Rendering SS2011, Andrea Weidlich

Onur Dogangönül

November 11, 2011

Contents

Part I
Rendering Theory

1 Rendering Equation

1.1 Surface Area Formalism

- 1986 Kajia version of RE (Fredholm Integro-differential equation)

$$I(x,x') = g(x,x')[\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx'']$$

- $I(x,x')$: Amount of light from x' to x
- $g(x,x')$
  - Geometric term desribes setting between x' and x
  - Usually light energy reduced by $\frac{1}{r^2}$
  - Is zero, if an object lies between these points
- $\epsilon(x,x')$: Direct light energy, if x' emitter
- $\rho(x,x',x'')$: Indirect light, which is recived from x'' via x' (BRDF etc.)
- $\int_S$: Must be evaluated over all surfaces in the scene
- Completely describes problem of image synthesis

1.2 Directional Formalism

- Alternative RE geometry

$$L(\vec{x},\omega) = L^e(\vec{x},\omega) + \int_\Omega L(h(\vec{x},-\omega'),\omega') \cdot f_r(\omega',\vec{x},\omega) \cdot \cos\theta \mathrm{d}\omega'$$

- $L(\vec{x},\omega)$
  - Describes amount of light on pos x in direction ω
  - Correlates to $I(x,x')$
- $h(\vec{x},-\omega')$: Correlates to $g(x,x')$.
- $L^e(\vec{x},\omega)$

- Direct light emitted from x in ω direction
- Correlates to $\epsilon(x, x')$

- $f_r(\omega', \vec{x}, \omega)$: Correlates to $\rho(x, x', x'')$.
- $\int_\Omega$: Evaluated over whole hemisphere.
- Emission term is outside of sintegral

1.3 Short Form

- Given the directional formalism, integral operator T introduces

$$TL(\vec{x}, \omega) = \int_\Omega L(h(\vec{x}, -\omega'), \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos\theta \mathrm{d}\omega'$$

- Short: $L = L^e + TL$
- Analytical solutions are usually impossible
- Contractivity of the integral operator T or adjunct operator T'
  - In physically plausible environments, operators T and T' are contractive since surface reflectances are < 1
  - Successive application yields smaller results
  - Highly specular surfaces less contractive than diffuse environments (i.e. iterative algorithms take longer to converge)

2 Potential Equation

- From viewpoint of emitter

$$(T'W)(\vec{y}, \omega') = \int_\Omega W(h(\vec{y}, \omega'), \omega') \cdot f_r(\omega', h(\vec{y}, \omega'), \omega) \cdot \cos\theta \mathrm{d}\omega$$

- T' is the adjoint operator of T from RE
- Short: $W = W^e + T'W$

2.1 RE, PE Symmetry and Difference

- Symmetry
  - RE: From viewpoint of the receiver
  - PE: From viewpoint of the emitter
  - Both equations of similar tpye and approached in similar ways

  - PE suites better for certain methods

- Differences
  - RE: Radiance values for viewing ray (immediately useful for rendering)
  - PE: Radiance state of entire scene, results are stored and evaluated later

3 Global Illumination Solution Strategies

- Solution Technique Classification
  - Local illuminations models
    * Coupling ignored
    * No recursion
  - Recursive ray-tracing
    * Easy couplings are followed
    * Specular and transmission
  - Global illumination methods: Full treatment of RE coupling

- GI solutions
  - Inversion: Not used in practice, forget it
  - Expansion: Mostly stochastic techniques (ray tracers are exception)
  - Iteration: Both stochastic and deterministic (finite element) approaches

3.1 Inversion

- Inversion brings variables to one side
- Is a solution for simple equations but does not apply to the infinite dimensional integral operator (cannot be inverted in closed form)
- Finite element approaches
  - Yield a system of linear equations
  - Have to be inverted
  - Algorithms not used due to cubic complexity and numerical instability

3.2 Expansion

3.2.1 Recursive Substiution

- $L = L^e + TL$
- $L = L^e + T(L^e + TL) = L^e + TL^e + T^2L$
- Repeating n times results in neumann series

$$L = \sum_{i=0}^{n} T^i L^e + T^{n+1} L$$

- T is a contraction and the infinith iteration is zero.

$$\lim_{n \to \infty} T^{n+1} L = 0$$

- Leads to

$$L = \sum_{i=0}^{\infty} T^i L^e$$

  as a solution for rendering problem
- Intractable equation with infinite series of integrals is replaced by integrals with successively higher dimensionality (T^i has i integrals)
- Gathering expansion (Rec. subst. RE): Corresponds to recursion levels during ray-casting from eye
- Shooting expansion (Rec. sub. PE): Corresponds to recursion levels during ray shooting from emitter

3.2.2 Integral Evaluation

- Integral evaluation
  - At each recursion level integral (high dimensional) is sampled
  - Numerical quadrature
    - Finite number of samples taken from integration domain
    - Integrand evaluated for samples
    - Numerical result of integral is weighted sum of evaluated samples
  - Classical Numerical Integration
    - Brick rule, Simpsons rule - simple and effective for low-dim integrands
    - Effort rises exponentially with dimension of integrand

    - Monto carlo only viable method

- Monte Carlo Quadrature
  - Converts calculation of integral to expected value problem.
  - Number of samples needed not dependent on dimension
  - Random sampling basis for determing result
  - Choosing samples
    - Two strategies
      - Importance sampling: Find best sample points by guessing distribution
      - Stratification: Using samples which cover integrant very evenly
    - Discrepancy measure for sampling quali
      - Regular grid high discrepancy
      - Real random numbers also high discrepancy
    - Quasi Monte Carlo (Stratification)
      - Deterministic low-discrepancy sequences
      - Halton sequences for arbitrary numbers of points
      - Hammersley sequences if number of needed points is known before
      - TMS nets for 2d integrands
  - QMC substantial performance and quality gains

3.2.3 Path termination

- Fractional propagation / Attenuation
  - Absorbed energy is subtracted at every step
  - Path terminated when energy is less than threshold
  - Biased but more intuitive
- Russian Roulette
  - Random termination of ray according to propagation probability
  - No bias, less intuitive

3.2.4 Expansion Advantages and Disadvantages

- Advantages:
  - Gathering expansion: No temp rep of complete radiance function required

- Shooting expansion: Flexible storage techniques
- Algorithms work on original geometry (no tesselation)
- Walks independent -can be parallelised.
- No temporary rep of the radiance function required.

