Lecture 12: Multi-view geometry / Stereo III

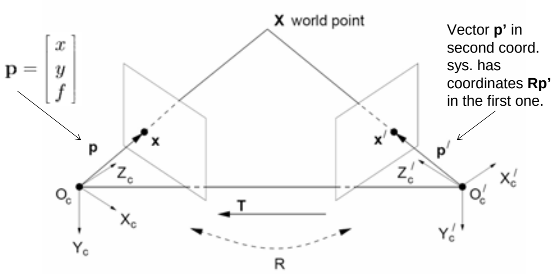
Tuesday, Oct 23

CS 378/395T
Prof. Kristen Grauman

Outline

- Last lecture:
  - stereo reconstruction with calibrated cameras
  - non-geometric correspondence constraints
- Homogeneous coordinates, projection matrices
- Camera calibration
- Weak calibration/self-calibration
  - Fundamental matrix
  - 8-point algorithm

Review: stereo with calibrated cameras



Camera-centered coordinate systems are related by known rotation **R** and translation **T**.

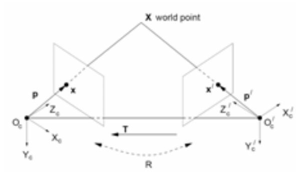
Review: the essential matrix

$$\mathbf{p} \cdot [\mathbf{T} \times (\mathbf{R}\mathbf{p}')] = 0$$

$$\mathbf{p} \cdot [\mathbf{T}_x]\mathbf{R}\mathbf{p}' = 0$$

Let $\mathbf{E} = [\mathbf{T}_x]\mathbf{R}$

$$\boxed{\mathbf{p}^T \mathbf{E} \mathbf{p}' = 0}$$



E is the **essential matrix**, which relates corresponding image points in both cameras, given the rotation and translation between their coordinate systems.

Review: stereo with calibrated cameras

- Image pair
- Detect some features
- Compute **E** from **R** and **T**
- Match features using the epipolar and other constraints
- Triangulate for 3d structure

Review: disparity/depth maps

image I(x,y)

Disparity map D(x,y)

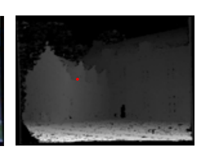
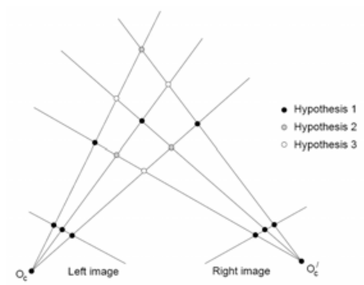


image I´(x´,y´)

(x´,y´)=(x+D(x,y),y)

Review: correspondence problem



Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Figure from Gee & Cipolla 1999

Review: correspondence problem

- To find matches in the image pair, assume
  - Most scene points visible from both views
  - Image regions for the matches are similar in appearance
- Dense or sparse matches
- Additional (non-epipolar) constraints:
  - Similarity
  - Uniqueness
  - Ordering
  - Figural continuity
  - Disparity gradient

Review: correspondence error sources

- Low-contrast / textureless image regions
- Occlusions
- Camera calibration errors
- Poor image resolution
- Violations of brightness constancy (specular reflections)
- Large motions

Homogeneous coordinates

- Extend Euclidean space: add an extra coordinate
- Points are represented by equivalence classes
- Why? This will allow us to write process of perspective projection as a matrix

2d: (x, y)' → (x, y, 1)'
3d: (x, y, z)' → (x, y, z, 1)'
} Mapping to homogeneous coordinates

2d: (x, y, w)' → (x/w, y/w)'
3d: (x, y, z, w)' → (x/w, y/w, z/w)'
} Mapping back from homogeneous coordinates

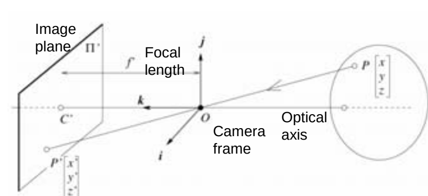
Homogeneous coordinates

•Equivalence relation:

(x, y, z, w) is the same as (kx, ky, kz, kw)

Homogeneous coordinates are only defined up to a scale

Perspective projection equations



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

Scene point → Image coordinates

Projection matrix for perspective projection

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

From pinhole camera model

Projection matrix for perspective projection

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

From pinhole camera model

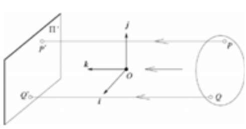
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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

$$x = \frac{fX}{Z} \quad y = \frac{fY}{Z}$$

Same thing, but written in terms of homogeneous coordinates

Projection matrix for orthographic projection



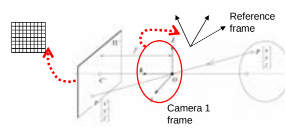
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

$$x = \frac{X}{1} \quad y = \frac{Y}{1}$$

Camera parameters

- **Extrinsic: location and orientation of camera frame with respect to reference frame**
- Intrinsic: how to map pixel coordinates to image plane coordinates



Rigid transformations

Combinations of rotations and translation

- Translation: add values to coordinates
- Rotation: matrix multiplication

Rotation about coordinate axes in 3d

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Express 3d rotation as series of rotations around coordinate axes by angles α, β, γ

Overall rotation is product of these elementary rotations:

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$

Extrinsic camera parameters

$$\mathbf{P}_c = \mathbf{R}(\mathbf{P}_w - \mathbf{T})$$

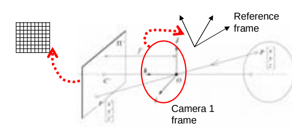
Point in camera reference frame ($\mathbf{P}_c$)

Point in world ($\mathbf{P}_w$)

$$\mathbf{P}_c = (X, Y, Z)$$

Camera parameters

- Extrinsic: location and orientation of camera frame with respect to reference frame
- **Intrinsic: how to map pixel coordinates to image plane coordinates**



Intrinsic camera parameters

- Ignoring any geometric distortions from optics, we can describe them by:

$$x = -(x_{im} - o_x)s_x$$

$$y = -(y_{im} - o_y)s_y$$

Coordinates of projected point in camera reference frame (x, y)

Coordinates of image point in pixel units (x_{im}, y_{im})

Coordinates of image center in pixel units (o_x, o_y)

Effective size of a pixel (mm) (s_x, s_y)

Camera parameters

- We know that in terms of camera reference frame:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z}$$

- Substituting previous eqns describing intrinsic and extrinsic parameters, can relate *pixels coordinates* to *world points:*

$$-(x_{im} - o_x)s_x = f\frac{\mathbf{R}_1^{\mathrm{T}}(\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3^{\mathrm{T}}(\mathbf{P}_w - \mathbf{T})}$$

$$-(y_{im} - o_y)s_y = f\frac{\mathbf{R}_2^{\mathrm{T}}(\mathbf{P}_w - \mathbf{T})}{\mathbf{R}_3^{\mathrm{T}}(\mathbf{P}_w - \mathbf{T})}$$

$\mathbf{R}_i$ = Row i of rotation matrix

Linear version of perspective projection equations

- This can be rewritten as a matrix product using homogeneous coordinates:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{M}_{int}\,\mathbf{M}_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

point in camera coordinates

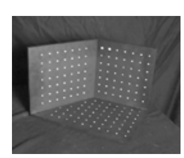
$$x_{im} = x_1 / x_3$$

$$y_{im} = x_2 / x_3$$

$$\mathbf{M}_{int} = \begin{pmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M}_{ext} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & -\mathbf{R}_1^{\mathrm{T}}\mathbf{T} \\ r_{21} & r_{22} & r_{23} & -\mathbf{R}_2^{\mathrm{T}}\mathbf{T} \\ r_{31} & r_{32} & r_{33} & -\mathbf{R}_3^{\mathrm{T}}\mathbf{T} \end{pmatrix}$$

