Poster: Immersive Point Cloud Virtual Environments

Gerd Bruder* Frank Steinicke† Andreas Nüchter‡

Department of Computer Science
University of Würzburg

ABSTRACT

Today's three-dimensional (3D) virtual environments (VEs) are usually based on textured polygonal 3D models, which represent the appearance and geometry of the virtual world. However, some application domains require other graphical paradigms, which are currently not adequately addressed by 3D user interfaces. We introduce a novel approach for a technical human-robot telepresence setup that allows a human observer to explore a VE, which is a 3D reconstruction of the real world based on point clouds. Such *point cloud virtual environments (PCVEs)* represent the external environment, and are usually acquired by 3D scanners. We present an application scenario, in which a mobile robot captures 3D scans of a terrestrial environment, which are automatically registered to a coherent PCVE. This virtual 3D reconstruction is displayed in an immersive virtual environment (IVE) in which a user can explore the PCVE. We explain and describe the technical setup, which opens up new vistas of presenting a VE as points rather than a polygonal representation.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1 HUMAN-ROBOT TELEPRESENCE SYSTEM

Our exploration system consists of two sides: (i) the real world in which a mobile robot scans a terrestrial region of interest, whereas (ii) a human observer explores the reconstructed PCVE in a head-mounted display (HMD) environment.

1.1 Real Remote Exploration

Our terrestrial reconstruction is based on a 3D scanning robot that acquires and registers point clouds in a coherent PCVE.

3D Scanning Robot

For the reconstruction of terrestrial scenes we make use of a small, lightweight, battery-powered robotic mobile laser scanning system called the *Intelligent Robot for Mapping Applications in 3D* (Irma3D) [2]. Using a WLAN connection the robot can be controlled remotely via steering commands. In this mode, the robot is controlled using two QuickCam Pro 9000 webcams that are mounted at the front of the chassis. Alternatively, the robot can be run in a fully autonomous mode, in which it attempts to explore its surroundings within preset limits and create a 3D representation of the environment. Therefore, a SICK LMS100 2D laser scanner is mounted at the front of the chassis, acquiring 2D range scans at a rate of 50Hz which are used for collision avoidance and route planning. The Irma3D robot is capable of acquiring 3D scans while moving through an environment or using a *stop-and-go* approach, i. e., the robot remains stationary while a 360 degrees point cloud is acquired. It is equipped with a RIEGL VZ-400 3D laser scanner and a Canon 1000D DSLR color camera (which can be replaced by an Optris Imager PI thermal camera) that are able to freely rotate. The fastest 360 degrees rotation takes 6 seconds. At this speed each point cloud will contain about 750000 points in a field of view of 360×100 degrees. The minimum angular resolution of a range scan is 0.0024 degrees in both directions, which equates to more than 6 billion points per scan.

Figure 1: Illustration of a user exploring an immersive PCVE of the cultural heritage site of Ostia Antica, Italy.

Point Cloud Registration

For precise registration of 3D scans in a coherent PCVE we make use of the 3D Toolkit (3DTK) software library. The library implements among other things the Iterative Closest Point (ICP) algorithm, which is used to iteratively revise the pose difference between two point clouds from subsequently taken 3D scans. Pairwise ICP is implemented to improve pose estimates and SLAM algorithms are used to minimize the extent of introduced registration errors. The motors of the robot are equipped with encoders that measure wheel rotations to provide pose estimates. An additional xSens MTi IMU is used to improve the odometry. This pose information is used as a starting guess for the ICP algorithm. Finnaly, a full 6D SLAM algorithm computes a globally consistent PCVE [1].

With the available 3D point cloud, surface reconstruction methods could be used to generate a triangle mesh approximating the points. This topic has been widely studied in the computer graphics community, and several approaches have been offered for this task. However, most of these algorithms are computationally expensive and potentially introduce errors in spatial representations. Since a fast representation of the reconstructed remote environment is essential for many application scenarios, we directly display the registered "raw data", i. e., the point clouds acquired from the real world, to users as described in the following section.

1.2 Virtual Remote Exploration

After a PCVE has been created, it can be visualized, and a user can explore the VE using an immersive HMD setup.

---

*e-mail: gerd.bruder@uni-wuerzburg.de
†e-mail: frank.seinicke@uni-wuerzburg.de
‡e-mail: nuechter@informatik.uni-wuerzburg.de

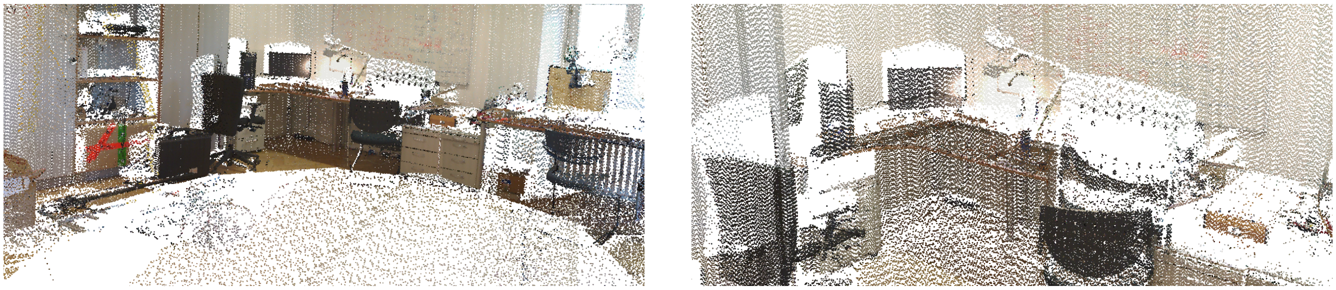

Figure 2: Views of a user walking towards a table in the Automation Lab PCVE (ca. $3.4 \cdot 10^6$ points in 62 scans rendered at about 250 fps).

