

First steps towards Live-action Virtual Reality Games

Alexandre Ribeiro Silva^{*}, Esteban Clua[§], Luis Valente[§], Bruno Feijó[†]

^{*}Instituto Federal do Triângulo Mineiro
email: alexandre@iftm.edu.br

[§]Universidade Federal Fluminense
email: esteban@ic.uff.br, lvalente@ic.uff.br

[†]VisionLab/PUC-Rio
email: bfeijo@inf.puc-rio.br

Abstract— In this paper we describe the main concepts of a new game genre (live-action virtual reality game) made possible by the advances of virtual reality technologies and context-awareness. This set of games requires that players wear HMD devices, from where they see a complete virtual world that is built using part of the physical configuration as the basic geometry and context information. Games keep both the virtual and real-worlds superimposed, requiring players to physically move in the environment and to use different interaction paradigms (such as tangible & embodied interaction) to complete game activities. As tracking physical elements is a key issue in the implementation of games, in this paper we also describe an architecture that addresses indoor navigation and tracking in live-action virtual reality games. The system we propose is based on infrared markers, working on the infrared spectral region that provides low optical noise and better precision when compared to traditional solutions based on fiducial markers. Furthermore, this paper describes our system and presents two case studies based on our architecture.

Keywords— virtual reality, indoor navigation, Oculus Rift, infrared optical tracking system, mixed-reality, augmented-virtuality, live-action virtual reality games

I. INTRODUCTION

In the past, virtual reality hardware (such as HMD devices) has been restricted mostly to research labs due to high cost and application specificity (high-end VR systems). Recently, there is a growing trend in industry to bring to the mass market (e.g., Oculus Rift, Samsung VR, HTC Vive). Hardware devices, besides being portable, increase the required computing power and sensor capabilities to applications, opening up possibilities for mainstream games and entertainment.

Hardware advances enable new experiences with virtual reality, most of them as demo installations and theme park facilities for multiplayer systems (e.g. The VOID [1], Artanim Interactive's VR system [2]). In July-2015 we proposed a new genre of game based on mixed reality, context awareness and embodied interaction, called "Live-action Virtual Reality Game" [3]. In type of game, players wear HMD devices and see a virtual world that is constructed using parts of the physical elements as the basic geometry and contextual information. Physical objects that reside in the physical world are also mapped to virtual elements. Live-action virtual reality games keep the virtual and real-worlds superimposed. We say that games are "live-action games" because the players need to move in the environment to complete game activities, using

different interaction paradigms (such as tangible and embodied interaction). Setup enables the players to touch physical architectural elements (such as walls) and other physical objects, "feeling" the mixed-reality environment. We remark that interface is completely different from Augmented Reality, since the user does not see any real elements projected in the virtual stage.

One of the main goals in virtual reality applications is to provide means to immerse the user's senses in an artificial virtual environment (VE) through an interactive experience. A key factor regarding how this interactive immersive experience is successful refers to the sense of presence. Presence can be commonly defined as "a sense of being in a VE rather than the place in which the participant's body is actually located" [4]. Properly integrating real-world information (such as physical objects, people, and other devices) with the virtual world (through tracking technologies) is an important technological aspect related to immersion. Other important aspects include the extent of the field of view, number of sensory systems stimulated by the VR system, the quality of rendering in each sensory modality, among other aspects.

In live-action virtual reality games, the players are able to experience the mixed-reality with at least five senses: sight, touch, hearing, the kinesthetic sense (e.g., sense of movement and body awareness), and the vestibular sense (e.g., sense of balance). The stimulation of sensory modalities may help reinforcing the feeling of "presence" that the player perceives in the environment. Section II describes the main concepts of live-action virtual reality games.

This paper extends our previous work [5] with a revised definition of live-action virtual reality games and a detailed presentation of basic components and characteristics. We also present a more complete discussion about how this new genre compares other related game genres. Moreover, we present more details of a navigation system, which is one of the most important features for this new type of game.

Tracking physical features and physical objects is of utmost importance in this type of game. Traditional solutions of tracking problems usually explore magnetic sensors, ultrasound systems, and even mechanical devices. However, in many of solutions the results have inherent imprecisions due to the technology, which contribute to degrade the sense of presence. In this paper we describe a solution to indoor navigation and tracking of physical elements to implement live-action virtual reality games, within a specific scenario configuration. Our

system uses IR markers (i.e., infrared light-emitting diode markers) to create the virtual representation of the physical environment. The game designer places these markers in the environment as a pre-processing step. In sequence, a calibrating stage is started when a user walks in the physical environment wearing the respective hardware in order to track and register all markers. Our system records the stored marker information to generate the virtual content, which includes the virtual geometry and other elements that the game designer wishes to add to the game. Our system could be applied to entertainment and serious games (e.g., police and fireman training). Due to the infrastructure required to create the mixed-reality environment, in the current version our system should be used in a dedicated physical place where the tracking infrastructure is deployed. For future versions, we plan to improve the techniques applied to simplify the installation and calibration processes required to play the game.

Some similar industry initiatives [1, 2] have appeared almost simultaneously with our work [3], but we were unable to access technical documentation except for advertising websites. However, live-action virtual reality games include aspects related to context-awareness and pervasiveness, which (as far as we know) we believe are not present in those industry initiatives. Also, we are the first ones to propose definitions and conceptual design characteristics that can help formalizing and developing this new type of immersive game in a more robust way.

This paper is organized as follows: Section II presents concepts of live-action virtual reality games. Section III presents related works. Section IV describes our architecture and Section V presents two case studies that we developed to test our architecture. The first study aimed to test if the tracking system worked properly. The second study tested the degree of acceptance of the system within a group of twelve players. The main objective of this second study is having the player walk in the environment with quick and accurately moves. Finally, Section VI presents conclusions and future works.

II. LIVE-ACTION VIRTUAL REALITY GAMES

In live-action virtual reality games players wear HMD devices and see a virtual world that is built using three main information sources: architectural elements (e.g. doors, walls), physical objects, and context information. The key concepts of this kind of game are: game stage, information flow, context-awareness, and interaction paradigms. These concepts are presented through the seven conceptual design characteristics of Live-action VR Games (C1 to C7, below). These characteristics not only formalize this new game genre more deeply, but they also help game designers develop more immersive and robust live-action VR games.

A. Mixed-Reality Environment (C1)

Live-action virtual reality games create a mixed-reality environment, which we name as the *game stage*. A “mixed-reality environment” can be defined as an environment that fuses virtual and real-world (physical) information. Fig. 1 illustrates a simplified “reality-virtuality” continuum that Milgram and co-authors [6] proposed. This continuum

represents a spectrum describing possible combinations of virtual and real elements, resulting in different kinds of “mixed-realities”.

The *game stage* is a kind of “augmented virtuality” (Fig. 1) because the virtual world is constructed through real-world information sources and enriched through virtual and real-world information. We roughly categorize these real-world information sources as “physical structure information” and “context information”. Live-action virtual reality games output information in the virtual and physical worlds. Physical world output can take form as *real-world effects*, such as manipulating environment properties (e.g., temperature, ambient light), generating vibrations, and outputting smells. Live-action virtual reality games are on the right side of Fig. 1.

