CS5670: Computer Vision

Two-view geometry
Reading

• Reading: Szeliski 1st Edition, Ch. 7.2
Fundamental matrix song

http://danielwedge.com/fmatrix/
Announcements

• Project 4 (Stereo) released today, due 4/20 by 7pm
  – To be done in groups of 2
• Two more topics on geometry, then on to recognition and deep learning
• Please enter any midterm regrade requests by this Wednesday

• Questions? sli.do, enter code #cs5670
Project 4 Demo
Back to stereo

- Where do epipolar lines come from?
Two-view geometry

• Where do epipolar lines come from?

- 3D point lies somewhere along \( r \)
- Epipolar plane
- Epipolar line
- Epipolar line (projection of \( r \))
- Image 1
- Image 2
**Fundamental matrix**

- This *epipolar geometry* of two views is described by a Very Special 3x3 matrix \( \mathbf{F} \), called the *fundamental matrix*.
- \( \mathbf{F} \) maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point \( \mathbf{p} \) is: \( \mathbf{Fp} \)

*Epipolar constraint* on corresponding points: \( \mathbf{q}^T \mathbf{Fp} = 0 \)
Fundamental matrix

- Two Special points: $e_1$ and $e_2$ (the *epipoles*): projection of one camera into the other
Fundamental matrix

- Two Special points: \( e_1 \) and \( e_2 \) (the epipoles): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole
- Epipoles may or may not be inside the image
Epipoles
Properties of the Fundamental Matrix

- $Fp$ is the epipolar line associated with $p$
- $F^T q$ is the epipolar line associated with $q$
- $Fe_1 = 0$ and $F^T e_2 = 0$
- $F$ is rank 2
- How many parameters does $F$ have?
Example
Demo
Fundamental matrix

• Why does $F$ exist?
• Let’s derive it...
Fundamental matrix – calibrated case

\[ \begin{align*}
K_1 & : \text{intrinsics of camera 1} & \quad K_2 & : \text{intrinsics of camera 2} \\
R & : \text{rotation of image 2 w.r.t. camera 1} \\
\tilde{p} = K_1^{-1} p & : \text{ray through } p \text{ in camera 1’s (and world) coordinate system} \\
\tilde{q} = K_2^{-1} q & : \text{ray through } q \text{ in camera 2’s coordinate system}
\end{align*} \]
Fundamental matrix – calibrated case

• $\tilde{p}$, $R^T\tilde{q}$, and $t$ are coplanar
• epipolar plane can be represented as with its normal $t \times \tilde{p}$

$$(R^T\tilde{q})^T(t \times \tilde{p}) = 0$$
Fundamental matrix – calibrated case

\[
(R^T \tilde{q})^T (t \times \tilde{p}) = 0
\]

\[
\tilde{q}^T R (t \times \tilde{p}) = 0
\]
Fundamental matrix – calibrated case

- One more substitution:
  - Cross product with \( \mathbf{t} \) (on left) can be represented as a 3x3 matrix

\[
[t]_\times = \begin{bmatrix}
0 & -t_z & t_y \\
t_z & 0 & -t_x \\
-t_y & t_x & 0
\end{bmatrix}
\]

\[
\mathbf{t} \times \tilde{\mathbf{p}} = [\mathbf{t}]_\times \tilde{\mathbf{p}}
\]
Fundamental matrix – calibrated case

\[
\tilde{q}^T R (t \times \tilde{p}) = 0
\]

\[
\tilde{q}^T R [t] \times \tilde{p} = 0
\]
Fundamental matrix – calibrated case

\[ \tilde{p} = K_1^{-1} p \quad \text{: ray through } p \text{ in camera 1’s (and world) coordinate system} \]

\[ \tilde{q} = K_2^{-1} q \quad \text{: ray through } q \text{ in camera 2’s coordinate system} \]

\[
\tilde{q}^T R [t] \times \tilde{p} = 0
\]

\[
\tilde{q}^T E \tilde{p} = 0
\]

\[ E \leftarrow \text{the Essential matrix} \]
Cross-product as linear operator

**Useful fact:** Cross product with a vector \( \mathbf{t} \) can be represented as multiplication with a \((\text{skew-symmetric})\) 3x3 matrix

\[
[t]_\times = \begin{bmatrix}
0 & -t_z & t_y \\
t_z & 0 & -t_x \\
-t_y & t_x & 0
\end{bmatrix}
\]

\( \mathbf{t} \times \tilde{\mathbf{p}} = [\mathbf{t}]_\times \tilde{\mathbf{p}} \)
**Fundamental matrix – uncalibrated case**

\[ q^T K_2^{-T} R [t] \times K_1^{-1} p = 0 \]

- \( K_1 \): intrinsics of camera 1
- \( K_2 \): intrinsics of camera 2
- \( R \): rotation of image 2 w.r.t. camera 1
- \( F \): the Fundamental matrix
Rectified case

\[ R = I_{3 \times 3} \]
\[ t = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \]
\[ E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \]
\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
-1 \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 \\
-1 \\
1
\end{bmatrix}
= 0
\]

\[
-b + y = 0
\]

\[
b = y
\]
Stereo image rectification

• reproject image planes onto a common plane
  – plane parallel to the line between optical centers

• pixel motion is horizontal after this transformation

• two homographies, one for each input image reprojection
Relationship between F matrix and homography?

Images taken from the same center of projection? Use a homography!
Questions?
Estimating F

• If we don’t know $K_1$, $K_2$, $R$, or $t$, can we estimate $F$ for two images?

