Homework #9

ECE661

Name:Chengzhang Zhong

Email:zhongc@purdue.edu

Step 1. Detecting corners

1.1 Extract edges from image by using Canny edge detector. In my experiment, I used the build in function with threshold 0.7.

1.2 Apply the Hough transform for fitting straight lines. I used the MATLAB build-in function for this step. And group lines into horizontal and vertical line sets.

1.3 For the accuracy consideration, Harris corner detector was applied to find the nearby corners to the intersection of horizontal and vertical lines.

1.4 Detected corners were labeled with numbers, from left to right, top to bottom.

Step 2. Calibration

2.1 The calibration was based on Zhang's algorithm. To implement this algorithm, a calibration pattern need to be printed. Use camera to take pictures of this pattern from different distances and viewing angles.

Also, a world coordinate was defined by choosing the top left black block 's corner in the pattern as original point of the world as (0,0). One side of the black square and the distance between each square were measured as 20mm.

2.2 Use the world and image coordinate information as described above, we were able to compute homographies which mapped corners on the pattern from world onto image. This problem can be solved as finding null vector from a matrix, and the answer corresponds to the eigenvector with the smallest eigenvalue from SVD decomposition.

2.3 With the homographies provided by solving the previous step, we were able to find the intrinsic and extrinsic parameters of a camera by using the following methods:

As we learned from class, the absolute conic in the image can be represented as:

$$W = \mathrm{K}^{-\mathrm{T}}\mathrm{K}^{-1}$$

where K is the intrinsic matrix.

The homography maps pattern points from world onto image:

$$x = HX, H = K[r_1\, r_2\, t]$$

where $r_1\, r_2$ are the first and second column of rotational matrix R. t is the translation vector. R and t together build the extrinsic matrix for camera.

We know that there exist two circular points on the absolute conic:

$$HI = h_1 + ih_2, HJ = h_1 - ih_2, H = [h_1\, h_2\, h_3]$$

So now we have, ${h_1}^T W h_2 = 0\ and\ {h_1}^T W h_1 - {h_2}^T W h_2 = 0$

Let W = $\begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ w_{13} & w_{23} & w_{33} \end{bmatrix}$. Since W is symmetric, we can just use 6 of these parameters $W = [W_{11}\, W_{21}\, W_{22}\, W_{31}\, W_{32}\, W_{33}]^T$. Then, we have a new format to represent these equations:

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

$${V_{ij}}^T = [h_{i1}h_{j1}\ h_{i1}h_{j2} + h_{i2}h_{j1}\ h_{i2}h_{j2}\ h_{i3}h_{j1} + h_{i1}h_{j3}\ h_{i3}h_{j2} + h_{i2}h_{j3}\ h_{i3}h_{j3}]$$

Use similar idea above, the solution of this problem would be the SVD of the V matrix of the eigenvector corresponding the smallest eigenvalue.

Now, the intrinsic matrix K can be written as:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The parameters will be found by following:

$$x_0 = \frac{w_{12} - w_{13} - w_{11}w_{23}}{w_{11}w_{22} - w_{12}w_{12}}$$

$$\lambda = w_{33} - \frac{{w_{13}}^2 + x_0(w_{12}w_{13} - w_{11}w_{23})}{w_{11}}$$

$$\alpha_x = \sqrt{\frac{\lambda}{w_{11}}}$$

$$\alpha_y = \sqrt{\frac{\lambda w_{11}}{w_{11}w_{22} - {w_{12}}^2}}$$

$$s = -\frac{w_{12}{\alpha_x}^2\alpha_y}{\lambda}$$

$$y_0 = \frac{s x_0}{\alpha_y} - \frac{w_{13}{\alpha_x}^2}{\lambda}$$

In here, the intrinsic matrix is the same for every image from the same camera, but the extrinsic matrix need to be computed individually for each image:

$$R = [\mathrm{r_1\ r_2\ r_3}], \mathrm{K^{-1}[h_1\ h_2\ h_3] = [r_1\ r_2\ t]}$$

$$scalefactor = \frac{1}{\|\mathrm{K^{-1}h_1}\|}$$

$$\mathrm{r_1 = scale\ factor * K^{-1}h_1}$$

$$\mathrm{r_2 = scale\ factor * K^{-1}h_2}$$

$$\mathrm{r_3 = r_1\ x\ r_2}$$

$$\mathrm{t = scale\ factor * K^{-1}h_3}$$

Step 3. Calibration refine

We can refine the calibration parameters by minimizing the projection error. Let's define $\overrightarrow{\mathrm{x_{ij}}}$ as the j-th point in the i-th image. $\overrightarrow{\mathrm{x_{M,j}}}$ as the j-th salient point on the calibration pattern. Then $\widehat{\mathrm{x_{ij}}}$ be the projected image point for $\overrightarrow{\mathrm{x_{M,j}}}$ using P for the i-th camera position.

$$d_{geom}^2 = \sum_{\mathrm{i}}\sum_{\mathrm{j}}\left\|\overrightarrow{\mathrm{x_{ij}}} - \widehat{\mathrm{x_{ij}}}\right\|^2$$

which can also be written as

$$d_{geom}^2 = \sum_{\mathrm{i}}\sum_{\mathrm{j}}\left\|\overrightarrow{\mathrm{x_{ij}}} - \mathrm{K[R_i|t_i]}\overrightarrow{\mathrm{x_{M,j}}}\right\|^2$$

We stack all the parameters K,R and t from each image into a vector p. Then, use LM algorithm to refine this vector.

The matrix R can be converted into the representation as:

$$\vec{w} = \frac{\varphi}{2sin\varphi}\begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}$$

$$\varphi = cos^{-1}\frac{trace(R) - 1}{2}$$

After the refinement, we can use parameters extracted for each image to build their extrinsic parameters:

$$R = \mathrm{I} + \frac{\sin(\varphi)}{\varphi}\mathrm{W_x} + \frac{1 - \cos(\varphi)}{\varphi}\mathrm{W_x^2}$$

$$\mathrm{W_x} = \begin{bmatrix} 0 & -\mathrm{w_z} & \mathrm{w_y} \\ \mathrm{w_z} & 0 & -\mathrm{w_x} \\ -\mathrm{w_y} & \mathrm{w_x} & 0 \end{bmatrix}$$

$$\varphi = \|\mathrm{w}\|$$

Step 4. Condition rotational matrix

As we know, the calculation of the extrinsic parameters will result in rotational matrix R which is not orthonormal. The following method was applied:

Let Q be the computed rotation matrix. For a given Q, we to find:

$$min_R\|R - Q\|_F^2$$

The solution is $[U\ D\ V] = SVD(Q)$ and $R = UV^T$

Result demonstration

Step 1. Edge detection, hough line fitting and corner detection result

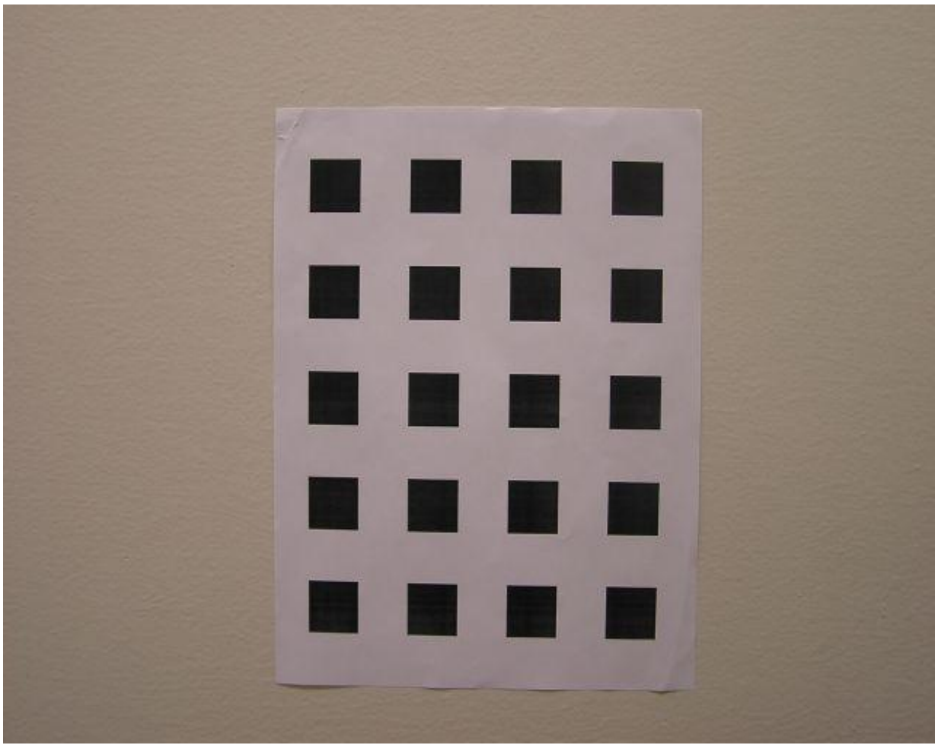


Figure 1.1 Original image

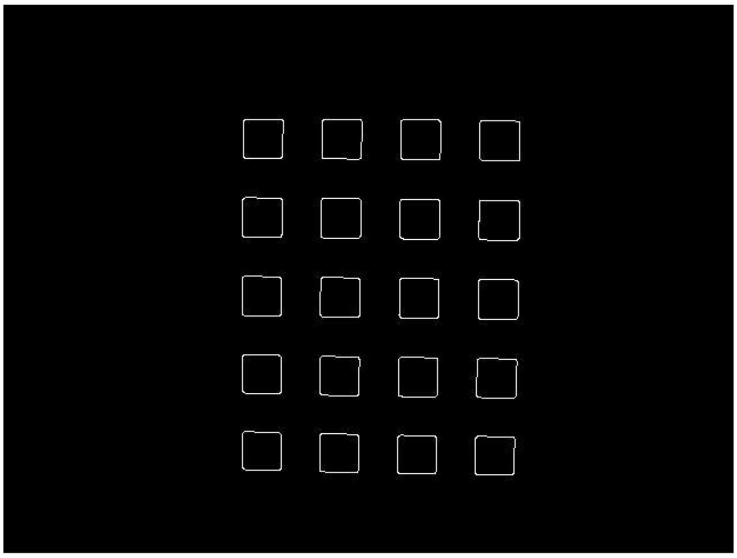


Figure 1.2 Canny edge detection

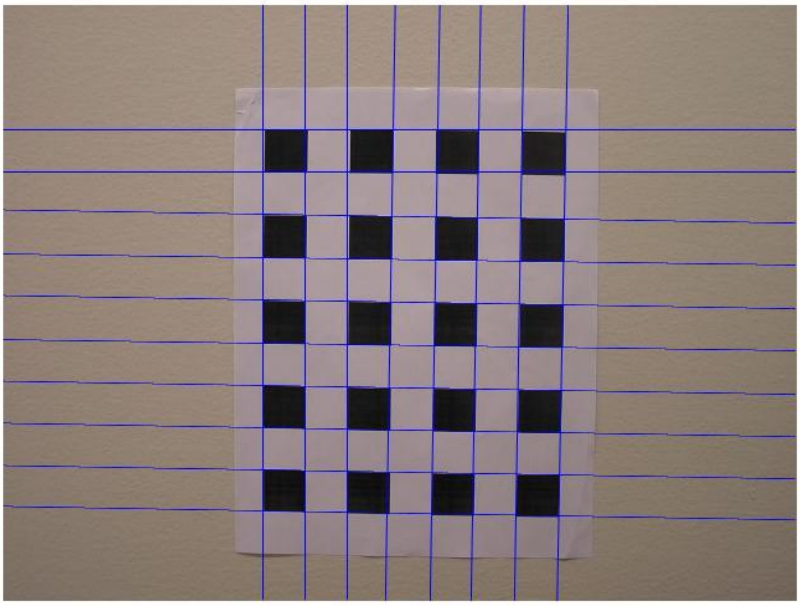


Figure 1.3 Hough line fitting

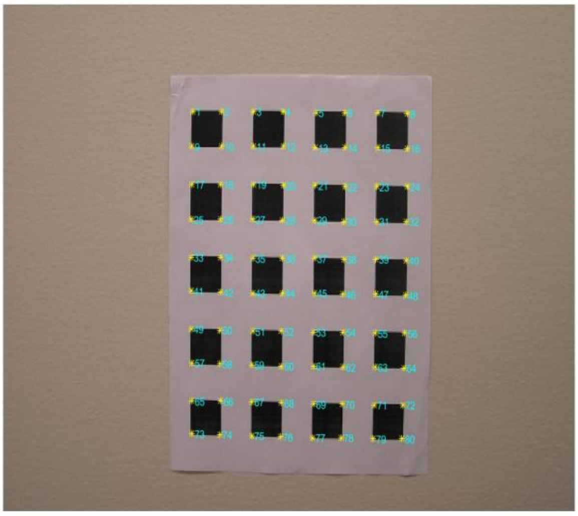


Figure 1.4 Detected corners with labels

