ECE 661 Homework 2

Michael Goldberg

September 3 2024

1 Task 1

1.1 Finding Homographies Between Frame and Test Image

The goal of this task is to map a region of interest from a test image to different regions of interests (ROIs) across three planar surfaces of different angles of a picture frame. The following images show the setup, with the test image being of Alex Honnold free-soloing El Capitan.

(a) Frame 1

(b) Frame 2

(c) Frame 3

(d) Test Image

I used python's matplotlib library to show the images and find the ROI bounding points I wanted to use. This was really easy because I could zoom in and get a very accurate pixel location of where my cursor was, and the window autosized to my monitor despite the images' varying resolutions. Once the ROIs were established, it was time to find the homographies. Considering points in the domain to be denoted X and points in the range to be denoted X', the following relationship is true: $X' = HX$, where X' and X are in homogeneous coordinates (HC) and H is the 3x3 homography matrix we are trying to find. Let's substitute in some intermediate variables to expand this equation out. Consider $X' = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix}$, $X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, and $H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$

Multiplying out, we get:

$$x_1' = h_{11}x_1 + h_{12}x_2 + h_{13}x_3 \tag{1}$$

$$x_2' = h_{21}x_1 + h_{22}x_2 + h_{23}x_3 \tag{2}$$

, and

$$x'_3 = h_{31}x_1 + h_{32}x_2 + h_{33}x_3 \tag{3}$$

Then, converting from HC back to physical coordinates, we can find the direct relationship between the domain physical points and the range physical points. This can be achieved through using the following that we know from HC relationships: $u = \frac{x_1}{x_3}$ and $v = \frac{x_2}{x_3}$

Therefore, we are left with:

$$u' = \frac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \tag{4}$$

and

$$v' = \frac{h_{21}x_1 + h_{22}x_2 + h_{23}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \tag{5}$$

We can further proceed by multiplying out the denominator to both sides of the equation for both u' and v'. Additionally, we can set x_3 to 1 due to the rational nature of HC, which makes x_1 and x_2 become u and v respectively. We are therefore left with:

$$h_{31}u'u + h_{32}u'v + h_{33}u' = h_{11}u + h_{12}v + h_{13} \tag{6}$$

and

$$h_{31}v'u + h_{32}v'v + h_{33}v' = h_{21}u + h_{22}v + h_{23} \tag{7}$$

We are therefore left with 8 unknowns (as h_{33} can be set to 1), which means we need at least 4 different points between the two images to find all homography matrix parameters, as each point solves one pair of the above equations (6) and (7). This is really easy, as we can use the four corners of the picture frame as well as the four corners of the test image photo. By setting up these equations for all four points and rearranging, we are left with the following relation:

$$\begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u'_1u_1 & -u'_1v_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v'_1u_1 & -v'_1v_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u'_2u_2 & -u'_2v_2 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -v'_2u_2 & -v'_2v_2 \\ u_3 & v_3 & 3 & 0 & 0 & 0 & -u'_3u_3 & -u'_3v_3 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -v'_3u_3 & -v'_3v_3 \\ u_4 & v_4 & 1 & 0 & 0 & 0 & -u'_4u_4 & -u'_4v_4 \\ 0 & 0 & 0 & u_4 & v_4 & 1 & -v'_4u_4 & -v'_4v_4 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} u'_1 \\ v'_1 \\ u'_2 \\ v'_2 \\ u'_3 \\ v'_3 \\ u'_4 \\ v'_4 \end{pmatrix} \tag{8}$$

If we label this equation in the following manner $Ah = p$, where A is the 8x8 matrix, h is the vector containing the homography parameters, and p is the vector containing the range points, we can solve directly for the homography parameters by doing: $h = A^{-1}p$. This method was used between the test image and each frame image to get corresponding homographies mapping Alex Honnold to the 3 ROIs found earlier. Once the homography is found, it can be applied to all points within the domain image to map it to the range image. The following shows the results for each of the frames:

Figure 2: Direct Mapping to Frame 1

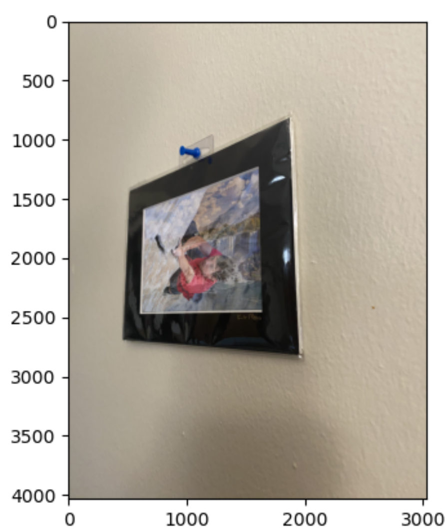


Figure 3: Direct Mapping to Frame 2

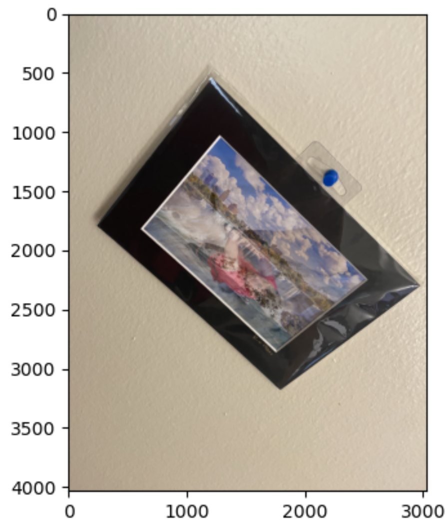


Figure 4: Direct Mapping to Frame 3

This direct mapping approach leads to the test image being placed in the correct spot, however there seems to be some transparency in the resultant image. This is because the range image that the test image is being mapped to is of higher resolution. This means that certain integer locations within the range are actually subpixel locations within the domain. This is a problem, as we only have integer pixel RGB data from the test image, which means we need to be more clever about the mapping if we want to circumvent this transparency issue. The solution that worked for me was to actually perform a reverse mapping - that is, find the inverse of the homography found earlier, apply it to the range image to map it into the domain, and find the corresponding rounded integer RGB value for the mapped point. Then, directly replace the point in the range with the sampled domain's RGB value. This can be better seen in a mathematical sense. The first step was to find the inverse of H, which is easily done with np.linalg.inv(H), as H is guaranteed to have an inverse due to its constraints of being non-singular and square. Therefore, any domain point X can be found from any range point X' through:

$$X = H^{-1}X' \tag{9}$$

The corresponding integer location of this mapped range point can be found by taking $int(\frac{x_1}{x_3})$ and $int(\frac{x_2}{x_3})$. We can then extract the RGB value of this point using array indexing of the integer location ($(R, G, B) = image_{domain}[v, u]$), and then directly set the original range point to this RGB value through:

$$image_{range}[X'] = (R, G, B) \tag{10}$$

Using this method for applying the homographies, we can see the following, which got rid of the transparency problem from before:

Figure 5: Mapping to Frame 1 with Inverse Homography Method

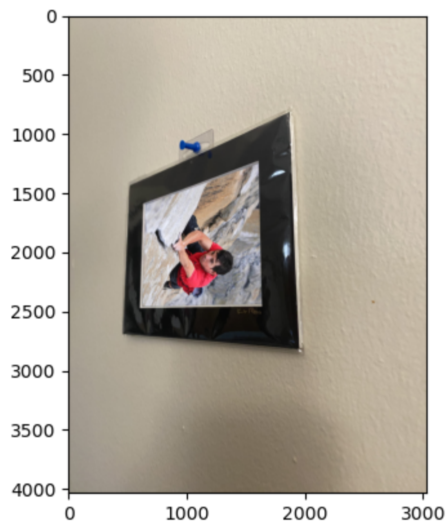


Figure 6: Mapping to Frame 2 with Inverse Homography Method

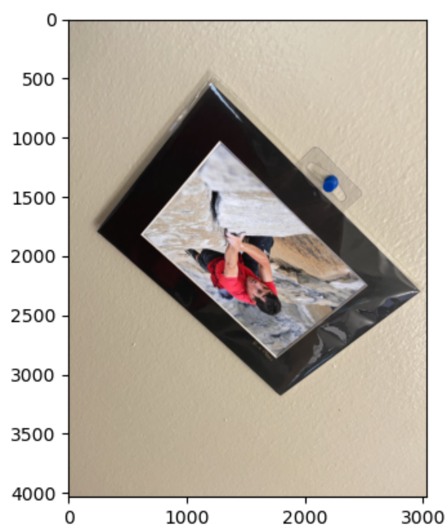


Figure 7: Mapping to Frame 3 with Inverse Homography Method

1.2 Combination Homography

This task is to find the homography from Frame 1 to Frame 2, and then Frame 2 to Frame 3. By then multiplying these two homographies together, we should get a homography from Frame 1 to Frame 3. I used the direct homography approach, as all of the images are the same resolution. The following is the result of this:

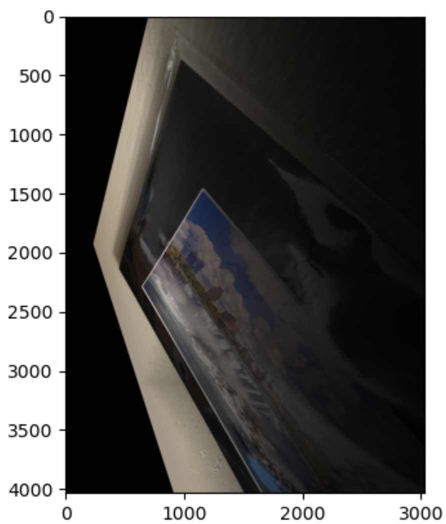


Figure 8: Mapping Frame 1 to Frame 3 Using Product of Intermediate Homographies H12 and H23

1.3 Affine-Only Homographies

The previous homographies were not strictly affine, and as a result, they were able to give a mapping that accounted for similarity, affine, and purely projective distortion. However, let's see what happens if we restrict the homographies to be strictly affine. This requires the last row of the homography matrix H to be $(0, 0, 1)$. Let's setup the equation again under this constraint and see the results. In order for the homography to be affine, the following relation is true:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{11}$$

Which can be expanded out to:

$$u' = h_{11}u + h_{12}v + h_{13} \tag{12}$$

and

$$v' = h_{21}u + h_{22}v + h_{23} \tag{13}$$

We can then construct an equation similar to before in the form $Ah = p$:

$$\begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_2 & v_2 & 1 \\ u_3 & v_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_3 & v_3 & 1 \\ u_4 & v_4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_4 & v_4 & 1 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \end{pmatrix} = \begin{pmatrix} u'_1 \\ v'_1 \\ u'_2 \\ v'_2 \\ u'_3 \\ v'_3 \\ u'_4 \\ v'_4 \end{pmatrix} \tag{14}$$

This time, however, to solve for h, A is not square which means it does not have a direct inverse. We can instead take the pseudo-inverse to find h using np.linalg.pinv(A). The following is the result of each affine homography:

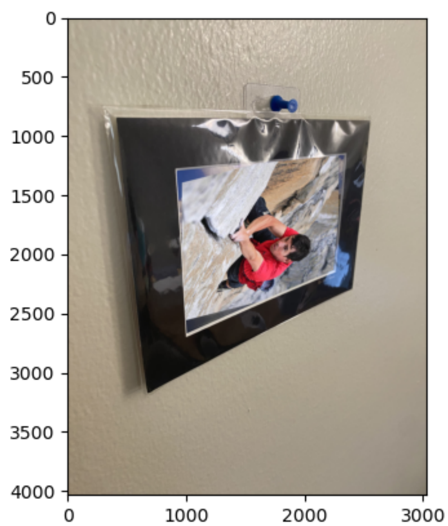


Figure 9: Affine Mapping to Frame 1

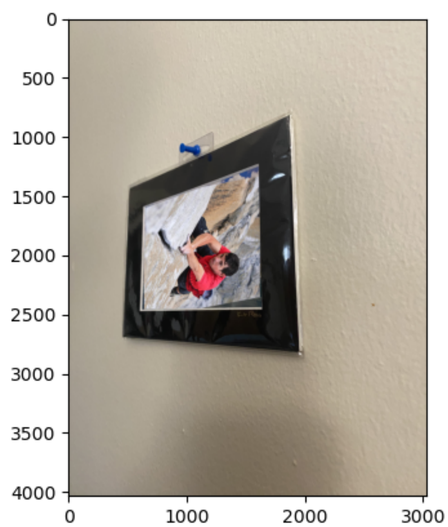


Figure 10: Affine Mapping to Frame 2

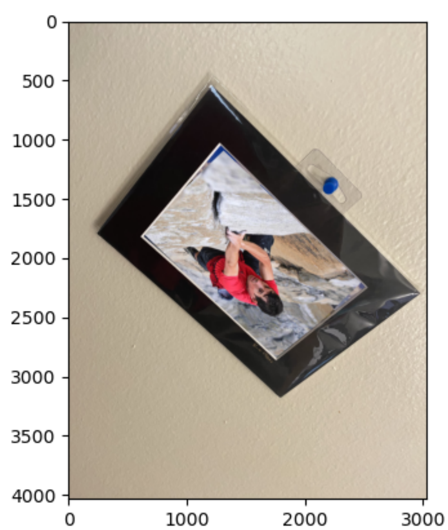


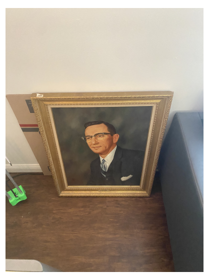
Figure 11: Affine Mapping to Frame 3

It can be seen that the Frame 3 affine mapping works the best - this is because this frame orientation is the one which keeps the lines the most parallel across all of the different frame angles. Frames 1 and 2 introduce non-trivial, non-affine distortion, whereas with Frame 3 that is kept to a minimum.

2 Task 2

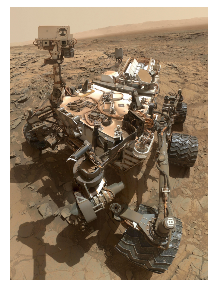
This task required me to use my own images. I found a framed picture of a random guy in a dumpster near my apartment, and I thought he looked quite distinguished, so I fished it out and kept it as an apartment decoration. His name is Bill Johnson. I will use three different angles of him to then map a picture of the Mars Curiosity Rover onto. Here are the original 3 images of Mr. Bill Johnson as well as the test image of Curiosity:

(a) Frame 1



(b) Frame 2

(c) Frame 3



(d) Test Image

And using the same methods from above, here are the mapped images:

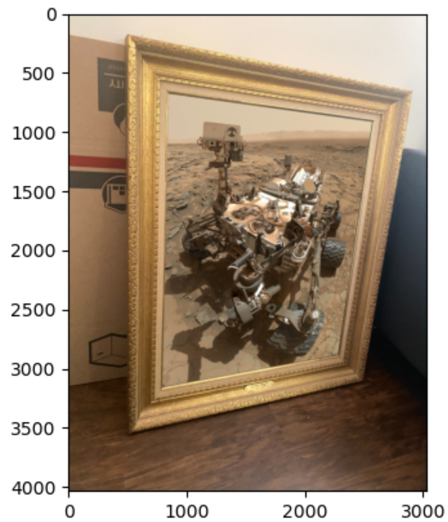


Figure 13: Affine Mapping to Frame 1

Figure 14: Affine Mapping to Frame 2

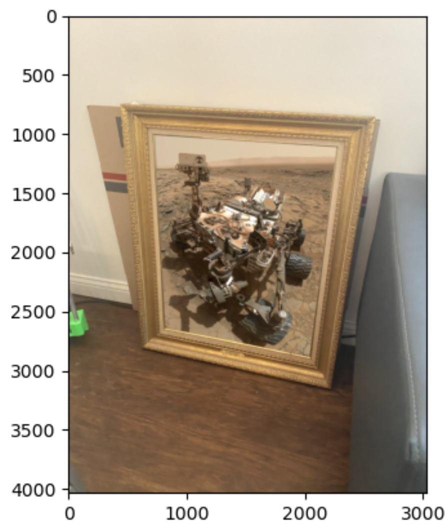


Figure 15: Affine Mapping to Frame 3