Media Synchronization in 3D Tele-Immersion Applications: an architecture

Rufael Mekuria[1], Michele Sanna[2], Pablo Cesar[1]

[1]CWI: Centrum Wiskunde & Informatica
Amsterdam, The Netherlands

[2]Queen Mary, University of London
London, UK

R.N.Mekuria@cwi.nl, michele.sanna@eecs.qmul.ac.uk, p.s.cesar@cwi.nl

Abstract—**This article discusses the challenges ahead for assuring media synchronization in 3D tele-immersion applications. The discussion is based on an architecture and use cases that are defined in the European FP7 project REVERIE. The architecture allows capturing, transmission, and rendering, in real-time, various types of 3D media streams (e.g., geometry, movements of the participants, 3D audio), with the final objective of enabling immersive communication (and interactions) between remote users. For achieving the final goal, media synchronization is a key requirement. In particular, this article will focus on two types of synchronization: between real-time streams, and between downloaded and real-time streams. The first case refers to the synchronization needed between the different real-time captured media (e.g., 3D audio and visual streams). The second one aims at synchronizing downloaded content (e.g., 3D models) and the media captured in real-time. The solution reported in this article is implemented as a novel real-time streaming engine that can handle various types of 3D media streams. Moreover, based on global timestamps, the engine can provide synchronization support for a variety of scenarios.**

Keywords—Mixed Media, 3D Tele-immersion, Real-time communcation, Media synchronization

I. INTRODUCTION

Tele-conferencing systems enable participants in different locations to share a common experience, where specially designed room tables and real-time high-definition video increase the feeling of proximity. The principal advantage of current-generation tele-conferencing systems is that participants can talk to each other as if they were in the same location. The shortcoming is that participants cannot collaboratively perform a task together: they remain captives in a 2-D screen projection.

The next challenge in tele-presence is tele-immersion, which will enable individuals that are geographically distributed to interact naturally with each other in a shared 3D synthesized environment. Where tele-conferencing allows participants to share a common space, tele-immersion allows them to share an activity.

3D tele-immersion pushes the limits of current infrastructures because of the high volume of synchronized data that needs to be transmitted in real-time between different locations. Moreover, current limitations will be a blocking factor when tele-immersion becomes a widespread technology - with households equipped with such technology. Current research [1] is starting to provide valuable results in specialized 3D capturing systems, real-time data transmission, and advanced rendering technology. Still, there are a number of challenges that need to be considered.

This article focuses on two main challenges: real-time streaming and synchronization. First, we will introduce an architecture that allows for efficient capturing, transmission, and rendering of 3D media streams. The architecture provides support as well for downloaded 3D content, such as representations of avatars and virtual worlds. Then, based on the requirements imposed by a number of use cases, further details about the envisioned streaming and synchronization engine will be reported.

While research in the past has provided some solutions for streaming geometric objects, none of them handled the critical real-time constraints imposed by 3D tele-immersion. In these types of infrastructures, synchronization between media streams is particularly challenging, as the pipeline for visual data introduces large delays compared to

ISBN - 978-90-5986-410-8

the pipeline for immersive audio. Moreover, current solutions intended for video (e.g., RTP/RTSP or MPEG-TS) are not suitable for captured geometric representations, unless they are extended with support for specific 3D graphics codecs. The remaining of the article will be dedicated to further detail our design decisions for solving these issues, enabling real-time streaming and synchronization of various types of 3D media streams: spatial audio, geometry and movement data. The use of live captured geometry is novel, fast implementations of 3D reconstruction algorithms with one or more commercial depth camera's as described in [2], make such approach realistic for low cost deployment in the future.

This article is structured as follows: Section II introduces the use cases, highlighting a number of requirements regarding synchronization in future 3D tele-immersion applications. The following section overviews the state of the art, indicating the specific contributions of this article. Section IV describes the proposed architecture that meets the novel requirements, advancing current solutions in the problem space. Finally, Section V focuses on the streaming and synchronization engine.

