Pinhole Camera Model

Computational Photography
Derek Hoiem, University of Illinois
Next classes: Single-view Geometry

How tall is this woman?

How high is the camera?

What is the camera rotation?

What is the focal length of the camera?

Which ball is closer?
Today’s class

Mapping between image and world coordinates

– Pinhole camera model
– Projective geometry
  • Vanishing points and lines
– Projection matrix
Let’s design a camera

– Idea 1: put a piece of film in front of an object
– Do we get a reasonable image?
Pinhole camera

Idea 2: add a barrier to block off most of the rays

– This reduces blurring
– The opening known as the aperture
Pinhole camera

\[ f = \text{focal length} \]
\[ c = \text{center of the camera} \]

Figure from Forsyth
Camera obscura: the pre-camera

- First idea: Mozi, China (470BC to 390BC)

- First built: Alhacen, Iraq/Egypt (965 to 1039AD)
Camera Obscurea used for Tracing

Lens Based Camera Obscurea, 1568
First Photograph

Oldest surviving photograph
  – Took 8 hours on pewter plate

Joseph Niepce, 1826

Photograph of the first photograph

Stored at UT Austin

Niepce later teamed up with Daguerre, who eventually created Daguerrotypes
Dimensionality Reduction Machine (3D to 2D)

3D world

2D image

Point of observation

Figures © Stephen E. Palmer, 2002
Projection can be tricky...
Projection can be tricky...
Projective Geometry

What is lost?

• Length

Who is taller?

Which is closer?
Length is not preserved
Projective Geometry

What is lost?

• Length
• Angles
Projective Geometry

What is preserved?

• Straight lines are still straight
Vanishing points and lines

Parallel lines in the world intersect in the image at a “vanishing point”
Vanishing points and lines

Vanishing Point

Vanishing Line
Vanishing points and lines

Vanishing point

Vertical vanishing point (at infinity)

Vanishing line

Vanishing point

Credit: Criminisi
Vanishing points and lines

Photo from online Tate collection
Vanishing objects
Projection: world coordinates $\rightarrow$ image coordinates

Optical Center $(u_0, v_0)$

$p = \begin{bmatrix} u \\ v \end{bmatrix}$

Camera Center $(t_x, t_y, t_z)$

$f$

$Z$

$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$
Homogeneous coordinates

Conversion

Converting to *homogeneous* coordinates

\[(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\]

homogeneous image coordinates

\[(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\]

homogeneous scene coordinates

Converting *from* homogeneous coordinates

\[
\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)
\]

\[
\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)
\]
Homogeneous coordinates

Invariant to scaling

\[
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
= 
\begin{bmatrix}
  kx \\
  ky \\
  kw
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
  \frac{kx}{kw} \\
  \frac{ky}{kw} \\
  \frac{kw}{kw}
\end{bmatrix}
= 
\begin{bmatrix}
  \frac{x}{w} \\
  \frac{y}{w} \\
  \frac{w}{w}
\end{bmatrix}
\]

Homogeneous Coordinates \hspace{2cm} Cartesian Coordinates

Point in Cartesian is ray in Homogeneous
Basic geometry in homogeneous coordinates

• Line equation: \( ax + by + c = 0 \)

\[
\text{line}_i = \begin{bmatrix}
a_i \\
b_i \\
c_i
\end{bmatrix}
\]

• Append 1 to pixel coordinate to get homogeneous coordinate

\[
p_i = \begin{bmatrix}
u_i \\
v_i \\
1
\end{bmatrix}
\]

• Line given by cross product of two points

\[
\text{line}_{ij} = p_i \times p_j
\]

• Intersection of two lines given by cross product of the lines

\[
q_{ij} = \text{line}_i \times \text{line}_j
\]
Another problem solved by homogeneous coordinates

Intersection of parallel lines

Cartesian: (Inf, Inf)
Homogeneous: (1, 1, 0)

Cartesian: (Inf, Inf)
Homogeneous: (1, 2, 0)
Pinhole Camera Model

\[ x = K[R \ t]X \]

- **x**: Image Coordinates: \((u,v,1)\)
- **K**: Intrinsic Matrix \((3 \times 3)\)
- **R**: Rotation \((3 \times 3)\)
- **t**: Translation \((3 \times 1)\)
- **X**: World Coordinates: \((X,Y,Z,1)\)
Interlude: when have I used this stuff?
When have I used this stuff?

