Structure from motion

- Cameras
- Epipolar geometry
- Structure from motion
Pinhole camera

f = focal length
o = center of the camera
Pinhole camera

$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$

$x' = f \frac{x}{z}$
$y' = f \frac{y}{z}$

$(x, y, z) \rightarrow (f \frac{x}{Z}, f \frac{y}{Z})$

Derived using similar triangles
From retina plane to images
From retina plane to images
Converting to pixels

1. Off set

\[(x, y, z) \rightarrow (f \frac{x}{z} + c_x, f \frac{y}{z} + c_y)\]
Converting to pixels

1. Off set
2. From metric to pixels

\[(x, y, z) \rightarrow (f \frac{k}{z} + c_x, f \frac{l}{z} + c_y)\]

Units: \(k, l:\) pixel/m \hspace{1cm} \text{Non-square pixels}
\(f:\) m \hspace{1cm} \alpha, \beta:\) pixel

\(C = [c_x, c_y]\)
Camera Matrix

\[(x, y, z) \rightarrow (\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y)\]

- Matrix form?
Homogeneous coordinates

Converting from homogeneous coordinates

\[
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix} \Rightarrow (x/w, y/w)
\]

For details see lecture on transformations in CS131A
Camera Matrix

\((x, y, z) \rightarrow (\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y)\)

\[ P' = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \]
Camera Skew

\[ P' = M P = K \begin{bmatrix} I & 0 \end{bmatrix} P \]
• The mapping so far is defined within the camera reference system
• What if an object is represented in the world reference system
World reference system

In 4D homogeneous coordinates:

\[ P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w \]

\[ P' = K[I \ 0] P = K[I \ 0] \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w \]

Internal parameters

External parameters
Properties of Projection

• Points project to points
• Lines project to lines
• Distant objects look smaller
Properties of Projection

- Angles are not preserved
- Parallel lines meet!

Vanishing point
Camera Calibration

- $P_1 \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots, p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

$$P' = K[I \ 0] P = K[R \ T] P_w$$
Camera Calibration

\[ \begin{bmatrix} P' \end{bmatrix} = \begin{bmatrix} K & 0 \end{bmatrix} \begin{bmatrix} P \end{bmatrix} = \begin{bmatrix} K & R & T \end{bmatrix} \begin{bmatrix} P_w \end{bmatrix} \]

- \( P_1 \ldots P_n \) with known positions in \([O_w,i_w,j_w,k_w]\)
- \( p_1, \ldots p_n \) known positions in the image

**Goal:** compute intrinsic and extrinsic parameters
Structure from motion

- Cameras
- Epipolar geometry
- Structure from motion
Can we recover the structure from a single view?

Why is it so difficult?

Intrinsic ambiguity of the mapping from 3D to image (2D)
Can we recover the structure from a single view?

Intrinsic ambiguity of the mapping from 3D to image (2D)

Courtesy slide S. Lazebnik
Two eyes help!

This is called \textbf{triangulation}
Structure from motion problem

Given \( m \) images of \( n \) fixed 3D points

\[
\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n
\]
From the $m \times n$ correspondences $x_{ij}$, can we estimate:

- $m$ projection matrices $M_i$
- $n$ 3D points $X_j$
Study relationship between $X$, $x_1$ and $x_2$

Epipolar geometry!
Epipolar geometry

- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles $e_1, e_2$
  = intersections of baseline with image planes
  = projections of the other camera center
Epipolar geometry

- Epipolar Plane
- Baseline
- Epipolar Lines

- Epipoles $e_1, e_2$
  - intersections of baseline with image planes
  - projections of the other camera center

For details see CS131A Lecture 9
Example: Converging image planes
Example: Parallel image planes

- Baseline intersects the image plane at infinity
- Epipoles are at infinity
- Epipolar lines are parallel to x axis
Example: Parallel Image Planes

e at infinity

\[
\begin{array}{c}
\text{e at infinity} \\
\end{array}
\]
Why are epipolar constraints useful?

- Two views of the same object
- Suppose I know the camera positions and camera matrices
- Given a point on left image, how can I find the corresponding point on right image?
Why are epipolar constraints useful?
Why are epipolar constraints useful?

