Outline

• Recap camera calibration
• Epipolar Geometry
Oriented and Translated Camera
Degrees of freedom

\[ \mathbf{x} = \mathbf{K}[\mathbf{R} \mathbf{t}] \mathbf{X} \]

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
= 
\begin{bmatrix}
  \alpha & s & u_0 \\
  0 & \beta & v_0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  r_{11} & r_{12} & r_{13} & t_x \\
  r_{21} & r_{22} & r_{23} & t_y \\
  r_{31} & r_{32} & r_{33} & t_z
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]
How to calibrate the camera?

\[
x = K \begin{bmatrix} R & t \end{bmatrix} X
\]

\[
\begin{bmatrix}
su \\
sv \\
s
\end{bmatrix} = \begin{bmatrix}
* & * & * & * \\
* & * & * & * \\
* & * & * & * \\
1 & & & 
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]
How do we calibrate a camera?

\[
\begin{bmatrix}
S_u \\
S_v \\
S
\end{bmatrix}
= \begin{bmatrix}
* & * & * & * \\
* & * & * & * \\
* & * & * & * \\
1
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

\[
\begin{array}{cccc}
880 & 214 & 312.747 & 309.140 & 30.086 \\
43 & 203 & 305.796 & 311.649 & 30.356 \\
270 & 197 & 307.694 & 312.358 & 30.418 \\
886 & 347 & 310.149 & 307.186 & 29.298 \\
745 & 302 & 311.937 & 310.105 & 29.216 \\
943 & 128 & 311.202 & 307.572 & 30.682 \\
476 & 590 & 307.106 & 306.876 & 28.660 \\
419 & 214 & 309.317 & 312.490 & 30.230 \\
317 & 335 & 307.435 & 310.151 & 29.318 \\
783 & 521 & 308.253 & 306.300 & 28.881 \\
235 & 427 & 309.317 & 312.490 & 30.230 \\
665 & 429 & 307.435 & 310.151 & 29.318 \\
655 & 362 & 308.253 & 306.300 & 28.881 \\
427 & 333 & 309.317 & 312.490 & 30.230 \\
412 & 415 & 307.435 & 310.151 & 29.318 \\
746 & 351 & 308.253 & 306.300 & 28.881 \\
434 & 415 & 309.317 & 312.490 & 30.230 \\
525 & 234 & 307.435 & 310.151 & 29.318 \\
716 & 308 & 308.253 & 306.300 & 28.881 \\
602 & 187 & 309.317 & 312.490 & 30.230 \\
\end{array}
\]
Method 1 – homogeneous linear system

\[
\begin{bmatrix}
    su \\
    sv \\
    s
\end{bmatrix}
= \begin{bmatrix}
    m_{11} & m_{12} & m_{13} & m_{14} \\
    m_{21} & m_{22} & m_{23} & m_{24} \\
    m_{31} & m_{32} & m_{33} & m_{34}
\end{bmatrix}
\begin{bmatrix}
    X \\
    Y \\
    Z \\
    1
\end{bmatrix}
\]

- Solve for m’s entries using linear least squares

\[
Ax = 0 \text{ form}
\]

\[
\begin{bmatrix}
    X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\
    0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\
    \vdots \\
    X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\
    0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n
\end{bmatrix}
= \begin{bmatrix}
    0 \\
    0 \\
    \vdots \\
    0 \\
\end{bmatrix}
\]

\[
[U, S, V] = \text{svd}(A); \\
M = V(:, \text{end}); \\
M = \text{reshape}(M, [], 3)';
\]
For project 3, we want the camera center
Estimate of camera center

1.0486  -0.3645  
-1.6851  -0.4004  
-0.9437  -0.4200  
1.0682   0.0699   
0.6077  -0.0771   
1.2543  -0.6454   
-0.2709  0.8635   
-0.4571  -0.3645  
-0.7902  0.0307   
0.7318   0.6382   
-1.0580  0.3312   
0.3464   0.3377   
0.3137   0.1189   
-0.4310  0.0242   
-0.4799  0.2920   
0.6109   0.0830   
-0.4081  0.2920   
-0.1109  -0.2992  
0.5129  -0.0575   
0.1406  -0.4527   

1.5706  -0.1490  0.2598  
-1.5282  0.9695  0.3802  
-0.6821  1.2856  0.4078  
 0.4124 -1.0201 -0.0915  
 1.2095  0.2812 -0.1280  
 0.8819 -0.8481  0.5255  
-0.9442 -1.1583 -0.3759  
 0.0415  1.3445  0.3240  
-0.7975  0.3017 -0.0826  
-0.4329 -1.4151 -0.2774  
-1.1475 -0.0772 -0.2667  
-0.5149 -1.1784 -0.1401  
 0.1993 -0.2854 -0.2114  
-0.4320  0.2143 -0.1053  
-0.7481 -0.3840 -0.2408  
 0.8078 -0.1196 -0.2631  
-0.7605 -0.5792 -0.1936  
 0.3237  0.7970  0.2170  
 1.3089  0.5786 -0.1887  
 1.2323  1.4421  0.4506
Oriented and Translated Camera
Recovering the camera center

\[ \mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X} \]

