Hindawi
International Journal of Computer Games Technology
Volume 2019, Article ID 2705734, 9 pages
https://doi.org/10.1155/2019/2705734

Research Article

IMOVE: A Motion Tracking and Projection Framework for Social Interaction Applications

Marie Kegeleers, Raoul Bruens, Maxim Liefaard, Nestor Z. Salamon, and Rafael Bidarra

Computer Graphics and Visualization Group, Delft University of Technology, Delft, Netherlands

Correspondence should be addressed to Rafael Bidarra; r.bidarra@tudelft.nl

Received 25 October 2018; Revised 22 December 2018; Accepted 27 January 2019; Published 3 March 2019

Academic Editor: Michael J. Katchabaw

In public places such as malls, train stations, and airports, there is a constant flow of people either waiting or commuting. Even though people at these locations are surrounded by many other individuals, mostly there is little social interaction, which generally creates a gloomy atmosphere. Any applications promoting social interactions are a welcome addition. We present IMOVE, an interactive framework aimed at facilitating the development of such applications. It offers a combination of motion tracking and projection methods which makes it easier to create interactive experiences and games, tailored to motivate people to move around, explore, and, most importantly, interact with each other in a fun way. People moving around trigger events and effects, interacting with the applications using their body movements or even collaboratively working towards an outcome. IMOVE was validated by means of a variety of applications in a real scenario, the entrance hall of a busy public building: the classic *Pong* game, a collaborative and accessible casual game (*Save the Turtles!*), and a procedural visual art generator based on game mechanics (*Light Trails*). All applications have been successfully running for the past year. The IMOVE framework is freely available online and it has been shown to be particularly suited and accessible to novice game and interactive application developers for large public spaces.

1. Introduction

In our modern and busy world, there are a lot of public places with large open spaces and a constant flow of people. Some examples are train stations, airports, and waiting lines at theme parks. Commuting or waiting, people at these locations are surrounded by many other individuals. The lack of a conversation-trigger or the void created when these places are partially empty usually discourages any kind of social interaction. Activities that promote social interaction in such circumstances, even if only casually, can make a tremendous difference (See Figure 1).

IMOVE is a motion tracking and projection framework which aims to facilitate the development of interactive systems in public locations. Applications developed using IMOVE are meant to encourage people to explore a scene or participate in a game and, therefore, trigger social interaction. The IMOVE setup consists of a normal camera and a projector attached to the computer, both hanging close to the ceiling. The camera and projector face down on an empty, open space where people can pass through. The camera records the environment to analyze people's movements, which controls the developed application. The projector will display the visual output onto the surrounding floor.

The IMOVE framework consists of two interchangeable modules. The People Detection module is first calibrated on the installation environment. The acquired images are used to extract participant's information parsed as input for the application, hereafter called Scene module. The Scene module handles the application logic and outputs the to-be-projected graphics. On the implemented game *Pong*, for example, the movement of two participants is acquired by the People Detection module. The Scene module uses such information to control the paddles and bounce the balls. New applications (either games or interactive scenes) can be developed by changing the Scene module.

To validate the flexibility of IMOVE, we also developed and tested *Save the Turtles!*, a cooperative game where participants work together to guide the turtles to the ocean, and *Light Trails*, a procedural art generation in which the

