Mani Golparvar-Fard
Department of Civil and Environmental Engineering
Department of Computer Science
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu
Reminders

- "App Turns iPhone into a Smarted Camera"
  - Technology Review:
    - http://www.technologyreview.com/computing/32235/?pl=MstRcnt&a=f
    - http://www.youtube.com/watch?v=b0zLgCF42Vk&feature=player_embedded

- "Google’s Art Project"
  - http://www.googleartproject.com/
    (Based on an image-based reconstruction algorithm):
    - http://www.youtube.com/watch?v=RAvnJCBYHgE&feature=player_embedded

- Camera Calibration
- Piazza
Single View Metrology

Vermeer’s *Music Lesson*

Reconstructions by Criminisi et al.
New Papers in this area

Outline

Single View Metrology

• Review calibration
• Points and Lines
• Vanishing Points
• Measuring Height

Reading: [HZ] Chapters 2,3,8

Some slides in this lecture are courtesy to Profs. D. Hoiem and S. Savarese
Outline

Single View Metrology
• Review calibration
• Points and Lines
• Vanishing Points
• Measuring Height
Calibration Problem

\[ P_i \rightarrow M P_i \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \]

World ref. system \hspace{1cm} \text{In pixels}

\[ \mathbf{M} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} \]

\[ \mathbf{K} = \begin{bmatrix} \frac{1}{\lambda} & 0 & 0 \\ 0 & \frac{1}{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & -\alpha \cot \theta & u_o \\ \beta & \frac{\beta}{\sin \theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \]
Calibration Problem

\[ (W) \]

\[ P_i \rightarrow M \rightarrow p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \]

World ref. system \hspace{1cm} In pixels

\[ M = K[R \hspace{0.5cm} T] \]

11 unknown

Need at least 6 correspondences
Calibrating the Camera

**Method 1**: Use an object (calibration grid) with known geometry

- Correspond image points to 3d points
- Get least squares solution (or non-linear solution)
Estimating the Projection Matrix

- Place a known object in the scene
  - Identify correspondence between image and scene
  - Compute mapping from scene to image

\[
\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\]

\[
M = K [R \quad T]
\]
Direct Linear Calibration

\[
\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}
\]

\[
u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}
\]

\[
v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}
\]
Direct Linear Calibration

\[
\mathbf{X} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}
\]

\[
u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}
\]

\[
v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}
\]

\[
u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}
\]

\[
v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}
\]
Direct Linear Calibration

\[ \lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \]

\[ u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \]

\[ v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \]

\[ u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03} \]

\[ v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13} \]

\[ \begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

Ax=0 form
Direct Linear Calibration

\[
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_1 & -u_1 Y_1 & -u_1 Z_1 & -u_1 \\
0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1 X_1 & -v_1 Y_1 & -v_1 Z_1 & -v_1 \\
\vdots & & & & & & & & & & & \\
X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\
0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n
\end{bmatrix}
\begin{bmatrix}
m_{00} \\
m_{01} \\
m_{02} \\
m_{03} \\
m_{10} \\
m_{11} \\
m_{12} \\
m_{13} \\
m_{20} \\
m_{21} \\
m_{22}
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

Ax=0 form

Can solve for \( m_{ij} \) by linear least squares
Calibration with linear method

- Advantages: easy to formulate and solve

- Disadvantages
  - Doesn’t tell you camera parameters
  - Doesn’t model radial distortion
  - Can’t impose constraints, such as known focal length
  - Doesn’t minimize right error function (see HZ p. 181)

- Non-linear methods are preferred
  - Define error as difference between projected points and measured points
  - Minimize error using Newton’s method or other non-linear optimization
Once the camera is calibrated...

\[ M = K[R \quad T] \]

- Internal parameters \( K \) are known
- \( R, T \) are known – but these can only relate \( C \) to the calibration rig

Can I estimate \( P \) from the measurement \( p \) from a single image?

No - in general \( \odot \) [\( P \) can be anywhere along the line defined by \( C \) and \( p \)]
Recovering structure from a single view

unknown

known

Known/
Partially known/
unknown
Outline

Single View Metrology

• Review calibration
• Points and Lines
• Vanishing Points
• Measuring Height
Camera Calibration

- What if world coordinates are not known?
- Can we use scene features (vanishing points)?
Calibrating the Camera

Method 2: Use vanishing points

- Find vanishing points corresponding to orthogonal directions
Vanishing Points and Lines

- Scene contains lines along directions that are orthogonal
Calibration by orthogonal vanishing points

- Intrinsic camera matrix
  - Use orthogonality as a constraint
  - Model K with only $f, u_0, v_0$

For vanishing points

$$p_i = KRX_i$$

$$X_i^T X_j = 0$$

- What if you don’t have three finite vanishing points?
  - Two finite VP: solve $f$, get valid $u_0, v_0$ closest to image center
  - One finite VP: $u_0, v_0$ is at vanishing point; can’t solve for $f$
Calibration by vanishing points

- Intrinsic camera matrix

\[ p_i = KRX_i \]

