Ground plane synchronization in VR applications using indoor robots for enhancing immersion

Divya J Udayan[1] ,Hrishikesh P[2] , Nithin Sylesh[3] Madhav M Menath[4],Yadukrishnan J[5]

[1] Department of Computer Science & Engineering
Amrita Vishwa Vidyapeetham
Amritapuri Campus, India

Abstract. Many techniques have been developed to improve locomotion while using VR applications. Although VR platforms or large open spaces allow natural locomotion, they are not feasible for all users. As a result, many users are confined to small workspaces which ensure their safety but severely affect immersion. In most cases, the VR application need not be aware of the spatial layout of the user's environment. This results in a workplace which does not take full advantage of the user's environment where, every time the user steps out of bounds in either the real world or the virtual world, their immersion with the virtual world detaches. Our work explores the possibility of synchronizing the ground plane of the virtual world to the ground plane of the real world using 2D mapping in an effort to create custom workspaces which ultimately enhances immersion by providing the user with freedom of natural movement without compromising their safety. We propose a pipeline that involves the use of indoor robots which could allow VR developers to customize their applications according to the end user's environment thereby solving the restricted workspace problem.

Keywords: Ground plane synchronization, Telepresence robots, VR, SLAM, Sub-optimal workspace.

1 Introduction

Natural motion is one of the key factors that contribute towards an immersive experience. Incorporating natural movement as a mechanic in VR applications makes them intuitive and significantly reduces the learning barrier while maintaining immersion. However, any restriction imposed on movement or causing the need for unnatural movement patterns without a believable cause can adversely impact user immersion and raise the learning barrier. Using large open spaces while using VR applications provides the user with a larger workspace where they can move freely. Another way to solve the limited workspace problem is to use VR platforms However, these methods are not feasible for all users who are often confined to a small workspace because the VR application isn't aware of the spatial layout of the user's environment. Most VR applications don't mimic the user's ground plane. This could lead to the user experiencing desynchronization with the real world and the virtual world. Subsequently, their immersion breaks. This also creates the possibility of accidents and therefore restricts

the use of such applications to safe and controlled workspaces that often do not take full advantage of the user's environment. The limited workspace problem often falls in the user experience domain where workarounds in the form of mechanics such as teleportation or encouraging the user to limit movement to a few strides are adopted. However, whenever a restriction of movement is felt, the immersion is affected adversely.

1.1 Ground plane synchronization

When the VR application is aware of the spatial layout of the user's environment, the workspace can be customized to take up more space without compromising safety. However, synchronizing the virtual world ground plane to the real world's ground plane has an added advantage - VR developers could customize their application to adapt to the user's environment. Users could traverse the whole environment without compromising their safety since all the inaccessible areas in the real world would be mirrored in the virtual world. This could also mitigate motion sickness to some extent as in-game movement would be in complete sync with the real-world movement. Figures 1 demonstrates ground synchronization of the virtual world to the real world. Fig. 1.1 shows the ground plane of a room in the real world. The space where the user can traverse is marked with green and the non-traversable areas are marked with red. There are 4 non-traversable areas in the room denoted as A, B, C, D respectively. Fig. 1.2 shows the ground plane of the virtual world. The space where the digital avatar can traverse is marked with green and the non-traversable areas are marked with red. There are 3 non-traversable areas in the virtual world denoted as 1,2,3 respectively. Fig. 1.3 shows the synchronization of the virtual world ground plane to the real-world ground plane. The goal is to alter the ground plane of the virtual world such that all the non-traversable areas in the real world are non-traversable in the virtual world. The scale, rotation and rotation of areas 1,2,3 is aligned with A, B and C respectively. An additional non-traversable area 4 is added in the virtual world in alignment with D. The modified virtual ground plane is shown in Fig. 1.4. A user accessing the virtual world using a VR headset can now navigate freely in the room because the traversable areas of the real world are the same as the traversable areas of the virtual world. By synchronizing the ground plane of the virtual world to the real world, they can use natural movement methods which increase their immersion without compromising their safety.

