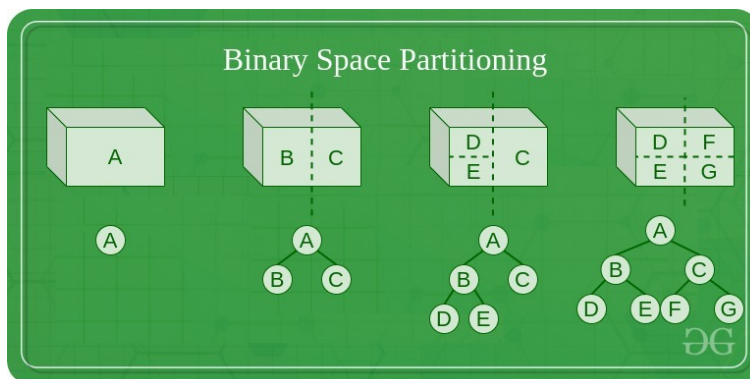


## CURS 10 -- Algoritmul BSP (Binary Space Partitioning).

Binary Space Partitioning este implementată pentru a subdiviza recursiv un spațiu în două mulțimi convexe, folosind hiperplanuri ca partiții. Acest proces de subdivizare dă naștere reprezentării obiectelor în spațiu sub forma unei structuri de date arborescente cunoscută sub numele de arborele BSP. Partiționarea spațiului binar a apărut în contextul graficii 3D pe computer în 1969, unde structura unui arbore BSP permite accesarea rapidă a informațiilor spațiale despre obiectele dintr-o scenă care sunt utile în randare, cum ar fi obiectele ordonate din față în spate în raport cu un vizualizator dintr-o anumită locație.



Partiționarea spațiului binar a apărut din nevoia de a desena rapid scene tridimensionale compuse din poligoane în grafica pe calculator. O modalitate simplă de a desena astfel de scene este algoritmul pictorului, care produce poligoane în ordinea distanței față de privitor, din spate în față, pictând pe fundal și poligoane anterioare cu fiecare obiect mai apropiat. Această abordare are două dezavantaje: timpul necesar pentru sortarea poligoanelor în ordinea din spate în față și posibilitatea de erori în poligoane suprapuse. Fuchs și coautorii au arătat că construirea unui arbore BSP a rezolvat ambele probleme, oferind o metodă rapidă de sortare a poligoanelor în raport cu un anumit punct de vedere (liniar în numărul de poligoane din scenă) și prin subdivizarea poligoanelor suprapuse pentru a evita erorile care pot apărea cu algoritmul pictorului.

Partiționarea spațiului binar este tratată ca un proces generic de împărțire recursiv a unei scene în două până când partiționarea satisface una sau mai multe cerințe. Poate fi privit ca o generalizare a altor structuri de arbore spațial, cum ar fi arborii k-d și arborii patru, una în care hiperplanele aceluși spațiu de partiție putând avea orice orientare în loc să fie aliniate cu axele de coordonate, așa cum sunt în arbori k-d sau în patru arbori. Când sunt implementate în grafica computerizată pentru a reda scene compuse din poligoane plane, planele de partiție

sunt frecvent selectate pentru a coincide cu planele definite de poligoane din scenă. Alegerea specifică a planului de partiționare și criteriul pentru finalizarea procesului de partiționare variază în funcție de scopul arborelui BSP.

#### **Algoritm de generare a unui arbore BSP dintr-o listă de poligoane**

- Selectați un poligon P din listă.
- Faceți un nod N în arborele BSP și adăugați P la lista de poligoane de la acel nod.
- Pentru fiecare alt poligon din listă:
  - Dacă acel poligon se află în întregime în fața planului care conține P, mutați acel poligon în lista de noduri din fața lui P.
  - Dacă acel poligon se află în întregime în spatele planului care conține P, mutați acel poligon în lista de noduri din spatele P.
  - Dacă acel poligon este intersectat de planul care conține P, împărțiți-l în două poligoane și mutați-le în listele respective de poligoane în spatele și în fața lui P.
- Dacă acel poligon se află în planul care conține P, adăugați-l la lista de poligoane de la nodul N.
- Aplicați acest algoritm la lista de poligoane din fața lui P.
- Aplicați acest algoritm listei de poligoane din spatele P.

#### **Dezavantajul BSP**

Generarea unui arbore BSP poate consuma mult timp.

BSP nu rezolvă problema determinării suprafeței vizibile.

#### **Utilizări ale BSP**

Este folosit în detectarea coliziunilor în jocurile video 3D și robotică.

Este folosit în ray tracing.

Este implicat în manipularea scenelor spațiale complexe.

Arborii BSP vor împărți obiectele astfel încât algoritmul pictorului să le deseneze corect fără a fi nevoie de un Z-buffer și să elimine necesitatea de a sorta obiectele:

o simplă traversare a arborelui va produce obiectele în ordinea corectă.