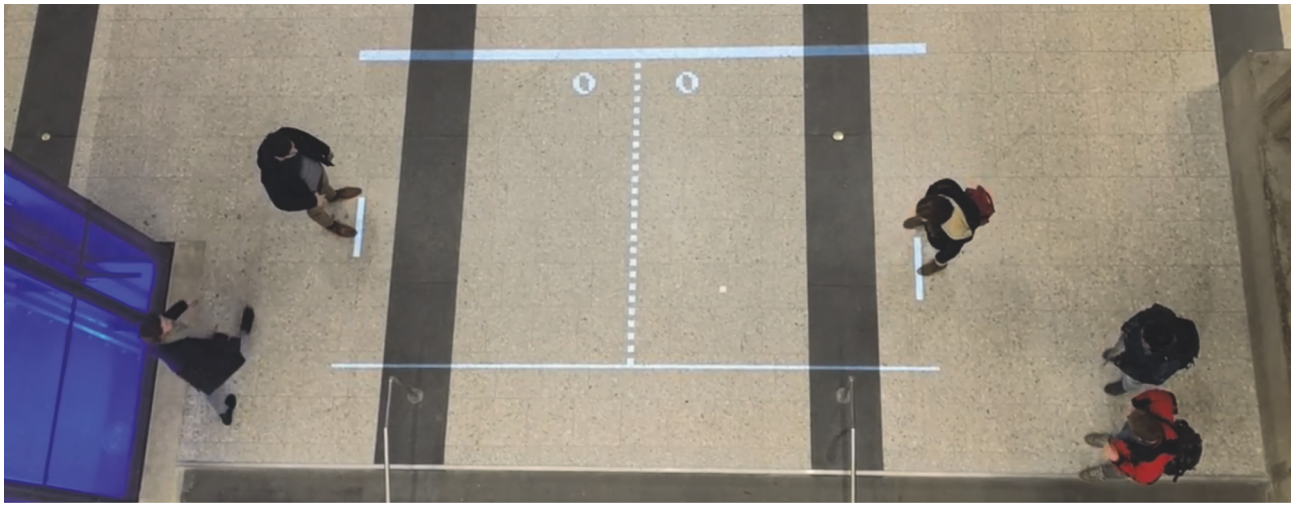

Figure 1: A group of people interacting while playing *Pong* in a public space.

participants' positions yield a colorful field of light trails, which interact with those of other nearby participants.

Nonetheless, while a comprehensive framework facilitates the construction of such public installations, the applications must consider ways to attract and engage participants. For the applications presented in this paper, we followed game design principles to keep interaction fun and engaging. Such principles were effective and can further serve as guidelines when developing new applications.

Specifically, our work makes the following contributions:

(i) a generalized tracking method paired with a calibration module to enable deployment in a variety of locations with heterogeneous light conditions (Section 3);

(ii) a low-latency tracking-projection pipeline to facilitate the creation of event-based scenes (Section 3);

(iii) two novel deployed installations specifically designed for social interaction (Section 4).

2. Background and Related Work

Public interactive installations have already been deployed in different situations. The usage and goals range from simple exhibitions to interactive scenes. One classic example was Loren Carpenter's experiment at SIGGRAPH 1991 Electronic Theater [1]. Carpenter showed how the crowd could organize itself to interactively participate into a Pong game. The study also demonstrated how swarm behavior occurs in crowds, which is commonly observed in different public installations [2, 3]. Interactive installations are now becoming increasingly widespread, employing various types of interactions.

Several approaches track individual people and react according to movements or gestures. ACCESS [4] tracks anonymous individuals in public places by pursuing them with a robotic spotlight and acoustic beam system. The spotlight is moved by a human watching a live camera feed using a joystick. The reception was mixed: while some individuals may not like being monitored, others may love the attention. The Gymnastics Motion Art installation [5] shows a series of animations based on the motion of athletes. Even though it is not real time and created by human hands, it is derived from the original footage to create motion art from movement. People tracking was also part of a multimodal art installation where galaxies are projected and moved by participants; projection colors changed based on the participant's voices [6]. Analyzing the installation outcome, authors observed a convergence on user interaction, both imitating other participant's behavior or collaboratively joining efforts to find out different outcomes.

Aiming at collaborative participation, huge installations with tailored equipment have previously been deployed. The Red Nose Game [7] is a collaborative game played on *the BBC Big Screens*. The game begins with small red blobs, like clown noses, placed randomly across the screen. The camera is put above the space in front of the Big Screen where the players are located. The goal of the game is to push the virtual blobs together into a single big blob. Participant movement is determined from the live feed of the camera. GRiD [8] is a collaborative Pong game using LiDAR to track peoples' movements in a public space. The solution offers an immersive experience, but requires specialized sensors and specific deployment locations.

Location and displacement of an installation can influence crowd interaction [9, 10]. To develop a location independent system and avoid specific equipment, an adaptable and responsive framework is desirable. To that extent, Godbehere et al. [11] developed a framework to track people under variable lighting conditions, using the detection to trigger the installation. No collaborative interaction is taken into account in their work.

Nonetheless, an interactive system involving multiple participants must attract people and focus on aspects to keep them engaged [12]. Several techniques to encourage participation and its effects have already been studied [13–16]. For public and interactive installations, we consider the design principles presented by Maynes-Aminzade et al. [17]:

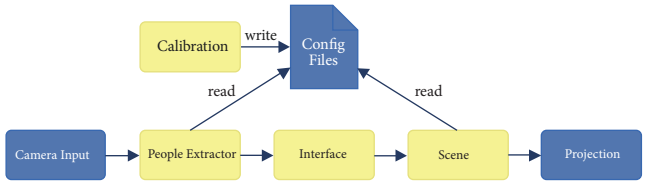

Figure 2: The high-level architecture of the framework.

