



CENTERIS - International Conference on ENTERprise Information Systems /
ProjMAN - International Conference on Project MANagement / HCist - International
Conference on Health and Social Care Information Systems and Technologies,
CENTERIS/ProjMAN/HCist 2018

A rapid prototyping tool to produce 360° video-based immersive experiences enhanced with virtual/multimedia elements

Telmo Adão^{*,a}, Luís Pádua^b, Miguel Fonseca^b, Luís Agrellos^c, Joaquim J. Sousa^a, Luís Magalhães^d, Emanuel Peres^a

^a INESC TEC and University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal

^b University of Trás-os-Montes e Alto Douro, School of Sciences and Technology, 5000-801 Vila Real, Portugal

^c GEMA - Digital and Technology Agency, Cais das Pedras N.º 8, 4050-465 Porto, Portugal

^d ALGORITMI Center, University of Minho, 4800-058 Guimarães, Portugal

Abstract

While the popularity of virtual reality (VR) grows in a wide range of application contexts – e.g. entertainment, training, cultural heritage and medicine –, its economic impact is expected to reach around 15bn USD, by the year of 2020. Within VR field, 360° video has been sparking the interest of development and research communities. However, editing tools supporting 360° panoramas are usually expensive and/or demand programming skills and/or advanced user knowledge. Besides, application approaches to quickly and intuitively set up such 360° video-based VR environments complemented with diverse types of parameterizable virtual assets and multimedia elements are still hard to find. Thereby, this paper aims to propose a system specification to simply and rapidly configure immersive VR environments composed of surrounding 360° video spheres that can be complemented with parameterizable multimedia contents – namely 3D models, text and spatial sound –, whose behavior can be either time-range or user-interaction dependent. Moreover, a preliminary prototype that follows a substantial part of the previously mentioned specification and implements the enhancement of 360° videos with time-range dependent virtual assets is presented. Preliminary tests evaluating usability and user satisfaction were also carried out with 30 participants, from which encouraging results were achieved.

© 2018 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Selection and peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies.

Keywords: Virtual Reality; 360° Videos Editing; Immersive Digital Panoramas; Multimedia; 3D model; 3D text; Spatialized Sound; Prototyping.

* Corresponding author. Tel.: +351 259 350 357.

E-mail address: telmoadao@utad.pt

1. Introduction

Experience simulation using computer-based synthetic environments is the core of virtual reality (VR) research and development field [1], which has been successfully applied to several other areas (such as medicine [2], product design and manufacturing [3], education [4], tourism [5], cultural heritage [6] and human-computer interaction [7]), in a wide range of contexts (e.g. entertainment, training and rehabilitation) and goals (e.g. learning improvement, knowledge dissemination, public participation promotion and medical patient recovery support). Currently, VR is an area of great relevance, whose economic impact is expected to be around 15bn USD, by the year of 2020 [8].

Among the VR environment classes, there is one that has been triggering buzz nearby development and research communities: 360° videos that allow a spherical view of streamed imagery data. These kind of videos has been used to documentaries, tourism, houses and interior design, hobby programs, debate programs, making of, stand-up comedy, among others [9] and several are the modalities to present such content as, for example, computer automatic virtual environments (CAVE), conventional screening devices and head-mounted displays (HMD) [9], [10]. Of course, modality selection should follow the intended requirements for immersiveness and presence, since this aspect ends up having impact in user experience (for example, HMDs have higher capabilities of immersiveness when compared with conventional displays).

More specifically, 360° video allows to set up immersive environments, by providing a complete circular (previously recorded or live) video streaming around a fixed central acquisition system, thus overcoming reduced covering angles that is a limitation of traditional videography [11][12]. Regarding acquisition system, two types of equipment can be applied: a single camera with curved lenses for a full field of view (FoV) covering or a multi-camera setup, where each camera shoots part of the space to be later stitched in a single group [13]. The resulting product is a closed loop video that allows an user to visualize it in portions or as a whole – depending on available FoV provided by viewer application –, by simply and freely change viewpoint orientation [13], [14], with potential benefits for sense of presence [12] (a topic addressed by other studies [15]–[17]). Efforts to provide 360° video with more interactivity and usage simplicity have been presented ([18], [19], [11], [12], [20], [21], [22]), while others [23] focused attentions on quality of experience (QoE), pointing out the strong impact of fast camera motion videos in users' cybersickness.

Currently, a wide variety of consumer-grade cameras to record videographic panoramas is available, which has leveraging new markets that rely their business models in this type of multimedia contents. Social media networks and video platforms are also following this trend, by providing support to 360° video streaming [21]. Although, in what regards to video editing, besides most of the existing tools are costly and demand professional level knowledge to operate with, not all of them support 360° panoramas and even fewer provide freedom to intuitively set up VR immersive environments complemented with parameterizable virtual elements, more specifically 3D models, text and spatial sound, at least, not without programming and/or advanced user skills. Scarcity of such authoring tools has been identified a few years ago [24].

In [25]–[27], text and photos were positioned in certain areas of interest to complement the narrator's voice or to enhance the spatial and contextual perception. Berning et al. [24] used panoramic 360° videos as a base for fast augmented reality (AR) scenarios prototyping. The developed tool enables to insert 2D elements into the video feed, as well as to transform their positions, sizes and opacities. Extensible markup language (XML) files are used to save scenarios and to later play them in a viewer application. Authors also reported that their tool could be easily handled – even by novice users – and the creation of both simple and advanced scenarios could be carried in less than one hour. Freitas et al. [28] proposed an information system for the management and visualization of multisensory contents. Following the same line, Coelho et al. [29] proposed a tool that enables to edit and visualize 360° videos enhanced with audio, haptic and olfactory stimuli. The developed tool enabled to select each stimulus' time range and intensity (e.g. quantity of smell spray to be released), to preview and to fine tune the experience. A pilot evaluation regarding system's usability was carried out with 10 participants, being all of them able to successfully perform the requested tasks. Only one of them asked for help.

