Real-time 3D rendering using depth-based geometry reconstruction and view-dependent texture mapping

Chih-Fan Chen, Mark Bolas, and Evan Suma*
USC Institute for Creative Technologies

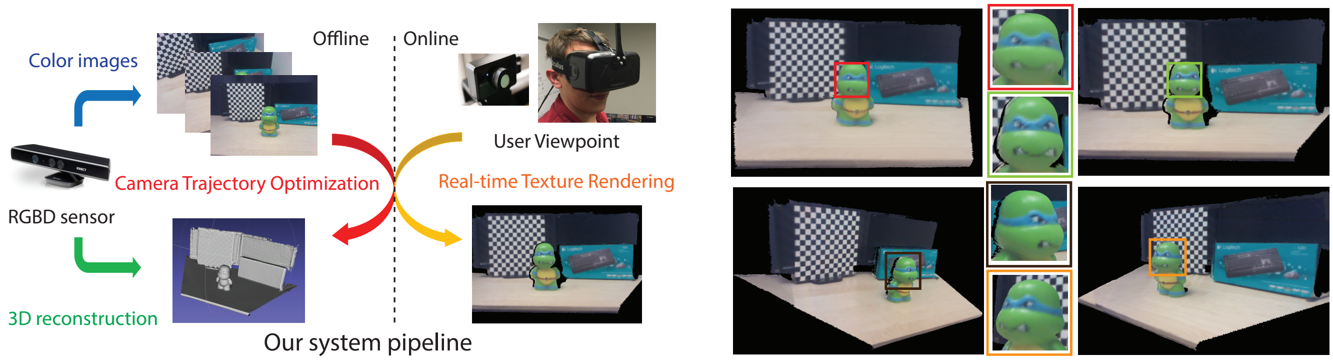


Figure 1: *(left) An overview of the Real-Time View-Dependent 3D Rendering pipeline. (Right) Comparison of our approach (green) with KinectFusion color blending (red). Our approach provides finer texture detail and better replicates illumination changes and specular reflections that become noticeable viewing the object from different viewpoints.*

Keywords: 3D reconstruction, view-dependent texture mapping, real-time rendering

Concepts: •**Computing methodologies** → **Image-based rendering;** *Virtual reality; Appearance and texture representations;*

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. 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 right). 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.

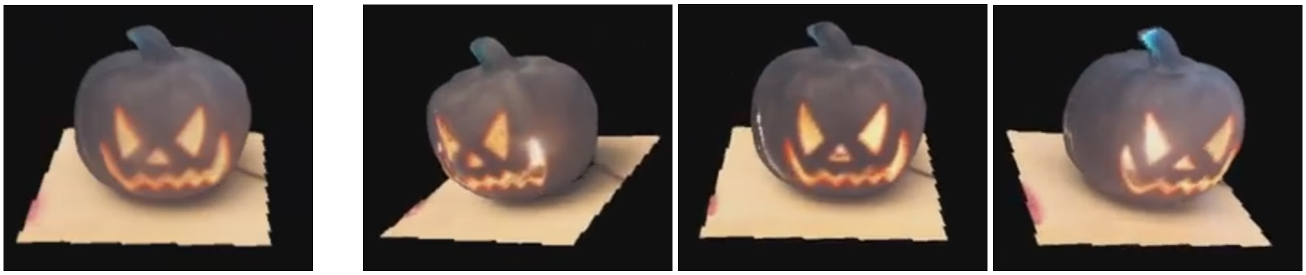
To improve color fidelity, techniques such as View-Dependent Texture Mapping have been introduced [Nakashima et al. 2015]. In this approach, the system finds observed camera poses closest to the view point and use the corresponding color images to texture the model. Previous work has used Structure-from-Motion 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 1 left). 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.

2 Our Approach

- **Input Data Capture** We use a Microsoft Kinect, which streams VGA resolution depth and color images at 30 frames per second. The color images are captured with fixed exposure and white balancing.
- **3D Reconstruction and Camera Trajectory Optimization** The geometric model is generated using KinectFusion. However, the camera trajectories are purely based on geometry information, which is not sufficiently accurate to produce a correct color mapping. This is particularly noticeable at the boundary of objects. To maximize the color and geometry agreement, first we calibrate and align the color and IR camera. We then apply Color Mapping Optimization to yield more accurate camera poses [Zhou and Koltun 2014].
- **Real-time Texture Rendering** To create a colormap that smoothly transitions with head movement, we sample images from multiple camera poses based on the HMD position and camera trajectory. Each vertex is then mapped to image planes to retrieve their corresponding RGB values. Before

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

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4371-8/16/07
DOI: http://dx.doi.org/10.1145/2945078.2945162



(a) Fixed Texture (b) View-Dependent Texture

Figure 2: *(a) A 3D model textured using KinectFusion color blending. The texture appears the same from every viewpoint. (b) The output of our view-dependent texture pipeline from three different camera locations.*

combining these values, we must perform a visibility check to detect occlusion. This is done by using a pre-computed depth map obtained by projecting the 3D model to color camera space. RGB values that fail the visibility check are then discarded. The remaining values are then weighted based on their barycentric coordinates.

- **Results** The result of this rendering pipeline can be seen in Figures 1 and 2. Note that the position of specular reflection changes varies by viewpoint. Furthermore, per-vertex rendering is parallelized in our design by passing the depth map to the shader for the visibility check. As a result, our method can render an entire model in only 10-20 milliseconds, which is sufficient for real-time high frequency rendering.

References

NAKASHIMA, Y., UNO, Y., KAWAI, N., SATO, T., AND YOKOYA, N. 2015. Ar image generation using view-dependent geometry modification and texture mapping. *Virtual Reality 19*, 2, 83–94.

ZHOU, Q.-Y., AND KOLTUN, V. 2014. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Trans. Graph. 33*, 4 (July), 155:1–155:10.