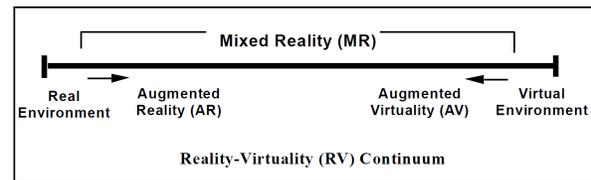


Fig. 1 Reality-virtuality continuum, extracted from Milgram and co-authors [6]

B. Physical Structure as Input (C2)

A live-action virtual reality game uses physical structure information as basic building blocks to create the game stage. When a player enters the physical environment where the game happens (e.g., a room), the game tracks all architectural elements (such as walls) to use as the basic geometry to create a virtual 3D environment. After creating the virtual structure, the game may augment this raw structure with virtual content. Metaphorically, we compare this process to the process of mapping images (textures) to raw polygon data in 3D environments. The game keeps the virtual and real worlds superimposed as the player moves in the environment, creating the game stage.

The game may also track physical objects that are part of this physical environment and map them in the virtual world as 3D models. Examples of physical objects are furniture, interactive objects carried by a player, and even the players’ own bodies. However, the player does not see the physical world and physical objects – the player only sees virtual representations through the HMD. The virtual building process may happen in realtime or before the game session (as a preparation step). In this preparation step, the game system may track all relevant physical features to generate a virtual world to be optimized later by an artist, who may augment the virtual world with other virtual elements.

C. Context Information as Input (C3)

Context-awareness is a key component of live-action virtual reality games. Dey [7] defines context as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a player and an application, including the user and themselves”. A system can

be considered as “context-aware” if “*it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task*” [7].

In this regard, “context-awareness” means that a live-action virtual reality game is able to adapt the gameplay according to the current context conditions. Context-awareness enables live-action virtual reality games to collect information from the real-world and from players (while the game session is happening), using these data as input information. Live-action virtual reality games use context information for two main purposes:

- Augment the game environment with dynamic information that originates in the realworld;
- Generate game content dynamically to create unpredictable game experiences, giving rise to emergent gameplay.

Here are some examples of information that we consider as “context” in live-action virtual reality games (this list is by no means an exhaustive list):

- Physical environment information. Examples include temperature, humidity, and lighting conditions, time of the day, weather information, and environmental noise;
- Existing research on Human-computer interaction [8] suggests that the concept of context can be enough to consider the ways a user interacts with a device as context information. we include interactions with input devices and with the physical environment where the game takes place;
- Player information. Examples include physiological state, personal preferences, social network profiles, and personality traits;
- Information derived from the possible forms of relationships and interactions among players – the social context [9].

A live-action virtual reality game is able to sense context information through several means. Some are:

- The player carries or wears devices that sense context from the environment and/or from the player (e.g., physiological information);
- The physical place where the game happens may host customized devices that are equipped with sensors – defined by us as smart objects. Smart objects may be connected to other smart objects and to central game servers. devices may provide interaction interfaces to the players (Section II.E) and may be able to output information (such as audio and images) and create real-world effects (Section II.D);
- The physical place may be augmented with dedicated infrastructure that enables the game to sense context information. Examples include tracking technology and sensors collect environmental information. In this solution, objects are hidden (“invisible”) from players, part of the overall game infrastructure. On the other

hand, smart objects are complex, visible physical objects that reside inside the game stage;

- The game queries remote information about the player based on the player identity (e.g., social network profiles).

D. Real-World Effects as Output (C4)

Live-action virtual reality games may output information through wearable devices, mobile devices, and *environment devices*. devices may output digital media information and create real-world effects (e.g., vibrations, smells, smoke). These devices may be connected to other devices that comprise the game infrastructure, such as input devices, game servers, and other output devices. devices may also be input devices and context-aware.

The wearable devices are worn by a player, which may also work as input devices and capable of sensing context information. Examples include HMDs, isolating or non-isolating headphones, gloves, vests, and clothes. They might provide haptics feedback as output, for example.

The mobile devices are devices that a player may carry around, touch, and throw away. These devices may output digital information and may be equipped with actuators, motors, and other components capable of generating real-world effects.

Environment devices are objects that reside in the physical environment and generate effects or behavior in the physical world [10]. Some possibilities for environment devices are (this is by no means an exhaustive list):

- Simple output devices, not limited to digital media (e.g., audio speakers, smell generators, weather effects, heat generators, smoke generators);
- Mechanical or electrical devices equipped with actuators (e.g., sliding doors, elevators, wind generators);
- Smart objects equipped with output hardware. An example is the trash can in Fun Theory [11].

E. Full Interaction (C5)

In live-action virtual reality games, players interact with the game and with other players through multiple modalities (e.g., voice, body movements, and gestures), ordinary physical objects, customized input devices, and context-aware devices, supporting tangible, embedded, embodied, and context-aware interaction paradigms.

The player may interact with the game by manipulating mobile devices equipped with sensors and ordinary physical objects (e.g., wood sticks, rocks). The mobile devices may be equipped with networking capabilities. These objects and devices realize the “tangible object metaphor”, which stands for “*human-machine interfaces that couple digital information to physical objects*” [12]. These interfaces can be implemented through tracking technologies or sensors. Examples of suitable sensors for this purpose are RFIDs, magnetic sensors, proximity sensors, and light sensors.

On a general context, for instance, tangible objects could be pieces of augmented table-top games [13]. Another example is using wood sticks (an ordinary object) that the game tracks to represent swords (the virtual representation in a game).

Tangible objects may also have embedded computing capabilities. Existing research suggests that tangible objects contribute to increasing player immersion in games [14]–[16].

In live-action virtual reality games, players may touch, manipulate, and throw these objects and devices (among other possibilities) to interact with the game. The player does not see objects and devices directly – the game presents their virtual representation to the player through the HMD. Objects and devices may be carried by the player or may be spread in the game stage. For example, a game may present a puzzle that requires finding an object in the environment and placing it on top of another object or device that resides in the environment.

Live-action virtual reality games may use customized input devices or general-purpose controllers as interaction interfaces. These devices may include joysticks and custom hardware that features a number of affordances that are specific for the game. An example of a customized controller is a controller modeled as a gun. An example of a general purpose controller is the Razer Hydra [17].

Another interaction possibility in live-action virtual reality games is through wearable devices, devices worn by a player, providing embedded, embodied, and context-aware interaction. Examples of devices include smart bands, motion capture devices, and brain-machine interfaces. Wearable devices usually are context-aware and may be able to sense the player's physiological state to use as contextual input information.

Players in live-action virtual reality games may be required to physically walk to move in the mixed-reality environment to complete game tasks. In this regard, players act out their own avatar, which is a form of full body interaction. Full body interactions in live-action virtual reality games can be mediated by wearable devices or not. With the expression “full body interactions not mediated by wearable devices”, we refer to interaction paradigms made possible by devices such as the Kinect sensor [18].

Another way in live-action virtual reality games takes form as smart objects (Section II.C) enable interactions of “implicit nature” [19]. In implicit interactions, the interaction interface is invisible (i.e., based on sensors). Schmidt [20] defines an implicit interaction as “*an action, performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input*”. For example, a game stage may host a smart object containing a proximity sensor that opens a door when someone gets close to it. Implicit interactions occur inadvertently from the point of view. On the other hand, explicit interactions occur through direct (conscious) player commands, meaning that the player has conscious intention to start something to achieve a goal. As the player does not see the physical world, live-action virtual reality game designers should be careful when designing

implicit interactions to avoid accidents that might injure players.

F. Mixed-Reality Infrastructure and Management (C6)

A game stage requires a dedicated physical place to create a customized physical installation, due to the infrastructure required to support the mixed-reality in live-action virtual reality games. Before game sessions take place, the required infrastructure is deployed to the physical environment, as a preparation step. Live-action virtual reality games designers and developers may exploit this requirement to their advantage by deploying custom hardware (such as smart objects) in the environment that otherwise would not be generally accessible to end users, which might help in creating more sophisticated game experiences.

Using a dedicated, customized installation also helps to implement an “uncertainty handling policy” [10] to remove (or minimize) problems caused by limitations of the involved technologies, such as networking and.

