



**Universitat de les
Illes Balears**

360° Hypervideo: an interactive documentary around the refugee crisis in Greece

MSC Candidate

Antoni Oliver Tomàs

A MSc thesis submitted to *Departament de Ciències Matemàtiques i Informàtica* of the University of Balearic Islands in accordance with the requirements for the degree of **Màster Universitari Enginyeria Informàtica (MINF)**

Author _____

Certified by _____

Toni Bibiloni Coll
UIB Master's Thesis Supervisor

Accepted by _____

Antonia Mas Pichaco
Director of Master on Information Technologies

7 September 2017

360° Hypervideo: an interactive documentary around the refugee crisis in Greece

Antoni Oliver Tomàs
Tutor: Toni Bibiloni Coll

Treball de fi de Màster Universitari Enginyeria Informàtica (MINF)
Universitat de les Illes Balears
07122 Palma de Mallorca
<antoni.oliver.tomas@gmail.com>

Abstract

We present an interactive 360° documentary around the migratory crisis in Greece. Omnidirectional content is recorded using a 6-camera array and rendered in an equirectangular projection and played later by an HTML5 web application. Interactive hotspots are placed on specific coordinates of the space-time of the production, introducing a connection of the viewer with the story by playing additional multimedia content. Usage analytics are recorded and will permit us to obtain metrics on the user behavior, like most popular areas or most viewed additional content, and detect possible usability issues.

Keywords: Hypervideo, 360° video, interactive documentary, 360° documentary.

1. Introduction

Over a million people have arrived in Greece since 2015. Refugees flee from war horrors and dreadful-armed conflicts. Babies, children and elderly people are especially vulnerable. Migrants arrive and will be still arriving while conflicts last¹. Proem-aid² is a group of emergency professionals who voluntarily help in Greece.

Equipped with a 6-camera array, the team recorded 360° footage in the refugee camp and, together with additional, regular video content, we present a use case for a novel format of interactive documentaries, letting the user to explore a 360° video and obtain additional information in the form of audiovisual content at their request.

We propose the following format for describing an interactive experience with a 360° video: a main video track plays as a panorama, allowing the user to pan the

scene and look at any direction. This track serves as a common thread for additional media that is linked to specific points in the space-time of the main track. In these moments, a marker is displayed in the scene, linking to a follow-up video of what is seen in the main track.

The users' actions are recorded in real-time by the player, which stores these micro transactions using the Experience API into a Learning Record Store in order to obtain metrics and analytics on the user behavior.

The Experience API (xAPI) [23] is a Representational State Transfer (RESTful) API in which any statement of experience (it can happen anywhere and on any platform) can be tracked as a learning record. A statement is a simple construct, represented in JavaScript Object Notation (JSON) format, used to track an aspect of a learning experience consisting of, at least, three parts: an actor, a verb and an object. A set of several statements, each representing an event in time, may be used to track complete details about any experience. The Learning Record Store (LRS) is the key part of the xAPI specification: a server responsible for receiving, securely storing, and providing access to learning records. There are several implementations of open-source LRS, like Brightcookie's lxHive³ or HT2Labs's Learning Locker⁴. A Learning Record Provider is a client that delivers data to an LRS, whereas a Learning Record Consumer is a client that accesses data from a LRS with the intent of processing that data.

This section includes a revision of the current state of the art in interactive multimedia and 360° videos, as well as an overview of the previous work. Section 2 details the creation of such media and the tools needed to make it interactive, play it and collect the users' actions. Section 3 introduces the initial results obtained from this work. Finally, the conclusion can be found in section 4, with some thoughts in the future.

¹ Adapted from <http://www.proemaid.org/en/lesbos/>

² <http://www.proemaid.org/en/>

³ <http://www.lxhive.com/>

⁴ <https://learninglocker.net/>

1.1. State of the art

A Hypervideo is a navigable stream of video that offers the viewer the possibility of choosing a path in the narrative combining several multimedia tracks through spatial and temporal links present in the media [1]. We extend this concept to support 360° media, including links around the sphere of the 360° panorama. Interactive documentaries have been analyzed in [2], encouraging users to explore and navigate the production, rather than simply viewing it.