- Rotation matrix
  - Set directions of vanishing points
    - e.g., \( \mathbf{X}_1 = [1, 0, 0] \)
  - Each VP provides one column of \( \mathbf{R} \)
  - Special properties of \( \mathbf{R} \)
    - \( \text{inv}(\mathbf{R}) = \mathbf{R}^T \)
    - Each row and column of \( \mathbf{R} \) has unit length
Vanishing points

- Vanishing point
  - projection of a point at infinity
Vanishing points (2D)
Vanishing points

Properties
- Any two parallel lines have the same vanishing point \( v \)
- The ray from \( C \) through \( v \) is parallel to the lines
- An image may have more than one vanishing point
  - in fact every pixel is a potential vanishing point
Vanishing lines

- **Multiple Vanishing Points**
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes define different vanishing lines
Vanishing lines

- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes define different vanishing lines
Computing vanishing points

\[ P = P_0 + tD \]
Computing vanishing points

Properties

- \( P_\infty \) is a point at infinity, \( v \) is its projection
- They depend only on line direction
- Parallel lines \( P_0 + tD, P_1 + tD \) intersect at \( P_\infty \)

\[
P_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix} \quad \text{as} \quad t \rightarrow \infty \quad P_\infty = \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}
\]
Computing vanishing lines

Properties

• \( l \) is intersection of horizontal plane through \( C \) with image plane
• Compute \( l \) from two sets of parallel lines on ground plane
• All points at same height as \( C \) project to \( l \)
  ▪ points higher than \( C \) project above \( l \)
• Provides way of comparing height of objects in the scene
Is the Parachute higher than the person who is taking this picture?
Fun with vanishing points

- Perspective Cues?
Perspective cues
Perspective cues
Perspective cues
Ames Room

http://www.youtube.com/watch?v=6ajlX0AEWys&feature=related
http://www.youtube.com/watch?v=hCV2Ba5wrcs&feature=related
http://www.youtube.com/watch?v=6ajlX0AEWys&feature=related
Comparing heights
Comparing heights

Camera height

5.4
3.3
2.8
Computing vanishing points (from lines)

- Intersect $p_1q_1$ with $p_2q_2$

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Least squares version

- Better to use more than two lines and compute the “closest” point of intersection
- See notes by Bob Collins for one good way of doing this:
Measuring height without a ruler

Compute \( Z \) from image measurements

- Need more than vanishing points to do this
The cross ratio

- A Projective Invariant
  - Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points

\[
\frac{\|P_3 - P_1\| \|P_4 - P_2\|}{\|P_3 - P_2\| \|P_4 - P_1\|}
\]

\[
P_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}
\]

Can permute the point ordering
  - \(4! = 24\) different orders (but only 6 distinct values)

This is the fundamental invariant of projective geometry

Proof available on Scholar
Measuring height

\[
\begin{bmatrix}
|T - B| \\
|R - B|
\end{bmatrix}
\begin{bmatrix}
\infty - R \\
\infty - T
\end{bmatrix} = \frac{H}{R}
\]

scene cross ratio

\[
\begin{bmatrix}
|t - b| \\
|r - b|
\end{bmatrix}
\begin{bmatrix}
v_z - r \\
v_z - t
\end{bmatrix} = \frac{H}{R}
\]

image cross ratio

scene points represented as

\[
P = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\]

image points as

\[
p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
\]
Measuring height

\[ t \approx (v \times t_0) \times (r \times b) \]

vanishing line (horizon)

\[ v \approx (b \times b_0) \times (v_x \times v_y) \]

image cross ratio

\[
\frac{|| t - b ||}{|| r - b ||} \frac{|| v_z - r ||}{|| v_z - t ||} = \frac{H}{R}
\]
Measuring height

What if the point on the ground plane $b_0$ is not known?

- Here the guy is standing on the box, height of box is known
- Use one side of the box to help find $b_0$ as shown above
Computing (X,Y,Z) coordinates

- Okay, we know how to compute height (Z coords)
  - how can we compute X,Y?

- Exact same idea as before, but substitute X for Z (e.g., need a reference object with known X coordinates)
3D Modeling from a photograph

The Virtual Museum

A. Criminisi @
Microsoft, 2002
Some Related Techniques

- **Image-Based Modeling and Photo Editing**
  - Mok et al., SIGGRAPH 2001

- **Single View Modeling of Free-Form Scenes**
  - Zhang et al., CVPR 2001

- **Tour Into The Picture**
  - Anjyo et al., SIGGRAPH 1997
  - [http://koigakubo.hitachi.co.jp/little/DL_TipE.html](http://koigakubo.hitachi.co.jp/little/DL_TipE.html)
Some Related Techniques

- Image-Based Modeling and Photo Editing
  - Mok et al., SIGGRAPH 2001
  - http://graphics.csail.mit.edu/ibedit/

- Single View Modeling of Free-Form Scenes
  - Zhang et al., CVPR 2001

- Tour Into The Picture
  - Anjyo et al., SIGGRAPH 1997