Welcome!
Today's Agenda:

- Gamma Correction
- Depth of Field
- Skybox
- Spots, IES Profiles
- Many Lights
Gamma Correction

Human Eye

Digital representation of intensities is discrete: for ARGB32, we have 256 levels for red, green and blue.

The human eye is more sensitive to differences in luminance for dark shades. When encoding luminance, it is advantageous to have more detail in the lower regions, e.g.:

\[ L = V^\gamma \Rightarrow V = \frac{1}{\gamma} \]

For the human eye, \( \gamma = 2.33 \) is optimal*.

*: Ebner & Fairchild, Development and testing of a color space (IPT) with improved hue uniformity, 1998.
Gamma Correction

CRT Power Response

A classic CRT display converts incoming data to luminance in a non-linear way.

\[ L = V^\gamma \Rightarrow V = L^{\frac{1}{\gamma}} \]

For a typical monitor, \( \gamma = 2.2 \).

In other words:

- If we encode our luminance using \( V = L^{\frac{1}{\gamma}} \), it will be linear on the monitor.
- At the same time, this yields a distribution that has more detail for darker shades, which suits the human eye.
Gamma Correction

Practical Gamma Correction

To ensure linear response of the monitor to our synthesized images, we feed the monitor adjusted data:

\[ V = L^{1/2.2} \approx \sqrt{L} \]

What happens if we don’t do this?

1. \( L \) will be \( V^{2.2} \); the image will be too dark.
2. A linear gradient will become a quadratic gradient; a quadratic gradient will become a cubic gradient \( \rightarrow \) your lights will appear to have a very small area of influence.
Gamma Correction

Survive = Survive * Probability; diffuse;

estimation = doing it properly, class;

diffuse = true;

Radiance = SampleLight: Brand, I, AI, Ambient

x = radiance.y + radiance.z) > 0) && (depth < MAXDEPTH)

x = true;

If breadth = EvaluateDiffuse( L, N ) * Power

SampleFactor = diffuse = M2026;

weight = M12( direct, brother ),

and coefficient = dot( N, L );

E = ((weight + cosTheta�ok) / direct)

If random walk - done properly, class following:

E = SampleDiffuse( diffuse, N, r1, r2,

Random; probability;

pdf;

E = E * pdf * (dot( N, R ) / pdf);
Gamma Correction

Legacy

The response of a CRT is \( L = V^{2.2} \); what about modern screens?

Typical laptop / desktop screens have a linear response, but expect applications to provide \( \sqrt{L} \) data... So \( V \) is modified (in hardware, or by the driver): \( V = V^2 \).

\[
L \Rightarrow \sqrt{L} \Rightarrow L^2
\]

Not all screens take this legacy into account; especially beamers will often use \( \gamma = 1 \).

Gamma correct only if the hardware or video driver expects it!
Gamma Correction

Gamma Corrected Or Not?

Open gamma.gif using the windows image previewer, and zoom to the smallest level (1:1). Which bar in the right column is most similar in brightness to the right column?
Gamma Correction

Gamma Corrected Or Not?

The circle on the right consists of two halves. The left half is grey, with an intensity of 16. Is it visible on your machine?

Note: 1/16th of full power is quite significant: if this looks black, clearly \( L \) became \( L^2 \) somewhere (and thus: 1/16 became 1/256).
Gamma Correction

Consequences

How are your digital photos / DVD movies stored?

1. With gamma correction, ready to be sent to a display device that expects $\sqrt{L}$
2. Without gamma correction, expecting the image viewer to apply $\sqrt{L}$

For jpegs and mpeg video, the answer is 1: these images are already gamma corrected.

➤ Your textures may require conversion to linear space:

$$L = V^2$$
Gamma Correction

Today's Agenda:

- Gamma Correction
- Depth of Field
- Skybox
- Spots, IES Profiles
- Many Lights
Focus

A pinhole camera ensures that each pixel receives light from a single direction.

