

Real-time object scanning and manipulation in the CAVE

Islam Akef Ebeid
Emerging Analytics Center
Department of Information
Science
University of Arkansas at Little
Rock

iaebeid@ualr.edu



Figure 1. A realtime render of the generated mesh in the CAVE using VR-Juggler (Personally Taken)



Figure 2. The object being scanned using the RealSense RGBD camera

ABSTRACT

Since the introduction of the CAVE in 1993 [1], there hasn't been much work done on rendering scanned small objects in real time. I developed a real time application that segments, identifies, captures and scans a small object and renders interactively inside the CAVE using a cluster window management system (VR-Juggler). The application allows scanning small objects on a custom made turntable using a RealSense [4] camera creating a custom defined binary file that loads and renders instantly inside the CAVE. The developed application is a point on the way of improving and solving the problems of reconstruction of small objects in real-time.

Keywords

Virtual Reality, Real-time Graphics

1. INTRODUCTION

A lot of work has been done in the area of 3D scanning through high resolution depth cameras specially generating meshes from objects being recognized as iso-surfaces through using the famous Marching Cube algorithm or other algorithms that has been adopted in the recent years by projects like MeshLab. Recently an idea has been introduced to capture scanned objects based on light field rendering through creating

billboard rendered images inside the VR environment in real-time [3], this method is computationally cheap and efficient however due to the lack of resources it will be hard to implement in the current project. This work does not intend to explore the problems in the field of 3D scanning but rather to explore and introduce other ways to capture objects and render them in real-time inside cluster based VR environments like the CAVE.

Thanks to Intel's Real Sense [4] provided SDK it is made easy to scan a relatively small object and generating an OBJ file that can be parsed and rendered using OpenGL. Integrating this real-time render inside the CAVE using a C++ library called VR-Juggler [2] which provides a stereo synchronized windowing system that operates on cluster based systems like the CAVE which usually run on quad buffered graphics cards giving a smooth and consistent frame rate and viewport for stereo images.

In my proposal I mentioned that Leap Motion will be introduced in the CAVE, unfortunately I ran out of time to be to develop a functional way to integrate a Leap Motion controller.

My pipeline starts with placing an object on a turntable that is positioned adjacent to the CAVE and by running the program a message will appear

directing the user inside the CAVE that an object is currently being scanned, the object will appear instantly inside the environment where a Leap Motion sensor is introduced inside the CAVE, then user will be able to manipulate the object in the virtual space.



Figure 3. Operating Scenario



Figure 4. Program's Pipeline

2. DESIGN & CHALLENGES

The main efforts were all invested in getting the scanning pipeline to work. The RealSense SDK is not as intuitive as it has to be though they provide several implemented scanning algorithms.

I started by configuring the provided SDK, the SDK version changed several times, this gave me a lot of problems.

2.1 Real Sense Camera

2.1.1 Raw Stream

Acquiring raw streams from the Real Sense Sensor provided the SDK is relatively an easy task.

2.1.2 Segmentation

Applying segmentation algorithms to each rendered frame of the raw streams is made easy also with the Real Sense SDK.

2.1.3 Object Detection

Detection comes after the segmentation process which segments the desired pixels based on colors then detecting the position of the object and the boundaries by rendering a green box around the detected object

2.1.4 Scanning

The scanning starts when by acquiring an image frame and a depth frame aligning them together and producing the vertex data.

2.1.5 Reconstruction

The RealSense SDK provides a black box function that can be called after generating all the image information and applying it to the vertex data stored in the scanner object.

2.2 Rendering in the CAVE

2.2.1 Serializing

One major problem that I faced was the loading time of the mesh file, since I am loading the file within the rendering loop, so I had to serialize the mesh file into my own format to speed up the loading time. I experimented with several ways including GPU accelerated triangulation and other

ways to reduce the amount of time required for loading big meshes.

2.2.2 Problems

Perhaps the main obstacle that I faced was the absence of a USB3 port in the CAVE system since the RealSense sensor requires a USB 3 port so I couldn't implement an actual real-time system that provides an almost zero loading time.

Other problems including updating the RealSense SDK with major changes makes hard to keep up with the new implementations.

3. KNOWLEDGE ACQUIRED

This project allowed to learn a whole set of skills, including using opengl to load a mesh file and applying materials and drawing bounding boxes, applying shaders and reducing the time to load a mesh.

Also learning VR-Juggler is a big advantage since it provides a generic framework to develop cluster based VR- Application reducing the overhead of integrating VR-Technology and rendering.

4. FUTURE WORK

Incorporating Leap Motion in the CAVE hasn't been explored before.

5. REFERENCES

[1] CRUZ-NEIRA, C., SANDIN, D. J., & DEFANTI, T. A. (1993). Surround- screen

projection-based virtual reality: the design and implementation of the CAVE. SIGGRAPH 1993 Proceedings.

- [2] CRUZ-NEIRA, C., Morillo, P., Hartling, P., Bierbaum, A. implementing immersive clustering with VR juggler. ICCSA'05 Proceedings of the 2005 international conference on Computational Science and Its Applications - Volume Part III Pages 1119-1128.
- [3] Mark Bolas*, Ashok Kuruvilla, Shravani Chintalapudi, Fernando Rabelo, Vangelis Lympouridis, Christine Barron, Evan Suma, Catalina Matamoros, Cristina Brous, Alicja Jasina, Yawen Zheng, Creating Near-Field VR Using Stop Motion Characters and a Touch of Light-Field Rendering. SIGGRAPH 2015 Poster Proceedings.
- [4] Intel Real Sense Technology.
<http://www.intel.com/content/www/us/en/arc/hitecture-and-technology/realsense-depth-camera.html>
- [5] Leap Motion Sensor.
<https://www.leapmotion.com/>
- [6] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzugli, MeshLab: an Open-Source Mesh Processing Tool. Eurographics Italian Chapter Conference (2008)