, object detection is performed in a per-frame basis, although an extension for object detection on a video segment using a frame-by-frame approach is straightforward. Since the object detector usually detects a specific object or a specific category of objects, this information can be used to semantically annotate the detected object with its type. For instance, a face detector detects faces and thus we know that all objects detected by such a detector are faces and we can store this information within the object. On the other hand, object tracking is the process of finding the trajectory of a predefined object (e.g. a face) in a sequence of frames. Usually, object tracking is performed in a video segment in a frame-by-frame basis where the spatial position of the tracked object (usually in the form of a bounding box) is calculated for each frame. The results of such process are the object trajectory.

In the case of an object detection module the AVDP profile provides us with a StillRegionDS which can be used to store the location of the detected object(s) (usually in terms of a bounding box) as well as other relevant information for this region. This is presented in Figure 5.

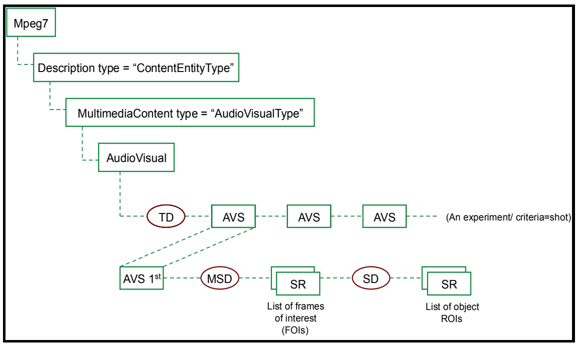

Mpeg7
Description type = "ContentEntityType"
MultimediaContent type = "AudioVisualType"
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=shot)
AVS 1st
MSD
SR
SD
SR
List of frames of interest (FOIs)
List of object ROIs

Figure 5: Object Detection schematic representation

In the AVS-1st level of the above description, we use a MediaSourceDecompositionDS (MSD) to decompose the VS into a list of StillRegionType elements where each of them represents an entire frame of the video segment. This list can be considered as the frames of Interest (FOIs). Subsequently, we decompose each frame into further StillRegionsDS through a Spatial Decomposition DS, each StillRegion representing a detected object.

To store the results of object tracking algorithms, the AVDP profile provides us with a MovingRegionDS (MR), which can be used to store information regarding a spatial region (e.g. a bounding box) that moves over time.

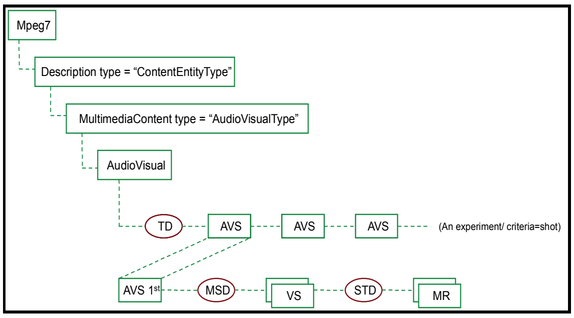

Mpeg7
Description type = "ContentEntityType"
MultimediaContent type = "AudioVisualType"
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=shot)
AVS 1st
MSD
VS
STD
MR

Figure 6: Object tracking schematic representation

It has to be noted that, as in the case of object detection, the tracker may give a first identification of the tracked object through the criteria of the spatiotemporal decomposition description scheme as well as its structural units. This is possible only if an object detection algorithm is used to initialize the tracking algorithm.

3.4 Human Activity Recognition

Human activity recognition aims at recognizing specific, predefined human activities in a video segment (i.e. running, walking, sit down etc). The video segment may be an entire shot or a video frame within a shot.

In the case of human activity recognition an annotation of a StillRegionDS or a MovingRegionDS is generated. Since both types derive from the SegmentDS type and since the AVDP profile includes the StructuredAnnotation of the SegmentDS, we can use it to provide human activity recognition description. Thus in both cases we use the WhatAction tag of the StructuredAnnotationDS to tag the recognized activity. Since we might have more than one actions taking place in a MovingRegionDS, we have to provide a way to be able to describe such different actions within the same MovingRegionDS. To do so, we can further decompose the initial MovingRegionDS into its semantically meaningful entities, i.e. moving regions, each representing a single activity. In more detail, we use the MovingRegionTemporalDecompositionType and thus create MovingRegionDSs (without the intialRegion tag) in order to characterize the activities occurring in different segments of the initial MovingRegionDS. It has to be noted, that we can allow overlapping MovingRegionDSs within the MovingRegionTemporalDecompositionType for different activities that can take place in the same time. This is useful to describe both the activity and the facial expression of a person, since (see Section 3.6) essentially we consider affect display as an activity. For instance, if we are able to characterize frames 1 to 10 of a video segment with action "Walk", while in the same time we can characterize frames 5 to 10 with an affect action of "Happiness" then we create two different MovingRegionDSs within the MovingRegionTemporalDecompositionType of the initial MovingRegionDS (containing the tracking information). These two MovingRegionDSs will overlap for frames 5 to 10. Moreover, if we have only one characterization for the entire initial MovingRegionDS (i.e. the same action is performed in the entire moving region), such decomposition can be discarded and we can directly characterize the initial MovingRegionDS. Both descriptions are valid and consistent with the AVDP profile.

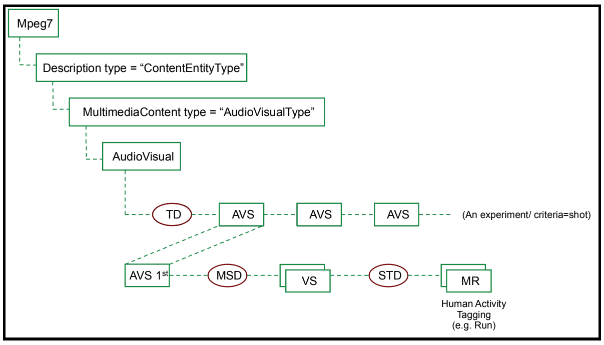

Mpeg7
Description type = "ContentEntityType"
MultimediaContent type = "AudioVisualType"
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=shot)
AVS 1st
MSD
VS
STD
MR
Human Activity Tagging (e.g. Run)

Figure 7: Human Activity Recognition schematization

Moreover, in cases where extra information about the human activity may be extracted (e.g. by geometric reasoning over the trajectory of the person), we can use the How tag of the structured annotation to describe this information as well. Such information may be for instance the direction of the activity as "Left" or "Right" (e.g. for walking).

3.5 Face/Object Clustering

Face/Object Clustering is defined as the analysis task, which partitions different facial or object images into clusters based on the actor or object they represent. In the case of 3D video, video frame regions representing faces or objects can be clustered into clusters of actors/objects. For storing the results of face/object clustering, we simply update appropriately the Who/WhatObject tag respectively in the structured annotation of each involved segment type (i.e. MovingRegionDS or StillRegionDS). The values that are associated with Who/WhatObject tags are mainly values such as “Face”, “Car”, “Actor_1” etc.

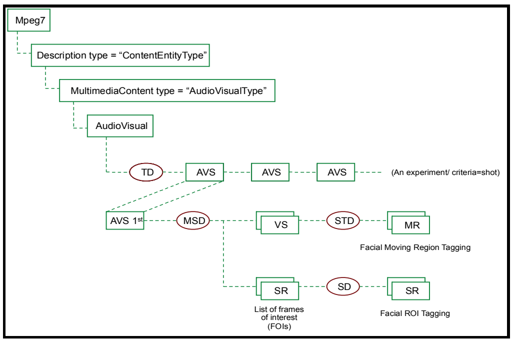

Mpeg7
Description type = “ContentEntityType”
MultimediaContent type = “AudioVisualType”
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=shot)
AVS 1st
MSD
VS
STD
MR
Facial Moving Region Tagging
SR
SD
SR
List of frames of interest (FOIs)
Facial ROI Tagging

Figure 8: Face/Object Clustering schematic representation

3.6 Facial Expression Recognition

Facial expression recognition is used in order to recognize predefined facial expressions such as happiness, anger, fear etc. In the case of Facial Expression Recognition an annotation of a StillRegionDS that defines the image area (bounding box) where a face is depicted is needed. A WhatAction tag with value “affect” is used within the StructuredAnnotation whereas the How tag is used to label the recognized expression.

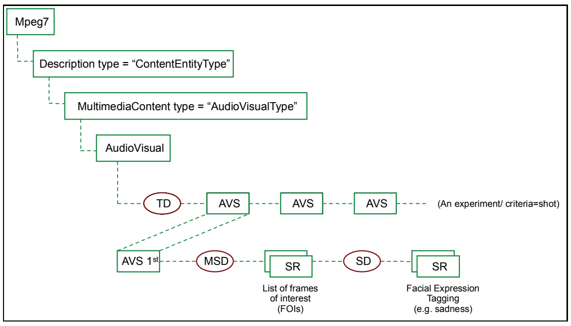

Mpeg7
Description type = “ContentEntityType”
MultimediaContent type = “AudioVisualType”
AudioVisual
TD
AVS
AVS
AVS
(An experiment/ criteria=shot)
AVS 1st
MSD
SR
SD
SR
List of frames of interest (FOIs)
Facial Expression Tagging (e.g. sadness)

Figure 9: Facial Expression schematic representation

4 DEALING WITH 3D VIDEO SPECIFIC ISSUES IN AVDP

4.1 Correspondences and Discrepancies Between Still or Moving Regions in Two or More Channels

When Object/person detection or tracking algorithms, are applied on stereo data (left and right video channels) or stereo data and their respective disparity channels may result in the derivation of correspondences between still regions (e.g. bounding boxes) or moving regions (e.g. bounding boxes in successive frames that correspond to the same object or person) in two channels, e.g. between the right and left video channel. These correspondences denote that the two still regions or moving regions depict the same physical entity (object or person) in the two channels. For example, an object detection algorithm may use the two video channels and detect the depictions of the same physical object in these channels. Thus, this relation/correspondence should be encoded in the XML file. To do so, we use the RelationDS from the AVDP profile. This description scheme allows us to connect two different SegmentDS types (SegmentDS is the abstract parent type for VideoSegmentDS, MovingRegionDS, StillRegionDS as well as many other decomposition types). Typically, the RelationDS type will have a strength component showing the strength of the specific relation, as well as an annotation such as “dialogue”, “handsake” and other to show the nature of the relation.

Furthermore, discrepancies (mismatches) between the content of corresponding (moving or still) regions in the various channels can also be described. Colorimetric mismatch is such a discrepancy. The RelationDS is used again, in order to designate the type of the discrepancies between two such regions. To this end, a RelationDS is inserted in the MovingRegionDS and StillRegionDS instantiation, where the type of the RelationDS designates the type of discrepancy. Note that, there can be many types of discrepancies for a specific region pairs (moving or still).

4.2 Storage of analysis results to video channels, consistency/inconsistency between analysis results derived from different channels

The availability of 2 (left+right or video+depth) or 4 (left + right + left horizontal disparity + right horizontal disparity) or 6 (left + right + left horizontal and vertical disparity + right horizontal and vertical disparity) video channels poses the question of how to analyse them and where to store the derived information. The rule that we have adopted is the following: The derived information will be stored in one, more, or all channels, depending on the algorithm. For example if a tracking algorithm is applied on the disparity channels only, the derived moving regions will be stored in these channels and the video (color) channels may contain no moving regions. Alternatively, one may choose to “copy” the moving regions derived from the disparity channels to the video (color) channels. In the particular case of information related to depth (e.g. extent/position in the z/depth axis of an object), this will be stored only in the disparity/depth channels, since it can be derived only if such channels are available.

Furthermore, in some cases, we may have conflicting results from an algorithm when applied into different channels of the same video. Such cases might arise for example in human activity recognition. If such an algorithm is applied on each channel (left/right) separately, it may derive (due to errors) different activity characterizations (e.g. walking vs. running) of the same segment in the two channels. In this case there are two options, both of which can be adopted:

a) The analysis algorithm itself combines the results stored in the two channels (in a subsequent fusion step) and tags both channels with the same consistent tag using the appropriate AVDP descriptors in the XML file.

b) There is no combination of results in the two channels and each of them is tagged (inconsistency) with a different tag in the XML file.

4.3 Geometrical Reasoning

By geometrical reasoning we define the ability to annotate 3D video content with information related to the geometric properties or relations of objects/ persons. Such properties may refer to the geometrical position of an actor in the 3D world, the geometrical proportions (size) of an actor/object, the spatial relation between two objects (near or far), the speed and direction of movement of an object or person etc. In order to derive such annotations the location/motion information derived from detection and tracking algorithms should be processed.

Tagging objects or persons with semantic concepts, like big/fast etc., requires in most cases a set of rules and thresholds. For example, in order to characterize an object as being centrally located in screen space, a rule (e.g. “the distance of the center of mass of the object from the screen center should be smaller than threshold T”) and a threshold value (e.g. T=0.2) are required. In a more general case, tagging might require an algorithm along with its parameters.

It has to be noted that although depth information is very helpful for 3D semantic content analysis, the extraction of specific geometry related information can be performed even when absolute depth information is not available (hence, only relative depth is known), but also when there is total lack of depth information. Absolute depth information can be extracted if the stereo camera parameters are known. In case camera parameters are unknown, relative depth and, hence, geometry related information can be inferred based on the simple fact that the disparity is inversely proportional to depth. In this way, geometry related information on static objects, can be calculated using the disparity values of objects, as well as the overall disparity range. Similarly, for moving objects, disparity values can be used to obtain relative 3D trajectory of the object (e.g. moving towards the camera, term 2.30). Depth related tags will be stored in the disparity/depth channels only, provided that such channels are available.

The AVDP descriptors where such geometric descriptions will be stored are StillRegionDS, MovingRegionDS and VideoSegmentDS for still and moving objects/actors. StillRegionDS can refer either to an entire video frame or to a spatial region within a frame. We identify the following 4 description cases: a) one still object (alternatively: object properties in a single time instance), b) many objects ROIs, c) one moving object (alternatively: an object behavior during a time interval) d) many moving objects.

5 CONCLUSIONS

In this paper we have presented a new way of using the AVDP profile of the MPEG-7 standard for 3D video content analysis. We have detailed how several known algorithms can be described within such a context and how specific 3D metadata can be incorporated in the schema. The derived description can be used for storing the analysis results in a multimedia database (e.g., a 3DTV content database) or a media asset management system. Such a database can then be queried with high level queries such as “Return the video where such an actor is near the screen” in a 3D context. Finally, note that the extension of the proposed XML description method in order to support video coming from a multi-view setup (i.e., multiple cameras) is straightforward.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n 287674 (3DTVS). The publication reflects only the authors’ views. The EU is not liable for any use that may be made of the information contained therei.

References

[1] Smolic, A.; Kauff, P.; Knorr, S.; Hornung, A.; Kunter, M.; Müller, M.; Lang, M.; , "Three-Dimensional Video Postproduction and Processing," *Proceedings of the IEEE* , vol.99, no.4, pp.607-625, April 2011.

[2] I.S.O, “Information technology – multimedia content description interface - part 1: Systems,” , no. ISO/IEC JTC 1/SC 29 N 4153, 2001.

[3] I.S.O, “Information technology – multimedia content description interface - part 2: Description definition language,” , no. ISO/IEC JTC 1/SC 29 N 4155, 2001.

[4] I.S.O, “Information technology – multimedia content description interface - part 3: Visual,” , no. ISO/IEC JTC 1/SC 29 N 4157, 2001.

[5] I.S.O, “Information technology – multimedia content description interface - part 4: Audio,” , no. ISO/IEC JTC 1/SC 29 N 4159, 2001.

[6] I.S.O, “Information technology – multimedia content description interface - part 5: Multimedia description schemes,” , no. ISO/IEC JTC 1/SC 29 N 4161, 2001.

[7] John P. Eakins, “Retrieval of still images by content,” pp. 111–138, 2001.

[8] A. Vakali, M.S. Hacid, and A. Elmagarmid, “Mpeg-7 based description schemes for multi-level video content classification,” IVC , vol. 22, no. 5, pp. 367–378, May 2004.

[9] J. K. Aggarwal and Q. Cai, “Human motion analysis: A review,” Computer Vision and Image Understanding , vol. 73, no. 3, pp. 428–440, 1999.

[10] J.J. Wang and S. Singh, “Video analysis of human dynamics - a survey,” Real-Time Imaging , vol. 9, no. 5, pp. 321–346, October 2003.

New usability evaluation model for a personalized adaptive media search engine based on interface complexity metrics

Silvia Uribe, Federico Álvarez, José Manuel Menéndez

Visual Telecommunications Applications Group (G@TV), E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid, UPM, Madrid, Spain

{sum, fag, jmm}@gatv.ssr.upm.es

Abstract:
Nowadays, search engines are responsible for allowing the current unprecedented amount of digital data to be quickly and easily accessible to the final users. However, the advantages given by these systems do not assure a complete acceptance of the solution. It is also necessary to provide an efficient, effective and satisfactory product for the users, and this is especially related to the usability level of the development. With this regard, in this paper we present a new usability evaluation model for graphic interfaces. In this model the main process evaluation is done before the implementation. Furthermore, it is based on the application of a set of previous defined screen complexity metrics together with a group of new weight coefficients, which are in charge of providing an optimal algorithm by maximizing the correlation between the metrics' results and users' opinion. Finally, we present the BUSCAMEDIA project, which represents a new semantic search engine for the multimedia environment and an excellent solution for applying this evaluation model.

Keywords: usability model, complexity metrics, weight coefficient, multimedia searching, user interface, content adaptation.

1 INTRODUCTION

During the last few years the number of multimedia content pieces has increased enormously, due to the capacity of users to produce, share and offer their own generated content, becoming "prosumers" [1]. For this reason, the management and the content search have become more difficult tasks. Taking this into account, multimedia search engines have become essential tools for making an unprecedented amount of data easily and quickly accessible.

In another aspect, along these years the user's role in the media chain has changed, evolving from passive users to active ones. In fact, current users participate in the creation of new solutions by providing new challenges to fulfil their needs and expectation and making possible the so called user-centered design.

But the success of a new application is not yet assured even if it represents a complete, innovative, functional and personalized solution. Moreover, the development has to fulfil also a set of specific requirements related to its own operation and presentation, to allow an easy and intuitive execution.

In this way, the usability evaluation of a solution lets us determine the correctness of a solution in terms of effectiveness, efficiency and satisfaction. This assessment is usually done once the implementation process is over by making use of different techniques of users testing. But changing the solution once it has been implemented has an enormous cost, so the improvements are usually not taken into account.

Considering this, a new evaluation model is needed, based on different metrics applicable before the implementation is done. This may helps us to reduce the cost and time required for its improvement.

According to these ideas, in this paper we present a new evaluation model whose main characteristics are not only the moment that it is applied, but also the used techniques. In this way, our model is based on two main elements: on one hand, on the application of a set of complexity screen metrics, which are closed related to usability [18], and on the other , on the establishment of a new group of weight coefficients. Besides, we also present BUSCAMEDIA as a new semantic search engine in order to provide a novel application where apply this new evaluation method.

The remainder of this paper is structured as follows: in Section 2 we are going to present an overview of the technologies and concepts related to this paper's objective. Next, Section 3 is in charge of showing our new search engine BUSCAMEDIA. Then, Section 4 and 5 presents both the new usability evaluation model and the BUSCAMEDIA interface evaluation. Finally conclusions and future work are located in Section 6.

2 STATE OF THE ART

As it was pointed out in [8], nowadays the use of search engines to find specific contents on the web is the most important activity of the Internet. For this reason, during the last few years the search engines development has been linked to the Internet evolution. Primitive solutions as Aliweb [9] (focused on the search of web pages titles) or WebCrawler [9] (based on full text queries), evolve to

Corresponding author: Silvia Uribe, G@TV, E.T.S.I.T UPM, Avda. Complutense 30, 28040 Madrid, Spain, sum@gatv.ssr.upm.es, Phone: +34 913367366 ext. 8073

the current tools, such as Yahoo! [11] or Google [12], But, although these engines are widely used, they are still making progresses and working hard for including an innovative concept, called Semantic Web, introduced by Tim Berners-Lee, and widely explained in [13].

The Web2.0, as an evolution of the Internet, represents a new development environment where the current search engines have tried to combine different functionalities to build event-driven user experience. Even so, current solutions present some lacks related to the multimedia management, such as not making use of the complete multimedia potential in both the input and the output process, or not presenting the content in an optimal way regardless the device and the network in use. For this reason, we have designed and implemented BUSCAMEDIA [2], with the purpose of being an innovative and important solution for the new environment challenges.

Nevertheless, the goodness of the solution is not guaranteed if it doesn't fulfill the user's needs. In consequence, a usability analysis of the development is needed because it provides important data to obtain a good solution, giving a good way to assure a positive acceptance by the user.

Related to this, over the last years many methods have been proposed to assess usability, but most of them are focused in the user's or experts' evaluation of the implemented system ([14] and [15]). This kind of evaluation is called *summative evaluation*, and it takes place after the product or at least a final prototype version of it is ready. Although this assessment usually reveals many usability problems, it is costly and time-consuming, so sometimes it is difficult to make the necessary improvements.

Given these points, we have focused in the usability since the earliest stages of the project, which is called *formative evaluation*. According to [16], the application of different metrics in the design stage provides some important benefits as:

- Improving software: a proper usability engineering leads to a usable software, which means satisfied customers.
- Saving user's money, because a usable solution provides more efficient tools to achieve the user's objectives.
- Minimizing engineering costs: the earliest the designer detects a usability mistake, the cheapest the change is. Besides, the costs of the iterations are low if they are made before the implementation.

This formative evaluation is included in the usability predictive models, whose main objective is to establish the usability level of a product in the first design steps in order to reduce the number of iterations and, with this, the total cost of the solution.

Taking that into account, we propose a new evaluation method for search engine interfaces based on screen complexity metrics. We have chosen these metrics because they don't need the participation of usability experts in the evaluation to obtain the results, like in other methods. This allows the universalization of its application even in small organizations, where usability does not count on specific research workers.

3 BUSCAMEDIA AS A NEW MULTIMEDIA SEMANTIC SEARCH ENGINE

BUSCAMEDIA [2], as a new semantic search engine, is based on a new ontology, and it is able to adapt itself to any network, device, context and user in a dynamic way.

The searching process is done by means of natural language. This is possible thanks to both an advanced content annotation system and the use of the 'M3 ontology', which represents a Multilingual, Multidomain and Multimedia ontology.

Besides, another important characteristic of the solution is the adaptation of the search results in order to present them to the users in an optimal way, according to their devices, preferences and contexts.

Regarding these capabilities, the main advantages of BUSCAMEDIA are in consonance with the new available possibilities in the Internet, which are pointed out in [3]. According to this, the relation between these points and the BUSCAMEDIA's characteristics are:

- To provide output results according to the query done by the user: one of the main requirements of BUSCAMEDIA is to record and analyze user's habits. Hence the system can process this information and then model the user behaviour. These data, in combination with the user's preferences and needs, helps the system to provide the search result in the optimal type and format. The adaptation and the personalization module of BUSCAMEDIA is in charge of doing these tasks.
- To apply the Web2.0 concept on the search engines, in order to provide user-oriented and personalized services. In this way, BUSCAMEDIA enables users to interact not only with the system, but also with the results. This helps the users to get the best information in a personalized way.

Furthermore, another important area of research and improvement in BUSCAMEDIA is to assure a good quality of experience in the use of the solution. This quality is not only related to the content itself, but also to its presentation, personalization, interaction with and so on. Therefore, and important research for modelling and measuring quality is done. And to that end, this research work includes the usability analysis of the system, which is in charge of helping us to obtain a optimized design of the interface.

4 NEW USABILITY EVALUATION MODEL

Since its beginning, the usability concept has been usually compared to other terms to clarify its meaning. In this way, although there are several different definitions in the

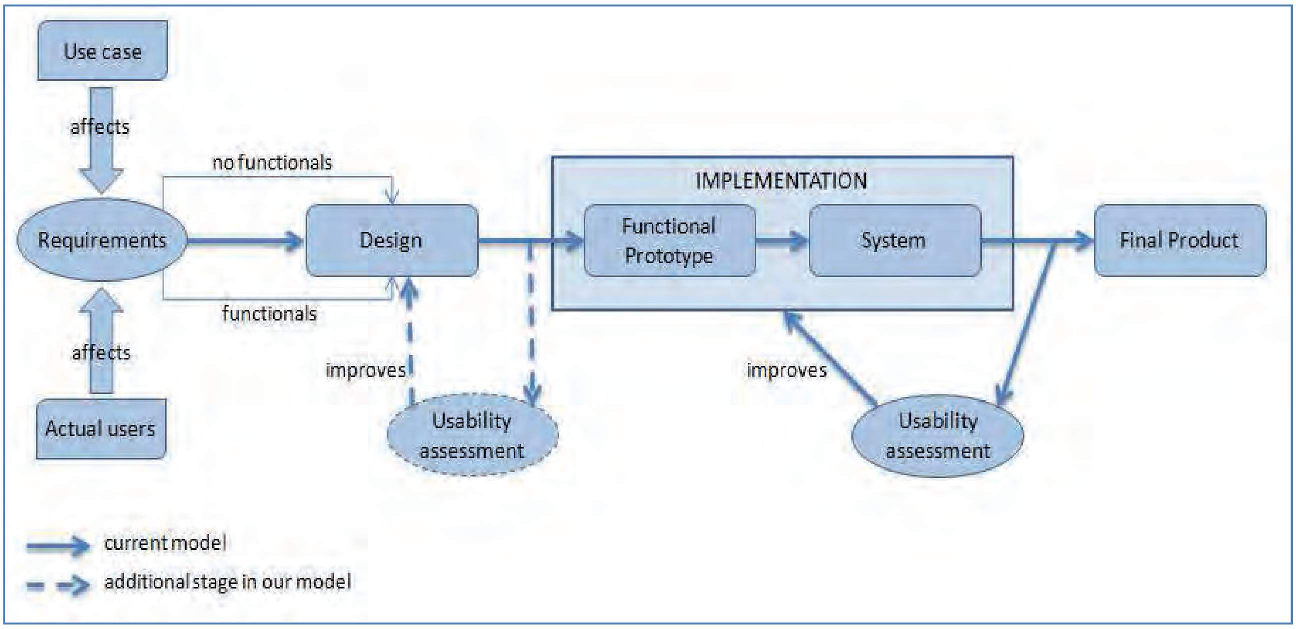


Fig. 1. New usability evaluation model for multimedia search engines interfaces

interface literature, such as [14] and [20], in short usability can be explained as *the capacity of a product* or a system to provide and efficient, effective and satisfactory way to achieve specific goals in a particular context of use. Its main dimensions are assessed:

- **effectiveness** can be evaluated through the number of necessary steps to achieve a specific task.
- **efficiency** can be measured by the number of particular tasks finished in a period of time, or the ease of use of the solution.
- **satisfaction**, that can only be measures directly by users.

Fig. 1 presents our evaluation model proposal based on formative assessment. We keep the current life cycle of a common software implementation, but the main difference among other models is the inclusion of a new evaluation stage focused on the design process.

As we can see, the process begins with the requirements establishment, which are deeply affected by both the use case defined and the regular users who are going to interact with the system.

After that, two main stages come to obtain the final product. First of all, a full design of the solution, based on the previous obtained requirements. Then, the implementation itself, composed by two subtasks: the obtaining of a functional prototype, whose main objective is to provide a tool to prove the system before the complete implementation, and the final implementation of the system. In the current model, the usability assessment is done after the system implementation with real users. This provides useful information about the solution execution, but it makes the changes to be done more costly in terms of implementation effort, time consumption and money.

In order to improve the solution quality, to save customer's money and to minimize engineering costs, we propose to include an additional usability evaluation stage to be applied after the design of the solution and before the implementation begins. This allows finding usability problems in the earliest stages of the process, before the real implementation starts. Besides, this evaluation does not need the user's participation, which makes the iterations to correct usability errors easier and faster.

Furthermore, this previous evaluation allows detecting the main usability problems, which can be corrected in the next design iterations. Thanks to that, the usability assessment to be applied after the implementation is more agile, taking into account the correlation between the metrics to be applied in the design stage and the usability test performed by the users over a functional system presented in [17]. Other design metrics could be applied in this evaluation according to their definitions, such as the predictive metrics defined in [15], but these metrics make necessary a wide knowledge about both the interface operation and its semantic information. By contrast, the selected metrics are based only in the analysis of the solution's aesthetic, which provides important results to determine the usability level of the search engine interface, as it is shown in [18].

Regarding this, in Section 5 is in charge of presenting the BUSCAMEDIA usability evaluation by means of complexity metrics. This assessment is going to be applied to a design of its main interface, in order to detect the main usability problems before the implementation starts. Furthermore, with the aim of providing an optimized result, a new metrics weighting is going to be applied, based on a modification of the one presented in [17].

5 INTERFACE EVALUATION

In this section we present the BUSCAMEDIA interface evaluation. According to the previous model, this evaluation is based on different complexity metrics, which help us to obtain a clear assessment of the usability level of the solution before its implementation.

Fig. 2 shows, on the left, the interface which is going to be evaluated. On the right it presents the interface

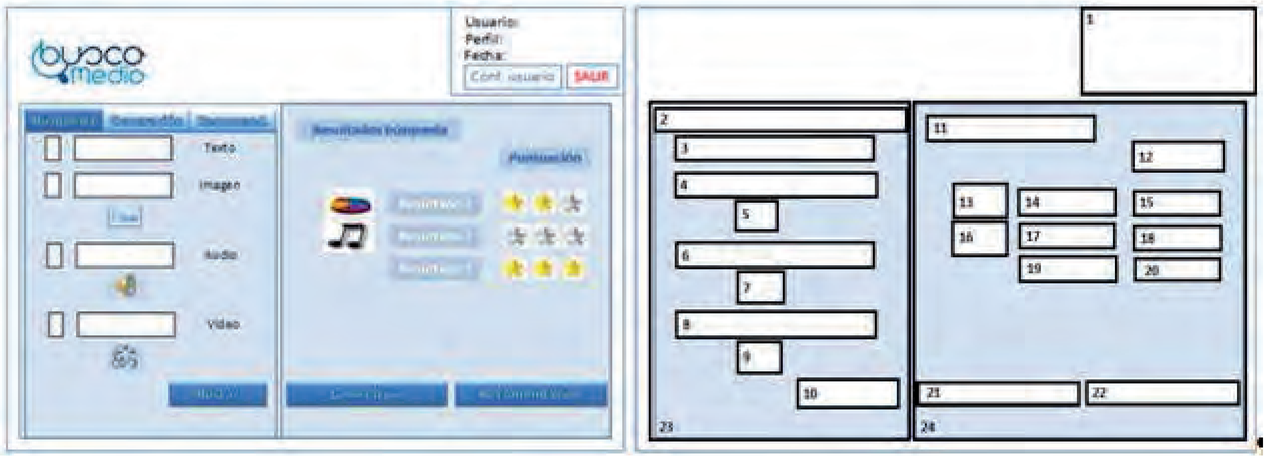


Fig. 2. **BUSCAMEDIA interface (left) and its model (right).**

model obtained according to [21], which facilitates the interpretation of the data analysis in interface aesthetics. Moreover, the elements' sizes are presented in Table 1.

Table 1. interface elements' size

Element	Size	Element	Size
1	179x81	13,16	55x31
2	253x23	14,17,19	99x25
3,4,6,8	201x43	15,18,20	87x25
5	43x25	21	161x21
7,9	48x29	22	155x21
10	101x25	23	254x313
11	170x23	24	335x313
12	92x26	Interface	631x413

The metrics used in this evaluation are a set of complexity metrics obtained in [17], as a specification of the previous work for web pages. In fact, they are derived from the research work done in [22] and [23], which was later synthesized in [21] with the development of fourteen aesthetic measures for graphic displays. The following subsections explain them and show their application over a real prototype.

5.1 Size Complexity

This metric is in charge of categorizing the interface's elements into groups according to their different sizes by applying:

$$CS = 1 - \frac{\sum_{I}^{type}(n_{size} - 1)}{n} \in [0,1]$$

where n_{size} is the total number of different sizes and n is the total number of elements.

Taking into account this definition and according to the different elements' sizes and the number of elements (n_{size}=15 and n=24), the size complexity of this interface is CS=0.4166.

5.2 Local Density

It is in charge of evaluating the density of the interface. Considering the 50% of the interface the optimal percentage for graphic screens, it is calculated as follows:

$$LD = 1 - 2\left|0.5 - \frac{\sum_{i}^{n} a_i}{a_{frame}}\right| \in [0,1]$$

where a_i and a_{frame} are the areas of object i and the frame. Taking into account the interface and main elements (No. 1, 23 and 24) sizes, the local density of the solution is LD= 0.484.

5.3 Group Complexity

This metric is in charge of evaluating the coherence of a solution by measuring the degree of which different elements appear to be as one and only piece. In this case, the grouping complexity is evaluating as follows:

$$CG = \frac{g_i}{g} \in [0,1]$$

where g_i represents the number of groups with a clear boundary and g is the total number of groups. Taking into account that every object of the interface is grouping with similar elements by a clear boundary line, the grouping complexity of this solution is GC=1.

5.4 Alignment complexity

In order to simplify the use of the interface, the fewer number of different alignment points, the better, due to its influence over the searching time on menus. This metric is calculated as follows:

$$AS = \frac{3}{(n_{vap} + n_{hap} + n)}$$

where n_{vap} and n_{hap} are the number of vertical and horizontal alignment points and n is the total number of objects in the layout. Taking into account the position of the different elements, there are 14 different horizontal alignments and 22 verticals, with 24 elements in total. That provides an AS= 0.05

Once applied these four metrics, the next step is to provide a global result for the usability level evaluation.

In [17] the authors explain that the complexity value of the evaluated interface obtained by means of the average of the four metrics differ from the users' evaluation. For this reason, we have found a set of different weighting coefficients to be applied to their metrics results in order to maximize the correlation between them and their users' evaluations. In fact, these coefficients have been obtained by determining their influence over the final count of the metrics result when they are compared to the users' evaluation.

Considering the information about the users' preferences and the metrics results provided in [17], we have obtained these four coefficients as follows:

- First, we obtain the correlation between the users' opinions and the metrics results for the four different graphic interfaces analyzed.
- Then, according to this matrix, we determine which metric has the bigger impact in the users by detecting the higher value of the correlation with users' opinion. This metric is CG.
- Finally, we obtain a different coefficient for each metric according to the lineal relationship between them and the correlation matrix and giving the highest value to CG in the form:

$$LD = \frac{corr\ (LD, Users)}{corr\ (CG, Users)} = 0.24$$

$$AS = \frac{corr\ (AS, Users)}{corr\ (CG, Users)} = 0.69$$

$$CS = \frac{corr\ (CS, Users)}{corr\ (CG, Users)} = 0.68$$

Taking this process into account , the different coefficient weights for each metric are:

CG=38%, CS= 26%, LD=9%, AS= 27%.

