Aurélien Bernier Levalois & Yoann Fleytoux
Skip to Page 8 for answers
COMPUTER VISION
2015/2016
ESTIMATION OF FUNDAMENTAL MATRIX
ESSENTIAL MATRIX AND 3D RECONSTRUCTION

Consider two cameras placed in two different positions. And are P_e P_d the coordinates of a point P (in space) respectively in referential the left chamber and the right chamber. Are p_e and p d the coordinates of the images from that point on the left camera and right camera respectively. Note that P_e and P_d are the coordinates of the same point in space expressed in two different coordinate systems while p_e and p_d are coordinates of two different points.

Coordinate systems associated with each chamber are related each other through a rigid body transformation, which is defined by a translation vector T = (O_d - O_e) and a rotation matrix R.

Given a point P in 3D space the relationship between their P_e and P_d coordinates is given by:

$$P_d = R\left(P_e - T\right) \quad (1)$$

The fundamental matrix formalizes the geometric relationships between the images of a pair of cameras that "observes" the same part of the 3D space. The geometry corresponding to this configuration is generally known as epipolar geometry. The plane defined by the 3D point P and the projection centers of the left chambers (O_e) and right (O_d) intersects the flat-image of the left camera and right lines that are called epipolares straight. On the other hand a straight line defined by the camera projection centers intersects the corresponding flat-image points por epipolos designated. Thus, the left epipolo is the image of the center of the right camera projection and vice versa.

1. MATRIX ESSENTIAL

The P_e, T vectors and P_e- T are contained in a plane which is the epipolar plane. These three vectors are defined in the left camera coordinate system. Because they belong to the same plane check / satisfy the following equation:

$$\left(P_e - T\right)^T (T \times P_e) = 0 \qquad (2)$$

On the other side of the equation (1) it follows that:

$$P_e - T = R^t P_d \qquad (3)$$

Substituting in Equation (2) gives:

$$(R^T P_d)^T (T \times P_e) = 0 \qquad (4)$$

Moreover the vector product can be algebraically expressed by a product of a skew-symmetric matrix (composed of the elements of the vectors) by another vector that is:

$$T \times P_e = S P_e \qquad (5)$$

Where S is given by:

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \qquad (6)$$

Therefore equation (4) can be rewritten as follows:

$$P_d^T E P_e = 0 \qquad (7)$$

Where

E = RS

On the other hand we can replace this equation by P_d by p_d P_and P_e by p_e because

$$p_e = \frac{f_e}{Z_e} P_e \text{ e } p_d = \frac{f_d}{Z_d} P_d \text{ em que } p_e = \begin{bmatrix} x_e & y_e & f_e \end{bmatrix}^T \text{ e } p_d = \begin{bmatrix} x_d & y_d & f_d \end{bmatrix}^T$$

Where f_e and f_d are respectively the focal lengths of the left and right camera. So you can be written:

$$p_d^T E p_e = 0 \qquad (8)$$

The matrix E is referred to as essential matrix. Note that this equation the coordinates of the points in the images are not expressed in pixels but in any metric unit (eg mm). The matrix E has 2 characteristic.
If we interpret the coordinates of the points in projectivamente images then u_d given by:

$$u_d = E p_e \qquad (9)$$

It represents the coordinates of a line in the right image. This line is the corresponding epipolar line to p_e point of the left image. This epipolar line contains the p_d point (the right image). The epipolar line also passes through epipolo the right image.

2. FUNDAMENTAL MATRIX

Are K_e and K_d matrices of the intrinsic parameters of the left and right cameras respectively. The coordinates of p and p_d points in pixels are then given by:

$$\hat{p}_e = K_e p_e \text{ e } \hat{p}_d = K_d p_d$$

where one strip,

$$p_e = K_e^{-1} \hat{p}_e \text{ e } p_d = K_d^{-1} \hat{p}_d$$

that substituted in equation (8) gives

$$\hat{p}_d^T F \hat{p}_e = 0 \qquad (10)$$

At where

$$F = K_d^{-T} E K_e^{-1} \qquad (11)$$

The matrix F is called the fundamental matrix. As in the case of the key matrix

$$\hat{u}_d = F \hat{p}_e$$

is the epipolar line on the right image corresponding to the point P^ e. Note that in this case the coordinates of the points in the images are in pixels. The fundamental matrix as the essential, characteristic has 2.

