Rendering Transparent and Turbid Fluids

PhD Course on the Deformable Simplicial Complex Method

Jeppe Revall Frisvad

August 2010

To learn more about rendering, see
http://www.imm.dtu.dk/courses/02576/

Teaser: natural waters using wind-based wave generation

Recommended reference

- Premože, S., and Ashikhmin, M. Rendering natural waters. *Computer Graphics Forum 20*(4), pp. 189–199, 2001.

Teaser: direct-able breaking waves

Recommended reference

- Bredow, R., Schaub, D., Kramer, D., Hausman, M., Dimian, D., and Duguid, R. S. *Surf's Up: The Making of an Animated Documentary*, ACM SIGGRAPH 2007 Course Notes, Course 12, 2007.

Teaser: watch Surf's Up if you didn't ☺

Recommended reference

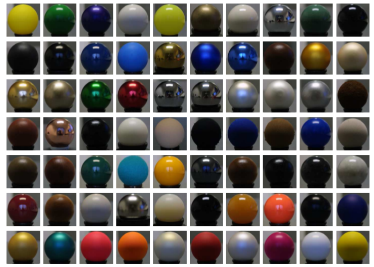
- Bredow, R., Schaub, D., Kramer, D., Hausman, M., Dimian, D., and Duguid, R. S. *Surf's Up: The Making of an Animated Documentary*, ACM SIGGRAPH 2007 Course Notes, Course 12, 2007.

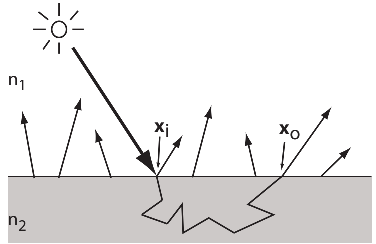
Models needed for physically based rendering

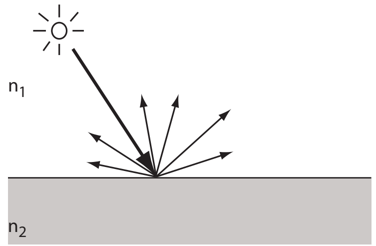
- Think of the experiment: "taking a picture".
- What do we need to model it?
  - Camera
  - Scene geometry
  - Light sources
  - Light propagation
  - Materials
- Mathematical models for these physical phenomena are required as a minimum in order to render an image.
- We can use very simple models, but, if we desire a high level of realism, more complicated models are required.
- To get started, we will go through the simpler models (in opposite order).

Materials (scattering and absorption of light)

- Optical properties (index of refraction, $n(\lambda) = n'(\lambda) + i\,n''(\lambda)$).
- Reflectance distribution functions, $f(\mathbf{x}_i, \mathbf{x}_o, \vec{\omega}_i, \vec{\omega}_o)$.

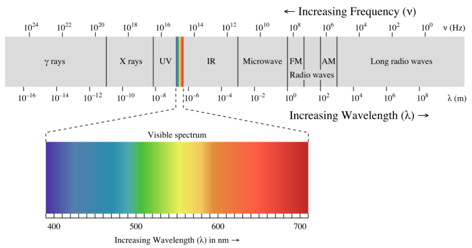






Light propagation

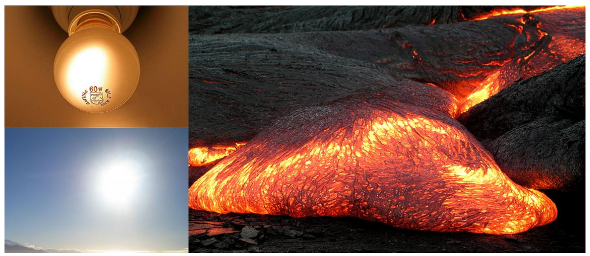
- Visible light is electromagnetic waves of wavelengths (λ) from 380 nm to 780 nm.



- Electromagnetic waves in transparent media propagate as *rays of light* for $\lambda \to 0$.
- Rays of light follow the path of least time (Fermat).
- How does light propagate in air? In straight lines (almost).
- The parametrisation of a straight line in 3D is therefore a good, simple model for light propagation.

Light sources

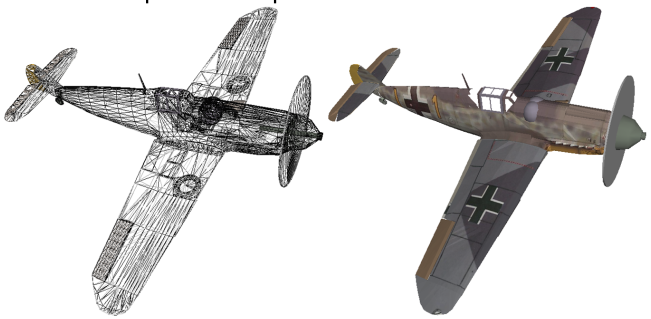
- A light source is described by a spectrum of light $L_{e,\lambda}(\mathbf{x}, \vec{\omega}_o)$ which is emitted from each point on the emissive object.
- A simple model is a light source that from each point emits the same amount of light in all directions and at all wavelengths, $L_{e,\lambda} = \text{const.}$
- The spectrum of heat-based light sources can be estimated using Planck's law of radiation. Examples:



- The surface geometry of light sources is modelled in the same way as other geometry in the scene.

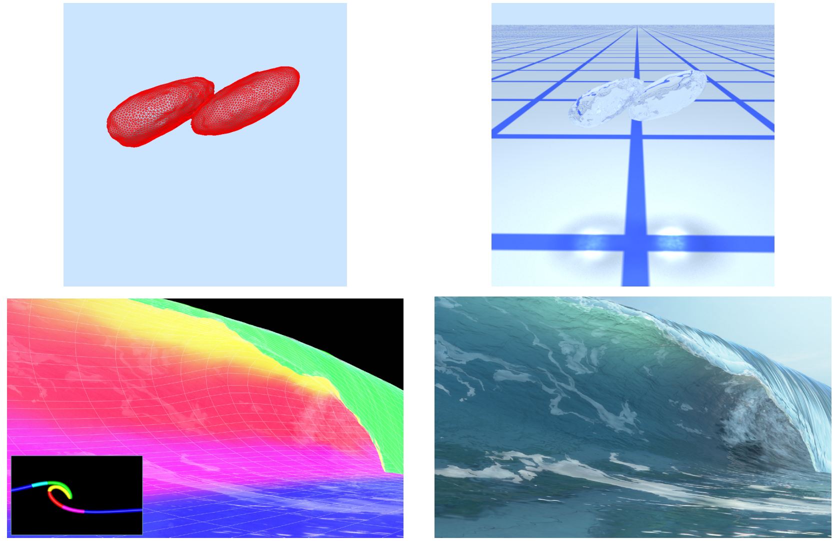
Scene geometry

- Surface geometry is often modelled by a collection of triangles some of which share edges (a triangle mesh).
- Triangles provide a discrete representation of an arbitrary surface. A quick example:



- Triangles are useful as they are defined by only three vertices. And ray-triangle intersection is simple.

