CS5670: Computer Vision

Two-view geometry
Reading

- Reading: Szeliski (2nd Edition), Chapter 11.3 and 12.1
Announcements

• Project 4 (stereo) due this Friday, March 29, at 8pm
Back to stereo

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

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

- Where do epipolar lines come from?

3D point lies somewhere along \( r \)

\( r \)

epipolar line (projection of \( r \))
Two-view geometry

- Where do epipolar lines come from?

![Diagram showing two viewpoints with epipolar lines]

- 3D point lies somewhere along $r$
- Epipolar line (projection of $r$)
Two-view geometry

- Where do epipolar lines come from?
Fundamental matrix

- This epipolar geometry of two views is described by a very special 3x3 matrix, called the fundamental matrix $F$. 
**Fundamental matrix**

- **Epipolar geometry**, very special 3x3 fundamental matrix $F$
- $F$ maps (homogeneous) **points** in image 1 to **lines** in image 2!
Relationship between F matrix and homography?

Images taken from the same center of projection? Use a homography!
Fundamental matrix

- Epipolar geometry, very special 3x3 fundamental matrix $F$
- $F$ maps (homogeneous) points in image 1 to lines in image 2!
- The epipolar line (in image 2) of point $p$ is: $Fp$

\[
p = \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix}, \quad q = \begin{bmatrix} q_x \\ q_y \\ 1 \end{bmatrix}, \quad l' = Fp = \begin{bmatrix} l'_a \\ l'_b \\ l'_c \end{bmatrix}
\]

\[
q^Tl' = \begin{bmatrix} q_x & q_y & 1 \end{bmatrix} \begin{bmatrix} l'_a \\ l'_b \\ l'_c \end{bmatrix} = q_xl'_a + q_yl'_b + l'_c = 0
\]
Fundamental matrix

- **Epipolar geometry**, very special 3x3 fundamental matrix $F$
- $F$ maps (homogeneous) **points** in image 1 to **lines** in image 2!
- The epipolar line (in image 2) of point $p$ is: $Fp$
- **Epipolar constraint** on corresponding points: $q^T Fp = 0$
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

- $F p$ is the epipolar line associated with $p$
- $F^T q$ is the epipolar line associated with $q$

\[ q^T F p = 0 \implies (F^T q)^T p = 0 \]
Properties of the Fundamental Matrix

- $\mathbf{Fp}$ is the epipolar line associated with $\mathbf{p}$
- $\mathbf{F}^T \mathbf{q}$ is the epipolar line associated with $\mathbf{q}$
- $\mathbf{F} \mathbf{e}_1 = 0$ and $\mathbf{F}^T \mathbf{e}_2 = 0$
- $\mathbf{q}^T \mathbf{Fp} = 0$  $\mathbf{e}_2^T \mathbf{Fp} = 0$  $\mathbf{F}^T \mathbf{e}_2 = 0$  $\mathbf{F} \mathbf{e}_1 = 0$
Properties of the Fundamental Matrix

• $F p$ is the epipolar line associated with $p$

• $F^T q$ is the epipolar line associated with $q$

• $F e_1 = 0$ and $F^T e_2 = 0$

• $F$ is rank 2
Properties of the Fundamental Matrix

- $F_p$ is the epipolar line associated with $p$
- $F^T q$ is the epipolar line associated with $q$
- $F e_1 = 0$ and $F^T e_2 = 0$
- $F$ is rank 2

Q: How many parameters (degrees of freedom) does $F$ have?

$\mathbf{q}^T F_p = 0$ 

$F$ is rank 2 

7 degrees of freedom
Example
Demo

https://www.cs.cornell.edu/courses/cs5670/2023sp/demos/FundamentalMatrix/?demo=demo1
Fundamental matrix

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

$\mathbf{K}_1$ : intrinsics of camera 1

$\mathbf{K}_2$ : intrinsics of camera 2

$\mathbf{R}$ : rotation of image 2 w.r.t. camera 1

$\tilde{\mathbf{p}} = \mathbf{K}_1^{-1} \mathbf{p}$ : ray through $\mathbf{p}$ in camera 1’s (and world) coordinate system