This paper aims to propose a general system specification to easily and rapidly configure immersive VR environments constituting visual experiences composed of a single surrounding 360° video sphere that can be complemented with parameterizable multimedia contents namely 3D models, text and spatial sound. Moreover, these contents are planned to be time range dependent or responsive to interaction triggers (e.g. camera eye gaze, user's raycast intersection,

etc.). A preliminary prototype following most of the proposed specification – more specifically, time-based asset embedding – has been implemented in Unity 3D (San Francisco, California, United States of America - USA) and is under analysis as well. Moreover, results of tests made with 30 participants that volunteered to experiment and evaluate usability and overall satisfaction are discussed.

The remaining of this paper is organized as follows: Section 2 describes the system's specification; in Section 3, the first prototype version is presented; prototype's evaluation results are presented and discussed in Section 4. Finally, Section 5 ends this work by presenting some conclusions and remarks towards future developments.

2. General system specification to edit and enhance 360° video spheres with virtual/multimedia contents

The proposed system to edit and enhance 360° video-based environments with virtual/multimedia contents is composed of two fundamental functionality groups: one for setting up experiences (manager-side) and other to visualize the compiled result (spectator-side). Regarding the former, it includes functionalities to setup an immersive stage with the respective 360° video, to import parameterizable assets, to specify events for each imported asset – time dependent and/or based in interaction – and, finally, to preview the experience before delivering it to the spectator, allowing timely adjustments. Assets and events follow a configuration sequence. Firstly, 3D models, sounds or textual descriptions must be imported and then events can be associated to each one. In what regards events, two types are planned to be integrated: a time-based event specifies the objects' time-to-live while interaction-based ones define a response to an interaction (e.g. eye-gaze or contact). The latter group interfaces with the spectator by providing a 360° experience that, besides visualization, also supports interaction. Thereby, during experiences, a spectator may: (1) see visual/audible objects being timely activated/deactivated with the video's course; and (2) be provided with interaction-based objects' feedback. Fig. 1 presents the groups of functionalities and respective user type assignment.

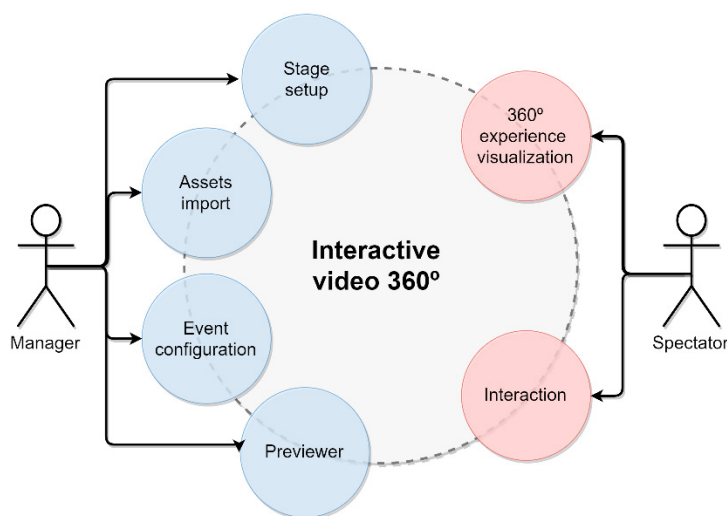


Fig. 1. Groups of general functionalities included in the proposed 360° video editing system. While the manager is responsible by setting up experiences that can include time- or interaction-based assets (objects), spectator is the system's actor to whom experiences are delivered.

Regarding experience manager tasks, the very first step for configuring a 360° video-based experience is to create a new project or load an existing one. Then, video 360° needs to be selected by user or loaded accordingly with the loaded project settings. As long as the manager remains in the application (until quitting), assets can be loaded into the project and, then, a parameterization step may take place, which includes context-based setting definition and time-/interaction-based event attachment. Fig. 2 presents the simple form of a workflow depicting the aforementioned functionalities.

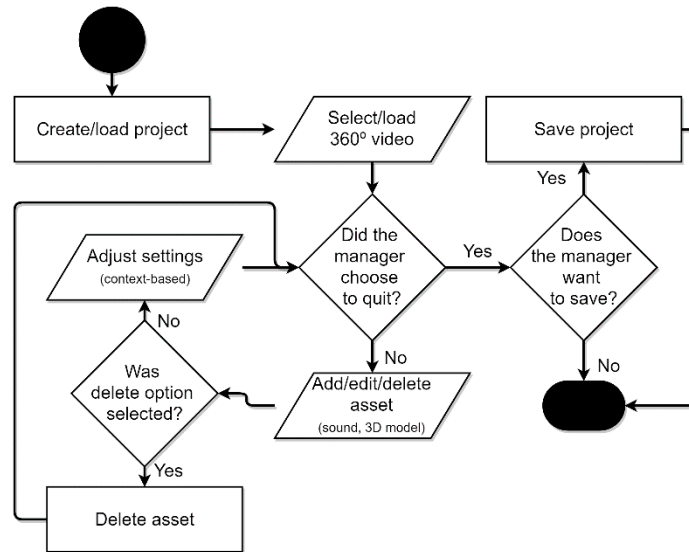


Fig. 2. Experience setup from the manager perspective. Projects can be initialized or loaded from history. To setup an immersive environment, video 360° can be selected (for new projects) or loaded (from existing project settings) and assets can be added, edited or deleted. Moreover, the possibility of adjusting each asset settings is available, which includes directly related parameters or associated triggers (time-/interaction-based).

