Self-calibration based on robust point matching and fundamental matrix estimation

ECSE 6650 Computer Vision Final Project
Bhavani Shankar Yanamadala, Lei Zhang, Xiaoli Zhang

Self camera calibration generally is not as effective as other calibration methods because it does not have the inherent 3D information embedded as compared to the conventional camera calibration methods. Among many factors, the accuracy of camera calibration technique depends on three major steps, they are:

- Points used for calibration
- Fundamental matrix estimation
- The accuracy of 3D information used for camera rotation/translation.

We make an effort to improve the performance of automated self calibration system by using more robust techniques for the defining steps.

1. Computation of the Fundamental Matrix using RANSAC

1.1 Harris Corner detector

Corners are the intersections of two edges of sufficiently different orientations. They are important two dimensional image features to represent object shapes. Corners are stable across image sequences and useful in image matching for stereo and object tracking for motion, therefore playing an important role in matching, pattern recognition, robotics, and measuration.

Corners are generally located in the region with large intensity variations in every direction. The instrument to detect corners lies in image derivatives. Let I_x and I_y be image gradients in horizontal and vertical directions, we can define a matrix *C* as

$$C = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \tag{1.1}$$

where the sum are taken over a neighborhood of the pixel in consideration. This matrix characterizes the structure of the image gray level patterns. In fart, the geometric interpretation of the gray levels is encoded in the eigenvector and eigenvalues of the matrix. *C* is symmetric and it has two nonnegative eigenvalues λ_1 and λ_2 . The eigenvectors of *C* encode the directions, while the eigenvalues encode the variational strength. A corner is detected if the minimum of the two eigenvalues is larger than a threshold. An alternative for detecting corners is: if $\det(C) - k \cdot trace(C^2)$ is larger than a threshold, where *k* is a small number (0.04). This method is called *Harris Corner Detector*, which is good for detecting corners with orthogonal edges.

1.2 Point Matching

Correlation-based method is utilized to establish the correspondence between the left and right image points. The principle of this method is to find two corresponding points based on the intensity distributions of their neighborhoods. The similarity between two neighborhoods is measured by their cross-correlation. The underlying assumptions include:

- Corresponding image regions are similar
- Point and distant or single light source
- Corresponding points are visible from both viewpoints

Given an image point on the left image, the task is to locate another point in a specified region on the right image that is maximally correlated with the one on the left. For each left image pixel, its correlation with a right image pixel is determined by using a small correlation window of fixed size.

$$c(\vec{d}) = \sum_{k=-W}^{W}\sum_{l=-W}^{W}\Psi(I_l(i+k, j+l), I_r(i+k-d_1, j+l-d_2)) \tag{1.2}$$

□where (i, j) are the coordinates of the left image pixel.
□$\vec{d} = (d_1, d_2)^T$ is the relative displacement between the left and right image pixels.
□$2W+1$ is the width of the correlation window.
□I_l and I_r are the intensities of the left and right image pixels respectively.
□$\Psi(u,v) = -(u-v)^2$ is the SSD correlation function.

By moving the correlation window in certain region of the right image, the corresponding point is set as the pixel that has the highest correlation value.

1.3 Computing Fundamental Matrix F and Essential Matrix E

Both the fundamental and essential matrices could completely describe the geometric relationship between corresponding points of a stereo pair of cameras. We derived the essential and fundamental followed by the eight-point algorithm.

The Essential matrix contains five parameters (three for rotation and two for the direction of translation). Given the relative rotation R and translation T matrices, the skew symmetric singular matrix S is

$$S = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{1.3}$$

From the epipolar geometry, we have

$$P_l^T E P_r = 0 \tag{1.4}$$

where P_l , P_r represent the 3D coordinates of P in the left and right camera coordinate systems respectively. $E = S^t R$ is the essential matrix.

Let U_l, U_r be the homogeneous pixel coordinates of point P in the left and right images. From full perspective projection, we know

$$\begin{aligned} \lambda_1 U_l &= W_l P_l \\ \lambda_r U_r &= W_r P_r \end{aligned} \tag{1.5}$$

where W_l and W_r are two 3×3 matrices involving the intrinsic left and right camara parameters. Substitute the above equations into the essential matrix equation (1.4) yields

$$U_l^T F U_r = 0 \tag{1.6}$$

where fundamental matrix $F = W_l^{-T} E W_r^{-1}$. Knowing F, we can compute the essential matrix $E = W_l^T F W_r$.

For each pair of matching 2D points $U_l = (c_l, r_l, 1)^T$, $U_r = (c_r, r_r, 1)^T$, the fundamental equation ,$U_l^T F U_r = 0$, offers a linear equation of the 9 unknows

$$\begin{pmatrix} c_l \\ r_l \\ 1 \end{pmatrix}^T \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} c_r \\ r_r \\ 1 \end{pmatrix} = 0 \Rightarrow A^{N \times 9} f^{9 \times 1} = 0 \tag{1.7}$$

where f is a nine-vector containing the entries of the matrix F, and

$$A = \begin{pmatrix} c_{l_1} c_{r_1} & c_{l_1} r_{r_1} & c_{l_1} & r_{l_1} c_{r_1} & r_{l_1} r_{r_1} & r_{l_1} & c_{r_1} & r_{r_1} & 1 \\ . & . & . & . & . & . & . & . & . \\ c_{l_n} c_{r_n} & c_{l_n} r_{r_n} & c_{l_n} & r_{l_n} c_{r_n} & r_{l_n} r_{r_n} & r_{l_n} & c_{r_n} & r_{r_n} & 1 \end{pmatrix} \tag{1.8}$$

Given a minimum of 8 corresponding pairs (cannot be co-planar), F can be solved up to a scale factor. The solution for f is given by the null vector of A. For computing this, we evaluate the SVD of A and the eigenvector corresponding to the smallest eigenvalue of A is f. Ideally the rank of the matrix F should be 2, but in most cases, due to inaccuracies, the rank of the matrix so obtained is 3. To make F singular, we do the SVD of F and then set the value of the smallest element in the diagonal to zero. This algorithm is called the e*ight-point algorithm.*

The linear solution from eight-point is often not accurate and it can be improved with a non-linear optimization method. This non-linear method minimizes function

$$\sum_{i=1}^{n} [(U_{l_i}^t F U_{r_i})^2 + ((U_{r_i}^t F^t U_{l_i})^2)] \tag{1.9}$$

subject to rank(F)=2.

Normally, we could use normalization of the coordinates of the corresponding points to decrease the influence of numerical instability. This is because if the coordinates of points are let's say in the order of 100 or more then, the fundamental matrix obtained becomes a very ill conditioned matrix. Normalization procedure is done to improve the conditioning of the matrix and this is often helpful in getting an accurate estimate of F which can be used to obtain E and the epipolar geometry. We used the centroid of the sets of points in left image plane and right image plane to do normalization. X_L, Y_L (X_R, Y_R) denote the x and y coordinates of the centroid of the left(right) image plane. d_L (d_R) denotes the average distance of the points in left(right) image plane with respect to their centroid. We used the following matrix H_L, H_R to normalize the coordinate of points in image plane.

$$H_L = \begin{bmatrix} 1/d_L & 0 & -X_L/d_L \\ 0 & 1/d_L & -Y_L/d_L \\ 0 & 0 & 1 \end{bmatrix}$$
$$H_R = \begin{bmatrix} 1/d_R & 0 & -X_R/d_R \\ 0 & 1/d_R & -Y_R/d_R \\ 0 & 0 & 1 \end{bmatrix} \qquad (1.10)$$

We used the normalized coordinates of image points in eight-point method to calculate the fundamental matrix F'. The normalization is done to both 2D and 3D coordinates in such a way that the average distance of each point from the origin in 2D case is $\sqrt{2}$ and the average distance of a 3D point from the origin is $\sqrt{3}$ [1,2]. The normalization factors are recorded and the image coordinates obtained at the end are again renormalized using the normalization factors recorded above. The actual fundamental matrix F could be calculated by the inverse process of normalization $F = H_R^T F^{'} H_L$.

1.4 Robust estimation of F and matching points using RANSAC

The linear method is sensitive to image errors and outliers. Eight-point method will corrupt due to the outliers resulting from mismatching. However, by using randomly selected subsets during the implementation of linear method, we could end up with correct solution. RANSAC is an algorithm for robust model fitting by selecting a minimum sample set required for the model. Models containing outliers are rejected since they don't generate sufficient consensus. The assumptions in RANSAC are

- The model can be estimated from *K* data items;
- There are *N* (*N*>*K*) data items in total.

Assume that the whole set of data may contain up to a fraction ε of outliers, then the probability that all K data in a subset are good is $(1-\varepsilon)^K$, and the probability that all *s* different subsets will still contain at least one or more outliers is $(1-(1-\varepsilon)^K)^s$. So the probability that at least one random subset has no outliers is given by

$$P = 1-(1-(1-\varepsilon)^K)^s$$

Thus the number of iterations needed is computed as

$$s = \frac{\ln(1-P)}{\ln(1-(1-\varepsilon)^K)} \qquad (1.11)$$

In this project, we aim to find an optimal subset to compute the fundamental matrix *F* using RANSAC. Each time a minimal sample (8 pairs) is selected from the matches based on SSD from which a tentative *F* is calculated from eight-point method. The epipolar lines for each pair of matching points can be derived from *F*. Those matches with the distance from the corresponding epipolar lines smaller than a threshold are kept as inliers. The process is iterated until a sufficient number of samples have been taken. The *F* is calculated from the optimal subset with the largest number of inliers. Setting *F* as initial value, we continue to improve *F* with a non-linear optimization method, then we can compute the epipolar geometry to guide more additional matches. At this point only feature points being in epipolar correspondence should be considered for matching. The final *F* is derived from all correct matches.

The whole algorithm for this part of project can be described as follows:

- Extract feature points in left and right images.
- Construct a set of potential matches using SSD.
- Implement RANSAC to randomly select subset to compute F, determine inliers, and find the optimal subset with the largest number of inliers to refine *F*.
- Improve *F* with non-linear minimization.
- Look for additional matches based on the epipolar geometry.
- Derive F from all correct matches.

4. Experimental results

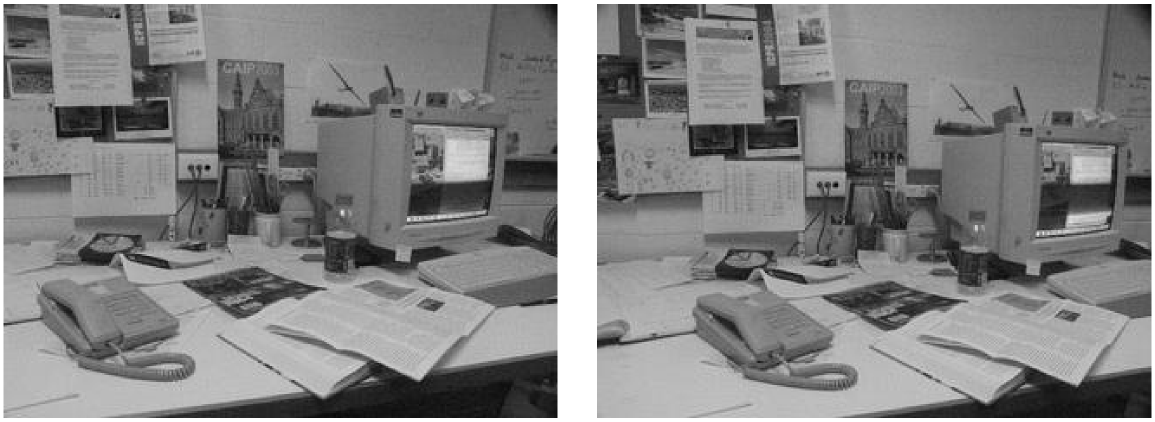

Figure 1.1. Left and right desk image

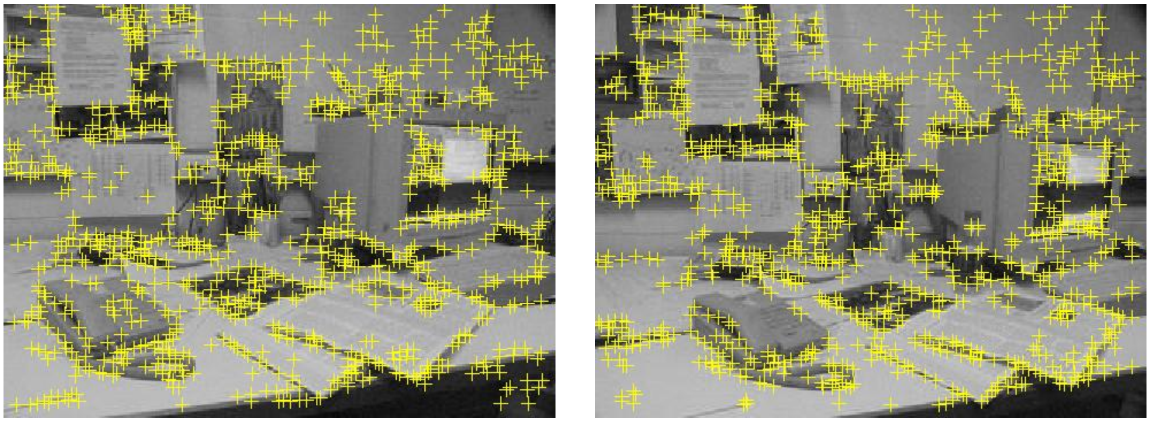

Figure 1.2 Corner feature points in left and right image from Harris corner detector

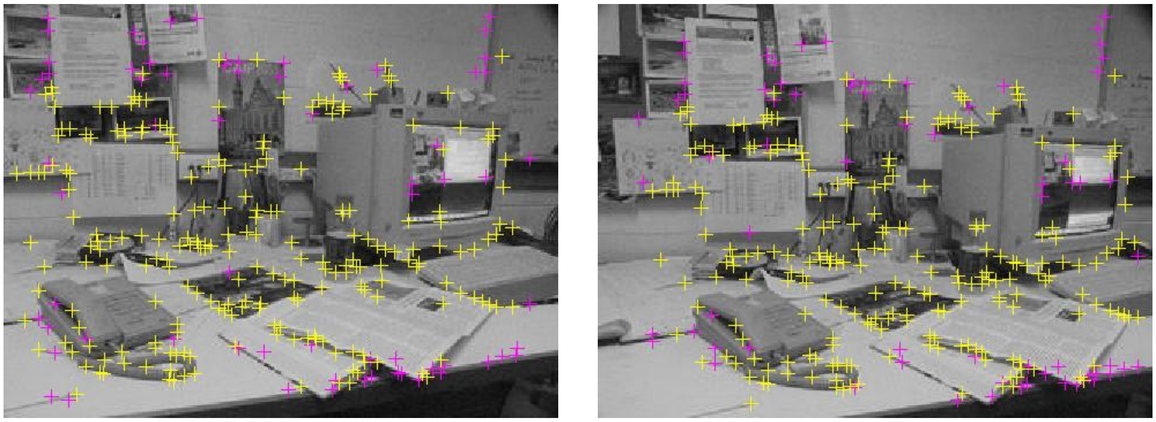

Figure 1.3a Matching points using RANSAC

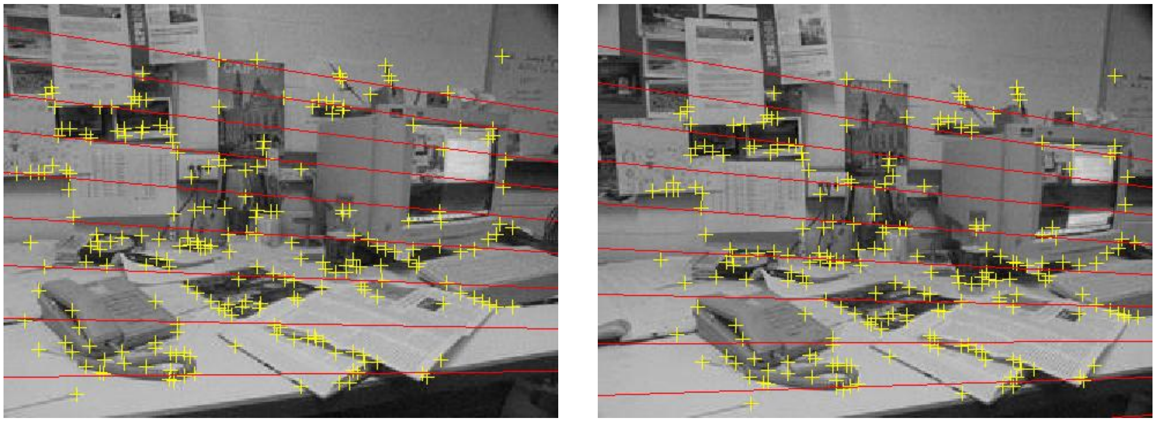

Figure 1.3b Epipolar lines using computed *F* from RANSAC

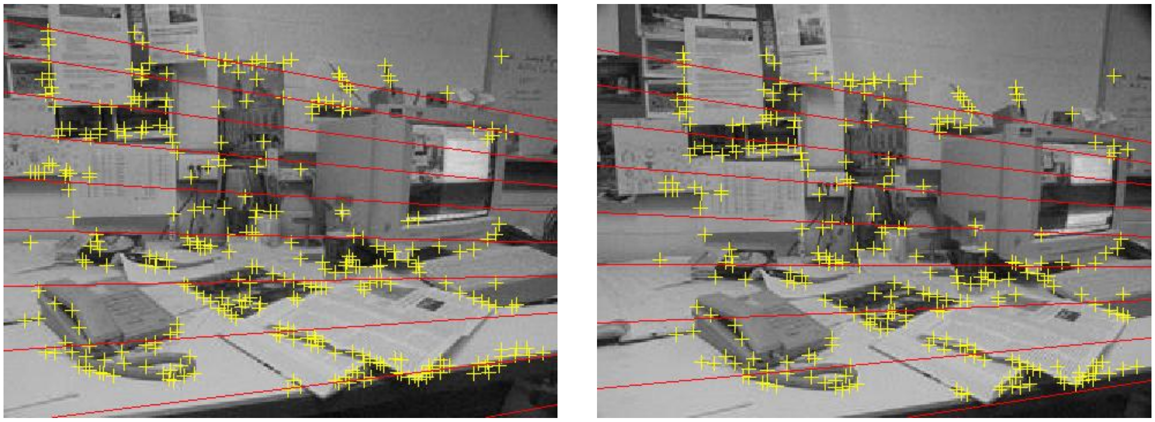

Figure 1.4 Matching points with improved *F*

We first use Harris corner detector to extract the corner features in left image and right image (Figure 1.1). Those corners shown in Figure 1.2 are candidate feature points for point matching. We can observe that corners at the telephone handle is not detected since Harris corner detector is good for detecting corners with orthogonal edges. The matching points based on SSD are shown in Figure 1.3. They include both magenta and yellow marked points. Magenta points are rejected by RANSAC since they don't satisfy epipolar geometry. The inliers are those yellow marked matches in Figure 1.3b with the distance from the corresponding epipolar lines smaller than a threshold (1 pixel). These inliers are used to compute the fundamental matrix *F* from which we draw the epipolar lines in Figure 1.3b. Using improved *F* by a non-linear optimization method, we can compute the epipolar geometry to guide more additional matches. For a feature point in left image, only the corners in right image that are within a small range (1 pixel) around the corresponding epipolar line are considered for matching. The final *F* is derived from all correct matches. The matching feature points and epipolar line in both left and right images are illustrated in Figure 1.4. Compared with the inliers from RANSAC (Figure 1.3b), more good candidates of matching points are obtained to guarantee a correct and stable *F* matrix.

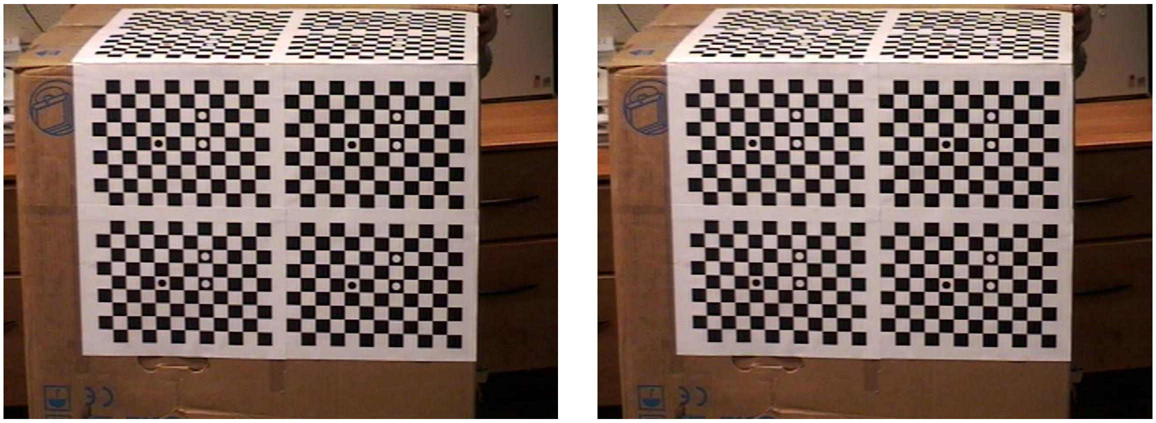

Figure 1.5 Left and right camera calibration board image

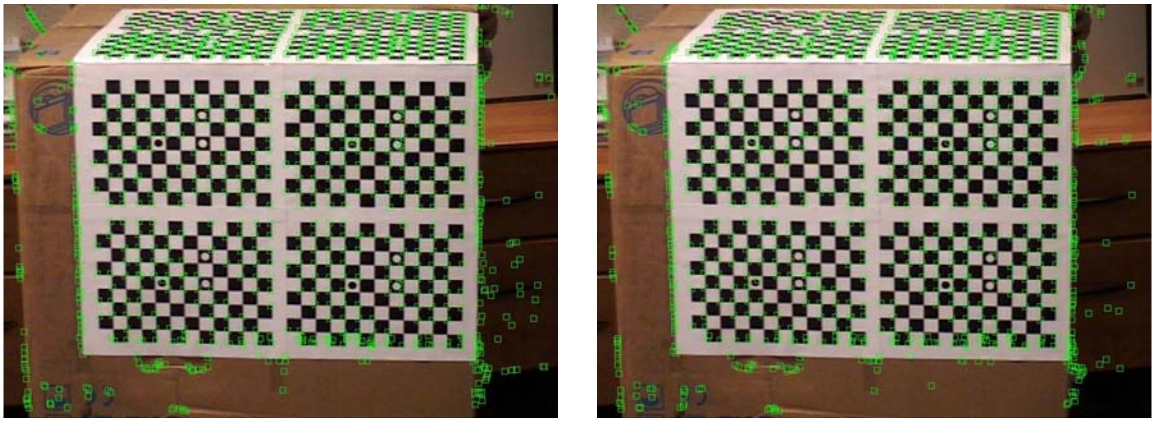

Figure 1.6 Corner feature points in left and right image from Harris corner detector

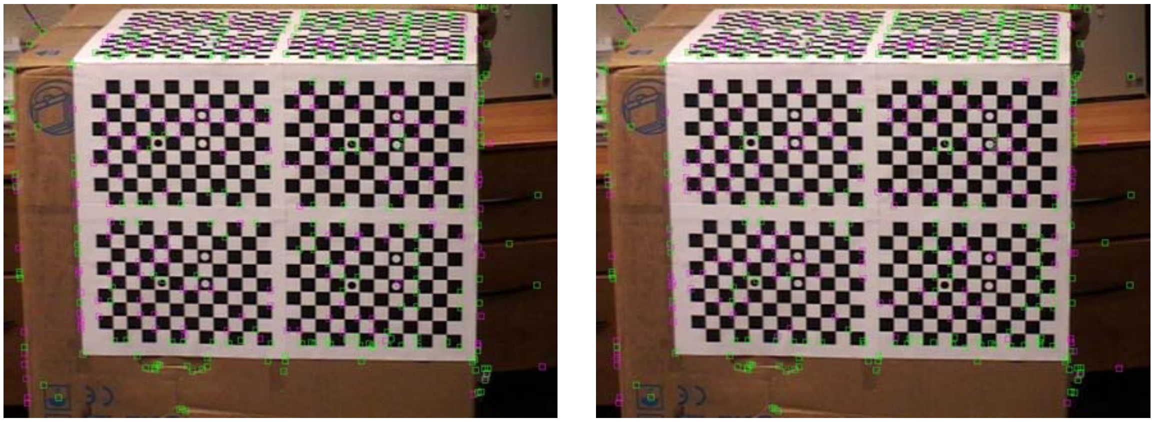

Figure 1.7a Matching points using RANSAC

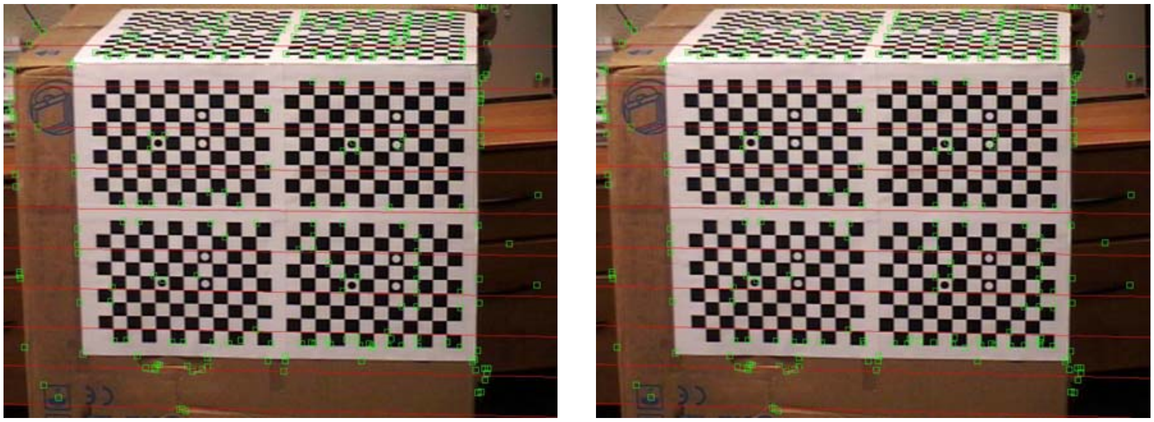

Figure 1.7b Using computed *F* from RANSAC

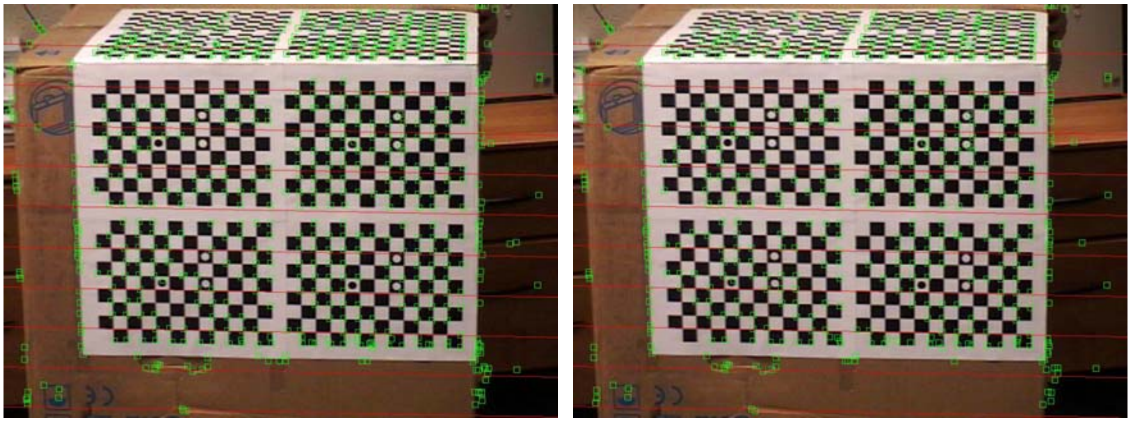

Figure 1.8 Matching points with improved *F*

We implement the same procedure on another camera calibration board images. The results are shown in Figure 1.6, 1.7, and 1.8. Due to the similar grid in camera calibration board, our correlation method gives many wrong matches. Figure 1.7a shows that RANSAC successfully rejects those mismatches (magenta squares) and retain most of correct match points (magenta squares) in left and right images. Based on the epipolar geometry derived from improved *F* matrix, we add more matching features. However, it also adds some mismatches.

Table 1.1 All the Parameters set in the algorithm

Image	Corner threshold	SSD window	RANSAC						Nonlinear
			ε	p	s	Inlier distance threshold	Inliers	Error	Error
Desk	100	11×11	0.5	0.99	8	1.0 pixel	241/295	0.21	0.18
Board	180	9×9	0.6	0.99	8	1.0 pixel	246/443	0.15	0.17

The parameters in the whole algorithm are listed in Table 1.1. The error is the nonlinear minimization error, which is calculated from the nonlinear objective function (formula 1.9) based on the matching points and *F* matrix. For camera calibration board, the improved F matrix gives higher error, which means that some mismatches appear. The fundamental matrices we obtained for these two sequences of images are as follows (Table 1.2).

Table 1.2 *F* matrix and number of matching points of two images

Image	Matching points	*F* matrix
Desk	316	$\begin{pmatrix} 0.00000039756912 & -0.00010344066759 & 0.01025149810761 \\ 0.00009925074566 & 0.00000742648782 & -0.09381208755582 \\ -0.01121827458737 & 0.09274944097038 & 0.99118299976579 \end{pmatrix}$
Board	521	$\begin{pmatrix} 0.00000001980648 & -0.00000576528799 & -0.01777073875812 \\ 0.00000480980401 & -0.00000189261424 & 0.96613911406288 \\ 0.01801435695895 & -0.96497133090921 & 0.06737150198919 \end{pmatrix}$

A sequence of *F* matrices is computed for mutually correlated pairs in a sequence of camera calibration images which are used by camera calibration in section 3.

2. Fundamental matrix estimation

The basic theory of fundamental matrix estimation is discussed in the previous sections, the normalized 8 point method and the 8 point methods is discussed, In this section we discuss implementation of another algorithm which can be used to improve the results of the existing algorithms.

2.1 Least squares regression

Consider the following equation

$$a_i = A_i^T f + \varepsilon \qquad i = 1 \ldots n, \tag{2.1}$$

In the above equation, a is the measured scalar with error ε, which is a gaussian error having standard deviation σ. Here the matrix *A* is a p dimensional matrix that is known and the matrix *f* is a *p* dimensional vector to be estimated. Then if the value *n* is greater than *p* the set of equations is overdetermined. The noise, however renders these equations inconsistent. Thus the least squares method is used to obtain the solution, here we obtain the pseudo inverse of *A* and then obtain the solutions for the values of *f*. We obtain the value of *f* by minimizing *f* in the direction of a particular coordinate, thus this becomes unstable sometimes as can be seen in figure 2.1(b)

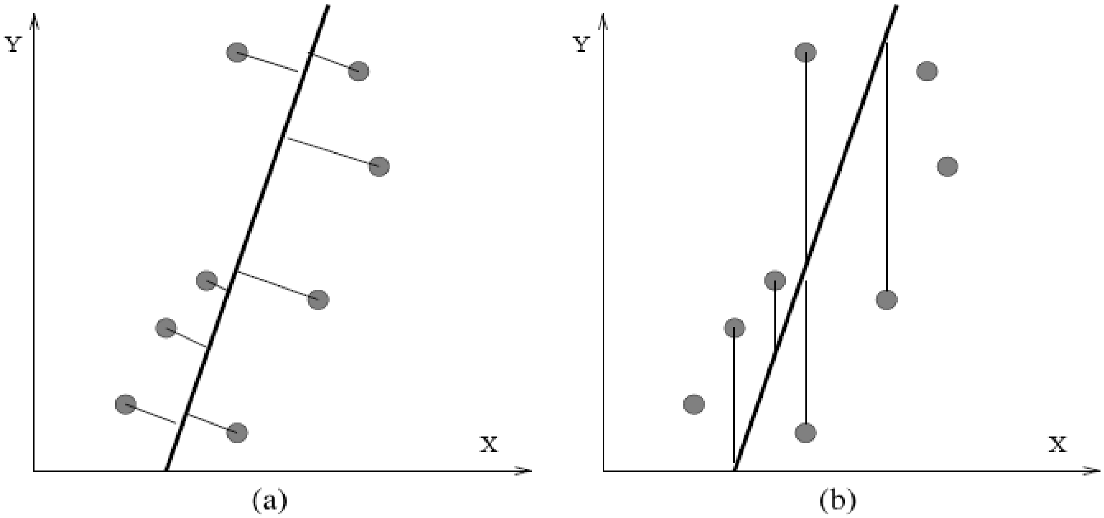

figure 2.1 : (a) the orthogonal distance and (b) ordinary least squares distance. It can be seen that the latter distance becomes less stable as the angle between the line and y-axis decreases, leading to unstable solutions.

But if we instead of obtaining *f* from *A* which is *n* x *p* matrix, if we obtain *f* by applying the constraint $f^T f = 1;$, it is equivalent to calculating the algebraic Euclidean distance as shown in figure 2.1 (a)[6,7,8].

Unfortunately both these methods produce inaccurate results. This is because the residual error is assumed to e a gaussian and also because there is no proper normalization done while computing *F*.

2.2 Invariant Linear Fitting

As described in [3], we describe fundamental matrix in this method as a 4D conic. For simplicity, we estimate *f* using a the equation $f^T J f = const.$ If the coordinate system is changed in one or both images, then the best fitting $\hat{F}$ to the transformed points must be exactly the result of the same transformation (s) applied to the F of the original points. In [5] an invariant norm for conics was suggested for Euclidean transformation It has been observed that the fundamental matrix is like a conic in four dimensional image space [7] . More on constructing these invariants is given in [6]

Consider the transformations of the image coordinates G in one image are such that $G\widetilde{x} = x$, and image two $G\widetilde{x}' = x'$ which leads to a transformation of F such that $\widetilde{F} = G'FG$ with

$$G = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}, \quad G' = \begin{bmatrix} R' & t' \\ 0^T & 1 \end{bmatrix}, \quad F = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix}, \quad \widetilde{F} = \begin{bmatrix} \widetilde{A} & \widetilde{b} \\ \widetilde{c}^T & \widetilde{d} \end{bmatrix} \tag{2.2}$$

Thus, from the above equations, it can be seen that

$$\widetilde{F} = \quad G'FG = \begin{bmatrix} \widetilde{A} & \widetilde{b} \\ \widetilde{c}^T & \widetilde{d} \end{bmatrix} = \begin{bmatrix} R'^T AR & R'^T At + R'^T b \\ t'AR + c^T R & t'At + t'^T b + c^T t + d \end{bmatrix} \tag{2.3}$$

From this it can be seen that if $t, t' = 0$, then the norm $\sum_{i=1}^{9} f_i^2 = 1$ and is invariant to rotations of the image plane. This leaves the upper left 2x2 submatrix to define the norm, because only these values are unaffected by the changes due to translation, and due to the orthogonality property of the rotation matrices, we have $\left\| R'^T AR \right\| = 1$ where $\|()\|$ represents the determinant of the quantity inside the brackets. The number of invariants is given by 2 which is dim(A) – dim(R', R) thus there are at least two invariants. Thus the matrix $J = diag(1,1,0,1,1,0,0,0,0)$ and the problem reduces to optimizing the value of $f^T M_k f$ subject to $f^T J f = const$. Here the value M is called the moment matrix and is given by $A^T A$, where A is the matrix encoding the information about the points in the image. This can be rewritten as a generalized eigenvector problem :

$$Jf + \lambda Mf = 0 \tag{2.4}$$

but this is not so stable hence a faster and more stable method is proposed by [5] where f is partitioned into f_1, f_2 where the former contains the elements corresponding to the first2x2 submatrix of f and the rest are contained in the latter. The matrix M is partitioned correspondingly into

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{2.5}$$

then,

$$f^T M f = f_1^T M_{11} f_1 + 2 f_1^T M_{12} f_2 + f_2^T M_{22} f_2 \tag{2.6}$$

Then we minimize $f^T M \ f$ subject to $f_1^T J_{11} f_1$, where $J_{11} = diag(1,1,1,1) = I$, for any fixed f_1, $f^T M \ f$ is minimal when

$$\frac{\partial f^T M_k f}{\partial f_2} = 2M_{12}^T f_1 + 2M_{22} f_2 = 0 \tag{2.7}$$

Which implies that

$$f_2 = -M_{22}^{-1} M_{12}^T f_1 \tag{2.8}$$

Then,

$$f^T M \ \ f = f_1^T \left(M_{11} - M_{12} M_{22}^{-1} M_{12}^T \right) f_1 = f_1^T Q f_1 \tag{2.9}$$

to minimize this for $f_1^T J_{11} f_1 = const$, if λ is a lagrangian multiplier, then we must set the derivative w.r.t f_1 of $f_1^T Q f_1 - \lambda f_1^T f_1$. This yields,

$$Qf_1 = \lambda f_1 \tag{2.10}$$

which is the eigenvector solution of the above equation. It is important to note that unless the data points all lie on a single line, the matrix M_{22} always has an inverse. Similarly the other quadric constraints can be imposed on the elements of F to restrict them to a certain subspace of fundamental matrices. We can also add an arbitrary linear constraint to F into the optimization as can be seen in [5].

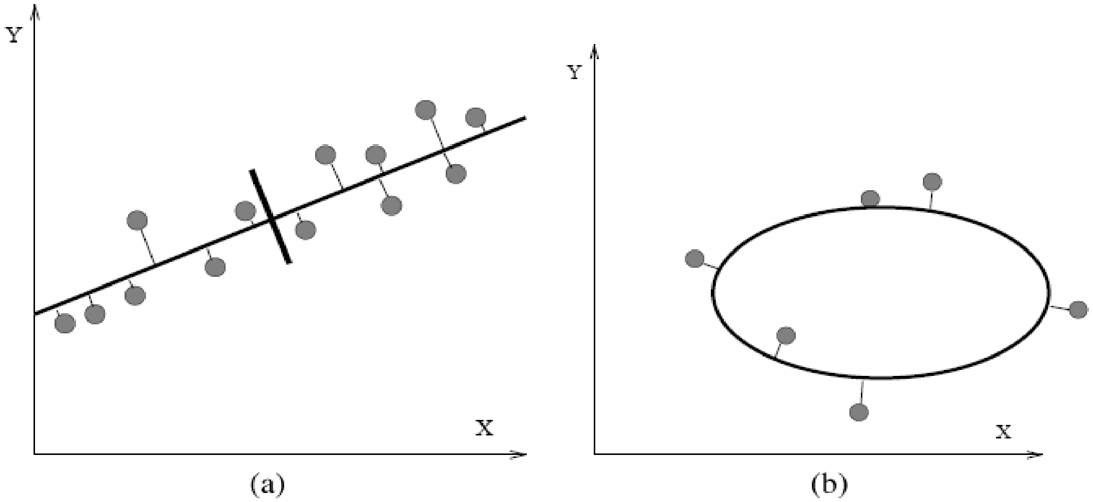

Figure 2.2 : Both (a) and (b) show the perpendicular distances to the fit. Under gaussian assumptions minimizing this gives the maximum likelihood estimate. (a) Taking eigenvalue minimizes the sum of squares of distances of the points to the line. Here the algebraic distance is equal to the geometric distance (b) The eigenvector of the moment matrix does not minimize the sum of squares of the points to a conic, here the algebraic distance is not equal to the geometric distance.

Also the above algorithm could be used for iterative least squares minimization using the method described in [11]. The nonlinear constrained/unconstrained optimization can also be done to improve the results of the program. All the methods discussed above were implemented and tested on synthetic data.

2.3 Discussion

The results obtained using all the techniques mentioned above are discussed below one set is tested using the synthetic data, and the results are compared. As we already have information about the intrinsic camera parameters of the camera used we can analyze the performance of various methods used for obtaining the F matrix.

1) Result using the conventional 8 point method/ least squares regression:

$$F = \begin{bmatrix} 0.00001007\ 088884 & 0.00009441\ 007768 & -0.01270216\ 860574 \\ -0.00010656\ 182708 & 0.00000960\ 598735 & 0.02257572\ 084021 \\ 0.01090409\ 583385 & -0.01280295\ 642921 & -0.99952296\ 435828 \end{bmatrix}$$

$$E = WL'*F*WR = \begin{bmatrix} 3.26689094\ 088018 & 30.0603628\ 0602320 & 1.94961562\ 695010 \\ -33.8114483\ 5175446 & 2.99166634\ 013016 & 3.24032135\ 977316 \\ -1.04899215\ 265914 & 1.10480852\ 313359 & 0.17597922\ 973710 \end{bmatrix}$$

$$E^T E = \begin{bmatrix} 1155 & -4.1076 & -103.38 \\ -4.1076 & 913.8 & 68.495 \\ -103.38 & 68.495 & 14.332 \end{bmatrix}$$

$$D = \begin{bmatrix} 1164.7 & 0 & 0 \\ 0 & 918.41 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [U \quad D \quad V] = svd(E^T E)$$

2) Result using the normalized 8 point method

$$F = \begin{bmatrix} -0.00000152\ 706275 & 0.00005302\ 064868 & 0.00055946\ 820068 \\ -0.00003846\ 743488 & -0.00000011\ 211317 & 0.03082955\ 563876 \\ 0.00057965\ 331706 & -0.03861893\ 834343 & 0.01934494\ 652774 \end{bmatrix}$$

$$E = WL'*F*WR = \begin{bmatrix} -0.49536317\ 339738 & 16.8818835\ 2978953 & 4.84614317\ 670212 \\ -12.2054935\ 0063668 & -0.03491626\ 403121 & 13.7481669\ 9132888 \\ -2.66633342\ 892564 & -17.0085995\ 5611661 & -1.31472147\ 133728 \end{bmatrix}$$

$$E^T E = \begin{bmatrix} 156.33 & 37.414 & -166.7 \\ 37.414 & 574.29 & 103.69 \\ -166.7 & 103.69 & 214.23 \end{bmatrix}$$

$$D = \begin{bmatrix} 602.15 & 0 & 0 \\ 0 & 342.7 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [U \quad D \quad V] = svd(E^T E)$$

3) Result using the invariant linear fitting:

$$F = \begin{bmatrix} 0.11463 & 0.66321 & -98.267 \\ -0.73555 & 0.077326 & 166.6 \\ 69.556 & -102.33 & -5522.7 \end{bmatrix}$$

$$E = WL'*F*WR = \begin{bmatrix} 37183 & 2.1117e+005 & 12831 \\ -2.3338e+005 & 24082 & 29434 \\ -6817.3 & 1692.1 & 998.02 \end{bmatrix}$$

$$E^T E = \begin{bmatrix} 5.5898e+010 & 2.22e+009 & -6.3992e+009 \\ 2.22e+009 & 4.5175e+0103. & 4201e+009 \\ -6.3992e+009 & 3.4201e+009 & 1.032e+009 \end{bmatrix}$$

$$D = \begin{bmatrix} 5.692e+010 & 0 & 0 \\ 0 & 4.5185e+010 & 0 \\ 0 & 0 & 3255.4 \end{bmatrix} \quad [U \quad D \quad V] = svd(E^T E)$$

4) Result using the nonlinear unconstrained minimization:

$$F = \begin{bmatrix} -0.13536 & -0.4271 & 164.1 \\ -0.88987 & -0.27669 & 1418.3 \\ -85.861 & -1089.6 & -6944.9 \end{bmatrix}$$

$$E = WL'*F*WR = \begin{bmatrix} -43910 & -1.3599e+005 & 44154 \\ -2.8235e+005 & -86172 & 6.9508e+005 \\ -1.2512e+005 & -6.5101e+005 & -12838 \end{bmatrix}$$

$$E^T E = \begin{bmatrix} 9.7305e+010 & 1.1176e+011 & -1.9659e+011 \\ 1.1176e+011 & 4.4973e+011 & -5.7543e+010 \\ -1.9659e+011 & -5.7543e+010 & 4.8524e+011 \end{bmatrix}$$

$$D = \begin{bmatrix} 6.2403e+011 & 0 & 0 \\ 0 & 4.0824e+011 & 0 \\ 0 & 0 & 6961.5 \end{bmatrix} \quad [U \quad D \quad V] = svd(E^T E)$$

From all the results obtained above we can see that the results obtained using both the invariant linear fitting and the results obtained using the normalized 8 point method are good because the first two diagonal values in matrix D are very near each other. It is very difficult to describe the effectiveness of the matrices this manner, the methods can be evaluated better after their performance using the self calibration method is observed.

3. Camera self-calibration using the singular value decomposition of the fundamental matrix

Intrinsic parameters of cameras are very important for 3D reconstruction. Traditionally, we can use calibration board to calibrate a camera and obtain its intrinsic parameters. This method is suitable for off-line calibration only. In practice, it often happens that we have no calibration data, but only a sequence of images. These images can be easily obtained by a hand-hold video camera. Can we get the intrinsic parameters using only a sequence of images? The answer is 'YES'. This can be done by camera self-calibration, which refers to determining the interior camera parameters of a camera by using only image data obtained from different view points.

Many methods have been proposed to do camera self-calibration. Maybank and Faugeras [12] have demonstrated that the calibration problem can be solved by tracking a set of points among images of a rigid scene, captured while the camera is pursuing unknown, unrestricted motion, the calibration parameters can be computed by determining the image of the absolute conic. The absolute conic is a special conic lying at the plane at infinity, having the property that its projection depends on the intrinsic parameters only. This fact is expressed mathematically by the so-called Kruppa equations. Several researchers have investigated the application of the Kruppa equations to the self-calibration problem. We have implemented the method presented in [13] and finished some experiments.

3.1 Introduction of the theory

In order to keep correspondence with the paper, the notations here are different what we used in class. In the perspective projection, a 3D point $M = [x,y,z]^t$ is projected to a 2D image point $m=[u,v]^t$ through a 3×4 projection matrix P, as $s\hat{m} = P\hat{M}$,where s is a scale factor and the notation $\hat{m}$ and $\hat{M}$ are the coordinates in homogeneous system.

$$\hat{m} = [u,v,1]^t, \hat{M} = [x,y,z,1]^t \tag{3.1}$$

In the case of a stereo system, every 3D point M yields a pair of 2D projections m_1 and m_2 on the two images. Those projections are defined by the following relations:

$$s_1\hat{m}_1 = P_1\hat{M}, s_2\hat{m}_2 = P_2\hat{M} \tag{3.2}$$

Assuming that the two cameras are identical and that the world coordinate system is associated with the first camera, the two projection matrices are given by:

$$P_1 = [A \mid 0], P_2 = [AR \mid At] \tag{3.3}$$

where R and t represent respectively the rotation matrix and the translation vector defining the rigid displacement between the two cameras. The intrinsic matrix has the following form

$$A = \begin{bmatrix} \alpha_u & -\alpha_u \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

The parameters α_u and α_v correspond to the focal length in pixels along the axes of the image, θ is the angle between the two image axes and (u_0, v_0) are the coordinates of the image principal point. In practice, θ is very close to 90° for real camera.

Let K denote the symmetric matrix AA^t. By eliminating the scalars s_1 and s_2 in equation (3.2), we get the relationship between the pair of projections of the same 3D point M.

$$\hat{m}_2^t F \hat{m}_1 = 0 \tag{3.5}$$

where F is the fundamental matrix, given by

$$F = (A^{-1})^t [t]_\times RA^{-1} \tag{3.6}$$

$[x]_\times$ denotes the antisymmetric matrix of vector x that is associated with the cross product. The matrix has the property $[x]_\times y = x \times y$ for each vector y and has the following analytic form:

$$[x]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{3.7}$$

The fundamental matrix F describes the epipolar geometry between the pair of views. It is the equivalent of the essential matrix $E = [t]_\times R$. The relationship between fundamental matrix and essential matrix is as follows:

$$E = A^t FA \tag{3.8}$$

As pointed out by Trivedi [14], the symmetric matrix EE^t is independent of the rotation R since

$$EE^t = [t]_\times RR^t ([t]_\times)^t = [t]_\times ([t]_\times)^t \tag{3.9}$$

Substitution of Eq.(3.8) into the above equation yields

$$FKF^t = (A^{-1})^t [t]_\times ([t]_\times)^t A^{-1} \tag{3.10}$$

This equation will be employed to derive the simplified Kruppa equations.

We shall not introduce the derivation of the classical Kruppa equations here. We will only introduce the result of the simplified Kruppa equations presented in [13]. Let *e`* denote the epipole in the second image. Using the singular value decomposition (SVD), the fundamental matrix *F* can be expressed:

$$F = UDV^t \tag{3.11}$$

The fundamental matrix *F* has a rank of 2, the diagonal matrix *D* has the following form:

$$D = \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.12}$$

where *r* and *s* are the eigenvalues of the matrix FF^t, *U* and *V* are two orthogonal matrices. By making use of this relation, the epipole in the second image e' can be deduced simply.

$$F^t e^{'} = VD^t U^t e^{'} = 0 \tag{3.13}$$

Since *D* is a diagonal matrix with a last element equal to zero, the following direction solution for e' is obtained:

$$e^{'} = \gamma Um, \gamma \neq 0 \tag{3.14}$$

with m=$[0,0,1]^t$. Therefore, the matrix $[e']_\times$ is equal to

$$[e^{'}]_\times = \mu UQU^t \tag{3.15}$$

where μ is a nonzero scale factor and Q=$[m]_\times$ is given by:

$$Q = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.16}$$

By substituting Eq.(3.16) into Eq.(3.10), a new expression for the Kruppa equations is obtained:

$$FKF^t = \mu UQU^t KUQ^t U^t \tag{3.17}$$

Since *U* is an orthogonal matrix, left and right multiplication by U^t and *U* respectively, yields the following notably simple expression for the Kruppa equations:

$$DV^tKVD^t = \mu QU^tKUQ^t \tag{3.18}$$

Because of the simple forms of the matrices *D* and *M*, relation (3.18) corresponds to three linearly dependent equations. Indeed, denoting by u_1, u_2, u_3 the column vectors of *U* and by v_1, v_2, v_3 the column vectors of *V*, the matrix equation (3.18) is equivalent to

$$\begin{aligned} DV^tKVD^t &= \begin{bmatrix} r^2v_1^tKv_1 & rsv_1^tKv_2 & 0 \\ srv_2^tKv_1 & s^2v_2^tKv_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ QU^tKUQ^t &= \begin{bmatrix} u_2^tKu_2 & -u_2^tKu_1 & 0 \\ -u_1^tKu_2 & u_1^tKu_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3.19}$$

The above expressions finally yield the following three linearly dependent equations for the matrix K:

$$\frac{r^2v_1^tKv_1}{u_2^tKu_2} = \frac{rsv_1^tKv_2}{-u_2^tKu_1} = \frac{s^2v_2^tKv_2}{u_1^tKu_1} \tag{3.20}$$

Only two of these three equations are linearly independent. They are the simplified Kruppa equations.

3.2 Self-calibration using the simplified Kruppa Equations

The simplified Kruppa equations can be applied to the problem of self-calibration. These equations are embedded a non-linear optimization framework to iteratively solve the problem of self-calibration.

Let $S_F = [r, s, u_1^t, u_2^t, u_3^t, v_1^t, v_2^t, v_3^t]^t$ be the vector formed by the parameters of the SVD of *F*. Let also $\frac{\rho_i(S_F, K)}{\Phi_i(S_F, K)}, i = 1,2,3$ be the three ratios defined by Eq.(3.20). Each pair of images defines a fundamental matrix, which in turn yields the following two equations regarding the elements of *K*:

$$\begin{aligned} \rho_1(S_F, K)\Phi_2(S_F, K) - \Phi_1(S_F, K)\rho_2(S_F, K) = 0 \\ \rho_1(S_F, K)\Phi_3(S_F, K) - \Phi_1(S_F, K)\rho_3(S_F, K) = 0 \end{aligned} \tag{3.21}$$

The above system of equations is of degree two in five unknowns defining the *K* matrix. Let $\pi_{ij}(S_F, K)$ denote the difference $\rho_i(S_F, K)\Phi_j(S_F, K) - \Phi_i(S_F, K)\rho_j(S_F, K)$ and let $\sigma^2_{\pi_{ij}}(S_F, K)$ be its variance. This variance can be computed as

$$\sigma^2_{\pi_{ij}}(S_F, K) = \frac{\partial \pi_{ij}(S_F, K)}{\partial S_F} \Lambda_{S_F} \frac{\partial \pi_{ij}(S_F, K)^t}{\partial S_F} \tag{3.22}$$

where Λ_{S_F} is the 20×20 covariance matrix associated with S_F. The variances $\sigma^2_{\pi_{ij}}(S_F, K)$ can be used to automatically weight the residuals $\pi_{ij}(S_F, K)$ according to their uncertainty.

Matrix K is computed as the solution of a non-linear least squares problem, namely

$$K = \arg\ \min_K \sum_{i=1}^{N} \frac{\pi^2_{12}(S_{F_i}, K)}{\sigma^2_{\pi 12}(S_{F_i}, K)} + \frac{\pi^2_{13}(S_{F_i}, K)}{\sigma^2_{\pi 13}(S_{F_i}, K)} + \frac{\pi^2_{23}(S_{F_i}, K)}{\sigma^2_{\pi 23}(S_{F_i}, K)} \tag{3.23}$$

Recalling that each fundamental matrix yields two independent equations and K consists of five unknowns, the minimum number of required displacements (i.e. N) is in general case equal to three. Additional constraints provided by more than three fundamental matrices can improve the accuracy of the solution. The reason for including the third simplified Kruppa equation, although it is dependent on the other two, is that it further constrains the solution and provides slightly better results. The minimization of Eq.(3.23) can be done using a classical Levenberg-Marquardt algorithm. The intrinsic matrix A can be extracted from K in three steps. First, A^{-t} is computed by employing the Cholesky decomposition of K^{-1}, then it is transposed and inverted to yield A.

3.3 Experiments and discussion

In practice, we implement the algorithm a little different from the above theory. The MATLAB has optimization toolbox that can iteratively solve the non-linear least square problem. We can directly use the unknown variables in intrinsic matrix A as the input parameters in iterative process. Besides, the Eq.(3.23) requires calculation of Jacobian and covariance matrix in order to get the weight $\sigma^2_{\pi_{ij}}(S_F, K)$. If we assume the weights are uniform, the implementation difficulty will decrease very much. The simplified optimization problem becomes

$$K = \arg\ \min_K \sum_{i=1}^{N} \pi^2_{12}(S_{F_i}, K) + \pi^2_{13}(S_{F_i}, K) + \pi^2_{23}(S_{F_i}, K) \tag{3.24}$$

Since the key idea of solving the self-calibration problem using non-linear optimization is to satisfy the constraints described in Eq.(3.21), the kind of implementation is also reasonable.

We use eight images of a image sequence taken by a video camera to do our experiments. Figure (3.1) shows two of the images (the first one and the eighth one). We find some matching points between each pair of images and use these points to calculate the fundamental matrix F. Based on the fundamental matrices between pairs of images, we implement a MATLAB code to solve the optimization problem in Eq.(3.24). We got some interesting results.

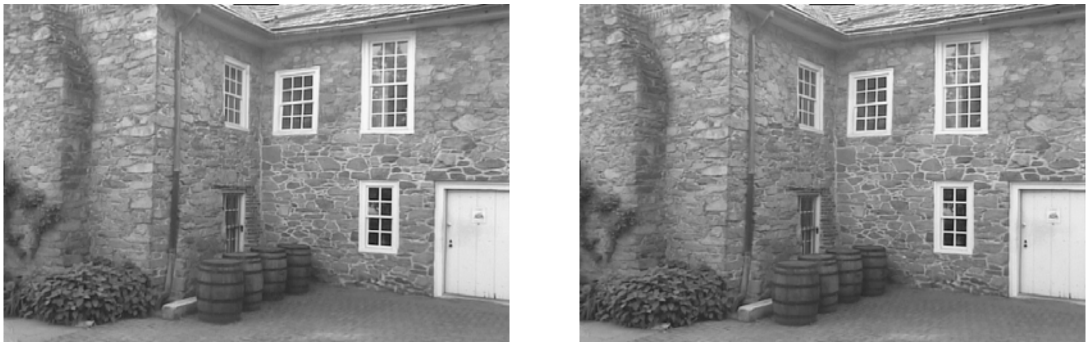

Figure 3.1 The first and the eighth images in image sequence 1

In order to solve the non-linear optimization problem, we need some initialization. For real camera, the θ in intrinsic matrix *A* is normally 90^{o}. Hence we only take four variables in intrinsic matrix, i.e., $\alpha_u, \alpha_v, u_0, v_0$. Generally speaking, we use the image center as the initialization of principal points. We must guess the initial value for α_u, α_v. We noticed that if the initialization is far away from the real values, the result by non-linear optimization is not correct. For example, if we use [500 800 360 240]^{t} as the initialization for $\alpha_u, \alpha_v, u_0, v_0$, the optimization converges to wrong solution [52.484 77.012 148.51 -193.48]^{t}. This is a general problem for non-linear optimization. The result depends on the initialization. If the initialization is too far away, the optimization will converge to another unreasonable local minima. Table 3.1 shows some of our experimental results. We also calculate a relative deviation to qualify the converging results, which is the ratio of standard deviation to the mean of the results. Let r_i denote the optimization result of a variable, μ_i denote the mean of its results, we calculate the quality number as

$$\lambda_i = \frac{|r_i - \mu_i|}{\mu_i} \tag{3.25}$$

We notice that even the initialization changes in a relatively large range, the results of the non-linear optimization process do not change significantly, which means the robustness of the non-linear optimization.

Table 3.1 initialization and results of our experiments

Initializations				Result			
α_u	α_v	u_0	v_0	α_u	α_v	u_0	v_0
650	750	360	240	700.3	683.1	377.9	228
500	800	360	240	703.21	698.96	410.93	253.1
800	800	360	240	710.18	698.7	406.87	247.5
400	900	360	240	52.484	77.012	148.51	-193.48

Table 3.2 relative deviation of our experimental results

	α_u	α_v	u_0	v_0
μ	704.5633333	693.5866667	398.5666667	242.8666667
λ	0.006051029	0.015119476	0.051852471	0.061213286
	0.001920811	0.007747169	0.031019486	0.042135603
	0.007971841	0.007372306	0.020832985	0.019077683

We also found a problem with our non-linear optimization process. For some special data set, the convergence range for different initializations is small. For the image sequence like figure 3.3, it's difficult to find a good initialization that can produce a stable result. We think this is due to the special structure of the image pattern, i.e., most feature points are coplanar. Besides, we know that this image sequence was taken by a auto-zoom camera. Since the focus length always changed, it's more difficult to estimate the intrinsic matrix. Even though, we can get some estimation with our implemented algorithms, which are shown in table 3.3.

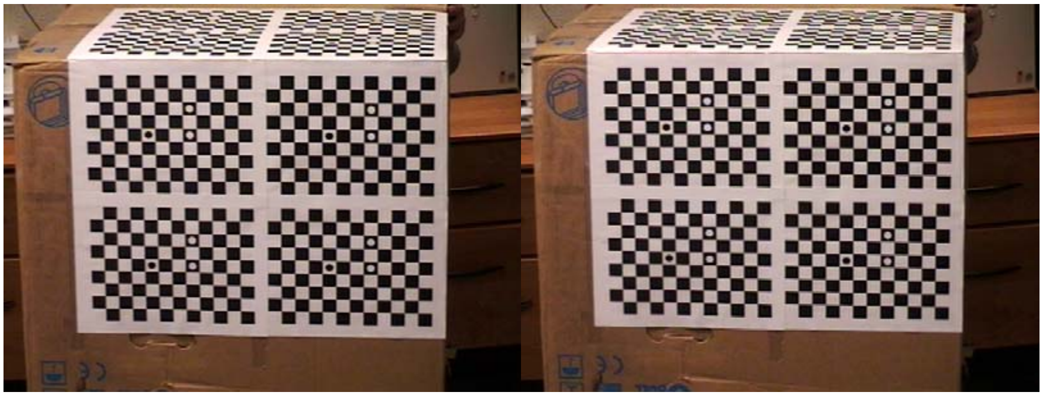

Figure 3.2 the first and eighth image in the second image sequence

Initializations				Result			
α_u	α_v	u_0	v_0	α_u	α_v	u_0	v_0
1326.5	1382.5	345.69	225.67	1400	1400	320	240
1035.9	1259.2	311.19	350.18	1500	1400	320	240

Table 3.3 results on the second image sequence

Conclusion

We use SSD for establishing correspondences and use RANSAC method to robustly calculate fundamental matrix. Since fundamental matrix is the key data for self-calibration implemented in this project, we also implement other two methods presented in [5][11] to calculate the fundamental matrix more accurately and robustly. After we got the fundamental matrix between each pair of images in the image sequence, we can use the simplified Kruppa equations to solve for intrinsic matrix using a non-linear optimization process. The results by non-linear optimization process can converge to a stable solution even if the initialization varies in a relatively large range.

Contribution of each team member

Bhavani Shankar Yanamadala, Fundamental Matrix Estimation (Section 2)
Lei Zhang, Camera self-calibration Using the SVD of the *F* Matrix (Section 3)
Xiaoli Zhang, Computation of the Fundamental Matrix using RANSAC (Section 1)

References

[1] Rovid A., Annamaria R., Varlaki P., 3D model estimation from multiple images, *IEEE International conference on fuzzy systems* 3(25-29), pp. 1661-1666, July 2004.

[2] P. H. S. Torr and D.W. Murray, "The development and comparison of robust methods for estimating fundamental matrix", International Journal of Computer Vision, 24(3), pp. 271-300 1997

[3] R. I. Hartley: Minimizing Algebraic Error in Geometric Estimation Problems Proc.*DARPA Image Understanding Workshop, Pages 631-637, 1997*

[4] R. I. Hartley: In Defense of 8 point algorithm. In Proc. *International Conference on Computer Vision, Pages 1064-1070, 1995.*

[5] F. Bookstein Fitting Conic Sectioins to Scattered Data. *Computer Vision Graphics and Image Processing,9:56-71, 1979.*

[6] J. Mundy and A. Zisserman *Geometric Invariance in Computer Vision.* MIT Press 1992.

[7] P. H. S. Torr, *Outlier Detection and Motion Segmentation*, Ph.D thesis, Department of Engineering Science, University of Oxford, 1995

[8] K. Pearson, On lines and planes of closest fit to systems of points in space. *Philos. Mag.Ser.*6 2:559.1901.

[9] M. G. Kendall and A. Stuart *The advanced theory of statistics*. Charles Griffin and company, London 1983.

[10] G. H. Golub and C. F. Van Loan *Matrix Computations*, The John Hopkins University Press, 1901.

[11] P. D. Sampson. Fitting Conic Sections to 'very' Scattered Data: An iterative implementation of Bookstein Algorithm. *Computer Vision, Graphics and Image Processing,* 18:97-108, 1982.

[12] S.J. Maybank and O.D.Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision*, 8(2):123-152, Aug. 1992.

[13] Manolis I.A. Lourakis and Rachid Deriche. Camera self-calibration using the singular value decomposition of the fundamental matrix. 1999

[14] H.P.Trivedi. Can multiple views make up for lack of camera registration. *Image and Vision Computing*, 6(1):29-32, Feb. 1988