- Two views of the same object
- Suppose I know the camera positions and camera matrices
- Given a point on left image, how can I find the corresponding point on right image?
Assume camera matrices are known

\[ p^T \cdot E \cdot p' = 0 \]

\[ E = [T_x] \cdot R \]

E = essential matrix

(Longuet-Higgins, 1981)
Cross product as matrix multiplication

\[ \mathbf{a} \times \mathbf{b} = \begin{bmatrix}
0 & -a_z & a_y \\
a_z & 0 & -a_x \\
-a_y & a_x & 0
\end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a} \times] \mathbf{b} \]
Epipolar Constraint

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

\[ F = K^{-T} \cdot [T_x] \cdot R \cdot K'^{-1} \]

\( F = \) Fundamental Matrix

(Faugeras and Luong, 1992)
Epipolar Constraint

\[ p_1^T \cdot F \cdot p_2 = 0 \]

- \( F p_2 \) is the epipolar line associated with \( p_2 \) \((l_1 = F p_2)\)
- \( F^T p_1 \) is the epipolar line associated with \( x_1 \) \((l_2 = F^T p_1)\)
- \( F e_2 = 0 \) and \( F^T e_1 = 0 \)
- \( F \) is a 3x3 matrix; 7 DOF
- \( F \) is singular (rank two)
Why F is useful?

- Suppose F is known
- No additional information about the scene and camera is given
- Given a point on left image, how can I find the corresponding point on right image?

\[ l' = F^T x \]
Why F is useful?

• F captures information about the epipolar geometry of 2 views + camera parameters

• **MORE IMPORTANTLY:** F gives constraints on how the scene changes under view point transformation (without reconstructing the scene!)

• Powerful tool in:
  • 3D reconstruction
  • Multi-view objectScene matching
Estimating F

The Eight-Point Algorithm

(Longuet-Higgins, 1981)
(Hartley, 1995)

\[
P \rightarrow p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad P \rightarrow p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad p^T F p' = 0
\]
Estimating F

OPENCV: findFundamentalMat
Structure from motion

- Cameras
- Epipolar geometry
- Structure from motion
From the $m \times n$ correspondences $x_{ij}$, can we estimate:

- $m$ projection matrices $M_i$
- $n$ 3D points $X_j$

**Structure from motion problem**
Similarity Ambiguity

• The scene is determined by the images only up a similarity transformation (rotation, translation and scaling)

• This is called **metric reconstruction**

[Image: Diagram showing similarity transformation between two images with a cube in the middle]

• The ambiguity exists even for (intrinsically) calibrated cameras
• For calibrated cameras, the similarity ambiguity is the only ambiguity

[Longuet-Higgins '81]
Similarity Ambiguity

- It is impossible based on the images alone to estimate the absolute scale of the scene (i.e. house height)

http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/hut/hutme.wrl
Structure from Motion Ambiguities

In the general case (nothing is known) the ambiguity is expressed by an arbitrary affine or projective transformation.

\[ x_j = M_i X_j \]

\[ M_i = K_i [R_i \ T_i] \]

\[ H X_i \]

\[ M_i H^{-1} \]

\[ x_j = M_i X_j = (M_i H^{-1})(H X_j) \]
Projective Ambiguity

Metric reconstruction (upgrade)

• The problem of recovering the metric reconstruction from the perspective one is called **self-calibration**

• Stratified reconstruction:
  • from perspective to affine
  • from affine to metric
Mobile SFM

- Intrinsic camera parameters are known or can be calibrated.
- For calibrated cameras, the similarity ambiguity is the only ambiguity [Longuet-Higgins '81]
- No need for stratified solution or auto-calibration

Metric reconstruction can be determined if a calibration pattern is used or the absolute size of an known object is given.
Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment
Algebraic approach (2-view case)

Apply a projective transformation $H$ such that:

$$M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix}$$  \hspace{2cm}  $$M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix}$$

Canonical perspective cameras

$$x_{ij} = M_i X_j$$
Algebraic approach (2-view case)

1. Compute the fundamental matrix $F$ from two views (eg. 8 point algorithm)

2. Compute $b$ and $A$ from $F$

   Compute $b$ as least sq. solution of $Fb = 0$, with $|b| = 1$ using SVD; $b$ is an epipole

   $A = -[b \times] F$

3. Use $b$ and $A$ to estimate projective cameras

   $M_1 = \begin{bmatrix} I & 0 \end{bmatrix} \quad M_2 = \begin{bmatrix} -[b \times] F & b \end{bmatrix}$

4. Use these cameras to triangulate and estimate points in 3D

For details, see CS231A, lecture 7
Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment
Affine structure from motion
(simpler problem)

From the $m \times n$ correspondences $x_{ij}$, estimate:
- $m$ projection matrices $M_i$ (affine cameras)
- $n$ 3D points $X_j$
Affine cameras