2.1. Assets and parameters

Each asset type has its own group of directly related parameters. Thereby settings are done in a context-aware basis. Supported types are the following:

- 3D model: corresponds to a mesh group that can be imported from file (e.g. obj, fbx). Besides position inside the environment, manager can also define object's rotation and scale;
- Text: is a mesh supporting character chains. As 3D models, it can be placed inside the environment, according to manager's will. Also, textual description inside the mesh can be edited, as well as its scale;
- Sound: represents an auditory asset. It can be assigned to any position inside the environment and its sound volume can be parameterized. A noteworthy feature of this asset type is the spatiality as it provides to the user an intuitive insight about the sound playing source position, during experiences.

2.2. Assets' controlling triggers

Along with assets parameterization, a manager can attach triggers. As it was previously mentioned, two types can be specified: time- and (interaction) event-based. Regarding the former, it consists in defining a time-to-live for a given asset, i.e., a time range to show up the object in the 360° environment. Two values within video time boundaries must be provided, corresponding to a start and to an end time. The latter type regards to interaction events, in which an interaction listener and a post-interaction trigger event – or simply post-trigger event – must be specified. Essentially, interaction listener establishes which user action activates post-trigger event. In spite of still being opened to new hypothesis, the following list provides the planned interaction listeners to be considered for integration in the proposed system:

- Eye-gaze: detected when the user stares to a given object for a given period of time (typically, a few seconds);
- Contact: occurs if some interaction metaphor is applied to actually reach the virtual object of interest. In a 3D environment, such metaphor can be implemented by, for example, remote raycast – i.e., virtual adjustable ray capable of reaching virtual assets – or proximity picking controller – i.e., physical device with position tracking capabilities, such as HTC Vive (HTC corporation, Xindian, New Taipei City, Taiwan), to convert user's real location into a virtual environment position.

Post-trigger events are elements that define which actions will be performed upon a given object, after an interaction event detection. The following provisional list presents the planned post-trigger events to be integrated in the proposed system:

- Customized behavior engagement: displaying of pre-configured visual/auditory stimuli (for example, translations, pathing definitions, rotations, scaling operations and/or material/color changings affecting 3D models/text meshes and, for audio elements, mixable sound effects such as acoustic transformations and/or playing speed alterations);
- Self-exclusion: in practice, this corresponds to an object's self-deactivation or destruction;
- New asset spawning: assets may be configured to be spawned after interaction with a given object of interest, either to be added to the virtual environment or to replace the referred object.

A simplified flowchart depicting a possible workflow for setting up the aforementioned time/interaction triggers is presented in Fig. 3.

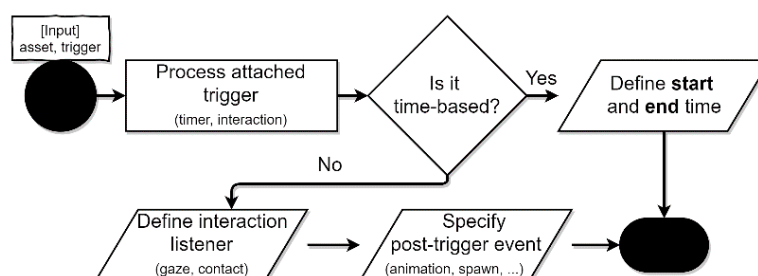


Fig. 3. Assets' trigger configuration workflow. In case of time-based trigger, a start and an end time must be provided (asset's time-to-live). Otherwise, an interaction event listener is required to be defined (eye-gaze or contact), along with a post-interaction trigger (customized behaviour engagement, self-exclusion, new asset spawning).

The following section presents a prototype application implementing the addressed system specification, but only considering time-based triggers. It aims to show proposed system's current development state.

3. Prototype implementation addressing time-based asset's integration in 360° video-based VR environments

To take advantage of a powerful mainstream game engine for multi-platform development, which provides flexible support to easily operate in 3D environments, proposed system's prototype was implemented in Unity 3D (version 2018.1). Along with a powerful graphical interface enabling to directly interact with a 3D stage to which components can be added by simple drag-and-drop mouse operation, Unity 3D also supports collision mapping, texture manipulation, screen space ambient occlusion and behavior-oriented programming, whose code scripts might be directly attached to game objects placed in the 3D stage.

At current stage, this prototype is implementing manager's side authoring tool to set up 360° video-based experiences that can be pre-visualized. It consists in a 3D tool that isolates the manager inside to a sphere and allows him to select the video and to freely place objects and parameterize them according to their type. Moreover, each object can be set with timers to show up and fade out from the video. Projects can be saved into JavaScript Object Notation (JSON) format files, to be later accessed and edited or, simply, to be used to display immersive experiences for spectators.

3.1. Immersive 360° videos

Prototype implements a sphere of 25 Unity units (equivalent to 25m, by default) that constitutes the cocoon for 360° video display. To be able to reproduce videos to its central point direction – i.e., to have interior projection – sphere normals were inverted. A series of 10 experiments were carried out to visually check the adaptability of the sphere in displaying different MP4 videos, in which a proper rendering and a consistent texture application behavior

was observed. However, some slightly noticeable distortions were identified, which is a quite common occurrence related with spherical videos stitching. This issue was considered negligible, since it falls out of this paper's scope.

3.2. Supported Objects

Three types of virtual objects are supported by the developed prototype: 3D models, spatialized sound and text meshes (Fig. 4). All of them can be freely placed within the spherical space.

In case of virtual 3D models, manager can select among cubic primitives or 3D polygonal meshes imported from hard drive. For the latter, a free asset from Unity Asset Store [30] was integrated in the prototype project to support wavefront files (.OBJ) import at runtime. Regarding operations that can be made to transform virtual 3D models, rotations, scaling and animations are allowed. While rotations can occur around X, Y and Z axis, individually, scaling is applied to the 3-axis locked together. Regarding animations, at the current development stage, only a non-stopping rotation around Y axis is supported.

