Real-Time High Quality Rendering

CSE 274 [Fall 2015], Lecture 2

Graphics Hardware Pipeline, Reflection and Rendering Equations, Taxonomy of Methods

http://www.cs.ucsd.edu/~ravir

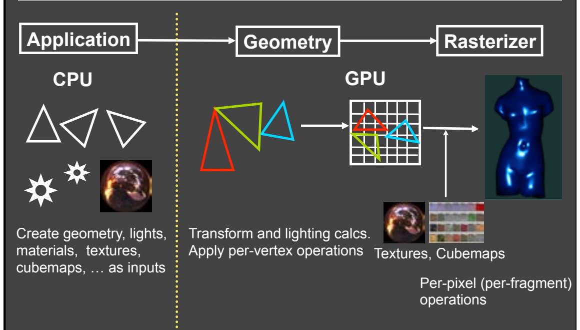
Outline of Lecture

- *Basics of hardware pipeline*
- Reflection and Rendering equations
- Typical Lighting, shading in hardware
- Taxonomy of methods/papers

- Assignment: Sign up for paper presentations
  - And send basic info:
    - Name, e-mail, status (Senior, PhD etc.) Background in graphics/comments
    - Will you be taking course grades or P/F

Basic concepts/review only. If interested in more background/in depth detail, refer to handouts

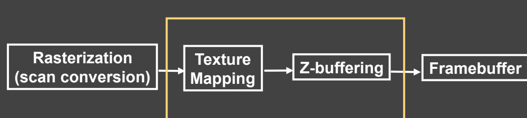
Basic Hardware Pipeline



Geometry or Vertex Pipeline

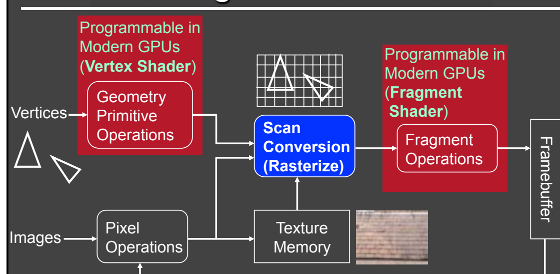
These fixed function stages can be replaced by a general per-vertex calculation using vertex shaders in modern programmable hardware

Pixel or Fragment Pipeline



These fixed function stages can be replaced by a general per-fragment calculation using fragment shaders in modern programmable hardware

GPU Programmable Shaders



Traditional Approach: Fixed function pipeline (state machine)
New Development (2003-): Programmable pipeline

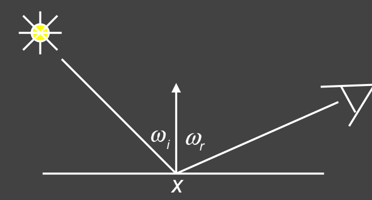
Outline of Lecture

- Basics of hardware pipeline
- *Reflection and Rendering equations*
- Typical Lighting, shading in hardware
- Taxonomy of methods/papers

- Assignment: Sign up for paper presentations

Basic concepts/review only. If interested in more background/in depth detail, refer to handouts

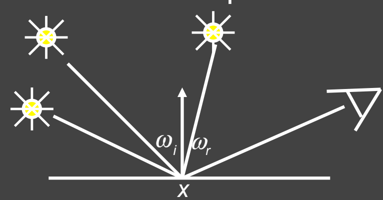
Reflection Equation



$$L_r(x,\omega_r) = L_e(x,\omega_r) + L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

Reflection Equation

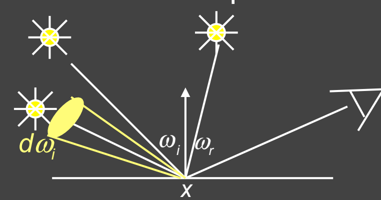


Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \sum L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

Reflection Equation



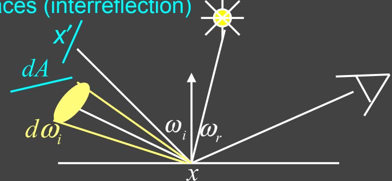
Replace sum with integral

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_i(x,\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

Rendering Equation

Surfaces (interreflection)



$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

Reflected Light (Output Image) UNKNOWN | Emission KNOWN | Reflected Light UNKNOWN | BRDF KNOWN | Cosine of Incident angle KNOWN

Rendering Equation (Kajiya 86)

Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.

