Computer Graphics

Raytracing

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2018/2019 · April 04, 2019

Rendering

Generating an image for a given representation
(3D models in our case)

Rendering

Generating an image for a given representation
(3D models in our case)

Rendering

Generating an image for a given representation
(3D models in our case)

The result is called a render

The 3D models, particles, etc. are stored in a scene file

Scene file

The scene file describes the virtual scene and contains:

- Geometry
- Viewpoint
- Texture and colors
- Lighting
- Materials
- Shading
- . . .

Scene file

The scene file describes the virtual scene and contains:

- Geometry
- Viewpoint
- Texture and colors
- Lighting
- Materials
- Shading
- . . .

These are defined in a specific language which is read by a rendering program to output a digital image

The challenge

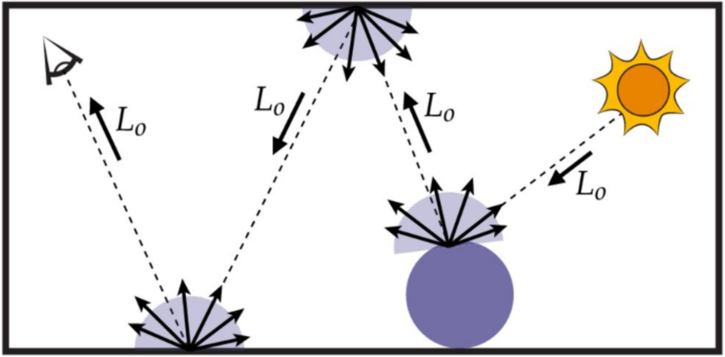
Rendering programs solve rendering equations for each point in the scene

These are integral equations describing the total amount of light emitted from each point along a particular viewing direction

The challenge

Rendering programs solve rendering equations for each point in the scene

These are integral equations describing the total amount of light emitted from each point along a particular viewing direction



The catch

Photorealism is not always a requirement

In general we just want our images to look nice, and quickly!

The catch

Photorealism is not always a requirement

In general we just want our images to look nice, and quickly!

Rendering involves several disciplines:

- Light physics
- Visual perception
- Aesthetics
- Mathematics
- Software engineering
- Algorithmics
- . . .

Rendering techniques: rasterization

Find the image pixels affected by each primitive, and modify them

Rendering techniques: rasterization

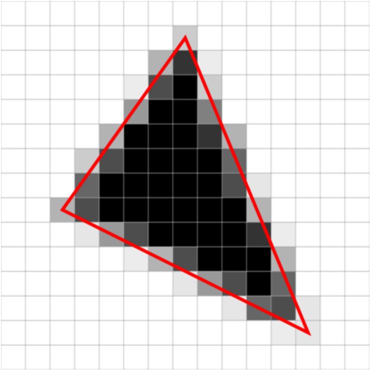
Find the image pixels affected by each primitive, and modify them

A primitive is a high-level description of a graphical element (curve, polygon, mesh, ...)

Rendering techniques: rasterization

Find the image pixels affected by each primitive, and modify them

A primitive is a high-level description of a graphical element (curve, polygon, mesh, ...)

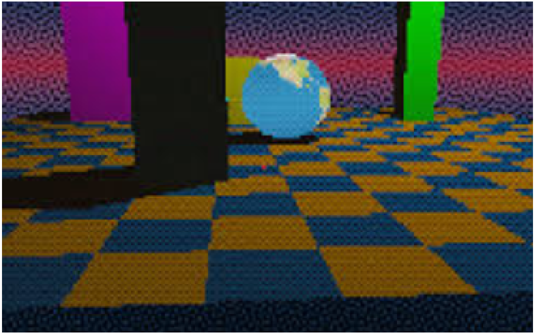


Rendering techniques: rasterization

Find the image pixels affected by each primitive, and modify them

A primitive is a high-level description of a graphical element (curve, polygon, mesh, ...)

For 3D models, it's just the mapping from scene geometry to pixels, does not prescribe a particular way to compute the colors



Rendering techniques: rasterization

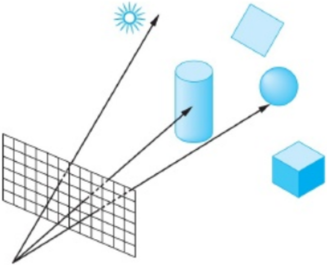
Find the image pixels affected by each primitive, and modify them

A primitive is a high-level description of a graphical element (curve, polygon, mesh, ...)

For 3D models, it's just the mapping from scene geometry to pixels, does not prescribe a particular way to compute the colors

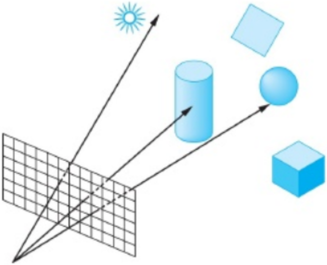
Rendering techniques: ray casting

- Cast straight rays from the point of view



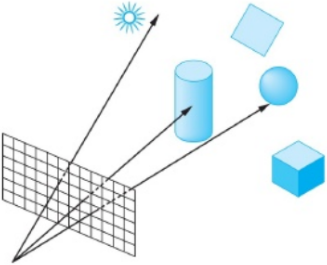
Rendering techniques: ray casting

- Cast straight rays from the point of view
- If something is intersected, compute color



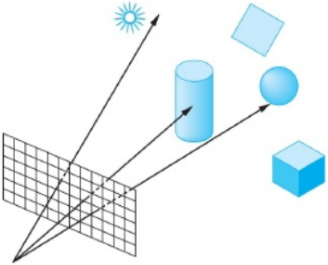
Rendering techniques: ray casting

- Cast straight rays from the point of view
- If something is intersected, compute color
- Rays do not bounce off surfaces
  So no reflection, no refraction, no decaying shadows, etc.