An overview of omnidirectional video is introduced by Bleumers et al. in [3]. 360° multimedia belongs to the field of Virtual Reality (VR) [4, 5]. This field has traditionally made use of several gadgets and devices to create an immersive experience, like head-mounted displays (HMD) or the CAVE [6].

Despite this, there is an increasing trend to display 360° video in an interactive scene inside web browsers in desktop and mobile devices, given the recent adoption of this novel medium by popular online video platforms, like YouTube⁵ or Facebook⁶, that have begun offering immersive 360° video upload and visualization services. This is possible thanks to the ability to display a 3D scene to project the equirectangular video texture inside the browser itself, since the introduction of WebGL⁷, in 2011.

The current state of 360° video playback is shared between HMD like the Oculus Rift⁸, and web based navigation in video services like the ones mentioned before, featuring a desktop drag-and-drop interface and a sensor-enabled view for smartphones. Particularly, VR head-mounted displays were used to provide 360° educational lectures in [7], a drag interface for 360° hypervideo is introduced in [8], and more recently smartphones have been used to present Augmented Reality [9].

While the most known solution in the field of analysis of user data in online video is YouTube Analytics⁹, it is missing reports in the omnidirectional aspect of 360° videos. Facebook, on the other hand, is bringing narrative to the 360 video experience and enabling video publishers to obtain insight through heat maps of the most viewed portions of the 360° video field of view¹⁰. Several other linear video interaction analytics solutions have been proposed, most of them oriented at Massive Open Online Courses (MOOCs) [10–13]. Commercial software solutions, like Wistia¹¹ or LiveHEAT¹² provide graphically highlighted points of the video screen in which the user has changed the orientation, clicked or zoomed the visualization of the video through heat maps. In the literature, similar approaches are followed, with the purpose to minimize

motion sickness when playing these videos using a HMD [14], or to describe user selection in an spherical multi-touch setting [15].

1.2. Previous work

The concept behind this project is based on the Hypervideo platform, an interactive solution for interactive linear videos to be played in interactive TVs [16–18]. This concept was later extended to include a second screen controller, using a smartphone [19] and a usability experiment followed to compare the previous design with the use of a second screen [20].

We began to work with 360° videos in [21], where we introduced a 360-degree video player remotely controlled by a handheld device. The controller received pieces of information at specified moments, notably when a point of interest appeared in the video.

This last system evolved into a platform for interactive 360° videos, dropping the support of the remote controller and introducing an interface to pan the video and select the elements that appeared in it. The use of xAPI to record the users' actions is introduced and initial results are obtained. This work is attached in Appendix A.

Finally, from this platform we designed the interactive documentary, going back to our first experiments with hypervideos. This system features a revision of the creation tools and the player module was redone. This work is attached in Appendix B.

2. The interactive 360° documentary

In this section, we describe how this production was created. The process follows Figure 1: 360° video files from the filming process are stitched together using specialized software and then, using audiovisual editors are joined into a single clip. This file, plus multiple other videos are encoded into several resolutions and then a DASH manifest is generated for each set.

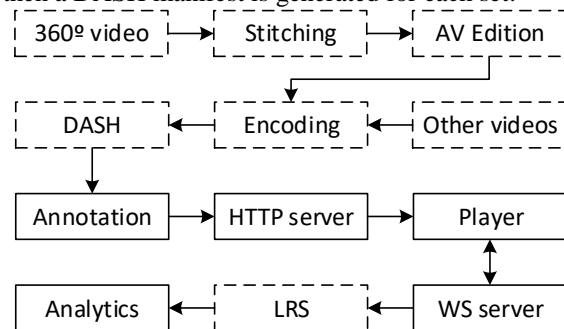


Figure 1: General overview.