Finally, Table 2 shows the difference between the usability assessment obtained by applying these weight coefficients and the average of the metrics results,

Table 2. Usability evaluation results.

CS	LD	CG	AS	Aver.	New Weight
0.4166	0.49	1	0.05	0.48915	0.5459

As we can see, this new weighting process provides a higher result than the simple average, although both of them represent high level of usability in the interface. Furthermore, thanks to the coefficients obtained from the data in [17], we are able to say that this result would be closer to the users' evaluation than the metrics average.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a new usability evaluation model which includes an assessment stage at the design process. This is done before the implementation in order to solve usability problems in an early stage, making the future evaluation easier, more agile and with lower costs. Furthermore, according to this model, this evaluation is based on the application of complexity screen metrics, which are in charge of determining the usability level of the interface. In fact, the obtained results provide important information about the goodness of the interface in terms of usability, giving the possibility of improving it by maximizing each metric.

Beside this, we have presented a new weighting process for providing the complexity metrics results, with new formulae based on the identification of four different weighting coefficients. These coefficients have been proved over the actual results show in [17] and, thanks to them, we obtain a closer result to the users' opinion.

With regards to the future work, there are two main research lines. The first one is related with the purpose of confirming the goodness of these weighting coefficients by checking them with real users' opinion in this case. Then, the second one is to determine if the usability level noticed by the users depends on the specific user, that is, if we can establish a personalized usability scale according to the user's characteristics and preferences.

Acknowledgment

This publication is based on the work performed in the framework of the Spanish national project BUSCAMEDIA (CEN- 20091026), which is partially funded by the CDTI - Ministry for Science and Innovation, and the AMURA project (TEC2009-14219-C03-01).

References

[1] W. Gerhardt, "Prosumers. A New Growth Opportunity", Cisco internet Business Solution Group (IBSG), March 2008.

[2] M. Alduan, F. Sanchez, F. Alvarez, D. Jimenez, J.M. Menéndez, C. Cebrecos, "System Architecture for Enriched Semantic Personalized Media Search and Retrieval in the Future Media Internet". IEEE Communications Magazine, vol. 49 issue 3, pp. 144 - 151. March 2010.

[3] S. Huang, T. Tsai, H. Chang, "The UI Issues for the Search Engine". 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, pp:330-335. 2009.

[4] P. Langley, "User Modeling in Adaptive Interfaces". Proceedings of the 7th International Conference on User Modeling. Banff, Alberta: Springer, pp. 357-370. 1999

[5] S. Romitti, C. Santoni, P. François, "A Design Methodology and a Prototyping Tool dedicated to Adaptive Interface Generation". Proceedings of the 3th ERCIM Workshop, Obernai, France. 1997

[6] N. van Meurs "Context-aware and Adaptative Interfaces". European Media Master of Arts in Interaction Design. 2008-2009.

[7] L. Arnáiz, J.M. Menéndez, D. Jiménez, "Efficient Personalized Scalable Video Adaptation Decision-taking Engine based on MPEG-21", IEEE Trans. on Consumer Electronics. Vol 57, n.2, pp.763-770. 2011.

[8] W. Wiza, K. Walczak and W. Cellary, "Periscope: a System for Adaptative 3D Visualization of Search Results". Proceedings of the 9th international Conference on 3D Web Technologies. ACM, 2004.

[9] Aliweb, http://www.aliweb.com/. 1993.

[10] WebCrawler, http://www.webcrawler.com/. 1994.
[11] Yahoo!, http://search .yahoo.com/. 1994.
[12] Google's corporate history page, http://www.google.com/about/corporate/company/history.html
[13] S. B. Palmer, "The Semantic Web: An Introduction". http://www.infomesh.net/2001/swintro/. 2001.
[14] J. Nielsen, "Usability Engineering". Ac. Press, Boston, MA, 1993.
[15] L.L. Constantine, L. A. D. Lockwood, "Software for use", Addison-Wesley, pp. 417-442. 1999.
[16] E. Folmer, J. Bosch, "Architecting for Usability; a Survey", Journal of Systems and Software, issue 70-1, pp. 61-78. 2004.
[17] F. Fu, S. Chiu, C. H. Su, "Measuring the screen complexity of web pages", Proc. Of the Conference on Human interface: Part II, pp. 720-729. 2007.
[18] T. Miyoshi, A. Murata, "A method to evaluate properness of GUI design based on complexity indexes of size, local density, aliment, and grouping," IEEE International Conference on Systems, Man, and Cybernetics, vol.1, pp.221-226, 2001.
[19] A. Parush, R. Nadir and A. Shtub. "Evaluating the layout of graphical user interface screens. Validation of a numerical computerized model". International Journal of Human-Computer Interaction, 10 (4), pp. 343-360. 1998.
[20] B. Shackel, "Usability - Context, Framework, Definition, Design and Evaluation". Human Factors for Informatics Usability. Eds Cambridge University Press, pp. 21-38. 1991
[21] D.L.C. Ngo, L.S. Teo, J.G. Byrne, "Modeling Interface Aesthetics" Information Science 152, pp. 25-46, 2003.
[22] T.S. Tullis, "Screen Design, Handbook of Human-Computer Interaction", Elsevier Science Publisher, 1988, 377-411
[23] A. Sears, "Layout appropriateness: guiding user interface design with simple task descriptions". IEEE Transactions on Software Engineering 19 (1993), 707-719.

Connected Media Worlds II

Session 2B

Chaired by Andy Bower, BBC

HTTP Adaptive Streaming: MPEG-DASH Proxy for Legacy HTML5 Clients

Stefan Kaiser[1], Stefan Pham[1], Stefan Arbanowski[1]

[1]Fraunhofer FOKUS, Berlin, Germany

E-mail: [1]{stefan.kaiser, stefan.pham, stefan.arbanowski}@fokus.fraunhofer.de

Abstract: **The importance of Dynamic Adaptive Streaming over HTTP (DASH) has increased in the last months, but up to now, Web browser-based solutions are still in development. Currently, various proprietary solutions exist for HTTP adaptive streaming, thus a standardized adaptive streaming solution, such as DASH, is needed to unify video delivery across the Internet. We propose a streaming proxy architecture that enables DASH in browser-based environments. This solution enables clients using only the plain HTML5 media element to play DASH-compatible content. Advantages and limitations of this solution will be discussed based on usage scenarios.**

Keywords: DASH, client, browser, adaptive streaming, legacy, proxy, HTML5, progressive download

1 INTRODUCTION

Audio and video streaming over Hypertext Transfer Protocol (HTTP) has received much attention in recent months due to its ability to be accessed by any consumer device that is connected to the Web due to increasing bandwidths and affordable provider contracts. In addition, advances in streaming like dynamic adaptation on changed connection quality or serving media resolution depending on the current demands of the user's device are applied to HTTP streaming as well. The recently published ISO standard Dynamic Adaptive Streaming over HTTP (DASH) [1] defines media presentation and content formats and is growing in popularity as well as becoming generally accepted by most vendors of pre-existing technologies. It has various possible use cases in the field audio and video content distribution, including live, on-demand and premium content. This area is as popular as never before with increasing customer numbers for services such as Spotify [2] or Netflix [3] (represents 29.7% of North American Peak Downstream Traffic [4]). In combination with ubiquitous computing, from computer via tablet and smartphone through to TV, over any network, the need for DASH support in browsers for these devices is growing as well. Streaming media has become a commodity. Moreover open standards such as HbbTV have adopted MPEG-DASH in recent specification releases (HbbTV Version 1.5 [5]). However, current efforts in adding adaptive streaming technologies in browsers (not via plug-ins) are still a work in progress and legacy browsers, with no DASH support ever added, will continue to exist.

One way for a content provider to support a wide range of clients is to provide multiple representations of the media (e.g. different formats, resolutions and bitrates) to serve content compatible with the clients' capabilities. In order to support both DASH-capable clients and so-called legacy clients (no DASH support), a Content Delivery Network (CDN) would have to store media data at least in duplicate, in DASH compliant segmented format and as plain media files. Since segmented DASH content itself consists of multiple representations this leads to a big storage overhead due to various combinations of audio and video representations, as well as more traffic, which in turn leads to higher costs for the content provider. The following proposal of a DASH proxy for legacy clients allows a content provider to serve all clients with DASH content as the only source, which also means that the media has to be encoded only once. Thus, storage demand is decreased and costs are saved.

In this paper we present a DASH proxy as concept to enable DASH-unaware clients (legacy clients) to handle DASH content and take advantage of adaptive streaming features. With this novel approach we demonstrate a DASH client acting as proxy between a legacy client and the DASH server. The proxy server includes a file server emulator, serving a virtual file created on demand which is then requested by the legacy client using progressive download. This combination enables the client to receive and play DASH content in the form of a plain video file. This proxy approach aims to be a solution for HTML5 Web clients until native DASH support is adopted. Further, in the long run the proxy can be used for niche clients, which are not further developed or will not support DASH.

This paper is structured as follows: Section 2 introduces MPEG-DASH and section 3 gives a detailed view on related work regarding adaptive streaming support in browser. Section 4 covers the proposed DASH proxy approach. Section 5 discusses pros and cons of the approach. The paper ends with a discussion and a summarizing outlook.

2 MPEG-DASH

The Web is ubiquitous in nearly all areas of current media services. According to statistics the amount of video and audio content already represents the biggest part of overall Internet traffic and is still growing, according to :

Corresponding author: Stefan Kaiser, Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany, stefan.kaiser@fokus.fraunhofer.de

"The sum of all forms of video (TV, VoD, Internet, and P2P) will continue to be approximately 90 percent of global consumer traffic by 2015" [6].

Traditional video delivery of live and on-demand content is realized by streaming over protocols like Real Time Messaging Protocol (RTMP) or Real Time Streaming Protocol (RTSP). These protocols can cause problems in NAT/firewall traversal, leading to user frustration [7].

Plain Web video delivery is currently realized by a technique called progressive download over HTTP, which circumvents the problem of NAT/firewall mentioned above. For large files it is unnecessary to wait until the whole media is downloaded so that the client starts playback just when enough data is available. HTTP progressive downloading allows media to be downloaded to a temporary place on the hard disk (for caching). This local temporary file is then used for playback.

Video streaming on the Web is done using browser plugins like Adobe's Flash [8] or Microsoft Silverlight [9] to wrap the native controlling of video streaming using, e.g. RTMP. These types of plugins have to be installed by the user manually. Therefore advances in Web technologies, e.g. HTML5 video tag [10], try to minimize the need for third-party plugins, by enabling built-in Web browser support for media streaming.

Now that available bandwidth increases (e.g. 3G, LTE etc.), the user is able to watch video with any device, from anywhere and at any time. But bandwidth is not constant all the time (e.g. in case of roaming) and it is possible that the video freezes and does not play fluently. Another aspect is the video content itself. The enormous number of devices differing in resolution and media format support is a problem for content providers (device fragmentation). To minimize the bandwidth used, the user should get the representation of the content that fits best to the device.

The concept generally known as HTTP adaptive streaming aims at improving the user experience by addressing the limitations previously mentioned. It uses the HTTP protocol to prevent media accessibility issues and enables seamless switching between representations (e.g. in terms of resolution, codec or bitrate) that fit the client's device and network connection best. Besides quality improvements, faster start up times for media playback can be expected. This is realized by segmenting video streams, on provider side, into small pieces, which are transferred via HTTP and are concatenated again on client side.

Adaptive streaming technologies are growing in popularity for content providers. They can be deployed on top of CDNs and hereby allow reuse of existing scalable and efficient HTTP servers for transport and caching. Hence proprietary efforts from major players like Apple's HTTP Live Streaming [11], Microsoft's Smooth Streaming [12] and Adobe's HTTP Dynamic Streaming [13] that all do HTTP adaptive streaming in a similar way, but are not interoperable, exist. As a consequence the 3GPP defined an open standard for HTTP streaming combining the advantages of existing technologies (container formats, DRM etc.) in one standardized specification [14]. After refinement of MPEG this lead to the MPEG DASH standard (see Figure 1) called "Dynamic adaptive streaming over HTTP" published as an international standard by ISO/IEC in April 2012 [1].

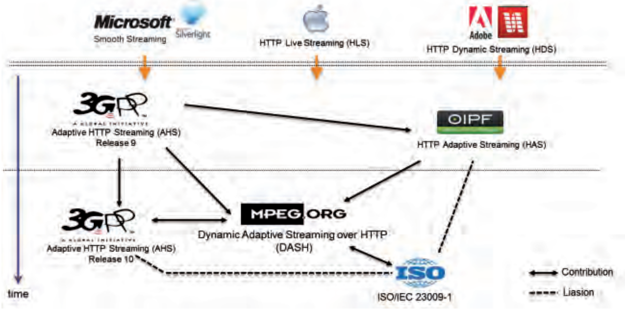

Microsoft
Smooth Streaming
Silverlight
HTTP Live Streaming (HLS)
Adobe
HTTP Dynamic Streaming (HDS)
3GPP
A GLOBAL INITIATIVE
Adaptive HTTP Streaming (AHS)
Release 9
OIPF
HTTP Adaptive Streaming (HAS)
3GPP
A GLOBAL INITIATIVE
Adaptive HTTP Streaming (AHS)
Release 10
MPEG.ORG
Dynamic Adaptive Streaming over HTTP
(DASH)
ISO
ISO/IEC 23009-1
Contribution
Liasion
time

Figure 1 MPEG-DASH combines existing standards

MPEG DASH specifies the two segment formats ISO base media file format (ISOBMFF) and MPEG-2 Transport Stream (M2TS). It is codec independent, thus existing codecs used for adaptive streaming technologies can be implemented. The XML manifest format, which is called the Media Presentation Description (MPD) (see Figure 2) describes alternative streams (e.g. codec, DRM, language, resolution or bandwidth), their respective HTTP URLs and timing information. Further, DASH specifies support for trick modes (seeking, fast-forward and rewind) and enables common encryption with the help of content descriptors for protection.

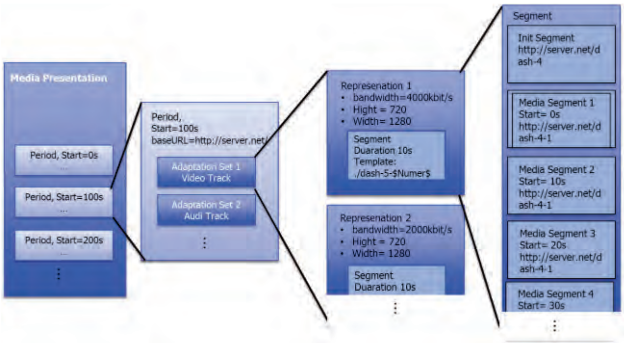


Figure 2 Media Presentation Description (MPD) Data Model [7]

3 ADAPTIVE STREAMING WITH HTML5 CLIENTS

The majority of currently available DASH clients consist of native implementations, such as the Osmo player [15], or media player plugins, e.g. VLC [16]. As far as we know, DASH is not yet natively supported by any browser. However, there are efforts to enable HTTP adaptive streaming in HTML5 with major contributions of the Web Hypertext Application Technology Working Group (WHATWG) and the W3C Media Pipeline Taskforce (MPTF). Summarized in [17] and extended by Duncan Rowden in [18] three different types of adaptive streaming facilitation in the Web browser have been characterized:

1. Native handling of content and adaptation

2. Native handling of content, but scripted adaptation
3. Scripted handling and adaptation of content

The first approach is transparent for the user as well as for the application. The manifest file describing the content and adaptation possibilities is set as source of the media element and is completely handled by the browser. Regarding DASH this means that it is the browser's responsibility to parse the Media Presentation Description (MPD), start fetching segments and feed the media player with data. Bandwidth measurements and proper adaptation of the representation to fetch have to be done natively in the browser in case of bandwidth variations. An example for this kind of support is already implemented in the Safari browser on iOS. Safari supports Apple's HTTP Live Streaming [19] by specifying a m3u8-manifest file as source attribute of a HTML5 video element.

The second approach allows the user to influence the adaptation process by offering manual switch functionality between different quality levels from Web application code. Parsing of the MPD and fetching of segments is still done by the browser. This also means that the Web application has to take care of bandwidth measurement and a dynamic switching model.

The third kind of adaptive streaming support shifts the whole client logic to the web application. Parsing of manifest files like MPD is done in script as well as fetching media data and bandwidth measurements with an appropriate dynamic switching model. The application logic fetches the media data that it assumes to be appropriate and passes them to media element. Subsequent segment data will be appended to the media element. For this purposes a comprehensive proposal for changes to the media element API is made by Aaron Colwell et al. in the Media Source Extensions draft [20].

To sum up, all mentioned approaches differ in the effort for implementation. The first two of them handle parsing of manifest files and segment fetching natively, which leads to the need for additional capabilities in the browser's source code. The first approach handles adaptive quality switching logic internally. The second approach does adaptation on client side, thus needs a smart logic implemented in the Web application to ensure a fine viewing experience. As a consequence this requires appropriate possibilities for the client to tell the user agent what to do. This can be done using the interface of the HTML5 media element, which can be accessed via JavaScript. Compared with that, the third approach needs more enhancements in the scriptable media element interface, since it has to be able to receive segments for appending to the media queue that will be played consecutively by the media element. In return, manifest interpretation and handling of segment fetching as well as adaptation is done in web application code (e.g. using XMLHttpRequest) and does not have to be implemented in the user agent.

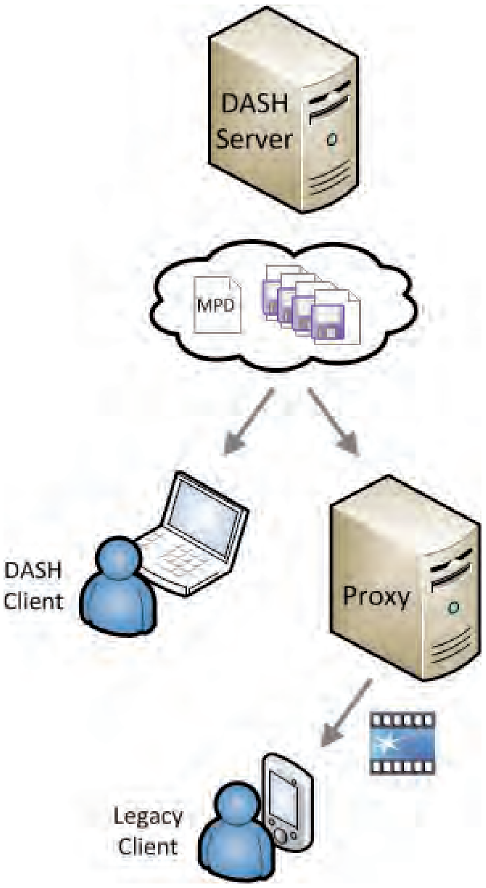


Figure 3 Overall DASH architecture including the DASH proxy

4 DASH PROXY ARCHITECTURE

Our proposed proxy approach can be placed into the classic DASH architecture as illustrated in Figure 3. The DASH server provides media presentation description (MPD) files and corresponding media segment files. A DASH client is able to use this kind of media representation to play a video and/or audio. However, not every client is aware of DASH at the moment and some may not be in the future. We call this kind of clients legacy clients and our proxy works as an intermediate between a DASH server and a legacy client. The proxy acts as DASH client and is able to understand DASH content. With this content it emulates a media file and provides this to the legacy client. From the view of the legacy client it seems to be a normal media file that can be smoothly embedded into a Web application.

The detailed workflow of the proxy is depicted in Figure 4. As can be seen, the DASH proxy architecture consists of two main components: Firstly, the DASH client and secondly the Media File Emulator. To initiate a client session to access DASH content, the Legacy Client sends a request containing the URI of the MPD, which the client is not able to play, to the DASH proxy. The proxy creates an internal session and forwards this URI to a new DASH client session, which is then mapped to a new media file URI generated by the emulator. This media file URI is then returned to the legacy client and can be used, e.g. as source attribute in a HTML5 video element.

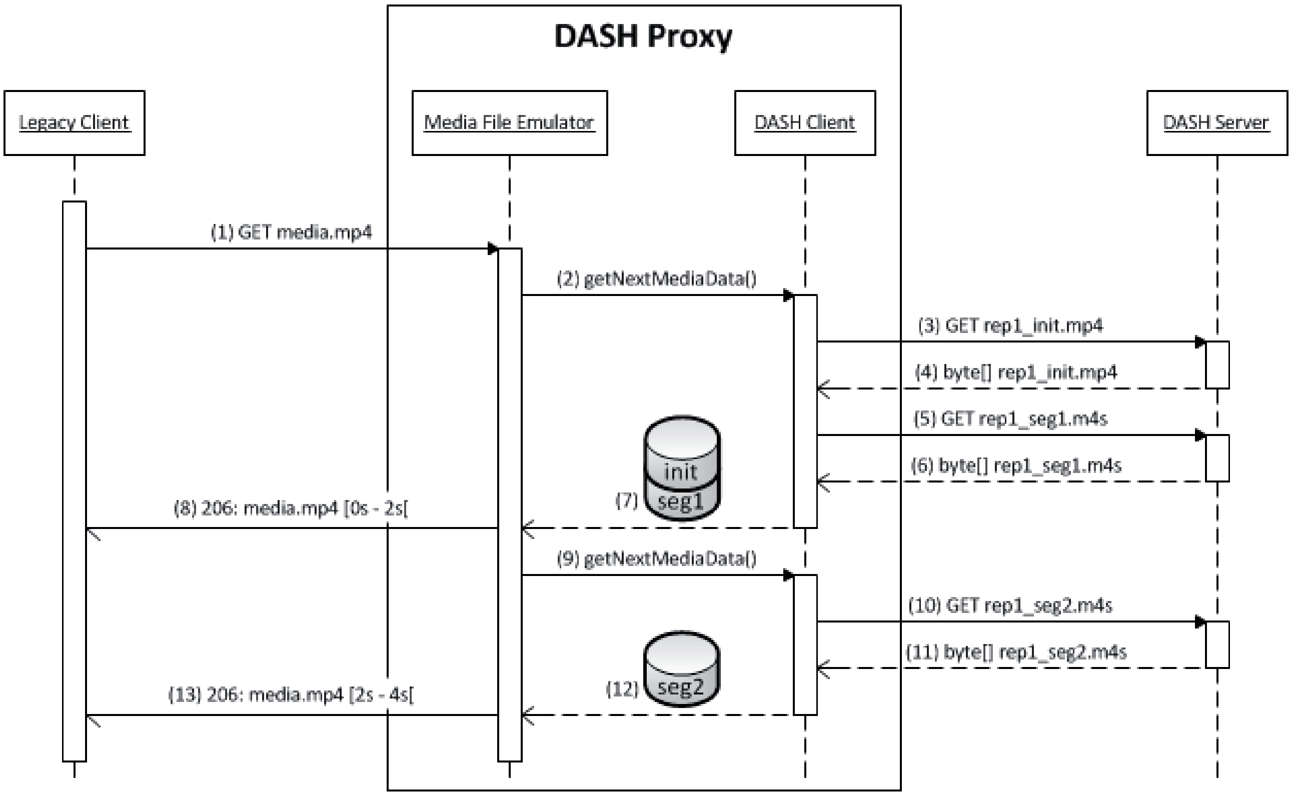


Figure 4 Example information flow of DASH proxy on client request

4.1 DASH client

The DASH client, which is part of the DASH proxy, is utilized for media retrieval from a DASH server and covers the specific DASH mechanisms by providing an interface for requesting the next media data package. It is responsible for fetching and parsing the MPD and interpreting it, in order to be able to fetch media segments. An internal model stores the MPD interpretation as long as the client session lasts. This enables adaptive switching between representations and easy processing of requests for subsequent media segments.

To satisfy the requirements of the "minBufferTime" attribute specification and in order to make these DASH mechanisms transparent, only media data packages are returned. A media data package consists of the number of media segments needed to reach the minimum buffer time.

Media segments within a media data package are aligned according to their playing time. For the first media segment, the initialization segment has to be prepended.

For validation we used ISOBMFF-based (ISO base media file format) [21] MP4 files. Segmenting these files in DASH manner leads to fragmented MP4. Modifications to the file structure may be necessary for some clients to enable playback. However, using content created in such way, without modifications, worked for us in latest Google Chrome browser [22].

4.2 Media File Emulator

Media file URIs used in the source attribute of the HTML5 video element do not need to be downloaded entirely to start playback. Modern browsers support progressive download, which downloads only a minimum amount of content to start playback and then downloads the rest of the media while playing it.

To allow the client to do the same, the DASH proxy generates a virtual media file with a specific URI. This URI is internally mapped to the MPD URI that the client would like to play originally. In doing so, incoming requests for this media file can be processed in combination with an established DASH client session for the corresponding MPD. This also means that the proxy acts as the server side of subsequent progressive download requests and has to handle them properly.

Progressive download requests typically utilize range header fields of HTTP to specify the next data to be responded. Range header fields are specified in byte unit, but segment size in the MPD is specified in time unit. Therefore this byte unit has to be translated by the internal session management.

On request, the next desired media data package is identified and requested by the DASH client. The legacy client will have its own minimum buffer time so that more media segment data than specified as the MPD-specified minimum buffer time will be requested before the playback begins.

4.3 Adaptive Client Behaviour

Adaptation to current client bandwidth conditions is usually done by switching between representations within an adaptation set defined in the MPD. Representations either differ in media resolution or bandwidth. Obviously, the most significant attribute is bandwidth and thus has to be measured in a proper way. For regular DASH clients this can be done on an application level or simply by measuring the time of segment fetching divided by size of the segment. In regards to the DASH proxy, measurements can be done in this simplified way or in a more complex way, since there are two components involved in media retrieval in addition to the DASH server. There are two paths that the data segment has to be transferred on. The first path goes from DASH server to DASH proxy and the second path from DASH proxy to legacy client.

The first trivial approach is realized on the proxy's side. It is the same as for a native DASH client, which the DASH proxy basically represents in the architecture. Dependent on the bandwidth measurements, the representation can be switched automatically and will be delivered in subsequent legacy client requests. However, this approach does not consider the bandwidth situation on legacy client side, except the proxy runs on the user agent.

In opposite to the trivial approach, a second more sophisticated one enables the client to influence the proxy's adaptive behaviour. The precondition for this is to have a client side script communicating with the DASH proxy. This script manages the bandwidth measurements for the DASH proxy to legacy client path and informs the proxy about the results. Even more, the proxy could provide capabilities to directly trigger representation switches to lower or higher quality from client side. The proxy behaviour has to consider both paths for bandwidth computations to serve the legacy client adaptively for a fluent media playback.

5 DISCUSSION

Our approach described above is designed with modularity in mind, so that it is applicable as a standalone server independent of a CDN providing DASH content. This allows any client, including so called legacy clients, to play back DASH content from any DASH server source. Obviously, this will utilize two separate HTTP transfers of the same data. This also means the time for media delivery can double. On proxy side the buffering of media segments that are expected to be fetched by the client can exceed the amount of data actually requested by the client. If the proxy to DASH server connection is faster than or equal to the legacy client to proxy connection, legacy client requests for next parts of media data are already available at proxy side and can be served with no trade-offs in terms of media representation.

For a CDN provider, who would like to use DASH as a single format for media delivery, it would make sense to deploy a DASH proxy in the CDN architecture. Adaptation requires the additional communication between proxy and legacy client, as there can be no assumptions made about the client's network connectivity from measuring fetch times between DASH server and proxy. Within the CDN, the segment files can be accessed via a local network, which on the one side is faster than the Web and on the other side makes the first path mentioned in 4.3 obsolete or at least can be neglectable. A further positive aspect of internal proxy deployment is the convergence of response time on legacy client side to that of a real DASH client. The bottleneck for client-server communication is basically reduced to the path between client and DASH proxy.

Another aspect that is critical for comparison with other DASH client implementations in benchmarks is the minimum buffer time of a legacy media player (HTML5 video object implementation). In DASH the minimum buffer time is specified by the corresponding attribute in the MPD. In the browser's media player this seems to be an implementation dependent value that cannot be influenced. This leads to different playback start times that are probably higher than a native DASH client according to the value defined in the MPD.

The current stage of our proposed proxy architecture to enable DASH on legacy HTML5 clients is in an early proof of concept phase. Future works will focus on support of more media formats (including live content), the possibilities and drawbacks of using DRM in a proxy solution and more details on deployment in existing CDNs.

6 CONCLUSION

With the DASH proxy approach we propose a solution to enable common HTML5 clients to use DASH content. Today not many native DASH clients are available and implementation work for built-in browser support or corresponding APIs are still in progress.

With DASH support growing and implementations improving, there will be DASH client models as described in chapter 3 available sooner or later. Two approaches that have been identified in this paper describe the use cases for the DASH proxy solution.

The first approach is the utilization as a standalone proxy. It basically acts as a native DASH client. Therefore the majority of client logic (e.g. adaptive behaviour) can be reused in future implementations. Moreover, the DASH client can be helpful in this early stage of interoperability testing with DASH server implementations.

The second approach depicts the deployment of the DASH proxy server in existing CDNs. This approach assumes that legacy HTML5 clients will exist (e.g. no further maintenance). These legacy clients can be supported from a single DASH source with the help of the proxy. This means that the content needs to be encoded only once to serve all devices.

References

[1] *Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats*, ISO/IEC 23009-1:2012.

[2] Spotify Ltd. website [Online]. Available: http://www.spotify.com

[3] Netlfix Inc. website [Online]. Available: http://www.netflix.com

[4] "Global Internet Phenomena Report", Sandvine, Spring 2011.

[5] *HbbTV Specification*, Version 1.5, March 2012 [Online]. Available: http://www.hbbtv.org/pages/about_hbbtv/HbbTV-specification-1-5.pdf

[6] "Forecast and Methodology, 2010-2015", Cisco Visual Networking Index, June, 2011.

[7] Thomas Stockhammer, "MPEG's Dynamic Adaptive Streaming over HTTP (DASH) - Enabling Formats for Video Streaming over the Open Internet," *Qualcomm Incorporated Webinar at EBU*. November 22, 2011.

[8] Adobe Systems Inc. Flash Media Streaming website [Online]. Available: http://www.adobe.com/de/products/flashmediastreaming/

[9] Microsoft Corp. Silverlight website [Online]. Available: http://www.microsoft.com/silverlight/

[10] W3C HTML5 Specification – Working Draft (March 29, 2012). The video element [Online]. Available: http://www.w3.org/TR/html5/the-video-element.html

[11] Apple Inc. HTTP Live Streaming website [Online]. Available: https://developer.apple.com/resources/http-streaming/

[12] Microsoft Corp. Smooth Streaming website [Online]. Available: http://www.microsoft.com/silverlight/smoothstreaming/

[13] Adobe Systems Inc. HTTP Dynamic Streaming website [Online]. Available: http://www.adobe.com/products/hds-dynamic-streaming.html

[14] *Transparent end-to-end Packet-switched Streaming Service; Protocols and codecs*, ETSI TS 126 234 V9.3.0, Jun, 2010

[15] GPAC Osmo4 Player website [Online]. Available: http://gpac.wp.mines-telecom.fr/player/

[16] Christopher Müller and Christian Timmerer, "A VLC Media Player Plugin enabling Dynamic Adaptive Streaming over HTTP," *Proceedings of the ACM Multimedia 2011, Scottsdale, Arizona*. November 28, 2011.

[17] Web Hypertext Application Technology Working Group (WHATWG). (2012, May). Adaptive Streaming. Available: http://wiki.whatwg.org/index.php?title=Adaptive_Streaming&oldid=6385

[18] Duncan Rowden, W3C MPTF. (2011, December). Adaptive Bitrate calls for HTML5 <video> tag [Online]. Available: http://www.w3.org/2011/webtv/wiki/index.php?title=MPTF/HTML_adaptive_calls&oldid=1690

[19] R. Pantos and W. May. (2011, September). "HTTP Live Streaming". Apple Inc. Informational Internet-Draft [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-07

[20] Aaron Colwell, Kilroy Hughes, Mark Watson. (2012, May). Media Source Extensions v.0.5, Draft Proposal [Online]. Available: http://html5-mediasource-api.googlecode.com/svn/trunk/draft-spec/mediasource-draft-spec.html

[21] *ISO base media file format*, ISO/IEC 14496-12:2008 Part 12.

[22] The Chromium Projects, Google Chrome 20.0.1130.1 dev-m.

Enrichment of News Show Videos with Multimodal Semi-Automatic Analysis

Daniel Stein[1], Evlampios Apostolidis[2], Vasileios Mezaris[2], Nicolas de Abreu Pereira[3], Jennifer Müller[3], Mathilde Sahuguet[4], Benoit Huet[4], and Ivo Lašek[5]

Fraunhofer Institute IAIS, Sankt Augustin, Germany[1] Information Technologies Institute CERTH, Thermi-Thessaloniki, Greece[2] Rundfunk Berlin-Brandenburg, Potsdam, Germany[3] Eurecom, Sophia Antipolis, France[4] Czech Technical University in Prague, and University of Economics, Prague, Czech Republic[5]

Abstract: **Enriching linear videos by offering continuative and related information via, e.g., audiostreams, webpages, as well as other videos, is typically hampered by its demand for massive editorial work. While there exist several (semi-)automatic methods that analyse audio/video content, one needs to decide which method offers appropriate information for an intended use-case scenario. In this paper, we present the news show scenario as defined within the LinkedTV project, and derive its necessities based on expected user archetypes. We then proceed to review the technology options for video analysis that we have access to, and describe which training material we opted for to feed our algorithms. Finally, we offer preliminary quality feedback results and give an outlook on the next steps within the project.**

Keywords: Speaker Recognition, Video Segmentation, Concept Detection, News show scenario, LinkedTV

1 Introduction

Many recent surveys show an ever growing increase in average video consumption, but also a general trend to simultaneous usage of internet and TV: for example, cross-media usage of at least once per month has risen to more than 59% among Americans [5]. A newer study [14] even reports that 86% of mobile internet users utilize their mobile device while watching TV. This trend results in considerable interest in interactive and enriched video experience, which is typically hampered by its demand for massive editorial work.

A huge variety of techniques, both new and established, exists that analyse video content (semi-)automatically. Ideally, the processed material will offer a rich and pervasive source of information to be used for automatic and semi-automatic interlinking purposes. However, the information produced by video analysis techniques is as heterogeneous as is their individual approach and the expected complexity, which is why careful planning is crucial, based on the demands of an actual use-case scenario. This paper introduces a news broadcast scenario for video enrichment as envisioned in the EU-funded project "Television linked to the Web" (LinkedTV),[1] and analyses the needs for practical usage. These needs form the basis for the technology that we employ for video analysis, and for the databases that we use to feed the training algorithms. We present our decisions on automatic speech recognition, keyword extraction, shot/scene segmentation, concept detection, the detection and tracking of moving and static objects, as well as unsupervised face clustering. Finally, we offer preliminary results based on human feedback.