In live-action virtual reality games, the problems of “tracking physical world features” and “keeping the virtual world properly overlaid on the physical world (e.g., synchronized)” are of central importance.

Traditional computer vision techniques (such as the ones based on QR codes) do not have the required accuracy to address these problems. For this reason, live-action virtual reality games usually require other solutions based on other kinds of sensors and tracking techniques.

Live-action virtual reality games may require ongoing supervision while game sessions happen. This real-time supervision, also known as “orchestration” [21] aims to anticipate and/or correct technical issues that may happen during a game session, so that the game experience is not broken due to these problems. Orchestration also can be used to help players who are experiencing difficulties in playing the game.

G. Multi-Sensorial Presence (C7)

Important aspects of presence include the extent of the user field of view, number of sensory systems stimulated by the VR system, the quality of rendering in each sensory modality, the quality of tracking, among other aspects [4]. Some possibilities to stimulate presence in live-action virtual reality games are:

- Live-action virtual reality games stimulate sight through mobile, lightweight HMDs.
- Live-action virtual reality games may stimulate touch through wearable devices (capable of providing haptic feedback), the underlying physical structure, and physical objects (e.g., touching physical walls, tables, and holding small objects).

- Hearing may be stimulated by isolating or non-isolating headphones and smart objects.
- Live-action virtual reality games stimulate the kinesthetic and vestibular senses by requiring players to move in the physical environment and by using interaction interfaces based on gestures.
- In a customized game stage, live-action virtual reality game designers may deploy smart objects and output devices that stimulate other senses, such as smell and the sense of heat.

III. RELATED WORK

This section presents game genres that might be related to live-action virtual reality games (III.A), and other initiatives that are similar to live-action virtual reality games (III.B)

A. Game Genres

The Virtuality Group [22] provides earlier examples (in 1990) of using VR in consumer games, as virtual reality arcade machines. machines contained a HMD, speakers, microphone, and joysticks. That system was able to track head and joystick movements through a magnetic tracking system. Games released in platform include Dactyl Nightmare [23] and Grid Busters [24]. Another attempt at bringing VR-based games into the mainstream came from Nintendo and its Virtual Boy [25] in 1995. Although the system was original, it was a commercial failure. King and Krzywinska [26] suggested that the high price and physical discomfort while playing the game contributed to the demise of this device.

The earlier virtual reality games focused mainly on immersing the user's visual sense in the virtual world. The users interacted in the virtual environment with limited mobility – physically walking in the environment was not possible as the equipment running the simulation was not portable.

1) Augmented Reality Games

Live-action virtual reality games are fundamentally different from games based on “augmented reality” (left side of Fig. 1), because in live-action virtual reality games the player does not see any real element from the physical world. In augmented reality games the player sees the physical world (through HMD or a mobile device camera) and virtual contents that are placed on top of physical world elements. On the other hand, in live-action virtual reality games the player sees a virtual world that is built based on part of the physical characteristics, including its architectural elements and contextual information. A classic reference on augmented reality can be found in [27].

A classic, early example of augmented-reality game is ARQuake [28], an indoor and outdoor game where player wears a backpack containing a laptop, an orientation sensor and a GPS sensor, which allows the player to walk freely in a physical environment. The player uses a see-through HMD that is connected to these devices. The ARQuake tracking system is based on GPS, digital compass, and fiducial markers. The game takes place on a physical environment that is modeled as

a Quake 3D level. When the game is played in the real world, ARQuake uses this 3D model to superimpose virtual world information into the physical world.

2) Pervasive Games

The term “pervasive games” refers to games that are played in the real world, exploring mobility, mixed-realities, and context-awareness, among other aspects.

The literature lacks consensus on definitions and formalisms about these kinds of games [10]. Also, there are terms in the literature that are used as synonyms of “pervasive games”, such as “mixed-reality games”, “ubiquitous games”, and even “augmented reality games”. In practice, the literature presents various interpretations and scopes for defining what “pervasive games” mean – for a detailed discussion on these issues, the reader should refer to [10], [29]. Valente and co-authors [10] some characteristics commonly found in pervasive games, which we reproduce below:

1. The game is constantly coming back to the real-world, which means that the game is played in physical places and it is not constrained to stationary computing devices;
2. The physical world (places, objects) is part of the game and it is combined with the virtual world, creating a mixed-reality;
3. Mixed reality is always existent and it is created through pervasive computing technologies (e.g., sensors, context-aware systems);
4. The spatial mobility occurs in a physical “open” environment, that is, the “game world boundary” is not “well-defined”, and sometimes it can be unconstrained;
5. The players use mobile devices (e.g., smartphones, tablets, custom hardware) to interact with the game and with other players;
6. The game may last for several days or weeks (or more), blending with the daily lives of players. The game may define a persistent world that progresses without player intervention. If some important event happens in the game, the game may notify the player to take action. These aspects are not mandatory;
7. The game may focus on promoting social interaction among players. Social interaction in a pervasive games may happen directly (face-to-face interaction) or indirectly (mediated through technology). This aspect is not mandatory, as pervasive games may be single-player games.

Despite all these very particular characteristics, there are similarities when comparing “pervasive games” and “live-action virtual reality games” That is, pervasive games may also use elements that we described in Sections II.B to II.F. Some of these similarities are:

- Live-action play, which is more prevalent in pervasive game genres such as technology-based LARPs [30];
- Game activities where players move in the physical world;

- Gameplay based on context-awareness;
- Interaction paradigms, such as tangible interactions and embodied interactions, found in some games (e.g. REXplorer [31], Spellbound [32]).

However, the key differences among these two genres are how the game constructs the mixed-reality environment and how the player experiences mixed-reality. These differences can be identified on the virtuality continuum (Fig. 1), in which pervasive games are near the left side of the scale and live-action virtual reality games are near the right side of the scale. In fact, pervasive games are based on the idea of a real world (indoors or outdoors) that is augmented with virtual content. On the other hand, in live-action virtual reality games, players do not see the real world, that is, they see a virtual representation of the world that was built by the game using physical elements and context information.

3) Alternate Reality Games

The term “alternate reality game” (ARG) may suggest that these games bring the player to a kind of reality that is very different from a real-world setting.

This idea is metaphorically similar to some of the ideas in live-action virtual reality games, but these game styles are very different. ARGs suggest a surrealistic setting where the game denies its existence “as a game”.

The main slogan of those games is “this is not a game”. ARGs use the real world as a platform and create a comprehensive interactive narrative, as massive puzzles that span different media such as web sites, emails, and phone calls. Game masters create (real-world and virtual) content and steer the story according to players’ reactions. The game is purposely ambiguous, so that players always question the game activities are indeed part of the game, or part of real-world life.

This includes discovering how to enter the game and guessing if it is over.

An example of early ARG is The Beast [33], which was part of a marketing campaign for the movie A.I. Artificial Intelligence [34]. The game began with a question, “Who killed Evan Chan”, and then evolved to an interactive story that had been deployed over the internet and the real world. The game itself was not advertised as a game, and its entry-point was hidden in A.I. movie trailers and posters. After following the clues, the player could access “real-world elements” (like voice mails from the game) that opened-up the gate for the storyline. The game designers have created fake websites and other multimedia content to support the game through puzzles and other interactions. Also, sometimes the game would make phone calls to the players. Some researchers classify ARGs as a kind of “pervasive game” [35], [36]. The reader should refer to [37] for more information about ARGs.

B. Tracking Systems

Considering tracking devices, [38] presented a hybrid system that combines outdoor tracking technologies (such as

GPS) with indoor tracking systems based on fiducial markers. The fiducial markers are placed on ceiling and walls to track the position.