II. USE CASES

The REVERIE project has defined two use cases to showcase the technological innovations that will be developed. Each of the use cases poses a different set of synchronization challenges. The first use case addresses scalability, where many participants are interacting in a common space. In this case, avatars represent the participants and captured motions are used for modeling interactions. The second use case aims at high-detail reconstruction and rendering, but it is feasible for a lower number of participants. In the following sections we will detail the use cases and highlight the specific challenges regarding media synchronization impose by each of them.

A. *European parliament*

In the first use case (see Figure 1) many students will participate in a live debate that takes place in the European Union parliament. Avatars represent the students, while virtual autonomous avatars (i.e., computer controlled avatars responding to students behavior and emotions) represent the instructors. The main challenge in this use case is scalability; as the actions from the debaters (and instructions) need to be streamed live to a high number of participants. While the transmission problem can be handled by applying some form of application layer multicast / CDN, synchronization of this content between the participants is still a challenge. This use case requires of inter-destination media synchronization between them. Otherwise, situations of inconstancy or unfairness may arise in case the instructor raises a question that reaches some students before than others, the former will have more possibilities of answering first, or the debater streams is are not received at the same time .

Figure 1 Use case 1 European parliament

B. *Birthday Party*

In the second use case (see Figure 2) up to four people will interact in a birthday party. During the party, special events and games can happen. For example, the children may play rock, paper, and scissors. In this case, synchronization between the representations of the users (3D objects) can be evaluated for fainess and consistency as in the study performed in[8]. In particular, the implementation has to allow for inter-sender synchronization (different lags between objects coming from different senders) and inter-media synchronization (between different captured objects/representations). The major challenge will be on provide synchronization mechanisms for a very demanding visual pipeline that enables highly realistic representations to be captured, transmitted and renderered.

ISBN - 978-90-5986-410-8

Figure 2 use case 2 Birthday party

III. RELATED WORK

The following sections will overview related work. They highlight the novel challenges imposed by the use cases presented in the previous sections. In particular, we focus on previous solutions for representing 3D objects, for streaming geometry representations, and for synchronization in 3D tele-immersion environments.

A. *3D Representations*

When discussing what 3D video is, different researchers have different interpretations. For example, 3D stereo video, in cinema, introduces an artificial depth perception, but it only consists of a left and a right image. Free-view point video, on the other hand, allows for viewpoint navigation. In the past, Kang provided a useful categorization of the different types of 3D representations [4] from image-based to geometry-based methods for renderering. Figure 3 shows the categorization. Image-Based methods are similar to traditional video and use multiple interpolated views. Geometry-Based representations include triangle meshes and point clouds, and contain full geometric information of the scene. While previous works on 3D tele-immersion have used image-based methods or points samples, we focus on live captured geometry-based representations.

The reasons why geometry-based representations have been selected are: they are considered renderer-friendly; they allow to seamlessly integrate virtual worlds; they enable N-viewpoint rendering for stereo, multi-view, and free-viewpoint; they can possibly make use of efficient compression algorithms that have been developed in the graphics community.

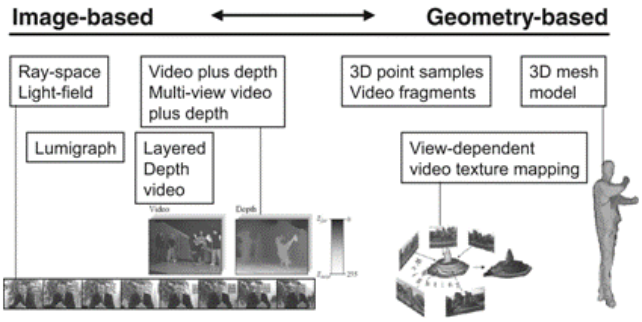

Figure 3 3D representations, from image based to geometry based, from [4]

B. *Geometry Streaming*

Streaming protocols for geometry-based 3D models are still in their infancy. They are not generically available and are not standardized. One example is the system developed by Li et al. [5], allowing efficient transmission of various types of (stored) compressed geometry-based objects. This system takes tackles possible effects of packet loss by providing specific properties on the encoding phase. It performs an offline analysis of the object, measuring possible degradations (L-2 norm of the distance between the original and the reconstructed surface, or Haussdorf distance). Unfortunately, such optimization is done offline, and it does now allow for real-time transmission of live captured geometry 3D data. This is an upcoming challenge that will be considered in the architecture presented in Section IV.

