1. **Asemanarile si deosebirile dintre efectele proiectiilor perspectiva si cele ale proiectiilor axonometrice.**

Asemanari:

* + - ambele redau mai multe fete ale obiectului
    - modifica felul in care arata poza
      * + Perspectiva:

**Efectul de micsorare :** *Dimensiunea obiectului in proiectia 2D este invers proportionala cu distanta de la centrul de proiectie la obiect.*

**Modifica unghiurile dintre dreptele care nu sunt paralele cu planul de proiectie :** *Proiectiile lor converg catre un punct de convergenta.*

* + - * + Axonometrice: se face scalarea laturilor :

Izometrice : laturile sunt scalate cu factori egali pe cele 3 axe

Dimetrice : laturile sunt scalate cu factori egali pe 2 axe

Trimetrice : laturile sunt scalate cu factori diferiti pe axe

Deosebiri:

* + - Perspectiva:
      * + centrul de proiectie la distanta finita de planul de proiectie
        + proiectorii = drepte convergente in centrul de proiectie
    - Axonometrice:
      * + centrul de proiectie este la infinit
        + proiectorii = paralele ce trec prin varfurile obiectului proiectat si au directia specificata.

1. **Care sunt parametrii care definesc volumul vizual al proiectiei perspectiva in OpenGL? Care este planul in care se efectueaza proiectia? Forma geometrica a volumului vizual in proiectia perspectiva.**

* Parametrii: planul din fata (near), planul spate (far), pozitia observatorului.
* Se efectueaza proiectia pe planul din fata, cu centrul de proiectie in pozitia observatorului.
* Trunchi de piramida (frustum) cu baza mica in pozitia observatorului si delimitat de planurile far & near.

1. **BSP (Binary Space Partitioning)**

Cum se construieste arborele bsp si ce se memoreaza in noduri

♣Fiecare nod al arborelui corespunde unui plan de partitionare a spatiului 3D (de regula, planul unui poligon al scenei).  
♣ Fiecare plan de partitionare imparte spatiul in 2 semi-spatii:  
–cel din faţa planului (de aceeasi parte cu normala la plan)  
–cel din spatele planului.  
♣ Se incepe cu un poligon oarecare din lista (de regula primul), pentru care se creaza nodul radacina al arborelui.  
♣ Primitivele din semi-spatiul “faţă” formeaza “lista-faţă”, care va genera subarborele “faţă” al nodului.  
♣ Primitivele din semi-spatiul “spate” formeaza “lista-spate”, care va genera subarborele “spate” al nodului.  
♣ Se alege un poligon din lista-faţă si se creaza nodul radacina al subarborelui “faţă”, etc.

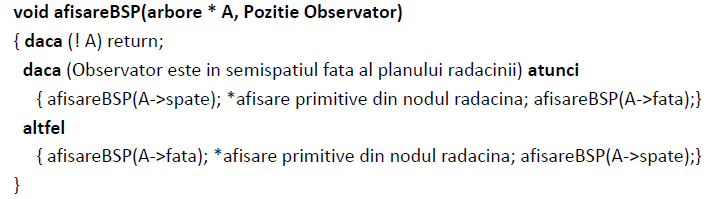
Cand trebuie reconstruit arborele BSP?

Nu trebuie reconstruit la fiecare cadru imagine : avantaj pentru scenele statice.

Cum se afiseaza o scena reprezentata prin BSP?

-Tine cont de pozitia observatorului  
-Afisare “back-to-front”: se incepe cu primitiva cea mai indepartata de observator  
- Se porneste din radacina arborelui si se avanseaza in arbore pana la frunze, in functie de pozitia  
observatorului fata de planul atasat fiecarui nod.  
- In fiecare nod: se avanseaza in subarborele nodului care este opus observatorului fata de plan.

