Automatic Virtual Portals Placement for Efficient VR Navigation

Lili Wang *[1,2,3], Yi Liu[1], Xiaolong Liu[1], and Jian Wu[1]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[2]Peng Cheng Laboratory, Shengzhen, China
[3]Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing, China.

Figure 1: The user study of our methods in the desert village scene. Users can walk to five white spheres in the scene efficiently through our automatic portal placement. (a) is one user's view frame during VR navigation. A mini-map is observed when triggered by the user, on which the portals and trajectory are marked. Circles of the same color represent a pair of portals. The blue arrow indicates the user's current location. The solid red line represents the natural walking trajectory of the user. The red dotted line represents the connection line of the used portal. (b) visualizes the user's final trajectory in the top view map when the user finished the task. Five white small circles in the scene indicate the locations the user needs to reach, and they are not visible on users' mini-map during the experiments. (c) shows the user's current view in (b). The user has just passed a portal, and he/she looks back at that portal. (d) shows the corresponding physical position and moving direction of the user.

Abstract

Portals placement in a large virtual scene can help users improve navigation efficiency, but determining the number and the positions of the portals has some challenges. In this paper, we proposed two automatic virtual portals placement methods for efficient VR navigation. To reduce the number of reverse redirections, we also proposed a real-time portal orientation determination algorithm. For any given single-floor outdoor virtual scene, our methods can automatically place the portals.

Index Terms: Virtual reality—Navigation—Teleportation techniques—Portal placement;

1 Introduction

Navigation and exploration are fundamental interactions in virtual reality (VR). Real walking (RW) is the most natural way to navigate the virtual environment (VE). However, when the physical space that hosts VR applications is much smaller than the virtual environment, it becomes troublesome that users frequently meet the physical boundaries. Redirected walking (RDW) [6] uses translation gain, rotation gain, and curvature gain to scale the transformation of the user's viewpoint. However, if the virtual scene is much larger than the physical scene, the value ranges of these gains become larger so that users can obviously feel that their viewpoint changes do not match their actions, which may cause a certain amount of cybersickness [4]. Teleportation [1] is a more efficient navigation and exploration method in VR, allowing users to jump to locations pointed by a handheld controller. The disadvantage of teleportation is that users have to teleport multiple times to reach an invisible target region. Some applications allow users to teleport by pointing on the scene map, however, it often leads to over-reliance on the handy translations and eventually, breaks presence. What's more, when the user is transferred to the destination, the mismatch between visual effects and acceleration makes users uncomfortable.

Virtual portal is another way to navigate and explore in VR [2, 3, 5, 7]. A pair of portals connect two remote locations. The user walks through the first portal to reach the other end outside the second portal, which improves the presence and performance of VR navigation. When placing the virtual portals in a large virtual scene, two critical factors affect the efficiency of user navigation and exploration. The first is the number of portals that need to be placed, and the second is the positions and orientations of those portals. The previous methods place the portals manually.

In this paper, we proposed two automatic virtual portals placement methods for efficient VR navigation and exploration. For a given scene, we first introduced a visibility importance-based portal placement method (VIPP) to determine the positions and the number of portals. Then we proposed a simulated annealing optimized portal placement method (SAPP) to optimize the positions of the portals generated by the first method. Fig. 1 shows the user study of our methods in the desert village scene.

2 Automatic Portals Placement Method

In this section, we introduce two automatic methods to place the portals in VE. After that, we also propose a real-time portal orientation determination algorithm to decide the orientations of the portals.

2.1 Visibility Importance base Portal Placement method

We calculate positions and number of portals for the scene according to the visibility importance of the scene. We first render the depth map of the scene from the top view orthographically. Then we construct a portal placement weight map ($PPWM$) and the value in PPWM is from 0 to 1 (Fig. 2 a). We use pre-defined depth threshold

*e-mail: wanglily@buaa.edu.cn

to initialize $PPWM$ with 0 and 1, where 0 means unwalkable, 1 means walkable.

A greedy algorithm is proposed to place the virtual portals based on visibility of the scene and the distance of each portal pair. In this greedy algorithm, two maps are constructed and updated. The first map is a scene visibility importance map (VIM) (Fig. 2 b), and it stores the value of the visibility importance from the current walkable position *pos* to the entire scene. If the current position is unwalkable, the value of the visibility importance is set to 0. We place the first portal of a portal pair at the position with the highest visibility importance in VIM for each greedy iteration. Then we update the weights in $PPWM$ (Fig. 2 c) by deceasing the weights of the position *pos* around the first portal *p*. The second map is a visibility and distance importance map ($VDIM$), and it stores the value of the visibility and distance importance from *pos* to the entire scene(Fig. 2 d). We place the second portal in a portal pair at the position with the highest visibility and distance importance in $VDIM$ for each greedy iteration. Then we update the $PPWM$ value of the second portal's visible region. We update VIM and $VDIM$ in the same way to guide the placement of the next portal pair. This process is repeated until the ratio of the area of the visible region in the current $PPWM$ to the area of the visible region in the original $PPWM$ is less than a threshold.

