

# aCAVE: Augmented CAVEs for entertainment applications

Elias Lawrence Marques Júnior\*

Luis Paulo Santos Valente

Esteban Walter Gonzalez Clua

Universidade Federal Fluminense, Ciência da Computação, Brasil

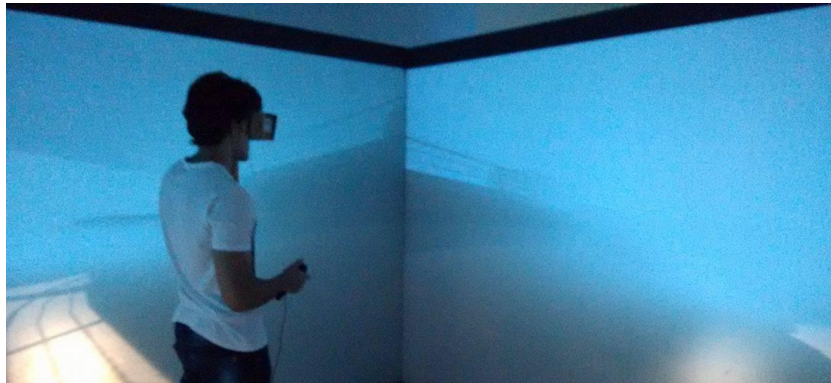


Figure 1 – User interacting with the aCAVE

## ABSTRACT

CAVE systems have been around virtual reality for quite some time. These systems immerse users in a cube-like environment where images are projected into its walls. Usually, in these systems interaction occurs only in the virtual space -- there is no interaction with virtual or physical objects in the physical space that comprises the CAVE. Meanwhile, there is a growing trend about mobile and accessible HMD devices arriving at the mass market, which brings exciting opportunities for research in virtual reality and entertainment applications. Considering these two issues, in this paper we present the aCAVE system (augmented CAVE) for entertainment applications that addresses the first issue by using HMD devices. In aCAVE, a user wears a see-through HMD that enables him/her to interact with other users, physical objects, and virtual objects while seeing the physical (CAVE interior) and virtual spaces (the virtual world projected in the CAVE walls). As a proof of concept we discuss the architecture and implementation of a testbed aCAVE application, which enables the user to navigate under a simulated sea and interact with virtual fishes.

**Keywords:** Augmented reality, aCAVE, virtual reality, entertainment.

## 1 INTRODUCTION

Over many years, the concepts of virtual reality and augmented reality have been related as a high cost hardware and resources, being primarily used in high-end scenarios. Examples include the automotive industry, the oil industry, and simulators.

However, recently there is a growing trend in the industry to bring VR hardware (head mounted displays – HMDs) to the consumer market at affordable prices. Some examples are examples the Facebook Oculus Rift, Samsung VR, HTC Vive, and Sony PlayStation VR. Also, currently it is possible to turn a smartphone into a HMD by using attachable equipment such as Google's Cardboard.

These HMD devices are sensitive to head movements, enabling the user to control virtual movements in a natural way. Several

innovations have been displayed in the hardware in order to mitigate physical discomforts typically felt while using this kind of device (e.g. headache, nausea and dizziness). Some of these HMDs (e.g. Project Morpheus) have OLED screens (5.7 inches), resolutions up to 1080p, and frame rates up to 120 FPS, which enables smooth and flowing images and animations. Another example is the HTC RE Lives, a partnership between Valve (responsible for Steam) and HTC (known company in the technological environment). The device will feature ergonomic controls for each hand, allowing users full interaction with the virtual world objects.

Presented in 1992 CAVE (CAVE Automatic Virtual Environment) is a virtual reality and scientific visualization system. The general CAVE design consists of a room whose walls, ceiling and floor surround a viewer with projected images. This design overcomes many of the problems faced by other virtual reality systems (e.g. physical occlusion) and can be constructed from currently available technology. Suspension of disbelief and viewer-centered perspective, are often used to describe such systems [1]. This system assures great quality and resolution of viewed images, and wider field of view in comparison to HMD based systems [2].

The ultimate VR means that no user interface is needed at all – every interaction task should be as natural as in the (real) world. In this way, many techniques may be used to enhance the interaction model [4], but they still use some metaphors to make human-computer dialog easier. The aCAVE is a way that we found to get closer to the natural user interfaces, where just the movement of the user's head is sufficient to interact with the application.

