Lecture 2: CS6250 Graphics & Visualization

- Rendering
- Image order vs. Object Order
- Surface vs. Volume Rendering
- Color
- Lights
- Surface Properties
- Cameras

Rendering

Converting graphical data into an image.

Ray tracing or ray casting:

> Simulates the interaction of light with objects by following the path of each ray of light. Usually we cast the rays from the viewer's eye back towards the scene (world) to determine what the ray strikes.

Why start at the eye, why not at the light source?

What do we do when we get an intersection with an object?

What about mutual illumination, etc.?

Ray tracing continued.

Why do we not use ray tracing for visualization?

What can we use instead?

Image-Order Methods

Methods of rendering that produce an image in the order the image is in memory are called ***image-order*** methods.

Ray tracing is an example of this type of rendering algorithm.

There are other scan-line algorithms that also use this order to render an image.

Object-Order Methods

Another way to render a scene is to look at each object in the scene sequentially. Each object is independently rendered onto the image plane. The results must be combined to form the complete image.

What are some possible problems?

Initial hardware was capable of rendering lines. Raster devices (like pixmapped displays) required that lines be drawn a pixel at a time.

Bresenham developed efficient means of rendering lines.

Surface vs. Volume Rendering

How can we render clouds?

Water?

Fog?

Smog?

Glass?

Surface Rendering

Surface rendering involves transforming object models into surfaces (usually flat planar polygons) that can then be rendered. Obtaining the surface patches for some kinds of objects is extremely difficult (e.g. fog, clouds, etc.).

Objects are modeled by some sort of surface primitive (e.g. points, lines, triangles, polygons, or 2D or 3D splines). No description of the interior of objects usually exists.

Can transparent surfaces be modeled in this way?

X-rays?

Volume Rendering

Volume rendering involves rendering not only the surfaces of objects but also their interiors. The difficulty in volume rendering is how to model the interior of objects. Usually some interior properties must be modeled, and these properties determine how the rendering must be performed.

Volume rendering can use a form of ray tracing or casting to perform its rendering operations. Imagine an X-ray passing through your body. What interactions should be modeled?

Hardware Support

What determines which rendering technique we will use?

There exists efficient hardware to support surface rendering. Hardware support for volume rendering is also available but rarer. Hardware support for ray tracing is also now available but extremely expensive.

Hardware Model

How does a CRT work?

What is a frame buffer?

What can a graphics card do?

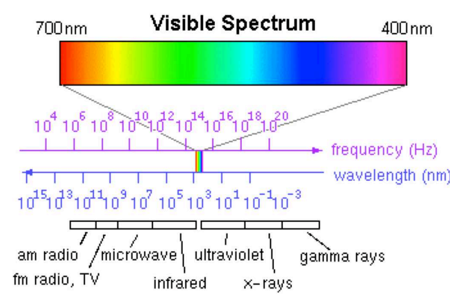
Color

What is color?

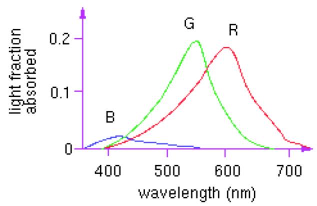
Why do we see colors?

What is a spectrum?

What wavelengths do humans see?



Human Perception of Color



Human color sensitivity:

There are four light sensitive types of cells in your retina, three are types of cones (color sensors), and the other is for low-light (rods). Each has a different light sensitivity curve as in the above diagram.

How can we explain color blindness?

Color Perception

- Different spectra can result in perceptually identical sensations called metamers.
- Color perception results from the simultaneous stimulation of the 3 cone types.
- Surround effects and adaptation also affect our perception of color.

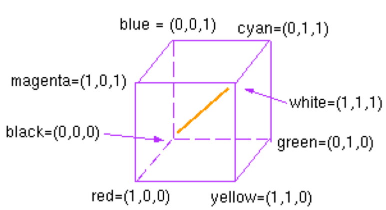
Simplified Color Models

RGB vs. HSV

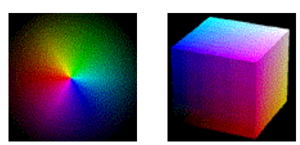
What do these abbreviations stand for?