3. ESTIMATION OF FUNDAMENTAL MATRIX: ALGORITHM OF 8 POINTS

Consider a set of corresponding points on the left and right images. Each pair of corresponding points satisfies the equation (10). The matrix M is a 3x3 matrix. So we have 9 unknowns. However the equation (10) is a homogeneous equation which makes it can be multiplied by any constant other than zero. Consequently, the matrix F is defined within a scale factor. This degree of freedom decreases in the number of unknowns to 8. So if we have eight or more pairs of corresponding points in the left and right images can estimate the F matrix (each pair of points allows you to set an equation whose unknowns elements of F) matrix. However it is important to ensure that the points used do not constitute a degenerate configuration, for example, all belonging to a plane. To estimate the matrix F in the sense of least squares, imposing the restriction that the norm of the vector of unknowns (the elements of the matrix F) is unitary, you can use it to "singular value decomposition" (SVD). The equation is the system matrix obtained from the equation (10). The singular value decomposition of the matrix A gives

$$A = UDV^T \qquad (11)$$

The matrices U and V are orthogonal matrices. The matrix D contains on its diagonal the
singular values. The solution of the equation system is given by the column of the matrix V to the single null singular value of the matrix A. In practice, due to noise, the matrix 9 has characteristic that is, no single value is zero. The solution then is the column of matrix V corresponding to the smallest singular value of the matrix A.

The F matrix that results from this estimation process should have characteristic 2. However due to the noise and inaccuracies in the estimation process only rarely F matrix has characteristic 2. It is therefore necessary to impose this condition. To this it makes the singular value decomposition of the matrix F obtaining:

$$F = UDV^T$$

then replaced with D matrix by a new matrix D 'that is equal to matrix D with except the smallest singular value becomes equal to 0. recalculate the matrix F is then to ensure the characteristic 2. Thus one obtains the matrix F 'is given by:

$$F' = UD'V^T$$

A very important element for a good estimation of the matrix F is the normalization of the data. Standardization aims to make the most well-conditioned problem numerically. No standardization of the elements of the matrix have large variations. Interests reduce the range of possible variations in the elements of the matrix A, which depend on the coordinates of the pixel. To this makes up a translation and a scaling factor change of each mode images at the origin of the coordinate system is situated at the center of mass of the pixels used in the estimation of the matrix F and also such that the mean square distance of the origin pixel is 1. Consider the points P i of an image with coordinates

$$\hat{p}_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^t \text{, em que } i = 1,\ldots,n\,.$$

The center of mass is then coordinates

$$\bar{x} = \sum\nolimits_i x_i / n \text{ e } \bar{y} = \sum\nolimits_i y_i / n\,.$$

Be

$$\bar{d} = \frac{\sum_i \sqrt{\left(x_i - \bar{x}\right)^2 + \left(y_i - \bar{y}\right)^2}}{n\sqrt{2}}$$

With the values _x, _y _d and for each of the two images are constructed and T_e T_d matrices that allow obtaining the two images in the normalized coordinate by:

$$\hat{p}_e^{'} = T_e \hat{p}_e \qquad \text{e} \qquad \hat{p}_d^{'} = T_d \hat{p}_d$$

The estimation matrix F is done so with these coordinates. The resulting matrix of the estimation process is the F matrix is given by:

$$\hat{F} = T_d^{-T} F T_e^{-1}$$

Estimated F matrix F is obtained matrix by:

$$F = T_d^T \hat{F} T_e$$

Algorithm of 8 points

Consider-n corresponding points (the left and right images) with n => 8. Mark the points corresponding to the hand or else automatically determine them first (using the corner detector from previous work) and establish correspondence by hand.
<u>Please note: the points should not all be coplanar.</u>

1. Normalize the coordinates corresponding points in the two images using
$\hat{p}_e^{'} = T_e \hat{p}_e$ e $\hat{p}_d^{'} = T_d \hat{p}_d$.

2. Build the homogeneous system of equations defined by equation 10. Let A be the matrix of the system nx 9 and A ^ T = UDV its singular value decomposition.
3. The elements of the matrix F (within a scale factor) are given by the elements of column V of the matrix corresponding to the smallest singular value of A.

4. To impose the restriction of the characteristic matrix F decomposes the matrix F singular value:
$\hat{F} = \hat{U}\hat{D}\hat{V}^T$

5. Place the smallest singular value in the array D to zero and becomes the matrix D 'equal to that array.
6. Calculate the fundamental matrix again using:
$F^{'} = UD^{'}V^T$

7. "Denormalisation": the fundamental matrix F is then obtained by,
$F = T_d^T F^{'} T_e$

4. PETS OF EPIPOLOS

The Pipolo and the left image e_e belongs to all epipolares straight. Thus, whatever ^ is obtained P_d

$$\hat{p}_d^T F \hat{e}_e = 0$$

Which means

$$F \hat{e}_e = 0$$

Since the F matrix has characteristic 2 (being a 3x3 matrix), it is concluded that the left epiplo e_e matches to space null right gives matrix F. Per other side O epipolo right ê_d corresponds to the right null space of the matrix F ^ T. Consequently to determine the two epipolos are estimated spaces zero duty of the matrices F and F ^ t.

