Panoramas & Stereo

Nooreldean Koteb and Abdallah Soliman

I. Introduction

This project consisted of various parts centered around spherical re-projection, triangulation, and stereo patch matching. In the first part we apply spherical re-projection to images generated in the computer program blender. We then use spherically re-projected images to create a panorama image. In the triangulation section points were found on an image and there was an attempt to find the length using triangulation. In the final section stereo patch matching was used to create depth maps of images.

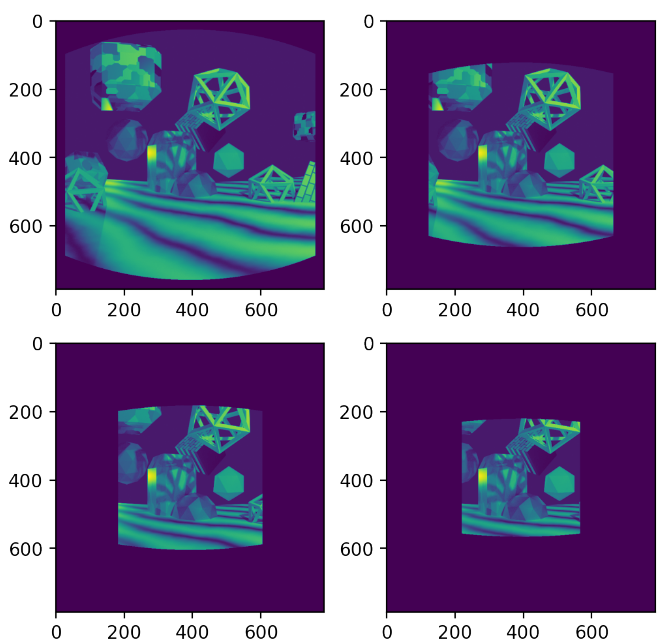
II. Spherical Re-projection

For this section the formulas below were used to spherically re-project 4 images with different focal lengths between 20mm and 50mm.

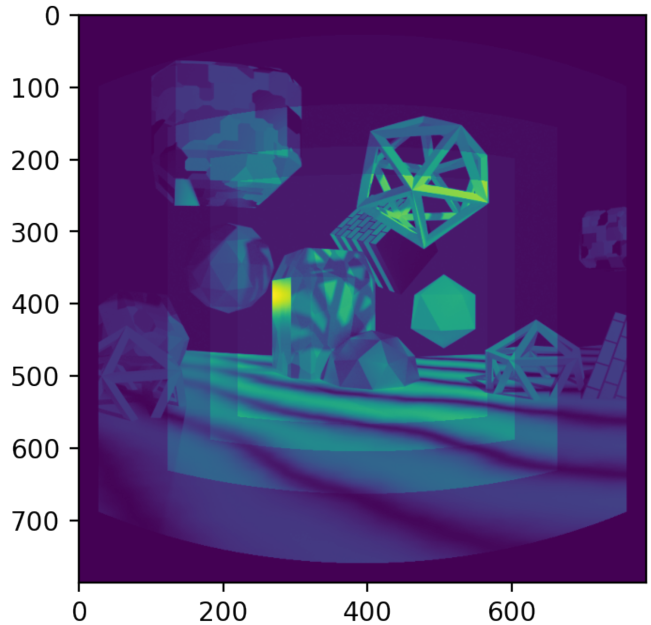
$$x_t = \sin\theta * \cos\phi$$
$$y_t = \sin\phi$$
$$z_t = \cos\theta * \cos\phi$$

In order to project onto the image plane the following was added to the new_x and new_y.

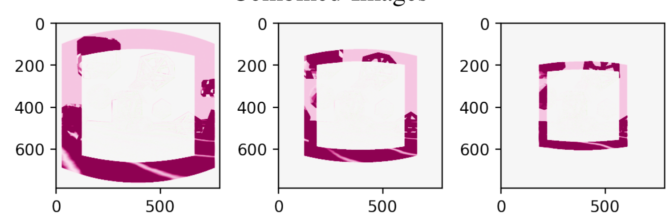
$$new\_x = len(x)//2 + \frac{x_t}{z_t} * focal\_length\_px$$
$$new\_y = len(x)//2 + \frac{y_t}{z_t} * focal\_length\_px$$