This paper is structured as follows: we present the envisioned news show scenario of LinkedTV (Section 2), describe the analysis techniques and training material that are currently used (Section 3), provide manual examination of first experimental results (Section 4), and finally elaborate on future directions that will be pursued (Section 5).

1.1 Related Work

We will continue to review three recent projects with a related overall focus as LinkedTV. For the more detailed analysis methods as listed in Section 3, we will give appropriate citations along with their descriptions later in the paper.

inEvent The Project "Accessing Dynamic Networked Multimedia Events" (inEvent)[2] works on video material analysis and search-ability, using A/V processing techniques enriched with semantics, and recommendations based on social network information. The project's main targets are meetings, video-conferences and lectures, which are more restricted in a sense that they only include a limited set of persons and domains within one video.

TOSCA-MP "Task-oriented Search and Content Annotation for Media Production" (TOSCA-MP),[3] aims at content annotation and search tools, with its main target being professionals in the networked media production as well as archives, i.e., it is not directly aimed for non-professional end-users. The media scope is broader than LinkedTV, since TOSCA-MP also allows for radio pod-casts and written text from websites as seed content.

AXES The scope of the project "Access to Audiovisual Archives" (AXES)[4] is even broader, looking for possible linking information in scripts, audio tracks, wikis or blogs. A main focus is cross-modal detection of various entities such as people or places, i.e., drawing knowledge from several sources at once to improve the accuracy. The aimed content is audiovisual digital libraries rather than television broadcast.

2 News Broadcast Scenario

The audio quality and the visual presentation within videos found in the web, as well as their domains, are very heterogeneous. To cover many aspects of automatic video analysis, we have identified several possible scenarios for interlinkable videos within the LinkedTV project, e.g., (a) news show, (b) cultural heritage, and (c) visual arts [12]. In this paper, we focus on the specific demands of the news show scenario. The scenario uses German news broadcast as seed videos, provided by Public Service Broad-

[1] www.linkedtv.eu

[2] www.inevent-project.eu

[3] www.tosca-mp.eu

[4] www.axes-project.eu

caster Rundfunk Berlin-Brandenburg (RBB).[5] The main news show is broadcast several times each day, with a focus on local news for Berlin and Brandenburg area. For legal and quality reasons, the scenario is subject to many restrictions as it only allows for editorially controlled, high quality linking. For the same quality reason only links selected from a restricted whitelist are allowed. This whitelist contains, for example, videos produced by the Consortium of public service broadcasting institutions of the Federal Republic of Germany (ARD) and a limited number of approved third party providers.

The audio quality of the seed content can generally be considered to be clean, with little use of jingles or background music. Interviews of the local population may have a minor to thick accent, while the eight different moderators have a very clear and trained pronounciation. The main challenge for visual analysis is the multitude of possible topics in news shows. Technically, the individual elements will be rather clear: contextual segments (shots or stories) are usually separated by visual inserts and the appearance of the anchorperson, and there are only few fast camera movements.

2.1 Scenario Archetypes

LinkedTV envisions a service that offers enriched videos which are interlinked with the web, and targets a broader audience. For the sake of a convincing scenario, however, we have sketched three archetypal users of the LinkedTV news service and their motivations to use it:

Ralph comes home from working on a building site in Potsdam, and starts watching “rbb AKTUELL”. The first spots are mainly about politics and about Berlin. Ralph is not particularly interested, neither in politics nor in Berlin as he lives in a small town in Brandenburg. After a while there is the first really interesting news for Ralph: a spot about the restoration of a church at a nearby lake; as a carpenter, Ralph is always interested in the restoration of old buildings. Therefore, he watches the main news spot carefully and views an extra video and several still images about the church before and after its restoration. Finally, the service also offers links to a map and the website of the church which was set up to document the restoration for donators and anyone else who would be interested. Ralph saves these links to his smartphone so he can visit the place on the weekend.

Nina’s baby has fallen asleep after feeding, so she finds some time for casually watching TV, to be informed while doing some housework. Browsing the programme she sees that yesterday’s enhanced “rbb AKTUELL” evening edition is available and starts the programme. Nina watches the intro with the headlines while starting her housework session with ironing some shirts. Watching a news spot about Berlin’s Green Party leader who withdrew from his office yesterday, Nina is kind of frustrated as she voted for him and feels her vote is now “used” by someone she might not have voted for. She would like to hear what other politicians and people who voted for him think about his decision to resign. She watches a selection of video statements of politicians and voters and bookmarks a link to an online dossier about the man and his political carrier which she can browse later on her tablet. Eventually, the baby awakes so Nina pauses the application so she can continue later.

Socially active retiree **Peter** watches the same news show with different personal interest and preferences. One of the spots is about a fire at famous Café Keese in Berlin. Peter is shocked. He used to go there every once in a while, but that was years ago. As he hasn’t been there for years, he wonders how the place may have changed over this time. In the news spot, smoke and fire engines was almost all one could see, so he watches some older videos about the history of the famous location where men would call women on their table phones – hard to believe nowadays, he thinks, now that everyone carries around mobile phones! After checking the clips on the LinkedTV service, he returns to the main news show and watches the next spot on a new Internet portal about rehabilitation centres in Berlin and Brandenburg. He knows an increasing number of people who needed such facilities. He follows a link to a map of Brandenburg showing the locations of these centres and bookmarks the linked portal website to check some more information later. At the end of the show, he takes an interested look at the weather forecast, hoping that tomorrow would be as nice as today so he could go out again to bask in the sun.

3 Technical Background

Now that we have defined the motivation and needs of the archetypes above, we need access to a very heterogeneous set of information to be (semi-)automatically derived from the A/V content. We will proceed to list the technology employed to address these requirements. This section is divided into four sub-parts, being automatic speech recognition, temporal segmentation, spatiotemporal segmentation, person recognition, and other meta-information.

All this material is joined in a single xml file for each video, and can be visualized and edited by the annotation tool EXMARaLDA [9]. Note that for the time being, EXMARaLDA does not support spatial annotation; we plan to extend the tool at a later stage of the project. Also note that the EXMARaLDA format is only used internally and has been chosen because of its simplicity rather than a broad, standardized international usage (like in, e.g., MPEG-7).

3.1 Automatic Speech Recognition

Whenever there are no subtitles available, an automatic speech recognizer is needed in order to employ keyword extraction as well as named entity recognition. If subtitles are given, forced alignment techniques to match the timestamps to the video on a word level could be useful, because the utterance timestamp provided by the subtitles might be to imprecise and coarse-granular for our needs. In both cases, we need a strong German acoustic model. We employ a state-of-the-art speech recognition system as described in [10]. For training of the acoustic model, we employ 82,799 sentences from transcribed video files. In accordance with the news show scenario, they are taken from the domain of both broadcast news and political talk shows. The audio is sampled at 16 kHz and can be considered to be of clean quality. Parts of the talk shows are omitted when, e.g., many speakers talk simultaneously or when music is played in the background. The language model consists of the transcriptions of these audio files, plus additional in-domain data taken from online newspapers and RSS feeds. In total, the material consists of 11,670,856 sentences and 187,042,225 running words. Of these, the individual subtopics were used to train trigrams with modi-

[5] www.rbb-online.de

fied Kneser-Ney discounting, and then interpolated and optimized for perplexity on a with-held 1% proportion of the corpus.

For Dutch, as foreseen in later parts of the project, the SHOUT speech recognition toolkit, as described in [6] will be used.

3.2 Temporal Segmentation

Larger videos should be temporally segmented based on their content. In our scenario, this would enable to skip parts of the video and directly jump to the weather forecast, for example. Also, we need to provide reasonable timestamp limits for hyperlinks, since we do not want further information on Berlin's Green Party to still be active once the clip about the rehabilitation center starts.

Currently, we segment the video into *shots* (i.e., fine-granular temporal segments which correspond to a sequence of consecutive frames captured without interruption by a single camera) and *scenes* (higher-level temporal segments composed by groups of shots, covering either a single event or several related events taking place in parallel).

Video shot segmentation is based on an approach proposed in [13]. The employed technique can detect both abrupt and gradual transitions between consecutive shots. The detection of gradual transitions is beneficial in cases where video production effects, such as fade in/out, dissolve etc., are used for the transition between successive shots of the video, which is a common approach, e.g., at the production of documentary videos. However, in certain use cases, like e.g., in news show videos where transition effects are rarely used between shots, it may be advantageous for minimizing both computational complexity and the rate of false positives to consider only the detected abrupt transitions. Specifically, this technique exploits image features such as color coherence, Macbeth color histogram and luminance center of gravity, in order to form an appropriate feature vector for each frame. Then, given a pair of selected successive or non-successive frames, the distances between their feature vectors are computed, forming distance vectors, which are then evaluated with the help of one or more SVM classifiers. In order to further improve the results, we augmented the above technique with a baseline approach to flash detection. Using the latter we minimize the number of incorrectly detected shot boundaries due to camera flash effects.

Video scene segmentation groups shot segments into sets which correspond to individual events of the video. The employed method was proposed in [11]. It introduces two extensions of the Scene Transition Graph (STG) algorithm [15]; the first one aims at reducing the computational cost of shot grouping by considering shot linking transitivity and the fact that scenes are by definition convex sets of shots, while the second one builds on the former to construct a probabilistic framework towards combination of multiple STGs. The latter allows for combining STGs built by examining different forms of information extracted from the video (i.e., low-level audio or visual features, and high-level visual concepts or audio events), while at the same time alleviating the need for manual STG parameter selection.

3.3 Spatiotemporal Segmentation

Objects of interest should be detected and tracked so that they can be clicked on for further information. Due to the broad target domains it cannot be guaranteed that established databases contain enough instances for local entities, which is why we need strong clustering and re-detection techniques so that an editor only needs to label them once and can automatically find other instances within the video itself, or within a larger set of surrounding videos.

For the purpose of spatiotemporal segmentation of the video stream, we differentiate between *moving* and *static* object detection. Spatiotemporal segmentation of a video shot into differently moving objects is performed as in [2]. This unsupervised method uses motion and color information directly extracted from the MPEG-2 compressed stream. The bilinear motion model is used to model the motion of the camera (equivalently, the perceived motion of static background) and, wherever necessary, the motion of the identified moving objects. Then, an iterative rejection scheme and temporal consistency constraints are employed for detecting differently moving objects, accounting for the fact that motion vectors extracted from the compressed stream may not accurately represent the true object motion.

For detecting occurrences of static objects of interest in consecutive or non-consecutive video frames, like the church or the café in the scenario, a semi-automatic approach based on object re-detection will be adopted. The human editor will manually specify the object of interest by marking a bounding box on one frame of the video. Then, the additional instances of the same object in subsequent frames will be automatically detected via object matching. Matching between image regions will be performed based on SURF descriptors [1] and some geometric restrictions defined by the RANSAC algorithm [2]. A baseline OpenCV implementation will initially be adopted for this. For each pair of images, feature vectors will be extracted using the SURF algorithm and will be compared. False matches will be filtered out using a symmetrical matching scheme between the pair of images, and the remaining outliers will be discarder by applying some geometric constraints calculated from the RANSAC method.

3.4 Person Detection

Persons seen or heard within a video are arguably the most important information for a viewer. They should be identified via their face and/or their voice, i.e., one can rely both on face detection as well as speaker recognition. For local content where some of the persons might be unknown with regard to the training material, there is also demand for unsupervised person clustering similar to objects of interest above, so that, e.g., the former leader of Berlin's Green Party needs only to be labelled once by a human editor.

3.4.1 Face Analysis

Face analysis is performed on keyframes extracted from the video (i.e., by temporal subsampling), and employs the face.com API.[6] The process can be divided into three components: face detection, face clustering and face recognition. Face detection is used as a prior tool to perform the other tasks, which both stem from the calculation of the similarity between detected faces. Our implementation is based on an iterative training/recognition approach [3] to make accurate groupings: the training process is initialized with the first detected face in the video. For each subsequent picture, the detected faces are matched against the

[6] `http://developers.face.com`

initial face. If the recognition confidence level is higher than a threshold (80% performed well in our experiments), both faces are associated with the same id, otherwise a new face id is created. After every assignment, the corresponding face model is retrained before performing recognition on the next candidate face. We output our results in a xml file that provides the coordinate of the faces detected in each picture (center, length and width of the bounded-box), together with the id of the face (id of the cluster). In order to guaranty the highest accuracy possible, we rely on a human annotator to label the face clusters with the appropriate person name.

3.4.2 Speaker Recognition

For speaker identification (SID), we follow the approach of [7], i.e., we make use of Gaussian Mixture Models (GMMs) using spectral energies over mel-filters, cepstral coefficients and delta cepstra of range 2. An overall universal background model (UBM) is merged from gender-dependent UBMs and forms the basis for the adaptation of person-dependent SID models. To train and assess the quality of the speaker identification, we listed German politicians as a possible requirement within our scenario description in Section 2.1. Thus, we downloaded a collection of speeches from 253 German politicians, taken from the archive of the German parliament.[7] In total, this consists of 2581 files with 324 hours of training material. To make training of the models feasible, we use 2 minutes per file to adapt the UBM.

3.5 Meta-Information

For keywords needed to tag the videos, they can either be extracted from textual sources, or derived from video-based concept detectors. These tags can then be used to recommend similar videos like, e.g., of churches in the local area.

3.5.1 Concept Detection

A baseline concept detection approach is adopted from [4]. Initially, 64-dimension SURF descriptors are extracted from video keyframes by performing dense sampling. These descriptors are then used by a Random Forest implementation in order to construct a Bag-of-Words representation (including 1024 elements) for each one of the extracted keyframes. Following the representation of keyframes by histograms of words, a set of linear SVMs is used for training and classification purposes, and the responses of the SVM classifiers for different keyframes of the same shot are appropriately combined. The final output of the classification for a shot is a value in the range [0, 1], which denotes the Degree of Confidence (DoC) with which the shot is related to the corresponding concept. Based on these classifier responses, a shot is described by a 323-element model vector, whose elements correspond to the detection results for the 323 concepts defined in the TRECVID 2011 SIN task.[8] These 323 concepts were selected among the 346 concepts originally defined in TRECVID 2011, after discarding a few that are either too generic (e.g., "Eukaryotic Organism") or irrelevant to the current data being considered in LinkedTV (cf. [12]).

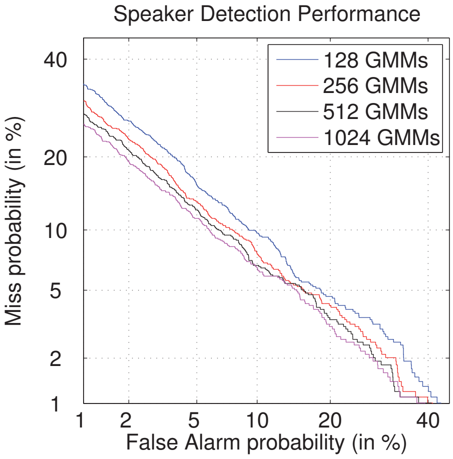


Figure 1: Speaker identifaction for German politians: DET Curve for different mixture sizes of the GMM, on a withheld test corpus of 994 audio files from the German parliament.

Moreover, in order to improve the detection accuracy we used some relations between concepts as they were determined in TRECVID 2011. When a concept implies another concept (e.g., the concept "Man" implies the concept "Person") then the confidence level of the second concept is reinforced with the help of an empirically set factor α. On the contrary, when a concept excludes another concept (e.g., the concept "Daytime Outdoor" excludes the concept "Nightime") and if the confidence score of the first concept is higher than the second, the first one is enhanced and the second one is penalized accordingly.

3.5.2 Keyword Extraction

The objective of keyword extraction or glossary extraction is to identify and organize words and phrases from documents into sets of glossary-items or keywords. For this particular scenario, we have access to several sources of textual information about a particular video. These include subtitles, manual annotations of videos, and finally the transcripts obtained from the ASR. Since LinkedTV is a multilingual project, we decided to refrain from using linguistically based, i.e., language dependent techniques here but employ a statistical approach based on text frequency – inverse document frequency (TF-IDF) [8] weights of words extracted from videos.

4 Experiments

In this section, we present the result of the manual evaluation of a first analysis of "rbb AKTUELL" videos.

The ASR system produces reasonable results for the news anchorman and for reports with predefined text. In interview situations, the performance drops significantly. Further problems include named entities of local interest, and heavy distortion when locals speak with a thick dialect. We manually analysed 4 sessions of 10:24 minutes total (1162 words). The percentage of erroneous words were at 9% and 11% for the anchorman and voice-over parts, respectively. In the interview phase, the error score rose to 33%, and even worse to 66% for persons with a local dialect.

To evaluate the quality of the SID, a distinct set of 994 audio files has been used to evaluate the quality of the mod-

[7] http://webtv.bundestag.de/iptv/player/macros/bttv/index.html

[8] www-nlpir.nist.gov/projects/tv2011/

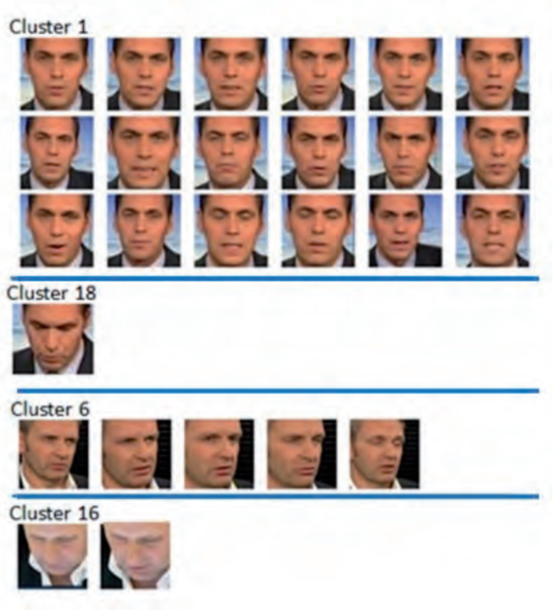


Figure 2: Clusters 1, 6, 16 and 18 of the face clustering result. Clusters 1 and 18 contain the anchorman and no noise face: two groups are made depending on the angle of his head. We have the same effect for clusters 6 and 16.

els, containing 253 different speakers. A GMM with 128 mixtures has an Equal Error Rate (EER) of 9.86, whereas using 1024 mixtures improves the EER to 8.06. See Figure 1 for Detection Error Trade-Off (DET) curves.

We performed face clustering on 200 keyframes extracted from the seed video at regular intervals. Results give a total of 54 clusters, most of which containing a single face (people that appear once). All 54 clusters are pure (i.e., contain only 1 person's face), and 2 ids (persons) appear in more than 1 cluster (as seen in Figure 2) due to significant viewing angle difference. Such clusters can easily be processed by an annotator. To make this process easier, we consider to let aside the clusters that contain only one face image, on the assumption that a person that appears only once is not of primary importance for the video.

In preliminary experiments on shot segmentation, the algorithm performed remarkably well. The effect from reporters' flashlights has been significantly restricted and the detection accuracy based on human defined ground-truth data was over 90%. A small number of false positives and false negatives was caused due to rapid camera zooming operations and shaky or fast camera movements. In a second iteration with conservative segmentation, most of these issues could be addressed.

Indicative results of spatiotemporal segmentation on these videos, following their temporal decomposition to shots, are shown in Figure 3. In this figure, the red rectangles demarcate automatically detected moving objects, which are typically central to the meaning of the corresponding video shot and could potentially be used for linking to other video, or multimedia in general, resources. Currently, the algorithm detects properly over 80% of the presented moving objects. However, an unwanted effect of the automatic processing is the false recognition of name banners which slide in quite frequently during interviews, which indeed is a moving object but does not yield additional information.

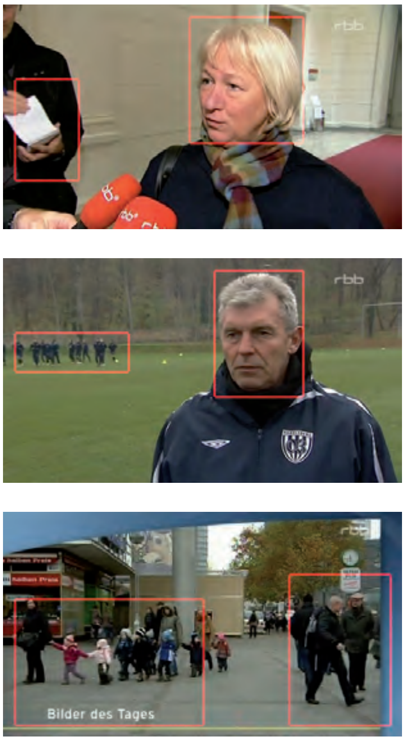


Figure 3: Spatiotemporal Segmentation on video samples from news show "RBB Aktuell"

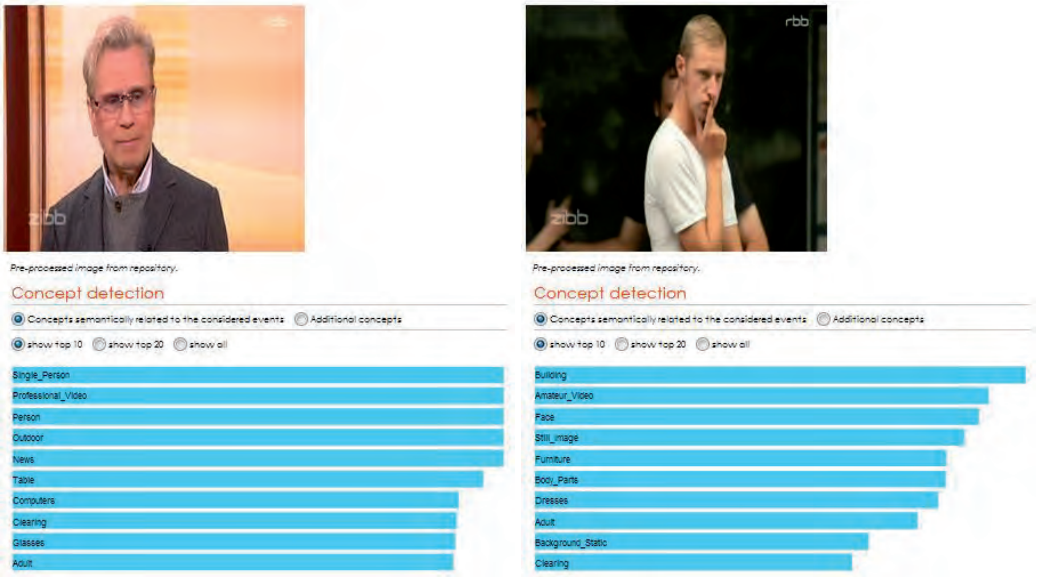
Manually evaluating the top-10 most relevant concepts according to the classifiers' degrees of confidence revealed that the concept detectors often succeed in providing useful results; yet there is significant room for improvement. See Figure 4 for two examples, the left one with good results, and the right one with more problematic main concepts detected.

5 Conclusion

In this paper, we presented the news show scenario as pursued by the LinkedTV consortium. We have access to state-of-the-art techniques that can analyse the video content in order to derive the needed information, and reported reasonable preliminary results.

The main challenge will be to interweave the single results into refined high-level information. For example, the person detection can gain information from automatic speech recognition, speaker recognition and face recognition. Also, in order to find reasonable story segments in a larger video, one can draw knowledge both from speech segments, topic classification, and video shot segments. As a final example, video similarity can be estimated with feature vectors carrying information from the concept detection, the keywords, the topic classification and the entities detected within the video.

We already collected all (semi-)automatically produced information bits into a single annotation file that can be viewed with a proper tool, and will use this in order to establish ground truth material on the scenario data in a next step. Then, we plan to validate and extend the methods to increase their accuracy. Finally, we hope to gain more insight from the multimodal feature combination, which includes combining different confidence scores from across



Pre-processed image from repository.

Concept detection

● Concepts semantically related to the considered events ○ Additional concepts

● show top 10 ○ show top 20 ○ show all

Single_Person
Professional_Video
Person
Outdoor
News
Table
Computers
Clearing
Glasses
Adult

Pre-processed image from repository.

Concept detection

● Concepts semantically related to the considered events ○ Additional concepts

● show top 10 ○ show top 20 ○ show all

Building
Amateur_Video
Face
Still_image
Furniture
Body_Parts
Dresses
Adult
Background_Static
Clearing

Figure 4: Top 10 TREC-Vid concepts detected for two example screenshots. Wrong concepts in the left one are "outdoor","table","computers", and "clearing", whereas the rest is correct. In the right one, only three concepts "face", "body part" and "adult" are correct.

the analysis results to emphasize or reject certain hypotheses, but also will introduce high-level features that can offer new interlinking enhancements for the viewers' experience.

Acknowledgements

This work has been funded by the European Community's Seventh Framework Programme (FP7-ICT) under grant agreement n° 287911 LinkedTV.

References

[1] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.

[2] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

[3] Liu, X. and Huet, B. (2010). Concept detector refinement using social videos. In *VLS-MCMR — International workshop on Very-large-scale multimedia corpus, mining and retrieval, October 29, 2010, Firenze, Italy*, Firenze, ITALY.

[4] Moumtzidou, A., Sidiropoulos, P., Vrochidis, S., Gkalelis, N., Nikolopoulos, S., Mezaris, V., Kompatsiaris, I., and Patras, I. (2011). ITI-CERTH participation to TRECVID 2011. In *TRECVID 2011 Workshop*, Gaithersburg, MD, USA.

[5] Nielsen (2009). Three screen report. Technical report, Nielsen Company.

[6] Ordelman, R. J. F., Heeren, W. F. L., de Jong, F. M. G., Huijbregts, M. A. H., and Hiemstra, D. (2009). Towards Affordable Disclosure of Spoken Heritage Archives. *Journal of Digital Information*, 10(6).

[7] Reynolds, D., Quatieri, T., and Dunn, R. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10:19–41.

[8] Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 232–241, New York, NY, USA. Springer-Verlag New York, Inc.

[9] Schmidt, T. and Wörner, K. (2009). EXMARaLDA – Creating, analysing and sharing spoken language corpora for pragmatic research. *Pragmatics*, 19:4:565–582.

[10] Schneider, D., Schon, J., and Eickeler, S. (2008). Towards Large Scale Vocabulary Independent Spoken Term Detection: Advances in the Fraunhofer IAIS Audiomining System. In *Proc. SIGIR*, Singapore.

[11] Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., and Trancoso, I. (2011). Temporal video segmentation to scenes using high-level audiovisual features. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(8):1163 –1177.

[12] Stein, D., Apostolidis, E., Mezaris, V., de Abreu Pereira, N., and Müller, J. (2012). Semi-automatic video analysis for linking television to the web. In *Proc. FutureTV Workshop*, pages 1–8, Berlin, Germany.

[13] Tsamoura, E., Mezaris, V., and Kompatsiaris, I. (2008). Gradual transition detection using color coherence and other criteria in a video shot meta-segmentation framework. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 45 –48.

[14] Yahoo! and Nielsen (2010). Mobile shopping framework – the role of mobile devices in the shopping process. Technical report, Yahoo! and The Nielsen Company. `l.yimg.com/a/i/us/ayc/article/mobile_shopping_framework_white_paper.pdf`.

[15] Yeung, M., Yeo, B.-L., and Liu, B. (1998). Segmentation of video by clustering and graph analysis. *Comput. Vis. Image Underst.*, 71(1):94–109.

The Creation of a Perceptive Audio Drama

Anthony Churnside[1], Ian Forrester[2]

[1 and 2]BBC R&D, Salford, UK

E-mail: [1]Anthony.Churnside@bbc.co.uk, [2]Ian.Forrester@bbc.co.uk

Abstract: **This paper describes the design and creation of an object-based audio drama, entitled 'Breaking Out'. The audio drama is designed to provide a narrative that varies, depending on the listener's geographic location and the date/time the listener is experiencing the drama. This paper considers the impact of object based production and personalisation on the creative process (the story design and writing), and the recording and production workflows.**

Keywords: Audio, Objects, Personalisation, Client-Side, New-Media Experiences.

1. INTRODUCTION

Radio drama workflows have not changed significantly since techniques were developed 75 years ago [1]. Linear, static narratives are created and scripted. These scripts are rehearsed and performed by actors. These performances are recorded and mixed with music and sound effects to create a single static stereo recording. This recording is then transmitted to the audience and each individual audience member experiences the same linear narrative, at the same time. The most significant change in radio drama production was perhaps the move from analogue audio tapes to digital audio workstations [2]. While this change meant the roles in radio drama production altered slightly and multichannel post-production was made more easy, the general approach to the recording process remained largely unchanged. There has also been some work in surround sound, although surround sound radio drama production is still very niche.

2. PERCEPTIVE MEDIA

Perceptive media is a term coined by the authors to describe a piece of content that automatically undergoes minor adaptations in response to information that it perceives about the individual viewers or listeners [3]. A similar concept to context aware computing, Perceptive Media adapts stories and narratives to fit the audiences [4]. The aims of the the perceptive media project are to explore two things:

- how might a producer approach the creation of such a piece of content?

- how will the audience react to a piece of content that adapts for them?

This paper focuses on the first of these questions.

1. Audio Drama as a Set of Data

To allow these minor adaptations a new way of thinking about audio programmes is required. Rather than considering an audio drama as a final audio file, it must be thought of as a data set representing all of the programme assets and metadata. This approach as been proposed in the context of MPEG-4 [5], but there is not yet widespread use of MPEG-4 for object audio/visual coding. Taking this approach different versions of a programme can be assembled from a common set of assets, allowing different adaptations. For the remainder of this paper each of these audio assets (audio file) is considered an audio object.

2. Adaptation

The first stage in the creation of the experimental audio drama "Breaking Out", was to identify what information the drama would perceive and how the drama would react to the perceived information. Some examples of the type of data it is possible to detect are shown in table 1. For this initial investigation, only data easily collected from the listener was used. The listener was not required to manually provide any information (such as their name, social network login details etc.). It was decided that the drama would be delivered by IP and would be a browser based experience. There is to potential for a hybrid (Broadcast/IP) solution, it was felt a simpler prototype using IP would be more appropriate at this early stage in the investigation. There are limited data that can be determined by an internet browser. It was decided the drama adaptations would depend on a single piece of information; the user's location.

To allow realtime adaptation of parts of the script a browser based (JavaScript) text-to-speech engine [6] would be employed to speak variables collected as results various online resources.

3. Scalability

To allow the a large number of listeners to hear the drama, the audio playback and processing (level control, fading in and out, adding reverberation, etc.) had to take place at the client-side, in the listener's web-browser. The traditional approach to broadcasting was for the broadcaster to perform the computationally expensive processes, allow the client processes to be less complex and therefore cheaper. However, modern computers have the processing power to enable this rendering and recent HTML5 standards [7] allow complex processing of audio. Web audio standards are currently in development and at this early stage, only the Chrome browser fully supports this implementation of the audio drama.

An alternate approach would be a server-side audio render, but this could risk high contention if a large number of users simultaneously required audio renders.

Corresponding author: Anthony Churnside, BBC R&D, MediaCityUK, Salford, Anthony.Churnside@bbc.co.uk

3. PRODUCTION WORKFLOW

The processes followed to create and deliver this audio drama differed to that of a traditional radio drama. Figure 3 shows a workflow typical of a traditional radio drama, and that used for this drama production respectively.

With traditional radio workflows the programme is fully designed and produced and a single 'final mix' is created before it is distributed to the audience. Using an object based production workflow, a single final mix never exists and the final mix experienced by the listener is assembled just prior to the point it is received.

The fact that they don't have full control of the final mix may be daunting to producers more familiar with traditional workflows. However, it is possible to argue that this belief of control is misplaced, given the variety of devices and listening environments used by audiences to consume audio content.

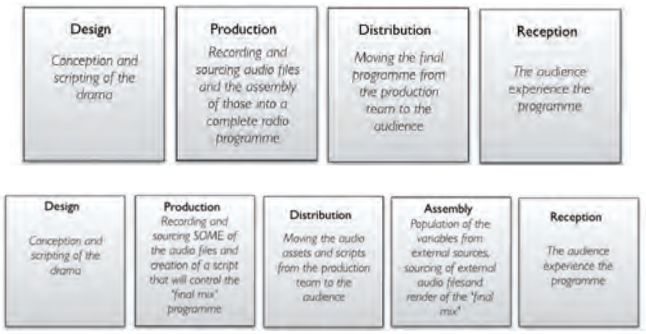

Design
Conception and scripting of the drama
Production
Recording and sourcing audio files and the assembly of those into a complete radio programme
Distribution
Moving the final programme from the production team to the audience
Reception
The audience experience the programme
Design
Conception and scripting of the drama
Production
Recording and sourcing SOME of the audio files and creation of a script that will control the 'final mix' programme
Distribution
Moving the audio assets and scripts from the production team to the audience
Assembly
Population of the variables from external sources, sourcing of external audio filesand render of the 'final mix'
Reception
The audience experience the programme

Figure 3: Radio drama workflow comparison

4. DESIGN PROCESS

In traditional radio drama workflows a scriptwriter is given a brief and delivers a first draft to the drama producer. The producer then provides feedback to the writer. This is an iterative process which concludes with a final version of the radio drama script. This project used a similar process, but the writer (who would normally deliver the final script) had to work with the producers to develop a creative idea that used the location detecting technology for the benefit of the story.

1. Constraining the writer

In order to allow the localisation of the story constraints were placed on the writer. Although intelligible, the browser based JavaScript text-to-speech engine did not sound particularly natural, therefore it was stipulated that one of the characters should be a computer or artificial intelligence.

2. Identifying variables

Due to the nature of perceptive media a final version of the script never existed. Each time the radio drama is listened to it is different. This leads to a potentially infinite number of final versions. The need for these different versions meant the producer had to work closely with the writer to identify how the location data could enhance the story.

While the overall arc of the story remained the same regardless of listener location, variables were woven into the narrative to allow the localisation of the story.

Datum type	Method/Examples
Audio	Analysis of audio from attached microphones.
Video	Analysis of video from attached webcams.
Geo-location	Location can be determined and used to look up - time of day, language, local events/news, weather.
User-Agent	Can be used to identify the browser, operating system from which the device my be inferred.
Feature Detection	JavaScript can be used to detect features such as screen resolution.
Referrer string	The previously visited website can can determined form the referee string.
Downloaded files	Previously downloaded torrents can be determined for, the user's IP address.

Table 1: User data

Examples of the variables identified are listed below:

- town where the story is set.
- three well known places located nearby where the story is set.
- the weather at the time and place the listener is listening.
- films being shown at a cinema near to the listener's location.
- the date the listener is listening.
- the news on the date the listener is listening.

5. PRODUCTION PROCESS

The drama featured three characters; two actors and a computer. The drama also called for the sound effects of a lift, such as 'dings', doors opening and closing, lift movements and buttons being pressed.

