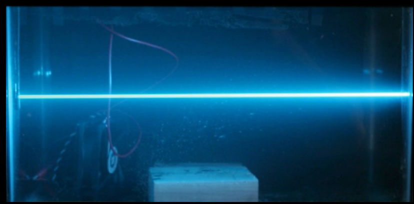
Scattering



70001 – Advanced Computer Graphics: Photographic Image Synthesis

Abhijeet Ghosh

Lecture 16, Feb. 27th 2024

1

Volumetric Scattering

- Global light transport in participating media
- Single scattering
  - optically thin media
- Multiple scattering
  - optically dense media
- Subsurface scattering

2

Atmospheric scattering

Diego Gutierrez

- Single and multiple scattering

3

Clouds

http://mev.fopf.mipt.ru/

- Multiple scattering inside clouds, single scattering in God rays!

4

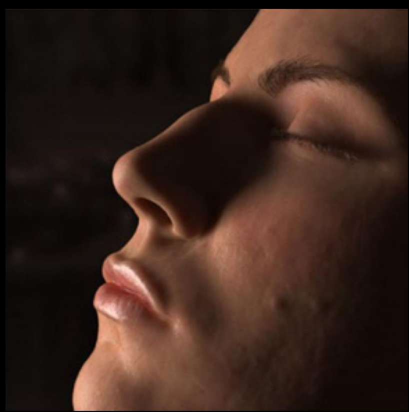
Light beams

Wojciech Jarosz

- Single scattering of light in dust particles

5

Subsurface scattering

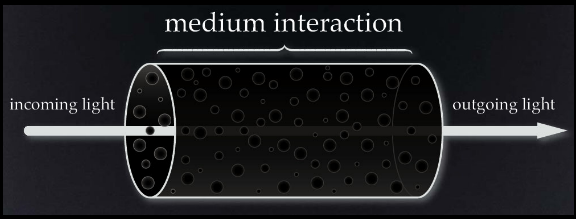


Henrik Wann Jensen

- Multiple scattering inside a material

6

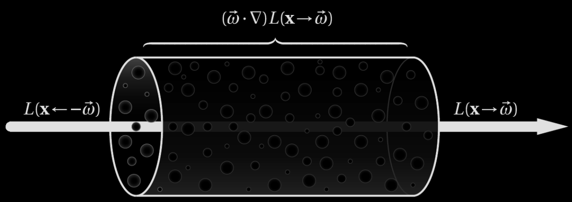
Volumetric Scattering



- Participating media
  - particles distributed in a volume of 3D space
- Absorption
- Emission
- Scattering

7

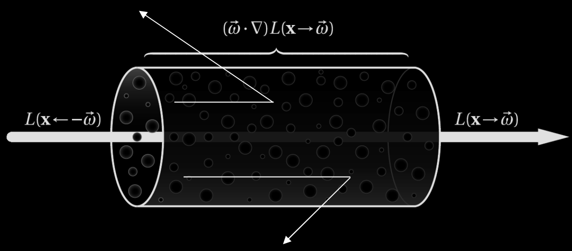
Absorption



- $L_o(x, \omega) - L_i(x, \omega) = dL_o(x, \omega) = - \sigma_a(x, \omega) L_i(x, -\omega)\, dt$,
  - Absorption cross-section σ_a: probability density of absorption per unit distance (m^{-1})

8

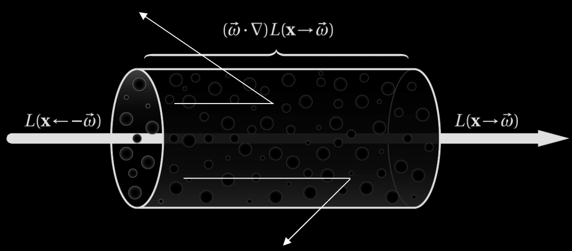
Out-scattering



- $dL_o(x, \omega) = - \sigma_s(x, \omega)\, L_i(x, -\omega)\, dt$,
  - Scattering coefficient σ_s: probability density of out-scatter per unit distance (m^{-1})

9

Extinction

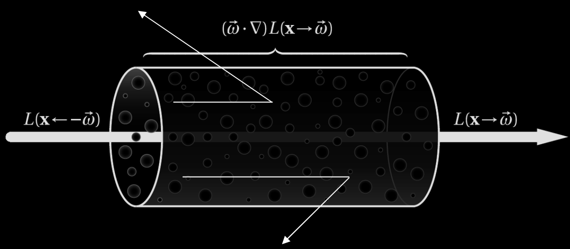


- $\sigma_t(x, \omega) = \sigma_a(x, \omega) + \sigma_s(x, \omega)$
  - Extinction coefficient σ_t: probability density of radiance reduction per unit distance (m^{-1})
- Integrating the differential equation:

  $T_r(x \to x') = \exp(- \int_0^d \sigma_t(x + td, \omega)dt)$, transport coefficient

10

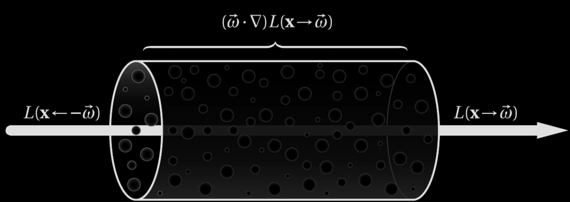
Homogeneous media



- $\sigma_t(x, \omega)$ constant in homogeneous media
- $T_r(x \to x') = \exp(-\sigma_t d)$, Beer's Law