Spherical Re-projection



Combined Images

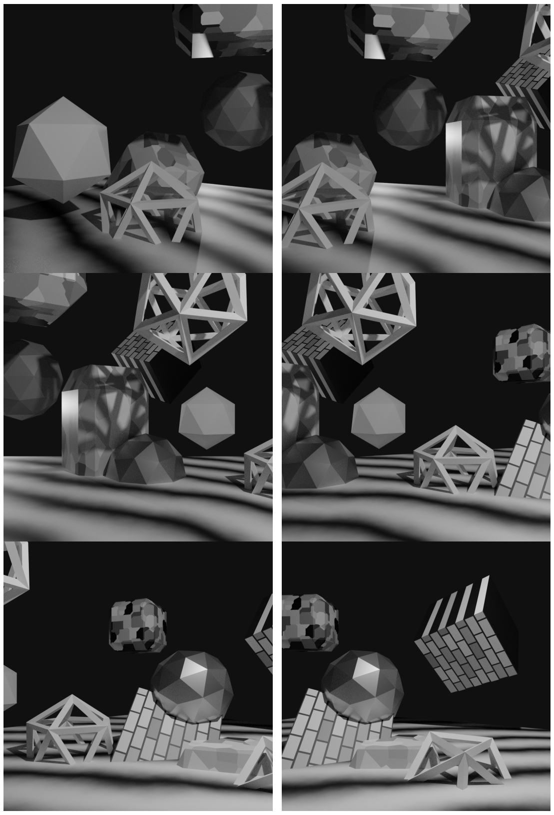


Change

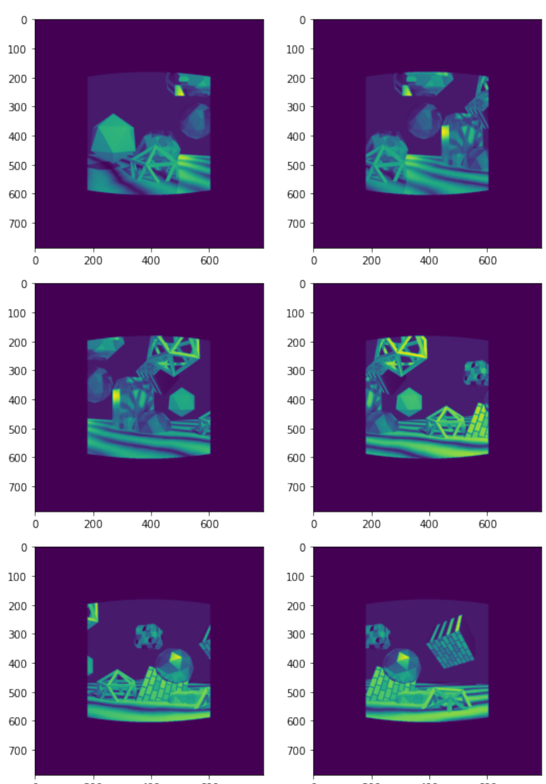
III. Panorama Stitching

For this section images were rendered from the given blender file from position $-40°$ to $60°$ using $20°$ increments. This resulted in 6 image files which were used in the panorama stitching. The spherical re-projection was then calculated and applied for all images. These images were then sent into a SIFT function that resulted in transformation matrices for the panorama stitching. A new empty image with the size of the new panorama was then created and images were padded into their correct location. The images were then combined together with the warpPrespective() function from openCV and blended together with the numpy maximum() function.

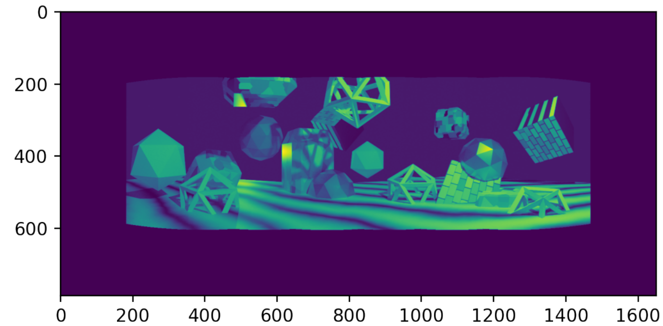
For the 20mm+Panorama image the spherically re-projected 20mm image from the previous section was taken and blended with the stitched panorama image that was created in this section. Since both images were spherically re projected they lined up perfectly even though the focal lengths differed.



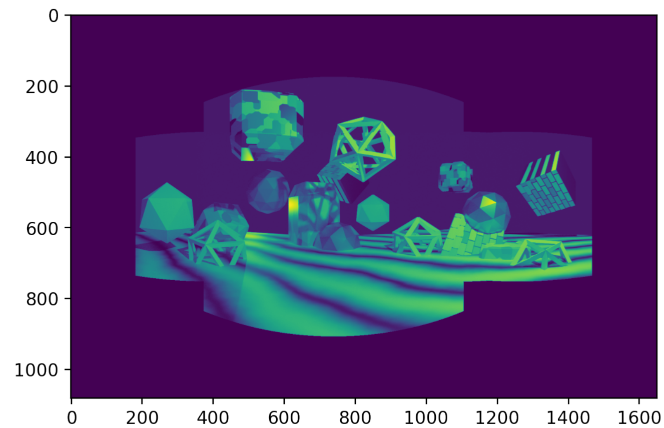
Blender Generated Images



Spherically Re-projected Images



Stitched Panorama Image



20 mm + Panorama

IV. TRIANGULATION

A. Projection Into Image Space

QUESTION What are the camera matrices P_a and P_b?