• Disadvantages:

- Paths independet, no coherency between them
- Evaluation of very high dimensional integrals
  * Walk termination when biased threshold is reached or russian roulette
  * Reduces sampling of higher recursion levels (scenes with mirrors)

3.3 Iteration

• Solution of RE is fixed point of series

$$L_n = L^e + TL_{n-1}$$

• If T contractive, series converges from any initial distribution L_0

• Finite element techniques used, with discretisation error

3.3.1 Iteration Advantages and Disadvantages

• Advantages

- Coherence exploited well
- Approximating functions L_n viewpoint-independent: potential advantage for animations
- Provides implicit smoothing through discretisation, i.e. more visually pleasing images than noisy expansion
- More robust for highly reflective environments

• Disadvantages

- Requires object tesselation and finite element rep
  * Geometric accuracy and coherence lost
  * Substantial storage requirements also for moderately complex scenes
- Accuracy of high frequency shadows, reflections, caustics problematical
- Solution computed even for invisible parts of scene

4 Expansion Algorithms

- Random Walk Algorithms Overview
  - Gathering type RW
  - Shooting type RW
  - Bi-directional algorithms
  - Global methods
- Heckbert's Taxonomy
  - Provides further information
  - Used to categorize rendering algorithms
  - Short names
    - E ... eye
    - L ... lightsource
    - D ... non-ideal reflection or refraction
    - S ... ideal relfection or refraction
    - * ... iteration
    - [] ... optional
    - | ... selection

4.1 Gathering Type Random Walk

- Starts at eye, gathers emission of the visited points
- Differences in trace() function determine actual algorithm
- Pseudo code illustrated in algorithm 1

Algorithm 1 Gathering Type Random Walk

```
for all pixels do
  colour = 0
  for i = 1 → N do
    ray = random ray through pixel
    sampleColour = c * trace(ray)
    colour += sampleColour / N
  end for
end for
```

- Gathering Type Algorithms
  - Ray Casting (LDE): Intersect single ray with a scene

- Ray Tracing: $L[D]S^*E$
  - Photorealistic rendering algorithm
  - Raytracers and more sophisticated renderes use raycasters
- Distribution raytracing: $L[D|S]^*E$
- Path tracing: $L[D|S]^*E$

4.1.1 Ray Casting

- First hit raytracing or nonrecursive raytracing
- Possible as real time renderer
- Potentially more efficient than OpenGL for complex scenes (> 10M polygons)
- Pseudo code illustrated in algorithm 2

Algorithm 2 Ray Casting Trace Function

```
hit = FirstIntersect(ray)
if no intersection then
  return bgColour
else
  return emission(@hit,-ray.dir) + directLighting(@hit,-ray.dir)
end if
```

- Realtime RT
  - Based on ray casting
  - Certain limited types of recursion possible
  - GI also possible
  - State of the art: 1 billion polygons 640x480

4.1.2 Ray Tracing

- Classical raytracing (CG1), known as Whitted Raytracing (Turner Whitted)
- Hybrid algorithm
  - Recursion is evaluated for perfect mirrors
  - Coupling is ignored otherwise
- Steps
  - Ray through pixel: Visibility calculation, Object intersection

  - Shading based on direction to lightsource
  - Multiple lightsources are taken into account
  - Determining shadows
  - Handling of reflections
  - Shading of reflections
  - Sum of influences

- Pseudo code illustrated in algorithm 3

Algorithm 3 Ray Tracing Trace Function

```
hit = firstIntersect(ray)
if no intersection then
  return bgColour
end if
colour = emission(@hit, -ray.dir) + directLighting(@hit, -ray.dir)
if kr > 0 then
  colour += kr * trace(reflectedRay)
end if
if kt > 0 then
  colour += kt * trace(transmittedRay)
end if
return colour
```

4.1.3 Distribution RayTracing

- Distribution raytracing: $L[D|S]^*E$
- Impractical due to fan out of N at every level
- Very slow convergence
- Pseudo Code: Pdf page 12 slide 3
- Pros and Cons
  - + Simple
  - + Process eventually converges to true solution
  - - Bad convergence
- Other algorithms are used instead of this one

4.1.4 Path Tracing

- Path tracing: $L[D|S]^*E$
  - Single path traced through scene
  - New ray is generated within hemisphere on diffuse object intersection
  - With or without intermediate light source evaluation exist
- Simple path tracing (Pseudo Code: Pdf page 13 slide 2)
  - + Fast
  - + Simple to code: Follow path untillight source or threshold
  - - Does not work for small lightsources
- Improved Path Tracing
  - Sampling of the light sources: Multiple importance sampling
  - Key problem: correct weighting of two samples
- Path tracing: Sample weighting
  - At each surface intersection there are two possibilities to continue the ray
    - According to the BRDF
    - Through sampling of lightsources
  - Both have merits depending on situation
  - Problem: Choosing right one requires knowledge of the solution
  - Simple averaging: Retains worst properties of both
  - Both samples without scaling yields wrong results
  - No simple solution: Heuristic base on relative sample probabilities
  - The Balance Heuristic
    - Only used when both rays hit same light source, all other cases are simply added
    - Key idea: Weight each sample according to its relative probability
    - Weights sum up to one, no energy is counted twice
    - Each ray has two probabilities
      - Brdf probability (how probable that this ray gets created, e.g. Lambertian surface, phong-type specular lobe, mirror)

      · Lightsource sample probability (how probable that ray gets created as a light source sample, e.g. area light source probability)

- Path Tracing: Pros and Cons
  - + Simple
  - + Converges to true solution
  - + Better convergence than distribution RT
  - - bad converge for arbitrary scenes, especially when the light-sources are small

4.2 Shooting Type Random Walk

- Starts at lightsources, spreads the emission to visited points, which are projected into image
- Differences in shoot() function determine algorithm
- Pseudo code illustrated in algorithm 4

Algorithm 4 Shooting Type Random Walk

```
clearImage
for i = 0 ← N do
  ray = random ray from light with selection probablity p_e
  power = L_e * cos(φ) / (P_e * N)
  shoot(ray, power)
end for
```

4.2.1 Forward Raytracing - Photon Tracing

- LS^*DE
- Inverse of raytracing
- Unuseable convergence speed
- Limited to RT-type images
- Pseudo Code on page 17 slide 5

4.2.2 Light Tracing - Forward Path Tracing

- $L[D|S]^*E$
- Inverse of path tracing
- High variance
- Converges to true solution
- Unuseable in practice
- Pseudo Code on page 18 slide 1