11

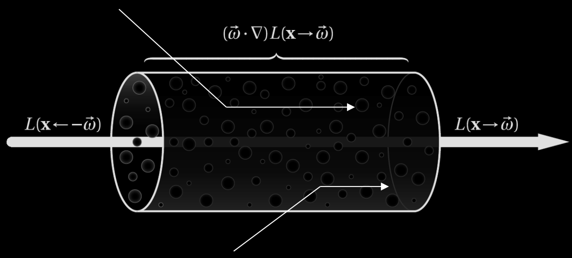
Emission



- $dL_o(x, \omega) = Q(x, \omega)\, dt$,
  - $Q(x, \omega)$: emitted radiance added per unit distance

12

In-scattering

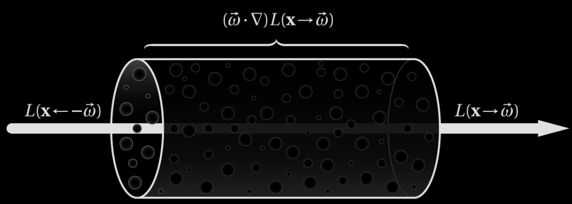


- In-scattering described by phase function $p(\omega \to \omega')$
  - Volumetric analog of BRDF

- $\frac{1}{4\pi} \int_{\Omega} p(\omega \to \omega')\, d\omega' = 1$

13

Added radiance



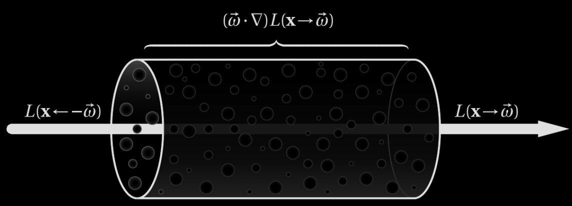
- Total added radiance: emission + in-scattering

  $dL_o(x, \omega) = S(x, \omega)\, dt,$

- $S(x, \omega) = Q(x, \omega) + \sigma_s(x, \omega) \int_{\Omega} p(x, -\omega' \to \omega)\, L_i(x, \omega')\, d\omega'.$

14

Radiative transfer



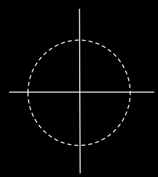
- Volume rendering equation :

  $dL_o(x, \omega) = - \sigma_t(x, \omega)\, L_i(x, -\omega)\, dt + S(x, \omega)\, dt.$

15

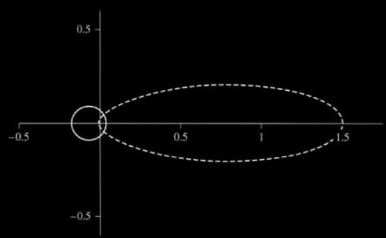
Phase Function

- Isotropic: $p(\omega \rightarrow \omega') = 1/4\pi$



- Anisotropic: Henyey-Greenstein

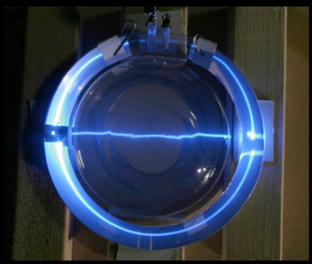
  $$p_{HG}(\cos\theta) = \frac{1 - g^2}{4\pi\ (1 + g^2 - 2g\cos\theta)^{3/2}}$$



  – θ is the angle between ω & ω'
  – $g \rightarrow [-1, 1]$,
  – g = + 1 forward scatting, -1 backward scattering, 0 isotropic scattering.

16

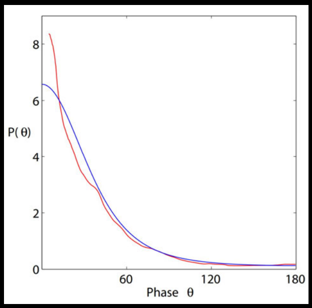
Measuring scattering in smoke Hawkins et al. 05

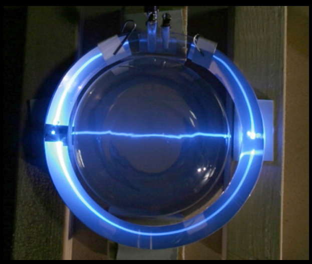


- Phase function measurement
  - Conical mirror at 45°, laser beam through volume of smoke trapped in glass bowl
  - 1D measurement of the phase function via conical mirror to fit the g parameter of the H-G phase function
  - highly forward scattering g

17

Measuring scattering in smoke Hawkins et al. 05

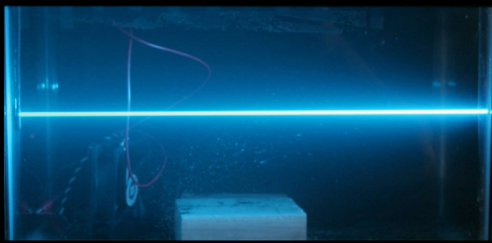




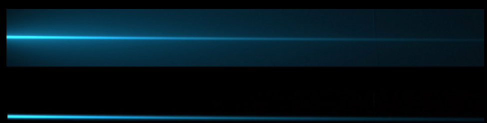
- Phase function measurement
  - Conical mirror at 45°, laser beam through volume of smoke trapped in glass bowl
  - 1D measurement of the phase function via conical mirror to fit the g parameter of the H-G phase function
  - highly forward scattering g

18

Measuring scattering in smoke
Hawkins et al. 05

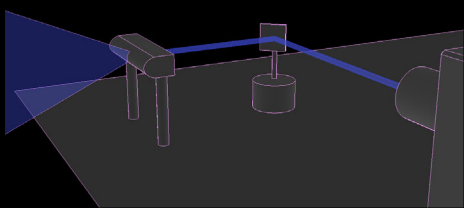


- Albedo $\alpha = \sigma_s / \sigma_t$
  - extinction in tank
  - removal of multiple scattering by subtracting background airlight