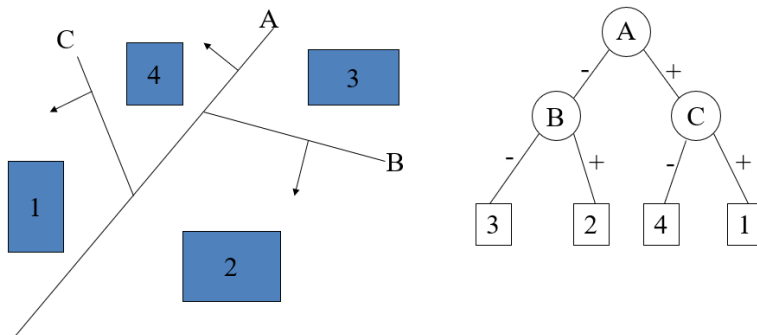
De asemenea, servește drept bază pentru alți algoritmi, cum ar fi listele de vizibilitate, care urmăresc să reducă *overdraw*.

Cu toate acestea, construirea arborelui pentru preprocesarea scenei necesită timp, ceea ce face dificilă și ineficientă implementarea directă a obiectelor în mișcare

într-un arbore BSP. Acest lucru este adesea depășit prin utilizarea arborelui BSP împreună cu un Z-buffer (folosind tamponul Z pentru a îmbina corect obiectele mobile pe scena de fundal).

## 1 Exemplu de aplicare a algoritmului BSP

În figura de mai jos este prezentat un exemplu de aplicare a arborelui BSP.



Pentru trasarea unui arbore BSP avem nevoie de pași următori:

Alegeți poligonul (arbitrar)

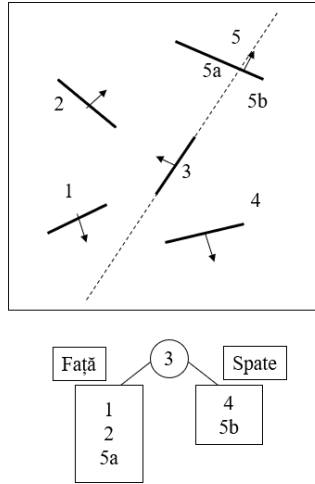
Împărțiți celula folosind planul în care se află poligonul.

Poate fi necesar să tăiați poligoane în două (decupare!)

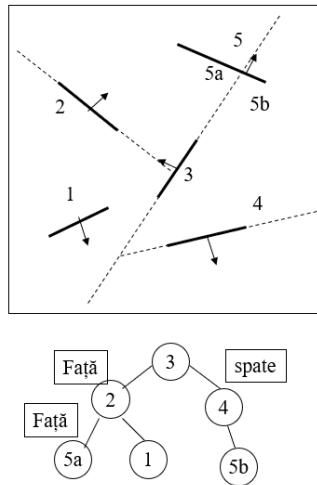
Continuați până când fiecare celulă conține doar un fragment de poligon.

Planele de divizare ar putea fi alese în alte moduri, dar nu există un algoritm optim eficient pentru construirea arborilor BSP.

Optimal înseamnă numărul minim de fragmente de poligon într-un arbore echilibrat.



În final, arborele BSP arată astfel:

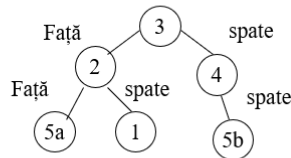


Determinați ordinea obiectelor - din spate în față.  
 Observatorul este în fața rădăcinii (3), așa că trasăm toate în spatele (3), apoi (3), apoi toate în fața nodului (3).  
 Când desenați în fața lui (3), vedem că observatorul este în spatele rădăcinii sub-arborelui (2).

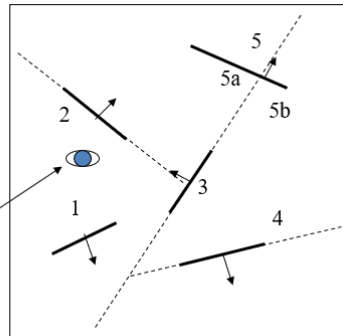
Așa că desenăm toate în fața (2), apoi (2), apoi în spatele lui (2)..... ordinea desenării este 4, 5b, 3, 5a, 2, 1.

În acest fel, obiectele parcurse mai târziu pot fi desenate peste obiectele anterioare cu rezultate corecte.

## Trasare arbore BSP



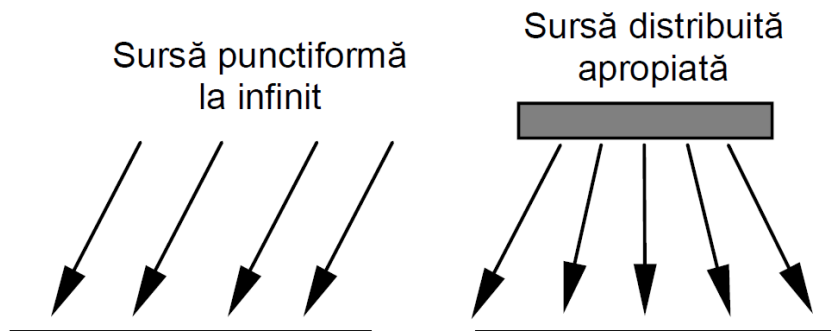
*Presupunem observatorul poziționat aici*