Focus Design on the Activity, Not on Technology. While interactive systems like this are not common and people are easily impressed by the technology, it is important to shift their focus to the activity to ensure participation.

Make the Control Mechanisms Obvious. Intuitively, mechanics must be easy to understand and execute. In our system, people can arrive and start participating at any time during the experience. Thus, it is not possible to show everyone a tutorial to explain the game mechanics. Therefore, they need to be easily derivable from the combination of movement and projection. However, making the control mechanisms obvious also means showing people that they actually are in control. This can be done by optimizing the controls to make projections accurately follow movements.

Ramp up the Difficulty of the Activity. As previously mentioned, a tutorial is not possible to explain mechanics. It is better to begin with simple interactions and make the crowd discover more advance interactions and additional effects.

Vary the Pacing of the Activity. Interactions with high intensity work best when combined with periods of relaxation.

In the next section, we describe how IMOVE was built to facilitate the creation of interactive installations with engaging content.

3. The Framework

The IMOVE framework consists of two major parts: the IMOVE core and the calibration application. The IMOVE core contains the People Detection module and the Scene module. Initially, calibration is executed to determine the correct mappings between the real and the virtual environment. The People Detection module, after reading the calibration and the input frames from the camera, detects people in the scene, pushing such information into a queue. Finally, the Scene module pops the detected people from the queue and deploys that information to the applications. An overview of the different parts and the interaction between them can be viewed in Figure 2.

We designed the framework to have the People Detection and Scene modules both running in their own process. We used the *OpenCV* [18] library for all image-related operations, which can also run operations on separate threads. Such architecture allows us to read, process, and output at a frame rate by which the participants notice no stuttering. However, this adds a communication challenge. For fast communication we use a shared memory between the modules as interprocess communication with Boost library [19].

3.1. Calibration. The deployment of the framework will differ for every installation. The type of camera and projector, the height of the camera and the placement of the projection, to name a few, can all influence properties like real size per pixel, and alignment of the projection and camera. To make the system as flexible as possible, the calibration module finds the parameters to properly link the framework to the environment.

As a base metric, the coordinates of extracted people on the camera frame need to be mapped to scene (virtual) coordinates. The calibration is done by indicating points that represent (i) the four corners of the projection field and (ii) a meter-to-pixel mapping. Using an input frame from the installed camera, we can define such points by clicking in the four corners of the field and in two places that are physically one meter apart, respectively. The framework uses this to adapt to camera-projector distance and their height and angle towards the floor. Before every new location setup, the calibration runs and writes the results to the corresponding configuration file, which will later be used by the People Detection module and Scene module. Some examples of values written in a configuration file are the coordinates of each corner of the projection in the camera frame and the number of pixels that represent one meter in reality.

3.2. People Detection. One of the biggest challenges on using people's movement for controlling applications is to extract people's positions from each frame using a regular camera and under different light constraints [20]. We aimed at a framework able to run indoors at all times of the day, either with natural or artificial lighting. It should ignore disturbances like changes in the background or bright spots on the floor when the sun shines through the windows. Additionally, the desired flexibility in camera height does not allow the usage of low resolution images. To cope with these challenges, we detect and identify the players in two steps:

3.2.1. Detection. We start by separating static and moving content from the input stream, which should represent the playing field and potential players, respectively. For such task, we use the K-nearest neighbor background subtraction of Zivkovic and van der Heijden [21] as available in standard computer vision libraries (e.g., [18]). The method implements a pixel-level segmentation which regularly updates the background model to avoid external disturbances, such as sudden spots of sunlight being detected as a moving region. We did not focus on improving detection accuracy and known artifacts such as region merging and occlusion, as well as static content fading into the background, are still a problem. Fortunately, occlusions are avoided by the top view perspective and fading content can be minimized, in practice, during the identification step. In terms of performance, we observed up to six regions being detected without any measurable delay, which showed to be more than enough

for any applications with a reasonable occupancy of the projection field. Additionally, to deal with different deployment locations, only detected regions are detected whose size is smaller than the threshold from the configuration file.

3.2.2. Identification. Next, the identification step matches people already present in the scene to the last retrieved regions. We based the matching criteria on the reasonable assumption that, running at a high frame rate, the person's movement will be small enough to become the closest region in the next frame. A match occurs when there is a new region which is the closest point to a person. For each existing person with no match, we check the distance to the playing field borders. If the person was close to a border, we assume the person left the scene and his/her position is removed. If the person was in any other part of the field, a no-match can indicate the person is standing still inside the playing field. For this case, we keep the person's position for a short period of time (i.e., 5 seconds) or until it moves and is matched again. Such delay is necessary to avoid discarding persons that might have faded into the background during the detection step but are actually still playing. We then look for new regions with no match, which are treated as follows. When close to the edge, it indicates a new person who has entered the field and the region will be included as a new player. New regions in any other place are discarded, as people cannot appear in the middle of the field. Finally, the updated persons' positions are forwarded to the Scene module.

