Binocular Stereo (cont’d.),
Structure from Motion

Aykut Erdem
Dept. of Computer Engineering
Hacettepe University

Basic stereo matching algorithm

- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match
  - Triangulate the matches to get depth information

- Simplest case: epipolar lines are scanlines
  - When does this happen?

Simplest Case: Parallel images

- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

Simplest Case: Parallel images

- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then, epipolar lines fall along the horizontal scan lines of the images
Essential matrix for parallel images

Epipolar constraint:
\[ x^T E x' = 0, \quad E = [t_x] R \]
\[ R = I \quad t = (T, 0, 0) \]
\[ E = [t_x] R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \]

\[ [a_x] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \]

Stereo image rectification

- reproject image planes onto a common plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies (3x3 transform), one for each input image reprojection

\[ (u' v 1) \begin{bmatrix} 0 & 0 & -T \\ 0 & T & 0 \\ 1 \end{bmatrix} \begin{bmatrix} a_x' \\ a_y' \end{bmatrix} = 0 \]
\[ (u' v 1) \begin{bmatrix} 0 & 0 & -T \\ 0 & T & 0 \\ 1 \end{bmatrix} = 0 \]
\[ Tv = Tv' \]

The y-coordinates of corresponding points are the same!
Rectification example

Why rectification is useful?

- Makes the correspondence problem easier
- Makes triangulation easy

Depth from disparity

\[
\text{disparity} = x - x' = \frac{B \cdot f}{z}
\]

Disparity is inversely proportional to depth!
Correspondence search

- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correlation Method

- Pick up a window around \( p(u,v) \)
- Build vector \( W \)
- Slide the window along \( v \) line in image 2 and compute \( w' \)
- Keep sliding until \( w w' \) is maximized.

Normalized Correlation; minimize:

\[
\frac{(w - \bar{w})(w' - \bar{w}')}{{\|w - \bar{w}\|}{\|w' - \bar{w}'\|}}
\]
Correspondence search

Effect of window size

- Smaller window
  + More detail
  • More noise

- Larger window
  + Smoother disparity maps
  • Less detail

Failures of correspondence search

- Textureless surfaces
- Occlusions, repetition
- Non-Lambertian surfaces, specularities

Stereo results

- Data from University of Tsukuba
- Similar results on other images without ground truth
Results with window search

Window-based matching (best window size)  
Ground truth

Better methods exist...

Graph cuts  
Ground truth

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

For the latest and greatest: http://www.middlebury.edu/stereo/

How can we improve window-based matching?

- The similarity constraint is **local** (each reference window is matched independently)
- Need to enforce **non-local** correspondence constraints

Non-local constraints

- **Uniqueness**
  - For any point in one image, there should be at most one matching point in the other image
Non-local constraints

- **Uniqueness**
  - For any point in one image, there should be at most one matching point in the other image

- **Ordering**
  - Corresponding points should be in the same order in both views

- **Smoothness**
  - We expect disparity values to change slowly (for the most part)

Scanline stereo

- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized
"Shortest paths" for scan-line stereo

Can be implemented with dynamic programming
Ohta & Kanade ’85, Cox et al. ’96

Stereo as energy minimization

- What defines a good stereo correspondence?
  1. Match quality
     - Want each pixel to find a good match in the other image
  2. Smoothness
     - If two pixels are adjacent, they should (usually) move about the same amount

Stereo matching as energy minimization

\[ E(D) = \sum_i \left( W_1(i) - W_2(i + D(i)) \right)^2 + \lambda \sum_{i,j} \rho(D(i) - D(j)) \]

- Energy functions of this form can be minimized using graph cuts

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001
Active stereo with structured light

- Project “structured” light patterns onto the object
  - Simplifies the correspondence problem
  - Allows us to use only one camera


Kinect: Structured infrared light


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

Digital Michelangelo Project
Levoy et al.
http://graphics.stanford.edu/projects/mich/

