

Transparenta în modelul de iluminare locala

Prof. univ. dr. ing. Florica Moldoveanu

*Curs Elemente de Grafică pe Calculator – UPB, Automatică și Calculatoare
2020-2021*

MODELAREA TRANSPARENTEI (1)

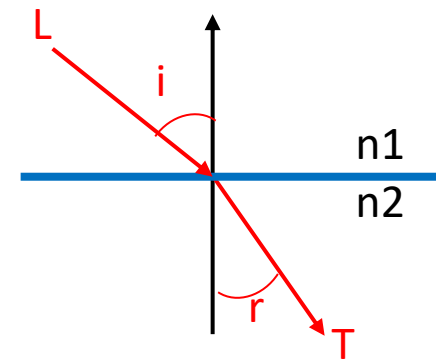
Unele obiecte ale scenei sintetizate pot fi construite din materiale transparente sau translucide.

Transmisia luminii prin obiectele transparente este speculară, în timp ce prin cele translucide este difuză.

- Atunci când lumina trece dintr-un mediu într-altul (de exemplu, din aer în apă), direcția sa se modifică datorită refracției.
- Relația dintre unghiul razei incidente, i , și cel al razei refractate, r , este dată de legea lui Snell:

$$\sin(r)/\sin(i) = n1/n2$$

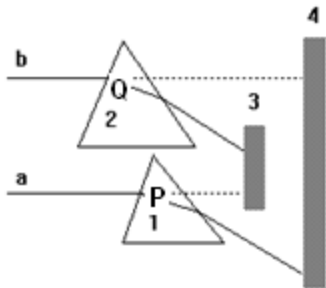
$n1$ și $n2$ sunt indicii de refracție ai celor două medii (materiale) traversate de lumină.



MODELAREA TRANSPARENTEI (2)

Indicele de refracție al unui material este dependent de lungimea de undă a luminii incidente și chiar de temperatură, dar în modelele de sinteza a imaginilor el este considerat constant.

Culoarea vizibila intr-un punct al unei suprafete transparente provine de la suprafata din spate aflata pe directia razei transmise.



3 si 4 sunt poligoane opace

1 si 2 sunt poligoane transparente, aflate in fata poligoanelor 3 si 4

a si b sunt raze de lumina provenind de la surse, incidente in P si Q

Tinand cont de refractie:

- Culoarea in P este data de poligonul 4
- Culoarea in Q este data de poligonul 3

Neglijand refractia (T are directia razei incidente)

- Culoarea in P este data de poligonul 3
- Culoarea in Q este data de poligonul 4

MODELAREA TRANSPARENTEI (3)

❑ Refracția produce, de asemenea, o distorsionare a obiectelor, asemănătoare cu aceea produsă de o proiecție perspectivă → dacă se dorește obținerea de imagini realiste, trebuie să se țină cont de refracție.

❑ Multe metode practice de modelare a transparenței ignoră refracția, astfel încât obiectele vizibile printr-o suprafață transparentă sunt cele aflate pe direcția razei incidente.

Motivul ignorării:

- reducerea volumului de calcule;
- obținerea realismului fotografic în totalitate (fără deformare)

Atunci când suprafața vizibilă într-un pixel este transparentă, culoarea în care va afișat pixelul se poate obține combinând culoarea suprafeței vizibile cu aceea a suprafeței aflată imediat în spatele său (sau a fondului), folosind următoarea formulă de interpolare:

MODELAREA TRANSPARENTEI (4)

$$I_{\lambda} = (1-kt_1)*I_{\lambda 1} + kt_1*I_{\lambda 2}$$

$I_{\lambda 1}$ este culoarea suprafeței vizibile în pixel

$I_{\lambda 2}$ este culoarea suprafeței din spatele celei vizibile, care se proiectează în același pixel

Coeficientul de transmisie, kt_1 , este o măsură a transparenței suprafeței vizibile în pixel,

$$0 \leq kt_1 \leq 1$$

$kt_1 = 0 \rightarrow$ suprafața vizibilă este opacă și deci pixelul va fi afișat în culoarea sa, $I_{\lambda 1}$

$kt_1 = 1 \rightarrow$ suprafața vizibilă este perfect transparentă și nu contribuie la culoarea pixelului.

