

## Part 1

### Recovering the Homographies

A homography is given by a transformation sending  $v_1 = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$  to  $v_2 = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$  mod scaling through a matrix:

$$c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Multiplying everything out, we get:

$$\begin{aligned} cx' &= h_1x + h_2y + h_3 \\ cy' &= h_4x + h_5y + h_6 \\ c &= h_7x + h_8y + 1 \end{aligned}$$

and substituting:

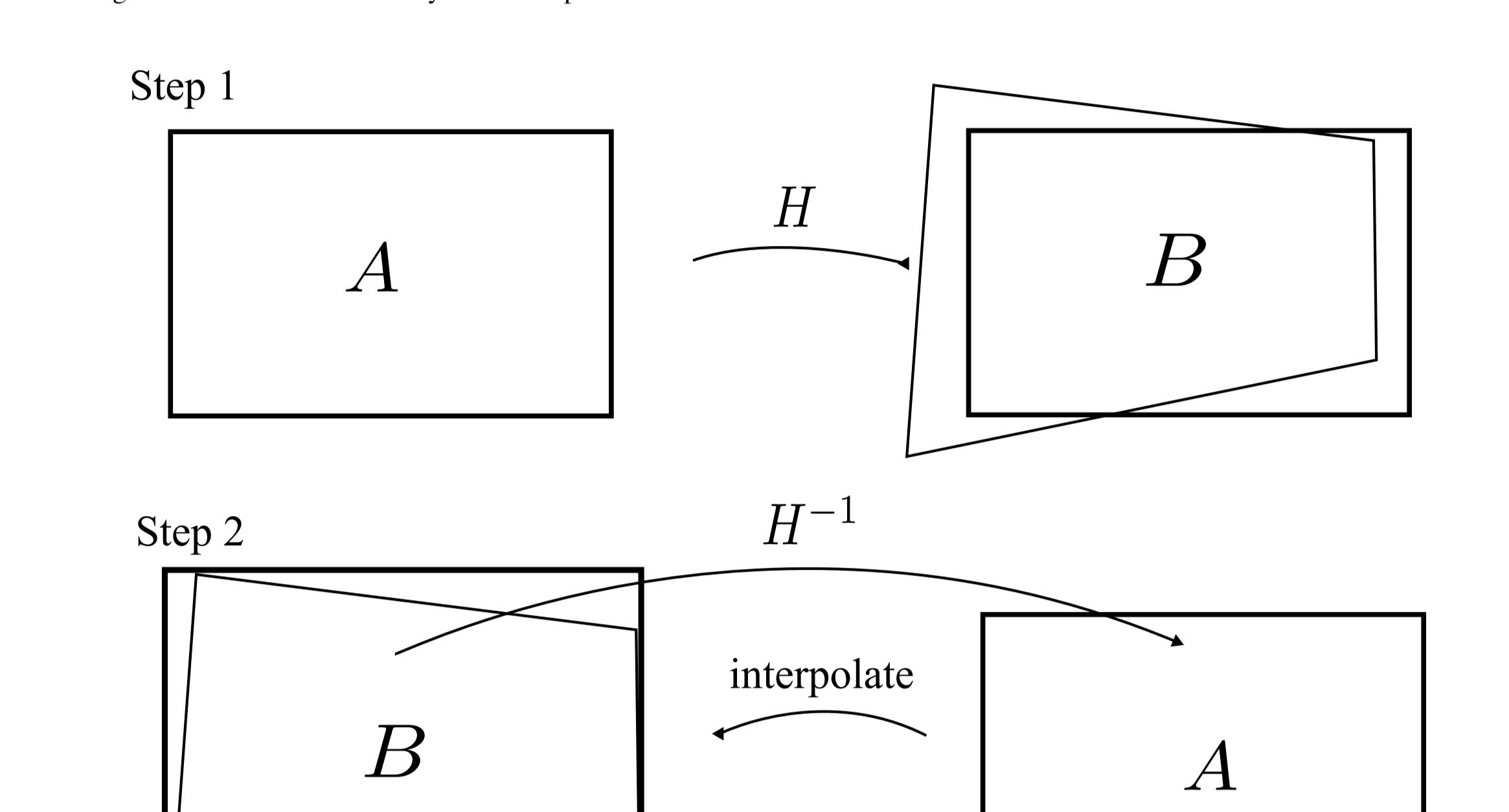
$$\begin{aligned} (h_7x + h_8y + 1)x' &= h_1x + h_2y + h_3 \\ (h_7x + h_8y + 1)y' &= h_4x + h_5y + h_6 \end{aligned}$$