Laser scanned models

The Digital Michelangelo Project, Levoy et al.
Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Beyond two-view stereo

Slide credit: S. Lazebnik

Structure from motion

Multiple views of a static scene from different cameras

One camera, a moving object

Slide credit: B. Freeman and A. Torralba

Volumetric stereo

Scene Volume $V$

Input Images (Calibrated)

Goal: Determine occupancy, “color” of points in $V$

Slide credit: N. Snavely
Discrete formulation: Voxel Coloring

**Goal:** Assign RGBA values to voxels in V \( photo-consistent \) with images

Space Carving

**Space Carving Algorithm**
- Initialize to a volume \( V \) containing the true scene
- Choose a voxel on the outside of the volume
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence

**Complexity and computability**

\( \text{Input Images} \quad (\text{Calibrated}) \)

\( C^N \) scenes

True Scene

Photo-Consistent Scenes

All Scenes \( (C^N) \)

Space Carving Results: African Violet

Input Image (1 of 45)  Reconstruction

Reconstruction  Reconstruction

Source: S. Seitz

Structure from motion

• Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates

\[ R_1, t_1 \quad R_2, t_2 \quad R_3, t_3 \]

Slide credit: N. Snavely

Structure from motion

• Given: m images of n fixed 3D points

\[ x_{ij} = P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]

• Problem: estimate m projection matrices \( P_i \) and n 3D points \( X_j \) from the mn correspondences \( x_{ij} \)

Slide credit: S. Lazebnik

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:

\[ x = PX = \left( \frac{1}{k} P \right) (kX) \]

It is impossible to recover the absolute scale of the scene!

Slide credit: S. Lazebnik

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

• More generally: if we transform the scene using a transformation \( Q \) and apply the inverse transformation to the camera matrices, then the images do not change

\[ x = PX = (PQ^{-1})QX \]

Slide credit: S. Lazebnik
Types of ambiguity

- **Projective ambiguity**
  - 15dof
  - Preserves intersection and tangency

- **Affine ambiguity**
  - 12dof
  - Preserves parallelism, volume ratios

- **Similarity ambiguity**
  - 7dof
  - Preserves angles, ratios of length

- **Euclidean ambiguity**
  - 6dof
  - Preserves angles, lengths

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.

---

**Projective ambiguity**

\[ Q_p = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix} \]

\[ x = PX = (PQ_p^{-1})Q_p X \]

---

**Affine ambiguity**

\[ Q_A = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} \]

\[ x = PX = (PQ_A^{-1})Q_A X \]
**Affine ambiguity**

Slide credit: S. Lazebnik

---

**Similarity ambiguity**

\[ Q_s = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} \]

\[ x = PX = \left( P Q_s^{-1} \right) Q_s X \]

---

**Structure from motion**

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

Slide credit: S. Lazebnik
Recall: Orthographic Projection

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

- Projection matrix:
  \[
  \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y \\
  z \\
  1 \\
  \end{bmatrix}
  =
  \begin{bmatrix}
  x \\
  y \\
  0 \\
  1 \\
  \end{bmatrix}
  \Rightarrow (x, y)
  \]

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:
  \[
  P = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  \end{bmatrix}
  \begin{bmatrix}
  a_{11} & a_{12} & a_{13} & b_1 \\
  a_{21} & a_{22} & a_{23} & b_2 \\
  0 & 0 & 0 & 1 \\
  \end{bmatrix}
  =
  \begin{bmatrix}
  A & b \\
  0 & 1 \\
  \end{bmatrix}
  \]

- Affine projection is a linear mapping + translation in inhomogeneous coordinates
  \[
  x = \begin{bmatrix}
  X \\
  Y \\
  Z \\
  \end{bmatrix}
  =
  \begin{bmatrix}
  a_{11} & a_{12} & a_{13} & b_1 \\
  a_{21} & a_{22} & a_{23} & b_2 \\
  0 & 0 & 0 & 1 \\
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y \\
  \end{bmatrix}
  = AX + b
  \]

