View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by ZENODO

Creating and broadcasting video-based multi-platform experiences

Joan Llobera, Juan A. Núñez° and Sergi Fernández Langa¥*
** i2cat Foundation, joan.llobera@i2cat.net*
° i2cat Foundation, juan.antonio.nunez @i2cat.net
¥ i2cat Foundation, sergi.fernandez@i2cat.net

1. Introduction

The majority of TV consumers now watch TV programs in a multi-display environment [1]. Second screens –most often smartphones - are generally used to check information not directly related to the events in the TV content being watched. Broadcasters have tried to orchestrate these different platforms, and there is reason to believe this contributes to user engagement [2]. However, their success has been limited. This might be caused, at least in part, by the different formats of content being used: mobile apps show graphics and text similar to web content, while TV renders a continuous stream of audiovisual content. The arrival of virtual reality displays to the living room further increases the need for consistent experiences across displays.

ImmersiaTV[1] is an European H2020 Research and Innovation action that is redesigning the production and delivery process to offer a new form of broadcast multi-platform video. This novel form of multi-platform video is based on the following assumptions:

1. The arrival of head-mounted displays (HMD) to the living room further requires the delivery of content that is consistent across displays, giving freedom to the end-user to engage with content in one or another device. This means that content must be adapted to the specificities of each device, both in terms of content format and interactive input (or lack of interaction, for the case of the TV), but the end-user should be free to adopt one or another device at different moments of the program being broadcasted.
2. This new form of broadcast multi-platform video must seamlessly integrate with and further augment traditional TV and second screen consumer habits. For TV, this means that content can be consumed simply by sitting on the couch and watching, without further input, and that the audiovisual language used follows established conventions. For tablets and smartphones, it means that user input and social media integration must work seamlessly with the specificities of each device.

To address these requirements, in the first year of the project we have designed and implemented an end-to-end production, delivery and rendering pipeline for offline content production which specifically addresses these needs. In this short article we outline the design principles, production and delivery methods as well as the tools used and developed for this purpose.

Section 2 outlines the design principles adopted, and related academic work. Section 3 introduces the elements constituting the ImmersiaTV end-to-end pipeline and, where there exist, commercial alternatives. Section 4 introduce our main conclusions and further planned work within ImmersiaTV to address some of the limitations detected. The content examples are adapted from the first pilot of ImmersiaTV.

2. Design principles and related work

2.1 Synchronous multi-platform playout

To create content for all devices, we need to create content that is adapted to each of them, and play it synchronously [3]–[5] . To play synchronized content, we have adapted emerging standards[2] and Gstreamer's version of the Precision Time Protocol (IEEE 1588)[3], as done, for example, in [6]. We have also embraced the use of omnidirectional video for HMDs and smartphones, in order to allow the user to visualize the scene in different directions. In other terms: the audience is still able to watch TV sitting on their couch, or tweet comments about it. However, the audience can also use immersive displays to feel like being inside the audiovisual stream, or use tablets and smartphones to explore these omnidirectional videos, or even, in the future, to zoom in, or share portions of it through social media.

---

[1] www.immersiatv.eu
[2] http://www.etsi.org/deliver/etsi_ts/103200_103299/10328601/01.01.01_60/ts_10328601v010101p.pdf
[3] https://gstreamer.freedesktop.org/data/doc/gstreamer/head/gstreamer-libs/html/GstPtpClock.html

