Lecture 18: Image formation
Announcements

• Please see project guidelines
• Projects: 1-4 people
• Either pick a project idea from the list or (better!) come up with your own project
• Main deliverables: writeup and presentation
Today

- This section of the course: physical models
- Camera models
- Projection equations
The structure of ambient light

Source: Freeman, Torralba, Isola
The structure of ambient light

Source: Freeman, Torralba, Isola
The Plenoptic Function

Adelson & Bergen, 91

The intensity $P$ can be parameterized as:

$$P(X, Y, Z)$$

Eye position

Source: Freeman, Torralba, Isola
The Plenoptic Function

Adelson & Bergen, 91

The intensity $P$ can be parameterized as:

$$P(\theta, \phi, X, Y, Z)$$

Angle
The intensity $P$ can be parameterized as:

$$P(\theta, \phi, \lambda, t, X, Y, Z)$$

Wavelength, time

Source: Freeman, Torralba, Isola
The intensity $P$ can be parameterized as:

$$P(\theta, \phi, \lambda, t, X, Y, Z)$$
Let’s design a camera
  – Idea 1: put a piece of film in front of an object
  – Do we get a reasonable image?
  – No. This is a bad camera.

Source: N. Snavely
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?

Source: N. Snavely
The pinhole camera only allows rays from one point in the scene to strike each point of the paper.
Pinhole camera

Photograph by Abelardo Morell, 1991

Source: Freeman, Torralba, Isola
Pinhole camera

Photograph by Abelardo Morell, 1991

Source: Freeman, Torralba, Isola
Pinhole camera

Photograph by Abelardo Morell, 1991

Source: Freeman, Torralba, Isola
Pinhole camera

Photograph by Abelardo Morell, 1991
http://www.foundphotography.com/PhotoThoughts/archives/2005/04/pinhole_camera_2.html

Source: Freeman, Torralba, Isola
Shrinking the aperture

- Why not make the aperture as small as possible?
  - Less light gets through
  - Diffraction effects...

Source: N. Snavely
Shrinking the aperture

Source: N. Snavely
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

Source: N. Snavely
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

Source: N. Snavely
Eyes in nature: eyespots to pinhole camera

Source: Freeman, Torralba, Isola
Shadows?
Accidental pinhole camera

Source: Freeman, Torralba, Isola
Window turned into a pinhole

View outside

Source: Freeman, Torralba, Isola
Window open

Window turned into a pinhole

Source: Freeman, Torralba, Isola
Accidental pinhole camera

Pinhole and Anti-pinhole cameras

Adam L. Cohen, 1982

Source: Freeman, Torralba, Isola
Mixed accidental pinhole and anti-pinhole cameras
Mixed accidental pinhole and anti-pinhole cameras

Source: Freeman, Torralba, Isola
Mixed accidental pinhole and anti-pinhole cameras

Source: Freeman, Torralba, Isola
Mixed accidental pinhole and anti-pinhole cameras

Body as the occluder

View outside the window

Source: Freeman, Torralba, Isola
Looking for a small accidental occluder

Source: Freeman, Torralba, Isola
Looking for a small accidental occluder

Body as the occluder
Hand as the occluder
View outside the window

Source: Freeman, Torralba, Isola
Dimensionality Reduction Machine (3D to 2D)

3D world

Point of observation

2D image

Figures © Stephen E. Palmer, 2002
Projection
Projection

Source: N. Snavely
Müller-Lyer Illusion

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

Source: N. Shavely
Modeling projection

• The coordinate system
  – We use the pinhole model as an approximation
  – Put the optical center (aka Center of Projection, or COP) at the origin
  – Put the Image Plane (aka Projection Plane) in front of the COP
  – The camera looks down the positive $z$-axis, and the $y$-axis points down

Source: N. Snavely
Modeling projection

- Projection equations
  - Compute intersection with PP of ray from (X,Y,Z) to COP
  - Derived using similar triangles
    \[ (x, y, z) \rightarrow (f \frac{x}{z}, f \frac{y}{z}, f) \]
  - We get the projection by throwing out the last coordinate:
    \[ (x, y, z) \rightarrow (f \frac{x}{z}, f \frac{y}{z}) \]

Source: N. Snavely
Perspective projection

Similar triangles: \( \frac{y}{f} = \frac{Y}{Z} \)

\[ y = f \frac{Y}{Z} \]

How can we represent this more compactly?