For a true pinhole, the amount of light is zero.

Actual cameras use a lens system to direct a limited set of directions to each pixel.
Focus

Objects on the focal plane appear in focus:

Light reflected from these objects to the lens end up on a single pixel on the film.
Focus

Objects before the focal plane appear out of focus:

Light reflected from these objects is spread out over several pixels on the film (the ‘circle of confusion’).
Depth of Field

Focus

Objects beyond the focal plane also appear out of focus:

Light reflected from these objects is again spread out over several pixels on the film.
Depth of Field

Circle of Confusion

Ray tracing depth of field:

Spreading out the energy returned by a single ray over multiple pixels within the circle of confusion.
Depth of Field

Circle of Confusion

Efficient depth of field:

*We place the virtual screen plane at the focal distance (from the lens).* Rays are generated on the lens, and extend through each pixel.

- All rays through the pixel will hit the object near the focal plane;
- Few rays through the pixel hit the ‘out of focus’ objects.
- Rays through other pixels may hit the same ‘out of focus’ objects.
Generating Primary Rays

Placing the virtual screen plane at the focal distance:

Recall that a $2 \times 2$ square at distance $d$ yielded a FOV that could be adjusted by changing $d$.

We can adjust $d$ without changing FOV by scaling the square and $d$ by the same factor.

Random point on the lens: generate an (ideally uniform) random point on a disc. This is non-trivial; see Global Illumination Compendium, 19a or b. Alternatively, you can use rejection sampling.

Also nice: replace the disc with a regular $n$-gon.
Depth of Field

```cpp
// Depth of Field

if (depth > 1.0)
  t = Inside ? 1.1 : 0.9;
else
  t = Inside ? 0.9 : 1.1;

w2x = w2x - t * (w2x - w2y);

R = (R - t) * (R - t);

I = diffuse;

if (refr) & (depth > 0.0)
  I = refr * I;

I = Survived ? I : 0.0;

radius = SampleLight x + SampleLight y + rad(x);

if (radius) {
  E = ((radius > 0.0) ? radius : 0.0);
}

if (rad > 0.0)
  E = SampleDiffuse( diffuse, N, r1, r2, &E, pdf );

w2x = w2x - t * (w2x - w2y);

w1x = w1x - t * (w1x - w2y);

w2y = w2y - t * (w2y - w3y);

w1y = w1y - t * (w1y - w3y);
```

Survived = SampleRend

Estimation: doing it right

Radiance = SampleLight

E = true;

if (rad > 0.0)
  E = SampleDiffuse( diffuse, N, r1, r2, &E, pdf );

// SampleRend

SampleRend = SampleLight x + SampleLight y + rad(x);

Survived = SampleDiffuse( diffuse, N, r1, r2, &E, pdf );

w2x = w2x - t * (w2x - w2y);

w1x = w1x - t * (w1x - w2y);

w2y = w2y - t * (w2y - w3y);