C. *Synchronization for 3D tele-immersion*

In the past, several works have studied 3D tele-immersion, and the associated synchronization issues. For example, Huang et al. [3] focused on how to stream live-captured immersive 3D (stereo) videos (background subtracted) between multiple sites. They also took into account the skew level between the different streams. They developed a scheme, called SyncCast, that allows video streams in an overlay (with multiple immersive) sites to be forwarded based on bandwidth, synchronization and latency requirements. The main novelty of this system is the introduction of synchronization logic in the network (the forwarding mechanisms is based on synchronization). Nevertheless, it is restricted to

ISBN - 978-90-5986-410-8

image-based representations, and does not deal with live-captured geometry.

IV. MEDIA PIPELINE

Figure 4 shows the media pipeline at the sender site. First, different types of media are captured: audio, motion (for avatars representation in use case 1), and visual data. For this article, we are particularly interested in the visual pipeline for transporting geometry data when capturing users. As users interact, this part of the pipeline is the most challenging in terms of real-time streaming and synchronization. After capture, the representation has to be encoded using an efficient compression method. Subsequently, the streams are packetized and adapted, so they can be sent over the network. Channel coding is applied to cope with lossy transmission over UDP. TCP could be used to avoid losses, but the end-to-end delay would be compromised. The data is then transmitted to the remote sites. Figure 5 shows the streaming module at the receiver site. The received packets are first buffered and synchronized. Then, re-construction of the stream takes place. The stream is decoded and rendered by a high-performance set of rendering components.