Spatialized sound is always positioned rotated towards central sphere position, to shoot the audible stimulus directly to the user, during experiences. This type of asset requires a wave file picked up from hard drive by the manager, which represents the sound to play. Its graphical representation in editing mode is a 3D sound speaker that can be selected and repositioned. Parameterization includes sound volume (ranging from 0 – softer – to 1 – louder) and a boolean flag defining if the asset should play in a loop or not. Sound spatialization is managed by Unity's Audio component, as long as spatial blend propriety is set to 1 (which is not, by default). Additionally, 3D sound settings such as doppler level and volume rolloff can be modified. In the proposed prototype, each sound asset spawned at runtime has its spatial blend propriety set to 1 and uses the default 3D audio settings, since they are good enough for the current sound spatialization needs.

For informative purposes, the proposed prototype allows to integrate parameterizable virtual text meshes within the immersive sphere space. Just as sound assets, textual meshes keep their rotation towards sphere center, regardless position modifications. Locked XYZ scaling and text editing constitute the set of parameters that can be changed by the manager to transform textual assets.

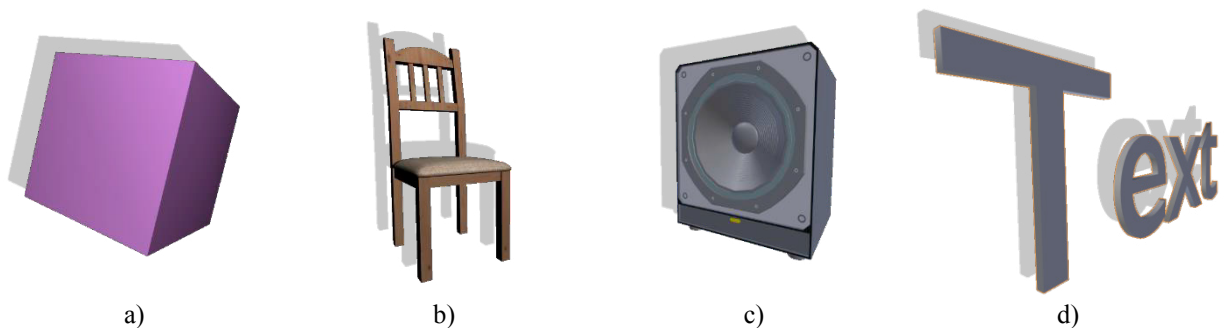


Fig. 4. Types of virtual assets supported for integration by the proposed prototype for setting up 360° video-based environments: a) 3D model primitive (only cubes are supported, currently); b) 3D model imported from hard drive; c) Spatial sound representation (visible in editing mode); and d) textual mesh.

3.3. Interaction

Prototype's interaction is implemented considering a minimalist set of inputs, for the sake of simplicity: mouse dragging, left mouse button, keyboard's left Alt and, also, keyboard's numbers 1 and 2 from the typewriter section (TS). Table 1 discriminates supported prototype's actions and corresponding controls' combinations.

Table 1. Prototype's available actions and respective keyboard/mouse controls to perform them.

Action	Controls' combination
To navigate with the mouse in free mode, enabling menus access (main menu, dialog controls, etc.).	Mouse cursor dragging.
To visualize 360° video all around, by moving raycast.	Left Alt + mouse cursor dragging.
To move away or to bring closer raycast positioner.	Left Alt + TS 1 or left Alt + TS 2, respectively.
To place an asset in the environment.	Left mouse click in one of the first four main menu buttons (from left to right).
To drag elements in the environment (repositioning).	(1) Left Alt + left mouse click + mouse dragging.
To move away or to bring closer assets, using raycast.	(1) Left mouse click + TS 1 or left mouse click + TS 2, respectively.
To parameterize assets through context-based menu (for more information, check previous subsections).	(1) Mouse dragging to navigate to the right sided context-based menu; mouse clicking to select option.
To change asset's time-to-live, through context-based dialog.	(2) Mouse clicking/dragging in the minimum/maximum handlers belonging to the context-based dialog's time-line.
To change asset's additional parameters, through context-based dialog	(2) Mouse clicking to navigate check/uncheck: animation for 3D models and loop playing for sound assets.

(1) Requires to have an asset selected, by orienting the raycast towards element of interest.

(2) Requires (1) to be satisfied, following context-based dialog invocation through editing options selection in the context-based menu.

A feature worthy to highlight regarding rotations interaction is the circular colored gizmo that is shown to provide a visual intuition of the axis under which a given asset will be rotated. This visual aid is activated whenever the mouse cursor is over a rotation button (by event trigger) and will assume the orientation of the desired rotation axis (X, Y or Z, depending on the button the mouse is overlaying).

In what concerns to editing menus, these are (exclusively) presented considering the context of the task that is being performed at a given moment. When the manager is freely navigating, besides a raycast for positioning/selecting virtual assets in the 360° video-based environment, main menu is presented in the screen bottom to allow virtual assets' attachment (that assume the raycast pointer position), save, preview and quit the experience project. If asset attachment is requested, a dialog shows up with a parameterizable temporal line allowing to define start and ending time for respective rendering (full video time range is assumed by default). Another option depending on the asset type that is being added is shown along with that time-line: for virtual 3D models, a checkbox to select/unselect animation (rotation around Y axis); for spatial sound, there is also a checkbox to toggle loop playing; text meshes do not have this second parameter available. The same type of dialogs is shown if the manager is editing a previously attached asset, but with a delete button available to wipe out the element that is being handled (in selection), in case of need. When the manager is selecting an asset lying in the scene, an asset type-based menu is properly presented at the right margin of the screen. Edit option is transversal to all types. For unanimated virtual 3D models, scaling, rotation and reorientation (to turn the asset's face towards the sphere center) options are made available. Otherwise (if the 3D model is set to be animated), rotation options are hidden. Regarding audio assets, a volume slider ranging between 0 (softer) and 1 (louder) is shown in the selection-active menu. For text meshes, such menu presents – besides an editing button – scaling options to make the descriptions bigger or smaller. Finally, when the option to save the experience project is requested, a window pops up prompting a storing folder and a file name.