Why use these models?

RGB



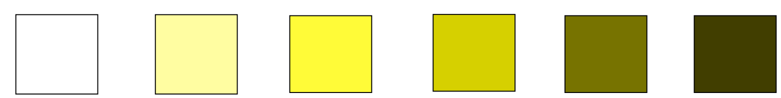
HSV



Color Continued

HSV -- Value

42 255 255	42 255 200	42 255 150	42 255 100	42 255 50	42 255 25

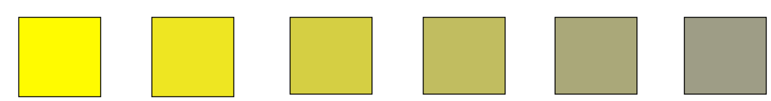


RGB

255 255 255	255 255 145	255 255 45	205 200 0	100 98 0	50 49 0

HSV -- Saturation

42 255 128	42 200 128	42 150 128	42 100 128	42 50 128	42 25 128



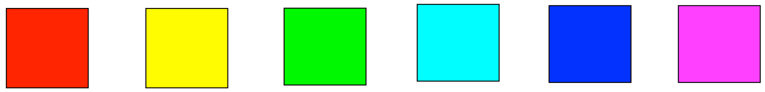
RGB

255 255 1	233 228 28	203 199 53	180 178 78	154 153 103	141 140 116

Hue

HSV -- Hue

0 255 128 | 42 255 128 | 85 255 128 | 127 255 128 | 170 255 128 | 212 255 128



RGB

255 0 0 | 255 255 0 | 0 255 0 | 0 255 255 | 0 0 255 | 255 0 255

Lights

There are two basic kinds of light source we will model – point sources and sources at infinite distance.

In addition, there is general background diffuse lighting. This light is called ***ambient*** light. It comes from light that is reflected or scattered from objects in a scene.

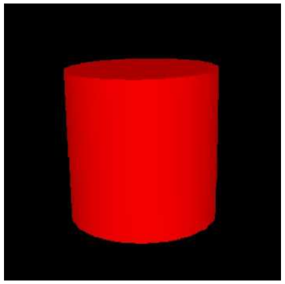
Surface Properties

How does ambient light interact with objects?

Surface Properties continued

In our simplified model for computer graphics, we will model two components of light sources that take into account the direction of the light source.

Diffuse lighting, also known as Lambertian reflection, is dependent on the angle of incidence of a light source on an object. It reflects the concept that surfaces reflect less light when they are angled more steeply from a light source. For example,



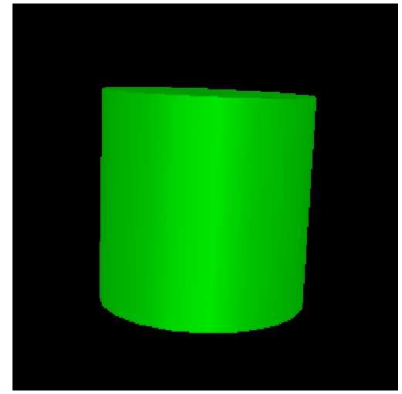
Diffuse Light Equations

What does this equation mean?

Remember that the vectors in this equation are assumed to be unit vectors.

Note: the diffuse light is independent of the observer's position!

Specular Light Sources



What is different about this image?

What equation could achieve this effect?

Specular Light Equations

where $\vec{C}_n$ is the direction of projection for the camera

and $\vec{S}$ is the direction of specular reflection.

O_{sp} is the specular power, and controls how shiny a surface looks. It controls the rate of drop-off in intensity as you get away from the specular direction.

Diagrams

Combined Lighting Model

$R_C = L_C O_{ai} O_{ac} - L_C O_{di} O_{dc}(\vec{O}_n \cdot \vec{L}_n) + L_C O_{si} O_{sc}(\vec{S} \cdot -\vec{C}_n)^{O_{sp}}$

O_{ai}, O_{di}, O_{si} are the relative amounts of ambient, diffuse, and specular lighting for an object.

O_{ac}, O_{dc}, O_{sc} specify the colors for each type of lighting.

What light source model is implicit in these equations?

Cameras

Review of camera geometry

Actors

Modeling