$\tilde{\mathbf{q}} = \mathbf{K}_2^{-1} \mathbf{q}$ : ray through $\mathbf{q}$ in camera 2’s coordinate system
Fundamental matrix – calibrated case

- \(~\mathbf{p}, \mathbf{R}^T \mathbf{q}\) and \(\mathbf{t}\) are coplanar
- epipolar plane can be represented as with its normal \(\mathbf{t} \times \mathbf{\tilde{p}}\)

\[
(R^T \mathbf{q})^T (\mathbf{t} \times \mathbf{\tilde{p}}) = 0
\]
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 \( \mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix} \) (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

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

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

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

\[ q^T F p = 0 ; \]

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

the Essential matrix
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
- \( q^T F p = 0 \)
- \( F \): the Fundamental matrix
Rectified case

\[
R = I_{3 \times 3} \\
\begin{bmatrix} t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \\
E = R [t]_\times = \begin{bmatrix} 0 & 0 & 0 \\
0 & 0 & -1 \\
0 & 1 & 0 \end{bmatrix}
\]
Working out the math

- For a point \([a, b, 1]^T\) in image 1:
  \[
  \begin{bmatrix}
  0 & 0 & 0 \\
  0 & 0 & -1 \\
  0 & 1 & 0
  \end{bmatrix}
  \begin{bmatrix}
  a \\
  b \\
  1
  \end{bmatrix}
  =
  \begin{bmatrix}
  0 \\
  -1 \\
  b
  \end{bmatrix}
  \]

- Its corresponding point \([x, y, 1]^T\) in image 2 must satisfy:
  \[
  \begin{bmatrix}
  x \\
  y \\
  1
  \end{bmatrix}
  \cdot
  \begin{bmatrix}
  0 \\
  -1 \\
  b
  \end{bmatrix}
  = 0 \iff y = b
  \]
Rectified case

\[ \mathbf{R} = \mathbf{I}_{3 \times 3} \]
\[ \mathbf{t} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \]

\[ \mathbf{E} = \mathbf{R} \begin{bmatrix} \mathbf{t} \end{bmatrix}_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \]
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
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

\[
\begin{bmatrix}
  u_1 u'_1 & v_1 u'_1 & u'_1 & u_1 v'_1 & v_1 v'_1 & v'_1 & u_1 & v_1 & 1 \\
  u_2 u'_2 & v_2 u'_2 & u'_2 & u_2 v'_2 & v_2 v'_2 & v'_2 & u_2 & v_2 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  u_n u'_n & v_n u'_n & u'_n & u_n v'_n & v_n v'_n & v'_n & u_n & v_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{22} \\
f_{23} \\
f_{31} \\
f_{32} \\
f_{33}
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

- Like with homographies, instead of solving $A\mathbf{f} = 0$ we seek unit length $\mathbf{f}$ to minimize $\|A\mathbf{f}\|$: least eigenvector of $A^T A$
8-point algorithm – Problem?

- \( \mathbf{F} \) should have rank 2
- To enforce that \( \mathbf{F} \) is of rank 2, \( \mathbf{F} \) is replaced by \( \mathbf{F}' \) 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

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

    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';

\[
A = U \Sigma V^T \quad \text{SVD of } A (V \text{ is orthogonal)}
\]
\[
A^T A = (U \Sigma V^T)^T (U \Sigma V^T)
\]
\[
A^T A = V \Sigma^T U^T U \Sigma V^T
\]
\[
A^T A = V \Sigma^2 V^T \quad \text{Eigen decomposition of } A^T A
\]
8-point algorithm

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

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

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 $[-1,1] \times [-1,1]$
**Normalized 8-point algorithm**

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

\[
\begin{align*}
\mathbf{x}'^T \mathbf{F} \mathbf{x} &= 0 \\
\hat{\mathbf{x}}'^T T'^{-T} \mathbf{F} T^{-1} \hat{\mathbf{x}} &= 0 \\
\hat{F} &= 
\end{align*}
\]
Normalized 8-point algorithm

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

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,:)'
     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)
Results (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...
Next time: 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
Fundamental matrix song

http://danielwedge.com/fmatrix/
Questions?