Colorado School of Mines

Computer Vision

Professor William Hoff
Dept of Electrical Engineering & Computer Science
http://inside.mines.edu/~whoff/
Fundamental Matrix

There’s even a song about it! See http://danielwedge.com/fmatrix/
Recall the Essential Matrix

• Is the matrix $E$, that relates the image of a point in one camera to its image in the other camera, given a translation and rotation

$$p_0^T E p_1 = 0$$

• where

$$E = [t]_x R$$

• and

- $p_0, p_1$ are corresponding points (normalized image coordinates)
Fundamental Matrix

• To work with the essential matrix we have to know the intrinsic camera parameter matrix \( K \)
  – We use \( p_0, p_1 \) which are normalized image coordinates (i.e., \( x = X/Z, y = Y/Z \))
  – We find normalized image coords using \( p = K^{-1} u \), where \( u \) are the un-normalized image coords

• If we don’t know the intrinsic parameter matrix
  – all we have is the un-normalized image points
  – we can still relate the views
  – We use the fundamental matrix \( F \)
Fundamental Matrix

• We have
  \[ p_0^T E p_1 = 0 \]

• Let
  \[ p_1 = K^{-1} u_1 \]
  \[ p_0^T = (K^{-1} u_0)^T = u_0^T K^{-T} \]

• Then
  \[ u_0^T K^{-T} E K^{-1} u_1 = 0 \]

• or
  \[ u_0^T F u_1 = 0 \]

• where F is the fundamental matrix
  \[ F = K^{-T} E K^{-1} \]

• Note
  – \( F \) is defined in terms of pixel coordinates
  – You can still reconstruct the epipolar lines using \( F \)

Also note that

\[ E = K^T F K \]
Solving for F

• We solve for F using the same methods as we used to solve for E
  – Except the corresponding points are in un-normalized coordinates

• We have
  \[ \mathbf{u}_0^T \mathbf{F} \mathbf{u}_1 = 0 \]
  \[ \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} x_0 & y_0 & 1 \end{pmatrix} = 0 \]

• Write as \( \mathbf{A} \mathbf{x} = \mathbf{0} \), where \( \mathbf{x} = (F_{11}, F_{12}, F_{13}, \ldots, F_{33}) \)
  \[ \begin{pmatrix} x_0 & y_0 & 1 \end{pmatrix} \begin{pmatrix} x_0 & x_1 & y_0 & y_1 & x_1 & y_1 & 1 \end{pmatrix} = 0 \]
Residual Error

• For each image point $p_2$, the corresponding point $p_1$ in the other image should ideally lie exactly on the epipolar line $l = F \ast p_2$

• If there is noise, the residual error = distance from the actual point $p_1$ to the epipolar line

• Distance from point $p_1 = (x_1, y_1, 1)^T$ to line with parameters $l = (a, b, c)^T$ is

$$d = \frac{\text{abs}(p_1^T \ast l)}{\sqrt{a^2+b^2}}$$

See http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html
Matlab code to compute residuals

% Get error residuals for all points, using the final F.
dp = zeros(N,1);
for i=1:N
    % The product l=F*p2 is the equation of the epipolar line corresponding
    % to p2, in the first image. Here, l=(a,b,c), and the equation of the
    % line is ax + by + c = 0.
x2 = pts2(i,:);         % Point in second image
l = F * [x2;1];         % Epipolar line in first image
    % The equation of the line is ax + by + c = 0.
    % The distance from a point p1=(x1,y1,1) to a line with parameters
    % l=(a,b,c) is   d = abs(p1' * l)/sqrt( a^2 + b^2 )
    % (see
    % http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html)
x1 = pts1(i,:);        % Point in first image
dp(i) = abs(([x1;1]' * l))/sqrt( l(1)^2 + l(2)^2 );
end
Reconstruction

• With the essential matrix we could reconstruct the scene points to a scale factor (Euclidean reconstruction)

• We can’t do Euclidean reconstruction with the fundamental matrix; however we can do a projective reconstruction
  – Orthogonal lines or planes in the world may not end up being reconstructed as orthogonal

http://www.cse.iitd.ernet.in/~suban/vision/multiple/node11.html