3.2.3. Detection Optimization and Adaptability. Depending on the deployed application, different optimizations can be added to the detection step. For games that have control mechanics outside of the projection, for example, detection zones can be created outside of the projection area. Only people walking in these zones will be detected. This also prevents others from disturbing the game if they walk through the field or behind the players. Additionally, given the modular and flexible design of the framework, the detection step can easily be adapted to use different hardware (e.g., infrared or heat camera) that best suits the application and location.

For more details on the IMOVE implementation, the reader is referred to the documentation and sources, available on the project repository (see Section 5).

3.3. Scenes as Applications. We call *Applications* the games, scenes or activities participants can interact with, developed using IMOVE and designed to entertain and engage players. These applications integrate the aforementioned Scene module.

Event-Based Design. New applications can be built from scratch but, to speed up the design process, a convenient event-based framework is provided. Each event statement is defined as a pair *conditions-actions*. Examples of conditions can be 'two objects collided' or 'a player changed position'; examples of actions can be 'change an object's color' or 'modify the state of some entity (e.g., the game score)'. Per pair, when the defined conditions are satisfied, the respective actions have to be executed. Describing every statement of the design or mechanics of a scene using such *conditions-actions* statements provides a very intuitive and declarative method of implementing desired game mechanics. New conditions and actions are easily created, edited or combined, providing an efficient link between the People Detection module and designed mechanics.

Figure 3: IMOVE setup and installation location.

Graphics. For developing the feedback projected onto the floor by the applications, IMOVE uses the free graphics library SFML [22]. The created conditions and actions can be used to trigger visual events, like, for example, the displacement or replacement of a virtual object.

4. Applications

IMOVE projects the scene onto the floor which catches peoples' attention and attracts them to the applications. However, to convince them to participate, the scenes or games must be interesting and engaging. For validation of the framework, we implemented three applications: *Pong*, *Save the Turtles!*, and *Light Trails*.

For all the applications presented, we used a Logitech HD Pro C920 webcam and Panasonic DLP PT-RZ670B projector, suspended 7 meters from the ground, and running on a consumer level desktop at 30 frames per second. Figure 3 shows the setup deployed in the hall of a public location.

4.1. Pong. *Pong* is a classic game that does not require any explanation or tutorial. Participants need only to move sideways to control the paddle, allowing everyone to participate.

4.1.1. Game Design. *Pong* is one of the first computer games ever created. It resembles tennis, is played by two players who control a paddle on either side of the field, and has a ball that bounces between them. The goal is to achieve a higher score by forcing the opponent to miss the ball. The IMOVE *Pong* scene is very similar to the original game. It consists of

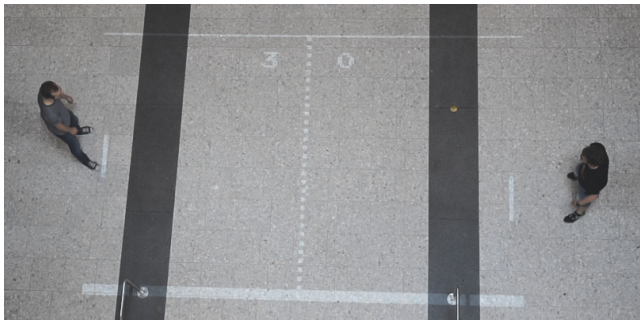

Figure 4: Players interacting with the *Pong* game (the two dark stripes are, unfortunately, part of the floor decoration).

Figure 5: Players interacting with the *Save the Turtles!* game.

a black field, two white paddles and a white ball. The score is displayed at the top. A scene-shot is shown in Figure 4.

4.1.2. Mechanics. To make the game more suitable for the IMOVE framework as an interactive movement game, a few small game mechanics were added or changed. Borders on the field were added to clarify where the ball would bounce back and how far players were able to move the paddle. Secondly, to prevent the game from always running, an automatic start and stop mechanism was added. When there is a player missing on either side of the field (there is no one being detected on that side of the field), no ball will be launched. As soon as the system detects players on both sides, the score resets and a ball is launched to start the game.