Examples of fluid geometry

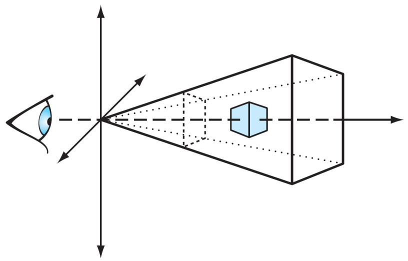


Reference for the two renderings in the second row

- Bredow, R., Schaub, D., Kramer, D., Hausman, M., Dimian, D., and Duguid, R. S. *Surf's Up: The Making of an Animated Documentary*, ACM SIGGRAPH 2007 Course Notes, Course 12, 2007.

Camera

- A camera consists of a light sensitive area, a processing unit, and a storage for saving the captured images.
- The simplest model of a camera is a rectangle, which models the light sensitive area (the chip/film), placed in front of an eye point where light is gathered.



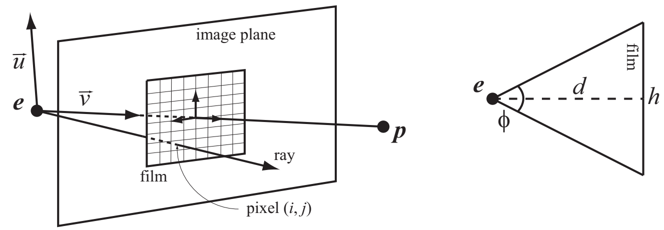
- We can use this model in two different ways:
  - Follow rays from the eye point through the rectangle and onwards (ray casting).
  - Project the geometry on the image plane and find the geometry that ends up in the rectangle (rasterization).

Ray casting

- Camera description:

Extrinsic parameters		Intrinsic parameters	
$\mathbf{e}$	Eye point	ϕ	Vertical field of view
$\mathbf{p}$	View point	d	Camera constant
$\vec{u}$	Up direction	W, H	Camera resolution

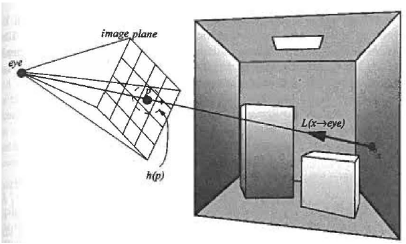
- Sketch of ray generation:

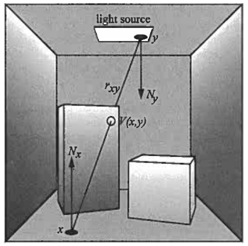


- For each ray, find the closest point where it intersects a triangle.

What is a ray?

- Parametrisation of a line: $\mathbf{x}(t) = \mathbf{e} + t\vec{r}$.
- Camera provides origin ($\mathbf{e}$) and direction ($\vec{r}$) of "eye rays".





- The user sets origin and direction when tracing rays recursively.
- But we need more properties:
  - Maximum distance ($\max t$) for visibility detection.
  - Information about what was hit and where (hit normal, position, distance, material, etc.).
  - A counter to tell us the trace depth: how many reflections or refractions the ray suffered from (no. of recursions).

Ray tracing

- What do you need in a ray tracer?
  - Camera (ray generation and lens effects)
  - Ray-object intersection (and accelleration)
  - Light distribution (different source types)
  - Visibility testing (for shadows)
  - Surface scattering (reflection models)
  - Recursive ray tracing (rays spawn new rays)
- How to use a ray tracer? Trace radiant energy.
- The energy travelling along a ray of direction $\vec{r} = -\vec{\omega}$ is measured in radiance (flux per projected area per solid angle).
- The outgoing radiance L_o at a surface point $\mathbf{x}$ is the sum of emitted radiance L_e and reflected radiance L_r:

$$L_o(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_r(\mathbf{x}, \vec{\omega}) \ .$$

- Reflected radiance is computed using the BRDF (f_r) and an estimate of incident radiance L_i at the surface point.

The rendering equation

- Surface scattering is defined in terms of
  - Radiance:
$$L = \frac{\mathrm{d}^2\Phi}{\cos\theta\,\mathrm{d}A\,\mathrm{d}\omega} \ .$$
  - Irradiance:
$$E = \frac{\mathrm{d}\Phi}{\mathrm{d}A} \ , \quad \mathrm{d}E = L \cos\theta\,\mathrm{d}\omega \ .$$
  - BRDF:
$$f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) = \frac{\mathrm{d}L_r(\mathbf{x}, \vec{\omega})}{\mathrm{d}E(\mathbf{x}, \vec{\omega}')} \ .$$
- The rendering equation then emerges from $L_o = L_e + L_r$:

$$L_o(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{2\pi} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega})\, L_i(\mathbf{x}, \vec{\omega})\, \cos\theta'\,\mathrm{d}\omega' \ .$$

- This is an integral equation.
- Integral equations are recursive in nature.

A Monte Carlo estimator for the rendering equation

- The rendering equation:

$$L_o(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{2\pi} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}') \cos\theta\,\mathrm{d}\omega' \ .$$

- The Monte Carlo estimator:

$$L_N(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \frac{1}{N}\sum_{i=1}^{N} \frac{f_r(\mathbf{x}, \vec{\omega}_i', \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}_i') \cos\theta}{\mathrm{pdf}(\vec{\omega}_i')} \ .$$

- The Lambertian (perfectly diffuse) BRDF:

$$f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) = \rho_d/\pi \ .$$

- A good choice of pdf would be:

$$\mathrm{pdf}(\vec{\omega}_i') = \cos\theta/\pi \ .$$

Cosine-weighted hemisphere sampling

- Sampling directions according to the distribution:

$$\mathrm{pdf}(\vec{\omega}'_i) = \cos\theta/\pi \ .$$

- In spherical coordinates: $\mathrm{pdf}(\theta, \phi) = \cos\theta \sin\theta/\pi$.
- With two uniform random variables $\xi_1, \xi_2 \in [0, 1]$, this pdf is sampled by

$$(\theta, \phi) = (\cos^{-1}\sqrt{\xi_1},\ 2\pi\xi_2) \ .$$

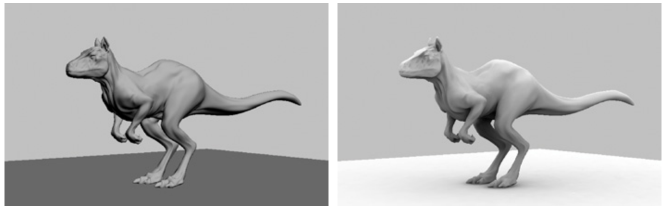
- In a local coordinate system, where the z-axis is the zenith direction, the corresponding direction vector is