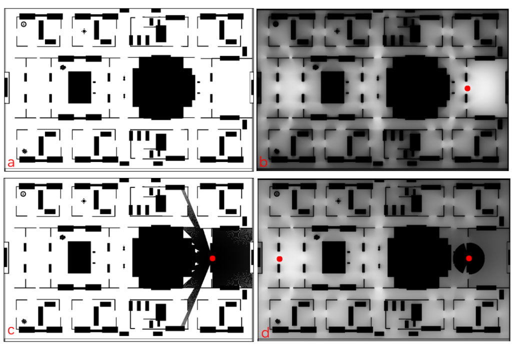

Figure 2: (a) is the initialized portal placement weight map ($PPWM$), (b) is the visibility importance map (VIM), redpoint has the highest importance, (c) is the updated $PPWM$ after the first portal is inserted, (d) is the visibility and distance importance map ($VDIM$), left redpoint has the highest importance.

2.2 Simulated Annealing Optimized Portal Placement Method

In many VR applications, users need to arrive at target positions or pick up target objects in VE. The visibility importance base portal placement method becomes less rational. Instead of visibility, walkability, defined as the average walking path length between any two locations in the scene, becomes important. The shorter the length, the better the walkability of the scene. Therefore, we introduce the second portal placement method to consider the walkability.

We first divide the virtual scene evenly into blocks, and we call each block a sub-region. To improve the walkability between each sub-region of VE, a simulated annealing optimization framework is used to optimize the portals' number and positions according to walkable path of the virtual scene. In order to achieve the low cost solution of the simulated annealing algorithm, we have designed a cost function. This function can measure the total walking distance between any two positions when the portals are placed. The shorter the distance, the better the placement effect.

2.3 Real-time Portal Orientation Determination

To reduce the number of reverse redirections in the navigation after determining the positions and number of the portals, we proposed a real-time portal orientation determination algorithm.

The orientation of the first portal in the pair is set dynamically to face users to help them entering it easily. Then our portal orientation determination method is used to determine the orientation of the export portal, in order to reduce the number of reverse redirections. Our algorithm has two steps. The first step is a pre-process, which initializes the orientations of all portals based on the scene structure with a navigation possibility based method, which makes the portal face the direction the user is most likely to walk. The second step is to estimate the orientation offset for the second portal the user leaves in real time when user navigates.

3 Discussions

In the visibility importance base portal placement method, we design two parameters that allow the user to manually adjust the placement of the portal. The first parameter is the expected ratio of portals' visible area, which could adjust the overall visible area of the portal placement. The second parameter is the weight of distance between portal pairs. The greater the weight, the farther away each pair of portals will be. In our implementation, the expected ratio of portals' visible area is set to 0.7, the weight of distance is set to 0.5.

4 Conclusions

We have proposed two automatic virtual portals placement methods for efficient VR navigation and exploration. We also introduced a real-time portal orientation determination algorithm to naturally guide users to the center of the physical space, reducing the number of reverse path redirections. Our methods can automatically place the portals on the single-floor outdoor virtual scene.

Acknowledgments

This work was supported in part by National Key R&D plan 2019YFC1521102, by the National Natural Science Foundation of China through Projects 61932003 and 61772051, by the Beijing Natural Science Foundation L182016, by the Beijing Program for International S&T Cooperation Project Z191100001619003.

References

[1] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Point amp; teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '16, p. 205–216. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2967934.2968105

[2] G. Bruder, F. Steinicke, and K. H. Hinrichs. Arch-explore: A natural user interface for immersive architectural walkthroughs. In *2009 IEEE Symposium on 3D User Interfaces*, pp. 75–82, 2009. doi: 10.1109/3DUI.2009.4811208

[3] S. Freitag, D. Rausch, and T. Kuhlen. Reorientation in virtual environments using interactive portals. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 119–122, 2014. doi: 10.1109/3DUI.2014.6798852

[4] E. D. Ragan, S. Scerbo, F. Bacim, and D. A. Bowman. Amplified head rotation in virtual reality and the effects on 3d search, training transfer, and spatial orientation. *IEEE transactions on visualization and computer graphics*, 23(8):1880–1895, 2016.

[5] F. Steinicke, G. Bruder, K. Hinrichs, M. Lappe, B. Ries, and V. Interrante. Transitional environments enhance distance perception in immersive virtual reality systems. APGV '09, p. 19–26. Association for Computing Machinery, New York, NY, USA, 2009. doi: 10.1145/1620993.1620998

[6] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):17–27, 2010. doi: 10.1109/TVCG.2009.62

[7] L. Wang, J. Wu, X. Yang, and V. Popescu. Vr exploration assistance through automatic occlusion removal. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2083–2092, 2019. doi: 10.1109/TVCG.2019.2898782