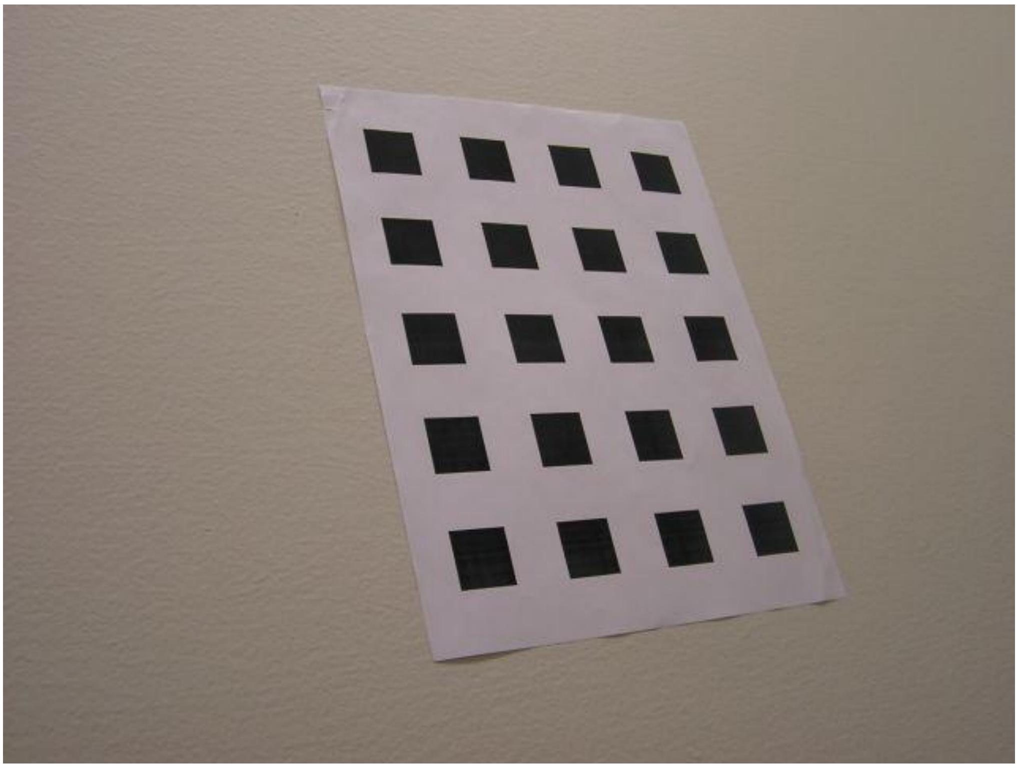


Figure 2.1 Original image

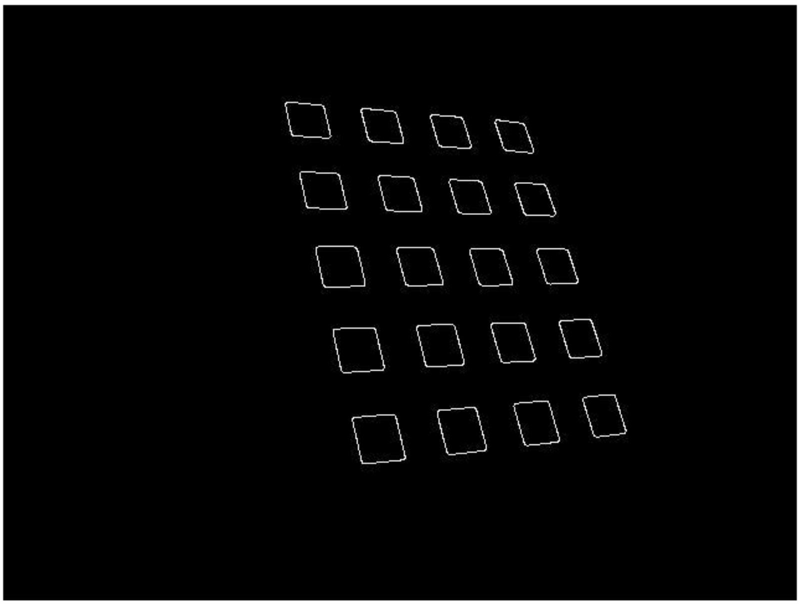


Figure 2.2 Canny edge detection

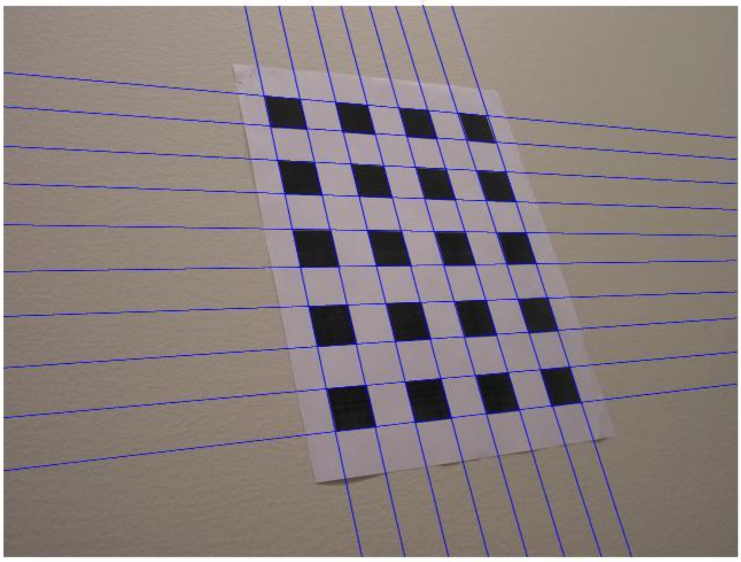


Figure 2.3 Hough line fitting

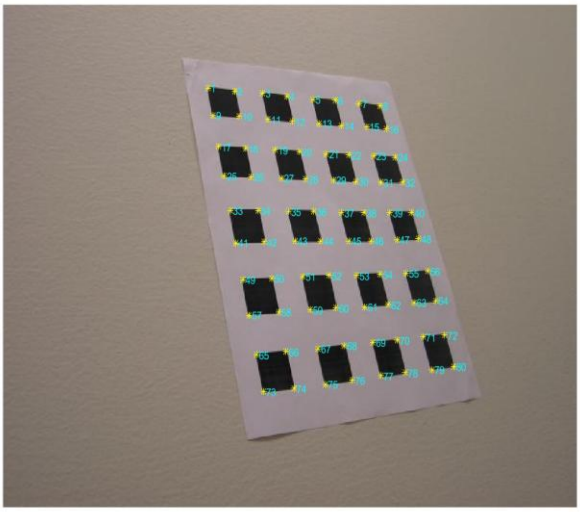


Figure 2.4 Detected corners with labels

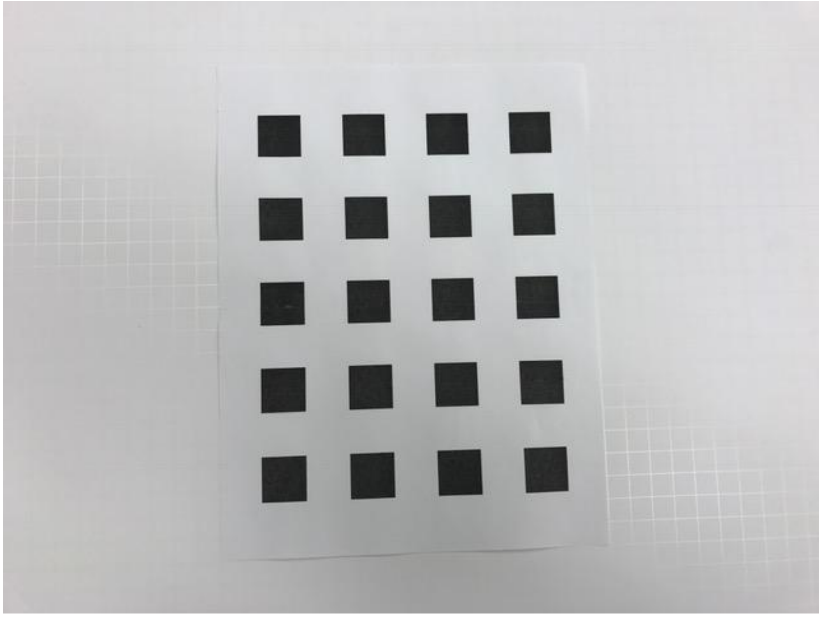


Figure 3.1 Original image

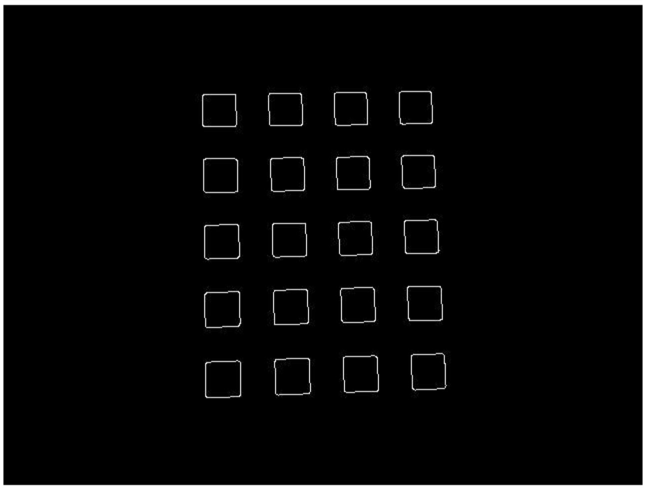


Figure 3.2 Canny edge detection

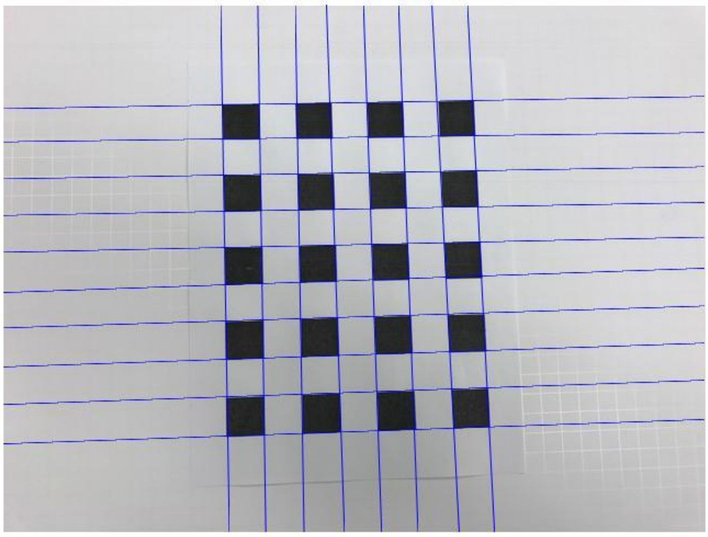


