Comprehensive Overview

Image Synthesis
Torsten Möller
Holy Grail
Topics

• Focus on image synthesis
• Physically based rendering
  – Global illumination
  – Radiometric quantities, radiance
  – Rendering equation
  – Local illumination models
  – Monte-Carlo methods and ray-tracing
  – Radiosity
• Interactive rendering
  – Direct illumination
  – Scanline conversion and projection methods
  – Textures and texture mapping
  – Shaders
  – Graphics hardware, GPUs (graphics processing units)
  – Real-time shadows
Topics cont.

• Image synthesis pipeline
  – Human visual perception
  – Tone mapping

• However, not covered in this course:
  – Non-photorealistic rendering (NPR)
    • Artistic rendering
    • Illustrations, visualization
Photorealism vs. Interactivity

• Realistic images (“like photographs“) by physical simulation of
  – Light emission and propagation
  – Interaction between light and matter (reflection, scattering, …)
  – Image formation (image recording, human visual perception)
  – In general, realistic image synthesis is computationally expensive and non-interactive

• Interactive graphics
  – Less realistic
  – Lots of approximations and “dirty” tricks
  – Programmable graphics hardware (GPUs) allow for increasingly realistic real-time rendering
Holy Grail of Photorealistic Rendering
Example
Example

© blue moon rendering
Example
Example
Example
Course topics

• Laws of image generation
  – Physics of Light - Optics, Transport
• Mechanics of image generation
  – Global Illumination
• Display Mechanism
  – Perception
• Efficient Mechanics
Image formation
Framework

Research Framework for Realistic Image Synthesis

© 1997 Cornell Program of Computer Graphics
Generic Framework

Acquisition -> Light Transport -> Visual Display -> Observer


Geometry BRDFs Lights Textures -> Radiometric Values -> Image

[Greenberg et al. 97]
Key Terms

- **Description of light**
  - Physical quantities, photometry

- **Illumination model**
  - Determine the color of a surface (data) point by simulating some light attributes

- **Local illumination**
  - Deals only with isolated surface (data) point and direct light sources

- **Global illumination**
  - Takes into account the relationships between all surfaces (points) in the environment

- **Shading model**
  - Applies the illumination models at a set of points and colors the whole scene
Description of Light

© 1994 by John Mardaljevic – Radiance Software
Measure It!

Equipment in the Light Measurement Laboratory (circa 1995) of the Cornell University Program of Computer Graphics

© Machiraju/Möller
Local Illumination

• Local reflection
  – Building block for complete illumination computation

• Local illumination models
  – Heuristic
  – Physics-based
  – Measured data
Global Illumination

- Global problem: each point might affect the illumination of any other point
- This makes illumination computations complicated
  - Use appropriate numerical methods
  - Efficient algorithms
  - Approximations and simplifications

What is the “intensity” of this surface in all possible directions?
Shading

Flat shading

Gouraud shading

© 1995 Foley, van Dam et al.

Gouraud shading

Phong shading
Physically Based Illumination

- Everything so far has been pretty heuristic
- cannot model:
  - wavelength dependent phenomena
  - anisotropic behaviours
  - many other physical phenomena (real physics)
- ongoing research - main contributions
  - Hanrahan, Krüger (1993)
Light Simulation - Global Illumination

Streaming

\[ \vec{\omega} \cdot \nabla L(x, \vec{\omega}) = L_o(x, \vec{\omega}) - \sigma_a(x)L(x, \vec{\omega}) - \sigma_s(x)L(x, \vec{\omega}) \]

Absorption

Out-scattering

Outgoing Emittance

\[ L_o(x, \vec{\omega}_o) = L_e(x, \vec{\omega}_o) + \sigma_s(x) \int_{\Omega} f_r(x, \vec{\omega}_o, \vec{\omega}_i)L_i(x, \vec{\omega}_i)d\vec{\omega}_i \]

In-scattering

© Machiraju/Möller
Challenges

• Primitives complex: lights, materials, shapes

• Materials
  – Interfaces: reflectance and texture
  – Medium: scattering

• Camera

• Large number of paths

• Solution:
  – Radiosity - Finite Elements
  – Ray Tracing - Monte Carlo
Radiosity

• Finite element methods
• Not efficient - storage
• Meshing problems
  – curved surfaces
  – shadows
• Cannot do it all:
  – Complex effects beyond diffuse
Monte Carlo Ray Tracing?

- Distributed Ray Tracing
- Path Tracing
- Metropolis Light Transport
- Bi-Directional Path Tracing
- Photon Maps
Caustics

Indirect Illumination
SubSurface Scattering

"A Practical Model for Subsurface Light Transport"
Henrik Wann Jensen, Steve Marschner, Marc Levoy, and Pat Hanrahan
Proceedings of SIGGRAPH'2001, pages 511-518, Los Angeles, August 2001
Translucency

© 2001 Jensen, Marschner, Levoy, Hanrahan
"Visual Simulation of Smoke"
Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen
Display Limitations

© 2001 Richard Sharp
Real-Time Rendering

- Approximations of the full light transport equation
  - Up to shadow computation and reflections
- Use of special hardware: GPUs (graphics processing units)
  - Shaders and shading languages
- Textures to encode detail information
Tone Mapping and Visual Perception

- Issue: How do we perceive images?
- There is a lot of processing done by the human visual system
  - Especially in stages of early vision
  - Just a direct display of physical light quantities is not good enough
- Add a perception-aware stage to the image processing pipeline
- Related issue: restrictions of output display
  (color gamut, contrast, and brightness)

- a) daylight: 1000 cd/m^2
- c) moonlight: 0.04 cd/m^2
Non-Photorealistic Rendering

• Rendering alternatives

Model

Photorealism

Non-photorealism (NPR)

[from A. Finkelstein, SIGGRAPH 05 Course Notes]
Non-Photorealistic Rendering cont.

• Imitate artistic and illustrative rendering styles
  – Pen-and-ink drawings (hatching etc.)
  – Painterly rendering
  – Watercolor

• Improve visual perception
  – Silhouette and feature lines
  – Emphasize important features
  – Hide unimportant details

• Goals
  – Explanation
  – Illustration
  – Storytelling
Non-Photorealistic Rendering cont.

• Technical and algorithmic aspects
  – How can we simulate artistic rendering styles?
  – Efficient extraction of feature lines etc.?
  – Can we adapt photorealistic rendering / illumination models to incorporate NPR? How?
  – Interactive rendering?