CS 4495 Computer Vision

N-Views (2) – Essential and Fundamental Matrices

Aaron Bobick
School of Interactive Computing
Administrivia

• Today: Second half of N-Views (n = 2)

• PS 3: Out tomorrow.
  • Will be due October 12th.
  • Will be based upon last week and today’s material
Two views…and two lectures

• Projective transforms from image to image

• Some more projective geometry
  • Points and lines and planes

• Two arbitrary views of the same scene
  • Calibrated – “Essential Matrix”
  • Two uncalibrated cameras “Fundamental Matrix”
    • Gives epipolar lines
Last time

• Projective Transforms: Matrices that provide transformations including translations, rotations, similarity, affine and finally general (or perspective) projection.

• When 2D matrices are 3x3; for 3D they are 4x4.
Last time: Homographies

• Provide mapping between images (image planes) taken from same center of projection; also mapping between any images of a planar surface.
Last time: Projective geometry

- A line is a *plane* of rays through origin
  - all rays \((x,y,z)\) satisfying: \(ax + by + cz = 0\)

  in vector notation: \(0 = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \)

- A line is also represented as a homogeneous 3-vector \(l\)
**Projective Geometry: lines and points**

**2D Lines:**  
\[ ax + by + c = 0 \]

\[
\begin{bmatrix}
  a & b & c \\
  x & y & 1
\end{bmatrix} = 0
\]

\[ l^T x = 0 \]

Eq of line

\[ l = [a \ b \ c] \Rightarrow [n_x \ n_y \ d] \]

\[ p_1 = [x_1 \ y_1 \ 1] \]

\[ p_2 = [x_2 \ y_2 \ 1] \]

\[ l = p_1 \times p_2 \]

\[ l_1 = [a_1 \ b_1 \ c_1] \]

\[ l_2 = [a_2 \ b_2 \ c_2] \]

\[ x_{12} = l_1 \times l_2 \]
Motivating the problem: stereo

- Given two views of a scene (the two cameras not necessarily having optical axes) what is the relationship between the location of a scene point in one image and its location in the other?
Stereo correspondence

• Determine Pixel Correspondence
  • Pairs of points that correspond to same scene point

Epipolar Constraint
  • Reduces correspondence problem to 1D search along *conjugate epipolar lines*
Epipolar geometry: terms

- **Baseline**: line joining the camera centers
- **Epipole**: point of intersection of baseline with image plane
- **Epipolar plane**: plane containing baseline and world point
- **Epipolar line**: intersection of epipolar plane with the image plane

- All epipolar lines intersect at the epipole
- An epipolar plane intersects the left and right image planes in epipolar lines
Example: converging cameras

Figure from Hartley & Zisserman
From Geometry to Algebra

• So far, we have the explanation in terms of geometry.
• Now, how to express the epipolar constraints algebraically?
Stereo geometry, with calibrated cameras

Main idea
If the stereo rig is calibrated, we know:

how to rotate and translate camera reference frame 1 to get to camera reference frame 2.

Rotation: 3 x 3 matrix $\mathbf{R}$; translation: 3 vector $\mathbf{T}$. 
Stereo geometry, with calibrated cameras

If the stereo rig is calibrated, we know:

- how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

\[
X'_c = RX_c + T
\]
An aside: cross product

\[ \vec{a} \times \vec{b} = \vec{c} \]

Vector cross product takes two vectors and returns a third vector that’s perpendicular to both inputs.