Algorithm to estimate the location of epipólos
Given the fundamental matrix F

1 Decomposes the singular value matrix F

$$F = UDV^T$$

2. e_e epipolo is the column of the matrix V corresponding to singular value null matrix F.

3. ê_d is epipolo of the corresponding column of the singular value matrix U null matrix F.

Work objectives
Part 1

1 - Estimate the fundamental matrix. To this must use the three images taken by each camera, because The matrix fundamental no can to be calculated with The images in one only plan (Lef01.jpg, left02.jpg. left03.jpg like this as right01.jpg, right02.jpg, and right03.jpg). Use the "detectCheckerboardpoints" to detect the coordinates corners. Establish correspondence manually and calculate the fundamental matrix using the algorithm of 8 points.

```
clear all;
close all;
clc

l1 = imread('left01.jpg');
l2 = imread('left02.jpg');
l3 = imread('left03.jpg');
r1 = imread('right01.jpg');
r2 = imread('right02.jpg');
r3 = imread('right03.jpg');

[imagePointsL1,boardSizeL1] = detectCheckerboardPoints(l1);
[imagePointsL2,boardSizeL2] = detectCheckerboardPoints(l2);
[imagePointsL3,boardSizeL3] = detectCheckerboardPoints(l3);
[imagePointsR1,boardSizeR1] = detectCheckerboardPoints(r1);
[imagePointsR2,boardSizeR2] = detectCheckerboardPoints(r2);
[imagePointsR3,boardSizeR3] = detectCheckerboardPoints(r3);

% figure('Name','left_1');
% imshow(l1); hold on;
% plot(imagePointsL1(:,1), imagePointsL1(:,2), 'ro');
%
% figure('Name','right_1');
% imshow(r1); hold on;
% plot(imagePointsR1(:,1), imagePointsR1(:,2), 'ro');
%
% figure('Name','left_2');
% imshow(l2); hold on;
% plot(imagePointsL2(:,1), imagePointsL2(:,2), 'ro');
%
% figure('Name','right_2');
% imshow(r2); hold on;
% plot(imagePointsR2(:,1), imagePointsR2(:,2), 'ro');
%
% figure('Name','left_3');
% imshow(l3); hold on;
% plot(imagePointsL3(:,1), imagePointsL3(:,2), 'ro');
%
% figure('Name','right_3');
% imshow(r3); hold on;
% plot(imagePointsR3(:,1), imagePointsR3(:,2), 'ro');

%1 - Algorithm of the 8 points
pointsLeft = [imagePointsL1; imagePointsL2; imagePointsL3];
```

```
pointsRight = [imagePointsR1; imagePointsR2; imagePointsR3];

allLeftX = pointsLeft(:,1);
allLeftY = pointsLeft(:,2);
allRightX = pointsRight(:,1);
allRightY = pointsRight(:,2);

%Estimate camera Params
%squareSize = 38;
%[worldPointsL1] = generateCheckerboardPoints(boardSizeL1,squareSize);
%[cameraParamsL1,imagesUsed,estimationErrors] =
estimateCameraParameters(imagePoints,worldPoints);
nbPoints = size(pointsLeft);

%Means and delta
deltaL = 0;
deltaR = 0;
meanLX = mean(allLeftX);
meanRX = mean(allRightX);
meanLY = mean(allLeftY);
meanRY = mean(allRightY);

for i=1:nbPoints(1)
    deltaL = deltaL+sqrt((allLeftX(i)-meanLX)^2+(allLeftY(i)-meanLY)^2);
    deltaR = deltaR+sqrt((allRightX(i)-meanRX)^2+(allRightY(i)-meanRY)^2);
end
deltaL = deltaL/(nbPoints(1)*sqrt(2));
deltaR = deltaR/(nbPoints(1)*sqrt(2));

TL = [1, 0, -meanLX; 0, 1, -meanLY; 0, 0,  deltaL];%I*means+delta
TR = [1, 0, -meanRX; 0, 1, -meanRY; 0, 0,  deltaR];%I*means+delta

%fill the hats
pLHat = (TL*[pointsLeft ones(nbPoints(1),1)]'/deltaL)';
pRHat = (TR*[pointsRight ones(nbPoints(1),1)]'/deltaR)';

%A and FHat
A = [pLHat(:,1).*pRHat(:,1), pRHat(:,1).*pLHat(:,2), pRHat(:,1)...
    pLHat(:,1).*pRHat(:,2), pLHat(:,2).*pRHat(:,2), pRHat(:,2)...
    pLHat(:,1), pLHat(:,2), ones(nbPoints(1),1)];
[U,D,V]=svd(A);
[M,I] = min(diag(D));
FHat = [V(1:3,I)';V(4:6,I)';V(7:9,I)'];
[U2,D2,V2] = svd(FHat);
[M2,I2] = min(diag(D2));
D2(I2,I2)=0;

%FPrime and F
FPrime = U2*D2*V2';
F = TR'*FPrime*TL;
```