4.1.3. Interaction and Engagement. The level of interaction and engagement of this game can be established by verifying the requirements defined in Section 2. Our goal for *Pong* was to have a game in which two people can interact with each other. Initially, the dark screen projected on a light-colored floor attracts a crowd in order to start the game. Next, the participants need to be entertained and motivated to keep playing and interacting. The design should encourage participants to explore and interact with each other and with the system. The *Pong* design follows the principles to optimize audience participation in interactive systems. The game starts as soon as the system detects two players to immediately engage them, and does not draw attention to any technical aspects of the system. The controls of the scene are as simple as the classic game but instead of pressing a button to move the paddle up or down, the participants move themselves. This becomes clear as the paddles always follow anyone who passes by. To ramp up the difficulty, the speed of the ball increases as long as no point is scored. Variety in pacing already happens naturally in *Pong*. Depending on the angle of incidence of the ball on the paddle, the angle with which the ball is sent away from the paddle varies. In practice, it sometimes takes a long time for the ball to travel across the field while bouncing from side to side while other times it quickly flies straight to the other side.

4.2. Save the Turtles! The second application, *Save the Turtles!*, is an awareness game that requires players to collaborate towards a common goal: to guide newborn turtles to a safe marine area [23]. Players walk towards the projected turtles to push them into the correct direction. Multiple players may be needed to ensure all turtles safely reach the sea without colliding with obstacles. A scene-shot is shown in Figure 5.

4.2.1. Game Design. The game starts with a nest of eggs at the center of the playing field from which turtles spawn. One or more players can push the turtles to make them move in specific directions. The goal of the game is to push as many turtles as possible to the sea, at each of the four corners of the projected field. To increase the difficulty, a number of sea urchins are spread around the area, behaving as obstacles: turtles vanish when touching them. To avoid linearity of the gameplay, the turtles wander around randomly when they are not being pushed by players. Collaboration between players is also encouraged, as it can be needed to make sure that no turtle walks, or is accidentally pushed, into the urchins.

4.2.2. Mechanics. Any person who walks through the projection field can interact with the turtles. By walking towards a turtle, it is pushed towards the walking direction. When a turtle is pushed into the sea (goal achieved), it is removed from the game and the score increases. On the other hand, when a turtle collides with an obstacle, it is removed from the game without increasing the score. The obstacle that was hit disappears and a new obstacle spawns at a different location. Such reallocation ensures that players continuously have to adapt their strategy to the dynamic playing field. Furthermore, edges of the playing field behave as walls. Players can walk around the turtle to push them away from the edge of the projection, stopping turtles from becoming stuck at the edges. Nonetheless, players can enter and leave the game whenever they want. In case all participants leave the area, the turtles slowly return to the nest and the game resets. This reset mechanic prevents any harsh restarting, making the game loop more natural.

4.2.3. Interaction and Engagement. We also verify the design principles earlier defined in Section 2. By extensively user-testing the interaction of the game, we focused explicitly on technical changes to improve interaction, thereafter making the game engaging and intuitive. The game acts as an interactive environment when no one is actively playing. This serves

Figure 6: Players interacting with the *Light Trails* scene.

for people to get familiar with the mechanics before playing the game, effectively ramping up the difficulty when they finally start the game. A person walking into the projection is immediately tracked and a dust trail is projected beneath their feet. The dust trail and possible turtle interaction serve as a way to grab the attention of potential players and to make the control mechanisms intuitive. The way the turtles move by design delivers a variation in pacing. One can start a new game as the turtles leave the nest in the center. If the game is already in progress, players sometimes need more people to successfully push turtles past an obstacle. Players can encourage passers-by to help them and to join the game. Moreover, interested passers-by can notice that a game is being played and decide to help. Such design choice encourages collaboration between players as they join and leave at any moment, giving continuity to the game without waiting for another turn. Furthermore, it is easier for players to push turtles together than alone. Elaborate strategies can emerge from well-coordinated cooperation. For example, the turtles can be pushed more accurately by surrounding them from different angles by multiple players. Similarly, a player can stand on top of an obstacle to prevent turtles from colliding with it.

4.3. Light Trails. Our third developed application, *Light Trails*, is a scene with procedural art generation, in which participants create expressive light trails using their movements. A light trail consists of glowing dots flying through the field, following the participants and leaving behind a glowing trail. A number of dots of a specific color swarm each person entering the scene. Participants can mix and change colors and discover other mechanics by exploring the experiment. An example of the scene is shown in Figure 6.