Affine structure from motion

- Given: m images of n fixed 3D points:
  - \( x_i = A_iX_j + b_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \)

- Problem: Use the mn correspondences \( x_i \) 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):
  \[
  \begin{bmatrix}
  A & b \\
  0 & 1 \\
  \end{bmatrix}
  \begin{bmatrix}
  A & b \\
  0 & 1 \\
  \end{bmatrix}^{-1},
  \begin{bmatrix}
  X \\
  \end{bmatrix}
  \rightarrow Q
  \begin{bmatrix}
  X \\
  \end{bmatrix}
  \]

- 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

• Centering: subtract the centroid of the image points
  \[
  \hat{x}_j = x_j - \frac{1}{n} \sum_{i=1}^{n} x_{i_j} = A_i X_j + b_j - \frac{1}{n} \sum_{i=1}^{n} (A_i X_i + b_i)
  \]
  \[
  = A_i \left( X_j - \frac{1}{n} \sum_{i=1}^{n} X_i \right) = A_i \hat{X}_j
  \]

• For simplicity, assume that the origin of the world coordinate system is at the centroid of the 3D points
• After centering, each normalized point \(x_{i_j}\) is related to the 3D point \(X_i\) by
  \[
  \hat{x}_{i_j} = A_i X_j
  \]


Slide credit: S. Lazebnik
Factorizing the measurement matrix

- Singular value decomposition of $D$:

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

This decomposition minimizes $\| D - MS \|_2$

- Obtaining a factorization from SVD:

\[ D = U_3 W_3 V_3^T \]

Possible decomposition:

\[ M = U_3 W_3^{1/2} \quad S = W_3^{1/2} V_3^T \]

This decomposition minimizes $\| D - MS \|_2^2$
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).

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 \(U_3\) by taking the first 3 columns of \(U\)
  - Create \(V_3\) by taking the first 3 columns of \(V\)
  - Create \(W_3\) by taking the upper left \(3 \times 3\) block of \(W\)
- Create the motion and shape matrices:
  - \(M = U_3 W_3^{1/2}\) and \(S = W_3^{1/2} V_3^T\) (or \(M = U_3\) and \(S = W_3 V_3^T\))
- Eliminate affine ambiguity

Eliminating the affine ambiguity

- Orthographic: image axes are perpendicular and scale is 1
- This translates into \(3m\) equations in \(L = CC^T\): \(A_i L A_i^T = I\), \(i = 1, ..., m\)
  - Solve for \(L\)
  - Recover \(C\) from \(L\) by Cholesky decomposition: \(L = CC^T\)
  - Update \(M\) and \(S\): \(M = MC, S = C^{-1}S\)

Reconstruction results

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:

Slide credit: S. Lazebnik

Projective structure from motion

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

Slide credit: S. Lazebnik

Projective structure from motion

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

- Compute fundamental matrix $F$ between the two views
- First camera matrix: $[I|0]$
- Second camera matrix: $[A|b]$
- Then $b$ is the epipole ($F^T b = 0$), $A = -[b_x]F$

---

**Sequential structure from motion**

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - calibration
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera - triangulation

---

**Refine structure and motion: bundle adjustment**
Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing reprojection error

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

Summary: 3D geometric vision

- Single-view geometry
  - The pinhole camera model
    - Variation: orthographic projection
  - The perspective projection matrix
  - Intrinsic parameters
  - Extrinsic parameters
  - Calibration

- Multiple-view geometry
  - Triangulation
  - The epipolar constraint
    - Essential matrix and fundamental matrix
  - Stereo
    - Binocular, multi-view
  - Structure from motion
    - Reconstruction ambiguity
    - Affine SFM
    - Projective SFM

Self-calibration

- Self-calibration (auto-calibration) is the process of 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 \( P_i = K [R_i | t_i] \)
- Can use constraints on the form of the calibration matrix: zero skew
- Can use vanishing points