⁵ <https://youtube-creators.googleblog.com/2015/03/a-new-way-to-see-and-share-your-world.html>

⁶ <https://facebook360.fb.com>

⁷ <https://www.khronos.org/webgl/>

⁸ <https://www.oculus.com/>

⁹ <https://www.youtube.com/analytics>

¹⁰ <https://media.fb.com/2016/08/10/new-publisher-tools-for-360-video/>

¹¹ <https://wistia.com/>

¹² <http://www.finwe.mobi/main/liveheat/>

These files are uploaded into the HTTP server through the annotation module, creating the links between these media files as a hypervideo. The player application obtains these files from the HTTP server and establishes a real-time communication with the Socket.IO¹³ WebSocket Server, that posts this data into the LRS to be retrieved by the analysis module. Both annotation and player tools are React¹⁴ applications, sharing some components. In the previous figure, dashed borders symbolize the use of external tools whereas modules developed for this platform are marked by solid borders.

The work done by the MSc candidate includes from the encoding of the video files up to the storage of xAPI statements in the LRS. The audiovisual files and the analysis module were responsibility of other members of the team. Please check Appendices A and B for further details.

2.1. Recording and editing

Two kinds of footage were recorded in the refugee camps in Lesbos and Piraeus, Greece: the 360° media and several linear clips. Four 360° scenes of approximately 90 seconds each were captured using a 6-camera array, placed on a stick. The media was stitched using Kolor's Autopano Video Pro¹⁵, obtaining an equirectangular clip.

```
ffmpeg -y -i vid.mp4 -strict -2 -c:a aac
-ac 2 -ab 128k -c:v libx264 -x264opts
'keyint=24:min-keyint=24:no-scenecut'
-b:v 1500k -maxrate 1500k -bufsize 1000k
-vf "scale=-1:720" vid_q1.mp4
```

Figure 2: Command used to encode a video file.

Since both the equirectangular and the linear clips were recorded using high bit rates, not easily supported by most network conditions, they were encoded into multiple bit rates using the ffmpeg command-line tool¹⁶ forcing a Group of Pictures of size 24, to ensure smooth quality changes. Figure 2 includes an example command to encode a video file at 720p.

```
MP4Box -dash 2000 -rap -frag-rap
-profile onDemand -out vid.mpd
vid_q1.mp4#video vid_q2.mp4#video
vid_q1.mp4#audio vid_q2.mp4#audio
```

Figure 3: Command used to generate a DASH manifest.

To enable adaptive streaming, we followed the MPEG DASH standard [22] and generated the MPD manifest files for each clip using the MP4Box command-line tool¹⁷. Figure 3 includes an example command to generate the manifest for two different qualities (q1, q2) of a single original clip.

2.2. Introducing the interactivity

The process of creating the interactive production from the main 360° video track and multiple linear video files is supported by a web application that generates the metadata needed to display the markers.

This annotation tool provides an interactive interface to edit the list of objects that appear in the main audiovisual content and setting their properties, most notably the list of key positions an object takes in the 360° video. The user is able to navigate the media to select a desired position in space and time to place one of these key positions. This application is designed and built with React, and makes use of a scene component to preview and place the key positions of the markers that is also included in the player.

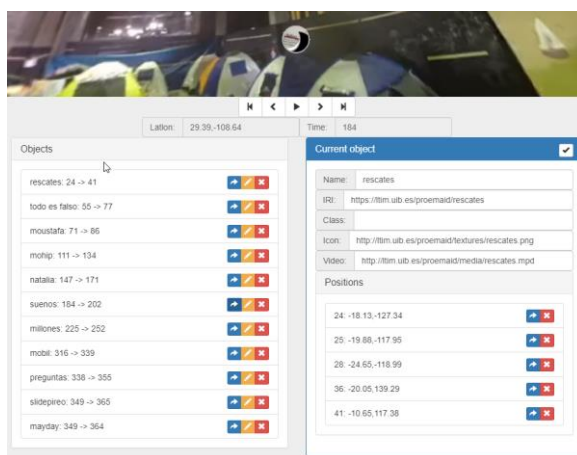


Figure 4: User interface of the annotation tool, featuring the markers in the scene, external video controls, the list of points of interest and the detail of a selected PoI.