Mazuryk and Gervautz [2] describe one of the difficulties and main goals of the computer science, to achieve the “invisible interface”. They state that “*VR means that no interface is needed*”: *every kind of human-computer-human interaction should be so natural and intuitive that neither learning nor adaptation should be necessary. Though, we are far from this: today's interfaces are clumsy, often require heavy hardware devices, complicated calibration steps and non-intuitive interaction*

---

\*e-mail: eliaslawrence@id.uff.br

paradigms. Hence they are not easy to operate by the unskilled user.”

The main goal of this paper is to show tools that can be used as means of integration between the technologies employed into CAVE-like displays and the Head Mounted Display devices. The idea is to combine virtual and real worlds, where the user is inserted in a virtual world and it is able to see real elements. To realize this vision we use augmented reality (AR) techniques. AR is “a technology that presents a virtual world that enriches, rather than replaces the real world” [1]. Still, Mazuryk and co- Gervautz [2] say that “*This is achieved by means of see-through HMD that superimposes virtual three-dimensional objects on real ones. This technology was previously used to enrich fighter pilot’s view with additional flight information (VCASS). Thanks to its great potential – the enhancement of human vision – augmented reality became a focus of many research projects in early 1990s.*”

As far as we are aware, there is no other work on integration of CAVEs and head mounted displays in the literature. The purpose of this new paradigm is not only the entertainment but also the possibility to build therapies environments or even simulators general tasks, since the immersion and realism are the major premises.

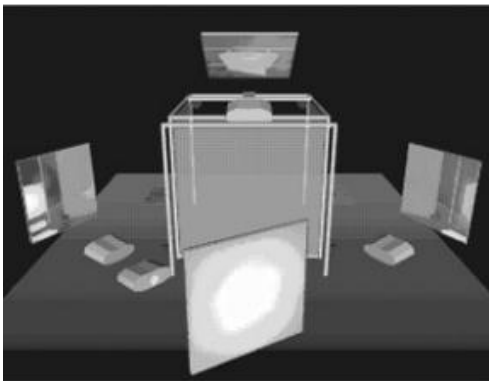


Figure 2 - Surround display diagram: CAVE. [4]

In this paper, we propose a new concept of CAVE where there is the mutual relation between virtual and physical world. We propose a system with 3 layers (two virtual layers and a physical one) in which there is a total immersion of the physical world into the virtual one, but some real elements can and should be stayed put. This way, our work is based on the relation between these layers and the application developed in order to exemplify this paradigm.

The whole idea was not only to provide a fully immersive system, but also a system that could support natural interaction paradigms. The aCAVE pursues this idea by combining a CAVE and a see-through HMD – the motivation for this latter component is that “*seeing one’s own body or those of other people may be an important factor for realistic immersion.*” [5].

## 2 THE aCAVE CONCEPT

The aCAVE (augmented CAVE) is a mixed-reality environment that expands a traditional CAVE (CAVE automatic virtual environment) through HMD devices and integration with the physical world.

A traditional CAVE (also known as a surround display) consists of a cube-like room where its walls are typically projection screens. A user enters this room and experiences the virtual environment. In these environments, the user interacts only with virtual contents that are projected in the CAVE walls. In some CAVE setups the projection screens are stereoscopic, which helps

to create the illusion of 3D depth. These systems require users to wear 3D glasses. This setup may enable the system to create the illusion that a user walks around specific 3D virtual objects. However, if there are physical objects between the user and the projection screen (e.g. the user’s hands, other users), this effect collapses due to physical occlusion.

On the other hand, augmented reality (AR) is a paradigm where the real world is enhanced with virtual objects. As Azuma [6] compares augmented reality to virtual reality, as “*VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world.*”

The aCAVE concept is inspired by augmented reality through enabling the application to superimpose virtual content on the real-world environment and on the projected virtual world (i.e. the original CAVE). To realize this vision, in aCAVE, users wear see-through HMD devices that the system uses to deliver virtual content.

In this regard, an aCAVE system consists of three layers: virtual layer 1 (HMD-enabled AR), physical environment, and virtual layer 2 (traditional CAVE), as illustrated in Figure 3. The remaining of this section describes each of these layers.

With a system that uses the AR technology along with the CAVE, we could provide to the user a better feeling of immersion, once this unity from both technologies allows us to elaborate an environment divided in layers giving a sense of depth and it also treats the problem of occlusion from the traditional CAVE. In Figure 4 we can see the representation of the user’s view.

