Last lecture

- Passive Stereo
- Spacetime Stereo
Today

• Structure from Motion:
  Given pixel correspondences,
  how to compute 3D structure and camera motion?

Slides stolen from Prof Yungyu Chuang
Epipolar geometry &
fundamental matrix
The epipolar geometry

What if only $C, C', x$ are known?
The epipolar geometry

$C, C', x, x'$ and $X$ are coplanar
The epipolar geometry

All points on $\pi$ project on $l$ and $l'$
The epipolar geometry

Family of planes $\pi$ and lines $l$ and $l'$ intersect at $e$ and $e'$
The epipolar geometry

epipolar pole = intersection of baseline with image plane
epipolar pole = projection of projection center in other image

epipolar plane = plane containing baseline
epipolar line = intersection of epipolar plane with image
The fundamental matrix F

\[ T = C' - C \]

\[ R = p \times p' \]

The equation of the epipolar plane through X is

\[ x \propto KX \]

\[ x' \propto K'R(X - T) \]

\[ p = K^{-1}x \propto X \]

\[ p' = K'^{-1}x' \propto R(X - T) \]

The equation of the epipolar plane through X is

\[ (X - T)^T (T \times p) = 0 \]

\[ (R^T p')^T (T \times p) = 0 \]
The fundamental matrix $F$

$$ (R^T p')^T (T \times p) = 0 $$

$$ T \times p = Sp $$

$$ S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} $$

$$ (R^T p')^T (Sp) = 0 $$

$$ (p'^T R)(Sp) = 0 $$

$$ p'^T E p = 0 \quad \text{essential matrix} $$
The fundamental matrix $F$

$$p'^T E p = 0$$
The fundamental matrix $F$

$$p'^{T}E p = 0$$

Let $M$ and $M'$ be the intrinsic matrices, then

$$p = K^{-1}x \quad \quad \quad p' = K'^{-1}x'$$

$$\begin{align*}
(K'^{-1}x')^{T}E(K^{-1}x) &= 0 \\
x'^{T}K'^{-T}EK^{-1}x &= 0 \\
x'^{T}Fx &= 0 \quad \text{fundamental matrix}
\end{align*}$$
The fundamental matrix $F$

\[ p'^T E p = 0 \]
\[ x'^T F x = 0 \]
The fundamental matrix $F$

- The fundamental matrix is the algebraic representation of epipolar geometry.

- The fundamental matrix satisfies the condition that for any pair of corresponding points $x \leftrightarrow x'$ in the two images

$$x'^T F x = 0 \quad \left( x'^T l' = 0 \right)$$
The fundamental matrix F

F is the unique 3x3 rank 2 matrix that satisfies $x'^{\top}Fx=0$ for all $x \leftrightarrow x'$

1. Transpose: if F is fundamental matrix for $(P,P')$, then $F^T$ is fundamental matrix for $(P',P)$
2. Epipolar lines: $l'=Fx$ & $l=F^Tx'$
3. Epipoles: on all epipolar lines, thus $e'^{\top}Fx=0$, $\forall x$  
   $\Rightarrow e'^{\top}F=0$, similarly $Fe=0$
4. F has 7 d.o.f., i.e. $3x3-1$(homogeneous)-1(rank2)
5. F maps from a point x to a line $l'=Fx$ (not invertible)
The fundamental matrix $F$

- It can be used for
  - Simplifies matching
  - Allows to detect wrong matches
Estimation of 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

\[
\begin{bmatrix}
u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\
u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
f_{11} \\
f_{12} \\
f_{13} \\
f_{21} \\
f_{22} \\
f_{23} \\
f_{31} \\
f_{32} \\
f_{33} \\
\end{bmatrix}
= 0
\]

- In reality, instead of solving \( A\mathbf{f} = 0 \), we seek \( \mathbf{f} \) to minimize \( \| A\mathbf{f} \| \), least eigenvector of \( A^T A \).
8-point algorithm

• To enforce that $F$ is of rank 2, $F$ is replaced by $F'$ that minimizes $\|F - F'\|$ subject to $\det F' = 0$.

