Cameras

UCSD CSE 167
Tzu-Mao Li
Going to 3D!
Goal: given 3D shapes, project them to an image
Goal: given 3D shapes, project them to an image

today’s topic
How does a camera work?
Camera attempt #1:
put a film in front of an object

scene

Q: what will we see on the film?
Camera attempt #2: pinhole camera

Q: what will we see on the film?
Camera obscura

“A short account of the eye and nature of vision”, 1755, James Ayscough
Camera obscura @ San Francisco
Camera lucida

https://en.wikipedia.org/wiki/Camera_lucida

Hockney-Falco thesis

paintings become much more photorealistic after the invention of camera obscura

*no strong evidence that this thesis is true!

Jan van Eyck, “Arnolfini Portrait”

also check out Aaron Hertzmann’s nice blog post
Tim’s Vermeer

a cool documentary about a 3D graphics guy
trying to reproduce Vermeer’s painting by building a camera lucida
Commercial pinhole cameras

ONDU 4x5", €160.00
Shrinking the aperture

Q: why not make the aperture as small as possible?
Q: why not make the aperture as small as possible?
- less light will come through the hole (need longer exposure time)
- diffraction effect
In computer graphics, we can simulate perfect pinhole cameras!
Simulating pinhole cameras
= projecting points

Given a point in 3D, how do we determine where it’s projected to?

scene

pinhole/aperture

film
Let’s get to a top down view
First, let’s create an “image plane” so that things are not flipped around.

In optics, sometimes image plane means the film itself.

Want to know where is \( p' \)?
First, let’s create an “image plane” so that things are not flipped around.

Want to know where is $p'$

$p'$

image plane
Next, let’s define a coordinate system

\[ p = (x, y, z) \]

\[ p' \]

\( z = -1 \)

\((0,0,0)\)
Similar triangle!

\[ p = (x, y, z) \]

\[ z = -1 \]
Similar triangle!

\[ p = (x, y, z) \]

\[ p' = \left( \frac{-x}{z}, \frac{-y}{z}, -1 \right) \]

\[ z = -1 \]

\[ (0,0,0) \]
Perspective projection = divide by $-z$

so things that are farther become smaller

$p = (x, y, z)$

$p' = \left( \frac{-x}{z}, \frac{-y}{z}, -1 \right)$

$z = -1$

$(0,0,0)$
Remember that the film has finite size

the film can only see things inside this blue area

do not hit the film
The angle spanned = Field of View (FOV)

the film can only see things inside this blue area

field of view
does not hit the film
Large field of view v.s. small field of view

https://en.wikipedia.org/wiki/Focal_length#In_photography
Two factors that affect the field of view

\[ \text{FOV} = 2 \tan^{-1} \left( \frac{1}{2} \frac{\text{film length}}{\text{focal length}} \right) \]
Two kinds of FOVs

Horizontal FOV

Vertical FOV

focal length

film width

film height

focal length
Fun things about projections
Things get distorted at large FOV

“wide-angle” cameras

small FOV

small FOV, rotated the camera
Things get distorted at large FOV

“wide-angle” cameras

large FOV

large FOV, rotated the camera
Things get distorted at large FOV

“wide-angle” cameras

run my HW3 code!
Things get distorted at large FOV

“wide-angle” cameras

happens even in phone cameras!

FOV \approx 97 \text{ degree}

https://people.csail.mit.edu/yichangshih/wide_angle_portrait/
Things get distorted at large FOV

"wide-angle" cameras

happens even in phone cameras!

perspective distortion correction in Google Pixel cameras

FOV \approx 97\ degree

https://people.csail.mit.edu/yichangshih/wide_angle_portrait/
Orthographic projection

a projection that does not introduce perspective distortion
(in real life these are called “telecentric lenses”)

thing that are farther don’t become smaller

area we can see

image plane

(0,0,0)
Orthographic vs perspective projections

orthographic projection is useful for 3D editing since parallel lines remain parallel

images from Albert Chern
Orthographic vs perspective projections

orthographic projection is useful for engineering since parallel lines remain parallel

Perspective illusion

famous illusion from Roger Shepard

human brains would try to infer the original 3D size from perspective projection
Perspective drawing using vanishing points

parallel lines in 3D will intersect at the “vanishing point” on the image

https://en.wikipedia.org/wiki/Vanishing_point
Perspective drawing using vanishing points

parallel lines in 3D will intersect at the “vanishing point” on the image

https://en.wikipedia.org/wiki/Vanishing_point
Perspective drawing using vanishing points
Perspective drawing using vanishing points
Perspective drawing using vanishing points
Perspective drawing using vanishing points
Perspective drawing using vanishing points

vanishing point
Perspective drawing using vanishing points
Perspective drawing using vanishing points
Perspective drawing in art

The Healing of the Cripple and Raising of Tabitha, by Masolino, 1426-1427

http://www.essentialvermeer.com/technique/perspective/history.html
Back to cameras
Pinhole cameras are not perfect in real life

blurry images, vignetting, etc
Most cameras use “lenses” to concentrate lights
Most cameras use “lenses” to concentrate lights.

An “in focus” point will be concentrated to one location.
Most cameras use “lenses” to concentrate lights

an “out of focus” point will spread out

“circle of confusion”
Depth of field

Most cameras use an “aperture” to control depth of field.
Large vs small aperture

Real lens systems

- sharpness
- optical aberration
- lens flare
- size
- ease to manufacture

lens design is a fun and vast field!

Optical aberration

distortion
Optical aberration

Vignetting

$L_3 \quad L_2 \quad L_1$

A

B

more light from A than B!
Optical aberration

https://en.wikipedia.org/wiki/Chromatic_aberration
Motion blur

Slow shutter speed

Fast shutter speed

https://www.csie.ntu.edu.tw/~cyy/courses/vfx/21spring/lectures/handouts/lec02_camera.pdf
Human (and animal) eyes are advanced cameras.

https://www.ucl.ac.uk/ioo/research/research-labs-and-groups/carr-lab/bestrophinopathies-resource-pages/eye/anatomy-camera-eye
Computational photography

a cool field about using computers to create better photographs (and more!)

Heide et al.: use a simple lens system with a single element, postprocess to make the image sharp (no deep learning!)

http://www.cs.ubc.ca/labs/imager/tr/2013/SimpleLensImaging/
Next: rasterization