Spatial Augmented Reality Meets Robots: Human-Machine Interaction in Cloud-based Projected Gaming Environments

Giovanni PIUMATTI[1], Andrea SANNA[1], Marco GASPARDONE[2], Fabrizio LAMBERTI[1]
[1]Politecnico di Torino – DAUIN, Corso Duca degli Abruzzi, 24, 10129, Torino, Italy
[2]TIM JOL Connected Robotics Applications laB, Corso Montevecchio 71, 10129 Torino, Italy
{giovanni.piumatti, andrea.sanna, fabrizio.lamberti}@polito.it, marco.gaspardone@telecomitalia.it

Abstract-- **Augmented Reality (AR) is expected to change the way we play, by transforming the world around us in an incredibly rich gaming environment. In this work, connected robots and natural interaction means are combined with projected AR to create a gaming experience more physical and engaging.**

I. Introduction

Continuous advancement in technologies enabling Virtual Reality (VR) and Augmented Reality (AR) applications will change the way we perform many of our daily life activities in a variety of fields. One of these fields is indeed entertainment and, in particular, gaming. According to [1], for the gaming industry AR will probably play a more important compared to VR, since it will give players the possibility to physically move in open spaces and interact with real objects, bringing the game experience to a more primordial level.

Outdoor AR gaming is generally tackled by means of handheld solutions on mobile devices, whereas indoor gaming often relies on large screens (tabletops or projected displays). In the middle sits wearable AR, which makes use of see-through glasses to provide the players with a wide field of view which, however, required for a long time bulky equipment that severely limited portability.

Although developments on portable holographic displays will probably boost wearable AR, today the most effective way for providing players with a strong physical engagement thanks to the affordances offered by immersiveness is via projected AR (often referred to as spatial AR, or SAR).

Besides immersing the players in the virtual environment and letting them naturally interact with the game, e.g., with their body, based on the findings of a number of behavioral studies a strategy that has been pursued to improve the physicality of the experience has been to introduce robotic elements acting as active gaming objects. In fact, compared to virtual objects, robots are generally expected to be capable of providing the users with a more engaging experience [2]. Robots could be either tele-controlled or endowed with autonomous behaviors, e.g., making them react to external stimuli from virtual and real objects or from human players.

In the literature, factors that could be used to improve the naturalness and immersiveness of gaming have been considered for a long time only in separate works. Only recently, the concept of a SAR-based architecture for robotic gaming fusing the above factors in a unique design based on cloud computing technologies was introduced [3]. In this work, an implementation based on the above architecture is proposed, and used to create a shooting game incorporating a commercial robot where the players can embody both virtual and robotic elements that can be controlled both using body gestures and a mobile app. The aim was to investigate players' reaction to different interaction means and roles in the game, as well to study the effectiveness of AR robotic games, by possibly identifying guidelines for the development of future platforms and gaming scenarios.

II. Background

According to [1], an interesting way to classify AR could be to consider the technology used for visualizing the game, which broadly encompasses handheld, wearable (head-mounted see-through), and projected displays.

Thanks to the tremendous diffusion of smartphones and tablet devices, handheld AR is by far the technology that is best known by consumers, which is foreseen to be adopted by more than a billion users in the next few years. This translate into an incredible market for AR apps, including games.

For years, the situation of wearable and projected AR has been slightly different, mainly due to setup requirements. However, compact holographic displays that are appearing on the market promise to revolutionize the idea itself of head-mounted AR devices [4]. Similarly, prototype solutions like the one in [5] let us envision for the near future the possibility to turn a whole room in an immersive gaming environment.

Besides focusing on output technology, researchers from both the academy and industry are working towards the identification of the input technologies best suitable for effective AR (and VR) HMI. In this context, many forms of natural and multi-modal interaction paradigms capable to make the user experience more engaging and intimate have been experimented [6].

The physical empowerment offered by AR and input/output technologies can be further boosted by introducing robots behaving as active gaming elements. The use of robots in games is not new. In fact, robots have been exploited as game companions in static scenarios, e.g., for playing chess, cards, etc., since from the early '90s [7]. More recently, advancements in the field of mobile robots and object tracking made it possible to implement more dynamic games, from soccer, to "red light – green light" (a.k.a. "statues"), etc. [8].

