

# 5

## Iluminări și Umbriri

### 1. Introducere

Vedem obiectele numai datorită luminii pe care o reflectă către ochiul nostru. Pe lângă aceasta, modul în care obiectele reflectă lumina ne furnizează informații importante despre forma lor tridimensională. Printre primele lucruri pe care le învață pictorii este capacitatea pe care o au luminile și umbrele de a conferi imaginii un aspect *realist*. Similar, în încercarea de a da realism în grafica pe calculator, trebuie să luăm în considerație efectele iluminării asupra obiectelor pe care le modelăm.

De fapt, creăm imagini afișând colecții de pixeli de culori și intensități adecvate. Avem nevoie de metode pentru determinarea exactă a acestor valori.

În capitolul precedent, Culoare în Grafica pe Calculator, am văzut cum determinăm culoarea unei surse de lumină pornind de la spectrul său de emisie. Am studiat de asemenea culoarea luminii reflectate de un obiect, pe baza spectrului de emisie al luminii și a spectrului de reflexie al obiectului. S-a văzut că sursa de lumină are importanța ei – o lampă de birou iluminează în jur cu totul diferit față de o lampă fluorescentă, de exemplu.

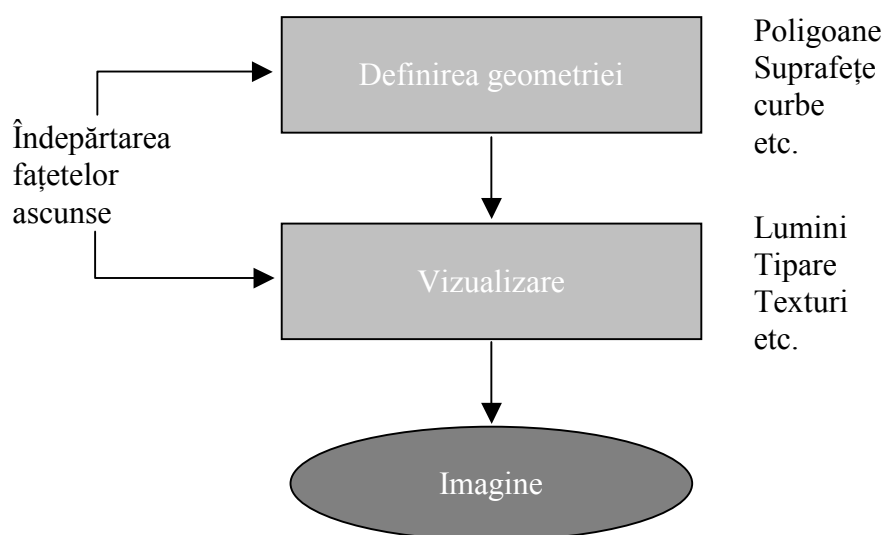
Acestea ne-au fost suficiente pentru a defini culoarea intrinsecă a unui singur obiect, izolat. La reprezentarea unei scene, însă, trebuie să ținem cont de mai mulți factori, printre care:

- Aranjamentul geometric al surselor de lumină, al obiectului și al observatorului. Experiența ne arată că sub anumite unghiuri vedem pete luminoase reflectate de obiect, și acestea nu au aceeași culoare ca și obiectul. Deci spectrul de reflexie este dependent de geometrie. De asemenea, suprafețele orientate direct spre o sursă de lumină vor apărea mai strălucitoare decât suprafețele înclinate.
- Geometria iluminării – este o sursă punctiformă sau distribuită, cât de îndepărtată este de fiecare obiect.
- Natura fizică a suprafeței – este lucioasă, mată, colorată, transparentă, netedă sau cu denivelări?
- Efectul *celorlalte* suprafețe din scenă.
- Mediul prin care trece lumina – fum, ceață, subacvatic, plouă.

În acest capitol vom studia tehnicile pentru cuantificarea acestor idei destul de vagi, astfel încât să putem calcula cum să desenăm imagini realiste. În primul rând trebuie reținut că nu există o metodă care să dea rezultatul 'corect' în orice împrejurare. Vom prefera să modelăm efectul luminii asupra obiectelor. Cât de aproape este acest model de realitatea fizică depinde de ce fenomene alegem să modelăm și de cât efort suntem dispuși să

cheltuim ca să obținem un anumit efect. Vom prefera să utilizăm un model care este suficient de adecvat pentru un scop anume. Efortul necesar pentru îndeplinirea acestui scop depinde de la situație la situație. Un sistem de pre-vizualizare 3D simplu necesită mult mai puține resurse computaționale decât o simulare a iluminării arhitecturale, de exemplu.

Figura 1 ilustrează faptul că sinteza imaginilor este o combinație în care întâi trebuie să definim *ce* plasăm în imagine, și apoi *cum* o afișăm. Aceste două etape sunt cunoscute sub denumirile de *modelare* și respectiv *vizualizare (rendering)*.



**Figura 1** Lanțul vizualizării

În mare parte, cele două faze sunt independente. Dându-se o descriere geometrică, putem genera imaginea ei în mai multe moduri. Capitolul de față se concentrează asupra modelării și vizualizării iluminării; modelarea geometrică este tratată numai în măsura în care ea afectează vizualizarea.

Pe parcursul procesului de vizualizare trebuie să luăm în considerație următoarele distincții:

- Iluminarea obiectelor de către sursele de lumină (oricare ar fi originea lor).
- Reflexia luminii pe obiecte.
- Umbrirea obiectelor (pentru afișare pe un ecran grafic).

În privința primului punct, trebuie să modelăm geometria sursei de lumină, atât dimensiunile fizice cât și distribuția intensității, precum și distribuția spectrală, respectiv culoarea ei. Pe urmă, trebuie să luăm în considerație modul în care lumina incidentă este modificată de o anumită suprafață, în special în privința culorii luminii. Pentru aceasta, vom defini un *model al reflexiei luminii*, și avem la dispoziție mai multe modalități. Al treilea punct este necesar întrucât în majoritatea cazurilor doar calculăm lumina care părăsește câteva puncte specifice de pe o suprafață și utilizăm diferite tehnici de interpolare pentru a determina lumina care pleacă de pe suprafață în alte puncte.

Există două moduri de calcul al efectelor iluminării: modele de iluminare *locală* și modele de iluminare *globală*.

Modelarea locală consideră fiecare obiect dintr-o scenă izolat, fără nici o legătură cu celelalte obiecte din scenă. Ca urmare, scenele vizualizate au aspect artificial și nu conțin umbre sau reflexii. Întrucât sunt simple și durată necesară vizualizării unui obiect nu

depinde de complexitatea de ansamblu a scenei, aceste modele de iluminare locală sunt rapide. De asemenea, această simplitate permite implementarea acestor modele în hardware-ul stațiilor grafice, lucru care conferă un plus de viteză. Modelele de iluminare locală sunt o alternativă acceptabilă atunci când scopul vizualizării este să ofere informații de bază despre forma 3D a unui obiect. Un asemenea exemplu ar fi o moleculă compusă din mii de sfere colorate. Nu este necesar doar o indicație asupra formei și este important ca vizualizarea să fie rapidă, astfel încât să putem roti și examina molecula în timp-real.

Prin contrast, modelele de iluminare globală țin cont de relațiile dintre obiectele unei scene și pot produce imagini mai realiste, încorporând umbre, reflexii și refracții. Aceste modele atunci când se urmărește redarea unui aspect al realității fizice, și când realismul este mai important decât viteza de afișare. În general, cu cât mai realist este un model, cu atât este mai costisitor computațional.

În primul capitol vom dezvolta un model simplu de reflexie locală, care ia în considerație:

- Iluminarea de la o sursă punctiformă de lumină
- Iluminarea de la lumina ambientă
- Reflexia difuză
- Reflexia speculară (direcționată).

În capitolele următoare vom vedea cum putem utiliza acest model pentru a produce imagini umbrite ale obiectelor și cum putem îl putem rafina pentru a include efectele datorate gradului de finisare al suprafeței, cum ar fi texturarea, precum și cum să tratăm transparența.

Apoi ne vom concentra asupra problemei modelării reflexiei globale a luminii. Vom discuta despre două dintre cele mai cunoscute tehnici, *ray-tracing* și *radiosity*, care modelează două aspecte destul de diferite ale reflexiei luminii.

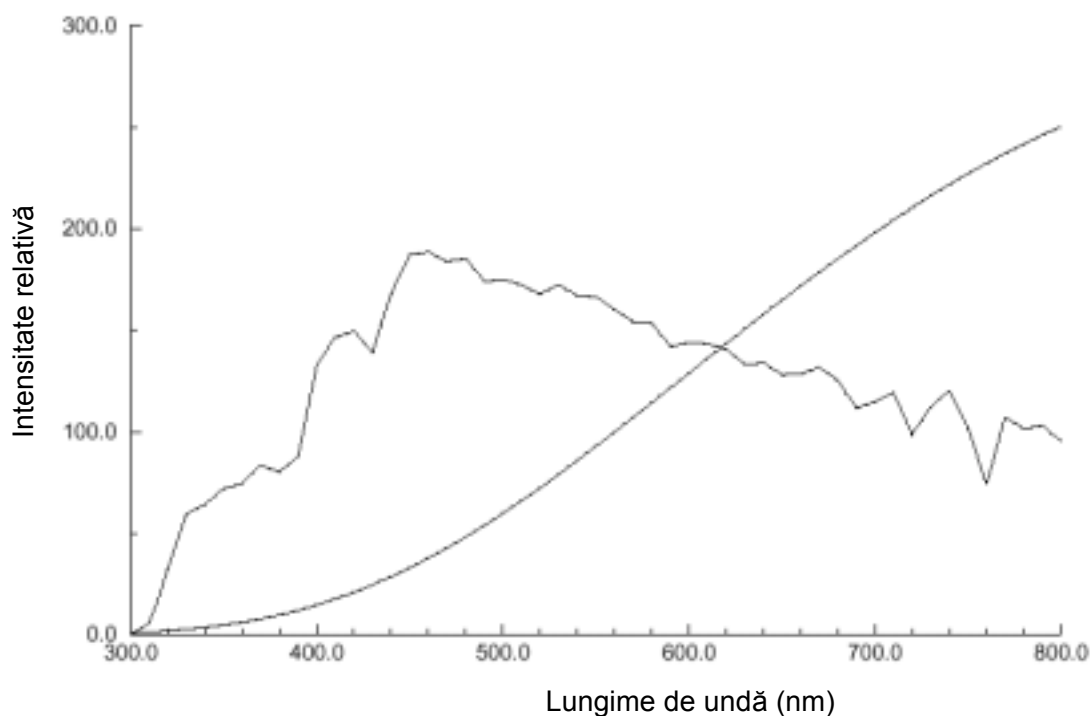
## 2. Modele de reflexie locală

### 2.1. Surse de lumină

Există două tipuri de bază de surse de lumină: emițători și reflectori.

#### 2.1.1. Emițători de lumină

Orice obiect care emite lumină, cum este soarele, un bec incandescent sau un tub fluorescent este un emițător. Printre caracteristicile emițătorilor se includ *intensitatea* generală a luminii și *spectrul* său, care determină culoarea. În Figura 2 sunt prezentate spectrele unor surse uzuale.



**Figura 2** Spectrul solar și al unui bec cu incandescență

Mai putem clasifica emițătorii de lumină în două categorii. Atunci când sursa de lumină este suficient de îndepărtată de o suprafață, o putem considera sursă *punctiformă* și toate razele de lumină întâlnesc suprafața sub același unghi. Atunci când sursa de lumină este în apropierea suprafeței, razele ajung în diferite puncte de pe suprafață sub unghiuri diferite, după cum se vede în Figura 3.

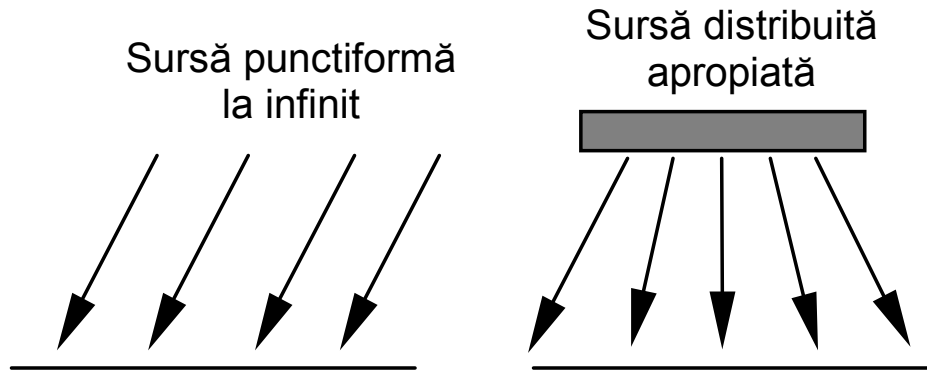


Figura 3 Surse de lumină îndepărtată și apropiată.

Mai putem lua în considerație geometria sursei de lumină. Din acest punct de vedere, putem clasifica sursele de lumină în *punctiforme* și *distribuite* (un singur punct al obiectului este iluminat sub mai multe unghiuri de la aceeași sursă de lumină, vezi Figura 4).

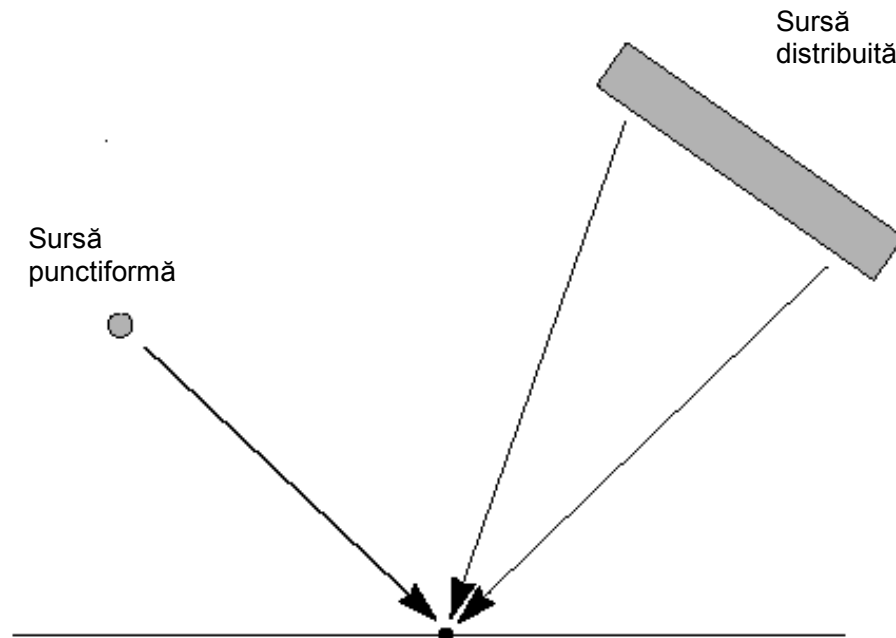


Figura 4 Sursă punctiformă și sursă distribuită.

Întrucât orice sursă fizică de lumină ocupă un anumit spațiu, toate sursele de lumină sunt distribuite! Putem însă să reducem din complexitate, acceptând compromisul că unele surse mici sunt punctiforme.

### 2.1.2. Reflectori de lumină

Într-o scenă reală, chiar dacă pe unele obiecte lumina nu cade direct de la o sursă, acele obiecte vor fi totuși iluminate de către lumina reflectată de pereți, tavan ș.a.m.d. Deci reflectori de lumină sunt toate celelalte obiecte dintr-o scenă care nu sunt emițători. Deși este evident că un emițător poate să și reflecte lumina, cantitatea reflectată este atât de mică în comparație cu lumina emisă încât porțiunea reflectată poate fi ignorată. Culoarea unui reflector depinde de spectrul luminii incidente și de spectrul de reflectanță al obiectului.

În plus, pe lângă aceste reflexii directe, există și reflexii indirecte – reflexii la reflexii etc. Aceste reflexii multiple produc într-o scenă o iluminare generală, numită lumină ambiantă, care va fi tratată mai târziu.

## 2.2. Tipuri de reflexii

Trebuie să dezvoltăm un model care să ne permită calcularea cantității de lumină reflectate de către diferitele obiecte dintr-o scenă. Reflexia se petrece în două moduri:

- *Reflexie difuză* – lumina este împrăștiată uniform de suprafața obiectului (Figura 5a), și suprafața are aspect mat. Suprafața absoarbe lumina incidentă și o re-radiază numai pe lungimi de undă preferențiale. De exemplu, un obiect verde arată verde când este iluminat cu lumină albă deoarece absoarbe toate lungimile de undă cu excepția verdei, pe care îl reflectă.
- *Reflexie direcționată (speculară)* – o proporție mare din lumina incidentă este reflectată într-un domeniu restrâns de unghiuri, iar obiectul arată lucios (Figura 5b). Ca o primă aproximație, culoarea suprafeței nu afectează culoarea luminii reflectate.

În practică, majoritatea materialelor prezintă ambele efecte, în proporții diferite, iar reflexia totală este de genul celei prezentate în Figura 5c.

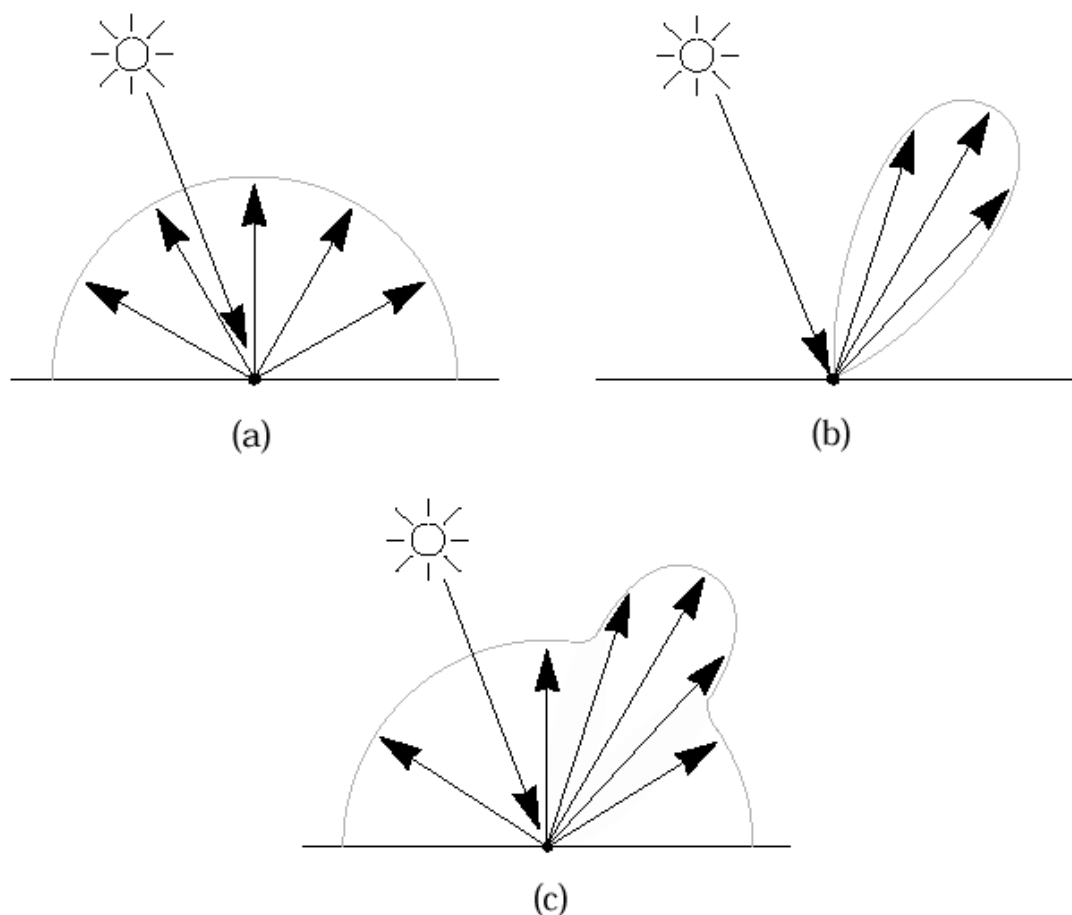


Figura 5 Reflexii difuză, speculară și combinată

## 2.3. Geometria surselor de lumină

Să considerăm o singură sursă de lumină, modelată sub forma unui punct, plasat la distanță infinită de obiectul iluminat. Evident, cantitatea de lumină reflectată de o suprafață va depinde de orientarea ei față de sursa de lumină. Descriem aceasta cantitativ prin intermediul a doi *vectori*, prezentați în Figura 6:

- Vectorul *normală*  $\mathbf{N}$  – este un vector normal la suprafață (formează un unghi drept). De exemplu, pentru o suprafață plasată în planul  $xz$ , vectorul normală va fi paralel cu axa  $y$ .
- Vectorul *luminii incidente*  $\mathbf{L}$  – indică *înspre* sursa de lumină.

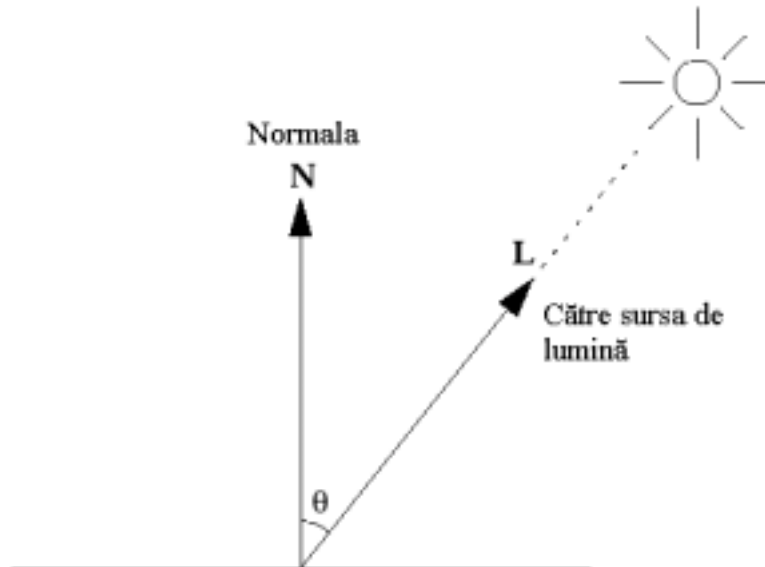


Figura 6 Descrierea unei suprafețe și a orientării sale

Întrucât ne sunt necesare numai *direcțiile* acestor vectori pentru a preciza orientarea suprafeței față de sursa de lumină este convenabil (și convențional) să utilizăm vectori *unitate*, așa că vom presupune că  $\mathbf{N}$  și  $\mathbf{L}$  au fost normalizați. Unghiul  $\theta$  dintre  $\mathbf{N}$  și  $\mathbf{L}$  se numește *unghi de incidență*.

## 2.4. Lumina ambientă

Reflexiile multiple ale luminii într-o scenă produc împreună o iluminare uniformă, numită *lumină ambientă*. De fapt, este vorba despre o simplificare. În majoritatea cazurilor putem presupune existența unui nivel constant al luminii ambiante într-o scenă. În realitate, totuși, efectele luminii ambiante sunt mult mai subtile.

Intensitatea  $I$  a reflexiei difuze într-un punct de pe un obiect iluminat numai de lumina ambientă cu intensitatea  $I_a$  este:

$$I = k_d I_a \quad (2-1)$$

unde  $k_d$  se numește coeficient de reflexie difuză și este o constantă subunitară. Valoarea sa depinde de proprietățile suprafeței materialului respectiv. O suprafață care reflectă puternic va avea  $k_d$  apropiat de 1. De notat că în literatură se mai preferă notarea lui cu  $k_a$ .

Am putea utiliza Ecuația 2-1 pentru calculul iluminării unui obiect, dar fiecărui punct de pe suprafața obiectului i se va atribui aceeași intensitate. Acest lucru nu este realist, noi știm că în practică umbrirea obiectelor *nu* este constantă. Dacă lucrurile ar fi stat așa, nu am fi capabili să percepem “tridimensionalitatea” unui obiect. În Figura 7 se arată cum ar trebui să apară un cub dacă ar fi iluminat numai de lumina ambientă. Vom obține efecte mult mai realiste dacă luăm în considerație surse de lumină localizate, cum ar fi sursele punctiforme și dacă ținem cont de orientarea suprafețelor față de aceste surse.

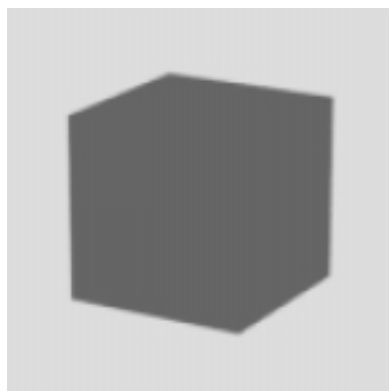


Figura 7 Obiect iluminat numai de lumina ambiantă

## 2.5. Reflexia difuză

Cantitatea de lumină reflectată de o suprafață depinde de cantitatea recepționată de la sursa de lumină, care la rândul său depinde de intensitatea iluminării, și de orientarea suprafeței față de sursa de lumină. Dacă  $N$  și  $L$  au aceeași direcție ( $\theta = 0$ ), reflexia va fi maximă. Dacă cei doi vectori sunt ortogonali ( $\theta = 90^\circ$ ), reflexia va fi minimă. Relația exactă este exprimată de Legea Cosinusului a lui Lambert, care spune că intensitatea efectivă  $I_e$  incidentă pe o suprafață iluminată cu intensitatea  $I_p$  incidentă la unghiul  $\theta$  este:

$$I_e = I_p \cos \theta$$

Exprimând reflectivitatea difuză prin  $k_d$ , cantitatea de lumină reflectată difuz de o suprafață iluminată cu intensitatea  $I_p$  incidentă sub unghiul  $\theta$  este:

$$I = k_d I_p \cos \theta \quad (2-2)$$

Întrucât  $N$  și  $L$  sunt normalizați, produsul lor scalar ne dă chiar unghiul dintre ei:

$$\cos \theta = N \cdot L$$

deci putem rescrie Ecuația 2-2 sub formă vectorială:

$$I = k_d I_p (N \cdot L) \quad (2-3)$$

Putem acum să combinăm efectele luminii ambiante cu cele ale sursei punctiforme pentru a obține modelul de iluminare:

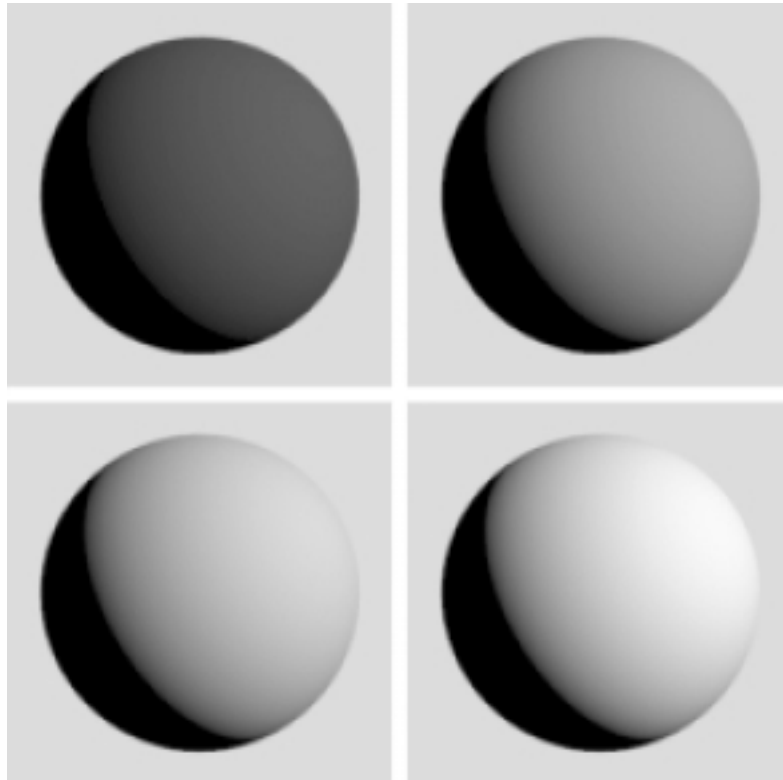
$$I = \text{componenta ambiantă} + \text{componenta difuză}$$

sau,

$$I = k_a I_a + k_d I_p (N \cdot L) \quad (2-4)$$

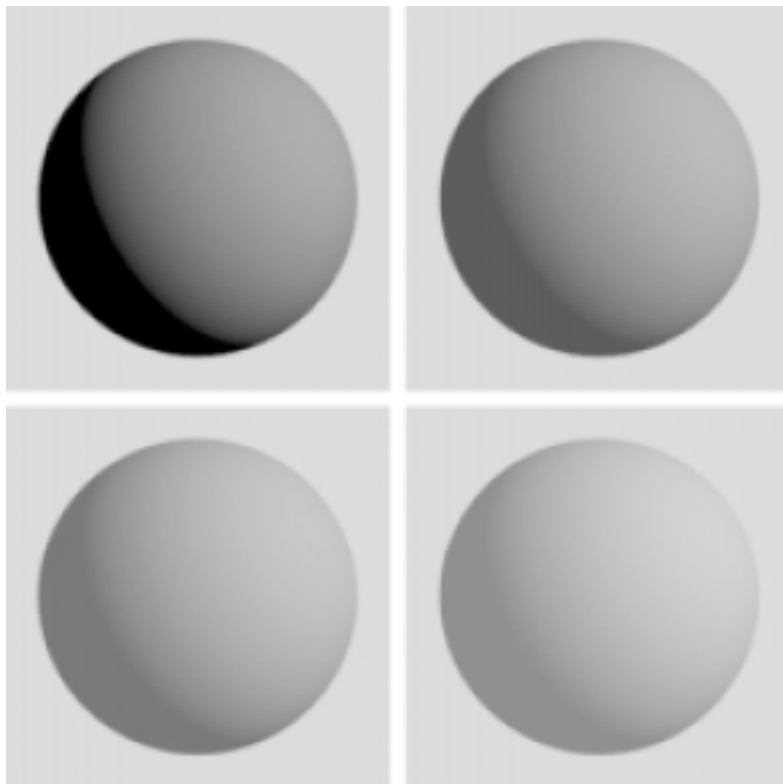
În Figura 8 sunt prezentate patru sfere ale căror suprafețe au fost umbrite conform modelului de reflexie difuză dat de Ecuația 2-3. De la stânga la dreapta și de sus în jos,  $k_d=0.1, 0.4, 0.7$  și  $0.95$ .





**Figura 8** Sfere umbrite pe baza modelului de reflexie difuză

În Figura 9 avem din nou patru sfere, dar de data aceasta suprafețele sunt umbrite conform Ecuației 2-4, care include și lumina ambiantă. Fiecare sferă are  $k_d = 0.4$ , dar  $I_a = 0\%$ ,  $20\%$ ,  $40\%$  și respectiv  $60\%$  din  $I_p$ , de la stânga la dreapta și de sus în jos.



**Figura 9** Sfere umbrite utilizând lumina ambiantă și reflexia difuză

De notat faptul că în Ecuația 2-4 intensitatea luminii reflectate de o suprafață este independentă de distanța de la suprafață până la sursa punctiformă. Aceasta înseamnă că dacă două obiecte au suprafețe paralele și dacă din punctul de observație suprafețele se suprapun, ele vor fi umbrite la fel și se vor confunda într-o singură suprafață. De asemenea, în acest model nu s-au luat în considerație culoarea luminii și nici a suprafeței. Vom reveni asupra acestui subiect.

## 2.6. Poziția sursei de lumină

Aceasta se poate modela observând că intensitatea luminii scade proporțional cu pătratul distanței parcurse. Dacă lumina de la o sursă cu intensitatea  $I_p$  parcurge o distanță  $d$ , intensitatea efectivă este  $I_p/4\pi d^2$ . Putem îngloba aceasta în Ecuația 2-4:

$$I = k_d I_a + \frac{k_d I_p (N \bullet L)}{4\pi d^2}$$

(De notat că termenul corespunzător luminii ambiante  $k_d I_a$  nu depinde de distanță, întrucât prin definiție sursa este *peste tot*). Totuși, în practică acest model nu dă rezultate realiste. În aplicații tipice de modelare, fie vom avea o sursă punctiformă plasată la infinit, deci  $d$  este infinit, fie avem o sursă de lumină la distanță finită. În oricare din cazuri, în mod obișnuit vom avea valori ale lui  $d$  care diferă prea puțin între obiecte, și aceasta, combinat cu domeniul finit de valori pentru pixeli, poate să însemne că efectul termenului  $d^2$  poate fi prea mare. Experimental s-a demonstrat că un model mai bun ar fi să se utilizeze o aproximare liniară a scăderii intensității cu distanța:

$$I = k_d I_a + \frac{k_d I_p (N \bullet L)}{(d + d_0)} \quad (2-5)$$

unde  $d$  este distanța de la sursa de lumină la obiect, iar  $d_0$  este o constantă determinată experimental (în grafica pe calculator avem de-a face cu o sumedenie de „constante determinate experimental”).

## 2.7. Reflexia speculară

Un obiect modelat numai pe baza reflexiei difuze are un aspect mat. În realitate, multe suprafețe sunt *lucioase*. Sub anumite unghiuri de observație, o suprafață lucioasă reflectă o parte importantă din lumina incidentă – aceasta se numește reflexie *speculară*. Atunci când se întâmplă aceasta, vedem o pată strălucitoare pe obiect, cu aceeași culoare ca și lumina incidentă.

Mai întâi vom considera un reflector perfect, cu ar fi o oglindă bine lustruită. În Figura 10, vectorul unitate  $\mathbf{R}$  reprezintă direcția reflexiei speculare.  $\mathbf{V}$  este direcția din care este observată suprafața. În cazul unui reflector perfect, unghiul de incidență este egal cu cel de reflexie, astfel încât reflexia speculară poate fi observată numai atunci când  $\mathbf{V}$  și  $\mathbf{R}$  coincid, adică  $\phi = 0^\circ$ .

