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

Photometric Stereo – 1 camera, 3 lights

Lambertian case:

\[ I = \rho \mathbf{n} \cdot \mathbf{s} \]

Image irradiance:

\[ I_1 = \rho \mathbf{n} \cdot \mathbf{s}_1 \]
\[ I_2 = \rho \mathbf{n} \cdot \mathbf{s}_2 \]
\[ I_3 = \rho \mathbf{n} \cdot \mathbf{s}_3 \]

• We can write this in matrix form:

\[
\begin{bmatrix}
I_1 \\
I_2 \\
I_3
\end{bmatrix}
= \rho
\begin{bmatrix}
\mathbf{s}_1^T \\
\mathbf{s}_2^T \\
\mathbf{s}_3^T
\end{bmatrix}
\mathbf{n}
\]
Original Images

Results - Shape
Results - Albedo

- The task is underconstrained.
- But adding smoothness based regularization helps

\[ I = \frac{\rho}{\pi} k c \cos \theta_i = \rho \mathbf{n} \cdot \mathbf{s} \]

\[ \rho, \mathbf{n}, \mathbf{s} = ? \]

Reconstruction from a single image?

Shape from shading

Lambertian case:

\[ I = \frac{\rho}{\pi} k c \cos \theta_i = \rho \mathbf{n} \cdot \mathbf{s} \]

\[ \rho, \mathbf{n}, \mathbf{s} = ? \]

- The task is underconstrained.
- But adding smoothness based regularization helps

Minimize

\[ e = \int_{\text{image}} \left( \left( n_x \right)_x^2 + \left( n_x \right)_y^2 \right) + \left( \left( n_y \right)_x^2 + \left( n_y \right)_y^2 \right) + \lambda \left( I_{\text{given}} (x, y) - I_{\text{synth}} (n) \right)^2 dx dy \]
Depth with stereo: basic idea

Related/same: Shape from motion, structure from motion
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

Geometry for a simplified stereo

Assume:
parallel optical axes, calibrated cameras, correspondence,
- Fixation
- 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?
- 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:

\[
\begin{bmatrix}
wx_{\text{im}} \\
wy_{\text{im}} \\
w
\end{bmatrix} = 
K[R \ t] 
M
\]

Let \( M_i \) be row \( i \) of matrix \( M \)

\[
x_{\text{im}} = \frac{M_1 \cdot P_w}{M_3 \cdot P_w}
\]

\[
y_{\text{im}} = \frac{M_2 \cdot P_w}{M_3 \cdot P_w}
\]

Every point gives two constraints on \( M \)
### Estimating the projection matrix

For sufficiently large number of point ....

an over-constrained equation \( Pm = 0 \)

\[
\begin{bmatrix}
    X^{(1)} & Y^{(1)} & Z^{(1)} & 1 & 0 & 0 & 0 & 0 & 0 & -x^{(1)}X^{(1)} & -x^{(1)}Y^{(1)} & -x^{(1)}Z^{(1)} & -x^{(1)}
    \\
    0 & 0 & 0 & X^{(1)} & Y^{(1)} & Z^{(1)} & 1 & -y^{(1)}X^{(1)} & -y^{(1)}Y^{(1)} & -y^{(1)}Z^{(1)} & -y^{(1)}
\end{bmatrix}
\begin{bmatrix}
m_1 \\
m_2 \\
m_3 \\
m_4 \\
m_5 \\
m_6 \\
m_7 \\
m_8 \\
m_9 \\
m_{10}
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \end{bmatrix}
\]

Solve for \( m_i \)'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 \( P_w \)

Therefore \( P_w \) is on a line (ray)

\[
x_{im} = \frac{M_1 \cdot P_w}{M_3 \cdot P_w}
\]

\[
y_{im} = \frac{M_2 \cdot P_w}{M_3 \cdot P_w}
\]
Stereo calibration

Some options:

- Calibrate every camera separately,
  - Internal and external calibration.
  - Given a point image location, the ray is specified.

- Calibrate the cameras relative to each other
  - NOT relative to the world coordinates
  - Reason:
    - Often calibration with 3D data is not possible.
    - Sufficient for helping correspondence
  - Two options: Internal calibration known/unknown

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: terms

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

- All epipolar lines intersect at the epipole
Example: converging cameras

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

Example: motion parallel with image plane

Figure from Hartley & Zisserman
For a given stereo rig, how do we express the epipolar constraints algebraically?