19

Measuring scattering in smoke
Hawkins et al. 05



- Laser sweep for density measurement of each plane of smoke
- Stacking of planes to construct a volume of smoke for rendering

20

Measuring scattering in smoke
Hawkins et al. 05

- MC path tracing of smoke volume using PBRT

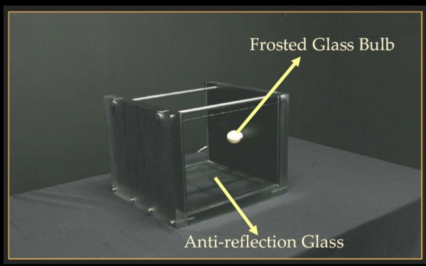
21

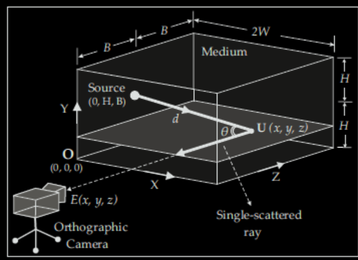
Measuring scattering in water – Narasimhan et al. 06

- Dilute water measurement of single scattering

22

Measuring scattering in water – Narasimhan et al. 06





- Dilute water measurement of single scattering

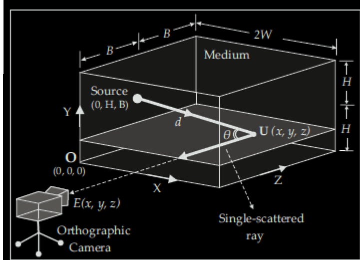
Measuring scattering in water – Narasimhan et al. 06

$$E(x,y,z) = \frac{I_0}{d^2} \,.\, e^{-\sigma d} \,.\, \beta P(g, \pi - \theta) \,.\, e^{-\sigma z} \,.$$

$$d = \sqrt{x^2 + (y-H)^2 + (z-B)^2} \;,\; \cos\theta = (z-B)/d \quad (1)$$

$$E(x,y) = \int_0^{2B} E(x,y,z)\, dz$$

$$= \beta \int_0^{2B} \frac{I_0\, e^{-\sigma\left(z + \sqrt{x^2+(y-H)^2+(z-B)^2}\right)}}{x^2 + (y-H)^2 + (z-B)^2} P(g, \pi - \theta)\, dz \,.$$



- Dilute water measurement of single scattering

Measuring scattering in water – Narasimhan et al. 06

measurement concentration

simulated actual concentration

25

Radiative Transport

$$
\begin{aligned}
(\vec{\omega}\cdot\nabla)L(x,\vec{\omega}) \;=\;& -\sigma_t L(x,\vec{\omega}) + && \text{extinction}\\
& \sigma_s \int_{4\pi} p(\vec{\omega},\vec{\omega}')L(x,\vec{\omega}')d\vec{\omega}' + && \text{in-scattering}\\
& s(x,\vec{\omega}) && \text{source term}
\end{aligned}
$$

- Costly computation for rendering of highly scattering materials
- Light also diffuses with multiple scattering!

26

Accelerating Cloud Rendering

[Kallweit et al. 2017]

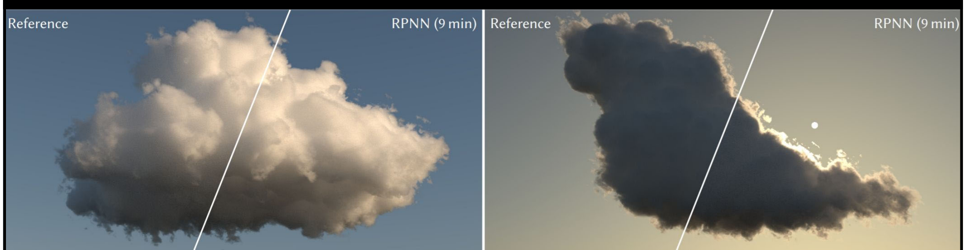
(a) 1 interaction Render time = 0.017 h | (b) 4 interactions Render time = 0.89 h | (c) 16 interactions Render time = 6.5 h | (d) 64 interactions Render time = 16 h | (e) All interactions Render time = 34 h

- Multiple scattering in clouds can be very expensive due to high albedo of cloud particles!

27

Accelerating Cloud Rendering

[Kallweit et al. 2017]



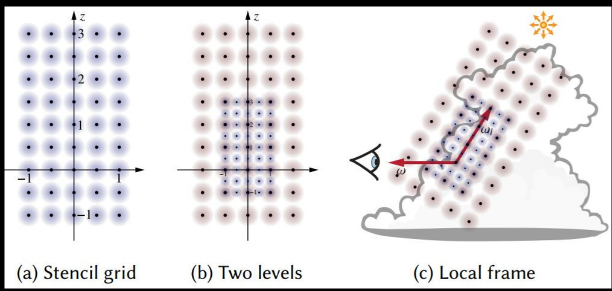
- Multiple scattering is clouds predicted using neural networks. Single scattering and direct reflection still rendered using MC rendering.

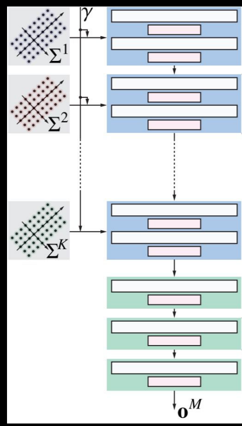
- Multiple scattering can very directional in a cloud volume. Hence, a low-resolution 3D descriptor of the cloud volume is employed to correctly predict the spatial and direction distribution of radiance using an MLP.

