

Game Room Map Integration in Virtual Environments for Free Walking

Marilyn Keller*
Gfi Informatique

Frédéric Exposito†
Gfi Informatique

ABSTRACT

Current tracking systems now enable real walking in a virtual scene with a Head Mounted Display (HMD). However, the play area usually remains limited to a few square meters because of tracking limits and the lack of free space in game rooms. This paper describes our demonstration showing how we can use an RGB-D sensor to increase the real game surface by dynamically acquiring a map of the actual game room and integrating it into the virtual environment. Our system was designed to be integrable to any virtual environment and aims to enable free walking with a HMD by showing the position of the real obstacles to the user.

Index Terms: Human-centered computing—Visualization—Visualization techniques; Human-centered computing—Visualization—Interaction Paradigm—Virtual Reality Computing methodologies—Computer Graphics—Image Manipulation

1 INTRODUCTION

Currently, virtual reality (VR) devices are multiplying and becoming more and more accessible. They allow to immerse oneself in a virtual environment, offering the possibility to look and move around, as well as interacting with the environment. However, there are still challenging issues that have to be addressed to enhance presence. One of them is the ability to move around in the virtual scene. Indeed, it has been shown that using walking to navigate in VR generally improves the feeling of immersion [3]. Devices like the Oculus Rift and HTC Vive are able to track the headset position and thus allow the user to move in the real world by walking. But the tracked area remains limited to a few square meters. Existing solutions include showing the boundaries of this restricted area as the HTC Vive does with Chaperone. Another solution is Redirected Walking that forces the user to stay within a defined area by applying transformations to the displayed scene [2].

Another approach called Simultaneous Localization and Mapping (SLAM) can perform tracking on wider surfaces. Mostly used in robotics, this technique allows to map the environment and determine the user position at the same time. It can work with various sensors including cameras or depth cameras. With such a method, it is possible to explore virtual environments by walking. The remaining limit is then the available free space and the presence of obstacles, especially in typical homes or office environments. Obstacle avoidance is a classic issue in robotics and navigation for visually impaired people. To avoid collisions, Triebel et al. [5] use depth cameras to create a traversability map of the environment. This map represents the safe path one can follow through the environment and the obstacles. It can be used to generate navigation instructions, which opens up various applications for navigation in virtual environments.

Some other works exploited point clouds and SLAM for free walking in virtual environments. Nescher et al. [1] combine redirected walking with tracking based on SLAM and Sra et al. [4]

*e-mail: marilyn.keller@gfi.fr

†e-mail: frederic.exposito@gfi.fr

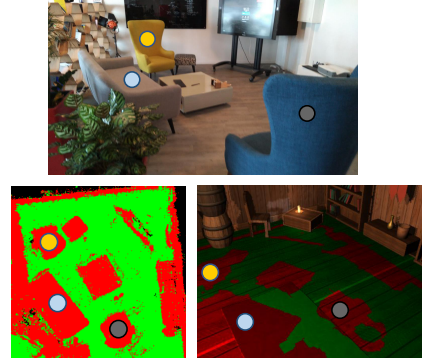


Figure 1: Illustration of our system. (top) The mapped game room. (left) The computed traversability map. (right) We integrate the traversability map to the virtual scene.

build a 3D scan of the room from captured point clouds and extract the traversable zone to generate procedurally a virtual environment. Compared to those works, our approach allows for the exploration of any virtual environment without requiring neither an external computer, nor a prior scan of the room, nor a procedural description of the virtual scene.

We demonstrate a system that enables the user to explore any virtual environment by walking in any space like an apartment or an office. The principle is to integrate navigation information into the virtual scene, such that the user knows where the real obstacles are and can avoid them (Fig. 1). Our approach consists in using a RGB-D sensor to generate and update in real time a 2D map of the game area and SLAM based tracking to allow the user to walk around the virtual environment with a HMD.

2 TECHNICAL DESCRIPTION

Our system is composed of a Googles Project Tango¹ smartphone attached to an adapted Cardboard VR viewer. The Tango smartphone is equipped with a color and depth camera, it can capture point clouds and perform tracking based on SLAM. We use the captured point cloud to generate a traversability map of the real game space and the tracking of the smartphone is used to control the virtual camera in the scene, such that the player can walk around.

Each cell of the map corresponds to a 2D position in the horizontal plane of the smartphone reference frame, so this map can then be positioned and scaled properly in the virtual scene, such that the user can walk on it and know where he or she is with respect to the real obstacles. We use this map to generate virtual content showing to the user the presence of obstacles or the free path.

2.1 Map Acquisition

To generate the traversability map, we project the point cloud captured by the Tango smartphone on the floor plane. We adopt the same map representation as [5]. The traversability map is thus represented as a 2D cell grid covering the game space. We tested different quantization and found that 2x2 cm cells give a good compromise

¹Googles Project Tango: <http://get.google.com/tango/>

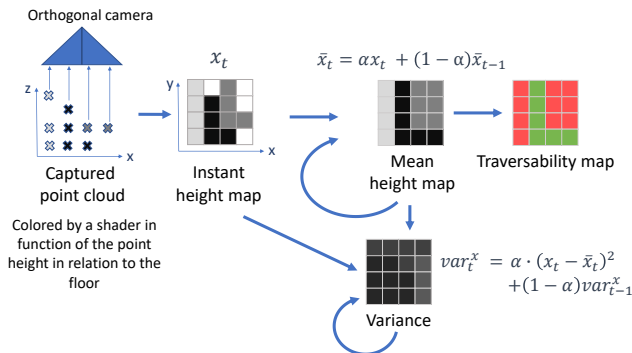


Figure 2: Maps computation from the point cloud. The computations from one map to another are done by Unity shaders. α controls the weight of a new frame measure.



