CS 280: Computer Vision
Fall 2012
Assignment 2: 3D reconstruction from Two Views

September 13, 2012

The goal of this assignment is the three dimensional reconstruction of an object depicted from two different views. In particular, from corresponding 2D points in the two images you will have to estimate their 3D location in the scene as well as estimate the camera positions and orientations.

1 Camera Models

Assume $OXYZ$ is the global coordinate system in the 3D scene, while $O_cX_cY_cZ_c$ is the camera's coordinate system. Note that in general, these two systems are not identical. If $\widetilde{\mathbf{X}}$ is a (non-homogeneous) point in the scene wrt the global coordinate system, while $\widetilde{\mathbf{X}}_{cam}$ is the same point wrt the camera coordinate system. It holds that

$$\widetilde{\mathbf{X}}_{cam} = R(\widetilde{\mathbf{X}} - \widetilde{\mathbf{C}}) \tag{1}$$

where $\widetilde{\mathbf{C}}$ is the center of the camera wrt the global coordinate system and R is a 3×3 rotation matrix which describes the orientation of the camera coordinate system wrt the global coordinate system.

If $\mathbf{X}$ is the 3D point in homogeneous coordinates and $\mathbf{x}$ the corresponding 2D point in the image, it holds

$$\mathbf{x} = KR[I| - \widetilde{\mathbf{C}}]\mathbf{X} \tag{2}$$

where K is the camera's calibration matrix which is defined by the intrinsic parameters of the camera.

The camera matrix is defined from the above equation. In other words, $P = K[R|\mathbf{t}]$, where $\mathbf{t} = -R\widetilde{\mathbf{C}}$.

In case of two cameras, it is often easier to locate the global coordinate system to the coordinate system of the first camera. In that case, $P_1 = K_1[I|\mathbf{0}]$ and $P_2 = K_2[R|\mathbf{t}]$.

2 Fundamental Matrix

The fundamental matrix F is a 3×3 matrix which relates corresponding points in stereo images. Assume $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ are corresponding points in an image pair, then it holds

$$\mathbf{x}_2^T F \mathbf{x}_1 = 0 \tag{3}$$

The fundamental matrix is of rank 2.

3 The Essential Matrix

Assume $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ are corresponding points in an image pair, the essential matrix E relates the uncalibrated corresponding image points. Alternatively, it can defined from the fundamental matrix as follows

$$E = K_2^T F K_1 \tag{4}$$

If $P_1 = K1[I|\mathbf{0}]$ and $P_2 = K_2[R|\mathbf{t}]$, the essential matrix can also be written as

$$E = [\mathbf{t}]_x R \tag{5}$$

where $[\mathbf{t}]_x$ is the representation of the cross product with $\mathbf{t}$ (a detailed proof of Eq. 5 can be found in *Multiview Geometry* by Hartley & Zisserman).

From Eq. 5 we can show that the essential matrix is of rank 2 and has two identical real singular values while the third one is zero. (Optional, prove it!)

From Eq. 5 it is apparent that knowledge of the essential matrix can lead to estimates of the camera orientation and center location (see below).

4 Algorithms

In this section, we describe algorithms that are used to solve various problems related to the task of 3D reconstruction.

4.1 Eight-Point Algorithm

This algorithm fits the fundamental matrix defined by corresponding points $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ in an image pair. In particular, assume $\mathbf{x}_1^{(i)} = (x_1^{(i)}, y_1^{(i)})$ in the first image corresponds to $\mathbf{x}_2^{(i)} = (x_2^{(i)}, y_2^{(i)})$ in the second image for $i = 1, ..., N$. It should be $N \geq 8$, which means we need at least 8 corresponding points.

Normalization. Usually, the point correspondences are not accurate. Arithmetic inaccuracies due to noise can be eliminated by normalizing the points in the image, by translating them by μ so that the mean of the points is the origin and scaling them by σ so that the mean distance from the origin is a constant (e.g. $\sqrt{2}$). Assume the linear transformation described above is represented by the matrices T_1 and T_2 for the two images.