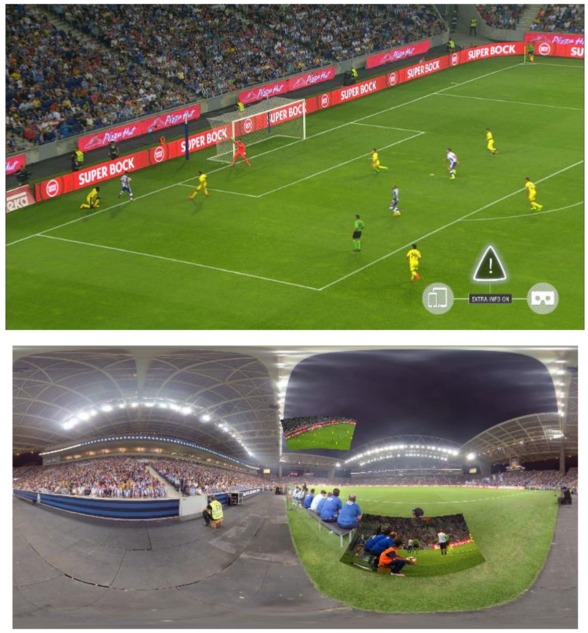


Figure 1: Top: an image typical of a traditional TV showing a football match. An insert informs the consumer that content is also available for tablets and HMDs. Bottom: a capture of an omnidirectional video with inserts of traditional cameras. This content is delivered synchronized with the main TV stream. Image courtesy of Lightbox (www.lightbox.pt).

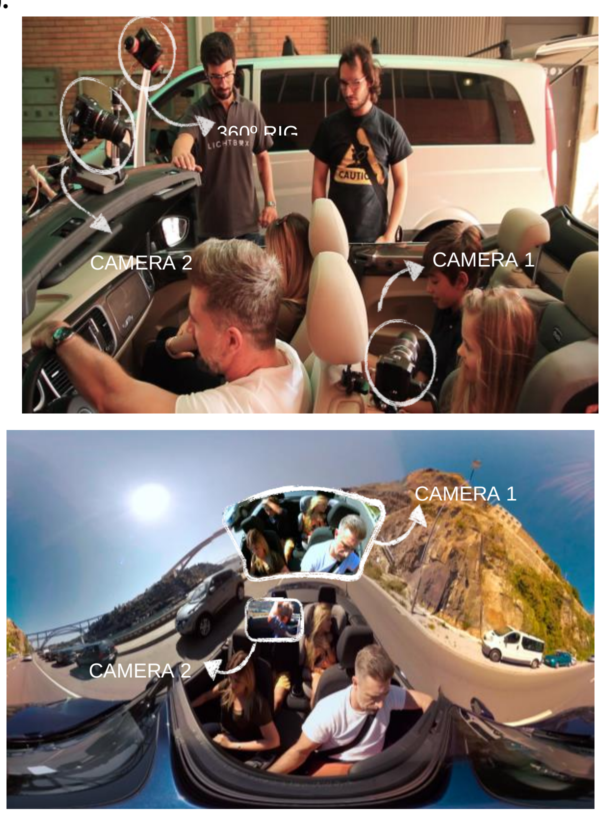


Figure 2: Top: a camera setup to record traditional and omnidirectional video simultaneously. Bottom: a schematic diagram of possible directive inserts located within the omnidirectional video. Image courtesy of Lightbox (www.lightbox.pt).

