Streaming And Presentation Architectures For Extended Video Streams

Emmanouil Potetsianakis
LTCI, Télécom ParisTech, Université Paris-Saclay
75013, Paris, France
potetsianakis@telecom-paristech.fr

ABSTRACT

Mobile devices able to capture multimedia streams (audio and video) have now the capability to produce numerous associated data (GPS, Gyroscope, etc.). The real-time exploitation of these *extra-data*, in different application scenarios, raise the issue of integrating these data in existing broadcasting architectures. These architectures are designed to work with the audiovisual data and should be extended to match the extra-data requirements (processing time, bandwidth, visualization techniques etc.), and be able to adapt to their environment (network, device) accordingly. The major part of the applications utilizing these data are developed in an "ad-hoc" way, without taking all of their characteristics in account, and require complex maintenance and update efforts in case new data types or new network/device support is introduced. Our study is focused on optimizing the present distribution and presentation architectures for Audiovisual (AV) content, in order to find an efficient way to represent the associated data.

ACM Classification Keywords
H.5.1. Information Interfaces and Presentation (e.g. HCI): Multimedia Information Systems; H.3.4. Information Storage and Retrieval: Systems and Software; D.2.12. Programming Techniques: Interoperability

Author Keywords
Synchronization; Meta-data; Media Orchestration; GPS; MS-Kinect

INTRODUCTION

Over the past years, the process of video recording has evolved from an on-occasion effort to an everyday commodity. The decrease in cost and size of digital cameras has driven their embedding in various "objects" - from mobile phones and tablets, to drones and cars - that also carry several other sensors (GPS Receiver, Motion Sensors, Light Sensors etc.). Moreover, dedicated devices, using novel types of sensors (Kinect, Myo, MindWave, Leap Motion, etc.) were introduced to the commercial market, that either feature a camera, or are frequently utilized jointly with one.

Despite many developments in the areas of acquisition and presentation of these data, their use in real-time presentation contexts and their use in complementing the Audiovisual (AV) applications is sparsely covered in literature. A quick review shows two classes of methods: methods related to the transport of AV content (such as adaptive streaming over HTTP) and methods related to data fetching (eg AJAX) [4]. These methods were not designed to work together, and multiple channels of communication are often necessary in the same application. Moreover most of the work focus in this area is on audio-visual broadcast using data related to the enforcement [9], or on very specialized application cases [7] leading to ad-hoc architectures that are difficult to reuse. Standardization efforts of sensor and actuator data formats were conducted in the 2000s by MPEG via its MPEG-V initiative [13]. We aim on using the aforementioned relevant work, to design a processing chain, from recording, to transmission and representation, for sensor *extra-data*[1] and multimedia data (audio, video).

BACKGROUND

Due to the plethora of possible video-centric applications, we created a loose classification method, in order to be able to describe relevant systems. We used for the classification the following three criteria:

- **Device Type**: Common, Specialized, Network
- **Content Availability**: On-demand, Live, Real-time
- **Synchronization**: Loose, Strict, Critical

First, according to the *Device Type*, we can have *common* video-recording devices (such as cameras and mobile phones) that also support other data; *specialized* (such as Depth cameras and BCIs) that support video-recording or used in conjunction with cameras; or several connected devices forming a *network* - usually supporting multiple modalities (Body Sensor Networks, or e-Health Points for example). The technical incentive for these classes occurs from the number of potential different streams per class, as well as differences in modalities and their data rates. Also, as the number of devices grows in the network case, so does the provision required for inter

[1] also referred as accompanying data, or meta-data

TVX '17 Adjunct, June 14-16, 2017, Hilversum, Netherlands
© 2017 ACM. ISBN 978-1-4503-5023-5/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3084289.3084298

