Recap: A General Point

- Equations of the form
  \[ Ax = 0 \]
- How do we solve them? (always!)
  - Apply SVD
    \[ A = U \Sigma V^T \]
    Singular values Singular vectors
    - Singular values of \( A \) = square roots of the eigenvalues of \( A^T A \).
    - The solution of \( Ax = 0 \) is the nullspace vector of \( A \).
    - This corresponds to the smallest singular vector of \( A \).

Recap: Camera Parameters

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - Skew (non-rectangular pixels)
  - Radial distortion
- Extrinsic parameters
  - Rotation \( R \)
  - Translation \( t \)
    (both relative to world coordinate system)
- Camera projection matrix
  \[ P = K[R|t] \]
  - General pinhole camera: 9 DoF
  - CCD Camera with square pixels: 10 DoF
  - General camera: 11 DoF

Recap: Calibrating a Camera

Goal

- Compute intrinsic and extrinsic parameters using observed camera data.

Main idea

- Place “calibration object” with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image: estimate \( P = P_{\text{int}} P_{\text{ext}} \)

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera calibration & Uncalibrated Reconstruction
  - Structure-from-Motion
- Motion and Tracking
Recap: Camera Calibration (DLT Algorithm)
\[
\begin{bmatrix}
0'
X_1'
X_2'
... \\
0'
X_1'
X_2'
\cdots
\end{bmatrix}
\begin{bmatrix}
0 \\
-\gamma_1X_1' \\
0 \\
-\gamma_2X_2' \\
\cdots \\
0 \\
-\gamma_1X_1' \\
0 \\
-\gamma_2X_2'
\end{bmatrix}
= \begin{bmatrix}
P_1 \\
P_2 \\
\cdots \\
P_2 \\
\cdots \\
P_2
\end{bmatrix}
\Rightarrow \text{Ap} = 0
\]
- P has 11 degrees of freedom.
- Two linearly independent equations per independent 2D/3D correspondence.
- Solution corresponds to homography estimation.
- Solution corresponds to smallest singular vector.
- 5 \frac{3}{2} correspondences needed for a minimal solution.

Recap: Triangulation - Linear Algebraic Approach
\[
\lambda_1x_1 = P_1x \\
x_1 \times P_1x = 0 \\
\lambda_2x_2 = P_2x \\
x_2 \times P_2x = 0
\]
- Two independent equations each in terms of three unknown entries of X.
- Stack equations and solve with SVD.
- This approach nicely generalizes to multiple cameras.

Recap: Epipolar Geometry - Calibrated Case
\[x \cdot (x' \times (Rx')) = 0 \quad \Rightarrow \quad x^TEx' = 0 \quad \text{with} \quad E = [t_1]R\]
Essential Matrix (Longuet-Higgins, 1981)

Recap: Epipolar Geometry - Uncalibrated Case
\[\hat{x}^TEx' = 0 \quad \Rightarrow \quad x^TFx' = 0 \quad \text{with} \quad F = K^{-1}EK^{-1}\]
Fundamental Matrix (Faugeras and Luong, 1992)

Recap: The Eight-Point Algorithm
\[x = (u, v, 1)^T, \quad x' = (u', v', 1)^T\]
\[
\begin{bmatrix}
F_{11} & F_{12} & F_{13} \\
F_{21} & F_{22} & F_{23} \\
F_{31} & F_{32} & F_{33}
\end{bmatrix}
\begin{bmatrix}
u \\
u' \\
1
\end{bmatrix}
= 0
\]
This minimizes: \[\sum_{i=0}^{N}(x_i^TFx_i')^2\]
Recap: Normalized Eight-Point Algorithm

1. 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.
2. Use the eight-point algorithm to compute $F$ from the normalized points.
3. Enforce the rank-2 constraint using SVD.
   \[ F = U D V^T = U \begin{bmatrix} d_{11} & \cdots & \cdots & d_{16} \\ \vdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ d_{16} & \cdots & \cdots & d_{11} \end{bmatrix} V^T \]
   Set $d_{33}$ to zero and reconstruct $F$
4. Transform fundamental matrix back to original units: if $T$ and $T'$ are the normalizing transformations in the two images, then the fundamental matrix in original coordinates is $T^T F T'$.

Active Stereo with Structured Light

- Idea: Project "structured" light patterns onto the object
  - simplifies the correspondence problem
  - Allows us to use only one camera

Laser Scanning

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Laser Scanned Models

The Digital Michelangelo Project

http://graphics.stanford.edu/projects/mich/
Laser Scanned Models

The Digital Michelangelo Project, Levoy et al.

Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #1: assume surface continuity

Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #2: colored stripes (or dots)

Active Stereo with Color Structured Light


Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #3: time-coded stripes
Time-Coded Light Patterns

- Assign each stripe a unique illumination code over time [Posdamer 82]

Better codes...

- Gray code
  - Neighbors only differ one bit

Poor Man’s Scanner

Slightly More Elaborate (But Still Cheap)

- Built-in IR projector
- IR camera for depth
- Regular camera for color

Under Everybody’s Christmas Tree...

Topics of This Lecture

- Structure from Motion (SfM)
  - Motivation
  - Ambiguity
- Affine SfM
  - Affine cameras
  - Affine factorization
  - Euclidean upgrade
  - Dealing with missing data
- Projective SfM
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations
- Applications
Structure from Motion

- Given: $m$ images of $n$ fixed 3D points
  \[ x_i = P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]
- Problem: estimate $m$ projection matrices $P_i$ and $n$ 3D points $X_j$ from the $mn$ correspondences $x_{ij}$

What Can We Use This For?

- E.g. movie special effects

Structure from Motion Ambiguity

- If we scale the entire scene by some factor $k$ and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points in the image remain exactly the same:
  \[ x = PX = \left( \frac{1}{k} P \right) (kX) \]
  \[ \Rightarrow \text{It is impossible to recover the absolute scale of the scene!} \]

Reconstruction Ambiguity: Similarity

- \[ x = PX = \left( PQ - X \right) QX \]

Reconstruction Ambiguity: Affine

- \[ x = PX = \left( PQ_A - X \right) Q_A X \]
Reconstruction Ambiguity: Projective

\[ x = P X = (PQ_P)^t Q_R X \]

Topics of This Lecture

- Structure from Motion (SfM)
  - Initialization
  - Ambiguity
- Affine SfM
  - Affine cameras
  - Affine factorization
  - Affine upgrade
  - Dealing with missing data
- Projective SfM
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations
- Applications

Hierarchy of 3D Transformations

- Projective 15 dof
- Affine 12 dof
- Similarity 7 dof
- Euclidean 6 dof

- With no constraints on the camera calibration matrix or on the scene, we get a projective reconstruction.
- Need additional information to upgrade the reconstruction to affine, similarity, or Euclidean.

From Projective to Affine

From Affine to Similarity

Slide credit: Svetlana Lazebnik
Structure from Motion

- Let’s start with affine cameras (the math is easier)

Orthographic Projection

- Special case of perspective projection
  - Distance from center of projection to image plane is infinite

Affine Cameras

- A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:

  \[
  P = \begin{bmatrix} 3 \times 3 & \text{affine} \\ 4 \times 4 & \text{affine} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & b_1 \\ a_2 & a_3 & a_4 & b_2 \\ a_3 & a_4 & a_5 & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
  \]

  \[
  \text{Affine projection is a linear mapping + translation in inhomogeneous coordinates}
  \]

  \[
  x = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad \begin{bmatrix} a_1 & a_2 & a_3 & b_1 \\ a_2 & a_3 & a_4 & b_2 \\ a_3 & a_4 & a_5 & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = AX + b
  \]

  \[
  \text{Projection of world origin}\]

Affine Structure from Motion

- Given: m images of n fixed 3D points:
  - \(x_i = A_i X_j + b_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n\)
  - Problem: use the mn correspondences \(x_{ij}\) to estimate m projection matrices \(A_i\) and translation vectors \(b_i\), and n points \(X_j\)

- The reconstruction is defined up to an arbitrary affine transformation \(Q\) (12 degrees of freedom):

  \[
  \begin{bmatrix} A \\ b \end{bmatrix} \rightarrow \begin{bmatrix} A \\ b \end{bmatrix} Q^{-1}, \quad \begin{bmatrix} X \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} Q \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix}
  \]

  - We have 2mn knowns and 8m + 3n unknowns (minus 12 dof for affine ambiguity).

- Thus, we must have 2mn > 8m + 3n - 12.

- For two views, we need four point correspondences.
Affine Structure from Motion

- 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)


Slide credit: Svetlana Lazebnik

Factorizing the Measurement Matrix

- Singular value decomposition of $D$:

\[
D = U \Sigma V^T
\]

$n \times n$ $n \times n$ $n \times n$

2m

$n$

Shape

Motion

Measurements

$n$

3

D = MS

Slide credit: Martial Hebert

Factorizing the Measurement Matrix

- Obtaining a factorization from SVD:

\[
D = U_k \times \begin{bmatrix} \mathbf{W} & \mathbf{V}_k^T \end{bmatrix}
\]

$n \times n$

$n \times n$

$3 \times n$

$3 \times 3$

$3 \times 3$

$3 \times 3$

To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3.