$$(x, y, z) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \ .$$

- If we find an orthonormal basis $\{\vec{b}_1, \vec{b}_2, \vec{n}\}$ which includes the surface normal $\vec{n}$, we can rotate to world coordinates by a change of basis

$$\vec{\omega}'_i = x\vec{b}_1 + y\vec{b}_2 + z\vec{n} \ .$$

Ambient occlusion



- Using the Lambertian BRDF for materials, $f_r = \rho_d/\pi$; the cosine weighted hemisphere for sampling, $\mathrm{pdf}(\vec{\omega}'_i) = \cos\theta/\pi$; and a visibility term V for incident illumination, the Monte Carlo estimator for ambient occlusion is simply:

$$L_N(\mathbf{x}, \vec{\omega}) = \frac{1}{N}\sum_{i=1}^{N} \frac{f_r(\mathbf{x}, \vec{\omega}'_i, \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}'_i)\cos\theta}{\mathrm{pdf}(\vec{\omega}'_i)} = \rho_d(\mathbf{x})\frac{1}{N}\sum_{i=1}^{N} V(\vec{\omega}'_i) \ .$$

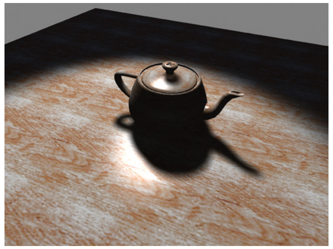
Sampling a triangle mesh

- Uniformly sample a triangle ($\mathrm{pdf} = 1/n$, where n is the number of triangle faces in the mesh).
- Uniformly sample a position on the triangle ($\mathrm{pdf} = 1/A_\Delta$, where A_Δ is the triangle area):
    1. Sample barycentric coordinates ($u, v, w = 1 - u - v$):

$$\begin{aligned} u &= 1 - \sqrt{\xi_1} \\ v &= (1 - \xi_2)\sqrt{\xi_1} \\ w &= \xi_2\sqrt{\xi_1} \end{aligned}$$

    2. Use the barycentric coordinates for trilinear interpolation of triangle vertices to obtain a point on the triangle.
    3. Use the barycentric coordinates for trilinear interpolation of triangle vertex normals to obtain the normal at the sampled surface point.

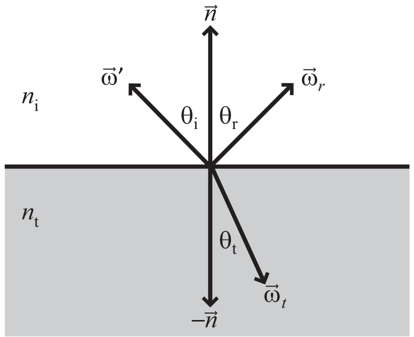
Soft shadows



- From solid angle to area: $\mathrm{d}\omega' = \frac{\cos\theta_{\text{light}}}{r^2}\mathrm{d}A$.
- Using the Lambertian BRDF, $f_r = \rho_d/\pi$, and triangle mesh sampling of a point $\mathbf{x}_i$ on the light, $\mathrm{pdf} = 1/(nA_{\Delta,i})$, the Monte Carlo estimator for area lights is:

$$L_N(\mathbf{x}, \vec{\omega}) = \frac{\rho_d(\mathbf{x})}{\pi}\frac{1}{N}\sum_{i=1}^{N} L_e(\mathbf{x}_i \to \mathbf{x}) V(\mathbf{x}_i \leftrightarrow \mathbf{x}) \frac{\cos\theta\cos\theta_{\text{light}}}{\|\mathbf{x} - \mathbf{x}_i\|^2} nA_{\Delta,i} \ .$$

The laws of reflection and refraction



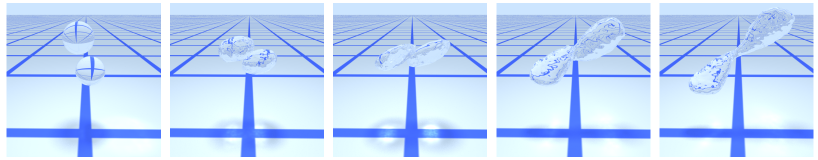
- The law of reflection ($\theta_r = \theta_i$)

$$\vec{\omega}_r = 2(\vec{\omega}' \cdot \vec{n})\vec{n} - \vec{\omega}' \ .$$

- The law of refraction ($n_i \sin\theta_i = n_t \sin\theta_t$)

$$\vec{\omega}_t = \frac{n_i}{n_t}((\vec{\omega}' \cdot \vec{n}) - \vec{\omega}') - \vec{n}\sqrt{1 - \left(\frac{n_i}{n_t}\right)^2 (1 - (\vec{\omega}' \cdot \vec{n})^2)} \ .$$

Rendering transparent fluids



- Choose a maximum recursion depth.
- When the fluid is intersected:
  - Trace a ray in the direction of the reflected ray.
  - Trace a ray in the direction of the transmitted/refracted ray.
- But how much radiance is reflected and how much is refracted?

Multiscale light modelling

- Light at different scales:
  - Quantum electrodynamics (photons)
  - Electromagnetic radiation (waves)
  - Geometrical optics (rays)
  - Radiative transfer (ray bundles)
- Why do we need to know? To understand:
  - how light interacts with materials;
  - where the material properties come from;
  - when rays don't work.
- Geometrical optics include:
  - the law of reflection (Euclid ∼300 B.C. and before);
  - the law of refraction (Ibn Sahl ∼984, Snel van Royen 1621).
- Electromagnetic radiation (Maxwell 1873) includes:
  - the index of refraction;
  - the amount of reflection and transmission at a specular surface (Fresnel 1832).

The Fresnel equations for reflection

- Consider a plane wave incident on a smooth surface.
- The wave gives rise to two other waves. Altogether we have
  - An incident wave (subscript i)
  - A reflected wave (subscript r)
  - A transmitted wave (subscript t)
- We resolve all waves into two independent components:
  - The wave with the electric vector perpendicular to the plane of incidence: $\perp$-polarised light.
  - The wave with the electric vector parallel to the plane of incidence: $\parallel$-polarised light.
- The Fresnel reflectances of the two polarisations are then given by the amplitude ratios

$$\tilde{r}_\perp = \frac{\mathbf{E}_{0r}^\perp}{\mathbf{E}_{0i}^\perp} = \frac{n_i\cos\theta_i - n_t\cos\theta_t}{n_i\cos\theta_i + n_t\cos\theta_t} \quad , \quad \tilde{r}_\parallel = \frac{\mathbf{E}_{0r}^\parallel}{\mathbf{E}_{0i}^\parallel} = \frac{n_t\cos\theta_i - n_i\cos\theta_t}{n_t\cos\theta_i + n_i\cos\theta_t} \ ,$$

where θ_i and θ_t are the angles of incidence and refraction.

