Lecture 10: Cameras

Source: S. Lazebnik
Announcements

• Project 2 is out, due next Thursday, March 8
  – Report due Friday, March 9
  – Project to be done in pairs
    • Please form groups on CMS!

• Take-home midterm
  – To be distributed in class next Thursday, March 8
  – Due at the beginning of class the following Tuesday (March 13)

• Planning on in-class final, last lecture of class
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?
• 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
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

http://upload.wikimedia.org/wikipedia/commons/6/6d/Mantis_shrimp.jpg
Projection
Projection
Müller-Lyer Illusion

http://www.michaelbach.de/ot/sze_muelue/index.html
Modeling projection

• 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 like 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—again!

\[
(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{homogeneous image coordinates}
\]

\[
(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{homogeneous scene coordinates}
\]

Converting from homogeneous coordinates

\[
\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \frac{x}{w}, \frac{y}{w}
\]

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

Projection is a matrix multiply using homogeneous coordinates:

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

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?

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

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

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

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

• Scaled orthographic
  – Also called “weak perspective”
    
    \[
    \begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1/d \\
    \end{bmatrix}
    \begin{bmatrix}
    x \\
    y \\
    z \\
    1
    \end{bmatrix}
    =
    \begin{bmatrix}
    x \\
    y \\
    1/d \\
    \end{bmatrix} \Rightarrow (dx, dy)
    \]

• 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)

What have we lost?

- Angles
- Distances (lengths)
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

How do we project a given point \((x, y, z)\) in world coordinates?
Camera parameters

• To project a point \((x,y,z)\) in \textit{world} coordinates into a camera
• First transform \((x,y,z)\) into \textit{camera} coordinates
• Need to know
  – Camera position (in world coordinates)
  – Camera orientation (in world coordinates)
• Then project into the image plane to get a pixel coordinate
  – Need to know camera \textit{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

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix}sx & * & * & * \\ sy & * & * & * \\ s & * & * & * \\ 1 & & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \Pi X$$

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

$$\Pi = \begin{bmatrix} -f_s x' & 0 & x'_c & 1 \\ 0 & -f_s y' & y'_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3x3} & 0_{3x1} \\ 0_{1x3} & 1 \\ 0_{1x3} & 1 \end{bmatrix}$$

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

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

(in homogeneous image coordinates)
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 \(-c\)

How do we represent translation as a matrix multiplication?

\[
T = \begin{bmatrix}
I_{3 \times 3} & -c \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
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 \(-\mathbf{c}\)
Step 2: Rotate by \(\mathbf{R}\)

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

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 $R$

$$R = \begin{bmatrix} u^T \\ v^T \\ w^T \\ 0 \end{bmatrix}$$
(with extra row/column of $[0 \ 0 \ 0 \ 1]$)
Perspective projection

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

\(K\) (intrinsics)

(converts from 3D rays in camera coordinate system to pixel coordinates)

in general, \(K = \begin{bmatrix}
-f & s & cx \\
0 & -\alpha f & cy \\
0 & 0 & 1
\end{bmatrix}\) (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”

24mm

50mm

200mm

800mm

• Also related to field of view
# Field of view

## APS-C Crop Body Measurement Table

<table>
<thead>
<tr>
<th>Lens</th>
<th>After 1.62 Multiplier</th>
<th>APS-C Sensor (1.62 lens multiplier) Canon 60D, 7D, 70D, T3i, T4i</th>
<th>Hand Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>18mm</td>
<td>29.16mm</td>
<td>Three hands wide at full arms length.</td>
<td></td>
</tr>
<tr>
<td>28mm</td>
<td>45.36mm</td>
<td>Slightly less than two hands wide at full arms length.</td>
<td></td>
</tr>
<tr>
<td>35mm</td>
<td>56.7mm</td>
<td>One hand + width of one fist at full arms length.</td>
<td></td>
</tr>
<tr>
<td>50mm</td>
<td>81.0mm</td>
<td>One hand wide + width of thumb at full arms length.</td>
<td></td>
</tr>
<tr>
<td>55mm</td>
<td>89.1mm</td>
<td>Slightly less than one hand wide at full arms length.</td>
<td></td>
</tr>
<tr>
<td>85mm</td>
<td>137.7mm</td>
<td>Inside edge of thumb to tip of forefinger wide with hand in “L” shape, thumb up.</td>
<td></td>
</tr>
</tbody>
</table>

Focal length in practice

- 24mm
- 50mm
- 135mm
Focal length = cropping

- 24mm
- 50mm
- 135mm
Focal length vs. viewpoint

- Telephoto makes it easier to select background (a small change in viewpoint is a big change in background).
This part converts 3D points in world coordinates to 3D rays in the camera’s coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

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

The \( K \) matrix converts 3D rays in the camera’s coordinate system to 2D image points in image (pixel) coordinates.
\[ \Pi = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3 \times 3} & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

(t in book’s notation)
Projection matrix

\[ P \in \mathbb{R}^{3 \times 4} \]

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

(in homogeneous image coordinates)
Questions?
Perspective distortion

• Problem for architectural photography: converging verticals

Source: F. Durand
Perspective distortion

- Problem for architectural photography: converging verticals

- Solution: view camera (lens shifted w.r.t. film)

Source: F. Durand

http://en.wikipedia.org/wiki/Perspective_correction_lens
Perspective distortion

• Problem for architectural photography: converging verticals

• Result:

Source: F. Durand
Perspective distortion

• What does a sphere project to?

Image source: F. Durand
Perspective distortion

• The exterior columns appear bigger
• The distortion is not due to lens flaws
• Problem pointed out by Da Vinci
Perspective distortion: People
Distortion

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens
Correcting radial distortion

from [Helmut Dersch](https://www.helmutdersch.com)
Distortion
Modeling distortion

Project \((\tilde{x}, \tilde{y}, \tilde{z})\) to “normalized” image coordinates

\[
x'_n = \frac{\tilde{x}}{\tilde{z}}
\]
\[
y'_n = \frac{\tilde{y}}{\tilde{z}}
\]

Apply radial distortion

\[
r^2 = x'_n^2 + y'_n^2
\]
\[
x'_d = x'_n(1 + \kappa_1 r^2 + \kappa_2 r^4)
\]
\[
y'_d = y'_n(1 + \kappa_1 r^2 + \kappa_2 r^4)
\]

Apply focal length, translate image center

\[
x' = f x'_d + x_c
\]
\[
y' = f y'_d + y_c
\]

• To model lens distortion
  – Use above projection operation instead of standard projection matrix multiplication
Other types of projection

• Lots of intriguing variants...
• (I’ll just mention a few fun ones)
360 degree field of view...

• Basic approach
  – Take a photo of a parabolic mirror with an orthographic lens (Nayar)
  – Or buy one a lens from a variety of omnicam manufacturers...
    • See http://www.cis.upenn.edu/~kostas/omni.html
Tilt-shift

http://www.northlight-images.co.uk/article_pages/tilt_and_shift_ts-e.html

Tilt-shift images from Olivo Barbieri and Photoshop imitations
Rotating sensor (or object)

Rollout Photographs © Justin Kerr
http://research.famsi.org/kerrmaya.html

Also known as “cyclographs”, “peripheral images”