1. Clean capture

The adaptation of the drama occurred at the point of listening, not the point of production. This meant the individual sounds of the actors speaking and the sound effects needed to be recorded separately and treated as separate audio objects for the whole of the production chain. Performances of each of the actors, the lift sound effects and the music were captured discretely in dry acoustic conditions (see figure 1). These remained as separate audio files and were played back with the correct timing, level and processing at the time of listening.

Corresponding author: Anthony Churnside, BBC R&D, MediaCityUK, Salford, Anthony.Churnside@bbc.co.uk

6. ASSEMBLY

Once all the individual audio files were captured the drama script was translated into a JavaScript which contained the following metadata for each audio object.

- a unique ID
- the location of the audio file (a URL)
- a textual description of the sound
- timing/synchronisation information
- processing information

Figure 1: Recording the actor's performance

1. Robot Voice

The scalability requirement meant that the text-to-speech was also performed in the client browser. Other more advanced and natural sounding text-to-speech algorithms are available [8] the browser based requirement for this audio drama meant less than natural sounding speech. This limitation is acceptable as it is believed that the technology will improve in future.

2. Variables

The variables identified in section 4.2 are populated using a number of online resources, such as the BBC's weather information [9], the BBC news podcast feed [10] and a Cinema release RSS feed. When text based variables are returned by these feeds the variables are voiced by the text-to-speech engine, in the case of audio files (for example the news podcast) the audio is played.

3. Sound Effects

The majority of the drama occurs in a small lift. In traditional radio drama production the sense of a small room would be created either by recording the actors inside a room with similar acoustics to the desired space or by processing the audio to make it sound like it occurred inside the desired space. The lift's speech generation is performed inside the browser so it is impossible to pre-process the sound. Client side processing had to be used to make the actor's and the lift's lines sound like they were spoken in the same acoustic space. Using HTML5 audio convolution, all the reverberation was applied by the web-browser.

4. Architecture

Figure 2 shows a simple system diagram indicating the sources of the different data used in the audio drama. There were a number of different types of audio that when together to create the audio drama, these are shown in table 2.

Audio type	Source	Type
Harriet's voice	Recorded by the production	mp3
Sound effects/ Music	Sourced from sound effects and music libraries	mp3
Room Impulse Responses	Captured by the Production	wav
Dynamic lift voice	Online RSS feeds	Text-to-speech
News report	Online podcast	mp3

Table 2: Audio Sources and Types

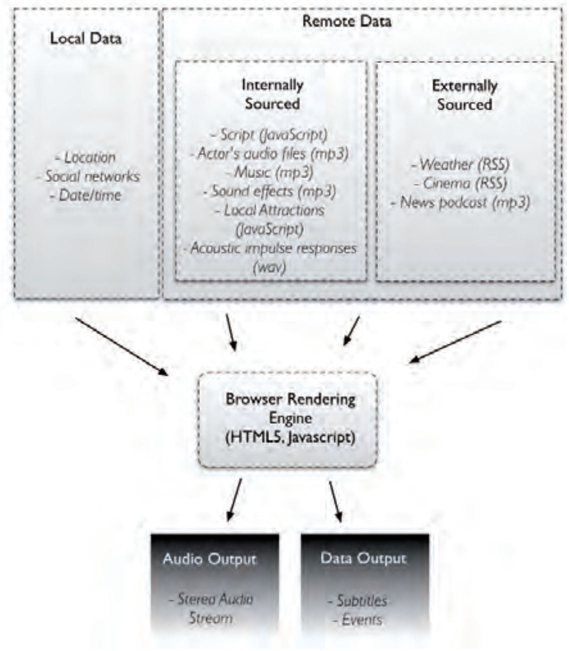


Figure 2: System diagram

7. POTENTIAL ADVANTAGES

An object based approach was taken in order to allow for the personalisation. The flexibility of this approach allows some other potential benefits outlined below.

1. Background level control

The listener has the ability to alter the relative level between the foreground and background sounds. This could be a benefit for listeners with hearing difficulties or English as a second language. A recent experiment with the BBC's coverage of Wimbledon has proven this technology to be useful to certain listeners [11].

2. Pace

The pace of playback can be determined by the listener. The change in pace is achieved by altering the time gap

Corresponding author: Anthony Churnside, BBC R&D, MediaCityUK, Salford, Anthony.Churnside@bbc.co.uk

between each of the lines rather than speeding up or slowing down the entire performance for listeners who prefer a slower or faster pace. This avoids some of the negative impacts on speech intelligibility and audio quality associated with audio time-stretching technologies. Figure 5 shows the existing control panel.

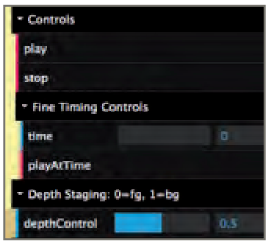


Figure 5: Background/foreground and pace controls (screenshot)

3. Subtitles

The audio playback is controlled by a JavaScript. This JavaScript is created directly from the original script and includes the lines delivered by the characters, textual descriptions of the sound effects, the effects/reverberation parameters and timing information.

This information can be easily displayed in synchronisation with the audio playback. Doing so can effectively provide the listener with subtitles for radio drama.

8. CONCLUSIONS AND FUTURE WORK

This paper assesses the impact of an object based approach to radio drama on existing workflows. A number of potential advantages to object based audio production are also identified.

This prototype has been created but there are plans to continue to work in this area.

This programme, 'Breaking Out', is going to be put on the web and the public will be invited to listen and provide feedback. In addition focus groups will be held that will explore advantages and disadvantages of the technology.

The potential advantages identified in section 7 will also be verified.

This paper focuses on using location data to adapt audio content. However, there are more data that could be used to adapt content (some of which are shown in table 1). The use of other sensors, such as web-cams and microphones could be explored in this context.

There are also plans to investigate how the concept of perceptive media could be applied to video content.

References

1. Swaguchi M, *Practical Surround Sound Production Part-1: Radio Drama*, AES, 2001.
2. McCarthy J, Stewart J, *Producing Radio Drama Using Networked Digital Audio Workstations*, AES, 1995.
3. Perceptive Media, http://thenextweb.com/media/2012/02/08/the-bbc-is-experimenting-with-perceptive-media-and-it-could-transform-tv-forever/
4. Context Aware Stories, http://connectedsocialmedia.com/6359/future-lab-context-aware/
5. MPEG-4 standard, http://mpeg.chiariglione.org/standards/mpeg-4/mpeg-4.htm
6. eSpeak, http://espeak.sourceforge.net/
7. W3C (HTML5 audio group), http://www.w3.org/
8. Lemmetty S, *Review of Speech Synthesis Technology*, Helsinki University, 1999.
9. Weather Source, http://open.live.bbc.co.uk/weather/feeds/en/
10. News podcast source, http://downloads.bbc.co.uk/podcasts/radio/newspod/rss.xml
11. NetMix experiment, http://www.bbc.co.uk/blogs/5live/2011/06/netmix.shtml

Acknowledgements

The authors would like to thank Mark Boas and Mark Panaghiston of Happy Worm, Henry R Swindell of BBC Writer's Room, Sharon Sephton, Sarah Glenister and Maeve Larkin.

Corresponding author: Anthony Churnside, BBC R&D, MediaCityUK, Salford, Anthony.Churnside@bbc.co.uk

SVM-based Shot Type Classification of Movie Content

Ioannis Tsingalis, Nicholas Vretos, Nikos Nikolaidis, Ioannis Pitas

Department of Informatics Aristotle University of Thessaloniki, Thessaloniki, Greece

E-mail: pitas@aiia.csd.auth.gr

Abstract: **In this paper, we propose a Support Vector Machine (SVM) based shot classification method for movies. This method classifies shots into seven different classes, namely eXtreme Long Shot (XLS), Long Shot (LS), Medium Long Shot (MLS), Medium Shot (MS), Medium Close Up (MCU), Close Up (CU) and eXtreme Close Up (XCU). The proposed method uses two features. The first one is the ratio of the height of the actor's facial image to the height of video frame. The second one is the ratio of the corresponding widths. These two ratios constitute the 2-D feature vectors which are fed into the SVM. A ground truth labeled shot database was created in order to experimentally test the proposed method performance. The corresponding results are very promising.**

Keywords: Shot type classification, Feature extraction, Support Vector Machines

1 INTRODUCTION

Due to the flourish of the movie industry during the last decades the automatic analysis, description, indexing and retrieval of video content became an urgent necessity. In these applications, shot type classification is undoubtedly one of the most useful techniques for analysing, characterizing and subsequently retrieving video. A shot is a continuous filming from a single camera for a certain period of time [5]. Shots constitute the main building blocks of film editing process. Some of the shot types used in cinematography can be found in [6], [3] and [5]. As far as research in shot type classification is concerned, a vast amount of work has been done, mostly in sports and news videos. For example, S.F.Chang et. al. [13] classify shots into CU, LS and MS using grass-ratio, and based on this classification they categorize sports video in play or break segments using a heuristic rule. Other shot classification methods for sports content that use a ratio on the apparent grass of the football field, the so called grass-ratio, can be found in [6] and [4]. Another interesting approach for shot type analysis in tennis videos is presented by X. Yu. et. al. [14] and is based on MPEG motion vectors and other features. Moreover, in [9] and [11] histogram color information is used for shot type classification. Unlike sports whose shot types are limited and more restricted, movie shots are more diverse over the various film types and genres. In [1] and [12], information from saliency maps, geometric composition of the scene, color and motion distribution are considered in order to classify shots. Other alternative approaches for determining the shot types work by estimating either the *absolute* or *relative* depth of the scene. The former technique is based on the actual distance between the filmed object and the camera, whereas the latter is based on the estimation of texture gradients [8], shape from shading, fractal dimensions [7] and other features.

The method in [3] is based on a combination of the height of the bounding box that frames the actor’s face and the distance between the bottom of the bounding box and the bottom of the frame. However, this algorithm does not perform very well since it is based on a simple thresholding of the derived value that involves the human body golden ratio. Figure 1 shows the basic features of the approach described in [3]. These two features are capable to represent the dominance of the actor on the frame and thus provide the shot type classification.

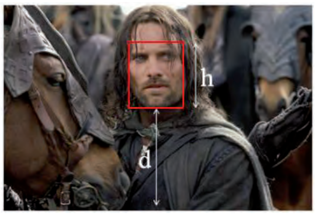


Figure 1: Low level features used in [3]

In this paper, a shot classification method for movies is presented. The major assumption in that method is that an actor is present in each shot and the bounding box of the actor’s face is available through the application of a face detection and/or tracking algorithm. The dimensions of the bounding boxes are used for the extraction of the features that are used in the proposed classification method. The proposed method is not restricted to a specific movie genre, like the method in [1] that can operate only in action movies, but covers different genres. The rest of the paper is organized as follows: in Section 2 the main shot types are introduced. In Section 3 the proposed method is presented. Experimental results are drawn in Section 4. Section 5, concludes the paper.

2 MOVIE SHOT TYPES

There are seven major types of video shots in movies, also known as field sizes [5].

eXtreme Close Up Shot (XCU): A part of the actor's face is visible. In this type the frame contains no information for the background.

Close Up Shot (CU): In this category the head of the actor is visible from the top of the actor's hair till the top of the shoulders. Sometimes it is called a “head shot”.

Medium Close Up Shot (MCU): The human body is framed from the elbow joint and above.

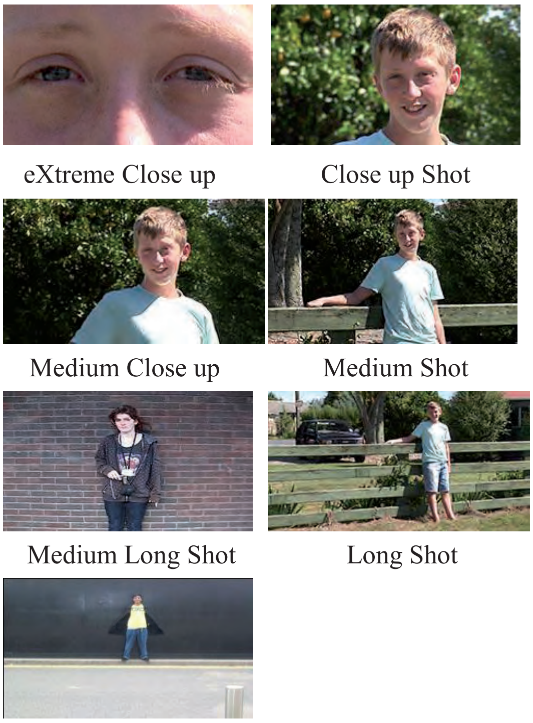
Medium Shot (MS): In this type the human is visible down to the waist level and hand gestures are visible.

Medium Long Shot (MLS): In this shot type, the actor's body is usually framed from the knees and up.

Long Shot/Wide Shot (LS/WS): In this case almost the entire body of the actor is visible and is usually considered as a full "body shot".

eXtreme Long Shot (XLS): It is usually used as an establishing shot and unlike previous types that can occur either in indoor or outdoor scenes, it usually occur in outdoor scenes. In this case the background is the most dominant element in the frame.

It should be noted that the previous shot types are not equiprobably in movies. For example, eXtreme Close Up and eXtreme Long Shot rarely appear. On the other hand, shots such as Close Up and Medium Close Up are used more frequently. Figure 2, shows the various type of shots.



eXtreme Close up

Close up Shot

Medium Close up

Medium Shot

Medium Long Shot

Long Shot

eXtreme Long Shot

Figure 2: Types of shots

3 SVM-BASED SHOT TYPE CLASSIFICATION

In the proposed method the height and the width of the facial bounding box are used, combined with the height and the width of the video frame. More specifically the extracted features are:

- Let the height of the face bounding box be h_{bb} and the height of the video frame H_F. The first feature involved in the proposed method is the ratio h_{bb}/H_F.
- The second feature is $w_{bb}/\ W_F$, where w_{bb} and W_F are the width of the bounding box and the video frame respectively.

Figure 3 shows the abovementioned features. These two features are the elements of the feature vectors. Figure 4, illustrates the value of these features for the ground truth data.

The feature vector, $\left(\frac{h_{bb}}{H_F}, \frac{w_{bb}}{W_F}\right)$ of each frame is fed to an appropriate trained Support Vector Machine. SVM consists of the following optimization problem: let $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i \in R^n$ and $\mathbf{y}_i \in \{0,1,\dots,m\}$ is the sample label index. The optimization problem is formulated as:

$$\min_{w,b,\xi} \frac{1}{2}\mathbf{w}^T\ \mathbf{w} + \mathrm{C}\sum_{i=1}^{n}\xi_i \qquad (1)$$

Subject to: $\mathrm{y}_i(\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x_i}) + \mathrm{b}) \geq 1 - \xi_i,\ i = 1, \dots, N \geq 0$ (2)

$$\xi_i\ \geq 0,\ i = 1, ..., \mathrm{N} \qquad (3)$$

where C is the tuning parameter used to balance the margin and training error and ξ_i are called the slack variables that are related to the soft margin. In this work in order to find the best parameterization of the SVM, "grid-search" along with cross validation is applied. The applied SVM uses the RBF kernel, i.e. $K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right),\ \gamma > 0$. Thus, the grid search is used to define the parameter C of the optimization problem, as well as the kernel parameter γ. The libSVM [2] library has been used for the SVM implementation.

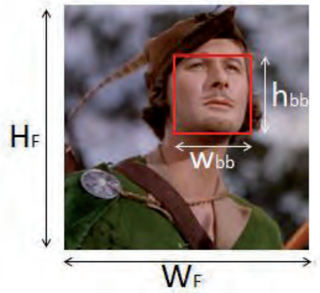


Figure 3: Example of a face tracked by the tracking algorithm in [15], along with the quantities involved in the proposed features.

4 EXPERIMENTAL EVALUATION

In this section experimental results are provided. In addition 5-fold cross validation was applied in the sample dataset that was created (see bellow).

Two types of results are presented. The frame based results refer to the classification of each frame of the shot in one of the shot types, whereas the shot-based results refer to the classification of the entire shot based on the classification of the individual frames of the shot. In order to decide on the type of an entire shot majority voting on the derived labels of the frames contained in the shot is applied, similar to [3], [6] and [10]. In other worlds, the majority of the labeled frames, that compose a shot, characterize the type of the entire shot.

In order to calculate the classification accuracy at the frame/shot level we use the same metric as in [3]. That is,

$$A = \frac{N_{CC}}{N_{GT}}$$

where N_{CC} is the frames/shots correctly classified to a specific shot type and N_{GT} is the total number of the frames/shots labeled in the ground truth data as belonging to this shot type. In this Section confusion matrixes are also evaluated.

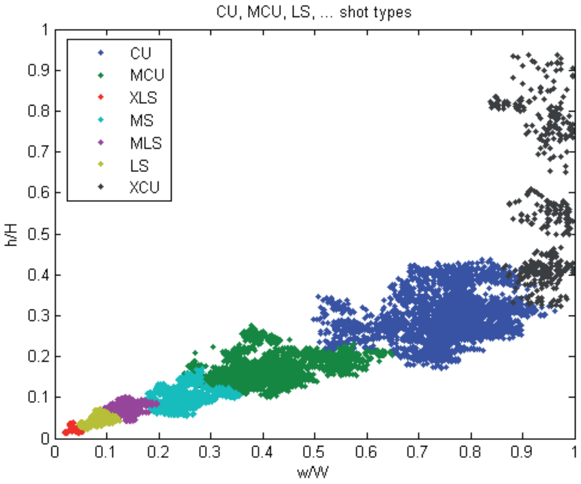


Figure 4: Feature values for frames from various shot types in the ground truth data

Our dataset is composed of 173 shots and 12178 frames, compiled from different movie genres by different directors. In each frame an actor is present. Most of the actors are looking towards the camera. Adhering to the definitions in Section 2, we have labeled each shot manually to construct the ground truth data (shot labels). The database includes only shots where only one actor is depicted so as to simplify its construction. However, the proposed method can easily generalized to work on shots depicting more than one actor, by selecting the most "dominant" one. This can be done for example by using a rule similar to the one proposed in [3] which decides on the dominant subject from all depicted ones.

For the extraction of facial bounding boxes, we manually mark the face bounding box in the first frame of each shot and then we apply the tracking algorithm described in [15] in order to track the face in the remaining frames. A common problem in tracking is the occlusion of the tracked object, in our case the face, that might lead to loss of target. To cope with this problem one can reinitialize the tracked region or periodically use an effective face detector. In our database, for reasons of simplicity, we exclude shots where occlusions occur since solving tracking problems is beyond the scope of this paper.

The method in [3] was applied to the database described above in order to obtain frame-based and shot-based experimental results. Frame-based results are shown in the confusion matrix in Table 1.

Table 1: Frame-based results according to algorithm in [3]

	XCU	CU	MCU	MS	MLS	LS	XLS
XCU	0.92	0.08	0	0	0	0	0
CU	0.01	0.86	0.13	0	0	0	0
MCU	0	0.08	0.92	0	0	0	0
MS	0	0	0	0.9	0.1	0	0
MLS	0	0	0	0	0.88	0.12	0
LS	0	0	0	0	0.007	0.96	0.033
XLS	0	0	0	0	0	0.12	0.88

Based on this metric the frame-based accuracy is 90,3%. The shot-based overall classification accuracy for this method is 91,4% and the corresponding confusion matrix is presented in Table 2.

Table 2: Shot-based results according to algorithm in [3]

	XCU	CU	MCU	MS	MLS	LS	XLS
XCU	1	0	0	0	0	0	0
CU	0	0.78	0.22	0	0	0	0
MCU	0	0	0.94	0.06	0	0	0
MS	0	0	0	1	0	0	0
MLS	0	0	0	0	0.85	0.15	0
LS	0	0	0	0	0	1	0
XLS	0	0	0	0	0	0.17	0.83

When the proposed method was applied on the same dataset, the frame-based overall accuracy was 98,3% and the corresponding confusion matrix is presented in Table 3, whereas the shot based accuracy was 100%.

Table 3: Frame-based classification results of the proposed method

	XCU	CU	MCU	MS	MLS	LS	XLS
XCU	0.97	0.03	0	0	0	0	0
CU	0.004	0.99	0.006	0	0	0	0
MCU	0	0.003	0.98	0.017	0	0	0
MS	0	0	0.019	0.976	0.005	0	0
MLS	0	0	0	0.023	0.97	0.007	0
LS	0	0	0	0	0.004	0.996	0
XLS	0	0	0	0	0	0	1

5 CONCLUSION

Movie shot type classification is a challenging problem. In this paper, we propose a method based on the height and width of the facial image in combination with the corresponding height and width of the video frame. The ratios are fed in a properly trained SVM classifier obtaining 100% accuracy at the shot type level. The main drawback of our method is that it is based on facial images, and thus, it is capable of shot type classification only in videos that contain faces.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n 287674 (3DTVS). The publication reflects only the authors' views. The EU is not liable for any use that may be made of the information contained therein.

References

[1] S. Benini, L. Canini, and R. Leonardi. Estimating cinematographic scene depth in movie shots. In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 855-860, July 2010.

[2] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu. edu.tw/_cjlin/libsvm.

[3] I. Cherif, V. Solachidis, and I. Pitas. Shot type identification of movie content. In Proceedings of Signal Processing and its Applications (ISSPA), 2007.

[4] S. Chen, M. Shyu, C. Zhang, L. Luo, and M. Chen. Detection of soccer goal shots using joint multimedia features and classification rules. In Proceedings of 4th International Workshop on Multimedia Data Mining (MDM/KDD), pp. 36-44, 2003.

[5] D. Arijon. Grammar of the Film Language. Silman-James Press, 1991

[6] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. IEEE Transactions on Image Processing, vol. 12, pp. 796-807, July 2003.

[7] J. M. Keller, R. M. Crownover, and R. Y. Chen. Characteristics of natural scenes related to the fractal dimension. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, no. 5, pp. 621-627, September 1987.

[8] B. J. Super and A. C. Bovik. Shape from texture using local spectral moments. IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 17, no. 4, pp. 333-343, April 1995.

[9] T. Xiaofeng, L .Qingshan, and L. Hanqing. Shot classification in broadcast soccer video. Electronic Letters on Computer Vision and Image Analysis, vol. 7, no. 1, 2008.

[10] C. J. W. Engsiong, and X. Changsheng. Soccer replay detection using scene transition structure analysis. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. 433-436, March 2005.

[11] L. Wang, M. Lew, and G. Xu. Offense based temporal segmentation for event detection in soccer video. In Proceedings of 6th ACM SIGMM International Workshop on Multimedia information retrieval, MIR, pp. 259-266, New York, USA, 2004.

[12] M. Xu, J. Wang, M. A. Hasan, X. He, C. Xu, H. Lu, and J.S. Jin. Using context saliency for movie shot classification. In Proceedings of 18th IEEE International Conference on Image Processing (ICIP), pp. 3653-3656, September 2011.

[13] P. Xu, L. Xie, S. F. Chang, A. Divakaran, A. Vetro, and H. Sun. Algorithms and system for segmentation and structure analysis in soccer video, In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 721-724, August 2001.

[14] X. Yu, L. Duan and Q. Tian. Shot classification of sports video based on features in motion vector field. In Proceedings of 3rd IEEE Pacific-Rim Conference on Multimedia Proceedings, pp. 235-260, December 2002.

[15] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Transactions on Image Processing, vol. 13, pp. 1434-1456, 2004.

A System for Task-Oriented Content Analysis and Search in Media Production

W. Bailer[1], P. Altendorf[2], A. Messina[3], F. Negro[3], G. Thallinger[1]

[1] JOANNEUM RESEARCH Forschungsgesellschaft mbH, DIGITAL – Institute for Information and Communication Technologies, Austria
[2] Institut für Rundfunktechnik GmbH, Germany
[3] RAI - Radiotelevisione Italiana - Centre for Research and Technological Innovation, Italy

E-mail: [1] {werner.bailer,georg.thallinger}@joanneum.at, [2] altendorf@irt.de,
[3] {a.messina,f.negro}@rai.it

Abstract: **The project TOSCA-MP aims at providing more efficient ways of annotating and searching audiovisual content in professional media production processes. To achieve this goal, automatic content analysis tools and novel search methods are developed. In this paper, we first present an overview of the components in a system addressing this challenge. We then analyse the scenarios, use cases and user tasks that such a system needs to support. Based on this, a logical and a technical view of the system design are presented, strongly relying on service oriented architectures and the recent FIMS standard.**

Keywords: media search, content analysis, content annotation, usage scenarios, system design, SOA, FIMS

1 INTRODUCTION

The work presented in this paper focuses on professional audiovisual media production and archiving workflows, in particular workflows deployed at broadcasters and media production houses. Many of the media professionals' tasks in such a workflow include searching for content in different modalities, such as finding appropriate clips from recent material to be included in the production, locating relevant archive material or reviewing relevant sections of other media's coverage on the same event or topic. Media production and archiving workflows are changing rapidly due to the appearance of a wide range of new ways of media production, distribution and consumption. These changes pose several challenges for the technology used in media production and archiving.

For reasons of flexibility and efficiency, the workflows need to be increasingly networked and distributed. Journalists on location are taking over more and more tasks that were previously done by other dedicated staff at a local studio or central facility. Due to increased collaboration and content syndication, media search is no longer a problem of searching an in-house repository, but one of searching distributed large-scale repositories, which are multilingual and heterogeneous in terms of their structure and data models. As content owners want to keep control over their repositories, centralised indexing is not an option. Novel networked media search technologies are required. The main challenge for this approach is the successful development of an infrastructure capable of handling multiple heterogeneous repositories distributed across the network, encompassing a sophisticated user interface for the seamless integration of different distributed repository types and services.

The transition from traditional tape-based production to an entirely file-based workflow is now a reality that needs to be managed efficiently. This change, however, requires the adoption of information technology (IT) as a fundamental technical enabler for production and archiving of content. Broadcasters are now in the position to embrace a new era, where barriers between dedicated technologies like VTRs (Video Tape Recorders) or SDI (Serial Digital Interface) networks and IT components are fading to nought. However, using IT means interfacing with a highly heterogeneous market made up of a broad panel of different standards and industries. A solution is to apply for the workflow principles of open service-oriented architecture (SOA), that allow customising systems on the basis of standardised component interfaces. The availability of off-the-shelf distributed, IT-based services for search and retrieval, seamlessly integrated with the production, publication, and archival processes is considered crucial for the near future. Distributed and networked search technologies represent a viable solution for implementing systems supporting media production processes in a sustainable and future-proof way.

The paradigm shift we see in media production will not result in a new set of fixed workflows, but the new workflows will be dynamic and constantly changing, as novel IT-based production and distribution technologies emerge. The tools for tomorrow's media production and archiving workflows must thus be able to adapt to the user's tasks, content types and production context. This adaptation can be enabled by task models, means of implicit and explicit user feedback and integration of benchmarking capabilities. Despite the complexity of search systems distributed throughout the network, tools must be user-centric, and provide a single, integrated user

Corresponding author: Georg Thallinger, JOANNEUM RESEARCH, DIGITAL – Institute for Information and Communication Technologies, Steyrergasse 17, 8010 Graz, Austria, phone: +43 316 876 1240

interface, in order to support professionals in their daily tasks.

The rest of this paper is organised as follows. Section 2 presents an overview of a system as being developed by the TOSCA-MP project[1], addressing these challenges. In Section 3 we discuss the use cases and tasks considered, as well as the derived requirements. Section 4 presents the system design and the standardised service interfaces being implemented. Finally, Section 5 provides conclusions and outlook.

2 OVERVIEW

The system described in this paper is developed in the context of the TOSCA-MP project, which aims at developing user-centric content annotation and search tools for professionals in networked media production and archiving (television, radio, online), addressing their specific use cases and workflow requirements.

The project performs research and development in the following technology areas. For advanced multimodal information extraction and semantic enrichment, scalable and distributed content processing methods are developed, focusing on visual content and speech. Existing approaches are adapted in order to be applied in task and genre adaptive ways. Task adaptation is achieved by formalised models of user tasks in the media production workflow. These models are used for the orchestration and configuration of automatic services as well as for benchmarking the tools and services.

A second key area concerns methods for searching across heterogeneous networked content repositories. In order to enable professionals in media production and archiving to seamlessly access content and indexes across distributed heterogeneous repositories in the network, a distributed repository framework is being developed. This repository framework will allow instant access to a large network of distributed multimedia databases and including beyond state-of-the-art metadata linking and alignment. The distributed repositories can be accessed through a single user interface that provides novel methods for result presentation, semi-automatic annotation and means of providing implicit user feedback. All these components are integrated in an open service-oriented architecture.

3 SCENARIOS, TASKS & REQUIREMENTS

3.1 Usage Scenarios and Requirements

The TOSCA-MP consortium chose to use the S-Cube methodology [3] to identify a two-level hierarchical description of usage scenarios. This methodology distinguishes between high-level goals (named business goals), i.e. target conditions which are to be met by the system from mainly a business process-oriented perspective, and more detailed scenarios, which describe more practical settings in which actors and systems interact to achieve a specific result.

Overall, the consortium identified 10 different business goals and a total number of 15 scenarios, i.e. at least one scenario for each identified business goal. Each business goal comes with a textual description which illustrates the rationale underlying the goal and its main objectives. Business goals and scenarios span a considerable range of real-world media production processes, and capture an important portion of the media production value chain, all of which could actually benefit from the employment of TOSCA-MP results [4].

The identified business goals and scenarios fall into broadly four categories: content access & retrieval, news service distribution, assisted production, and infrastructure. Table 1 summarises the description of the main identified business goals and scenarios.

Business goal	Scenario for business goal	Category
Fast retrieval of very recent material	Fast content discovery for news production	Content access & retrieval
Efficient retrieval of historical archive material	Searching archived material, including deep archive search	Content access & retrieval
Access to international feeds and their use in news production	Distributed semantic search and retrieval of multilingual content, dynamic configuration of features for content enrichment, machine-supported subtitle generation	Content access & retrieval, assisted production
News daily report with event detection and impact analysis	Assisted production of news stories using distributed multilingual sources	News service distribution
Assisted production of sports events	Summary of downhill race, summary of downhill world cup season	Assisted production
Distributed repository for all steps in metadata production and usage chain	Distributed content metadata production and post-production, distributed search and recommendation	Infrastructure

Table 1. Summary of identified business goals and scenarios.

[1] http://www.tosca-mp.eu

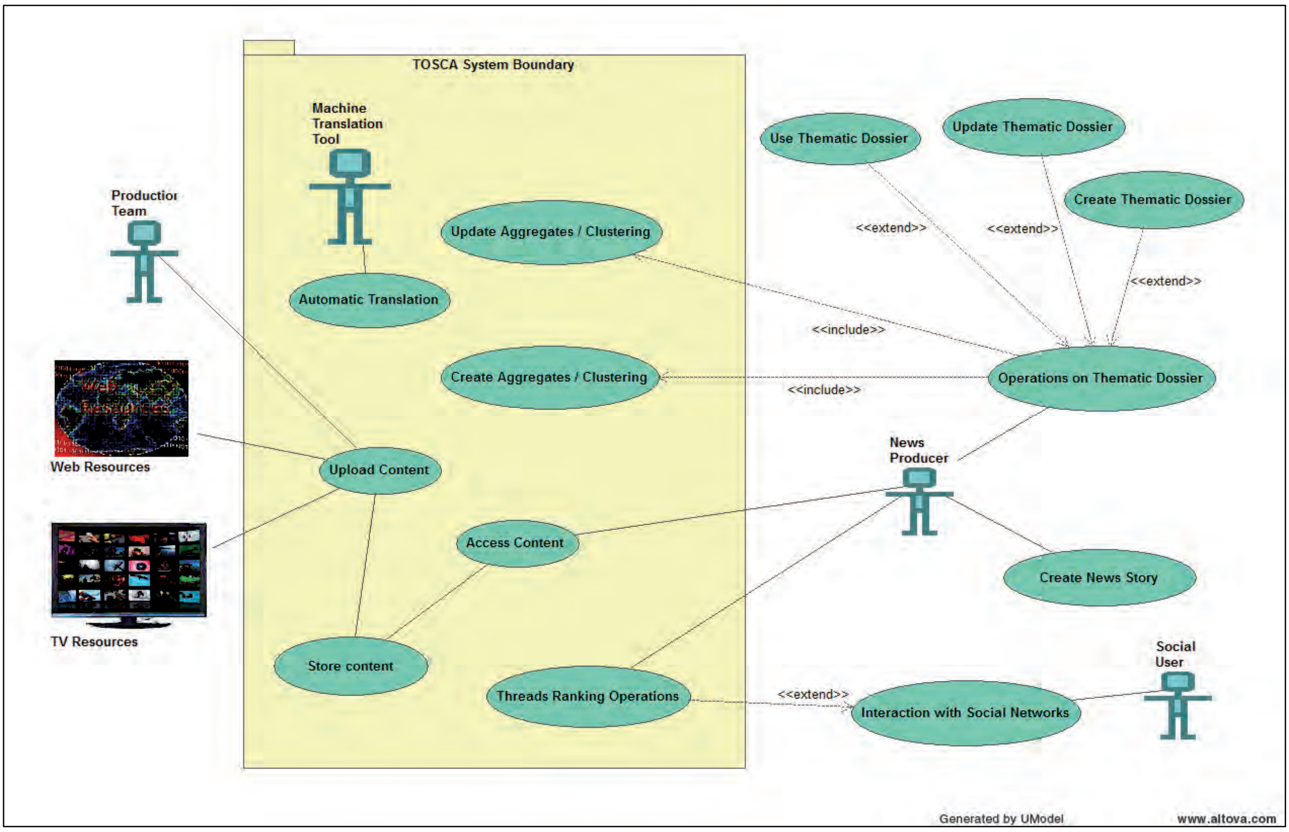


Figure 1. Example of usage scenario connected with business goal "Assisted production of news stories using distributed multilingual sources".

Scenario descriptions are complemented by UML use case diagrams, which go into more detail by breaking down scenarios into the actors, components and functionalities involved. Thus they provide insights about what functionalities will be part of the system and what actors will be involved in the practical test cases. An example of scenario description is depicted in Figure 1. At the headquarters of an Italian broadcaster, a news producer has to create a story for the evening edition of the newscast. He can choose the subject to talk about, so he would like to select the one which is, at the moment, considered the main thread on the net. The news producer queries a central system to give him the threads ranked according to the number of published items on the web and on television and to the behaviour of users on different social networks. These items are uploaded in the system by a dedicated production team. After having selected the subject, he asks the central system to get a constant update on the selected theme. The update comes in form of a multimedia report (or dossier) containing multilingual material properly aggregated/clustered coming from several distributed sources of information, both television and web. The Italian news producer can enrich his news story taking suggestions in his own language coming from these heterogeneous materials.