Computing reflectances and transmittances (energy ratios)

- To translate Fresnel's amplitude ratios into energy ratios, one uses the squared absolute value:

$$R_\perp = |\tilde{r}_\perp|^2 \quad \text{and} \quad R_\parallel = |\tilde{r}_\parallel|^2 \ .$$

- The Fresnel reflectance for unpolarized light is then

$$R = \frac{1}{2}(R_\perp + R_\parallel) \ .$$

- The transmittances are one minus the reflectances:

$$T_\perp = 1 - R_\perp \quad , \quad T_\parallel = 1 - R_\parallel \quad , \quad T = 1 - R \ .$$

- The Fresnel equations work for complex indices of refraction (n_i and n_t).

Reflection and transmission of radiance

- Requiring energy conservation at the surface boundary:

$$\Phi_i = \Phi_r + \Phi_t \ ,$$

where Φ_i, Φ_r, and Φ_t are the incident, reflected, and transmitted energy fluxes.
- Translated to radiances:

$$\begin{aligned} L_i \cos\theta_i \, \mathrm{d}A \, \mathrm{d}\omega_i &= L_r \cos\theta_r \, \mathrm{d}A \, \mathrm{d}\omega_r + L_t \cos\theta_t \, \mathrm{d}A \, \mathrm{d}\omega_t \\ &\Updownarrow \\ L_i \cos\theta_i \, \mathrm{d}\omega_i &= L_r \cos\theta_r \, \mathrm{d}\omega_r + L_t \cos\theta_t \, \mathrm{d}\omega_t \ . \end{aligned}$$

- Another way to write the differential solid angles:

$$\begin{aligned} \mathrm{d}\omega_i &= \sin\theta_i \, \mathrm{d}\theta_i \, \mathrm{d}\phi_i \\ \mathrm{d}\omega_r &= \sin\theta_r \, \mathrm{d}\theta_r \, \mathrm{d}\phi_r \\ \mathrm{d}\omega_t &= \sin\theta_t \, \mathrm{d}\theta_t \, \mathrm{d}\phi_t \ . \end{aligned}$$

Using the law of reflection

- Using the law of reflection $\theta_i = \theta_r$ and the plane of incidence $\phi_i = \phi_r = \phi_t$, the solid angles become

$$\begin{aligned} \mathrm{d}\omega_i &= \sin\theta_i \, \mathrm{d}\theta_i \, \mathrm{d}\phi_i \\ \mathrm{d}\omega_r &= \sin\theta_i \, \mathrm{d}\theta_i \, \mathrm{d}\phi_i \\ \mathrm{d}\omega_t &= \sin\theta_t \, \mathrm{d}\theta_t \, \mathrm{d}\phi_i \ . \end{aligned}$$

- By insertion, we can then change the energy conservation condition as follows:

$$\begin{aligned} L_i \cos\theta_i \, \mathrm{d}\omega_i &= L_r \cos\theta_r \, \mathrm{d}\omega_r + L_t \cos\theta_t \, \mathrm{d}\omega_t \\ &\Updownarrow \\ L_i \cos\theta_i \sin\theta_i \, \mathrm{d}\theta_i &= L_r \cos\theta_i \sin\theta_i \, \mathrm{d}\theta_i + L_t \cos\theta_t \sin\theta_t \, \mathrm{d}\theta_t \ . \end{aligned}$$

Compression of solid angle upon transmission

- Using the law of refraction (Snell's law), we have (neglecting the imaginary part of the IOR)

$$\sin\theta_t = \frac{n'_i}{n'_t} \sin\theta_i \quad \text{and} \quad \frac{\mathrm{d}\theta_t}{\mathrm{d}\theta_i} = \frac{\mathrm{d}}{\mathrm{d}\theta_i} \sin^{-1}\left(\frac{n'_i}{n'_t} \sin\theta_i\right) = \frac{n'_i \cos\theta_i}{n'_t \cos\theta_t} \ .$$

- By insertion, we can then change the energy conservation condition as follows:

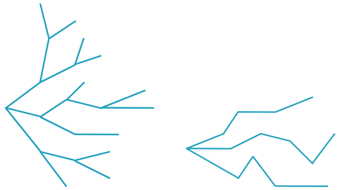
$$\begin{aligned} L_i \cos\theta_i \sin\theta_i \, \mathrm{d}\theta_i &= L_r \cos\theta_i \sin\theta_i \, \mathrm{d}\theta_i + L_t \cos\theta_t \sin\theta_t \, \mathrm{d}\theta_t \\ &\Updownarrow \\ L_i &= L_r + \left(\frac{n'_i}{n'_t}\right)^2 L_t \ . \end{aligned}$$

- Finally, using Fresnel reflectance R, we have

$$L_o = RL_i + \left(\frac{n'_i}{n'_t}\right)^2 (1 - R) L_i \ .$$

Doing efficient rendering

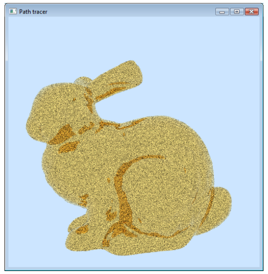
- Time vs. variance.
  - Efficiency $= (\text{variance} \times \text{rendering time})^{-1}$.
- Tree vs. path.



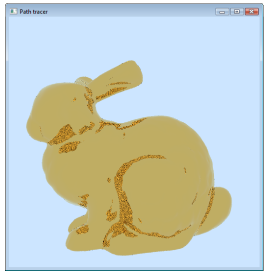
- Splitting vs. Russian roulette.
  - Splitting is a sum over all the branches.
  - Russian roulette is sampling a path.

Demo: The golden bunny

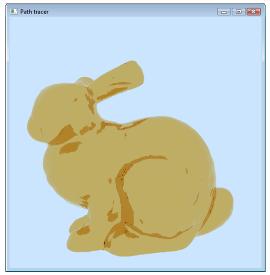
- Using 10 samples per pixel.



Russian roulette
No splitting
Time: 14.4 s



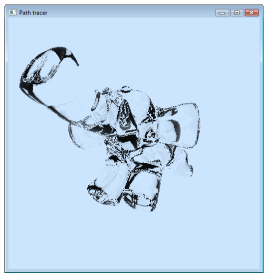
Russian roulette
Split on 1 reflection
Time: 14.8 s



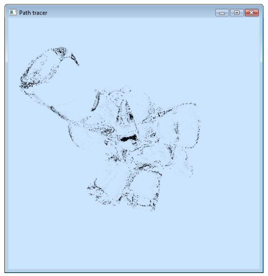
Russian roulette
Split on 2 reflections
Time 14.7 s

Demo: The glass elephant

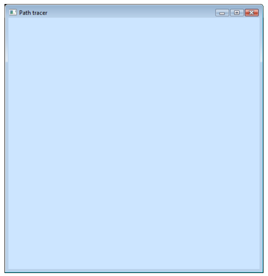
- What does a glass elephant on a flatly coloured background look like?
- Using one sample per pixel.



