

Real-time Rendering of Translucent Materials with Directional Subsurface Scattering

Alessandro Dal Corso

Laurea Magistrale in Ingegneria Informatica
M.Sc. in Digital Media Engineering
T.I.M.E. Double Degree Program

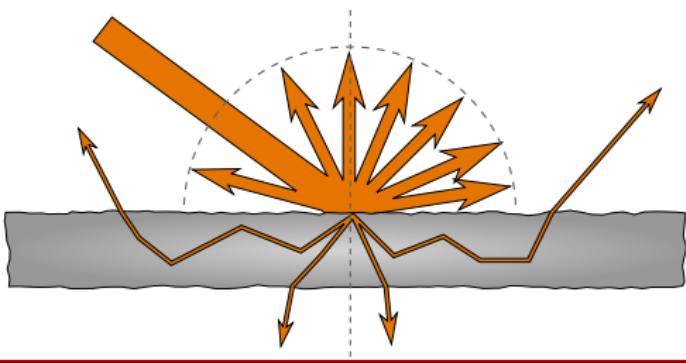
Relatore: Emanuele Menegatti
Correlatore: Jeppe Revall Frisvad

7 Luglio 2014



Introduzione

- Rendering di materiali traslucidi, come frutta, marmo, pelle
- Materiali con alto scattering sottosuperficiale
- Rendering interattivo (> 10 FPS)
- Applicazioni: computer games e visualizzazione digitale

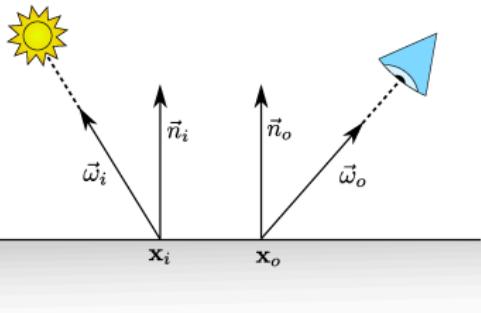
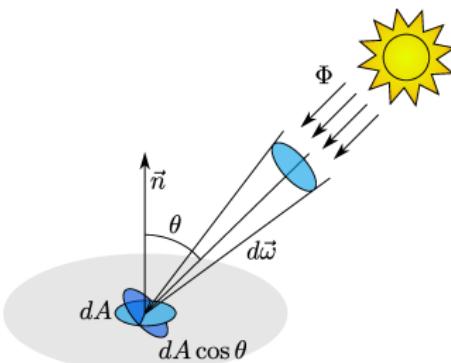


Light transport



- Definiamo una quantità fisica per descrivere la luce, detta *radianza*:

$$L(\vec{\omega}) = \frac{d^2\Phi}{d\omega dA \cos \theta}$$



- Avendo due punti su una superficie, definiamo il BSSRDF come:

$$S(x_i, \vec{\omega}_i, x_o, \vec{\omega}_o) = \frac{dL_o(x_o, \vec{\omega}_o)}{L_i(x_i, \vec{\omega}_i) \cos \theta_i d\vec{\omega}_i dA_i}$$

Che collega il flusso entrante con la radianza uscente.

The rendering equation



- Dalla definizione di BSSRDF ottieniamo la cosiddetta *rendering equation* [Jensen et al., 2001]:

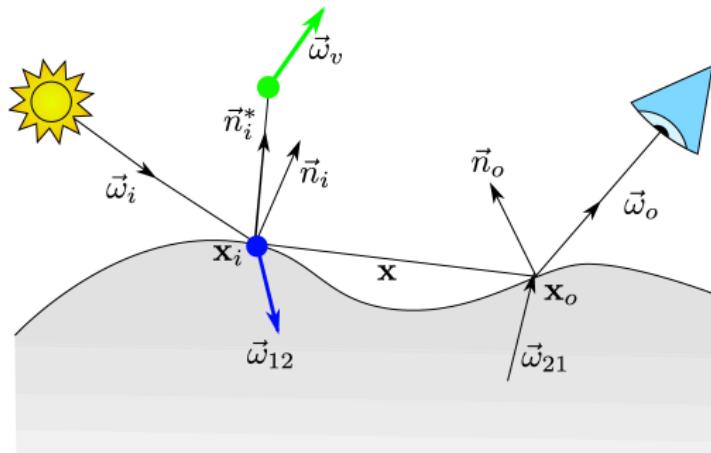
$$L_o(\mathbf{x}_o, \vec{\omega}_o) = L_e(\mathbf{x}_i, \vec{\omega}_i) + \int_A \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o, \vec{\omega}_o) L_i(\mathbf{x}_i, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i dA_i$$

- Ci sono vari BSSRDF in letteratura [Jensen et al., 2001; d'Eon et al., 2011; Frisvad et al., 2014]

BSSRDF Direzionale



- [Frisvad et al, 2014] hanno definito un nuovo BSSRDF che tiene conto della direzione della luce incidente.
- L'effetto di scattering viene modellato come due sorgenti, una *reale* e una *virtuale*.