În realitate, materialele nu se comportă atât de exact. Lumina care cade sub un unghi de incidență  $\theta$  este reflectată într-un con centrat pe  $\phi$ , unde intensitatea are valoarea maximă. Intensitatea reflexiei speculare scade puternic pe măsură ce crește unghiul  $\phi$  dintre direcția reflexiei ideale  $\mathbf{R}$  și direcția de observație  $\mathbf{V}$ . Avem nevoie de o modalitate de-a include acest efect în modelul nostru de iluminare.

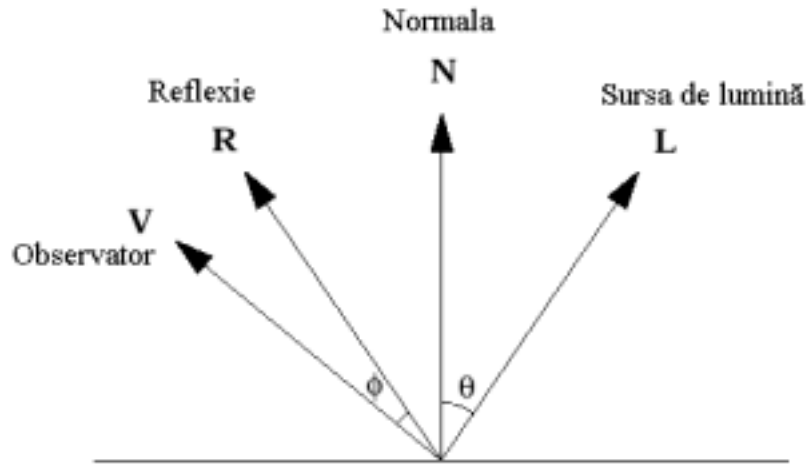


Figura 10 Cei patru vectori de referință

### 2.7.1. Modelul Phong

Phong a dezvoltat un model empiric în care intensitatea reflexiei speculare este proporțională cu  $\cos^n \phi$ . În Figura 11 se prezintă graficul funcției  $\cos^n \phi$  pentru  $-\pi/2 \leq \phi \leq \pi/2$ .  $n$  reprezintă proprietățile suprafeței și ia valori în domeniul  $[1,200]$ . Pentru un reflector perfect,  $n=\infty$ , iar pentru un reflector foarte slab  $n=1$ .

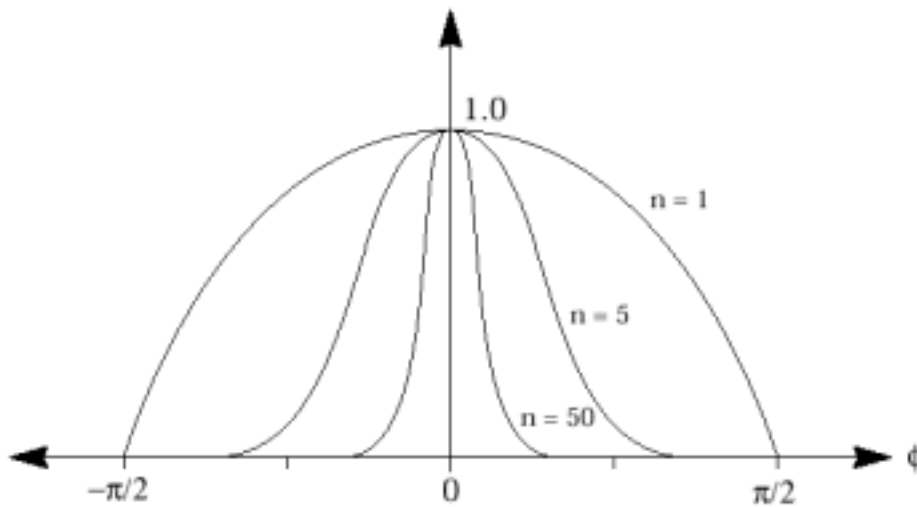
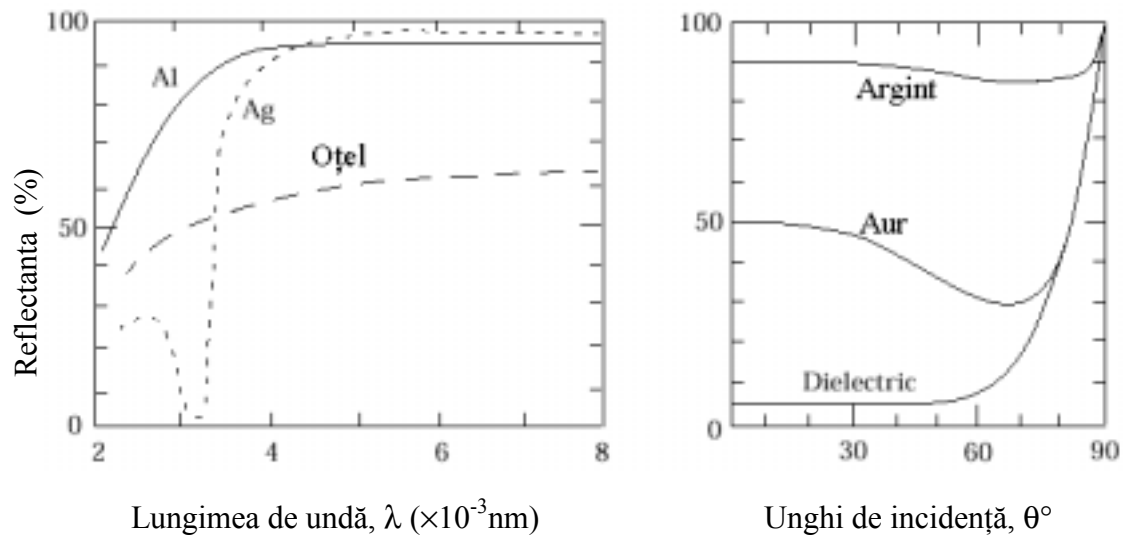


Figura 11 Comportarea lui  $\cos^n \phi$

Pentru materiale reale, cantitatea de lumină reflectată specular depinde și de unghiul de incidență. În general, intensitatea reflexiei speculare crește cu creșterea unghiului de incidență. Relația exactă dintre ele depinde de proprietățile concrete ale suprafeței și reprezentăm aceasta printr-o funcție de reflexie speculară  $W(\theta)$ , cu valori în domeniul  $[0,1]$ . Comportarea exactă a lui  $W(\theta)$  variază de la o suprafață la alta. În Figura 12 este reprezentată dependența de lungimea de undă a luminii incidente și de unghiul de incidență pentru diferite materiale. De exemplu, în cazul sticlei, reflexia speculară crește de la aproape zero la intensitatea maximă pe măsură ce unghiul de incidență crește de la  $0^\circ$  la  $90^\circ$ . Alte materiale prezintă o variație mult mai mică și de aceea  $W(\theta)$  este deseori înlocuit cu o constantă  $k_s$ , determinată experimental, numită coeficient de reflexie speculară.



**Figura 12** Curbe de reflexie

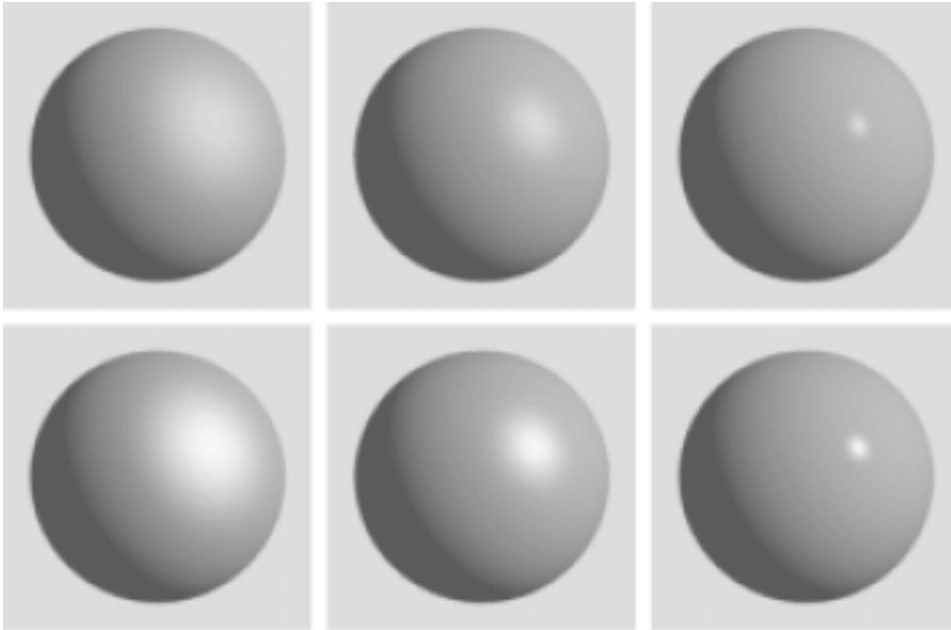
Acum putem să includem componenta de reflexie speculară în modelul nostru de iluminare din Ecuația 2-5:

$I = \text{componenta ambientă} + \text{componenta difuză} + \text{componenta speculară}$   
sau

$$I = k_d I_a + \frac{I_p}{(d + d_0)} [k_d (N \cdot L) + k_s \cos^n \phi] \quad (2-6)$$

Întrucât R și V sunt normalizați, putem rescrie Ecuația 2-6 sub formă vectorială:

$$I = k_d I_a + \frac{I_p}{(d + d_0)} [k_d (N \cdot L) + k_s (R \cdot V)^n] \quad (2-7)$$



**Figura 13** Sfere iluminate cu modelul lui Phong

În Figura 13 este prezentată o sferă iluminată conform Ecuației 2-7, pentru diferite valori ale lui  $k_s$  și  $n$ . În fiecare caz,  $k_d=0.4$  și  $I_a=20\%$  din  $I_p$ . Primul rând are  $k_s=0.25$ , iar al doilea  $k_s=0.5$ . De la stânga la dreapta,  $n=3, 10$  și respectiv  $100$ .

## 2.8. Mai multe surse de lumină

Evident, putem extinde Ecuația 2-7 pentru a include mai multe surse de lumină. Fiecare sursă va determina o reflexie difuză și una speculară, deci trebuie să *însușim* aceste efecte. Dacă avem  $n$  surse punctiforme diferite, modelul nostru devine:

$$I = \text{ambient} + \sum_{i=1}^n \text{difuz}_i + \sum_{i=1}^n \text{specular}_i$$

Însumarea poate însă să ridice probleme de depășire în valoarea lui  $I$ , deși aceeași problemă poate să apară și în cazul unei singure surse de lumină. În scopul afișării, vom limita domeniul lui  $I$ , de obicei între 0 și 1 în formă normalizată. Există mai multe modalități de evitare a depășirii. Mai întâi, putem ajusta parametrii de reflexie sau intensitățile surselor de lumină. Dacă ajungem la concluzia că o asemenea abordare nu este potrivită, putem ajusta valoarea calculată a lui  $I$ . În acest caz putem:

- să ajustăm valoarea *local*, de exemplu limitând valoarea la domeniul necesar, sau
- să ajustăm valoarea *global*, luând în considerare toate intensitățile calculate pentru imaginea noastră și să scalăm toate valorile astfel încât să nu avem valori care să depășească domeniul.

A doua metodă este în general de preferat, întrucât evită erorile care pot să ducă la efectul de bandă Mach.

## 2.9. Culoare

Până acum am presupus că avem de-a face cu o sursă de lumină monocromă, și am ignorat faptul că obiectele au culoare. Vom exprima culoarea utilizând un *model de culoare*, care ne va permite să atribuim culorilor valori numerice. Un model uzual simplu este

RGB, în care o culoare este reprezentată ca un amestec de trei culori primare, roșu, verde și albastru, corespunzătoare ecranului, după cum se vede în Figura 14.

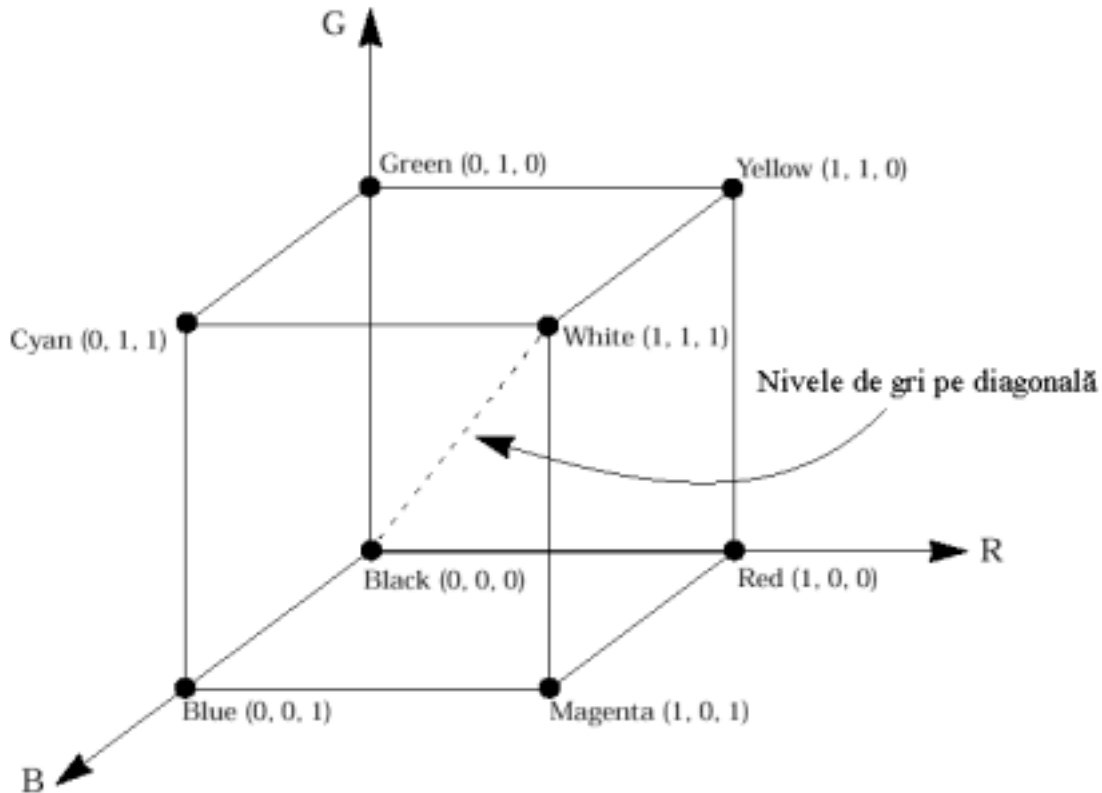


Figura 14 Modelul de culoare RGB

Pentru a lua în considerație efectul iluminării colorate asupra obiectelor colorate, trebuie să aplicăm modelul nostru de iluminare pentru fiecare componentă a modelului de culoare utilizat. De exemplu, dacă utilizăm modelul RGB, vom trata separat componentele roșie, verde și albastră.

Putem exprima culoarea suprafeței obiectelor printr-un coeficient de reflexie difuză dependent de culoare ( $k_{dR}$ ,  $k_{dG}$ ,  $k_{dB}$ ). De asemenea, avem componente de culoare pentru lumina incidentă,  $I_{pR}$ ,  $I_{pG}$ ,  $I_{pB}$ . Deci, pentru componenta roșie modelul nostru devine:

$$I_R = k_{dR} I_{aR} + \frac{I_{pR}}{(d + d_0)} \left[ k_{dR} (N \cdot L) + k_s (R \cdot V)^n \right]$$

și similar pentru verde și albastru. Presupunem că pata datorată reflexiei speculare va fi de aceeași culoare ca și sursa de lumină, deci  $k_s$  nu depinde de culoare. Având cele trei intensități,  $I_R$ ,  $I_G$  și  $I_B$ , și fiecare componentă în domeniul acceptat de display, le putem utiliza pentru controlul direct al monitorului RGB. Valori din afara domeniului se tratează după cum s-a prezentat mai înainte, prin limitare sau scalare globală. De asemenea, mai putem scala local culoare prin dividerea fiecărei componente cu valoarea maximă, o metodă care păstrează tonalitatea și saturația, dar modifică luminozitatea.

Deși utilizarea modelului RGB duce de obicei la rezultate rezonabile, este o simplificare grosolană a interacțiunilor dintre lumină și obiecte. Pentru rezultate mai bune trebuie să aplicăm modelul de iluminare cu mai mult de trei culori primare. În loc să ne restrângem la un model de culoare cu trei componente, o formă mai generală a modelului de iluminare ar putea fi scrisă ca:

$$I_{\lambda} = k_{d\lambda} I_{a\lambda} + \frac{I_{p\lambda}}{(d + d_0)} [k_{d\lambda} (N \cdot L) + k_s (R \cdot V)^n]$$

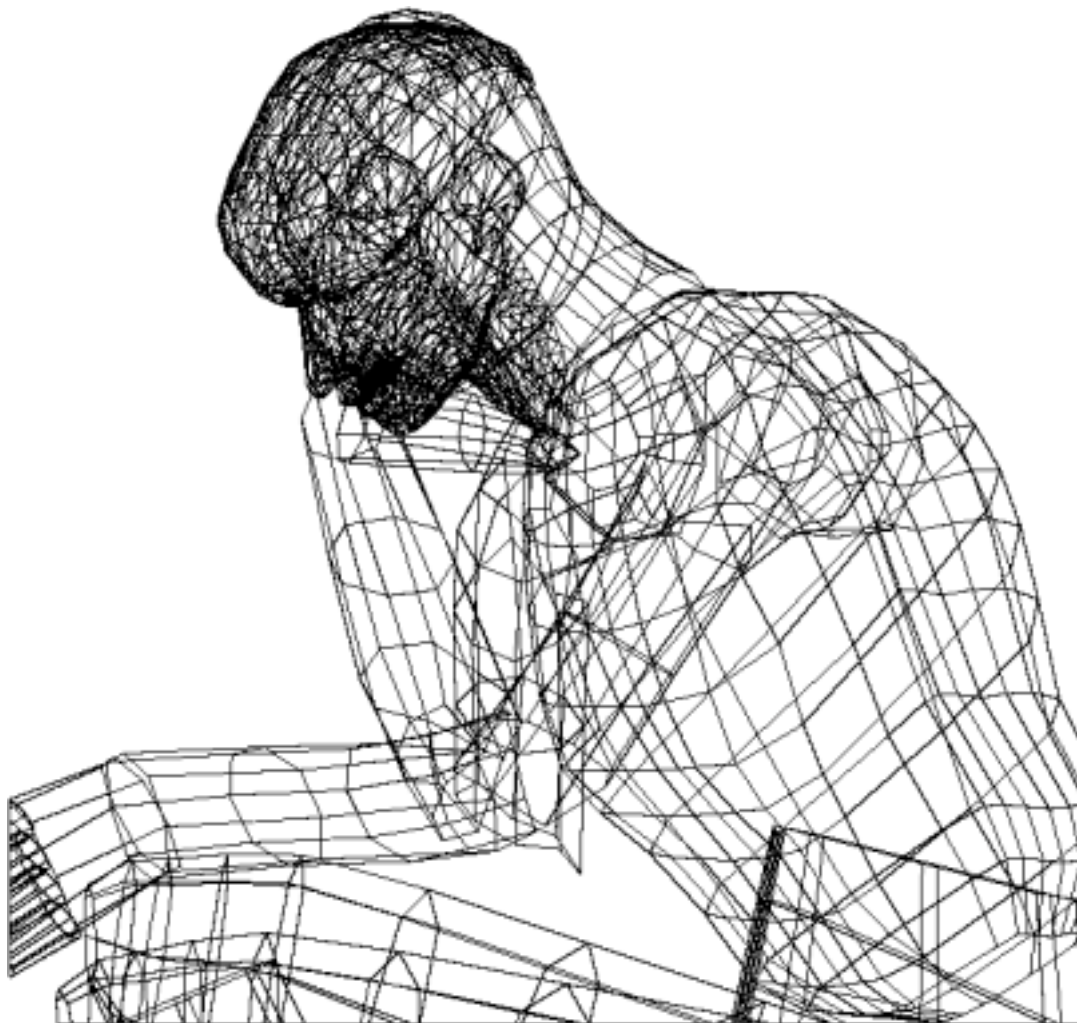
în care oricare termeni dependenți de lungimea de undă sunt precizați prin indicele  $\lambda$ . Însă cum utilizăm aceste *componente spectrale* pentru a specifica un triplet RGB, de care avem nevoie pe display? O modalitate ar fi să le convertim în spațiul CIE XYZ și apoi în spațiul RGB al monitorului, ceea ce necesită cunoașterea aspectelor cromatice ale fosforului din monitor. Mai multe detalii pot fi găsite în capitolul precedent, *Culoarea în Computer Graphics*.

Cu toate că această tehnică este mai corectă din punct de vedere fizic, să nu uităm că modelele de reflexie discutate până acum sunt în principal bazate pe observații empirice: ele sunt construite astfel încât rezultatele să arate rezonabil. În general deci ar fi nepotrivit să utilizăm o asemenea metodă. Cu toate acestea, când vom discuta modelele globale de reflexie vom vedea că aceste modele sunt mai apropiate de realitatea fizică și ar beneficia de utilizarea mai multor componente spectrale. Din păcate, deși aceasta este adevărat, în practică se utilizează cel mai des modelul RGB.

### 3. Umbrirea suprafețelor

Modelele de iluminare dezvoltate până aici ne pot calcula intensitatea și culoarea luminii care pleacă de pe o anumită suprafață într-un punct dat. Cum vom calcula intensitățile pentru întregul obiect?

Suprafețele curbe și obiectele complexe de obicei sunt approximate utilizând rețele de poligoane, după cum se vede în Figura 15.



**Figura 15** Un model poligonal

Putem aplica modelul nostru de iluminare fiecărei fațete poligonale, însă avem la dispoziție mai multe modalități: am putea calcula o singură intensitate pentru fiecare poligon, de exemplu, pe baza centrului poligonului. Altă variantă ar fi să aplicăm modelul fiecărui punct al poligonului, astfel încât intensitatea să nu fie aceeași. A doua abordare va avea rezultate mai realiste, însă ar fi foarte costisitoare. Ca o alternativă, calculăm intensitatea în câteva puncte ale poligonului, iar apoi determinăm intensitatea în celelalte puncte utilizând metode de interpolare.

#### 3.1. Umbrirea constantă

Umbrirea constantă are o aparență nerealistă atunci când două sau mai multe poligoane au muchii comune și fiecare este umbricit diferit. În unele aplicații, cum ar fi modelarea cu element finit, acest efect este chiar dorit. Însă în cazul unei modelări realiste, modificarea



bruscă a iluminării în zona muchiilor este deranjantă și nerealistă. În plus, prin umbrirea constantă fiecare poligon care aproximează o suprafață devine vizibil ca entitate de sine stătătoare, cu toate că este posibil să reducem acest efect utilizând mai multe poligoane mai mici pentru aproximarea suprafeței.

### 3.1.1. Efectul de bandă Mach

Vederea umană are o abilitate deosebită în detectarea muchiilor – chair și acolo unde nu există. Diferențele abrupte în umbrirea a două fațete sunt percepute chiar mai mari decât sunt ele în realitate; este vorba despre un fenomen denumit efectul de bandă Mach.

În Figura 16 se prezintă (exagerat, pentru ilustrare), diferențele dintre intensitatea *percepută* și cea *reală* la suprafața unui obiect.

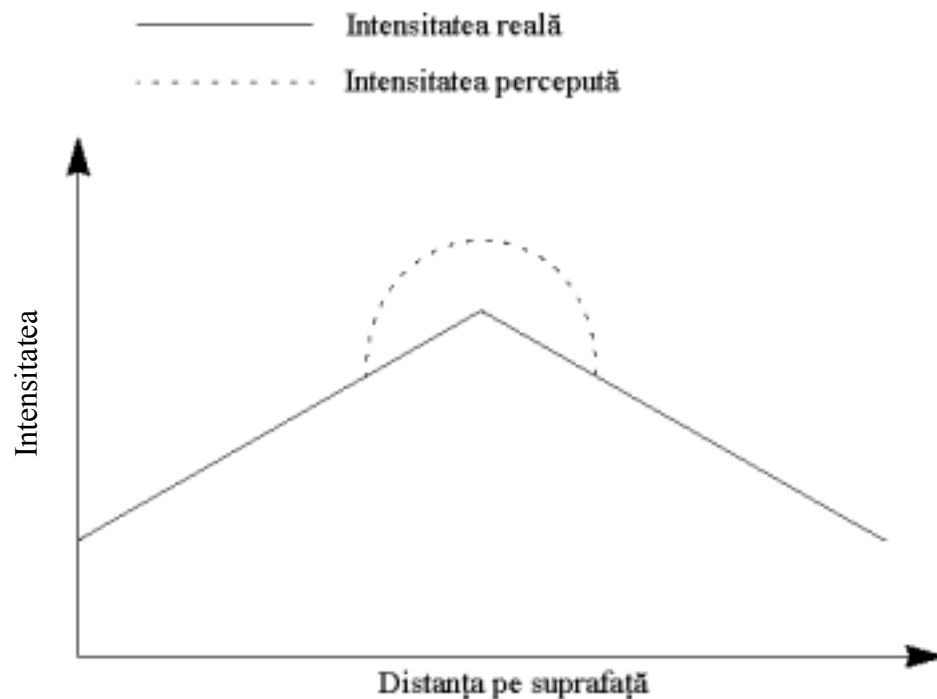
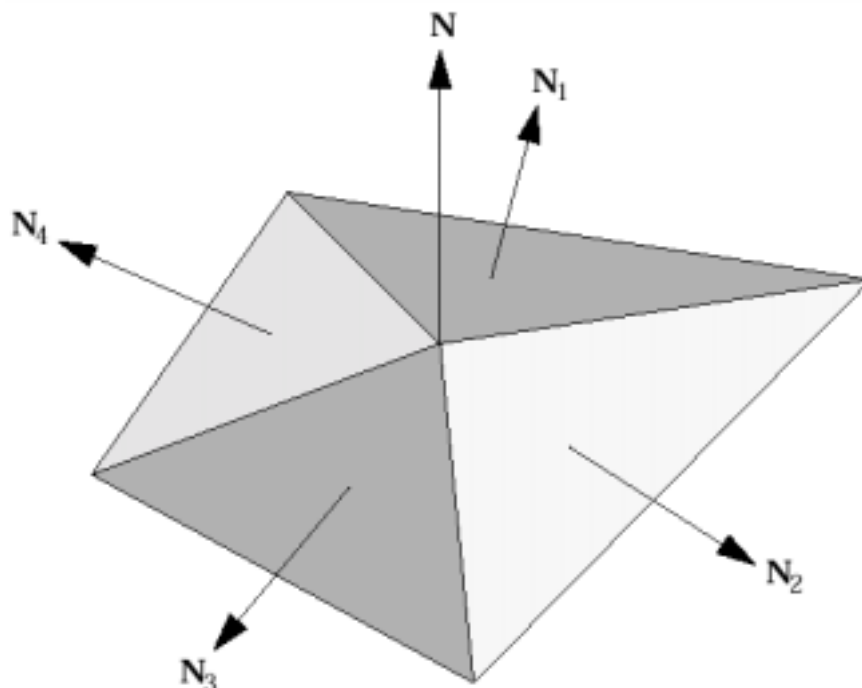


Figura 16 Efectul de bandă Mach

Pentru reducerea efectului de bandă Mach și eliminarea salturilor bruște de intensitate între fațete, este necesar să netezim aceste schimbări, astfel încât o fațetă să anticipeze umbrirea celei care urmează. Vom studia trei tehnici: *interpolarea intensității*, *interpolarea vectorului normală* și *interpolarea produsului scalar*.

## 3.2. Interpolarea intensității

În această metodă, se calculează valorile intensităților în vârfurile poligonului, iar apoi acestea sunt interpolate, pentru a determina valoarea intensității în orice punct al poligonului. Metoda poartă numele de *Umbrire Gouraud*, după numele inventatorului său, Henri Gouraud (1971).



**Figura 17** Calculul normalei în vertex

Mai întâi trebuie să determinăm intensitățile în vârfurile poligonului. Pentru a calcula intensitatea într-un vertex pe baza modelului nostru de iluminare, avem nevoie de *normala în vertex*. Nu putem utiliza normala poligonului, întrucât trebuie să luăm în considerație orientarea poligoanelor adiacente pentru a evita discontinuitățile în zona muchiilor. De fapt avem nevoie de normala la suprafața care este de fapt aproximată prin poligoane. Putem *aproxima* normala la suprafață într-un vertex calculând media normalelor poligoanelor de care aparține acel vertex (Figura 17).

Fie poligonul din Figura 18. Intensitatea  $I_Q$  în punctul  $Q$  se calculează prin interpolarea intensităților  $I_A$  în  $A$  și  $I_B$  în  $B$ :

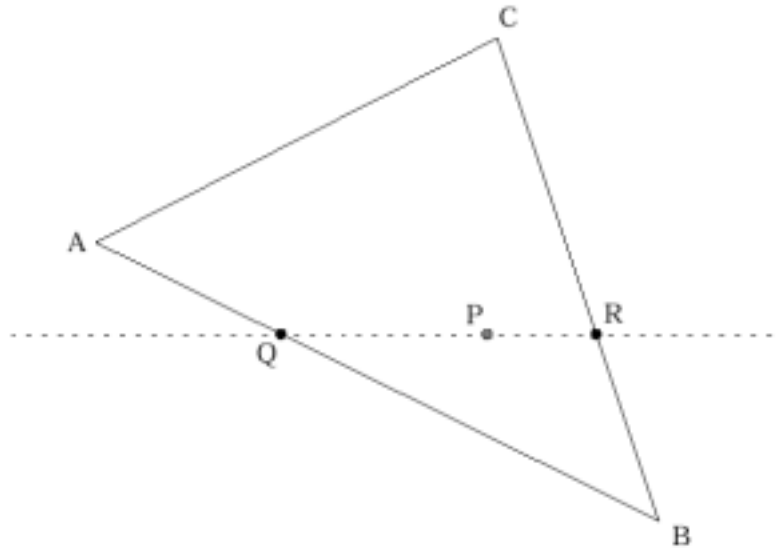
$$I_Q = uI_B + (1-u)I_A \text{ unde } u = \frac{AQ}{AB}$$

Similar pentru punctul  $R$ ,

$$I_R = wI_B + (1-w)I_C \text{ unde } w = \frac{CR}{CB}$$

Acum putem interpola  $IQ$  și  $IR$  pentru a obține intensitatea în  $P$ :

$$I_P = vI_R + (1-v)I_Q \text{ unde } v = \frac{QP}{QR}$$



**Figura 18** Interpolarea intensității

Pentru fiecare linie de scanare, putem accelera procesul dacă utilizăm calcul *incremental*. Pentru doi pixeli  $p_1$  și  $p_2$  la  $v_1$ , respectiv  $v_2$  pe linia de scanare, avem

$$I_{P_2} = v_2 I_R + (1 - v_2) I_Q$$

și

$$I_{P_1} = v_1 I_R + (1 - v_1) I_Q$$

Scăzând  $I_{P_1}$  din  $I_{P_2}$  obținem

$$I_{P_2} = I_{P_1} + (I_R - I_Q)(v_2 - v_1)$$

Dacă notăm  $(I_R - I_Q)$  cu  $\Delta I$  și  $(v_2 - v_1)$  cu  $\Delta v$ , avem

$$I_{P_2} = I_{P_1} + \Delta I \Delta v$$

Trebuie să calculăm  $\Delta I$  și  $\Delta v$  o singură dată pentru o linie de scanare, reducând calculul pentru un pixel la o singură adunare.

Nu vom mai avea discontinuități în umbrire pentru poligoane adiacente. Intensitatea calculată prin interpolare pe o muchie va fi punctul de plecare pentru interpolarea pe următorul poligon.

### 3.2.1. Probleme ale interpolării intensității

Ce facem însă atunci când dorim ca o muchie să fie într-adevăr vizibilă? Metoda descrisă mai sus “netezește” muchiile ca ele să dispară. Rezolvarea este să efectuăm acțiuni speciale atunci când dorim o muchie reală: calculăm două normale în vertex, câte una pentru fiecare fațetă, fiecare calculată prin medierea normalelor la suprafața poligoanelor de pe partea corespunzătoare.

O altă problemă este prezentată în Figura 19a.

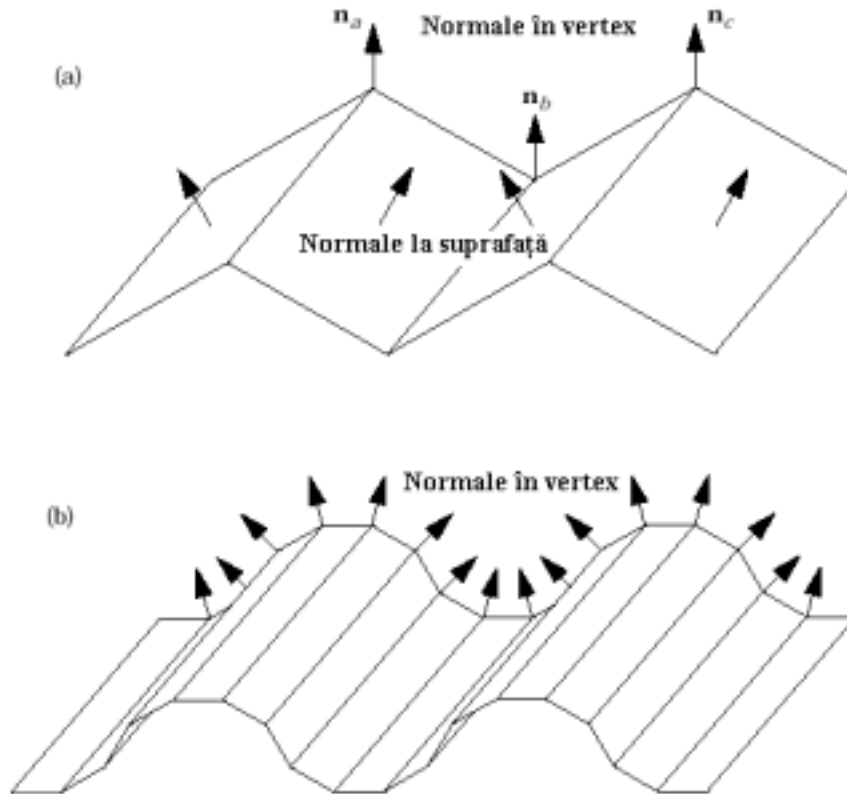


Figura 19 Probleme la interpolarea intensității

Aici fațetele adiacente au fost astfel aranjate încât normalele mediate  $\mathbf{n}_a$ ,  $\mathbf{n}_b$  și  $\mathbf{n}_c$  sunt paralele. Ca urmare fiecare fațetă va avea aceeași umbrire și suprafața va apărea incorect ca fiind plată. Una dintre soluții ar fi să introducem poligoane suplimentare, ca în Figura 19b.