So here, \( \vec{c} \) is perpendicular to both \( \vec{a} \) and \( \vec{b} \), which means the dot product = 0.

\[ \vec{a} \cdot \vec{c} = 0 \]
\[ \vec{b} \cdot \vec{c} = 0 \]
From geometry to algebra

\[
X' = RX + T
\]

\[
T \times X' = \text{Normal to the plane}
\]

\[
T \times X' = T \times RX
\]

\[
X' \cdot (T \times X') = X' \cdot (T \times RX)
\]

\[
0 = X' \cdot (T \times RX)
\]
Another aside: Matrix form of cross product

\[
\vec{a} \times \vec{b} = \begin{bmatrix}
0 & -a_3 & a_2 \\
a_3 & 0 & -a_1 \\
-a_2 & a_1 & 0
\end{bmatrix}
\begin{bmatrix}
b_1 \\
b_2 \\
b_3
\end{bmatrix} = \vec{c}
\]

Can be expressed as a matrix multiplication!!!

\[
[a_x] = \begin{bmatrix}
0 & -a_3 & a_2 \\
a_3 & 0 & -a_1 \\
-a_2 & a_1 & 0
\end{bmatrix}
\]

Notation:

\[
\vec{a} \times \vec{b} = [\vec{a}_x][\vec{b}]
\]

Has rank 2!
From geometry to algebra

\[ X' = RX + T \]

\[ T \times X' = \text{Normal to the plane} \]

\[ = T \times RX \]

\[ X' \cdot (T \times X') = X' \cdot (T \times RX) \]

\[ 0 = X' \cdot (T \times RX) \]
Essential matrix

\[
X' \cdot \left( T \times RX \right) = 0 \\
X' \cdot \left( [T_x]RX \right) = 0
\]

Let \( E = [T \ x]R \)

\[
X'^T EX = 0
\]

\( E \) is called the **essential matrix**, and it relates corresponding image points between both cameras, given the rotation and translation.

Note: these points are in **each camera coordinate systems**.

We know if we observe a point in one image, its position in other image is constrained to lie on line defined by above.
Essential matrix example: parallel cameras

\[
R = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & B \\
0 & -B & 0
\end{bmatrix}
\]

\[
T = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & B \\
0 & -B & 0
\end{bmatrix}
\]

\[
E = [T \times] R = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & B \\
0 & -B & 0
\end{bmatrix}
\]

\[
E = [T \times] R = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & B \\
0 & -B & 0
\end{bmatrix}
\]
Essential matrix example: parallel cameras

\[
E = [T_x]R = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & B \\
0 & -B & 0
\end{pmatrix}
\]

\[
p'\mathbf{E}p = 0 : \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\
0 & 0 & B \\
0 & -B & 0 \end{bmatrix} \begin{bmatrix} x \\
y \\
f \end{bmatrix} = 0
\]

\[
\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\
Bf \\
-By \end{bmatrix} = 0
\]

\[
Bfy' = Bfy \Rightarrow y' = y
\]

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

Given a known point \((x,y)\) in the original image, this is a line in the \((x',y')\) image.
Weak calibration

- Want to estimate world geometry without requiring calibrated cameras
  - Archival videos (already have the pictures)
  - Photos from multiple unrelated users
  - Dynamic camera system

**Main idea:**
- Estimate epipolar geometry from a (redundant) set of point correspondences between two uncalibrated cameras
From before: Projection matrix

- This can be rewritten as a matrix product using homogeneous coordinates:

$$\begin{bmatrix}
S x_{im} \\
S y_{im} \\
S
\end{bmatrix} = K_{int} \Phi_{ext} \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}$$

where:

$$\Phi_{ext} = \begin{bmatrix}
\begin{array}{ccc}
\alpha & \beta & \gamma \\
\delta & \epsilon & \zeta \\
\eta & \theta & \iota
\end{array}
\end{bmatrix}$$

$$K_{int} = \begin{bmatrix}
\frac{-f}{s_x} & 0 & 0 & o_x \\
0 & \frac{-f}{s_y} & 0 & o_y \\
0 & 0 & 1
\end{bmatrix}$$

Note: Invertible, scale x and y, assumes no skew
From before: Projection matrix

- This can be rewritten as a matrix product using homogeneous coordinates:

\[
\begin{bmatrix}
SX_{im} \\
SY_{im} \\
S
\end{bmatrix} = \begin{bmatrix}
K_{int} \Phi_{ext} \\
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}
\]

\[
p_{im} = K_{int} \Phi_{ext} P_w
\]

\[
p_{im} = K_{int} p_c
\]
Uncalibrated case

For a given camera:

\[ p_{im} = K_{int} p_c \]