Based on the scenario descriptions and use case diagrams, also workflow and interaction diagrams had been developed. An analysis of these workflow diagrams has helped identifying similar workflow patterns and to isolate the related functionalities into workflow building blocks that can be plugged into higher level workflow descriptions. As an example of this abstraction process, Figure 2 depicts the generic sequence diagram for metadata generation, which involves some of the main components of the architecture (see Section 4). The requirements for the components involved in each of these workflows, and in particular workflow management and repository components which form the core of the system, have been derived by means of this approach.

3.2 User Tasks

The project has collected a set of real-world tasks in the media production workflow that are considered within the scope of TOSCA-MP. In order to help to characterise them and to formalise task models and success metrics based on them, properties of these tasks have been collected by performing a survey[2], and using information gathered by the EBU MIM/SCAIE working group[3]. A task is defined as a sequence of actions performed by one or more users to achieve a defined goal in the production process, possibly using a set of tools. The task has a defined set of input documents and produces a set of output documents. For example, a "Content Search Task" would be defined as "The action performed by a journalist to find an audiovisual content item with a specified title".

The collected/described tasks cover many aspects of the audiovisual media production workflows, such as

[2] The survey is still open at http://www.tosca-mp.eu/tasksurvey

[3] http://tech.ebu.ch/groups/pscaie

annotation and documentation of incoming news and sports material as well as archive content, search for multilingual news content, personalised news production and live subtitling of news. Other tasks deal with gathering material for a documentary, performing editing in a distributed environment and creating highlight summaries for news and sports content. A complete list and more detailed information about the tasks can be found in [4].

We are currently working on a formalisation of the collected tasks descriptions, in order to obtain machine readable descriptions of tasks. The ConcurTaskTrees (CTT) formalism [6], a graphical model for tasks, is used. These will be used to orchestrate the services in the metadata production management framework described in the next sections, and to derive benchmarks from the expected tasks results for evaluating individual tools or sets of tools developed in the project. The latter enables the integration of benchmarking into workflows and to dynamically adapt the processing tools to new types of input.

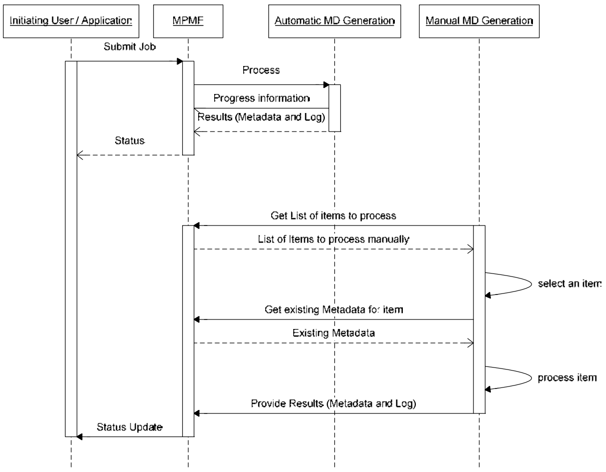

Initiating User / Application
MPMF
Automatic MD Generation
Manual MD Generation
Submit Job
Process
Progress information
Results (Metadata and Log)
Status
Get List of items to process
List of Items to process manually
select an item
Get existing Metadata for item
Existing Metadata
process item
Provide Results (Metadata and Log)
Status Update

Figure 2. Interaction diagram of metadata generation process.

4 SYSTEM DESIGN

4.1 Logical View

The analysis of the above mentioned tasks and requirements first resulted in a logical view of the system: the Logical System Design (LSD). The LSD as depicted in Figure 3 is composed of four subsystems: the Metadata Production Management Framework (MPMF), the Distributed Repository Framework (DRF) and sets of services and graphical user interface components.

The MPMF consists of an Enterprise Service Bus (ESB) and a process engine and is the central system component which integrates all other components. Tasks and workflows are modelled in this integration platform and orchestrated by means of message mediation between the components.

The DRF is the data management centre of the system where internal databases, triple stores, file systems and other types of storage are administrated. It is a scalable integrated repository for data used at different stages of a media production workflow.

All components providing dedicated functionalities like the processing of data in the production workflow e.g. for automatic metadata generation, semantic enrichment and linking etc. are exposed as services. This allows loosely coupling of components and thus a highly flexible architecture.

Several GUIs are needed to allow users to interact with the system, i.e. to administer the system, to manage and monitor processes, to launch queries, inspect results etc. In the LSD all GUIs are packaged together as one logical component which is connected to the MPMF and the DRF, but there will be several GUIs providing specific functionalities.

Figure 3 also shows the relations and the different connections between the four main subsystems. We differentiate between connections used to exchange essence and metadata between the system components; and those for control purposes only and thus only exchange metadata.

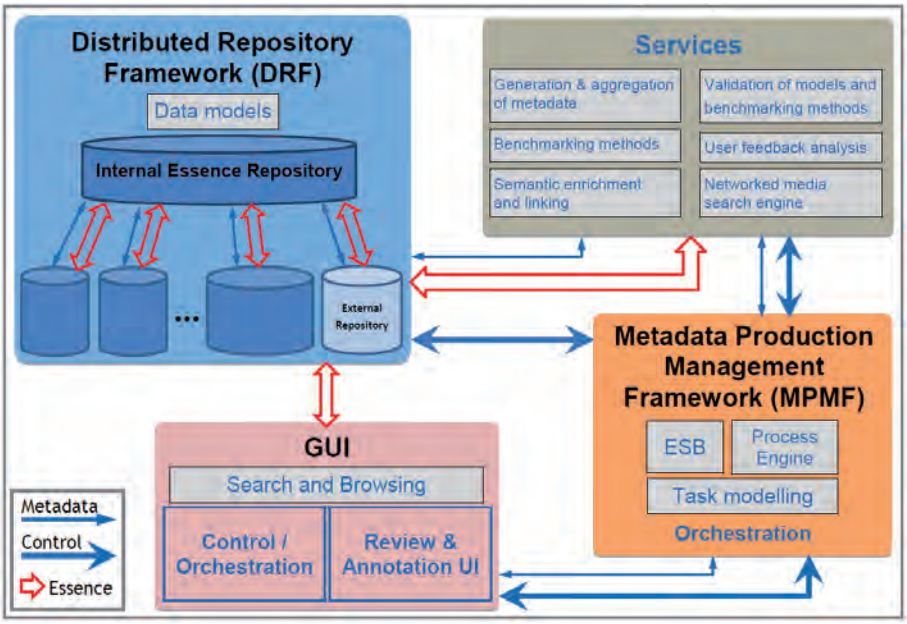

Distributed Repository Framework (DRF)
Data models
Internal Essence Repository
...
External Repository
Services
Generation & aggregation of metadata
Validation of models and benchmarking methods
Benchmarking methods
User feedback analysis
Semantic enrichment and linking
Networked media search engine
Metadata Production Management Framework (MPMF)
ESB
Process Engine
Task modelling
Orchestration
GUI
Search and Browsing
Control / Orchestration
Review & Annotation UI
Metadata
Control
Essence

Figure 3: Logical System Design (simplified version).

4.2 Technical View

Systems for file-based media production became increasingly complex in the past years. Standard IT-based hardware and software components are typically tightly coupled in spider web-like system environments. To reduce the complexity in such heterogeneous systems the paradigm of Service Oriented Architectures (SOA) is increasingly adopted. SOA-based systems consist of individual services that are loosely connected with each other by a service bus (see Figure 4) [1].

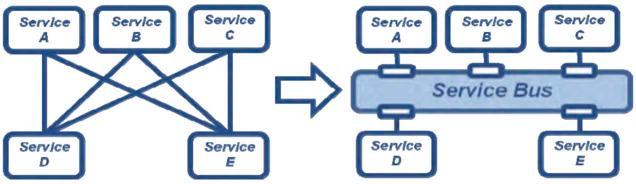


Figure 4: From spider web to service bus [5].

The SOA-based Technical System Design (TSD) consists of five layers and provides a more technical view on the system (see Figure 5). The Application Layer forms the top layer which provides GUIs allowing users to access the system. Applications implementing the GUIs interact with the services in the Orchestration Layer.

The Orchestration Layer is a wrapping layer for the Service Layer and the MPMF. The Service Layer provides access to and control of the different components in the system through web service interfaces. The MPMF contains the definitions of service orchestrations, called processes. An ESB as open integration platform allows in combination with a process engine the execution of processes started by the Application Layer to invoke business logics.

The Component Layer below the Orchestration Layer contains the actual components for metadata extraction, search etc. as well as the DRF. The functionalities of these components are exposed as services to the Orchestration Layer using web service interfaces in the Service Layer.

The Data Layer handles the exchange of content (essence and metadata) between the DRF and the components or the GUI.

4.3 A SOA Framework for Media Services

In addition to a reduced system complexity, the SOA approach enables easier integration of new components and better scalability. Many vendors in the broadcast industry already offer service-oriented interfaces for their systems [5]. However, even if the systems generally serve the same purpose, their interfaces typically differ in their functionality, complexity and data model. To reduce the integration efforts a joint task force initiated by EBU and AMWA called FIMS [4] targets the standardisation of interfaces and formats.

The FIMS specification of a common SOA framework for media services [2] describes a high-level architecture and framework. In its initial version it also defines service interfaces for three basic media services: capture,

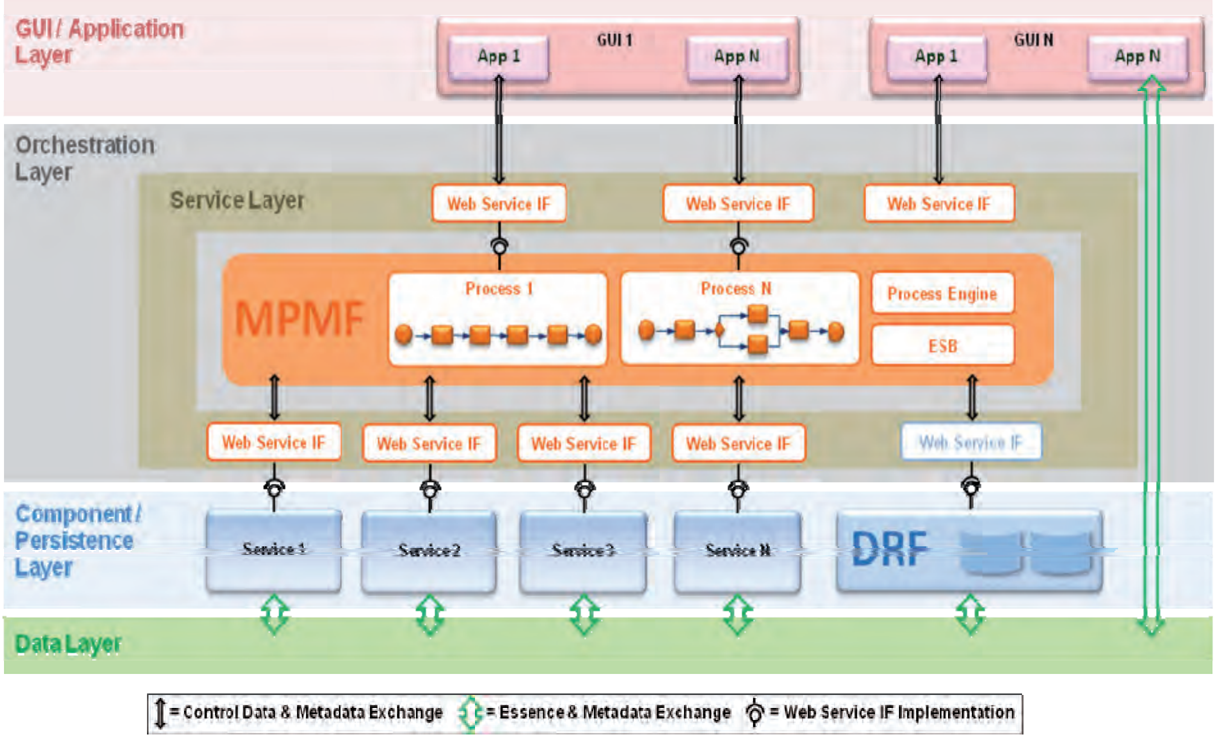


Figure 5: Technical System Design.

[4] Framework for Interoperable Media Service, http://wiki.amwa.tv/ebu/

transform and transfer.

The services used by the system described in this paper will be implemented following the baseline of the FIMS specification as far as possible.

5 CONCLUSION & OUTLOOK

The TOSCA-MP project aims at providing more efficient ways of annotating and searching audiovisual content in professional media production use cases. To achieve this goal, automatic content analysis tools and novel search methods are developed.

In this paper, we have analysed the scenarios, use cases and user tasks that a system for this purpose needs to support. Based on this, a logical and a technical view of the system design was presented. The system design is based on service-oriented architectures and will make use of the recent FIMS standard.

At the time of writing this paper, the specification of the systems has been completed. Many of the related documents are public, available at http://tosca-mp.eu/publications/public-deliverables. The implementation of initial components is in progress. A first integrated version of the system is expected to be ready by mid 2013.

Acknowledgement

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287532, TOSCA-MP – Task-oriented search and content annotation for media production (http://www.tosca-mp.eu). The project brings together 10 partners from 5 European countries, including industry partners providing solutions for the media industry, public service broadcasters as well as their European association, a university and research centres.

References

[1] J. Footen, “The Service-Oriented Media Enterprise”, Focal Press, Burlington 2008.

[2] FIMS, “FIMS Media SOA Framework 1.0”, September 2011, http://wiki.amwa.tv/ebu/index.php/Framework_Specification (retrieved May 9th, 2012).

[3] E. Di Nitto, P. Plebani, “Describing Case Studies: the S-Cube approach”, Technical Report, Dipartimento di Elettronica ed Informazione Politecnico di Milano, 2010. Available at http://www.s-cube-network.eu/pdfs/EC-document-20100713.pdf

[4] W. Bailer, A. Messina, “Relevant Tasks in A/V media Production Workflow”, TOSCA-MP Public deliverable D4.1.

[5] Elser, Matthias et al., “Integration von Produktionssystemen (Proof of Concept) – ein anderer Ansatz”, FKT 6/2012, 2012.

[6] F. Paterno, C. Mancini, S. Meniconi. "ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models," Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction Pages, pp.362-369, Syndey, 1997.

Digital Media Content II

Session 3A

Chaired by Pierre-Yves Danet, France Telecom

Scene Modelling For Richer Media Content

Chris Budd, Jean-Yves Guillemaut, Martin Klaudiny, Adrian Hilton
University of Surrey
{c.budd, j.guillemaut, m.klaudiny, a.hilton}@surrey.ac.uk

Abstract: **The SCENE project aims to develop a novel scene representation for digital media which bridges the gap between sample-based (video) and model-based (CGI) methods allowing production and delivery of richer media experiences. The usability and potential benefits of this representation hinge on the development of tools to utilize this format in a production environment. The capture of 3D video and subsequent temporal alignment into a 4D representation provide a key set of tools on which to demonstrate the potential benefits of the representation. In this paper we present a pipeline of tools for the capture, reconstruction and temporal alignment of multi-camera video data, which are currently being developed as part of the project. We demonstrate this pipeline on both full-body and facial capture scenarios, showing high quality temporally consistent results. We also provide some insights into how future developments will enable the extension of the pipeline to more general scenarios containing arbitrary scene content.**

Keywords: Reconstruction, Mesh Tracking, Temporal Consistency, Multi-View Video

1 INTRODUCTION

The SCENE project aims to develop a novel scene representation for digital media content that improves on the capabilities of either purely sample-based (video) or model-based (CGI) methods. Sample-based approaches directly capture a scene representation using video cameras or other types of visual sensors. This mode of acquisition produces high-fidelity representations but is very rigid as obtained representations cannot be modified in an intuitive manner (for example viewpoint or content cannot be altered without need to re-capture the entire scene). In contrast, model-based techniques generate a virtual representation providing ultimate flexibility in terms of manipulation but lacking realism compared to a sample-based approach (this loss of realism is referred to as the "Uncanny Valley" in the field of computer graphics and animation). SCENE aims to develop a novel scene representation which will bridge the gaps between the two previous representations introducing a highly realistic representation of the real world that is fully manipulatable.

Achieving such a representation will require several key advances in the areas of sensor development, scene geometry and appearance representation, algorithms and data structures for storage, distribution and rendering, and content manipulation. In this paper we focus on the specific problem of scene geometry representation which is central to this aim. To support data manipulation without loss of realism, it is essential that the representation is not only accurate at each time instant but also temporally consistent (that is that surface displacements are correctly tracked across frames). Temporal consistent is for example necessary to allow scene manipulation occuring in a single frame to be automatically propagated across a sequence or database of related content. We address this issue by introducing a generic framework for spatio-temporally consistent scene modelling. This uses a multi-view studio capture system able to extract accurate scene models and a non-rigid dense surface tracking approach.

The paper is structured as follows. We start by providing an overview of the pipeline used to extract a spatio-temporally consistent representation. Next we present a method for high-quality scene modelling from a multi-camera studio system; at this stage, we ignore the temporal aspect of the problem and focus on extracting an accurate model at each frame. In the following section, we show how this frame-dependent reconstruction can be made temporally consistent via our proposed dense non-rigid surface tracking approach. The results section illustrates the performance of the pipeline on a range of example sequences and demonstrates the validity of the approach for full-body and face capture. Finally we conclude and describe ways to extend the approach to more general scenarios containing arbitrary scenes.

2 PIPELINE OVERVIEW

The pipeline presented allows the capture, reconstruction and temporal alignment of 3D video data. The first stage after the initial multiple-view video capture is segmentation, which yields silhouette images. The silhouette images are used in creation of visual hulls which provide a rough estimate of shape at each time frame. The visual hulls are combined with stereo matching between pairs of the multiple video streams to produce refined and accurate surface models [20]. A single frame from this sequence of reconstructed meshes is then deformed into alignment with all other frames to yield a temporally consistent representation [3]. To achieve this a non-sequential alignment approach is taken which first compares all frames from the sequence (or multiple sequences) to be aligned based on shape similarity. This comparison allows construction of a tree which orders the reconstructed meshes into multiple paths where consecutive frames have minimal difference in shape. Tracking along these paths allows temporal alignment of all frames.

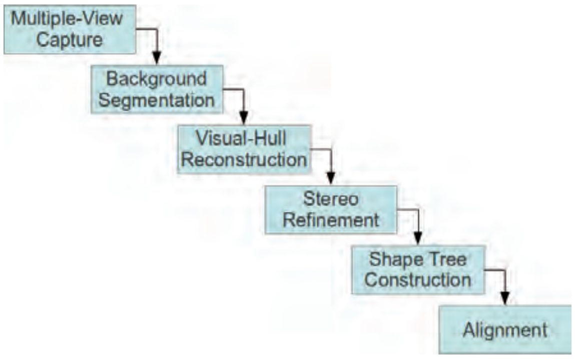


Figure 1: Pipeline Overview for Full-Body Capture

For facial capture and processing the process differs only in the frame independent surface reconstruction stages. With facial data no visual hull reconstruction takes place. Instead stereo matching combined with Poisson surface reconstruction is employed to produce the surface meshes.

3 3D RECONSTRUCTION

The ability to make meaningful digital observations of the real world with cameras, range scanners and other sensory devices has lead to the production of many forms of media and digital content. Transmission of this media has to date been largely 2D in the form of video. Demand is however emerging for more interactive and flexible content. SCENE is targeting the production of a new scene representation format that allows transmission and interaction with 3D content at the viewers discretion. An important stage in this process is creating the 3D models and scene content which can be interacted with and viewed from user selected angles.

Based on multiple-view video data it is possible to reconstruct models of people within the scene. Kanade and Rander [11] pioneered surface capture for sequences of human motion with their Virtualized Reality System. Laurentini et al [14] introduce the concept of a visual hull as the bounding region for geometry represented by the intersection of the projection of multiple silhouette images. Sand et al [17] combine information from shape from silhouette techniques with skeletal reconstruction from a commercial motion capture system. Visual hull reconstruction gives the surface shape whilst the traditional marker-based motion capture system allows capture of the underlying skeleton. Seitz and Dyer [18] developed a voxel colouring technique which uses photometric information. Their algorithm identifies voxels within the visual hull of consistent colour in multiple views and uses them as constraints in reconstruction. The concept of the Photo Hull takes this a step further [13] and provides a means to deal with occlusions in visual hull reconstruction.

Another significant body of research on character reconstruction has come from multi-view stereo techniques. One of the earliest techniques for dealing with multiple view stereo is presented by Okutomi and Kanade [15]. Their approach uses a number of cameras at increasing base-lines to produce multiple estimates of depth. Roy and Cox [16] introduce a global approach to the multiple camera stereo problem. They transform the problem into a maximum flow problem and solve for the disparity surface as the minimum cut. Vogiatzis et al [25] use volumetric graph cuts. They use the visual hull as a prior model for photo-consistency, however fail to take account of silhouette information in the final reconstruction. Sinha and Pollefeys [19] include silhouette constraints however there method is restricted to genus-0 surfaces preventing the reconstruction of concavities in objects such as a ring formed by linking ones arms.

The work demonstrated here combines visual hull reconstruction with multi-view stereo refinement [20]. Figure 2 details the steps involved in this process. First, performances are captured in our studio using 8 Thompson Viper cameras HD-SDI 20-bit 4:2:2 format with 1920 x 1080 resolution at 25Hz progressive scan. Gen-locked and timecode synchronized video is recorded using 8 DVS HD capture cards direct to disk. Cameras are intrinsically and extrinsically calibrated using checker board and wand calibration techniques respectively.

The next step is foreground extraction. By virtue of the studio environment with consistent background colour it is possible to employ chroma-key matting. An alpha matte is produced which represents the foreground opacity in case of clear segmentation and foreground colour where a pixel is a mix of foreground and background. Using the extracted silhouette image, shape from silhouette is used to extract a visual hull. The visual hull represents the maximum possible volume that the character mesh can occupy. The visual hull alone only provides a rough estimate of shape. Concavities within the mesh are not reconstructed and no consistency is enforced between the separate camera views which can result in false positive volumes.

To refine the visual hull, stereo matching between pairs of the cameras is used to give further estimation of depth at matched regions of the images. Features are matched between the image pairs using a Canny-Deriche edge detector. Correspondence is constrained to satisfy the epipolar geometry and to be feasible based on the visual hull. These correspondences are validated by enforcing left-right consistency between views; a feature in the left view matching one in the right must be reciprocated by the corresponding feature in the right. Feature matching provides a sparse set of corresponding line segments which potentially lie on the target surface.

Stereo refinement of the visual hull involves extracting a surface based on these features and is constrained to lie within the visual hull. To achieve this a global optimization approach is adopted which formulates the problem in a maximum-flow / minimum-cut framework yielding a volumetric representation of character mesh. Each node of this graph represents a discrete volume element or voxel

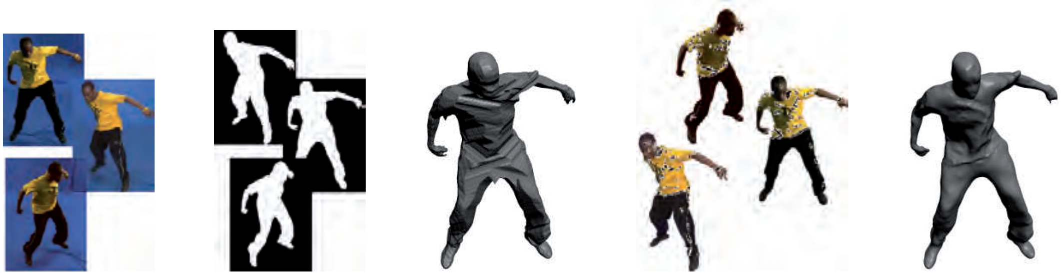

(a) Captured Images (b) Foreground Matting (c) Visual Hull (d) Feature Matching (e) Stereo Refinement

Figure 2: Overview of Reconstruction Process

within the scene. The edges are weighted by a cost function which quantifies consistency in appearance between camera images. The final surface is subsequently extracted as the minimum cut of this graph and a triangulated mesh can be extracted using the marching cubes algorithm.

4 DENSE SURFACE TRACKING

A key aim of the SCENE project involves the development of techniques for editing and manipulating 3D scene information in a re-usable and adaptable manner. These techniques should enable support for interaction, scene modification and object replacement. Traditionally, reconstruction and 3D capture techniques have focused on producing high quality, representative meshes for each frame. These meshes are however time independent; that is the number of vertices and topology of the meshes vary at each time frame. The result is that a number of common editing tasks are time consuming and laborious. Mesh edits applied to a single time frame are not easily propagated throughout a sequence. Texture edits need to be applied at every time frame rather than on a single instance of the mesh. With a temporally consistent representation edits can be propagated with techniques such as space-time editing and texture edits would need only be applied to a single texture.

With this in mind recent work has focused on the production of a temporally consistent sequence of meshes [7, 6, 21]. Much work has focused on tracking features in the original multiple view video [20, 23, 22]. Work from Aguiar et al [7] takes a laser scanned model and aligns it with each frame of a sequence using SIFT features and refines the model with silhouette rim constraints. While Aguiar et al fit a laser scanned model Vlasic et al [24] and Gall et al [8] fit a predefined linear blend skinned model throughout a sequence. Moving away from the use of predefined models Cagniart et al [5] fit a well reconstructed frame taken from a sequence of non-temporally consistent meshes using geometric surface patches and a Laplacian regularizer. Subsequently, they improve the patch based approach with the introduction of a probabilistic framework which better deals with noise and reconstruction errors [6]. All these approaches share one inherent flaw; they sequentially align the sequences on a frame-to-frame basis. Errors in the tracking thus accumulate over time leading to progressively worse alignment results.

Attempting to overcome the issues involved with sequentially aligning long sequences of video the authors of this paper have presented techniques for non-sequential alignment of both full-body [3, 10] and face capture data [4]. It is based on these techniques that the tools presented in this paper have been developed. Within these techniques a fully connected graph is produced based on a measure of similarity between the frames of the sequence. A spanning tree is subsequently constructed which represents the multiple paths of deformation via which the temporally consistent representation is built. This methodology also permits alignment of multiple sequences into a single temporally consistent representation, something which has not been previously possible without manual interaction.

The output of the employed stereo reconstruction technique is a sequence of unstructured 3D meshes. That is meshes which have a unique set of vertices and varying topology at each frame. From these unstructured meshes we wish to construct a sequence of meshes with consistent topology, where only vertex position varies at each frame. A traditional approach would take a single frame of this sequence and deform that mesh from frame-to-frame based on feature correspondences between the frames. This can result in large frame-to-frame deformations in the presence of fast motion, which can lead to erroneous correspondences and thus poor alignment. Additionally with the quality of alignment at each frame depending on the result of the previous frame errors can accumulate, sometimes significantly, as the sequence is processed.

Here however we order the frames according to how similar they are into a tree structure, thus reducing the possibility of error in alignment, and producing multiple shorter chains of alignment which result in less accumulation of error. Figure 3 details the steps involved in this process. First, every mesh in the sequence or database of sequences is compared to all other meshes using shape histogram comparison [9]. This quantifies the difference in shape between each of the meshes and provides the data to construct a shape similarity matrix. This matrix is equivalent to a fully connected graph in which each node represents a mesh and each edge the shape difference between connected meshes. The minimum spanning tree of this graph provides a logical tree on which alignment can proceed.

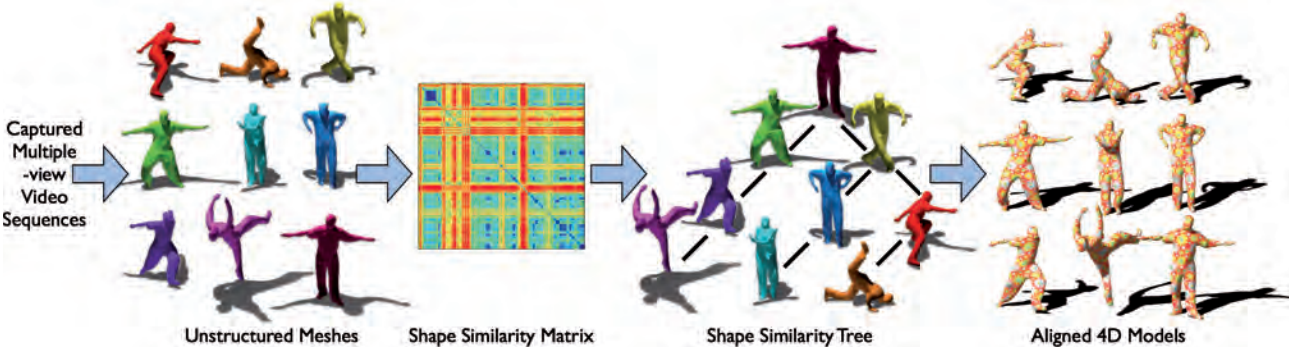


Figure 3: Overview of global temporal mesh sequence alignment

The minimum spanning tree represents a series of paths of alignment over which the sum difference in shape is minimised. Alignment now proceeds from the frame at the root of this tree down the branches. Each frame-to-frame step now represents the smallest possible change in shape and by virtue of the tree structure the multiple shorter paths of alignment result in less error accumulation. Additionally, by the nature of the minimum spanning tree, all larger alignment steps are forced towards the leaves of the tree. These larger and thus more error prone steps therefore affect few subsequent alignment stages; again reducing accumulation of error. Alignment along the branches of the shape similarity tree makes use of techniques developed by Budd and Hilton [2].

5 RESULTS

Character reconstruction and subsequent temporal alignment is demonstrated on the JP-Street Dancer database which was created and made publicly available as part of the SurfCap project [20]. The top line of images in figure 4 shows a selection of frames from the 1800 frame JP database of motions which were reconstructed using the approach described in section 3. The reconstructed surfaces yield an accurate representation of the character with a high level of surface detail. The bottom line of images show the results of temporally aligning the reconstructed frames using the technique described in section 4. The patched pattern rendered on the surface of the mesh is produced by texturing the first frame of the sequence and projecting that texture through the sequence using the temporal consistency of the mesh. The patches remain stable and consistently placed throughout the aligned database whilst the shape matches the original reconstructed meshes with an RMS Error of approximately 0.5mm.

The non-sequential alignment technique described in section 4 can also be applied to video sequences of facial animation. With facial data shape similarity as defined by the shape histogram is not suitable for quantifying inter-frame differences. Instead similarity is assessed based on a sparse set of points tracked using an LP tracker. These tracked points yield a sparse 3D point cloud representing a sparse estimate of the facial expression at each frame. The difference in expression is quantified as the mean Euclidean distance between corresponding points in the cloud after rigid alignment. Rigidly aligning the point clouds effectively discards head pose and considers only the facial expression. With this shape information a shape similarity tree can be produced for facial animation. Alignment along the branches of the shape tree proceeds using the work of the Klaudiny et al [12]. Figure 4 shows the result of applying this approach to a 346 frame publicly available dataset [1]. Again the patterned face presents stable texture throughout the sequence whilst the mesh accurately follows the original facial expressions.

6 CONCLUSIONS & FUTURE WORK

In this paper we have presented a pipeline for capture, reconstruction and temporal alignment of multiple view video and facial capture. Based on a combination of visual hull reconstruction and stereo refinement the presented techniques are capable of producing accurate frame-by-frame models of the captured character. These models are frame dependent representations of the character with time varying topology. They lack the temporal consistency required to make mesh editing, relighting, texture editing and other post production tasks less laborious. Subsequently we presented a technique for temporal alignment of these meshes demonstrating the capability to fabricate a sequence of meshes with consistent topology. We prove this technique to yield a sequence of meshes which not only accurately model the characters of the scene but show a high degree of temporal correspondence with low levels of surface drift.

Additionally, we demonstrate the use of variations of these techniques to construct and track facial capture. These variations are capable of producing temporally consistent facial models which again accurately represent the physical nature of the characters facial expression whilst maintaining low levels of drift in tracking throughout the sequence. The non-sequential nature of the these alignment techniques allow accurate alignment of not only very long single sequences but entire databases of capture data. Mesh and texture edits can thus be automatically propagated, not only within a sequence, but database-wide.

This methodology for reconstruction and tracking surface deformations will be core in achieving our goal of bridging the gap between sample-based and model-based repre-

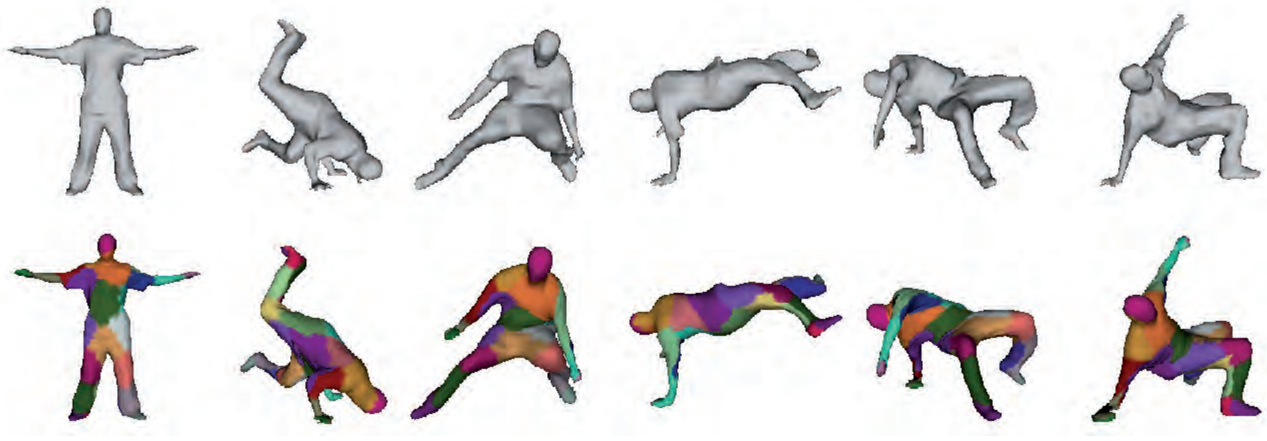
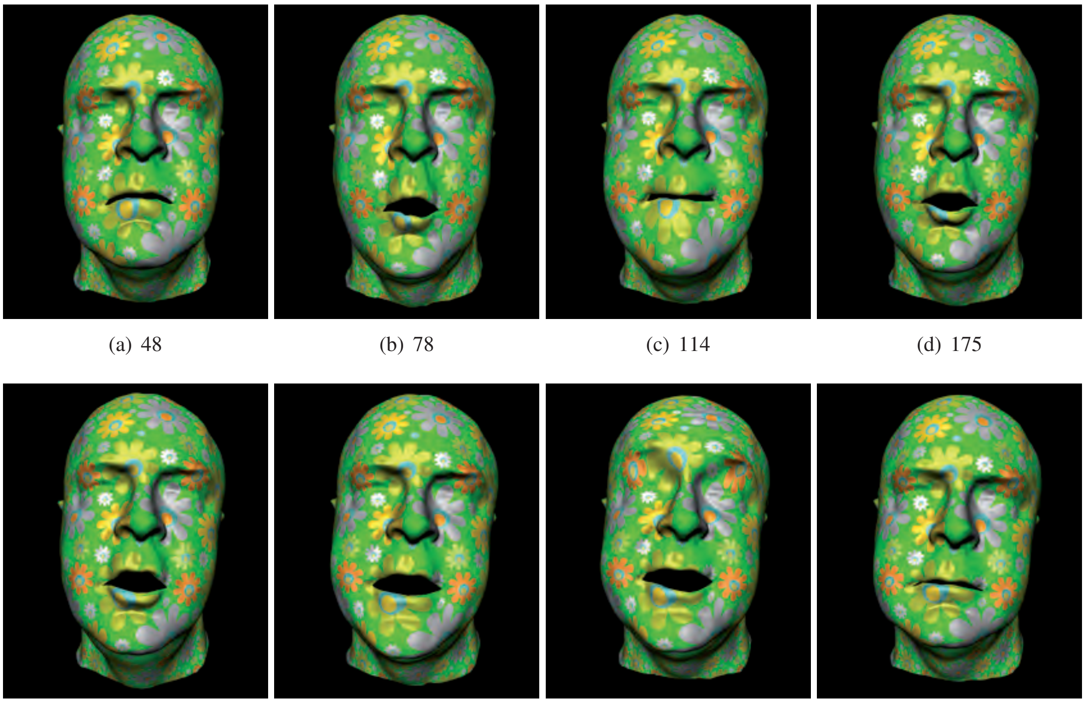


Figure 4: Non-sequential Full Body Alignment of a Street Dancer Motion Database



(a) 48

(b) 78

(c) 114

(d) 175

(e) 216

(f) 258

(g) 333

(h) 394

Figure 5: Non-sequential Facial Capture Alignment

sentations. Although excellent results have been demonstrated in the case of studio face or body capture, further work is required before being able to model more general scenes containing multiple dynamic non-rigid objects possibly captured in a natural environment. This poses major challenges in terms of robustness as algorithms must be able to operate in environments with limited control for which they have not normally been tailored. To give an example, segmentation (a first step for most reconstruction pipelines) in a natural environment is usually ill-posed as different objects may have overlapping colour distributions. In terms of reconstruction, weakly textured surfaces cannot be reliably reconstructed or tracked.