Interpolarea intensității nu elimină complet efectul de bandă Mach, dar rezultatele sunt mult mai realiste decât în cazul umbririi constante. Această tehnică de interpolare poate de asemenea să distorsioneze petele luminoase datorate reflexiei speculare.

O altă problemă apare atunci când o pată luminoasă datorată reflexiei speculare este plasată în interiorul unui poligon – ea va fi netezită și nu va mai fi vizibilă.

### 3.3. Interpolarea vectorilor normali

Această tehnică utilizează o metodă de interpolare similară, cu diferența că se interpolează *vectorul normală la suprafață*, și nu intensitatea. Această metodă mai este cunoscută sub numele de *interpolare Phong*. Normalele în vertex-uri se calculează la fel ca în cazul metodei interpolării intensității. În Figura 18, dacă normala în punctul P este  $\mathbf{n}_p$ , avem

$$\begin{aligned} n_Q &= un_B + (1-u)n_A \\ n_R &= wn_B + (1-w)n_C \\ n_P &= vn_R + (1-v)n_Q \end{aligned}$$

unde  $u$ ,  $v$  și  $w$  sunt aceiași ca mai înainte. De asemenea, normala poate fi calculată incremental.

Rezultatul este mult mai realist decât în cazul metodei interpolării intensității, mai ales atunci când se ia în considerație și reflexia speculară – petele luminoase sunt mult mai

bine reprezentate, iar efectul de bandă Mach este mult redus (din păcate, în cazul sferelor și cilindrilor mai este posibil să apară efectul de bandă Mach).

Dezavantajul metodei constă în necesarul mai mare de calcul pentru fiecare punct din poligon; trebuie efectuate calcule complete de umbrire, pe baza normalei interpolate în acel punct.

### **3.4. Interpolarea produsului scalar**

Această metodă este un compromis între cele două metode de interpolare anterioare. În această metodă, cantitățile interpolate între capetele liniei de scanare sunt produsele scalare  $N \cdot L$  și  $(R \cdot V)^n$ . Această metodă mai este denumită “interpolare Phong redusă”.

### **3.5. Sumar**

Am dezvoltat un model simplu al iluminării locale care ține cont de:

- Reflexia difuză directă de la surse punctiforme.
- Reflexia speculară directă de la surse punctiforme.

Nu s-au luat în considerație:

- Surse luminoase distribuite
- Reflexii indirecte:
  - Specular la specular
  - Specular la difuz
  - Difuz la specular
  - Difuz la difuz

Am preferat să modelăm reflexiile difuze indirecte ca lumină ambiantă. Modelul nostru poate trata geometrii poligonale, interpolând normalele la suprafață pentru a simula o suprafață curbă. Dependențele de lungimea de undă sunt modelate utilizând trei culori primare, roșu, verde și albastru, iar intensitățile sunt mapate direct pentru semnalele de control ale monitorului color.

Vom căuta în continuare să îmbunătățim acest model, în vederea simulării unor efecte particulare cum ar fi transparența, sau pentru modelarea mai realistă a unor factori care au fost simplificați sau omiși.

## 4. Textură și transparență

Metodele de umbrire discutate anterior pot reprezenta suprafețe netede iluminate de una sau mai multe surse de lumină. Vom vedea în continuare cum să modelăm suprafețe care nu sunt netede sau conțin mai multe culori.

Vom trata două tipuri de texturi:

- *Tipare sau detalii de culoare* – Suprapunem un tipar peste o suprafață netedă și suprafața rămâne în continuare netedă. De exemplu, putem modela simplu o planetă aplicând o hartă peste o sferă. Adăugarea de tipare la o suprafață este în esență o *funcție de mapare*, iar procesul poartă denumirea de mapare de tipar (*pattern mapping*).
- *Rugozitate* – Majoritatea obiectelor din natură nu sunt netede. Suprafața lor posedă o micro-structură. De exemplu putem modela o portocală prin intermediul unei sfere de culoarea respectivă. Oricât de realistă ar fi umbrirea, modelul nostru nu prea va semăna cu o portocală, întrucât suprafața unei portocale nu este netedă. În termeni de modelare, trebuie să alterăm uniformitatea suprafeței cu ajutorul unei *funcții de perturbare* care să modifice într-adevăr geometria suprafeței.

### 4.1. Tipare

#### 4.1.1. Maparea spațiului obiect

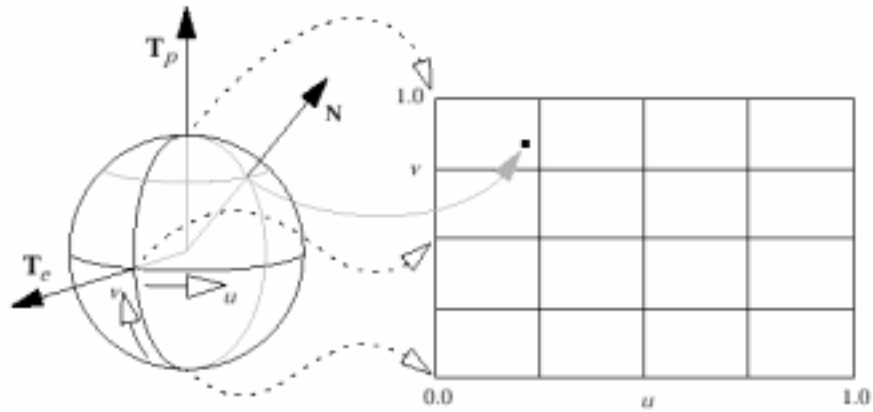
Pentru adăugarea de detalii de culoare la o suprafață netedă putem să îi suprapunem un tipar, memorat sub forma unui fișier imagine. Tiparul și obiectul sunt ambele definite în sistemul de coordonate propriu, iar texturarea este definită ca o funcție de mapare între aceste două sisteme. În mod curent, spațiul tiparului este specificat prin intermediul unor coordonate  $(u,v)$ , iar dacă coordonatele spațiului obiect sunt  $(\theta,\phi)$ , atunci vom determina funcția de mapare ca

$$u = f(\theta, \phi) \quad v = g(\theta, \phi) \quad (4-1)$$

Acum putem calcula coordonatele  $(u,v)$  ale unui punct în spațiul tiparului corespunzător unui punct de pe suprafața obiectului. Culoarea corespunzătoare acestor coordonate este culoarea pe care o vom aplica pe suprafață.

Pentru a ilustra determinarea funcției de mapare vom exemplifica pornind de la o sferă. Trebuie să cunoaștem vectorul normală  $\mathbf{N}$  în punctul de pe sferă care ne interesează și procesul este echivalent cu găsirea latitudinii și longitudinii.

Sistemul local de coordonate atașat sferei (cu originea în centrul sferei) este definit prin doi vectori unitate,  $\mathbf{T}_p$  și  $\mathbf{T}_e$ , care indică dinspre origine spre polul nord, respectiv dinspre origine către un punct de referință pe ecuator. Definim maparea (vezi Figura 20) astfel încât coordonata  $u$  să varieze între 0 și 1 de-a lungul ecuatorului, iar coordonata  $v$  să varieze de asemenea între 0 și 1 de la polul sud către polul nord. La poli,  $u$  este egal cu 0.



**Figura 20** Funcția de mapare pentru o sferă

Pe baza acestor definiții obținem coordonata  $v$ :

$$v = \frac{\phi}{\pi} \quad (4-2)$$

unde  $\phi$  este parametrul latitudine dat de

$$\phi = \cos^{-1}(-T_p \cdot N) \quad (4-3)$$

Dacă  $v$  este egal cu 0 sau 1,  $u$  este zero. Dacă nu, atunci coordonata  $u$  este calculată ca

$$u = \frac{\cos^{-1}[(T_e \cdot N) / \sin \phi]}{2\pi} \quad (4-4)$$

Dacă

$$(T_p \times T_e) \cdot N \quad (4-5)$$

este mai mic decât zero, atunci trebuie să înlocuim pe  $u$  cu  $u-1$ . Acest test este necesar pentru a decide de care parte a lui  $T_e$  este situat punctul. Pe baza coordonatelor  $u$  și  $v$  astfel calculate, selectăm culoarea din imaginea care conține tiparul și o aplicăm pe punctul respectiv de pe sferă.

În Figura 21 este prezentat un exemplu de mapare de tipar pe o sferă. Tiparul constă dintr-un tablou de pătrate aleator colorate, iar maparea pe o sferă produce distorsiunile prezentate. De remarcat concentrarea din jurul polilor și lățirea de-a lungul ecuatorului.

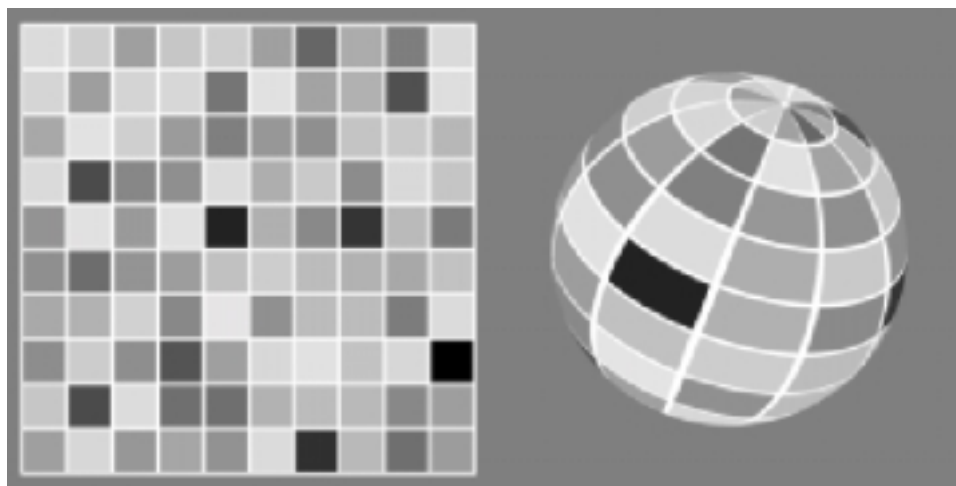


Figura 21 Maparea unui tipar pe o sferă

### 4.1.2. Maparea parametrică

Aceasta este similară mapării spațiului obiect, prin aceea că trebuie să determinăm o mapare de la spațiul tiparului la cel al suprafeței, însă în acest caz suprafața este definită parametric, ceea ce ușurează mult.

### 4.1.3. Maparea mediului

Această tehnică de mapare este o metodă eficientă de simulare a reflexiei în oglindă a mediului înconjurător pe un obiect. Asemenea tehnici au putut fi văzute în filmele *The Abyss* sau *Terminator 2*, unde au fost mixate imagini 3D pe calculator cu imagini reale.

Tehnica de mapare a mediului se bazează pe existența unei sau mai multor imagini ale scenei din jurul obiectului. Pentru a înțelege cum funcționează, trebuie să ne imaginăm că plasăm un cub în jurul obiectului și fotografiem scena prin fiecare dintre cele șase fațete. Cele șase imagini constituie tiparul de mediu. Pe parcursul reprezentării, atunci când trebuie să colorăm un anumit punct de pe suprafața obiectului, construim un vector de reflexie în acel punct, relativ la vectorul de observație (vezi Figura 10). Determinăm apoi punctul de intersecție cu suprafața cubului a acestui vector de reflexie, și deci încotro indică acest vector în tiparul imaginii. Acea culoare este culoarea pe care trebuie să o aplicăm pe suprafața obiectului.

Tiparele de imagine sunt în acest caz simple proiecții perspective ale scenei pe cele șase fațete ale cubului. Totuși, întrucât fiecare vedere este decalată cu  $90^\circ$  față de vecinii săi, apar distorsiuni semnificative în acele zone care corespund fiecărei muchii a cubului, care pot să pară ne-naturale în vizualizarea suprafeței. În majoritatea cazurilor, această limitare poate fi tolerată, mai ales în cazul obiectelor cu o formă complexă.

## 4.2. Rugozitate

În natură avem de-a face cu diferite scale de rugozitate, fiecare producând un efect diferit.

- Rugozitatea macroscopică poate să fie atât de mare încât ea afectează geometria, iar suprafața este evident că nu este plată. Asemenea exemple include textura unei portocale, suprafața unei anvelope etc.
- Rugozitatea la scară microscopică dă suprafeței aparența de netezime, dar ea nu este o oglindă perfectă. Printre exemple enumerăm materiale obișnuite ca hârtia,



pereti, lemnul neted etc. Deși este mică în termeni geometrici, această rugozitate este mare în comparație cu lungimea de undă a luminii incidente.

- În fine, rugozitatea la scară nanometrică este similară ca magnitudine lungimii de undă a luminii și produce o plajă de culori de interferență. Asemenea efecte nu se obișnuiește să fie modelate în grafica pe calculator.

#### 4.2.1. Rugozitate macroscopică

Majoritatea obiectelor naturale nu sunt netede; suprafața lor posedă un fel de microstructură. De exemplu, am putea modela o portocală utilizând o sferă de culoarea adecvată. Oricât de realist ar fi modelul de umbrire, nu prea va semăna cu o portocală reală; aceasta se datorește faptului că suprafața portocalei nu este netedă.

O posibilă abordare ar fi să modelăm geometria într-un mod mai complex. În unele cazuri, aceasta este o alternativă acceptabilă. Pentru a genera o portocală realist, cu mii de ridicături și adâncituri, totuși, aceasta ar fi extrem de complicat. Obiectul rezultat ar fi de asemenea dificil și lent de vizualizat.

O alternativă des utilizată este să simulăm o geometrie mai complexă în faza de reprezentare, prin alterarea uniformității suprafeței utilizând o *funcție de perturbare* care să modifice geometria suprafeței.

O tehnică uzuală pentru modelarea suprafețelor care nu sunt netede este să perturbăm vectorul normală în fiecare punct al suprafeței, înaintea calculării intensității în acel punct pe baza modelului de iluminare.

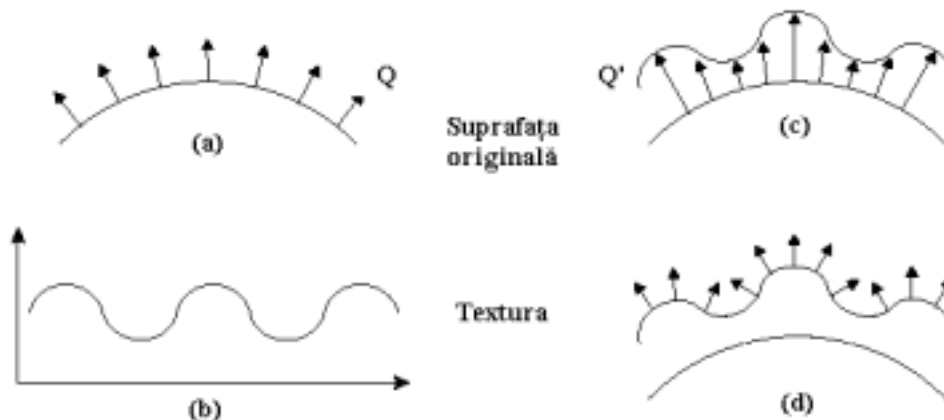


Figura 22 Bump mapping

Dacă adăugăm o componentă mică aleatoare rezultă o suprafață rugoasă. Se pot obține și alte efecte interesante dacă perturbarea normalei este funcție de poziția pe suprafață. De exemplu, o suprafață poate să apară cu valuri dacă perturbăm normala cu o funcție sinus. Această tehnică este cunoscută în literatura de specialitate sub denumirea de *bump mapping*.

Pentru o suprafață  $Q$  definită parametric sub forma  $Q(u,v)$ , putem defini o nouă suprafață (rugoasă)  $Q'(u,v)$  adăugând o funcție de perturbare  $f(u,v)$  la  $Q$  în fiecare punct pe direcția normalei originale în acel punct. Pentru un punct  $\mathbf{q}$  de pe suprafață, noul punct, perturbat,  $\mathbf{q}'$  este

$$\mathbf{q}' = \mathbf{q} + f(\mathbf{q})\mathbf{n}_q$$

unde  $\mathbf{n}_q$  este vectorul normală unitate în  $\mathbf{q}$ .

În Figura 23 este prezentat un tor a cărui suprafață a fost denivelată utilizând această tehnică. Funcția de perturbare poate fi definită ca o funcție analitică sau ca un set de valori plasate într-o tabelă de căutare. Definirea valorilor pentru întreaga suprafață ar necesita o tabelă mult prea mare, astfel încât se preferă memorarea unui număr mai mic de valori, care să fie interpolate pentru punctele intermediare.

De remarcat că această tehnică evită modelarea explicită a geometriei noii suprafețe rugoase. Aceasta are un efect vizibil important: dacă privim *silueta* suprafeței transformate în acest mod, ea este totuși netedă. Rugozitatea devine vizibilă numai atunci când aplicăm modelul de umbrire.

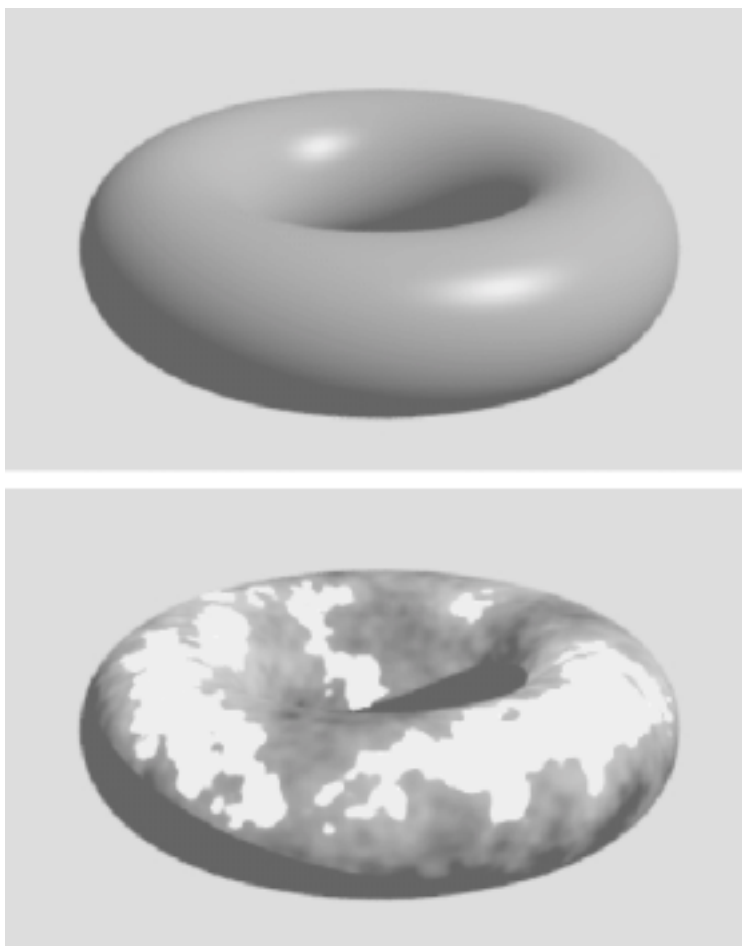
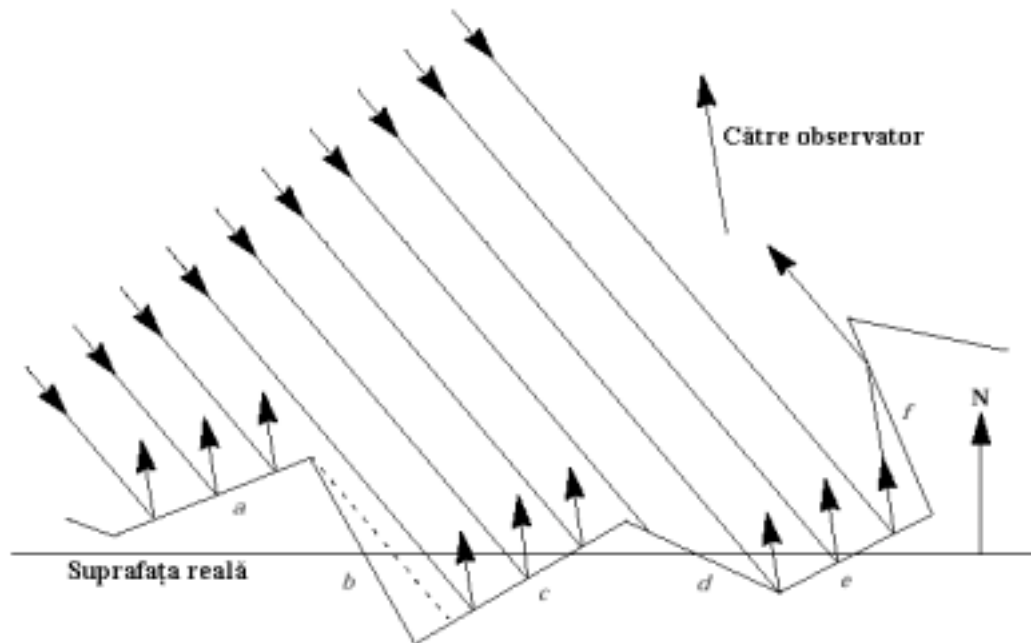


Figura 23 Exemplu de bump mapping

### 4.2.2. Rugozitate microscopică

Modelul Cook-Torrance a fost dezvoltat pentru a descrie suprafețe ale căror rugozitate microscopică este mare în comparație cu lungimea de undă a luminii incidente.

O suprafață reflectantă se modelează ca o colecție de fațete microscopice orientate aleator, fiecare fiind considerată un reflector perfect. În Figura 24 se prezintă o colecție de micro-fățete, și ne vom ocupa numai de acele fațete care reflectă lumina înspre observator. În figură , acestea sunt fațetele a, c și e.



**Figura 24** Modelarea unei suprafețe cu microfațete

Modelul ia în considerație trei situații:

- O fațetă reflectă direct înspre observator – aceasta este fațeta *a* din figură.
- O fațetă este în umbra unei fațete vecine, astfel încât nu primește întreaga cantitate de lumină – fațeta *c* este parțial umbrată de fațeta *b*, prezentat cu linie punctată.
- O fațetă reflectă, dar razele reflectate sunt *re-reflectate* de o altă fațetă și nu ajung la observator – razele reflectate de fațeta *e* ajung la fațeta *f*. Aceste reflexii multiple contribuie la *reflexia difuză* a suprafeței.

Modelul Cook-Torrance ia în considerație aceste efecte, rezultând mărimea reflexiei speculare a întregii suprafețe  $k_s$ :

$$k_s = \frac{DGF}{\pi(N \cdot V)(N \cdot L)}$$

$D$  este funcția de distribuție care descrie direcția fiecărei fațete pentru întreaga suprafață (de obicei o funcție Gauss de probabilitate), iar  $G$  este un factor care descrie măsura în care fațetele se umbresc și se maschează reciproc.  $F$  este factorul Fresnel, care ne dă fracțiunea din lumina incidentă care se reflectă și nu este absorbită. Este definit ca

$$F = \frac{1}{2} \left[ \frac{\sin^2(\phi - \theta)}{\sin^2(\phi + \theta)} + \frac{\tan^2(\phi - \theta)}{\tan^2(\phi + \theta)} \right]$$

### 4.2.3. Comparație între metodele Phong și Cook-Torrance

Cele două modele dau rezultate diferite. În particular

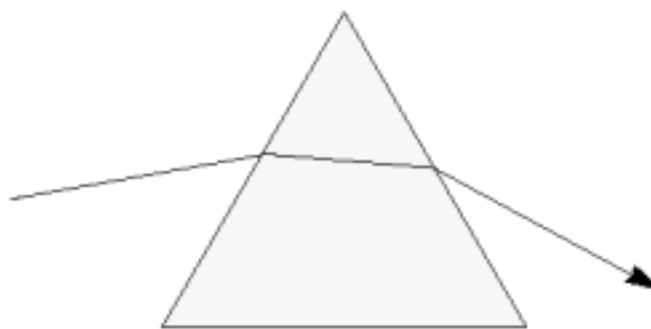
- Poziția și intensitatea petelor luminoase produse de cele două modele sunt similare pentru valori mici ale lui  $\theta$ , însă diferă considerabil cu mărirea lui  $\theta$ .

- Modelul Phong presupune că petele luminoase au aceeași culoare ca și lumina incidentă, ceea ce nu se întâmplă în realitate, întrucât reflectanța materialului depinde de lungimea de undă. Modelul Cook-Torrance încorporează acest efect (prin intermediul funcției Fresnel), dând rezultate mai realiste.
- Modelul Cook-Torrance este mai costisitor din punct de vedere computațional decât modelul Phong și nu se justifică în unele cazuri.

### 4.3. Aspectele optice ale transparenței

Nu toate materialele sunt opace. Unele materiale permit trecerea luminii și spunem că lumina este *transmisă* sau *refractată*. Există două tipuri de refracție:

- *Refracție difuză* – Apare atunci când o suprafață este translucidă, dar nu chiar transparentă. Lumina care trece este atenuată și împrăștiată de neregularitățile interne și de suprafață. O scenă privită printr-o suprafață cu refracție difuză va apărea mai mult sau mai puțin turbure.
- *Refracție speculară* – Apare atunci când lumina traversează un obiect transparent. O scenă privită printr-un refractor specular va fi clară, deși în general razele de lumină sunt deviate la trecerea prin material (Figura 25).



**Figura 25** Refracție

Comportarea luminii la trecerea dintr-un mediu în altul este descrisă de *Legea lui Snell*. Conform cu cele din Figura 26,

$$\eta_{1\lambda}\sin\theta = \eta_{2\lambda}\sin\theta' \quad (4-6)$$

Aici  $\eta_{1\lambda}$  și  $\eta_{2\lambda}$  sunt indicii de refracție ai mediilor și, întrucât  $\eta$  depinde de lungimea de undă a luminii incidente, culorile diferite vor fi deviate diferit. De obicei utilizăm o valoare medie a lui  $\eta$  pentru un material, ignorând dependența de lungimea de undă. Aceasta înseamnă că, de exemplu prismele de sticlă, nu produc curcubeu. Totuși, ele deviază lumina, altfel ele ar fi invizibile!

Efectul Legii lui Snell este semnificativ: o rază de lumină care trece din aer într-o sticlă densă ( $\eta \approx 1.5$ ) la un unghi de  $30^\circ$  va fi deviat cu aproximativ  $11^\circ$ . La ieșirea din sticlă lumina va fi deviată din nou, iar efectul constă în faptul că razele incidentă și refractată sunt paralele, dar deplasate.

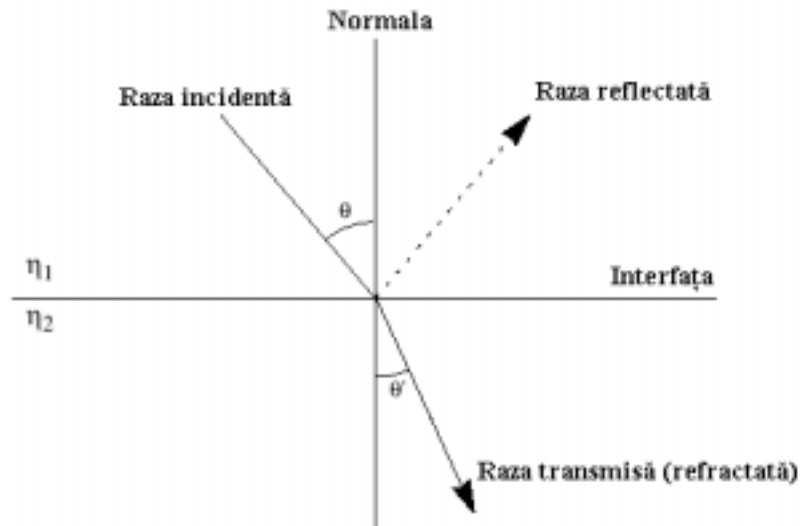


Figura 26 Legea lui Snell

De remarcat faptul că refracția are efecte vizibile. În Figura 27 avem două obiecte care refractă,  $A$  și  $B$ , ambele cu  $\eta$  mai mare decât mediul înconjurător, și două obiecte opace  $C$  și  $D$ . Dacă privim din punctul de observație  $v_1$ , raza  $r_1$  va traversa obiectul  $A$  și vom vedea obiectul  $C$ . Dacă nu ar fi refracție, am vedea obiectul  $D$ . Similar, privind din  $v_2$ , vedem, după refracție, obiectul  $D$  – din nou, în absența refracției am vedea obiectul  $C$ .

În practică, putem obține rezultate bune dacă pur și simplu deplasăm traiectoria luminii incidente cu o constantă (mică). Aceasta evită calcule complicate de trigonometrie. Pentru obiect foarte subțiri, deplasarea este atât de mică încât poate fi ignorată. Vom ignora deplasarea cauzată de refracție și ne vom concentra asupra scăderii în intensitate a luminii care traversează materialul transparent.

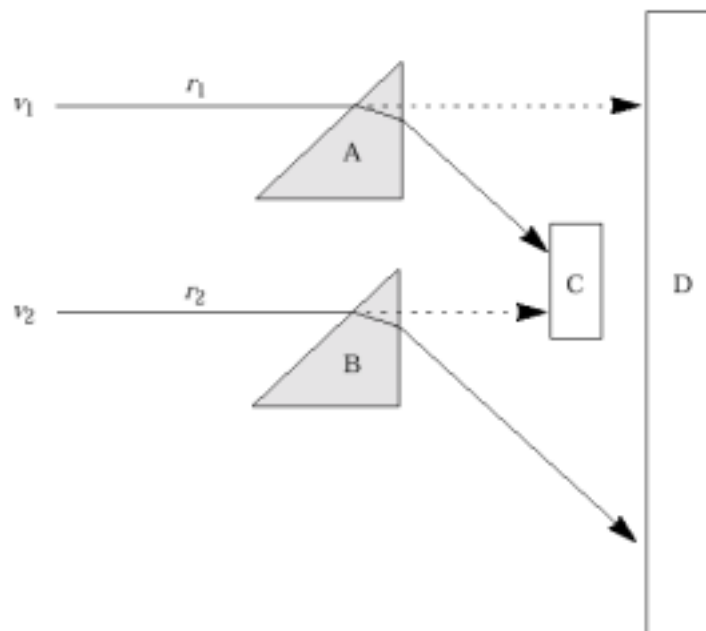


Figura 27 Efectele refracției (exagerat)

#### 4.4. Modelarea transparenței

Atunci când un obiect care refractă este plasat între punctul de observație și scena grafică, porțiunea de scenă din spatele obiectului este vizibilă *prin* obiect, dar și obiectul însuși

este vizibil la rândul său. În Figura 28 este prezentată o scenă privită de sus. Poligonul  $p_f$  este în față și poligonul  $p_b$  este în spate. Pentru a modela refracția datorată lui  $p_b$ , adăugăm o fracțiune din intensitatea punctului  $b$  de pe obiectul  $p_b$  vizibil prin  $p_f$ .

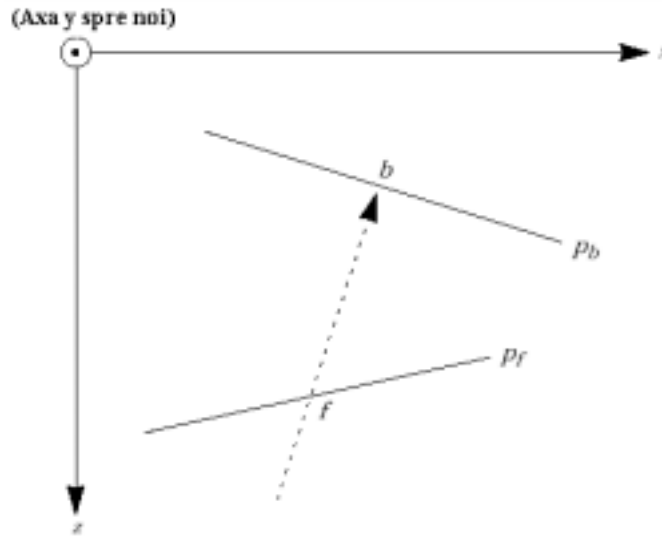


Figura 28 Modelarea transparenței

Introducem un *coeficient de transmisie*,  $t$ , care măsoară *transparența* unui obiect. Pentru un obiect complet opac,  $t=1$ , iar pentru un obiect perfect transparent,  $t=0$ . Atunci intensitatea observată  $I$  este

$$I = tI_f + (1-t)I_b$$

Această abordare ne permite de asemenea să modelăm suprafețe neuniforme: putem considera  $t$  ca funcție de poziția pe suprafață. De exemplu, dacă privim scena printr-o vază de sticlă, transparența aparentă a marginilor siluetei va fi mai mică decât cea a suprafeței îndreptată spre observator. În Figura 29a se vede cum lumina care trece prin marginile obiectului trebuie să traverseze mai mult material decât lumina care travează direct. Figura 29b prezintă o vedere de sus a unei scene în care un obiect  $B$  este privit printr-un obiect  $A$  din sticlă transparentă.