### 2.1 Virtual layer 1 (AR)

The virtual layer 1 uses see-through HMDs to enable users to see other users, physical objects placed in the CAVE environment, and to enable augmented reality and virtuality.

Using see-through HMDs enables the aCAVE to solve the problem of physical occlusion in the mixed-reality environment, as the system uses the HMD to display 3D objects that are closer to the user and the CAVE walls to display the virtual scenario and farther objects. Without the presence of this layer, no virtual object could be projected in front of real elements, due to occlusion.

### 2.2 Virtual layer 2 (VR)

The virtual layer 2 corresponds to a traditional CAVE. In the current version, our system does not use stereoscopic images, but in future works we intend to include synchronization of the HMD camera with the projectors, in order to produce an active 3D display. These CAVEs offer a wide field of view, which makes them very attractive for VR applications.

The walls in which the images are projected form a cube-like space that surrounds the user. This space corresponds to the physical environment layer. The user is able to walk around the mixed-reality environment using dedicated controllers (e.g. joysticks) and their own bodies (e.g. physically walking or real object manipulation).

### 2.3 Physical environment layer

The physical environment corresponds to the CAVE interior space. This environment supports multiple users who are able to interact freely among themselves.

The physical environment may house other physical objects, sensors, and actuators. The aCAVE may use these objects as direct input devices (i.e. through conscious user manipulation) or indirect input devices (i.e. sensor-based interactions that may

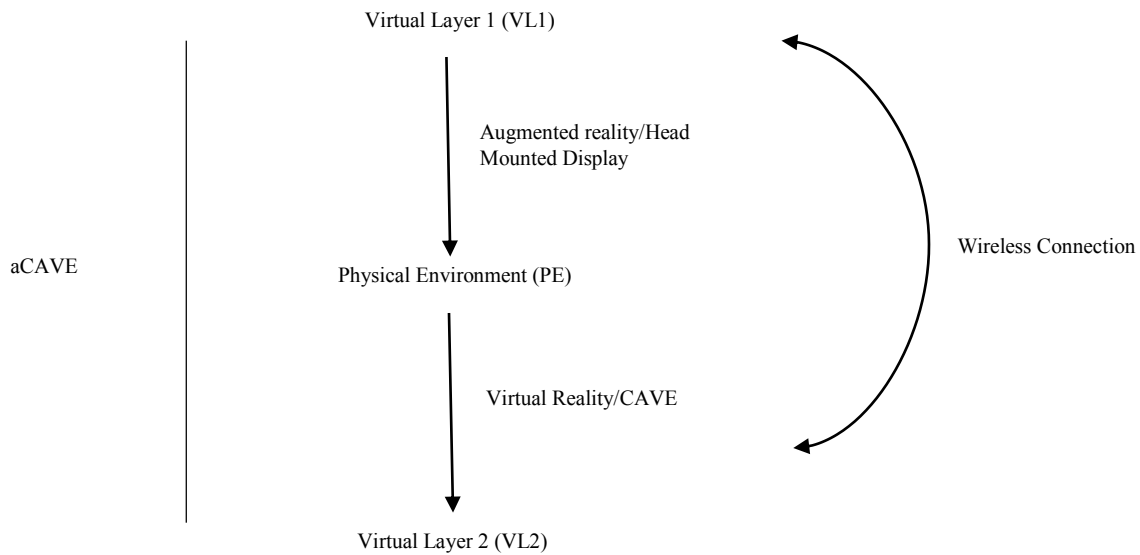


Figure 3 – aCAVE layers

occur inadvertently without conscious user command). Valente and co-authors [7] define these kinds of objects as “smart objects”. The aCAVE may also use these objects as “environment devices” [7] to output information.

The physical environment may house other physical objects, sensors, and actuators. The aCAVE may use these objects as direct input devices (i.e. through conscious user manipulation) or indirect input devices (i.e. sensor-based interactions that may occur inadvertently without conscious user command). Valente and co-authors [7] define these kinds of objects as “smart objects”. The aCAVE may also use these objects as “environment devices” [7] to output information.

### 3 RELATED WORK

Some studies have established that the use of just the technologies of VR or AR separately presents some limitations that need to be solved by integrating these technologies with others.

According to [8], for a good virtual reality experience, “it is very important that the participants have the feeling that they are really in the environment. This feeling of ‘being’ in a mediated environment is described as presence. Two important factors that influence presence are the level of immersion and the navigation method”. However, if we used just a CAVE, we would face some difficulties either in the immersion parameter such as the navigation one. In this paper, on the contrary from the conference [8], we proposed an improvement on immersion issues.