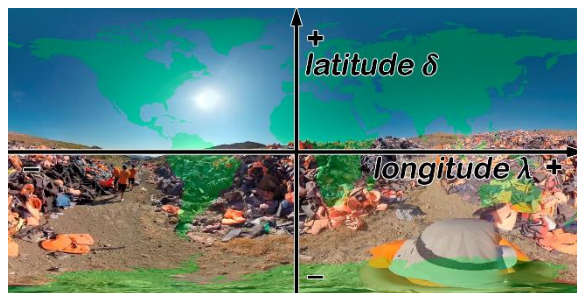


Figure 5: Notation of latitude and longitude in the equirectangular frame.

The representation of the positions of a given object is a sequence of at least two points in the space-time of the 360° video track, where t is the time in seconds; δ is the latitude in degrees from the equator of the scene; and λ is the longitude in degrees from the meridian zero of the equirectangular frame (see (1) and Figure

¹³ <https://socket.io/>

¹⁴ <https://facebook.github.io/react/>

¹⁵ <http://www.kolor.com/autopano-video/>

¹⁶ <https://www.ffmpeg.org/>

¹⁷ <https://gpac.wp.imt.fr/mp4box/>

5). An object that follows a simple trajectory may be described by as few as two key positions, but objects whose trajectory is erratic will need more positions to compensate for this.

$$(t_0, \delta_0, \lambda_0), \dots, (t_N, \delta_N, \lambda_N) \quad (1)$$

Additional metadata for each point of interest include a descriptive name, an IRI (Internationalized Resource Identifier) to identify that object, the IRI of the video that should play when selected and an optional image to customize the marker that appears in the scene. This information is stored in a JSON file in the HTTP server and retrieved by the player.

2.3. Playing the production in a web browser

Once the interactive 360° video has been defined, it can be played in a WebGL enabled browser. Following the same design as the annotation tool, the player is also a web application built with React, and it recycles the same scene component included in the editor to display the 360° video. An overview of the player application is given in Figure 6 and detailed next.

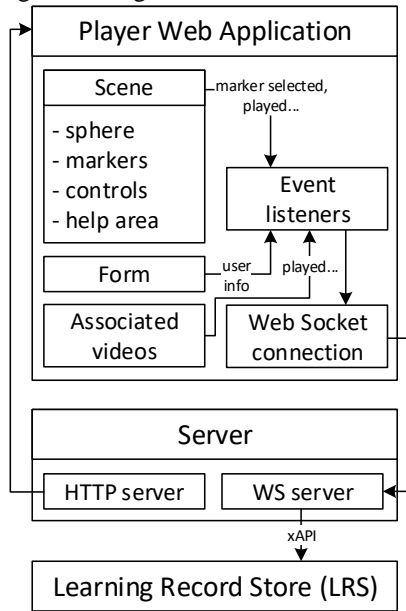


Figure 6: Architecture of the interactive 360° video player environment.

Before playing the production, a form prompts the user to share some personal information (gender, age, profession, country) with us, so we will be able to obtain some statistics by this information.

By listening these events, the player is notified when a marker is selected, so its associated video will play in a pop-up view, interrupting the main track playback. When this video ends or the user closes the pop-up view, the main playback continues.

A Web Socket connection is established with the server keeping track of these events, which are stored in a Learning Record Store (LRS) via the Experience API (xAPI).

2.3.1. Dynamically adapting bitrate

After encoding every video file at multiple bit rates and supplying the MPD manifests that detail the different options in quality for each clip, the player needs to adapt from one quality to the other dynamically. In the player side, the manifests are loaded by the DASH.js JavaScript player¹⁸ and, when the media starts playing, it automatically adapts its bit rate.

This process is done by creating a dashjs.MediaPlayer using the video elements for both the main 360° video in the Scene component as well as the ones in the Associated videos. In order to listen for play events in these videos listeners for dashjs.MediaPlayer.events have to be set up in that MediaPlayer instead of using the native <video> events.