will give the system of equations:

$$\begin{aligned} h_1x + h_2y + h_3 + 0h_4 + 0h_5 + 0h_6 - h_7x^2 - h_8xy' - 1x' &= 0 \\ 0h_1 + 0h_2 + 0h_3 + h_4x + h_5y + h_6 - h_7xy' - h_8yy' - 1y' &= 0 \end{aligned}$$

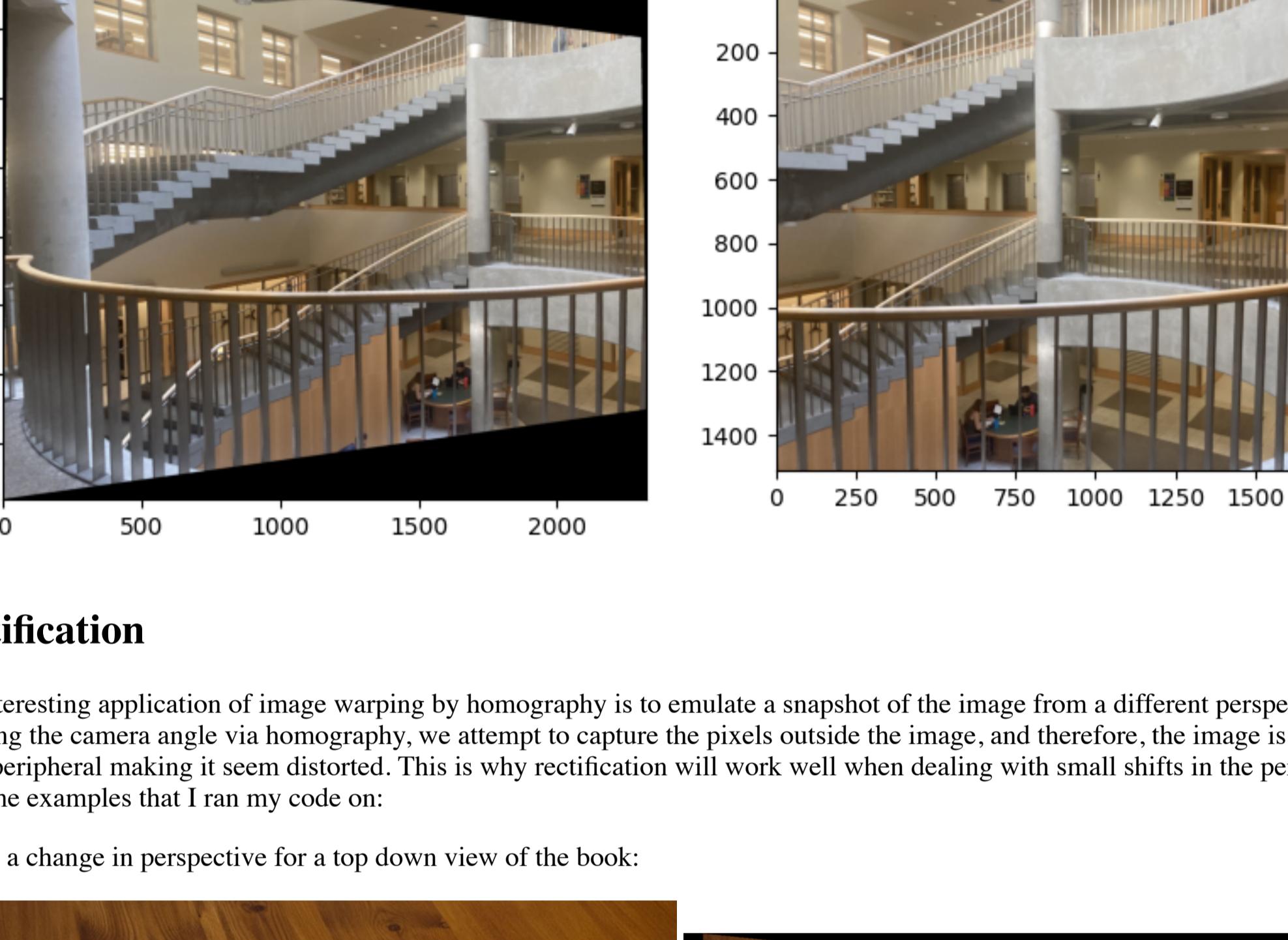
This gives a way to solve for the entries of the matrix  $H$  through a different linear equation  $Ax = b$  where  $A$  is a  $2 \times 8$  matrix,  $x$  is the vector containing the entries  $h_1, \dots, h_8$ , and  $b$  is the vector containing the values  $x', y'$ . This is extendable to include more points in our homography mapping and can be solved through least squares to find the projective mapping