Pixelul va fi afișat în culoarea suprafeței din spate, $I_{\lambda 2}$

Dacă $kt_1=1$ și suprafața din spatele celei vizibile este la rândul său transparentă, metoda de calcul se aplică recursiv, până când se întâlnește o suprafață opacă sau fondul.

Aproximarea liniară din model nu dă rezultate bune pentru suprafețele curbe: în apropierea siluetei unei suprafețe curbe (de exemplu, o vază sau o sticlă) grosimea materialului reduce transparența.

MODELAREA TRANSPARENTEI (5)

Solutia propusa de Kay: k_t se calculeaza in functie de normala la suprafata in punctul considerat:

$$k_t = k_{t_{\min}} + (k_{t_{\max}} - k_{t_{\min}})(1 - (1 - N_z))^m$$

unde

$k_{t_{\min}}$ și $k_{t_{\max}}$ reprezintă transparența minimă și cea maximă a suprafeței

N_z este componenta z a normalei normalizate la suprafață în punctul pentru care se calculează k_t (normala fragmentului),

m este un exponent ce caracterizează transparența. Valorile uzuale pentru m sunt 2 și 3.

$$N_z = 1 \rightarrow k_t = k_{t_{\max}} \quad I_\lambda = (1 - k_{t_{\max}}) * I_{\lambda 1} + k_{t_{\max}} * I_{\lambda 2}$$

$$N_z = 0 \rightarrow k_t = k_{t_{\min}} \quad I_\lambda = (1 - k_{t_{\min}}) * I_{\lambda 1} + k_{t_{\min}} * I_{\lambda 2}$$

Majoritatea algoritmilor de eliminare a partilor nevizibile la afișarea scenelor 3D pot fi adaptați pentru a îngloba transparența.

- ❑ In algoritmi care afișează poligoanele scenei 3D în ordinea “din spate în față” (back to front), de ex. algoritmul BSP și Pictorului, $I_{\lambda 1}$ corespunde fragmentului curent iar $I_{\lambda 2}$ este culoarea pixelului în care se afișează fragmentul.

MODELAREA TRANSPARENTEI (6)

- ❑ Adăugarea efectului de transparentă în algoritmul Z-Buffer este mai dificilă, deoarece poligoanele sunt rasterizate în ordinea în care sunt transmise în banda grafică, neținându-se cont de distanța lor față de observator.
- Nu întotdeauna fragmentul afisat într-un pixel (x,y) aparține suprafeței aflate imediat în spatele celei din care face parte fragmentul curent.
- Se poate realiza folosind mai multe buffere-imagine și generând imaginea în mai multe etape, imaginea finală obținându-se prin combinarea imaginilor poligoanelor opace cu cele ale poligoanelor transparente; nu se folosește în practică.

In OpenGL, transparenta poate fi simulată prin amestecul dintre culoarea fragmentului curent și cea a pixelului în care se afișează, folosind pentru primitivele grafice modelul (R, G, B, A): pentru fiecare fragment va rezulta o culoare (R, G, B, A), în care:

A – opacitatea culorii: 0 – transparentă; 1 – opacă

MODELAREA TRANSPARENTEI (7)

Se folosesc functiile:

`glEnable(GL_ALPHA_TEST)` – activeaza testul alfa

`glAlphaFunc(func, ref)` – specifica functia “alpha test”

`func` - functia de comparare a valorii `A` a fragmentului cu o valoare de referinta, $0 \leq \text{ref} \leq 1$;

Exemplu: `glAlphaFunc(GL_GREATER, 0.5)`

– fragmentul este afisat daca opacitatea sa este > 0.5

Daca fragmentul curent

- nu “trece” testul alfa, culoarea pixelului nu se modifica
- altfel, el va modifica culoarea pixelului, dacă trece testul “stencil” si testul de adancime

`glEnable(GL_BLEND)` - activeaza amestecul culorilor

`glBlendFunc(..)` – specifica modul de amestec al culorilor, daca fragmentul trece testul alfa.

Exemplu:

`glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA):`

$\text{noua_culoare_pixel} = \text{Culoare_fragment} * A_fragment + \text{Culoare_pixel} * (1 - A_fragment)$

- Amestecul culorilor este o operatie raster: efectuata de procesorul de rasterizare dupa testul de adancime, inainte de actualizarea valorii pixelului in bufferul imagine (frame buffer) (<https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/glBlendFunc.xhtml>).