2.3.2. The 3D scene

This component, shared by the editor and the player, houses a WebGL scene managed by Three.js¹⁹. The core of this scene is the sphere used to display the frames of the 360° texture provided by the main equirectangular video track. This component is provided the list of points of interest, from which circles are created in their respective positions, firing an event when selected. That position is calculated via linear interpolation in spherical coordinates using Ed William's aviation formula²⁰, adapted as following:

For a particular time t , $t_0 \leq t \leq t_1$, and knowing two key positions at t_0 and t_1 , $(t_0, \delta_0, \lambda_0)$ and $(t_1, \delta_1, \lambda_1)$, the position p_t in Cartesian coordinates for that marker in that time at a distance of R is obtained via (2).

$$f = \frac{t - t_0}{t_1 - t_0}$$

$$d = \cos^{-1}[\cos \delta_0 \cos \delta_1 \cos(\lambda_0 - \lambda_1) + \sin \delta_0 \sin \delta_1]$$

$$A = \frac{\sin(1 - f)d}{\sin d}$$

$$B = \frac{\sin f \cdot d}{\sin d}$$

$$p_t = R \begin{bmatrix} A \cos \delta_0 \cos \lambda_0 + B \cos \delta_1 \cos \lambda_1 \\ A \sin \delta_0 + B \sin \delta_1 \\ A \cos \delta_0 \sin \lambda_0 + B \cos \delta_1 \sin \lambda_1 \end{bmatrix} \quad (2)$$

To be able to detect when a marker is selected, intersections are looked for with a ray casted from the camera position to the selected position. Other events are fired when play conditions change, that will be used by the component that includes the scene, either the editor or the player.

¹⁸ <https://github.com/Dash-Industry-Forum/dash.js/>

¹⁹ <https://threejs.org/>

²⁰ <http://edwilliams.org/avform.htm#Intermediate>



Figure 7: User interface of the interactive player, featuring the markers in the scene, the help zone with its markers and the controls with its PoI appearances.

Additional parameters can be passed to the scene to enable two player-specific features: the controls to adjust playback of the 360° video, featuring a progress bar that hints the times when points of interest appear and a help area that displays again the markers that are shown somewhere in the scene, so the user can still see them if they are outside the camera's field of view. As an example, see Figure 7, where the first point of interest in the help area can be seen on the right of the scene, but the second one is outside the visible portion of the scene.

2.3.3. Usage tracking

The events fired by the scene are listened by the player and then forwarded via a Web Socket connection to the server. Instead of collecting all the data and submitting it in the end, this approach was chosen to permit us to visualize in real time the users' actions. Once this information is received by the server, it is stored into the LRS via xAPI.

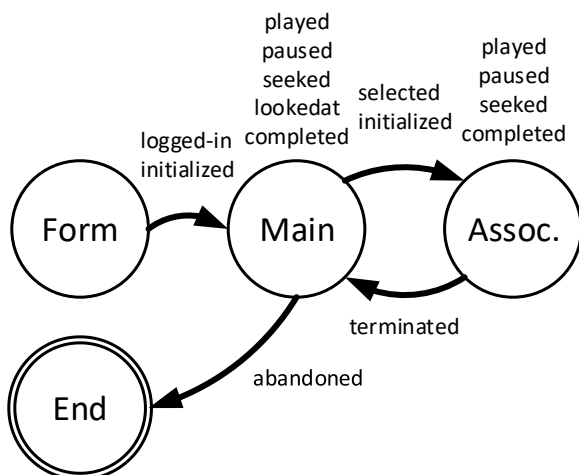


Figure 8: States of the player application, with events.

The vocabulary used in this application is based on ADL's video vocabulary²¹ and on the general xAPI vocabulary registry²². Additional vocabulary has been

created and submitted to the general registry to cope with the 360° nature of the application.

We will follow Figure 8 to explain the statements that are generated in the use of the player application, indicating the actor, verb, object, context and extensions used. For a more in-depth explanation, please refer Appendix C.

The actor for these statements is specified either with using their e-mail address, if provided, or a version 4 UUID as the name of a local account, to deal with anonymous users, stored in a cookie. The logged-in statement is sent to store the users' personal information after they submit the form. Just after that, when the main video loads, an initialized statement is submitted specifying the main video's IRI, creating a video session from its ID, according to the video vocabulary.