This is not the camera center \( \mathbf{C} \). It is \(-\mathbf{RC}\) (because a point will be rotated before \( t_x, t_y, \) and \( t_z \) are added)

So we need \(-\mathbf{R}^{-1} \mathbf{K}^{-1} \mathbf{m}_4\) to get \( \mathbf{C} \)

This is \( \mathbf{t} \cdot \mathbf{K} \)

So \( \mathbf{K}^{-1} \mathbf{m}_4 \) is \( \mathbf{t} \)

\( \mathbf{Q} \) is \( \mathbf{K} \cdot \mathbf{R} \). So we just need \(-\mathbf{Q}^{-1} \mathbf{m}_4\)
### Estimate of camera center

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0486</td>
<td>-0.3645</td>
<td>1.5706</td>
</tr>
<tr>
<td>-1.6851</td>
<td>-0.4004</td>
<td>-1.5282</td>
</tr>
<tr>
<td>-0.9437</td>
<td>-0.4200</td>
<td>-0.6821</td>
</tr>
<tr>
<td>1.0682</td>
<td>0.0699</td>
<td>0.4124</td>
</tr>
<tr>
<td>0.6077</td>
<td>-0.0771</td>
<td>1.2095</td>
</tr>
<tr>
<td>1.2543</td>
<td>-0.6454</td>
<td>0.8819</td>
</tr>
<tr>
<td>-0.2709</td>
<td>0.8635</td>
<td>-0.9442</td>
</tr>
<tr>
<td>-0.4571</td>
<td>-0.3645</td>
<td>0.0415</td>
</tr>
<tr>
<td>-0.7902</td>
<td>0.0307</td>
<td>-0.7975</td>
</tr>
<tr>
<td>0.7318</td>
<td>0.6382</td>
<td>-0.4329</td>
</tr>
<tr>
<td>-1.0580</td>
<td>0.3312</td>
<td>-1.1475</td>
</tr>
<tr>
<td>0.3464</td>
<td>0.3377</td>
<td>-0.5149</td>
</tr>
<tr>
<td>0.3137</td>
<td>0.1189</td>
<td>0.1993</td>
</tr>
<tr>
<td>-0.4310</td>
<td>0.0242</td>
<td>-0.4320</td>
</tr>
<tr>
<td>-0.4799</td>
<td>0.2920</td>
<td>-0.7481</td>
</tr>
<tr>
<td>0.6109</td>
<td>0.0830</td>
<td>0.8078</td>
</tr>
<tr>
<td>-0.4081</td>
<td>0.2920</td>
<td>-0.7605</td>
</tr>
<tr>
<td>-0.1109</td>
<td>-0.2992</td>
<td>0.3237</td>
</tr>
<tr>
<td>0.5129</td>
<td>-0.0575</td>
<td>1.3089</td>
</tr>
<tr>
<td>0.1406</td>
<td>-0.4527</td>
<td>1.2323</td>
</tr>
</tbody>
</table>
Epipolar Geometry and Stereo Vision

Chapter 7.2 in Szeliski
• Epipolar geometry
  – Relates cameras from two positions
Depth from Stereo

- Goal: recover depth by finding image coordinate $x'$ that corresponds to $x$
Depth from Stereo

• Goal: recover depth by finding image coordinate $x'$ that corresponds to $x$

• Sub-Problems
  1. Calibration: How do we recover the relation of the cameras (if not already known)?
  2. Correspondence: How do we search for the matching point $x'$?
Correspondence Problem

- We have two images taken from cameras with different intrinsic and extrinsic parameters

- How do we match a point in the first image to a point in the second? How can we constrain our search?
Where do we need to search?
Key idea: Epipolar constraint
Key idea: Epipolar constraint

Potential matches for \( x \) have to lie on the corresponding line \( l' \).

Potential matches for \( x' \) have to lie on the corresponding line \( l \).
Wouldn’t it be nice to know where matches can live? To constrain our 2d search to 1d.
VLFeat’s 800 most confident matches among 10,000+ local features.
Epipolar geometry: notation

- **Baseline** – line connecting the two camera centers
- **Epipoles**
  - intersections of baseline with image planes
  - projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)
Epipolar geometry: notation

- **Baseline** – line connecting the two camera centers
- **Epipoles**
  = intersections of baseline with image planes
  = projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)
Example: Converging cameras
Example: Motion parallel to image plane
Example: Forward motion

What would the epipolar lines look like if the camera moves directly forward?
Example: Forward motion

Epipole has same coordinates in both images.
Points move along lines radiating from $e$: “Focus of expansion”
Epipolar constraint: Calibrated case

\[ \hat{x} = K^{-1} x = X \]
\[ \hat{x}' = K'^{-1} x' = X' \]
\[ \hat{x} \cdot [t \times (R\hat{x}')] = 0 \]

