Reconstruction = Shape from X

Task: Find 3D shape from 1+ images

• 3D Shape from stereo
• 3D Shape from shading
• 3D Photometric stereo
• 3D Shape from defocus
• 3D Shape from motion
• 3D Shape from structured light
• 3D Shape from time of flight

Depth with stereo: basic idea

Related: Shape from motion, structure from motion

Source: Seitz, Avidan, Luong
Depth with stereo: basic idea

Basic Principle: Intersecting rays

Requires:
• Relative camera pose
• Point correspondence
• Triangulation

Lecture:
• Simplified idealized case
• Finding (absolute / relative) camera pose
• Finding correspondences
• Triangulation

Source: Steve Seitz

Geometry for a simplified stereo

Assume:
parallel optical axes, calibrated cameras, correspondence,

• Fixation (Experiment)
• Baseline
Geometry for a simplified stereo

Similar triangles \((p_l, P, p_r)\) and \((O_l, P, O_r)\):

\[
\frac{T - (x_l - x_r)}{Z - f} = \frac{T}{Z}
\]

\[Z = f \frac{T}{x_l - x_r}\]

disparity

Depth from disparity

image \(I(x,y)\)

Disparity map \(D(x,y)\)

image \(I'(x',y')\)

\[(x',y') = (x + D(x,y), y)\]
General case

• The two cameras need not have parallel optical axes.

We need to know something about their pose
• Do two points correspond to intersecting rays?
  – Relative calibration
• Where is the intersection in camera coordinates?
  – Relative pose
• Where is the intersection in World coordinates?
  – Calibration

Calibrating a camera

• Compute intrinsic and extrinsic parameters using observed camera data

Main idea
• Place (3D) “calibration object” with known geometry in the scene
• Get correspondences
• Solve for mapping from scene to image: estimate \( M = K[R \ t] \)
The projection may be written as a matrix product using homogeneous coordinates:

$$\begin{bmatrix}
wx_{im} \\
wz_{im} \\
w
\end{bmatrix} = K[R \; t] M w \begin{bmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{bmatrix}$$

Let $M_i$ be row $i$ of matrix $M$.

Every point gives two constraints on $M$: 

$$0 = (M_i - x_{im}M_1) \cdot P_w$$

$$0 = (M_i - y_{im}M_3) \cdot P_w$$
Estimating the projection matrix

For sufficiently large number of point .... an over-constrained equation \( \mathbf{P}_m = 0 \)

\[
\begin{pmatrix}
X^{(i)}_x & Y^{(i)}_x & Z^{(i)}_x & 1 & 0 & 0 & 0 & -X^{(i)}_w X^{(i)}_x & -X^{(i)}_w Y^{(i)}_x & -X^{(i)}_w Z^{(i)}_x & -X^{(i)}_w \\
0 & 0 & 0 & 0 & X^{(i)}_w & Y^{(i)}_w & Z^{(i)}_w & 1 & -Y^{(i)}_m X^{(i)}_w & -Y^{(i)}_m Y^{(i)}_w & -Y^{(i)}_m Z^{(i)}_w & -Y^{(i)}_m
\end{pmatrix}
\begin{pmatrix}
m_{i1} \\
m_{i2} \\
m_{i3} \\
m_{i4} \\
m_{i5} \\
m_{i6} \\
m_{i7} \\
m_{i8} \\
m_{i9} \\
m_{i10} \\
m_{i11} \\
m_{i12} \\
m_{i13} \\
m_{i14} \\
m_{i15} \\
m_{i16} \\
m_{i17} \\
m_{i18} \\
m_{i19} \\
m_{i20} \\
m_{i21} \\
m_{i22} \\
m_{i23} \\
m_{i24} \\
m_{i25} \\
m_{i26} \\
m_{i27} \\
m_{i28} \\
m_{i29} \\
m_{i30} \\
m_{i31}
\end{pmatrix}
= \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

Solve for \( m_{ij} \)'s (the calibration information) [F&P Section 3.1]

Image points and 3D rays

Note: when \( M \) is known every point imposes two linear constraints on the 3D point \( \mathbf{P}_w \)

Therefore \( \mathbf{P}_w \) is on a line (ray)

\[
X_{im} = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}
\]

\[
y_{im} = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}
\]
Stereo calibration

Some options:
- Calibrate every camera separately, using 3D data.
  - Internal and external calibration.
  - Given a point image location, the ray is specified.

Already learned …
- Calibrate the cameras relative to each other
  - NOT relative to the world coordinates
  - Using only image data, not 3D data.
  - Sufficient for:
    - Helping correspondence
    - Limited reconstruction

The connection between two images of the same scene

• The match (right image) for p (left image) is non-unique.
• The match for p must be on the “epipolar” line l.
• The matches for p’ must lie on … epipolar line l.

Source: M. Pollefeys
Epipolar geometry

- **Baseline**: a line joining the camera centers
- **Epipolar plane**: plane containing baseline and world point
- **Epipolar line**: intersection of epipolar plane with the image plane
- **Epipole**: point of intersection of baseline with the image plane. Also intersection of all epipolar lines.