A screenshot of the working prototype is presented in Fig. 5, in which several assets can be seen, namely a selected cube (with the raycast pointing to it), a virtual model of a chair, a spatial audio source and several text meshes tagging each one of them.

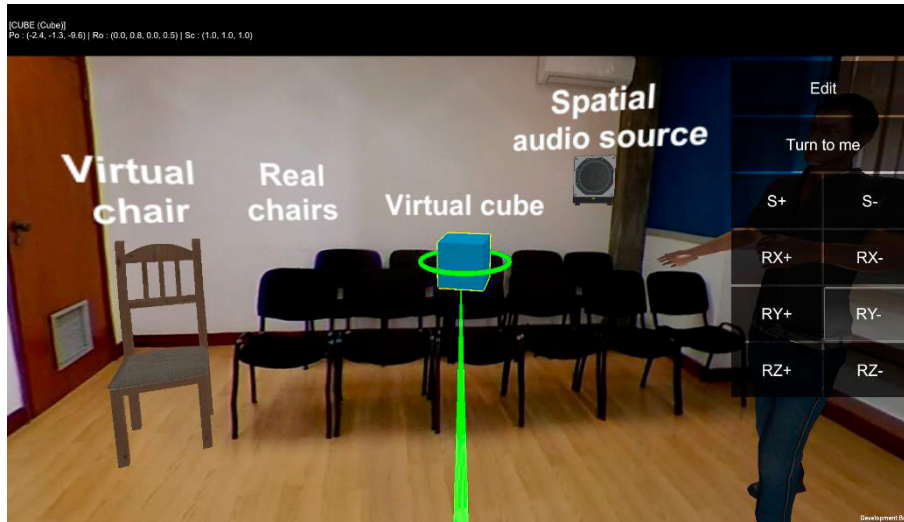


Fig. 5. Prototype's screenshot, depicting several types of virtual assets embedded in a 360° video-based environment. The following elements can be seen: a virtual chair (next to real chairs), a spatial audio source, a virtual cube that is selected by raycast, a context-based menu (right part of the screen) regarding cube selection and an avatar partially occluded by that menu.

3.4. JSON format for saving/loading/previewing experience projects

To support VR experience projects' saving/loading/previewing, JSON files are used to store the state of the assets lying in the environment – and respective proprieties – at a given moment. The main difference between a JSON file to save and a JSON file to preview is that the former stores data definitively – for future access (e.g. experience loading) –, while the latter is temporarily generated only to support the preview mode and is deleted when user quits previewing. Regardless, both files keep the very same structure: a path for the 360° video to play takes the top of each file and the remaining of it is reserved for arrays, grouping the virtual environment's assets per type, i.e., arrays of primitives, 3D models, sounds and text meshes.

Next section regards to prototype evaluation, wherein tests with participants – to evaluate usability and user satisfaction – and respective results are analyzed and discussed.

4. Prototype evaluation

Experimental design used to evaluate the proposed prototype and respective results processed over data gathered from questionnaires are both presented and discussed in this section. To assess the proposed system's prototype with respect to both usability and satisfaction, tests with volunteers were carried out, in an experience assembling context. They were provided with a protocol specifying a set of tasks to accomplish. In the end, they were invited to answer to a couple of questionnaires.

4.1. Material and methods

An academic dissertation defense simulation was the experiment selected to evaluate the proposed prototype. Thereby, the public academic events room of the School of Sciences and Technology (*Escola de Ciências e Tecnologia* – ECT, Pole I), integrated in the University of Trás-os-Montes e Alto Douro (UTAD) campus, was recorded with a 360° camera to be used as experiment's video-sphere background.

The experiment consisted in asking to 30 participants to properly populate the video-sphere with virtual assets – more specifically, 3D models, sound and text – configured with a time range, i.e., time-to-live in the scene. An adaptation stage was carried out to let the participants get familiarized with the prototype's interaction. Firstly, they were instructed how to use the software and then, they were asked to freely explore it. Afterwards, they were

challenged to accomplish a protocol with 6 concrete tasks. In the end, feedback regarding functionality-centered usability and overall satisfaction was gathered through a couple of questionnaires, which will be analyzed in a further subsection.

4.1.1. Experimental 360° video-based environment

The 360° video content used for this experience was recorded using a Ricoh Theta V (Ricoh Company Ltd., Tokyo, Japan) capturing device that is constituted by a couple of 180° FOV cameras and supports 4K resolution at 30 frames per second (fps). As previously mentioned, video recording took place in the academic events room of ECT, UTAD, which can be divided in three main sections that must be explained to participants before they perform the experimental trial: audience, jury's table (both placed in opposite room's sides) and candidate's spot (placed in the room's center).

Most of the virtual models to enhance the 360° video-based environment were downloaded from public access repositories¹, with free usage licensing: a bottle of water, a laptop, an avatar and an audio speaker representing sound asset were the models used from those repositories. Regarding the cube primitive that is made available for integration in this development stage, it is natively provided by Unity framework. These elements were placed in a specific folder to be easily accessed by participants during experimental tests.