Most of the commercial gaming solutions developed for existing toy robots are deployed as AR mobile apps. An example of handheld AR-based robotic games developed by the academy is reported in [9], where multiple tele-operated robots can be used as avatars in a shooting game for handheld

978-1-5090-5544-9/17/$31.00 ©2017 IEEE

devices. Examples based on wearable technology are quite rare [10], whereas several projected AR gaming implementations can be found in the literature. In some cases, tabletop setups are used, like in [11] and [12], where mobile robots can be either controlled remotely by means of a gaming controller or touch gestures, or used as tangible props to interact with other physical and virtual objects. In other cases, floor projection is used, like in [13] and [14].

In order to gain the required control capabilities, researchers often exploited do-it-yourself (DIY) robots rather than off-the-shelf hardware. Although the use of custom-made robots could be regarded already as a possible drawback of current solutions in terms of market possibilities, the major limitation of the above examples is that interaction with the robot and the environment is generally based on traditional interfaces (e.g., joystick, touch, or other robot's native interfaces) rather than on mechanisms that are supposed to be capable of boosting physicality and engagement.

III. Proposed SAR-Based Robotic Gaming System

The aim of this work is to present a system capable to cope with the limitations of existing AR-based gaming solutions. As shown in Fig. 1, the functional architecture of the proposed SAR-based robotic gaming system can be split into two parts: a front-end, i.e., the game itself, which implements graphics, sounds, characters' artificial intelligence (AI), game logic, etc., and a back-end, which takes care of the more computationally complex tasks, such as player and robot tracking. The two parts communicate through a websocket. This way, they can be executed on separate machines and, according to the scalable design philosophy adopted in [3], the back-end can be deployed in the cloud and re-used to implement different games possibly with other kinds of robots.

A. Back-end

The back-end is built upon the messaging infrastructure of the Robot Operating System (ROS) [15], and consists of four main modules (i.e., ROS nodes, in gray): *people tracking*, *robot tracking*, *robot control* and *coordinate system management*.

The people tracking module is in charge of detecting and tracking human players in the game area. In the current implementation, it exploits the OpenPTrack library [16], which is fed with frames captured by a RGB-D camera[1]. The positions of the various human players are then projected onto the floor plane and clipped to the projected game area.

The robot tracking module is responsible for detecting and tracking mobile robots. Unlike the people tracking module, which leverages universal features of the human body to implement detection, according to the design principles stated in [3] the robot tracking module cannot make any assumption about the aspect of the robot used. Therefore, it was decided to implement it using TLD (Tracking-Learning-Detection) [17], one of the most robust and efficient general-purpose online tracking algorithms reported in the literature. The module can be initialized either manually, by defining a bounding box around the robot to be tracked, or with a pre-existing model, e.g., created by a previous run of the algorithm. Currently, a second RGB camera with its own field of view is used to track the robot, in order to ensure proper performance. Should the robot provide an odometry, the devised tracking module can fuse it with the TLD tracking data by means of a Kalman filter, thus making the tracking more robust (when the robot is occluded or has fallen outside the field of view of the camera).

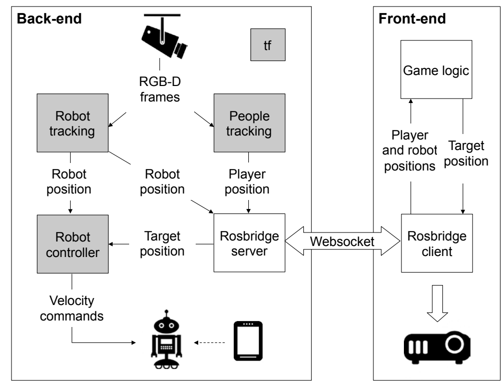


Fig. 1. Architecture of the devised SAR-based robotic gaming platform.

The robot control module is based on a Proportional Integral Derivative (PID) controller, which adjusts robot velocity based on current and target positions. Once correctly tuned, the controller proved to be capable to move the selected robot[2] quickly and precisely on the target, with little to no overshoot. This approach simplifies high-level control of the robot, as the game logic will just need to constantly provide a target position. The controller will ensure that the robot follows the target as closely as possible (currently, the PID controls speed only, as the tested game works on a single dimension; however, it should be trivial to extend to yaw control as well).