Another example of the limitations of the CAVE is presented in [9], where the authors stand that “the raw CAVE system doesn’t provide semantic understanding of players’ behaviors and emotions in real-time, which leads to a challenge for application developers to create a personalized virtual environment for various players.”

Aspin and Hoang Le [10] use the AR technologies as a complement of a CAVE-like environment. In his work, a tablet is used as a way to get further information than the CAVE could

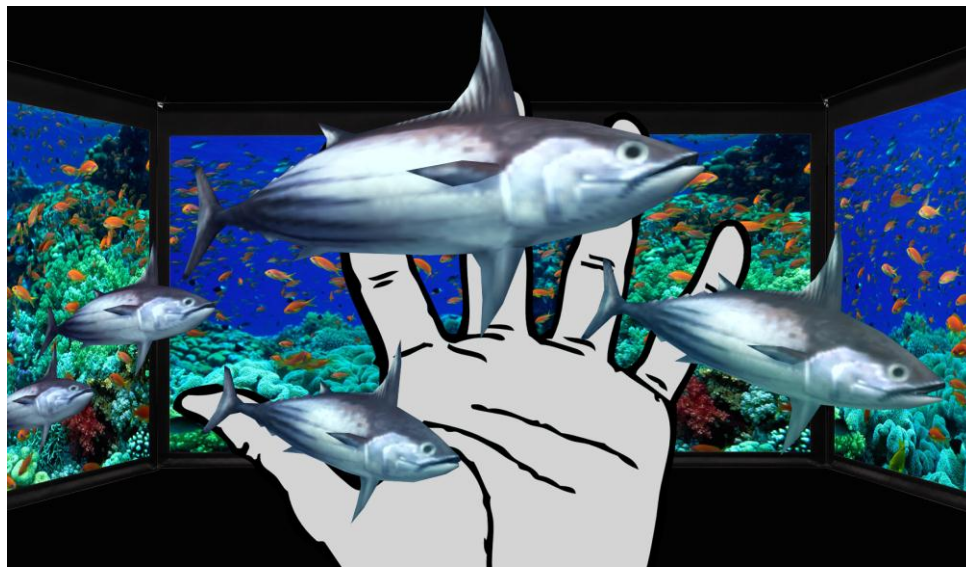


Figure 4 – Graphic representation of layer composition: virtual layer 1 (fishes), physical environment (hand), and virtual layer 2 (seabed on the background)

give. Their work demonstrates that is feasible to integrate AR and VR technologies.

In this way, this paper is motivated by works that point some limitations on the CAVE and try to provide some improvements that could be done.

#### 4 SYSTEM DESIGN

This section describes the system architecture we designed to realize the aCAVE concept. This system consists of a hardware setup and a sample application. The example application simulates an undersea diving.

The aim of the project was to test the technology setup of this new mixed-reality paradigm. We designed a virtual scenario to simulate a seabed (virtual layer 2). In virtual layer 1, the application contains a plane where it projects the smartphone video stream and the virtual fishes. The fishes keep on swimming in front of the user and have basic AI behavior – they follow a target location in the virtual space, and once they reach this location, the target location changes. The fishes are also influenced by the virtual scenario. For example, if the fishes pass through a stone or any other part of the scenario that occludes them, they disappear from the user's view. We developed the demo application with Unity3D.

##### 4.1 Virtual layer 2 (CAVE)

We built a CAVE theater (2.70m x 2.70m x 2.70m) made up of three rear-projection screens (i.e. left, right, and forward) as illustrated in Figure 2. The room that houses the CAVE contains a computer connected to the three projectors. The projected images have the same size. The three projection walls are arranged at a 90 degree angle to one another, as **Erro! Fonte de referência não encontrada.** 2 illustrates.

The movement around the seabed scenario is done through a joystick, where the user can go forward/backward and turn around.



Figure 5 - Mirror reflecting image from projector

##### 4.2 Virtual layer 1 (HMD)

As our goal was to develop a basic system so it could be easily replicated, we decided to use a simple Android smartphone. In this part of the application, the user could see the fishes swimming and in the background was projected the image from the camera, giving the impression that the fishes were at the same place as the user.