4.2.3 Gathering vs Shooting Algorithms

- Dual algorithms, solve same problem
- Performance depends on factors
  - Image size vs scene size
  - Surface types
  - Light sources

4.3 Bidirectional Random Walk

- Overcome difficulties of gathering and shooting by combining them
- Two algorithms exist
  - Bidirectional path tracing
  - Metroplois light transport

4.3.1 Bidirectional Path Tracing

- Similar to path or light tracing, except
  - Two paths randomly cast, from the eye, and the lightsources
  - Convert shooting type walk to gathering type walk, the radiance ise multiplied by a facter (see pdf)
- One version, all mutual inter-connections evaluated, in the other just last one
- Both paths are stopped when below importance threshold, or russian roulette is applied

4.3.2 Metropolis Light Transport

- Bidirectional tracing until a useful path found (costly and rare)
- Attempt to change path a little bit, to find another usefull path
- Problem: not breaking the stochastic simulation - metropolis sampling
- Changing a path
  - Bidirectional Mutations: Changes in path length - vertices are added or deleted
  - Perturbations: Directions are slightly altered at points in path
- Pros and Cons
  - + Handles difficult situations well
  - - Startup bias not optimal for normal scenes
  - - very difficult to implement

5 Iterative Algorithms

5.1 Idea, Advantages and Disadvantages

- Idea
  - Solution to RE is a fixed point of

$$L_n = L^e + TL_{n-1}$$

  - If T contractive, converges from any initial distribution L_0
  - Finite element techniques (discretisation error) used for storing intermediate approximating functions L_n
  - Viewpoint independent, also more computations involved
- Advantages
  - Coherency can be better exploited
  - Approximating functions L_n viewpoint independent: advantage for animations
  - Implicit smoothing due discretization, i.e. better than noisy expansion
  - More robust for highly reflective environments
- Disadvantages

- Requires object tesselation and finite element representation
  - Geometric accuracy and coherency is lost
  - High storage requirements even for moderately complex scenes
- Accuracy of high frequency shadows, reflections and caustics problematical
- Solution also for occluded parts

5.2 Storage-Based Methods Overview

- Classical Radiosity - Iteration
  - Deterministic
  - Form-factor based
  - Discretization of scene into patches
- Stochastic Approaches - Expansion
  - Global lines
  - Photon tracing

5.3 Form-Factor Radiosity

5.3.1 Radiosity Equation

- Assume perfectly diffuse scattering
- Radiosity B is engery per unit area leaving patch surface per discrete time interval and is the combi
- Derive Discrete Radiosity Equation from RE (surface patches - finite element mehtod)

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij}, 1 \leq i \leq n$$

- B_i: Radiosity Patch i (Sum of emission und reflection per surface area)
- E_i: Emission patch i
- ρ_i: Reflection patch i
- n: Num patches
- F_{ij}: Amount of energy from patch j to i (form factor)

5.3.2 Algorithm

- Surfaces divided up into patches
- View factor (form factor) computed for each pair of patches
- Factors describe visibility between patches
- Small view factors - patches are far away or at oblique angle (zero if occluded)
- Factors used as coefficients in linearized form of RE - yields a linear system of equations
- Solving equations yields radiosity (brightness) of patches with diffuse interreflections and soft shadows
- Progressive radiosity solves system iteratively, after each iteration tmp radiosity values for patches (bounce levels)
- PR usefull for getting interactive preview
- No numerical converging needed - stop iteration when looking good
- Compute matrix solution at once too expensive: Jacobi-Iteration and Gauss-Seidel method (Gathering Radiosity), Progressive-Refinement and Southwell-Iteration (Shooting Radiosity)
- Shooting solutions converge better

5.3.3 Form Factor Calculation

- Nusselt
  - Point of receiving surface patch is center of sphere
  - Project other surface on sphere
  - Project projected surface on sphere on plane
  - Approximates form factor
- Hemicube
  - Cube with grid
  - Each grid point on cube has delta formfactor depending on position of grid plane
  - Surface patch projected on hemi cube
  - Delta form factors are summed up on projected areas: form factor relationship

- Binary space partitioning tree (BSP-tree): better performance for determining hidden patches
- Adaptive integration: newer methods

5.3.4 Limitations and Scope

- Only polygonal scenes
- Only diffuse materials
- Extensions for mirrors possible
- Newer, stochastic methods are state of research
- Form factor radiosity is state of the art in commercial products

5.3.5 Progressive-Refinement-Radiosity

- Shoot radiosity from all patches
- First take patches with high radiosity then lower radiosity

5.3.6 Hierarchical Radiosity

- Patchhierarchie: Bigger patches contain many smaller patches
- Radiosity on different levels: If error small stay on higher level, else lower level
- Cluster enhancement: bigger hierarchie level

5.4 Global Lines

- Stochastic radiosity solver
- Cast rays through scene, record all light transport along intersections
- Difusse environments only
- Also tesselated geometry
- Sometimes faster than photon tracing
- Good for animation sequences
- Theoretical better than form-factor radiosity but not used in practice (research only)

5.5 Photon Mapping

- Multipass renderer (usually 2-pass)
- Photon paths traced through scene and intersection point and incoming dir stored in cache (photon map)
- Usually different photon maps (Global, Caustics, etc..)
- Photon map usually arranged (kD-tree) for optimal k-nearest neighbor algorithm (photon look-up time depends on spatial distri)
- Caustic Photon Map
  - Many photons are emitted towards specular objects
  - Stored upon intersection with diffuse surfaces
  - Visualize directly by nearest n photons for illumination reconstruction
- Global Photon Map
  - Photons are emitted towards all objects in a scene
  - Used as a rough approximation of light transport in a scene
  - Not visualized directly
- Shadow Photon Map
  - Rays with origin at lightsource traced through entire scene
  - First intersection recorded as light, subsequent as shadow
  - Used for improvement raytracing pass
- Photon Maps Disadvantages
  - Slow
  - Memory consumption
  - Illumination reconstruction depends on distance
  - Biased (Averaging many renders does not converge to correct solution of RE)
- Lightmaps
  - 2d light textures on objects
  - Each texture element averages the energy of all photon hits it receives
  - Higher order representations possible

- Area of texels has to be known and has to be computed as a preprocessing step
  - Interpolation over texels after tracing pass

- Photon Tracing
  - Memory consumption: texels on all primitives are wasteful
  - Preprocessing: area computation for large numbers of texels takes too long
  - Execution time: far too many photons have to be cast for a stable estimate
  - Impossible to attach to implicit objects, e.h. L-systems