4.3.1. Game Design. The setup of the scene is a black background with a light source in each corner. Each light source has a specific color (initially blue, yellow, red and green) and emits the light trails. The light trails orbit the source when it is not assigned to a participant, and goes back to its source when the participant leaves the scene. As soon as a participant enters the scene, a color is assigned to him/her and light trails (of that color) starts following and orbiting around them. Up to four participants can enter the scene at the same time. To keep participants engaged - and exploring the game design guidelines, we gamified the scene by adding a *color hole* mechanic: there is (randomly) one person who is not surrounded by light trails but instead by shrinking concentric circles, indicating a different player role.

4.3.2. Mechanics. When entering the scene, the color of the closest light source is assigned to the participant. Moving around the scene causes the light trails to follow the participant and create patterns. Walking up to one of the corners causes a change of color. If the current light trails are not the same color as the source, the light trails will leave and be replaced by new trails in the color of the source in the corner. When two participants with different colors walk up to each other and stay within a certain radius, their colors will start to mix. After a few seconds, when mixing is complete, a new unique color is created. Only the two involved participants have light trails of such color orbiting around them. Therefore, beside the drawing patterns of each follower light trails, interaction is also encouraged on searching for new colors. Additionally, when someone walks too close to the color hole participant, his/her light trails will vanish into the color hole. When the person with the color hole leaves the scene, all the captured trails are released, making them explode across the scene for a very colorful effect. Last but not least, when no people are detected in the scene, two points at randomly generated locations are orbited by light trails from all colors to keep the scene dynamic and attract passers-by.

4.3.3. Interaction and Engagement. The design principles defined in Section 2 are verified as follows. The scene is always dynamic. Light trails are always moving which draws attention to the scene and away from any technological aspects of the system. The light trails immediately follow anyone who passes through the scene, showing the control mechanisms to the person being followed and anyone who is watching from outside the scene. Because this scene is more focused on enjoying the generated graphics and less on game-related aspects, there is no difficulty mode. Therefore, there is also no need to vary the pacing of the activity.

4.4. Evaluation. We initially used *Pong* to validate the game mechanics and the functionality of the framework. It has now been deployed for several months at the entrance of a conference center, with players having no issues understanding the game mechanics. Before developing the *Light Trails* as a different category of application, we deployed *Save the Turtles!* to perform a user evaluation.

Save the Turtles! is an application that uses cooperative and more immersive mechanics. During the gameplay testing phase it became apparent that it was difficult to communicate to players how the game worked. Because the game was deployed in a public location, it had to be self-explanatory. However, despite the simple game mechanics, player testing interviews revealed that players ran into two problems. The first problem appeared as limited visibility of the projection: it prevented the players from seeing where the seas (the objectives) were located. Despite the partial glass ceiling

in the deployment location, *Pong* still had better visibility because it only uses black and white shapes. Visibility of the game elements in *Save the Turtles!* was then improved by maximizing contrast and strategically choosing colors. The second problem was reported as people not realizing that they were being tracked, or they did not know what happened when walking around. To clarify this important mechanic, additional animations were added for (i) turtles successfully guided into the ocean, (ii) turtles fainting, and (iii) a tutorial arrow pointing towards the objectives. This approach made the game more intuitive without disrupting its flow by means of an extensive tutorial. Finally, adding an animated dust trail as players moved around helped communicating to the players that they were being tracked.

The player testing process showed that game mechanics which may seem obvious when interacting with a conventional computer setup are not necessarily clear to the player in a tracking and projection context. Extensive testing with multiple iterations uncovers issues that can often be solved by developing better visual representations. Furthermore, it is important to keep in mind where the application is going to be deployed, as both the projecting environment and the audience can notoriously change.

Reception of the *Save the Turtles!* game was overly positive. In player testing questionnaires, participants enjoyed the game and its deployment in public installation: 33 players stated that they had a lot of fun playing the game and the remaining 9 players remained neutral. When asked if they would play the game in different locations, 23 out of 25 people answered with positive interest. Additionally, participants reported that they would play it once more at the current location: 28 out of the 42 players were certain that they would play the game again. Two players would not play it again and all other players might play it again.

Finally, the amount of time that people spend on the game varies. At the entrance of the conference center used for testing, the passers-by often tend to quickly move towards their destination. An application should therefore allow players to have both short experiences and longer play sessions. Both *Pong* and *Save the Turtles!* facilitate such heterogeneous engagement using simple and quickly recognizable mechanics, as well as a recurring game loop to start over again. The test showed that the average of time spent on a game was 1 minute and 45 seconds per player. The shortest playtime was 50 seconds and the longest playtime was 5 minutes and 41 seconds.