Here is an example of the point correspondences that I used to compute the homography

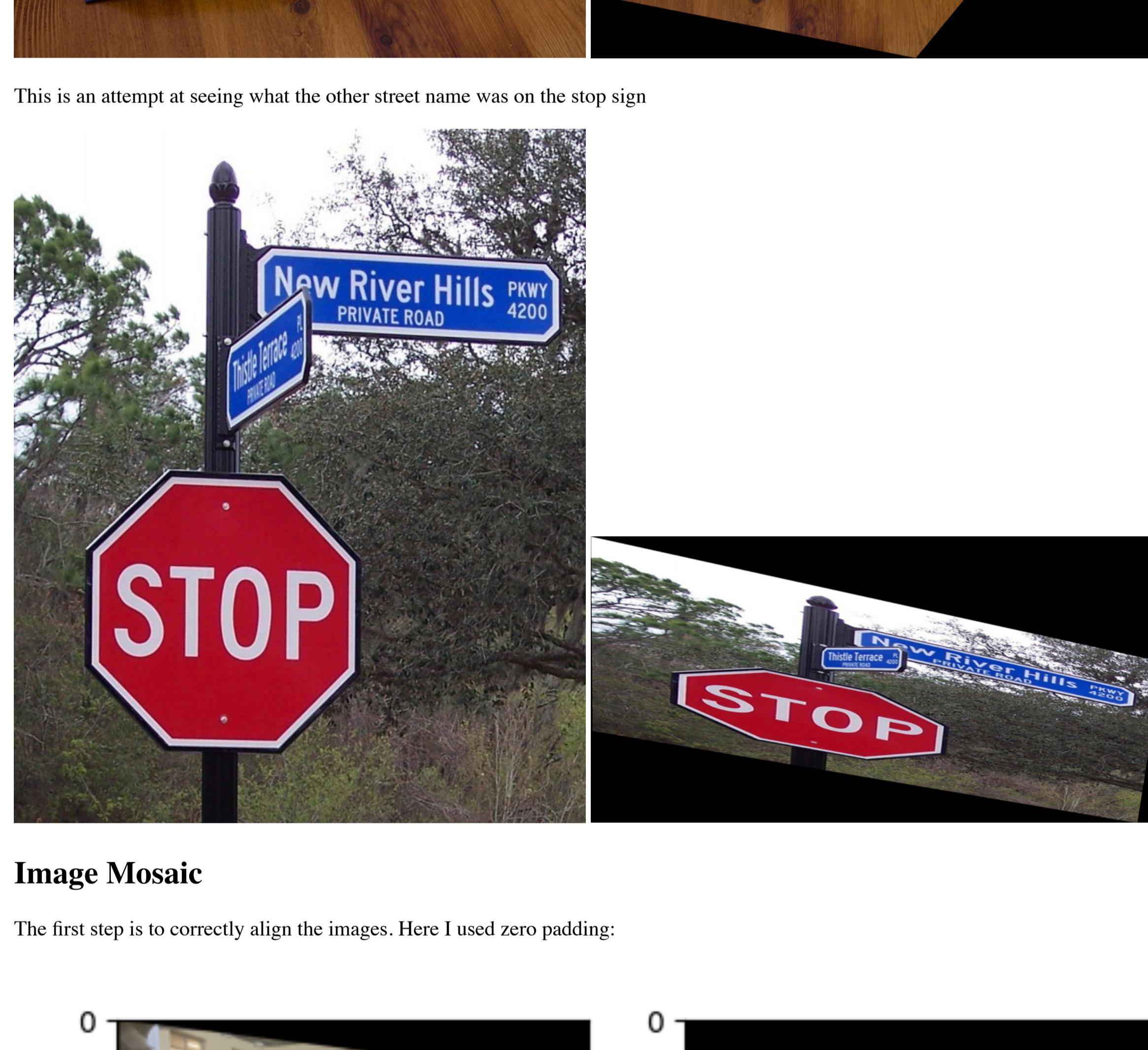


### Warping the Image

This process was very similar to the last project, but instead, the warp would be determined by four corners under the homography transformation. After calculating the new corners, a resulting image would be set to encompass the warped image. The pixel