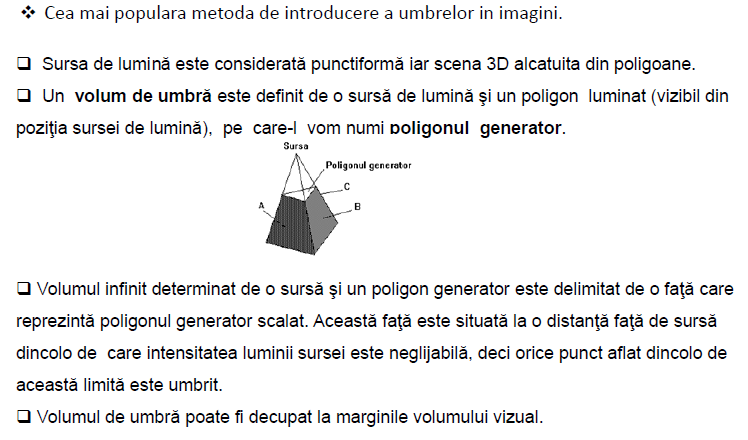
Functia de afisare BSP

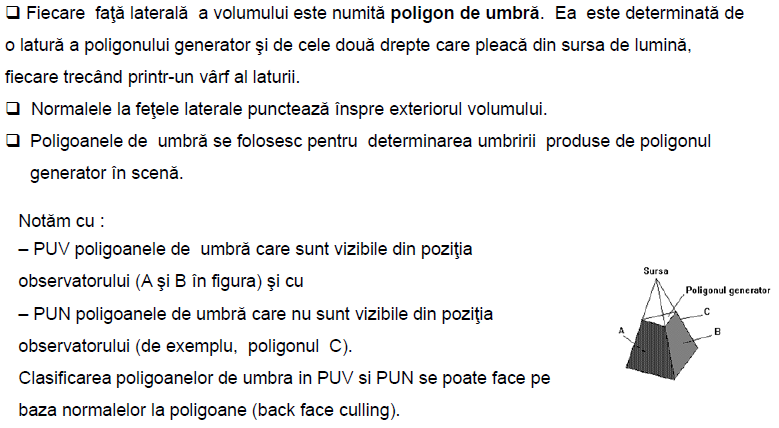


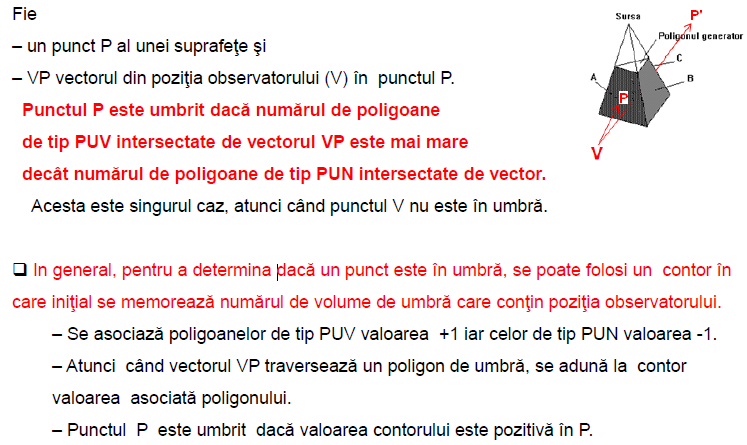
Cum poate fi utilizat algoritmul pentru eliminarea obiectelor nevizibile din banda grafica?

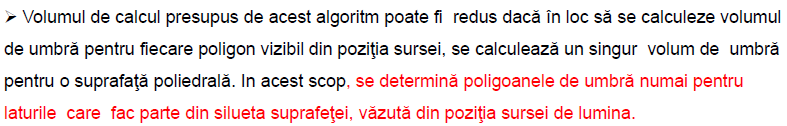
Daca planul de partitionare al unui nod nu intersecteaza volumul vizual, atunci numai  
subarborele aflat de aceeasi parte cu volumul vizual va fi afisat, celalalt subarbore fiind eliminate din banda grafica.

1. **Introducerea umbrelor in imagini prin metoda volumelor de umbra (fara implementare)**