4.1.2. Sample characterization

Most of the participants were students of Communication and Multimedia (CM) course (~66,67%) while the remaining sample was composed of students of Informatics and other similar courses (from now on referred as IOSC). Elements in both groups were mostly good or advanced computer users and had contacted with VR environments and 360° videos in previous occasions. A balanced regarding video post-/production skills was verified: approximately half of the participants pointed out to be inexperienced or to have a basic/average knowledge in such topic, while the remaining sample considered to have good/advanced capabilities. Other observations include academic qualification and gender: among the participants, 15 were bachelor students, 12 were attending for master's degree, the remaining 3 were in PhD programs; 4 individuals were female.

4.1.3. Proceedings and task protocol

Volunteers provided individual contribution for prototype's testing. They started of being informed about this tests' context, as well as prototype's goal. Then, they were presented with a consent declaration to sign, thus agreeing to be voluntarily integrated in the testing sample and, also, to provide their anonymized data for scientific processing purposes. Prototype and main functionalities were shown and explained, as well as the organization of the public academic events room recorded for the 360° video that composes the immersive environment. Afterwards, participants were invited to freely explore the proposed software with the goal of letting them get familiar with it (adaptation stage). When they felt comfortable to start the official test, a protocol to produce a 360° video-based immersive environment simulating a public dissertation defense was provided to them, in which the following 6 concrete tasks' sets were requested:

- To place a primitive (cube) nearby audience's area, along with a textual description "Audience"; required time range for both elements is 2-10 seconds;
- To place two bottles of water upon juries' table and make them look like they are seamlessly supported; and, also, a textual description "arguer" around the same table; these elements are shown between 20th and 30th second;
- To edit the last textual element and to change its content to "Jury", as well as to adjust its showing interval to 20-35 seconds; to delete one of the bottles previously imported;
- To place a laptop upon the candidate's table and change its rotation, scale and position at will; this asset is shown the entire video;

¹ TurboSquid (<https://www.turbosquid.com/>) and CGTrader (<https://www.cgtrader.com/>).

- To add an avatar representing the candidate and make it look like it is lying somewhere between candidate's table and audience's area; to place the available sound source nearby the previously added asset, aiming to simulate candidate's voice. Both elements are shown the whole video time;
- Finally, to previsualize the freshly created environment.

Errors per task and overall execution time were registered during experimental trials, for each participant. In the end, questionnaire-based feedback was requested.

4.1.4. SUS-like Questionnaires

Two short questionnaires (Table 2) – each one having 10 queries – were developed to obtain participant's feedback: one to evaluate functionality-centered usability and another to assess overall satisfaction. While the former is composed by questions regarding the functionalities that were made available for this prototype version (e.g. asset introduction and repositioning, rotation and scaling operations, etc.), the latter contains questions of general purpose to assess overall interaction, frustration/pleasure and interest (e.g. assets manipulation, protocol execution difficulty, desire for exploring more). Likert-based scale with five classification options (very hard to very easy and strongly disagree to strongly agree) was adopted for both questionnaires, to enable not only fast answering but, also, discrimination between levels. Then, following a SUS-like metric, participants' answers were converted to a range of 0-100 points, to retrieve the final score.

Table 2. Questionnaires used to retrieve user's feedback: Questionnaire 1 regards to functionality-centered usability, while overall satisfaction relates to Questionnaire 2.

Nº.	Questionnaire 1 (very hard – very easy) – usability evaluation by functionality –	Questionnaire 2 (strongly disagree – strongly agree) – overall satisfaction –
1	Raycast navigation	3D content seemed to be seamlessly mixed with the 360° video.
2	Sound asset introduction	The set of assets made available (virtual 3D models, sound and text) was satisfactory.
3	3D model asset introduction	It was very difficult to follow and accomplish the proposed protocol.
4	Select/release assets	Interaction with virtual assets seemed adequate.
5	Parameterize/edit assets through context-based popup dialog	I was disappointed with the assets integration in the environment, due to the lack of naturalness.
6	Assets repositioning	It was very complicated to deal with presented set of menus and options.
7	To carry out scale and rotation operations to affect 3D models	Experience was fun and I did not even notice the time passing by.
8	Change sound assets volume	I would like to have explored more.
9	Delete assets	I was craving for the experience to end.
10	Save/preview experiences	Overall, I am satisfied with the proposed prototype

Questionnaires' processing results are presented in the following subsection, wherein participants' feedback is analyzed globally, as well as some interesting tendencies.

4.1.5. Error and time registering

During tests, the number of errors committed by each participant in each task was dully registered. Mistaken actions (e.g. insertion of wrong assets types) and ignored requests (incomplete tasks) were considered as errors, which could occur more than one time per task. Overall time spent by each participant for protocol execution was also registered.

4.2. Results and discussion

From the processed questionnaires, a mean SUS score of ≈ 76 points for functionality-centered usability and ≈ 75 points for overall user satisfaction were obtained as results. According to [31], such score is enough to adjectively classify the proposed prototype as good. Regarding usability, the questions that contributed the most for score lowering were numbers 1 (raycast navigation) and 6 (assets positioning), with ≈ 65 points each one, which seems to be related with the racayst interaction issues. Questions 2 (sound asset introduction) and 10 (save/preview experience) were the ones obtaining the best SUS score – ≈ 86 and ≈ 95 points, respectively –, due to the simplicity and directness associated with inherent tasks. On the other hand, overall satisfaction results presented their worst score on questions 1 (content seamless mixing with the environment) and 5 (asset integration naturalness) – with ≈ 55 and ≈ 60 points, respectively –, pointing out the need for improving software’s realism. Virtual re-lightning/shadowing can be a key-factor influencing this issue. Still addressing overall satisfaction, best scores were obtained for questions 3 (level of difficulty in performing requested tasks) and 9 (user interest) – the former with ≈ 90 points and the latter with ≈ 88 points –, which denotes that prototype’s usage is easy and pleasant. Both functionality-centered usability and overall user satisfaction scores are depicted in the chart of Fig. 6.