• Yes, given enough correspondences
Estimating F – 8-point algorithm

• The fundamental matrix $F$ is defined by

$$x'^T F x = 0$$

for any pair of matches $x$ and $x'$ in two images.

• Let $x = (u, v, 1)^T$ and $x' = (u', v', 1)^T$, $F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$

each match gives a linear equation

$$uu'f_{11} + vu'f_{12} + u'f_{13} + uv'f_{21} + vv'f_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$
8-point algorithm

\[
\begin{bmatrix}
    u_1u'_1 & v_1u'_1 & u'_1 & u_1v'_1 & v_1v'_1 & v'_1 & u_1 & v_1 & 1 \\
    u_2u'_2 & v_2u'_2 & u'_2 & u_2v'_2 & v_2v'_2 & v'_2 & u_2 & v_2 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    u_nu'_n & v_nu'_n & u'_n & u_nv'_n & v_nv'_n & v'_n & u_n & v_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
    f_{11} \\
    f_{12} \\
    f_{13} \\
    f_{21} \\
    f_{22} \\
    f_{23} \\
    f_{31} \\
    f_{32} \\
    f_{33}
\end{bmatrix}
= 0
\]

- Like with homographies, instead of solving \( Af = 0 \), we seek \( f \) to minimize \( \|Af\| \), least eigenvector of \( A^TA \).
8-point algorithm – Problem?

- **F** should have rank 2
- To enforce that **F** is of rank 2, **F** is replaced by **F'** that minimizes \( \| \mathbf{F} - \mathbf{F}' \| \) subject to the rank constraint.
  
  - This is achieved by SVD. Let \( \mathbf{F} = \mathbf{U} \Sigma \mathbf{V}^T \), where

\[
\Sigma = \begin{bmatrix}
\sigma_1 & 0 & 0 \\
0 & \sigma_2 & 0 \\
0 & 0 & \sigma_3 \\
\end{bmatrix}, \quad \text{let} \quad \Sigma' = \begin{bmatrix}
\sigma_1 & 0 & 0 \\
0 & \sigma_2 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

then \( \mathbf{F}' = \mathbf{U} \Sigma' \mathbf{V}^T \) is the solution (closest rank-2 matrix to \( \mathbf{F} \))
8-point algorithm

% Build the constraint matrix
A = [x2(1,:)'.*x1(1,:)'  x2(1,:)'.*x1(2,:)'  x2(1,:)'
    x2(2,:)'.*x1(1,:)'  x2(2,:)'.*x1(2,:)'  x2(2,:)'
    x1(1,:)'            x1(2,:)'          ones(npts,1) ];

[U,D,V] = svd(A);

% Extract fundamental matrix from the column of V
% corresponding to the smallest singular value.
F = reshape(V(:,9),3,3)';

% Enforce rank2 constraint
[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';
8-point algorithm

- Pros: linear, easy to implement and fast
- Cons: susceptible to noise
Problem with 8-point algorithm

\[
\begin{bmatrix}
    u_1 u_1' & v_1 u_1' & u_1' & u_1 v_1' & v_1 v_1' & v_1' & u_1 & v_1 & 1 \\
    u_2 u_2' & v_2 u_2' & u_2' & u_2 v_2' & v_2 v_2' & v_2' & u_2 & v_2 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    u_n u_n' & v_n u_n' & u_n' & u_n v_n' & v_n v_n' & v_n' & u_n & v_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
    f_{11} \\
    f_{12} \\
    f_{13} \\
    f_{21} \\
    f_{22} \\
    f_{23} \\
    f_{31} \\
    f_{32} \\
    f_{33}
\end{bmatrix} = 0
\]

Orders of magnitude difference between column of data matrix → least-squares yields poor results
Normalized 8-point algorithm

normalized least squares yields good results

Transform image to $\sim[-1,1] \times [-1,1]$
Normalized 8-point algorithm

- Transform input by $\hat{x}_i = T x_i$, $\hat{x}'_i = T x'_i$
- Call 8-point on $\hat{x}_i, \hat{x}'_i$ to obtain $\hat{F}$
- $F = T'^T \hat{F} T$

\[
x'^T F x = 0
\]

\[
\hat{x}'^T T'^T F T \hat{x}^{-1} \hat{x} = 0
\]
Normalized 8-point algorithm

[x1, T1] = normalise2dpts(x1);
[x2, T2] = normalise2dpts(x2);

A = [x2(1,:)'.*x1(1,:)'
x2(2,:)'.*x1(1,:)'
x1(1,:)'  x1(2,:)'  ones(npts,1) ];

[U,D,V] = svd(A);

F = reshape(V(:,9),3,3)';

[U,D,V] = svd(F);
F = U*diag([D(1,1) D(2,2) 0])*V';

% Denormalise
F = T2'*F*T1;
Results (ground truth)

- **Ground truth** with standard stereo calibration
Results (ground truth)

8-point algorithm
Results (normalized 8-point algorithm)
What about more than two views?

• The geometry of three views is described by a $3 \times 3 \times 3$ tensor called the *trifocal tensor*

• The geometry of four views is described by a $3 \times 3 \times 3 \times 3$ tensor called the *quadrifocal tensor*

• After this it starts to get complicated...
Large-scale structure from motion

Dubrovnik, Croatia. 4,619 images (out of an initial 57,845).
Total reconstruction time: 23 hours
Number of cores: 352
Questions?