Rendering Equation as Integral Equation

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i)\, f(x,\omega_i,\omega_r)\cos\theta_i\, d\omega_i$$

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

Is a Fredholm Integral Equation of second kind [extensively studied numerically] with canonical form

$$l(u) = e(u) + \int l(v)\, K(u,v)\, dv$$

Kernel of equation

Linear Operator Equation

$$l(u) = e(u) + \int l(v)\, K(u,v)\, dv$$

Kernel of equation
Light Transport Operator

$$L = E + KL$$

Can be discretized to a simple matrix equation [or system of simultaneous linear equations] (L, E are vectors, K is the light transport matrix)

Ray Tracing and extensions

- General class numerical ***Monte Carlo*** methods
- Approximate set of all paths of light in scene

$$L = E + KL$$
$$IL - KL = E$$
$$(I-K)L = E$$
$$L = (I-K)^{-1}E$$

Binomial Theorem

$$L = (I + K + K^2 + K^3 + \ldots)E$$
$$L = E + KE + K^2E + K^3E + \ldots$$

Ray Tracing

$$L = E + KE + K^2E + K^3E + \ldots$$

- E: Emission directly From light sources
- KE: Direct Illumination on surfaces
- K^2E: Global Illumination (One bounce indirect) [Mirrors, Refraction]
- K^3E: (Two bounce indirect) [Caustics etc]

Ray Tracing

$$L = E + KE + K^2E + K^3E + \ldots$$

- E: Emission directly From light sources
- KE: Direct Illumination on surfaces
- K^2E: Global Illumination (One bounce indirect) [Mirrors, Refraction]
- K^3E: (Two bounce indirect) [Caustics etc]

OpenGL Shading

Outline of Lecture

- Basics of hardware pipeline
- Reflection and Rendering equations
- *Typical Lighting, shading in hardware*
- Taxonomy of methods/papers
- Assignment: Sign up for paper presentations

Basic concepts/review only. If interested in more background/in depth detail, refer to handouts

OpenGL: Sum of Components

Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \sum L_i\, f(\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) · Emission · Incident Light (from light source) · BRDF · Cosine of Incident angle

- Ambient
- Emission
- Diffuse
- Specular

Ambient term

Global constant (sometimes per light) added to everything

- In addition to other terms in reflection equation
- Fakes indirect illumination, broad area lights
- Prevents completely black regions etc.
- Hack, no physical basis

Emissive Term (for Light Sources)

Sum over all light sources

$$L_r(x,\omega_r) = \boxed{L_e(x,\omega_r)} + \sum L_i\, f(\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image) · Emission · Incident Light (from light source) · BRDF · Cosine of Incident angle

Diffuse Term (Lambertian Reflection)

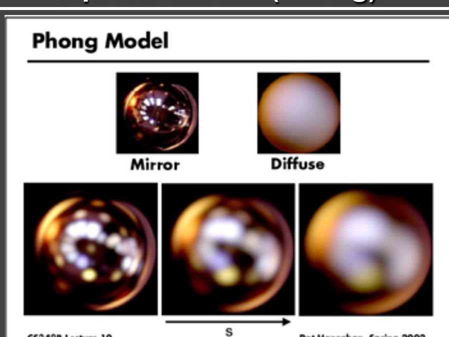
Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \boxed{\sum L_i\, f_d\, (\omega_i \bullet n)}$$

Reflected Light (Output Image) · Emission · Incident Light (from light source) · albedo · Cosine of Incident angle

BRDF constant (Lambertian) ; mult. by diffuse albedo

Specular Term (Phong)

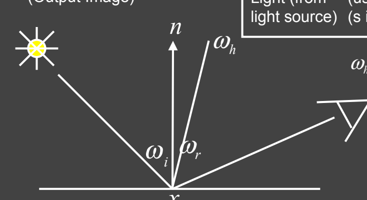


Specular Term (Blinn-Phong)

Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \boxed{\sum L_i\, (\omega_h \bullet n)^s}$$

Reflected Light (Output Image) · Emission · Incident Light (from light source) · Blinn-Phong mode (using half-angle) (s is shininess)



$$\omega_h = \frac{\omega_i + \omega_o}{|\omega_i + \omega_o|}$$

Outline of Lecture

- Basics of hardware pipeline
- Reflection and Rendering equations
- Typical Lighting, shading in hardware
- *Taxonomy of methods/papers*

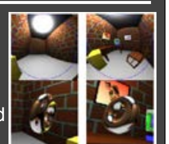
- Assignment: Sign up for paper presentations