28

Accelerating Cloud Rendering

[Kallweit et al. 2017]



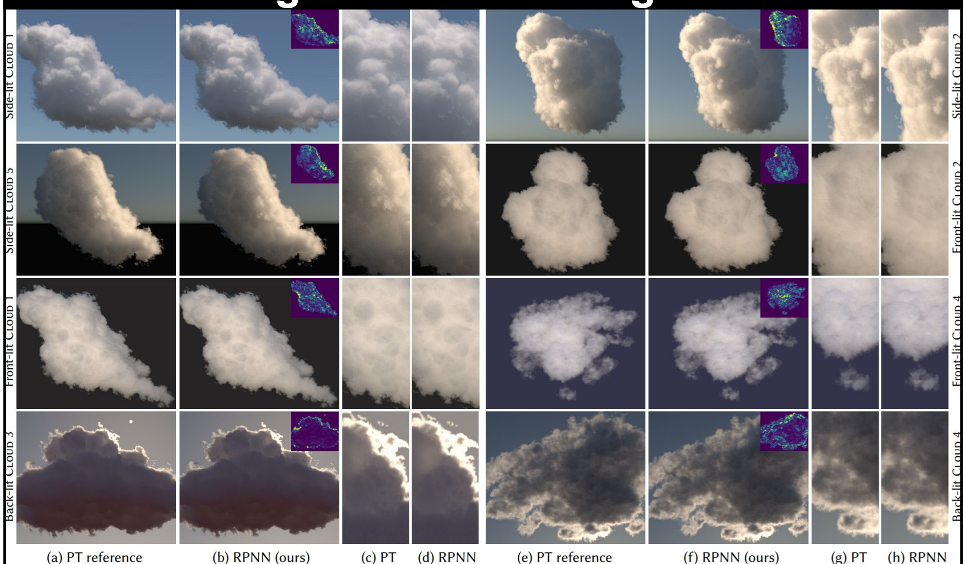


- A 5x5x9 hierarchical descriptor is constructed and oriented towards the light inside the volume.
- The descriptor is an input to an MLP network for prediction of the directional radiance distribution. The MLP network processes the stencil hierarchically from coarse scale to fine scale for the prediction.

29

Accelerating Cloud Rendering

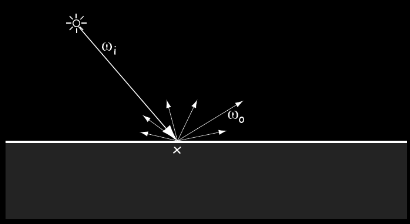
[Kallweit et al. 2017]



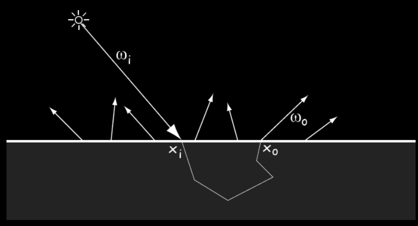
24x speedup over pathtracing for environmental illumination, >1000x speedup for just sunlight

30

Subsurface scattering



BRDF $f_r(x, \vec{\omega}_i, \vec{\omega}_o) = \frac{dL(x,\vec{\omega}_i)}{dE(x,\vec{\omega}_o)}$



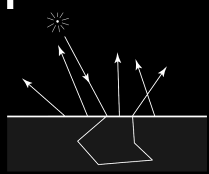
BSSRDF $S(x_i, \vec{\omega}_i, x_o, \vec{\omega}_o) = \frac{dL(x_o,\vec{\omega}_o)}{d\Phi(x_i,\vec{\omega}_i)}$

- Bidirectional surface scattering distribution function (BSSRDF), 8D function
  – generalizes the 4D BRDF of surface reflection to transport across material

31

Diffuse BSSRDF – Jensen et al. 01

$$R_d(r) = \frac{(\vec{n} \cdot \vec{E}(x_o))}{d\Phi_i(x_i)}$$

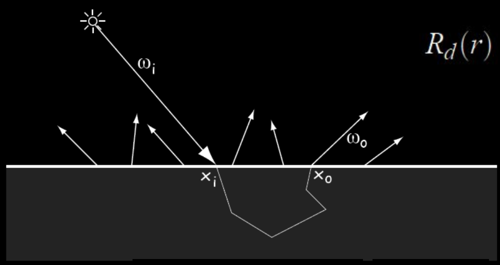


- Diffuse BSSRDF Rd given by ratio of radiant exitance and incident flux. No dependence on angle of incidence or exitance, only distance of transport!

- $R_d(r)$ is a 1-D radially symmetric diffusion kernel that depends on distance r of scattering within material, i.e., distance between x_i and x_o.

32

Diffuse BSSRDF [Jensen et al. 01]



$$R_d(r) = \frac{\alpha'}{4\pi}\left\{z_r\left(\sigma_{tr}+\frac{1}{d_r}\right)\frac{e^{-\sigma_{tr}d_r}}{d_r^2} + z_v\left(\sigma_{tr}+\frac{1}{d_v}\right)\frac{e^{-\sigma_{tr}d_v}}{d_v^2}\right\}$$

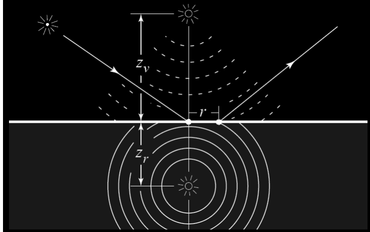
BSSRDF $S_d(x_i,\omega_i,x_o,\omega_o) \approx R_d(x_i,x_o)$