values in the new image  $B$  would be determined by where the pixel landed under the transformation  $H^{-1}$ .



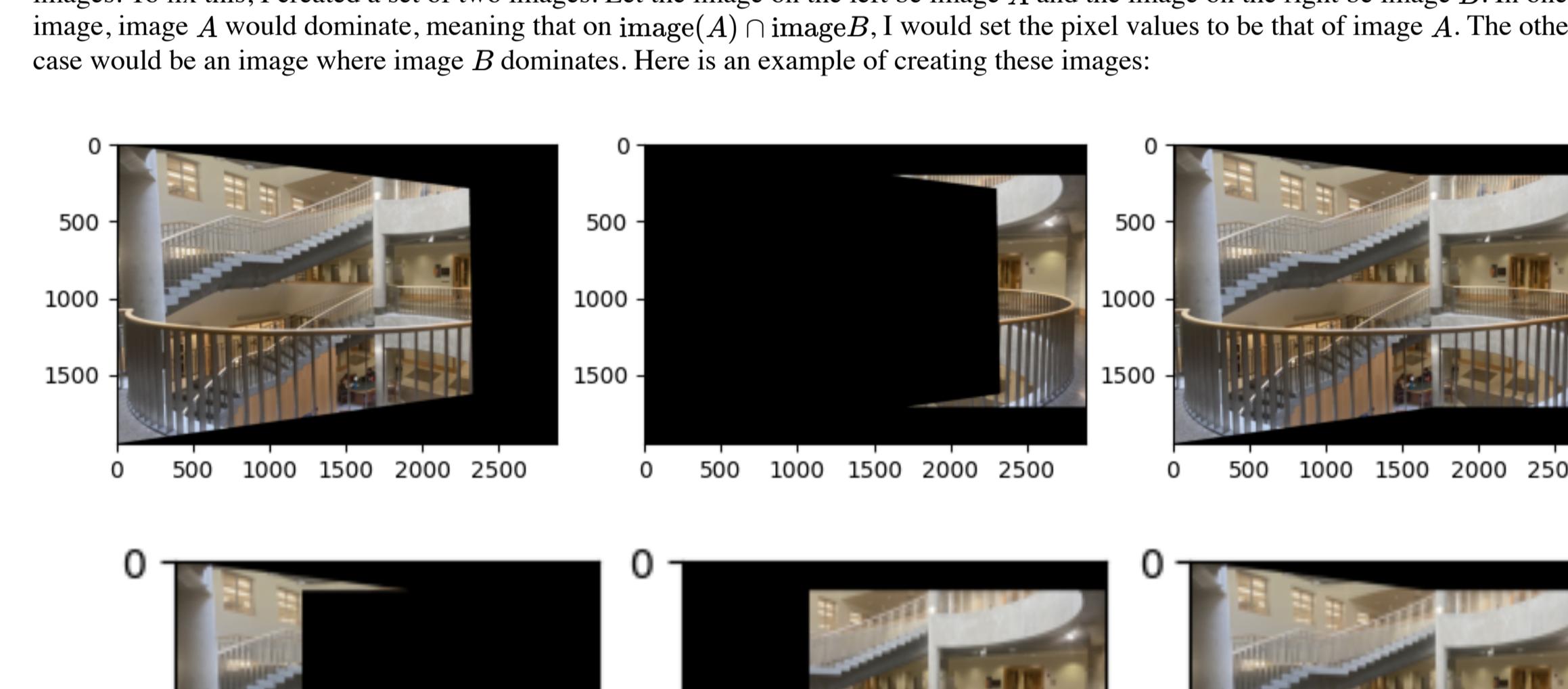
Since the pixel location under  $H^{-1}$  is not exact, I used griddata to interpolate for the pixel value using the location and values in the original image as the data. Here is the result of the warp on the left and the base image on the right