• It is achieved by SVD. Let $F = U\Sigma V^\top$, 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 $F' = U\Sigma' V^\top$ is the solution.
8-point algorithm

% Build the constraint matrix
\[ A = \begin{bmatrix} x2(1,:)'\cdot x1(1,:)' & x2(1,:)'.x1(2,:)' & x2(1,:)' \\ x2(2,:)'.x1(1,:)' & x2(2,:)'.x1(2,:)' & x2(2,:)' \\ x1(1,:)' & x1(2,:)' & \text{ones}(npts,1) \end{bmatrix} ; \]

\[ [U,D,V] = \text{svd}(A); \]

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

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

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

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

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

\[
[x1, T1] = \text{normalise2dpts}(x1);
[x2, T2] = \text{normalise2dpts}(x2);
\]

\[
A = \begin{bmatrix}
  x2(1,:)'.*x1(1,:)' & x2(1,:)'.*x1(2,:)' & x2(1,:)' & \ldots & x2(2,:)'.*x1(1,:)' & x2(2,:)'.*x1(2,:)' & x2(2,:)' & \ldots & x1(1,:)' & x1(2,:)'
  & \text{ones}(npts,1)
\end{bmatrix};
\]

\[
[U,D,V] = \text{svd}(A);
\]

\[
F = \text{reshape}(V(:,9),3,3)';
\]

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

% Denormalise
\[
F = T2'*F*T1;
\]
Normalization

function [newpts, T] = normalise2dpts(pts)

c = mean(pts(1:2,:))';  % Centroid
newp(1,:) = pts(1,:) - c(1);  % Shift origin to centroid.
newp(2,:) = pts(2,:) - c(2);

meandist = mean(sqrt(newp(1,:).^2 + newp(2,:).^2));
scale = sqrt(2)/meandist;

T = [scale 0 -scale*c(1)
     0 scale  -scale*c(2)
     0 0 1];
newpts = T*pts;
RANSAC

repeat
    select minimal sample (8 matches)
    compute solution(s) for F
    determine inliers
until \( \Gamma(\#\text{inliers},\#\text{samples}) > 95\% \) or too many times

compute F based on all inliers
Results (ground truth)

- **Ground truth** with standard stereo calibration
Results (8-point algorithm)
Results (normalized 8-point algorithm)
From F to R, T

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

\[ x'^T M'^-T E M^{-1} x = 0 \]

\[ E = M'^T F M \]

If we know camera parameters

\[ E = R[T]_x \]

Hartley and Zisserman, Multiple View Geometry, 2nd edition, pp 259
Application: View morphing
Application: View morphing
Main trick

- Prewarp with a homography to rectify images
- So that the two views are parallel
  - Because linear interpolation works when views are parallel

Figure 4: View Morphing in Three Steps. (1) Original images $I_0$ and $I_1$ are prewarped to form parallel views $\hat{I}_0$ and $\hat{I}_1$. (2) $\hat{I}_s$ is produced by morphing (interpolating) the prewarped images. (3) $\hat{I}_s$ is postwarped to form $I_s$. 
Problem with morphing

- Without rectification

Figure 2: A Shape-Distorting Morph. Linearly interpolating two perspective views of a clock (far left and far right) causes a geometric bending effect in the in-between images. The dashed line shows the linear path of one feature during the course of the transformation. This example is indicative of the types of distortions that can arise with image morphing techniques.
Figure 10: Image Morphing Versus View Morphing. Top: image morph between two views of a helicopter toy causes the in-between images to contract and bend. Bottom: view morph between the same two views results in a physically consistent morph. In this example the image morph also results in an extraneous hole between the blade and the stick. Holes can appear in viewmorphs as well.
Figure 6: View Morphing Procedure: A set of features (yellow lines) is selected in original images $I_0$ and $I_1$. Using these features, the images are automatically prewarped to produce $\hat{I}_0$ and $\hat{I}_1$. The prewarped images are morphed to create a sequence of in-between images, the middle of which, $\hat{I}_{0.5}$, is shown at top-center. $\hat{I}_{0.5}$ is interactively postwarped by selecting a quadrilateral region (marked red) and specifying its desired configuration, $Q_{0.5}$, in $I_{0.5}$. The postwarps for other in-between images are determined by interpolating the quadrilaterals (bottom).
Figure 7: Facial View Morphs. Top: morph between two views of the same person. Bottom: morph between views of two different people. In each case, view morphing captures the change in facial pose between original images $I_0$ and $I_1$, conveying a natural 3D rotation.
Video demo
Triangulation