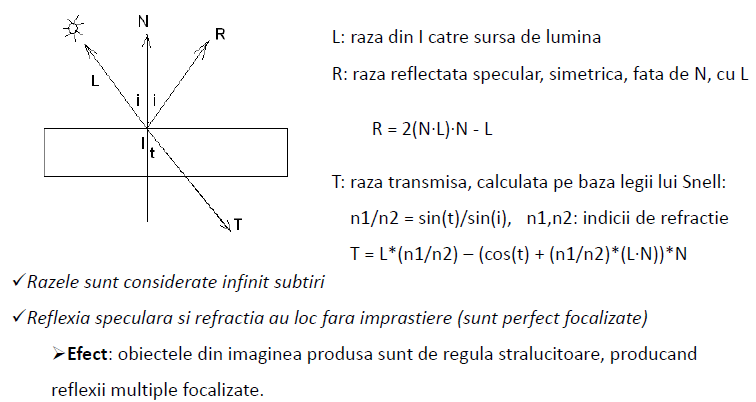


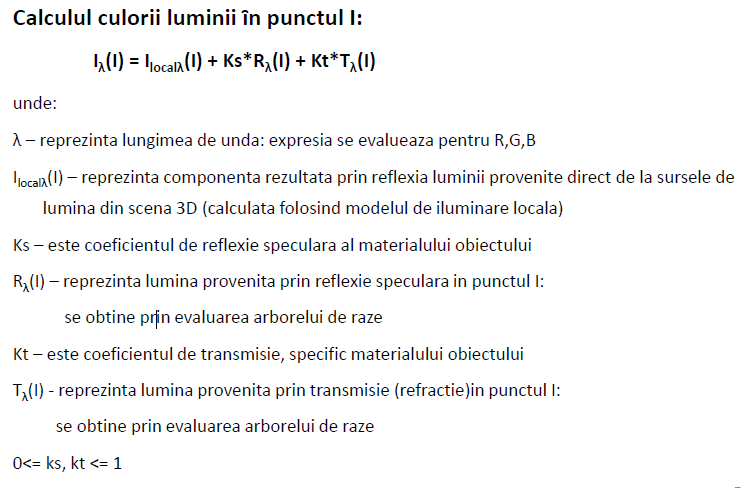




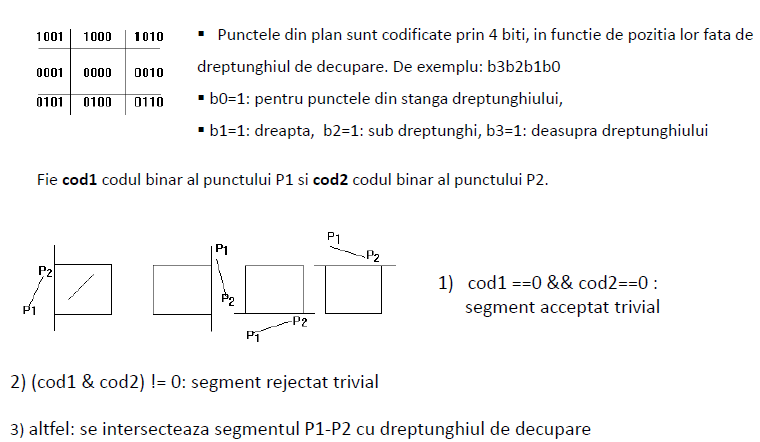


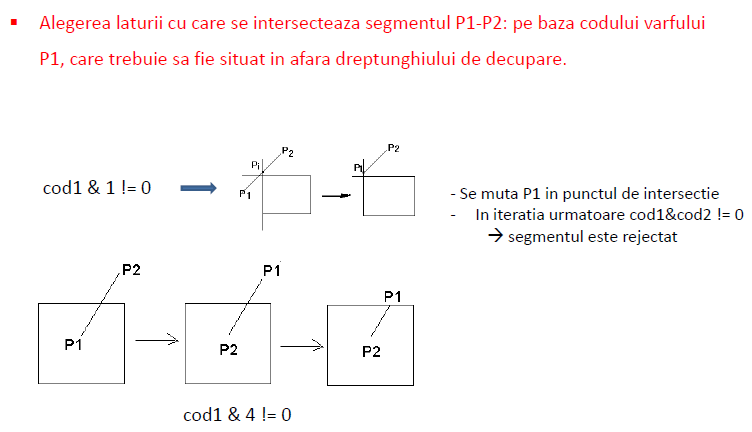
1. **Care sunt razele folosite in calcului culorii unui pixel in algoritmul Ray-Tracing? Cum se calculeaza directia lor? Formula de calcul a culorii unui pixel?**