No Russian roulette
Split on 5 bounces
Time: 4.4 s



No Russian roulette
Split on 10 bounces
Time: 13.7 s



Russian roulette
No splitting
Time 2.6 s

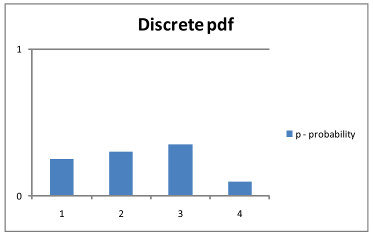
Splitting

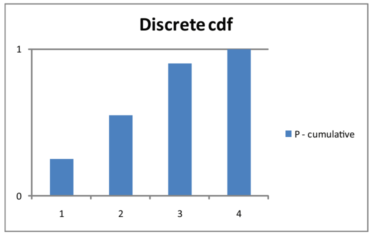
- Sum all the events.
  - Example: Reflection multiplied by reflectance.
  - Example: Sum reflection and transmission.
- Exact but expensive.
- When does the recursion stop?
  - Often not before the combinatorial explosion.
  - It might never.
- When to use splitting?
  - Where variance is most striking.
  - Example: The first (or first few) light bounces.

Russian roulette

- Either - or.
- Sample an event.
  - Example: Reflection or absorption.
  - Example: Reflection or transmission.
- How? Sample a step function.
- What are the steps?
  - The probabilities of an event.
  - Example: Fresnel reflectance.

Sampling a step function

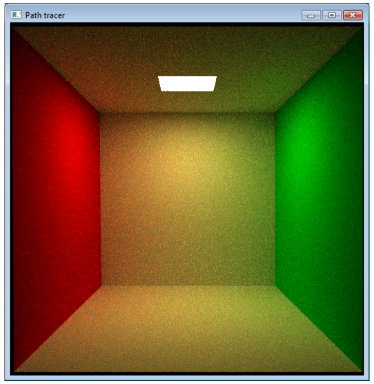


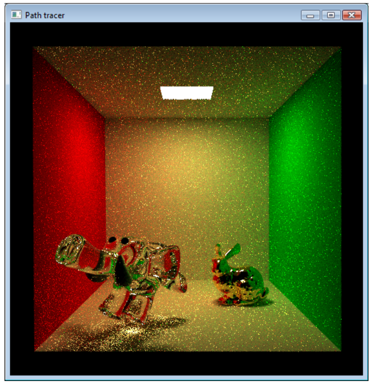


- Algorithm:

  sample $\xi \in [0,1]$ uniformly;
  if $(\xi < P_1)$
      call event 1;
      divide by p_1;
  else if $(\xi < P_2)$
      call event 2;
      divide by p_2;
  else if $(\xi < P_3)$
      ...
  else if $(\xi < P_4)$
      ...

What's wrong with path tracing?





- Low probability paths result in bright dots.
- A bright dot is visually unacceptable.
- A bright dot takes a long time to mean out.
- Soft caustics take "forever".

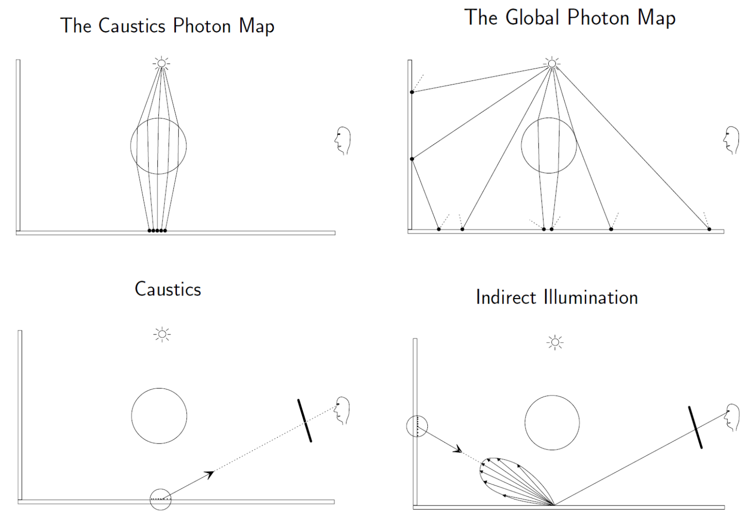
The variance versus bias trade-off

- What if we assume that illumination is always soft?
- Then we can reuse illumination computed at nearby points.
  - Eliminates high frequency noise.
  - Saves computational effort.
- But then the method is *biased*.
  - Error estimates are not well-defined.
  - Images with no visual artifacts may still have substantial error.
  - Artifacts are not eliminated in a predictable way.
- The trade-off is *high frequency* versus *low frequency* noise.

How to pre-compute illumination?

- Sparse tracing from the eye (irradiance caching).
  - Very efficient.
  - Not good for caustics and glossy surfaces.
- Trace rays from the light sources (particle tracing).
  - Rays from the lights are good for finding caustics.
  - Caustics are particularly prone to variance in path tracing.
- A popular particle tracing algorithm: *photon mapping*.
  - Emit.
  - Trace.
  - Store at non-specular surfaces.
  - Use density estimation to reconstruct illumination.

Photon mapping - overview



References

- Jensen, H. W., and Christensen, N. J. A Practical Guide to Global Illumination Using Photon Maps, ACM SIGGRAPH 2000 Course Notes, Course 8, 2000.

Photon emission

- "Photons" in photon mapping are *flux packets*.
- We must ensure that our rays carry *flux*.
- The definition of radiance:

$$L = \frac{\mathrm{d}^2\Phi}{\cos\theta\,\mathrm{d}A\,\mathrm{d}\omega} \ .$$

- Flux emitted from a source:

$$\Phi = \int_{2\pi}\int_A L_e \cos\theta\,\mathrm{d}A\,\mathrm{d}\omega = \int_{2\pi}\int_A L_e(\mathbf{x},\omega)(\mathbf{n}\cdot\omega)\,\mathrm{d}A\,\mathrm{d}\omega \ .$$

- A ray is a Monte Carlo sample of this integral.
  - Shooting N rays.
  - Uniformly sampling source area A for origin of ray.
  - Sampling cosine-weighted hemisphere for direction of ray.
  - Assuming homogeneous, diffuse light.

$$\Phi_p(\mathbf{y},-\omega) = \frac{1}{N}\frac{L_e(\mathbf{x},\omega)(\mathbf{n}\cdot\omega)}{\mathrm{pdf}(\mathbf{x})\,\mathrm{pdf}(\omega)} = \frac{1}{N}L_e A\pi \ .$$