Kim and Jun [39] proposed an indoor location system based on computer vision for navigation in augmented reality applications. Their calculates automatically after recognizing sequences of images taken in the indoor environment. This strategy creates augmented reality seamlessly by overlaying location information on the user’s view. According to [40] the solutions presented [38] and [39] require the following pipeline:

1. Image acquisition;
2. Detection of potential markers and discard of obvious non-markers;
3. Marker identification and decoding;
4. Calculation of marker poses.

Our system shares some similarities with the proposals of [38] and [39]. However, we use an optical system to decrease the computational cost required in the recognition stage. The recognition stage corresponds to the detection of potential markers and discard of obvious non-markers. The optical system is based on infrared markers instead of fiducial markers. We are aware that recognizing fiducial markers can be very fast due to the current state of algorithms and optimizations to fulfill this task. Yet we opted to create a system that uses modified cameras to capture only infrared light, which makes the marker recognition process easier (as this eliminates background details and background noise in captured images) and more robust, as it diminishes possible interferences caused by environmental lighting. Also, optical systems can employ a variety of detectors to acquire different forms of light patterns (such as ambient light and infrared light). Infrared light is often used in these systems to prevent interferences with other activities. This kind of system is able to track objects over a larger area with fast processing, but they can be heavy and expensive [41].

C. Industry initiatives with similar concepts

There are some industry and academic initiatives that have similarities with the concept of live-action virtual reality games that we present in this paper. However, information about these initiatives is scarce and only available as advertising websites and short videos.

The first one is “The VOID” [1], which defines a mixed-reality based on the real world structure, where the players move and walk. The environments where the game happens are named “gaming pods”. They intend to provide different kinds of sensations for players, including “*elevation changes, touching structures & objects, feeling vibrations, air pressure, cold & heat, moisture, simulated liquids and smell*” [42]. Players will use a proprietary HMD and wear proprietary vests and gloves, both capable of providing haptics feedback. The players do not see the physical world.

Another related initiative comes from Kenzan/Artanim Interactive [2], [43], who named their vision as “real virtuality”. In “real-virtuality”, players wear HMDs and have

their body and movements captured by motion capture devices installed in the physical environment installation. The players also have to walk physically to move in the mixed-reality environment. Players may also carry physical objects inside the environment, while the game displays their virtual representation (similar to the idea of interacting with physical objects described in this paper).

IV. AN INDOOR NAVIGATION SYSTEM FOR LIVE-ACTION VIRTUAL REALITY GAMES

This section presents an indoor navigation system that we developed for live-action virtual reality games. In our system, the user wears a tracking system hardware and wanders around a room equipped with several markers (Fig. 2). This section describes the tracking hardware (IV.A), the game stage configuration, and the system processing pipeline.

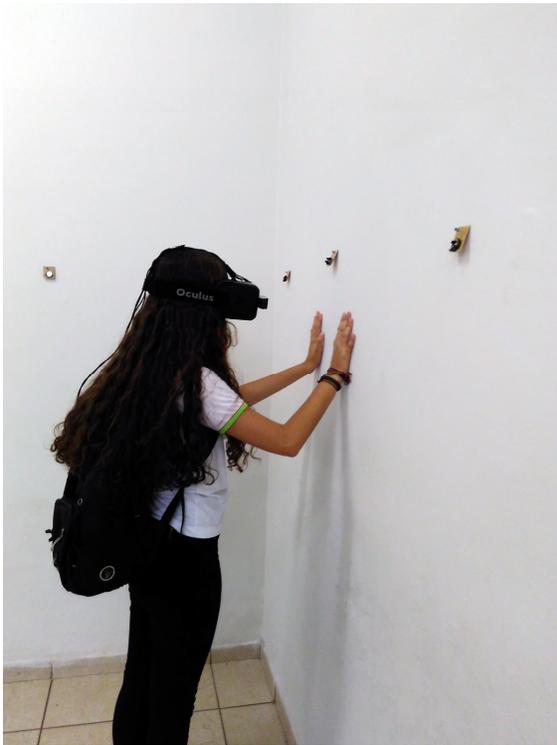


Fig. 2 User wearing the tracking hardware in the game stage

A. Tracking System Hardware

The tracking hardware consists of an Oculus Rift device equipped with two regular cameras (one for each eye position), which have been modified to capture only infrared light Fig. 3. These devices are connected to a notebook computer, all of them stored in a backpack. This setup enables a user to wear the hardware and walk in the game stage (i.e., mixed-reality environment) without restrictions – the equipment has no attached wires that might limit movement.

The system uses three ultrasonic distance sensors for accuracy calibration (e.g., left, right, and top directions). To

accomplish this task, we implemented a camera calibration method described in the OpenCV documentation [44]. The system has a compartment (i.e., an attached box) that is able to store four sensors, which can be customized according to application requirements. Examples of sensors that can be stored in this compartment are sensors for gas, light, temperature, and humidity.

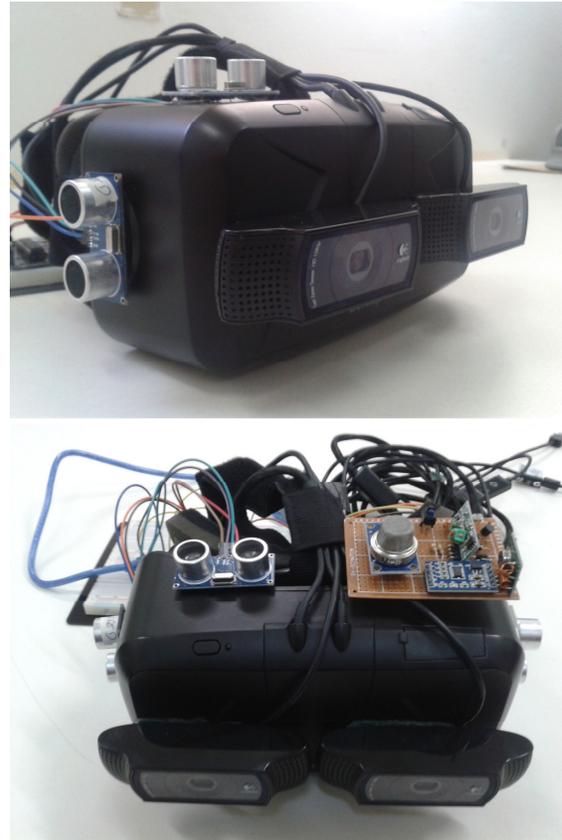


Fig. 3 Tracking system.

B. The Game Stage

The game stage consists of an indoor physical place (e.g., a room) that contains two types of objects that we use for tracking: static markers and smart. Objects are composed of IR markers, while smart markers contain embedded sensors. Fig. 4 presents a schematic view of a physical environment equipped with these markers.

Our system uses these IR markers to produce some of the virtual content that is part of the game stage and to calculate the user position and orientation in the environment. To calculate user position and orientation, the tracking system uses the modified cameras to detect the IR markers through stereo matching process.

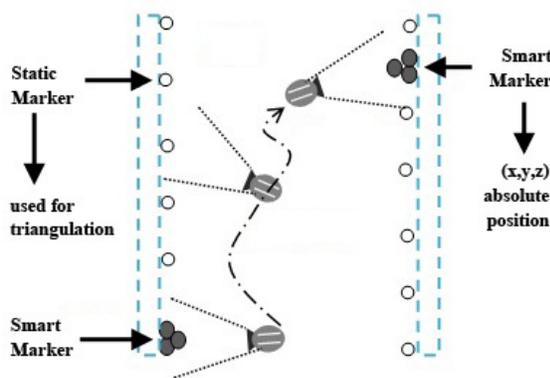


Fig. 4 A user walks in a sample environment with walls (dashed lines), smart markers (shaded three-ball objects), and static markers (white ball objects).

