Lecture Outline

• Fundamental Matrix estimation
• Binocural Stereo
  – Matching criteria

Based on slides by R. Hartley, A. Zisserman, M. Pollefeys, R. Szeliski and P. Fua
Projective Transformation and Invariance

\[ \hat{x} = Hx, \quad \hat{x}' = H'x' \implies \hat{F} = H'^{-T}FH^{-1} \]

F invariant to transformations of projective 3-space

\[ x = PX = (PH)(H^{-1}X) = \hat{P}\hat{X} \]
\[ x' = P'X = (P'H)(H^{-1}X) = \hat{P}'\hat{X} \]

\( (P, P') \mapsto F \) unique
\( F \mapsto (P, P') \) not unique

canonical form

\[ P = \begin{bmatrix} I & 0 \end{bmatrix}, \quad P' = \begin{bmatrix} M & m \end{bmatrix}, \quad F = \begin{bmatrix} m \end{bmatrix}_x M \quad (F = [e']_x P'P^+) \]
The Projective Reconstruction Theorem

If a set of point correspondences in two views determine the fundamental matrix uniquely, then the scene and cameras may be reconstructed from these correspondences alone, and any two such reconstructions from these correspondences are projectively equivalent.

\[ x_i \leftrightarrow x_i' \]

\[
\begin{align*}
\begin{bmatrix} P_2 \end{bmatrix} &= \begin{bmatrix} P_1 \end{bmatrix} H^{-1} \\
\begin{bmatrix} P'_2 \end{bmatrix} &= \begin{bmatrix} P'_1 \end{bmatrix} H^{-1} \\
X_{2i} &= HX_{1i} \\
\text{except: } & Fx_i = x_i'F = 0
\end{align*}
\]

\[ P_2 \left( HX_{1i} \right) = P_1 H^{-1} HX_{1i} = P_1 X_{1i} = x_i = P_2 X_{2i} \]

⇒ along same ray of \( P_2 \), idem for \( P'_2 \)

two possibilities: \( X_{2i} = HX_{1i} \), or points along baseline

**key result:** allows reconstruction from pair of uncalibrated images
Stratified Reconstruction

(i) Projective reconstruction
(ii) Affine reconstruction
(iii) Metric reconstruction

Out of scope of CS 532
The Essential Matrix

~fundamental matrix for calibrated cameras (remove K)

\[ E = [t]_x R = R[R^T t]_x \]

\[ \hat{x'}^T E \hat{x} = 0 \quad \left( \hat{x} = K^{-1} x; \hat{x'} = K^{-1} x' \right) \]

\[ E = K'^T FK \]

5 d.o.f. (3 for \( R \); 2 for \( t \) up to scale)

E is an essential matrix if and only if two singular values are equal (and the third=0)

\[ E = U \text{diag}(1,1,0)V^T \]
Given E and setting the first camera matrix $P = [I | 0]$, there are four possible solutions for $P'$ (only one solution, however, where a reconstructed point is in front of both cameras)
Fundamental Matrix Estimation
Epipolar Geometry: Basic Equation

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

\[ x' x f_{11} + x' y f_{12} + x' f_{13} + y' x f_{21} + y' y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0 \]

separate known from unknown

\[ [x', x' y, x', y' x, y' y, y', x, y, 1 \| f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^T = 0 \]

(data) (unknowns)

(linear)

\[ \begin{bmatrix}
    x'_1 & x_1 & x'_1 & y_1 & x'_1 & y_1 & x'_1 & y_1 & y_1 & x_1 & y_1 & 1 \\
    x'_n & x_n & x'_n & y_n & x'_n & y_n & x'_n & y_n & y_n & x_n & y_n & 1
\end{bmatrix} f = 0 \]

\[ A f = 0 \]
The Singularity Constraint

\[ \mathbf{e}'^T \mathbf{F} = 0 \quad \mathbf{F}_e = 0 \quad \det \mathbf{F} = 0 \quad \text{rank } \mathbf{F} = 2 \]

SVD from linearly computed \( \mathbf{F} \) matrix (rank 3)

\[ \mathbf{F} = \mathbf{U} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} \mathbf{V}^T = \mathbf{U}_1 \sigma_1 \mathbf{V}_1^T + \mathbf{U}_2 \sigma_2 \mathbf{V}_2^T + \mathbf{U}_3 \sigma_3 \mathbf{V}_3^T \]