• Problem: Given some points in correspondence across two or more images (taken from calibrated cameras), \{(u_j,v_j)\}, compute the 3D location \(X\)
**Triangulation**

- **Method I**: intersect viewing rays in 3D, minimize:

\[
\arg \min_X \sum_j \left\| \mathbf{C}_j + s \mathbf{V}_j - \mathbf{X} \right\|
\]

- $\mathbf{X}$ is the unknown 3D point
- $\mathbf{C}_j$ is the optical center of camera $j$
- $\mathbf{V}_j$ is the viewing ray for pixel $(u_j, v_j)$
- $s_j$ is unknown distance along $\mathbf{V}_j$
- Advantage: geometrically intuitive
Triangulation

- **Method II**: solve linear equations in \( X \)
  - advantage: very simple
  \[
  u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}
  \]
  \[
  v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}
  \]

- **Method III**: non-linear minimization
  - advantage: most accurate (image plane error)
Structure from motion
Structure from motion

structure from motion: automatic recovery of **camera motion** and **scene structure** from two or more images. It is a self calibration technique and called **automatic camera tracking** or **matchmoving**.
Applications

- For computer vision, multiple-view shape reconstruction, novel view synthesis and autonomous vehicle navigation.
- For film production, seamless insertion of CGI into live-action backgrounds
Structure from motion

2D feature tracking → 3D estimation → optimization (bundle adjust) → geometry fitting

SFM pipeline
Structure from motion

• Step 1: Track Features
  • Detect good features, Shi & Tomasi, SIFT
  • Find correspondences between frames
    – Lucas & Kanade-style motion estimation
    – window-based correlation
    – SIFT matching
Structure from Motion

• Step 2: Estimate Motion and Structure
  • Simplified projection model, e.g., [Tomasi 92]
  • 2 or 3 views at a time [Hartley 00]
Structure from Motion

- Step 3: Refine estimates
  - “Bundle adjustment” in photogrammetry
  - Other iterative methods
Structure from Motion

- Step 4: Recover surfaces (image-based triangulation, silhouettes, stereo... )
Example: Photo Tourism

Photo Tourism
Exploring photo collections in 3D
Factorization methods
Problem statement
Other projection models

perspective

weak perspective

increasing focal length

increasing distance from camera
SFM under orthographic projection

\[ \mathbf{q} = \Pi \mathbf{p} + \mathbf{t} \]

**2D image point** \( \mathbf{q} \)

**orthographic projection matrix** \( \Pi \)

**3D scene point** \( \mathbf{p} \)

**image offset** \( \mathbf{t} \)

**Trick**
- Choose scene origin to be centroid of 3D points
- Choose image origins to be centroid of 2D points
- Allows us to drop the camera translation:

\[ \mathbf{q} = \Pi \mathbf{p} \]
factorization (Tomasi & Kanade)

projection of $n$ features in one image:

$$
\begin{bmatrix}
q_1 & q_2 & \cdots & q_n
\end{bmatrix}_{2 \times n} = \prod_{p \leq n} \begin{bmatrix}
p_1 & p_2 & \cdots & p_n
\end{bmatrix}_{2 \times 3}
$$

projection of $n$ features in $m$ images

$$
\begin{bmatrix}
q_{11} & q_{12} & \cdots & q_{1n} \\
q_{21} & q_{22} & \cdots & q_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
q_{m1} & q_{m2} & \cdots & q_{mn}
\end{bmatrix}_{2m \times n} = \begin{bmatrix}
\Pi_1 \\
\Pi_2 \\
\vdots \\
\Pi_m
\end{bmatrix}_{2m \times 3}
\begin{bmatrix}
p_1 & p_2 & \cdots & p_n
\end{bmatrix}_{3 \times n}
$$

