Final Project

Colin Bunker

For this project I selected the augmented reality assignment.

Calibrating Camera

First, point correspondences must be found between the checkboard corners in the calibration sequence and the real world checkboard corner locations in 3D.

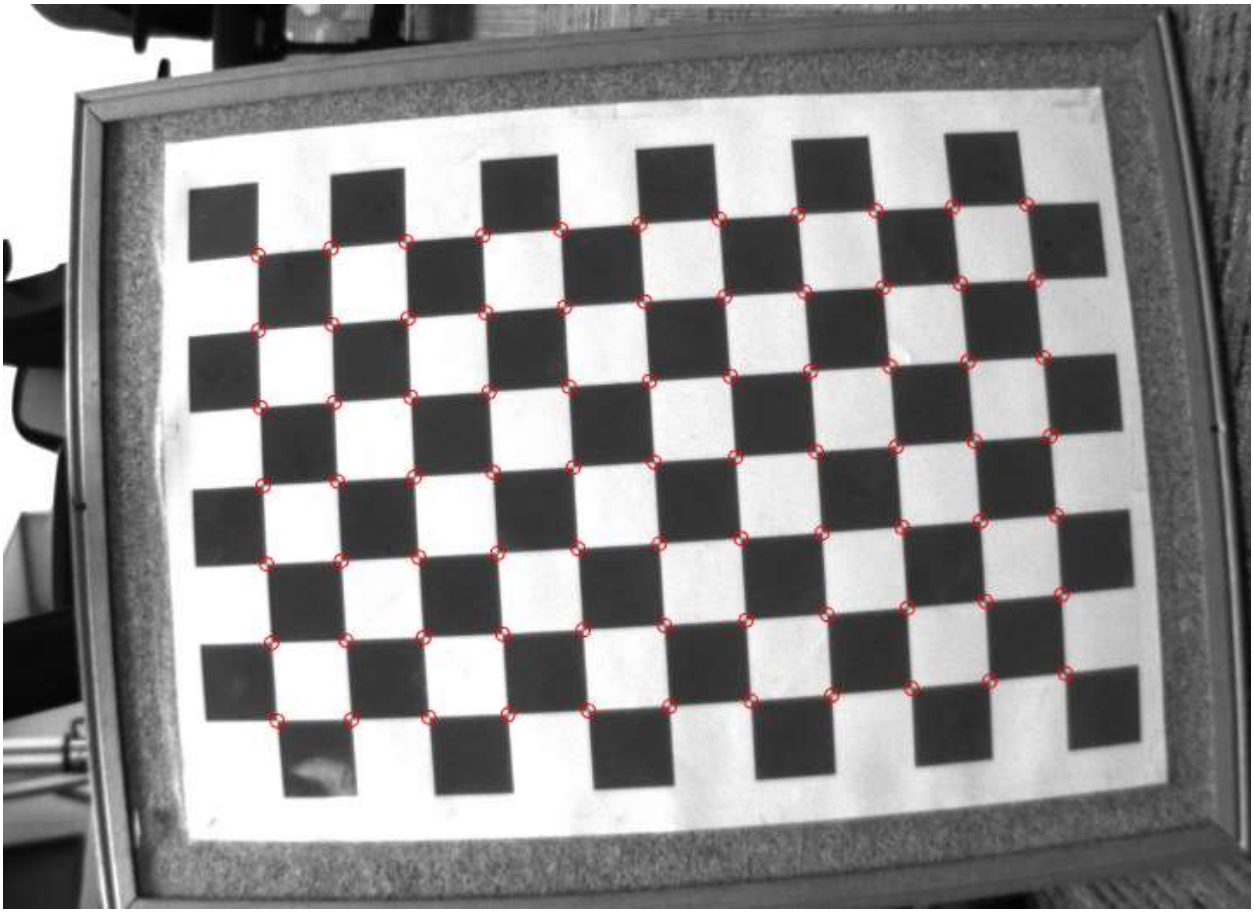

Figure 1: Detected Checkboard Corners

Then, a two-step process was used to estimate the intrinsic matrix and distortions from the calibration sequence. The first step was to find a closed-solution to solving for the intrinsic matrix, K. The second step was to use nonlinear optimization to not only adjust K, but also solve for the radial and tangential distortions.

There are two key constraints put on K as related to the homography, H.

$$h_1^T K^{-T} K^{-1} h_2 = 0$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2$$

Where h_i are column vectors of H.