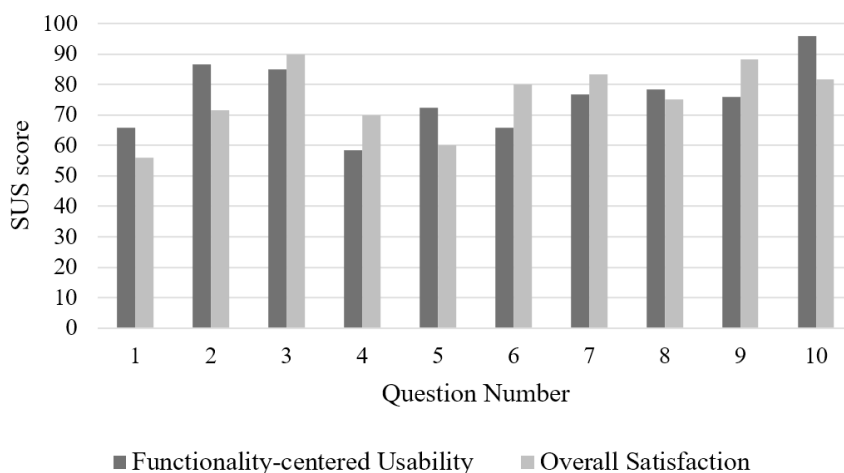


Fig. 6. SUS-based scores for functionality-centered usability (dark-grey bars) and overall user satisfaction (light-grey bars). Values correspond to participants’ average, per question.

In what regards to the number of errors during tasks performing, an average of 1,43 errors was observed. Participants also lasted an average time of 7 minutes and 46 seconds to accomplish the proposed protocol.

Data interaction in some subgroups was analyzed, more specifically, by confronting male vs. female individuals and, also, CM vs. IOSC students. In spite of having only 4 female elements, this subgroup was averagely faster and more accurate than male subgroup. Moreover, in female individuals’ evaluation, higher rates regarding functional-centered usability and user satisfaction were achieved. A resembling situation occurred when CM and IOSC were compared. Students of the former course were more accurate in performing protocol’s tasks and the SUS rates obtained by their feedback for functionalities-centered usability and overall satisfaction were higher. Additionally, the latter students’ group was slightly slower accomplishing the protocol. The better results achieved by CM students can be associated to that course’s intrinsic audio-visual component, which is stronger when compared to IOSC. Referred data can be found in Table 3.

Table 3. Mean values for SUS scores, errors and time spent performing protocols tasks, presented by 4 subgroups, besides general sample: (1) male individuals; (2) female individuals; (3) CM students; and (4) IOSC students.

	Average usability (points)	Average satisfaction (points)	Average errors (points)	Average time (H:m:s)
Male subgroup	74,81	74,62	1,5	00:07:51
Female subgroup	84,38	81,86	1	00:07:14
CM students	77,75	76,75	1,35	00:07:47
IOSC students	72,75	73,25	1.6	00:07:56

Next section will sum up paper's main conclusions and discuss future developments, including prototype's improvements.

5. Conclusions and future work

In this paper, a general system specification to set up 360° video-based immersive VR environments that can be complemented with parameterizable multimedia contents namely 3D models, text and spatial sound was introduced. A preliminary prototype implementing time-based asset embedding developed in Unity 3D was also presented, along with tests made with 30 participants that volunteered to experiment and evaluate it. Two SUS-like questionnaires were used to perform that evaluation, from which encouraging results were obtained. Accordingly to previously defined classification metrics [31] and in spite of being in an initial development stage, this prototype already reached the level of "good", at least in what regards to functionality-centered usability and overall satisfaction.

In a near future, the weaker aspects of the prototype must be addressed, namely to improve realism and raycast interaction. While the former requires the integration of re-lightning/shadowing approaches, the latter demands a broader study on usability heuristics and interaction metaphors/devices, as well as user-based fine-tuning. Moreover, responsive interaction triggers (e.g. camera eye gaze, user's raycast intersection, etc.) implementation – considered in the proposed specification – will be carried out to consolidate the proposed solution.

Acknowledgements

This work was financed by project "CHIC – Cooperative Holistic View on Internet and Content" (N° 24498), financed the European Regional Development Fund (ERDF) through COMPETE2020 - the Operational Programme for Competitiveness and Internationalisation (OPCI).

References

- [1] D. Ochi, K. Niwa, A. Kameda, Y. Kunita, and A. Kojima, "Dive into Remote Events: Omnidirectional Video Streaming with Acoustic Immersion," in *Proceedings of the 23rd ACM International Conference on Multimedia*, New York, NY, USA, 2015, pp. 737–738.
- [2] K. Laver, S. George, S. Thomas, J. E. Deutsch, and M. Crotty, "Virtual reality for stroke rehabilitation: an abridged version of a Cochrane review.," *Eur J Phys Rehabil Med*, vol. 51, no. 4, pp. 497–506, Aug. 2015.
- [3] L. P. Berg and J. M. Vance, "Industry use of virtual reality in product design and manufacturing: a survey," *Virtual Reality*, vol. 21, no. 1, pp. 1–17, Mar. 2017.
- [4] S. Marks, D. White, and M. Singh, "Getting Up Your Nose: A Virtual Reality Education Tool for Nasal Cavity Anatomy," in *SIGGRAPH Asia 2017 Symposium on Education*, New York, NY, USA, 2017, pp. 1:1–1:7.
- [5] I. P. Tussyadiah, D. Wang, T. H. Jung, and M. C. tom Dieck, "Virtual reality, presence, and attitude change: Empirical evidence from tourism," *Tourism Management*, vol. 66, pp. 140–154, Jun. 2018.
- [6] T. Adão et al., "Bringing together UAS-based land surveying and procedural modelling of buildings to set up enhanced VR environments for cultural heritage," in *24º Encontro Português de Computação Gráfica e Interação (EPCGI)*, 2017, pp. 1–8.

