

## Tour Into Picture

### Introduction

Since I haven't had much experience dealing with 3D objects in programming, I had really difficult time to understand the concept of switching between different perspectives (spaces). So it took a while for me to actually get started in programming. Initially, I started with C++ up to selecting background wall and vanishing point, but I switched to Matlab because there was sample codes provided at the class canvas and Matlab allowed to me to debug easier and more efficiently with all variables and their values displayed.

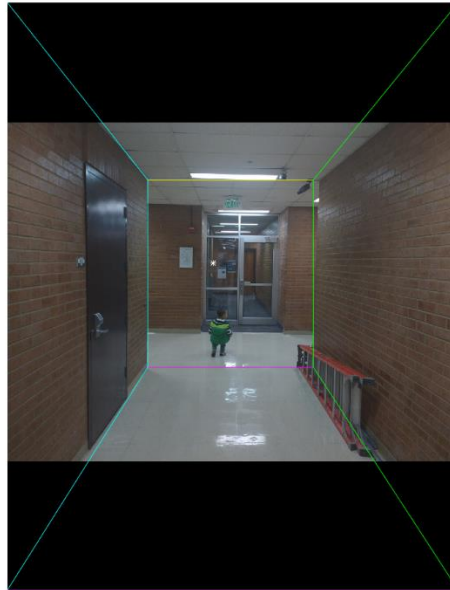
The below figures show the results of dividing 5 faces of a box with vanishing point.  
byu\_expanded.png



jerome\_expanded.png



woojoo\_expanded.png



## Procedures

For the Graphcut part, I had to use GIMP image editor to cut out the foreground objects because my graphcut program didn't give me the results I wanted. The reason was that my graphcut algorithm was not iterative. Anyway, I segmented the foreground object out to make a separate image with only foreground object. In my Matlab program, I calculated the correct depth in 3D plane to place the object in my 3D box by using the similar triangle equation with the bottom coordinate of the object in image plane. I also had to make the background transparent before I placed it in the box.

I set the focal length to be the same length as the width of the original image and then was able to get the maximum depth of the 3D box. I also made my box width and height to be the same size as my expanded image. From here, I was able to switch coordinates between 3D space and image space since I already know the dimension of the box at this point. I also realized that each size of the box has one fixed value for one plane. For example, my back wall is at the maximum depth of the box and top surface is at the maximum height of the box. For each surface, I operated double For loops to find where is the exact point in the image plane when a point is selected in the 3D plane. Once I got this exact point in the image plane, I bilinearly interpolated to get a appropriate pixel value to fill a rectangle image corresponding the quadrilateral. After this point, five rectangle images representing five surfaces are generated.

Here are my foreground pictures and each side of the boxes  
byu\_faces.png

**foreground object**



**back**



**top**



**bottom**



**left**



**right**



jerome\_faces.png

foreground object



back



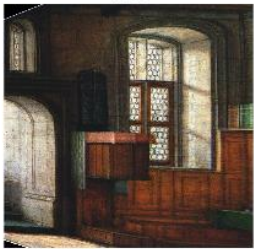
top



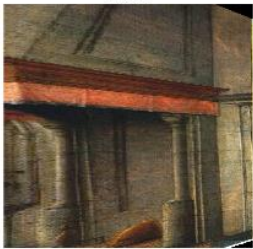
bottom



left



right



woojoo\_faces.png

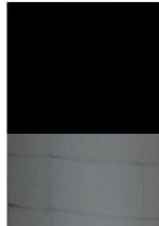
foreground object



back



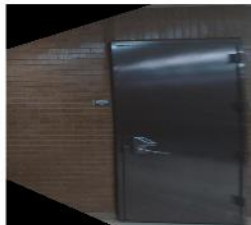
top



bottom



left



right



## Result

Since I programmed in Matlab, I couldn't use the provided OpenGL C++ code to display 3D box. However, this was not too difficult since there was a way to do this using correct built-in functions. I also had to deal with making the background of my foreground image transparent so that I could see the back wall. This was also matter of finding correct built-in functions in Matlab. Anyway, after long hours of puzzling and programming, this project is finished.

Some box images  
byu\_box.png



jerome\_box.png

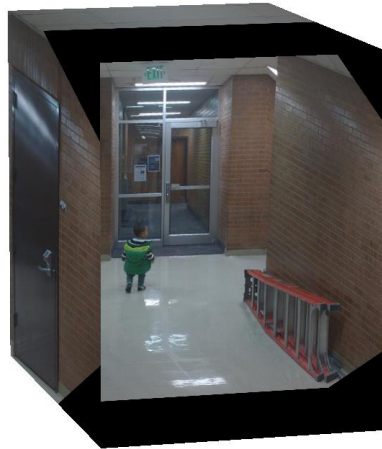




woojoo\_box1.png – the foreground is misplaced because Woojoo(the baby) was included in the back wall.



woojoo\_box2.png



## Conclusion

Even if it was difficult to get to know what I need to program, once I started I was able to do it. This is the only class lab that I did in Matlab and I am glad that I did in Matlab because otherwise I am probably still spending my time to do this lab. Although I had to ask many questions about the concept and how to use some Matlab functions to finish this project, it was worth to learn and it is cool to see the result of my program.