Let $B = K^{-T} K^{-1}$

$$\mathrm{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & \frac{-\gamma}{\alpha^2\beta^2} & \frac{\gamma o_y - \beta o_x}{\alpha^2\beta^2} \\ \frac{-\gamma}{\alpha^2\beta^2} & \frac{\gamma^2 + \alpha^2}{\alpha^2\beta^2} & \frac{(-\gamma^2 - \alpha^2) o_y + \beta\gamma o_x}{\alpha^2\beta^2} \\ \frac{\gamma o_y - \beta o_x}{\alpha^2\beta^2} & \frac{(-\gamma^2 - \alpha^2) o_y + \beta\gamma o_x}{\alpha^2\beta^2} & \frac{(\gamma^2 + \alpha^2)}{\alpha^2\beta^2} \end{bmatrix}$$

This is symmetric and can be described simply as:

$$b = [b_{11} \quad b_{12} \quad b_{22} \quad b_{13} \quad b_{23} \quad b_{33}]$$

Thus:

$$h_1^T K^{-T} K^{-1} h_2 = 0$$

And

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2$$

Can be written as:

$$h_i^T B h_j = v_{ij}^T b$$

Where:

$$v = \begin{bmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{bmatrix}$$

So then we have:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0$$

Which can be solved in a Least Squares sense. The intrinsic matrix parameters are then:

$$v_0 = \frac{b_{12}b_{13} - b_{11}b_{23}}{b_{11}b_{22} - b_{12}^2}$$

$$\lambda = b_{33} - \frac{b_{13}^2 + v_0(b_{12}b_{13} - b_{11}b_{23})}{b_{11}}$$

$$\alpha = \sqrt{\frac{\lambda}{b_{11}}}$$

$$\beta = \sqrt{\frac{\lambda b_{11}}{b_{11}b_{22} - b_{12}^2}}$$

$$\gamma = -\frac{b_{12}\alpha^2\beta}{\lambda}$$

$$u_0 = \frac{\gamma v_0}{\alpha} - \frac{b_{13}\alpha^2}{\lambda}$$

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Once the intrinsic matrix is estimated, the extrinsic matrix for each image is the only thing that needs to be approximated. A point on a 2D image, p, can be calculated using a homography from its corresponding 3D point, P by:

$$\lambda x = HX$$

This is because, in our case, all 3D points are planar, so the Z component drops. This is accurate up to scale, hence the λ. Also:

$$\lambda x = PX$$

$$\lambda x = K[r_1 \quad r_2 \quad r_3 \quad t]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Where r_i and t are column vectors denoting rotation and translation. It is obvious that r_3 and the 0 term in X can be removed.

$$\lambda x = K[r_1 \quad r_2 \quad t]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = [h_1 \quad h_2 \quad h_3]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Where h_i are column vectors of H. r_1, r_2, and r_3 are orthogonal, thus:

$$r_1 = \lambda K^{-1}h_1$$

$$r_2 = \lambda K^{-1} h_2$$

$$r_3 = r_1 \times r_2$$

$$t = \lambda K^{-1} h_3$$

$$\lambda = \frac{1}{\|K^{-1} h_1\|} = \frac{1}{\|K^{-1} h_2\|}$$

Using this algorithm, r_1, r_2, and r_3 will not generally be orthogonal, so Singular Value Decomposition can be use:

$$R = [r_1 \quad r_2 \quad r_3]$$

$$R = USV'$$

$$R_{orthogonal} = UV'$$

Once this closed-form solution is completed, a nonlinear estimation of a better K and distortions can be completed. The tangential distortion was assumed to be 0. The radial distortion matrix can be modelled as:

$$D = \begin{bmatrix} \frac{1}{1 + \kappa_1 r^2 + \kappa_2 r^4} & 0 & 0 \\ 0 & \frac{1}{1 + \kappa_1 r^2 + \kappa_2 r^4} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$r^2 = x^2 + y^2$$

Then, the nonlinear optimization algorithm (Levenberg-Marquardt) can be used to minimize:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \left\| m_{ij} - DK[R|t]M_j \right\|^2$$

Where m_{ij} is the j[th] point in image I, and M_j is the 3D coordinate of the j[th] point.