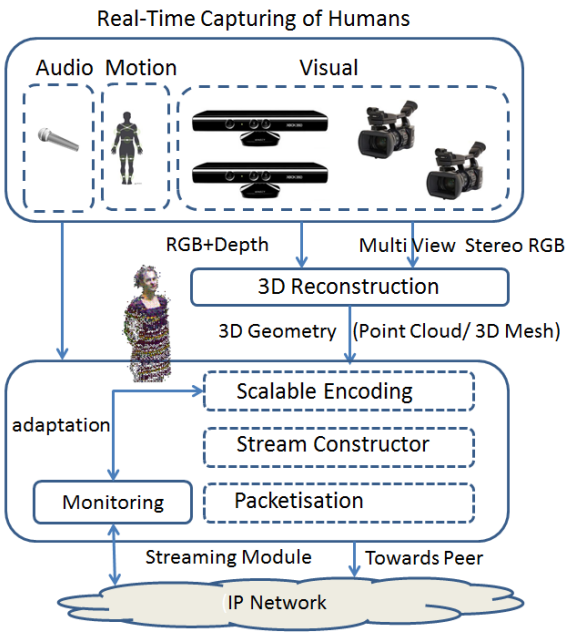

Figure 4 Media pipeline, sender site

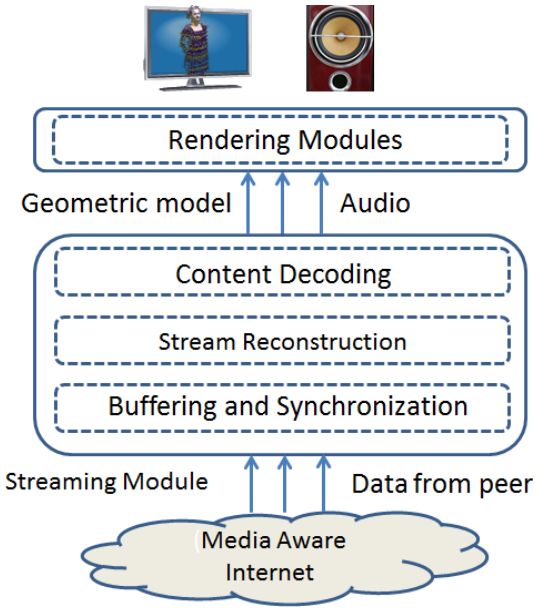

Figure 5 Streaming Pipeline, recipient site

V. REAL-TIME STREAMING AND SYNCHRONIZATION ENGINE

The architecture introduced in the previous section uses a number of state of the art technologies for 3D capturing and rendering. In order to enable networked communication, a specific engine has been designed. It meets the requirements imposed by our use cases, as it supports real-time streaming and synchronization of various types of 3D formats (geometry-based, audio). The next sections detail the decisions taken for designing such component.

A. 3D Representation

The streaming engine supports various types of 3D representation (including 3D audio). First, live-captured full geometry (triangle mesh/ point clouds) can be delivered. In particular, we use octree compression for point clouds [6] (available PCL[1]) and MPEG-4 SC3DMC coding [5] for triangle meshes. Second, MPEG BBA/AFX is used for transmitting motion commands, when a mesh (e.g., the avatar) has been downloaded in advance. Third, the engine can stream stored 3D videos (video plus depth) for extra material to be shown in the virtual world (e.g., a movie in the common space). Finally, in the coming future, the streaming engine will allow the delivery of a mixed representation approach that allows human reconstruction by using

[1] Point Cloud Library www.pointclouds.org

ISBN - 978-90-5986-410-8

hybrid geometry/image based methods by reconstructing from a previously initialized database.

B. *Channel and Network Coding*

In order to prime efficiency over data completeness, real-time streaming is performed over UDP, that unlike TCP does not allow for retransmission. Since the 3D compression mechanisms are not designed to work with information losses, the architecture has to provide some level of protection against packet losses. Channel coding provides such mechanisms at the transport level.

In the streaming engine we deploy channel coding on segments of data. These segments are consistently produced with a time window that allows keeping a target end-to-end delay (currently a per frame basis). For each segment, redundancy is added by encoding with increased information persistence (as long as the rate of received packets is equal to the coding rate, the source information can be entirely be decoded). This mechanism yields to rate-less inter-packet error correction, similar to fountain codes (e.g., raptor codes) as proposed in the IETF standard in [7].

Symbol-Based linear coding is performed to produce as many packets as needed to match and seamlessly adapt to the fluctuations of the available channel rate. The encoded representation is self-contained in each packet, allowing network coding via re-encoding, and mixing at intermediate nodes the information coming from different paths. This achieves a superior level of spatial coding diversity as well as resistance against drop-offs of specific paths, and an increased end-to-end throughput.

C. *Time client*

In order to support media synchronization and time coherence, the streaming engines includes a time client that regularly updates its time reference to a common NTP server. A virtual clock (application-specific) is used to avoid interferences with other system components. The values of this clock are the ones used for achieving media synchronization. The implementation uses the OS-specific format handled by the Boost C++ library[2].

D. *Support for Synrchonization: Sender and Receiver*

Both the sender and receiver provide support for synchronization. At the sender side, timestamps (based on a common clock) are generated, and later used by the receiver side. Delay estimation can be used for easing or enforcing encoding and scheduling parameters (rate estimation, priority encoding), so a target end-to-end delay is achieved. Moreover, by matching the encoding mechanisms to be developed with the data from the capturing process, superior coding speed and efficiency can be achieved.

At the receiver side, streams are aligned for ensuring synchronization between the different data streams (audio, geometry, and movement); and synchronization between users is performed, so a coherent scene comprising the actions at the right moment is constructed.

E. *Support for Syncrhonization: Renderers*

The architecture foresees a number of renderers, each of them specialized in different types of 3D data. The final implementation will integrate a spatial audio renderer; a rendering platform for triangle meshes and point clouds; and a renderer that supports body animation and facial animation parameters (FAP/BAP). Eventually, the system will include as well a renderer for hybrid human geometry/image based interpolated models. A key challenge, then, is inter-renderer synchronization. For solving this problem, each of the renderers provides feedback on the actual rendering times (timestamp) to the monitoring module in the engine. In addition, each of the renderers offers a number of quality options (e.g., shading, global illumination), allowing the engine to speed up rendering when necessary.