The above modules work in different coordinate systems. Thus, a way to manage transformations between multiple coordinate frames is needed. This functionality is provided by the *tf* package [18], which allows any ROS node to publish dynamic transformation matrices between coordinate frames, and to transform data from one frame to another. The main coordinate systems that are defined in the devised architecture are: the projection, the cameras, the robot and its odometry reference. Camera-projection transformation matrices are required to translate people or robot tracking information into 2D positions within the game area. The robot's odometry reference is an arbitrary coordinate system, typically defined as the robot's starting pose. The camera-projection and projection-odometry transformation matrices allow to compare and fuse odometry with optical tracking. Finally, the robot's own reference frame is used to convert absolute velocity commands (i.e., defined in the projection reference frame) into relative ones, actuated by the robot.

B. Front-end

The front-end is implemented as a library for a well-known 3D game engine[3]. The library consists of a Rosbridge client[4]

[1] https://developer.microsoft.com/en-us/windows/kinect/develop

[2] http://www.parrot.com/it/prodotti/jumping-sumo/

978-1-5090-5544-9/17/$31.00 ©2017 IEEE

that handles the communications with the back-end, and a set of high-level scripts that facilitate interaction between physical and virtual elements. This is achieved by creating a virtual replica of the physical world, or at least of the main actors participating in the game. In other words, every physical element of the game (i.e., robot and human players) has a virtual counterpart, which mimics the movement of the physical entity using tracking data from the robot tracking or the people tracking modules. The game engine is thus able to simulate interactions between the game elements, creating the illusion that the physical world can affect the digital one, and vice versa. Moreover, any robot in the game features an additional virtual counterpart, which represents its target destination. By moving this object in the virtual world, the back-end controller moves the robot in the real world, closing the loop. Finally, the library offers various utility features, such as tracking area configuration, robot movement scheduling, etc.

C. Game Implementation

The game implemented for testing the system is a simple shooting game featuring two characters that move only in one direction: one drops bombs from the top of the gaming area (the *bomber*), the other tries to avoid them (the *runner*). The game is configurable along multiple dimensions: characters' avatar (whether its representation is virtual or robotic), avatar control (AI, or human), control interface (body tracking or robot's native, which for the selected robot runs on a mobile device), and player's role. The goal is to survive as long as possible (when playing the role of the runner), or to hit the runner as quickly as possible (if playing the role of the bomber). As time passes, the game logic makes AI-controlled characters smarter (for instance, the bomber becomes faster and more precise). One of the possible configurations is illustrated in Fig. 2. In this case, the player is controlling the virtual avatar close to his feet (runner) by means of body movements. The robot (bomber) is controlled by the game AI.

Fig. 2. Gameplay for a configurations where the player is using his body to move the virtual avatar, whereas the game AI is controlling the robot.

[3] http://unity3d.com/
[4] http://wiki.ros.org/rosbridge_suite

IV. EXPERIMENTAL RESULTS

Several preliminary experimental tests were carried out to study the effectiveness of the proposed design and get insights about user experience, which could be possibly exploited to drive the development of next-generation AR-based robotic gaming solutions. In particular, the focus of the analysis was on the added value provided by the presence of the robot (anticipated already by previous studies, but in different settings [2]), on the suitability of the considered interaction mechanisms (specifically, on advantages that were expected to be linked with the adoption of a body tracking-based natural interface compared to traditional ones), as well as on players' preference about the character controlled and the role played.

Out of all the possible configurations that can be set up in the implemented gaming environment, three of them were regarded as more interesting, being capable, in principle, to shed some light on all of the above aspects: (a) body-controlled virtual runner with AI-controlled robot bomber, (b) body-controlled robot bomber with AI-controlled runner, and (c) mobile device-controlled robot bomber with AI-controlled runner (in this case, the native app of the selected robot is used, which allowed the player to move the robot left and right by means of a slider or the gyro sensors of a handheld device).

Configurations above were analyzed through a user study that involved 17 volunteers (13 males and 4 females, 30-year old, on average) selected among MS students at our university.

Volunteers were asked to fill in a pre-test questionnaire aimed to assess their prior experience with interfaces different than mouse and keyboard, video games and service robots.