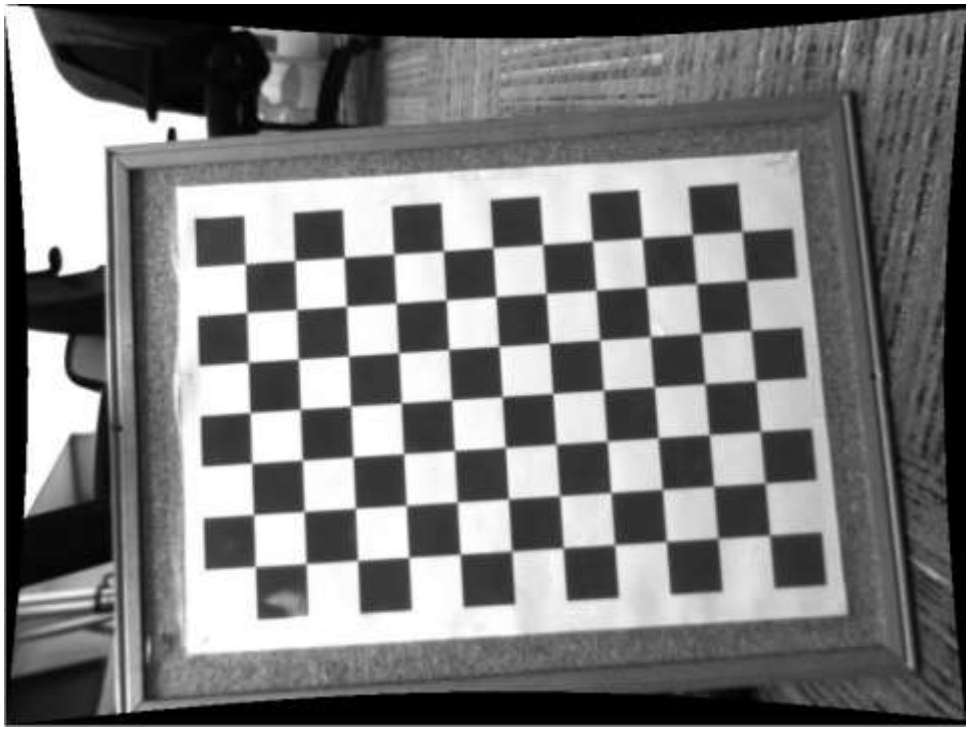

Figure 2: Undistorted Image

Locating Fiducial Points

In order to find an appropriate homography between the layout of the paper in the real world (shown in Figure 1) and the image of the paper, the fiducial points had to be located. Originally, I attempted to do this by finding SIFT points from the template given (shown in Figure 2) and matching them to SIFT points from the image itself. This proved to work for some of the images; however, for the most part this resulted in very inaccurate results.