Basic concepts/review only. If interested in more background/in depth detail, refer to handouts

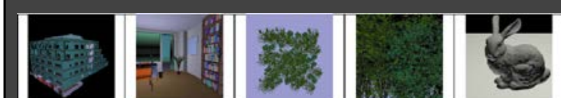
Syllabus / Taxonomy: Basic Hardware

- Basic Hardware Techniques (next wk)
  - Shadow Mapping
  - Environment Mapping
- Graphics Hardware
  - Geometry Engine (82)
  - Reality Engine (83)
  - *Realistic Hardware-Accelerated Shading and Lighting*
  - *Real-Time Procedural Shading (01)*
  - Ray Tracing on Graphics Hardware (02)
  - Brook: Stream Computing (04)
  - Photon mapping on hardware (03)
  - *GPU-Based Interactive Global Illumination (09)*

High Quality Hardware Shading



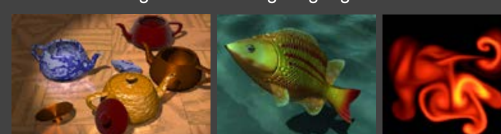
- Heidrich 99, one of first papers. Hardware tricks (before current programmable graphics; in standard OpenGL)
- Purcell et al. 02, map ray tracing to standard programmable hardware scanline pipeline



- Lindholm et al. 01, first nVidia vertex shaders (GeForce 3)

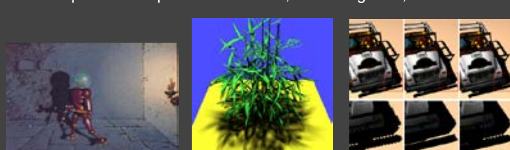
General Programmable Shading

- First programmable shading systems for hardware
  - Peercy et al. 00: Multipass OpenGL implementation of RenderMan (OpenGL as general SIMD machine)
  - Proudfoot et al. 01: Multiple computation frequencies (precursor to vertex, fragment shaders)
- Mark et al. 03: Describes nVidia's Cg: one of first commercial high-level shading languages



Shadow Mapping

- Classic technique to add complex curved shadows
  - Williams 78, oldest paper we read in course
- Many recent extensions for programmable hardware
  - Soft shadows: Agrawala 00
  - Complex geometry like hair: Lokovic and Veach 00
  - Adaptive techniques: Fernando 01, Stamminger 02, Sen 03



Syllabus/Taxonomy

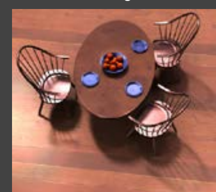
- Hardware Ray Tracing
  - Multi-Level CPU Ray Tracing (05)
  - OptiX (10)
  - Embree (14)
- Image-Based Rendering (background)
  - Light Fields and Lumigraphs (96)
  - Surface Light Fields (00)
  - Reflectance Fields (00)
- Signal Processing (background)
  - Plenoptic Sampling (00)
  - Signal-Processing Framework (01)
  - Frequency Analysis of Light Transport (05)

New Ideas

- Precomputed Rendering
  - Precomputed Radiance Transfer (02)
  - Clustered PCA (03)
  - All-Frequency Shadows (03)
  - Newer Papers (09, 11, 13)
- Sparse Sampling and Reconstruction
  - (background) (08,09,12)
  - Axis-Aligned Filtering (12,13)
- Sampling of Recent Work
  - Anisotropic Spherical Gaussians (13)
  - Convolution Shadows, Volumes (13)
  - Axis-Aligned Distribution Effects (14)
  - Interreflections all-frequency BRDFs (14)
  - Dynamic ray stream traversal (14)
  - Machine learning filtering noise (15)

Precomputed Transfer

- Precompute on static scenes, followed by real-time relighting, changing view. Can capture most complex shading effects.
- Sloan et al. 02,03: Low-frequency SH, compression
- Ng et al. 03, 04: Wavelets all-frequency, relight, view change
- Sloan 04, Wang 04: Factored BRDFs all-frequency relighting



Outline of Lecture

- Basics of hardware pipeline
- Reflection and Rendering equations
- Typical Lighting, shading in hardware
- Taxonomy of methods/papers

- *Assignment: Sign up for paper presentations*

Basic concepts/review only. If interested in more background/in depth detail, refer to handouts

Paper Presentations

- E-mail me the top 3-5 papers you want to present
- Only those that say “presented by students”
- Brief description of projects next week (see schedule)
- Project milestone and final proposal 3 weeks after

Questions?