Photon tracing

- Once emitted the flux is confined by the solid angle of the ray.
- Flux carried by a ray changes like radiance upon reflection.
- Tracing "photons" is like tracing ordinary rays.
- Whenever the "photon" is traced to a non-specular surface:
  - It is stored in a kd-tree.
  - Position is stored
  - Direction from where it came (opposite ray direction)
  - Flux (Φ_p) is stored
- Russian roulette is used to stop the recursive tracing

Photon density estimation

- The rendering equation in terms of irradiance $E = \mathrm{d}\Phi/\mathrm{d}A$:

$$\begin{aligned} L(\mathbf{x},\omega) &= L_e(\mathbf{x},\omega) + \int f_r(\mathbf{x},\omega',\omega)\,\mathrm{d}E(\mathbf{x},\omega') \\ &\approx L_e(\mathbf{x},\omega) + \sum_{p\in\Delta A} f_r(\mathbf{x},\omega'_p,\omega)\,\frac{\Delta\Phi_p(\mathbf{x},\omega'_p)}{\Delta A} \end{aligned}$$

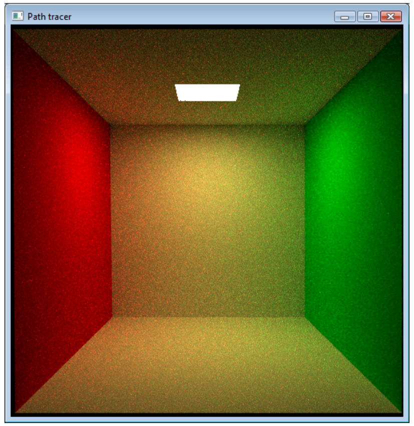
- The Δ-term is called the *irradiance estimate*.
- It is computed using a *kernel method*.
  - Consider a circular surface area $\Delta A = \pi d^2$
  - Then the power contributed by the “photon” $p \in \Delta A$ is

$$\Delta\Phi_p(\mathbf{x},\omega'_p) = \Phi_p\,K\left(\frac{\|\mathbf{x}-\mathbf{x}_p\|}{d}\right)$$

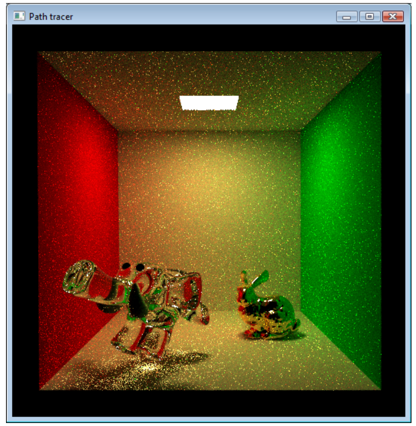
    where K is a filter kernel
  - Simplest choice (constant kernel)

$$K(a) = \begin{cases} 1 & \text{for } a < 1 \\ 0 & \text{otherwise} \end{cases}$$

Reference images (path tracing)

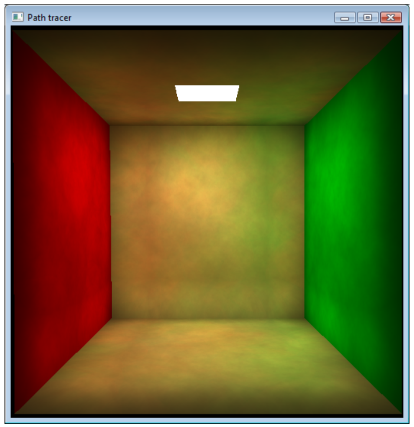


c. 3 min

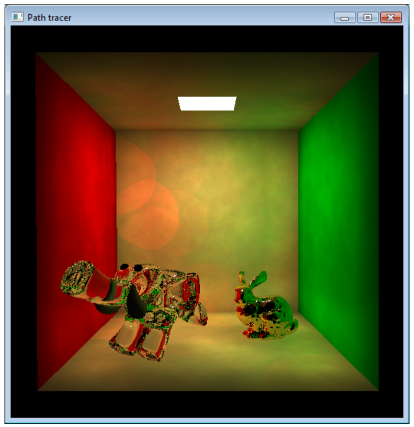


c. 5 min and 20 s

Photon mapping results (radiance estimate)



c. 10 s

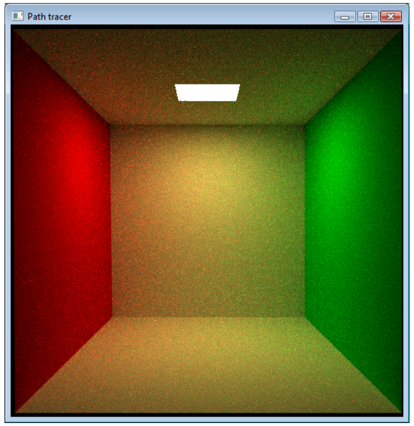


c. 10 s

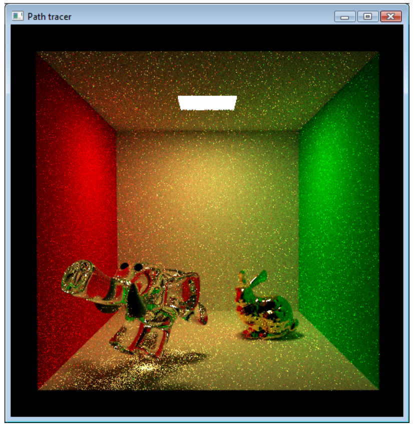
Final Gathering

- How to eliminate the worst low frequency noise:
- Use two photon maps:
  - A global map (all photons: L(S|D)*D)
  - A caustics map (only caustics photons: LS+D)
- At the first non-specular surface reached from the eye:
  - Do a final gathering (sample the hemisphere)
  - Add the radiance estimate from the caustics map
- At subsequent non-specular surfaces:
  - Use the radiance estimate from the global map

Reference images (path tracing)

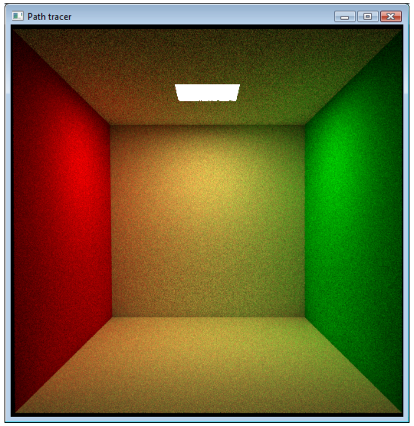


c. 3 min

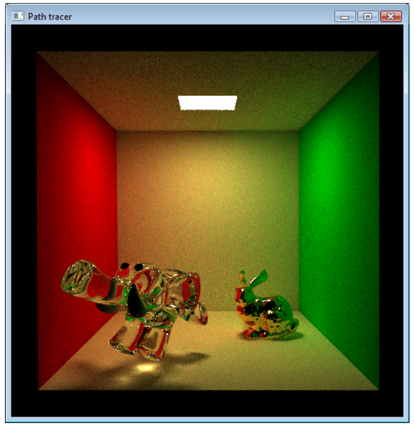


c. 5 min and 20 s

Photon mapping results (final gathering)



c. 3 min



c. 5 min and 20 s

What happens in a volume?