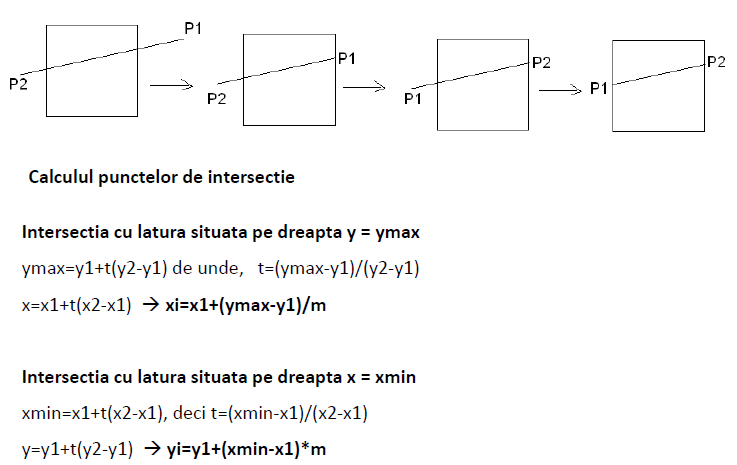
****

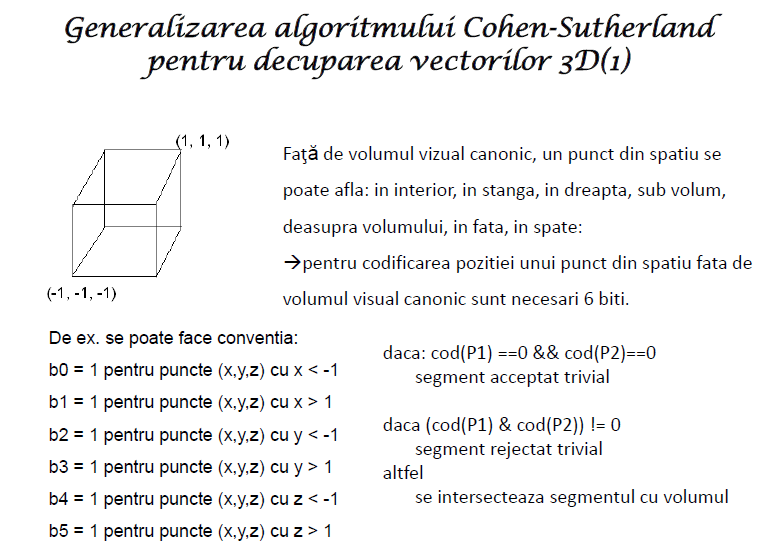
****

1. **Algoritmul Cohen-Sutherland pentru decuparea vectorilor 2D   
   a) descriere  
   b) pasii alg pentru decuparea vect din fig  
   c) implementare in C**

****







1. **Diferentele dintre proiectiile oblice si axonometrice**

Asemanari :

* Proiectorii sunt drepte paralele de directie data: directia de proiectie
* Sunt transformari afine (conserva paralelismul liniilor)
* Unghiurile se conserva doar pentru feţele obiectului paralele cu planul de proiectie.

Oblice :

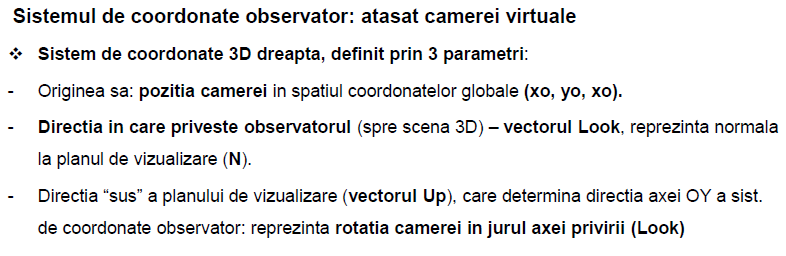
* planul de proiectie este perpendicular pe o axa principala
* proiectorii nu sunt perpendiculari pe planul de proiectie
* fetele paralele cu planul de proiectie se proiecteaza fara alterare unghiuri & laturi

Axonometrice :

* planul de proiectie este oarecare
* proiectorii = paralele ce trec prin varfurile obiectului proiectat si au directia specificata => sunt perpendiculari
* redau mai multe fete ale obiectului proiectat
* sunt de 3 tipuri: izonometrice, dimetrice, trimetrice

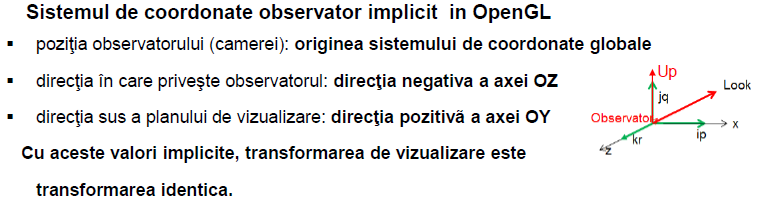
1. **Care sunt parametrii care definesc sistemul de coordonate observator. Care sunt valorile implicite ale acestor parametrii in OpenGL?**

Transformare vizuala: coordonate globale -> coordonate observator.