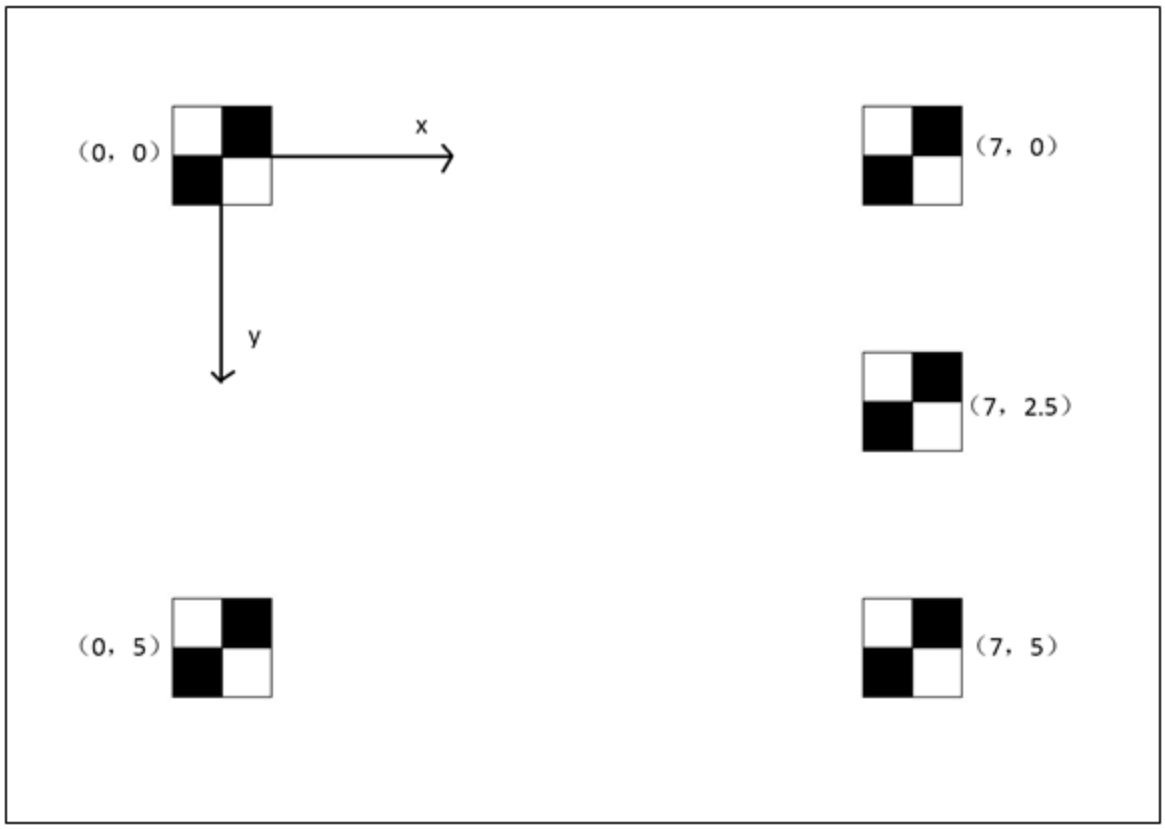

Figure 3: Real World Layout of Paper

Figure 4: Fiducial Marker

Next, I tried to an approach where I first found the homography by hand-selecting the correct points from the first image in the sequence. I then created a new image by transforming the first image in the sequence by the homography. I then cropped this new image so that all that was remaining was the paper this is shown in Figure 5. Finally, I used this image to find SIFT features which I matched to SIFT points in each of the following images in the video using RANSAC. Figures 6 and 7 show that, while for some cases (such as the first image) it did work, for this most part it yielded inaccurate results.