- $R_d(x_i, x_o)$ models isotropic Gaussian-like diffusion between points x_i and x_o
  - Dipole model for homogeneous semi-infinite medium
  - Sum of two Gaussian like fall-offs

33

Dipole diffusion - Jensen et al. 01

$$R_d(r) = \frac{\alpha'}{4\pi}\left\{z_r\left(\sigma_{tr}+\frac{1}{d_r}\right)\frac{e^{-\sigma_{tr}d_r}}{d_r^2} + z_v\left(\sigma_{tr}+\frac{1}{d_v}\right)\frac{e^{-\sigma_{tr}d_v}}{d_v^2}\right\}$$



distance of real source from x_o: d_r

distance of virtual source from x_o: d_v

z_r & z_v distance of sources to the surface

α' albedo, σ_{tr} is extinction coefficient

- Planar homogeneous semi-infinite medium
- Dipoles enforce the net inward flux at the boundary to be zero!
- Think of dipole diffusion as a sum of two Gaussian like fall-offs.

34

Dipole diffusion - Jensen et al. 01

BRDF

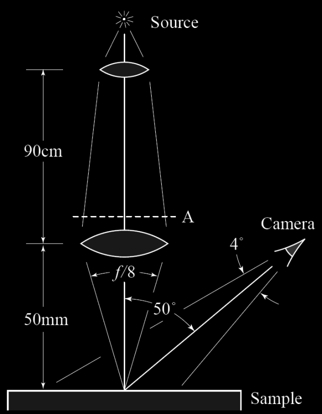
Rough appearance due to surface reflection

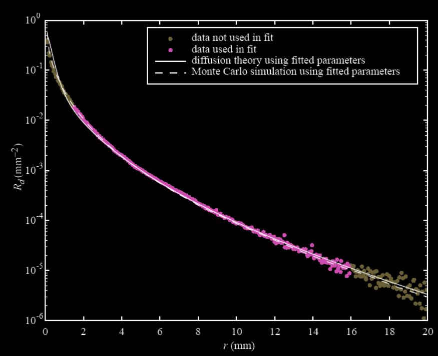
BSSRDF

Light bleed into the shadow regions due to diffusion

35

Diffusion measurement – Jensen et al. 01

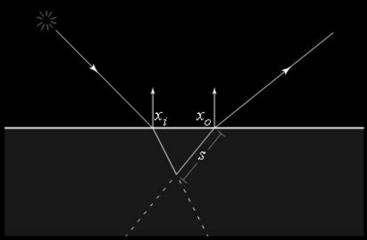




36

Single scattering BRDF - Jensen et al. 01

$$f_r^{(1)}(x, \vec{\omega}_i, \vec{\omega}_o) = \alpha F \frac{p(\vec{\omega}_i' \cdot \vec{\omega}_o')}{|\vec{n} \cdot \vec{\omega}_i'| + |\vec{n} \cdot \vec{\omega}_o'|}$$



Single scattering abedo: α

Phase function: *p, H-G model forward scatter*

Fresnel *F*

37

Skin rendering – BRDF model

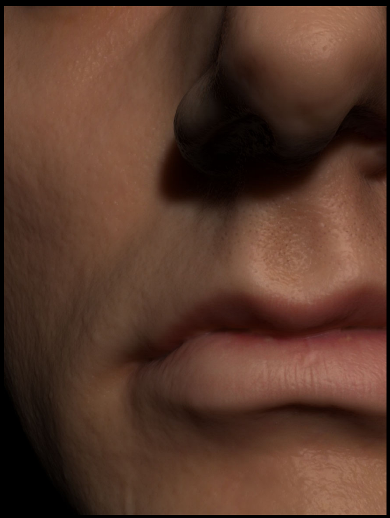
38

Skin rendering with dipole diffusion

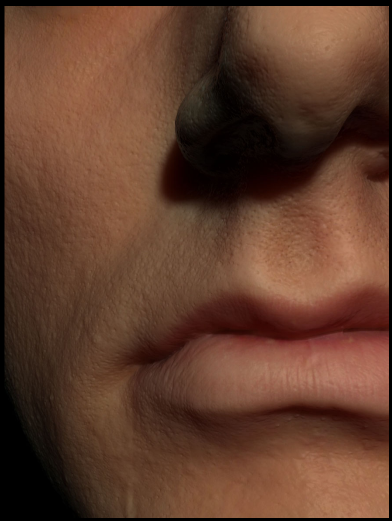
rendering with measured parameters

39

Multilayer diffusion – Donner & Jensen 05



rendering with dipole diffusion



rendering with 3 diffusion layers

40

Skin - layered media

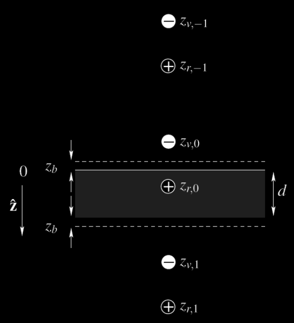
Skin diagram

(courtesy University of Iowa)

- Skin layered heterogeneous scattering media
- Single layer diffusion not sufficient!