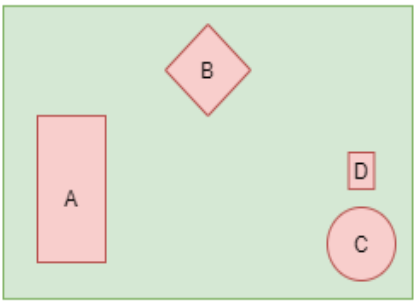

Fig. 1.1 Ground plane of real-world

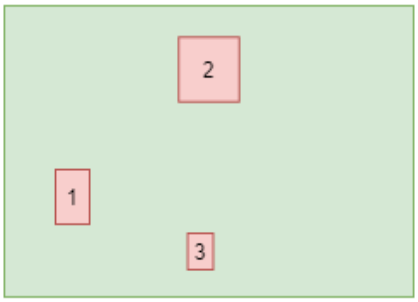

Fig. 1.2. Ground plane of virtual world

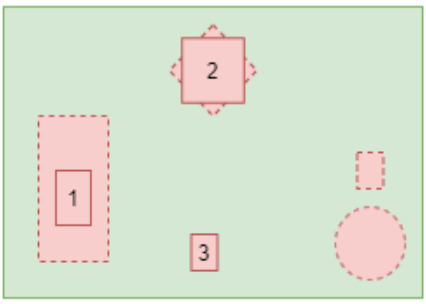

Fig. 1.3. Synchronizing ground plane of virtual world to real world

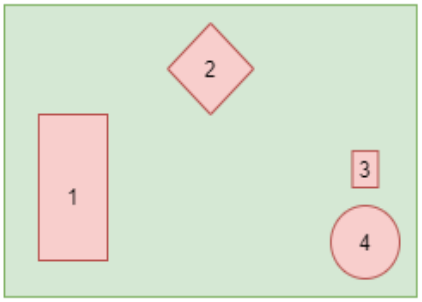

Fig. 1.4. Modified ground plane of virtual world

Fig. 5. Illustration of ground plane synchronization

1.2 2D mapping using indoor robots

This paper proposes a pipeline that could potentially enable VR application developers to synchronize the virtual world ground plane to the real-world ground plane using indoor robots that map the environment to find out which areas are accessible and where the obstacles in the environment are located. The map generated by the robot would be used to generate heightmaps and 3D models of the environment which could then be used by VR developers to customize their applications in order to create an optimal workspace that takes full advantage of the user's environment. The proposed method also encourages the use of robots and their digital avatars in VR environments. This paves way to numerous possibilities from a user experience perspective. The robot could assist the user in the real world and its actions could be mirrored using digital avatars in the virtual world.

2 Proposed Pipeline

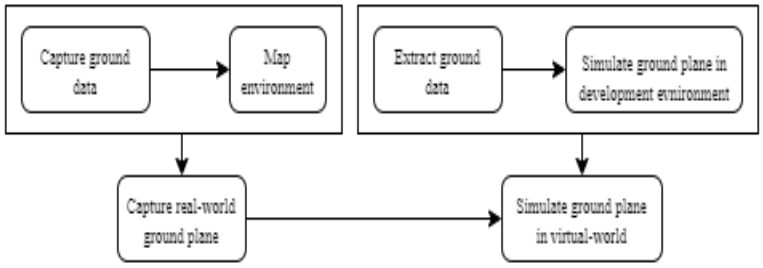

Fig. 6. Proposed pipeline for ground plane synchronization

Our pipeline, as shown in Fig. 2, encourages VR developers to synchronize the virtual-world ground plane of their VR applications to the real-world ground plane of the user. Developers might use different methods such as manual synchronization, procedural synchronization and so on. Therefore, deliberate care has been taken to make the pipeline independent of how the developer chooses to implement synchronization. The pipeline can easily be integrated into existing VR development pipelines as the end results of the pipeline are heightmaps and 3D models which are familiar assets for VR developers. The pipeline is also easily customizable. There are 2 main segments in the pipeline - capturing the real-world ground plane and simulating the real-world ground plane.

2.1 Capturing the real-world ground plane

The first segment in the pipeline deals with capturing the spatial layout of the user's environment. We use indoor robots to create a map of the environment. In order to map the entire environment autonomously, mapping algorithms such as SLAM and its variants can be used. This ensures that the robot creates a map of the entire environment without any redundancy in ground data. Using SLAM over other methods also ensures easy localization as well as object tracking. However, in our work, it has been assumed that the user works in a static environment without dynamic obstacles. The LIDAR SLAM method has been used instead of the Visual SLAM method and as a result we obtain a point cloud containing location data which is much simpler to handle in the later steps. Since the goal is to just recreate the ground plane, LIDAR data is preferred. Manual mapping methods such as capturing LIDAR data during tele-operation of the robot is also a viable option and also gives the user complete freedom in defining the workspace.