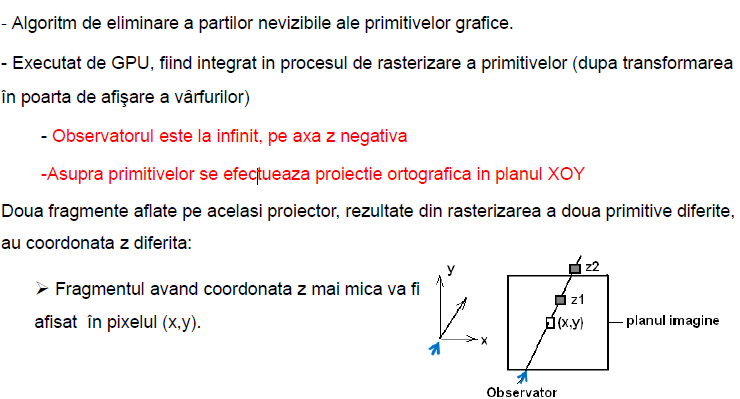


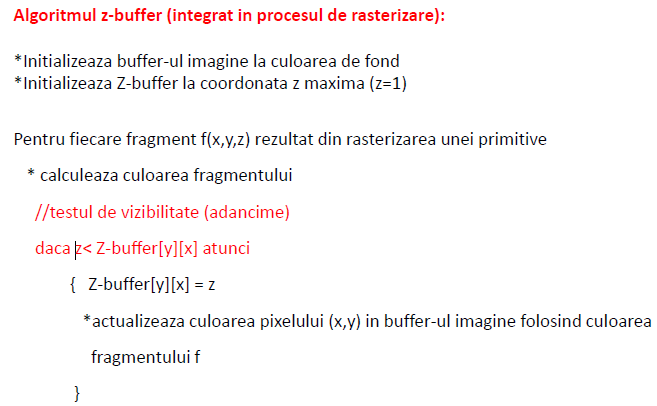
Functia OpenGL: glm::lookAt(glm::vec3(xo,yo,zo), glm::vec3(Px,Py,Pz), glm::vec3(Upx,Upy,Upz));

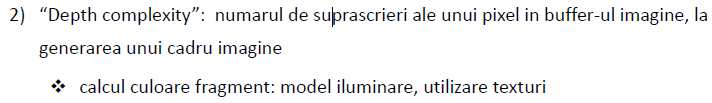
Valorile implicite:



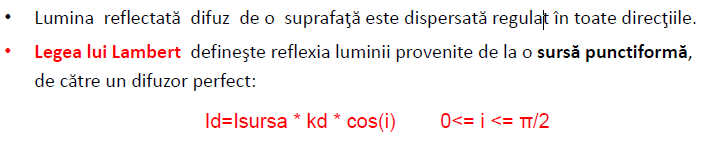
1. **Algoritmul z-buffer. Descriere. Cand se executa? Depth complexity**

****





1. **Reflexia difuza a luminii intr-un punct al unei suprafete 3D**

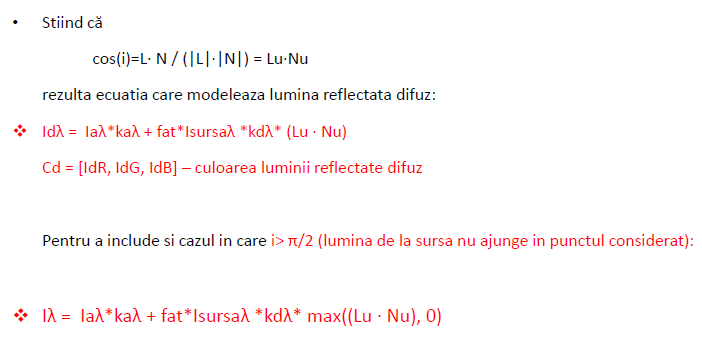
****

Isursa este intensitatea luminii incidente (provenita de la sursa de lumina)

Kd este coeficientul de difuzie a luminii incidente

Id este intensitatea luminii reflectate difuz de suprafata

* Dacă i este mai mare ca π/2, suprafaţa nu primeşte lumină de la sursă (sursa de lumină se află în spatele suprafeţei).



1. **Comparatie intre modelul Gourraud si modelul Phong.**

Dezavantaje Phong fata de modelul Gouraud

\* Componentele Nx, Ny, Nz ale normalei unui fragment se obţin printr-un calcul incremental (analog cu cel folosit pentru calculul culorilor in modelul Gouraud) dar, pentru folosirea în calculul culorii, normala trebuie să fie normalizată.  
\* De asemenea, vectorii L (catre sursa de lumina) si V (catre observator), folositi in modelul de iluminare local la nivel de fragment, sunt calculati si normalizati in programul fragment shader pentru a fi folositi in calculul culorii fragmentului.