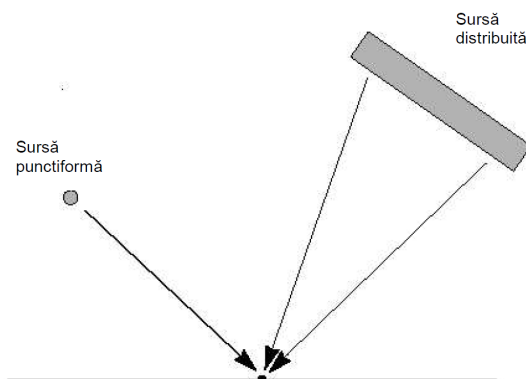
## 2 Modele de iluminare globală

Modelele de iluminare globală pot încorpora umbre, reflexii și refracții. Aceste modele sunt utilizate mai ales când urmărim redarea unui aspect al realității fizice și când realismul este mai important decât viteza de afișare. Din punct de vedere computațional, cu cât mai realist este un model cu atât este mai costisitor computațional.

Există două tipuri de bază de surse de lumină: emițători și reflectori. Mai putem clasifica emițătorii de lumină în două categorii. Atunci când sursa de lumină este suficient de îndepărtată de o suprafață, o putem considera sursa 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 de mai jos.



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).



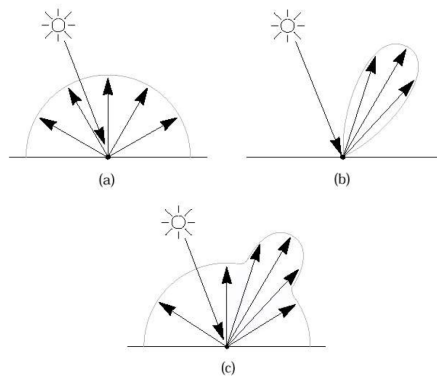
### 3 Reflectori de lumină

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. Deci 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.

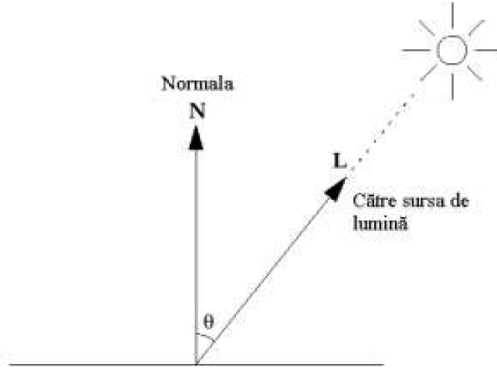
Culoarea unui reflector depinde de spectrul luminii incidente și de spectrul de reflectanță al obiectului. În plus, pe lângă aceste reflexii directe, există și reflexii indirecte reflexii la reflexii etc. Aceste reflexii multiple produc într-o scenă o

iluminare generală, numită lumină ambiantă, care va fi tratată mai târziu. Trebuie să dezvoltăm un model care să ne permită calcularea cantității de lumină reflectată de către diferitele obiecte dintr-o scenă. Reflexia se petrece în două moduri: Reflexie difuză lumina este împrăștiată uniform de suprafața obiectului, și suprafața are aspect mat. Suprafața absoarbe lumina incidentă și o reradiază 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 directionată (speculară) o proporție mare din lumina incidentă este reflectată într-un domeniu restrâns de unghiuri, iar obiectul arată lucios. 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 Figură.



Să considerăm o singură sursă de lumină, modelată sub forma unui punct, plasat la distanța 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. Vectorul normală  $N$  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  $L$  indică înspre sursa de lumină.



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  $N$  și  $L$  au fost normalizați. Unghiul  $\theta$  dintre  $N$  și  $L$  se numește unghi de incidență.

### 3.1 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 ( $\theta = 90$ ), 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:

$$I = k_d I_p \cos(\theta)$$

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

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

deci putem rescrie ecuația de mai sus sub formă vectorială:

$$I = k_d I_p (N \cdot L)$$



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

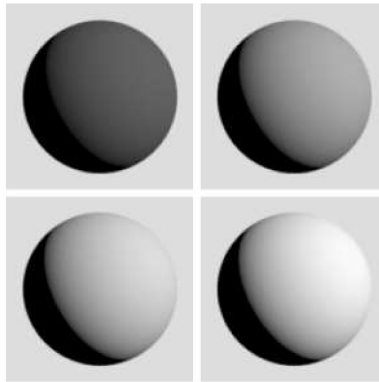
$I = \text{componenta ambiantă} + \text{componenta difuză}$

sau,

$$I = k_d I_a + k_d I_p (N \cdot L)$$

În figura de mai jos sunt prezentate patru sfere ale căror suprafețe au fost umbrite conform modelului de reflexie difuză dat de Ecuația de mai sus. De la stânga la dreapta și de sus în jos,

$$k_d = 0.1, 0.4, 0.7, 0.95.$$



## 4 BIBLIOGRAFIE

1. <https://www.geeksforgeeks.org/binary-space-partitioning>.