Calibrating a camera

- 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 $\mathbf{M} = \mathbf{M}_{int}\mathbf{M}_{ext}$

The Opti-CAL Calibration Target Image

Linear version of perspective projection equations

- This can be rewritten as a matrix product using homogeneous coordinates:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\mathbf{M}_{int}\,\mathbf{M}_{ext}}_{\mathbf{M}} \overbrace{\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}}^{\mathbf{P}_w \text{ in homog.}}$$

product **M** is single **projection matrix** encoding both extrinsic and intrinsic parameters

$$x_{im} = x_1 / x_3 = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}$$

$$y_{im} = x_2 / x_3 = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w}$$

Let $\mathbf{M}_i$ be row i of matrix **M**

Estimating the projection matrix

$$x_{im} = \frac{\mathbf{M}_1 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w} \longrightarrow 0 = (\mathbf{M}_1 - x_{im}\mathbf{M}_3) \cdot \mathbf{P}_w$$

$$y_{im} = \frac{\mathbf{M}_2 \cdot \mathbf{P}_w}{\mathbf{M}_3 \cdot \mathbf{P}_w} \longrightarrow 0 = (\mathbf{M}_2 - y_{im}\mathbf{M}_3) \cdot \mathbf{P}_w$$

Estimating the projection matrix

For a given feature point:

$$0 = (\mathbf{M}_1 - x_{im}\mathbf{M}_3) \cdot \mathbf{P}_w$$

$$0 = (\mathbf{M}_2 - y_{im}\mathbf{M}_3) \cdot \mathbf{P}_w$$

In matrix form:

$$\begin{pmatrix} P_w^{\mathrm{T}} & 0^{\mathrm{T}} & -x_{im} P_w^{\mathrm{T}} \\ 0^{\mathrm{T}} & P_w^{\mathrm{T}} & -y_{im} P_w^{T} \end{pmatrix} \begin{pmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Stack rows of matrix **M**

Estimating the projection matrix

$$0 = (\mathbf{M}_1 - x_{im}\mathbf{M}_3) \cdot \mathbf{P}_w$$
$$0 = (\mathbf{M}_2 - y_{im}\mathbf{M}_3) \cdot \mathbf{P}_w$$

$$\begin{pmatrix} P_w^{\mathrm{T}} & 0^{\mathrm{T}} & -x_{im} P_w^{\mathrm{T}} \\ 0^{\mathrm{T}} & P_w^{\mathrm{T}} & -y_{im} P_w^{T} \end{pmatrix} \begin{pmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Expanding this to see the elements:

$$\begin{pmatrix} X_w & Y_w & Z_w & 1 & 0 & 0 & 0 & 0 & -x_{im}X_w & -x_{im}Y_w & -x_{im}Z_w & -x_{im} \\ 0 & 0 & 0 & 0 & X_w & Y_w & Z_w & 1 & -y_{im}X_w & -y_{im}Y_w & -y_{im}Z_w & -y_{im} \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Estimating the projection matrix

This is true for every feature point, so we can stack up *n* observed image features and their associated 3d points in single equation: **Pm = 0**