w1y = w1y - t * (w1y - w3y);
```
Depth of Field

f/1.8
Depth of Field

f/2.8
Depth of Field

f/4.0
Depth of Field

f/5.6
Depth of Field

Accurately Approximating DOF using Rasterization

We can accurately simulate this process using rasterization:

Instead of using a single (pinhole) camera, we use $N$ cameras located on the ‘lens’, aimed at the center of the focal plane. By averaging their images, we obtain correct depth of field.

- All ‘rays’ for a given camera use the same origin on the lens: noise will be replaced by banding.
- $N$ must be fairly large to suppress objectionable banding artifacts.
Today's Agenda:

- Gamma Correction
- Depth of Field
- Skybox
- Spots, IES Profiles
- Many Lights
Skybox

Environment Imposter

Many games use a skybox to simulate distant geometry without actually storing this geometry.
Skybox

Environment Imposter

Many games use a skybox to simulate distant geometry without actually storing this geometry.

The skybox is a $1 \times 1 \times 1$ box centered around the camera: assuming the sky is at an ‘infinite’ distance, the location of the camera inside this box is irrelevant.

Which face of the cubemap we need to use, and where it is hit by a ray is determined on ray direction alone.
High Dynamic Range

Instead of using a skybox, we can also use an equirectangular mapping, which maps azimuth to $u$ and elevation to $v$:

$$\theta = \pi(u - 1), \varphi = \pi v; \quad u = [0,2], \quad v = [0,1].$$

Converting polar coordinates to a unit vector:

$$\vec{D} = \begin{pmatrix} \sin(\varphi)\sin(\theta) \\ \cos(\varphi) \\ -\sin(\varphi)\cos(\theta) \end{pmatrix}$$

Reverse:

$$u, v = \begin{pmatrix} 1 + \frac{\arctan(\vec{D}_x, -\vec{D}_z)}{\pi} \\ \frac{\arccos(\vec{D}_y)}{\pi} \end{pmatrix}$$
Skybox

High Dynamic Range

You can find HDR panoramas on Paul Debevec's page:

http://gl.ict.usc.edu/Data/HighResProbes

Note:

A HDR skydome can be used as a light source.
Next Event Estimation for Skydomes

Useful trick:

*Use the original skydome only for rays that stumble upon it.*

For next event estimation, use a tessellated (hemi)sphere; assign to each triangle the average skydome color for the directions it covers.
Today’s Agenda:

- Gamma Correction
- Depth of Field
- Skybox
- Spots, IES Profiles
- Many Lights
Spots & IES

Ray Tracing Spotlights

Spotlight parameters:

- Brightness
- Position, direction
- Inner angle, outer angle

We can use importance sampling for spotlights, taking into account potential contribution based on these parameters.
Spots & IES

IES Profiles

Photometric data for light sources:
Measurement of the distribution of light intensity.

Can be used in e.g. 3DS Max to model lights in virtual scenes.
IESNA:LM-63-1995

[TEST] 21307

[MANUFACT] ECLIPSE LIGHTING - PENDANT LUMINAIRE

[LUMCAT] ME-XL1-QL165-277VOLT

[LUMINAIRE] WHITE PLASTIC TUBE WITH TOP AND BOTTOM OPEN

[LAMP] ONE PHILIPS 165 WATT INDUCTION LAMP

[LAMPCAT] QL165W/840. LUMEN RATING = 8289 LMS.

[OTHER] ONE PHILIPS QL165W S/1 GENERATOR OPERATING AT 277 VAC AND 147 WATTS

TILT = NONE

1 8289 1 73 1 11 -1.00 0.00 1.92 1 1 147.0000
0 2.5 5.7 10 12.5 15 17.5 20 22.5 25 27.5 30 32.5 35 37.5
40 42.5 45 47.5 50 52.5 55 57.5 60 62.5 65 67.5 70 72.5 75
77.5 80 82.5 85 87.5 90 92.5 95 97.5 100 102.5 105 107.5 110
112.5 115 117.5 120 122.5 125 127.5 130 132.5 135 137.5 140
142.5 145 147.5 150 152.5 155 157.5 160 162.5 165 167.5 170
172.5 175 177.5 180
0

879.2 897.2 941.6 1001.1 1060.8 1116.3 1165.3 1171.1 1131.6 1064.3
986.6 910.8 845.1 792.7 760.3 745.7 735.6 724.5 714.6 703.8
963.0 683.0 675.6 672.1 672.0 673.9 675.9 677.6 679.3 680.9
682.3 682.9 682.7 681.2 678.5 676.6 680.7 684.8 683.3 680.9
676.9 671.2 664.1 655.9 646.6 636.3 625.0 612.7 599.6 586.1
572.3 558.1 544.0 530.5 517.8 506.6 496.4 486.5 477.0 469.6