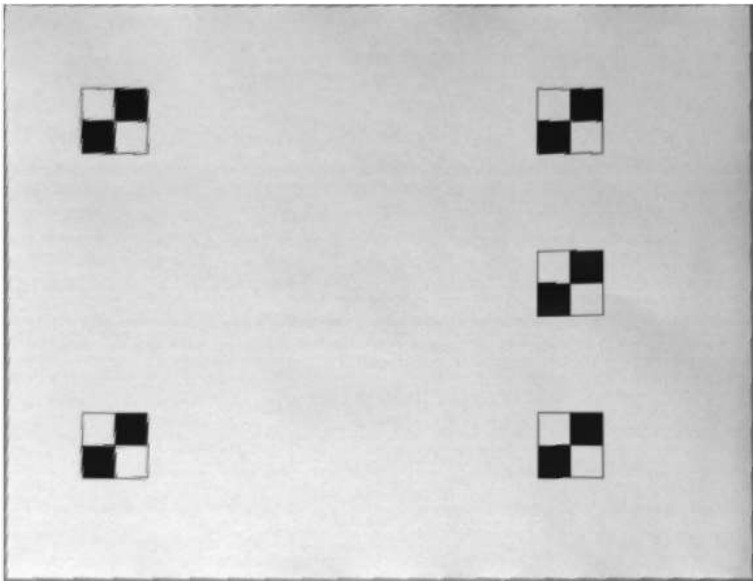

Figure 5: Paper Template

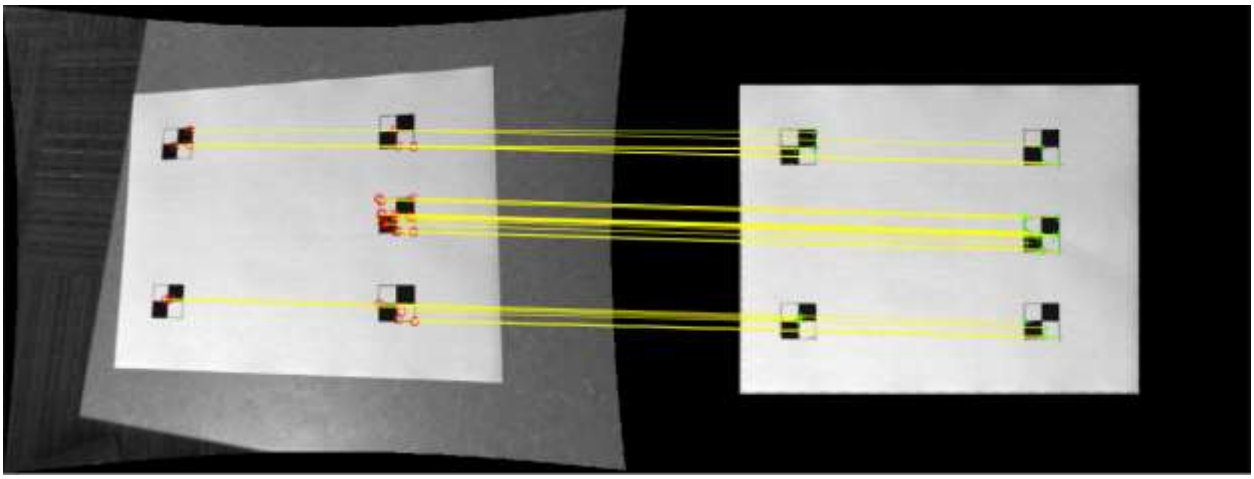

Figure 6: Matching SIFT Features using RANSAC, First Frame

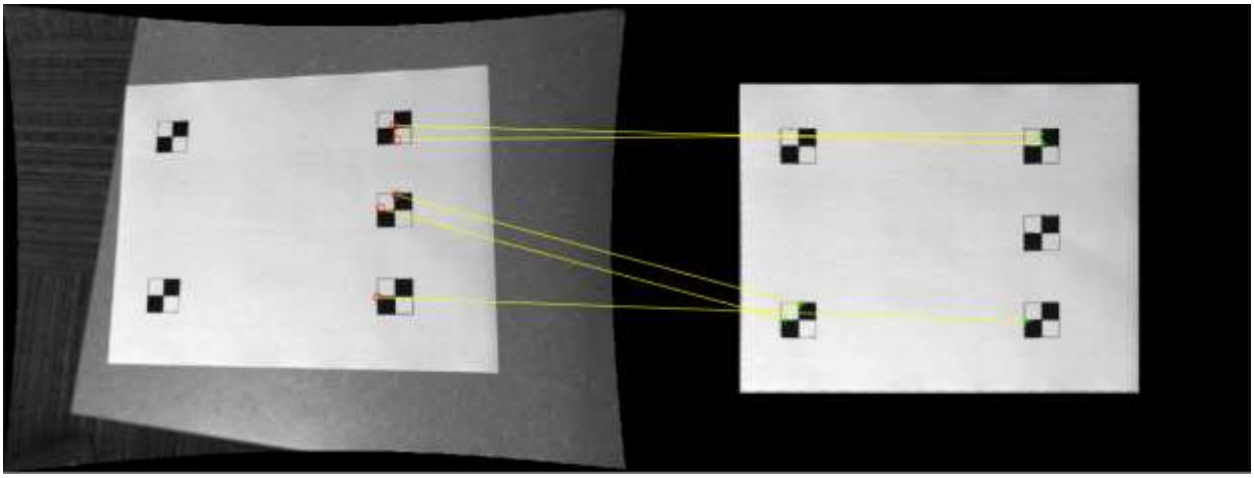

Figure 7: Matching SIFT Features using RANSAC, Second Frame

The final approach I took was to use the Normalized Cross Correlation algorithm. For each image, the Normalized Cross Correlation was found using the fiducial marker as the template. This outputted a map

of the normalized convolution of the marker to the image at each pixel. For each pixel in this map, I performed non-maxima suppression, and used the greatest five points as the five fiducial maker locations. This approach proved to be not only very accurate but also very robust during the entire video sequence.