Dorim să avem mai puțină transparență pe măsură ce ne deplasăm din  $p_1$  prin  $p_2$  către  $p_3$ , și pentru a o determina putem utiliza componenta  $z$  a normalei la suprafață,  $n_z$  în fiecare punct  $p$ . Putem formula o relație liniară simplă,

$$t = t_{min} + (t_{max} - t_{min})n_z$$

în care  $t_{min}$  și  $t_{max}$  sunt transparențele minimă și maximă. Punctul  $p_1$  are  $n_z=0$ , deci  $t=t_{min}$ . Similar, în punctul  $p_3$ ,  $n_z=1$  și  $t=t_{max}$ . Punctul  $p_2$  are transparența undeva între  $t_{min}$  și  $t_{max}$  conform valorii lui  $n_z$ .

De asemenea, s-a propus și o relație alternativă neliniară:

$$t = t_{min} + (t_{max} - t_{min})[1 - (1 - n_z)^m]$$

unde  $m$  este un factor de putere de ordinul 2 sau 3.

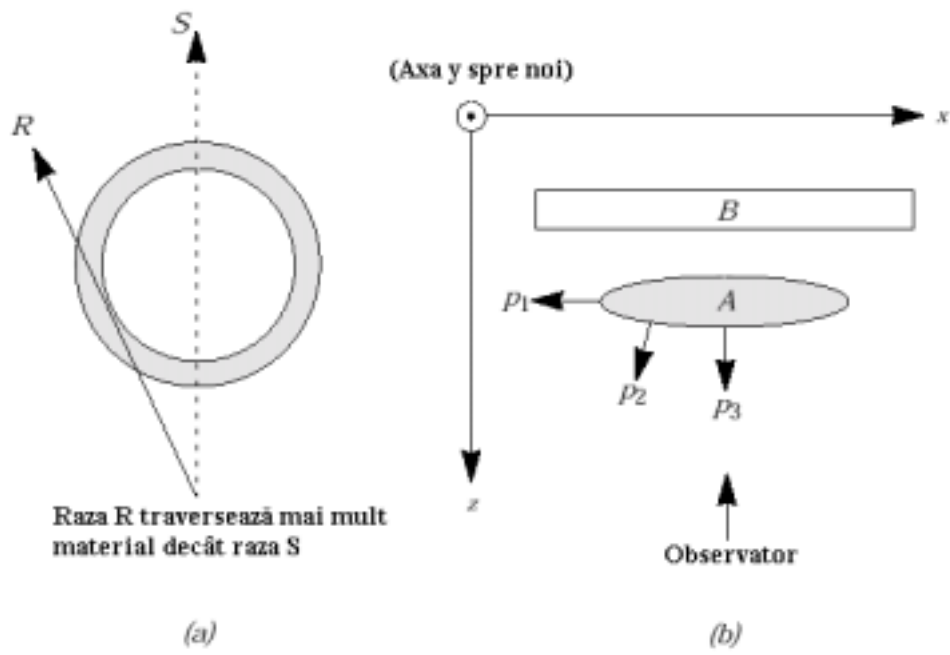


Figura 29 Transparență neliniară

## 5. Ray-tracing

Toate modelele de iluminare studiate până aici sunt locale, și ele au o deficiență: nu este posibil să modelăm reflexiile între obiectele dintr-o scenă. Aceste modele consideră fiecare suprafață suspendată în spațiu, independentă de celelalte suprafețe, și iluminată de una sau mai multe surse de lumină. Vom studia în continuare metode globale, care fac posibilă modelarea interacțiunilor între obiecte într-o scenă complexă.

Primul algoritm de iluminare globală care a fost dezvoltat a fost denumit *ray-tracing recursiv* și produce rezultate foarte realiste prin aplicarea legilor opticii. Această metodă urmează transportul energiei luminoase de-a lungul *razelor* într-o scenă, ceea ce înseamnă în practică faptul că luăm în considerație numai lumina reflectată sau transmisă specular. Desigur, în realitate avem de-a face cu o combinație complexă de lumină difuză și speculară, mai ales dacă ținem cont de faptul că reflexia difuză definește culoarea unei suprafețe. În practică, metodele ray-tracing includ un element care aplică o iluminare locală calculată pentru a simula lumina difuză. În lipsa acesteia, metoda ray-tracing ar produce imagini doar cu obiecte negre, lucioase!

O altă caracteristică a metodei este că această tehnică poate fi utilizată pentru diferite tipuri de suprafețe, atât pentru poligoane cât și pentru primitive simple cum ar fi sfere, conuri și toruri, ca și pentru suprafețe definite parametric.

O altă metodă cuscută puternică utilizată pentru calculul iluminării globale este cea denumită *radiosity*, care, prin contrast cu ray-tracing, este adecvată în particular mediilor cu suprafețe care reflectă sau transmit doar difuz. Imaginile produse prin metoda radiosity au un realism foarte bun, în unele cazuri mai bun decât cele generate prin ray-tracing. Din păcate, metoda radiosity este limitată la reprezentarea de obiecte definite prin rețele de poligoane. Vom studia această metodă în capitolul următor.

Principala motivație care a stat la baza dezvoltării metodelor globale este *realismul imaginii*. Se încearcă modelarea naturii fizice a energiei luminoase și a interacțiunii cu un material în scopul producerii de imagini apropiate de calitatea fotografică și aproape imposibil de distins de o scenă reală. Dorința de realism al imaginilor are însă neajunsurile sale, mai ales în privința costurilor computaționale, inerente dacă ne reamintim că încercăm să modelăm fenomene care sunt considerabil mai complexe decât cele considerate într-un model de iluminare locală.

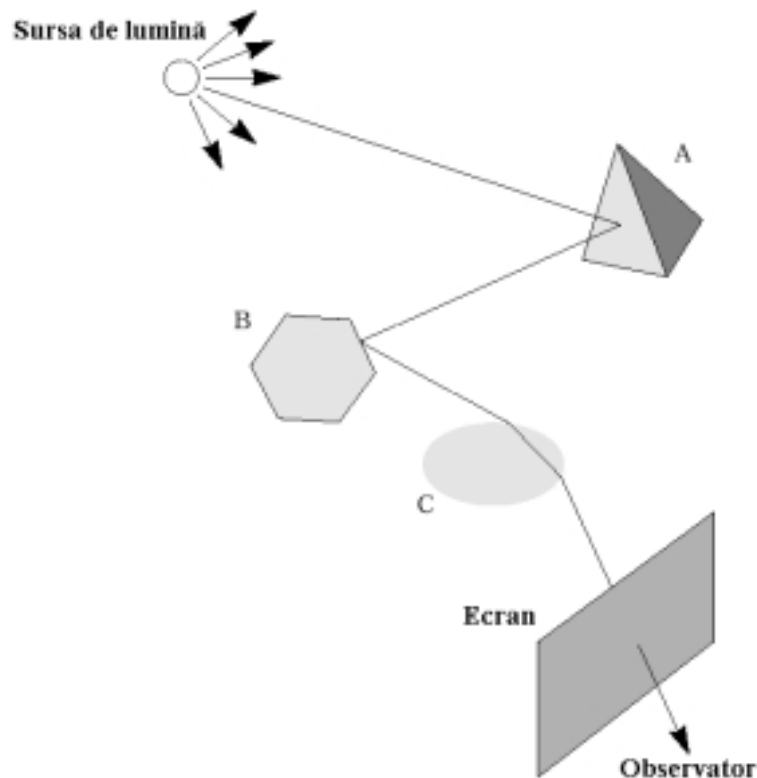
### 5.1. Algoritm de bază

Ideea de bază în ray-tracing este că un observator vede un punct de pe o suprafață ca rezultat al faptului că acea suprafață interacționează cu razele de lumină care provin din altă parte. Atunci când am discutat modelele de iluminare locală am luat în considerație numai interacțiunea suprafeței cu lumina provenită direct de la sursele de lumină. În general, o rază de lumină poate să ajungă la o suprafață și indirect, prin reflexie sau transmisie, de la alte suprafețe. Metoda ray-tracing calculează acea porțiune din iluminarea globală, rezultată în urma interacțiunii obiect-la-obiect, care este distribuită de lumina speculară. Ca urmare, aceasta este *dependentă de observator*, întrucât, de exemplu, poziția reflexiei unui obiect pe suprafața altui obiect se modifică dacă se modifică punctul de observație.

În principiu, algoritmul de ray-tracing este simplu: urmăm traiectoria fiecărei raze de lumină emisă de fiecare sursă de lumină, urmărind toate reflexiile și refracțiile, pe măsură ce interacționează cu mediul, și dacă în cele din urmă ajunge la punctul de observație, îi utilizăm intensitatea pentru vizualizarea unei porțiuni din imagine (vezi Figura 30). În



practică, aceasta nu se poate realiza. Fiecare sursă de lumină emite un număr infinit de raze, dintre care majoritatea nu vor ajunge la punctul de observație. Rezolvarea este să pornim din punctul de observație și să urmărim razele *înapoi* în scenă.



**Figura 30** Urmărirea razelor de la sursa de lumină

Pentru proiecția razelor în scenă definim atât un punct de observație cât și un plan de vedere, în coordonatele spațiului obiect (mai jos este descris un sistem simplu de vizualizare). Planul de vedere este divizat într-un număr mare de pixeli imaginari, care vor fi mapați peste pixelii reali dintr-un display grafic. Se urmărește o singură rază pentru fiecare pixel, pornind din punctul de observație, prin planul de vedere (raza *primară*) și se determină cel mai apropiat obiect pe care îl intersectează. Acest obiect astfel determinat va reprezenta suprafața care este vizibilă prin acel pixel (de remarcat faptul că avem de-a face cu o formă de îndepărtare a fațetelor ascunse).

Matematic, sub formă parametrică, o rază arbitrară este definită ca:

$$\mathbf{r}(t) = \mathbf{O} + \mathbf{D}t \quad (5-1)$$

unde  $\mathbf{r}$  este vectorul de poziție al unui punct pe dreapta unei raze cu parametrul  $t$ ,  $\mathbf{O}$  este punctul de plecare al razei, iar  $\mathbf{D}$  este un vector unitate pe direcția razei. Parametrul  $t$  este o valoare reală și reprezintă distanța (într-o unitate de măsură oarecare) de-a lungul razei din origine și pentru toate razele primare originea este situată în punctul de observație.

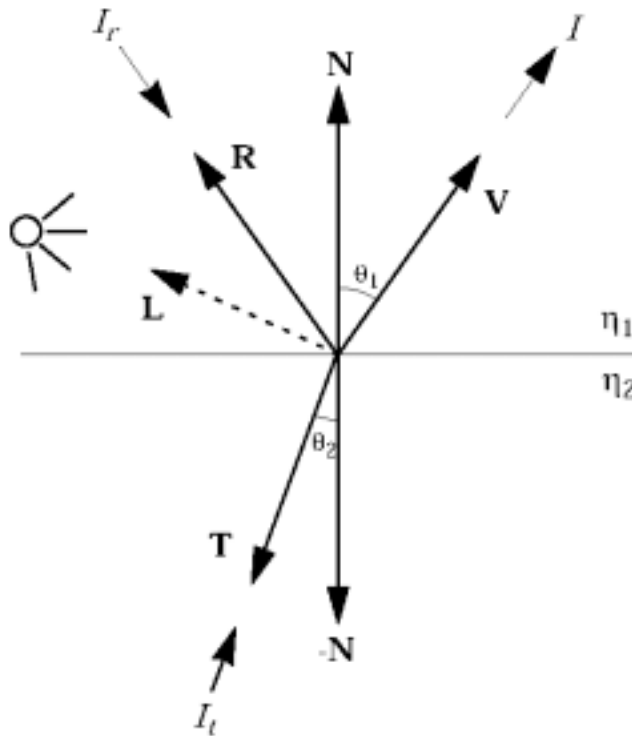
Următorul pas în algoritm este să calculăm intensitatea luminii care părăsește suprafața obiectului în punctul de intersecție (vezi Figura 31). Această intensitate se datorește unei componente locale similară ca formă celei din Ecuația 2-7, plus componentele datorate luminii reflectate sau transmise global. Pentru  $m$  surse de lumină, putem scrie aceasta sub forma:

$$I = k_d I_a + k_d \sum_{i=1}^m I_{p_i} \left[ (\mathbf{N} \cdot \mathbf{L}_i) + k_s (\mathbf{R} \cdot \mathbf{V})^n \right] + k_r I_r + k_t I_t \quad (5-2)$$

unde

- $k_d, k_s, I_a, n, \mathbf{N}, \mathbf{R}$  și  $\mathbf{V}$  sunt definite în capitolele anterioare,
- $I_{p_i}$  și  $\mathbf{L}_i$  sunt intensitatea și direcția sursei de lumină  $i$ ,
- $k_r$  este coeficientul de reflexie speculară globală,
- $I_r$  este intensitatea luminii care vine din direcția  $\mathbf{R}$ ,
- $k_t$  este coeficientul de transmisie speculară globală,
- $I_t$  este intensitatea luminii care vine din direcția  $\mathbf{T}$ .

Vom avea câte o asemenea expresie pentru fiecare componentă de culoare utilizată.



**Figura 31** Geometria ray-tracing-ului

Calculul ultimilor doi termeni din Ecuația 5-2 se realizează pornind noi raze din punctul de intersecție pe direcția reflexiei și refracției speculară (de-a lungul lui  $\mathbf{R}$ , respectiv  $\mathbf{T}$ ). Desigur, dacă  $k_r$  este zero, suprafața este perfect difuză și nu mai are loc reflexie speculară, deci nu este nevoie să pornim o nouă rază reflectată. Similar, dacă  $k_t$  este zero, suprafața poate fi considerată perfect opacă, deci nu se transmite lumina.

Aceste noi raze, numite *raze secundare*, vor fi urmărite prin scenă exact în același mod ca și razele primare, până când intersectează o altă suprafață. Atunci putem aplica din nou Ecuația 5-2 pentru noul punct de intersecție, rezultând valorile  $I_r$  și  $I_t$  pentru raza *primară*.

După cum se observă, acest proces poate fi repetat, într-o manieră recursivă, creînd noi raze secundare în fiecare punct de intersecție și returnând valorile  $I_r$  și  $I_t$  pentru raza trasată anterior. Natura recursivă a algoritmului este ilustrată în Figura 32, unde avem

patru suprafețe,  $S_1$ ,  $S_2$ ,  $S_3$  și  $S_4$  toate având un anumit grad de transparență, cu excepția lui  $S_4$ , care este opacă.

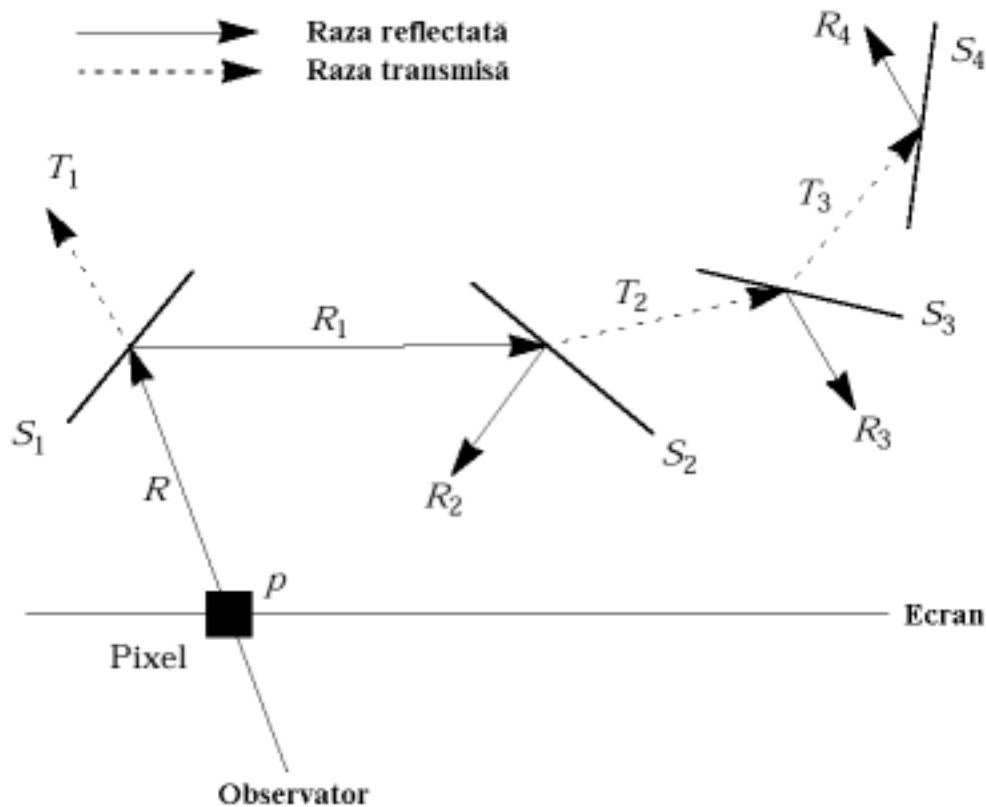


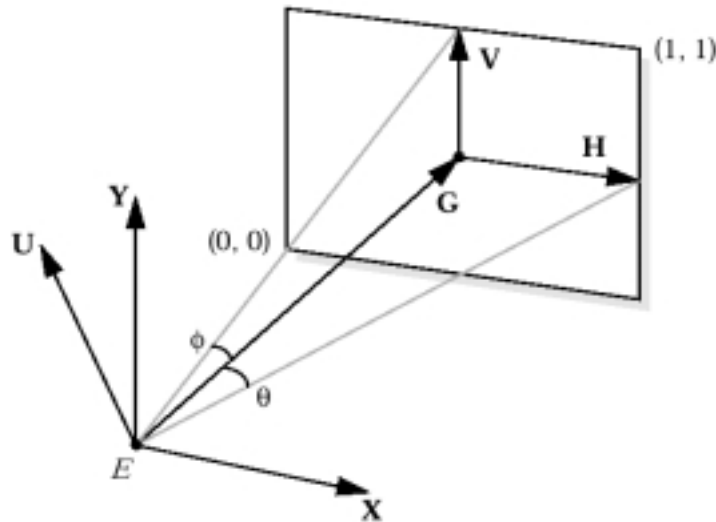
Figura 32 Ray-tracing pentru un pixel

Urmărim raza primară  $R$  pe direcția de observație prin pixelul  $p$ . Ea întâlnește mai întâi  $S_1$ , producând două raze secundare,  $R_1$  și  $T_1$ .  $T_1$  nu mai întâlnește alte suprafețe, însă  $R_1$  intersectează  $S_2$  și produce noile raze  $R_2$  și  $T_2$ . Raza  $R_2$  se pierde în fundal, în vreme ce  $T_2$  intersectează suprafața  $S_3$  și generează razele  $R_3$  și  $T_3$ . Raza  $R_3$  părăsește scena, iar  $T_3$  intersectează suprafața  $S_4$ . Întrucât  $S_4$  este opacă, se produce numai o rază reflectată  $R_4$ .

Se observă că razele  $T_1$ ,  $R_2$ ,  $R_3$  și  $R_4$  nu contribuie la intensitatea în pixelul  $p$ , întrucât ele nu pot să ajungă de la vreo sursă de lumină.

## 5.2. Un sistem simplu de vizualizare

Scopul unui sistem de vizualizare este să sprijine calculul razelor de la ochi prin pixelii imaginii. În această secțiune vom descrie arhitectura și utilizarea unui sistem simplu dar eficient care poate să producă vederi în perspectivă ale scenei noastre. Sistemul este prezentat în Figura 33.



**Figura 33** Sistem de vizualizare

Ochiul este plasat în punctul  $E$ , cu vectorul de poziție  $\mathbf{E}$  și privește în direcția lui  $\mathbf{G}$ . Vectorul  $\mathbf{G}$  poate să fie calculat direct din pozițiile ochiului și a punctului de referință. Cele două unghiuri de vedere  $2\theta$  și  $2\phi$  reprezintă câmpul vizual pe orizontală și pe verticală ale vederii planului de vedere (deci a imaginii) și indică o măsură a perspectivei imaginii. Dacă presupunem că pixelii din imagine sunt pătrați, atunci cele două unghiuri nu sunt independente. Trebuie specificat numai unul dintre unghiuri ( $2\theta$ ), celălalt poate fi calculat pe baza rezoluției imaginii.

Mai avem nevoie de o informație pentru a avea definit planul de vedere. Un vector “în sus” ( $\mathbf{U}$  în diagramă) este necesar pentru a indica orientarea planului de vedere față de direcția privirii.

Definim un sistem de coordonate în planul de vedere astfel încât colțul din stânga jos este originea, iar colțul din dreapta sus este la  $(1,1)$ . Apoi construim vectorii  $\mathbf{X}$  și  $\mathbf{Y}$  din

$$\mathbf{X} = \mathbf{G} \times \mathbf{U}$$

$$\mathbf{Y} = \mathbf{X} \times \mathbf{G}$$

Acești doi vectori sunt ortogonali și paraleli cu planul de vedere. Se observă că vectorii  $\mathbf{H}$  și  $\mathbf{V}$  pot fi scriși

$$\mathbf{H} = d \tan \theta \mathbf{X}$$

$$\mathbf{V} = d \tan \phi \mathbf{Y}$$

dacă  $\mathbf{X}$  și  $\mathbf{Y}$  sunt normalizați și  $d$  este distanța planului de vedere față de ochi. Utilizând  $\mathbf{H}$  și  $\mathbf{V}$ , un punct din planul de vedere,  $S$  poate fi exprimat ca

$$\mathbf{S} = \mathbf{E} + d\mathbf{G} + (2x'-1)\mathbf{H} + (2y'-1)\mathbf{V}$$

unde  $\mathbf{G}$  a fost normalizat și  $(x',y')$  este poziția lui  $S$  în sistemul de coordonate al planului de vedere. Dacă punctul  $S$  reprezintă centrul unui pixel sub forma poziției pe ecran  $(i,j)$

$$x' = \frac{i + 0.5}{R_x}$$

$$y' = \frac{j + 0.5}{R_y}$$

unde  $R_x$  și  $R_y$  sunt rezoluțiile imaginii. Acum putem calcula simplu direcția rayei care trece prin  $S$  ca

$$\mathbf{P} = \mathbf{S} - \mathbf{E} = d\mathbf{G} + (2x' - 1)\mathbf{H} + (2y' - 1)\mathbf{V}$$

Acum avem la dispoziție toți parametrii necesari pentru definirea unei raye care pleacă din ochi și trece prin pixelul  $S$ : după normalizarea lui  $\mathbf{P}$ , acesta ia locul lui  $\mathbf{D}$  în Ecuația 5-1, și îl facem pe  $\mathbf{O}$  să fie egal cu  $\mathbf{E}$ .

### 5.3. Vectorii de reflexie și de refracție

Pentru a urmări direcția unei raze, trebuie să putem calcula vectorii unitate  $\mathbf{R}$  și  $\mathbf{T}$  din Figura 31. În Figura 34, vectorii  $\mathbf{R}$ ,  $\mathbf{V}$  și  $\mathbf{N}$  sunt în același plan și utilizează triunghiuri similare.

Mai întâi calculăm vectorul  $\mathbf{N}'$  înmulțind normala  $\mathbf{N}$  cu  $\cos\theta_1$ . Întrucât  $\cos\theta_1$  este chiar produsul scalar dintre  $\mathbf{N}$  și  $\mathbf{V}$ , atunci

$$\mathbf{N}' = \mathbf{N}(\mathbf{N} \cdot \mathbf{V}) \quad (5-3)$$

Acest vector este denumit *vectorul cosinus* al lui  $\mathbf{V}$ , întrucât lungimea lui este chiar  $\cos\theta_1$ .

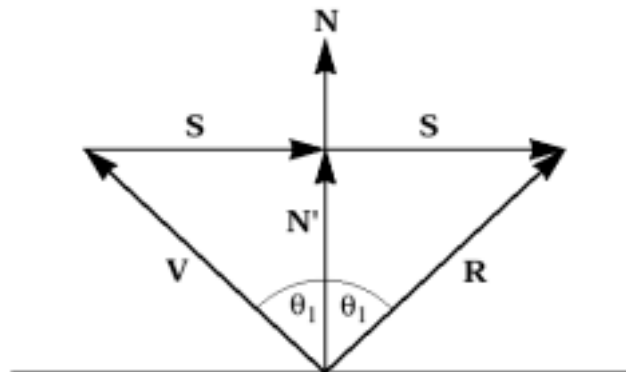


Figura 34 Calculul vectorului de reflexie

Din aceasta putem calcula *vectorul sinus*  $\mathbf{S}$  corespunzător:

$$\mathbf{S} = \mathbf{N}' - \mathbf{V} \quad (5-4)$$

Prin definiție lungimea vectorului sinus este  $\sin\theta_1$ , întrucât vectorii  $\mathbf{V}$ ,  $\mathbf{N}'$  și  $\mathbf{S}$  formează un triunghi dreptunghic. Se observă acum că vectorul de reflexie  $\mathbf{R}$  este dat de

$$\mathbf{R} = \mathbf{N}' + \mathbf{S} = 2\mathbf{N}' - \mathbf{V}$$

sau

$$\mathbf{R} = 2\mathbf{N}(\mathbf{N} \cdot \mathbf{V}) - \mathbf{V} \quad (5-5)$$

Direcția razei transmise  $\mathbf{T}$  se determină aplicând legea lui Snell din Ecuția 4-6 și remarcând faptul că  $\mathbf{T}$  este în același plan cu  $\mathbf{V}$  și  $\mathbf{N}$  (vezi Figura 35). Calculăm vectorul sinus ( $\mathbf{S}$ ) al vectorului incident, îl scalăm cu legea lui Snell pentru a obține vectorul sinus ( $\mathbf{S}'$ ) al vectorului transmis, calculăm vectorul cosinus ( $\mathbf{N}''$ ) pentru raza transmisă și apoi însumăm vectorii sinus și cosinus pentru a obține vectorul transmis.

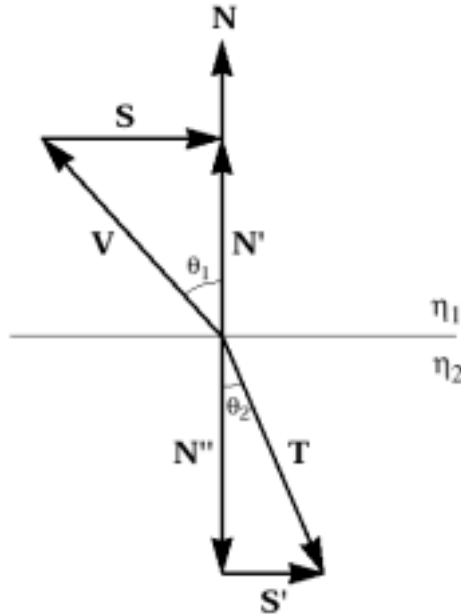
Vectorii sinus,  $\mathbf{S}$  și  $\mathbf{S}'$  sunt corelați prin legea lui Snell

$$\mathbf{S}' = \frac{\eta_1}{\eta_2} \mathbf{S} \quad (5-6)$$

Lungimea vectorului cosinus este dată de

$$|\mathbf{N}''| = \sqrt{1 - |\mathbf{S}'|^2}$$

întrucât  $\cos^2\theta + \sin^2\theta = 1$ , avem



**Figura 35** Calculul vectorului de transmisie

Ca urmare, vectorul transmis este dat de

$$\mathbf{T} = \mathbf{N}'' + \mathbf{S}' = \frac{\eta_1}{\eta_2} \mathbf{S} - \mathbf{N} \sqrt{1 - |\mathbf{S}'|^2}$$

Această ecuație poate fi rescrisă în funcție de vectorii originali  $\mathbf{N}$  și  $\mathbf{V}$  pe baza ecuațiilor 5-6, 5-4 și 5-3. După simplificări, obținem

$$\mathbf{T} = \frac{\eta_1}{\eta_2} [\mathbf{N}(\mathbf{N} \cdot \mathbf{V}) - \mathbf{V}] - \mathbf{N} \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 [1 - (\mathbf{N} \cdot \mathbf{V})^2]} \quad (5-7)$$

Dacă privim din nou Ecuția 5-6, observăm că este posibil ca lungimea lui  $\mathbf{S}'$  să fie mai mare decât unu dacă  $\eta_1 > \eta_2$ . Această situație apare atunci când lumina trece dintr-un mediu în altul mai puțin dens, cum ar fi de la sticlă la aer. Totuși, prin definiție lungimea unui vector sinus este mai mică sau egală cu 1 și orice altceva este imposibil fizic.

Semnificația acestei condiții este aceea că ne va avertiza că a apărut *reflexie totală internă*. Ca urmare nu există vector de transmisie și vectorul de reflexie internă poate fi calculat inversând normala  $\mathbf{N}$  și aplicând Ecuația 5-5.

## 5.4. Umbre

Revenind acum la calculul intensității unei raze luminoase care pleacă de pe o suprafață (vezi Ecuația 5-2 și Figura 31), de obicei se mai efectuează un calcul pentru a determina dacă punctul de intersecție se află în umbră relativ la sursa de lumină.

Teoretic, testarea umbrei este foarte simplă. Mai întâi construim o rază a cărei origine este în punctul de intersecție și a cărei direcție este către sursa de lumină, vectorul  $\mathbf{L}$  în Figura 31. Această rază, denumită *rază de umbrire*, este apoi testată dacă nu intersectează alte obiecte din scenă, ca și în cazul intersecției originale. Dacă se găsește un punct de intersecție și acesta este *mai apropiat* decât sursa de lumină atunci putem spune că punctul de pe suprafață este în umbră pentru acea sursă de lumină.

În cazul mai multor surse de lumină, razele de umbrire sunt urmărite fiecare, pentru a determina care dintre ele produc o umbră pe suprafață. La iluminarea locală contribuie numai acelea care nu sunt blocate. Lumina provenind de la celelalte surse de lumină nu va fi inclusă în calculul intensității pe suprafață, în Ecuația 5-2.

De obicei, obiectul a cărui suprafață o umbrim este inclus într-o listă de obiecte față de care testăm raza de umbrire. Totuși, putem întâlni o problemă care a fost denumită *acneea ray-tracing-ului* (!). Ea apare datorită erorilor de precizie la calcularea punctului de intersecție. El poate cădea la o distanță foarte mică dar nu nesemnificativă în afara suprafeței obiectului. Dacă punctul este situat în spatele suprafeței relativ la sursa de lumină, atunci se va detecta o intersecție a razei de umbrire cu această suprafață, ceea ce va produce o pată întunecată în imaginea obiectului.

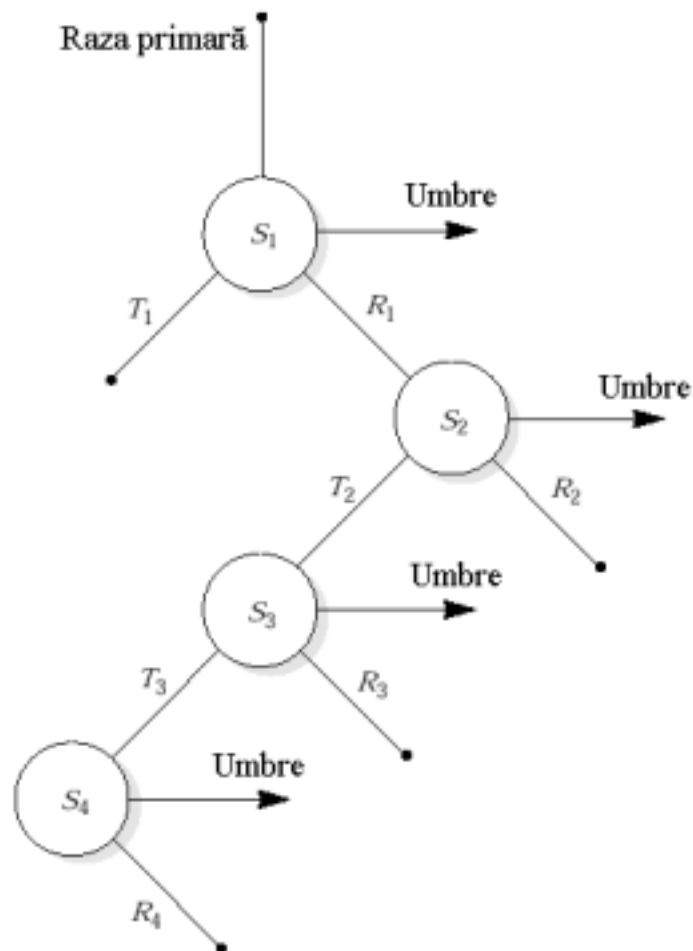
Evident, este un rezultat nedorit, însă el poate fi compensat în mai multe moduri. Am putea introduce o toleranță, de exemplu  $10^{-5}$ , și să precizăm numai distanțele de intersecție de-a lungul razei de umbrire mai mari decât această valoare să fie luate în considerație. Desigur, această toleranță trebuie să fie mai mare decât orice erori de precizie care ar putea să apară la calculul de intersecții, dar să fie suficient de mică pentru a nu pierde intersecții reale. Ca o alternativă, am putea utiliza această toleranță pentru deplasarea originii razei de umbrire înspre sursa de lumină, înainte de a efectua testele de intersecție. O altă modalitate ar fi să excludem această suprafață din testele de intersecție, dar este aplicabilă numai dacă știm că suprafața de plecare nu se poate 'vedea' pe sine, adică este *convexă*.

## 5.5. Arborele de raze

Putem vizualiza procesul recursiv utilizat în ray-tracing cu ajutorul unei diagrame numită *arborele de raze*, care evidențiază setul complet de intersecții care trebuie determinate pentru calculul intensității și culorii care ajunge la un anumit pixel. Arborele de raze corespunzător situației din Figura 32 este prezentat în Figura 36.

La implementarea unui *ray-tracer* trebuie să stabilim o adâncime maximă până la care sunt urmărite razele, altfel vom descoperi că se consumă o mare cantitate de timp pentru calculul unor raze care contribuie prea puțin la imaginea finală. Totuși, dacă nu se permite o adâncime suficientă a arborelui de raze, aspectul artificial va fi evident în imaginea generată.

Avem la dispoziție trei modalități prin care putem limita adâncimea arborelui. Prima posibilitate apare atunci când o rază se pierde în spațiu fără să intersecteze nici un obiect, terminând deci acea ramură din arbore. În exemplul nostru, aceasta se întâmplă pentru razele  $T_1$ ,  $R_2$ ,  $R_3$  și  $R_4$ . Culoarea returnată dinspre aceste direcții este de obicei o valoare de fond definită de utilizator.