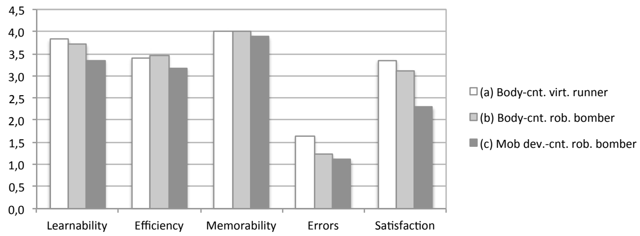
After the pre-test, they were invited to play the game in one of the three configurations (without any specific training). Configurations were selected in a random order to avoid bias. For each test, playing time and number of commands issued to move either the virtual character or the robot were recorded. Data collected are reported in Table I.

TABLE I
AVERAGE TIME AND NUMBER OF COMMANDS FOR EACH CONFIGURATION

Configuration	Time (s)	Number of comm.
(a) Body-control. virtual runner	59.1	15.6
(b) Body-control. robot bomber	42.4	17.6
(c) Mob. dev.-control. robot bomber	39.1	16.6

After the test, users were delivered a questionnaire based on Nielsen's Attributes of Usability [19] and SASSI metrics [20]. With the former tool, lernability, efficiency, memorability, (impact of) errors and satisfaction were measured. With the latter tool, complementary aspects about user interaction were evaluated, like system response accuracy, likeability, cognitive demand, annoyance, habitability and speed. Answers provided to questions in the two tools were collected in a 0 to 4 scale. Results are reported in Fig. 3 and Fig. 4, respectively.

The after-test questionnaire included a question that asked volunteers to say how much was the robot important in their opinion to make the game appealing (in 0 to 4 scale). Average rate was quite high, equal to 2.9 (standard deviation 1.1).



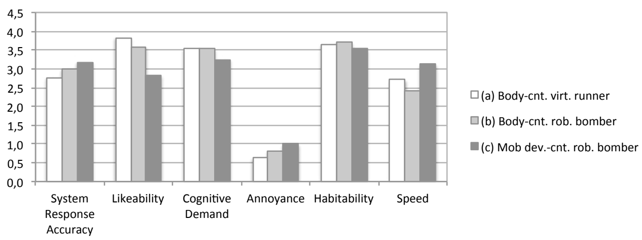


Fig. 3. Nielsen's usability factors for the three configurations (average rates).
Fig. 4. Usability (SASSI metrics) for the three configurations (average rates).

Finally, volunteers were asked to rank the three gaming configurations, by also providing motivations for the choices made. Results obtained are reported in Table II.

TABLE II
NUMBER OF TIMES A GIVEN CONFIGURATION WAS RANKED 1^{ST}, 2^{ND}, 3^{RD}.

Configuration	1^{st}	2^{nd}	3^{rd}
(a) Body-control. virtual runner	8	5	4
(b) Body-control. robot bomber	7	10	0
(c) Mob. dev.-control. robot bomber	2	2	13

Statistical significance of experimental observations was evaluated using Student's paired t-tests. Playing time was significantly higher in configuration (a) than in configurations (b) and (c). Since there was no significant difference in the number of commands issued, higher time in configuration (a) was probably related to the fact that, possibly due to differences in the AI for the bomber and the runner, it could be easier to avoid the bombs than to kill the runner by launching them. This fact should be taken into account in future game designs. With respect to interface usability in terms of Nielsen's factors, only satisfaction was significantly higher both in configurations (a) and (b) than in configuration (c). A comparable consideration can be made regarding likeability metric of the SASSI tool. This is an indication of the fact that, independent of the avatar used and the role played, volunteers liked more the natural interface than the mobile device-based one. The only other significant metric in the SASSI tool is speed, which was rated better for configuration (c) than for configurations (a) and (b). This was due to the fact that the tracking solutions adopted so far and the ROS infrastructure introduce a delay that make the system less responsive (and this was confirmed also by motivations provided for the rank).

About ranks, volunteers largely preferred configurations (a) and (b). Aggregated preferences for the 1^{st} and 2^{nd} ranks show that volunteers preferred configuration (b) since, as emerged in the after-test questionnaire, they liked more controlling the robot (with the body) than the virtual avatar (with the body).

V. CONCLUSIONS AND FUTURE WORK

In this work, a prototype implementation of a SAR-based robotic game has been presented, and a user study performed to preliminary assess the added value brought by the use of robots and by the adoption of natural control interfaces. Future work will be aimed to foster the scalability of the designed system, by making it capable to support the implementation of different types of games using many types of consumer robots.