1) Static Markers

The “static markers” (Fig. 5) are generic IR markers spread throughout the physical environment. The system uses these markers as a guide to estimate the user position (e.g., x, y, and z coordinates) in the physical space through triangulation algorithms. These markers have only one infrared LED emitter and use a standalone battery. To calculate user orientation, our system uses the static markers and Oculus Rift’s accelerometer and gyroscope.



Fig. 5 Static Marker

2) Smart Markers

The “smart markers” (Fig. 6) are composed of infrared LEDs arranged as a matrix, and embedded sensors. These markers have two purposes in our architecture: the first is to work as an “absolute position marker” – an anchor in the physical environment with known location. The second purpose is to enable the system to sense context information from the environment.

The infrared LED matrix in these markers can have variable size, according to the game/application requirements. For example, the marker in Fig. 6 has a 5x5 LED matrix. In the current version, these markers use radio frequency receiver/transmitter technology (RF-433 MHz) to communicate with other devices responsible for changing the matrix contents in realtime.

The current version of our system uses a computer with a transmitter to emit a signal to the marker containing the marker identifier (ID) and the content that the matrix should display. These markers have an internal controller that is responsible for turning the matrix LEDs on and off, when required.

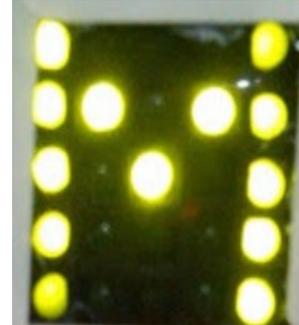


Fig. 6 A smart marker using a 5x5 LED matrix. The LEDs are arranged to represent the letter “M” in this example.

The main advantage of using markers based on infrared light is the lower computational cost to capture, extract, and recognize the marker, as well as greater accuracy, when compared to other available solutions for this kind of problem (i.e., fiducial markers). In our system, the complexity of the process to capture, extract, and recognize a marker is lower because the system works with images captured with infrared cameras, which is less complex than traditional RGB images. In other words, our system works with binary images that have no background. We present an overview of this process in Section IV.C and technical details in Section V.A).

The smart marker works as an “absolute position marker”. When we configure the physical environment with smart markers, we measure their location in the physical place and insert this information in a database that is available for the game or application based on our system. This means that the system knows the position of all smart markers before the game starts. A game session begins with a user near a smart marker, so that the system reads a known location to start calculating user positions (using static markers and relative position among them). As the user walks in the environment, the calculation of the user positions may become slightly incorrect due to error accumulation. Then, as the user bumps into a smart marker while walking in the environment, the system uses their known location to adjust user position estimation accordingly if necessary. This technique is similar to the one that Sezgin and Sankur [45] proposed to enhance accuracy of user position calculations.

Smart markers can be customized with a variety of sensors to enable the system to enhance context information. Fig. 7 illustrates a smart marker equipped with an infrared sensor, accelerometer and gyroscope. The smart markers may also work as a platform for tangible interaction (Section II). For example, in a fireman training application a smart marker may represent a device responsible for sensing toxic gases, smoke, and environmental information such as temperature and humidity. Although the user holds the marker (or an object made of several markers), the game displays to the user a

virtual representation of this object. As these markers have small sizes, they can be linked together to provide a network of information, such as pieces of trays in arm or a set of grenades in action games. These markers might also work in a similar way as traditional augmented reality markers – providing a physical base to augment with virtual content.

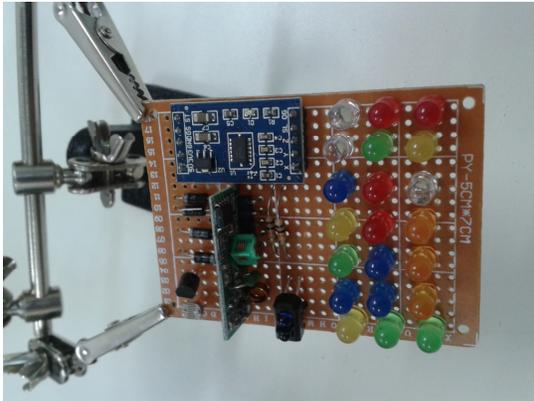


Fig. 7 Smart marker with embedded sensors.

C. System Processing Pipeline

This section presents an overview on the processing pipeline that the tracking system uses (Fig. 8). In Section V.A we present technical details about the tasks in this pipeline can be roughly grouped into categories: 1) image acquisition, 2) computer vision processing, 3) data acquisition and position calculation, and 4) VR/AR navigation. We describe these categories as follows:

1. Image Acquisition (Infrared Marker Detection): The system uses the cameras to capture the IR markers. As the cameras have attached filters to capture only infrared light, this process generates two-color images (black and white), which enables the system to segment IR markers by thresholding. In the resulting image, the IR LEDs are denoted by high brightness regions (white) and the background is denoted by low brightness regions (black).
2. Computer vision processing: The computer vision algorithm is responsible for image segmentation and background separation. This stage determines marker positions and removes camera distortions, as well as distinguishes the static markers from other markers that the system uses. The system is able to identify different markers by applying feature detection and recognition techniques. After a marker is identified, its position is marked in space map.
3. Data acquisition and position calculation: After identifying the markers and extracting data, the system starts data acquisition and user position estimation. The user position is calculated through stereo triangulation. Assuming that the game/application

starts with a known user position (through absolute position markers), the system uses an angular estimation for calculating the relative positions of static markers. The position data and corresponding marker information are stored in a database and marked in a 3D virtual environment, which allows a game/application developer to create a mixed-reality environment for the game/application based on these location data. In other words, this stage is also responsible for the correct mapping of virtual elements into the real environment.

4. VR/AR navigation: The navigation algorithm is responsible for superimposing the previously stored marker data (step 3) with the real physical markers that are spread in the physical environment. The idea is allowing the user move in the virtual world (created by the game/application developer) simultaneously while he moves physically in the real world. After this mapping process, the game/application developer is able to place 3D objects in the virtual world with the augmented reality markers, thus creating an augmented reality layer.

V. CASE STUDIES

This section discusses two experiments that we conducted to validate our indoor navigation system. The first experiment is a proof of concept test (V.A), while the second is a more sophisticated game-based demonstration conducted with twelve users (0).

A. Proof of Concept Test

The first experiment is a proof of concept to verify if our proposed architecture is suitable for practical applications such as games, entertainment, and training. The test application captures IR markers and extracts their information to locate a user in an environment. We distributed 16 static markers in a closed room (approximately 50cm apart from each other) and 4 smart markers positioned at known locations. This experiment had two stages.

The first stage comprised three steps: 1) Marker recognition; 2) Discrimination between static markers and smart markers; and 3) 3D mapping. The first step tests the marker tracking. Besides tracking markers, we were also interested in testing how far (the distance) the tracking system would be able to track a single marker without compromising accuracy. The second step process the tracked information to identify the different types of markers used in our system. The third step uses marker information to create a 3D map that represents the physical environment as defined by the markers.

The second stage corresponds to depth calibration, where we observed how accurate would be the superimposition of the 3D map into the physical world.