**Figura 36** Arbore de raze

O altă metodă este să stabilim explicit adâncimea maximă absolută pentru arborele de raze, astfel încât să nu se genereze raze dincolo de acest nivel. De exemplu, dacă definim razele pornind din ochi ca fiind de nivel 0, atunci o adâncime maximă de 2 nu va permite razelor  $T_3$ ,  $R_3$  și  $R_4$  să contribuie la intensitatea recepționată în final.

Experiența ne arată că numai un procent mic dintr-o scenă tipică este format din suprafețe cu reflectanță mare sau foarte transparente, deci ar fi inefficient să urmărim fiecare rază până la adâncimea maximă. Ne-ar fi mult mai convenabil să utilizăm un control *adaptiv* al adâncimii, care să ia în considerație proprietățile materialelor cu care interacționează razele. Aceasta este a treia metodă.

Razele sunt atenuate de coeficienții de reflexie și transmisie speculară. Ca urmare, de exemplu, contribuția razei  $T_3$  la culoarea finală este atenuată cu produsul coeficienților  $k_{i3}k_{i2}k_{i1}$ . Dacă această valoare este sub un anumit prag, nu are sens să mergem cu urmărirea mai departe de raza  $T_3$ .



## 5.6. Intersecții cu obiecte

În această secțiune vom trata problema determinării punctului de intersecție a unei raze oarecare cu o anumită suprafață. Pentru aceasta trebuie să testăm fiecare suprafață definită în scena grafică și să găsim o intersecție care să fie cea mai apropiată de originea razei, adică să determinăm valoarea minimă pozitivă pentru parametrul  $t$  din Ecuația 5-1. Ca exemplu de calcul al intersecțiilor vom demonstra procedura pentru două tipuri uzuale de suprafețe: o sferă și un poligon.

### 5.6.1. Intersecția cu o sferă

Sfera este una dintre cele mai simple suprafețe și calculul punctului de intersecție cu o rază oarecare este destul de ușor. Definim o sferă cu ajutorul a doi parametri: centrul său, prin vectorul de poziție  $\mathbf{C}$ , și raza  $R$ . Punctele de pe suprafața sferei, având vectorul de poziție  $\mathbf{r}$ , satisfac ecuația

$$(\mathbf{r} - \mathbf{C}) \cdot (\mathbf{r} - \mathbf{C}) = R^2 \quad (5-8)$$

Aceasta este evident, întrucât în partea stângă,  $\mathbf{r} - \mathbf{C}$  este un vector cu originea în centrul sferei, și cu celălalt capăt pe suprafața sferei, deci lungimea lui este egală cu raza sferei. Produsul scalar al unui vector cu el însuși ne dă pătratul lungimii lui, de unde rezultă Ecuația 5-8.

Pentru determinarea punctului de intersecție substituim Ecuația 5-1 în Ecuația 5-8 și rezolvăm ecuația care ne rezultă pentru parametrul  $t$ . După puțină algebră vectorială obținem ecuația

$$t^2 - 2t\mu + [\lambda - R^2] = 0 \quad (5-9)$$

în care am definit

$$\begin{aligned} \mu &= \mathbf{D} \cdot \mathbf{T} \\ \lambda &= \mathbf{T} \cdot \mathbf{T} \\ \mathbf{T} &= \mathbf{C} - \mathbf{O} \end{aligned} \quad (5-10)$$

Această ecuație de gradul II se poate rezolva cu formula standard, și obținem

$$t = \mu \pm \sqrt{\gamma} \quad (5-11)$$

unde

$$\gamma = \mu^2 - \lambda + R^2$$

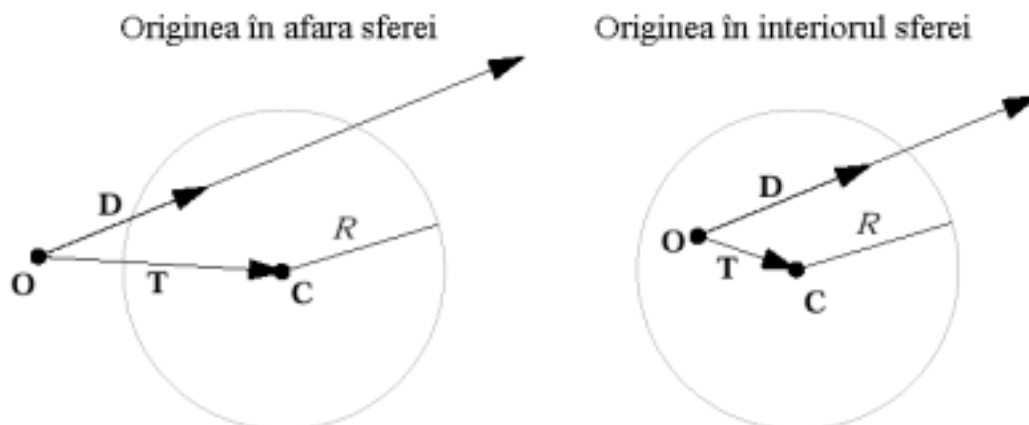
În funcție de semnul lui  $\gamma$  avem una dintre cele trei situații:

- $\gamma < 0$ : raza nu intersectează sfera
- $\gamma = 0$ : raza este tangentă la sferă
- $\gamma > 0$ : avem două intersecții distincte, cu  $t = t_1 = \mu + \sqrt{\gamma}$  și  $t = t_2 = \mu - \sqrt{\gamma}$

De obicei, primele două condiții se grupează împreună, întrucât o intersecție tangențială nu are sens.

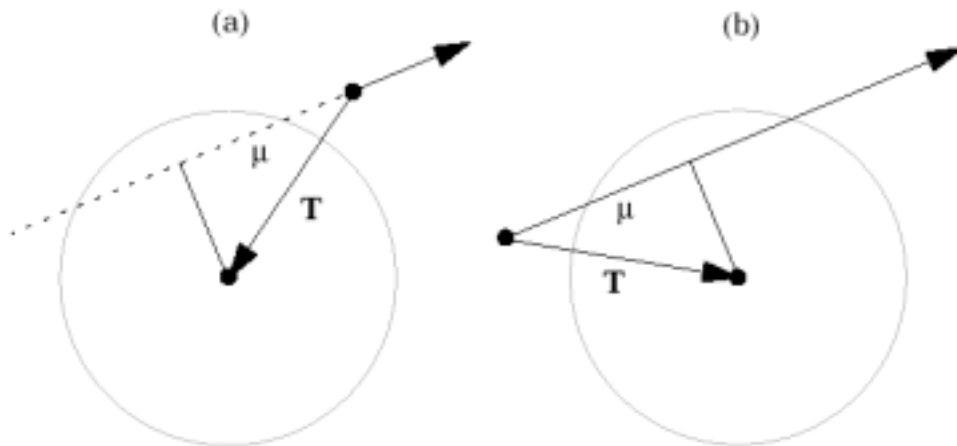
Se observă acum faptul că trebuie efectuate destul de multe calcule pentru a decide dacă avem vreo intersecție. Deseori aceasta se poate evita utilizând rezultatele intermediare, după cum se va vedea în continuare.

Mai întâi, trebuie să știm dacă originea razei se află în interiorul sferei. Pentru aceasta se calculează vectorul  $\mathbf{T}$  din Ecuația 5-10. Lungimea acestui vector este chiar distanța de la originea razei până la centrul sferei, pe care o vom compara deci cu raza sferei. Dacă raza are originea în interiorul sferei, ea trebuie să-i intersecteze suprafața. Altfel, originea se află pe sferă sau în afara ei, și putem să avem sau nu o intersecție. Aceste două cazuri sunt ilustrate în Figura 37. Convenim că o rază a cărei origine se află exact pe suprafața sferei se consideră că nu intersectează sfera în originea razei.



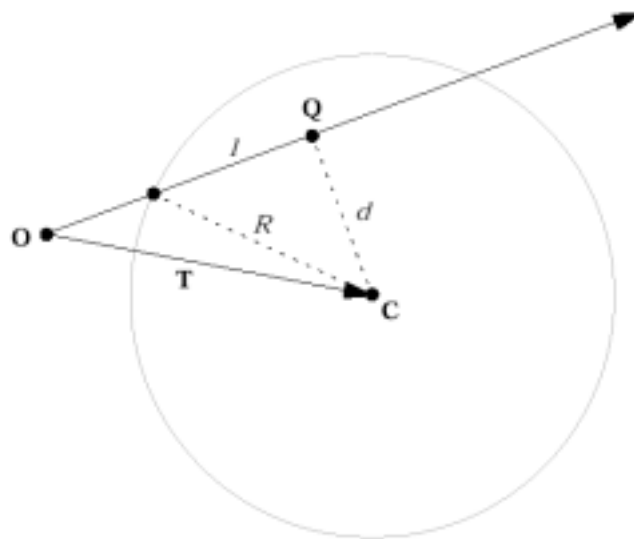
**Figura 37** Originea razei în raport cu sfera

Pasul următor este determinarea valorii parametrului  $t$  pentru punctul de pe rază cel mai apropiat de centrul sferei (vezi Figura 38, în care este prezentat cazul când originea razei este în afara sferei). Această distanță este dată de proiecția lui  $\mathbf{T}$  de-a lungul razei, adică  $\mathbf{D} \cdot \mathbf{T}$ , care, după cum am văzut, este egală cu  $\mu$  (avem acum o interpretare geometrică a parametrului  $\mu$ ). Dacă  $\mu \leq 0$ , intersecțiile, dacă acestea există, sunt plasate dincolo de originea razei, iar dacă originea este situată în afara suprafeței, raza nu poate intersecta sfera (Figura 38a). Ca urmare, putem opri calculul în acest punct. Dacă  $\mu > 0$  și originea este în afara sferei (Figura 38b), încă nu putem preciza dacă există vreo intersecție și trecem la următorul pas în algoritm.



**Figura 38** Punctul cel mai apropiat ( $t=\mu$ )

În continuare determinăm distanța de-a lungul razei dintre punctul cel mai apropiat și suprafața sferei. După cum se vede în Figura 39, ne folosim de două triunghiuri dreptunghice. Punctul **Q** este punctul cel mai apropiat, situat la distanța  $d$  de centrul sferei, și avem nevoie de distanța  $l$ .



**Figura 39** Determinarea punctelor de intersecție

Pe baza teoremei lui Pitagora,

$$l^2 = R^2 - d^2$$

și

$$d^2 = \lambda - \mu^2$$

întrucât  $OQ^2 = \mu^2$  și  $OC^2 = |\mathbf{T}|^2 = \mathbf{T} \cdot \mathbf{T} = \lambda$ , avem

$$l^2 = R^2 + \mu^2 - \lambda = \gamma$$

Acum se observă semnificația Ecuației 5-11:  $\mu$  este distanța până la punctul cel mai apropiat, în vreme ce rădăcina pătrată a lui  $\gamma$  este distanța (cu semn) de la punctul cel mai

apropiat până la punctele de intersecție, dacă acestea există. Acum putem testa valoarea lui  $\gamma$  pentru a verifica dacă există intersecții și care este poziția celei mai apropiate.

Dacă originea razei este în afara sferei, ne este necesară cea mai mică valoare a lui  $t$ ,

$$t = \mu - \sqrt{\gamma}$$

Dacă originea razei este situată pe sau în interiorul sferei, avem

$$t = \mu + \sqrt{\gamma}$$

Odată ce cunoaștem valoarea lui  $t$ , o putem insera în ecuația razei (Ecuația 5-1) și putem determina poziția punctului de intersecție.

### 5.6.2. Intersecția cu un poligon

Calculul intersecției razei cu un poligon se face în două etape:

- Găsirea intersecției razei cu *planul* în care este situat poligonul, și
- Dacă există vreo intersecție, testul dacă acest punct este în interiorul poligonului.

Planul în care este plasat poligonul este definit de ecuația

$$\mathbf{N} \cdot \mathbf{P} + d = 0 \quad (5-12)$$

unde  $\mathbf{N}$  este normala unitate,  $d$  este constanta planului iar  $\mathbf{P}$  este vectorul de poziție al unui punct oarecare din plan. Vectorul normala este egal cu normala poligonului și poate fi calculat din

$$\mathbf{N} = \frac{(\mathbf{V}_1 - \mathbf{V}_0) \times (\mathbf{V}_2 - \mathbf{V}_0)}{|(\mathbf{V}_1 - \mathbf{V}_0) \times (\mathbf{V}_2 - \mathbf{V}_0)|}$$

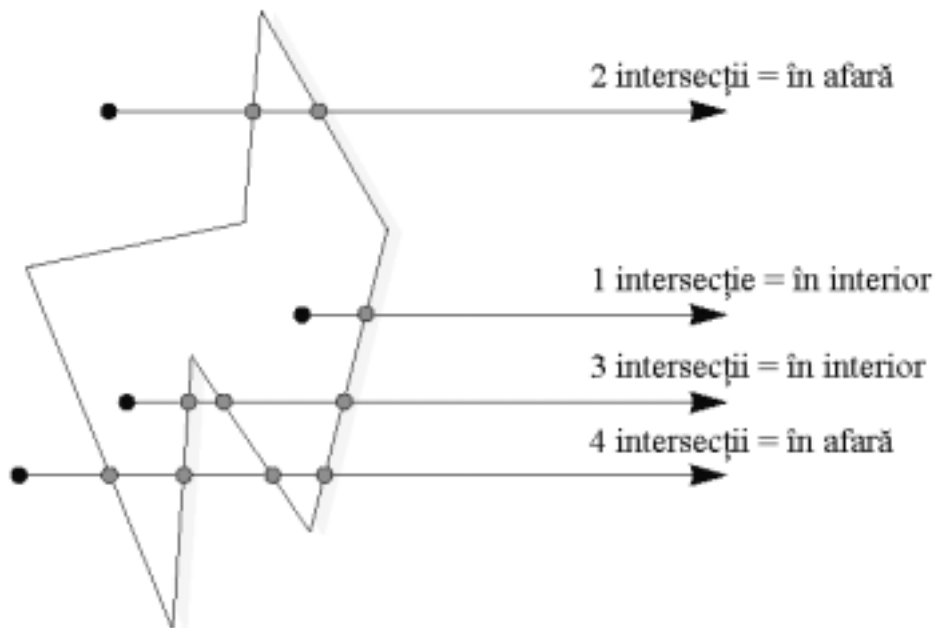
unde  $\mathbf{V}_0$ ,  $\mathbf{V}_1$  și  $\mathbf{V}_2$  sunt vectorii de poziție pentru trei dintre vârfurile poligonului. Constanta planului este dată de

$$d = -\mathbf{V}_0 \cdot \mathbf{N}$$

Din Ecuația 5-1 și Ecuația 5-12 putem evalua parametrul  $t$  corespunzător punctului de intersecție ca

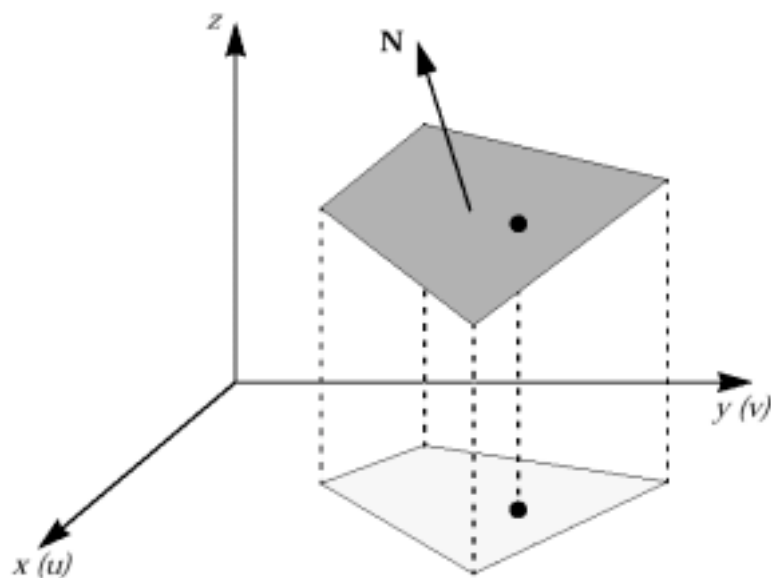
$$t = -\frac{d + \mathbf{N} \cdot \mathbf{O}}{\mathbf{N} \cdot \mathbf{D}} \quad (5-13)$$

De observat că dacă  $\mathbf{N} \cdot \mathbf{D} = 0$  atunci raza și planul (și deci poligonul) sunt paralele și nu există intersecție. Pe de altă parte, dacă  $t \leq 0$  intersecția este plasată în spatele originii razei și nu prezintă interes. În al treilea rând, dacă  $t$  este mai mare decât distanța de intersecție cea mai apropiată curentă, atunci intersecția actuală nu poate fi cea mai apropiată și de asemenea nu prezintă interes. Trecem la pasul al doilea numai dacă toate aceste teste eșuează.



**Figura 40** Teorema curbei lui Jordan

Acum vom determina dacă punctul de intersecție este plasat în interiorul poligonului. Pentru aceasta vom utiliza o idee cunoscută ca *teorema curbei lui Jordan*, ilustrată în Figura 40. Aici se prezintă un poligon complex bi-dimensional și un număr de puncte (cu negru) în jurul lui. Se trasează din fiecare punct o linie într-o direcție arbitrară și punctele de intersecție cu marginile poligonului sunt marcate cu gri. Teorema afirmă că dacă numărul de intersecții este *impar*, atunci punctul de plecare este situat *în interiorul* poligonului; altfel este în afară.



**Figura 41** Proiecția poligonului intersectat

Pentru a utiliza această teoremă, proiectăm poligonul pe unul din planele de coordonate (planul  $xy$ ,  $xz$  sau  $yz$ ). Pentru acuratețe, trebuie ales acel plan de proiecție care asigură cea mai mare arie a proiecției. Putem determina acest plan prin examinarea componentelor vectorului normala  $\mathbf{N}$ : dacă mărimea componentei  $z$  este cea mai mare, atunci vom face proiecția pe planul  $xy$ . Similar pentru celelalte componente. Proiecția pe un plan de coordonate se face simplu ignorând componenta corespunzătoare. De exemplu, dacă

$|\mathbf{N}_z|$  este componenta cea mai mare, proiectând punctul  $(x,y,z)=(a,b,c)$  pe planul  $xy$  ne dă punctul  $(u,v)=(a,b)$ ; vezi Figura 41. Aceleași operații le efectuăm asupra coordonatelor punctului de intersecție. Ajungem într-o situație similară cu cea din Figura 42a. Coordonate punctului de intersecție vor fi acum  $(u_p, v_p)$ .

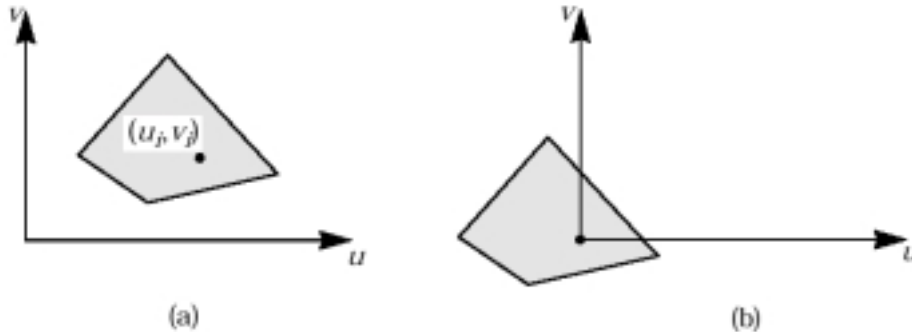


Figura 42 Testul de punct în interiorul poligonului

Acum efectuăm un test de interior translătând poligonul astfel încât punctul de intersecție să fie în origine (vezi Figura 42b). Aceasta se efectuează simplu prin scăderea coordonatelor punctului din coordonatele fiecărui vertex. Apoi utilizăm semiaxa pozitivă  $u$  pentru determinarea intersecțiilor cu fiecare latură a poligonului. Dacă numărul total de intersecții este impar, atunci din teorema curbei lui Jordan conchidem că punctul este situat în interiorul poligonului.

Ce se întâmplă dacă semiaxa  $u$  pozitivă trece exact printr-un vertex? Acest caz special poate fi evitat în felul următor: semiaxa  $u$  divide planul în două părți și declarăm vârful situat pe această linie ca fiind de partea lui  $+u$ . De fapt, redefinim axa  $u$  ca fiind apropiată infinitesimal de axa originală, dar nu trece prin nici un vârf.

## 5.7. Tehnici de accelerare

Căutarea unei intersecții pentru fiecare suprafață din scena grafică este o operație mare consumatoare de timp, și devine cu atât mai costisitoare cu cât crește numărul de obiecte din scenă. De asemenea, operația este cu atât mai costisitoare cu cât crește rezoluția imaginii, întrucât sunt generate mai multe raze care pornesc din punctul de observație. Estimările efectuate duc la concluzia că 90% din timpul de calcul este consumat de compararea rayelor cu obiectele din scenă. Ca urmare, ar fi avantajos să găsim modalități de reducere a timpului necesar determinării celei mai apropiate intersecții și deci să accelerăm calculul pentru o imagine. Există o serie de abordări, care fie limitează numărul de teste care se efectuează, fie amână un test costisitor până în momentul în care un test mai puțin costisitor confirmă necesitatea lui.

### 5.7.1. Volume de încadrare ierarhice

La începuturile ray-tracing-ului a fost dezvoltată o tehnică bazată pe conceptul de *volume de încadrare (bounding volumes)*. Volumul de încadrare al unui obiect este o primitivă simplă (de ex. o sferă sau un paralelipiped) care ocupă cel mai mic volum și care să încadreze obiectul. Pe parcursul testului de intersecție a unei raze cu un obiect, se efectuează mai întâi un test pentru volumul de încadrare, test care de obicei este mai simplu și mai rapid. Numai dacă volumul de încadrare este intersectat de rază se aplică testul de intersecție pentru obiectul respectiv.

Utilizarea unei sfere ca volum de încadrare este tehnica cea mai uzuală, iar intersecția unei raze cu o sferă este în particular cea mai eficientă, dacă ținem cont de faptul că ne

este necesară doar informația despre existența intersecției, fără să calculăm punctul de intersecție. Totuși, calculul parametrilor (centrul și raza) pentru o sferă de încadrare nu este chiar atât de simplu. În cazul ideal, sfera cu cel mai mic volum se determină modificând centrul și raza și determinând raza minimă pentru care este încadrat întregul obiect. În practică, se face o bună aproximare calculând centroidul obiectului, acesta urmând să fie centrul sferei, și apoi determinând distanța până la cel mai îndepărtat punct, aceasta fiind raza.

Volumele de încadrare pot fi utilizate într-o structură ierarhică, în care grupuri de volume de încadrare sunt la rândul lor încadrate într-un volum mai mare. Testul de intersecție începe prin detectarea unei intersecții cu volumul exterior și apoi decizia dacă trebuie să continuăm cu interiorul acestuia, efectuând teste pentru volumele interioare. Desigur, putem avea volume pe oricâte nivele. Această tehnică funcționează foarte bine pentru scene conținând obiecte care nu sunt uniform distribuite, astfel încât pentru o rază oarecare să nu avem prea multe calcule în zone ale scenei înspre care raza nu este îndreptată. O asemenea schemă ierarhică de volume de încadrare duce la o accelerare considerabilă a timpului de reprezentare.

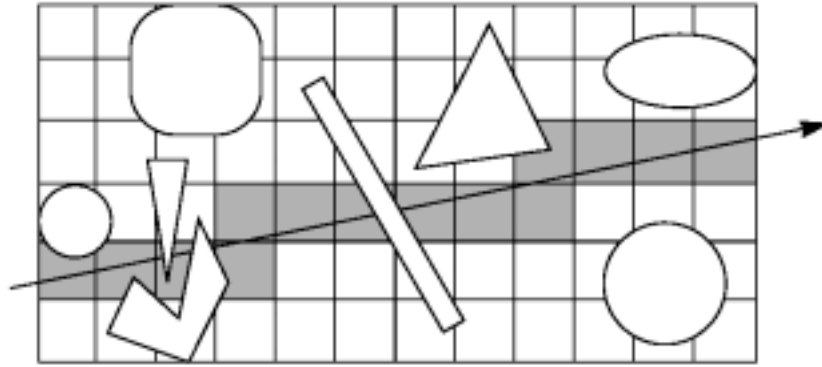
### 5.7.2. Subdivizarea 3D spațială

Au fost dezvoltate și alte scheme de accelerare, mai puternice, dar care nu sunt atât de simple de implementat. Aceste metode se bazează pe ideea de *voxeli*, care sunt prisme rectangulare aliniată cu axele, similare cu volumele de încadrare, cu diferența că ele umplu tot spațiul ocupat de scena grafică, fără să se suprapună. Există două abordări ale acestei metode: *uniformă* și *ne-uniformă*.

Într-o aranjare uniformă înconjurăm întreaga scenă cu un cub de încadrare, pe care apoi îl descompunem în  $L \times M \times N$  cuburi mai mici, care sunt voxelii. Pentru fiecare voxel păstrăm o listă a obiectelor care îi intersectează spațiul. Apoi, atunci când urmărim o rază prin scenă, trebuie să examinăm numai obiectele aparținând voxelilor prin care trece raza. În Figura 43 este prezentat un exemplu 2D a subdivizării spațiale uniforme pentru o scenă conținând 8 obiecte și o rază care trece prin scenă. Voxelii umbriți corespund celor prin care trece raza; se observă că raza intersectează un număr de obiecte.

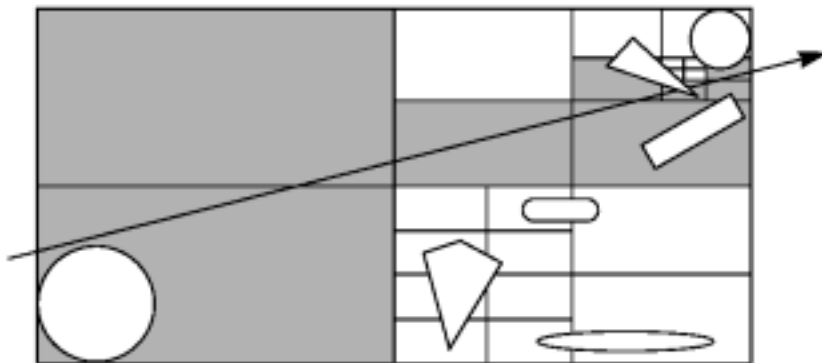
Urmărim raza voxel cu voxel. Primul voxel are asociat un obiect, dar testul arată că suprafața lui nu este intersectată. Trecem la următorul voxel, care conține două obiecte, dintre care unul a fost deja testat și s-a determinat că nu este intersectat. Această informație trebuia reținută în momentul când s-a efectuat primul test de intersecție. Testele de intersecție vor indica faptul că poligonul este intersectat, dar punctul de intersecție este situat înafara voxelului curent, deci nu putem fi siguri că este vorba despre cea mai apropiată intersecție. Din diagramă se observă că nu este cea mai apropiată intersecție, dar calculatorul trebuie să descopere aceasta! Trecem mai departe, dar păstrăm această informație pentru a evita reluarea calculelor. Următorul voxel conține un triunghi și poligonul anterior și ajungem la concluzia că triunghiul este primul întâlnit, și punctul de intersecție este în voxelul curent. Am determinat deci cea mai apropiată intersecție.

Se poate observa că raza intersectează și un alt obiect (de fapt puteau să fie mult mai multe obiecte), dar nu a fost necesar să efectuăm și alte teste de intersecție, cum ar fi fost în absența acestei metode de accelerare. În exemplul nostru ne-au fost necesare numai trei teste de intersecție din 8 posibile. Într-o scenă 3D mai realistă, conținând probabil sute de mii de obiecte, restrângerea testelor de intersecție la obiectele din vecinătatea traiectoriei razei poate reduce semnificativ timpul de generare a imaginii.



**Figura 43** Subdivizare spațială uniformă (analogie 2D)

În cadrul metodei ne-uniforme de subdivizare a spațiului 3D, scopul este să se genereze o partiționare care să sublinieze distribuția obiectelor în scenă. Structura utilizată în acest caz este denumită *octree*, care este un arbore ierarhic de voxelii care nu se suprapun, de mărimi diferite. Octree-ul este construit începând cu un cub de încadrare a întregii scene. Acest volum este apoi divizat în *opt* sub-volume, sau voxelii, de aceeași mărime, și pentru fiecare voxel se determină câte o listă a obiectelor asociate. Apoi, în funcție de atingerea unei adâncimi maxime a arborelui, fiecare voxel este subdivizat la rândul său în *opt*, dacă numărul de obiecte din acel voxel depășește un maxim prestabilit. În acest mod, subdivizarea în voxelii mai mici apare numai atunci când avem multe obiecte. În regiunile scenei în care nu avem obiecte sau foarte puține, voxelii sunt mari.



**Figura 44** Subdivizare spațială ne-uniformă (analogie 2D)

Vom demonstra utilizarea unui octree prin analogie cu situația din 2D, în care subdivizarea produce o structură denumită *quadtree* (vezi Figura 44). Din nou, voxelii umbriți sunt cei traversați de rază, iar subdivizarea continuă până când avem cel mult un obiect per voxel.

Procedura pentru determinarea celei mai apropiate intersecții este similară ca și în cazul subdivizării uniforme: procesăm în ordine numai acei voxelii prin care trece raza, până când găsim un punct de intersecție în voxelul curent. În exemplul nostru, se trece prin 6 voxelii și se efectuează numai trei teste distincte de intersecție. Se observă că regiuni mari ale scenei sunt goale, indicate de voxelii mari, prin care raza trece foarte rapid. Testele utile pentru găsirea unei intersecții se efectuează numai în zonele unde avem concentrare mare de obiecte.

În oricare dintre metodele de subdivizare spațială de obicei se face un compromis între finețea subdivizării scenei (influențată de distribuția obiectelor în scenă și de numărul maxim de obiecte per voxel) și timpul necesar trecerii de la un voxel la altul de-a lungul razei. În mod normal, aceste aspecte pot fi controlate de utilizator și pot fi ajustate în



vederea atingerii eficienței optime. În acest mod se pot obține reduceri semnificative ale timpului de generare a imaginii, în special pentru scene complexe, conținând un număr mare de obiecte.

Ca o indicație a performanțelor care pot fi atinse, cei care au dezvoltat metoda de subdivizare uniformă au anunțat că o scenă conținând 10 584 de obiecte, a fost prelucrată în 15 minute, în timp ce metoda tradițională ar fi necesitat 40 de zile!

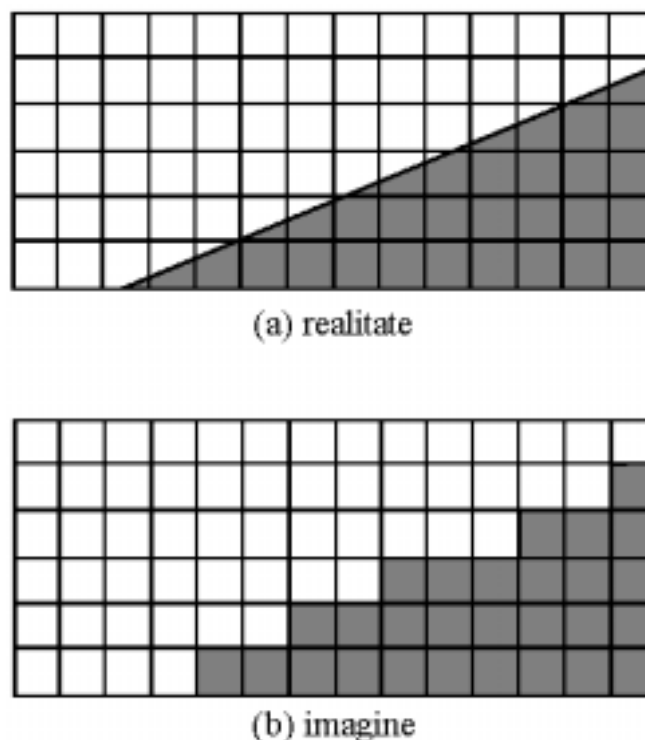
## 5.8. Aliasing

După cum am văzut, atunci când utilizăm tehnica ray-tracing pentru generarea unei imagini sintetice urmărim o rază pornind din ochi și trecând prin fiecare pixel din planul de vedere. De fapt, am eșantionat planul de vedere pe un număr de direcții discrete și acest proces de eșantionare introduce în imagine un efect denumit *aliasing*.

El este o consecință directă a încercării de a reprezenta un semnal continuu (distribuția intensității luminoase pe suprafața planului de vedere) în termenii unei secvențe de eșantioane, discrete: dacă încercăm să reconstruim semnalul pe baza eșantioanelor noastre (de exemplu prin afișarea eșantioanelor pe un ecran grafic), vor apărea distorsiuni și reconstrucția nu va fi identică cu originalul. Din teoria procesării semnalelor, se cunoaște faptul că dacă dorim să eșantionăm un semnal fără pierderi sau distorsiuni, atunci trebuie să-l eșantionăm cu o rată care este cel puțin dublul celei mai înalte frecvențe din semnalul original (aceasta este *teorema lui Nyquist*, iar frecvența minimă de eșantionare este denumită *limita Nyquist*). Totuși, întrucât semnalul reprezentând scena noastră are un domeniu infinit de frecvențe spațiale, nici o frecvență finită de eșantionare nu va fi suficient de înaltă încât să evite apariția aliasing-ului. Fenomenul de aliasing face ca frecvențele mai mari decât limita Nyquist să apară în semnalul reconstruit cu frecvențe mai mici: de fapt, frecvențele mari sunt înlocuite cu frecvențe scăzute. De aici provine termenul de aliasing.