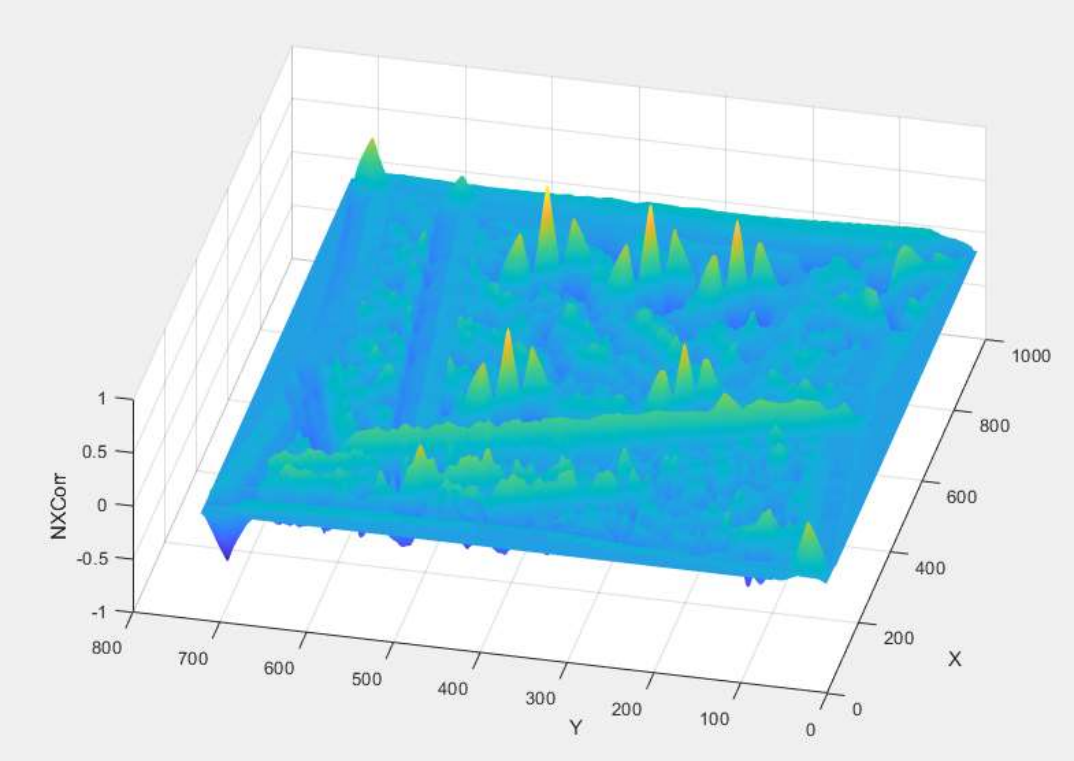

Figure 8: Normalized Cross Correlation with 5 Obvious Peaks

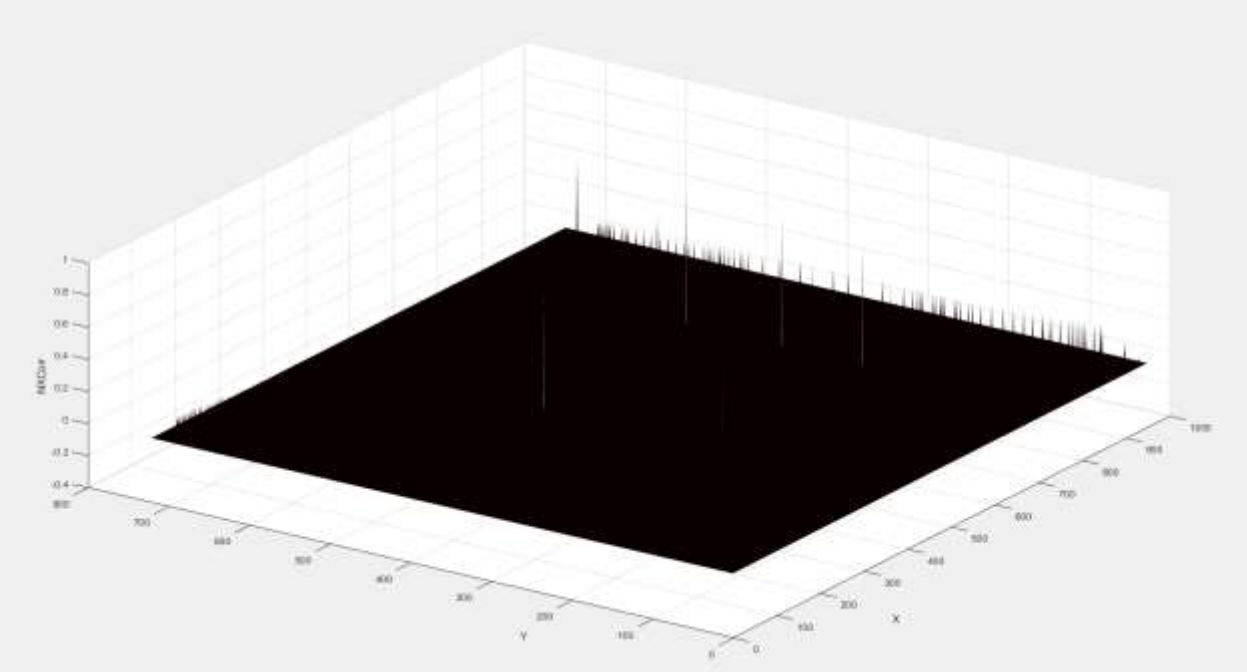

Figure 9: Non-maxima Suppression

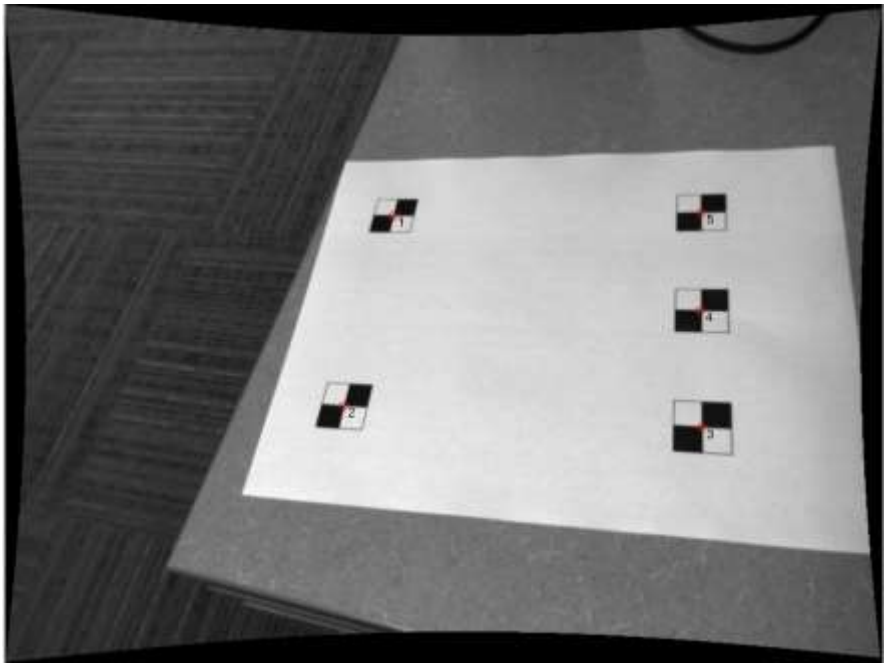

Figure 10: 5 Greatest Maximas Found to Calculate Fiducial Locations

Homography Calculation

For planar geometries a new point, x', is related to point x by:

$$x' = Hx$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In order to enforce only 8 degrees-of-freedom we can set $h_{33} = 1$.