For this section two camera matrices were calculated and a 3D point was projected onto an image. The camera matrices were calculated using t_a, t_b, and f.

$$\mathrm{t}_a = \begin{bmatrix} 0 \\ -0.2 \\ 5 \end{bmatrix} \qquad t_b = \begin{bmatrix} -1.5 \\ 0 \\ 5 \end{bmatrix}$$

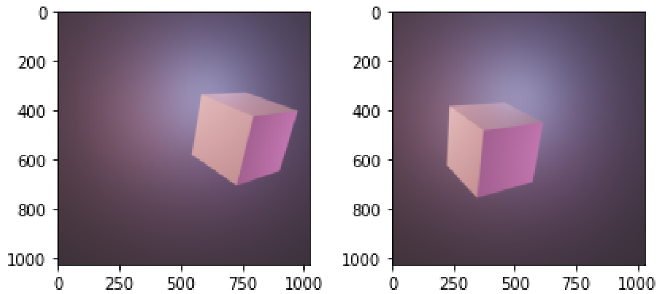
$$f = 1137.8$$

$$\mathrm{P}= \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & y_z \end{bmatrix}$$

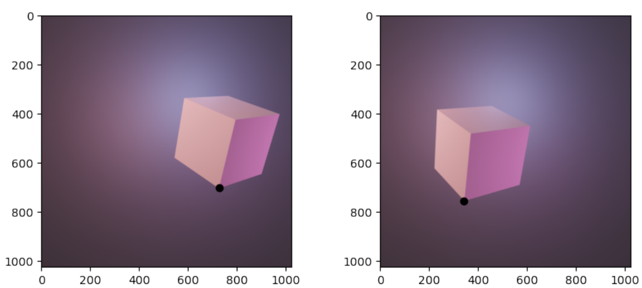
$$FinalMatrices:$$

$$\mathrm{P}_a = \begin{bmatrix} 1170.30 & 0.00 & -60.00 & -300.00 \\ 0.00 & 1170.30 & -40.00 & -434.06 \\ 0.00 & 0.00 & 1.00 & 5.00 \end{bmatrix}$$

$$P_b = \begin{bmatrix} 1170.30 & 0.00 & 245.00 & -530.45 \\ 0.00 & 1170.30 & -80.00 & -400.00 \\ 0.00 & 0.00 & 1.00 & 5.00 \end{bmatrix}$$



Original Images



Projecting on image spaces

B. Determining the Size of the Cube

TASK Pick two corners of the cube and include the (x, y) image coordinates for both image_a and image_b and the 3D world coordinate (X, Y, Z) in your write up.

(540, 580) and (220, 620) were the two points picked for this section, however for some reason my code won't align properly.

QUESTION What is the side length of the cube shown in the two images above? Although my code did not work, my partner was able to get it to run and the distance he got was 1.268551.

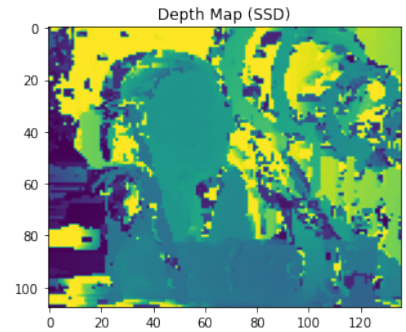
V. STEREO PATCH MATCHING

QUESTION The possible feature matches in the second image are along the epipolar line. Since the images are properly rectified, what is the epipolar line in the second image corresponding to coordinate (x_a, y_a) in the first image?

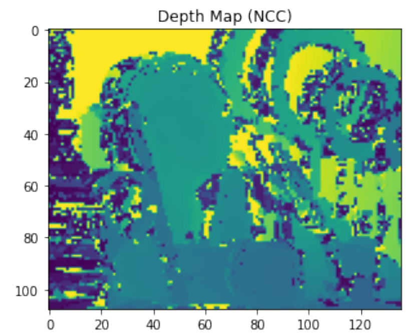
The epipolar line is the horizontal line y_a which consists of x_a values within the range patch_half_width and image width - patch_half_width.

QUESTION The left few columns of both depth maps is quite noisy and inaccurate. Give an explanation for why this is the case?

The left few columns' original image had a left shift that resulted in less features being matched. This resulted in less accurate depth maps.



Depth SSD



Depth NCC