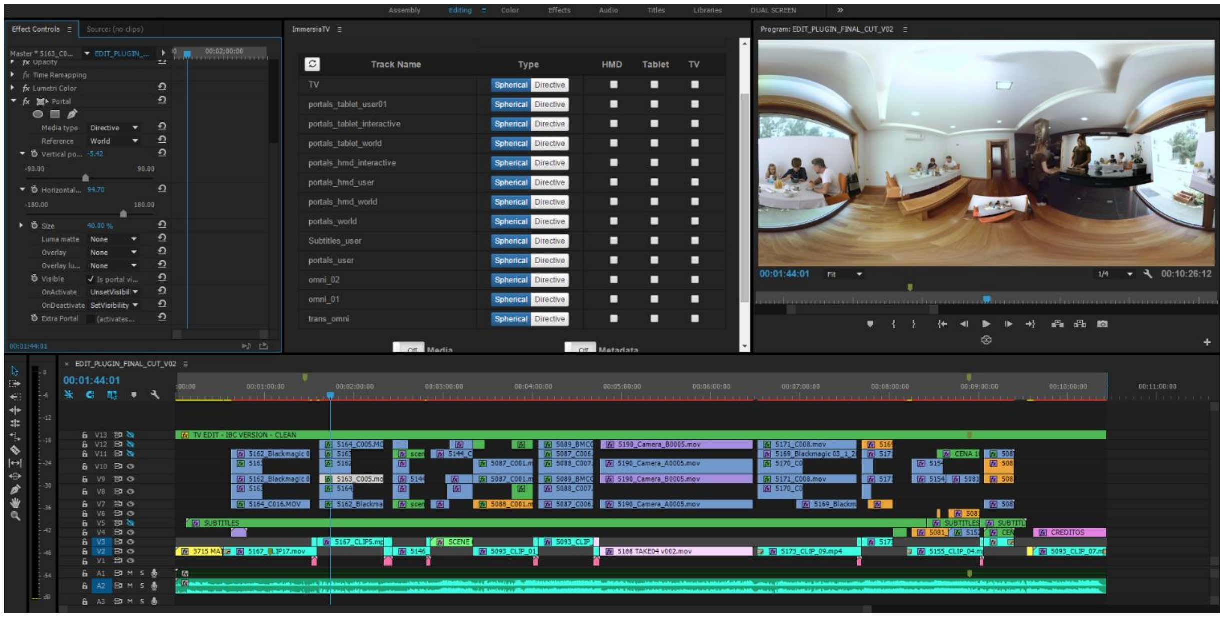


Figure 3: The Adobe Premiere ImmersiaTV panel, shown at the center, allows defining omnidirectional and directive (i.e., traditional) tracks, as well as which devices does each track target. The inserts added to the omnidirectional view, shown at right, can be edited with the ImmersiaTV Portal Effect, whose widgets are shown at the left. Image courtesy of Lightbox (www.lightbox.pt).

2.2 Portals

The idea of portals is inspired from the homonymous and famous videogame Portal.[4] In the context of streaming omnidirectional video, we introduce the idea of portals as video inserts that can be rendered in the HMD. These portals can be portions of other omnidirectional videos, which allows introducing basic interactive storytelling techniques such as scene selection or forking paths. These portals can also be inserts of traditional or directive videos. These traditional video inserts allow reintroducing classical audiovisual language that is not possible to render solely with omnidirectional videos, such as close-ups, slow motion, shot-countershot, etc. (see also Fig. 2). These strategies will not avoid the necessary precautions needed for shooting omnidirectional video [7], but they offer more options to integrate classic audiovisual conventions, particularly in a TV-centered consumption context..

3. An end-to-end pipeline

Designing and implementing a broadcast audiovisual production chain is challenging due to the diversity of processes, technologies and production practices that it requires. In this section we outline the main solutions, either adopted or implemented, for our purpose, with examples from the first pilot of ImmersiaTV.

3.1 Capture

The creation of content that is both omnidirectional and traditional requires shooting simultaneously in both content formats. Preliminary tests with separate shootings for omnidirectional and traditional cameras revealed it was unfeasible to synchronize two different shootings, even when the actors in the scene were repeating the same actions. The solution found by the production team was to use two *BlackMagic Micro Studio Camera 4k* micro-cameras for the traditional shooting, which could be hidden or, if visible, removed in post-production with a reasonably small amount of effort. This combined with an omnidirectional capture rig composed of 6 GoPro 3 Black Rig cameras allowed capturing simultaneously traditional and omnidirectional footage. However, for a joint shooting, we must address the fact that omnidirectional cameras capture the whole visual field, and therefore would show the traditional camera and

[4] http://store.steampowered.com/app/400/

the film crew behind it. This is not problematic for sports or music events, but it goes strongly against the conventions of fiction or documentary.

3.2. Edition

Dedicated stitching tools such as video-stitch studio by Video-stitch, or Autopano by Kolor, allow stitching video footage captured with camera rigs in order to create omnidirectional video. Tools for omnidirectional video edition, such as CaraVR[5] and Mettle's suite[6] allow further post-production. However, we are not aware of an editing tool targeting synchronous rendering across devices. To address this fact, we have designed and implemented a plugin for Adobe Premiere. The ImmersiaTV plugin shown in figure 3 allows defining the inserts that are placed within an omnidirectional video. It also allows selecting which tracks should be rendered in each of 3 possible devices (TV, tablet or HMD). It works both with Mac and Windows, and we have validated that, after going through a tutorial, can use it to create multi-platform content. Further work should refine and expand the possibilities to introduce interactive capabilities, in order the user's input affects the media being rendered or other aspects of the experience.