Part II
Advanced Topics

6 Surface Models

6.1 Light Matter Interaction

- Surface reflectance properties determine look
- Surface - light interaction
  - Key term for surface reflectance $\rho(x, x', x'')$
  - How light from a given direction is modified upon reflection from a surface
  - Has to be answereable for all directions and surface points

6.2 Bi-Directional Reflectance Distribution Function (BRDF)

6.2.1 Definition

- Returns ratio of reflected radiance along outgoing dir to irradiance incident on surface from ingoing dir
- $f_r(w_i \to w_r) \equiv \frac{L_r(w_r)}{L_i(\omega_i)cos\theta_i d\omega_i}$

6.2.2 Related Distribution Functions

- BTDF (Bi-directional Transmission Distribution) for rays that point into material
- BSDF (Bi-directional Scattering Distribution) combination of both, but rarely used

- BTF (Bi-directional Texture Function)
  - BRDF for entire texutres (carpet, cloth, ...)
  - Positional info to combine BRDFs
  - Problems: Seamless textures from BTF data, acquisition of samples

6.2.3 Isotropy vs Anisotropy

- Isotropy
  - Rotational invariance
  - Holds for large number of surfaces
  - Reduces number of variables by one
  - No alignment needed
- Anisotropy: Reflectance properties change with rotation of the surface around normal vector

6.2.4 Requirements

- Reasonable amounts of storage
- Capture key characteristics of surface
- Fast easy sampling by Monte Carlo
  - Apart perfectly diffuse andperfect mirrors, other reflection properties only tractable through MC rendering
  - Casting rays according to distribution function is crucial

6.2.5 Data Sources

- Tabulated measurements or simulation results
  - Gonioreflectometer: Expensive, hard to maintain and operate, generates huge data.
  - High memory when stored as set of finely spaced samples
  - Compression essential
  - Hard, time-consuming to measuring
  - Bad stochastic sampling characteristics: Rejection sampling
    * Given hard to sample probability distri other easy to sample envelope distri
    * Samples from envelope distri are accepted or rejected

        * Many unwanted samples possible

- Approximation by analytical functions and requirements
    - Reciprocity: Sampling directions can be interchanged (Helmholtz reciprocity principle)

$$f_r(\omega_i \to \omega_r) \equiv f_r(\omega_r \to \omega_i)$$

    - Energy conservation
    - Fast evaluation
    - Expressivity
    - Easy stochastic sampling

6.3 Analytical BRDFs

- Empirical models
    - Lambert, Phong, Blinn, Lafortune
    - Superposition of different components
- Physically based models
    - Torrance-Sparrow, Cook-Torrance, Kajiya, He-Sillion-Torrance-Greenberg (HTSG)
    - Physical material constants needed

6.3.1 Overview

- Lambert: only diffuse component
- Phong: generalized cosine lobe
- Ward: anisotropic
- Can be combined for higher realism
- Energy conservation dependent on coefficients and combination (esp. for Phong)
- Easy to sample
- Generalizations possible

6.3.2 Perfectly Diffuse

- Reflect the incoming light equally in all directions over the hemisphere
- Viewing direction independent
- E.g. Lambert, Oren-Nayar
- Lambert Surface
  - Percectly diffuse isotropic surface
  - Light proportional to cosine of incident angle
  - Color defined by wavelength-dependent diffuse absorption coefficient
- Oren-Nayar Surface
  - Microfacet-based diffuse surface
  - Takes into account masking, shadowing and interreflections
  - Often gaussian distri used to model distri of micro-facets (cavities)
  - Variance (sigma) of gaussian distri measure for roughness
  - Simplifies to Lambert's cosine model when sigma is zero
  - More and more retro-reflective for increasing sigma

6.3.3 Rough Specular

- Some light is reflected off ideal specular angle (highlights)
- Phong Surface
  - Visual approximation by ambient, diffuse and specular term
  - Adds specular highlight (parametrized through exponent)
  - Also combined phong lobes possible
- Ward's BRDF Model
  - Anisotropic micro-faceted surfaces
  - Physically correct (energy conserving) and fast
  - Again gaussian distri for cavities
  - Based on real measurements with a gonio-reflectometer

6.3.4 Perfectly Specular

- One outgoing dir (reality not existent)
- Incoming angle   outgoing angle
- Simulate smooth glass, metallic surfaces
- For realistic materials: Fresnel coefficients

6.3.5 Directional Diffuse

- Combination of rough specular reflector and ideal diffuse reflector
- E.g. Cook-Torrance, Ward, He,...
- Torrance-Sparrow Surface
  - Physically plausible BRDF model three main components
    - Microfacet model
    - Fresnel term for reflectance
    - Roughness term
  - Requires material constants to be known
  - Geometric Factor: Facets perfect reflectors except attenuation (self-shadowing, masking)
  - Evaulation
    - + Physically corrent
    - + Excellent results
    - - Hard to sample
    - - Hard to code
    - - Depends on material constants
- He, Torrance, Sillion, Greenberg
  - Based on wave optics and diffraction theory, can take polarization into account
  - Additional split between diffuse and directional diffuse term
  - Expensive to compute
  - Input: auto-correlation, variance of surface height

7 Advanced Materials

7.1 Beyond Normal BRDFs

- Some surfaces not characterized through standard BRDFs
- Phosphorescent paint
- Fluorescent paint
- Metallic paint
- Pearlescent paint
- Complex structure
  - Fibers (e.g. hair, textiles)
  - Sparkles (e.h. snow, lacquer)
  - Thin layers (e.g. leaves, skin)
- BSSRDFs

7.2 Fluorescence

- Re-radiation of incident energy at different wavelengths
- If only re-radiation to lower energy levels
  - Extends reflection spectra to matrices
  - Hard to handle otherwise
- Common effect, hard to measure: bispectral photometers needed
- In some areas the reflection intensity appears to be larger than 1
- Bi-Coloured Reflection Pattern
  - Rays reflected by the substrate retain color
  - Rays which interact with colorant molecules undergo wavelength shift
  - First Approximation: Phong lobes
    * Superposition of different phong lobes
    * Fixed ratio between lobes
    * Large diffuse fluorescent component
    * Small specular, non-fluorescent part
    * Advantage: simple
    * Disadvantage: results are not good