Figure 3.3 Hough line fitting

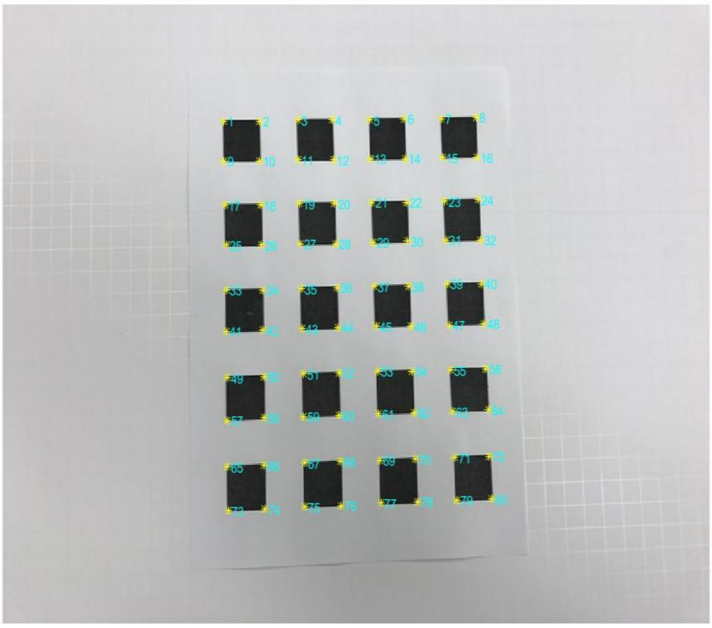


Figure 3.4 Detected corners with labels

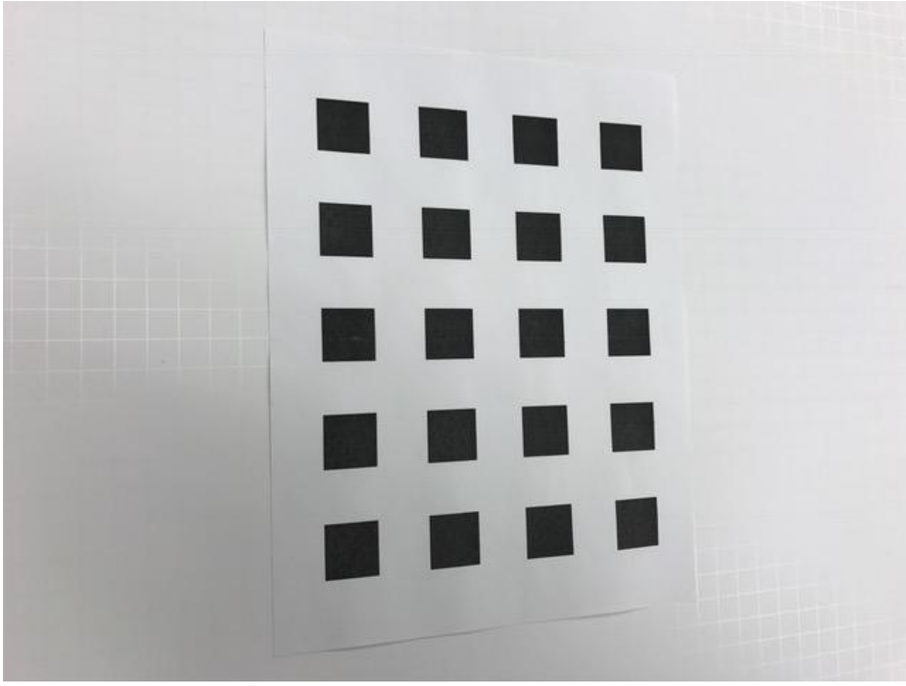


Figure 4.1 Original image

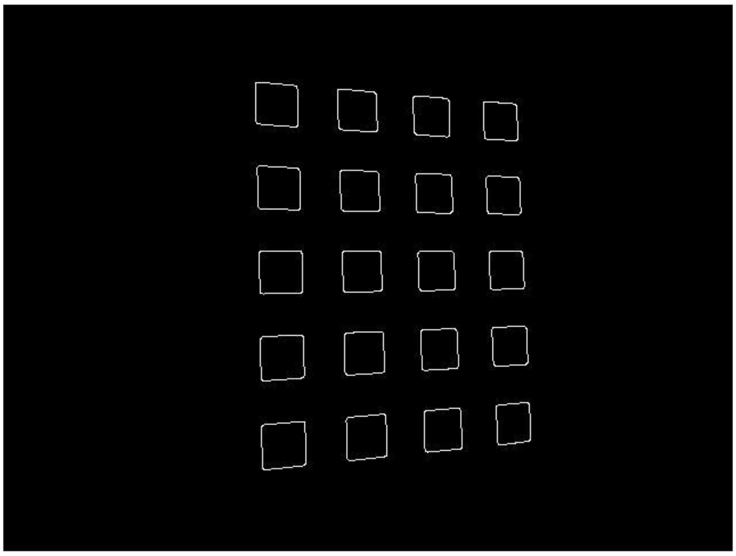


Figure 4.2 Canny edge detection

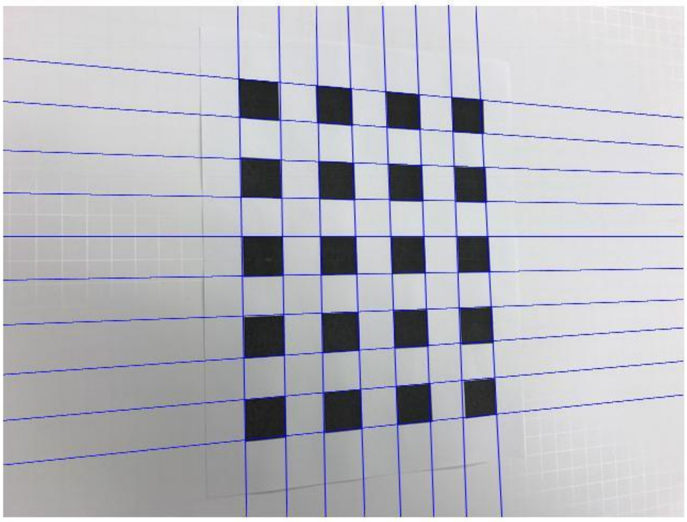


Figure 4.3 Hough line fitting

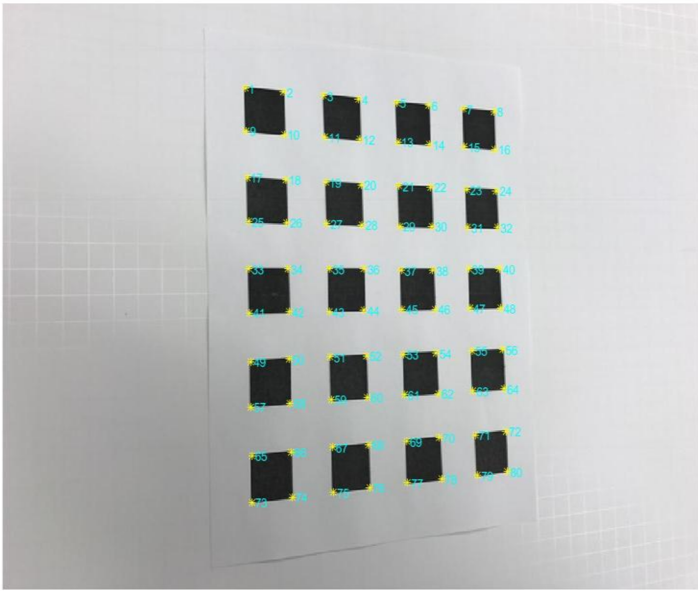


Figure 4.4 Detected corners with labels

Step 2. Reprojected corners on to fixed image(3 images for each dataset)

- Blue points indicate the estimated points, yellow points indicate the true points.
- Average error was calculated by computing the mean of the absolute value between each estimated and ground truth pair points.
- Variance error was calculated by computing the variance of the absolute value between each estimated and ground truth pair points

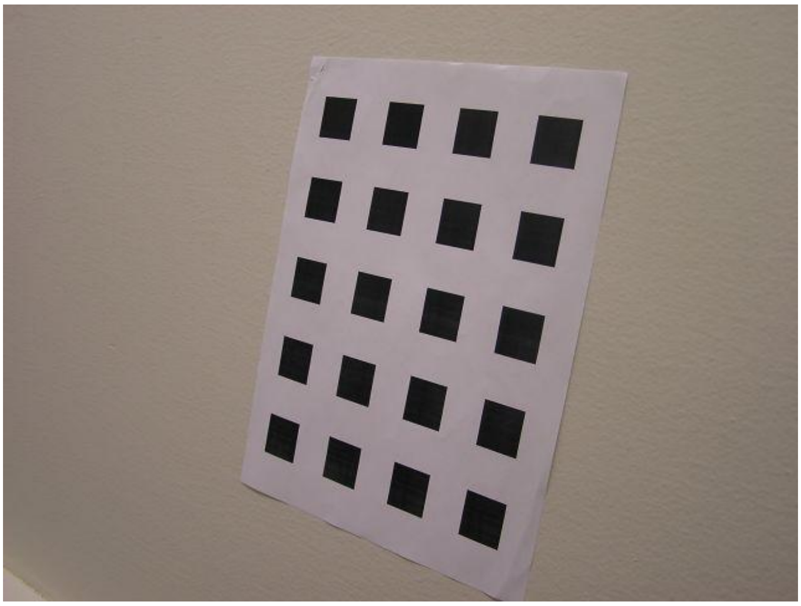


Figure 5.1 Fixed image chosen from dataset 1

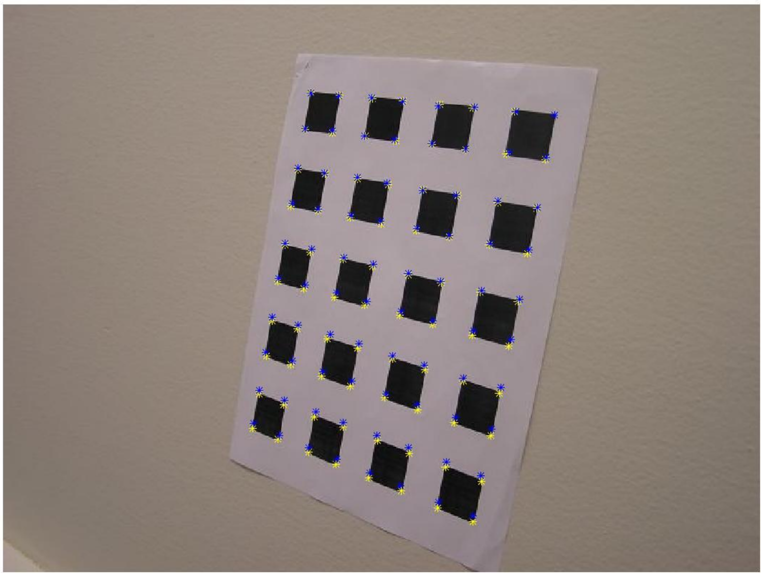


Figure 5.2 Estimated points from Pic_12 onto Pic_1

Average error = 32.2756, Variance error = 664.0059

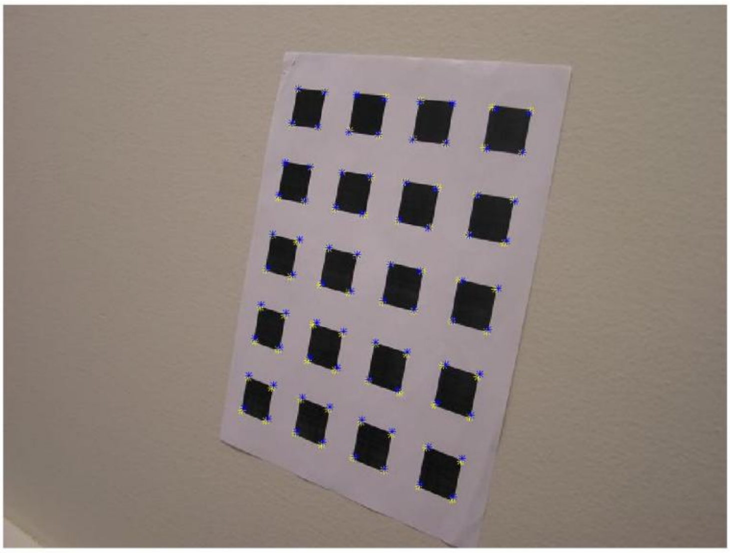


Figure 5.3 Estimated points from Pic_10 onto Pic_1