3.3 Distribution

The media encoding uses readily available and methods, concretely H.264 and AAC encoding, and adaptive bitrate streaming based on MPEG-DASH (ISO/IEC 23009-1:2014). Encoding is implemented as a cloud service, running on a Linux server using the Dockers virtualization tool as well as MP4Box from Gpac's MP4Box for MPEG-DASH multiresolution encoding[7].Video decoding uses the Gstreamer library[8].The additional metadata required for playout, which relates audiovisual streams with devices (i.e., allows selecting different streams for TVs and tablets), as well as to define interaction and media orchestration requirements, follows closely the format of MPEG-DASH manifests, and its XML specification is publicly available.[9] Content publication is performed through a custom built website which allows triggering media conversion, as well as monitoring progress on media encoding and publishing content generating a list of content parsed at the delivery stage.

3.4 Delivery

Metadata parsing is done with a custom parser, which also generates the appropriate geometry and provides the DASH player with the appropriate DASH manifests. Delivery of omnidirectional video requires integrating the user's input with the experience, in order to select the right portion of the omnidirectional view. We are currently exploring two different software architectures for media playout. A first version is based on a custom integration of Gstreamer and the Unity3D game engine which we have released open source[10] A second version is based entirely on web technologies, and content is rendered within a browser. We are using custom code combined with three.js[11] for the geometry generation and shaka-player[12] for DASH rendering.

We have not yet completed systematic evaluation of these implementations, initial benchmarking tests shows both players, when executed on a Samsung Galaxy s6 with android 6.0 can play one 4K video. However, the additional processing power required for metadata parsing imposes reductions on the playout quality. In addition, synchronization across devices necessarily imposes seeking through a video stream being consumed, which becomes slower at higher rates. Rendering for mobile-based HMD, either cardboard or Samsung GearVR, also imposes a double rendering process (one for each eye), which further decreases performance. Therefore, in practice we can currently reproduce synchronously video up to 4096x2048, bitrate of 50 Megabits per second (Mbps) and 30 frames per second on a Samsung Galaxy S6, but this resolution drops to 1024x512, bitrate of 2.3 Mbps and 25 frames per seconds when VR rendering is required. The browser-based rendering, in practice, can reproduce up to 2048x512, with 12Mbps of bitrate and 30frames per second. It seems clear that the limits in the quality that we can deliver is determined by hardware processing load, rather than bandwidth limitations. Further work will, first, develop more precise performance tests and, second, implement strategies to improve the data throughput that can be processed with off-the-shelf hardware, particularly when different media streams need to be synchronized in one device and where the orchestration of these media streams is affected by the consumer's input.

---

[5] https://www.thefoundry.co.uk/products/cara-vr/

[6] http://www.mettle.com/

[7] https://gpac.wp.mines-telecom.fr/mp4box/

[8] https://gstreamer.freedesktop.org/

[9] http://server.immersiatv.eu/public_http/metadata/ImmersiaTV.html

[10] https://www.assetstore.unity3d.com/en/#!/content/59897

[11] https://threejs.org/

[12] https://github.com/google/shaka-player

4. Conclusions and further work

The availability of open source tools as well as the existence of APIs to develop plugins for proprietary editing software makes possible, today, to design, implement and demonstrate an innovative end-to-end broadcast pipeline from scratch in less than a year. For our case, it required, approximatively, 8 software engineers with different profiles, and a mean dedication of 6 months each. Further work will be needed to optimize the media quality delivered, as well as to develop examples of content exploring more exhaustively the possibilities in terms of interaction that such a platform offers. Further work to adapt this paradigm to live production will also require implementing live production tools, as well as integrating a live MPEG-DASH encoding service. Other aspects missing, which we would like to address in the future are: access control for content conversion and publication, tiling techniques to optimize bandwidth consumption [8], [9], as well as social media integration on tablets and smartphones to further integrate with existing consumer habits.