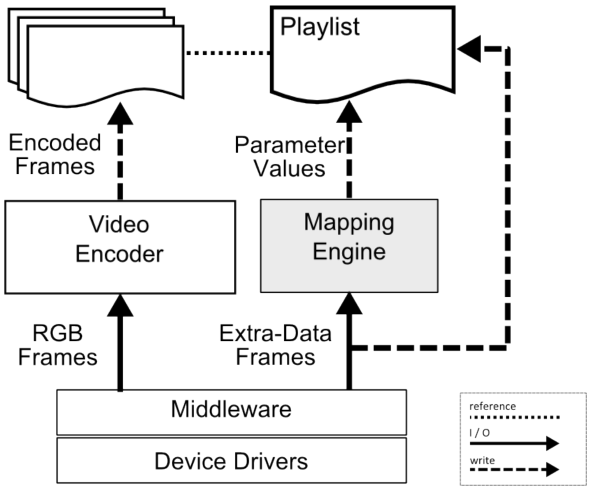


Figure 1. Server-side Architecture Overview

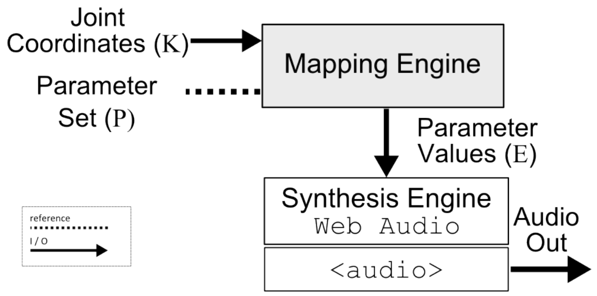


Figure 2. Audio Map-based Engine

and intra stream synchronizations - which are usually handled in firmware or middleware on the single-device case. This scalability synchronization issue occurs because various synchronization algorithms rely on bounded delays - that change as streams and network overhead is added, or global clocks that have to be in sync [6].

Regarding the *Content Delivery* delay requirements, we consider *on-demand* delivery, where the content is available after the recording is over, *live* scenarios when the content is transmitted during the recording process, and a *real-time* case in which the recording-to-delivery delay is as short as possible. Finally, on inter-bundle (between producers/devices) and inter-stream (between streams[2]) *Synchronization*, we identify *loose* (asynchronies over 500ms), *strict* (asynchronies between 100ms and 500ms) and *critical* (asynchronies bellow 100ms) classes. Our synchronization classification is based on principles derived from previous works focused on Interactive TV [3], standards [14] and generic synchronization principles and techniques [5].

SYSTEM ARCHITECTURE

Following the State of The Art review, from which we created the aforementioned classification system, we designed a high-level architecture schema for relevant client-server applications. The server part is responsible for gathering the Video and Extra frames (from the recorders), applying any processing - if needed, and sending them to the client, which in turn is responsible for the synchronous playback of the incoming streams. The distinct parts of the server, as described in our architecture, are shown in Figure 1. In that scenario we consider a single capture device (e.g. Kinect); otherwise a synchronization layer might be needed, responsible for inter-stream synchronization between the extra-data and video streams. The captured video frames are encoded, while the extra-data might be subject to a similar processing (in the drawn case an example "mapping"). Then they are prepared for transmission, in this case, by referencing the encoded Video frames, the Extra-Data Frames and the mapped Parameter Values to a playlist, and transmitted. In order to test this system architecture, we studied three different applications.

Kinect Audio

The first application scenario we examined uses the Kinect as input for an audio performance with accompanying dynamically-generated visualizations [11]. We implemented a server-client version where the content authoring takes place in the server side, while the data is transmitted on the client for a parameterizable rendering of the output [12]. The server-side architecture is the same as in in Figure 1, where "Extra-Data" frames are the Joint Coordinate frames and Mapping Engine uses a Parameter Map, also send to the client.

Figure 2 shows the Audio Engine architecture, located both in the server, for output monitoring, and on the client, for consuming the performance. In the server, where the Audio engine is used for monitoring purposes, Joint Coordinates (K) come directly from the device middleware, the Parameter Set (P) is provided by the producer and the Parameter Values (E) are the product of the Mapping Engine. On the client side, where the Audio engine is used to render the output, all of the aforementioned data (K, P, E) arrive from the network. Then, if the user wants to modify the performance output, changes can be made to elements of P, otherwise, for an exact reproduction of the output, the mapping is bypassed and the incoming Parameter Values E are directly send to the Synthesis Engine. The visualizations engine uses a similar mechanism, with the difference that if no visualization output is desired, the user can switch to the received RGB frames of the performer.

With this application we tested our architecture and were able to provide a modular multimedia system, while achieving inter-stream synchronization, by maintaining the intrinsic timing provided by the device middleware. Regarding the technical specifics of the implementation, on the server side, for H.264 encoding the RGB frames we used the ffmpeg library[3], while on the client the audio engine was build on the `Web Audio` API (`<audio>` HTML element). This solution is compatible with the MP4Box tool of the GPAC framework [8] for packaging the streams in MP4 containers, for an easily dis-

[2]in this paper, we do not differentiate inter-media synchronization, which is the synchronization of streams of different modalities, from inter-stream synchronization

[3]F Bellard, M Niedermayer - Available from:http://ffmpeg. org

tributed dataset, used in pair with the MP4Box.js library[4], for extracting and analyzing the contents in the browser [2].