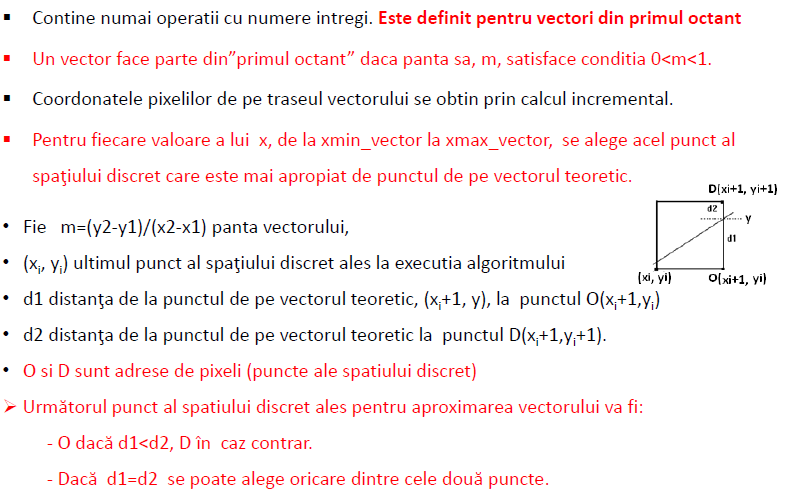
\* Calcule mai complexe la nivel de fragment decat in modelul Gouraud

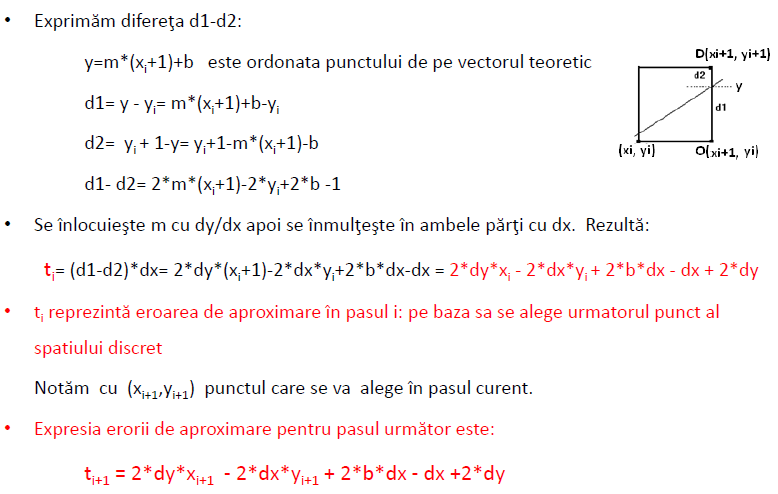
Avantaje Phong fata de modelul Gouraud

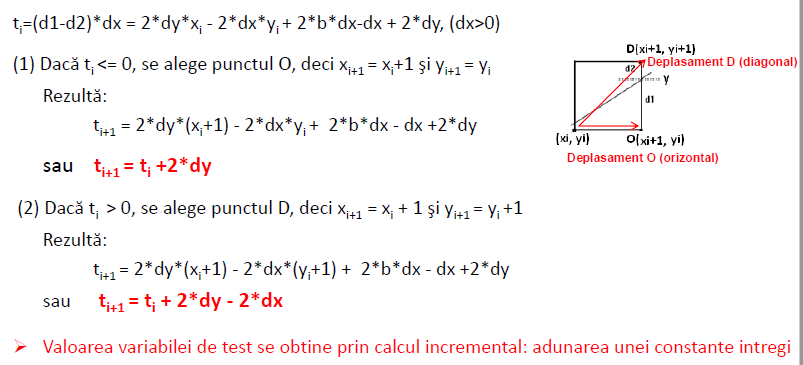
\* Permite redarea reflexiei speculare in orice fragment al unui poligon