2.2 Simulating the real-world ground plane

In the second segment, the real-world ground plane is simulated in the virtual world. Here, several requirements of VR developers have been taken into account. It was clear that instead of using the data captured by the robot in real-time, the developers would

be able to better visualize and work with the data if it was converted into a heightmap or a 3D model. For the purposes of this work, a standalone application has been developed for the simulation of the ground plane. It takes the map generated by the robot and converts it into a heightmap. As the heightmap stores the displacement of each pixel, developers will be able to easily visualize the real-world ground plane using it. They could also make custom changes and remove noise captured during the mapping process using a simple image editing tool. The heightmap could then directly be used to create terrains in game engines such as Unreal Engine or Unity. Users also have the option to construct a 3D projection of the heightmap using the application. In order to construct a 3D projection, the Delaunay triangulation method has been used. The use of 3D projections of the real-world ground plane would be very helpful for VR developers while synchronizing their virtual-world ground plane to the real-world ground plane.

2.3 Testbed

In order to capture the real-world ground plane, turtle Bot simulations as shown in Fig. 3 were performed in Gazebo. The results were visualized, tested and compared using Rviz. In order to simulate the real-world ground plane, a standalone application that generates height maps as well as 3D projections based on the mapping data was built using Python. The resulting height map or 3D model can be directly used in game engines such as Unreal or Unity and in 3D modeling software such as Blender or Maya.

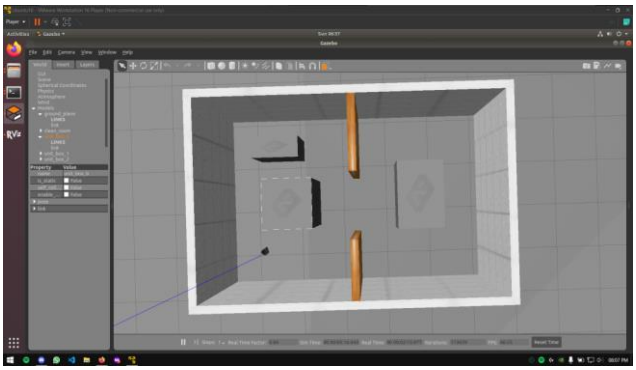

Fig. 7. Bot begins mapping the environment

3 Experiment Result

3.1 Capturing the real-world ground plane

For autonomous mapping, the LIDAR SLAM method was used. The 2D point cloud generated accurately captures the real-world ground plane. Using the SLAM algorithm also has the added advantage of tracking dynamic obstacles if needed. In the experiments conducted for this work, the robot was expected to map a room full of static

obstacles. It was able to do so successfully with the point cloud corresponding to the ground truth accurately and the margin of error was low. The robot could successfully map the entire room including the obstacles placed in them without entering into redundant loops or leaving areas un-mapped (shown in Fig. 4).

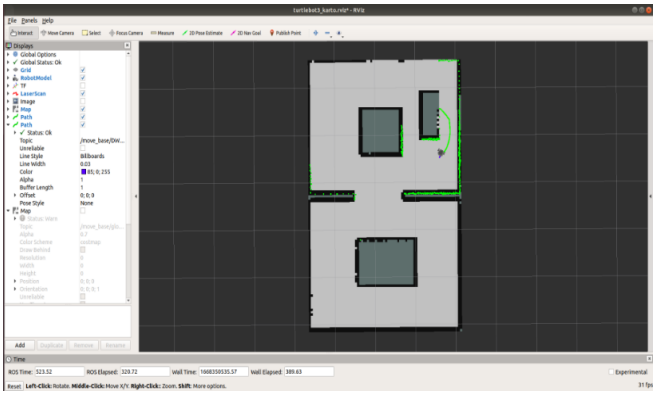

Fig. 8. Completely mapped environment

3.2 Simulation of real-world ground plane.

A standalone application, shown in Fig. 6, has been created that converts the 2D point-cloud generated by the robot into a heightmap. This allows the developers to easily visualize the real-world terrain beforehand and make adjustments using a simple image editor if needed. This step also helps in removing a lot of noise that might be captured by the robot, using thresholding. A 3D projection of the spatial layout can also be generated using the application. The VR developers can then use either the heightmap (shown in Fig. 7) or the 3D projection (shown in Fig. 8) to customize their applications so as to provide optimal workspaces for the users or they can synchronize the ground plane of their virtual world to the real-world to create exciting VR experiences.