5. Third-Party Development and Educational Deployment

In addition to the installations presented in this article, the IMOVE framework was also deployed as an auxiliary tool for a Game Development CS undergraduate course [24]. To illustrate the versatility and ease of use of the framework, we briefly describe how new applications can be created, using as reference *Reflect* and *The Tower*, two student projects developed during the Spring semester 2018.

Figure 7: Reflect: three players collaborating to guide the laser beam towards the target.

Figure 8: The Tower: participants collaborating on the level elements to guide the ball to the bottom right target. Overlaid gameplay screenshot on bottom left.

Reflect is a puzzle game in which participants have to move around to cooperatively place mirrors so that a laser beam is led around obstacles to reach a predefined target. *The Tower* is a turn-based puzzle game that inspires social interaction by having players cooperating to lead the falling ball from the top to the base of the tower, using their body to build or avoid elements on the ball's path. Figures 7 and 8 show, respectively, a view of *Reflect* and *The Tower* games.

Developing an application for IMOVE entails designing the game mechanics that will be triggered by an event on the scene, as well as the response that will be projected back to the players. The creation process carried out by the students can be followed by anyone to create a new IMOVE application.

Initially, students developed the game concept following the guidelines recommended in Section 2 in order to attract participants and promote engagement. Subsequently, they built a prototype application with virtual participants, which can be controlled used mouse and/or keyboard input. A mouse or keyboard event should trigger the application mechanics. At the same time, the calibration could be tested at the deployment location. Using the Calibration module, the physical and virtual environment are mapped and the configuration file is provided to initialize IMOVE.

Once running, IMOVE continuously provides a simple data structure representing where people are currently located in the playing field. This data indicates, for example, if a person left, entered or moved within the playing field.

The framework also monitors such data to steer the event-based design. Having any of these events triggered, the corresponding application-defined action is executed. Intuitively, the mouse/keyboard actions previously defined can then be replaced by the IMOVE data and the application becomes controlled by real participants.

We believe that the successful deployment of IMOVE in this undergraduate course, given the quality of the interactive games developed in just 10 weeks, can confirm the framework is powerful, flexible, and also provides an accessible development pipeline. A detailed manual of IMOVE, its source code and example applications, as well as a video trailer, can be found in the project repository (https://github.com/Mari3/IMOVE).

6. Conclusion and Future Work

IMOVE is a flexible and adaptable framework for motion tracking and projection. With a modular structure, it makes it easier to develop games and applications that promote social interaction in public spaces. In addition to the technical framework, the discussed guidelines for interaction design help in creating interactive and engaging new scenes and games for public installations. The system is currently up and running in a busy public space (the Foyer of the Aula conference center, at TU Delft) where passers-by join in and interact every day.

The IMOVE framework has been released as an open source project and also used as an auxiliary tool for game design courses. All together, the features make the IMOVE framework very versatile and suitable for game developers to express their creativity in many new applications, contexts, and physical public spaces.

As future work, we plan to extend the IMOVE framework with new capabilities, including new control tracking based on arm movements, and a flexible configuration for public spaces with a larger range of projection dimensions and angles. Finally, we also plan to investigate how the IMOVE framework could facilitate the creation of casual experiences involving procedural content generation [25] and adaptive gameplay [26].

Data Availability

All source code for the IMOVE project is available online, in the repository indicated in the paper. That is all that is required to support the conclusions in Section 4.4. Raw data can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank Gerbert Van Nieuwaal and Wouter Posdijk for their cooperation during the initial IMOVE project, Tim Huisman, Mariëtte Schönfeld and Thijmen Langendam for the joint development effort on *Save the Turtles!*, J. Timothy Balint for his sharp proofreading comments, and TU Delft Aula for generously providing the installation space.

Supplementary Materials

Video containing footage of all three applications described in the article was submitted. An updated version of that trailer is available at the project repository https://github.com/Mari3/IMOVE. *(Supplementary Materials)*

References

[1] L. Carpenter, "Cinematrix, video imaging method and apparatus for audience participation," https://vimeo.com/78043173, 1993, US Patents 5210604 (1993) and 5365266 (1994).

[2] J. E. Boyd, G. Hushlak, C. J. Jacob, P. Nuytten, and M. Sayles, "SwarmArt: Interactive art from swarm intelligence," in *Proceedings of the ACM Multimedia 2004 - proceedings of the 12th ACM International Conference on Multimedia*, pp. 628–635, USA, October 2004.