The played, paused, seeked and completed statements are used according to the video vocabulary, attaching the following video extensions: session-id, quality, screen-size, user-agent, volume, video-playbacksize and time (or time-from and time-to, in the case of a seeked statement, refer the video vocabulary). An additional extension is defined: orientation, containing an object with the current view's latitude and longitude.

Following the same design, we defined the lookedat verb that is submitted when the user changes the current view. Successive submissions are throttled to 250 ms so as not to flood the LRS. In this case, orientation-from and orientation-to are used instead of the orientation extension, following the design of the seeked statement with time.

The selected statement is sent to the LRS whenever a point of interest is selected through its marker, either in the scene or in the help zone. This statement includes the additional data present in the played statement as context extensions and the IRI of the point of interest as a result extension. From this moment, the associated video window opens and it begins to load. When it does, an initialized statement is generated following the previous behavior and still using the previous initialized statement ID as its session-id.

As it is natural, the user is free to interact with the associated video, and played, paused, seeked and completed statements will be generated much like previously specified, but the ID of the initialized statement of this associated video is used as the session-id and the orientation extension is unused.

Whenever the user closes the window of an associated video, a terminated statement is stored in the LRS, including the same additional data these previous statements would. This marks the end of the session of an associated video.

²¹ <http://xapi.vocab.pub/datasets/video/>

²² <https://registry.tincanapi.com/>

Finally, when the user closes the player, the Web Socket connection is closed and an abandoned statement is submitted, marking the end of the session.

2.4. The backend

To serve the web applications, host the media files and listen for Web Socket connections a Node.js²³ Express²⁴ server was developed.

For the HTTP server, routes are established to serve the HTML files that include the Javascript bundles generated by WebPack²⁵ from the React code. A cookie is set in every route to track the returning user. HTTP POST routes are set for the annotation tool in order to receive changes in the metadata that controls the hypervideo and wildcard routes serve static content, like videos, textures and WebPack bundles.

The Socket.IO Web Socket server listens to the events we listed before and submits the xAPI statements to the LRS.

Finally, we chose lxHive as our LRS and installed it in a Docker²⁶ container in a separate host.

2.5. Performing analysis

Storing the user actions in an LRS in the form of xAPI statements permits us to query this information in a standard way using a Learning Record Consumer. In this case, we developed a tool that lets us select multiple sessions (a session starts with their logged-in statement) and plot graphs from the interactions performed by the users.

Since this application is out of the scope of this document, we refer Appendix A for further details.

3. Results

A preliminary set of results has been obtained so far from the data collected in the LRS using xAPI. Knowing (almost) everything the user does with the application enables us to perform very accurate monitoring of their behavior. A simple example is given next to illustrate that the users' actions are stored in the LRS.

Figure 9 and Figure 10 contain two complementary graphs that display data from three different video sessions, represented in three different colors. These two graphs are complementary because the same information is represented in both graphs but with different coordinates, letting us to compare different sessions in the 360° dimension or the temporal dimension.

Figure 9 represents playback time (in seconds) versus longitude (in degrees). Dashed lines are superposed to help us know where the markers are displayed in the scene.

Figure 10 represents playback time (in seconds) versus relative timestamp (in seconds). The origin is relative to the timestamp of the initialized statement, to be able to coherently compare different sessions.

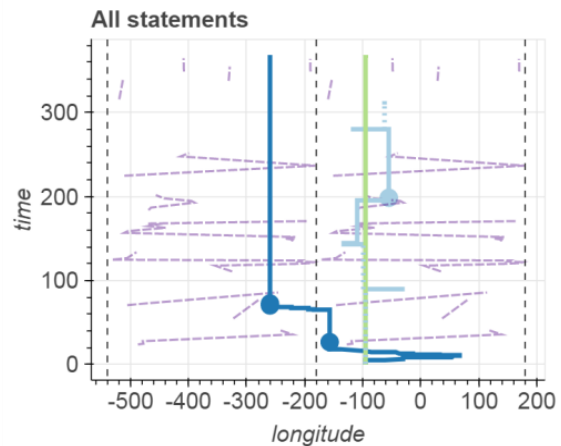


Figure 9: Longitude vs time graph.