41

Multipole model – Donner&Jensen 05



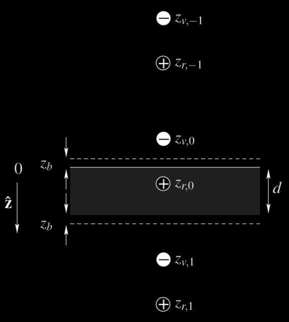
$$R(r) = \sum_{i=-n}^{n} \frac{\alpha' z_{r,i}(1+\sigma_{tr} d_{r,i}) e^{-\sigma_{tr} d_{r,i}}}{4\pi d_{r,i}^3} - \frac{\alpha' z_{v,i}(1+\sigma_{tr} d_{v,i}) e^{-\sigma_{tr} d_{v,i}}}{4\pi d_{v,i}^3}$$

$$T(r) = \sum_{i=-n}^{n} \frac{\alpha' (d - z_{r,i})(1+\sigma_{tr} d_{r,i}) e^{-\sigma_{tr} d_{r,i}}}{4\pi d_{r,i}^3} - \frac{\alpha' (d - z_{v,i})(1+\sigma_{tr} d_{v,i}) e^{-\sigma_{tr} d_{v,i}}}{4\pi d_{v,i}^3}$$

- Multipole models reflectance and transmissions through thin layers more accurately than dipole model

42

Kubelka-Munk theory



$$T_{12} = T_1 * T_2 + T_1 * R_2 * R_1 * T_2 + T_1 * R_2 * R_1 * R_2 * R_1 * T_2 + \ldots$$

$$\mathcal{T}_{12} = \frac{\mathcal{T}_1 \mathcal{T}_2}{1 - \mathcal{R}_2 \mathcal{R}_1}$$

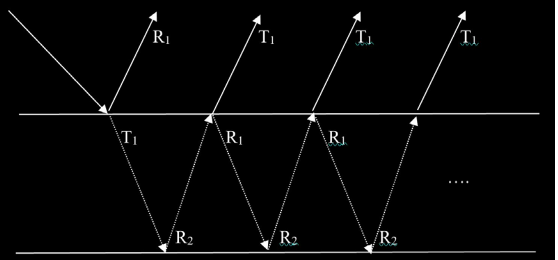
$$\mathcal{R}_{12} = \mathcal{R}_1 + \frac{\mathcal{T}_1 \mathcal{R}_2 \mathcal{T}_1}{1 - \mathcal{R}_2 \mathcal{R}_1}$$

- Derived from geometric series formula!

43

Kubelka-Munk theory – geometric series

$$\mathcal{R}_{12} = \mathcal{R}_1 + \frac{\mathcal{T}_1 \mathcal{R}_2 \mathcal{T}_1}{1 - \mathcal{R}_2 \mathcal{R}_1}$$



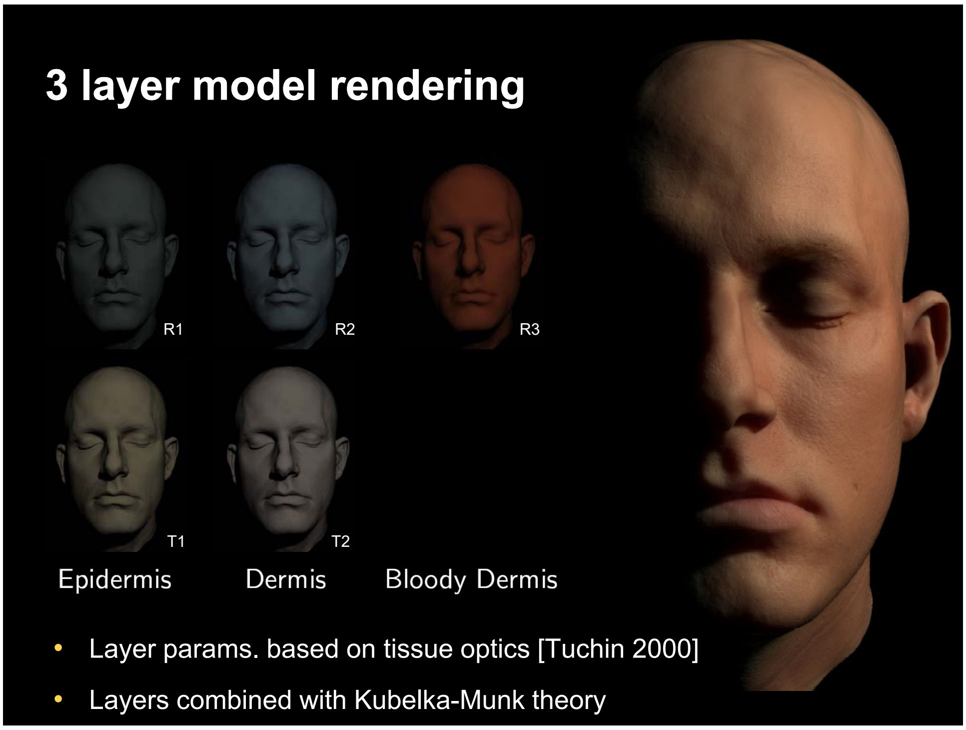
- $R_{12} = R_1 + T_1 * R_2 * T_1 + T_1 * R_2 * R_1 * R_2 * T_1 + T_1 * R_2 * R_1 * R_2 * R_1 * R_2 * T_1 + \ldots..$

  $= R_1 + T_1 * R_2 * T_1 * (1 + (R_1 * R_2) + (R_1 * R_2)^2 + (R_1 * R_2)^3 + \ldots)$