Stereo geometry, with calibrated cameras:

Camera-centered coordinate systems are related by known rotation $R$ and translation $T$: $X' = RX + T$
3d rigid transformation

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

\[X' = RX + T\]

Cross product

\(\bar{a} \times \bar{b} = \bar{c}\)

\(\bar{a} \cdot \bar{c} = 0\)

\(\bar{b} \cdot \bar{c} = 0\)

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

So here, \(c\) is perpendicular to both \(a\) and \(b\), which means the dot product = 0.
From geometry to algebra

\[
\begin{align*}
\mathbf{X}' &= \mathbf{RX} + \mathbf{T} \\
\mathbf{T} \times \mathbf{X}' &= \mathbf{T} \times \mathbf{RX} + \mathbf{T} \times \mathbf{T} \\
\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') &= \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX}) = 0 \\
&= \mathbf{T} \times \mathbf{RX}
\end{align*}
\]

Matrix form of cross product

\[
\begin{pmatrix}
0 & -a_z & a_y \\
a_z & 0 & -a_x \\
-a_y & a_x & 0
\end{pmatrix}
\begin{pmatrix}
b_x \\
b_y \\
b_z
\end{pmatrix} = \begin{pmatrix}
\bar{a} \cdot \bar{c} = 0 \\
\bar{b} \cdot \bar{c} = 0
\end{pmatrix}
\]

Can be expressed as a matrix multiplication.

\[
\bar{a} \times \bar{b} = [\mathbf{a}]_{\times} \bar{b}
\]
Essential matrix

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

Let \( E = T_x R \)

\[ X'^T EX = 0 \]

This holds for \( X \) and \( X' \) which are the same 3D point in two coordinate systems.
So it holds also for the rays \( p \) and \( p' \) and for the corresponding image points (in homogeneous coordinates)

\( E \) 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.
- 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

- Relates image of corresponding points in both cameras, given rotation and translation
- Assuming intrinsic parameters are known
  \[ E = TxR \]
- \( E \) is a rank 2 matrix with two equal eigenvalues

Essential matrix example: parallel cameras

\[ R = I \quad p \sim p_i \quad p' \sim p_r \]
\[ T = [-d, 0, 0]^T \]
\[ E = [T]_pR = \begin{pmatrix} 0 & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & -d \end{pmatrix} \]
\[ p' \top E p = 0 \]
\[ \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & -d \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0 \]
\[ \Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & df \\ -dy & 0 \end{bmatrix} = 0 \]
\[ \Leftrightarrow y = y' \]

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.

Adapted from Li Zhang, C. Loop and Z. Zhang, Computing Rectifying Homographies for Stereo Vision, CVPR 1999.

Stereo image rectification: example

Source: Alyosha Efros
Can we always use the Essential matrix?

In general

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

- K known - we may infer \( p \) from \( \overline{p} \) and calculate \( E \)
- K unknown - we may get a weaker constraint

Uncalibrated case: fundamental matrix

\[ p^T E p = 0 \]

\[ \left( K^{-1} \overline{p} \right)^T E \left( K^{-1} \overline{p} \right) = 0 \]

\[ \overline{p}^T \left( K^{-1} E K^{-1} \right) \overline{p} = 0 \]

\[ \overline{p}^T F \overline{p} = 0 \]

Fundamental matrix
Fundamental matrix

- Relates pixel coordinates in the two views
- More general form than essential matrix: we remove need to know intrinsic parameters
- By estimating fundamental matrix from correspondences in pixel coordinates, we can reconstruct epipolar geometry without intrinsic or extrinsic parameters

Computing F from correspondences

Each point correspondence generates one constraint on F

\[ \bar{p}^T F \bar{p} = 0 \]

\[
\begin{bmatrix}
  u' & v' & 1
\end{bmatrix}
\begin{bmatrix}
f_{11} & f_{12} & f_{13} \\
f_{21} & f_{22} & f_{23} \\
f_{31} & f_{32} & f_{33}
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} = 0
\]

Collect n=8+ of these constraints

**The 8-point algorithm**

Normalization (to reduce condition number) is crucial.
**Stereo geometry - summary - version 1**

a. Given two cameras (a stereo pair) and a 3D object with known 3D coordinates:
   - For each camera compute the direct linear transformation from 3D to 2D

b. Move the stereo pair to a different place, take a pair of pictures and recover 3D information:
   - Calculate E, Find correspondences
   - Given two camera projection matrices and a pair of matching image points, triangulate to get the 3D point's depth.

---

**Stereo geometry - summary - version 2**

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 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.)
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 not a global transformation

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, N. Correlation

- Constraints:
  - Limited disparity
  - Match monotonicity -> piecewise smooth disparity
  - Epipolar constraint

W = 3  W = 20
Stereo as energy minimization

- For good correspondence ....
  1. Match quality - Want each pixel to find a good appearance 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

\[
E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)
\]

\[
E_{\text{data}} = \sum_i (W_1(i) - W_2(i + D(i)))^2
\]

\[
E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \rho(D(i) - D(j))
\]

- Find D everywhere by minimizing E.
- Referring to D as an MRF (Markov Random Field)
- Effective algorithms using e.g. graph cuts exist.
Stereo results

Window-based matching
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
- Approach 2 (easy): Find all ray (line) equations and solve simultaneously by least squares
- 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 deformation image distance is smallest
Stereo reconstruction pipeline

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

What will cause errors?
- Camera calibration errors
- Poor image resolution
- 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
Active stereo with structured 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
Laser scanned models

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

Time of flight (TOF) cameras

(+ Simple to use & analyze
(+ Fast
(- Suffer from daylight

Becoming more popular

One option: RF modulated light
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 …