Lecture 9: Cameras
Reading

• Szeliski 2.1.3-2.1.6
Announcements

• Project 2 due Friday by 11:59pm
  – If you haven’t created your team on CMS, please do so ASAP

• Take-home midterm
  – To be handed out next Thursday, due the following Tuesday by the beginning of class

• Planning on in-class final, last lecture of class
Third Place
Second Place
Gabriel Ruttner
First Place
Jaldeep Acharya
Last time
Can we use homographies to create a 360 panorama?

- In order to figure this out, we need to learn what a camera is
• 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

- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening known as the **aperture**
  - How does this transform the image?
Camera Obscura

- Basic principle known to Mozi (470-390 BC), Aristotle (384-322 BC)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

Source: A. Efros
Camera Obscura
Home-made pinhole camera

Why so blurry?

http://www.debevec.org/Pinhole/
Pinhole photography


6-month exposure
Shrinking the aperture

• Why not make the aperture as small as possible?
  • Less light gets through
  • *Diffraction* effects...
Shrinking the aperture
Adding a lens

- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
    - other points project to a “circle of confusion” in the image
  - Changing the shape of the lens changes this distance
Lytro Lightfield Camera
The human eye is a camera

- **Iris** - colored annulus with radial muscles
- **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What’s the “film”? 
    - photoreceptor cells (rods and cones) in the retina
Top row: 1 Bengal tiger. 2 Asian elephant. 3 Zebra. 4 Chimpanzee. 5 Flamingo.
Second row: 1 Domestic cat. 2 Hairless sphynx cat. 3 Grey wolf. 4 Booted eagle. 5 Iguana.
Third row: 1 Macaw. 2 Jaguar. 3 Rabbit. 4 Cheetah 5 Horse.
Fourth row: 1 Lioness. 2 Bearded dragon (a type of lizard). 3 Leaf-tailed gecko. 4 Macaroni penguin. 5 Alligator.
Fifth row: 1 Great horned owl. 2 Mountain lion. 3 Boa constrictor. 4 Pufferfish. 5 African crested crane.
Eyes in nature: eyespots to pinhole
Projection
Projection
Müller-Lyer Illusion

http://www.michaelbach.de/ot/sze_muelue/index.html
• The coordinate system
  – We will use the pinhole model as an approximation
  – Put the optical center (Center Of Projection) at the origin
  – Put the image plane (Projection Plane) in front of the COP
    • Why?
  – The camera looks down the negative z axis
    • we need this if we want right-handed-coordinates
Modeling projection

• Projection equations
  – Compute intersection with PP of ray from \((x,y,z)\) to COP
  – Derived using similar triangles (on board)
  \[
  (x, y, z) \rightarrow \left(-d \frac{x}{z}, -d \frac{y}{z}, -d\right)
  \]
• We get the projection by throwing out the last coordinate:
  \[
  (x, y, z) \rightarrow \left(-d \frac{x}{z}, -d \frac{y}{z}\right)
  \]
Modeling projection

• Is this a linear transformation?
  • no—division by $z$ is nonlinear

Homogeneous coordinates to the rescue!

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

homogeneous image coordinates

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

homogeneous scene coordinates

Converting from homogeneous coordinates

\[
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix} \Rightarrow \left(\frac{x}{w}, \frac{y}{w}\right)
\]

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
  w
\end{bmatrix} \Rightarrow \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)
\]
Perspective Projection

Projection is a matrix multiply using homogeneous coordinates:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1/d & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
z/d \\
1
\end{bmatrix}
\Rightarrow \left( -d\frac{x}{z}, -d\frac{y}{z} \right)
\]

divide by third coordinate

This is known as perspective projection

• The matrix is the projection matrix

• (Can also represent as a 4x4 matrix – OpenGL does something like this)
Perspective Projection

- How does scaling the projection matrix change the transformation?

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1/d & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
-z/d \\
1
\end{bmatrix}
\Rightarrow (-\frac{x}{z}, -\frac{y}{z})
\]

\[
\begin{bmatrix}
-d & 0 & 0 & 0 \\
0 & -d & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
-dx \\
-dy \\
-z \\
1
\end{bmatrix}
\Rightarrow (-\frac{x}{z}, -\frac{y}{z})
\]
Orthographic projection

• Special case of perspective projection
  – Distance from the COP to the PP is infinite

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= \begin{bmatrix}
x \\
y \\
1
\end{bmatrix} \Rightarrow (x, y)
Orthographic projection
Perspective projection
Variants of orthographic projection

• Scaled orthographic
  – Also called “weak perspective”

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1/d
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x \\
y \\
1/d \\
1
\end{bmatrix}
\Rightarrow (dx, dy)
\]

• Affine projection
  – Also called “paraperspective”

\[
\begin{bmatrix}
a & b & c & d \\
e & f & g & h \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
Dimensionality Reduction Machine
(3D to 2D)

3D world

2D image

Point of observation

What have we lost?