Compute closest rank-2 approximation \( \min \| \mathbf{F} - \mathbf{F}' \|_F \)

\[ \mathbf{F}' = \mathbf{U} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} \mathbf{V}^T = \mathbf{U}_1 \sigma_1 \mathbf{V}_1^T + \mathbf{U}_2 \sigma_2 \mathbf{V}_2^T \]
The Singularity Constraint
The Minimum Case - 7 Point Correspondences

\[
\begin{bmatrix}
    x'_1 & x_1 & x'_1 & y_1 & x'_1 & y'_1 & x'_1 & y'_1 & x_1 & y_1 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x'_7 & x_7 & x'_7 & y_7 & x'_7 & y'_7 & x'_7 & y'_7 & x_7 & y_7 & 1
\end{bmatrix}
\]

\[f = 0\]

\[A = U_{7 \times 7} \text{diag}(\sigma_1, \ldots, \sigma_7, 0, 0)V_{9 \times 9}^T\]

\[\Rightarrow A[V_8 V_9] = 0_{9 \times 2}\]

\[x_i^T(F_1 + \lambda F_2)x_i = 0, \forall i = 1 \ldots 7\]

One parameter family of solutions – results in 1 or 3 real solutions

but \(F_1 + \lambda F_2\) not automatically rank 2
The NOT Normalized 8-point Algorithm

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

Transform image to $[-1,1] \times [-1,1]$

normalized least squares yields good results
Some Experiments
Some Experiments
Some Experiments

Residual error:

\[ \sum_{i} d(x'_i, Fx_i)^2 + d(x_i, F^T x'_i)^2 \]

(for all points!)
Recommendations:

1. Do not use unnormalized algorithms
2. Quick and easy to implement: 8-point normalized
3. Better: enforce rank-2 constraint during minimization
4. Best: Maximum Likelihood Estimation (minimal parameterization, sparse implementation)
RANSAC

Step 1. Extract features
Step 2. Compute a set of potential matches
Step 3. do
  Step 3.1 select minimal sample (i.e. 7 matches)
  Step 3.2 compute solution(s) for F
  Step 3.3 determine inliers (verify hypothesis)
until $\Gamma (#\text{inliers}, #\text{samples}) < 95\%$

Step 4. Compute F based on all inliers
Step 5. Look for additional matches
Step 6. Refine F based on all correct matches

$$\Gamma = 1 - \left(1 - \left(\frac{#\text{inliers}}{#\text{matches}}\right)^7\right)^{#\text{samples}}$$

<table>
<thead>
<tr>
<th>#inliers</th>
<th>90%</th>
<th>80%</th>
<th>70%</th>
<th>60%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>#samples</td>
<td>5</td>
<td>13</td>
<td>35</td>
<td>106</td>
<td>382</td>
</tr>
</tbody>
</table>
Finding more matches

restrict search range to neighborhood of epipolar line
(±1.5 pixels)
relax disparity restriction (along epipolar line)
Degenerate Cases

• Degenerate cases
  – Planar scene
  – Pure rotation

• No unique solution
  – Remaining DOF filled by noise
  – Use simpler model (e.g. homography)

• Model selection (Torr et al., ICCV’98, Kanatani, Akaike)
  – Compare H and F according to expected residual error
    (compensate for model complexity)
Stereo Matching

Slides by Rick Szeliski, Pascal Fua and P. Mordohai
Stereo Matching

• Given two or more images of the same scene or object, compute a representation of its shape

• What are some possible applications?
Stereo Matching

• Given two or more images of the same scene or object, compute a representation of its shape

• What are some possible representations?
  – depth maps
  – volumetric models
  – 3D surface models
  – planar (or offset) layers
Stereo Matching

• What are some possible algorithms?
  – match “features” and interpolate
  – match edges and interpolate
  – match all pixels with windows (coarse-fine)
  – use optimization:
    • iterative updating
    • dynamic programming
    • energy minimization (regularization, stochastic)
    • graph algorithms
Rectification

- Project each image onto same plane, which is parallel to the baseline
- Resample lines (and shear/stretch) to place lines in correspondence, and minimize distortion

- Take rectification for granted in this course
Rectification

(a) Original image pair overlayed with several epipolar lines.