Average error = 37.2013, Variance error = 899.0712

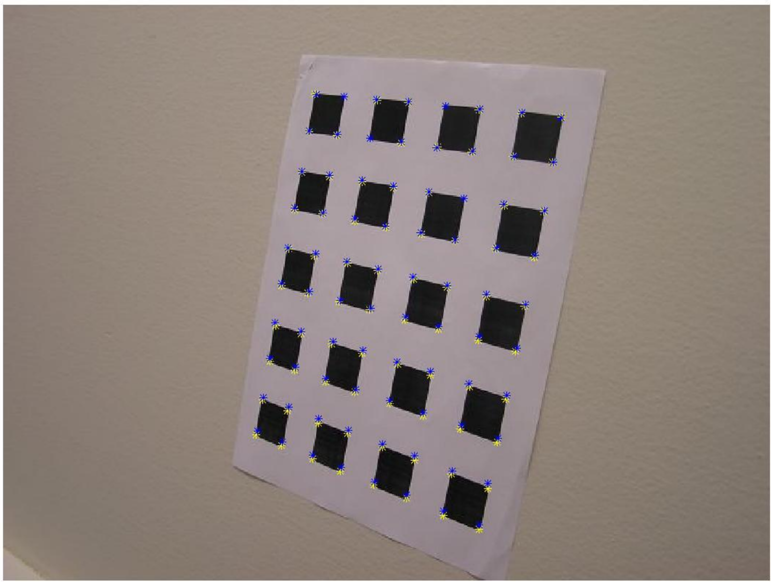


Figure 5.4 Estimated points from Pic_11 onto Pic_1

Average error = 16.6008, Variance error = 191.3762

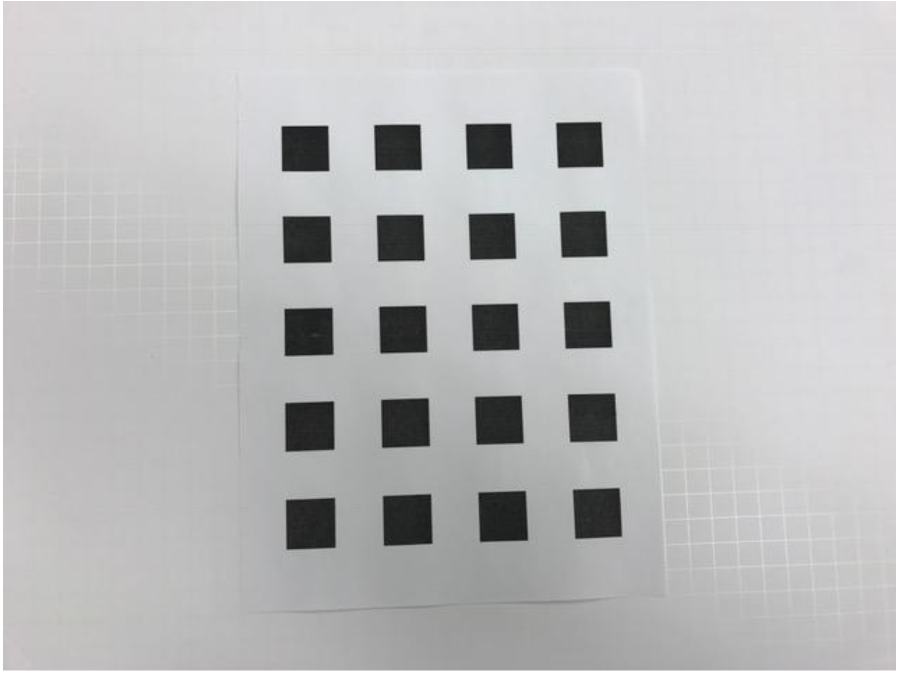


Figure 6.1 Fixed image chosen from dataset 2(IMG_1236)

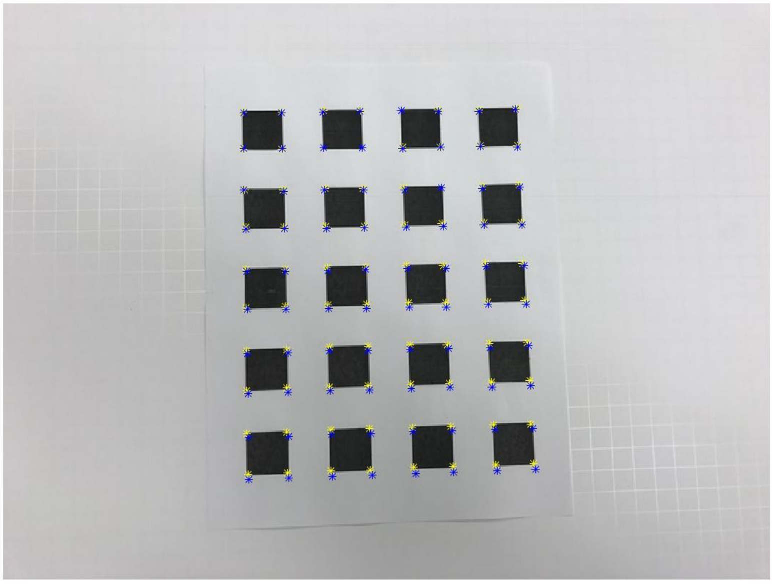


Figure 6.2 Estimated points from IMG_1245 onto IMG_1236

Average error = 33.7478, Variance error:720.5106

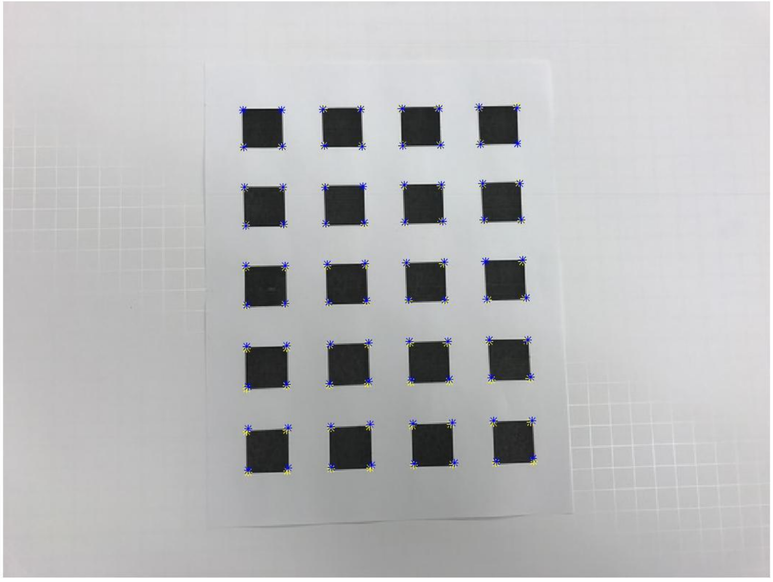


Figure 6.3 Estimated points from IMG_1252 onto IMG_1236

Average error = 21.5641 ,Variance error = 122.3340

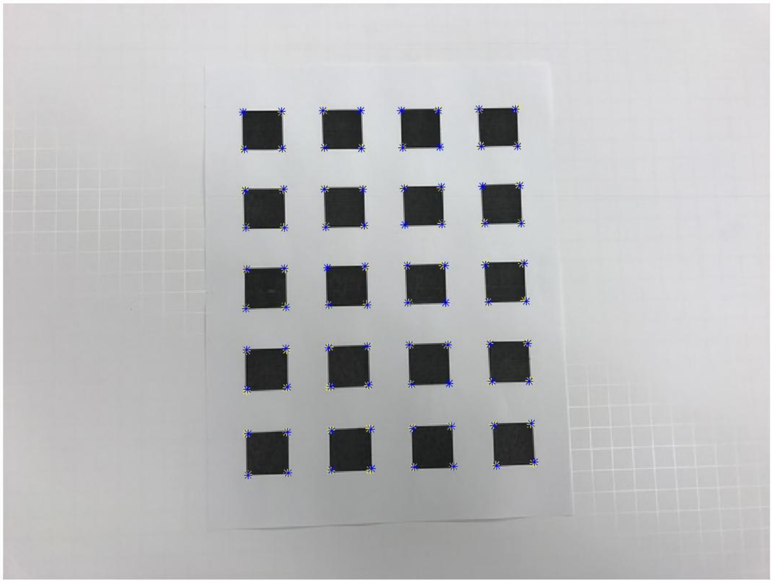
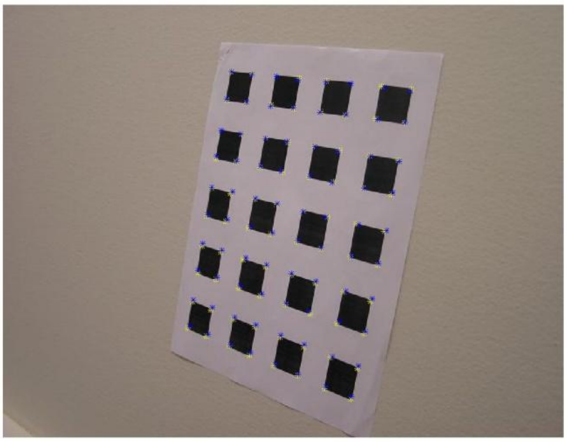


Figure 6.4 Estimated points from IMG_1253 onto IMG_1236

Average error = 13.3931 ,Variance error = 98.1084

Step 3. Demonstration of refinement through LM algorithm(2 images for each dataset)



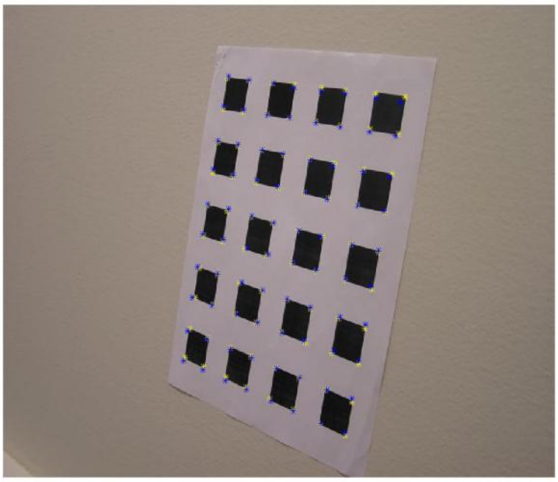
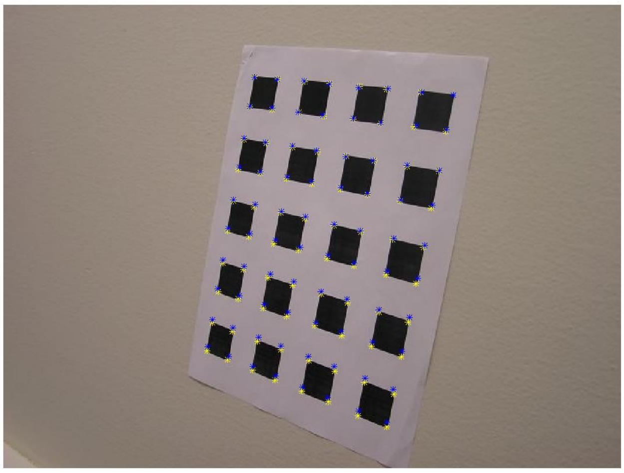


Figure 7.1 Before LM refinement(Top) and after LM refinement(Bottom) points from Pic_10 onto Pic_1

Before: Average error = 37.2013, Variance error = 899.0712

After: Average error = 34.2406, Variance error = 805.8043



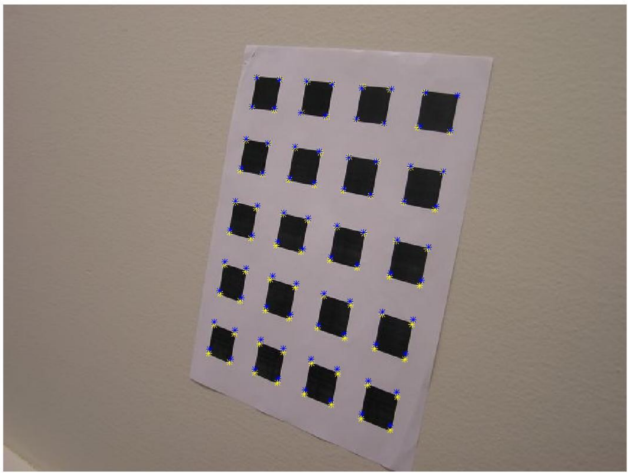
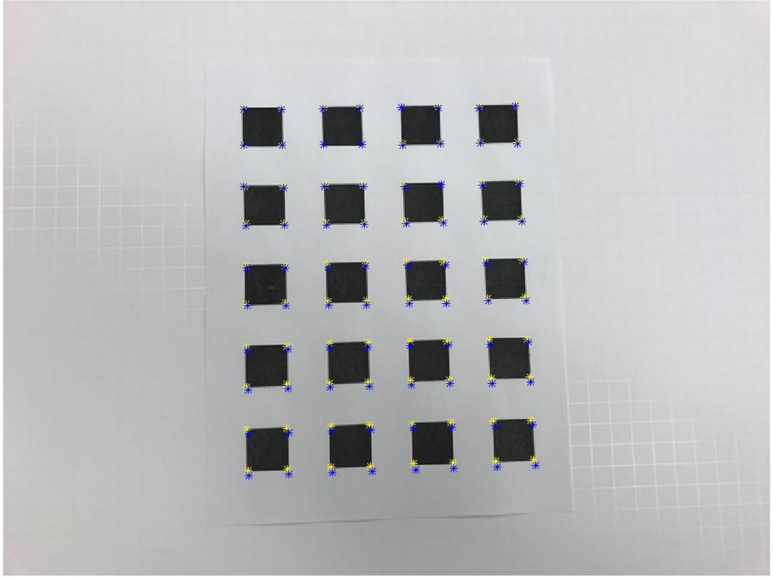


Figure 7.2 Before LM refinement(Top) and after LM refinement(Bottom) points from Pic_12 onto Pic_1

Before: Average error = 32.2756, Variance error = 664.0059

After: Average error = 30.8027, Variance error = 647.1240



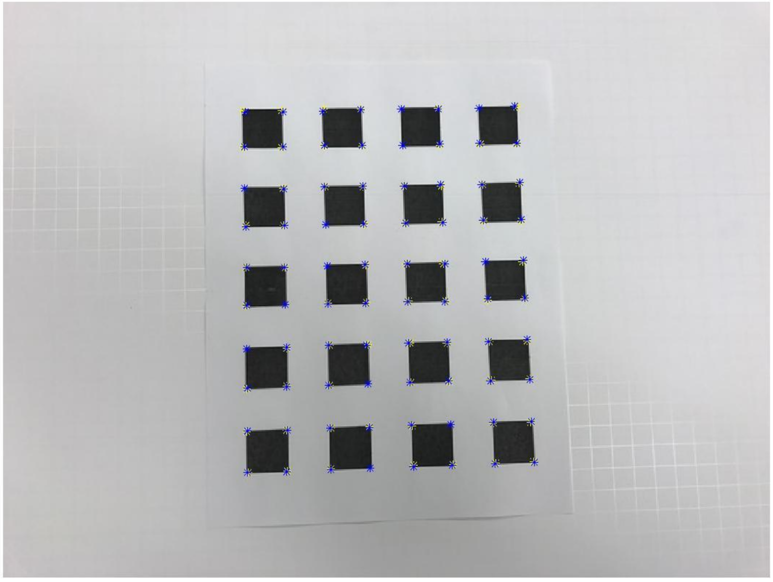
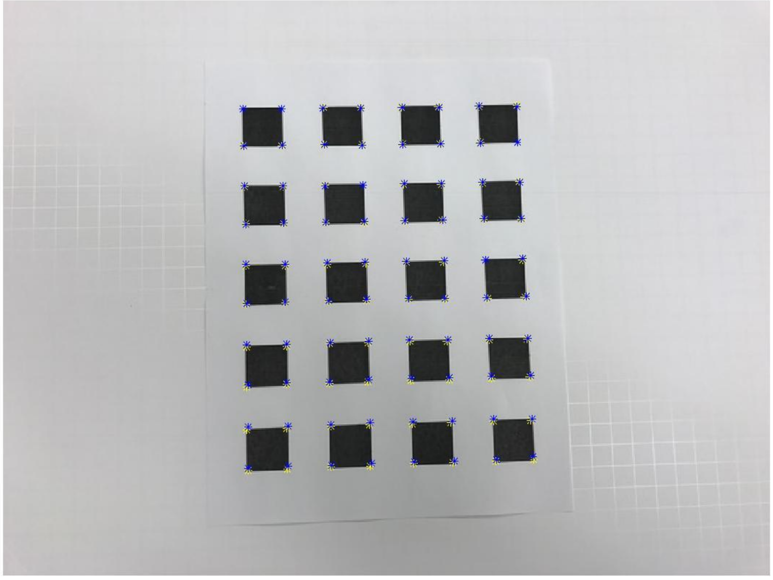