The technology currently being developed will address these shortcomings at both hardware and software levels. Firstly, it will capitalise on recent advances in depth sensing technology to provide a richer input necessary for processing of complex scenes. More specifically it will investigate how fusion of one or multiple low resolution depth sensors with a single or multiple view camera system can be used to enhance capture capabilities. The additional depth cue will facilitate processing of general scenes by reducing ambiguities and improving accuracy at all stages of the pipeline. This will be used to guide the spatio-temporal refinement of the scene geometry by providing additional constraints complementary to those pertaining to the image domain.

Finally, we will extend the dense surface tracking framework by combining the high quality image feature tracking developed for facial data with the geometry based tracking used with full body. This combination will provide stronger constraints required for robust processing of general scenes which may contains holes caused by occlusions or have weak texture information. It will also provide a basis with which to track surfaces reconstructed from 2.5D data. This has potential to simplify data acquisition by using a smaller number of cameras augmented with low resolution depth sensors as opposed to a more expensive multi-camera studio system.

7 ACKNOWLEDGEMENT

This work is currently supported by the EU FP7 project SCENE.

References

[1] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (SIGGRAPH 2011)*, 30(4), 2011.

[2] C. Budd and A. Hilton. Temporal Alignment of 3D Video Sequences Using Shape and Appearance. *Proc. CVMP*, pages 114–122, Nov. 2010.

[3] C. Budd, P. Huang, and A. Hilton. Hierarchical Shape Matching for Temporally Consistent 3D Video. *Proc. 3DIMPVT*, pages 172–179, May 2011.

[4] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Global Non-Rigid Alignment of Surface Sequences. *International Journal of Computer Vision*, 2012 (in press).

[5] C. Cagniart, E. Boyer, and S. Ilic. Iterative mesh deformation for dense surface tracking. *Proc. ICCV Workshops*, pages 1465–1472, Sept. 2009.

[6] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. *Proc. ECCV*, pages 326–339, 2010.

[7] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics*, 27(3):1, Aug. 2008.

[8] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. *Proc. CVPR*, pages 1746–1753, June 2009.

[9] P. Huang. Shape-colour histograms for matching 3d video sequences. *Computer Vision Workshops (ICCV Workshops)*, pages 1510–1517, Sept. 2009.

[10] P. Huang, C. Budd, and A. Hilton. Global temporal registration of multiple non-rigid surface sequences. *Proc. CVPR*, pages 3473–3480, June 2011.

[11] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997.

[12] M. Klaudiny and A. Hilton. Cooperative patch-based 3D surface tracking. *Proc. CVMP*, page 2011, 2011.

[13] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[14] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2), 1994.

[15] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(4), 1993.

[16] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *Proc. ICCV*, pages 492–499, 1998.

[17] P. Sand and L. McMillan. Continuous capture of skin deformation. *ACM Transactions on Graphics*, pages 578–586, 2003.

[18] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):141–173, 1999.

[19] S. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. *Proc. ICCV*, pages 349–356 Vol. 1, 2005.

[20] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[21] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-p. Seidel. Animation Cartography - Intrinsic Reconstruction of Shape and Motion. *ACM Transaction on Graphics*, 31(2), 2011.

[22] T. Tung and T. Matsuyama. Dynamic surface matching by geodesic mapping for 3d animation transfer. *Proc. CVPR*, pages 1402–1409, June 2010.

[23] K. Varanasi, A. Zaharescu, and E. Boyer. Temporal surface tracking using mesh evolution. *Proc. ECCV*, pages 563–576, 2008.

[24] D. Vlasic, I. Baran, and W. Matusik. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics (SIGGRAPH 2008)*, 27(3):1, Aug. 2008.

[25] G. Vogiatzis, C. Hern, P. H. S. Torr, and R. Cipolla. Multi-view Stereo via Volumetric Graph-cuts and Occlusion Robust Photo-Consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):1–15, 2007.

Enhancing Viewer Engagement Using Biomechanical Analysis of Sport

Robert Dawes[1], Bruce Weir[2], Chris Pike[2], Paul Golds[2], Mark Mann[2], Martin Nicholson[1]

[1]BBC Research & Development, London, UK; [2]BBC Research & Development, Salford, UK

E-mail: <firstname.lastname>@bbc.co.uk

Abstract: **The audience for television sport have high expectations in the analysis that forms part of that coverage. In a competitive broadcasting environment there is always a need to develop new features to engage the audience. This is particularly true of sports with relatively small audiences such as athletics that will only get large viewing figures during occasional big events such as the Olympics. In this paper we describe the results of our recent work in the field of biomechanics. This field of science is a key part of the training regime of almost all athletes and sportsmen and women. By making use of the tools and techniques of this field we have developed systems for both the next generation of television analysis systems and for distribution via the web to put the tools in the hands of the audience. These tools aim to offer a new level of insight and explanation to the audience – including those viewers who may rarely watch the sports in question – and so increase their engagement with the coverage. The web tool illustrates some of the possibilities that new forms of digital media content offer the viewer for direct interaction with video.**

Keywords: sport, analysis, biomechanics, Flash, augmented reality, image processing, computer vision

1 INTRODUCTION

One of the challenges of ensuring that sports coverage is as engaging and involving for the audience as possible is the need to help them understand the sport they are watching. This is particularly true of sports where most of the audience will have limited experience of participating in the sport to any level of quality, and so may not be able to appreciate the level of skill or ability they are watching. One method of addressing this problem is to use a co-commentator or pundit - commonly a former professional from the sport in question - who uses their expertise and experience to explain the situation to the viewer. In a team sport this will usually consist of an explanation of the tactics in use, but for more individual sports such as athletics the explanation is more personal and may concentrate on the specific actions of an athlete, explaining the technique or effort required to perform them.

It is this sort of explanation that our work aims to aid, by producing tools to help the pundit to explain the actions of athletes and to help the viewer relate to them.

Pundits are often provided with tools they can use to annotate the action, drawing on the video to illustrate the point they are making. These tools may produce simple flat drawings on the screen or use sophisticated systems to ensure that the drawings appear in the correct perspective, as if painted onto the field of play. However, we are looking beyond this passive annotation and developing tools that actively analyse the scene and extract or help to extract information about the performance.

We are also looking to give the viewer access to the tools and techniques currently only available to the studio pundit. By delivering video and data via the web we can create applications where the viewer can interact with the sport and give themselves a more involved experience of the event.

2 BACKGROUND

To know what is useful to extract from athletic sequences we have investigated the tools and techniques used by the athletes and their coaches and trainers. Increasingly athletes make use of sports scientists to help improve their performance and many of those scientists work in the field of biomechanics. Biomechanics is the application of mechanical principles to living organisms, examining the internal and external forces acting on them and the effects produced by these forces. It is a large and varied scientific field that combines the disciplines of biology and engineering mechanics and utilises the tools of physics, mathematics, and engineering to study everything from the molecular level up to the effect of gravity on entire skeletons.

Within classical mechanics there are two related fields, kinematics and kinetics:

Kinematics – The study of bodies in motion without regard for the causes of motion.

Kinetics – The study of the causes of motion.

Kinematics observes the quantities of motion such as position, velocity and acceleration both linear and angular, such as the angles of joints and the acceleration of a limb. Kinetics studies forces and moments of force and their characteristics such as work, energy, power and momentum.

Analyses of mechanical systems can be split into two categories, forward and inverse dynamics:

Forward Dynamics – prediction of the motion of bodies (kinematics) from forces and moments of force (kinetics).

Inverse Dynamics – prediction of forces and moments of force (kinetics) from the motion of bodies (kinematics) and their inertial properties.

Corresponding author: Robert Dawes, BBC R&D, BBC Centre House, 56 Wood Lane, London, W12 7SB, +44 (0) 3030 409703, robert.dawes@bbc.co.uk

In biomechanics, forward dynamics is often concerned with simulation of movements using a sequence of muscle actions as an input into a musculoskeletal model. The modelling is generally verified by comparison with a recorded real movement. Once a satisfactory correlation with the real world has been obtained, numerical methods can be utilised to search for variations in the sequence of actions that can obtain better sporting results. It can be used to discover what is within the range of human ability and verify our opinions on how movements are achieved. As a very simple example biomechanists have built mathematical models to simulate the 100 metres sprint, profiling how an athlete accelerates, reaches a top speed and attempts to maintain it. These are verified by comparison to real races were split times have been recorded at regular intervals along the course. The parameters of the model can be altered so it fits the race profile of the athlete. The parameters can then be manipulated to see which aspects of his or her race the athlete should concentrate on in order to improve his or her performance.

Inverse dynamics will normally involve measurement of movement using, for example, a marker based motion capture system, which is combined with the inertial properties of the bodies, to calculate the internal forces and powers, and direct measurement of external forces, such as ground reaction force which is measured using a force plate. These measurements are solved using a regression function to obtain values for the forces and moments involved in the system. This allows high-level biomechanical analysis of the real-world movement. An example of these techniques might be to use the motion capture system to record a long jumper taking off from a force plate. All this data can be used to model the forces and exact positions of the athlete's body parts allowing for a great deal of further analysis.

While much of this data will only be of interest to the athletes and coaches there is still a wealth of information that might be interesting to viewers at home to help them understand the events they are watching. For example, measuring the stride frequency and length of athletes in a 100 metre sprint can demonstrate what type of athlete they are: a tall long legged athlete who takes few strides and is slow to reach top speed or a smaller, quicker paced athlete who can get more strides in, but lacks the higher top speed, or perhaps somewhere in between. In the long and triple jumps the trajectory of the athlete can be modelled as a projectile, using the centre of mass as the location. The centre of mass begins at around waist height, then, as the athlete tucks in around it, ends up at ground level as he or she lands in the pit. Because the landing height is lower than that at take-off, the optimal take-off angle is less than 45 degrees, so they are able to retain more of their horizontal momentum. We can examine the take-off angle of an athlete to see how near they are to achieving their particular optimum angle.

The tools used by coaches and athletes to extract and record data of this sort typically make use of sensors or markers placed on the athlete or in the environment. However, we wish to analyse competitive events where such methods cannot be used easily because they interfere with the proceedings. This, combined with the logistical difficulties that come from working with sporting events that might be taking place all over the globe, means that we are effectively restricted to just working with the broadcast video of an event.

Some existing tools can work entirely with images. Examples include Dartfish's products which are used in both the sports science and broadcasting spheres [1]. However, when used with just broadcast video they are generally restricted to producing solely visual effects, while we hope to make use of and gain knowledge about the scene.

3 PREVIOUS WORK

We have previously developed tools for augmenting real scenes with annotations that appear to be "painted" into the pitch or arena [2]. For football, rugby and other team sports with reasonably standard pitch markings our existing system uses these markings with known real world positions to determine the position and pose of the camera. It then tracks the movement of the lines to determine how the camera was moving. As the camera moves the graphics are moved such that they appear to be fixed to the same part of the real world. More recent work has developed the system further such that is can track arbitrary points rather than just lines [3]. This allows for similar graphical effects but it can be applied in a greater number of environments, including less regular environments such as athletics grounds.

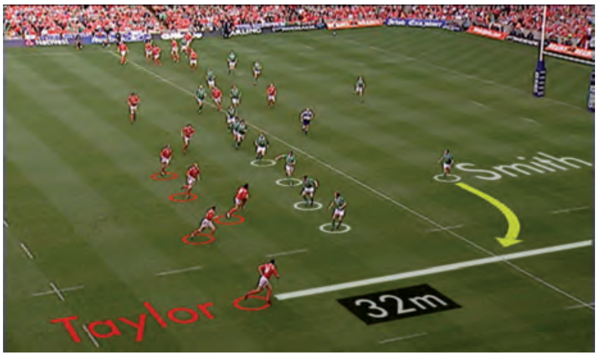


Figure 1: Measurement on a Rugby Field

It is a natural extension of this work to try and gain more information about the performance of the athletes rather than just overlay annotations. This extra data can be presented to the viewer to give them another level of information about the event they are watching. Indeed there is a predecessor of sorts in the existing graphics systems. The camera calibration allows measurements between two points of known position, such as two points on a football pitch. This is often used to measure the distance involved in an incident or activity such as how far a player has run or how far from goal a free kick or conversion is being taken. An example of this facility is shown in Figure 1. Our tools aim to extend and develop this idea.

4 ANALYSIS TOOLS

We have developed a series of standalone tools to analyse different aspects of athletic or sporting performance. These tools operate on video or image sequences, processing them to extract additional data from the scene. This data can then be presented to the viewer to offer insights into the event they are watching.

4.1 Calibration

Some of these tools require information about the position and pose of the camera – much like the graphics drawing tools described above. In order to get hold of this calibration information we first process the video sequences to calculate where the camera is and how it moves. This process can be performed using live video or offline from file. Unlike the graphics drawing tools these analysis tools are mostly not intended to be used live so the process described here is the offline version. However the live video process is very similar.

The video sequence is treated as a sequence of separate images, each of which will be accompanied by a description of the camera position and pose. The system is first calibrated on a single frame. This process requires the coordinates of known points in the scene to be identified manually. With this information the position and initial pose of the camera is computed using an iterative optimisation process to minimize the squared reprojection error of the annotated locations into the image, following the approach in [2].

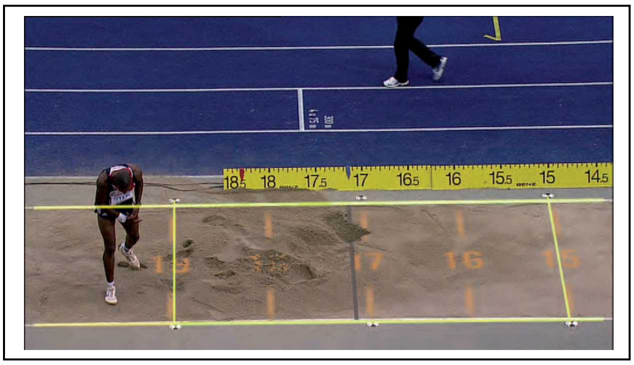


Figure 2: Known positions marked on an image

In Figure 2 known real world locations have been annotated and then highlighted in yellow. In this case these are the far and near edges of the pit and the 15m and 19m lines.

Once calibrated on a single image, a KLT-based tracker is used to track areas of rich texture from frame to frame. The camera position is assumed to be stationary and the movement of the texture patches is used to determine the changing pose of the camera throughout the sequence. This produces camera pose and position data for all of the images. The images and accompanying data can then be used by the other tools.

4.2 Stride Detection

We have developed a tool that can extract the positions in the real world of the feet of a running athlete. This can be used to automatically extract the stride frequency and length of a runner or the positions of the first two phases of a triple jump.

It uses a motion compensated temporal median filter to build up a background image for the scene with the athlete removed. We make use of the previously extracted frame-by-frame camera calibration data in order to perform the motion compensation.

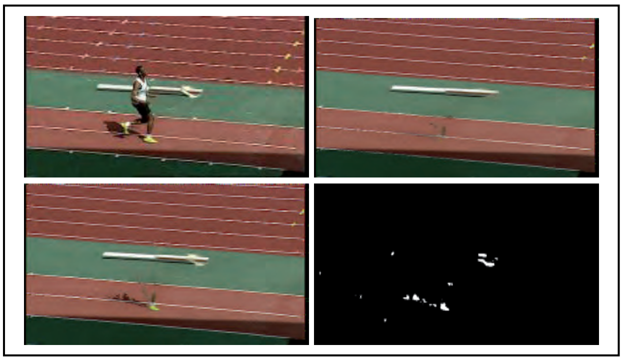


Figure 3: Original image (top left), background image (top right), background with visible foot (bottom left), difference between the two backgrounds (bottom right)

A second background image is then generated using a filter with a smaller temporal window. The size is chosen such that the temporarily stationary feet of the athlete "burn" into the background. A difference is then taken between the two background images resulting in a mask of possible locations for the foot. The calibration is used to find where these possible locations occur in real world 3D coordinates (making the assumption that stationary feet lie on the ground) and the most likely option is chosen as the foot. If there was no stationary foot in that frame then there will be no suitable candidate in the mask. This process is conducted over the whole sequence and a series of footsteps are extracted. This data is then available for further analysis or for presenting to the viewer.

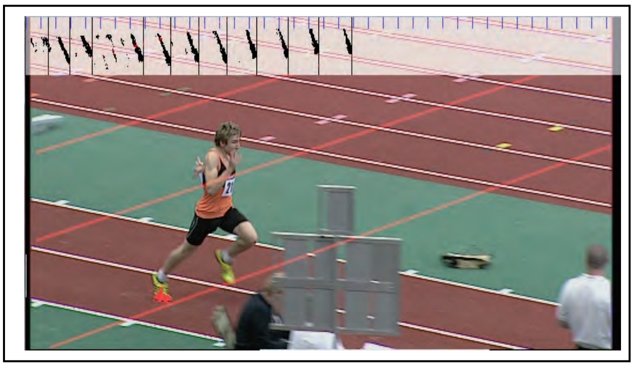


Figure 4: Step positions annotated onto the video sequence

4.3 Body Modelling

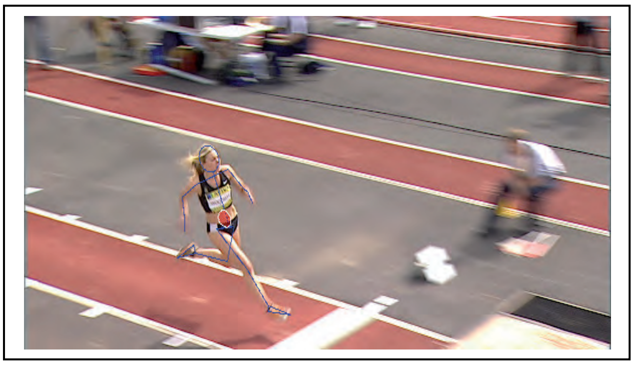


Figure 5: Athlete with body parts marked and derived centre of mass

In several events, particularly sprints and jumps, the athlete will only move in a single plane as they run down the track. We can combine this assumption with the camera calibration information to work out body positions in three dimensions. An operator can hand annotate a video sequence of an athlete, labelling the 2D positions of the body parts – a process known as "digitisation" by sports scientists. We can then calculate a line of sight from the camera towards this point and discover where it intersects the known plane of motion. For example, we may assume an athlete is running down the centre of his or her lane and that the head, neck and base of the spine can all be found along this plane while the limbs are in planes offset to the left and right accordingly. This plane gives one dimension, while the point of intersection provides us with the other two. Once we have positions for all the body parts we can display them within a virtual environment or we can place them into a mathematical model of the body to try and extract extra information such as the centres of mass of the various body parts and the whole body. The pundit can then make use of this information in his or her analysis or the data could be presented directly to the viewer.

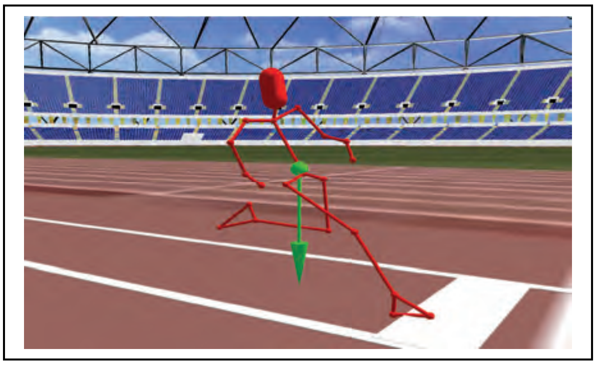


Figure 6: Visualisation of body positions in a 3D environment

4.4 Cadence Detection

This tool detects the position of a bicycle in a scene by using a Hough transform to locate the wheels. It then looks for the position of the cyclist's feet – segmenting them from the background using their colour - and tracks them as he or she pedals. From this information a value for cadence (i.e. the pedalling speed) can be extracted. This can inform the pundit or viewer about when a cyclist is accelerating, or in combination with the gear ratio how much power is being produced.

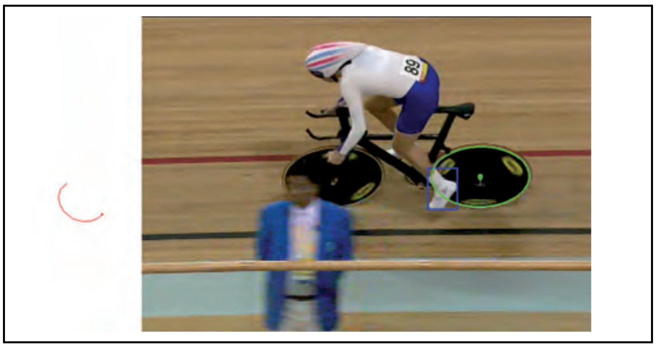


Figure 7: Annotated bicycle wheel and foot with extracted movement.

4.5 Diving

In the sport of diving points are awarded by a panel of judges. To the unskilled eye of the viewer it may often be unclear why one dive scored better than another. As the diver enters the water he or she aims to be as upright as possible and to minimise the splash which would result in a non straight and vertical entry:

"The entry into the water shall in all cases be vertical, not twisted, with the body straight, the feet together, and the toes pointed."[4]

The system measures the size of the splash and the angle of entry. It segments the largely white splash from the largely blue background and then measures the size of the resulting mask in order to get a figure for the splash. The angle is detected by segmenting the diver from the blue background and then fitting a line down the length of the extracted object.

These measurements give viewers some insight into why a dive might receive the score it does and offers them the opportunity to compare one dive with another. It is useful to have the angle value for a few frames as the diver enters the water. This can help communicate the speed and intricacy of motion involved in a dive.

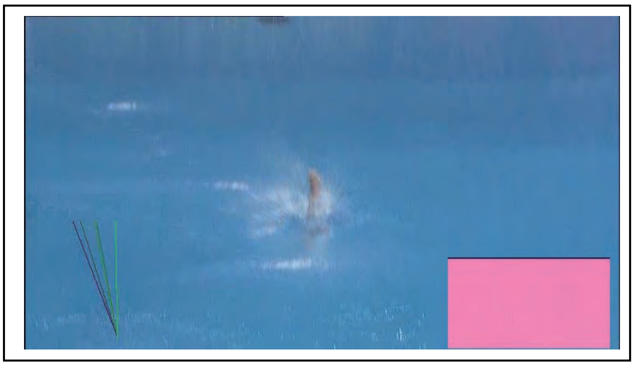


Figure 8: Diver entering the water with visualisation of the splash size (height of bar on the right) and angle of entry of the last 4 frames.

5 WEB BASED AUGMENTED REALITY

5.1 Overview

We have developed a Flash application to allow the viewer to interact with footage of sporting events and help them to get more involved in the action. By rendering a 3D scene on top of a background video sequence, the real footage can be 'augmented' with virtual objects in a similar manner to the broadcast graphics tools mentioned above. However, because the rendering is performed client-side by the Flash plug-in, as shown in Figure 9, we can offer an engaging interactive experience to the viewer where they can affect the augmented graphics themselves.

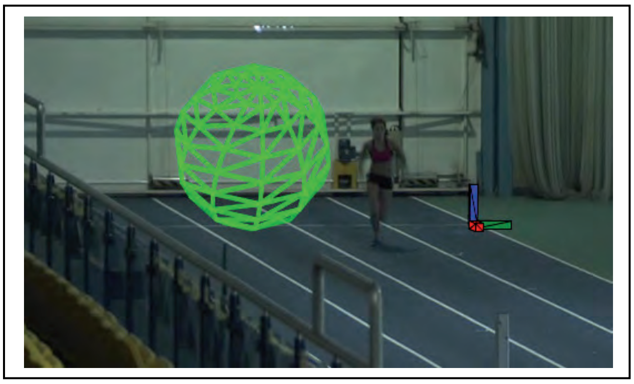


Figure 9: Virtual 3D objects added to a real scene at the client side. During early tests a rolling wireframe sphere was used to represent the virtual athlete.

5.2 Augmenting Graphics

The web application for client-side interaction with the biomechanics data was built using Flash and the Away3D ActionScript library[5]. Away3D is an open-source 3D graphics library and we use it to render a 3D scene on top of a background video sequence. Since we are trying to insert virtual objects into the scene so that they look like that are present in the real environment, the virtual elements must not drift relative to the real objects visible in the background video. In order to make this work, the frame-by-frame, camera-pose calibration data described above must be made available to the 3D renderer as each video frame is updated.

Some sequences are less suitable for the tracking process used to generate the calibration data than others, for example where there are very few distinct background features visible. The resulting camera data can sometimes have faults where the virtual camera wobbles or moves sharply. This is significant problem with a graphics drawing system such as this application. However, the offline nature of the process means there is significant time to make adjustments to cover the faults - a process that is not possible when tracking is used live. One method of fixing faults is to interpolate over bad frames between two known good frames.

The video sequences were converted into FLV format [6] with the video encoding undertaken by the x264 library[8]. The camera pose calibration data was embedded into the resulting file as frame-by-frame 'ScriptData' tags [7], time-stamped for 'presentation' with the same timestamp as the video frame to which they correspond. This camera pose data is made available to the Flash application during video playback via a handler method which is triggered whenever a ScriptData tag is encountered. The handler extracts the embedded camera pose data and uses it to control the pose of an Away3D Camera3D object modified to generate the correct transform for converting the 3D model coordinates into the screen coordinates when provided with a camera pose, field of view and aspect ratio. This method also initiates the render pass which ensures that the video updates and 3D overlay updates happen simultaneously. The 3D model elements are positioned in the scene graph such that they are rendered after the video image, this means that they will always appear in the foreground. In this version of the application, there is no alpha masking of the video, so virtual objects cannot appear behind objects in the video.

5.3 Final Application

The application we have developed using this approach allows users to compare their own sporting performance against that of professional athletes in a novel way. The sports included in the application are the 100m sprint, long jump, high jump and triple jump. The user enters details of their own sporting performance, such as their best long jump, or fastest 100m sprint then the video footage of the sporting event is augmented with an avatar of the user competing in the same event. Examples can be seen in Figure 10 and Figure 11. Other athletics events with a simple performance metric (time, distance, etc.) would be straightforward to add, but events such as synchronised swimming or beach volleyball would be much harder to simulate.

The user can first personalise the experience by entering their height and weight. If they wish they can also use a webcam to take an image of their face which is then texture mapped onto the avatar, or they can select the face of a sporting celebrity to represent them. The body of the avatar is textured with the colours from the 2012 British Olympic team.

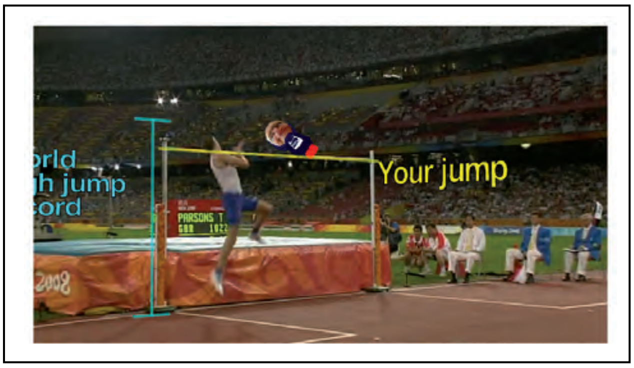


Figure 10: The user clears the bar in the high jump, matching the women's world record.

A degree of bespoke animation is required to integrate the avatar with the event. For example, the location of the bar in the high jump event needs to be known in the coordinate system of the foreground 3D model so that the avatar's run up and jump can be correctly positioned, or the orientation of the camera pose with respect to the 100m running track needs to be understood to make sure that the calculated motion vector of the avatar matches the direction of the other competitors in the race. The video footage is also augmented with markers showing world record distances, or the dimensions of well-known real-world objects such as double-decker buses. As well as seeing themselves compete, these real-world objects help the user relate to what they are seeing.

In order to ensure realistic positioning over the course of the whole race the movement of the avatar in the 100m sprint is controlled by a model based on Tibshirani's extension of the Hill-Keller model [9]. The equations of motion are

$$D(t) = kt - \frac{1}{2}c\tau t^2 + \tau k(e^{-t/\tau} - 1)$$

where $k = f\tau + \tau^2 c$

In this model f represents the acceleration force of the athlete. We calculate this by taking the user's chosen finish time for the 100m and make assumptions for c and τ , respectively representing the athlete's muscular endurance and a broader measure of flexibility, leg turnover rate, anaerobic response etc [10]. Our assumptions use a combination of typical figures for professional athletes [9] with some variation based on the personal attributes of the user entered earlier on. For example, a taller, heavier athlete gets a higher τ value, which gives him or her lower acceleration. c varies depending on the chosen finish time for the sprint, the longer the sprint took the lower the value, implying less energy is being drained due to a lower running speed.

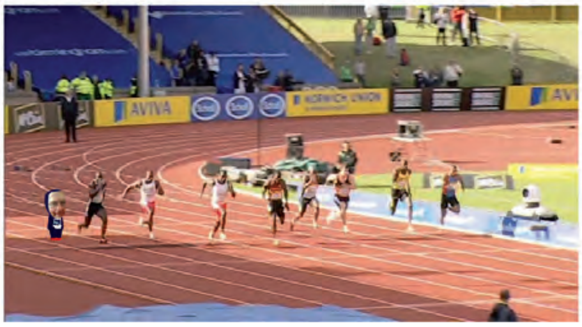


Figure 11: A personalised avatar taking part in a 100m race

Once f has been calculated the equations of motion can be used to find the location of the user's avatar throughout the race. We can also provide details of his or her velocity as the race progresses. This is provided in miles per hour, a measure of speed that viewers will be most familiar with, helping them to relate to and understand what they are watching. While this model is of course not a completely accurate representation of how the user might perform, it is at least indicative of relative performance and helps to demonstrate to the user the level of ability inherent in professional competition.

6 CONCLUSIONS

With relatively simple tools we are able to present to the viewer an extra level of detail in the events they are watching. For those with a particular interest in a sport this gives them the extra detail that the current coverage may lack. The extra analysis may also give a more mainstream viewer an insight into a sport that has never occurred to them before. This may well encourage a greater interest in a particular sport and engagement in its coverage. This is particularly true of the web application that offers a fun and accessible way to learn about a sport and to look at it in a new way.

In addition the augmented reality application described here only begins to scratch the surface of the possibilities offered by delivering analysis tools via the web. Tools that were previously only the domain of the television pundit may soon be put in the hands of the viewers. This development can be seen as part of the wider trend of giving the power to the audience. They expect to be able to choose, interact and play around with the media they consume and this application offers them just that.

References

[1] Dartfish Sports Enhancements http://www.dartfish.com/en/sports-enhancements/sport_performance_software/index.htm

[2] Thomas, G.A. Real-Time Camera Tracking using Sports Pitch Markings. Journal of Real Time Image Processing, Vol. 2, No. 2-3, November 2007, pp. 117-132. Available as BBC R&D White Paper 168. http://www.bbc.co.uk/rd/publications/whitepaper168.shtml

[3] Dawes, R., Chandaria, J., Thomas, G.A. Image-based Camera Tracking for Athletics. Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB 2009), Bilbao, May 13-15 2009. Available as BBC R&D White Paper 181 http://www.bbc.co.uk/rd/publications/whitepaper181.shtmlds

[4] Judging section of the Diving Rules. FINA. http://www.fina.org/

[5] Away3D. http://www.away3d.com

[6] Adobe Flash Video File Format Specification Version 10.1. http://download.macromedia.com/f4v/video_file_format_spec_v10_1.pd f August 2010.

[7] Ibid. p74

[8] VideoLan x264. http://www.videolan.org/developers/x264.html

[9] Tibshirani, R. "Who is the fastest man in the world?" The American Statistician. Vol. 51, No. 2 pp. 106-111. May 1997.