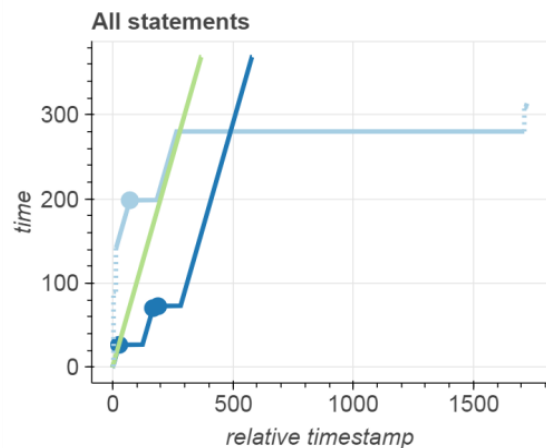


Figure 10: Relative timestamp vs time graph.

In both figures, solid lines indicate that the video track was played, was paused or was navigated. Dotted lines indicate that it was sought. Circles indicate that a marker was selected (and an associated video was played, partially at least).

4. Conclusion

The work presented in this document is an extension of the state of the art in interactive documentaries, introducing user-navigable 360° video and interaction in the form of additional videos that give further details about a specific topic. A use case was conducted with 360° and linear video footage in the refugee camps in Lesbos and Piraeus, which resulted in an interactive web application. Precise user behavior is logged in real

²³ <https://nodejs.org/>

²⁴ <https://expressjs.com/>

²⁵ <https://webpack.js.org/>

²⁶ <https://www.docker.com/>

time and analyzed to comprehend their decisions and navigation choices in this novel format.

Further enhancements are planned for the 360° hypervideo model, introducing questions or quizzes and for the video player, designing a mobile specific interface, using the device sensors and enabling the use of VR technologies like Google's Cardboard. We have realized there is an issue with bandwidth waste when streaming 360° videos, since a great part of the frames are not visible in the scene at a given time, strongly constraining playback in mobile devices. We will focus on solutions for this problem in the future.

We have shown a very specific way of representing stored information about how the video is used, just to explain the kind of system we have developed. As we have already mentioned, we know almost anything about video usage, so many other statistics could be obtained, depending on how we define our interests.

Finally, this work has been accepted to be published in *jAUTI 2017*, 6th Iberoamerican Conference on Applications and Usability of Interactive TV, that will be held 12-13 October (refer Appendix B) and the general platform for measurable interactive 360° videos is under review by the journal *Multimedia Tools and Applications* (refer Appendix A).

5. Acknowledgements

The authors want to thank the people that appear in the documentary for sharing their stories with us. We also want to thank PROEM-AID for their work where they are much needed.