2 - Estimate epipolos and represent - them on the images;

3 - Representing the straights images corresponding epipolares. What conclusion as to camera movement between images?

```
%2 - Epipoles
[UF,DF,VF] = svd(F);
[MF,IF]=min(diag(DF));

eHatLeft = VF(:,IF);
```

```
eHatLeft = eHatLeft/eHatLeft(3);
eHatRight = UF(:,IF);
eHatRight = eHatRight/eHatRight(3);

%Print that
%plot(eHatLeft(1), eHatLeft(2),'b+');
%hold on;
%imshow(l1);
% figure('name','Epipoles on left1')
% plot(eHatLeft(1),eHatLeft(2),'r.','MarkerSize',25)
% hold on, imshow(l1), hold on
% for i=1:nbPoints(1)
%     plot([eHatLeft(1) allLeftX(i)],[eHatLeft(2) allLeftY(i)])
%     plot(allLeftX(i),allLeftY(i),'r.','MarkerSize',4)
% end
%
% figure('name','Epipoles on right1')
% plot(eHatRight(1),eHatRight(2),'r.','MarkerSize',25)
% hold on, imshow(r1), hold on
% for i=1:nbPoints(1)
%     plot([eHatRight(1) allRightX(i)],[eHatRight(2) allRightY(i)])
%     plot(allRightX(i),allRightY(i),'r.','MarkerSize',4)
% end
% Values apparently too wrong to give a satisfying answer
```

4 - Compute the fundamental matrix using the command Matlab "EstimateFundamentalMatrix". Compare the two matrices visually, making display of epipolos and corresponding epipolares straight.

```
%4 - EstimateFundamentalMatrix
FMat = estimateFundamentalMatrix(pointsLeft,pointsRight);

[UMat,DMat,VMat] = svd(FMat);
[MMat,IMat]=min(diag(DMat));

eHatLeftMat = VMat(:,IMat);
eHatLeftMat = eHatLeftMat/eHatLeftMat(3);
eHatRightMat = UMat(:,IMat);
eHatRightMat = eHatRightMat/eHatRightMat(3);
%Same reason for no printing
```

	1	2	3
1	-2.4153e-04	-0.0123	3.9585
2	-0.0070	6.7826e-05	65.1666
3	1.1282	-59.6277	-2.1215e+03

Figure 1 : Computed Fundamental Matrix

	1	2	3
1	1.0695e-07	1.0360e-05	-0.0030
2	-3.6953e-07	-4.7361e-07	-0.0379
3	4.7948e-04	0.0357	0.9986

Figure 2 : Estimated Fundamental Matrix

Part 2

Knowing the parameter matrices intrinsic chambers left and right are respectively:

$$K_{esq} = \begin{bmatrix} 533.0031 & 0 & 341.586 \\ 0 & 533.1526 & 234.259 \\ 0 & 0 & 1 \end{bmatrix} e$$

$$K_{dir} = \begin{bmatrix} 536.9826 & 0 & 326.472 \\ 0 & 536.56938 & 249.3326 \\ 0 & 0 & 1 \end{bmatrix}$$

5 - Determine the essential matrix;
6 - From the essential matrix determines the rotation and translation between the two chambers using one of the algorithms described in the lecture;
7 - Make The reconstruction tion 3D (determine at coordinates 3D) of points what used for estimate The matrix essential. Use O algorithm described at class theoretical or O command Matlab "Triangulate". if use O command "Triangulate" has what create The structure "StereoParameters".

```
%5 - Essential Matrix
KR=[533.0031,0,341.568; 0,533.1526,234.259; 0,0,1];
KL=[536.9826,0,326.472; 0,536.56938,249.3326; 0,0,1];
E = KR'*F*KL

%6 - Translation and rotation
[UE DE VE] = svd(E);
TE=UE(:,3);
RE=UE*[0,-1,0;1,0,0;0,0,1]*VE';

%7 - 3D Reconstruction
```

	1	2	3
1	-69.1299	-3.5181e+03	433.0417
2	-2.0131e+03	19.4031	3.3529e+04
3	-323.0040	-3.4240e+04	-1.6107e+03

Figure 3 : Essential Matrix