[10] Mureika, J.R. A Simple Model for Predicting Sprint Race Times Accounting for Energy Loss on the Curve. Canadian Journal of Physics, 75: 837–851. August 1997

Foundations of a New Interaction Paradigm for Immersive 3D Multimedia

I. Galloso[1], F.P. Luque-Oostrom[1], L. Piovano[1], D. Garrido[1], E. Sánchez[1], C. Feijóo[1]

[1]Center for Smart Environments and Energy Efficiency (CEDINT), Technical University of Madrid, Madrid, Spain

E-mail: [iris, franluque, lpiovano, dgarrido, esanchez, cfeijoo]@cedint.upm.es

Abstract: **Immersion and interaction have been identified as key factors influencing the quality of experience in stereoscopic video systems. The work presented here aims to create a new paradigm for 3D Multimedia consumption exploiting these factors in order to increase user involvement. We use a 5-sided CAVE^{TM} environment to support 3D panoramic video reproduction, real-time insertion of synthetic objects into the three-dimensional scene and real-time user interaction with the inserted elements. In this paper we describe our system requirements, functionalities, conceptual design and preliminary implementation results emphasizing the most relevant challenges accomplished. The focus is on three main issues: the generation of stereoscopic video panoramas; the synchronous reproduction of immersive 3D video across multiple screens; and, the real-time insertion algorithm implemented for the integration of synthetic objects into the stereoscopic video. These results have been successfully integrated into the graphic engine managing the operation of the CAVE^{TM} infrastructure.**

Keywords: interactive and immersive 3D multimedia, real-time insertion algorithm, 3D video panorama

1 INTRODUCTION

Recent trends in the field of multimedia systems focus on 3D technologies. The industrial interest for them is constantly growing up, but only the adoption of a truly user-centered approach is seen by experts as the way to guarantee the ultimate success of these systems ([1]). With this goal in mind, interactivity and immersion have been identified as two of the most relevant factors influencing the quality of experience in stereoscopic systems, since the very early beginning of VR science (see, for instance, [2-5]). In particular, the actual society is devoting a lot of attention to the interactivity paradigm as one of the most promising ways to improve the technological and social impact of any new innovative media. Despite these huge expectations, the definition of a technical standard is still missing. This drives researchers, communication engineers, broadcasters and 3D content producers to experiment new setups, protocols and architectures in order to better define the potentialities of this feature. Anyway, this technology is reserving a centric role to the user, who through interactivity may take a more active control over the multimedia content progress. This is explicitly in contrast with more traditional and popular media (e.g., TV and cinemas) because of the constant bi-directional flow of information. Therefore, content and point of view selection, tri-dimensionality and low latency in the interaction are key features in defining a media interaction system.

On the other side, an immersive environment could help the user to better exploit the interaction capabilities. In one of its dimension, immersion is a perception mechanism dealing with the user's mental state of feeling the relationship between the self-awareness and the surrounding environment ([6]). Different categories of immersion exist ([7]). Without a loss of generality in the following of this paper we refer to the one defined as *a psychological state characterized by perceiving oneself to be enveloped by, included in and interacting with a virtual environment* ([8]). A full immersion is achieved when all the senses are engaged. The most common techniques allowing such an effect involve fooling only three senses: sight (by different stereoscopic techniques), hearing (by stereo audio systems) and the sense of touch (by using haptic devices). Nowadays, CAVE^{TM} infrastructures ([9, 10]) are among the most effective environments enabling high immersive experiences. They are made up with a variable number of large projection screens (*walls*) - usually from 2 up to 5 -, arranged according to different geometries, from rows to one-side opened cubes. In this theatre, high resolution projectors display 3D content on the walls which users could perceive through particular stereoscopic glasses. User's movements around this space are tracked in real time by infra-red cameras and used to adapt the subjective point of view of the scene for each wall and without any appreciable distortion. Moreover, speakers placed at different corners of this environment provide 3D sounds to complement the projected 3D content. Therefore, the main feature of such infrastructure is the higher level of stereo displaying technique with respect to the current commercial TV systems. In order to enrich the degree of immersion, the use of particular remote controllers introduces the added value of the interaction phase between final users and the displayed content.

The work presented here follows the lead of previous European experiences in this field. Indeed, an interesting research project exploring these issues is FascinatE (Format-Agnostic SCript-based INterAcTive Experience, [11]). FascinatE is focused on the development of technologies aimed to support user interaction with and navigation around an ultra-high resolution video panorama showing a live event. Immersive panoramic

Corresponding author: Luca Piovano, CeDInt – Technical University of Madrid, Campus de Montegancedo, 28223 Pozuelo de Alarcón – Madrid, España, tel. +34 91 452 4900 (Ext. 1747), e-mail: lpiovano@cedint.upm.es

displays, among other terminals, will be supported. The MUSCADE Project is another outstanding effort to define, develop, validate and evaluate technological innovations across the entire 3DTV value chain including capture, data representation and rendering ([12]). Its ultimate goal is to define a technically efficient and commercially successful 3DTV broadcast system. One specific objective of MUSCADE is to develop a 3D video interactive application platform. Among other demonstration scenarios, a 3D Interactive Advertisement demonstrator will be implemented in order to promote products that are featured within the A/V stream by complementing the A/V signal with interactive 3D content.

In this paper, we are discussing our contributions to the field of new content production for immersive environments. In particular, the conceptual design and preliminary implementation results of our Immersive and Interactive 3D Multimedia prototype (II-3DM) are described. Section 2 deals with the project objectives and provides the general picture leading to its definition. Section 3 shortly describes the functionalities defined for our system, while the logical architecture of our II-3DM prototype is discussed in Section 4. A description of preliminary results could be found in Section 5. In particular, greater emphasis is given to the description of our immersive environment (I-Space, Section 5.1), depth estimation from stereo videos (Section 5.3) and new 3D content generation and insertion (Sections 5.4 and 5.5). Conclusions and future works are then presented in Section 6.

2 RATIONALE AND OBJECTIVES

A deeper user involvement in the multimedia experience can be enabled through the combination of an enhanced visual immersion and a direct interaction among users and represented 3D objects ([7]). Previous research efforts have been mainly focused on the visualization of and interaction with synthetic environments according to the Virtual Reality paradigm. In such cases, the 3D information is inherently available as part of the scene model. Thus, user's manipulation of objects and scene rendering according to different user's points of view may be computed faster, easier and without distortions in projections. On the other side, little efforts have been reserved on the interaction with traditional audio-visual materials. In general, this is because of an inherent technical limitation: traditional videos present a fixed point of view which cannot be changed while playing them and, at the same time, there is a lack in the full 3D definition of the scene. Nonetheless, under well-defined conditions, it is possible to partially solve those problems. For instance, from a stereo video and by using some stereo matching algorithm, it is possible to retrieve the object spatial displacements. Though this information is not enough to have a complete idea of all the 3D object geometries (it is actually defined as 2D and ½), it could be successfully used to interact with 3D models to be merged into the original streaming. This way, a shift in the paradigm of reference is accomplished, that is we are moving from Virtual to Augmented Reality features.

The work presented in this paper moves towards the aforementioned direction by merging stereoscopic video and interactive 3D content into a 5-sided CAVE^{TM} infrastructure (see Figure 3 for a pictorial representation of it). Given the distinctive configuration of our environment, we decided to deal with stereoscopic videos of panoramic views because they fit better into a cubic CAVE^{TM} and, moreover, the immersion feeling could be experienced at a higher level. Panoramic videos are characterized by a field of view much greater than the one human sight is used to and could result into a 360º horizon arc. Moreover, the field of depth spans from the user's closer surroundings to the horizon, so that each frame usually represents a scene having kilometers in depth. In this kind of stereoscopic videos, tri-dimensionality is much more appreciable for objects laying in close-ups and middle ranges, because perspective laws tend to compress those planes of view in the background. Given this context, the final goal of our II-3DM prototype aims to provide an experimental environment where the influence of interaction and immersion in the user experience could be comprehensively studied. In order to achieve it, the main functionalities to be met by our prototype are as follows:

- Generate immersive (panoramic) video contents to be visualized in a multi-screen projection system.
- Merge interactive computer-generated objects into the stereoscopic video in order to create hybrid contents (i.e. real-time insertion of synthetic objects into the video frame).
- Visualize the generated content in the I-Space environment supporting real-time interaction with the inserted graphics including functions as zoom, point of view selection and manipulation of synthetic objects.

3 SYSTEM REQUIREMENTS

The definition and practical implementation of such a system entails a number of technical requirements whose objectives are to highlight the *desiderata* for each project phase and guarantee the overall quality of the final product. In this section, a short description for the most critical ones among them is provided.

3.1 3D Video Content Acquisition and Post Production

The creation of stereoscopic video panoramas entails a fundamental paradox. While an error-free 2D panorama can only be generated using single viewpoint images (parallax-free) ([13]), stereoscopic images use parallax to provide a 3D representation of the scene [14]. Anyway, as stated in [15], this contradiction could be overcome because the systematic approximation error related to concentric mosaics can be neglected in practice and therefore, a parallax-free stitching of stereoscopic video panoramas (two 2D panoramas of the scene with different perspectives, one for the left and one for the right eye) can be provided. Following that approach, a stereoscopic rig

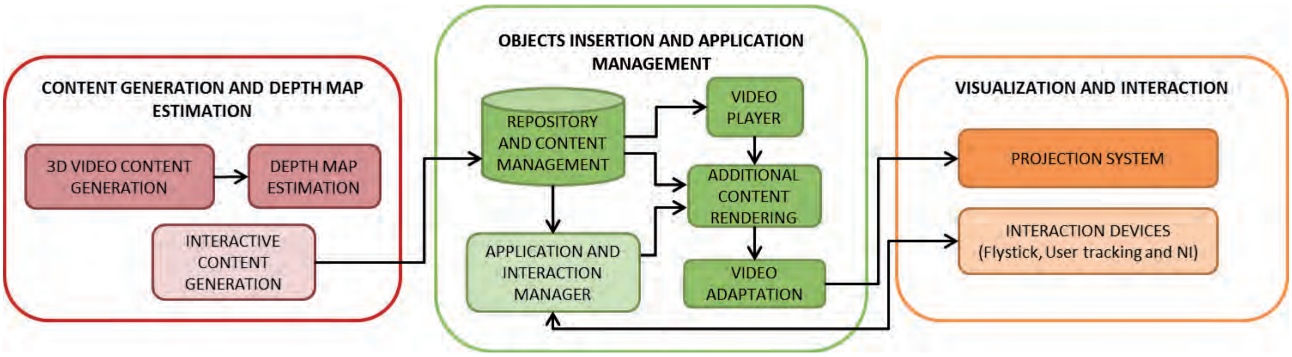

CONTENT GENERATION AND DEPTH MAP ESTIMATION
3D VIDEO CONTENT GENERATION
DEPTH MAP ESTIMATION
INTERACTIVE CONTENT GENERATION
OBJECTS INSERTION AND APPLICATION MANAGEMENT
REPOSITORY AND CONTENT MANAGEMENT
VIDEO PLAYER
ADDITIONAL CONTENT RENDERING
APPLICATION AND INTERACTION MANAGER
VIDEO ADAPTATION
VISUALIZATION AND INTERACTION
PROJECTION SYSTEM
INTERACTION DEVICES (Flystick, User tracking and NI)

Figure 1 - The architectural design for the II-3DM system

with two cameras has been used to obtain the two video sequences corresponding to the Left and Right Eye (LE&RE) points of view (see details in 5.2). In order to avoid visual discomfort, special attention should be paid to the final viewing conditions ([1]).

Instead, the post-production phase mainly deals on editing the acquired material so that it can easily fit the immersive environment requirements for its projection. This broadly means unify single views in order to produce a unique panoramic video. The main tasks to accomplish this goal concern the synchronization of each single view, the definition of a stitching procedure to unify corresponding frames, and the definition of the video technical features being able to maximize the projection quality and minimize the transmission overhead (for instance, bandwidth occupation and transmission delay).

3.2 Merging Synthetic Objects into the 3D Video

This comprises three main steps: first, a stereo matching (or equivalent) algorithm has to be applied in order to obtain the depth map of the scene from the disparity between the stereo images and the camera features; second, the insertion of an object into the scene at the desired depth and position, by properly calculating and representing the corresponding deformations and occlusions; and third, the so computed new stereo pair, in which the merged object appears as part of the scene, should be generated and sent to the stereoscopic player.

As our 3D video content will be generated under real shooting conditions, an eventual depth map estimation algorithm should be able to deal with varying lighting conditions and real world effects (e.g. reflections). Likewise, it is aimed to perform well in fast-varying scenes comprising moving objects, occlusions and variable disparity ranges, etc. and therefore, it should be robust enough to correctly take into account the complexity of a dynamic environment (see 5.3 for details).

On the other hand, given a stereoscopic video and its synchronized depth map sequence, our real-time rendering algorithm should be able to merge synthetic objects into the 3D scene at the desired depth and position. Real-time user interaction with the merged objects (e.g. free viewpoint observation and manipulation) is also a must.

3.3 Visualization and Interaction

Visualizing stereoscopic video panoramas in a multi-screen environment entails significant challenges, mainly related to the required image continuity between adjacent screens, synchronous reproduction of independent views and real-time support of user interaction.

Since the panoramic videos span all the horizontal walls in our I-Space, the six video streams corresponding to the three stereo pairs are to be fed in a synchronous way to the three projectors illuminating the vertical screens. This process can be heavily intensive considering that each video file can be in the order of hundreds of Gigabytes, or even more, depending on the format, duration and quality.

User interaction also stresses the visualization system with rotations, displacements, deformations and size and resolution scaling of the projected image, without detriment of maintaining accurate the synchronization and frame to frame alignment among views. A particularly problematic case takes place, for example, when a user rotation causes the displacement of an ongoing reproduction between two or more screens. To guarantee a continuous and transparent transition, the visualization system not only needs to split and send the exact part of the frame corresponding to each screen but it has to do it synchronously and following the corresponding image deformations and scale adjustments ([16]).

Finally, natural, intuitive and non-intrusive interaction devices and techniques should be used. Likewise, system responses to the actions of the user must be quick and fluid.

4 CONCEPTUAL DESIGN

The logical architecture of our II-3DM prototype is shown in Figure 1.

In the **acquisition** module, the different views to be combined into the stereo panorama are generated and stored with the highest possible quality. Each view comprises two HD video sequences corresponding to the Left and Right Eye, respectively. The output formats can be HD-SDI or 3G-SDI, although a lossless compression method could be applied to fit the available storage capacity.

The main objective of the **postproduction** module is to adapt the acquired video signals to the visualization infrastructure for an optimum user experience. Besides (and usually before) the stitching and time corrections, the

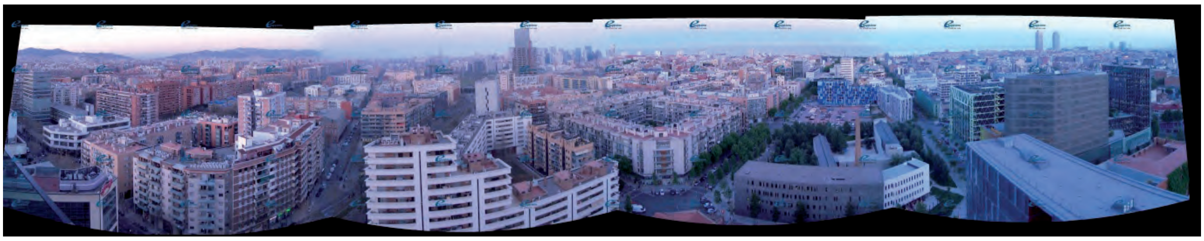


Figure 2 - Left eye frame of a panoramic view of the city of Barcelona. This image has been obtained by combining frames of four original, contiguous views. The geometrical distortion perceived in it comes after this stitching phase and it is due to the effort of matching corresponding points across two views.

input signals need to be rescaled to their original 1920x1080 pixels (the convergence adjustments may cause resolution losses). Additional adjustments including images rotation and vertical/horizontal displacements could also be necessary to alleviate the stitching algorithm load and optimize its results. In this module, the additional interactive content (3D geometry, animations and associated scripts) to be merged into the stereoscopic video is also generated

All the generated content is locally stored at the **repository and content management** module. Video content is accessed frame by frame by the video player to get the stereo pair images and compute depth maps, while additional 3D content can be asynchronously queried by any other component handling synthetic objects and interactivity.

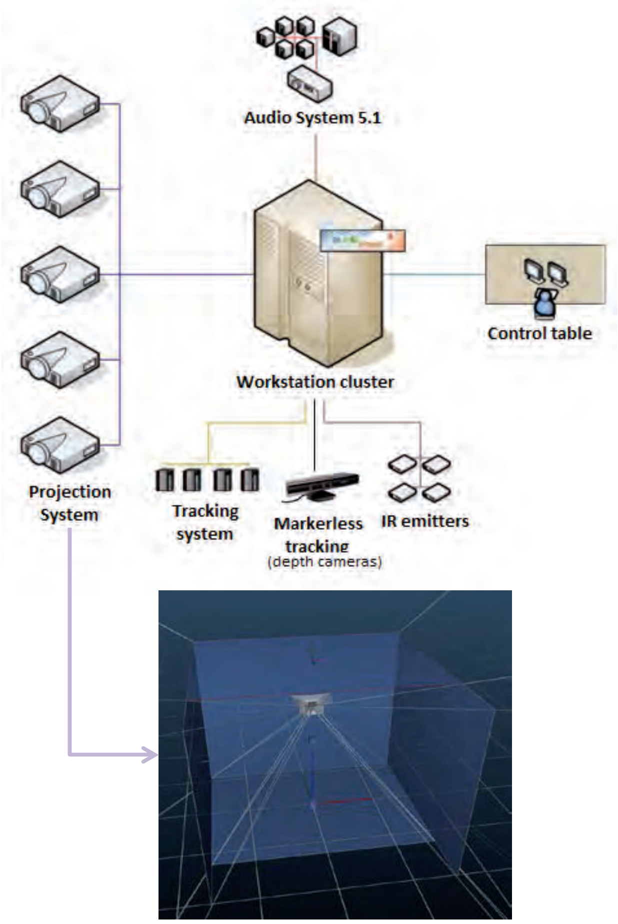


Figure 3 - Scheme for the I-Space environment at CeDInt laboratories.

The **application and interaction management** module executes the applications and animation scripts governing the features, behavior and responses to user actions of the video and additional contents. It also processes the VRPN inputs provided by the interaction devices (Flystick® and tracking system), and redirects the corresponding orders to the additional content rendering and/or video adaptation modules in order to execute the programmed actions.

The **additional content rendering** module merges the additional content (3D models) into the stereoscopic video at the depth, position and orientation indicated by the application or the user. At the output, stereoscopic video frames with the synthetic content embedded are provided with the same format as the input video.

The **visualization** module adapts the stored video content to be synchronously visualized in our I-Space multi-screen infrastructure. It must provide at every frame interval the instantaneous stereo pair corresponding to each projection channel. Therefore, its outputs consist of five stereoscopic video signals (one for each screen) with RGBHV format and SXGA+ (1400X1050) resolution, plus one common synchronization signal. These signals are synchronously projected into the five screens by the projection system.

5 PRELIMINARY RESULTS

5.1 CeDInt CAVE^{TM} Infrastructure

The I-Space environment, in which all the tests for II-3DM project have been conducted, is located into our laboratories at CeDInt center (see Figure 3). The infrastructure comprises:

- *Geometry and screen dimensions*: 5 rear-projection screens (frontal + 2 lateral + ceiling + floor) arranged in a parallelepiped shape. Each screen is 3.20 x 2.40m (4:3 format). Therefore, the viewing distances from the center of the room to the frontal and lateral screens are respectively 1.20m and 1.60m;
- *Projection features*: each screen is projected by a high resolution (SXGA+ 1400x1050) projector 3D BARCO Galaxy 12 HB+ with DLPTM technology, 12000 ANSI lumens of brightness and contrast 16000:1. Each projector is connected to the graphic output of one dedicated workstation;
- *Graphic and storage:* 6 HPxw8600 workstations in cluster configuration (master+5slaves), each equipped with Graphic Card nVidia Quadro FX5800 PCIe, 4.0GB of memory and 5751 NetXtreme network card

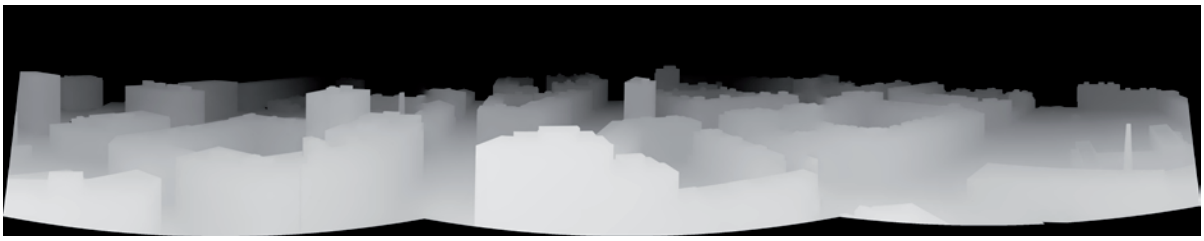


Figure 4 - Depth map of the panoramic frame in Figure 2 (limited to the first three original images). This map encodes the 3D information of the scene as different grey values: brighter objects are actually closer to the viewer, while further buildings and mountains are merged together into a single background plane (the darkest one).

at 1Gb/s. External cabinet with 8 HD SAS of 1TB each;

- *Tracking and interaction*: ARTtracking optical tracking with 6 degrees of freedom (position and orientation): four cameras (2 ARTtrack2 and 2 ARTtrack2/C) with CCD image sensors operating in the near infrared light spectrum and two reference systems. Stereoscopic glasses Infitec® deLuxe and Flystick ® for first person navigation.

5.2 Immersive 3D Content Generation

For a first test, four stereoscopic views of the city of Barcelona taken from the flat roof of Mediapro building have been generated. Each view is made up of stereoscopic video using a star-like arrangement. The field-of-view of each CCD camera is of 60º32'H x 36º18'V. Overlapping areas between adjacent views have been guaranteed. The stereo rig was mounted with two cameras at a distance of 0.94 m from each other (baseline) and having a convergence angle of 0.5º. Those parameters have been kept constant for each shooting sessions.

For each view, several shoots were taken using consecutive capture (1080i@25) at a bitrate of 100Mbps. These were edited to obtain one 20' stereo sequence comprising a few minutes at daybreak, noon, dusk and night. Then, the 2D sequences corresponding to each eye were combined to create a 230º panorama as shown in Figure 2.

5.3 Depth Information Retrieval

Extracting the 3D information from stereo videos is an essential task for profiting by a correct interaction with 3D content. This problem has been widely approached in last decades, especially in terms of stereo matching formulations (e.g. see [17-20]), but it still represents a research field opened to innovation, as several challenges are far to be eventually solved [21]. In the final analysis, the approach concerns solving the depth issue by triangulation of points. Therefore, a stereo matching algorithm is used to find a comprehensive set of pairs of corresponding points (*matches*) across distinct stereo views. These points are the projections in the image of a single point in the world. To retrieve how far this point is from the observer, the difference in match displacements (*disparity*), expressed through pixel coordinates, is used (see [22] for more details). The whole set of disparities could be shown in a map (*depth* or *disparity map*, as the one in Figure 4) and used as a good estimation of the 3D information of the scene.

One of the bigger issues in this research field concerns some simplified assumptions on the shooting conditions (such as constant brightness, well-controlled light conditions, Lambertian surfaces, lack of reflections and so on), since most of the stereo matching algorithm have been traditionally thought for and tested against "artificial", laboratory-created images or video. So far, as they are used on real video sequences (especially outdoor scenes), they usually under-perform because of the changing operating conditions ([23]). Since panoramic videos we are considering are falling in this category, our 3D multimedia content production will be affected by those light biases. Therefore, relying on a robust stereo matching algorithm taking into account the complexity of such a dynamic environment (described in terms of different weather conditions and light effects as well as moving objects, multiple occlusions and bigger disparity ranges) is the first mandatory requirement.

Another constraint to consider carefully is the level of quality of the final 3D reconstruction to achieve. Given the stereo algorithm taxonomy described in [17], it appears clear that the highest results come at relevant processing time and computational resources cost. Indeed, inferring the right 3D distance for each pixel in a frame is usually a task based on several iterative processes, assumptions on the whole image and energy function minimizations. Traditional techniques facing them require much more time to be computed than local methods, where similarity comparisons between local features in a small surrounding of each point are performed. Luckily, recent advances in GPU programming are making convenient global methods even for those applications where real-time performances are a compulsory requirement [24]. Given this context, high quality at a fraction of the time-consuming resources represents an interesting trade-off for 3D Multimedia applications too, where vision quality is essential and unavoidable.

For all the afore-mentioned reasons, we took our choice on the stereo algorithm described in [25]. Nonetheless, some of the problems introduced by panoramic videos are still partially solved by this algorithm. In particular, since a huge field of view is often expressed with a relatively small range of disparities, the correct depth estimation will occur only for objects relatively close to the observer (for instance those ones in the first and middle planes). This happens because disparity and real 3D values are in an inverse ratio: the smaller the disparity, the bigger the distance. So, by working at a pixel coordinates level, the complex 3D dynamic of the scene could not be fully caught. Another issue concerns some kind of inherent

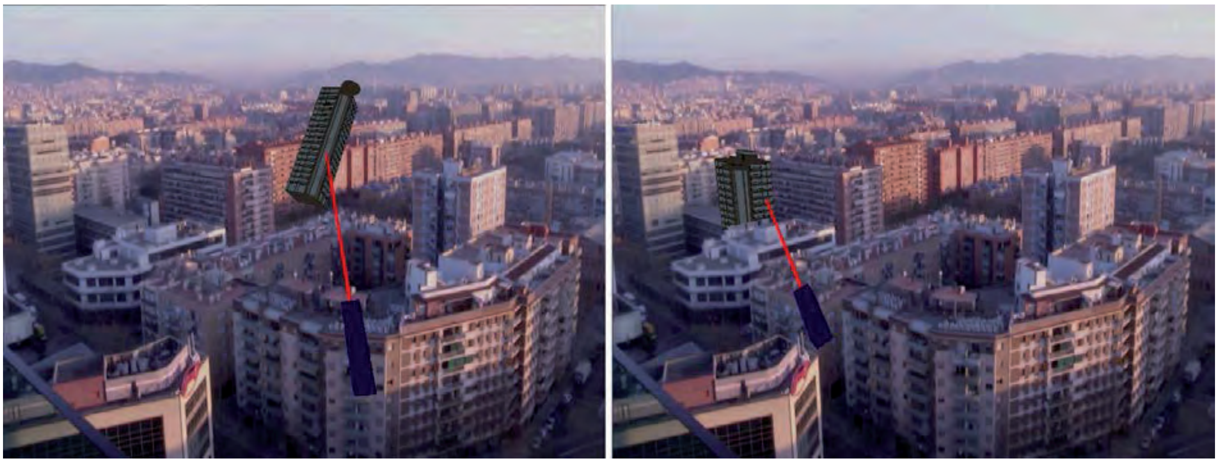


Figure 6 - Inserting a 3D model of a building into a video frame from Barcelona video. The user can move the model across the scene through a suitable interaction device (a symbolic representation of it has been super-imposed in the figure). The model is scaled according to its distance from the observer.

"noise" those videos are affected by, that is blur and lack of contrast. In this case, details are harder to be distinguished and therefore univocally matching correspondences turns to be a trickier task. Once again, far objects are most likely to be prone to such problems, but unfortunately it occurs also in closer planes, that is where the need of a better reconstruction is strictly required.

Our current research efforts in depth estimation are mainly focused on overcoming such issues. In this sense, the direction we have undertaken has two distinct paths. First of all, by detecting a greater number of stronger features (e.g. corners, edges; well known shapes, objects, silhouettes) as well as understanding how the scene is organized in terms of sets of homogeneous regions. Ideally such region segmentation could help in isolating macro areas where it is likely to assign the same disparity and / or a single object has been detected. The rationale here is to improve the matching reliability at well-defined places and propagate this information across group of pixels belonging to the same disparity layer. This geometrical reconstruction should be robust enough to ensure consistency across frames even in the most difficult cases, such as when occlusions occur or in extreme light conditions. The second idea consists in integrating external information in order to refine the so-computed disparity map. For instance, in the case of the Barcelona video, the city topology is well-known as well as the building models might be easily imported from other modeling tools (such as Google Maps). The main advantages in such approach are a better quality of the final depth map and the integration of real 3D information. Figure 4 shows an example of such a map computed with this approach.

5.4 Merging synthetic objects into the stereoscopic video

This section describes a method to merge synthetic objects with a stereoscopic video sequence using depth maps attached to a video frame image. For the purpose of this algorithm, it is assumed that the depth-map has been previously extracted from the stereo video pair. The general idea is to generate a depth displacement model of the 2D scene and compute object occlusions attending to the Z-buffer information of both, the scene and the synthetic objects.

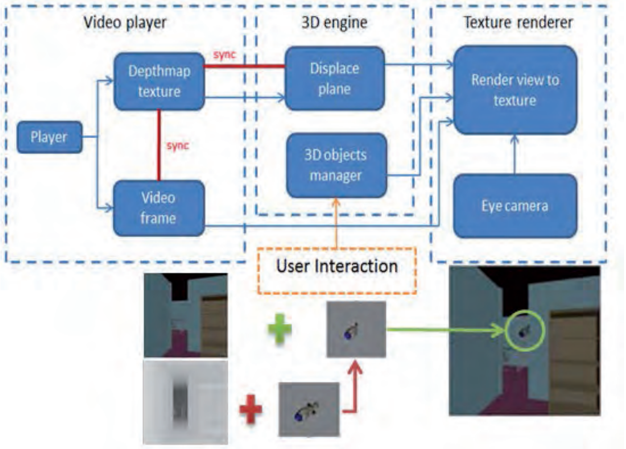


Figure 5 - Scheme and a practical example of the synthetic object insertion algorithm. When inserting a 3D model, possible occlusions are computed through the frame depth map in order to allow a realistic visualization of the new generated frame

All the components defined in this method have been implemented using 3DVia Virtools 5.0 from Dassault Systémes. This programming tool is aimed to develop applications based on virtual environments and also provides the capability to play video content. However, the video synchronization between eye streams is not supported by default and additional considerations have been necessary to take into account regarding this issue (see section 6.5). Figure 5 illustrates one cycle of the implemented insertion algorithm for a particular eye frame. At the end of each cycle, a new image pair with the synthetic objects inserted is obtained and used to replace the original. In order to prevent unwanted performance dropdowns, it is required that the whole process is fulfilled within the time imposed by the source video frame rate (e.g. a 25 fps video sets a maximum processing time of 40ms per cycle).

The components shown have been grouped inside three different modules attending their roles in the overall process. These modules are: i) the video player in charge

RE@CT - IMMERSIVE PRODUCTION AND DELIVERY OF INTERACTIVE 3D CONTENT

Oliver Grau[1], Edmond Boyer[2], Peng Huang[3],
David Knossow[4], Emilio Maggio[5], David Schneider[6]

[1]BBC R&D, London, UK; [2]INRIA, Grenoble, France; [3]Surrey University, Guildford, UK; [4]ARTEFACTO, Rennes, France; [5]Vicon Motion Systems (OMG group), Oxford, UK; [6]Fraunhofer/HHI, Berlin, Germany

E-mail: [1]Oliver.Grau@bbc.co.uk, [2]edmond.boyer@inria.fr , [3]peng.huang@surrey.ac.uk, [4]d.knossow@artefacto.fr, [5]emilio.maggio@vicon.com, [6]David.schneider@hhi.fraunhofer.de

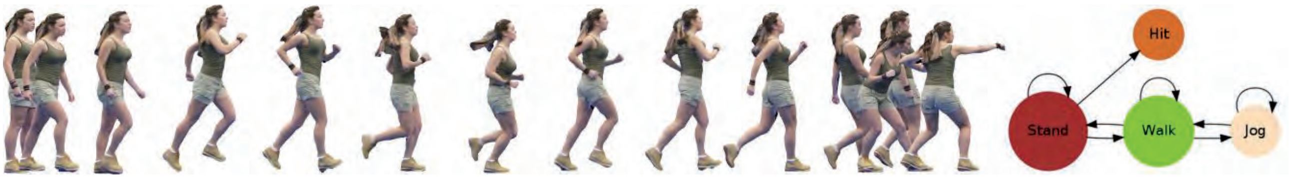


Figure 1 An example of a simple Surface Motion Graph and Animation results.

Abstract: **This paper describes the aims and concepts of the FP7 RE@CT project. Building upon the latest advances in 3D capture and free-viewpoint video RE@CT aims to revolutionise the production of realistic characters and significantly reduce costs by developing an automated process to extract and represent animated characters from actor performance capture in a multiple camera studio.**

The key innovation is the development of methods for analysis and representation of 3D video to allow reuse for real-time interactive animation. This will enable efficient authoring of interactive characters with video quality appearance and motion.

Keywords: Character animation, video game development, immersive media, motion capture.

1 INTRODUCTION

Computer animation is now an essential technique for the production of digital media. Recent advances in graphics hardware have produced video games with a degree of realism only achieved by offline rendered computer generated imagery (CGI) a few years ago. However, applications like games require interactive synthesised animations on the fly. RE@CT [1] aims to revolutionise the production of highly realistic animations of human actors at significantly reduced costs, by developing an automated process which extracts both the visual appearance and motion of actors in a multi-camera studio by combining the latest advances in 3D video into a new character and motion representation.

Technically the production of realistic looking animations of characters is probably the most challenging part of the production of interactive games applications and requires highly skilled experts. The production costs of high-end video games are reaching an average of $10 million for console games and $30k - $300k for casual and social games. Within this budget the production of animations can account for up to 20% of the total budget for smaller and independent production companies without specialist facilities [2]. In these cases animations are often produced with labour-intensive key-frame animation. Since RE@CT aims to automate the animation process to a great extend these costs are expected to be significantly reduced.

The traditional approach is to design the appearance of a character independent from the animation. This is flexible, as the appearance is designed with full artistic freedom on a computer system. The animation is added as a second, separate step using manual animation techniques and motion captured animation data. However, there is currently no seamless process to produce animated content of real people. Applications for this process include production of content with known actors alongside TV- or movie productions (transmedia), capture of heritage or historical events.

RE@CT is developing high-quality capture components based on multiple cameras to allow capture of both the appearance and motion of the actors. This builds upon previously developed techniques to extract 3D information from multiple cameras [4][5]. We extend these capture techniques by active tracking and high-detail face capture and modelling. We also include an immersive feedback system, previously developed for the production of special effects [4] to aid the actors in our capture studio.

Current techniques for the animation of captured content combine manual animation techniques with multi-camera video [3]. RE@CT on the other hand is developing new methods that automatically analyse the captured motion and transform it into a representation that can be used in a games engine to synthesise new movements from the stored data.