- Layered Torrance-Sparrow Model
  - Rough dielectric layer over lambertian fluorescent surface
  - Blinn microfacet distribution
  - No attenuation in the substrate
  - Simplified sub-surface scattering, re-emission at the point of incidence

7.3 Fibers

- Many materials composed of bundles of thin fiber
- E.g. hair, textiles, finished wood
- Hair
  - Not possible to model as a volume
  - Strands modelled as cylinders
  - Three light paths (R Reflect, T Transmission: R, TT, TRT)
  - Gaussian distribution (roughness) times attenuation

7.4 Asterism and Chatoyance

- Reflected light forms luminous band (star)
- Caused by small needle-like inclusions
- Simulated with a phong-like model

7.5 Heterogeneous Gemstones

- Appearance depends on viewing direction
- Often heterogenous
- Combining several textures

7.6 Sparkling Effects

- E.g. snow, metallic paint, gemstones
- Small flecks of material has a high specular reflectance
- Two options: Statistically, Explicitly
- Metallic Paint
  - Statistical model

  - Substrate: Lambert reflector
  - Flakes are modelled with a distribution
  - Top: clear coat (Fresnel reflectance)
  - Explicitly modelled flakes
    * Bigger flakes can be modelled explicitly
    * E.g. BTFs or voronoi textures

7.7 Pearlescent Paint

- Interference effect
- Aluminium or mica flakes coated with thin layers

7.8 Subsurface Scattering

- BRDFs: light scatters exactly at point where it hits the surface
- Subsurface scattering: light enters material, bounces around and leaves at different place
- Extension of BRDFs
- BSSRDFs: Bidirectional surface scattering reflectance distribution function
- 8 degrees of freedom (pos and direction)
- Monte Carlo evaluation expensive
- Diffusion Approximation
  - Light distri in highly scattering media becomes isotropic
  - Two point sources are placed near surface
  - Complete BSSRDF sum of diffusion approximation and single scattering term
- Rendering BSSRDFs
  - BSSRDF model applies to semi infinite homogeneous media
  - For practical model consider
    * Efficient integration of BSSRDF
    * Single scattering evaluation for arbitrary geometry
    * Diffusion approximation for arbitrary geometry
    * Texture (spatial variation)
  - Efficient Rendering (2 pass approach): first irradiance at surface points, then evaluate BSSRDF

8 Spectral Rendering

8.1 Introduction

- Basic Properties
  - Light as electromagnetic radiation out of particular region of entire spectrum
  - Distinguishing criterion: its frequency
- Spectrum
  - Ray of light contains many different waves with individual frequencies
  - Distribution of wavelength intensities per wavelength referred as spectrum of ray or lightsource

8.2 Colour Space Rendering

- Conventional
- Mostly RGB space, also CIE XYZ space
- RGB (Tristimulus) Rendering
  - RGB (3 wavelengths) define light and material props
  - Process 3 channels sep
  - Device dependent
  - RGB rep not ideal
- CIE XYZ
  - Derived from RGB
  - Outside the human visual gamut (only positive XYZ values)
  - Valid colors subspace of first octant
  - XYZ not closed under multiplication
- CIE XYZ vs RGB
  - RGB closed under multiplicaiton
  - RGB negative values to rep all colors
  - RGB corresponds to real colour
  - XYZ virtual positive values

8.3 Spectral Rendering

- Few products available: Maxwell renderer, luxrender, indigo
- Few known about internal details
- Rendering steps
  - Get spec
  - Prepare spec
  - Process spectral samples sep throughout rendering calculation
  - Compute final display color using CIE color matching functions and standard transformations

8.3.1 Measuring Spectra

- Measured with spectroradiometer
- First calibration of measurement device
- Reference standard needed
  - Source of known emissivity (blackbody, refernece lamps)
  - Detector with an exactly known response
  - Surface with exactly known reflectivity
- Reference Lamps
  - Desinged lamps
  - Repeatability of lamp manufacturing good enough to duplicate lamps
  - Tungsten halogen lamps of 1000, 200 and 45W developed for general use
  - Burning time limited
- Detectors: pdf...

8.3.2 Preparation

- Light as freq distri (usually smooth, sharp peaks - fluorescense, spectral colours)
- Colour collections: munsel, ncs, ral...
- Discritize distris
  - Regular sampling: aliasing, fast convolution

  - Linear or higher order rep: efficient storage, slow convolution
  - Hybrids: slow but more efficient storage
  - Amount of samples depends on spectrum
    * More samples for spectra with sharp peaks
    * CIE F11 representations
    * error map sampling points

8.4 Spectral vs RGB Rendering

- RGB
  - Fast, widely supported
  - Limited accuracy (sharp spectra, different illuminations)
- Spectral Rendering
  - Accuracy, prediction of nature
  - High cost, aliasing, data mixing, input data
  - Hard to code

8.5 Spectral Effects

- Metamerism (different spectra - same color, problem for paint and pigment industry)
- Volume absorption
- Dispersion (prisms, rainbows): wavelength dependency of interference and refraction
- Interference and diffraction
- Fluorescent materials and light sources
- Polarisation: essential for predictive rendering or crystals and transparent objects or outdoor scenes
  - Specular surfaces governed by fresnel terms
  - Show discrepancy for diff orientiations of polarised incoming light (.e.g. skylight)
  - Spec scenes with water, glass, car roots etc are affected
  - Describing polarisation see pdf

9 Participating Media

9.1 Introduction

- In vaccum, radiance is constant along the ray
- Real-world light scattered and attenuated (e.g. fog, smoke, ..)
- Two difficulties
  - Intersection phenomena within any point of the medium
  - Spectral dependence of medium characteristic parameters
- Radiation three different kinds of phenomena
  - Absorbtion
    * Energy is reduced (converted into e.g. heat)
    * Given by absorption coefficient
    * Beer's law: see pdf
  - Emission
    * Energy added from luminous particles and converted to visible light
    * Chemical, thermal or nuclear processes
  - Scattering
    * Out-scattering: Radiance reduced along ray (e.g. clouds)
    * In-scattering: Radiance increased from other dirs (e.g. mist)
- Phase Function: Spatial distri of the scattered light
  - Isotropic (counterpart of diffuse BRDF)
  - Rayleigh (small spherical particles, e.g. smoke)
  - Mie (particles have size of light, e.h. clouds)
  - Henyey-Greenstein (approximation of Mie)

9.2 Transport Equation

- TE takes all these phenomena into account
- Describes variation of radiance
- Challenges
  - Input data (homogeneous - constant parameters, inhomogeneous - properties are varying in the medium)
    * Explicit storage of measured data (voxel grids)