Figure 3: The traversability map is integrated in real time to the virtual room floor texture (a) and can be used to generate content like lava lakes (b) where obstacles are or vegetation (c).

between the number of points projected in each cell and the final map resolution. With this quantization, we can store in a texture of 1024x1024 px, the map of a game space up to 20,48x20,48 m. For each frame, we project the captured point cloud on a map to get an instant height map of the field of view. We use this map to update a height map of the entire room, a map of the measured height variance and a map of the number of points measured by cell. We then use those maps to generate and update a traversability map by setting a threshold on the height but also on the variance and number of points for each cell. To set the height threshold, we use Tango’s API to detect the floor height out of a point cloud.

The points are projected with a Unity Camera and all the computations from the instant height map to the traversability map are done by shaders applied successively on the different textures, so the computations are not heavier than a second camera render which allows the user to play while the map is acquired without major latency. The final map can then be simply displayed as a textured plane in a virtual scene on which the user can walk, being aware of the obstacles. The pipeline is summarized in Fig. 2.

To make possible the integration of our system to any virtual environment in Unity, we divide our Unity scene into three blocks to create a hierarchy we name *RMV* (Real world, Mixed world, Virtual world), in order to separate the different kinds of contents. The *Real world* block contains all the data we got from the real environment, namely the maps (updated on the fly or loaded from memory). The *Virtual world* block contains the virtual environment with all the possible interactions and the *Mixed world* block contains the content generated from the real world that is integrated into the virtual environment. With this separation, we can integrate navigation information in different forms in the *Mixed world* block independently of the virtual environment.

2.2 Map Integration

For virtual environment exploration with real obstacles, path planning is very important. Indeed, if the user wants to go from a point A to a point B in the virtual scene, there may be only one possible

path through the real obstacles, so the user has to know their positions in advance. Moreover, if the virtual content generated from the map is complex, it can be too time-consuming to update this content every frame from the updated traversability map. Thus, we added a second game mode that enables to scan a game space and use the map later. In this case, we post-process the traversability map with a median filter before saving it, which partly removes the noise. Beside SLAM, Tango is able to perform Area Learning. Thus, once a room has been learned, the Tango smartphone can orient and position itself in the room at any time. So, we are able to position and scale properly a map that has been saved before. Furthermore, the user only needs to scan a game space once, then the map can be used in different games or scenarios.

We can integrate the captured traversability data into a virtual scene in several ways. We implemented 3 visualization modes of the traversability map. Some are dynamics while others are generated from a saved map of a game room. The different representations are illustrated in Fig. 3. The first one integrates the traversability map into the existing virtual floor texture, the second one generates a mesh from the map to represent the obstacles as lava lakes and the third representation shows the possibility to generate virtual objects where the real obstacles are, like vegetation.

3 DEMONSTRATION

Our demonstration presents a virtual shack that the user can walk around wearing a Google Cardboard VR viewer. A game space of a few square meters will be defined, with some obstacles within (table, chair). The user will have to go get several objects dispersed across the room while the free path will appear as a green and red map mixed to the virtual floor texture. Once the game space has been mapped, the user can click on a button on the VR Viewer to generate lava lakes where the obstacles are, and thus enter a second visualization mode. By clicking again, the lava lakes disappear and virtual vegetation is generated where the real obstacles are. Fig. 3 shows the three visualization modes.

4 DISCUSSION AND CONCLUSION

Current VR devices enable free walking but not yet in any space. RGB-D cameras are more and more accessible and are a good way to get information on the occupancy of an area. Our demonstration shows how we can combine those tools to integrate navigation information into a virtual scene. Building a 2D map from the point cloud simplifies the traversability data from 3D to 2D and makes easier the generation of virtual content to indicate obstacles.

The limitation of such a system remains the map size and the limitation to game spaces with constant floor height since we do not handle steps. Moreover, knowing where the obstacles are does not guaranty that the user can go anywhere in the scene, which can be quite frustrating. Further work will study what are the best metaphors for representing real obstacles in VR and methods to improve the accessibility of a virtual scene by free walking.

REFERENCES

- [1] T. Nescher, M. Zank, and A. Kunz. Simultaneous mapping and redirected walking for ad hoc free walking in virtual environments. In *Proc. VR*. IEEE Computer Society, Los Alamitos, 2016.
- [2] S. Razaque, Z. Kohn, and M. C. Whittom. Redirected walking. In *Proc. EUROGRAPHICS*, vol. 9, pp. 105–106, 2001.
- [3] R. A. Ruddle and S. Lessels. The benefits of using a walking interface to navigate virtual environments. *ACM Transactions on Computer-Human Interaction*, 16(5), Apr. 2009.
- [4] M. Sra, S. Garrido-Jurado, C. Schmandt, and P. Maes. Procedurally generated virtual reality from 3d reconstructed physical space. In *Proc. VRST*, pp. 191–200. ACM, New York, NY, USA, 2016.
- [5] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proc. IROS*, pp. 2276–2282. IEEE Computer Society, Los Alamitos, 2006.