- Some light is absorbed.
- Some light scatters away (out-scattering).
- Some light scatters back into the line of sight (in-scattering).

  (absorption + out-scattering = extinction)

- Historical origins:

Bouguer [1729, 1760]	A measure of light. Exponential extinction.
Lambert [1760]	Cosine law of perfectly diffuse reflection and emission.
Lommel [1887]	Testing Lambert's cosine law for scattering volumes. Describing isotropic in-scattering mathematically.
Schuster [1905]	Scattering in foggy atmospheres (plane-parallel media). Reinventing the theory in astrophysics.
King [1913]	General equation which includes anisotropic scattering (the phase function).
Chandrasekhar [1950]	The first definitive text on radiative transfer.

How to describe scattering?

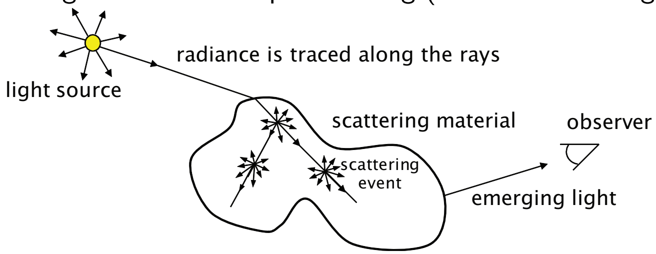
- We follow a ray of light passing through a scattering medium.
- The parameters describing the medium are
  - σ_a the absorption coefficient [m^{-1}]
  - σ_s the scattering coefficient [m^{-1}]
  - σ_t the extinction coefficient [m^{-1}] ($\sigma_t = \sigma_a + \sigma_s$)
  - p the phase function.
- The radiative transfer equation (RTE)

$$\begin{aligned}(\vec{\omega} \cdot \nabla) L(\mathbf{x}, \vec{\omega}) \;=\; & -\sigma_t(\mathbf{x}) L(\mathbf{x}, \vec{\omega}) \\ & + \sigma_s(\mathbf{x}) \int_{4\pi} p(\mathbf{x}, \vec{\omega}', \vec{\omega}) L(\mathbf{x}, \vec{\omega}) \, \mathrm{d}\omega' \\ & + L_e(\mathbf{x}, \vec{\omega}) \ ,\end{aligned}$$

where L is radiance at the position $\mathbf{x}$ along the ray in the direction $\vec{\omega}$ and L_e is emitted radiance.

Rendering volumes

- The general method: path tracing (Monte Carlo integration).



- The integral form of the radiative transfer equation (for a non-emitter):

$$L(s) = T_r(0,s)L(0) + \int_0^s T_r(s',s)\sigma_s(s') \int_{4\pi} p(s',\vec{\omega}',\vec{\omega})L(s',\vec{\omega}')\,\mathrm{d}\omega'\,\mathrm{d}s' ,$$

  where T_r is the beam transmittance and s is the distance travelled along a ray with direction $\vec{\omega}$ and origin $\mathbf{o}$ on the surface of the volume such that $\mathbf{x} = \mathbf{o} + s\vec{\omega}$ is a point along the ray inside the volume.

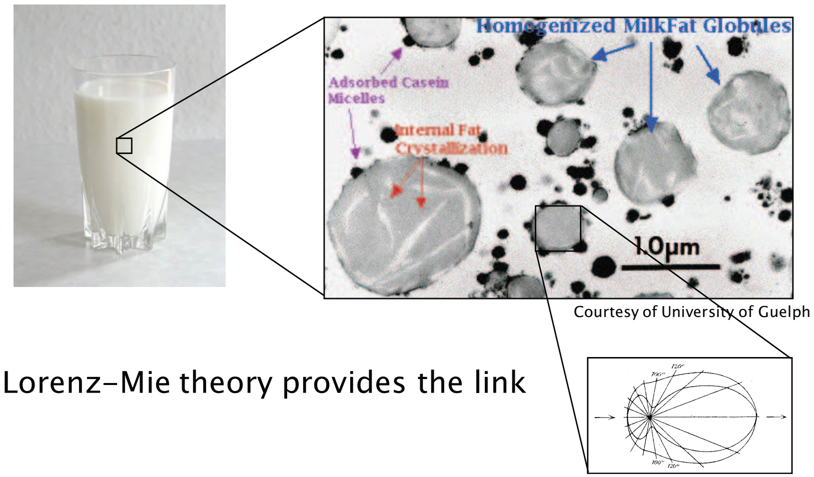
Path tracing volumes

- We need an estimator for each integral. In path tracing, we usually take only one sample for each estimator per frame.
- When a ray hits a scattering material, do the following.
  1. If the ray hit from outside, do a normal volume transmission and stop. If the ray hit from the inside, proceed.
  2. Sample the distance $d = -\ln(\xi)/\sigma_t$ to the next scattering event. If the scattering event is outside the volume ($d > s$), do a normal transmission and stop. Otherwise, proceed.
  3. Do a Russian roulette with the scattering albedo. If $\xi > \alpha$, the ray is absorbed. Otherwise, proceed.
  4. Find the position of the next scattering event ($\mathbf{x} = \mathbf{o} + d\,\vec{\omega}$).
  5. Sample the direction of the scattered ray (sample the phase function).
  6. Trace the scattered ray. If it hits something, proceed to step 2. Otherwise, stop.
- This procedure only works for monochromatic rays.

The colours of materials

- What causes materials to look the way they do?
- The appearance of a material is largely determined by its particle composition.
- We therefore turn our attention to the general theory of light scattering by spherical particles: the Lorenz-Mie theory.
- This is a general case of appearance modelling.
  - Lorenz showed in 1890 that the rainbow and the sky (Rayleigh scattering) are simply special cases.
  - Mie derived the theory again in 1908 to predict the appearance of colloidal gold solutions.
- In general, we can use the theory to obtain the optical properties of turbid materials consisting of approximately spherical particles.
- The obtained optical properties are the ones used in rendering: the scattering coefficient σ_s, the extinction coefficient σ_t, and the phase function p.