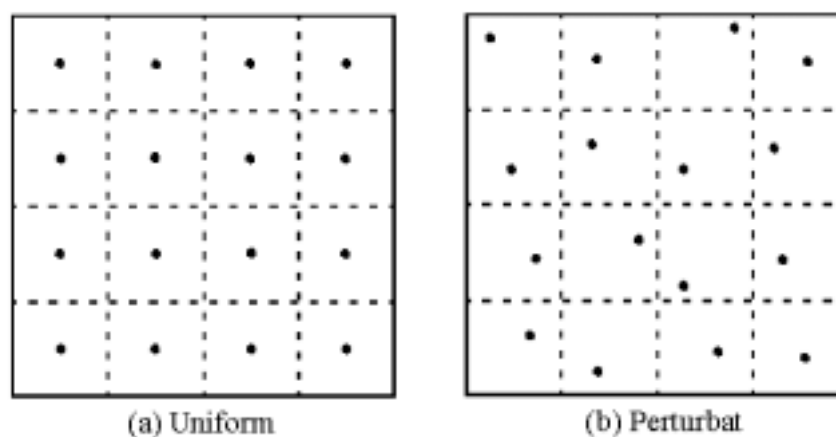
### 5.8.1. Anti-aliasing

În termeni practici, fenomenul de aliasing într-o imagine generată cu ray-tracing face ca muchiile netede (de exemplu de-a lungul marginilor suprafețelor) să apară cu asperități. Acest efect este ilustrat în Figura 45, unde în (a) este prezentată o porțiune dintr-o imagine cu limitele pixelilor, și proiecția unei părți dintr-un obiect pe planul imaginii, iar în (b) este prezentat rezultatul eșantionării proiecției în centrul fiecărui pixel.



**Figura 45** Aliasing

În termeni simpli, se poate observa că aliasing-ul provine din reprezentarea unei *regiuni finite* a imaginii (cea acoperită de un pixel), printr-un singur punct. Se mai poate observa că ar fi posibil să reducem efectele aliasing-ului dacă am putea reduce dimensiunea acestei regiuni. Acest lucru este posibil, și o modalitate de a reduce suprafața pixelului este să mărim rezoluția imaginii, deci să mărim rata de eșantionare. În practică, această tehnică de *anti-aliasing* este implementată printr-un proces echivalent de trimitere a mai multor raze prin același pixel, adică *supraeșantionăm* imaginea (*supersampling*). Ne stau la dispoziție două metode de distribuție a eșantioanelor în cadrul unui pixel: acestea efectuează eșantionarea *uniformă* și *perturbată* (*jittered*).

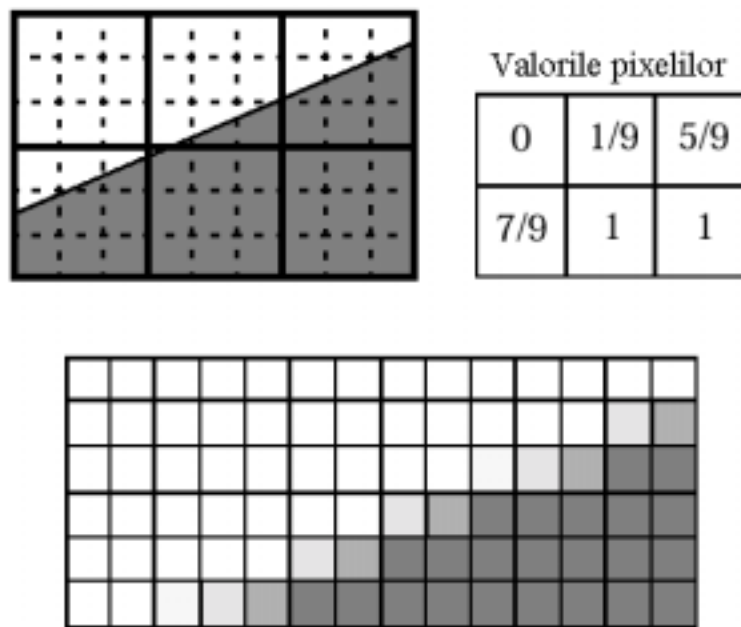


**Figura 46** Distribuția supraeșantionării

În supraeșantionarea uniformă, fiecare pixel este subdivizat într-un tablou de sub-pixeli, iar imaginea este eșantionată în centrul fiecărui sub-pixel, în vreme ce cu metoda perturbată punctele de eșantionare se aleg la distanțe mici aleatoare de poziția originală. În Figura 46 este prezentat câte un exemplu al celor două tipuri de eșantionare: în (a) un

pixel este supraeșantionat uniform utilizând 16 sub-pixeli, iar în (b) avem aceeași grilă, pe care s-a făcut supraeșantionarea cu perturbații.

După eșantionarea pixelului și trasarea razelor prin punctele de eșantionare, trebuie să combinăm intensitățile returnate de fiecare eșantion, pentru a determina valoarea reprezentativă pentru întregul pixel. Această procesare a eșantioanelor este denumită *filtrare* și se poate efectua în mai multe modalități. Varianta cea mai simplă ar fi să mediem eșantioanele (*“box filtering”*), și ea dă rezultate satisfăcătoare în majoritatea cazurilor. Se pot utiliza filtre mai realiste, cum ar fi un filtru gaussian, care ponderează valorile eșantioanelor în cadrul medierii, în funcție de distanța de centrul pixelului original. Acesta este de fapt un filtru util pentru imaginile de afișat pe un ecran grafic, întrucât redă foarte bine variația intensității pe suprafața unui pixel produs de fascicolul de electroni.



**Figura 47** Antialiasing cu supraeșantionare uniformă

În Figura 47 este prezentat un exemplu de supraeșantionare uniformă aplicată imaginii din Figura 45. Supraeșantionăm cu 9 valori per pixel și utilizăm un box filter pentru calculul valorilor în pixel. În partea superioară a diagramei este prezentată o porțiune mărită, cu cei 6 pixeli din regiunea centrului imaginii, împreună cu delimitarea sub-pixelilor. Numerele din tabelul alăturat sunt valorile pixelilor obținute prin eșantionarea centrului fiecărui sub-pixel și apoi mediere. Efectele acestei tehnici simple de anti-aliasing sunt prezentate în partea de jos din Figura 47 și se observă că muchia pare mai puțin în trepte: avem o tranziție mai lină de la pixelii care conțin numai obiectul la cei care nu conțin deloc obiectul.

Una dintre deficiențele supraeșantionării uniforme, în special dacă densitatea de eșantionare este redusă, și anume aceea că efectele aliasing-ului pot să rămână, deși la un nivel perceptual redus. Aparența de muchie cu asperități poate fi încă detectabilă, în particular în cazul muchiilor cu gradienti mici sau mari relativ la reaster-ul imaginii. Utilizarea eșantionării aleatoare îmbunătățește această situație, întrucât frecvențele de aliasing sunt înlocuite de zgomot în imagine, cu care sistemul vizual uman este mult mai tolerant.

### 5.8.2. Supraeșantionarea adaptivă

Până în acest punct al discuției despre aliasing am presupus că se aplică o tehnică anti-aliasing la fiecare pixel din imagine, indiferent de conținutul său. Totuși, este posibil să nu fie necesar ca un anumit pixel să fie supraeșantionat, întrucât nu toate porțiunile imaginii conțin muchii sau, mai general, modificări mari de contrast. Aceasta înseamnă că se depune mai mult efort decât este necesar: mulți pixeli vor avea, de exemplu, 9 eșantioane, în vreme ce este necesar doar unul.

Pentru aceasta, avem nevoie de o metodă de anti-aliasing care să concentreze majoritatea eforturilor în regiunile imaginii care necesită atenție: aceste metode sunt denumite *supraeșantionare adaptivă*. Prin aceasta, încercăm să localizăm acei pixeli care necesită atenție specială. Putem să rezolvăm aceasta mai întâi prin trasarea razelor prin colțurile pixelilor, nu prin centrele lor, și pentru fiecare pixel să examinăm diferențele în intensități între cele 4 eșantioane. Dacă variația pe pixel este prea mare, atunci divizăm pixelul în 4 sub-pixeli și reluăm procesul de câte ori este necesar. Pentru decizia dacă contrastul dintre eșantioane impune sau nu subdivizarea în continuare, putem utiliza următoarea formulă

$$C = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}$$

unde  $I_{\max}$  este valoarea cea mai mare returnată, iar  $I_{\min}$  este cea mai mică. Trebuie determinat valorile lui  $C$  pentru fiecare culoare primară, de obicei R, G și B, și dacă oricare dintre acestea depășește o valoare prestabilită se efectuează un nou nivel de supraeșantionare. Valori tipice de prag pentru R, G și B sunt 0.25, 0.20 și respectiv 0.4. În Figura 48 este prezentată o schemă de anti-aliasing adaptiv, cu trei nivele de supraeșantionare.

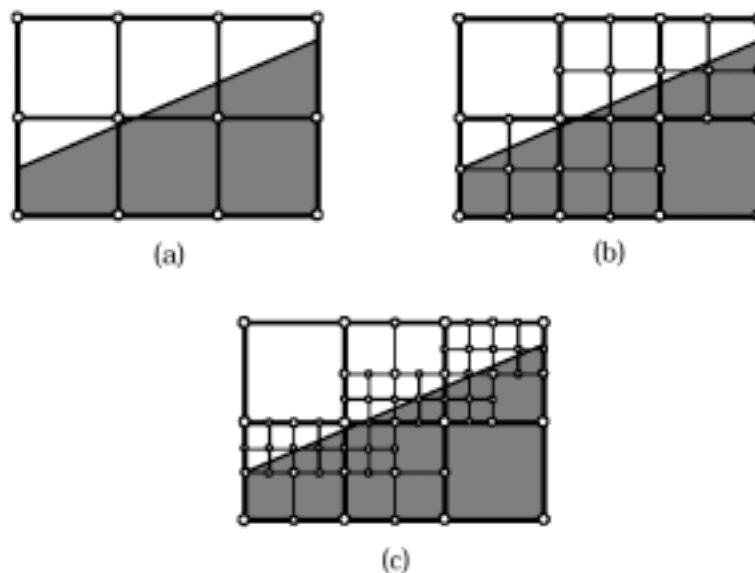
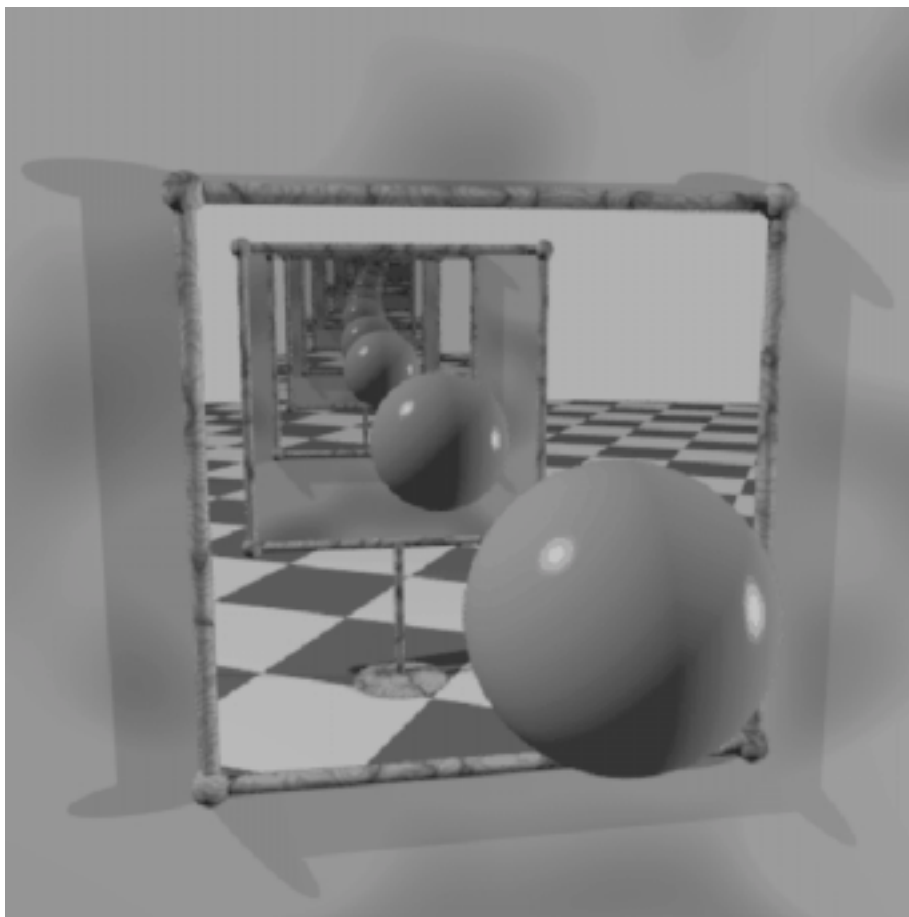


Figura 48 Antialiasing adaptiv

De remarcat mai întâi că la trecerea de la un nivel de supraeșantionare la următorul, putem reutiliza multe dintre eșantioanele deja calculate. De asemenea, este evident că ne sunt necesare mult mai puține eșantioane decât în cazul schemei de eșantionare globală. În exemplul prezentat, dacă am fi supraeșantionat toți cei 6 pixeli la același nivel, ar fi trebuit să urmărim 117 raze, în comparație cele 62 din cazul metodei adaptive. Desigur, aceasta economisește substanțial din timpul de generare a imaginii.

## 5.9. Exemple de imagini

În această secțiune finală vom prezenta câteva imagini sintetice calculate utilizând tehnicile prezentate în acest capitol<sup>1</sup>. Prima imagine este prezentată în Figura 49: o scenă stilizată conținând o minge mare netedă plasată între două oglinzi, una pe perete și cealaltă plasată în spatele punctului de observație. Cea de-a doua oglindă nu poate fi văzută, dar se observă reflexia ei în cealaltă oglindă.



**Figura 49** Suprafețe puternic speculare

Scena conține 9 sfere (o minge mare și 4 bile mai mici în colțurile fiecărei oglinzi), 9 cilindri (ramele oglinzilor și piciorul oglinzii din spatele punctului de observație), 2 poligoane (oglinzile), un plan (peretele) și 2 discuri (“talpa” piciorului oglinzii). De remarcat modul în care unele suprafețe au fost texturate: podeaua are un tipar în formă de tablă de șah, ramele oglinzilor au o textură lemnoasă, iar peretele este mat. Toate suprafețele au fost definite ca fiind pur difuze, cu excepția sferei mari, care mai are o componentă locală speculară de 0.9, și a oglinzilor, care au doar reflectivitate speculară globală de 0.9. Faptul că reflectivitatea oglinzilor nu este unitară produce o întunecare a imaginii pe direcția reflexiilor multiple.

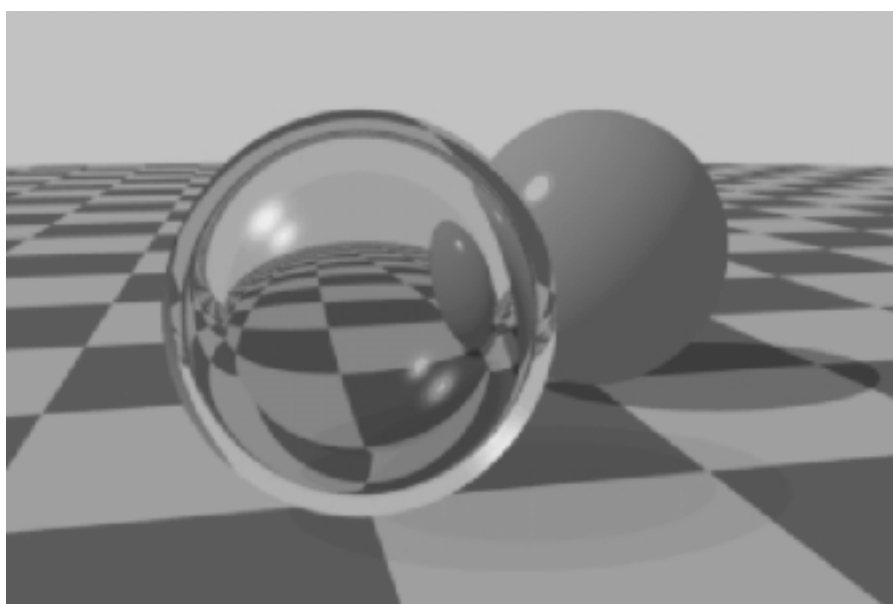
Mai avem două surse punctiforme de lumină, de intensitate 1, și o sursă de lumină ambiantă cu intensitatea 0.3. De observat umbrele de pe perete și podea, care ar fi fost reprezentate foarte întunecat în absența iluminării ambiante. Arborele de raze a fost

<sup>1</sup> Toate imaginile au fost generate inițial ca imagini full-colour pe 24 de biți, însă, pentru tipărire, fiecare a fost convertită la nivele de gri (maximum 256 de nivele). Ca urmare, calitatea imaginii s-a diminuat semnificativ și nu poate fi comparată cu originalul. Oricum, fișierele cu imaginile originale sunt disponibile.

parcurs până la adâncimea 12, pentru a se putea observa reflexiile multiple între oglinzi. Imaginea a fost generată la o rezoluție de 500×500 pixeli, utilizând o supraeșantionare adaptivă cu 16 eșantioane aleatoare per pixel. Apoi s-a utilizat un filtru gaussian pentru reconstrucția valorilor pixelilor.

Figura 50 prezintă un alt exemplu, care ilustrează posibilitățile ray-tracing-ului de a reprezenta medii transparente. Obiectul frontal a fost modelat ca un glob sferic de sticlă, cu un index de refracție de 1.3 și o transmisivitate speculară globală de 0.99. Suprafața de sticlă mai are de asemenea un coeficient de reflexie speculară locală de 0.9, care produce pete speculare luminoase atât pe exteriorul cât și pe interiorul sferei. Se observă foarte clar refracția luminii.

Scena conține o singură sursă de lumină, plus cea ambientă, la un nivel similar celui din imaginea precedentă. Razele au fost urmărite până la o adâncime de 5 și reprezentate la 750×500 de pixeli, utilizând o supraeșantionare adaptivă cu 9 eșantioane perturbate per pixel.



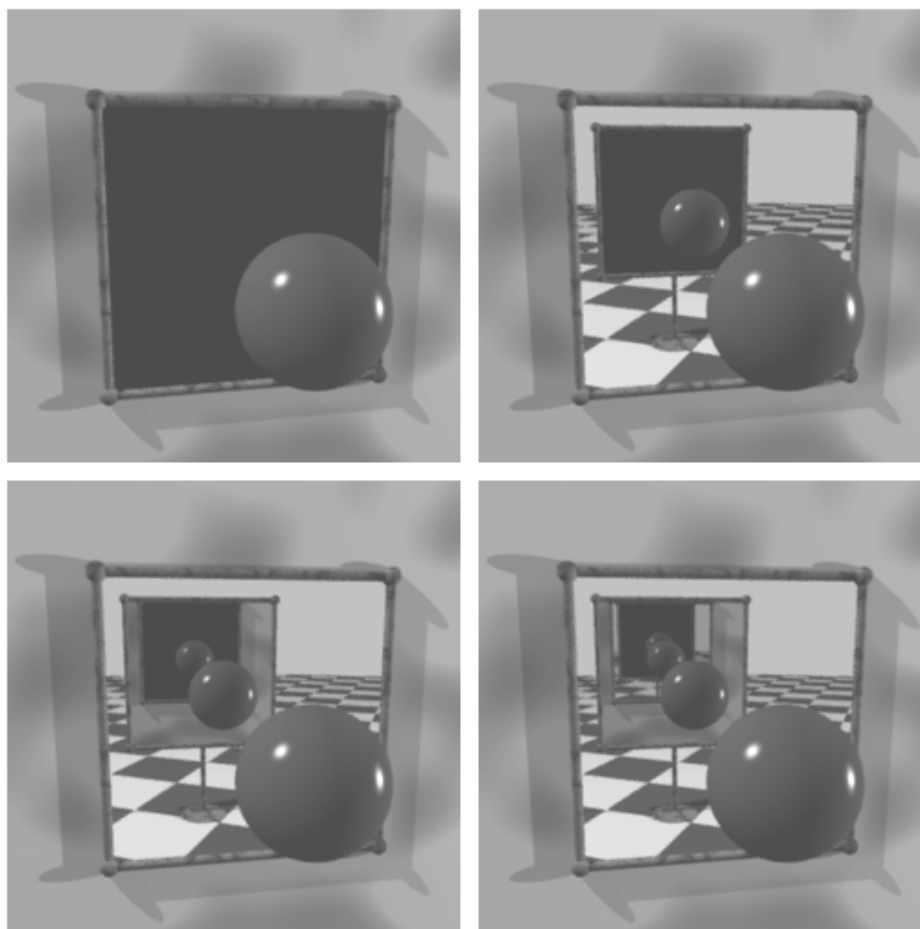
**Figura 50** Suprafețe transparente

Un aspect important de remarcat la această imagine este apariția umbrei în spatele obiectului de sticlă. Ceea ce se vede nu este deloc realist, din mai multe motive. Mai întâi, nu s-a ținut cont de refracția luminii prin sticlă. Intensitatea luminii în zona de umbră a fost calculată printr-o metodă similară celei descrise în Secțiunea 4.4 și ia în considerare numai atenuarea luminii de-a lungul unei drepte care traversează sticla (se observă cum zona centrală a umbrei este mai întunecată decât la periferie, unde parcurge un drum mai lung prin sticlă). De fapt, pe baza ray-tracing-ului convențional, este extrem de dificil de calculat umbre realiste în spatele obiectelor transparente. Aceste genuri de umbre, denumite *caustice*, sunt în realitate constituite dintr-un tipar de lumină și umbră determinat de efectul de focalizare al sticlei.

Pornind de la imaginea prezentată în Figura 49, vom ilustra efectul trunchierii premature a adâncimii arborelui de raze, și de asemenea cum este afectată calitatea imaginii de fenomenul de aliasing.

Figura 51 conține patru imagini ale aceleiași scene. Prima imagine are o adâncime 0, indicând că sunt urmărite numai razele pornind de la ochi: se efectuează numai îndepărtarea suprafețelor ascunse. De remarcat cum oglinda de pe perete este neagră,

întrucât nu se generează raze reflectate. Restul imaginii arată neschimbat, fiind compusă numai din suprafețe cu difuzie. În următoarea imagine adâncimea este 1, deci avem o reflexie în oglinda de pe perete, dar nimic în cealaltă oglindă. Crescând adâncimea, se mărește numărul de reflexii generate, crescând realismul imaginii.



**Figura 51** Ray-tracing la adâncimi de 0, 1, 2 și 3

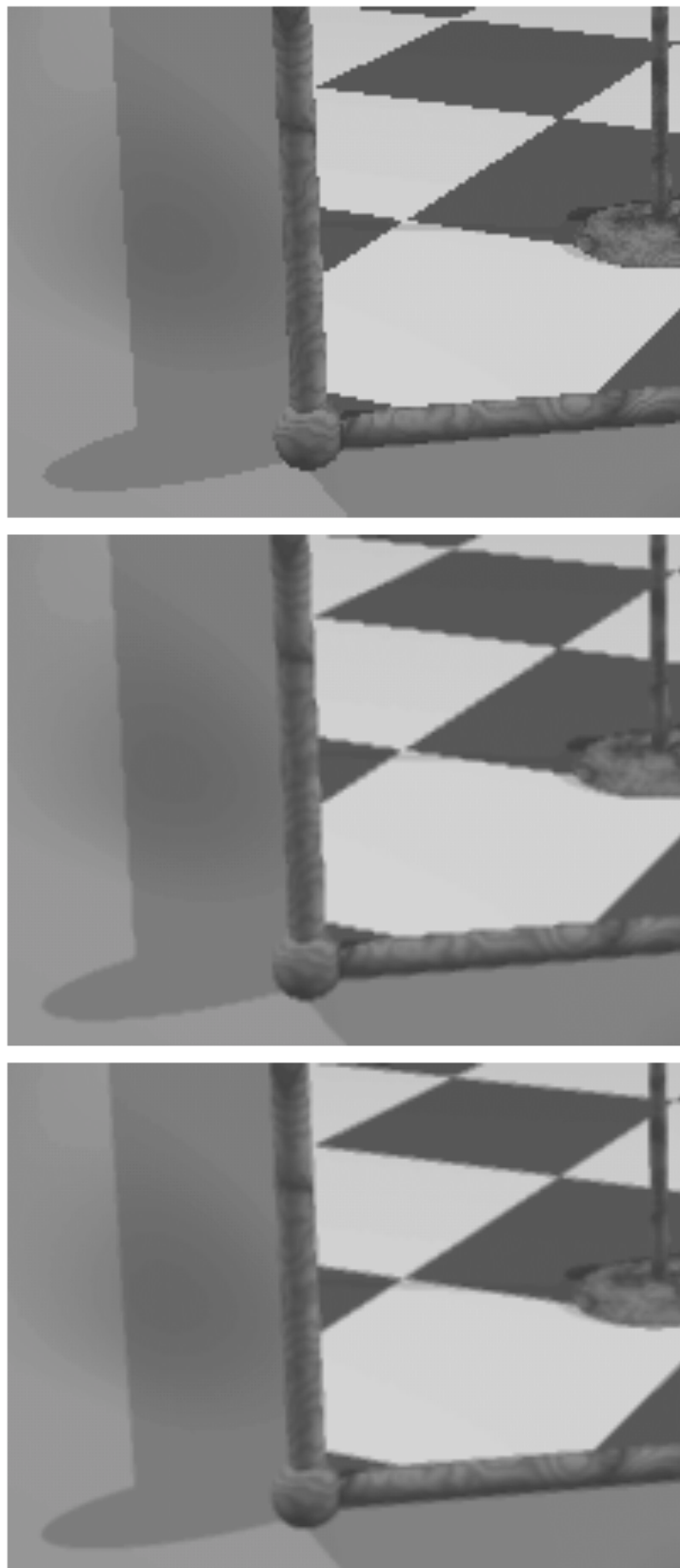
Pentru a ilustra efectul de aliasing, vom reprezenta o mică porțiune a imaginii din Figura 49, cu și fără anti-aliasing. De asemenea vom investiga efectul diferitelor metode de supraeșantionare.

În Figura 52 sunt prezentate trei imagini care au fost calculate utilizând supraeșantionarea uniformă. În prima imagine s-a utilizat o singură rază per pixel și se observă clar aliasing-ul. A doua și a treia imagine au utilizat 4, respectiv 9 eșantioane per pixel. De remarcat îmbunătățirea graduală a calității imaginii, însă cu prețul unei ușoare încețoșări în zona muchiilor. Utilizarea a 9 eșantioane per pixel pare adecvată în acest exemplu. Dacă privim cu atenție, se mai observă aliasing-ul, dar trebuie să ținem cont de faptul că este o imagine mărită; aliasing-ul va fi insesizabil în imaginea întregă. În toate aceste imagini a fost setat un prag de contrast redus, pentru a forța supraeșantionarea de-a lungul tuturor laturilor.

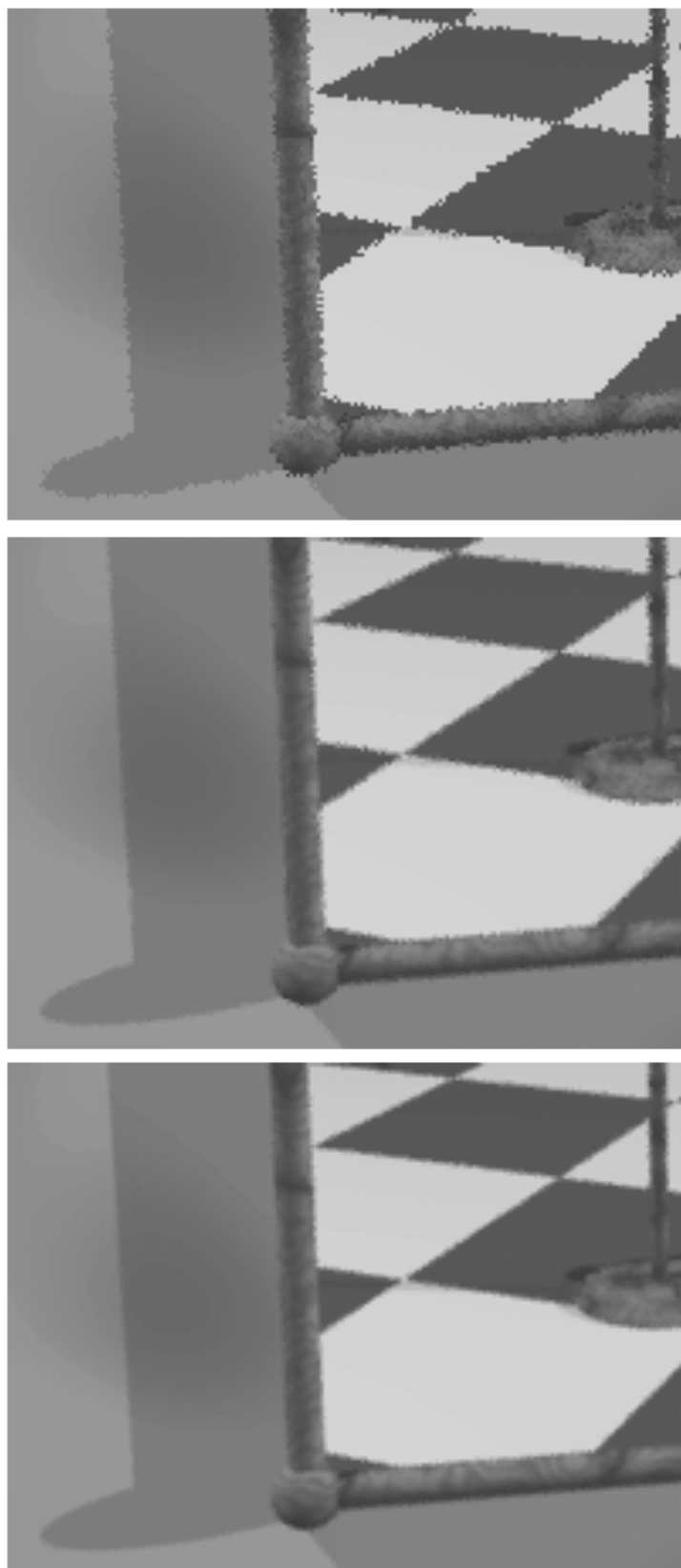
Acum să comparăm rezultatele cu supraeșantionarea perturbată din Figura 53. Din nou, au fost utilizate 1, 4 și 9 eșantioane per pixel. Putem utiliza perturbarea chiar și în cazul unui singur eșantion per pixel, dar rezultatele nu vor fi prea plăcute! Din nou, se observă îmbunătățirea calității imaginii la utilizarea mai multor eșantioane per pixel. În plus, toate urmele de aliasing au dispărut prin eșantionarea perturbată; aliasing-ul este înlocuit de

zgomot. Însă numai atunci când sunt utilizate suficiente eșantioane per pixel nivelul zgomotului este redus la un nivel acceptabil. Rezultatul utilizării de 9 eșantioane perturbate vor fi virtual imposibil de distins de cazul când se utilizează 9 eșantioane uniforme, atunci când privim întreaga imagine.





**Figura 52** Supraeșantionare uniformă cu 1, 4 și 9 eșantioane per pixel



**Figura 53** Supraeșantionare perturbată cu 1, 4 și 9 eșantioane per pixel

## 6. Radiozitatea clasică

*Radiozitatea* unei suprafețe este o măsură a ratei cu care energia părăsește acea suprafață. În același mod în care descrie un flux fizic, termenul de radiozitate este utilizat în grafica pe calculator pentru a descrie algoritmi care calculează distribuția luminii într-un mediu închis, pe baza principiului conservării energiei.

Metodele radiozității încearcă să modeleze cu acuratețe efectele luminii ambiante. Anterior, am considerat componenta ambiantă în modelul de iluminare ca fiind  $I_a k_{d,i}$ , unde  $I_a$  este constantă pentru întreaga scenă. În practică lucrurile nu stau tocmai așa, astfel încât modelul radiozității a fost dezvoltat în încercarea de a exprima cât mai exact acest termen.

### 6.1. Ecuația radiozității

Atunci când aplicăm metoda radiozității în problema generării imaginii unui spațiu închis, principala presupunere care se face este aceea că toate suprafețele reflectă sau emit difuz ideal. Această presupunere, deși ne limitează la medii difuze (făcând imposibilă reprezentarea suprafețelor metalice sau a oglinzilor), ne oferă o tehnică de reprezentare independentă de punctul de observație. Odată calculată distribuția luminii prin mediu, singurul pas necesar pentru reprezentarea diferitelor vederi ale scenei este aplicarea tehnicilor de îndepărtare a suprafețelor ascunse.

De asemenea, metoda radiozității presupune că scena este compusă dintr-un număr de porțiuni (patch-uri) discrete, fiecare cu radiozitate uniformă. Astfel, ținând cont de echilibrul energiei în scenă, putem scrie

$$A_i B_i = A_i E_i + \rho_i \sum_{j=1}^n B_j F_{ji} A_j \quad (6-1)$$

unde

- $B_i$  este radiozitatea patch-ului  $i$ , măsurată în energie/unitatea de timp/unitatea de suprafață (de ex.  $\text{Wm}^{-2}$ ) și reprezintă cantitatea totală de energie (emisă și reflectată) care părăsește unitatea de suprafață a patch-ului în unitatea de timp
- $A_i$  este aria patch-ului
- $E_i$  este lumina emisă de patch (de asemenea în  $\text{Wm}^{-2}$ )
- $\rho_i$  este reflectivitatea difuză a patch-ului (cu aceeași semnificație ca și  $k_{d,i}$ ), care este fracțiunea din energia incidentă suprafeței care este reflectată de aceasta (ca urmare  $\rho_i$  este în domeniul  $[0,1]$ )
- $F_{ji}$  este un factor de formă al patch-ului  $j$  relativ la patch-ul  $i$ , și reprezintă fracțiunea din energia care părăsește patch-ul  $j$  și ajunge la patch-ul  $i$  fără interacțiune cu alte patch-uri ( $F_{ji}$  este în domeniul  $[0,1]$ )

În Ecuația 6-1 observăm că termenul din partea stângă este totalul energiei în unitatea de timp (deci puterea) care părăsește patch-ul  $i$ . Aceasta este egală cu suma a doi termeni, primul reprezentând puterea totală emisă (dacă există), iar al doilea – puterea totală reflectată. Acest ultim termen este ușor de înțeles, întrucât  $B_j A_j$  este puterea emisă de patch-ul  $j$  și din aceasta numai fracțiunea  $F_{ji}$  ajunge direct la patch-ul  $i$ . Ca urmare suma tuturor expresiilor de acest gen ne dă puterea totală care ajunge la patch-ul  $i$  de la toate celelalte patch-uri și numai fracțiunea  $\rho_i$  este reflectată (restul este absorbită).