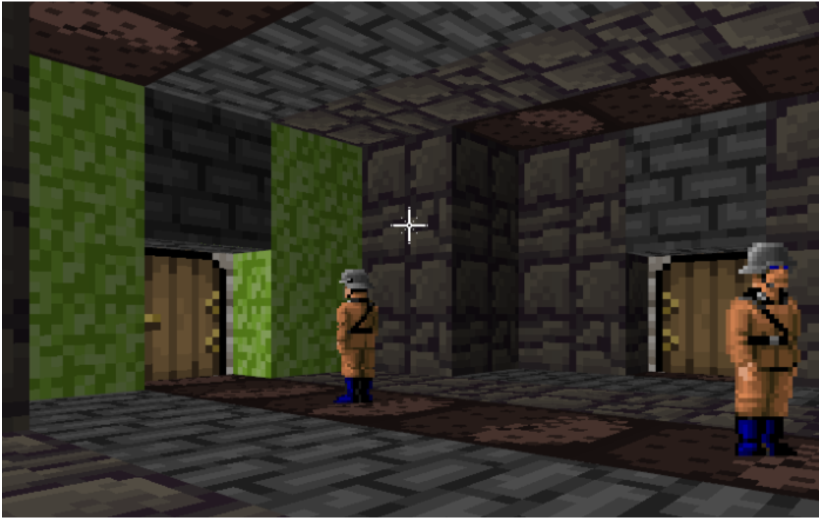


Rendering techniques: ray casting

- Cast straight rays from the point of view
- If something is intersected, compute color
- Rays do not bounce off surfaces
  So no reflection, no refraction, no decaying shadows, etc.
- Color depends on distance, angle of incidence, etc.

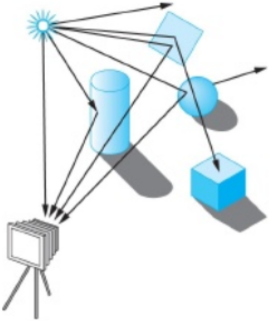


Rendering techniques: ray casting



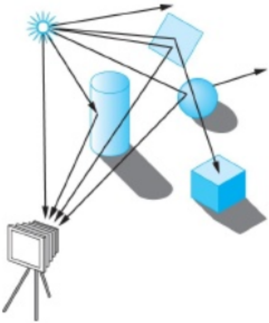
Rendering techniques: ray tracing

- Follow ("trace") rays of light from light source to viewer



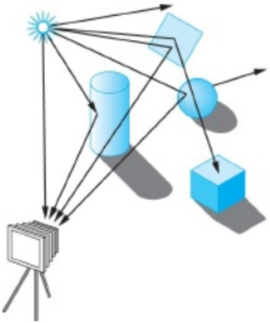
Rendering techniques: ray tracing

- Follow (“trace”) rays of light from light source to viewer
- If something is hit, simulate physical behavior (including bounces)
  Can account for reflection, dispersion, aberration, etc.



Rendering techniques: ray tracing

- Follow ("trace") rays of light from light source to viewer
- If something is hit, simulate physical behavior (including bounces)
  Can account for reflection, dispersion, aberration, etc.
- Better visual realism, higher computational cost



Rendering techniques: ray tracing

Ray tracing in hardware

Traditionally poorly suited for real-time applications such as video games

Ray tracing in hardware

Traditionally poorly suited for real-time applications such as video games

Not anymore (since March 2019)!

NVIDIA RTX GPU with dedicated raytracing core

Ray tracing in hardware

Traditionally poorly suited for real-time applications such as video games

Not anymore (since March 2019)!

NVIDIA RTX GPU with dedicated raytracing core

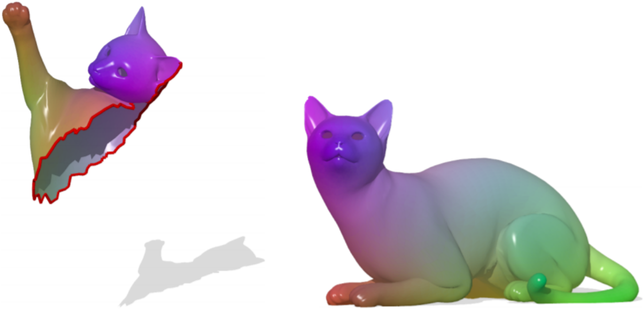
Pov-RAY

Exercise: Raytrace previous examples

Pick an example or exercise from the previous lectures, and reproduce it in POV-Ray.

Use materials and lights as you like.

Do a nice rendering!



Send me the .png image + the POV-Ray code