(because \( \hat{x}, R\hat{x}', \) and \( t \) are co-planar)
Essential matrix

\[ \hat{x} \cdot [t \times (R\hat{x}')] = 0 \quad \Rightarrow \quad \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t] \times R \]

Essential Matrix
(Longuet-Higgins, 1981)
The Fundamental Matrix

Without knowing $K$ and $K'$, we can define a similar relation using unknown normalized coordinates

$$\hat{x}^T E \hat{x}' = 0$$
$$\hat{x} = K^{-1} x$$
$$\hat{x}' = K'^{-1} x'$$

$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

**Fundamental Matrix**
(Faugeras and Luong, 1992)
Properties of the Fundamental matrix

\[ x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1} \]

- \( F x' = 0 \) is the epipolar line associated with \( x' \)
- \( F^T x = 0 \) is the epipolar line associated with \( x \)
- \( F e' = 0 \) and \( F^T e = 0 \)
- \( F \) is singular (rank two): \( \det(F)=0 \)
- \( F \) has seven degrees of freedom: 9 entries but defined up to scale, \( \det(F)=0 \)
Estimating the Fundamental Matrix

- **8-point algorithm**
  - Least squares solution using SVD on equations from 8 pairs of correspondences
  - Enforce $\det(F)=0$ constraint using SVD on $F$

- **7-point algorithm**
  - Use least squares to solve for null space (two vectors) using SVD and 7 pairs of correspondences
  - Solve for linear combination of null space vectors that satisfies $\det(F)=0$

- **Minimize reprojection error**
  - Non-linear least squares

Note: estimation of $F$ (or $E$) is degenerate for a planar scene.
8-point algorithm

1. Solve a system of homogeneous linear equations

   a. Write down the system of equations

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

\[
uu'f_{11} + uv'f_{12} + uf_{13} + vu'f_{21} + vv'f_{22} + vf_{23} + u'f_{31} + v'f_{32} + f_{33} = 0
\]

\[
Af = \begin{bmatrix}
u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_nu_n' & u_nv_n' & u_n & v_nu_n' & v_nv_n' & v_n & u_n' & v_n' & 1 \\
\end{bmatrix} \begin{bmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{33}
\end{bmatrix} = 0
\]
8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations
   b. Solve $f$ from $Af=0$ using SVD

   Matlab:
   
   ```matlab
   [U, S, V] = svd(A);
   f = V(:, end);
   F = reshape(f, [3 3])';
   ```
Need to enforce singularity constraint

Fundamental matrix has rank 2: $\det(F) = 0.$

Left: Uncorrected $F$ – epipolar lines are not coincident.

Right: Epipolar lines from corrected $F$. 
8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations
   b. Solve \( f \) from \( Af = 0 \) using SVD

   Matlab:
   
   ```matlab
   [U, S, V] = svd(A);
   f = V(:, end);
   F = reshape(f, [3 3] )';
   ```

2. Resolve \( \det(F) = 0 \) constraint using SVD

   Matlab:

   ```matlab
   [U, S, V] = svd(F);
   S(3,3) = 0;
   F = U*S*V';
   ```
8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations
   b. Solve $f$ from $Af=0$ using SVD

2. Resolve $\det(F) = 0$ constraint by SVD

Notes:
• Use RANSAC to deal with outliers (sample 8 points)
  – How to test for outliers?
Problem with eight-point algorithm

\[
\begin{bmatrix}
u' & u'v & u' & v'u & v'v & v' & u & v
\end{bmatrix}\begin{bmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{22} \\
f_{23} \\
f_{31} \\
f_{32}
\end{bmatrix} = -1
\]
Problem with eight-point algorithm

<table>
<thead>
<tr>
<th></th>
<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>
</tr>
</thead>
<tbody>
<tr>
<td>250906.36</td>
<td>183269.57</td>
<td>921.81</td>
<td>200931.10</td>
<td>146766.13</td>
<td>738.21</td>
<td>272.19</td>
<td>198.81</td>
<td></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></td>
</tr>
<tr>
<td>416374.23</td>
<td>871684.30</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></td>
</tr>
<tr>
<td>191183.60</td>
<td>171759.40</td>
<td>410.27</td>
<td>416435.62</td>
<td>374125.90</td>
<td>893.65</td>
<td>465.99</td>
<td>418.65</td>
<td></td>
</tr>
<tr>
<td>48988.86</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></td>
</tr>
<tr>
<td>164786.04</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></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></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></td>
</tr>
</tbody>
</table>

Poor numerical conditioning
Can be fixed by rescaling the data
The normalized eight-point algorithm

(Hartley, 1995)

• Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels
• Use the eight-point algorithm to compute $F$ from the normalized points
• Enforce the rank-2 constraint (for example, take SVD of $F$ and throw out the smallest singular value)
• Transform fundamental matrix back to original units: if $T$ and $T'$ are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T'^TFT'$
VLFeat’s 800 most confident matches among 10,000+ local features.
Epipolar lines
Keep only the matches that are "inliers" with respect to the "best" fundamental matrix.