(b) Image pair transformed by the specialized projective mapping $H_p$ and $H_p'$. Note that the epipolar lines are now parallel to each other in each image.

BAD!
Rectification

(c) Image pair transformed by the similarity $H_s$ and $H'_s$. Note that the image pair is now rectified (the epipolar lines are horizontally aligned).

(d) Final image rectification after shearing transform $H_s$ and $H'_s$. Note that the image pair remains rectified, but the horizontal distortion is reduced.

GOOD!
Finding Correspondences

- Apply feature matching criterion at all pixels simultaneously
- Search only over epipolar lines (many fewer candidate positions)
Basic Stereo Algorithm

For each epipolar line
  For each pixel in the left image
    - compare with every pixel on same epipolar line in right image
    - pick pixel with minimum match cost
  Improvement: match windows
Disparity

• Disparity $d$ is the difference between the $x$ coordinates of corresponding pixels in the left and right image

$$d = x_L - x_R$$

• Disparity is inversely proportional to depth

$$Z = \frac{bf}{d}$$
Stereo Reconstruction

\[ Z = \frac{bf}{d} \]
Finding Correspondences

• How do we determine correspondences?
  – *block matching* or *SSD* (sum squared differences)

  \[
  SSD(x, y; d) = \sum_{(x', y') \in N(x, y)} \left[ I_L(x', y') - I_R(x' - d, y') \right]^2
  \]

  – *d* is the *disparity* (horizontal motion)

• How big should the neighborhood be?
Neighborhood size

- Smaller neighborhood: more details
- Larger neighborhood: fewer isolated mistakes

\[ w = 3 \quad w = 20 \]
Challenges

• Ill-posed inverse problem
  – Recover 3-D structure from 2-D information

• Difficulties
  – Uniform regions
  – Half-occluded pixels
  – Repeated patterns
Pixel Dissimilarity

- Sum of Squared Differences of intensities (SSD)

\[
SSD(x, y; d) = \sum_{(x', y') \in N(x, y)} [I_L(x', y') - I_R(x' - d, y')]^2
\]

- Sum of Absolute Differences of intensities (SAD)

\[
SAD(x, y; d) = \sum_{(x', y') \in N(x, y)} |I_L(x', y') - I_R(x' - d, y')|
\]

- Zero-mean Normalized Cross-correlation (NCC)

\[
NCC(x, y, d) = \frac{\sum_{i \in W} (I_L(x_i, y_i) - \mu_L)(I_R(x_i - d, y_i) - \mu_R)}{\sigma_L \sigma_R}
\]
Cost/Score Curve
Cost/Score Curve
Fronto-Parallel Assumption

• The disparity is assumed to be the same in the entire matching window
  – equivalent to assuming constant depth
• Avoid having using matching windows that straddle two surfaces
  – Disparity will not be constant for all pixels
• Shift the window around the reference pixel
  – Keep the one with min cost (max NCC)
Rod-shaped Filters

• Instead of square windows aggregate cost in rod-shaped shiftable windows
• Search for one that minimizes the cost (assume that it is an iso-disparity curve)
Alternative Dissimilarity Measures

• Rank and Census transforms
  • Rank transform:
    – Define window containing R pixels around each pixel
    – Count the number of pixels with lower intensities than center pixel in
      the window
    – Replace intensity with rank (0..R-1)
    – Compute SAD on rank-transformed images
  • Census transform:
    – Use bit string, defined by neighbors, instead of scalar rank
  • Robust against illumination changes
Locally Adaptive Support

Apply weights to contributions of neighboring pixels according to similarity and proximity

(a) left support window (b) right support window (c) color difference between (a) and (b)
Locally Adaptive Support

• Similarity in CIE Lab color space:

\[ \Delta c_{pq} = \sqrt{(L_p - L_q)^2 + (a_p - a_q)^2 + (b_p - b_q)^2} \]

• Proximity: Euclidean distance

• Weights:

\[ w(p, q) = k \cdot \exp \left( -\left( \frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p} \right) \right) \]
Locally Adaptive Support: Results