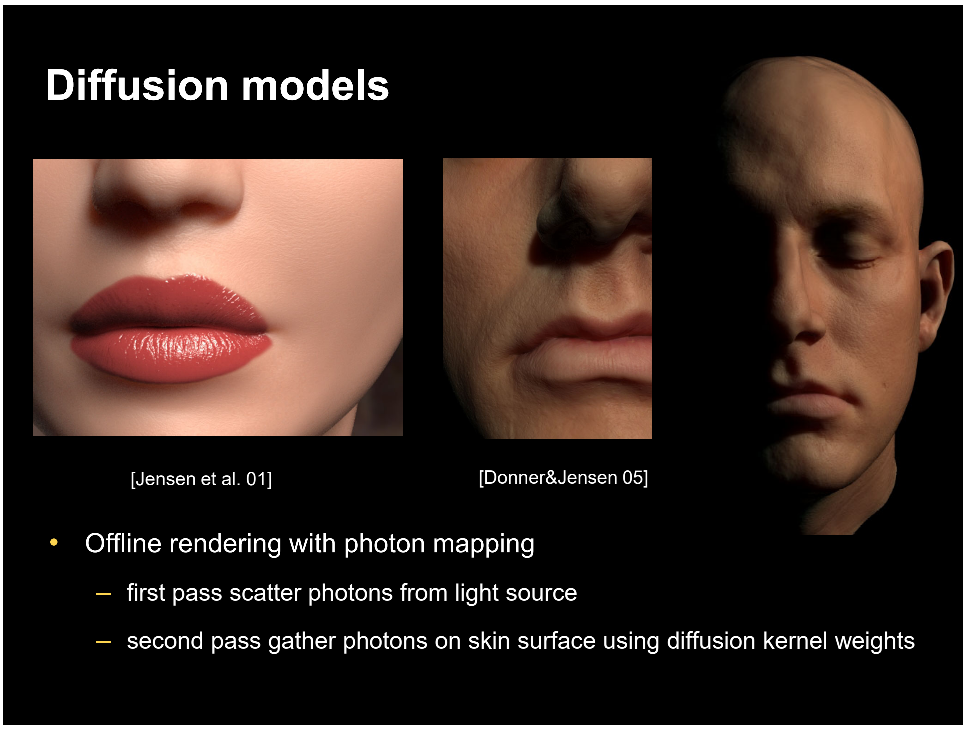
  $= R_1 + T_1 * R_2 * T_1 * (1/(1 - (R_1 * R_2))$

- Derived from geometric series formula!

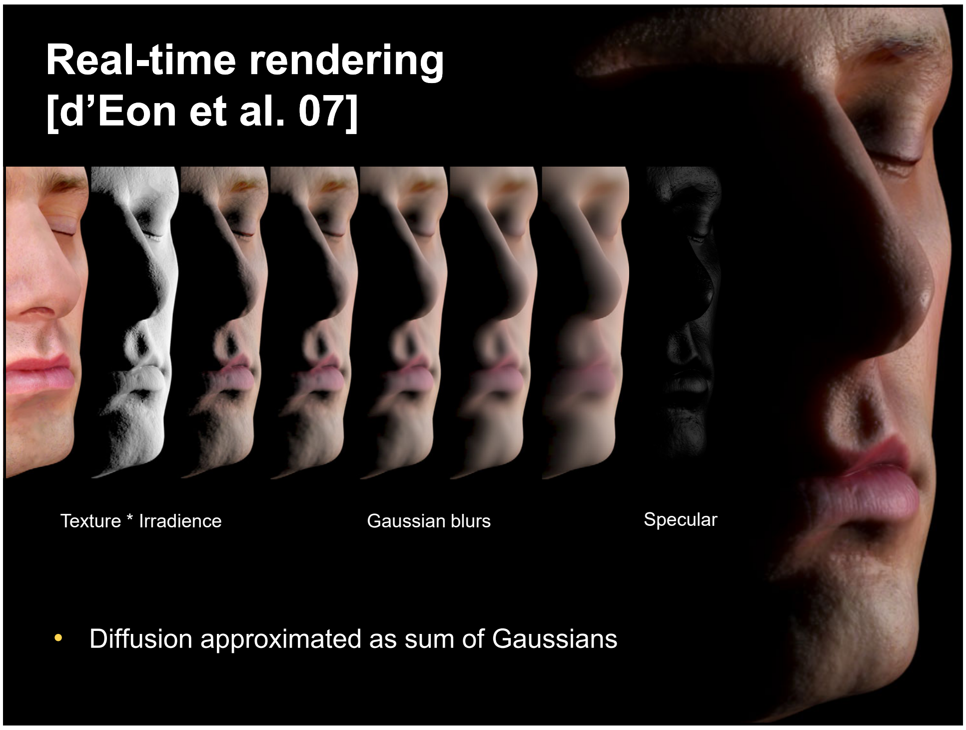
44


3 layer model rendering
R1
R2
R3
T1
T2
Epidermis
Dermis
Bloody Dermis
Layer params. based on tissue optics [Tuchin 2000]
Layers combined with Kubelka-Munk theory