- Angles
- Distances (lengths)

Slide by A. Efros
Figures © Stephen E. Palmer, 2002
Projection properties

• Many-to-one: any points along same ray map to same point in image
• Points → points
• Lines → lines (collinearity is preserved)
  – But line through focal point projects to a point
• Planes → planes (or half-planes)
  – But plane through focal point projects to line
Projection properties

- Parallel lines converge at a vanishing point
  - Each direction in space has its own vanishing point
  - But parallels parallel to the image plane remain parallel
Questions?
Camera parameters

• How can we model the geometry of a camera?

Two important coordinate systems:
1. World coordinate system
2. Camera coordinate system
Camera parameters

• To project a point \((x,y,z)\) in *world* coordinates into a camera
• First transform \((x,y,z)\) into *camera* coordinates
• Need to know
  – Camera position (in world coordinates)
  – Camera orientation (in world coordinates)
• The project into the image plane
  – Need to know camera *intrinsics*
Camera parameters

A camera is described by several parameters

- Translation $T$ of the optical center from the origin of world coords
- Rotation $R$ of the image plane
- focal length $f$, principle point $(x'_c, y'_c)$, pixel size $(s_x, s_y)$
- blue parameters are called “extrinsics,” red are “intrinsics”

Projection equation

$$
\begin{bmatrix}
    s_x \\
    s_y \\
    s
\end{bmatrix}
\begin{bmatrix}
    * & * & * \\
    * & * & * \\
    * & * & *
\end{bmatrix}
\begin{bmatrix}
    X \\
    Y \\
    Z
\end{bmatrix}
= \Pi X
$$

- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

$$
\Pi =
\begin{bmatrix}
    -fs_x & 0 & x'_c \\
    0 & -fs_y & y'_c \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    R_{3\times3} & 0_{3\times1} \\
    0_{1\times3} & 1 \\
    0_{1\times3} & 1
\end{bmatrix}
$$

- The definitions of these parameters are **not** completely standardized
  - especially intrinsics—varies from one book to another
Extrinsics

• How do we get the camera to “canonical form”? 
  – (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

Step 1: Translate by \(-c\)
Extrinsics

• How do we get the camera to “canonical form”?
  – (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

Step 1: Translate by $-\mathbf{c}$

How do we represent translation as a matrix multiplication?

$$
T = \begin{bmatrix}
I_{3 \times 3} & -\mathbf{c} \\
0 & 0 & 0 & 1
\end{bmatrix}
$$
Extrinsics

• How do we get the camera to “canonical form”?
  – (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

Step 1: Translate by $-c$
Step 2: Rotate by $R$

$R = \begin{bmatrix}
  u^T \\
v^T \\
w^T
\end{bmatrix}$

3x3 rotation matrix
Extrinsics

• How do we get the camera to “canonical form”?  
  – (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

Step 1: Translate by \(-c\)

Step 2: Rotate by \(R\)

\[
R = \begin{bmatrix}
    u^T \\
    v^T \\
    w^T
\end{bmatrix}
\]
Perspective projection

\[
\begin{bmatrix}
-f & 0 & 0 \\
0 & -f & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

\[K\] (intrinsic)
(converts from 3D rays in camera coordinate system to pixel coordinates)

in general, \[K = \begin{bmatrix}
-f & s & c_x \\
0 & -\alpha f & c_y \\
0 & 0 & 1
\end{bmatrix}\] (upper triangular matrix)

\(\alpha\): aspect ratio (1 unless pixels are not square)

\(s\): skew (0 unless pixels are shaped like rhombi/parallelograms)

\((c_x, c_y)\): principal point ((0,0) unless optical axis doesn’t intersect projection plane at origin)
Focal length

• Can think of as “zoom”

• Also related to field of view
Projection matrix

\[
\Pi = K \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
R & 0 \\
0 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
I_{3\times3} & -c \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(t in book’s notation)

\[
\Pi = K \begin{bmatrix}
R & -Rc \\
\end{bmatrix}
\]
Projection matrix

\[ q = (x, y, z, 1) \]  

(in homogeneous image coordinates)
Questions?