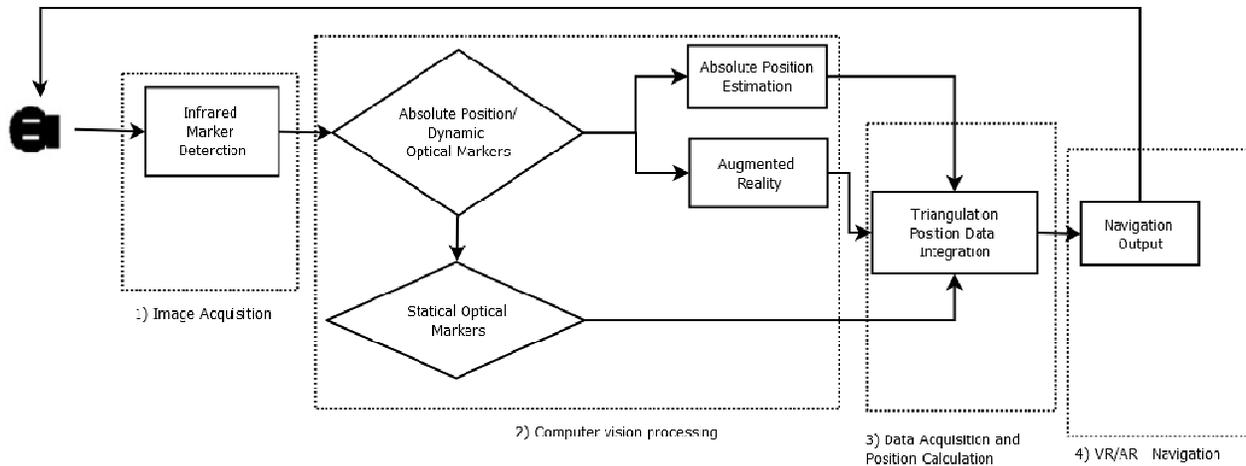


Fig. 8 System processing pipeline

1) First Stage

This section describes the three steps that comprise the first stage.

a) Marker Recognition. In this step, a user walks in the environment guided by another person. Firstly, the guide instructs the user to reach the augmented reality markers, as they are anchors for absolute positions. The guide follows the user closely to avoid that the user bumps accidentally into physical walls. After tracking the augmented reality markers, the user walks around the environment to track static markers. The system needs to track at least three markers simultaneously (any combination of static and smart markers) in order to estimate the user position in the environment through triangulation.

When using infrared light in an optical tracking system, the markers can be segmented by thresholding. While the markers reflect the infrared light, other wavelengths are filtered by the IR-pass filter mounted in front of the camera lenses. Thus, regions in the image representing a marker have significantly higher brightness values compared to the background [46]. In this regard, the system used a thresholding technique [45], [46] to simplify recognition and segmentation of infrared markers from the background. This process locates and extracts objects of interest in images, discarding redundant information.

b) Discrimination Between Static Markers and Smart Markers. After this segmentation and mask creation process, this step determines the position of markers using the OpenCV library [www.opencv.org]. This stage uses two functions (Blob Detection and SURF) to identify a marker as a “static marker” or “smart marker”.

The system uses the Blob Detection function to determine the position and size of a marker, similar to the approach applied by Mehling [46]. The system uses the Blob Detection function to fit tracker contour points into an ellipse shape, where the ellipse parameters describe the blob geometry.

When the number of calculated ellipses is greater than three, the system classifies the marker. After the system

identifies the marker, it starts feature detection and recognition using SURF (Speeded-Up Robust Features) [47]. SURF is a detector and a high-performance descriptor for points of interest in images where the image is transformed into coordinates, using a technique called multi-resolution. An important feature of SURF is the capability of finding corresponding points between two images or objects that are in the same scene.

Our system uses SURF to detect the pattern that a “smart marker” represents in its LED matrix. The pattern determines the purpose of this marker in the system: it could be used for a virtual superimposing layer, to denote an absolute position, or both. After determining the purpose of the marker, the system is able to calculate marker position relative to the camera and the physical environment using triangulation.

c) The 3D Mapping. This step creates a 3D marker map using the tracked marker data. The system calculates the position of markers in this map using angulation technique [48] for triangulation. Angulation techniques use angles to calculate the position of an object in an environment. More specifically, two dimensional angulation techniques require two angle measurements and one length measurement (such as the distance between two reference points) to calculate the position of an object. Fig. 9 illustrates an example of 2D angulation.

After the system determines the 3D position of a marker, it stores this information to create a 3D marker map. In a later stage, this map will be overlaid into the virtual map that represents the physical environment.

2) Second Stage

In the stage, the system uses triangulation to calculate depth information (i.e., how distant the user is from a physical wall). As the user does not see the physical world, the system should be able to calculate the distance between the user and physical walls with high accuracy; otherwise, the user may accidentally bump into physical walls. For our experiment, we tested markers with physical distances going from 1.5 through 3 meters between the obstacles (walls) and the HMD device.

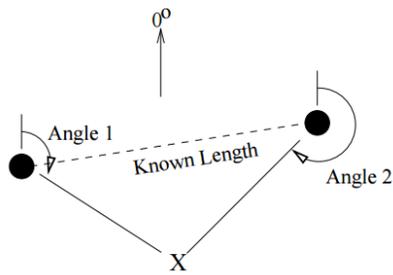


Fig. 9. 2D angulation illustrates locating object 'X' using angles relative to a 0° reference vector and the distance between two reference points [48].

3) Results

The computer vision algorithm used in marker detection was able to track infrared markers with few accuracy errors, which enabled the next step (data acquisition) to finish processing fast. For example, in this experiment we noticed that errors in the x axis increased when the distance between the user and a wall increased: 10 cm error (distance: 1.5 m), 33 cm error (distance: 2.5 m), and 45 cm error (distance: 3m). However, we noticed accumulation errors while the test was running, due to the sum of accuracy errors. Errors due to the camera calibration process also contributed to the overall accumulation error.

Fig. 10 and Fig. 11 illustrate examples of calibration errors. Physical walls are represented by solid lines and virtual walls are represented by dashed lines. Fig. 10 represents a situation where the virtual wall is in front of the physical wall. Fig. 11 represents a more severe case where the user may bump into a physical wall without being aware of it, as the virtual wall is behind the physical wall.

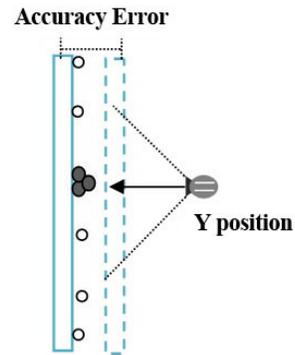


Fig. 10 Calibration error: the virtual wall is in front of the physical wall

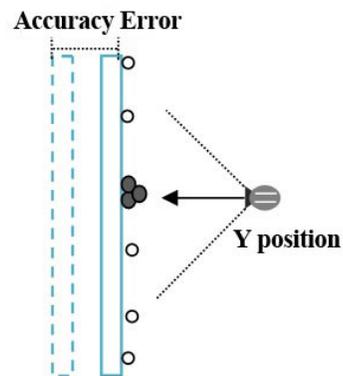


Fig. 11. Calibration error: the virtual wall is behind the physical wall



Fig. 12 User inside the room (left) and his view point (right)

After the camera calibration, we noticed significant reduction of distortion errors, although the accumulation errors still persisted. In future tests, we plan to use the Extended Kalman Filter to reduce these kinds of errors (related to triangulation), which has been an alternative proposed by Maeda and co-authors [49] to solve this kind of problem.

B. Game Demo Experiment

Our game experiment is based on the Tuscany Demo [50]. In this test, the user carries a smart marker that simulates a gun. Fig. 12 illustrates a user inside the room and his corresponding point of view.

The user wanders around the environment to destroy the alien invaders by shooting them (with the controller) and to keep the aliens from invading the house. The user is able to move freely in the environment to accomplish these goals. The field of view is solely determined by the position and orientation of the user's head. Twelve users (aged from 18 to 25) took part in this test, with two of them being women. This test has two main stages: room setup and game playing.