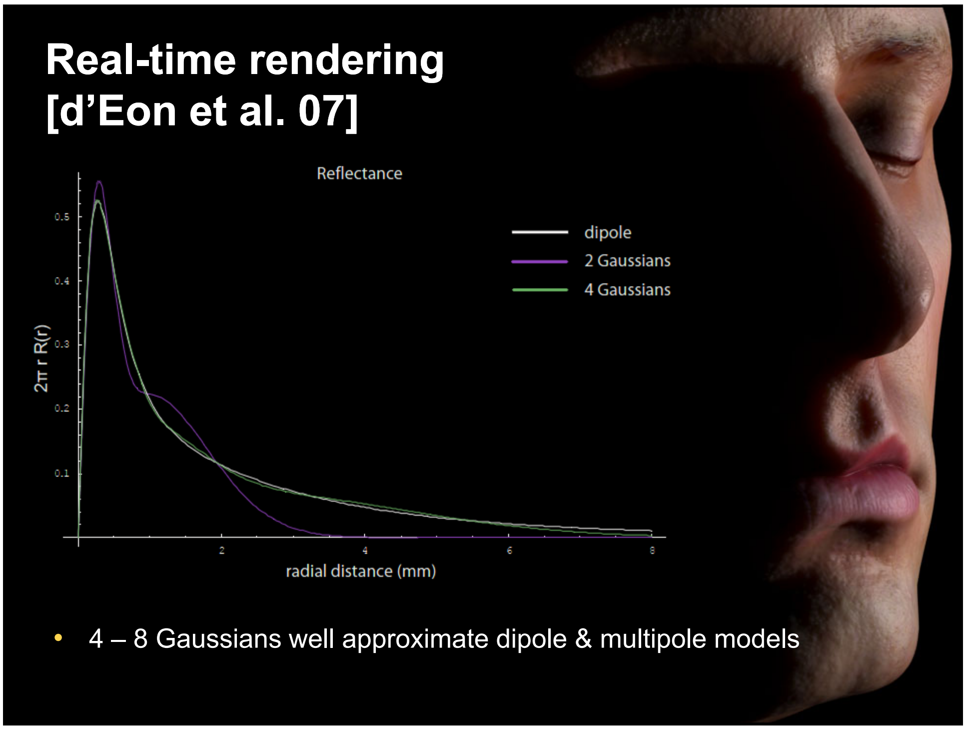
45


Diffusion models
[Jensen et al. 01]
[Donner&Jensen 05]
Offline rendering with photon mapping
– first pass scatter photons from light source
– second pass gather photons on skin surface using diffusion kernel weights

46

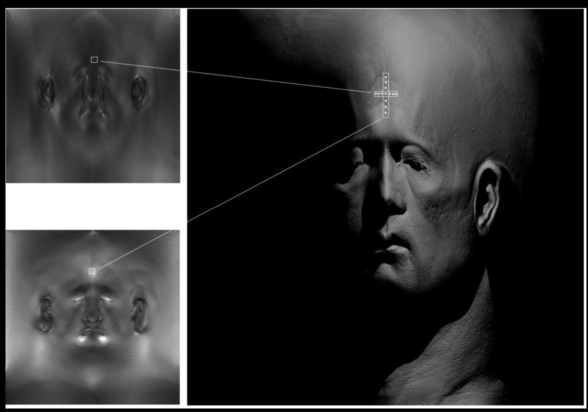

Real-time rendering
[d'Eon et al. 07]
Texture * Irradience
Gaussian blurs
Specular
• Diffusion approximated as sum of Gaussians

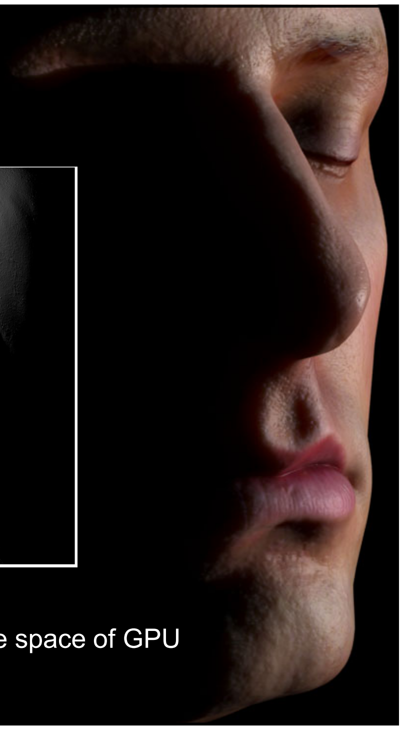
47


Real-time rendering
[d'Eon et al. 07]
Reflectance
dipole
2 Gaussians
4 Gaussians
2π r R(r)
0.5
0.4
0.3
0.2
0.1
2
4
6
8
radial distance (mm)
• 4 – 8 Gaussians well approximate dipole & multipole models

48

Real-time rendering [d'Eon et al. 07]





- Gaussian blur implemented in 2D texture space of GPU
- Need to account for texture distortion

49

Real-time rendering [d'Eon et al. 07]

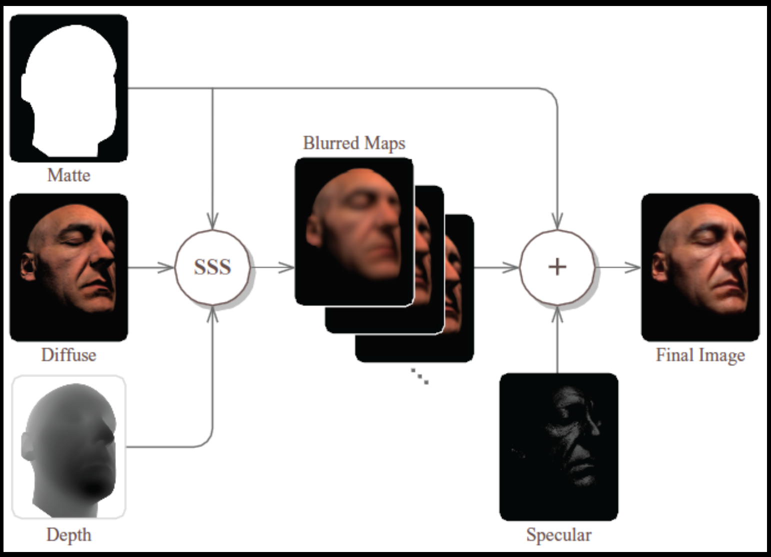
- 4 – 8 Gaussians well approximate dipole & multipole models

50

Screen-space diffusion [Jimenez et al. 09]

51

Screen-space diffusion [Jimenez et al. 09]



52

Screen-space diffusion [Jimenez et al. 09]

- Efficient rendering for many faces on screen
- Choice of rendering technique in games!

53