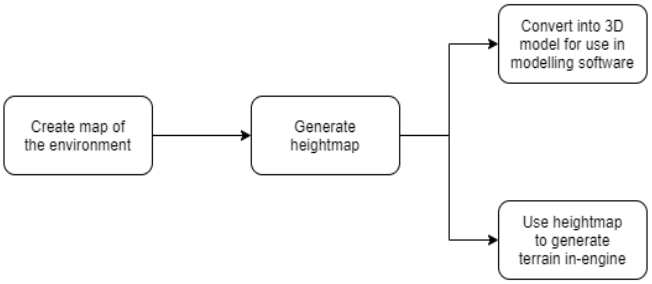

Fig. 9. Process flow for standalone application

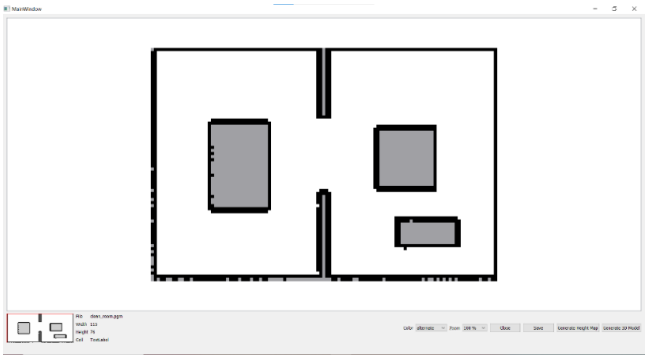

Fig. 10. GUI of standalone application

Fig. 11. Heightmap generated by application

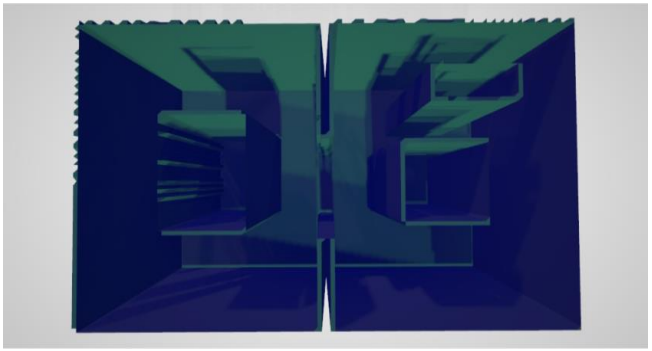

Fig. 12. 3D model generated by application

A blockout of a VR dungeon level, shown in Fig. 9, was created in Unreal Engine 5. Using the 3D model generated by the application as a reference, the ground plane of the dungeon was synchronized to the ground plane of the room (shown in Fig. 10). This would provide the user with increased workspace and would enable them to move naturally without the risk of entering non-traversable spaces inside the room.

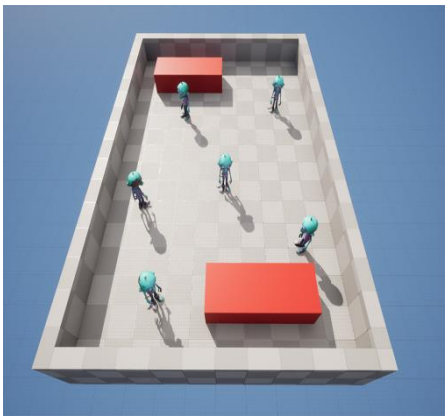

Fig. 13. Blockout of dungeon

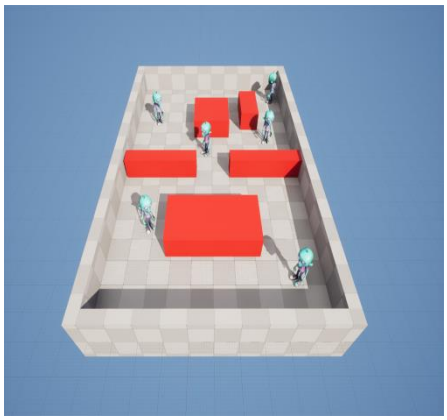

Fig. 14. Modified dungeon

4 Conclusion and Future Work

The highly customizable pipeline proposed in our work is easy to use and modify and can be easily integrated into existing VR application development pipelines. The use of robots for mapping also encourages the use of digital avatars of these robots in casual VR applications. This opens up a lot of applications in the field of teleoperation, gaming and so on. Using robots to capture the spatial layout of the environment is not an error-free process. Estimation of sequential movement ultimately leads to some margin of error. This can be due to several factors such as sensor calibration, wheel movement of the robot, etc. However, if this error is large then, it can accumulate and cause deviation from the actual coordinates which ultimately results in a distorted map. One countermeasure is to determine the calibration errors of the robot beforehand and correct the measurement values. Better algorithms could result in more precise and faster map generation. Sub-optimal workspaces in VR applications are a problem that has adversely impacted user immersion. Synchronization of the virtual world ground plane to the real-world ground plane using indoor robots is a novel method that provides the user with an optimal workspace that encourages them to take full advantage of the real-world environment without compromising safety. It also adds a whole new level of customizability and authenticity to VR applications.