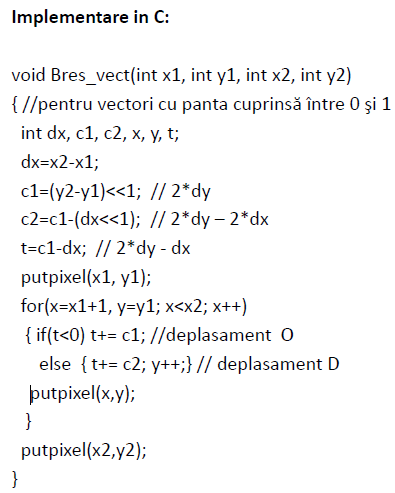
\* Reduce mult efectul de banda Mach

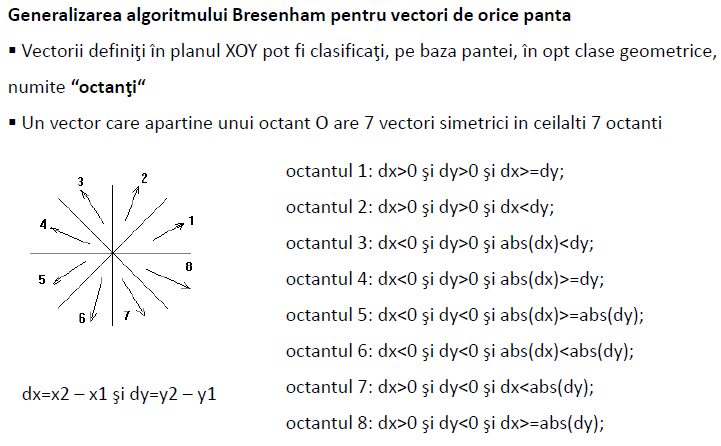
1. **Algoritmul Bresenham pentru rasterizarea vectorilor. Alegerea punctelor spatiului discret.Deducerea valorii expresiei de test. Implementare in C.**

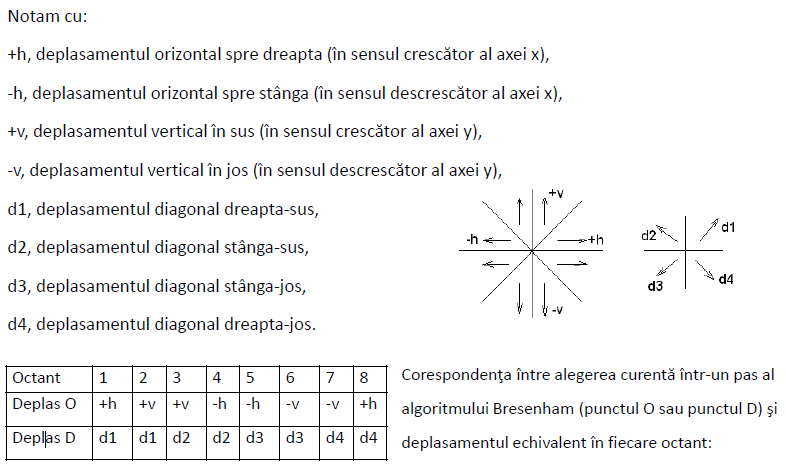
****











1. **Care dintre urmatoarele afirmatii sunt corecte:**

**a. Algoritmul z-buffer necesita rasterizarea poligoanelor scenei 3D in ordinea descrescatoare a distantei lor fata de observator**

*Fals, sunt rasterizate in ordinea in care sunt trimise.*

**b. Bufferul z memoreaza coordonatele z ale fragmentelor de primitive care sunt vizibile intr-o imagine**

*Adevarat, scopul algoritmului este de a elimina partile nevizibile ale primitivelor grafice.*

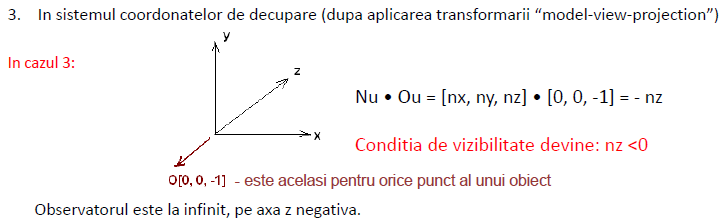
**c. Bufferul z se actualizeaza pentru fiecare fragment rezultat din rasterizarea primitivelor dintr-un cadru imagine**

*Fals, se actualizeaza doar daca trece testul de adancime.*

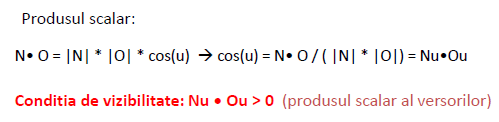
**d. Algoritmul z-buffer este exectutat de procesoarele placii grafice**

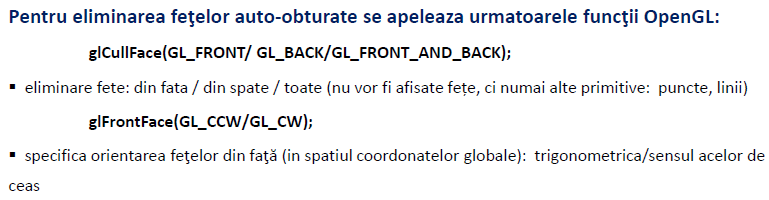
*Adevarat, este executat pe GPU si integrat in procesul de rasterizare a primitivelor.*