        * Numerical solutions (simple analytical functions: perlin noise, exp func, fluid sim)
    - Solving the TE (Full solution expensive - simplified models)

- TE simplifications
    - No scattering case (e.g. fire)
    - Single scattering case (not realistic, strongly related to the medium)
        * Scattering of light by a single particle
        * Material is either very thin or very transparent
    - For homogeneous non-emitting materials - there is another formula
    - For heterogeneous materials break up integral and compute it incrementally by ray marching
        * Compute distribution from medium, by dividing the ray into smaller segments
        * Different Ray Sampling strategies
    - Absorption and emission only
    - Multiple Scattering
        * Scattering of light from multiple particles
        * Two Stages
        * Illumination pass (source radiance is computed, e.g. volume photon mapping)
        * Visualization pass (TE is solved, e.g. ray marching)

9.3 Volume Photon mapping

- Extend surface photon maps to volume photon maps
- Photons stored in volume
- Two pass algorithm: First trace photons through volume, then evaluate photon maps using ray marching
- Photon can pass unaffected or interact with medium (scattered or absorbed)
- Russion roulette decides photon is scattered or absorbed
- Stored in photon map, if not from light source and interacts
- Importance sampling of phase function to find new direction

9.3.1 Estimating Radiance

- Different for surfaces and volumes
- Direct light by sampling of the light source using ray marching
- Indirect light: volume radiance estimate

9.3.2 Analytical Models

- CIE: Monochrome (luminance only), validated to some degree
- Perez: improved CIE model
- Preetham: based on perez model, spectral colours for each solar elevation
  - Five parameters
  - A darkening or brightening of the horizon
  - B luminance gradient near the horizon
  - C relative intensity of the circum-solar region
  - D width of the circum-solar region
  - relative backscattered light

10 Cameras

10.1 Camera Models

- Perspective Camera (perspective projection)
  - Pinhole (Perspective) Camera
  - Simplest device for taking photos
  - Light enters through small hole and falls on film, hole is the eye point
  - Includes foreshortening
  - Doesn't perserve distances or parallel lines
- Orthographic Camera (parallel projection)
  - Perserves relative distance between objects, parallel lines
  - No foreshortening
  - View volume is an aligned box
- Fisheye Camera

- Environment (Spherical) Camera
  - Rays are traced in all directions around a point
  - 2d view of everything that is visible from that point
  - All rays have same origin
  - Useful for environment lighting

10.2 Pinhole Concept vs Aperture

- Sharp for all parts of a scene
- Pinhole is an idealized concept, but not realistic
- Real cameras need an aperture and a lens
- Aperture
  - Pinhole has to have nonzero diameter - reason for aperture
  - Cameras have variable with - which is determined by a mechanical iris
  - Lens is needed to create an image with aperture and iris
- Thin Lens Assumption
  - For sophisticated CG renderings with DOF etc more realistic concept of lens systems is needed than a pinhole camera
  - For most purposes, it is sufficient to assume a planar lens with negligible curvature and fixed index of refraction
  - Fat lenses have to be explicitly simulated
- Focal Length Definition
  - Parallel rays that fall through lens are focused in F
  - Small F .. wide angle lens
  - Large F .. tele lens
- Thin Lens Camera Implications
  - Possible to perfectl focus any plane source image onto the receiving film
  - It is not possible to focus all objects which are in different depths
  - Resolution of the film is always limited
- Depth of Field

  - Small aperture... image is sharper over a wider range, but longer exposure needed
  - Big aperture ... small exposure, low DOF

- f-number: focal length divided by aperture diameter

- Real Camera Lens Errors

  - Spherical Aberration, Astigmatism and Coma
  - Distortion affects the image geometry
  - Field curvature is due to lens curvature: local lack of sharpness
  - Chromatic aberration (occurs near edges)
  - Diffraction upper-bounds the imaging capabilities of a given lens

11 Introduction to Stereo Projection

- Physical properties of stereo setup

  - Linear vs circular polarization
  - 2 beamers with filters for circular polarized light
  - Retroreflective screen: Perserves polarization state

- Basics

  - Render scene twice: With eye offset
  - Also view matrix changed sometimes
    * Toe in: shift towards focus point
    * Offaxis: shit view matrix
    * Create asymmetric proj: No vertical parallax

12 Post-processing: Tone Reproduction and White Balance

12.1 Introduction

- Image Synthesis Pipeline

  - Modelling
  - Rendering
    * Output RGB, XYZ or spectral
    * Predictive rendering yields hdr images
  - Display has limited range for luminance and color

- HDR
  - Dynamic range: contrast ratio between brightest and darkest parts
  - Hdr cannot be displayed on normal display hw
  - Special iamge formats necessary
  - Usually impossible to solve repro task perfectly
  - Strongly depnds on output device
  - Various heuristics of increasing complexity exist
  - Full perception models difficult
  - Animations pose additional challenges (frame to frame coherency)
- Image Types
  - Relative values: Measured as max output device capability
    - Screens: Two orders of magnitude
    - Printouts: Range of 10 luminance units
    - 8bit images: 256 steps
  - Absolute Radiometric Values
    - Captures of reality: Scene ref images
    - Digital cameras ought to capture

12.2 Image Formats

12.2.1 Conventional formats

- RGB (TIFF, PNG, JPEG)
- TIFF also in CIE L*a*b*
- Normally 8 bits per channel
- TIFF: 16 bit possible (JPEG 12bit)
- Brightness ends at 1 - device dependent
- No physical meaning of values
- Compact size, standardised but lots of info lost

12.2.2 HDR formats

- Values have physical meaning
- Floating point components: Large range
- Compact size, standardised, few quantization errors, Compression introduce artefacts
- Not understood by photoshop
- Radiance RGBE, Pixar Log and LogLuv TIFF, ART XYZ (uncompressed), OpenEXR
- OpenEXZ HDR Image Format
  - Tailored to needs of movie industry
  - 8bits unsuitable for movie work, 16bit limited post-processing
  - 16 and 32 bit floating point colors
  - 16 bit float compatible with Nvidia CG HALF (EXR directly used in hardware)
  - Lossless compression: 35%-55%
  - Arbitrary info can be stored alongside image data
  - Arbitrary image channels

12.2.3 Spectral Image Formats

- N spectral samples per pixel
- Floating point components: Large range
- Values have physical meaning
- FITS and ARTRAW lonely formats
- No quantization or compression errors, no info lost but huge files (400mb for 640x480) and rare support