So, for two cameras (left and right):

\[ p_{c, left} = K_{int, left}^{-1} p_{im, left} \]
\[ p_{c, right} = K_{int, right}^{-1} p_{im, right} \]

Internal calibration matrices, one per camera
Uncalibrated case

\[ \mathbf{p}_{c,\text{right}} = \mathbf{K}_{\text{int, right}}^{-1} \mathbf{p}_{\text{im, right}} \]
\[ \mathbf{p}_{c,\text{left}} = \mathbf{K}_{\text{int, left}}^{-1} \mathbf{p}_{\text{im, left}} \]

From before, the **essential matrix** \( \mathbf{E} \).

\[
\begin{align*}
(\mathbf{K}_{\text{int, right}}^{-1} \mathbf{p}_{\text{im, right}})^{T} \mathbf{E} (\mathbf{K}_{\text{int, left}}^{-1} \mathbf{p}_{\text{im, left}}) &= 0 \\
p_{\text{im, right}}^{T} \left( (\mathbf{K}_{\text{int, right}}^{-1})^{T} \mathbf{E} \mathbf{K}_{\text{int, left}}^{-1} \right) p_{\text{im, left}} &= 0
\end{align*}
\]

“**Fundamental matrix**” \( \mathbf{F} \)

\[
\begin{align*}
p_{\text{im, right}}^{T} \mathbf{F} p_{\text{im, left}} &= 0 \\
or \quad p^{T} \mathbf{F} p' &= 0
\end{align*}
\]
Properties of the Fundamental Matrix

\[ \mathbf{p}^T \mathbf{F} \mathbf{p'} = 0 \]

- \( \mathbf{l} = \mathbf{Fp'} \) is the epipolar line associated with \( \mathbf{p'} \)
- \( \mathbf{l'} = \mathbf{F}^T \mathbf{p} \) is the epipolar line associated with \( \mathbf{p} \)
- Epipoles found by \( \mathbf{Fp'} = 0 \) and \( \mathbf{F}^T \mathbf{p} = 0 \) (Why?)
- \( \mathbf{F} \) is singular (mapping from 2-D point to 1-D family so rank 2 – more later)
Fundamental matrix

- Relates pixel coordinates in the two views
- More general form than essential matrix: we remove need to know intrinsic parameters
- If we estimate fundamental matrix from correspondences in pixel coordinates, can reconstruct epipolar geometry without intrinsic or extrinsic parameters.
Any two views?

• Can we do this for any two views?

• Yes, as long as we can see some similar points between images?

• So how about these…
Different Example: forward motion

courtesy of Andrew Zisserman
Computing $F$ from correspondences

Each point correspondence generates one constraint on $F$

$$\mathbf{p}_{im,\text{right}}^T F \mathbf{p}_{im,\text{left}} = 0$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

Collect $n$ of these constraints

$$\begin{bmatrix} u' u_1 & u' v_1 & u' u_1 & v' u_1 & v' v_1 & v' u_1 & u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Solve for $f$, vector of parameters.
The (in)famous “eight-point algorithm”