$W$ measurement, $M$ motion, $S$ shape

Key Observation: $\text{rank}(W) \leq 3$
Factorization

- **Factorization Technique**
  - $W$ is at most rank 3 (assuming no noise)
  - We can use *singular value decomposition* to factor $W$:
    \[
    W = M' S'
    \]
    
    \[
    W = M' S' = (MA^{-1})(AS)
    \]
    
    - Solve for $A$ by enforcing *metric* constraints on $M$
Metric constraints

- Orthographic Camera
  - Rows of $\Pi$ are orthonormal:
    $\Pi_i \Pi_i^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

- Enforcing “Metric” Constraints
  - Compute $A$ such that rows of $M$ have these properties

$$M' A = M$$

Trick (not in original Tomasi/Kanade paper, but in followup work)

- Constraints are linear in $AA^T$:
  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \Pi_i \Pi_i^T = \Pi'_i A (\Pi'_i^T A)^T = \Pi'_i G \Pi'_i^T$ where $G = AA^T$

- Solve for $G$ first by writing equations for every $\Pi_i$ in $M$
- Then $G = AA^T$ by SVD
Results
Extensions to factorization methods

- Paraperspective [Poelman & Kanade, PAMI 97]
- Sequential Factorization [Morita & Kanade, PAMI 97]
- Factorization under perspective [Christy & Horaud, PAMI 96] [Sturm & Triggs, ECCV 96]
- Factorization with Uncertainty [Anandan & Irani, IJCV 2002]
Bundle adjustment
Structure from motion

\[
\hat{u}_{i,j} = f(K, R_j, t_j, x_i)
\]
\[
\hat{v}_{i,j} = g(K, R_j, t_j, x_i)
\]

- How many points do we need to match?
- 2 frames:
  \((R,t)\): 5 dof + 3\(n\) point locations \(\leq\)
  4\(n\) point measurements \(\Rightarrow\)
  \(n \geq 5\)
- \(k\) frames:
  6\((k-1)\)-1 + 3\(n\) \(\leq 2kn\)
- always want to use many more
Bundle Adjustment

\[ \hat{u}_{ij} = f(K, R_j, t_j, x_i) \]
\[ \hat{v}_{ij} = g(K, R_j, t_j, x_i) \]

- What makes this non-linear minimization hard?
  - many more parameters: potentially slow
  - poorer conditioning (high correlation)
  - potentially lots of outliers
Lots of parameters: sparsity

\[ \hat{u}_{ij} = f(K, R_j, t_j, x_i) \]
\[ \hat{v}_{ij} = g(K, R_j, t_j, x_i) \]

- Only a few entries in Jacobian are non-zero

\[ \frac{\partial \hat{u}_{ij}}{\partial K}, \frac{\partial \hat{u}_{ij}}{\partial R_j}, \frac{\partial \hat{u}_{ij}}{\partial t_j}, \frac{\partial \hat{u}_{ij}}{\partial x_i} \]
Robust error models

- Outlier rejection
  - use robust penalty applied to each set of joint measurements

- for extremely bad data, use random sampling [RANSAC, Fischler & Bolles, CACM’81]

\[
\sum_{i} \sigma_{i}^{-2} \rho \left( \sqrt{(u_{i} - \tilde{u}_{i})^{2} + (v_{i} - \tilde{v}_{i})^{2}} \right)
\]
Structure from motion: limitations

• Very difficult to reliably estimate metric structure and motion unless:
  • large (x or y) rotation or
  • large field of view and depth variation
• Camera calibration important for Euclidean reconstructions
• Need good feature tracker
• Lens distortion
Issues in SFM

- Track lifetime
- Nonlinear lens distortion
- Prior knowledge and scene constraints
- Multiple motions
Track lifetime

every 50th frame of a 800-frame sequence
Track lifetime

lifetime of 3192 tracks from the previous sequence
Track lifetime

track length histogram
Nonlinear lens distortion
Nonlinear lens distortion

effect of lens distortion
Prior knowledge and scene constraints

add a constraint that several lines are parallel
Prior knowledge and scene constraints

add a constraint that it is a turntable sequence
Applications of Structure from Motion
Jurassic park
PhotoSynth

“What if your photo collection was an entry point into the world, like a wormhole that you could jump through and explore...”

http://labs.live.com/photosynth/