REFERENCES

[1] B. H. Thomas, "A Survey of Visual, Mixed, and Augmented Reality Gaming," *Computers in Entertainment*, vol. 10, no. 3, pp. 1-33, 2012.
[2] A. Powers, S. Kiesler, S. Fussel, and C. Torrey, "Comparing a Computer Agent with a Humanoid Robot," in *Proc. ACM/IEEE Intl. Conf. on Human-robot Interaction*, pp. 145-152, 2007.
[3] M. L. Lupetti, G. Piumatti, and F. Rossetto, "Phygital Play: HRI in a New Gaming Scenario," in *Proc. 7th Intl. Conf. on Intelligent Tech. for Interactive Entert.*, pp. 17-21, 2015.
[4] L. Avila, and M. Bailey, "Augment your Reality," *IEEE Computer Graphics & Applications*, vol. 36, no. 1, pp. 6-7, 2016.
[5] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi, and L. Shapira, Room Alive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units," in *Proc. 27th Annual ACM Symp. on User Interface Software and Technology*, pp. 637-644, 2014.
[6] M. Billinghurst, H. Kato, and S. Myojin, "Advanced Interaction Techniques for Augmented Reality Applications," in *Proc. 3rd Intl. Conf. on Virtual and Mixed Reality*, pp. 13-22, 2009.
[7] F. C. A. Groen, G. A. der Boer, A. van Inge, and R. Stam, "A Chess-playing Robot: Lab Course in Robot Sensor Integration," in *Proc. 9th Instrumentation and Meas. Tech. Conf*, pp. 261-264, 1992.
[8] K. Sakai, Y. Hiroi, and A. Ito, "Playing with a Robot: Realization of "Red Light, Green Light" using a Laser Range Finder," in *Proc. 3rd Intl. Conf. on Robot, Vision and Signal Proc*, pp. 1-4, 2015.
[9] D. Estevez, J. G. Victores, S. Morante, and C. Balaguer, "Robot Devastation: Using DIY Low-Cost Platforms for Multiplayer Interaction in an Augmented Reality Game," in *Proc. 7th Intl. Conf. on Intelligent Tech. for Interactive Entert.*, pp. 32-36, 2015.
[10] S. O'Kane, "Microsoft Used this Adorable Robot to Show Off new HoloLens Features," *The Verge*, April 2015.
[11] D. Calife, J. L. Bernardes, R. Tori, "Robot Arena: An Augmented Reality Platform For Game Development," *Computers in Entertainment*, vol. 7, no. 1, art. 11, 2009.
[12] H. Mi, A. Krzywinsk, T. Fujita, and M. Sugimoto, "RoboTable: An Infrastructure for Intuitive Interaction with Mobile Robots in a Mixed-Reality Environment," *Adv. in Human-Comp Int.*, vol. 2012, art. 1, 2012.
[13] M. Kojima, M. Sugimoto, A. Nakamura, M. Tomita, H. Nii, and M. Inami, "Augmented Coliseum: An Augmented Game Environment with Small Vehicles," in *Proc. 1st IEEE Intl. Workshop on Horizontal Interactive Human-Computer Systems*, 2006.
[14] D. Robert, R. Wistorrt, J. Gray, and C. Breazeal, "Exploring Mixed Reality Robot Gaming," in *Proc. 5th Intl. Conf. on Tangible, Embedded, and Embodied Interaction*, pp. 125-128, 2011.
[15] S. Cousins, B. Gerkey, K. Conley, and W. Garage, "Sharing Software with ROS [ROS Topics]" *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 12-14, 2010.
[16] M. Munaro, A. Horn, R. Illum, J. Burke, and R. B. Rusu, "OpenPTrack: People Tracking for Heterogeneous Networks of Color-Depth Cameras," in *Proc. 1st Intl. Workshop on 3D Robot Perception with Point Cloud Library*, pp. 235-247, 2014.
[17] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, 2012.
[18] T. Foote, "tf: The Transform Library," in *Proc. IEEE Intl. Conf, on Tech. for Practical Robot Applications*, pp. 1-6, 2013.
[19] J., Nielsen, Usability Engineering. *Academic Press*, 1993.
[20] K. S. Hone, R. Graham, "Towards a Tool for the Subjective Assessment of Speech System Interfaces," *Natural Language Engineering*, vol. 6, no. 3-4, pp. 287-303, 2000.