$$\overbrace{\begin{pmatrix} X_w^{(1)} & Y_w^{(1)} & Z_w^{(1)} & 1 & 0 & 0 & 0 & 0 & -x_{im}^{(1)}X_w^{(1)} & -x_{im}^{(1)}Y_w^{(1)} & -x_{im}^{(1)}Z_w^{(1)} & -x_{im}^{(1)} \\ 0 & 0 & 0 & 0 & X_w^{(1)} & Y_w^{(1)} & Z_w^{(1)} & 1 & -y_{im}^{(1)}X_w^{(1)} & -y_{im}^{(1)}Y_w^{(1)} & -y_{im}^{(1)}Z_w^{(1)} & -y_{im}^{(1)} \\ & \ldots & & \ldots & & & \ldots & & & \ldots & & \\ X_w^{(n)} & Y_w^{(n)} & Z_w^{(n)} & 1 & 0 & 0 & 0 & 0 & -x_{im}^{(n)}X_w^{(n)} & -x_{im}^{(n)}Y_w^{(n)} & -x_{im}^{(n)}Z_w^{(n)} & -x_{im}^{(n)} \\ 0 & 0 & 0 & 0 & X_w^{(n)} & Y_w^{(n)} & Z_w^{(n)} & 1 & -y_{im}^{(n)}X_w^{(n)} & -y_{im}^{(n)}Y_w^{(n)} & -y_{im}^{(n)}Z_w^{(n)} & -y_{im}^{(n)} \end{pmatrix}}^{\mathbf{P}} \overbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix}}^{\mathbf{m}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

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

Summary: camera calibration

- Associate image points with scene points on object with known geometry
- Use together with perspective projection relationship to estimate projection matrix
- (Can also solve for explicit parameters themselves)

When would we calibrate this way?

- Makes sense when geometry of system is not going to change over time
- …When would it change?

Self-calibration

- Want to estimate world geometry without requiring calibrated cameras
  - Archival videos
  - Photos from multiple unrelated users
  - Dynamic camera system
- We can still reconstruct 3d structure, up to certain ambiguities, if we can find correspondences between points…

Uncalibrated case

$$\bar{\mathbf{p}} = \mathbf{M}_{\text{int}} \underbrace{\mathbf{M}_{ext} \mathbf{P}_w}_{\mathbf{p}}$$

So:

$$\mathbf{p}_{(left)} = \mathbf{M}^{-1}_{left,\text{int}} \bar{\mathbf{p}}_{(left)}$$

$$\mathbf{p}_{(right)} = \mathbf{M}^{-1}_{right,\text{int}} \bar{\mathbf{p}}_{(right)}$$

Camera coordinates

Image pixel coordinates

Internal calibration matrices

Uncalibrated case: fundamental matrix

$$\mathbf{p}_{(left)} = \mathbf{M}^{-1}_{left,\text{int}} \bar{\mathbf{p}}_{(left)}$$

$$\mathbf{p}_{(right)} = \mathbf{M}^{-1}_{right,\text{int}} \bar{\mathbf{p}}_{(right)}$$

$$\mathbf{p}_{(right)}^{\mathrm{T}} \mathbf{E} \mathbf{p}_{(left)} = 0$$

From before, the essential matrix

Dropped subscript, still internal parameter matrices

$$\left(\mathbf{M}^{-1}_{right} \bar{\mathbf{p}}_{right}\right)^{\mathrm{T}} \mathbf{E} \left(\mathbf{M}^{-1}_{left} \bar{\mathbf{p}}_{left}\right) = 0$$

$$\bar{\mathbf{p}}^{\mathrm{T}}_{right} \underbrace{\left(\mathbf{M}^{-\mathrm{T}}_{right} \mathbf{E} \mathbf{M}^{-1}_{left}\right)}_{} \bar{\mathbf{p}}_{left} = 0$$

$$\bar{\mathbf{p}}^{\mathrm{T}}_{right} \mathbf{F} \bar{\mathbf{p}}_{left} = 0$$

Fundamental matrix

Fundamental matrix

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

Computing F from correspondences

$$\mathbf{F} = \left(\mathbf{M}^{-\mathrm{T}}_{right} \mathbf{E} \mathbf{M}^{-1}_{left}\right)$$

$$\bar{\mathbf{p}}^{\mathrm{T}}_{right} \mathbf{F} \bar{\mathbf{p}}_{left} = 0$$

- Cameras are uncalibrated: we don't know **E** or left or right $\mathbf{M}_{\text{int}}$ matrices
- Estimate F from 8+ point correspondences.