Object Recognition (CVPR 2006)
When have I used this stuff?
Single-view reconstruction (SIGGRAPH 2005)
When have I used this stuff?

Getting spatial layout in indoor scenes (ICCV 2009)
When have I used this stuff?

Inserting photographed objects into images (SIGGRAPH 2007)
When have I used this stuff?

Inserting synthetic objects into images: http://vimeo.com/28962540
When have I used this stuff?

Creating detailed and complete 3D scene models from a single view (ongoing)
Projection matrix

Intrinsic Assumptions
- Unit aspect ratio
- Principal point at (0,0)
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

\[
X = K [I \quad 0] X
\]

\[
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix}
= 
\begin{bmatrix}
    f & 0 & 0 & 0 \\
    0 & f & 0 & 0 \\
    0 & 0 & 1 & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix}
\]
Remove assumption: known optical center

Intrinsic Assumptions

- Unit aspect ratio
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

\[
x = K[I \ 0]X
\]

\[
\begin{bmatrix}
    u \\
v \\
1
\end{bmatrix} = \begin{bmatrix}
f & 0 & u_0 & 0 \\
0 & f & v_0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
Remove assumption: square pixels

Intrinsic Assumptions
• No skew

Extrinsic Assumptions
• No rotation
• Camera at (0,0,0)

\[ x = K[I \quad 0]X \]  \quad \Rightarrow \quad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \]
Remove assumption: non-skewed pixels

Intrinsic Assumptions

Extrinsic Assumptions

• No rotation
• Camera at (0,0,0)

\[ x = K[I \ 0] X \]

\[ w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]

Note: different books use different notation for parameters
Oriented and Translated Camera
Intrinsic Assumptions  Extrinsic Assumptions
• No rotation

\[ x = K[I \quad t]X \]
3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:

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

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

\[ R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \]
Allow camera rotation

\[ x = K[R \ t]X \]
Degrees of freedom

\[
x = K [ R \ t ] X
\]
Vanishing Point = Projection from Infinity

\[ p = K[R \ t] \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \Rightarrow p = KR \begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow p = K \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \]

\[
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \Rightarrow
u = \frac{fx_R}{z_R} + u_0
\]

\[
\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \Rightarrow
v = \frac{fy_R}{z_R} + v_0
\]
Orthographic Projection

- Special case of perspective projection
  - Distance from the COP to the image plane is infinite
  - Also called “parallel projection”
  - What’s the projection matrix?

\[
\begin{bmatrix}
  u \\
  v \\
  w
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]
Scaled Orthographic Projection

- Special case of perspective projection
  - Object dimensions are small compared to distance to camera
  - Also called “weak perspective”
  - What’s the projection matrix?

\[
\begin{bmatrix}
  u \\
v \\
w \\
\end{bmatrix} =
\begin{bmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 0 & s \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1 \\
\end{bmatrix}
\]

Illustration from George Bebis
Take-home question

Suppose we have two 3D cubes on the ground facing the viewer, one near, one far.

1. What would they look like in perspective?
2. What would they look like in weak perspective?
Beyond Pinholes: Radial Distortion

No Distortion

Barrel Distortion

Pincushion Distortion

Corrected Barrel Distortion

Image from Martin Habbecke
Things to remember

- Vanishing points and vanishing lines
- Pinhole camera model and camera projection matrix

\[ x = K[R \ t]X \]
Next classes

• Tues: single-view metrology
  – Measuring 3D distances from the image

• Thurs: single-view 3D reconstruction