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

$$x'(h_{31}x + h_{32}y + 1) = h_{11}x + h_{12}y + h_{13}$$

$$x' = h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx'$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}$$

$$y'(h_{31}x + h_{32}y + 1) = h_{21}x + h_{22}y + h_{23}$$

$$y' = h_{21}x + h_{22}y + h_{23} - h_{31}xx' - h_{32}yx'$$

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \\ \vdots \\ \vdots \end{bmatrix}$$

This make solving for H a simple Least Squares problem. Care must be taken in order to precondition the points. This is done by first mean-centering them, then scaling so that the variance is $\sqrt{2}$.

$$\bar{x} = Tx$$

$$\overline{x'} = T'x'$$

Where T is a 3x3 normalizing matrix, $\bar{x}$ is the point x after normalization and $\overline{x'}$ is the point x' after normaliztion. After $\bar{H}$ is solved for the normalized points, it must then be unnormalized by:

$$H = (T')^{-1}\bar{H}T$$

An example is shown below which shows an image after being transformed by a calculated homography.

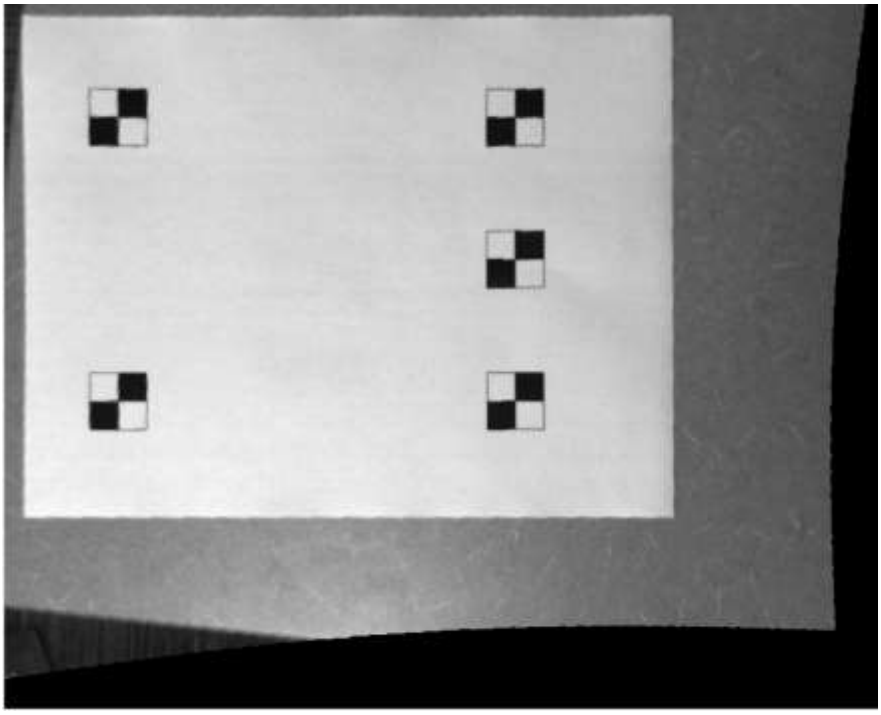

Figure 11: First Image, Transformed by Homography

Camera Resectioning

This task is very similar to the camera calibration task, but with fewer steps. First, the intrinsic matrix is already found, and second, no nonlinear adjustment was used.

The relevant equations are as follows:

$$r_1 = \lambda K^{-1} h_1$$

$$r_2 = \lambda K^{-1} h_2$$

$$r_3 = r_1 \times r_2$$

$$t = \lambda K^{-1} h_3$$

$$\lambda = \frac{1}{\|K^{-1} h_1\|} = \frac{1}{\|K^{-1} h_2\|}$$

$$R = [r_1 \quad r_2 \quad r_3]$$

$$R = USV'$$

$$R_{orthogonal} = UV'$$

Once this is completed, the projection of the 3D points to the 2D image coordinates is simply:

$$x = K[R|T]X$$

Polygon Drawing

For this task, I took some shortcuts that create a robust result, but is not very fast. The 3D shapes to be drawn are saved as a 4x3xN matrix where N is the number of faces. Each 4x3 layer of this matrix corresponds to the four vertices of a face of the object. The average X,Y, and Z components for each face is calculated, and the faces are sorted from farthest away from the camera pinhole to closest. Then, the faces are drawn in this order. This means that all faces will be drawn, but the closer ones will be drawn later, ensuring they are not covered up by farther away faces. Results can be found below.

