

Augmented Reality
Preparation and Framing Document Technical Review 2

Background and Context

Our project is a method of viewing conceptual objects in a 3D space using virtual reality (VR). The program (on the computer) should be able to track a post-it layout and project a model over the post-it. The layout is four post-its in square formation with the main corner having a different marker.

We are using Google Cardboard as our source of VR and obtaining our image from a Neato Raspberry Pi camera that has an image-splitting attachment. Once we read images, we augment the image in python and then output it into Google Cardboard-viewable form.

Currently, we have developed tracking of a post-it note using openCV. The program masks the image to obtain the centers of each of the post-its. It can then use a different mask to identify the main corner. The program then labels each corner's orientation using angle calculation and smart use of quadrants. Given these locations, the user then calibrates the camera to the image by changing the view of the image. This calibration enables the program to know the proper orientation and position of the model in relation to the camera. We can use this information to project simple 3D objects onto our post-its.

Since the last technical review, we have also purchased the app *Duet Display* which enables us to use a cabled connection between the camera and computer for faster display data transfer. We will be running a virtual machine of Ubuntu in Windows since the app only runs with the Windows operating system.

Using OpenGL, we have been able to project a 3D cube onto the frame of the camera's view. In order to draw onto an OpenCV camera frame, we have to convert each pixel on the camera's frame into OpenGL format. After doing this we are able to draw objects using OpenGL functions on the frame. One of the problems we are running into is working with both 2D shapes and 3D shapes in the same screen. The camera frame is a 2D OpenGL texture whereas the cube we are displaying is a 3D OpenGL shape. OpenGL 3D functions such as `gluPerspective()` do not allow us to render 3D shapes in a 2D defined system.

From now, we plan on projecting an OpenGL generated cube onto the interface and importing/reading STL files in OpenGL.

Key Questions

- How do you use both `gluPerspective()` and `gluOrtho2D()` to draw a 3D cube on a 2D texture?
- How do you render an STL file in OpenGL
- How do you combine OpenGL and OpenCV images
- Are there any alternatives besides using a virtual machine for our project?

Agenda for Technical Review Session

1 minute - Talk about what we hope to get out of it, general structure

- We generally understand how to do our project but would like advice on how to improve it and different ways to tackle the problems

2 minutes - Overview of project / presentation

- Talk about general background, as in what the project is and what we're doing, how we've been doing it

10 minutes - Demo of calibration and cube projection

- Feedback

10 minutes - Demo of OpenGL Cube

- Feedback, questions, discussion on combining OpenGL and OpenCV

2 minutes - Closing comments