Adapted from M. Pollefeys, UNC

Example: converging cameras

As position of 3d point varies, epipolar lines "rotate" about the baseline

Figure from Hartley & Zisserman
Example: motion parallel with image plane

This was a geometric interpretation

For a given stereo camera pair, how do we express the epipolar constraints algebraically?
Stereo geometry, with ideal cameras

Suppose the camera-centered coordinate systems are related by known rotation $R$ and translation $T$:

$$X' = RX + T$$

Reminder: 3d rigid transformation

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$X' = RX + T$$
Reminder: Cross product

\[ \vec{a} \times \vec{b} = \vec{c} \quad \vec{a} \cdot \vec{c} = 0 \quad \vec{b} \cdot \vec{c} = 0 \]

Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

From geometry to algebra

\[ X' = RX + T \]
\[ T \times X' = T \times RX + T \times T = T \times RX \]
\[ X' \cdot (T \times X') = X' \cdot (T \times RX) = 0 \]
Matrix form of cross product

\[
\vec{a} \times \vec{b} = \begin{bmatrix}
0 & -a_z & a_y \\
al_z & 0 & -a_x \\
-a_y & a_x & 0
\end{bmatrix}
\begin{bmatrix}
b_x \\
b_y \\
b_z
\end{bmatrix} = \vec{a} \cdot \vec{c} = 0 \\
\vec{b} \cdot \vec{c} = 0
\]

Can be expressed as a matrix multiplication.

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

\[
\vec{a} \times \vec{b} = [a_x] \vec{b}
\]

Essential matrix

\[
X' \cdot (T \times RX) = 0
\]

\[
X' \cdot (T \times RX) = 0
\]

Let \( E = T \times R \)

\[
X'^TEX = 0
\]

This holds for \( X \) and \( X' \) which describe the same 3D point in two coordinate systems.

Reminder: A 3D coordinate vector \( X \) (\( X' \)) is proportional to its corresponding image point vector \( p \) (\( p' \)) (in homogeneous coordinates). Therefore

\[
p'^T Ep = 0
\]

\( E \) is called the essential matrix [Longuet-Higgins 1981]
Essential matrix and epipolar lines

\[ p'^T Ep = 0 \]

Epipolar constraint: if we observe point \( p \) in one image, then its position \( p' \) in second image
- must satisfy this equation.
- or equivalently, must lie on a specific line

\[ E^T p' \]
represents the epipolar line associated with point \( p' \)

\[ Ep \]
represents the epipolar line associated with point \( p \)

Essential matrix: properties

- A 3x3 matrix
- Relates image of corresponding points in both cameras, assuming intrinsic parameters are ideal
- Calculated by \( E = T . R \)
- \( E \) has 5 degrees of freedom
- \( E \) is a rank 2 matrix
  
  Proof: For every point \( \), the corresponding point as well as any point lying on the epipolar line, must satisfy the constraint. In particular the epipole satisfies the constraints
  
  \( (P')^T E e = 0 \)

This implies that \( e \) in an e.vector of \( E \) with 0 e.value.
- \( (E \) has two equal singular values)
Essential matrix example: parallel cameras

\[ R = I \]
\[ T = \begin{bmatrix} -d, 0, 0 \end{bmatrix}^T \]
\[ E = [T] R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \]

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

For parallel axes cameras, epipolar lines are parallel (to scan lines if …)

Stereo image rectification

For general camera pairs, it would also be convenient if epipolar lines are scanlines.

reproject image planes onto a common plane parallel to the baseline, using two homographies.

Stereo image rectification: example

Can we always use the Essential matrix?

In general

\[ \bar{p} = Kp \propto K \begin{bmatrix} R & T \end{bmatrix} X \]

- \( K \) known - we may infer \( p \) from \( \bar{p} \) and use \( E \)
- \( K \) unknown - we may get a weaker constraint
Uncalibrated case: fundamental matrix

\[ p^T E p = 0 \]

\[
(K'^{-1} \bar{p}')^T E (K^{-1} \bar{p}) = 0
\]

\[
\bar{p}'^T (K'^{-1} E K^{-1}) \bar{p} = 0
\]

\[
\bar{p}'^T F \bar{p} = 0
\]

Fundamental matrix

Fundamental matrix

- A 3x3 matrix
- Relates pixel coordinates in the two views
- More general (and useful) from than essential matrix: no need to know intrinsic parameters
- F has 7 degrees of freedom
- F has rank 2.
- The fundamental matrix may be estimated from correspondences in pixel coordinates.
- Knowing the fundamental matrix = “knowing the epipolar geometry”.
Estimating $F$ from correspondences

Each point correspondence generates one constraint on $F$:

$$ \mathbf{p}'^T F \mathbf{p} = 0 $$

The 8-point algorithm

1. Collect $n=8+$ of these constraints
2. Normalize (to reduce condition number) (crucial!)
3. Solve using SVD
4. Rearrange to 3x3 and find a rank 2 approximation
Several approaches to Stereo

Calibrated Stereo
Given two cameras (a stereo rig) and a 3D object with known 3D coordinates:

a. (Calibration) For each camera compute the direct linear transformation from 3D to 2D
b. Calculate E.