Finally, a simple API will be available to enable synchronization at the renderer. Nevertheless it is still not clear if such type of synchronization will be

[2] www.boost.org

ISBN - 978-90-5986-410-8

used, as the main bottleneck seems to be the delay introduced by the network.

VI. CONCLUSION

The 3D tele-immersion environment introduced in this article raises a number of challenges regarding synchronization that have not been widely studied in the past. This paper presents our real-time streaming and synchronization engine, addressing such challenges.

From a transmission perspective, the engine is capable of delivering (in real-time) geometry-based 3D representations. Efficient channel coding allows us to take advantage of real-time lossy protocols (UDP), while still being resilient to packet losses. This solution is more efficient than TPC, as it does not require extra overheads and delays. On the other hand, stored objects are delivered in a caching network using adaptive streaming over HTTP (MPEG-DASH Standard [9]).

The engine supports as well various kinds of synchronization. First, it assures that live captured media of different types (geometry, audio, and motion) is synchronized. Second, stored media (3D video plus depth, 3D objects in the scene) can be synchronized with the real-time captured media streams. This is achieved thanks to a number of mechanisms implemented at the sender (universal timestamps), at the receiver (buffering and alignment), and at the rendering sides (feedback and monitoring).

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-287723 (REVERIE project). We would like to thank the Reverie Consortium for the various contributions to the architecture, platform, and use cases.

REFERENCES

[1] Z. Yang, Bin Yu, Ross Diankov, W. Wu, and R. Bajcsy. Collaborative Dancing in Tele-Immersive Environment. ACM International Conference on Multimedia (MM) 2006.

[2] D. Alexiadis, D.S. Kordelas, G. ; Apostolakis, K.C. ; Agapito, J.D. Vegas, J.M. ; Izquierdo, E. ; Daras, P(2012). (WIAMIS), 2012 13th International Workshop on Image Analysis for Multimedia Interactive Services pp.1-4.

[3] Huang, Z., Wu, W., Nahrstedt, K., Rivas, R. and Arefin, A., 2011. SyncCast: Synchronized dissemination in multi-site interactive 3D tele-immersion. 2nd Annual ACM Conference on Multimedia Systems (MMSys '11)., pp. 69—80.

[4] S.B. Kang, R. Szeliski, P. Anandan: The geometry-image representation tradeoff for rendering, in: ICIP 2000, IEEE International Conference on Image Processing, Vancouver, Canada, September 2000.

[5] H. Li, M. Li, and B. Prabhakaran.(2006) Middleware for streaming 3D progressive meshes over lossy networks. ACM Trans. Multimedia Comput. Commun. Appl. 2, 4 (November 2006), pp. 282-317.

[6] B. Jovanova, M. Preda, and F. Preteux, “Mpeg-4 part 25: A graphics compression framework for xmlbased scene graph formats”, in Signal Processing: Image Communication - Special issue on advances in three-dimensional television and video, vol.24, pages 101 – 114.

[7] Kammerl, J.; Blodow, N.; Rusu, R.B.; Gedikli, S.; Beetz, M.; Steinbach, E.; , "Real-time compression of point cloud streams," *Robotics and Automation (ICRA), 2012 IEEE International Conference on* , vol., no., pp.778-785, 14-18 May 2012.

[8] Luby M., Shokrallahi, A. Watson M., Stock Hammer T. RFC 5053 Raptor Forward Error Correction Scheme for Object Delivery (2007)

[9] Huang, Z.,Arefin, A., Agarwal,P., Nahrstedt,K., and Wu, W. 2012. Towards the understanding of human perceptual quality in tele-immersive shared activity. In proceedings 3rd Annual ACM conference on Multimedia Systems (MMSys '12). pp 29-34.

[10] ISO/IEC 23009-1 -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats(2012).

ISBN - 978-90-5986-410-8