Figure 6 - GOOGLE Cardboard and smartphone

#### 4.3 System Architecture

Our system uses a client/server architecture that uses Unity3D's NetworkManager to manage the communication between the virtual layers. In a traditional multiplayer games that use this module, both client and server components are inserted in the same game scene. However, our applications do not follow this pattern – the game scene content is split between the client (e.g. fishes) and the server (virtual scenario).

When the user starts the application, there are three possible configurations: “client”, “server only” or “host” (Figure 7). We just use the “client” and “server only” configurations. The client was set as a player prefab that is only spawned if the application is instantiated in the “client” mode.



Figure 7 – Network menu.

In the “client” mode, the app spawn automatically the player prefab that contains the elements from Figure 8. This way, the user is only able to see those components.

The user should select the client option. The whole environment of the sea would be disabled and the prefab with the fishes and the package of GOOGLE Cardboard would be instantiated.

In the “server only” mode, the app does not spawn the player prefab and the user can only see the seabed environment, where he/she can walk through.

When the user walks through the environment of the server selection, the movement is sent to the client, changing the position or rotation of the fishes, so the user feel like he/she is really moving.

So, when the application is instantiated as a client, the player prefab will be spawned and the others objects will be deactivated and not shown to the client, just to the server. And the opposite will happen if the application is instantiated as a server.

With the application instantiated and the connection ready, the server can now listen to the commands from the user in the server application and sent to the client one.



#### 4.3.1 Client (HMD)



Figure 8 - Application working at the mobile device.

The client component is responsible for rendering and synchronizing the HMD content. This component works like a common application from a HMD and shows to the user the elements that are in front of the real elements.

As Figure 10 illustrates, the client component contains four elements: 1) the projection plane, 2) smartphone camera stream, 3) virtual fishes, and 4) virtual aquarium. The camera and the plane are influenced by user's head movement, which means that any head rotation also rotates these elements. The "projection plane" is an ordinary plane that occupies the entire user field of view. The application maps the smartphone video stream as a texture to this plane.

The virtual aquarium is composed of four parallelepipeds that held the fishes in a small region between the camera and the projection plane. The net effect is that the fishes are located between the user and the projection plane most of the time, helping to create a 3D depth illusion.

When the user moves his head (translation or rotation), the system detects this event and notifies the server, which in turn updates the orientation of the fishes (i.e. the opposite movement is applied to the fishes). For example, if the user walks forward in

the virtual environment using a joystick, the fishes will move in the direction of the camera. This way, we try to connect the virtual objects from the "client" mode with the virtual environment from the "server".

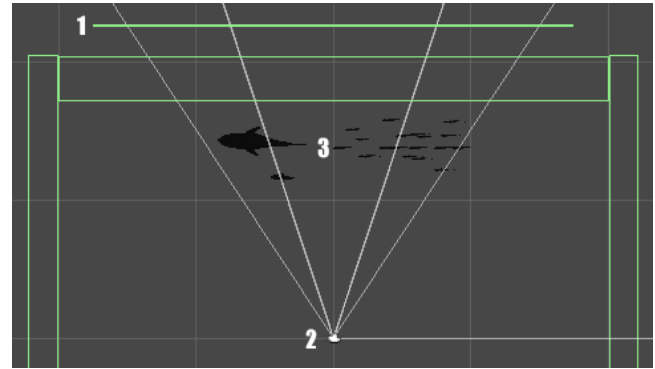


Figure 10 – Client application seen in Unity

#### 4.3.2 Server

This component is responsible for synchronizing the multiple projectors and for rendering the backstage scenario. Using regular IP connection, the module allows each movement generated on the computer (server) to be passed to the HMD (client).

#### 4.3.3 Connectivity between the virtual layers

Our architecture provides a communication system between the HMD, the CAVE. The solution uses the NetworkManager from Unity, a component for managing the network state of a multiplayer game as described in [11]:

"A Networking multiplayer game can run in three modes - as a client, as a dedicated server, or as a "Host" which is both a client and a server at the same time. Networking is designed to make the same game code and assets work in all of these cases. Developing for the single player version of the game and the multiplayer



Figure 9 - Seabed divided in 3 screens

version of the game should be the same thing.”

The big difficulty by using this component by our purpose was that in a multiplayer game, both client and server are inserted at the same game scene. This situation could not happen in our project, once the client should show the fishes and the server should show the whole scenario of seabed. To bypass this issue, we converted all the objects that we wanted to show to the client into a player prefab, since when “a player prefab is set, a player object will automatically be spawned from that prefab for each user in the game” [11].