6. References

1. Sawhney, N., Balcom, D., Smith, I.: HyperCafe. In: Proceedings of the the seventh ACM conference on Hypertext - HYPERTEXT '96. pp. 1–10. ACM Press, New York, New York, USA (1996).
2. Gaudenzi, S.: The Living Documentary: from representing reality to co-creating reality in digital interactive documentary.
3. Bleumers, L., Van den Broeck, W., Lievens, B., Pierson, J.: Seeing the bigger picture. Proc. 10th Eur. Conf. Interact. tv video - EuroITV '12. 115 (2012).
4. Sherman, W.R., Craig, A.B.: Understanding virtual reality : interface, application, and design. Morgan Kaufmann (2003).
5. Neumann, U., Pintaric, T., Rizzo, A.: Immersive panoramic video. In: Proceedings of the eighth ACM international conference on Multimedia - MULTIMEDIA '00. pp. 493–494. ACM Press, New York, New York, USA (2000).
6. Lantz, E., Ed: The future of virtual reality. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96. pp. 485–486. ACM Press, New York, New York, USA (1996).
7. Kavanagh, S., Luxton-Reilly, A., Wüensche, B., Plimmer, B.: Creating 360° educational video. In: Proceedings of the 28th Australian Conference on Computer-Human Interaction - OzCHI '16. pp. 34–39. ACM Press, New York, New York, USA (2016).
8. Neng, L.A.R., Chambel, T.: Get around 360° hypervideo. In: Proceedings of the 14th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '10. p. 119. ACM Press, New York, New York, USA (2010).
9. Berning, M., Yonezawa, T., Riedel, T., Nakazawa, J., Beigl, M., Tokuda, H.: pARNorama. In: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication - UbiComp '13 Adjunct. pp. 1471–1474. ACM Press, New York, New York, USA (2013).
10. Giannakos, M.N., Chorianopoulos, K., Chrisochoides, N.: Making sense of video analytics: Lessons learned from clickstream interactions, attitudes, and learning outcome in a video-assisted course. *Int. Rev. Res. Open Distrib. Learn.* 16, (2015).
11. Conglei Shi, Siwei Fu, Qing Chen, Huamin Qu: VisMOOC: Visualizing video clickstream data from Massive Open Online Courses. In: 2015 IEEE Pacific Visualization Symposium (PacificVis). pp. 159–166. IEEE (2015).
12. Chen, Q., Chen, Y., Liu, D., Shi, C., Wu, Y., Qu, H.: PeakVizor: Visual Analytics of Peaks in Video Clickstreams from Massive Open Online Courses. *IEEE Trans. Vis. Comput. Graph.* 22, 2315–2330 (2016).
13. Wachtler, J., Khalil, M., Taraghi, B., Ebner, M.: On Using Learning Analytics to Track the Activity of Interactive MOOC Videos.
14. Kasahara, S., Nagai, S., Rekimoto, J.: JackIn Head: Immersive Visual Telepresence System with Omnidirectional Wearable Camera. *IEEE Trans. Vis. Comput. Graph.* 23, 1222–1234 (2017).
15. Williamson, J.R., Sundén, D., Bradley, J.: GlobalFestival: Evaluating Real World Interaction on a Spherical Display. Proc. Jt. Int. Conf. Pervasive Ubiquitous Comput. Int. Symp. Wearable Comput. 1251–1261 (2015).
16. Bibiloni, T., Mascaró, M., Palmer, P., Oliver, A.: Realidad Aumentada en HbbTV, una plataforma Hypervideo para TV conectada.

- In: Iberian Conference on Information Systems and Technologies, CISTI. pp. 1–6. IEEE (2014).
17. Bibiloni, T., Oliver, A.: D-102: Augmented Reality on HbbTV, an Hypervideo approach. In: TVX 2014, ACM International Conference on Interactive Experiences for Television and Online Video. ACM, Newcastle Upon Tyne (2014).
 18. Bibiloni, T., Mascaró, M., Palmer, P., Oliver, A.: Hypervideo, Augmented Reality on Interactive TV. In: jAUTI2014 III Jornadas Iberoamericanas de Difusión y Capacitación sobre Aplicaciones y Usabilidad de la TVDi | III WTVDI Webmedia 2014. pp. 63–71. Red AUTI, João Pessoa (2014).
 19. Bibiloni, T., Mascaró, M., Palmer, P., Oliver, A.: A Second-Screen Meets Hypervideo, Delivering Content Through HbbTV. Proc. ACM Int. Conf. Interact. Exp. TV Online Video. 131–136 (2015).
 20. Bibiloni, T., Oliver, A., Manresa-Yee, C., Varona, J.: Usability testing of a Hypervideo-based Interactive TV application. In: VI International Conference on Interactive Digital TV IV Iberoamerican Conference on Applications and Usability of Interactive TV. pp. 127–137. Universitat de les Illes Balears, Palma (2015).
 21. Bibiloni, A., Ramis, S., Oliver, A., Perales, F.J.: An Augmented Reality and 360-degree Video System to Access Audiovisual Content through Mobile Devices for Touristic Applications. In: Communications in Computer and Information Science. pp. 44–58 (2016).
 22. Sodagar, I.: The MPEG-DASH Standard for Multimedia Streaming Over the Internet. IEEE Multimed. 18, 62–67 (2011).
 23. ADL: GitHub - adlnet/xAPI-Spec, <https://github.com/adlnet/xAPI-Spec>.
 24. Experience API (xAPI) - Video Vocabulary, <http://xapi.vocab.pub/datasets/video/>.
 25. The Registry, <https://registry.tincanapi.com/>.