Computing F from correspondences

Each point correspondence generates one constraint on F

$$\overline{\mathbf{p}}_{right}^{\mathrm{T}}\mathbf{F}\overline{\mathbf{p}}_{left} = 0$$

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

Collect n of these constraints

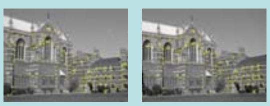
$$\begin{bmatrix} u'_1u_1 & u'_1v_1 & u'_1 & v'_1u_1 & v'_1v_1 & v'_1 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u'_nu_n & u'_nv_n & u'_1 & v'_nu_n & v'_nv_n & v'_n & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \mathbf{0}$$

Invert and solve for F. Or, if n > 8, least squares solution.

Robust computation

- Find corners
- Unguided matching – local search, cross-correlation to get some seed matches
- Compute **F** and epipolar geometry: find **F** that is *consistent with many of the seed matches*
- Now guide matching: using **F** to restrict search to epipolar lines

RANSAC application: robust computation



Interest points (Harris corners) in left and right images
about 500 pts / image
640x480 resolution

Putative correspondences (268)
(Best match,SSD<20)

Outliers (117)
(t=1.25 pixel; 43 iterations)

Inliers (151)

Final inliers (262)

Hartley & Zisserman p. 126

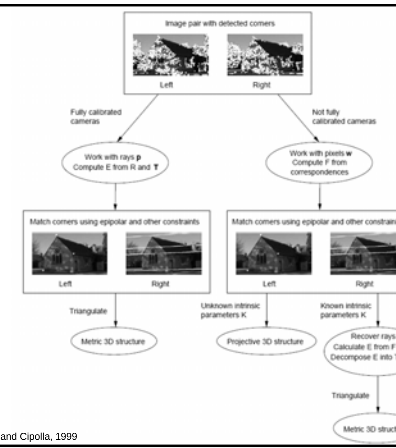


Figure by Gee and Cipolla, 1999

Need for multi-view geometry and 3d reconstruction

Applications including:

- 3d tracking
- Depth-based grouping
- Image rendering and generating interpolated or "virtual" viewpoints
- Interactive video

Z-keying for virtual reality

- Merge synthetic and real images given depth maps

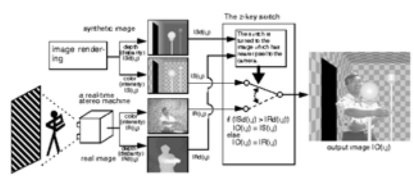
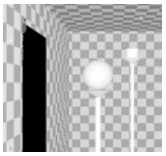


Figure 1: A schema of the z-key method

Kanade et al., CMU, 1995

Z-keying for virtual reality



http://www.cs.cmu.edu/afs/cs/project/stereo-machine/www/z-key.html

Virtualized Reality™

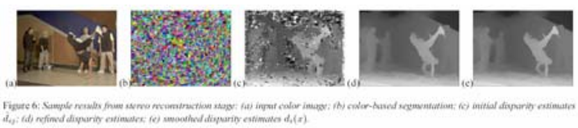
Capture 3d shape from multiple views, texture from images

Use them to generate new views on demand

Kanade et al, CMU

http://www.cs.cmu.edu/~virtualized-reality/3manbball_new.html

Virtual viewpoint video



C. Zitnick et al, High-quality video view interpolation using a layered representation, SIGGRAPH 2004.

Virtual viewpoint video

http://research.microsoft.com/IVM/VVV/

Noah Snavely, Steven M. Seitz, Richard Szeliski, "Photo tourism: Exploring photo collections in 3D," ACM Transactions on Graphics (SIGGRAPH Proceedings), 25(3), 2006, 835-846.

http://phototour.cs.washington.edu/, http://labs.live.com/photosynth/

Coming up

- Tuesday: Local invariant features
  - Read Lowe paper on SIFT
- Problem set 3 out next Tuesday, due 11/13
- Graduate students: remember paper reviews and extensions, due 12/6