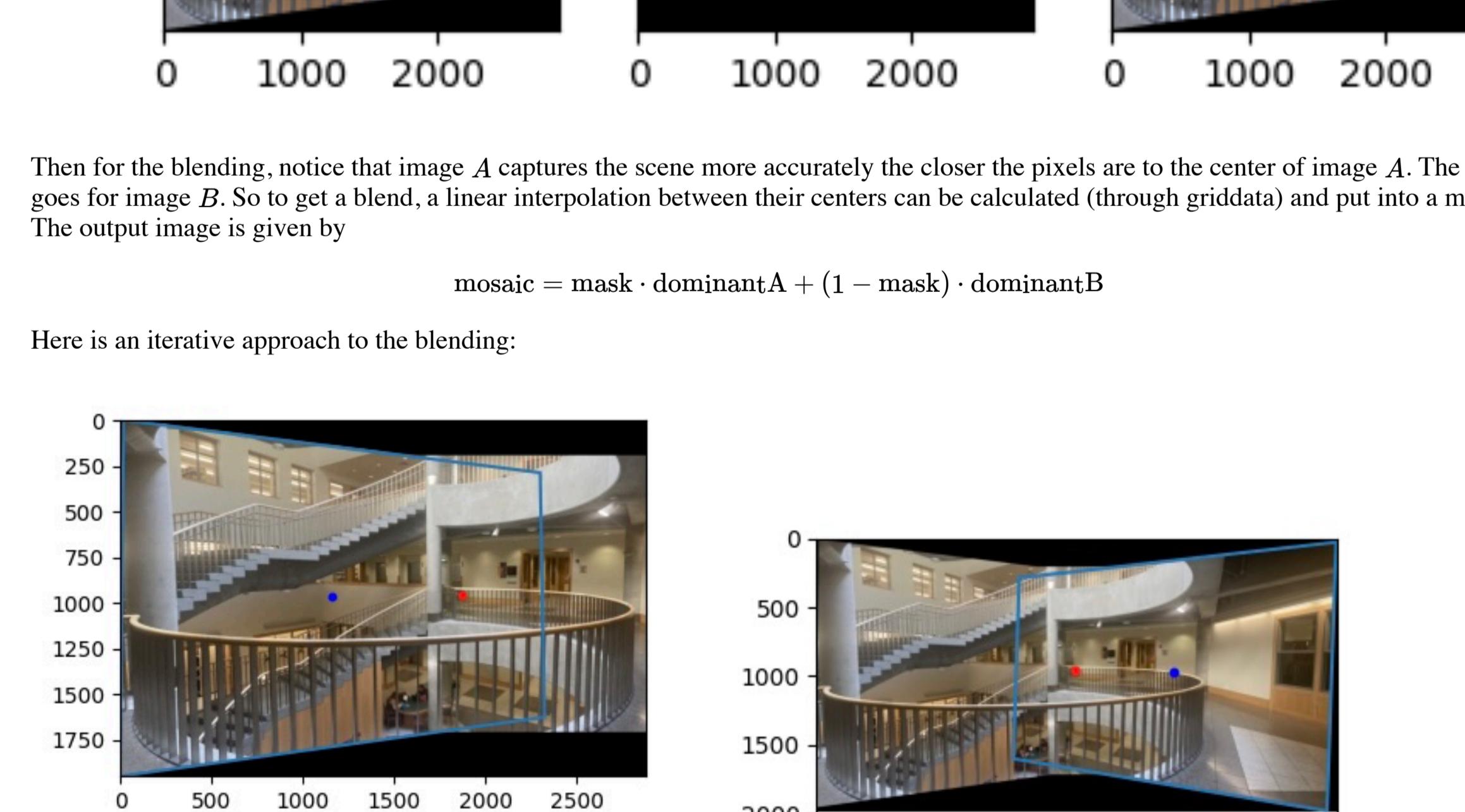
### Rectification

One interesting application of image warping by homography is to emulate a snapshot of the image from a different perspective. In changing the camera angle via homography, we attempt to capture the pixels outside the image, and therefore, the image is shifted further to the peripheral making it seem distorted. This is why rectification will work well when dealing with small shifts in the perspective. Here are some examples that I ran my code on:

Here is a change in perspective for a top down view of the book:

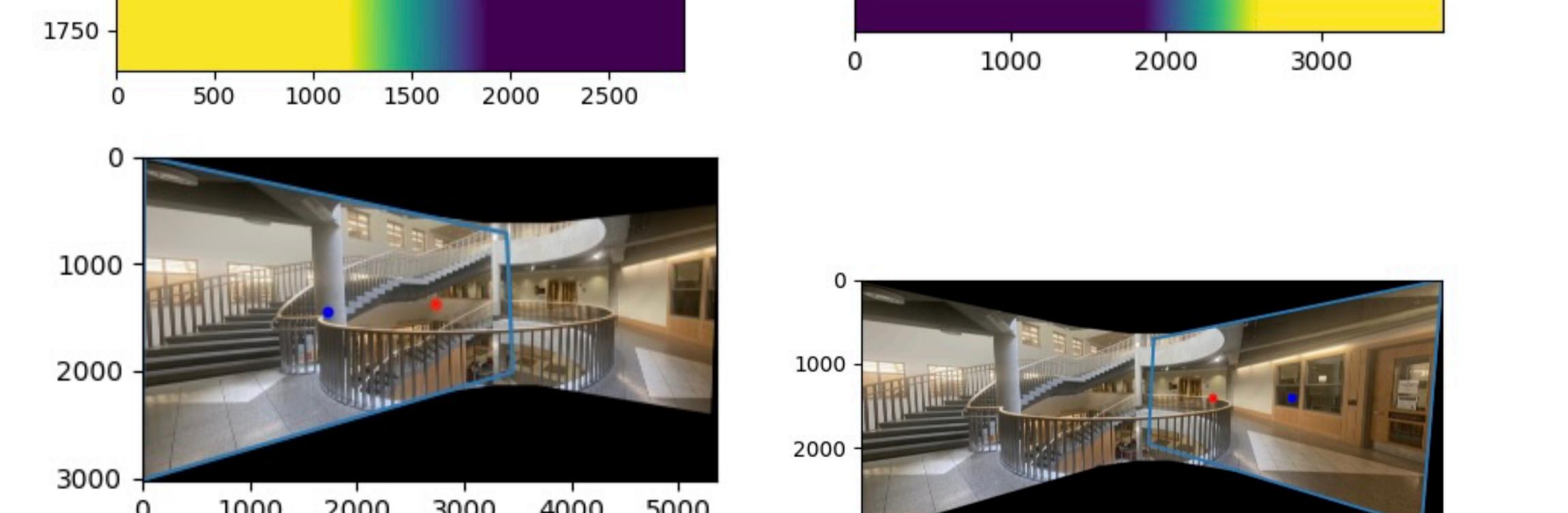


This is an attempt at seeing what the other street name was on the stop sign

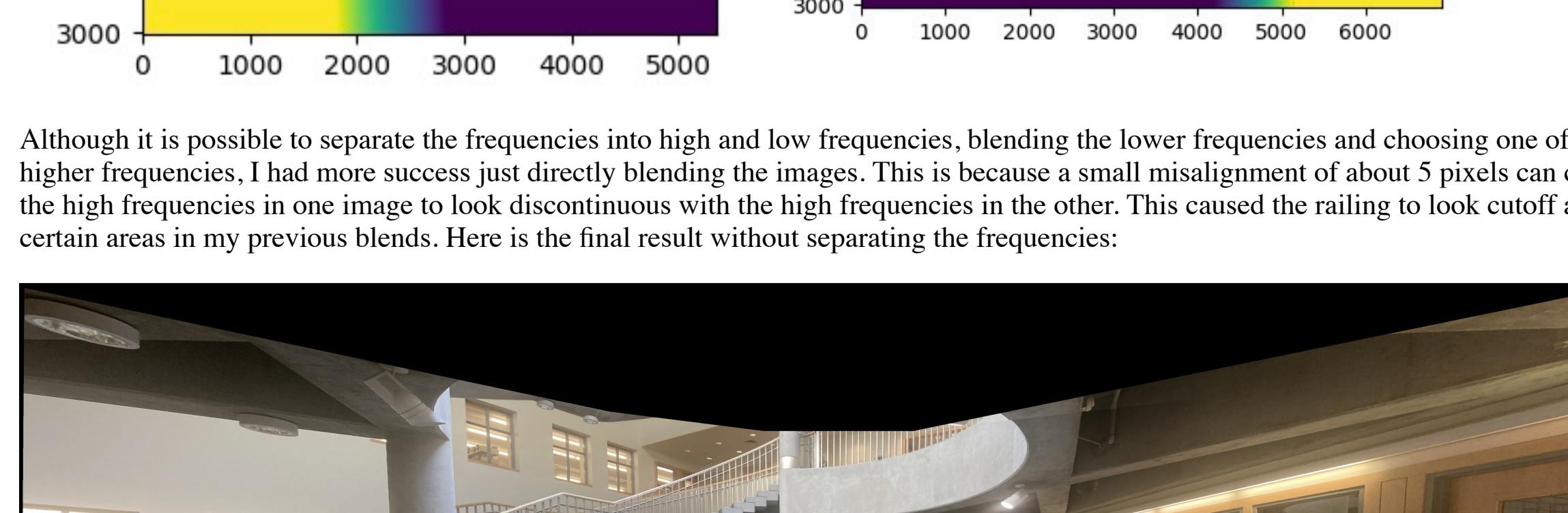


### Image Mosaic

The first step is to correctly align the images. Here I used zero padding:



The images cannot be directly added together and averaged at the intersection because there is different lighting and exposure across images. To fix this, I created a set of two images. Let the image on the left be image  $A$  and the image on the right be image  $B$ . In one image, image  $A$  would dominate, meaning that on  $\text{image}(A) \cap \text{image}(B)$ , I would set the pixel values to be that of image  $A$ . The other case would be an image where image  $B$  dominates. Here is an example of creating these images:



Then for the blending, notice that image  $A$  captures the scene more accurately the closer the pixels are to the center of image  $A$ . The same goes for image  $B$ . So to get a blend, a linear interpolation between their centers can be calculated (through griddata) and put into a mask. The output image is given by

$$\text{mosaic} = \text{mask} \cdot \text{dominantA} + (1 - \text{mask}) \cdot \text{dominantB}$$

Here is an iterative approach to the blending:



Although it is possible to separate the frequencies into high and low frequencies, blending the lower frequencies and choosing one of the higher frequencies, I had more success just directly blending the images. This is because a small misalignment of about 5 pixels can cause the high frequencies in one image to look discontinuous with the high frequencies in the other. This caused the railing to look cutoff at certain areas in my previous blends. Here is the final result without separating the frequencies:

