

Assignment 4: CS 763, Computer Vision

Sasank Chilamkurthy
Tharun Kumar Reddy
Rajeev Puppala

April 2, 2015

1. Answer the following questions pertaining to the essential/fundamental matrix in a binocular stereo system:

- (a) Consider two cameras with parallel optical axes, with the optical center of the second camera at a location $(a, 0, 0)$ as measured in the first camera's coordinate frame. What is the essential matrix of this stereo system?

Ans. Rotation matrix is Identity Matrix since the axis are parallel and $\mathbf{T} = (a, 0, 0)$

$$\Rightarrow \mathbf{T}_x = a; \mathbf{T}_y = 0; \mathbf{T}_z = 0$$

$$\Rightarrow \mathbf{S} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -a \\ 0 & a & 0 \end{bmatrix}$$

Therefore $\mathbf{E} = \mathbf{RS}$

$$\Rightarrow \mathbf{E} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -a \\ 0 & a & 0 \end{bmatrix}$$

- (b) Suppose I gave you the fundamental matrix of a stereo system, how will you infer the left and right epipoles?

Ans. The left epipole lies on all epipolar lines in the left image. Hence we can write

$$\tilde{P}_r^t \mathbf{F} \tilde{e}_l = 0$$

$$\therefore \mathbf{F} \tilde{e}_l = 0 \Rightarrow \tilde{e}_l \text{ lies in the nullspace of } \mathbf{F}$$

In the same way \tilde{e}_r is the nullspace of \mathbf{F}^T

When \mathbf{F} is written as $\mathbf{U}_\mathbf{F} \mathbf{S}_\mathbf{F} \mathbf{V}_\mathbf{F}^T$

\tilde{e}_l is column of $\mathbf{V}_\mathbf{F}$ corresponding to null singular value

\tilde{e}_r is column of $\mathbf{U}_\mathbf{F}$ corresponding to null singular value

- (c) Prove that any essential matrix will have one singular value which is zero, and that its other two singular values are identical. Derive a relationship between these singular values and the extrinsic parameters of the stereo system (*i.e.*, the rotation matrix \mathbf{R} and/or the translation vector \mathbf{t} between the coordinate frames of the two cameras). [Hint: Show that if \mathbf{E} is the essential matrix, then we can write $\mathbf{E}^T \mathbf{E} = \alpha (\mathbf{I}_{3 \times 3} - \mathbf{t}_u \mathbf{t}_u^T)$ where α is some scalar which you should express in terms of \mathbf{R} and/or \mathbf{t} , $\mathbf{I}_{3 \times 3}$ is the identity matrix with 3 rows and 3 columns, and \mathbf{t}_u is a vector of unit magnitude in the direction of \mathbf{t}].

$$\text{Ans. } \mathbf{E}^T \mathbf{E} = (\mathbf{RS})^T (\mathbf{RS}) = \mathbf{S}^T \mathbf{R}^T \mathbf{RS} = \mathbf{S}^T \mathbf{S}$$

$$\mathbf{S} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

$$\Rightarrow \mathbf{S}^T \mathbf{S} = \begin{bmatrix} T_z^2 + T_y^2 & -T_y T_x & -T_x T_z \\ -T_y T_x & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix} = (T_x^2 + T_y^2 + T_z^2) (\mathbf{I}_{3 \times 3} - \mathbf{t}_u \mathbf{t}_u^T)$$

$$\text{where } \mathbf{t}_u = \begin{pmatrix} \frac{T_x}{\sqrt{T_x^2 + T_y^2 + T_z^2}} \\ \frac{T_y}{\sqrt{T_x^2 + T_y^2 + T_z^2}} \\ \frac{T_z}{\sqrt{T_x^2 + T_y^2 + T_z^2}} \end{pmatrix} \text{ is a vector of unit magnitude in the direction of } \mathbf{t}$$

Let v be an eigen vector of $\mathbf{E}^T \mathbf{E}$ and λ be its eigen value. v can be written as $\lambda_1 \mathbf{t}_u + \lambda_2 \mathbf{t}_{u\perp}$ where $\mathbf{t}_{u\perp}$ is a unit magnitude perpendicular vector to \mathbf{t}_u and λ_1 and λ_2 are the corresponding dot products of v with \mathbf{t}_u and $\mathbf{t}_{u\perp}$.

Substituting v in the above equation gives

$$\begin{aligned} \alpha(\mathbf{I}_{3 \times 3} - \mathbf{t}_u \mathbf{t}_u^T) v &= \lambda v \text{ where } \alpha = T_x^2 + T_y^2 + T_z^2 \\ \Rightarrow (\alpha - \lambda)(\lambda_1 \mathbf{t}_u + \lambda_2 \mathbf{t}_{u\perp}) &= \alpha \mathbf{t}_u \mathbf{t}_u^T (\lambda_1 \mathbf{t}_u + \lambda_2 \mathbf{t}_{u\perp}) \\ \Rightarrow (\alpha - \lambda)\lambda_1 \mathbf{t}_u + (\alpha - \lambda)\lambda_2 \mathbf{t}_{u\perp} &= \alpha \lambda_1 \mathbf{t}_u + 0 \\ \Rightarrow \lambda \lambda_1 &= 0 \text{ and } (\alpha - \lambda)\lambda_2 = 0 \end{aligned}$$

Therefore as both λ_1 and λ_2 cannot be simultaneously 0

if $\lambda_1 = 0$ then $\alpha = \lambda$ and

if $\lambda_2 = 0$ then $\lambda = 0$

In the first case we can find two vectors that are not parallel to each other and perpendicular to \mathbf{t}_u

\therefore one eigen value is 0 and other two eigen values are same and equal to α

Singular values of essential matrix are square roots of eigen values of $\mathbf{E}^T \mathbf{E}$

\therefore singular values of \mathbf{E} are $0, \sqrt{T_x^2 + T_y^2 + T_z^2}, \sqrt{T_x^2 + T_y^2 + T_z^2}$.

- (d) In the noiseless case, what is the minimum number of corresponding pairs of points you must know in order to estimate the essential matrix? Or in other words, how many degrees of freedom does an essential matrix have? Justify your answer. (Think carefully).

Ans. $\mathbf{E} = \mathbf{R}\mathbf{S}$

\mathbf{R} is rotation matrix. \therefore DOF of $\mathbf{R} = 3$

\mathbf{S} is Translation matrix. \therefore DOF of $\mathbf{S} = 3$

\Rightarrow DOF of \mathbf{E} should be 6 But if \mathbf{E} is an essential matrix then $\alpha \mathbf{E}$ can also be an essential matrix for any scalar α for the same system

Therefore by fixing α we can reduce the DOF of \mathbf{E} by 1

\therefore Degrees of freedom for essential matrix $\mathbf{E} = 5$

\Rightarrow 5 pairs of corresponding pairs of points are required to estimate essential matrix

- (e) We have studied the eight-point algorithm in class for estimating the essential/fundamental matrix. There exist algorithms that require only 7 pairs of corresponding points. In robust estimation, what main advantage will a 7-point algorithm have over the 8-point version?

Ans. In robust estimation using RANSAC method the no. of sets of pairs of corresponding points to be taken to maximise the probability of finding a set with all inliers is less if we use 7-point algorithms instead of 8-point algorithm for a given percentage of outliers (p)

$$[1 + 1 + 5 + 1 + 2 = 10 \text{ points}]$$

2. In class, we have seen (or will soon see) a method of normalizing matching points from the two given images, before computing the fundamental matrix or the homography transformation (you would have seen this normalization in the code for homography that was handed out in assignment 2). This normalization involves zero-centering each individual point-set followed by a rescaling. Express this normalization in the form of a matrix, and write out the expression for the final fundamental matrix. Explain why this normalization is useful. [5 points].

Ans. The origin of the new coordinate system should be centered at the centroid of the image points. This is accomplished by a translation of the original origin to the new one and after the translation the coordinates are uniformly scaled so that the mean distance from the origin to a point equals $\sqrt{2}$.

The final transformation matrix will be a product of a scalar and transition matrix which will normalize the image coordinates.

New homogeneous image coordinates $\bar{\mathbf{y}}, \bar{\mathbf{y}}'$ are given by

$$\bar{\mathbf{y}} = \mathbf{T} \mathbf{y}$$

$$\bar{\mathbf{y}}' = \mathbf{T}' \mathbf{y}'$$

where \mathbf{T}, \mathbf{T}' are the transformations of the two images from the old to the new normalized image coordinates.

Epipolar constraint based on the fundamental matrix can now be rewritten as

$$(\bar{\mathbf{y}}')^T ((\mathbf{T}')^T)^{-1} \mathbf{F} \mathbf{T}^{-1} \bar{\mathbf{y}} = (\bar{\mathbf{y}}')^T \bar{\mathbf{F}} \bar{\mathbf{y}} = 0$$

$$\Rightarrow \bar{\mathbf{F}} = ((\mathbf{T}')^T)^{-1} \mathbf{F} \mathbf{T}^{-1} \text{ where } \bar{\mathbf{F}} \text{ is the new fundamental matrix for normalised coordinates.}$$

The de-normalised fundamental matrix is given by

$$\mathbf{F} = (\mathbf{T}')^T \bar{\mathbf{F}} \mathbf{T}$$

This Normalisation reduces the problem caused by the poor distribution of the homogeneous image coordinates in their space.

3. In this exercise, you will implement the Adaboost method for creating a strong binary classifier from a series of weaker classifiers. You will work with some synthetic datasets and also with a dataset containing images of digits.

Consider a training set consisting of N input vectors $\{\mathbf{x}_j\}_{j=1}^N$ in a d -dimensional space, and their respective labels $\{y_j\}_{j=1}^N$ where $\forall j, y_j \in \{-1, +1\}$. You will assign a scalar weight to each input vector. Before the first iteration of Adaboost, these weights will be set to be equal in value. In each round t , you will pick the best classifier from the following family of weak classifiers: $h_t(\mathbf{x}; i, p, \theta) = \text{sign}(p(x_i - \theta))$ where x_i is the i^{th} element of d -dimensional input vector \mathbf{x} , the parameter $p \in \{-1, +1\}$ and θ is a real-valued threshold parameter. Basically, this classifier assigns input vector \mathbf{x} the label '+1' if either (1) $x_i > \theta$ and $p = +1$, or (2) $x_i \leq \theta$ and $p = -1$. Otherwise it assigns \mathbf{x} the label '-1'. The best classifier refers to the classifier producing the least weighted error on the training set, i.e. least value of $\epsilon = \sum_{j=1}^N w_j I(h_t(\mathbf{x}_j) \neq y_j)$ where $I(\cdot)$ is an indicator function that returns 1 if the predicate passed as a parameter is true, and returns 0 otherwise. Note that the search for the best classifiers involves picking the tuple (i, p, θ) . After picking the best classifier, you will update the weights following the method in the standard Adaboost algorithm. This entire process is repeated for T rounds. You need to specify the value of T but $T = 30$ to 40 is sufficient. The final classifier after T rounds will have the form $H(\mathbf{x}) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}))$. You will work with the following datasets.

- (a) A dataset containing 2000 points in 2D drawn from a $[0,1]$ bounded uniform random distribution. Label all the points lying on or inside a rectangle bounded by the lines $x = 0.3, x = 0.7, y = 0.3, y = 0.7$ as +1 and the rest as -1. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset1).
- (b) A dataset containing 2000 points in 2D drawn from a $[0,1]$ bounded uniform random distribution. Label all the points satisfying any of the following conditions as '+1' and the rest as '-1': (1) lying on or inside a rectangle bounded by the lines $x = 0.3, x = 0.7, y = 0.3, y = 0.7$ as +1, (2) with x-coordinate between 0.15 and 0.25 or between 0.75 and 0.85, (3) with y-coordinate between 0.15 and 0.25 or between 0.75 and 0.85. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset2).
- (c) A dataset containing 2000 points in 2D drawn from a zero-mean Gaussian distribution of standard deviation 2. Label all the points whose distance from the origin is less than 2 as +1 and the rest as -1. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset3).
- (d) A dataset containing 2000 points in 2D drawn from a zero-mean Gaussian distribution of standard deviation 2. Label all the points whose distance from the origin is either less than 2, or between 2.5 and 3, as +1 and the rest as -1. Randomly divide this dataset into disjoint sets of 1000 training points and 1000 test points (called dataset4).
- (e) The text files can be read into MATLAB using 'dlmread'. Each image has size 13×13 , and there are 1000 images each in the training set as well as the test set. Label the images belonging to any one digit (say '2' or '3') as '+1' and all the others as '-1'. Thus for this database, your job is to determine whether a given image contains the selected digit or not.

For each of the five datasets, do the following after each round of Adaboost: (1) estimate and print the training error of the *strong* classifier created thus far, (2) estimate and print the error of the *strong* classifier created thus far on the *test* set. For the first four datasets, also plot the test points with their associated labels using the MATLAB function called 'scatter' (in each round of Adaboost). Sample scatter plots for the first four datasets can be found at (named as dataset1.jpg and so on). Finally, for all four datasets, plot a graph of the test set error versus the number of rounds of Adaboost. Do you notice something peculiar with the fourth dataset? Explain what you would do to remedy that situation (there is no need to implement).

[15 points]

- In part d, error rate is unusually high. This is because concentric circular boundaries are hard to capture with classifiers of the form $\text{sign}(p(x_i - \theta))$. So, this problem can be solved by including classifiers of the form $\text{sign}(p(r - \phi))$ where r is distance from the origin.

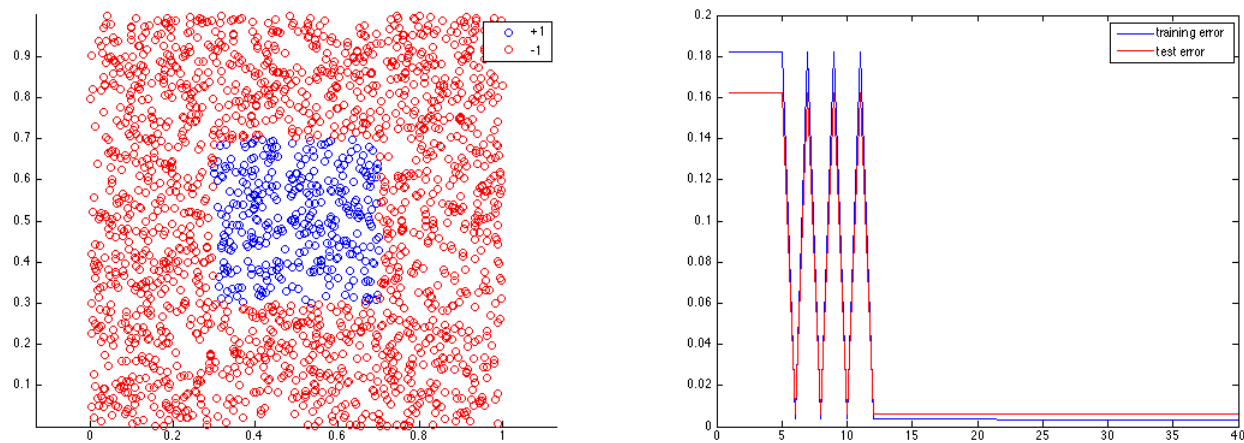


Figure 1: Part a

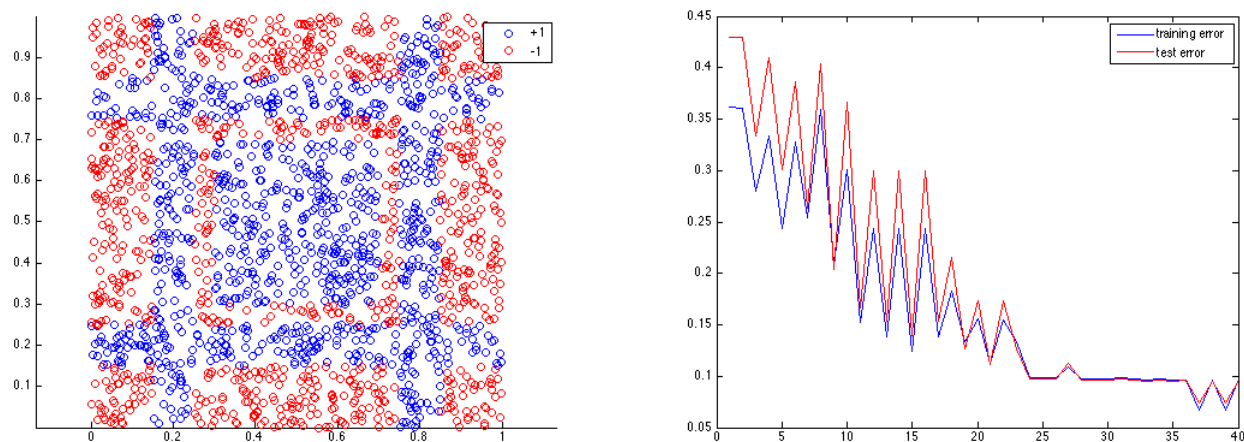


Figure 2: Part b

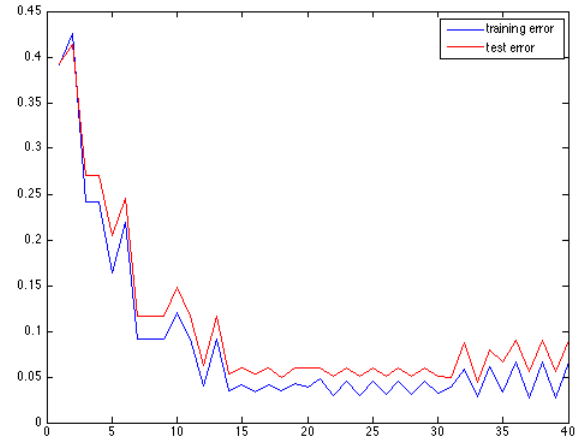
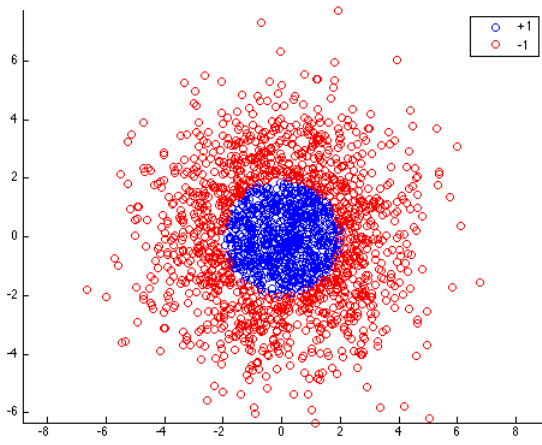


Figure 3: Part c

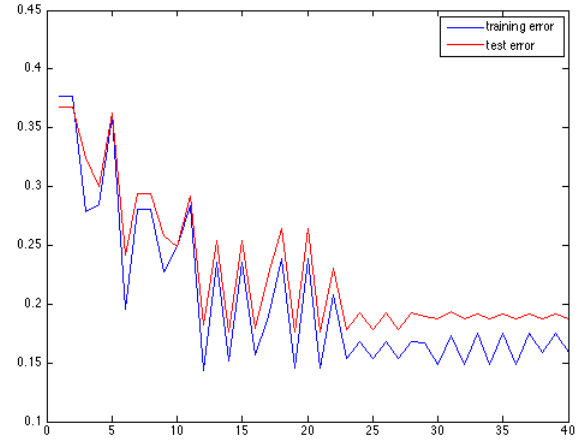
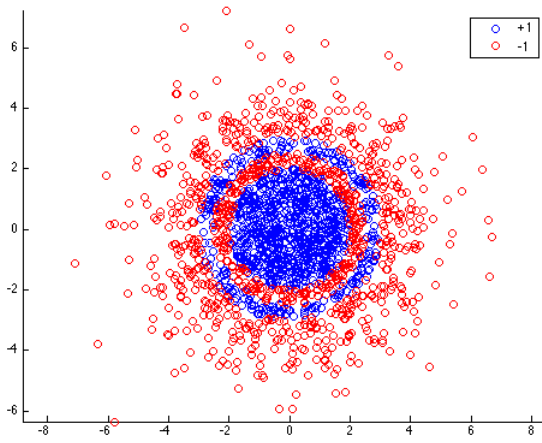


Figure 4: Part d

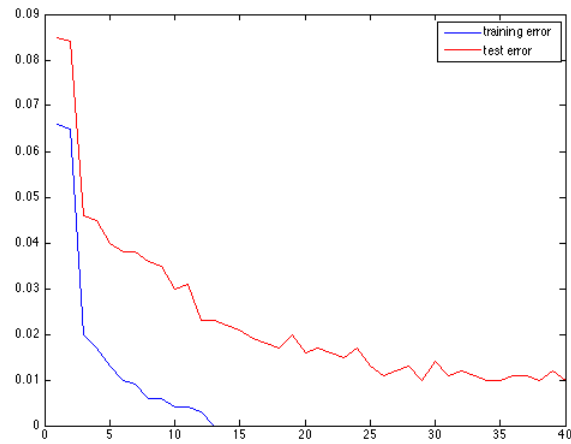


Figure 5: Part e