Rapid Creation of Photorealistic Virtual Reality Content with Consumer Depth Cameras

Chih-Fan Chen, Mark Bolas, and Evan Suma Rosenberg*

USC Institute for Creative Technologies, Los Angeles CA, USA

ABSTRACT

Virtual objects are essential for building environments in virtual reality (VR) applications. However, creating photorealistic 3D models is not easy, and handcrafting the detailed 3D model from a real object can be time and labor intensive. An alternative way is to build a structured camera array such as a light-stage to reconstruct the model from a real object. However, these technologies are very expensive and not practical for most users. In this work, we demonstrate a complete end-to-end pipeline for the capture, processing, and rendering of view-dependent 3D models in virtual reality from a single consumer-grade RGB-D camera. The geometry model and the camera trajectories are automatically reconstructed from a RGB-D image sequence captured offline. Based on the HMD position, selected images are used for real-time model rendering. The result of this pipeline is a 3D mesh with view-dependent textures suitable for real-time rendering in virtual reality. Specular reflections and light-burst effects are especially noticeable when users view the objects from different perspectives in a head-tracked environment.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.3.7 [Computing Methodologies]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture I.2.10 [Computing Methodologies]: Vision and Scene Understanding—3D/stereo scene analysis

1 INTRODUCTION

With the recent proliferation of high-fidelity head-mounted displays (HMDs), there is increasing demand for realistic 3D content that can be integrated into virtual reality environments. However, creating photorealistic models is not only difficult but also time consuming. A simpler alternative involves scanning objects in the real world and rendering their digitized counterpart in the virtual world. Capturing objects can be achieved by performing a 3D scan using widely available consumer-grade RGB-D cameras. This process involves reconstructing the geometric model from depth images generated using a structured light or time-of-flight sensor (Figure. 1 (a)). The colormap is determined by fusing data from multiple color images captured during the scan. Existing methods compute the color of each vertex by averaging the colors from all these images. Blending colors in this manner creates low-fidelity models that appear blurry. (Figure 1 (b)). Furthermore, this approach also yields textures with fixed lighting that is baked on the model. This limitation becomes more apparent when viewed in head-tracked virtual reality, as the illumination (e.g. specular reflections) does not change appropriately based on the user's viewpoint.

To improve color fidelity, techniques such as View-Dependent Texture Mapping (VDTM) have been introduced [2, 4]. In this approach, the system finds observed camera poses closest to the view point and use the corresponding color images to texture the model. Previous work has used Structure-from-Motion and Stereo Matching to automatically generate the model and the camera trajectory. Although these methods typically result in higher color fidelity, the reconstructed geometric model is often less detailed and more prone to error than depth-based approaches. In this work, we leverage the strengths of both methods to create a novel view-dependent rendering pipeline (Figure 2). In our method, the 3D model is reconstructed from the depth stream using KinectFusion. The camera trajectory computed during reconstruction is then refined using the images from the color camera to improve photometric coherence. The color of 3D model is then determined at runtime using a subset of color images that best match the viewpoint of the observing user (Figure 1 (c)).

*e-mail:{cfchen, bolas, suma}@ict.usc.edu

2017 IEEE Virtual Reality (VR)
March 18-22, 2017, Los Angeles, CA, USA
978-1-5090-6647-6/17/$31.00 © 2017 IEEE

2 SYSTEM OVERVIEW

In this work, we proposed a system pipeline (Figure 2) to generate a view-dependent 3D model from a consumer-grade RGB-D sensor through an offline stage and an online stage.

2.1 Offline Stage

- **3D Reconstruction** The depth information comes directly from RGB-D sensors, so visual odometry techniques such as SfM and stereo matching are not needed to generate models. Instead, the Kinectfusion system [3] is used to reconstruct a single 3D model. Note that the color images are captured with fixed exposure and white balancing.

- **Camera Trajectory Optimization** The camera trajectories are not sufficiently accurate for texture mapping because it is purely based on geometry information. This is particularly noticeable at the boundary of objects. To maximize the color and geometry agreement, we apply Color Mapping Optimization [5] to yield more accurate camera poses.