Optimization. Eq. 3 can be rewritten equivalently as $Af = 0$, where f is formed from the entries of F stacked to a 9-vector row-wise and A is a $N \times 9$ dimensional matrix. In particular, the i-th row of A is equal to

$$A_i = [x_1^{(i)}x_2^{(i)} \; y_1^{(i)}x_2^{(i)} \; x_2^{(i)} \; x_1^{(i)}y_2^{(i)} \; y_i^{(i)}y_2^{(i)} \; y_2^{(i)} \; x_1^{(i)} \; y_1^{(i)} \; 1] \tag{6}$$

where the point coordinates are the normalized ones, i.e. $\mathbf{x} \leftarrow T\mathbf{x}$

The linear system$Af = 0$ has an exact solution if $rank(A) = 8$, which happens if the point correspondences are exact. Usually and due to noise it holds that $rank(A) > 8$. In that case, there is no exact solution to the linear system and an approximate solution has to be found.

An approximate solution can be found by solving the following optimization problem

$$\min_f \; ||Af||_2 \quad s.t. \; ||f||_2 = 1 \tag{7}$$

which can be solved using the SVD decomposition of matrix A.

The solution F^* found by the approximate solution to the problem $Af = 0$, does not guarantee that the fundamental matrix will be of rank 2, or else $det(F) = 0$. We need to enforce that constraint, solving another optimization problem, in particular

$$\min_F \; ||F - F^*||_{Frob} \quad s.t. \; det(F) = 0 \tag{8}$$

Again, this problem can be solved using the SVD decomposition of F^*. In particular, if $F^* = USV^H$, where $S = diag(s_1, s_2, s_3)$ and $s_1 \geq s_2 \geq s_3$ then $F = U\hat{S}V^H$, where $\hat{S} = diag(s_1, s_2, 0)$.

Denormalization. After enforcing the rank-2 constraint, we need to remove the normalization such that the fundamental matrix corresponds to points in the actual 2D image space, i.e. $F \leftarrow T_2^T F T_1$.

4.2 Estimating Extrinsic Camera Parameters from the Essential Matrix

Another important problem is the estimation of the extrinsic camera parameters, i.e. $R, \mathbf{t}$ of the second camera matrix, when only the essential matrix E is known.

The operation $[\mathbf{t}]_x$ can also be written as

$$[\mathbf{t}]_x = SZR_{90^o}S^T \tag{9}$$

where $S = [\mathbf{s}_0 \ \mathbf{s}_1 \ \mathbf{t}]$ is an orthogonal matrix, $Z = diag(1, 1, 0)$ and R_{90^o} is the rotation matrix for rotation angle $\theta = 90^o$.

From Eq. 5

$$E = [\mathbf{t}]_x R = SZR_{90^o}S^t R = U\Sigma V^T \tag{10}$$

Since we know that $\Sigma = diag(1, 1, 0)$ (remember in homogeneous coordinates we define everything up to a multiplication factor), it holds that $U = S$ and $V = RU^T R_{90^o}^T$.

Thus, the direction of $\mathbf{t}$ as well as R can be determined by the SVD analysis of E. Notice that there is no way we can determine the actual norm of the vector $\mathbf{t}$ from just image point correspondences. Absolute values can only be determined if we have a known reference distance in the scene. In addition, the sign of both $\mathbf{t}$ and R can also not be determined from SVD decomposition. Therefore, this algorithm generates numerous candidate directions and orientations for the camera.

All the possible combinations of $\mathbf{t}$ and R result in different camera matrices $P2 = K_2[R|\mathbf{t}]$. After triangulation (3D reconstruction of the point cloud), the combination that gives the largest number of points in front of the image planes is kept.