Source: Freeman, Torralba, Isola
Homogeneous coordinates for affine transformations

Trick: add one more coordinate:

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

homogeneous image coordinates

Converting \textit{from} homogeneous coordinates

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

\[ T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \]

\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
= 
\begin{bmatrix}
x + t_x \\
y + t_y \\
1
\end{bmatrix}
\]
Affine transformations

\[ T = \begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix} \]

any transformation represented by a 3x3 matrix with last row \([ 0 0 1 ]\) we call an affine
Basic affine transformations

\[ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \]

Translate

\[ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \]

2D in-plane rotation

\[ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \]

Scale

\[ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s_{hx} & 0 \\ s_{hy} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \]

Shear

Source: N. Snavely
Perspective Projection

Projection is a matrix multiply using homogeneous coordinates:

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

divide by third coordinate

This is known as **perspective projection**

- The matrix is the **projection matrix**
Perspective Projection

How does scaling the projection matrix change the transformation?

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

Scale by \( f \): \[
\begin{bmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
=
\begin{bmatrix}
f x \\
f y \\
z \\
1
\end{bmatrix}
\Rightarrow 
(f \frac{x}{z}, f \frac{y}{z})
\]

Scaling a projection matrix produces an equivalent projection matrix!
Orthographic projection

- Special case of perspective projection
  - Distance from the COP to the PP is infinite
  - Good approximation for telephoto optics
  - Also called “parallel projection”: \((x, y, z) \rightarrow (x, y)\)
  - What’s the projection matrix?

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

Source: N. Snavely
Perspective projection

Source: N. Snavely
Projection properties

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

Source: N. Snavely
Projection properties

• Parallel lines converge at a vanishing point
  – Each direction in space has its own vanishing point
  – But lines parallel to the image plane remain parallel
Camera parameters

• How can we model the geometry of a camera?

Three important coordinate systems:
1. World coordinates
2. Camera coordinates
3. Image coordinates

How do we project a given world point \((x, y, z)\) to an image point?
Coordinate frames

- **World coordinates**
  - \( p_g \)

- **Camera coordinates**
  - \( p_i \)

- **Image coordinates**
  - \( p_{img} \)

Source: N. Snavely

Figure credit: Peter Hedman
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)
- Then project into the image plane to get image (pixel) coordinates
  - 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 $f$, principal point $(c_x, c_y)$, pixel aspect size $\alpha$
- blue parameters are called “extrinsics,” red are “intrinsics”

### Projection equation

\[
x = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \Pi X
\]

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

\[
\Pi = \begin{bmatrix}
    f & s & c_x \\
    0 & \alpha f & c_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    R_{3 \times 3} & 0_{3 \times 1} \\
    0_{1 \times 3} & 0
\end{bmatrix}
\begin{bmatrix}
    I_{3 \times 3} & T_{3 \times 1} \\
    0_{1 \times 3} & 0
\end{bmatrix}
\]

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

Source: N. Snavely
Projection matrix

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

(in homogeneous image coordinates)

Source: N. Snavely
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?

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

Source: N. Snavely
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

Source: N. Snavely
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 \(R\)

\[
R = \begin{bmatrix}
    u^T \\
    v^T \\
    w^T \\
    0 0 0 1
\end{bmatrix}
\]

(with extra row/column of \([0 \ 0 \ 0 \ 1]\))

Source: N. Snavely
Perspective projection

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

\( K \) (intrinsics)

(upper triangular matrix)

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

in general, \( K = \)

\[
\begin{bmatrix}
 f & s & c_x \\
 0 & \alpha f & c_y \\
 0 & 0 & 1 \\
\end{bmatrix}
\]

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

\( S \): skew \((0 \text{ unless pixels are shaped like rhombi/parallelograms})\)

\((c_x, c_y)\): principal point \(((w/2, h/2) \text{ unless optical axis doesn’t intersect projection plane at image center})\)

Source: N. Snavely
Typical intrinsics matrix

\[ K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \]

- **2D affine transform** corresponding to a scale by \( f \) (focal length) and a translation by \((c_x, c_y)\) (principal point)
- Maps 3D rays to 2D pixels

Source: N. Snavely
Focal length

• Can think of as “zoom”

24mm

50mm

200mm

800mm

• Also related to field of view
Projection matrix

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

The \( K \) matrix converts 3D rays in the camera’s coordinate system to 2D image points in image (pixel) coordinates.

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).

Source: N. Snavely
Projection matrix

\[ \Pi = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \text{intrinsics} \\ \text{projection} \\ \text{rotation} \\ \text{translation} \end{bmatrix} \]

(sometimes called \( t \))

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

Source: N. Snavely
Projection matrix

\[ \text{proj matrix} \]

Source: N. Snavely
Distortion

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

Source: N. Snavely
Modeling distortion

\[(\tilde{x}, \tilde{y}, \tilde{z})\]

Project to "normalized" image coordinates

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

\[ r^2 = x'_n^2 + y'_n^2 \]

Apply radial distortion

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

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

Source: N. Snavely
Correcting radial distortion

from Helmut Dersch
Next lecture: More geometry!