Spatiotemporal Video Navigation

Another scenario we are currently testing and is based on the same architecture outline, uses the Orientation/Location sensors information of a mobile phone to provide spatial video navigation [4]. This platform was inspired from existing works for identifying videos according to the recording location [10]. We implemented an Android application, responsible for the recording, a server processor that parses the data and interpolates the GPS points if needed, and a browser-based client for navigation as seen on Figure 3. The markers are placed on the recording coordinates (using the Location sensors) and the triangular shape is used to indicate the camera facing (using Orientation sensors). Because all of the data are timestamped with the video, the content consumer is able to navigate through the map with the video as it progresses, or select a coordinate of interest and skip to the relevant part of the video.

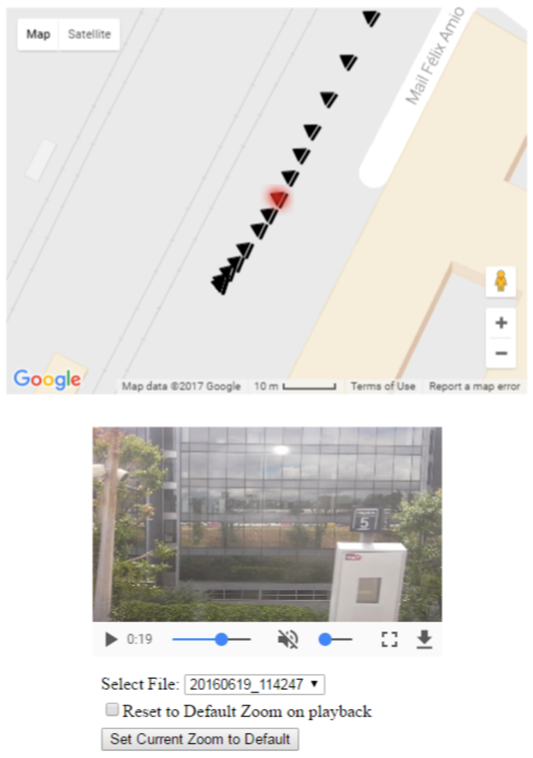
This implementation is used to study a different device class (i.e. *common*) and the challenges on simultaneous recordings. Also, the Content Availability and Synchronization classification criteria may vary according to the specific usecase. For example, in the video browsing example, where a user simply navigates through videos on a map, the inter-stream synchronization (between the location, orientation and video streams) is handled by the device. As for the inter-bundle synchronization, if the content consumer switches from one video recorded in the past to another (recorded from a different device at a different point in time), there can be significant drift of the two clocks without affecting the user experience, since there is no temporal overlap of the bundles. However, in the scenario that the content consumer wants to watch a specific event (e.g. a concert) via the submitted streams, the inter-bundle synchronization between content producers should be *strict* to minimize gaps and artifacts between stream switching. On top of the Synchronization restrictions of the scenario, if there is support for live event stream, the content availability delay must be as low as possible.

Sensor Network

Finally, we consider a scenario that the data source is the gateway of a network of sensors and cameras. Even though we are still in the assessment phase, this is a very challenging scenario, since the area span of the network and the number of devices gathering data, may vary from a few sensors on - or close to - the user [1], to sensors deployed over a city [16]. Also, even in the same network, the data rates and delays may vary significantly, especially in the case of multi-hop and/or energy-harvesting nodes [15].

The challenges are obvious in the environment monitoring example, where the nodes, in order to save power, transmit data sparsely and/or in inconsistent intervals. This delay grows as the size of the network increases, since data from the leaf nodes will take several re-transmissions to reach the gateway. As a result, on the gateway, data arrives with varying delays - according to the node transmission intervals and node-level

[4]https://github.com/gpac/mp4box.js/



Select File: 20160619_114247 ▾

Reset to Default Zoom on playback

Set Current Zoom to Default

Figure 3. Spatiotemporal video navigation client screenshot

in the network. However, when a camera is present, the video stream delay is orders of magnitude lower than the sensor data.

Assuming a wildlife monitoring example, where the cameras are on the gateways and temperature sensors are deployed throughout a forest, connected in a low-power wireless network with energy harvesting nodes, the temperature data might arrive several minutes late [15]. In case an event occurs (e.g. a fire), we would like to be able to see it on the video as soon as possible, without waiting for the temperature data to arrive. However, when having offline or time-shifted playback (e.g. to investigate the spread of the fire), we would like the video streams to be synchronized with the temperature measurement streams.

CONCLUSIONS AND FUTURE WORK

Our goal is to study the challenges for diffusion and efficient synchronized treatment of AV content with its associated data in different delivery architectures (broadcast, broadband, IP video, Adaptive Streaming), as well as different recording/processing architectures (embedded systems, mobile, web). In order to achieve this, we are identifying the constraints for different data types, devices and topologies, for which a first step is the classification method we devised. We design a model compatible with the different synchronization requirements, and validate this model by building demo applications and relevant simulations.