470.0 482.8 502.2 520.6 526.0 496.0 414.4 315.6 235.7 169.7
108.4 59.4 35.8

Format specification:
http://lumen.iee.put.poznan.pl/kw/iesna.txt

Lumens

Horizontal angles

Vertical angle

Candela values

Advanced Graphics – Various

Spots & IES
Projective Spotlight

A rectangular beam is cast from the spotlight. Illumination per direction is obtained from a bitmap.

\[ u, v = ? \]
Spots & IES
Today’s Agenda:

- Gamma Correction
- Depth of Field
- Skybox
- Spots, IES Profiles
- Many Lights
Many Lights

Multiple Lights, Efficiently

\[ L_0(p, \omega_i) \approx \text{lights} \times \frac{1}{N} \sum_{i=1}^{N} f_r(p, \omega_o, Q) L_d^J(p, Q) V(p \leftrightarrow Q) \frac{A_{Ld} \cos \theta_i \cos \theta_o}{\| p - Q \|^2} \]

Here, \( Q \) is a point on randomly chosen light \( J \).
Many Lights

Multiple Lights, Efficiently

\[ L_0(p, \omega_i) \approx \frac{\text{lights}}{N} \sum_{i=1}^{N} f(X) \]

Using importance sampling, we can pick lights with a specific probability:

\[ L_0(p, \omega_i) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(X)}{p(X)} \]

Sampling each light with an equal probability:

\[ L_0(p, \omega_i) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(X)}{1/3} = \frac{1}{N} \sum_{i=1}^{N} 3 f(X) = \frac{\text{lights}}{N} \sum_{i=1}^{N} f(X) \]
Multiple Lights, Efficiently

Note that *any* set of probabilities for the three lights will work, as long as:

- No probability is zero;
- The sum of the probabilities is one.

We can thus safely ‘guess’ a good probability for a light.

We will want to base our guess on *potential contribution*, which is proportional to:

- Solid angle
- Brightness of the light

The difference between potential contribution and actual contribution is *visibility*.

Dividing potential contribution by the sum of the potential contributions yields a valid probability for each light.
Many Lights
Many Lights

From Multiple to Many Lights

Potential contribution is proportional to:

- Solid angle
- Brightness of the light

Sadly we cannot precalculate potential contribution; it depends on the location and orientation of the light source relative to the point we are shading.

We can precalculate a less refined potential contribution based on:

- Area
- Brightness
Many Lights Array

The light array stores pointers to (or indices of) the lights in the scene. For $N$ lights, light array size $M$ is several times $N$.

Each light occupies a number of consecutive slots in is stored in the light array, proportional to its coarse potential contribution.

Selecting a random slot in the light array now yields (in constant time) a single light source $L$, with a probability of \( \frac{\text{slots for } L}{M} \).
Resampled Importance Sampling

The light array allows us to pick a light source proportional to importance. However, this importance is not very accurate.

We can improve our choice using resampled importance sampling.

1. Pick $N$ lights from the light array (where $N$ is a small number);
2. For each of these lights, determine the more accurate importance;
3. Chose a light with a probability proportional to the accurate importance.

This scheme allows for unbiased, accurate and constant time selection of a good light source.
Many Lights

Resampled Importance Sampling

Final probability for the chosen light $L$:

$$I_{\text{coarse}} \ast I_{\text{resampled}}$$

Where

$$I_{\text{coarse}} = \frac{\text{slots for } L}{M} \quad \text{and} \quad I_{\text{resampled}} = \frac{\text{potential contribution } L}{\text{summed potential contributions}}.$$

0|1|2|... | M-1
Today’s Agenda:

- Gamma Correction
- Depth of Field
- Skybox
- Spots, IES Profiles
- Many Lights
INFOMAGR – Advanced Graphics

Jacco Bikker - November 2018 - February 2019

END of “Various”

next lecture: “GPU Ray Tracing (1)”