REFERENCES

1. M. Jaswanth, N. K. L. Narayana, S. Rahul and M. Supriya, "Autonomous Car Controller using Behaviour Planning based on Finite State Machine," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 296-302, doi: 10.1109/ICOEI53556.2022.9776684.
2. Anupama K. Ingale, Divya Udayan J.,Real-time 3D reconstruction techniques applied in dynamic scenes: A systematic literature review,Computer Science Review,Volume 39,2021,100338,ISSN 1574-0137,https://doi.org/10.1016/j.cosrev.2020.100338.
3. Ingale, A.K., Divya Udayan, J., Patil, S.P. (2023). VREd: Virtual Reality in Engineering Education for Immersed and Interactive Learning: A Case Study. In: Fong, S., Dey, N., Joshi, A. (eds) ICT Analysis and Applications. Lecture Notes in Networks and Systems, vol 517. Springer, Singapore. https://doi.org/10.1007/978-981-19-5224-1_55
4. Yeonseo Lee, Hyeonjung Lim, Yeunhee Kim & Youngsu Cha (2020). "Thermal Feedback System From Robot Hand for Telepresence." IEEE Access. https://doi.org/10.1109/ACCESS.2020.3047036
5. Wei Li and Rong Xiong. "Dynamical Obstacle Avoidance of Task- Constrained Mobile Manipulation Using Model Predictive Control." IEEE Access (Volume: 7, 02 July 2019). https://doi.org/10.1109/ACCESS.2019.2925428.
6. Kai Zhu & Tao Zhang (2021). "Deep reinforcement learning based mobile robot navigation: A review." Tsinghua Science and Technology. https://doi.org/10.26599/TST.2021.9010012
7. Ke Zhang, Jiayu Cao & Yan Zhang (2022). "Adaptive Digital Twin and Multiagent Deep Reinforcement Learning for Vehicular Edge Computing and Networks" IEEE Transactions on Industrial Informatics. https://doi.org/10.1109/TII.2021.3088407.
8. Guoqi Xie, Kehua Yang, Cheng Xu, Renfa Li & Shiyan Hu (2022). "Digital Twinning Based Adaptive Development Environment for Automotive Cyber-Physical Systems" IEEE Transactions on Industrial Informatics. https://doi.org/10.1109/TII.2021.3064364
9. Yun Chang, Kamak Ebadi, Christopher E. Denniston, Muhammad Fadhil Ginting, Antoni Rosinol & Andrzej Rein (2022). "LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments" IEEE Robotics and Automation Letters. https://doi.org/10.1109/LRA.2022.3191204.
10. Haoyu Zhou, Zheng Yao, Zhuo Zhang, Penghao Liu & Mingquan Lu (2022)."An Online Multi-Robot SLAM System Based on Lidar/UWB Fusion" IEEE Sensors Journal. https://doi.org/10.1109/JSEN.2021.3136929
11. Feng Li; Wenfeng Chen, Weifeng Xu, Linqing Huang, Dan Li, Shuting Cai, Ming Yang& Xiaoming Xi. "A Mobile Robot Visual SLAM System With Enhanced Semantics Segmentation" IEEE Access. https://doi.org/10.1109/ACCESS.2020.2970238.
12. Yuting Xie, Yachen Zhang, Long Chen, Hui Cheng, Wei Tu, Dongpu Cao & Qingquan Li (2022). "RDC-SLAM: A Real-Time Distributed Cooperative SLAM System Based on 3D LiDAR." IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2021.3132375
13. Z. Meng, Z. Wang, Z. Han & Z. Ma. "Research on SLAM navigation of wheeled mobile robot based on ROS", 2020 5th International Conference on Automation Control and Robotics Engineering (CACRE) (pp. 110-116). 10.1109/CACRE50138.2020.9230186