So, when the application is instantiated as a client, the player prefab will be spawned and the others objects will be deactivated and not shown to the client, just to the server. And the opposite will happen if the application is instantiated as a server.

With the application instantiated and the connection ready, the server can now listen to the commands from the user in the server application and sent to the client one.

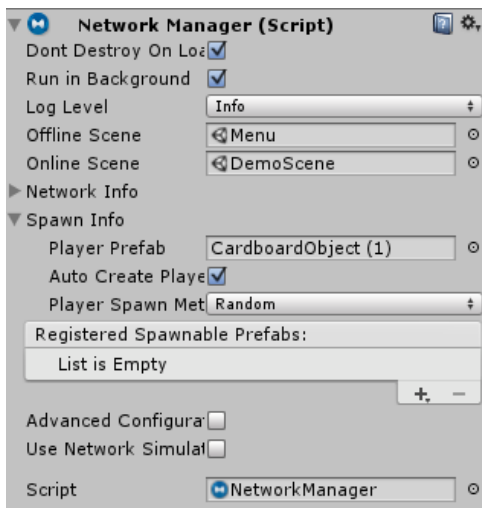


Figure 11 – Network Manager setup

## 5 PRELIMINARY EVALUATION

As the current system is still on the early stages, we conducted a preliminary qualitative evaluation with our system. This section presents the methodology and results.

### 5.1 Methodology

The goal of this evaluation was to get first impressions about the main project cornerstones: comfort, integration and immersion.

The methodology consisted of the following stages: user selection, pre-evaluation interview, users experiencing the aCAVE environment (demo stage), and post-use interview.

We selected four users (four men) for this first evaluation trial. These volunteers are undergraduate Computer Science of our university. Their average age was 22 and they are casual game players. In the pre-evaluation interview, we asked about their age, their game experience, and we introduced them the demo application and hardware equipment.

In the demo stage, we had the users experience the virtual environment for five minutes. In this stage, we observed the users and how they reacted to the environment.

Finally, after the demo experience was done, we conducted a post-use interview. This interview consisted of the following questions, where the answers are part of a Likert scale (0: very poor, 1: poor, 2: fair, 3: good, 4: very good, 5:excellent):

Q1: I felt uncomfortable side effects while using the system (e.g. headache, motion sickness, nausea)

Q2: The fish and the virtual scenario seemed to me as one integrated virtual environment

Q3: When I moved in the virtual scenario (CAVE), the fish also moved congruently

Q4: How would you classify the 3D depth experience that the system provided?

Q5: I enjoyed this aCAVE experience

These questions relate to the main project cornerstones (comfort, integration and immersion) as follows: Question Q1 refers to the comfort level that the user felt in the experience. Questions Q2 and Q3 refer to the feeling that the participants had about the integration between the virtual layers. Questions Q4 and Q5 relate to the immersion experience of the participants.

## 5.2 Results

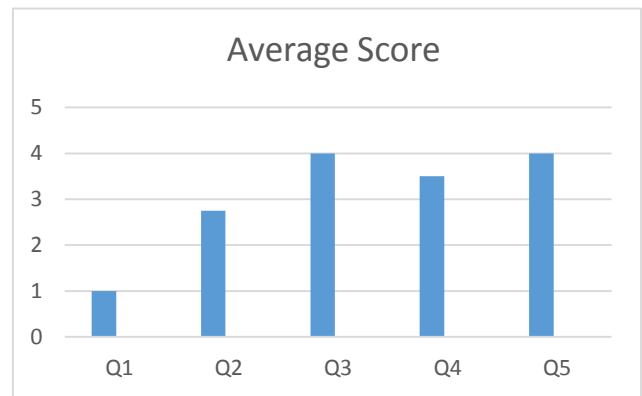


Figure 12 – Evaluation from the interview

Figure 12 illustrates the average scores of the post-use interview questions. This preliminary feedback (on this particular user group) suggested that the current aCAVE system provided “good” “immersion” and “comfort” to these users. Concerning “integration”, the response was average. We believe that the limitations of the HMD-smartphone setup contributed greatly to this feedback. The camera resolution and response delay are fundamental to achieve a truly integrated system. In the current version, as we are using a common smartphone (MOTO G) as HMD, its camera will manifest a certain delay in low light environments like the aCAVE system requires. Besides that, as the screen of the cellphone stays really close to the eyes of the user, any average resolution will be noticed.