Trebuie remarcat faptul că toate patch-urile din scenă sunt tratate ca surse de lumină, cu valori pentru  $E_i$  zero sau pozitive. Ca urmare, acum pot fi reprezentate imagini care conțin surse de lumină de formă oarecare!

## 6.2. Calculul factorului de formă

Pentru a rezolva Ecuația 6-1 în  $B_i$  trebuie mai întâi să calculăm factorul de formă din cadrul fiecărei perechi de patchuri din scenă.

După cum s-a menționat deja, factorul de formă  $F_{ji}$  este egal cu fracțiunea de energie care părăsește patch-ul  $j$  și ajunge la patch-ul  $i$  fără să interacționeze cu oricare alt patch din mediu. Factorul de formă încearcă să ia în calcul lumina care pleacă de la patch-ul  $j$  și ajunge la patch-ul  $i$  după ce a fost reflectată (de exemplu) de patch-ul  $k$  – aceasta intră în calculul lui  $B_k$  și  $F_{ki}$  în ecuația noastră.

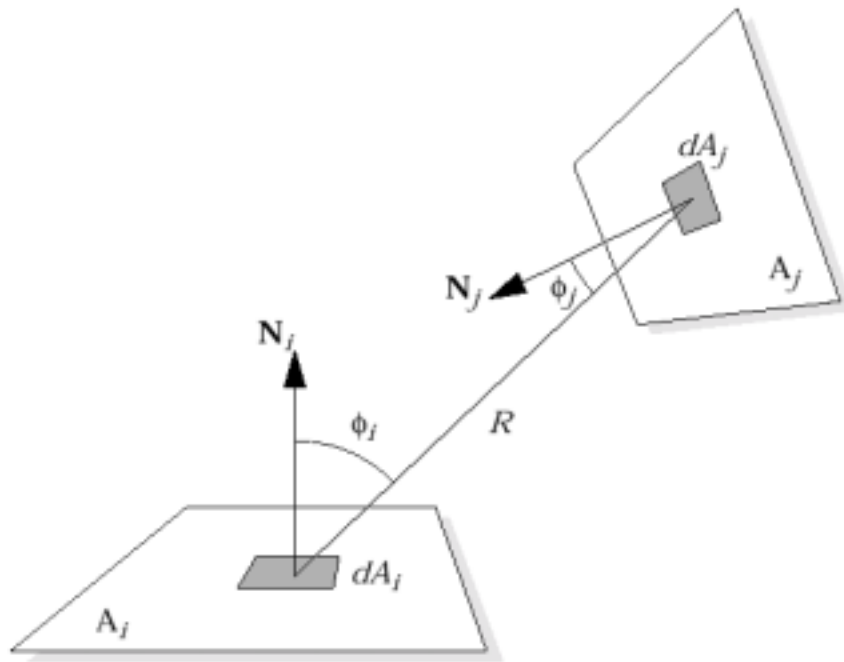


Figura 54 Geometria factorului de formă

În Figura 54 sunt prezentate două patch-uri  $i$  și  $j$ , descompuse într-un număr de regiuni elementare  $dA_i$  și  $dA_j$  (pentru claritate a fost reprezentată numai câte una). Regiunile elementare sunt la distanța  $R$  iar normalele sunt înclinate față de dreapta care le unește la unghiurile  $\phi_i$  respectiv  $\phi_j$ . Se poate demonstra că factorul de formă dintre cele două regiuni elementare este dat de

$$F_{dA_j dA_i} = \frac{\cos \phi_i \cos \phi_j}{\pi R^2} dA_i \quad (6-2)$$

Cu presupunerea pe care am făcut-o, că radiozitatea este constantă pentru întregul patch, factorul de formă pentru patch-urile întregi se calculează simplu din Ecuația (6-2) prin mediere pe toată suprafața,

$$F_{ji} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos \phi_i \cos \phi_j}{\pi R^2} dA_i dA_j \quad (6-3)$$

Desigur, limitarea cauzată de această presupunere poate fi depășită prin descompunerea patch-urilor care au gradienti de radiozitate ne-nuli pe suprafață, în patch-uri mai mici pentru care acest gradient este neglijabil.

Ecuția 6-3 presupune că fiecare porțiune a patch-ului  $i$  poate fi văzută dinspre fiecare porțiune a patch-ului  $j$ , ceea ce nu se întâmplă întotdeauna. Pentru a lua în calcul această acoperire, s-a introdus o funcție  $H_{ji}$  care are valoarea 1 dacă dreapta dintre  $dA_i$  și  $dA_j$  nu traversează vreun alt patch și 0 altfel. Ca urmare, Ecuția 6-3 devine

$$F_{ji} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos \phi_i \cos \phi_j}{\pi R^2} H_{ji} dA_i dA_j \quad (6-4)$$

### 6.2.1. Relația de reciprocitate

Între factorul de formă  $F_{ji}$  și reciprocul său  $F_{ij}$  (factorul de formă al patch-ului  $i$  relativ la patch-ul  $j$ ), există o relație care arată că

$$A_i F_{ij} = A_j F_{ji} \quad (6-5)$$

Aceasta este denumită *relația de reciprocitate pentru factori de formă*. Această relație este foarte utilă la implementarea unui algoritm care evaluează factorii de formă pentru o scenă, întrucât este evident că este necesar să calculăm factorii de formă  $F_{ij}$  pentru  $j > i$ . Cei  $n$  factori de formă  $F_{ii}$  vor fi nuli, iar restul  $n(n-1)/2$  factori pot fi calculați cu Ecuția 6-5.

O altă relație utilă care se poate aplica la calculul factorilor de formă este

$$\sum_{j=1}^n F_{ij} = 1 \quad (6-6)$$

care derivă din definiția factorului de formă ca fiind fracțiunea din energie care părăsește patch-ul  $i$  și ajunge direct la patch-ul  $j$ . Întrucât presupunem că mediul este închis, Ecuția 6-6 este o altă formulare a principiului conservării energiei. De obicei, Ecuția 6-6 este utilizată pentru a verifica acuratețea metodei care calculează factorii de formă.

### 6.3. Rezolvarea ecuației radiozității

Odată stabiliți factorii de formă între toate perechile de patch-uri din scenă, acum este posibil să rezolvăm Ecuția 6-1 pentru valorile radiozității  $B_r$ . Lucrurile se simplifică dacă aplicăm relația de reciprocitate din Ecuția 6-5 în Ecuția 6-1. Aceasta ne permite să scriem

$$B_i = E_i + \rho_i \sum_{j=1}^n B_j F_{ij} \quad (6-7)$$

Pentru a înțelege cum să rezolvăm aceste ecuații, să considerăm o situație simplă în care avem trei patch-uri:  $i=1,2$  și 3. Aplicând Ecuția 6-7, putem obține expresiile radiozității fiecărei suprafețe

$$\begin{aligned} B_1 &= E_1 + \rho_1 (F_{11} B_1 + F_{12} B_2 + F_{13} B_3) \\ B_2 &= E_2 + \rho_2 (F_{21} B_1 + F_{22} B_2 + F_{23} B_3) \\ B_3 &= E_3 + \rho_3 (F_{31} B_1 + F_{32} B_2 + F_{33} B_3) \end{aligned} \quad (6-8)$$

Avem deci un sistem de trei ecuații cu trei necunoscute,  $B_1$ ,  $B_2$  și  $B_3$ . Din geometria scenei putem calcula toți factorii de formă, și cunoaștem  $E$  pentru fiecare suprafață – dacă este o sursă de lumină îi cunoaștem intensitatea, iar dacă nu,  $E=0$ .

Putem rearanja ecuațiile într-o matrice de forma

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & -\rho_1 F_{13} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & -\rho_2 F_{23} \\ -\rho_3 F_{31} & -\rho_3 F_{32} & 1 - \rho_3 F_{33} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} \quad (6-9)$$

și să rezolvăm acest sistem cu necunoscutele  $B_i$  printr-o tehnică standard cum ar fi eliminarea Gaussiană.

Atenție, ecuațiile de forma celor din Ecuația 6-9 trebuie stabilite și rezolvate pentru fiecare culoare primară, de obicei roși, verde și albastru.

#### 6.4. Algoritmul hemi-cubului

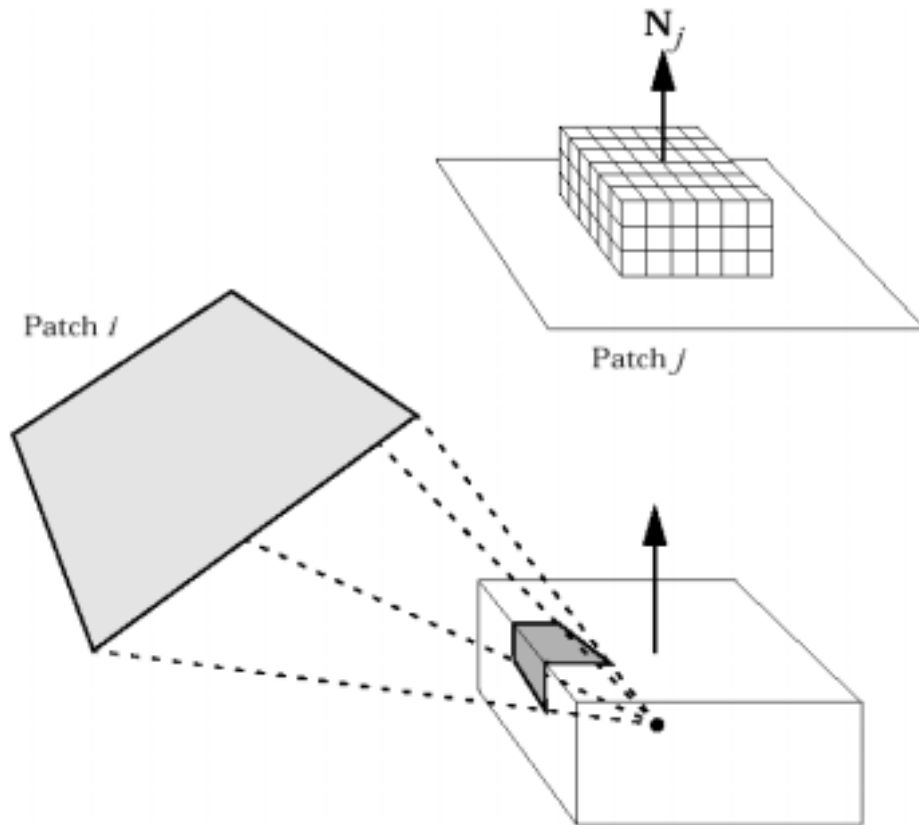
Metoda hemi-cubului este una dintre primele tehnici numerice pentru evaluarea factorilor de formă în medii complexe cu mai multe obiecte. Acest algoritm este o aproximare a integralei din Ecuația 6-4, în care se pornește de la ipoteza că distanța care separă patch-urile este mare în comparație cu dimensiunea lor. Acest lucru este de obicei adevărat, astfel încât integrala cea mai din interior din Ecuația 6-4 este aproximativ constantă, indiferent de poziția zonei elementare  $dA_j$  pe patch-ul  $j$ . Astfel putem scrie

$$F_{ji} = \left[ \int_{A_i} \frac{\cos \phi_i \cos \phi_j}{\pi R^2} H_{ji} dA_i \right] \frac{1}{A_j} \int_{A_j} dA_j = \int_{A_i} \frac{\cos \phi_i \cos \phi_j}{\pi R^2} H_{ji} dA_i \quad (6-10)$$

care reprezintă chiar factorul de formă dintre zona elementară  $dA_j$  și suprafața finită  $A_i$ .

Pentru evaluarea integralei din Ecuația 6-10 se concentrează un hemi-cub imaginar peste patch-ul  $j$ , adică un cub cu muchia de lungime 2 care are două fațete paralele cu patch-ul. Numai jumătatea cubului situată „în față” patch-ului este relevantă, de unde și termenul de *hemi-cub*. Situația este ilustrată în Figura 55.

Fațetele hemi-cubului se divid în pixeli (să nu se confunde cu pixelii de pe un ecran grafic) de aceeași mărime. Numărul și dimensiunea acestor pixeli sunt dependente de implemțare – densitatea mare oferă acuratețe ridicată a valorii factorului de formă. Rezoluțiile tipice pentru fațeta de deasupra variază între 50×50 la mai multe sute.



**Figura 55** Proiecția unui patch pe un hemi-cub

Următorul pas al algoritmului este proiectarea fiecărui patch din scenă pe hemi-cub și determinarea pixelilor acoperiți de proiecție. Dacă două patch-uri se proiectează peste același pixel atunci trebuie comparată adâncimea lor și se păstrează cel mai apropiat, întrucât celălalt nu poate fi văzut dinspre patch-ul  $j$ . Această operație reprezintă termenul  $H_{ji}$  din Ecuația 6-10. Fațetele hemi-cubului funcționează ca un Z-buffer, cu diferența că nu se păstrează informația de intensitate, doar identitatea patch-ului în cauză.

Factorul de formă asociat cu fiecare pixel al hemi-cubului, cunoscut ca *factor de formă delta*, este pre-calculat și accesat într-o tabelă de căutare la execuție. Odată determinați factorii de formă delta, putem calcula factorul de formă pentru un patch particular simplu, prin însumarea factorilor delta pentru toți pixelii pe care se proiectează patch-ul, adică

$$F_{ji} = \sum_q \Delta F_q$$

unde  $\Delta F_q$  este factorul de formă delta al pixelului  $q$ .

Odată calculați cei  $n$  factori de formă, hemi-cubul imaginar poate fi centrat peste alt patch din scenă și procesul se repetă până la determinare tuturor factorilor de formă.

#### 6.4.1. Calculul factorilor delta

Revenind la Ecuația 6-2, se observă că factorul de formă delta pentru un pixel  $q$  al hemi-cubului este dat de

$$\Delta F_q = \frac{\cos \phi_i \cos \phi_q}{\pi r^2} \Delta A \quad (6-11)$$

unde

- $\phi_i$  este unghiul dintre normala la patch-ul  $i$  și linia care unește centrul patch-ului cu pixelul  $q$
- $\phi_q$  este unghiul dintre normala la fațeta care conține pixelul  $q$  și linia care unește centrul patch-ului  $i$  cu pixelul  $q$
- $r$  este distanța dintre pixelul  $q$  până la centrul patch-ului  $i$
- $\Delta A$  este aria pixelului  $q$ .

Reamintim că în algoritmul hemi-cubului modelăm patch-ul  $i$  ca o suprafață elementară, astfel încât să putem utiliza rezultatul din Ecuația 6-10, și de asemenea că aproximăm un pixel al hemi-cubului ca o suprafață elementară. Astfel putem utiliza Ecuația 6-2.

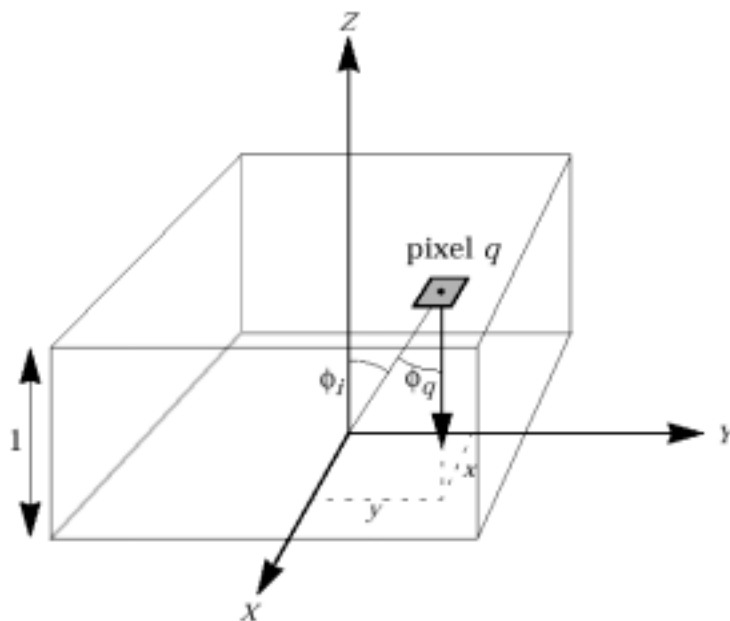
Pentru un pixel de pe fațeta superioară a hemi-cubului (vezi Figura 56)

$$\cos \phi_i = \cos \phi_q = \frac{1}{r}$$

unde

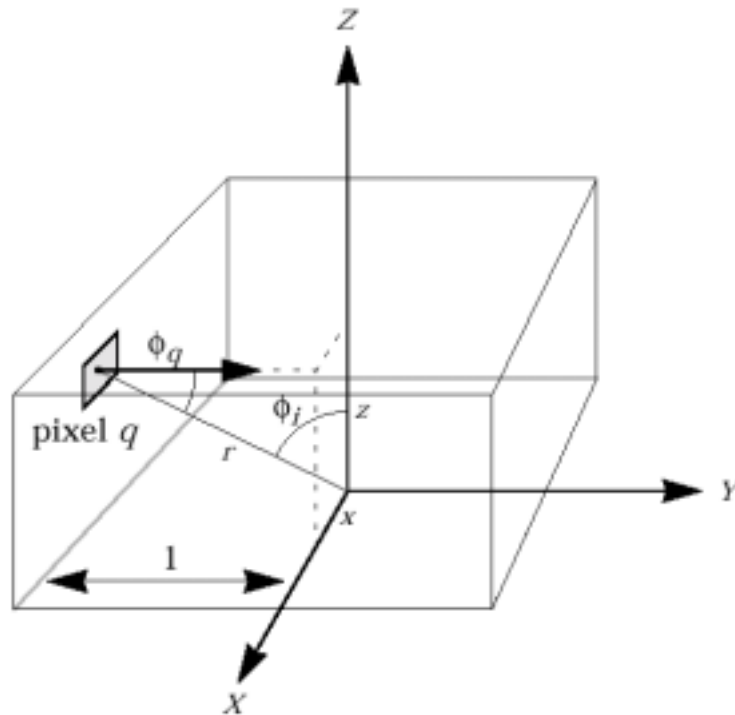
$$r = \sqrt{x^2 + y^2 + 1}$$

iar poziția pixelului  $q$  al hemi-cubului este dată de  $(x,y,z)$ , într-un sistem local de



**Figura 56** Factorul de formă delta pentru un pixel de pe fațeta superioară coordonate, centrat pe patch. Astfel din Ecuația 6-11 obținem





**Figura 57** Factorul de formă delta pentru un pixel de pe o fațetă laterală

Pentru o fațetă laterală (vezi Figura 57)

$$\cos \phi_i = \frac{z}{r} \text{ și } \cos \phi_q = \frac{l}{r}$$

unde

$$r = \sqrt{x^2 + z^2 + 1}$$

deci

$$\Delta F_q = \frac{z}{\pi [x^2 + z^2 + 1]^2} \Delta A$$

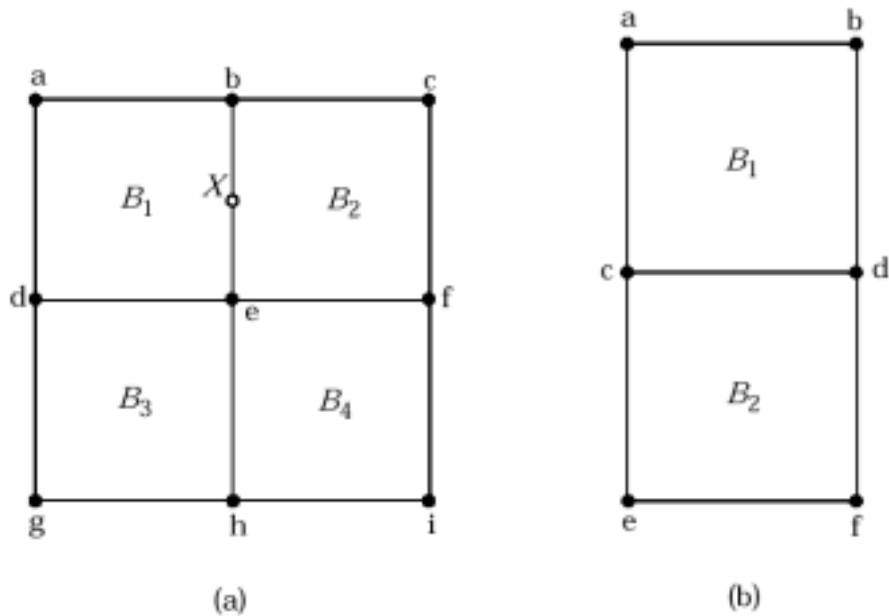
Pentru o fațetă laterală perpendiculară pe axa X, îl substituim pe  $x$  cu  $y$ .

Datorită simetriei hemi-cubului, nu trebuie memorați decât 1/8 dintre factorii delta asociați fațetei superioare și 1/4 dintre cei de pe o fațetă laterală.

## 6.5. Reprezentarea

Până aici, am aplicat algoritmul hemi-cubului fiecărui patch din scenă și am determinat toate intrările din matricea factorilor de formă. În acest moment putem rezolva Ecuația 6-7 pentru fiecare culoare primară și putem genera o vedere a scenei prin conversia valorii lui  $B_i$  în valori (R,G,B) și utilizarea unui algoritm convențional de îndepărtare a suprafețelor ascunse.

Totuși, rezultatul va fi o imagine compusă din fațete de colorate diferit – o situație nesatisfăcătoare care poate fi îmbunătățită dacă extrapolăm radiozitățile patch-urilor la vârfurile lor și apoi generăm imaginea utilizând interpolarea Gouraud pe suprafața fiecărui patch.



**Figura 58** Extrapolarea radioizității patch-urilor în vârfuri

Figura 58 prezintă două situații diferite care pot să apară atunci când încercăm să extrapolăm radioizțiile patch-urilor la vârfuri. În Figura 58a este prezentat cazul cel mai frecvent. Dacă un vertex este în interiorul suprafeței, de exemplu  $e$ , radioizitatea atribuită acestuia este media radioizțiilor patch-urilor de care aparține,

$$B_e = \frac{B_1 + B_2 + B_3 + B_4}{4}$$

Dacă este pe margine, de exemplu  $b$ , determinăm vertexul interior cel mai apropiat (din nou vertex-ul  $e$ ), și observăm că punctul median  $X$  dintre  $b$  și  $e$  se poate exprima radioizitatea în două modalități, astfel încât

$$\frac{B_b + B_e}{2} = \frac{B_1 + B_2}{2}$$

obținând

$$B_b = B_1 + B_2 - B_e$$

Pentru stabilirea radioizității unui vertex cum este  $a$ , din nou căutăm cel mai apropiat vertex interior,  $e$ , și observăm că media lui  $B_a$  și  $B_e$  trebuie să fie  $B_1$ , deci

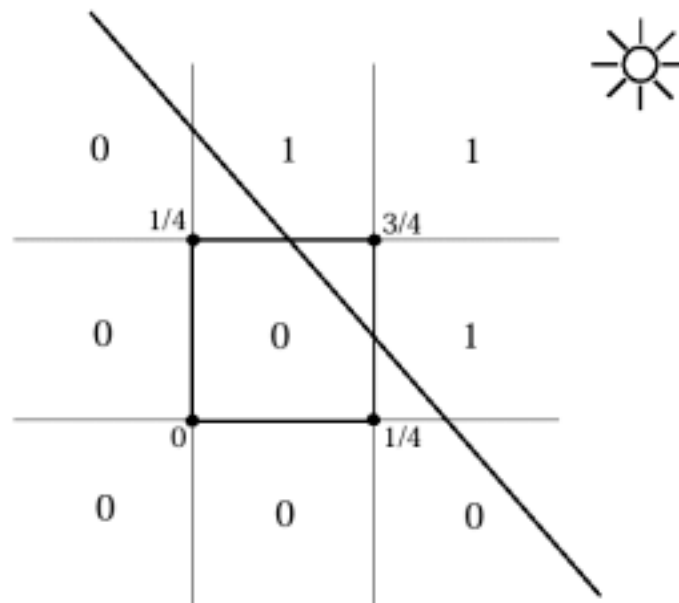
$$B_a = 2B_1 - B_e$$

În cazul ilustrat în Figura 58b nu poate fi aplicat nici unul dintre rezultatele de mai sus întrucât nu există puncte interioare. Cu toate că este posibil întotdeauna să facem extrapolarea și în acest caz, nu este chiar atât de simplu de recunoscut multitudinea de scenarii atât de diferite care pot să apară și se pare că nu există o formulă generală care să acopere toate cazurile.

### 6.5.1. Scurgeri de lumină și umbră

O problemă este des întâlnită în imaginile generate prin metoda radioizțiilor și rezultă atunci când două suprafețe se întâlnesc dar rețelele lor de elemente nu coincid. Pentru a

explica problema, să studiem diagrama schematică din Figura 59, care prezintă 9 elemente ale unei suprafețe și o altă suprafață, plasată la un unghi oarecare. De remarcat faptul că lumina cade pe suprafețe din dreapta, sus.



**Figura 59** Scurgeri de lumină și umbră

Fiecare element va recepționa un anumit nivel de iluminare de la sursa de lumină și suprafețele înconjurătoare, calculat utilizând un hemi-cub plasat în centrul elementului. Numerele plasate în centrele elementelor indică dacă ele recepționează radiozitate direct de la sursele de lumină. Atunci când generăm imaginea scenei trebuie să extrapolăm radiozitățile elementelor la vârfurile elementelor; elementul din mijloc va primi radiozități proporționale cu numerele prezentate. Ca urmare, acest element va fi reprezentat ca fiind iluminat gradat, de la stânga jos la dreapta sus, chiar dacă majoritatea suprafeței sale ar trebui să fie întunecată. Rezultatul vizual constă în faptul că lumina pare să se *scurgă* pe porțiuni ale suprafeței pe care fizic nu poate să ajungă. De asemenea, reciproca poate să apară, în cazul scurgerilor de *umbre*, în direcția opusă, într-o regiune în care nu ar trebui să apară umbre.

De obicei apare numai un singur gen de scurgere, în funcție de contrastul radiozităților vârfurilor elementelor. În exemplul discutat mai sus va predomina scurgerea de lumină, dând impresia că obiectul a cărui suprafață determină umbrirea pare să plutească deasupra celeilalte suprafețe. Vom prezenta exemple reale ale acestui efect ceva mai târziu. De asemenea vom vedea cum se poate reduce acest efect prin rafinarea adaptivă a patch-urilor.

## 6.6. Radiozitate cu rafinare progresivă

Metoda de bază descrisă până acum generează rezultate foarte realiste pentru medii cu reflectivitate difuză, dar poate necesita atât un timp îndelungat pentru calculul imaginii, cât și spațiu mare de memorie. Înainte ca o imagine să poată fi generată este necesar mai întâi să calculăm și să memorăm valorile a  $n \times n$  factori de formă. Aceasta necesită un timp de ordinul  $n^2$ . O altă durată mare de timp este necesară calculului radiozităților fiecărui patch, pentru fiecare culoare primară.

Metoda *rafinării progresive* încearcă să elimine aceste probleme printr-o modificare subtilă a algoritmului existent. Noua tehnică oferă costuri de memorie de ordinul  $n$ , și asigură o imagine inițială a scenei disponibilă după un timp de ordinul  $n$ .

Scopul radiozității cu rafinare progresivă este să combine realismul imaginii cu interacțiunea cu utilizatorul. Aceasta se realizează gășind rapid o aproximare a Ecuației 6-1 și apoi generarea imaginii. Această aproximare este apoi îmbunătățită iterativ și imaginea este actualizată de fiecare dată când se obține o îmbunătățire. Acest proces poate fi oprit de către utilizator atunci când se consideră că imaginea a atins o calitate acceptabilă.

### 6.6.1. Formularea metodei

Reformularea algoritmului radiozității este următoarea: Fie un singur termen al sumei din Ecuația 6-1 – acesta arată că porțiunea lui  $B_i$  datorată lui  $B_j$  este

$$B_i(\text{datorat lui } B_j) = \rho_i B_j F_{ji} \frac{A_j}{A_i} \quad (6-12)$$

Ca urmare, plasând un singur hemi-cub pe patchul  $j$ , putem calcula toți cei  $n$  factori de formă  $F_{ji}$ . Totuși, întrucât Ecuația 6-12 trebuie aplicată într-un proces *iterativ*, ar fi mai bine să calculăm *creșterea* lui  $B_i$  de la ultima contribuție a lui  $B_j$ . Putem face asta scriind

$$B_i(\text{datorat lui } \Delta B_j) = \rho_i \Delta B_j F_{ji} \frac{A_j}{A_i} \quad (6-13)$$

unde  $\Delta B_j$  este diferența dintre estimarea curentă a lui  $B_j$  și valoarea care a fost utilizată ultima dată când s-a calculat iluminarea provenind de la patch-ul  $j$ . Termenul  $\Delta B_j$  reprezintă efectiv radiozitatea care încă nu a fost luată în calcul dinspre patch-ul  $j$ .

Acum, având numai  $n$  factori de formă (prin plasarea *unui* hemi-cub pe patch-ul  $j$ ) este posibil să determinăm cât vor crește radiozitățile tuturor patch-urilor  $i$  datorită contribuției lui  $B_j$ . Acest proces este descris ca emiterea luminii dinspre patch-ul  $j$  către restul scenei. În particular, dacă patch-ul  $j$  este ales astfel încât să fie patch-ul cu cea mai mare contribuție ( $\Delta B_j A_j$  este cel mai mare) atunci o imagine poate fi generată după numai un singur calcul de hemi-cub). Dacă patch-ul este selectat aleator, atunci în general scena este încă întunecată (cu excepția surselor de lumină) după câteva astfel de calcule. Desigur, viteza de convergență a soluției depinde semnificativ de ordinea în care sunt selectate patch-urile.

După o iterație imaginea generată nu va fi ‘corectă’ întrucât valorile utilizate pentru  $\Delta B_j$  sunt toate *subestimate*. Algoritmul continuă iterațiile până când creșterea radiozității patch-urilor este considerată neglijabilă. De remarcat, totuși, că imaginea poate fi considerată corespunzătoare după o singură iterație dacă se utilizează o sortare corectă.

### 6.6.2. Contribuția ambientului

În general, primii pași ai procedurii rafinării progresive produc imagini de calitate scăzută, în particular în acele regiuni ale scenei care nu recepționează lumina direct. Pentru a compensa această deficiență (datorată faptului că algoritmul nu calculează radiozitățile *simultan*), se utilizează un termen de *ambient*, numai în ideea afișării. Termenul *ambien* se bazează pe media reflectivității mediului și de asemenea pe estimarea curentă a radiozității

tuturor patch-urilor. Aceasta înseamnă că acest termen se reduce treptat, pe măsura convergenței soluției reale.

Pentru a calcula contribuția ambientului, mai întâi facem presupunerea că pute determina o valoare rezonabilă a factorului de formă de la *oricare* patch  $j$  către un patch *dat*  $i$ , din

$$F_{ji}^{est} = \frac{A_i}{\sum_{k=1}^n A_k}$$

care este fracțiunea pe care o reprezintă suprafața patch-ului  $i$  din totalul suprafețelor din scenă.

Reflectivitatea medie a scenei este calculată ca o medie ponderată a reflectivității patch-urilor:

$$\rho_{mediu} = \frac{\sum_{j=1}^n \rho_j A_j}{\sum_{j=1}^n A_j} \quad (6-14)$$

Dându-se o energie emisă în scenă, o fracțiune din ea,  $\rho_{mediu}$ , va fi, în medie, reflectată înapoi în scenă, din care o fracțiune  $\rho_{mediu}$  va fi, în medie reflectată înapoi, și așa mai departe. Ca urmare coeficientul general de inter-reflexie pentru acea scenă este dat de

$$1 + \rho_{mediu} + \rho_{mediu}^2 + \rho_{mediu}^3 + \dots \quad (6-15)$$

Întrucât  $0 \leq \rho_j \leq 1$ , atunci din Ecuația 6-14 avem aceeași inegalitate și pentru  $\rho_{mediu}$  și deci Ecuația 6-15 poate fi scrisă

$$1 + \rho_{mediu} + \rho_{mediu}^2 + \rho_{mediu}^3 + \dots = \frac{1}{1 - \rho_{mediu}}$$

Termenul de radiozitate ambientă se calculează ca produsul dintre coeficientul de inter-reflexie cu media ponderată a tuturor radiozităților și a factorilor de formă:

$$B_{amb} = \frac{1}{1 - \rho_{mediu}} \sum_{i=1}^n \Delta B_i F_{ji}^{est}$$

Pentru vizualizarea realistă a patch-ului  $k$ , mai trebuie să adunăm  $\rho_k B_{amb}$  la valoarea returnată de algoritm pentru  $B_k$ . Această valoare revizuită nu are nici un rol în calculele următoare și se include numai în vederea afișării. Subliniem faptul că pe măsură ce soluția converge,  $\Delta B_i \rightarrow 0$  și la fel  $B_{amb} \rightarrow 0$ , astfel încât termenul ambient dispare *treptat*.

## 6.7. Îmbunătățirea soluției radiozității

Deseori se întâmplă ca într-o scenă umbrele și alte regiuni în care radiozitatea se modifică rapid să nu fie reprezentate corespunzător. Aceasta deoarece densitatea de eșantionare pentru calculul radiozității într-un punct dat este tot atât de fină ca și densitatea patch-urilor: cu cât avem mai multe patch-uri, cu atât va fi mai mare numărul de puncte în care cunoaștem valoarea radiozității și cu atât va fi mai bună soluția.

