Modeling Light

15-463: Rendering and Image Processing
Alexei Efros

On Simulating the Visual Experience

Just feed the eyes the right data
• No one will know the difference!

Philosophy:
• Ancient question: “Does the world really exist?”

Science fiction:
• Many, many, many books on the subject
• Latest take: The Matrix

Physics:
• Slowglass might be possible?

Computer Science:
• Virtual Reality

To simulate we need to know:
How and what does a person see?
Today

How do we see the world?
- Geometry of Image Formation

What do we see?
- The Plenoptic Function

How do we recreate visual reality?
- Sampling the Plenoptic Function
- Ray Reuse
- The “Theatre Workshop” metaphor

How do we see the world?

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**

The first camera

- Known to Aristotle
- Depth of the room is the focal length
- **Pencil of rays** – all rays through a point
- Can we measure distances?
Distant objects are smaller

Figure by David Forsyth

Camera Obscura

Drawing from "The Great Art of Light and Shadow"
Jesuit Athanasius Kircher, 1646.

How does the aperture size affect the image?
Shrinking the aperture

Why not make the aperture as small as possible?

- Less light gets through
- Diffraction effects…

Slide by Steve Seitz
Home-made pinhole camera

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

The reason for lenses

Slide by Steve Seitz
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

Modeling projection

The coordinate system
- We will use the pin-hole 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)
  \[
  (x, y, z) \to \left(-\frac{d}{z} x, -\frac{d}{z} y, -d\right)
  \]
- We get the projection by throwing out the last coordinate:
  \[
  (x, y, z) \to \left(-\frac{d}{z} x, -\frac{d}{z} y\right)
  \]

Homogeneous coordinates

Is this a linear transformation?
- no—division by \(z\) is nonlinear

Trick: add one more coordinate:
- \((x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\)
- \((x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\)

Converting \textit{from} homogeneous coordinates
- \[
  \begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)
  \]
- \[
  \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)
  \]
Perspective Projection

Projection is a matrix multiply using homogeneous coordinates:

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

divide by third coordinate

This is known as **perspective projection**

- The matrix is the **projection matrix**
- Can also formulate as a 4x4

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

divide by fourth coordinate

Orthographic Projection

Special case of perspective projection

- Distance from the COP to the PP is infinite

- Also called “parallel projection”
- What’s the projection matrix?

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

What if PP is spherical with center at COP?
In spherical coordinates, projection is trivial:
\[(\theta, \phi) = (\theta, \phi)\]
Note: doesn’t depend on focal length d!

The eye

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**
The Plenoptic Function

Q: What is the set of all things that we can ever see?
A: The Plenoptic Function (Adelson & Bergen)

Let’s start with a stationary person and try to parameterize everything that he can see…

Grayscale snapshot

\[ P(\theta, \phi) \]

is intensity of light
- Seen from a single viewpoint
- At a single time
- Averaged over the wavelengths of the visible spectrum
(can also do \( P(x,y) \), but spherical coordinate are nicer)
Color snapshot

\[ P(\theta,\phi,\lambda) \]

is intensity of light
- Seen from a single view point
- At a single time
- As a function of wavelength

A movie

\[ P(\theta,\phi,\lambda,t) \]

is intensity of light
- Seen from a single view point
- Over time
- As a function of wavelength
Holographic movie

\[ P(\theta, \phi, \lambda, t, V_X, V_Y, V_Z) \]

is intensity of light
- Seen from ANY viewpoint
- Over time
- As a function of wavelength

The Plenoptic Function

\[ P(\theta, \phi, \lambda, t, V_X, V_Y, V_Z) \]
- Can reconstruct every possible view, at every moment, from every position, at every wavelength
- Contains every photograph, every movie, everything that anyone has ever seen! It completely captures our visual reality! Not bad for function…
Sampling Plenoptic Function (top view)

Just lookup -- Quicktime VR

Ray

Let’s not worry about time and color:

5D

\[ P(\theta, \phi, V_X, V_Y, V_Z) \]

- 3D position
- 2D direction

Slide by Rick Szeliski and Michael Cohen
Ray Reuse

Infinite line
  • Assume light is constant (vacuum)

4D
  • 2D direction
  • 2D position
  • non-dispersive medium

Can still sample all images!
Lumigraph / Lightfield
Outside convex space

Slide by Rick Szeliski and Michael Cohen

Lumigraph - Organization
2D position
2D direction

Slide by Rick Szeliski and Michael Cohen
Lumigraph - Organization

2D position

2D position

2 plane parameterization
Lumigraph - Organization

Hold $s,t$ constant
Let $u,v$ vary
An image

Lumigraph / Lightfield

Slide by Rick Szeliski and Michael Cohen
Lumigraph - Capture

Idea 1
- Move camera carefully over s,t plane
- Gantry
  - see Lightfield paper

Lumigraph - Capture

Idea 2
- Move camera anywhere
- Rebinning
  - see Lumigraph paper
For each output pixel

- determine \( s, t, u, v \)

- either
  - use closest discrete RGB
  - interpolate near values

Nearest

- closest \( s \)
- closest \( u \)
- draw it

Blend 16 nearest

- quadrilinear interpolation
2D: Image

What is an image?

All rays through a point
  • Panorama?

Spherical Panorama

All light rays through a point form a panorama
Totally captured in a 2D array -- $P(\theta, \phi)$
Where is the geometry???

See also: 2003 New Years Eve
http://www.panoramas.dk/fullscreen3/f1.html
The “Theatre Workshop” Metaphor

(Adelson & Pentland, 1996)

Painter Lighting Designer Sheet-metal worker

desired image

Painter (images)
Lighting Designer (environment maps)

Show Naimark SF MOMA video
http://www.debevec.org/Naimark/naimark-displacements.mov

Sheet-metal Worker (geometry)

Let surface normals do all the work!
… working together

clever Italians

Want to minimize cost
Each one does what’s easiest for him
- Geometry – big things
- Images – detail
- Lighting – illumination effects

Façade demo

Campanile Movie
http://www.debevec.org/Campanile/
Next Time

Start Small:
  Image Processing
Assignment 1:
  Out by Monday (check the web)