Hard- and Software Setup

In our setup the PCVE is visualized on an Oculus Rift (Developer Edition) HMD (see Figure 1). The PCVE is rendered using the 3DTK point cloud library. We incorporated the Viargo middleware utility [4] in the native 3DTK rendering code to support natural locomotion via head position and orientation tracking peripherals as well as calibrated rendering on the display device. Using this approach, the Viargo middleware handles the acquisition and processing of tracking data and updates the 6 DoF pose of the rendering cameras via 4×4 matrices in the 3DTK environment.

Our virtual reality (VR) laboratory setup supports natural exploration via real walking with an HMD in a 6m×8m room. We track the user's head position using a WorldViz PPT X4 active optical position tracking system. An InterSense InertiaCube BT inertial tracker is used to track the user's head orientation. For rendering we use an Intel 64-bit computer with 3.5GHz Core i7−2700K processors, 8GB of main memory and Nvidia GeForce 580 GTX graphics card. Using the setup, a user can walk through the reconstructed PCVE. We support exploration of larger areas than the available workspace in the laboratory room via *redirected walking* (cf. [3]).

2 LESSONS LEARNED

To our knowledge, the described human-robot telepresence system is the first application in which point clouds of a remote environment that are scanned by a mobile robot are displayed to a user wearing an HMD. From our preliminary experience with the described setup we observed different challenges.

Performance Point cloud rendering frameworks are usually optimized for desktop environments, i. e., continuous rendering is often not necessary, and renderings can be improved over multiple frames, e. g., by rendering additional points per frame. For our HMD setup we had to introduce continuous rendering in the 3DTK software, which greatly limited such optimization strategies. As we observed for our setup, data sets with ca. 10−20 millions 3D points can efficiently be displayed in real time. While the performance results are discouraging for more dense data sets, the 3DTK software provides the means to set an arbitrary target frame rate, e. g., the 60Hz refresh rate of the Oculus Rift, which then can be enforced. Then, points are culled against a quadtree representation and rendered accordingly until the elapsed frame time exceeds the target frame rate. While this approach ensures optimal frame rates, it results in different numbers of points being displayed each frame.

Sparse Point Clouds When walking through a PCVE with an HMD it becomes apparent that virtual walls, and other objects only appear solid when viewed from a sufficiently large distance which depends on the point sampling density of the object. Such point clouds are usually explored in desktop environments resulting in apparently more dense surfaces. However, walking with an HMD causes objects that appear solid from a distance to "dissolve" when approaching the object. Figure 2 shows an example in which a user walks towards a virtual table that is sampled as a sparse point cloud. Even point clouds of the floor beneath the user's feet appear to dissolve if the user looks down.

Spatial Perception It is a challenging question what effects such sparse point cloud representations have on spatial perception. As discussed above, objects that appear solid from a distance, which arguably support similar distance cues as in polygonal VEs, dissolve when located closer to the user. Spatial perception with VR technologies using polygonal VEs have been extensively considered over the last years, which revealed significant differences in size and distance estimation compared to the real world [3]. So far it is not clear how spatial perception differs in PCVEs. In such situations retinal size, perspective, and occlusion cues can not be leveraged to the same extent to estimate the distance to an object. Moreover, the floating 3D points in space provide conflicting depth cues that humans do not encounter in the real world.

Interaction with the Void When interacting with sparsely represented objects, common implementations of selection, manipulation, and deformation techniques have to be revised. In particular, ray-based selection approaches, which are commonly used in polygonal environments, cannot be used to select objects that are represented as sparse point clouds. Interaction techniques will have to be adapted to take into account the density of points in the user's visual field.

3 CONCLUSION

We introduced a novel technical human-robot telepresence setup, established the context in which this work is conducted in and we discussed some of the limitations currently faced in the field of PCVEs. Some of the existing techniques and 3D user interfaces promise good results, but other approaches need to be improved for efficient handling and presentation of large amounts of point cloud data in real-time VR environments. In the future, we plan to evaluate the perceptual aspects of PCVEs and provide new rendering and interaction techniques. Finally, we will extend our telepresence setup in such a way that the motions of the human observer are mapped to movements of the remote 3D scanning robot.

REFERENCES

[1] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3D mapping with scan matching. *Journal of Robotics and Autonomous Systems (JRAS)*, 56(2):130–142, 2008.

[2] A. Nüchter, J. Elseberg, and D. Borrmann. Irma3D – an intelligent robot for mapping applications. In *Proceedings of the IFAC Symposium on Telematics Applications (TA)*, 6 pages, 2013.

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

[4] D. Valkov, B. Bolte, G. Bruder, and F. Steinicke. VIARGO: A generic VR-based interaction library. In *Proceedings of the Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, pages 23–28, 2012.