O modalitate simplă de depășire a acestei probleme este să creștem numărul de patch-uri din scenă. Totuși, aceasta va crește numărul de factori de formă de calculat; de exemplu, dublând numărul de patch-uri, numărul de factori de formă va crește de patru ori. Mai important, timpul de calcul va crește.

### 6.7.1. Substructurare

O soluție mai bună este să utilizăm o tehnică denumită *substructurare*, după cum urmează. Mai întâi trebuie să definim o serie de termeni noi, care să facă distincția dintre un *patch* (cum a fost descris anterior) și un *element*, care este o fracțiune dintr-un patch, obținută printr-o subdivizare binară a patch-ului sau a unui alt element. Elementele sunt cele mai mici unități pentru care se calculează valoarea radiozității. De asemenea, elementele poartă indicele  $q$ , în vreme ce patch-urile au indici  $i$  și  $j$ .

Presupunem că scena a fost decompusă în  $n$  patch-uri, ca anterior, și că unele dintre aceste patch-uri sau toate au fost decompuse într-un număr de elemente, astfel încât avem un total de  $m$  elemente în scenă. Să examinăm această situație în termeni de radiozități ale elementelor. Din Ecuația 6-7 știm că radiozitatea unui element  $q$  poate fi scrisă ca

$$B_q = E_q + \rho_q \sum_{j=1}^n B_j F_{qj} \quad (6-16)$$

Presupunem că patch-ul  $i$  a fost divizat în  $t$  elemente. Atunci radiozitatea patch-ului  $i$  poate fi reprezentată ca media ponderată pe suprafață a radiozității celor  $t$  elemente

$$B_i = \frac{1}{A_i} \sum_{q=1}^t B_q A_q$$

care, înlocuită în Ecuația 6-16 ne dă

$$B_i = \frac{1}{A_i} \sum_{q=1}^t \left[ E_q + \rho_q \sum_{j=1}^n B_j F_{qj} \right] A_q \quad (6-17)$$

Deja am presupus că radiozitatea globală nu variază pe suprafața unui patch. Dacă presupunem de asemenea că nici  $\rho_q$  nici  $E_q$  nu variază pe suprafața unui patch, atunci Ecuația 6-17 poate fi simplificată

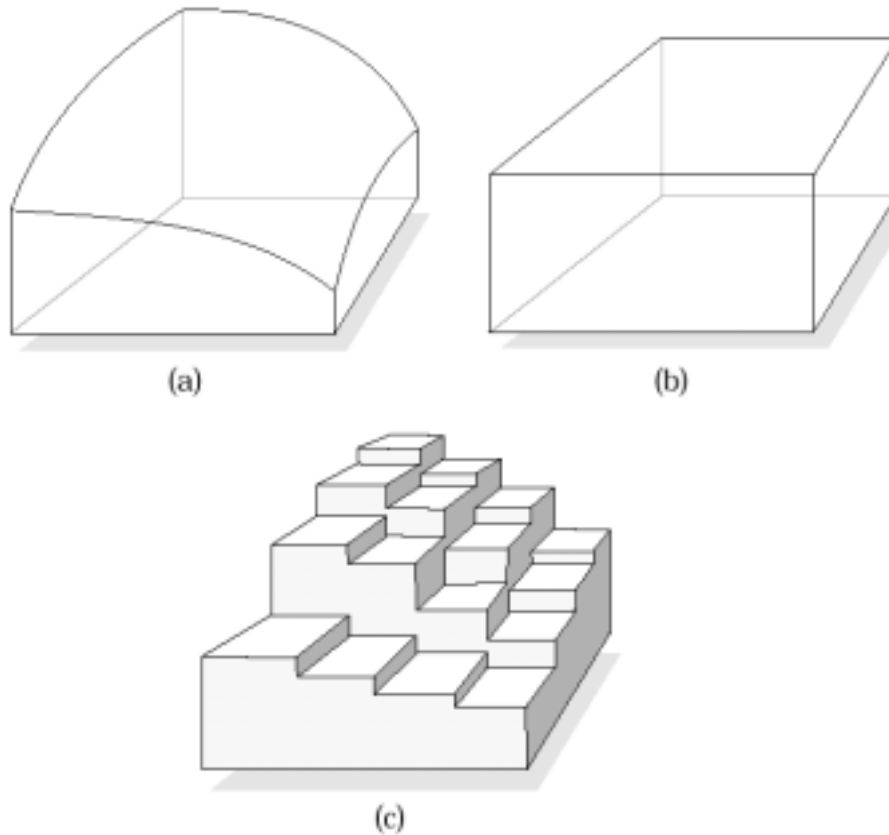
$$B_i = E_i + \rho_i \sum_{j=1}^n B_j \left[ \frac{1}{A_i} \sum_{q=1}^t F_{qj} A_q \right] \quad (6-18)$$

Comparând această ecuație cu Ecuația 6-7 observăm că

$$F_{ij} = \frac{1}{A_i} \sum_{q=1}^t F_{qj} A_q \quad (6-19)$$

care este o aproximare mai bună decât ipoteza că integrala interioară din Ecuația 6-4 este constantă pe întregul patch  $i$  (acum ipoteza este că integrala este constantă numai pe fiecare element). Datorită acestei ipoteze, factorii de formă pentru patch-uri vor avea o mai bună acuratețe.

Ca urmare, substructurarea ne permite să obținem valori mai bune pentru radiozitățile elementelor, utilizând valorile radiozităților deja calculate din soluția la nivel de patch-uri și utilizând Ecuația 6-16. Acest proces este ilustrat în Figura 60.



**Figura 60** Evidențierea gradientului radiozității

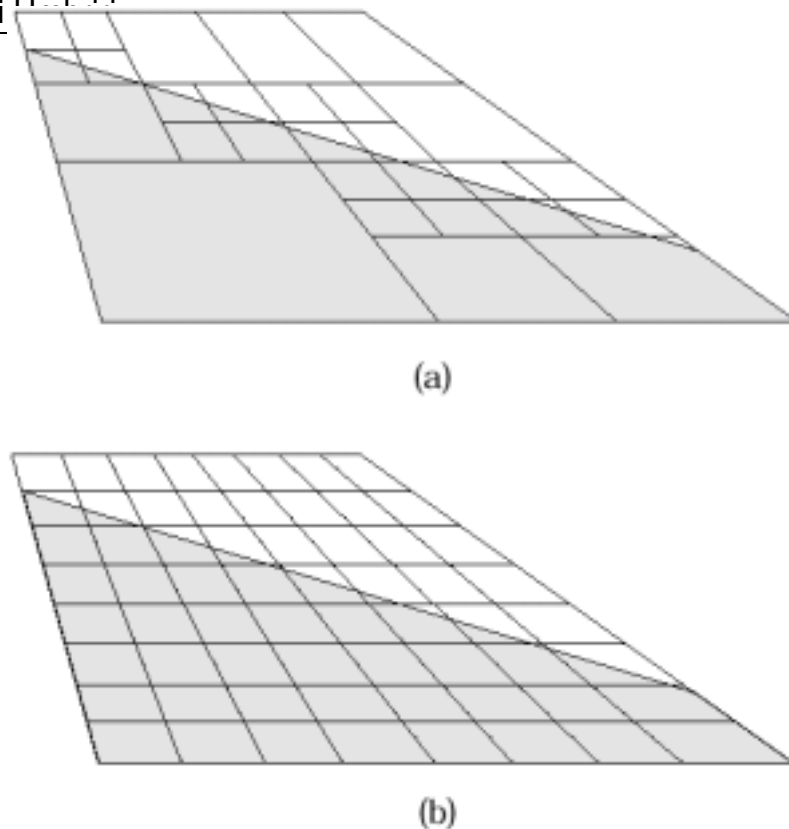
În Figura 60a este prezentată variația reală a radiozității peste un patch desenat ca o suprafață în 2D; Figura 60b este reprezentarea grosolană a patch-ului, care este evident o aproximare slabă. Totuși, la divizarea patch-ului în 16 elemente și calcularea radiozității fiecăruia, reprezentarea care rezultă este mult mai bună (Figura 60c).

### 6.7.2. Subdivizare adaptivă

Prin tehnica subdivizării adaptive, numai acele patch-uri (sau elemente) care prezintă un gradient al radiozității care depășește un nivel prestabilit sunt supuse subdivizării.

Pentru a afla care porțiuni din imagine trebuie subdivizate, este necesară o ‘primă încercare’, care se poate obține exact în modul descris în Secțiunea 6.3. Patch-urile și elementele pot fi acum examinate și verificate pentru gradienti prea mari ai radiozității, iar dacă este necesar se pot efectua subdivizări. Apoi se calculează o a doua soluție și procesul se repetă până când nici unul dintre gradientii radiozității nu mai este prea mare.

Este posibil ca imaginea rezultată să conțină mai puține elemente decât o imagine acceptabilă calculată pe baza substructurării simple, dar definiția umbrelor și umbrirea în general să fie net superioare, întrucât densitatea elementelor în punctele critice ale imaginii este mult mai mare.



**Figura 61** Subdivizare adaptivă

În Figura 61 sunt prezentate două situații, în care un patch, care are o porțiune umbrită, a fost subdivizat în încercarea de a reprezenta cu acuratețe gradientul radiozității. În ambele cazuri patch-ul a fost decompus (ca punct de pornire) în patru elemente. În Figura 61a elementele au fost descompuse utilizând subdivizarea adaptivă, iar în Figura 61b a fost atinsă aceeași rezoluție ca și în Figura 61a, de data aceasta utilizând substructurarea simplă. Se observă că în ambele cazuri rezultă aceeași imagine, dar este evident că structura prezentată în Figura 61a va necesita un consum mult mai scăzut de memorie și de timp de calcul.

Într-un mod similar putem utiliza această tehnică de subdivizare adaptivă în cadrul metodei radiozității progresive. Ca și în cazul algoritmului inițial, patch-urile se descompun în elemente, și aceste elemente se comportă ca receptori de lumină de la *patch-uri*, nu alte elemente. Astfel, se plasează un hemi-cub pe fiecare patch din scenă, iar elementele sunt proiectate pe suprafața hemi-cubului, pentru a determina cei  $m$  factori de formă patch-element,  $F_{iq}$ .

## 6.8. Exemple de imagini

Pentru a ilustra posibilitățile metodei radiozității, vom prezenta în această secțiune câteva exemple calculate utilizând tehnica rafinării progresive. În particular, vom demonstra superioritatea acestei tehnici în redarea unor aspecte naturale, cum ar fi umbrele difuze și amestecul de culori, pe care ray-tracing-ul le poate simula cu dificultate.

Primul exemplu este prezentat în Figura 62, în care avem 5 vederi ale unui mediu simplu închis. Scena este compusă dintr-o încăpere mare cu o sursă de lumină distribuită în centrul tavanului. Pereții încăperii sunt colorați în roșu, albastru, gri deschis și negru. În interior, pe podea, mai sunt două obiecte: o cutie mică galbenă și o cutie mai înaltă cu fațetele laterale de culoare gri, iar fațeta superioară este gri deschis.

Pereții, podeaua și tavanul au fost divizate uniform în câte 100 de elemente (52 de patch-uri), cutiile galbenă și gri – în 20 de elemente (5 patch-uri) iar sursa de lumină este



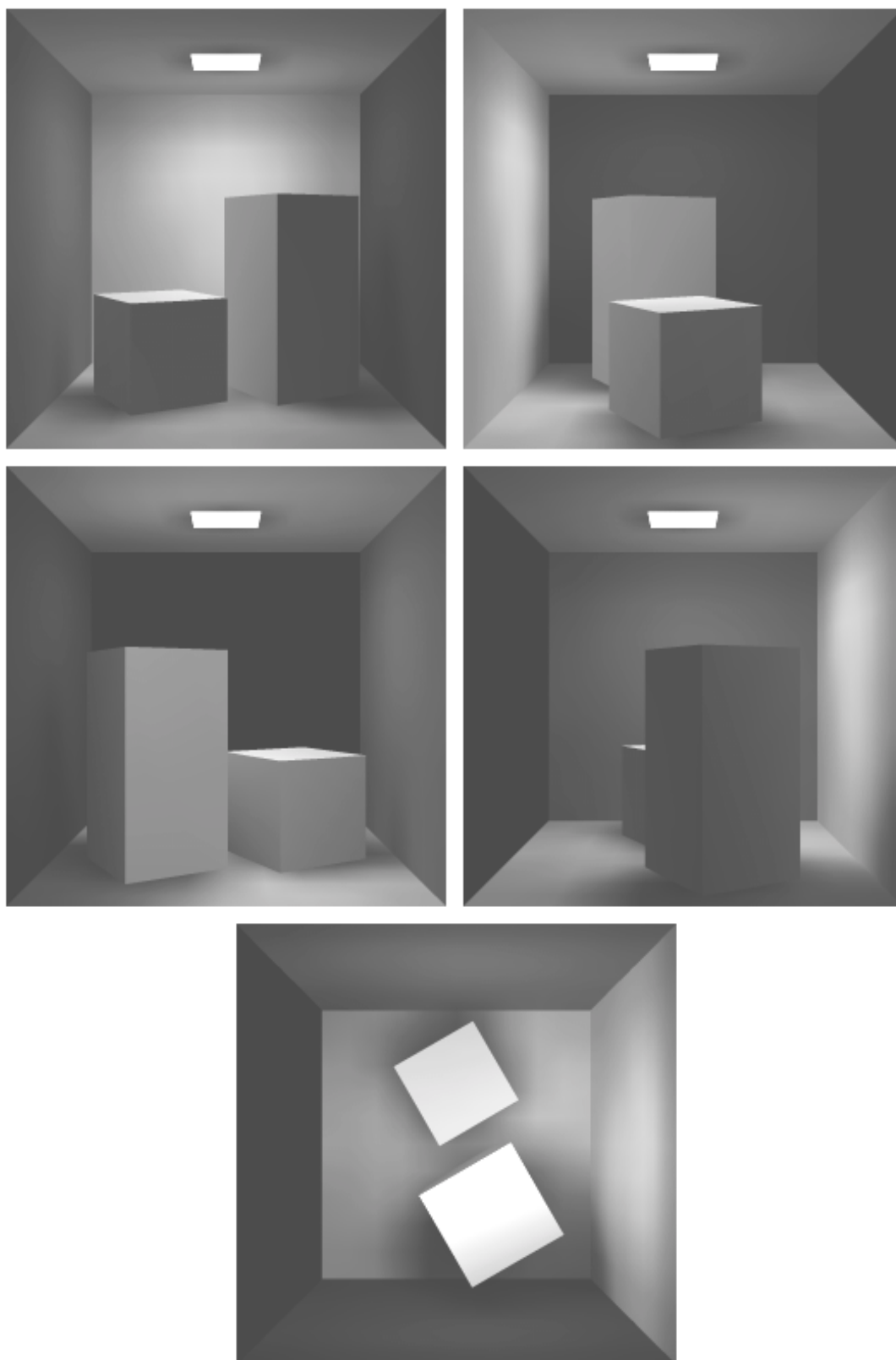
reprezentată printr-un element (1 patch). Calculul a necesitat pentru convergență 1506 iterații: soluția a început să converge atunci când energia totală absorbită de mediu a fost mai puțin de 0.1% din cea emisă de sursa de lumină.

Aceste imagini ilustrează clar modul în care metoda radiozității poate genera umbre realiste. O sursă de lumină distribuită, cum este cea utilizată în acest exemplu, produce întotdeauna umbre difuze, cu zone de umbră și penumbră. Totuși, trebuie remarcat cum cele două cutii par să plutească deasupra podelei. Aceasta se datorează unei combinații de “scurgeri” de lumină și umbră, după cum am discutat anterior. Se mai observă amestecul puternic de culori (*colour bleeding*) care apare de la o suprafață la alta, în particular efectul peretelui roșu pe lateralul cutiei galbene și al peretelui albastru pe cutia gri.

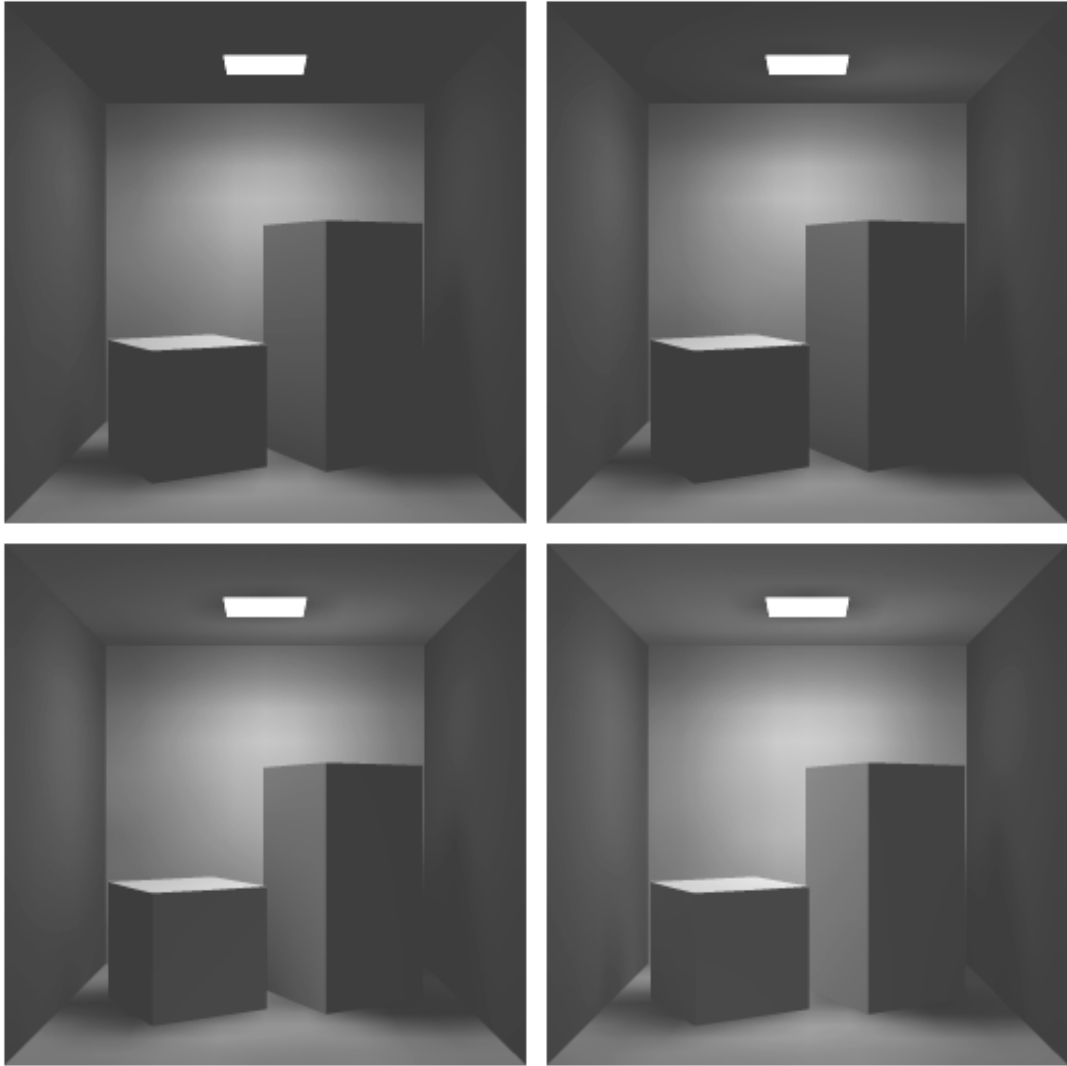
Figura 63 prezintă efectele tehnicii rafinării progresive asupra aceleiași scene descrisă anterior. Secvența de imagini reprezintă starea rafinării după 1, 2, 20 și 100 de iterații cu 100%, 45%, 34% și respectiv 21% din energie rămasă ne-emisă. Nu s-a utilizat o estimare a contribuției ambientului. După 1 iterație, numai energia de la sursa de lumină a fost distribuită la celelalte patch-uri și se observă că numai suprafețele iluminate direct de sursa de lumină au recepționat (și au emis) lumină. Restul scenei este întunecat. După 2 iterații, patch-ul cu cea mai mare cantitate de energie rămasă ne-emisă (un patch pe fațeta superioară a cutiei mai înalte) și-a emis energia către restul scenei, cel mai vizibil în porțiunea de tavan chiar deasupra, pe care o luminează corespunzător.

După fiecare iterație, cantitatea de energie ne-emisă scade, rezultând o iluminare graduală a scenei. Chiar și după 100 de iterații imaginea arată aproape la fel ca și soluția complet convergentă și am putea fi tentați să oprim aici calculele. Totuși, este important să continuăm, întrucât mai sunt necesare foarte multe iterații ca să devină vizibile detalii subtile în privința umbrelor și a inter-reflexiei luminii între diferitele suprafețe.

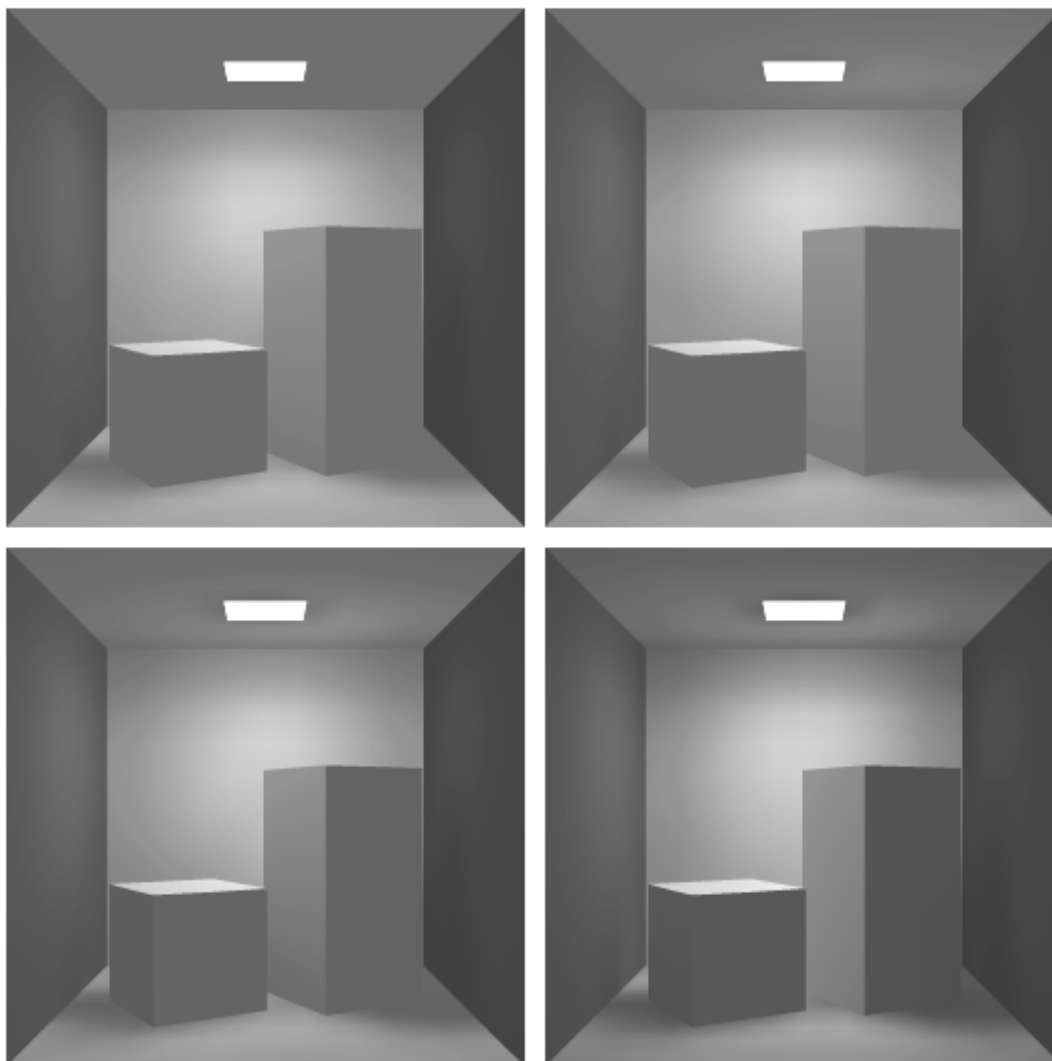
Secvența de imagini din Figura 64 utilizează rezultatele calculului cu rafinare progresivă prezentate în Figura 63, dar cu o contribuție a ambientului (estimat din totalul energiei ne-emisă).



**Figura 62** Cinci vederi ale unei scene calculată cu metoda radiozității



**Figura 63** Rafinare progresivă fără termenul ambient



**Figura 64** Rafinare progresivă cu termen ambient

În următoarea imagine vom ilustra tehnica rafinării adaptive pentru subdivizarea patch-urilor și vom prezenta utilizarea metodei pentru îmbunătățirea acurateții.

Figura 65 prezintă o altă scenă simplă, similară cu cea prezentată anterior, și rezultatele unei soluții complete. Se mai prezintă distribuția patch-urilor și a elementelor în care a fost divizată fiecare suprafață.

Se remarcă două regiuni ale scenei care au fost puternic subdivizate: zona în care tavanul se întâlnește cu pereții și în jurul muchiei cutiei și a umbrei sale. Regiunea din jurul umbrei a un contrast ridicat al radiozității, ceea ce a determinat rafinarea, rezultând o mai bună definiție a umbrei. Totuși, rafinarea adaptivă nu poate fi niciodată perfectă și se poate observa apariția unor defecte care fragmentează marginile umbrei. Se mai observă că rafinarea a eliminat majoritatea scurgerilor de umbră din jurul muchiei cutiei, astfel încât de data aceasta cutia pare bine plasată pe podea!

În final, în Figura 66 avem imaginea unei scene care este mult mai complexă decât în imaginile anterioare.

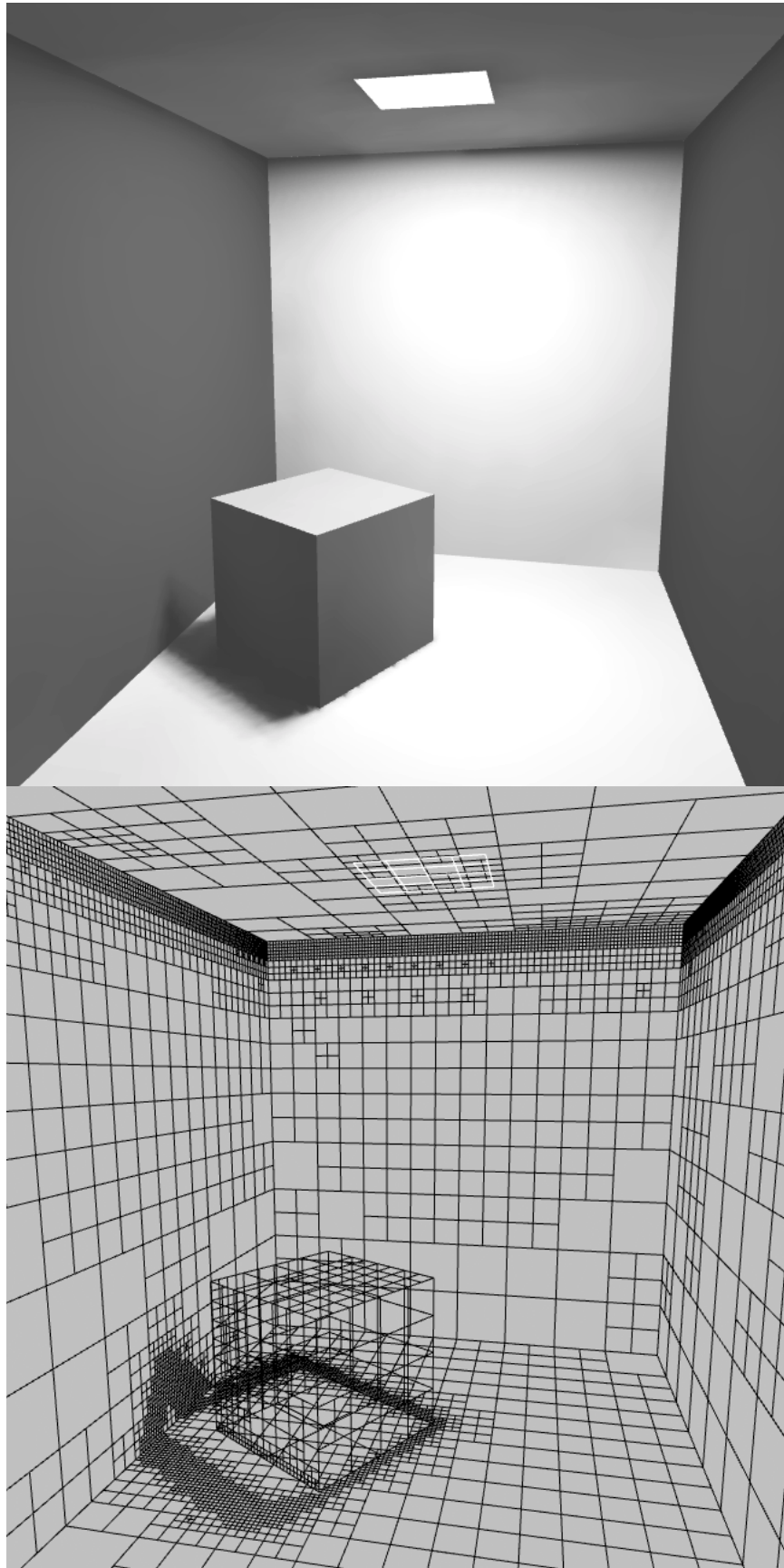


Figura 65 Rafinare adaptivă

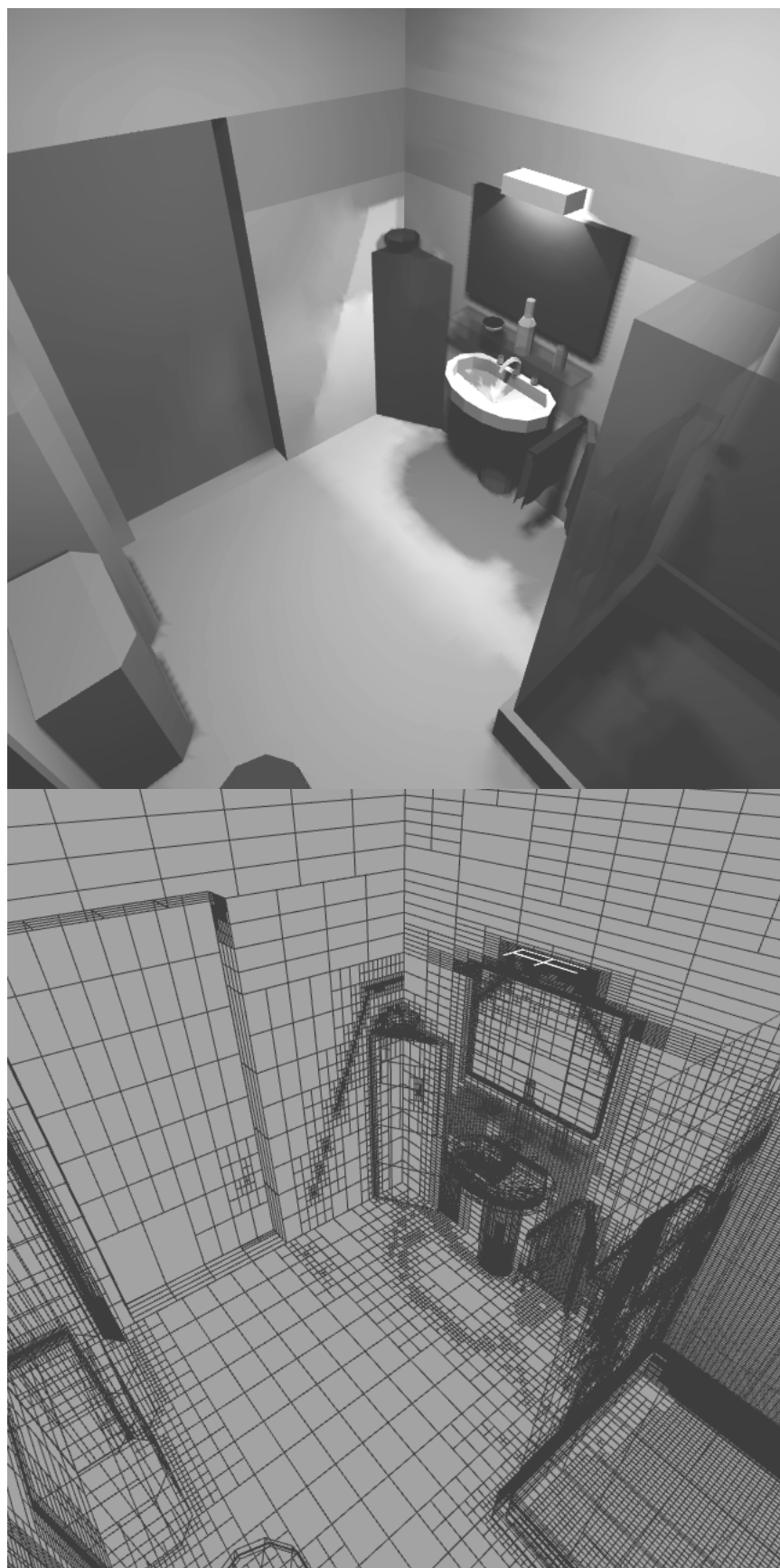


Figura 66 O scenă complexă cu rafinare adaptivă

## Software

Ray-tracing:

- Programul *Rayshade*:
  - <http://www-graphics.stanford.edu/~cek/rayshade/>
  - <ftp://graphics.stanford.edu/pub/rayshade/>
- Alte programe de ray-tracing:
  - <http://arachnid.cs.cf.ac.uk/Ray.Tracing/>

Radiozitate:

- *Helios* (disponibil numai împreună cu cartea). Un demo la el poate fi găsit la
  - <http://www.ledalite.com/library/rrt.html>
- RAD (disponibil momentan numai pe Silicon Graphics):
  - <http://www.cs.kuleuven.ac.be/~philippe/rad.html>