Given two new images of objects with unknown coordinates,

a. find correspondences.
b. Given a pair of matching image points, triangulate* to get the 3D point's

Uncalibrated stereo

Estimate world geometry without calibrated cameras
- Archival videos
- Photos from multiple unrelated users
- Dynamic camera system

Main idea:
- Estimate epipolar geometry (F) from a set of interest point correspondences
- Use it for finding correspondences
- Extract (some) projection matrices that satisfy the epipolar geometry.
- Use it to find a weaker, non-Euclidean, “projective” reconstruction (two reconstructions are equivalent if they differ by a projective transformation.)
Ambiguities in Reconstruction

• For calibrated cameras (intrinsic) and unknown R,T. The reconstruction is up to scale.

Ambiguities in Reconstruction - II

• For uncalibrated cameras the reconstruction is up to projective transformation:

\[ p_j = M_i P_j = (M_i Q^{-1})(Q P_j) \]
Stereo correspondence

Goal: find correspondence

Context
- Most points are seen in both images
- Most seen points are pairwise similar in most images
- The correspondence is "locally consistent"

Approach I: Sparse correspondence
- Use interest points - accurate location
- Invariance is less important
- Disadvantage: sparse reconstruction

Approach II: Dense correspondence
- Search a match for every point
  - Use epipolar geometry to make the search 1D
  - Similarity: SSD between patches,
    Norm. Correlation

Parallel camera example – epipolar lines are corresponding rastors
Correlation based window matching

Effect of window size

Smaller window implies that the disparity image is
- More detailed
- More noisy
- Less smooth
Correlation method – results

Stereo problems – I

- Normalized correlation does not work well
Stereo problems - II

- Repetitive patterns
- Patches are not unique

Stereo problems - III

- Fore shortening effect
- Occlusions
- It is desirable to have small B/z ratio!
- Patches are not similar
Helping Normalized correlation - I

- Uniqueness of match - every point should have at most one match

![Diagram showing uniqueness constraint]

Helping Normalized correlation - II

- Match location is
  - usually monotonic
  - Piecewise Smooth
  - Consistent between lines

![Diagram showing consistency and monotonicity]
Dynamic programming algorithm

- Use the monotonicity constraints to Match pixels on a full scan lines

Dynamic programming algorithm

- Cost depends on local similarity of matched points, on disparity smoothness, and on non-monotonicity and occlusion penalties.

- Can be viewed as finding minimal cost path from upper left to lower right
Stereo results

Window-based matching  Dynamic Prog.  Graph cuts (Boykov et al)

Triangulation

- Task: Given corresponding points from 2+ images (calibrated cameras), compute the 3D location $x$

- Approach 1 (intuitive): Find the rays, and then find a 3D point $X$ minimizing 3D distance to them

$$\arg\min_X \sum_j \|C_j + sV_j - X\|$$

- $X$ is the unknown 3D point
- $C_j$ is the optical center of camera $j$
- $V_j$ is the viewing ray for pixel $(u_j, v_j)$
- $s_j$ is unknown distance along $V_j$

- Advantage: geometrically intuitive
Triangulation

• **Approach 2 (Algebraic & easy):** Find all rays (lines) equations and solve simultaneously by least squares

\[
\begin{align*}
    u_i &= \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1} \\
    v_i &= \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + 1}
\end{align*}
\]

• **Approach 3 (Nonlinear (complex but most accurate)):**
  Find a perturbation of image points so that
  • All rays meet exactly at same 3D point (X)
  • The image perturbation distance is smallest

Stereo reconstruction pipeline

Steps
• Calibrate cameras
• Rectify images
• Compute disparity
• Estimate depth

What will cause errors?
• Camera calibration errors
• Far objects and low image resolution
• Correspondence problems
  • Occlusions
  • Violations of brightness constancy (specular reflections)
  • Large motions
  • Low-contrast image regions
Active stereo with structured light

Project “structured light” patterns onto the object
- simplifies the correspondence problem

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, Levoy et al.

Time of flight (TOF) cameras

One option: RF modulated light

(+): Simple to use & analyze
(+): Fast
(-): Suffer from daylight
(-): Suffer from multiple reflections
Structure from Multiview Motion

- Many cameras
- Many points

Task: Estimates the 3D set of points and camera projection matrices together

Algebraic approach
- With affine camera approximation, the problem becomes that of factorizing a data matrix into a product of two matrices, describing \(M, P\).

Nonlinear Minimization (Bundle adjustment)
- Minimize

\[
\text{Energy}(M, P) = \sum_{i=1}^{m} \sum_{j=1}^{n} \text{dist}(p_{ij}, M, P)^2
\]
Reconstruction from a single image?

Surface layout

- Divides image into super-pixels
- Characterize every s.p. by features
  - Location, area, color, lines to vanishing points, etc.
- Consider 3-7 classes for world patches
  - Horizontal, vertical, sky
- Learn a classifier
- Classify …