On other technical aspects, aside of the application testing, we are currently working on studying the fundamentals of an extended AV system. More specifically, our current work is focused on buffer management for multiple streams with many different modalities. By identifying the buffer behavior for different data types, we will design a model to scale according to the number of streams and the underlying platform. In order to achieve that, we are working on designing suitable low-overhead synchronization methods.

Finally, user studies will be designed and conducted, in order to examine the subjective performance of our findings. Also, since our work aims at integrating in the multimedia ecosystem (W3C, MPEG, IETF), the results will be discussed with the interested bodies, for reference and possible contributions.

ACKNOWLEDGMENTS

This thesis is supervised by Jean Le Feuvre and co-directed by Cyril Concolato and Jean-Claude Dufourd.

REFERENCES

1. Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor CM Leung. 2011. Body area networks: A survey. *Mobile networks and applications* 16, 2 (2011), 171–193.
2. Cyril Concolato, Jean Le Feuvre, and Emmanouil Potetsianakis. 2015. Synchronized Delivery of 3D Scenes With Audio and Video. In *Proceedings of the 20th International Conference on 3D Web Technology*. ACM, 245–248.
3. Ricardo Mendes Costa Segundo and Celso Alberto Saibel Santos. 2014. Systematic Review of Multiple Contents Synchronization in Interactive Television Scenario. *ISRN Communications and Networking* 2014 (2014).
4. Emmanouil Potetsianakis, Jean Le Feuvre and Cyril Concolato. 2016. Extended Video Streams for Spatiotemporal Navigation. In *The Graphical Web*. Exeter, Devon, UK.
5. Zixia Huang, Klara Nahrstedt, and Ralf Steinmetz. 2013. Evolution of temporal multimedia synchronization principles: A historical viewpoint. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1s (2013), 34.
6. Yutaka Ishibashi and Shuji Tasaka. 2000. A comparative survey of synchronization algorithms for continuous media in network environments. In *Local Computer Networks, 2000. LCN 2000. Proceedings. 25th Annual IEEE Conference on*. IEEE, 337–348.
7. Rose Johnson, Kenton O'Hara, Abigail Sellen, Claire Cousins, and Antonio Criminisi. 2011. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3323–3332.
8. Le Feuvre, Jean and Concolato, Cyril and Moissinac, Jean-Claude. 2007. GPAC: Open Source Multimedia Framework. In *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA '07)*. ACM, New York, NY, USA, 1009–1012. DOI: http://dx.doi.org/10.1145/1291233.1291452
9. Shih-Yao Lin, Yun-Chien Lai, Li-Wei Chan, and Yi-Ping Hung. 2010. Real-time 3D model-based gesture tracking for multimedia control. In *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 3822–3825.
10. Ying Lu, Hien To, Abdullah Alfarrarjeh, Seon Ho Kim, Yifang Yin, Roger Zimmermann, and Cyrus Shahabi. 2016. GeoUGV: User-generated mobile video dataset with fine granularity spatial metadata. In *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 43.
11. E. Potetsianakis, E. Ksylakis, and G. Triantafyllidis. 2014. A Kinect-based Framework For Better User Experience in Real-Time Audiovisual Content Manipulation. In *Telecommunications and Multimedia (TEMU), 2014 International Conference on*. 238–242. DOI: http://dx.doi.org/10.1109/TEMU.2014.6917767
12. Emmanouil Potetsianakis and Jean Le Feuvre. 2016. Streaming of Kinect Data for Interactive In-Browser Audio Performances. In *Proceedings of the Audio Mostly 2016 (AM '16)*. ACM, New York, NY, USA, 258–265. DOI:http://dx.doi.org/10.1145/2986416.2986438
13. Christian Timmerer, S Hasegawa, and SK Kim. 2009. Working Draft of ISO/IEC 23005 Sensory Information. *ISO/IEC JTC 1/SC 29/WG 11 N* 10618 (2009).
14. M. O. van Deventer, H. Stokking, M. Hammond, J. Le Feuvre, and P. Cesar. 2016. Standards for multi-stream and multi-device media synchronization. *IEEE Communications Magazine* 54, 3 (March 2016), 16–21. DOI:http://dx.doi.org/10.1109/MCOM.2016.7432166
15. Liviu-Octavian Varga, Gabriele Romaniello, Mališa Vučinić, Michel Favre, Andrei Banciu, Roberto Guizzetti, Christophe Planat, Pascal Urard, Martin Heusse, Franck Rousseau, and others. 2015. Greennet: An energy-harvesting ip-enabled wireless sensor network. *IEEE Internet of Things Journal* 2, 5 (2015), 412–426.
16. Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of things for smart cities. *IEEE Internet of Things journal* 1, 1 (2014), 22–32.