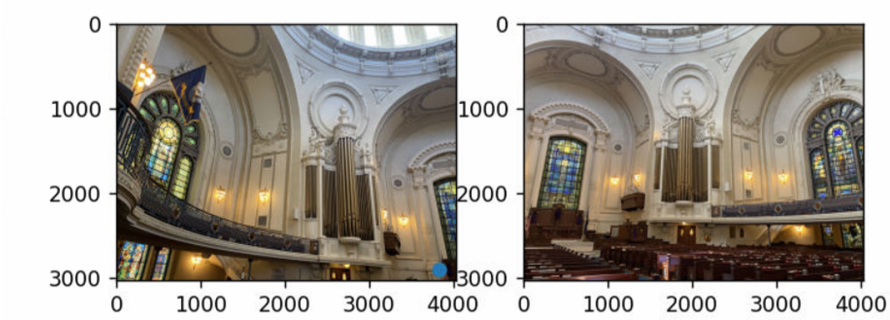
Project III: Image Mosaics

Adellar Irakunda

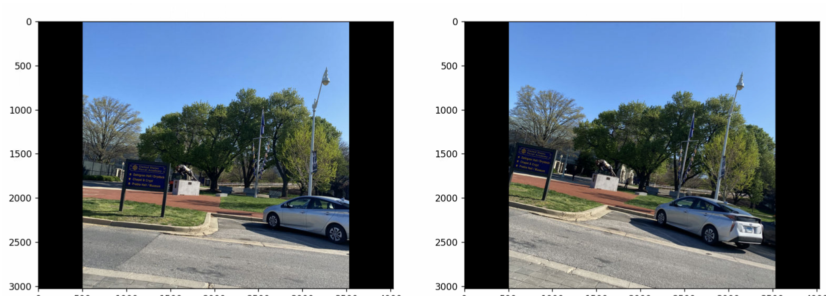
April 2022

1 Introduction

Image mosaicing is a technique used to create a single image by stitching multiple images together based on visual commonalities. This is done to create a single continuous image, reflecting a combination of the constituent projections. Visual identification of common feature sets is done manually via user input, from which homography matrices are calculated to represent the difference in projection between images. An attempt was made at creating a method for automatically identifying image commonalities, but it requires more work. In this project, this procedure is done between only two images such that this matrix is easily inversable for debugging. Below are the steps taken to calculate image correspondents, find the homography matrix, and warp two input images into a singular semi-panorama.



(a) Chapel at United States Naval Academy (USNA)



(b) Parking lot at USNA

Figure 1: Input Images for Mosaicing

2 Methods

2.1 Calculating Homographies

To have a basis on how to combine the two images, we must obtain a homography matrix, based on a selection of visually defined points. This can be represented as a system of linear equations such as $AX = B$. With 8 known points in this matrix X, we can arrange our matrix, where $p_1 - p_8$ are our knowns. p_i are user defined by clicking on points in both inputs that represent common features. We store these point pairs in the matrix p_i as such:

$$p_i = \begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_i x_i' & y_i x_i' & x_i' \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_i y_i' & y_i y_i' & y_i' \end{bmatrix} \tag{1}$$

We can then set our matrix = 0 and set the constraint that our $|H| = 0$ and solve for for our values of H. We expect the output matrix to be 3x3. For any point (i), we can use this general matrix form to arrive at H.

$$PH = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1x_1' & y_1x_1' & x_1' \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1y_1' & y_1y_1' & y_1' \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2x_2' & y_2x_2' & x_2' \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2y_2' & y_2y_2' & y_2' \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3x_3' & y_3x_3' & x_3' \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3y_3' & y_3y_3' & y_3' \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4x_4' & y_4x_4' & x_4' \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4y_4' & y_4y_4' & y_4' \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \\ h9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2}$$

We break down each point into its x and y coordinates as shown in Equation 1. To arrive at the form given in Equation 2, we can take the two points and stack them into an nx9 matrix. The column matrix $h_1 - h_9$ is H which we will reshape into 3x3. With our constraint $PH = 0$, we use singular value decomposition and reshape the output to 3x3. This is now our Homography matrix. We can also derive H by using the least squares method where we arrange our system as $PH = P$‘, this is the non-homogeneous method for generating H.

2.2 Image Warping

With our homography, representing the shift between our two inputs, we can now use it to stich our images together. We must warp the images such that overlapping characteristics aren't overly represented. We start by assuming our resultant image to be double the width of our singular inputs, then we consider individual RGB pixel values. We take each pixel intensity and its position in homogeneous coordinates, then multiply it by the homography matrix. In this sense, we're regarding pixels from one image and projecting it into the plane of the secondary image. When this is done, we convert our homogeneous dimensions to cartesian coordinates and again take pixel intensities. We can then use bilinear interpolation to fill the pixel holes based on their neighbors. These artifacts are the results of forward warping. We sum our pixel values from forward and backwards warping for each channel and are provided with our final image output.

3 Results

Results of the warped images are shown in Figures 2 a and b. Figure 2b has some black bars artifacts which I could not understand their origination, but overall I'm pleased with the results.

Computation Photography/Comp-Mosaic/Output1.png

(a) Chapel at United States Naval Academy (USNA) Stitched

Computation Photography/Comp-Mosaic/Output2.png

(b) Parking lot at USNA Stiched

Figure 2: Output Images from Mosaicing

4 Conclusion

This was a great project that provided insight into how to warp images. I had some difficulty udnerstanding some of the linear algebra involved, which is what took me the longest, but I'm glad I was able to pull it together. Definitely an engaging project!