<table>
<thead>
<tr>
<th>F_{11}</th>
<th>F_{12}</th>
<th>F_{13}</th>
<th>F_{21}</th>
<th>F_{22}</th>
<th>F_{23}</th>
<th>F_{31}</th>
<th>F_{32}</th>
<th>F_{33}</th>
</tr>
</thead>
<tbody>
<tr>
<td>250906.36</td>
<td>183259.37</td>
<td>921.81</td>
<td>200931.10</td>
<td>148766.13</td>
<td>738.21</td>
<td>272.19</td>
<td>196.81</td>
<td>1.00</td>
</tr>
<tr>
<td>2692.28</td>
<td>131633.03</td>
<td>176.27</td>
<td>6196.73</td>
<td>302975.59</td>
<td>405.71</td>
<td>15.27</td>
<td>746.79</td>
<td>1.00</td>
</tr>
<tr>
<td>416374.23</td>
<td>871694.30</td>
<td>935.47</td>
<td>408110.89</td>
<td>854384.92</td>
<td>916.90</td>
<td>445.10</td>
<td>931.81</td>
<td>1.00</td>
</tr>
<tr>
<td>191183.60</td>
<td>71759.40</td>
<td>410.27</td>
<td>416435.62</td>
<td>374125.90</td>
<td>839.65</td>
<td>455.99</td>
<td>418.65</td>
<td>1.00</td>
</tr>
<tr>
<td>48968.86</td>
<td>30401.76</td>
<td>57.89</td>
<td>298604.57</td>
<td>185309.58</td>
<td>352.87</td>
<td>846.22</td>
<td>525.15</td>
<td>1.00</td>
</tr>
<tr>
<td>164766.04</td>
<td>546559.67</td>
<td>813.17</td>
<td>1998.37</td>
<td>6628.15</td>
<td>9.86</td>
<td>202.65</td>
<td>672.14</td>
<td>1.00</td>
</tr>
<tr>
<td>116407.01</td>
<td>2727.75</td>
<td>138.89</td>
<td>169941.27</td>
<td>3982.21</td>
<td>202.77</td>
<td>838.12</td>
<td>19.64</td>
<td>1.00</td>
</tr>
<tr>
<td>135384.58</td>
<td>75411.13</td>
<td>198.72</td>
<td>411350.03</td>
<td>229127.78</td>
<td>603.79</td>
<td>681.28</td>
<td>379.48</td>
<td>1.00</td>
</tr>
</tbody>
</table>

- In principal can solve with 8 points.
- Better with more – yields homogeneous linear least-squares:
  - Find unit norm vector F yielding smallest residual
  - Remember SVD or substitute a 1?
- What happens when there is noise?
Doing the obvious thing
Rank of F

- Assume we know the homography $H_\pi$ that maps from Left to Right (Full 3x3)
  $$p' = H_\pi p$$
- Let line $l'$ be the epilolar line corresponding to $p$ – goes through epipole $e'$
- So: $l' = e' \times p'$
  $$= e' \times H_\pi p$$
  $$= [e']_x H_\pi p$$
  $$= F p$$
- Rank of $F$ is rank of $[e']_x = 2$
Fix the linear solution

- Use SVD or other method to do linear computation for $F$
- Decompose $F$ using SVD (not the same SVD):
  \[
  F = UDV^T
  \]

- Set the last singular value to zero:
  \[
  D = \begin{bmatrix}
    r & 0 & 0 \\
    0 & s & 0 \\
    0 & 0 & t
  \end{bmatrix} \Rightarrow \hat{D} = \begin{bmatrix}
    r & 0 & 0 \\
    0 & s & 0 \\
    0 & 0 & 0
  \end{bmatrix}
  \]

- Estimate new $F$ from the new $\hat{D}$
  \[
  \hat{F} = U\hat{D}V^T
  \]
That’s better...
Stereo image rectification
Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection

Rectification Example

Some example cool applications…
Photo synth


http://photosynth.net/
Based on [Photo Tourism](#) by Noah Snavely, Steve Seitz, and Rick Szeliski
3D from multiple images

Building Rome in a Day: Agarwal et al. 2009
Summary

- For 2-views, there is a geometric relationship that defines the relations between rays in one view to rays in the other:
  - Calibrated – Essential matrix
  - Uncalibrated – Fundamental matrix.

- This relation can be estimated from point correspondences – both in calibrated cases and uncalibrated.

- Extensions allow combining multiple views to get more geometric information about scenes:
  - SLAM (simultaneous localization and mapping) – you’ll hear about this if we fit it in…