[3] D. Bisig and P. Kocher, "Tools and abstractions for swarm based music and art," in *Proceedings of the 38th International Computer Music Conference, ICMC 2012*, pp. 297–300, Slovenia, September 2012.

[4] M. Sester, Access project. ACCESS Project, 2013, http://access-project.net/.

[5] T. Maly, Using motion capture and code, to turn gymnasts into data art. Fast CoDesign, 2018, https://bit.ly/2JgIi7E.

[6] G. Jacucci, A. Spagnolli, A. Chalambalakis et al., "Bodily explorations in space: Social experience of a multimodal art installation," in *Proceedings of the IFIP Conference on Human-Computer Interaction*, pp. 62–75, Springer, 2009.

[7] K. O'Hara, M. Glancy, and S. Robertshaw, "Understanding collective play in an urban screen game," in *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work, CSCW 08*, pp. 67–76, USA, November 2008.

[8] M. Factory, "Grid: Transforming public spaces with collaborative play," https://momentfactory.com/lab/grid, 2018.

[9] I. Akpan, P. Marshall, J. Bird, and D. Harrison, "Exploring the effects of space and place on engagement with an interactive installation," in *Proceedings of the 31st Annual CHI Conference on Human Factors in Computing Systems: Changing Perspectives, CHI 2013*, pp. 2213–2222, France, May 2013.

[10] S. R. Siegel, B. L. Haddock, A. M. Dubois, and L. D. Wilkin, "Active video/arcade games (exergaming) and energy expenditure in college students," *International Journal of Exercise Science*, vol. 2, no. 3, pp. 165–174, 2009.

[11] A. B. Godbehere, A. Matsukawa, and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *Proceedings of the 2012 American Control Conference, (ACC '12)*, pp. 4305–4312, IEEE, Montreal, QC, Canada, June 2012.

[12] D. Slappendel, F. Lie, M. de Vos, A. Kopla, and R. Bidarra, "Paintrix: Color up your life!," in *Proceedings of the Advances in Computer Entertainment 2013, LNCS 8253*, pp. 576–579, Springer International Publishing, 2013.

[13] S. S. Snibbe and H. S. Raffle, "Social immersive media: Pursuing best practices for multi-user interactive camera/projector

exhibits," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '09. ACM*, 2009.

[14] J. Hu, D. Le, M. Funk, F. Wang, and M. Rauterberg, "Attractiveness of an interactive public art installation," in *Distributed, Ambient, and Pervasive Interactions*, N. Streitz and C. Stephanidis, Eds., pp. 430–438, Springer, Berlin, Heidelberg, 2013.

[15] E. Grenader, D. G. Rodrigues, F. Nos, and N. Weibel, "The VideoMob interactive art installation connecting strangers through inclusive digital crowds," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 2, p. 7, 2015.

[16] M. Hudson and P. Cairns, "The effects of winning and losing on social presence in team-based digital games," *Computers in Human Behavior*, vol. 60, pp. 1–12, 2016.

[17] D. Maynes-Aminzade, R. Pausch, and S. Seitz, "Techniques for interactive audience participation," in *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, ser. ICMI '02*, pp. 1–6, IEEE Computer Society, Washington, DC, USA, 2002.

[18] Itseez, "OpenCV - open source computer vision library," https://github.com/itseez/opencv, 2016.

[19] I. Gaztanaga, Boost.interprocess - 1.54.0. 2016, https://www.boost.org.

[20] J. P. Oakley and B. L. Satherley, "Improving image quality in poor visibility conditions using a physical model for contrast degradation," *IEEE Transactions on Image Processing*, vol. 7, no. 2, pp. 167–179, 1998.

[21] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.

[22] SFML, "Simple and fast multimedia library," https://github.com/SFML/SFML, 2016.

[23] R. Hassan, N. K. Yahya, L. M. Ong et al., "Public awareness program and development of education toolkit for green sea turtle conservation in sarawak, Malaysia," *International Journal of Environmental and Science Education*, vol. 12, no. 3, pp. 463–474, 2017.

[24] R. Bidarra, J. Boers, J. Dobbe, and R. Huijser, "Bringing a pioneer games project to the next level," in *Proceedings of the 3rd Annual Academic Days on Game Development in Computer Science Education*, pp. 11–15, ACM, Miami, Fl, USA, February 2008.

[25] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes, "A survey on procedural modelling for virtual worlds," *Computer Graphics Forum*, vol. 33, no. 6, pp. 31–50, 2014.

[26] R. Lopes and R. Bidarra, "Adaptivity challenges in games and simulations: A survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 2, pp. 85–99, 2011.