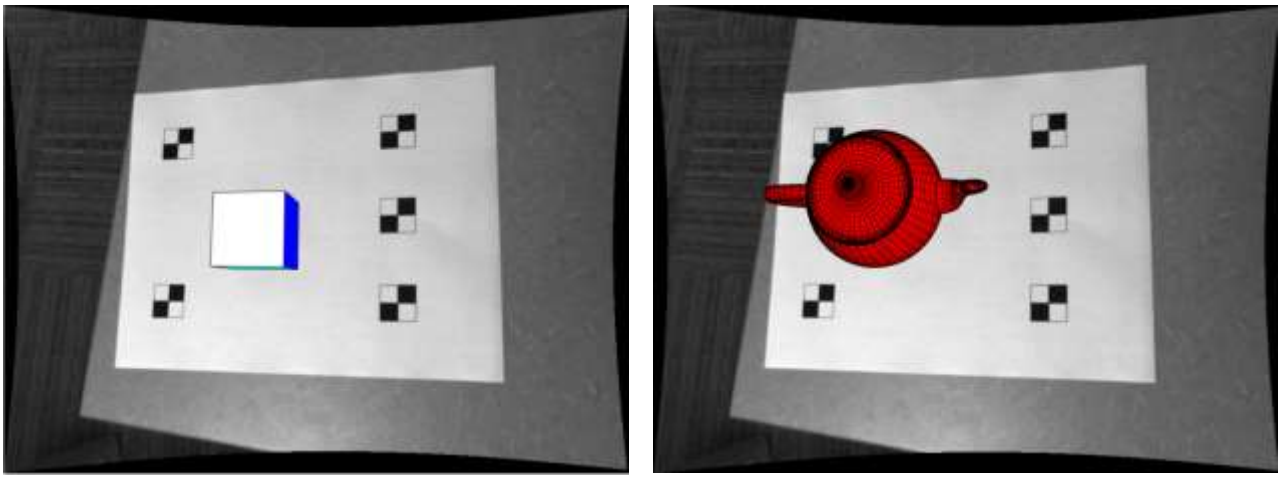

Figure 12: First frame, Cube and Teapot

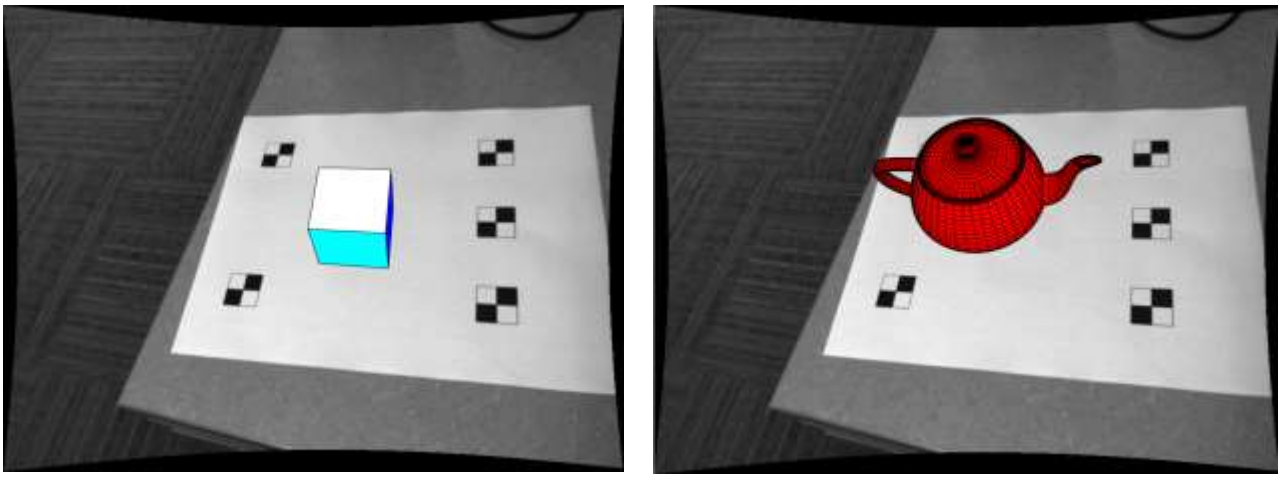

Figure 13: 100th frame, Cube and Teapot