1) Room Setup and Calibration

In the first stage, we needed to setup the markers in the physical room according to the virtual scene already designed in the Tuscany Demo. The first player to participate in the test was responsible for conducting room setup and calibration, which consisted in loading all marker data from the database and tracking all corresponding physical markers in the 3D map.

In this test we distributed 16 static markers in a closed room (approximately 50cm apart from each other) and 4 smart markers positioned at known locations. The markers in the room delimit the virtual scene and the system uses these markers to calculate user position while he is moving in the room. Fig. 13 illustrates the placement of smart markers (A, B, C, and D) and static markers (dots) in the room. Our plan was to position smart markers in the physical room to match their corresponding representation in the virtual room (Fig. 14).

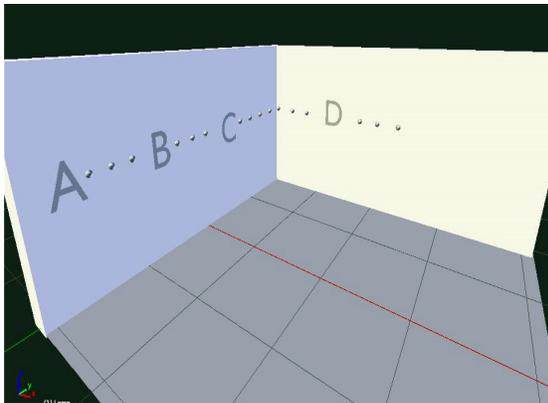


Fig. 13 Static markers (isolated dots) and smart markers (A, B, C, and D letter patterns).

After the room setup, the calibration process overlays the 3D marker map into the virtual world and verifies marker positions and the distances between the user and the markers.

In this test, we used a virtual environment (from a third-party demo) that was not modeled specifically for the physical room that we used. In this regard, we had to conduct multiple rounds of marker recognition stages (as in Section V.A.1)) to ensure that the overlay of virtual and physical worlds was correct, since we intended to minimize calibration errors and to avoid situations such as the one described in b. Currently, our system requires a pre-processing step where it tracks markers and creates a 3D marker map to use in a mixed-reality overlay. In future versions, we plan to create this 3D marker map and the mixed-reality overlay in real-time, eliminating the pre-processing stage.



Fig. 14 Virtual room used in the test.

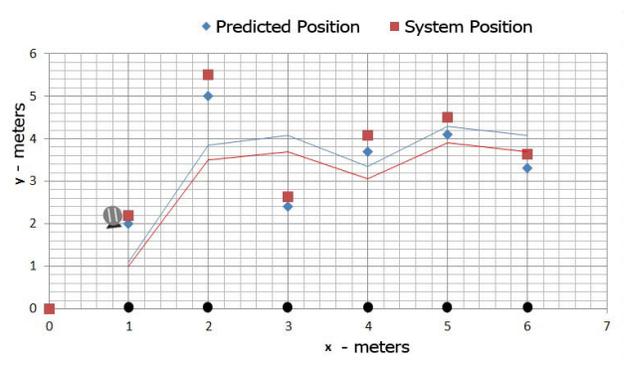


Fig. 15 Accuracy errors.

Fig. 15 presents a chart that illustrates the predicted user position (x and y coordinates) relative to the markers (blue) and the user position calculated by the system through triangulation (red). Triangulation errors increased as the distance from the camera to a marker increased, which results in slight inaccuracy.

The system presented some accuracy errors due to camera calibrations and other factors, but when considering the low-cost hardware employed, the test results exceeded our initial expectations. After the calibration step, the game administrator loaded the virtual scenario, overlaying it with the physical world, thus creating the mixed-reality that the specific game uses.

1) Game Playing

Each test session lasted for about 15 minutes (excluding the time required for room setup and calibration). However, some players needed to get used to the system before participating in the test due to their inexperience in using the test equipment.

Nine players (out of 12) were able to complete the established game mission, which consisted in shooting aliens enemies and protect the house entrance. Two players said that were not able to complete the game due the gameplay difficult rather than our immersion system. One player felt dizziness while using the Oculus Rift device and did not finish the game.

According to the players, the game stage was consistent (i.e., moving in the virtual world matched with moving in the physical environment), which enabled freedom for acting. However, the players complained that carrying the bag (with the notebook) was a bit uncomfortable. Many players suggested including other virtual characters played by real players.

VI. CONCLUSIONS.

There is a growing trend in the industry to bring VR hardware with affordable prices to the mass market. Affordable, lightweight, mobile, and powerful VR hardware facilitate new game possibilities. In this paper we defined a new genre of digital game – which we call live-action virtual reality game – that becomes feasible due to the advances in VR hardware. In a nutshell, live-action virtual reality games combine live-action, mixed-reality, context-awareness, and different interaction paradigms comprising tangible objects, embodied interactions, context-aware input devices and wearable devices. This paper discusses the main concepts of this new game genre, which is fundamentally different from other related game genres such as virtual reality games, augmented reality games, pervasive games, and alternate reality games (Section III.A).

In this paper we propose seven conceptual design characteristics for live-action virtual reality games. We believe that these characteristics help game designers develop more immersive and robust games. For example, one of the recommendations that arises from the characteristic C6 is the practice of implementing “uncertainty handling policies” to remove (or minimize) problems caused by limitations of the involved technologies, such as networking and sensors. Another example is that, among the recommendations in C5, there is the important warning about designing implicit interactions that avoid accidents with players. Other examples can be found in Section II.

In this paper, we also describe an indoor navigation system for live-action virtual reality games that combines a custom stereo-camera and infrared sensors to locate and orientate users in a mixed-reality environment. Our architecture enable the players to move freely in a mixed-reality scenario with small accuracy errors, considering the limitations imposed by physical elements. The user tracking processes enable the game to estimate user position correctly and enable interaction with

markers. Our solution of using multiple kinds of markers showed to be important for the whole proposed architecture.

As part of future works, the authors intend to explore the potential of this navigation system, improve the use of sensors that were not presented in this architecture and develop multi-user platform applying the markers in the body of the player as a motion capture system. Another line of future works is the development of a more complete game using all recommendations that are intrinsic to the proposed conceptual design characteristics of the live-action virtual reality games.

ACKNOWLEDGEMENTS

The authors thank CAPES, CNPq, FAPERJ, NVIDIA, ITFM and FINEP for the financial support to this work.

REFERENCES

- [1] The VOID, “The VOID,” 2015. [Online]. Available: <https://thevoid.com/>. [Accessed: 09-Jul-2015].
- [2] Artanim Interactive, “Real Virtuality,” 2015. .
- [3] L. Valente, E. Clua, A. Ribeiro Silva, and B. Feijó, “Live-action virtual reality games,” Departamento de Informática, PUC-Rio, Rio de Janeiro, MCC03/15, Jul. 2015.
- [4] M. V. Sanchez-Vives and M. Slater, “From presence to consciousness through virtual reality,” *Nat. Rev. Neurosci.*, vol. 6, no. 4, pp. 332–339, Apr. 2005.
- [5] A. R. Silva, E. Clua, L. Valente, and B. Feijó, “An indoor navigation system for live-action virtual reality games,” in *Proceedings of SBGames 2015*, Teresina, 2015, pp. 84–93.
- [6] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, “Augmented reality: A class of displays on the reality-virtuality continuum,” *Syst. Res.*, vol. 2351, no. Telem manipulator and Telepresence Technologies, pp. 282–292, 1994.
- [7] A. K. Dey, “Understanding and Using Context,” *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, Jan. 2001.
- [8] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Laerhoven, and W. Velde, “Advanced Interaction in Context,” in *Handheld and Ubiquitous Computing*, vol. 1707, H.-W. Gellersen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 89–101.
- [9] M. Endler, A. Skyrme, D. Schuster, and T. Springer, “Defining Situated Social Context for Pervasive Social Computing,” in *Proc. of the 2nd Workshop on Pervasive Collaboration and Social Networking (PerCol 2011)*, Seattle, 2011.
- [10] 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.
- [11] Volkswagen, “The World’s Deepest Bin | The Fun Theory,” 2009. [Online]. Available: <http://www.thefuntheory.com/worlds-deepest-bin>. [Accessed: 23-Jul-2015].
- [12] H. Ishii and B. Ullmer, “Tangible bits: towards seamless interfaces between people, bits and atoms,” *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, pp. 234–241, 1997.
- [13] C. Magerkurth, A. D. Cheok, R. L. Mandryk, and T. Nilsen, “Pervasive games: bringing computer entertainment back to the real world,” *Comput. Entertain. CIE*, vol. 3, pp. 4–4, Jul. 2005.
- [14] I. Lindt, J. Ohlenburg, U. Pankoke-Babatz, S. Ghellal, L. Oppermann, and M. Adams, “Designing Cross Media Games,” in *Proceedings of 2nd international workshop on pervasive gaming applications*, Munich, 2005, pp. 8–13.
- [15] V. Tuulos, J. Scheible, and H. Nyholm, “Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup,” in *Pervasive Computing*, vol. 4480, A. LaMarca, M. Langheinrich, and K. Truong, Eds. Springer Berlin / Heidelberg, 2007, pp. 37–54.

- [16] A. Waern, M. Montola, and J. Stenros, "The three-sixty illusion: designing for immersion in pervasive games," *Proc. 27th Int. Conf. Hum. Factors Comput. Syst.*, pp. 1549–1558, 2009.
- [17] Razer, "Razer Hydra," 2015. [Online]. Available: <http://www.razerzone.com/gaming-controllers/razer-hydra-portal-2-bundle>. [Accessed: 21-Jul-2015].
- [18] Z. Zhang, "Microsoft Kinect Sensor and Its Effect," *IEEE Multimed.*, vol. 19, no. 2, pp. 4–10, Feb. 2012.
- [19] L. Valente and B. Feijó, "Extending Use Cases to Support Activity Design in Pervasive Mobile Games," in *Proceedings of the 2014 Brazilian Symposium on Computer Games and Digital Entertainment*, Porto Alegre, Brazil, 2014, pp. 193–201.
- [20] A. Schmidt, "Implicit human computer interaction through context," *Pers. Technol.*, vol. 4, no. 2–3, pp. 191–199, Jun. 2000.
- [21] S. Benford, A. Crabtree, M. Flintham, A. Drozd, R. Anastasi, M. Paxton, N. Tandavanitj, M. Adams, and J. Row-Farr, "Can you see me now?," *ACM Trans. Comput.-Hum. Interact. TOCHI*, vol. 13, pp. 100–133, Mar. 2006.
- [22] Wikipedia, "Virtuality (gaming)," *Wikipedia*. 07-Jun-2015.
- [23] Gaming history, "Dactyl Nightmare (1991)," 2014. [Online]. Available: <http://www.arcade-history.com/?n=dactyl-nightmare&page=detail&id=12493>. [Accessed: 07-May-2015].
- [24] Gaming history, "Grid busters (1991)," 2013. [Online]. Available: <http://www.arcade-history.com/?n=grid-busters&page=detail&id=12498>. [Accessed: 07-May-2015].
- [25] Wikipedia, "Virtual Boy," *Wikipedia, the free encyclopedia*. 24-Jun-2015.
- [26] G. King and T. Krzywinska, *Tomb Raiders and Space Invaders: Videogame Forms and Contexts*. London ; New York : New York: I. B. Tauris, 2006.
- [27] R. Azuma, "A survey of augmented reality," *Presence*, vol. 6, no. 4, pp. 355–385, 1997.
- [28] W. Piekarski and B. Thomas, "ARQuake: The Outdoor Augmented Reality Gaming System," *Commun ACM*, vol. 45, no. 1, pp. 36–38, Jan. 2002.
- [29] E. Nieuworp, "The pervasive discourse: an analysis," *Comput. Entertain. CIE*, vol. 5, no. 2, p. 13, Apr. 2007.
- [30] "Live action role-playing game," *Wikipedia, the free encyclopedia*. 24-Nov-2015.
- [31] R. Ballagas and S. P. Walz, "REXplorer: Using Player-Centered Iterative Design Techniques for Pervasive Game Development," in *Pervasive Gaming Applications – A Reader for Pervasive Gaming Research*, vol. 2, C. Magerkurth and C. Roecker, Eds. Aachen: Shaker Verlag, 2007, pp. 255–284.
- [32] M. Sra and C. Schmandt, "Expanding social mobile games beyond the device screen," *Pers. Ubiquitous Comput.*, vol. 19, no. 3–4, pp. 495–508, May 2015.
- [33] D. Szulborski, "The Beast," in *Space Time Play*, F. Borries, S. P. Walz, and M. Böttger, Eds. Birkhäuser Basel, 2007, pp. 228–229.
- [34] S. Spielberg, *A.I. Artificial Intelligence*. Warner Bros., 2001.
- [35] J. McGonigal, "This might be a game: ubiquitous play and performance at the turn of the twenty-first century," PhD Thesis, University of California, Berkeley, Berkeley, CA, USA, 2006.
- [36] M. Montola, J. Stenros, and A. Wærn, *Pervasive Games: Theory and Design*. Morgan Kaufmann, 2009.
- [37] D. Szulborski, *This Is Not A Game: A Guide to Alternate Reality Gaming*, 1st ed. Lulu.com, 2005.
- [38] W. Piekarski, B. Avery, B. H. Thomas, and P. Malbezin, "Hybrid Indoor and Outdoor Tracking for Mobile 3D Mixed Reality," in *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, Washington, DC, USA, 2003, p. 266–.
- [39] J. Kim and H. Jun, "Vision-based location positioning using augmented reality for indoor navigation," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, pp. 954–962, Aug. 2008.
- [40] S. Siltanen, "Theory and applications of marker-based augmented reality," VTT Technical Research Center of Finland, 2012.
- [41] E. Gobbetti and R. Scateni, "Virtual reality: past, present, and future," Nov. 1998.
- [42] The VOID, "Gaming Pods," 2015. [Online]. Available: <https://thevoid.com/#the-centers/2>. [Accessed: 09-Jul-2015].
- [43] Kenzan, "Real Virtuality," 2015. [Online]. Available: http://www.kenzan.ch/?avada_portfolio=real-virtuality. [Accessed: 09-Jul-2015].
- [44] OpenCV, "Camera calibration With OpenCV — OpenCV 2.4.11.0 documentation," 2015. [Online]. Available: http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html. [Accessed: 26-Jul-2015].
- [45] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imaging*, vol. 13, no. 1, pp. 146–168, 2004.
- [46] M. Mehling, "Implementation of a Low Cost Marker Based Infrared Optical Tracking System," MSc dissertation, Hochschule der Medien, Fachhochschule Stuttgart, 2006.
- [47] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [48] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [49] M. Maeda, T. Ogawa, K. Kiyokawa, and H. Takemura, "Tracking of user position and orientation by stereo measurement of infrared markers and orientation sensing," in *Eighth International Symposium on Wearable Computers, 2004. ISWC 2004*, 2004, vol. 1, pp. 77–84.
- [50] Oculus VR, "Oculus Tuscany Demo," 2013. [Online]. Available: <https://share.oculus.com/app/oculus-tuscany-demo>. [Accessed: 11-Sep-2015].