- [7] C.-M. Wu, C.-W. Hsu, T.-K. Lee, and S. Smith, "A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment," *Virtual Reality*, vol. 21, no. 1, pp. 19–29, Mar. 2017.
- [8] T. text provides general information S. assumes no liability for the information given being complete or correct D. to varying update cycles and S. C. D. M. up-to-D. D. T. R. in the Text, "Topic: Virtual Reality (VR)," *www.statista.com*. [Online]. Available: <https://www.statista.com/topics/2532/virtual-reality-vr/>. [Accessed: 29-May-2018].
- [9] L. Bleumers, W. Van den Broeck, B. Lievens, and J. Pierson, "Seeing the Bigger Picture: A User Perspective on 360° TV," in *Proceedings of the 10th European Conference on Interactive TV and Video*, New York, NY, USA, 2012, pp. 115–124.
- [10] V. K. Kothapally, S. L. Grant, and Y. R. Zheng, "Immersive audio for human-machine interface of unmanned ground vehicles," in *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2016, pp. 1–5.
- [11] T. Chambel, M. N. Chhaganlal, and L. A. R. Neng, "Towards Immersive Interactive Video Through 360&Deg: Hypervideo," in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, New York, NY, USA, 2011, pp. 78:1–78:2.
- [12] T. Rhee, L. Petikam, B. Allen, and A. Chalmers, "MR360: Mixed Reality Rendering for 360 Panoramic Videos," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 4, pp. 1379–1388, Apr. 2017.
- [13] L. Bleumers, W. Van den Broeck, B. Lievens, and J. Pierson, "Seeing the Bigger Picture: A User Perspective on 360 TV," in *Proceedings of the 10th European Conference on Interactive TV and Video*, New York, NY, USA, 2012, pp. 115–124.
- [14] M. Budagavi, J. Furton, G. Jin, A. Saxena, J. Wilkinson, and A. Dickerson, "360 degrees video coding using region adaptive smoothing," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 750–754.
- [15] M. Bessa, M. Melo, D. Narciso, L. Barbosa, and J. Vasconcelos-Raposo, "Does 3D 360 video enhance user's VR experience?: An Evaluation Study," presented at the Proceedings of the XVII International Conference on Human Computer Interaction, 2016, p. 16.
- [16] M. Melo, S. Sampaio, L. Barbosa, J. Vasconcelos-Raposo, and M. Bessa, "The impact of different exposure times to 360° video experience on the sense of presence," 2016, pp. 1–5.
- [17] D. Narciso, M. Bessa, M. Melo, A. Coelho, and J. Vasconcelos-Raposo, "Immersive 360° video user experience: impact of different variables in the sense of presence and cybersickness," *Universal Access in the Information Society*, Oct. 2017.
- [18] J.-H. Kwon et al., "Be Closer As You Being There: HMD-based Social Interaction System," in *SIGGRAPH ASIA 2016 VR Showcase*, New York, NY, USA, 2016, pp. 2:1–2:2.
- [19] L. A. R. Neng and T. Chambel, "Get Around 360&Deg: Hypervideo," in *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, New York, NY, USA, 2010, pp. 119–122.
- [20] T. Pakkanen et al., "Interaction with WebVR 360 video player: Comparing three interaction paradigms," in *2017 IEEE Virtual Reality (VR)*, 2017, pp. 279–280.
- [21] Y.-C. Lin, Y.-J. Chang, H.-N. Hu, H.-T. Cheng, C.-W. Huang, and M. Sun, "Tell Me Where to Look: Investigating Ways for Assisting Focus in 360° Video," 2017, pp. 2535–2545.
- [22] L. A. Hernandez, J. Taibo, and A. J. Seoane, "Immersive video for virtual tourism," 2001, pp. 63–73.
- [23] H. T. T. Tran, N. P. Ngoc, C. T. Pham, Y. J. Jung, and T. C. Thang, "A subjective study on QoE of 360 video for VR communication," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, 2017, pp. 1–6.
- [24] M. Berning, T. Yonezawa, T. Riedel, J. Nakazawa, M. Beigl, and H. Tokuda, "pARnorama: 360 degree interactive video for augmented reality prototyping," 2013, pp. 1471–1474.
- [25] SamsungIndia, *360° Film on Taj Mahal by Samsung & UNESCO*. 2018.
- [26] National Geographic, *360° Great Hammerhead Shark Encounter | National Geographic*. 2016.
- [27] BuzzFeedVideo, *How Animals See The World (360° Video)*. 2016.
- [28] J. Freitas, C. Meira, M. Melo, L. Barbosa, and M. Bessa, "Information system for the management and visualization of multisensorial contents," 2015, pp. 1–7.
- [29] H. Coelho, M. Melo, L. Barbosa, J. Martins, M. Sérgio, and M. Bessa, "Immersive Edition of Multisensory 360 Videos," in *Trends and Advances in Information Systems and Technologies*, vol. 746, Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, Eds. Cham: Springer International Publishing, 2018, pp. 309–318.
- [30] "Runtime OBJ Importer - Asset Store." [Online]. Available: <https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547>. [Accessed: 03-Jun-2018].
- [31] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *J. Usability Studies*, vol. 4, no. 3, pp. 114–123, May 2009.