From particles to appearance



Lorenz–Mie theory provides the link

Scattering by spherical particles

- The Lorenz-Mie theory:

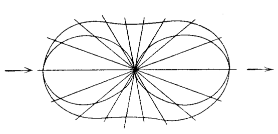
$$p(\theta) = \frac{|S_1(\theta)|^2 + |S_2(\theta)|^2}{2|k|^2 C_s}$$

$$S_1(\theta) = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \left(a_n \pi_n(\cos\theta) + b_n \tau_n(\cos\theta)\right)$$

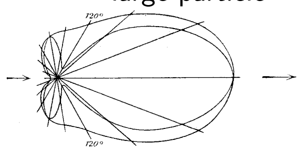
$$S_2(\theta) = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} \left(a_n \tau_n(\cos\theta) + b_n \pi_n(\cos\theta)\right) \quad .$$

- a_n and b_n are the Lorenz-Mie coefficients.
- π_n and τ_n are spherical functions associated with the Legendre polynomials.

small particle



large particle

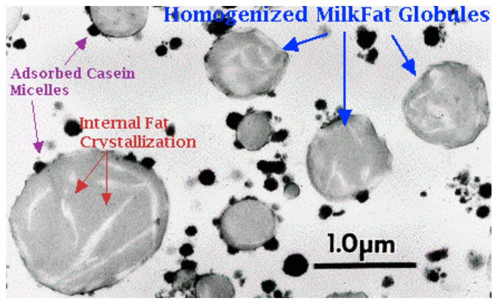


Quantity of scattering

- Lorenz-Mie theory continued:
  The scattering and extinction cross sections of a particle:

$$C_s = \frac{\lambda^2}{2\pi |n_{\mathrm{med}}|^2} \sum_{n=1}^{\infty} (2n+1) \left(|a_n|^2 + |b_n|^2\right)$$

$$C_t = \frac{\lambda^2}{2\pi} \sum_{n=1}^{\infty} (2n+1) \mathrm{Re}\left(\frac{a_n + b_n}{n_{\mathrm{med}}^2}\right) \quad .$$



Bulk optical properties of a material

- Start with volume fraction of a component v

$$v = \frac{4\pi}{3} \int_{r_{\min}}^{r_{\max}} r^3 N(r)\,\mathrm{d}r$$

- Use it to scale the number density distribution N.
- Use that to find the bulk properties σ_s (and σ_t likewise)

$$\sigma_s = \int_{r_{\min}}^{r_{\max}} C_s(r) N(r)\,\mathrm{d}r$$

Computing scattering properties

- Input needed for computing scattering properties:
  - Particle composition (volume fractions, particle shapes).
  - Refractive index for host medium n_{med}.
  - Refractive index for each particle type n_p.
  - Size distribution for each particle size (r).
- Lorenz-Mie theory uses a series expansion. How many terms should we include?
- Number of terms to sum $N = \lceil |x| + p|x|^{1/3} + 1 \rceil$.
  - Empirically justified [Wiscombe 1980, Mackowski et al. 1990].
  - Theoretically justified [Cachorro and Salcedo 1991].
  - For a maximum error of 10^{-8}, use $p = 4.3$.
- Code for evaluating the expansions in the Lorenz-Mie theory is available online [Frisvad et al. 2007]:
  http://www.imm.dtu.dk/pubdb/p.php?5288

Particle contents (examples)

- Natural water
  - Refractive index of host: saline water.
  - Mineral and alga contents: user input in volume fractions.
  - Refractive indices of mineral and algae: empirical formulae.
  - Shape of mineral and algal particles: spheres.
  - Size distributions: power laws.
- Icebergs
  - Refractive index of host: pure ice.
  - Brine and air contents: depend on temperature, salinity, and density.
  - Refractive index of brine and air: empirical formula, measured absorption spectrum, and $n_{\text{air}} = 1.00$.
  - Shape of brine pores and air pockets: closed cylinders and ellipsoids.
  - Size distributions: power laws.

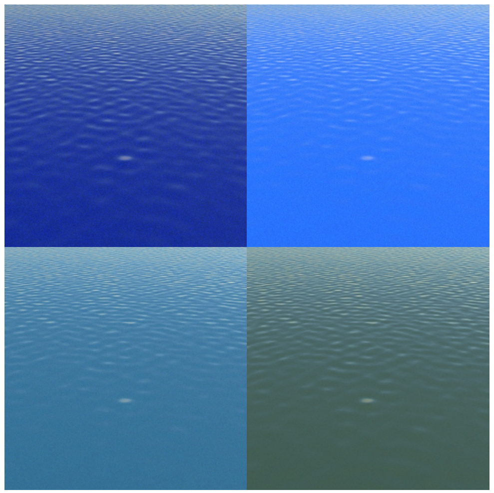
Case study: natural waters

- Glacial melt water with rock flour mixing with purer water from melted snow to give Lake Pukaki in New Zealand its beautiful bright blue colour.

Oceanic and coastal waters

Cold Atlantic

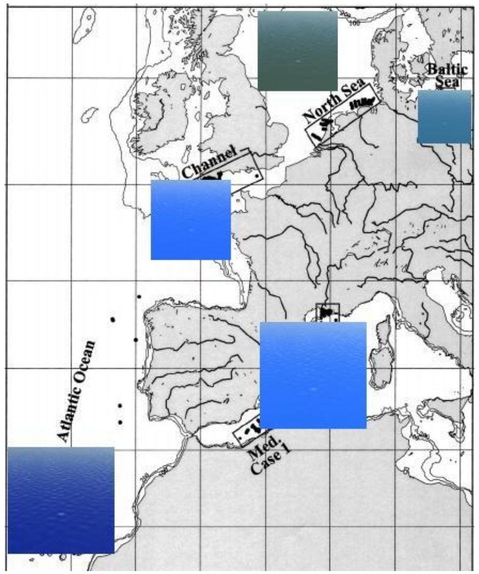
Mediterranean



Baltic

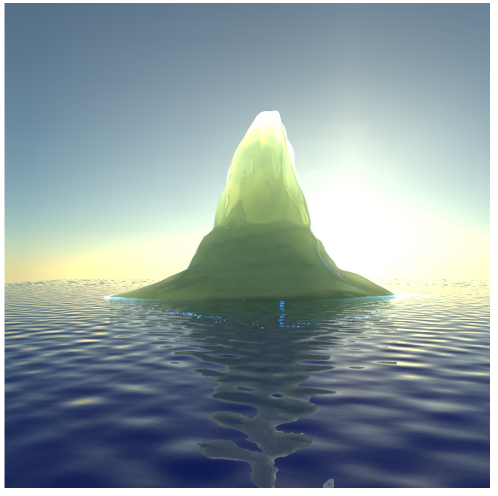
North Sea

Oceanic and coastal waters



Challenge

- Let us add DSC and dynamics and render a melting iceberg in turbulent sea water!



Green algae on iceberg.
(Photo courtesy Gerhard Dieckmann)