	Interviewed A	Interviewed B	Interviewed C	Interviewed D
Q1	0	0	3	1
Q2	4	2	2	3
Q3	3	4	4	5
Q4	4	4	2	4
Q5	5	4	3	4

Figure 13 – Answers from the participants

In possession of these answers, we could map this result into a graphic based on the three cornerstones categories (comfort, integration and immersion). The comfort level was calculated as the complementary of the average of question Q1. The immersion level was the average between questions Q2 and Q3. And the integration is the average between questions Q4 and Q5.

The users felt a little awkward at the first seconds until they get used to the system and could enjoy the experience. They felt that the limitation of resolution and delay of the camera from the smartphone spoiled their experience with the system a little bit. However, they seemed to enjoy system and were thrilled with the future possibilities.

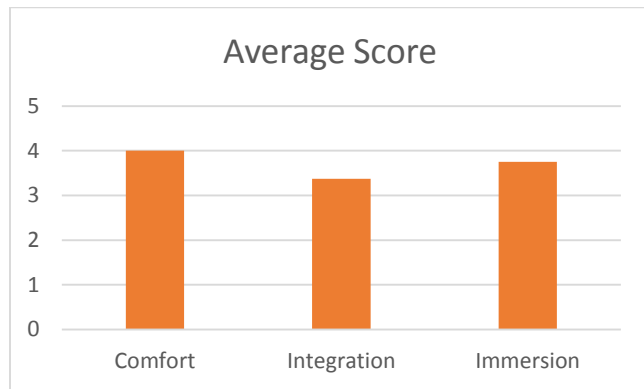


Figure 14 - Evaluation based on comfort, integration and immersion

## 6 CONCLUSIONS

In this work we proposed an extension to the concept of traditional CAVEs, merging their projections with a HMD layer. Due light occlusion, it is impossible to have real objects between two virtual elements. Our solution presents a framework capable of integrating two different virtual systems, enabling different entertainment possibilities and experiences.

While our system proved interesting results with the proposed test, there are still some limitations:

- The projection of the CAVE does not have 3D stereo effects, since the user is using a regular HMD. This can be solved in the future, developing a synchronization of the 3D CAVE projectors with the HMD embedded cameras.
- There is no match among the illumination of the virtual scenario with the real elements inside the CAVE space. For solving this problem it is necessary to include light sensors inside the CAVE space, in order to suggest a calibration to the virtual scenario lighting.

## REFERENCES

- [1] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The CAVE: Audio Visual Experience Automatic Virtual Environment," *Commun ACM*, vol. 35, no. 6, pp. 64–72, Jun. 1992.
- [2] T. Mazuryk and M. Gervautz, "Virtual reality-history, applications, technology and future," 1996.
- [3] D. Zeltzer, "Autonomy, Interaction, and Presence," *Presence Teleoperators Virtual Environ.*, vol. 1, no. 1, pp. 127–132, Jan. 1992.
- [4] J. Balaguer and E. Gobetti, "Leaving Flatland: From the Desktop Metaphor to Virtual Reality," *EUROGRAPHICS95 Tutor.*, no. 5, 1995.
- [5] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1993, pp. 135–142.
- [6] R. T. Azuma, "A Survey of Augmented Reality," *Presence Teleoperators Virtual Environ.*, vol. 6, no. 4, pp. 355–385, Aug. 1997.
- [7] L. Valente, B. Feijó, and J. C. S. do P. Leite, "Mapping quality requirements for pervasive mobile games," *Requir. Eng.*, pp. 1–29, Sep. 2015.
- [8] M. Lorenz, M. Busch, L. Rentzos, M. Tscheligi, P. Klimant, and P. Fröhlich, "I'm There! The influence of virtual reality and mixed reality environments combined with two different navigation methods on presence," in *2015 IEEE Virtual Reality (VR)*, 2015, pp. 223–224.
- [9] Y. Cai and Z. Shen, "i-CAVE: Intelligent Agent-Augmented CAVE System."
- [10] R. Aspin and K. H. Le, "Augmenting the CAVE: An Initial Study into Close Focused, Inward Looking, Exploration in IPT Systems," in *Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, Washington, DC, USA, 2007, pp. 217–224.
- [11] "Unity - Manual: Using the NetworkManager." [Online]. Available: <http://docs.unity3d.com/Manual/UNetManager.html>. [Accessed: 10-Jun-2016].