Slide credit: Martial Hebert
Factorizing the Measurement Matrix

- Obtaining a factorization from SVD:

\[ D = U \times W \times V^T \]

Possible decomposition:

\[ D = M \times S \]

This decomposition minimizes \( ID-MSI^2 \)

Affine Ambiguity

- The decomposition is not unique. We get the same \( D \) by using any \( 3 \times 3 \) matrix \( C \) and applying the transformations \( M \rightarrow MC, S \rightarrow C^{-1}S \).
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example). We need a Euclidean upgrade.

Estimating the Euclidean Upgrade

- Orthographic assumption: image axes are perpendicular and scale is 1.

\[ a_1 \cdot a_2 = 0 \]

\[ |a_1|^2 = |a_2|^2 = 1 \]

This can be converted into a system of \( 3m \) equations:

\[ \begin{align*}
  \alpha_1 \cdot \alpha_2 &= 0 \\
  |\alpha_1|^2 &= 1 \\
  |\alpha_2|^2 &= 1 \\
\end{align*} \]

for the transformation matrix \( C \) ⇒ goal: estimate \( C \)

Algorithm Summary

- Given: \( m \) images and \( n \) features \( x_{ij} \)
- For each image \( i \), center the feature coordinates.
- Construct a \( 2m \times n \) measurement matrix \( D \):
  - Column \( j \) contains the projection of point \( j \) in all views
  - Row \( i \) contains one coordinate of the projections of all the \( n \) points in image \( i \)
- Factorize \( D \):
  - Compute SVD: \( D = U W V^T \)
  - Create \( U_i \) by taking the first 3 columns of \( U \)
  - Create \( V_i \) by taking the first 3 columns of \( V \)
  - Create \( W_i \) by taking the upper left 3 \times 3 block of \( W \)
  - Create the motion and shape matrices:
    - \( M = U_i W_i \) and \( S = W_i V_i^T \) (or \( M = U_i \) and \( S = W_i V_1^T \))
  - Eliminate affine ambiguity
Dealing with Missing Data

- So far, we have assumed that all points are visible in all views.
- In reality, the measurement matrix typically looks something like this:

\[
\begin{array}{cccccccc}
\text{Cameras} & \text{Points} \\
6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\
\end{array}
\]

- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results.
  - Incremental bilinear refinement

1. Perform factorization on a dense sub-block
2. Solve for a new 3D point visible by at least two known cameras (linear least squares)
3. Solve for a new camera that sees at least three known 3D points (linear least squares)


Dealing with Missing Data

- Possible solution: decompose matrix into dense sub-blocks, factorize each sub-block, and fuse the results.
  - Finding dense maximal sub-blocks of the matrix is NP-complete (equivalent to finding maximal cliques in a graph)
  - Incremental bilinear refinement

1. Perform factorization on a dense sub-block
2. Solve for a new 3D point visible by at least two known cameras (linear least squares)
3. Solve for a new camera that sees at least three known 3D points (linear least squares)


Comments: Affine SfM

- Affine SfM was historically developed first.
- It is valid under the assumption of affine cameras.
  - Which does not hold for real physical cameras...
  - But which is still tolerable if the scene points are far away from the camera.

- For good results with real cameras, we typically need projective SfM.
  - Harder problem, more ambiguity
  - Math is a bit more involved...
  - (Here, only basic ideas. If you want to implement it, please look at the H&Z book for details).

Topics of This Lecture

- Structure from Motion (SfM)
  - Motivation
  - Ambiguity
  - Affine SfM
  - Affine cameras
  - Affine factorization
  - Euclidean upgrade
  - Dealing with missing data

- Projective SfM
  - Two-camera case
  - Projective factorization
  - Bundle adjustment
  - Practical considerations

- Applications
Projective Structure from Motion

- Given: \(m\) images of \(n\) fixed 3D points
- \(x_i = P_i x_j\), \(i = 1, \ldots, m\), \(j = 1, \ldots, n\)
- Problem: estimate \(m\) projection matrices \(P_i\) and \(n\) 3D points \(X_j\) from the \(mn\) correspondences \(x_{ij}\)

Projective Structure from Motion

- Given: \(m\) images of \(n\) fixed 3D points
- Problem: estimate \(m\) projection matrices \(P_i\) and \(n\) 3D points \(X_j\) from the \(mn\) correspondences \(x_{ij}\)
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation \(Q\):
  \[X \rightarrow QX, \ P \rightarrow PQ^{-1}\]
- We can solve for structure and motion when \(2mn > 11m + 3n - 15\)
- For two cameras, at least 7 points are needed.