12.3 Image Post-Processing

- Gamut Mapping: Getting colours into display gamut
  - Local: Outlying points are individually moved (fast but highlights lost)
  - Global: All points analysed, point cloud shrunk so it fits into gamut
    - Relation between colours is maintained

        * Desaturation of img

- Tone Mapping: Fitting the luminance range to a given device
- Tone Rep: Gamut and tone mapping together

12.3.1 Tone Reproduction Operators

- 3 diff approaches
- Global methods: spatially uniform, linear scale factor, non-linear scale factor
    - Scalling all luminance values by factor
    - Primitive and fast
    - Automatic determination of factor
    - Sufficient for many scenes
    - Result in dark images if the DR big
    - Linear operators: same fac for all values
        * Mean value mapping
            · Mean value of the hist is mapped to 0.5
            · Values outside the constrast interval are clipped
        * Interactive calib: Interactively define area and range of available contrast interval
        * Ward's scaling fac
            · MAx display luminance and environment adaption degree parametrization
            · Good results: just visible differences remain
            · Image has to be given in absolute units
    - Non-linear operators
        * Exponential Mapping
            · Corresponds to human perception
            · Reduces overproportional influence of few bright pixels
        * Schlick's Method
            · Behaviour similar to exponential mapping
            · Good for high contrast images
            · Can fail completely
        * Mapping by Tumblin and Rushmeier
        * Visual Adaption Model
- Local methods: spatially non-uniform

  - Differences between parts of the image are taken into account
  - Image sep into zones to determine brightness targets (similar to photography)
  - Local kernel of var size used for final tone rep step
  - Can look artificial

- Perceptual approaches
  - Results from physiology and psychology used to reproduce behaviour of human vis sys
  - 2 approaches
    * Determiens what person would see if scene was real
    * Try to rep sensation using on display
  - Takes into account
    * Threshold sensitivity
    * Color appearance
    * Visual acuity
    * Light adaption
    * Dark adaption

- Colour Correction / White balance
  - Challenging task
  - Most algorithms image based, only two scene driven
  - Workflow
    * Determining illuminant colour
    * Applying transform that compensates for the illuminant
  - Many algorithms:
    * Gray world
      · Avg of all pixels is gray
      · Avg is mapped to gray
      · Fails if assumption is violated
    * White patch
      · Always a white obj in the img,
      · Brightest pixel is mapped to white,
      · Fails if no white obj
    * Neural networks
  - CC State of the art
    * Scene based (better approach)

    - * Reliable CC method that used additional info (gathered during rendering)
  - Algorithm Overview
    - * All directly viewed surfaces set to neutral
    - * All lights set to neutral on directly viewed surfaces

13 Pixar RenderMan

13.1 Introduction

- REYES architecture
  - Render everything you ever saw
  - PRMan ref implementation
  - Assumptions and Goals
    - * High possible model complexity
    - * Diverse primitives: fractals, procedural models etc
    - * Shading complexity
      - · Complexity of scenes more comes from surface specs than geometry
      - · Programmable shaders needed
    - * Minimal ray tracing: Approximation of non-local effects through other means (e.g. shadow maps)
    - * Speed: 2h movie in 1 year
    - * Image quality: Anti-aliasing and proper pixel filtering needed
- RenderMan: Used for today's industrial CG work
- RenderMan Naming Confusion
  - RenderMan SL: 3d scene description language
  - RenderMan Interface: Interface between modelling and rendering
  - PRMan: The RenderMan-compliant hybrid scanline renderer
- For long time Pixar PRMan was the only Rman-compliant system
- Pixar Photorealistic RenderMan (PRMan)
  - Evolved since 1982/84 from Lucasfilm Renderer
  - Sophisticated scanline renderer
  - Currently at release 14.0
    - * Indirect illumination / GI
    - * Hair and fur optimizations
    - * Parallel network rendering
    - * On demand raytracing

13.2 REYES Desing Principles

- Bound: Computes bounding box
- Culling: Discard inv primitives
- Diceable test
  - Examines micropolygon size and number
    - All primitives are diced into micropolygons
    - Shading process operates on MPs
    - MP generation operates in eye space
    - Subdivision done in the primitives (u,v) space
  - Split: Subdivision into other geometric primitives
  - Dice: Perform the actual split into micropolygons
  - Backdoor feature intended to incorporate raytracing
  - Sampling
    - Micropolygons are nyquist limit for their pixels
    - Jittered samples trade aliasing artifacts for noise
    - Sampling can be influenced
  - Reconstruction functions are user choice: RiFilter

13.3 REYES Advantages and Disadvantages

- Advantages
  - Can handle arbitrary number of primitves
  - No inversions - projections of pixel onto textures
  - Computations can easily be vectorised (e.g. shading)
  - No clipping calculations
  - Frequently no texture filtering is needed
- REYES Disadvantages
  - No natural way to dice polygons
  - Shading before sampling causes problems for motion blur
  - Dicing is difficult for some types of primitives (like e.g. blobs)
  - No coherency for large uniform objects, everything is diced into micropolygons
  - No GI info of any kind is computed

13.4 RenderMan Interface

- Interface between rendering and modelling
- Powerful set of primitive surface types: quadric surf, polygons, parametric surfaces
- Hierarchical modeling, geometry
- Constructive solid geometry
- Camera model (orthographic, perspective)
- Generalized shading model

13.4.1 Using the RM Interface

- Two basic options exist
  - Use of RM function calls from high-level language (e.g. C) implementation of the RM API
  - Feeding archived RM function calls from a RenderMan Interface Bytestream (RIB) to a compliant renderer (Hand generated, output from modelling program)
- Acutal renderers are usually non-interactive
- Separate preview renderes are used during the design phase

13.4.2 RM Programm Structure

- Consistent naming of API calls (Ri..)
- All function calls bracketed between one pair of RiBegin and RiEnd
- One global graphics state is maintained within this bracket
- All API calls modify this state
- API calls are frequently varargs, and have to be terminated with RI NULL
- Most calls deal with surface properties

13.4.3 RIB File

- Sequence of requests to the renderer
- No loops, branches, ...
- Hierarchical attributes, transformations
- Geometry, lights and materials are specified insidde a WorldBegin, WorldEnd block
- Normally RIB file contains just one frame's worth of data

13.5 Shading

13.5.1 Shape vs Shading

- Shape: Geometry of object
- Shading
  - Appearance of object in scene
  - Defined by: light and surface specs and pos, pos and orientation of object