Camera matrix $M$ for the affine case

$$
\mathbf{x} = \begin{pmatrix} u \\ v \end{pmatrix} = M \begin{bmatrix} X \\ 1 \end{bmatrix} = A\mathbf{X} + \mathbf{b}; \quad M = \begin{bmatrix} A & \mathbf{b} \end{bmatrix}
$$
Centering the data

Normalize points w.r.t. centroids of measurements from each image

\[ \mathbf{x}_{ij} = \mathbf{A} \mathbf{X}_j + \mathbf{b} \rightarrow \hat{\mathbf{x}}_{ij} = \mathbf{A}_i \mathbf{X}_j \]
A factorization method - factorization

Let’s create a $2m \times n$ data (measurement) matrix:

$$D = \begin{bmatrix}
\hat{X}_{11} & \hat{X}_{12} & \cdots & \hat{X}_{1n} \\
\hat{X}_{21} & \hat{X}_{22} & \cdots & \hat{X}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{X}_{m1} & \hat{X}_{m2} & \cdots & \hat{X}_{mn}
\end{bmatrix}$$

cameras \ (2m \ )

points \ (n \ )
A factorization method - factorization

Let’s create a $2m \times n$ data (measurement) matrix:

$$
D = \begin{bmatrix}
\hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1n} \\
\hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{x}_{m1} & \hat{x}_{m2} & \cdots & \hat{x}_{mn}
\end{bmatrix}
= \begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_m
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n
\end{bmatrix}
$$

(2m x n)
cameras (2m x 3)

The measurement matrix $D = M S$ has rank 3
(it’s a product of a 2mx3 matrix and 3xn matrix)

The measurement matrix $D = M S$ has rank 3
(no specific camera number)

(2m x 3)
points (3 x n)
Factorizing the Measurement Matrix

\[ \text{Measurements} = \text{Motion} \times \text{Structure} \]

\[ D = MS \]
Factorizing the Measurement Matrix

- Singular value decomposition of $D$:

\[ D = U \times W \times V^T \]
Since rank (D)=3, there are only 3 non-zero singular values.
Factorizing the Measurement Matrix

\[ D = U_3 \times W_3 \times V_{3^T} \]

M = Motion (cameras)

S = structure
Factorizing the Measurement Matrix

What is the issue here? D has rank>3 because of:

- measurement noise
- affine approximation

Theorem: When \( D \) has a rank greater than \( p \), \( U_p W_p V_p^T \) is the best possible rank- \( p \) approximation of \( A \) in the sense of the Frobenius norm.

\[
D = U_3 W_3 V_3^T
\]

\[
\begin{align*}
A_0 &= U_3 \\
P_0 &= W_3 V_3^T
\end{align*}
\]

\[
\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}} \sigma_i^2}
\]
Affine Ambiguity

- The decomposition is not unique. We get the same $D$ by using any $3\times3$ matrix $C$ and applying the transformations:
  
  $$M \to MC$$
  $$S \to C^{-1}S$$

- Additional constraints must be enforced to resolve this ambiguity
Reconstruction results

Structure-from-Motion Algorithms

• Algebraic approach (by fundamental matrix)
• Factorization method (by SVD)
• Bundle adjustment
Bundle adjustment

Non-linear method for refining structure and motion

Minimizing re-projection error

\[
E(M, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, M_i X_j)^2
\]
Bundle adjustment

Non-linear method for refining structure and motion
Minimizing re-projection error

\[ E(M, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, M_i X_j)^2 \]

- **Advantages**
  - Handle large number of views
  - Handle missing data
  - Can leverage standard optimization packaged such as Levenberg-Marquardt

- **Limitations**
  - Large minimization problem (parameters grow with number of views)
  - Requires good initial condition

Used as the final step of SFM
Results and applications

Lucas & Kanade, 81
Chen & Medioni, 92
Debevec et al., 96
Levoy & Hanrahan, 96
Fitzgibbon & Zisserman, 98
Triggs et al., 99
Pollefeys et al., 99
Kutulakos & Seitz, 99

Levoy et al., 00
Hartley & Zisserman, 00
Dellaert et al., 00
Rusinkiewic et al., 02
Nistér, 04
Brown & Lowe, 04
Schindler et al, 04
Lourakis & Argyros, 04
Colombo et al. 05

Golparvar-Fard, et al. JAEI 10
Pandey et al. IFAC 2010
Pandey et al. ICRA 2011
Microsoft's PhotoSynth
Snavely et al., 06-08
Schindler et al., 08
Agarwal et al., 09
Frahm et al., 10

Courtesy of Oxford Visual Geometry Group
Next lecture:

Example of a SFM pipeline for mobile devices: **the VSLAM pipeline**