Fused AR Experience

David Dunn*
Abdul Rafay Khalid†

Figure 1: *Spring Training 2009, Peoria, AZ.*

Abstract

Traditionally Augmented Reality has taken three main forms each with its own drawbacks. Optical see-through gives us a good view of the real world but it is difficult to show virtual objects effectively as occlusion cannot be achieved. Spatial Augmented reality doesn't allow the user to view the virtual content at the correct depth. The closest to approach to ours is Video See through augmented reality however it suffers from mismatch between the location of eye and the camera center. In recent years there has been a lot of work on accurately reconstructing a 3D representation of the environment using RGBD sensors. We propose a system that combines such a realtime reconstruction method with a Head mounted display to create a wide field of view Augmnented Reality system.

Keywords: KinectFusion, RGBD, Augmented Reality

Concepts: •**Computing methodologies** → **Image manipulation;** *Computational photography;*

1 Introduction

Recent advances in reconstruction methods means that we can create a 3D representation of an environment in real time. The ability of these methods lead us to imagine an alternate form of Augmented Reality. By attaching an RGBD sensor to a VR HMD we can reconstruct the environment around the user as he moves around in his environment. As we have a 3D model of the environment we can render the view from each of the user's eyes. This enables us to accurately reproduce the real environment of the user in stereo. Virtual content can then be composited over the model of the real environment. By using the tracking from the reconstruction method we can align the virtual content with the reconstruction.

*e-mail:spencer@cs.washington.edu
†e-mail:spencer@cs.washington.edu

SIGGRAPH 2016 Posters, July 24-28, 2016, Anaheim, CA
ISBN: 978-1-4503-ABCD-E/16/07
DOI: http://doi.acm.org/10.1145/9999997.9999999

Figure 2: *Ferrari LaFerrari. Image courtesy Flickr user "gfreeman23."*

2 Related Work

3 System Design

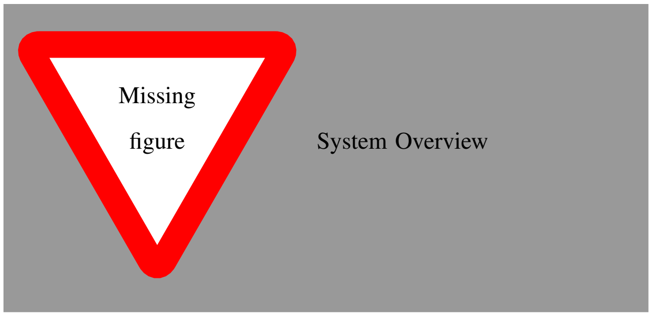
Our setup consists of a Kinect camera rigidly aligned to an Oculus Rift HMD (Figure

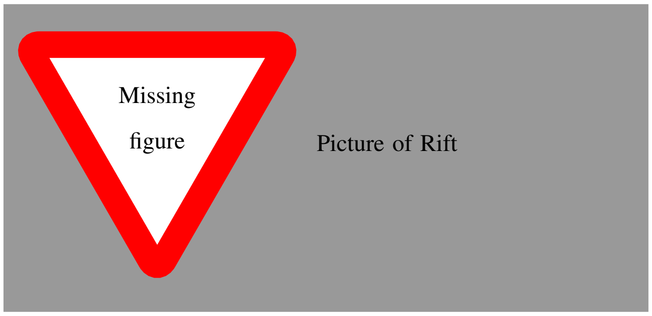
Add picture of rft and kinect

). The kinect acquires RGBD images which as used as input to the KinectFusion algorithm. Using the RGBD images, KinectFusion estimates the camera pose and uses this to integrate the current depthmap into the reconstruction. Virtual content is generated by animating a rigged model of a virtual character. The reconstruction is rendered from the viewpoint of the eyes. The virtual content is then composited onto this. If the virtual content is transformed using the same tracking as the reconstruction then it remains aligned to the environment.The combination of the reconstruction and virtual content is then rendered onto the HMD. Figure

Add system overview figure

shows the System Overview.