Figure 7.3 Before LM refinement(Top) and after LM refinement(Bottom) points from IMG_1245 onto IMG_1236

Before: Average error = 33.7478, Variance error:720.5106

After: Average error = 33.525, Variance error = 663.6003



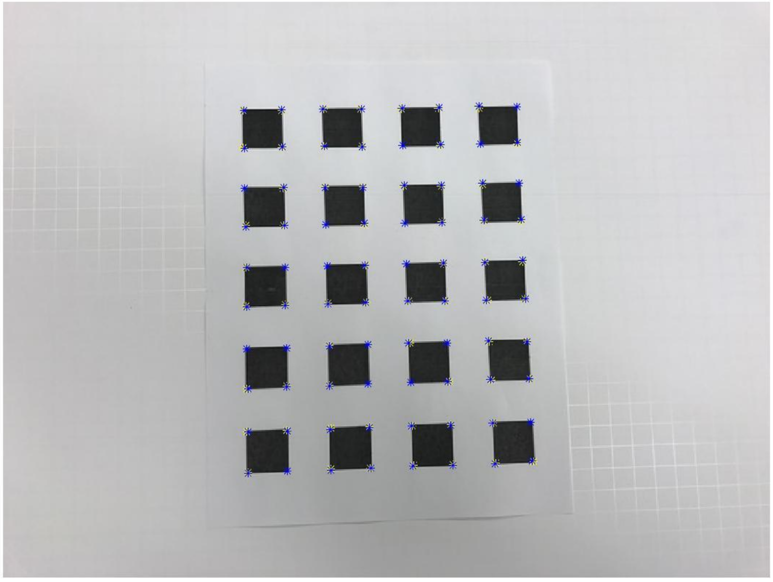


Figure 7.4 Before LM refinement(Top) and after LM refinement(Bottom) points from IMG_1252 onto IMG_1236

Before: Average error = 21.5641 ,Variance error = 122.3340

After: Average error = 19.7758 ,Variance error = 113.5433

Step 4. Intrinsic matrix and extrinsic matrix demonstration of 4 different matrix

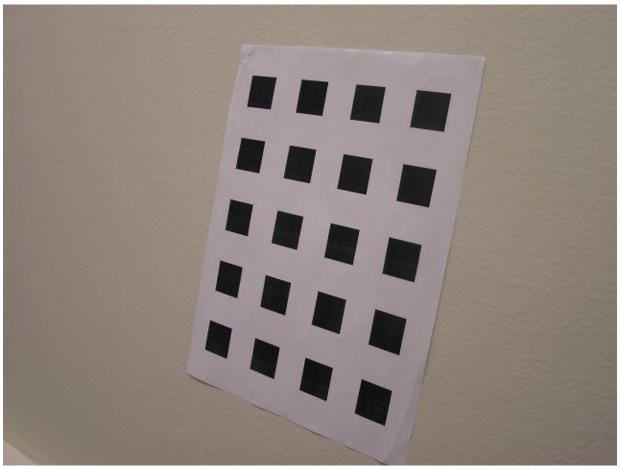
1. Intrinsic matrix for dataset1 before LM refinement:

717.691	1.7198	318.896
0	718.72	236.951
0	0	1

2. Intrinsic matrix for dataset1 after LM refinement:

872.016	1.601	319.64
0	874.55	236.13
0	0	1

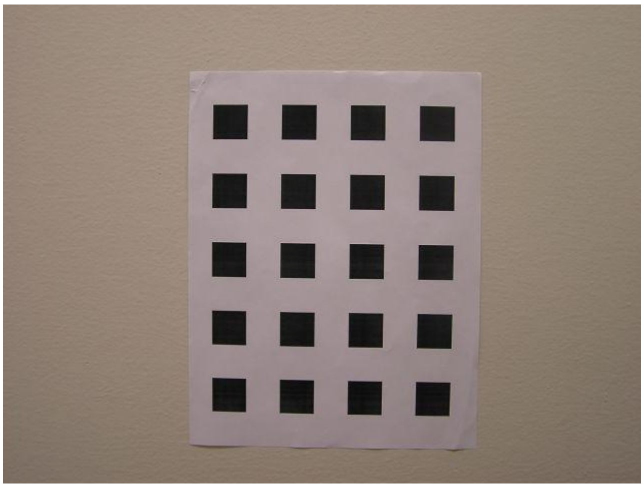
3. Extrinsic matrix for Pic_1 from dataset1



[R|t] =

0.788	-0.1837	0.5866	-35.405
0.1997	0.9791	0.0381	-100.02
-0.581	0.0871	0.8089	441.52

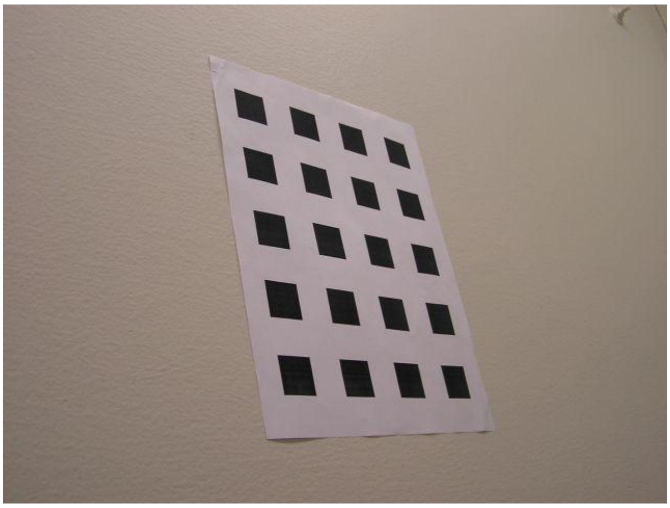
4. Extrinsic matrix for Pic_11 from dataset1



[R|t] =

0.999	-0.0118	0.0337	-62.03
0.0133	0.9989	-0.0432	-78.79
-0.0332	0.04365	0.9985	416.48

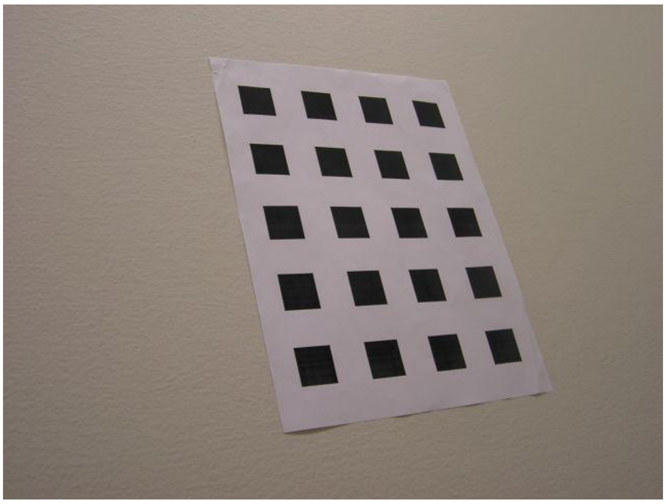
5. Extrinsic matrix for Pic_10 from dataset1



[R|t] =

0.756	0.1817	-0.628	-56.91
0.135	0.8963	0.422	-92.72
0.639	-0.4045	0.6534	428.19

6. Extrinsic matrix for Pic_12 from dataset1



[R|t] =

0.8855	0.219	-0.4098	-53.57
-0.0175	0.897	0.4416	-93.87
0.4644	-0.383	0.7981	427.11