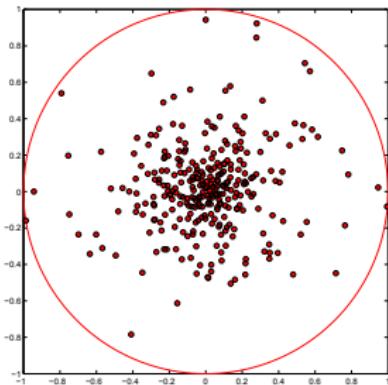
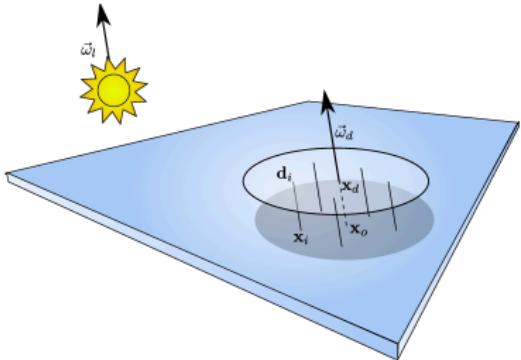


$$S_d(\mathbf{x}_i, \vec{\omega}_i, \mathbf{x}_o) = S'_d(\mathbf{x}_o - \mathbf{x}_i, \vec{\omega}_{12}, d_r) - S'_d(\mathbf{x}_o - \mathbf{x}_v, \vec{\omega}_v, d_v)$$

Metodo

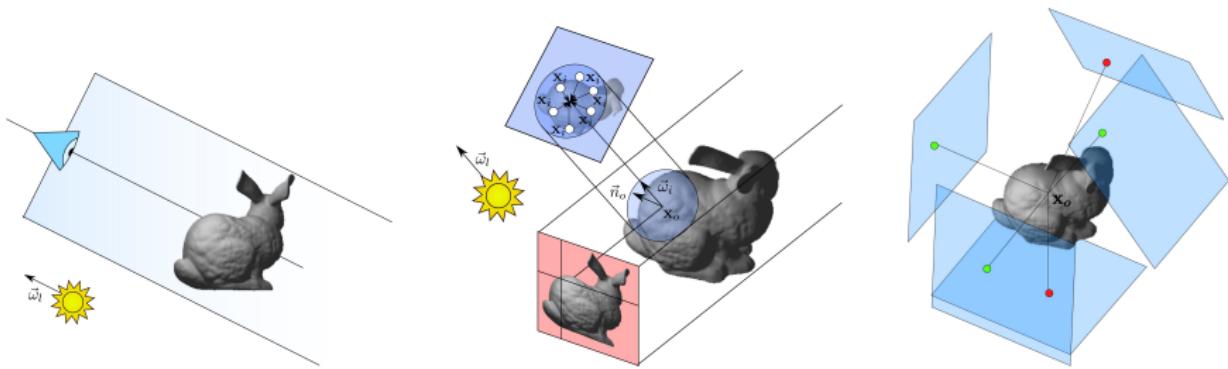
- Gli effetti di scattering hanno un range limitato, che dipende dalle proprietà ottiche del materiale.
- Creiamo un disco di campionamento di area A_c .
- Approssimiamo la rendering equation (per luci direzionali):

$$L_o(\mathbf{x}_o, \vec{\omega}_o) = L_d \frac{A_c}{N} \sum_{i=1}^N S(\mathbf{x}_i, \vec{\omega}_l, \mathbf{x}_o, \vec{\omega}_o) e^{\sigma_{tr} r_i}$$



Implementazione

- Abbiamo implementato il nostro metodo su hardware GPU, usando OpenGL 4.3 (2012) e C++.
- Otteniamo il risultato finale in tre passi:
 - 1 Rendering dell'oggetto dal punto di vista della luce
 - 2 Rendering dell'oggetto da camere direzionali e accumulo del risultato
 - 3 Ricostruzione del risultato.



Risultati (qualità)



Figure: Stanford Bunny, parametri per patata. Si notino le ombre naturalmente generate dall'algoritmo.

Risultati (qualità)



(a) $N = 100$



(b) $N = 1000$



(c) Riferimento



(d) $N = 100$



(e) $N = 1000$



(f) Riferimento

Figure: Comparazione dei risultati per patata e succo d'uva.

Risultati (qualità)



(a) $N = 50$ (ca. 90 millisecondi)



(b) Riferimento (6 ore, 16 milioni di samples)

Figure: Comparazione dei risultati per Stanford Dragon. Parametri per ketchup.

Risultati (performance)



Modello	# Δ	Numero di samples (N)			
		1	10	50	100
Bunny	10^4	2.1	5.3	19.8	38.2
Dragon	10^5	12.5	35.2	140.6	275.3
Buddha	10^6	96.7	97.7	128.0	216.0

Table: Timings in milliseconds of our method for different models and number of samples N (potato material properties). One directional light, 16 directions for rendering and reconstructing.

Conclusioni



- Abbiamo implementato una soluzione per il rendering veloce di materiali traslucidi usando BSSRDF direzionali
- Abbiamo ottenuto qualità paragonabili al path tracing
- Abbiamo ottenuto un miglioramento di 5 ordini di grandezza (millisecondi contro minuti/ore)
- Abbiamo ottenuto un metodo flessibile e applicabile a engine professionali esistenti