The remainder of this paper is structured as follows: The next section gives a brief overview of the project's

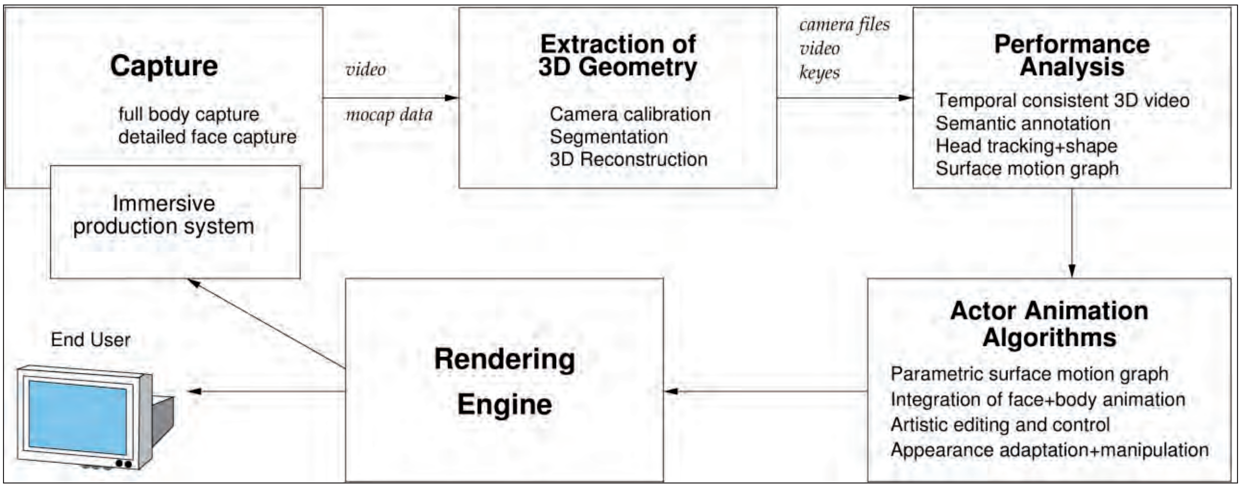

Capture
full body capture
detailed face capture
video
mocap data
Extraction of 3D Geometry
Camera calibration
Segmentation
3D Reconstruction
camera files
video
keyes
Performance Analysis
Temporal consistent 3D video
Semantic annotation
Head tracking+shape
Surface motion graph
Immersive production system
End User
Rendering Engine
Actor Animation Algorithms
Parametric surface motion graph
Integration of face+body animation
Artistic editing and control
Appearance adaptation+manipulation

Figure 2 RE@CT overview

components. Section 2 describes the immersive capture system and 3D processing pipeline. In section 4 some details of the performance analysis are given. Section 5 outlines the rendering engine. The paper finishes with first results and conclusions.

2 SYSTEM OVERVIEW

Figure 2 shows the main functional modules of the RE@CT production pipeline. The **capture** module includes the physical studio set-up to capture multiple video streams of an actor. Although the final goal of the project is an image-based system, we include a motion capture (mocap) sub-system for the initial phase of the project and as a reference for verification of the image-based algorithms developed in the project. The video streams are then processed in the **extraction of 3D geometry** module. This module extracts 3D information of the actor on a frame-by-frame basis. Techniques used here are based on visual hull computation and require calibrated cameras and keyed (segmented) images.

The **performance analysis** module performs a temporal alignment of the 3D data. The action is then analysed and a semantic annotation is added. This allows the storage of the action into a surface motion graph.

The **actor animation algorithms** perform a final structuring of the captured data. A parametric representation adds stylistic control and an operator is now able to manipulate the action and to define behaviour of the character as required by the interactive application.

The **rendering engine** is either standalone software or a plugin to a games engine. It allows the play-back and on-the-fly editing of the captured action as controlled by the interactive application. The rendering engine is also used in the **immersive production system** to integrate previously captured action into new capture scenarios.

3 CAPTURE AND EXTRACTION OF 3D GEOMETRY

One of the major aims of RE@CT is to design a studio system to provide high-quality capture of human actions that preserve the full repertoire of body language as well as highly detailed facial expressions used in acting. The project is developing a camera-based acquisition system for whole body and facial 3D video. In order to help actors interact with virtual objects, a novel immersive feedback system will also be investigated.

3.1 Capture

In order to allow for simultaneous development of temporally consistent data representations and actor animation algorithms (see Sections 4 and 5), the project will develop two capture systems: a preliminary hybrid marker-based/marker-less system, and a final video-only markerless system.

3.1.1 Hybrid system

To capture the actors' bodies the hybrid performance capture system will combine video from multiple HD cameras and state-of-the-art marker-based motion capture technology. The marker positions will drive classic articulated models and will provide additional information for 3D reconstruction. Synchronisation between the two systems is achieved by means of industry standard gen-lock signals.

To capture facial expressions in the hybrid system, head mounted capture systems from VICON will be used. Each system is composed of four monochrome cameras positioned as two stereo pairs as showed in Figure 3. The videos are compressed using H.264 and stored on portable logger SSD drives also worn by the actor. The head-mounted cameras synchronize with the rest of the capture system using jam-syncing. An external Time-Code signal is passed to the logger by plugging a cable at the beginning of the capture section. Once the cable is unplugged the logger uses an internal clock to maintain synchronisation with the external system.

3.1.2 Markerless system

The final performance capture system will use multi-view video only and will be composed of a set of commercial HD video cameras with different frame rates: 3CCD lower frame-rate cameras for high quality video, and single CCD high frame-rate cameras to capture fast motions.

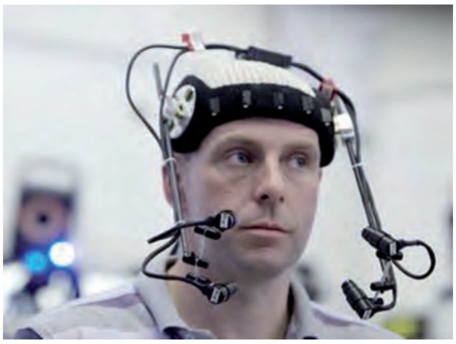


Figure 3 Head mounted camera system used to capture the facial expression of the actors.

In addition to the stationary camera array, a set of multiple pan/tilt/zoom cameras will simultaneously capture high-resolution views of the head. These additional views will provide more detailed information on actors' facial expressions.

3.1.3 System calibration

System calibration in terms of camera locations and lens distortion correction is achieved via standard bundle adjustment techniques. A wand with colour LEDs attached is used as a calibration device. The calibration procedure involves recording views of the wand waved in front of the cameras. Prior knowledge of the LED positioning is used to detect the 2D location of the wand in each frame. Then a bundle adjustment algorithm uses the wand locations to estimate the parameters of the lens distortion model, the camera intrinsic parameters (i.e., focal length, image format, and principal point), and the camera position and orientation.

3.1.4 Face capture

RE@CT features a dedicated capture and processing chain for the actors' heads. This allows cutting to head/face close-ups with a free choice of viewpoint at any time. Also, head shots can be re-animated with the RE@CT animation engine to synthesise new views on the fly as requested by the games engine.

The head capture process must capture data of sufficient quantity and quality for this application. Therefore, footage of the actors' heads is shot on-set by multiple dedicated cameras, recording at full HD or higher quality at a high frame-rate. The cameras have pan/tilt/zoom (PTZ) capability in order to follow the action. They are operated either manually by trained personnel or automatically by robotic camera heads.

The number of head cameras required depends on the number of actors involved in the scene and on the degree of freedom required for image synthesis. In practice, coverage from the front over a baseline angle of 120 degrees is most relevant, allowing the rendering of half-profile views from both directions. An angle of 180 degrees must be covered if full profiles are required. The PTZ cameras must be synchronized as well as calibrated. The RE@CT studio is equipped with PTZ cameras with electronic feedback of zoom settings and an optical calibration system for extrinsic camera parameters, consisting of markers on the studio ceiling which are recorded by a second camera mounted on the actual studio camera.

3.2 3D Processing

3.2.1 Whole body modelling

The 3D geometry of the captured action is computed frame-by-frame using a robust implementation of a visual hull computation. This requires known camera parameters, as computed by the system calibration and a segmentation of the scene into background and foreground, i.e. the actor's silhouette. We use chroma-keying for the segmentation.

3.2.2 Head modelling

To support the computation of 3D information from the PTZ sequences, a set of detailed, fully textured 3D models is generated in advance from the actors involved in the shot. The models are captured with an image-based head-capture rig built by Fraunhofer HHI. The rig comprises multiple stereo pairs of SLR cameras as well as studio flash lighting. The 3D models cover a head-and-shoulder view of the person over an angle of 120 to 180 degrees (frontally). As the reconstruction process is fully image-based the capture process is instantaneous. The actors are captured with different facial expressions, with an emphasis on expressions that have a large impact on the visual hull and overall appearance of the face.

After capture, the footage is analyzed in order to generate temporally and spatially consistent depth information. The captured 3D models support this step by providing detailed information about the head involved. However, pose and expression of the head in the PTZ footage will deviate from the 3D model, which therefore serves primarily as a proxy for analysis.

First, a relationship between each PTZ video stream and the captured model is established. To this end, the individual PTZ video frames are matched with textured renderings of the 3D model using feature matching and image-based optimization techniques. Then the computed relationship of the individual frames to the model is used together with the camera calibration data to establish a consistent spatial relationship between synchronous frames of the different PTZ streams. Finally, the content of each PTZ stream is tracked over time.

The output of the analysis stage is:

depth information for the region of each PTZ video frame showing the actor's head,

inter-view correspondence information, relating the head region of each PTZ frame to its synchronous frames in neighbouring views,

temporal correspondence information, relating the head region of each PTZ frame to the next frame of the same video stream

For motion-graph re-animation of the footage, semantic information on the head footage is required. This will be obtained by semantically annotating the captured 3D models to describe the facial expressions and propagating this information to the PTZ footage in the registration process. This may require some form of manual intervention for which appropriate tools will be developed.

3.3 Immersive Production System

The processing modules described in the previous section to extract 3D geometry of actors are based on the assumption that the scene can be segmented. Ideally actors are captured in isolation in a controlled studio environment with a chroma-keying facility. That raises the issue that actors have no visual reference: It is very difficult even for trained actors to interact with virtual objects they do not see. The unknown position of another person/object may lead to incorrect pose or gaze of the actor. Similarly, the temporal synchronisation with movements, events, or gestures is difficult to achieve without any feedback.

In order to simplify the capturing process and to obtain better results we will investigate the use of a feedback system previously developed to help the production of special effects [4]. It provides the actor with visual feedback of other virtual humans or objects with which they have to interact. The virtual scene is rendered with accurate timing and projected into the real studio environment without interfering with the capturing. This is achieved with the help of a special keying system, which makes use of retro-reflective cloth. The cameras are equipped with a ring of blue LEDs and the light reflected by the retro-reflective material makes the cloth appear saturated blue as required for chroma-keying. At the same time the actor can observe images from a video projector, as depicted in Figure 4. The light levels have to be balanced so that the projector does not interfere with the chroma-keying, but in practice the set-up is very robust, as the retro-reflective cloth has a high reflectivity peak at a narrow angle of only a few degrees (see [4] for details).

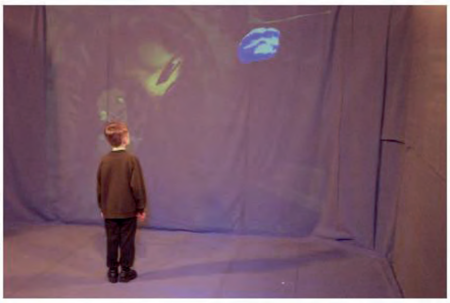


Figure 4: View-dependent projection on retro-reflective cloth

To give the actor an immersive visualisation of the scene, it must be rendered using a view-dependent approach. This requires a rendering system able to render a view of the scene depending on a) the head position of the actor, b) the position of the projector and c) the screen size and position. The position, size and internal projection parameters, i.e. b) and c) are static and can be calibrated and set-up in forehand.

For the actor's head position a real-time head tracking system is required. This is implemented using the video streams of the capture system and real-time processing of the IP-based system. The head position will then be streamed to a rendering module, based on the rendering engine described in section 5. The view-dependent rendering requires a specific setup of the projection.

4 PERFORMANCE ANALYSIS

The system described in the previous section captures shape geometry and appearance independently in each video frame. The performance analysis estimates additional temporal information from this data in order to enable further processing such as actor animation. This additional information includes shape motion that allows a temporally-consistent representation to be built. It also includes semantic information, such as body part labels, to provide more precise knowledge of the observed scene dynamics. This information is used for interaction and animation purposes.

4.1 Temporally Consistent Representations

In order to structure 3D video sequences into meaningful representations for animation, e.g., motion graphs as described in section 5, motion information must be recovered. This key step in the temporal modelling pipeline is implicitly solved when considering traditional motion capture data; however it is a difficult problem when considering 4D sequences of 3D models independently estimated over time. The strategy followed consists of representing 3D video sequences, within a dataset, as motions and deformations of a single known model, i.e. a 3D mesh.

A patch based deformation model [1] is used to track the evolution of a reference mesh over each sequence (see Figure 5). This model preserves the reference mesh consistency over time by enforcing local rigidity constraints. To this end, deformations are encoded as rigid motions of patches. Vertices of the reference mesh are associated to patches at different levels of detail, thereby enabling a coarse to fine strategy when tracking large deformations.

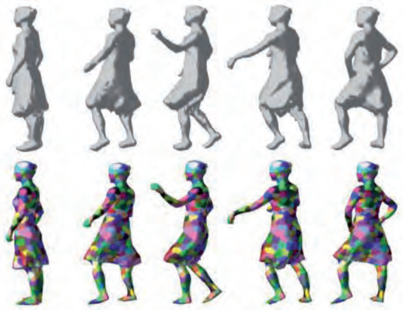


Figure 5: An example of mesh tracking (bottom) given independently estimated 3D models (top).

4.2 Semantic information

Temporally-coherent representations provide dense motion information for shapes in the form of mesh vertex trajectories over time. These representations do not account for intrinsic properties of shapes neither do they provide compact motion information nor semantic information on shapes, e.g. identification of head, hands, etc. This information however is useful for any shape motion analysis and will enrich the 3D video representations.

When markers are tracked, semantic information can be extracted by fitting a kinematic model to the labelled marker positions using standard marker-based motion capture algorithms. In order to recover similar information in a marker-less environment, 3D video sequences must be analysed. To this purpose, tracked meshes are segmented into parts that exhibit rigid motions over temporal sequences. This is achieved by clustering mesh vertices with respect to their displacement vectors over a time window [10]. Labels can then be associated to rigid parts by either fitting a known articulated model or by considering connectivity information between identified rigid parts.

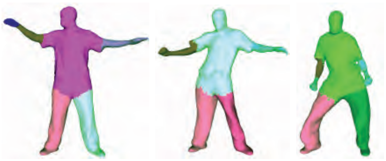


Figure 6: An example of time evolving segmentation of body parts into rigid segments with respect to vertex displacements.

5 ACTOR ANIMATION AND RENDERING

5.1 Actor Animation

The framework for actor animation comprises two stages: pre-processing the database of the 3D video sequences into a Surface Motion Graph [6] or a Parametric Motion Graph [7]; and synthesising actor animation by optimising a path on the graph which best satisfies user input. The user input could be user-defined global constraints (offline) for authoring purposes or interactive control (online) for game-play purposes.

5.1.1 Surface Motion Graph

The Surface Motion Graph consists of a set of motion classes and links between them. Each motion class contains a single motion sequence (e.g. a walk or a hit). Each edge represents a transition across motion classes. The transition is identified as a pair of frames (or a set of overlapped frames), which minimises transition cost from source to target motion. The transition cost is computed as 3D shape dissimilarity between transition frames. A spherical volume-based 3D shape histogram descriptor [1] is used to compare 3D shape dissimilarity between all meshes in the database and pre-compute a 3D shape similarity matrix. Transitions are automatically found and the Surface Motion Graph is constructed. The user is allowed to modify the graph (add/remove some transitions and smooth the transitions by linear/non-linear blending overlapped meshes) to increase the visual quality of final animation.

5.1.2 Parametric Motion Graph

The Parametric Motion Graph can be considered as a natural extension to the Surface Motion Graph. It also consists of a set of motion classes and links between them. Each motion class contains two or more motion sequences (e.g. a slow walk and a fast walk) parameterised to allow generation of any motion in between (e.g. speed of walking). Each link represents a transition between motion classes. A transition may be triggered during the rendering process, for example by a user or by an event in a game. The transition cost is a combination of responsiveness (delay time) and smoothness (3D shape dissimilarity). The 3D shape similarity for all 3D meshes is pre-computed. The dissimilarity for parameterized frames is then approximated as a linear/non-linear interpolation of nearby existing dissimilarity values at run-time to allow real-time transitions between motions.

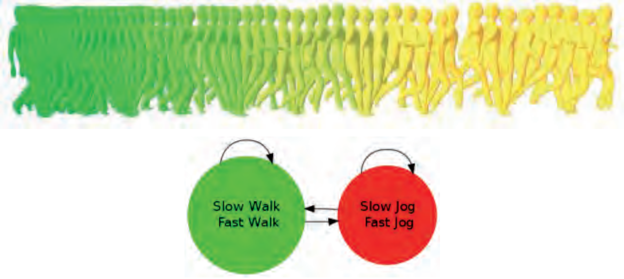


Figure 7: An example of a simple Parametric Motion Graph and Animation results.

5.2 Rendering Engine

5.2.1 Textured Mesh Rendering

View dependant rendering

View-dependent rendering uses a subset of the cameras that are closest to the virtual camera as texture images, with a weight defined according to the cameras' relative distance to the virtual viewpoint. By using the original camera images, the highest-resolution appearance in the representation can be retained, and view-dependent lighting effects such as surface specularity can be incorporated. In practice, the resulting rendering for both original and blended meshes looks realistic and maximally preserves the captured video quality.

Single texture rendering

View-dependent rendering is currently not supported by games engines and other online rendering SDKs. For that reason we will also consider a combined texture map that is compliant with commercially-available rendering SDKs. This approach is common within the context of CG production and will allow the rendering to be done using commercially-available software. Hence, using standard file formats, outputs from the project will be usable within a standard production process.

5.2.2 Interactive rendering tool

Using the technology described in the paragraph above, the rendering engine will be used to interactively animate the meshes. More precisely, an authoring tool will be integrated in the rendering engine to control the motions being rendered in real time. This will allow the end user to control the sequence of motions, defining the motion type (run, walk, jump, etc.), its speed (run slow, fast, etc.) and style.

We aim at proposing a set of very simple interactions for the end-user to control the overall motion and style possibilities. When available, tactile interfaces will be used to ease the motion control.

6 RESULTS

Currently the RE@CT project is focussing on the capture techniques and the integration of the processing modules outlined in previous sections. The first test production will address a cultural heritage application, e.g. for use in museum exhibitions.

RE@CT will provide cultural and historical heritage centres with highly-realistic applications. As an example, RE@CT will allow younger visitors, as well as adults, to control medieval characters and battle in a field using both augmented reality and realism provided by the RE@CT project. Figure 8 shows a typical scenario in such an application. The static assets are themed to a medieval setting in a castle.

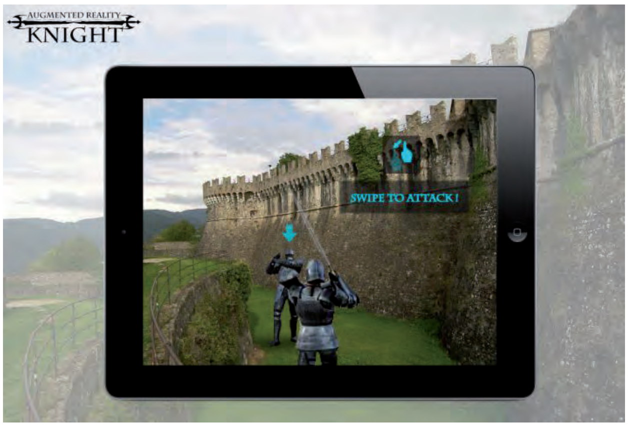


Figure 8: Augmented reality application in a cultural heritage setting.

The characters are captured and processed using RE@CT techniques. Figure 9 shows a picture of a recent test production.

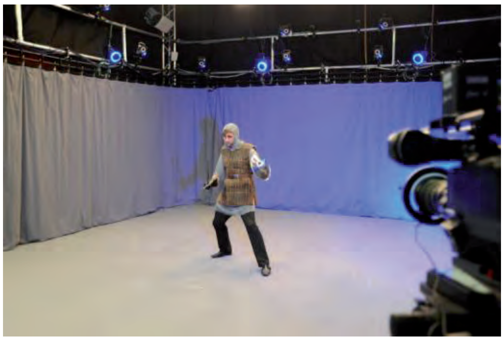


Figure 9: Production test in the studio

7 CONCLUSIONS

The RE@CT project aims to provide new tools for the production of interactive character animations. The application scenarios include cultural heritage, education, simulation, and live TV-productions. These applications are often called ‘serious gaming’. Furthermore, RE@CT also has the potential to be applied to traditional video game production.

At its core the project is developing tools that capture the action of real actors and then automatically build a database from these sampled actions to be used in the interactive context of a games engine to generate new animations on the fly.

This paper has given a brief overview of the individual components that the project is developing. Initial tests have already been carried out to demonstrate the benefits of the techniques. The initial test scenarios are in the domain of cultural heritage. Applications related to broadcast production will follow in the next phase of the project.

References

[1] RE@CT web-site: http://react-project.eu/

[2] The evolution of game animation, white paper, http://www.mixamo.com/c/articles/mixamo_whitepaper

[3] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, Christian Theobalt, Video-based Characters - Creating New Human Performances from a Multi-view Video Database, in ACM Transactions on Graphics 30(4) (Proc. of SIGGRAPH 2011).

[4] O. Grau, T. Pullen, G. Thomas, A combined studio production system for 3D capturing of live action and immersive actor feedback, IEEE Tr. on Systems and Circuits for Video Technology. March 2004.

[5] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. In Proc. Of ICCV, pages 915–922, 2003.

[6] P. Huang, J. Starck and A. Hilton. Human Motion Synthesis from 3D Video. In Proceedings of the Twenty-Second IEEE Conference on Computer Vision and Pattern Recognition (CVPR’09), pages 1478-1485.

[7] D. Casas, M. Tejera, J-Y Guillemaut and A. Hilton. 4D Parametric Motion Graphs for Interactive Animation. In Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2012 (I3D’12).

[8] P. Huang, J. Starck and A. Hilton. Shape Similarity for 3D Video Sequences of People. In International Journal of Computer Vision (IJCV) special issue on 3D Object Retrieval, Volume 89, Issue 2-3, September 2010.

[9] C. Cagniart, E Boyer and S. Ilic. Probabilistic Deformable Surface Tracking from Multiple Videos. In 11th European Conference on Computer Vision, Sep 2010.

[10] R. Arcila, C. Cagniart, F. Hétroy, E. Boyer and F. Dupont. Temporally coherent mesh sequence segmentations. INRIA Research Report RR-7856, 2012.

of decoding the original video frames along with its attached depth map texture; ii) the 3D engine managing all the external objects originally not related to the video and occlusions and iii) the texture renderer component which gives as an output the calculated video frame with the synthetic objects merged.

Initially, the *video player* module takes the stereo recording as an input, with the depth information and the separate views for each eye. These views can be loaded within a container file or from separate sources as long as the frame synchronization is maintained. The goal of the video player is then to synchronously extract the stereo frame image and depth map texture separately so that they can be processed by the rest of the components.

The *3D engine* manages all the synthetics objects of the virtual environment to be inserted in the video frames. Their position and orientation within the rendering space can be controlled dynamically by the user interactions so that occlusions are calculated in real time according to the video scene being played. This can be seen in Figure 6, where a real user interacts with the scene and places a virtual building into the video. As a result, the inserted 3D model appears to be occluded by the rest of buildings that have been recorded.

To generate the depth of the scene, a planar surface called the *displace plane* is positioned at the furthest distance that has been captured by the video camera. For each video frame, the vertices of the plane are moved towards the camera direction according to the value of its pixel position in the depth map texture. After processing the whole texture, a mesh describing the depth geometry of the scene is obtained and its Z-buffer information can be used to compute the occlusions with the objects.

The number of vertices used to model the displace plane has a significant impact in the overall performance of the algorithm. A greater number will result in a more detailed scene mesh and therefore more resolution to compute occlusions (particularly around borders). During tests, a plane mesh formed by 256x256 vertices has been found to be suitable for both, resolution and overall performance.

The final step of the algorithm cycle is to generate the output frame with the merged object so that it can be used to replace the original on the screen. For this purpose, a virtual eye camera reference configured with the same extrinsic parameters than the camera used to record the original source is configured to take a picture of the complete scene. In this phase, the displace plane is not to be rendered as only its Z-buffer information is relevant to calculate object occlusions.

Preliminary tests have been run on an Intel Xeon 3.2GHz CPU with 6GB RAM memory and an nVidia GTX 580 graphic card. Under these conditions, the implemented method was able to provide 70-90 fps which is approximately three times the frame rate needed to prevent video stutter. The video source used has been the time-lapse capture of Barcelona with a resolution of 1400x1050 (fixed by the native resolution of the projectors available in our CAVE^{TM} infrastructure).

5.5 Content Adaptation and Visualization

Previously, a way of merging synthetic objects inside a stereoscopic video image has been defined. As a result, a new video frame composed by the original image and the occluded object is created and can therefore be used to be presented inside the virtual environment. To let this image be visualized in the CAVE^{TM}, an auxiliary 3D sprite mapped with the generated textures is used. This sprite is located with any position, orientation and scale, in the same way than other generic object of the scene.

One important thing to be considered is that the generated video frames have to be presented to the user in stereoscopic mode. For this reason, the 3D sprite containing the video is mapped with a stereo material that takes as input parameters the video textures for each eye. During visualization, the material is synchronized with the CAVE^{TM} stereo glasses to switch its texture between both eye perspectives.

On the other hand, objects are also allowed be rendered in the virtual space located between the video 3D sprites and the user. Inside this volume, there is no need to calculate objects occlusions with video and they can also take benefit of the CAVE^{TM} capability to visualize the virtual elements from any view perspective. This issue has been exploited to implement user interactivity, allowing moving objects inside the stereoscopic video or bringing them at the front to be observed from any angle.

Calculating object occlusions and generating the video frames for both eyes is a costly operation in terms of CPU performance. This is an important issue to consider for the CAVE^{TM} layout, where it is needed to render a separate 3D sprite video for each screen to complete the panorama. As it has been described, the CAVE^{TM} is controlled by a set of workstations connected in a cluster, in which only one computer is responsible for the rendering of one of the screens. In order to guarantee a good frame rate performance, a different process for video playing and objects merging has been allocated into each workstation of the cluster. User position inside the environment is therefore restricted so that only one video sprite is inside each screen camera frustum at a given time.

Another problem that has been necessary to solve are the possible mismatches between the two streams of the stereoscopic video. This issue can affect the user 3D experience critically if not corrected conveniently. To accomplish this correction, a synchronization algorithm shown in Figure 7 has been implemented inside the video player module.

The threshold operator determines the maximum difference in time at which the user is aware of the asynchrony. When this value is surpassed by any of the video streams, then the speed of the affected video playback is reduced by 10% until it becomes synchronized again. During testing and for the Barcelona city user case, a threshold value of 100ms has been found to be suitable for the correction algorithm to work and also the speed reduction does not disturb the visualization experience. Nevertheless, this value would have to be reconsidered when dealing with more dynamic video

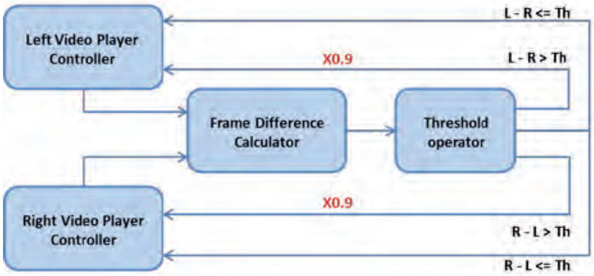

Left Video Player Controller
L - R <= Th
X0.9
L - R > Th
Frame Difference Calculator
Threshold operator
X1
X0.9
R - L > Th
Right Video Player Controller
R - L <= Th

Figure 7 - Synchronization algorithm for stereoscopic video playing.

sequences where image can vary substantially in a short time period.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the basic ideas about a new way of conceiving immersive environments – such as the I-Space at CeDInt laboratories in Madrid - and user interaction. Indeed, the paradigm is intended to lead to the massive production of new 3D multimedia content where users could play a more active role, by having direct access to the 3D models contained in it. In other words, it could lead to a shift from Virtual to Augmented Reality paradigm in a typical CAVE^{TM} environment. Both essential requirements and the architectural organization of our prototype have been discussed, as well as some of the most important technical issues deriving from them. In particular, we introduced the estimation of 3D information from stereo videos and the procedure of user-guided insertion of 3D synthetic models into the stereo flow. Preliminary results on a panoramic stereo video have been presented. Panoramic videos have been chosen because in a CAVE^{TM} room, they could enlarge the immersive sensation of the users. Even if in a prototype phase, it is already possible to envisage some practical uses of such paradigm. In particular, future scenarios could range from the game and entertainment industry, to sport and cultural events broadcasting, passing for sure from both architectural and urban development and educational training.

Acknowledgments

This work is part of a research project partially funded by the Spanish Ministry of Industry, Tourism and Commerce through the project TSI-020302-2010-61 (ImmersiveTV, available online at: http://www.immersivetv.es), developed in collaboration with a consortium of entities. Authors would like to thank all the participants involved into it and in particular Jordi Alonso and Jordi Ramon from MEDIAPRO for the support provided through the funded project.

References

[1] W. IJsselsteijn, P. J. Seuntiëns, and L. M. Meesters: “State-of-the-art in human factors and quality issues of stereoscopic broadcast television”, Deliverable ATTEST/WP5/01, Eindhoven (NL), 2002.

[2] J. Steuer: “Defining Virtual Reality: Dimensions determining telepresence”, Journal of Communication, vol. 42(2), pp. 73-93, 1992.

[3] G. Burdea and P. Coiffet: “Virtual reality technology”, New York, Wiley, 1994.

[4] D.R. Proffitt and M.K. Kaiser: “Human factors in Virtual Reality Development”, Tutorial at the Virtual Reality Annual International Symposium, Research Triangle, 1995.

[5] K.M. Stanney, R.R. Mourant, R.S. Kennedy: “Human factors issues in virtual environments: a review of the literature”, Presence: Teleoperators and Virtual Environments, vol. 7(4), pp. 327-351, 1998.

[6] M. Slater and S. Wilbur: “A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments,” *Presence-Teleoperators and Virtual Environments*, vol. 6(6), pp. 603–616, 1997.

[7] M. J. Schuemie, P. van der Straaten, M. Krijn, and C. A. P. G. van der Mast, “Research on Presence in Virtual Reality: A Survey,” *CyberPsychology & Behavior*, vol. 4, no. 2, pp. 183–201, 2001.

[8] B.G. Witmer and M.J. Singer, “Measuring Presence in Virtual Environments: A Presence Questionnaire,” *Presence: Teleoperators and Virtual Environments*, vol. 7(3), pp. 225–240, 1998.

[9] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R.V..Kenyon, and J.C. Hart: "The CAVE: Audio Visual Experience Automatic Virtual Environment",Communications of the ACM, vol. 35(6), pp. 64-72, 1992.

[10] C. Cruz-Neira, D.J. Sandin and T.A. DeFanti: "Surround-Screen Projection-based Virtual Reality: The Design and Implementation of the CAVE", SIGGRAPH'93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 135 - 142, 1993.

[11] “Format-Agnostic SCript-based INterAcTive Experience (FascinatE)", available online at: http://www.fascinate-project.eu/.

[12] “MUltimedia SCAlable 3D for Europe (MUSCADE).”, available online at: http://www.muscade.eu/index.html.

[13] P. Sturm, S. Ramalingam, and J.P. Tardif: "Camera Models and Fundamental Concepts Used in Geometric Computer Vision", now Publishers Inc, 2011.

[14] S. Peleg, M. Ben-ezra, and Y. Pritch: “Omnistereo: Panoramic stereo imaging,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 279 - 290, 2001.

[15] C. Weissig, O. Schreer, and P. Eisert: “The Ultimate Immersive Experience: Panoramic 3D Video Acquisition,” in Advances in Multimedia, 2012, vol. 7131, February 2010.

[16] S. K. Han and J. P. Schulze: “High Resolution Video Playback in Immersive Virtual Environments”, IEEE Virtual Reality Conference, pp. 247-248, 2009.

[17] D. Scharstein and R. Szelinski: “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International Journal of Computer Vision, no. 1, pp. 131-140, 2002.

[18] B. Zitová and J. Flusser, “Image registration methods: a survey,” Image and Vision Computing, vol. 21, no. 11, pp. 977-1000, Oct. 2003.

[19] M. Gong, R. Yang, L. Wang, and M. Gong: “A performance study on different cost aggregation approaches used in real-time stereo matching”, International Journal of Computer Vision, vol. 75(2), pp. 283-296, 2007.

[20] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski: “A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms”, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 519 - 528, 2006.

[21] “Middlebury Stereo Vision Home Page", available online at: http://vision.middlebury.edu/stereo/.

[22] O. Faugeras and Q.T. Luong: “The geometry of multiple images”, The MIT Press, March 2004.

[23] L. Nalpantidis and A. Gasteratos: “Stereo vision for robotic applications in the presence of non-ideal lighting conditions”, Image and Vision Computing, vol. 28(6), pp. 940-951, 2010.

[24] V. Vineet and P. J. Narayanan: “CUDA cuts: Fast graph cuts on the GPU", IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1-8, 2008.

[25] A. Geiger, M. Roser, and R. Urtasun: “Efficient Large-Scale Stereo Matching”, ACCV'10 Proceedings of the 10th Asian conference on Computer Vision, vol. 1, pp. 25-38, 2010.

Application & Experimentation track

Session 3B

Chaired by Jovanka Adzic, Telecom Italia

«Sourcing content from social media: a journalistic application of the SocIoS platform»
Authors: Birgit Gray, Sara Porat, Konstantinos Tserpes

«Society@school: social enhanced reading experiences for education»
Authors: F. L. Mondin, O. R. Rocha, M. Belluati, E. A. M. Guercio

«Synaesthesia Innovative music components for collaborating and creating music with objects in real space»
Authors: Michela Magas, Christopher Rea

«Connected TV: new opportunities for the accessibility»
Authors: Carlos Alberto Martín, José Manuel Menéndez, Guillermo Cisneros

«User-activated public service»
Authors: Miriam Lerkenfeld, Torsten Andreasen

European Commission

The NEM Summit is an annual Conference and Exhibition organised by the NEM Initiative under the aegis of the European Commission (DG Information Society and Media), supported by Sigma Orionis and Eurescom GmbH.

Privilege sponsor

TÜRK TELEKOM

Platinum sponsors

arçelik

technicolor

TÜBİTAK

TuR&Bo PUBLIC - PRIVATE PARTNERSHIP

Gold sponsors

orange™

Microsoft®

SAMPAŞ® AKILLI KENTLER

ST®

TELECOM ITALIA

ZTE中兴

Silver sponsors

3D-ConTourNet

CHORUS+ AUDIO-VISUAL SEARCH

EXPERIMEDIA

Explore

fascinate

FI-CONTENT

GUIDE Gentle User Interfaces for Elderly People

Ideal-ist www.ideal-ist.net

Inria INVENTEURS DU MONDE NUMÉRIQUE

QLectives

SCENE

SMARD Networked Media R&D for SMEs

THESEUS New Technologies for the Internet of Services

Vconect

Organised by

sigma orionis