13.5.2 Shading pipeline

- Three types of shaders
- Emission at the light source
- Interaction of light with surface
- Atmospheric effects between surface and viewpoint

13.5.3 Types of Shaders

- RenderMan Interface supports
  - Light source shaders
    * Calculates intensity and colour of light sent by the light source to a point on a surface
    * Describes lightsource
  - Surface shaders
    * Determines the colour of light reflecting from a point on a surface in a particular direction
    * Does not have to be physically plausible

- Volume shaders: Generalizes the idea of atmosphere affecting light passing through space between surface and eye
- Displacement shaders
  * Distort geometry of basic object
  * Difference to bump maps (surface shader): silhouette is correct
  * Costly to evaluate
- Imager shaders
  * Transform already computed colours to something else
  * Applications: e.g. cartoonish distortions of realistic renderings
  * Only colour information and z-Buffer data are provided
- Transformation Shaders
  * Similar to displacement shaders in that they modify object geometry resp. point coordinates
  * Difference: used at a different, earlier stage of the rendering pipeline
  * Used to transform entire objects
  * Restricted variable set

- Each shader has specific variables and result types
  - Floats
  - Colours: multi colour models, RGB default
  - Points
  - Strings
  - Uniform vs Varying variables
    * Uniform vars constant everywhere over area

- Shadows
  - Automatic shadow generation not classic RM
  - Shadow maps prepared for each lightsource
  - Surface shaders use this info
  - Raytracing and GI in newer RM obsolete this

- Deep Shadow Maps
  - Store rep of the fractional visibility through a pixel at all possible depths
  - Transmittance function describes light falloff

  - Stored as array of floating-point pairs
  - Handles volumentric effects and semi-transparent surfaces

- Reflections
  - Similar to depth images for the lights, reflections have to be precomputed
  - Reflection maps have artefacts: no multiple inter-reflections
  - Fast
  - (Sort of obsoleted by raytracing on demand)

- Reflection / Refraction Multipass Rendering
  - Multi-pass rendering
  - Tank shader: Each wall has unique pair of reflection and refraction camera
  - Texture is projected

- Ray Tracing On Demand
  - Scanline: fast, can handle complex scenes, shadows and reflections are problematic
  - Ray tracing can not deal with complex scenes
  - Ray differentials
  - First-level rays originate from REYES shading points

- Implementations
  - Original Pixar renderer (REYES): Micropolygon-based hybrid with raytracing capabilities
  - BMRT: Raytracer, has disappeared after legal action was taken against author
  - Pixie: Open-source RenderMan
  - Realtime techniques: Ongoing research topic

14 Unsolved Problems

- First suggested by Ivan Sutherland in 1965
- Become focus of future developments in 70/80s
- When is a problem solved? e.g. RE .. theoretically solved, but unsovled practical sense

- Cheap and fast solutions needed
- Sutherland 1966
  - Cheap machines with basic capability
  - Basic interaction techniques
  - Coupling simulation to their display
  - Describing motion
  - Continuous tone displays (haftoning)
  - Making structure of drawings explicit
  - Hidden line removal
  - Program instrumenation and visualization
  - Automatic placement of elements in network diagrams
  - Workin with abstractions (scientific visualization)
- Heckbert 1987
  - Converting implicit models to parametric
  - High-quality texture filtering
  - Antialiasing
  - Shadows without ray tracing
  - Practical ray tracing
  - Practical radiosity
  - Frame-to-frame coherence
  - Automating model culling
  - Smooth model transitions
  - Affordable real-time rendering hardware
- Jim Blinn's 10 Unsolved Problems (More sociological or marketing issues than technical problems)
  - Novely
    * Simply find something new
    * Easy problems have been solved
  - Education
    * Learning: Keeping up with what has been done (Related to problem 1, don't reinvent the wheel)
    * Teaching: Dissemination of new discoveries
    * System Integration: How to use all the tricks in one production

    * Simplicity: Make things simple
  - Better Pixel Arithmetic Theory
    * RGBA pixel concept incomplete
    * 3 probs
      · Pre-multiplication of colour channels by alpha channel (local / global distinction)
      · Correlated edges of foreground and background object
      · Combining compositing operations with light reflection models
    * Unified field theory of pixel arithmetic
  - Legacy compatibility
    * Technological improvements change trade-off
    * Legacy applications and data, e.g. 3d apis, file formats
    * How to no abandon the old while allowing the new
  - Arithmetic Sloppiness
    * Programmers tempted to do sloppy job of pixel arithmetic for speed
    * E.g. texture filtering: bilinear interpolation between four nearest texels (diamond shaped artifacts)
    * Phong BRDF model
    * How accurate do we need to be?
  - Antialiasing: Textures in perspective will be either too fuzzy or to jaggy
  - A Modeling, Rendering, Animation Challenge
    * 1 shape - piles of rope or string and even conceivably to protein folding
    * Modeling is figuring out the shape of it
    * Rendering is making picture of it
    * Animation is figuring out how it moves with time
  - Finding a use for real-time 3d
    * Find large-scale uses for it: Entertainment (movies, games), Engineering (CAD, CAM), Visualization
    * Interaction and communication (GUIs, printed media, art, ecommerce, web3d and virtual communities)

- Unsolved issues in PR
  - Acquisition and modeling BRDFs
  - Reliable, accurate and cheap

- Render and measurement time too long
- Measure more samples where more accuracy needed
- Acquisition of geometry and surface appearance
- Problematic for non-diffuse and transparent surfaces or when illuminant is unknown
- Self-adaptive light transport
- Some algorithms perform better in specific situations than others
- Adaptive overall global illumination algorithm picks mode depending on surface, illuminations,...
- Scalable and robust rendering
- Complex scenes without user intervention
- Especially important in interactive and dynamic applications, e.g. games
- Geometry-independent rendering
- Currently many ray-objects interection calculations
- What if geometry is not known explicitly (e.g. light field, photographs)
- Radiometric accuracy main dirven force
- Usually not necessary
- Psychoperceptual rendering: Perceptual correct images
- Viewer might still judge the image to be realistic
- Integration with real elements
- Put real objects in virtual scene and virtual object in real scene, e.g. with projectors
- Blend between real and virtual elements

• Ultimate Photorealistic Renderer
  - Interactivity
  - Any material, any geometry (pure specular to pure diffuse)
  - Many different input models
  - Realism slider

List of Algorithms