3.1 Environment Capture

In order to capture the environment, we rely on the KinectFusion[Newcombe et al. 2011] algorithm as our environment is static and we require the algorithm to be realtime. There are two main implementations of KinectFusion available. The Kinect Development Toolkit for Windows comes with an API to access KinectFusion. They do not provide source code but let the developer access KinectFusion in a limited matter. Also this version has only been demonstrated on small volumes and the tracking is not very stable when using large volumes with small voxel denisty. Given a camera pose and a camera matrix, the API allows the user to render the scene from that camera.
The other popular implementation is the one that comes with PCL. This contains extensions on KinectFusion that enable it to work in large areas. It does so by decomposing the environment into chunks that can be uploaded to the GPU. This allows it to reconstruct a large volume while maintaining a high voxel density.This results in more robust tracking.

3.2 Generate Virtual Content

3.3 Composit Real and Virtual Content

We use the textures from KinectFusion as background and composit it with our virtual content.

3.4 Tracking

KinectFusion uses the current depth image and tries to align it with the reconstruction to find out the camera pose. This camera pose is used to track the head-mount with respect to the world. The same transform needs to be applied to the virtual content to keep it fixed with respect to the real world. For our initial experiments however, the rotational tracking from the Oculus was used to align the virtual objects. This can be shifted without too much effort to use the tracking from KinectFusion ensuring that the objects stay aligned to the real world.

3.5 Display on HMD

To render on the HMD, we draw our content on the framebuffer provided by Oculus Rift SDK for each eye. In order to ensure that the content looks correct through the oculus, we provide the desired FOV and texture size expected by the Rift to the KinectFusion API. Some further calibration was performed to ensure that objects in the Oculus appeared the same size as the real world. In our setup, the Kinect is placed above the Rift and thus the perspective from which its looks at the environment is different than the eye. We transform the Kinect to the pose of the centre of the Rift. Eye Position translations are then applied to get the poses for the eyes. The environment is rendered from each of the poses and passed to the Rift. The additional virtual content is rendered on the same buffers. The rift applies lens distortion correction and displays it.

Acknowledgements

To Robert, for all the bagels.

References

AGARWAL, S., MIERLE, K., AND OTHERS. Ceres solver. https://code.google.com/p/ceres-solver/.

ANONYMOUS, 1976. Planes of the head. http://www.planesofthehead.com/.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 15–22.

JOBSON, D. J., RAHMAN, Z., AND WOODELL, G. A. 1995. Retinex image processing: Improved fidelity to direct visual observation. In *Proceedings of the IS&T Fourth Color Imaging Conference: Color Science, Systems, and Applications*, vol. 4, The Society for Imaging Science and Technology, 124–125.

KARTCH, D. 2000. *Efficient Rendering and Compression for Full-Parallax Computer-Generated Holographic Stereograms*. PhD thesis, Cornell University.

LANDIS, H., 2002. Global illumination in production. ACM SIGGRAPH 2002 Course #16 Notes, July.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital michelangelo project. In *Proceedings of SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, New York, K. Akeley, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 131–144.

NEWCOMBE, R. A., IZADI, S., HILLIGES, O., MOLYNEAUX, D., KIM, D., DAVISON, A. J., KOHI, P., SHOTTON, J., HODGES, S., AND FITZGIBBON, A. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, IEEE, 127–136.

PARK, S. W., LINSEN, L., KREYLOS, O., OWENS, J. D., AND HAMANN, B. 2006. Discrete sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics 12*, 2 (Mar./Apr.), 243–253.

PARKE, F. I., AND WATERS, K. 1996. *Computer Facial Animation*. A. K. Peters.

PELLACINI, F., VIDIMČE, K., LEFOHN, A., MOHR, A., LEONE, M., AND WARREN, J. 2005. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Transactions on Graphics 24*, 3 (Aug.), 464–470.

SAKO, Y., AND FUJIMURA, K. 2000. Shape similarity by homotropic deformation. *The Visual Computer 16*, 1, 47–61.

YEE, Y. L. H. 2000. *Spatiotemporal sensistivity and visual attention for efficient rendering of dynamic environments*. Master's thesis, Cornell University.