4.3 Triangulation

Triangulation refers to the estimation of the 3D position of a point when the corresponding points in the two image planes are given as well as the parameters of the camera. In particular, assume $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ are two corresponding points in the image plane and the camera matrices are P_1 and P_2 respectively. We want to find the 3D point $\mathbf{X} = (X, Y, Z, W)$ such that

$$\mathbf{x}_1 = P_1\mathbf{X} \quad and \quad \mathbf{x}_2 = P_2\mathbf{X} \tag{11}$$

Eq. 11 can also be written analytically as

$$x_j = \frac{P_{11}^{(j)}X + P_{12}^{(j)}Y + P_{13}^{(j)}Z + P_{14}^{(j)}W}{P_{31}^{(j)}X + P_{32}^{(j)}Y + P_{33}^{(j)}Z + P_{34}^{(j)}W} \tag{12}$$

$$y_j = \frac{P_{21}^{(j)}X + P_{22}^{(j)}Y + P_{23}^{(j)}Z + P_{24}^{(j)}W}{P_{31}^{(j)}X + P_{32}^{(j)}Y + P_{33}^{(j)}Z + P_{34}^{(j)}W} \tag{13}$$

where $j = 1, 2$ are the two camera indices and $\mathbf{x}_j = (x_j, y_j)$.

The above system of non linear equations can be turned into a linear system, if both equations are multiplied by the denominators. The system will be homogenous if the point in 3D is represented in homogenous coordinates and thus can be solved via SVD. Alternatively, if we set $W = 1$ then the system is no longer homogenous and can be solved using least squares.

5 Your 3D reconstruction

In this assignment, we give you two pairs of images called **house** and **library**. Since the intrinsic parameters of a camera are nowadays known and stored when a picture is taken, we provide the calibration matrices K_1 and K_2 for the two images for both pairs of images. In general, these parameters also need to be estimated via calibration when they are unkown (which is not taught in this class!). We also give you corresponding points for both pairs of images. In general, the corresponding points are found by detecting interest points in images and then comparing their descriptors across images to find matches. You will have the chance to work on this very interesting problem in another homework!

Given a pair of images and their corresponding points as well as the intrinsic parameters of the two cameras, you will have to estimate the 3D positions of the points as well as the camera matrices. In particular, initially you will have to fit the fundamental matrix. You can impelement the 8-point algorithm described above or any other algorithm you wish. Subsequently, you will estimate the extrinisic camera parameters of the second camera from the essential matrix. From all the possible combination of parameters, you will keep the one that results in the most points in front of the two image planes. Last, you will plot the 3D points as well as the two camera centers. It is up to you to show the point cloud and also come up with a nice visualization.

5.1 Instructions

Unzip **hwk2.zip** in a directory on your machine. Inside the subdirectory called *data* you will find the data for the two pairs of images, in particular for **house** and **library**, as described above. In the subdirectory called *code*, you will find a matlab function

reconstruct_3d.m

This function is the main function that you should run and should output the points in the 3D space as well as the camera matrices. The input argument to this function is either *'house'* or *'library'* according to which dataset you want to use.

You will write four functions that are needed to reconstruct the point cloud.

- fundamental_matrix.m
  This function fits the fundamental matrix F given the data provided. Your function should return F as well as the residual res_err. The residual is defined as the mean squared distance between the points in the two images and the corresponding epipolar lines. Is this what you are directly optimizing using SVD when solving the homogenous system? If yes, explain. If no, how does the objective relate to the residual?

- find_rotation_translation.m
  This function estimates the extrinsic parameters of the second camera. The function should return a cell array R of all the possible rotation matrices and a cell array t with all the possible translation vectors.

- find_3d_points.m
  This function reconstructs the 3D point cloud. In particular, it returns a $N \times 3$ array called $points$, where N is the number of the corresponding matches. It also returns the reconstruction error rec_err, which is defined as the mean distance between the 2D points and the projected 3D points in the two images. Is this reconstruction error what you are directly optimizing when solving the linear system of equations? If yes, explain. If no, how does the objective of your optimization problem relate to this error?

- plot_3d.m
  This function plots the 3D points in a 3D plot and also displays the camera centers for both cameras. Plotting in 3D can be done using the

plot3 command. A plot of the point cloud and the camera centers is enough. However, you are free to do more elaborate stuff and create nicer visualizations if you please.

Zip your functions (only the four functions needed) and a short report (not more than 3 pages) and send it to hwk2.CS280@gmail.com. In the email, please write your name or names if you collaborated. In case of collaboration, please send the email only once but with all the names written in the email.