Acknowledgements

Developing an end-to-end pipeline requires competent and motivated programmers. In ImmersiaTV, we acknowledge the contributions of Szymon Malevski and Wojtiech Kapsa, from PSNC, as well as Ibai Jurado, Isaac Fraile, David Gómez, Einar Meyerson and Juan Gordo from i2cat Foundation. David Cassany and Dr. Xavier Artigas, formerly at i2cat Foundation, also contributed in the early stages of the project. Pau Pamplona from i2cat Foundation and Maciej Glowiak from PSNC also helped with project management.

References

[1] T. Neate, M. Jones, and M. Evans, "Interdevice Media: Choreographing Content to Maximize Viewer Engagement," *Computer*, vol. 49, no. 12, pp. 42–49, 2016.

[2] R. C. Nee and D. M. Dozier, "Second screen effects: linking multiscreen media use to television engagement and incidental learning," *Convergence*, p. 1354856515592510, Jul. 2015.

[3] M. Montagud, F. Boronat, H. Stokking, and R. van Brandenburg, "Inter-destination multimedia synchronization: schemes, use cases and standardization," *Multimedia Systems*, vol. 18, no. 6, pp. 459–482, 2012.

[4] B. Rainer and C. Timmerer, "Self-organized inter-destination multimedia synchronization for adaptive media streaming," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 327–336.

[5] V. Vinayagamoorthy, R. Ramdhany, and M. Hammond, "Enabling Frame-Accurate Synchronised Companion Screen Experiences," in *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*, 2016, pp. 83–92.

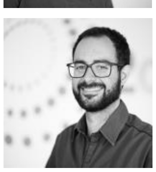
[6] A. Veenhuizen and R. van Brandenburg, "Frame accurate media synchronization of heterogeneous media sources in an HBB context," in *Media Synchronization Workshop*, 2012.

[7] A. Sheikh, A. Brown, Z. Watson, and M. Evans, "Directing attention in 360-degree video," presented at the International Broadcasting Convention / The Best of IET and IBC 2016-17, 2016, vol. 8, pp. 43–47.

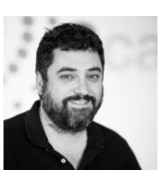
[8] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, "Tile Based HEVC Video for Head Mounted Displays," presented at the IEEE International Symposium on Multimedia (ISM), 2016.

[9] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "MPEG DASH SRD: spatial relationship description," in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016, p. 5.

Dr. Joan Llobera is an electrical engineer and a virtual reality scientist. He obtained a double diploma in Electrical Engineering (Universitat Politècnica de Catalunya and Télécom Paris) and holds 2 Masters, one in cognitive sciences (EHESS) and another in software (UPC). In 2012 he obtained a PhD at the Universitat de Barcelona on the topic of Stories in Virtual Reality and worked as a postdoc researcher at the Ecole Polytechnique Fédérale de Lausanne. In 2015 he joined the i2CAT Foundation.



Mr. Juan Antonio Núñez is a telecommunications engineer specialized in image and sound. He obtained a master in Audiovisual Communication Systems by the Universidad de Málaga. He worked six years for Tedial, a software developer for Media Asset Management and a Media IT solutions integrator. In 2016 he joined the i2CAT Foundation.



Mr. Sergi Fernández is a computer scientist and holds a Master in Artificial Intelligence by the Universitat Politècnica de Catalunya since 2007, obtained researching the field of Natural Language Processing. Since May 2008, he has given support to the coordination of i2CAT's Media & Internet Unit and led i2CAT's involvement in different national and European projects. Since 2012 he is the director of the Media & Internet Unit of i2CAT. He previously coordinated the European project TV-Ring (www.tvring.eu) and currently coordinates the H2020 Project ImmersiaTV.