Projective SfM: Two-Camera Case

- Assume fundamental matrix \(F\) between the two views
  - First camera matrix: \([I|0]Q\)
  - Second camera matrix: \([Alb|Q\)
- Let \(X = QX\), then \(z = [I|0]X\), \(z' = [Alb]X\)
- And
  \[z' = A[I|0]X + b = A X + b\]
  \[(z' \times b) \cdot x' = (A X + b) \cdot x'\]
  \[0 = (A X + b) \cdot x'\]
- So we have \(x^T[b]A = 0\)
- \(F = [b]A\): epipole \((F^Tb = 0)\), \(A = [b]F\)

Sequential Structure from Motion

- Initialize motion from two images using fundamental matrix
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - calibration
Self-Calibration

Determining intrinsic camera parameters directly from uncalibrated images.

For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images.

Compute initial projective reconstruction and find 3D projective transformation matrix \( Q \) such that all camera matrices are in the form \( P_i = K[Q_i, I] \).

Can use constraints on the form of the calibration matrix: square pixels, zero skew, fixed focal length, etc.
Practical Considerations (1)

1. Role of the baseline
   - Small baseline: large depth error
   - Large baseline: difficult search problem
   • Solution
     - Track features between frames until baseline is sufficient.

Practical Considerations (2)

2. There will still be many outliers
   - Incorrect feature matches
   - Moving objects
   ⇒ Apply RANSAC to get robust estimates based on the inlier points.

3. Estimation quality depends on the point configuration
   - Points that are close together in the image produce less stable solutions.
   ⇒ Subdivide image into a grid and try to extract about the same number of features per grid cell.

General Guidelines

• Use calibrated cameras wherever possible.
  - It makes life so much easier, especially for SfM.
• SfM with 2 cameras is far more robust than with a single camera.
  - Triangulate feature points in 3D using stereo.
  - Perform 2D-3D matching to recover the motion.
  - More robust to loss of scale (main problem of 1-camera SfM).
• Any constraint on the setup can be useful
  - E.g. square pixels, zero skew, fixed focal length in each camera
  - E.g. fixed baseline in stereo SfM setup
  - E.g. constrained camera motion on a ground plane
  - Making best use of those constraints may require adapting the algorithms (some known results are described in H&Z).

Structure-from-Motion: Limitations

• Very difficult to reliably estimate metric SfM unless
  - Large (x or y) motion
  - Large field-of-view and depth variation
• Camera calibration important for Euclidean reconstruction
• Need good feature tracker

Topics of This Lecture

• Structure from Motion (SfM)
  - Ambiguity
• Affine SfM
  - Affine cameras
  - Affine factorization
• Euclidean SfM
• Projective SfM
  - Two-camera case
  - Projective factorization
• Applications

Commercial Software Packages

• boujou
  (http://www.2d3.com/)
• PFTrack
  (http://www.thepixelfarm.co.uk/)
• MatchMover
  (http://www.realviz.com/)
• SynthEyes
  (http://www.ssontech.com/)
• Icarus
  (http://aig.cs.man.ac.uk/research/reveal/icarus/)
• Voodoo Camera Tracker
  (http://www.digilab.uni-hannover.de/)
boujou demo

(We have a license available, so if you want to try it for interesting projects, contact us.)

Applications: Matchmoving

- Putting virtual objects into real-world videos

<table>
<thead>
<tr>
<th>Original sequence</th>
<th>Tracked features</th>
</tr>
</thead>
<tbody>
<tr>
<td>SfM results</td>
<td>Final video</td>
</tr>
</tbody>
</table>

Applications: Large-Scale SfM from Flickr


References and Further Reading

- A (relatively short) treatment of affine and projective SfM and the basic ideas and algorithms can be found in Chapters 12 and 13 of


- More detailed information (if you really want to implement this) and better explanations can be found in Chapters 10, 18 (factorization) and 19 (self-calibration) of

  R. Hartley, A. Zisserman Multiple View Geometry in Computer Vision 2nd Ed., Cambridge Univ. Press, 2004