2.2 Online Stage

- **Render Image Selection** We sample images based on the euclidean distance of user's head position and all the camera positions in our database. The head position and orientation is obtained from the HMD tracker. Using only the closest image to render the model will create a sudden transition from one image to another. It also produces a sharp edge between updated vertices and the others. Selecting more images can achieve smooth transitions with head movement. However, it will lose details such as specular reflections and light-bursts (e.g., the fixed-texture model is colored by all images). In our experiment, we select three images to not only preserve the detail but also smoothly switch from different viewpoints.

- **Real-time Texture Rendering** Each vertex is mapped to image planes to retrieve their corresponding RGB values. We compute the vector from the model center to the HMD position and use the barycentric coordinates to compute the weight to combine the color from images. Note that before combining these values, we must perform a visibility check to detect occlusion. RGB values that fail the visibility check are

(a) Reconstructed 3D Model (b) Fixed Texture (c) View-Dependent Textures

Figure 1: (a) A 3D model reconstructed from images captured with a single consumer depth camera. (b) The textured model generated using the traditional approach of blending color images. (c) Our approach is capable of rendering view-dependent textures in real-time based on the user's head position, thereby providing finer texture detail and better replicating illumination changes and specular reflections that become especially noticeable when observed from different viewpoints in virtual reality.

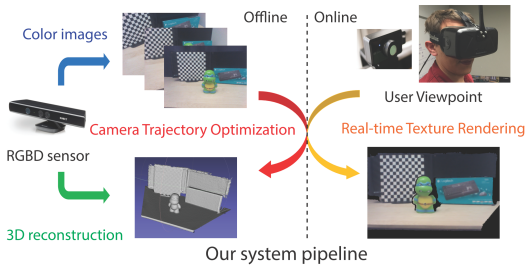

Figure 2: An overview of the complete pipeline, consisting of three phases: (1) object capture, (2) offline processing/optimization, and real-time view-dependent texture rendering.

then discarded (i.e., set the weight to zero). The remaining weights are normalized and the vertex color is updated by the new RGB value.

3 Demo Experience

The demo is a seated virtual reality experience using an Oculus Rift DK2 head-mounted display and position tracking camera (Figure 3). Users will be able to view a gallery of virtual objects that were scanned offline using consumer depth cameras (Microsoft Kinect and Intel RealSense) and subsequently processed through our pipeline. In addition to objects that were captured directly by us, we will also include objects that were reconstructed from the dataset provided by Choi et al. [1], which contains thousands of RGB-D sequences captured by non-experts in computer vision. The demo environment is implemented in Unity.

For each object, users will be able to freely toggle between the dynamic view-dependent textures and a single fixed texture generated from blending the color images (e.g. KinectFusion). Taking advantage of both the 6DOF head tracking and gamepad controls to view the objects from different perspectives, users will therefore be able to directly compare the results of our pipeline with the typical approach used in game engines.

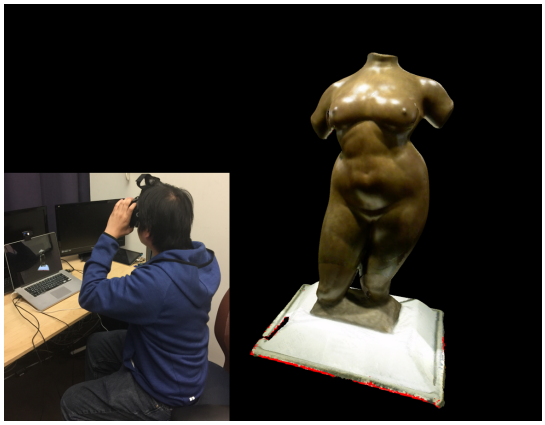

Figure 3: An example of the virtual reality experience for viewing the scanned 3D objects rendered with view-dependent textures.

Acknowledgements

This work is sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005. Statements and opinions expressed and content included do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016.

[2] P. Debevec, Y. Yu, and G. Boshokov. Efficient view-dependent image-based rendering with projective texture-mapping. Technical report, University of California at Berkeley, Berkeley, CA, USA, 1998.

[3] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.

[4] Y. Nakashima, Y. Uno, N. Kawai, T. Sato, and N. Yokoya. Ar image generation using view-dependent geometry modification and texture mapping. *Virtual Reality*, 19(2):83–94, 2015.

[5] Q.-Y. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Trans. Graph.*, 33(4):155:1–155:10, July 2014.