1. **Determinarea fetelor auto-obturate ale poliedrelor convexe. Conditia de vizibilitate atunci cand determinarea este efectuata:**

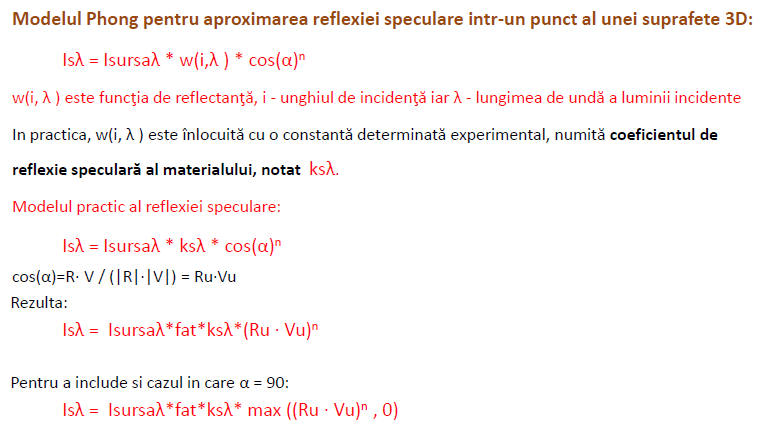
**a) in coordonate de decupare**

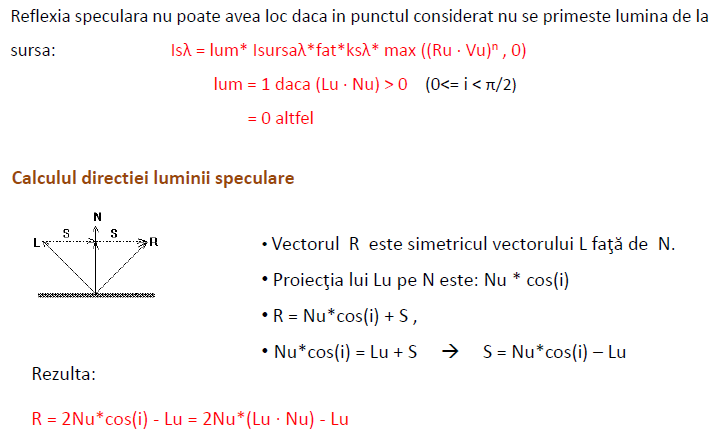
**b) in coordonate globale**

****

****

1. **Aproximarea reflexiei speculare a luminii intr-un punct al unei suprafete 3D, luminata de o sursa punctiforma (modelul empiric)**

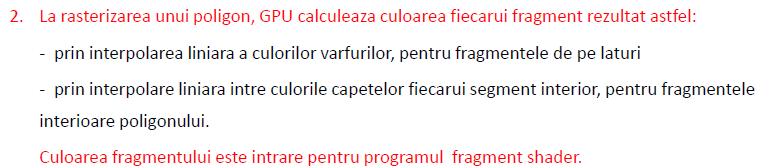
****

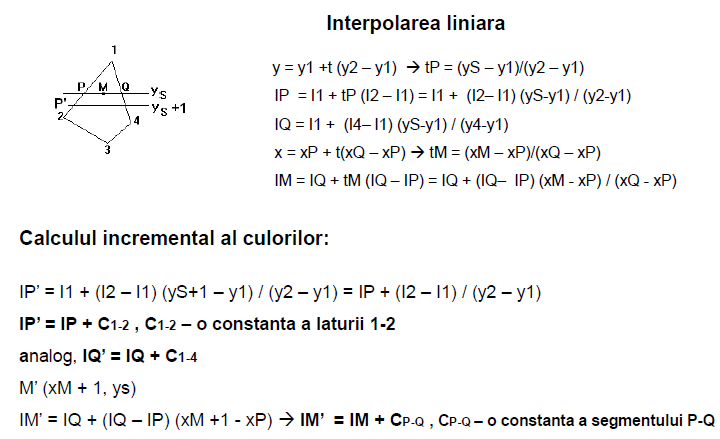
****

1. **Calculul culorilor fragmentelor la afisarea suprafetelor 3D folosind modelul Phong. Avantajele si dezavantajele modelului Phong in comparatie cu alte modele.**

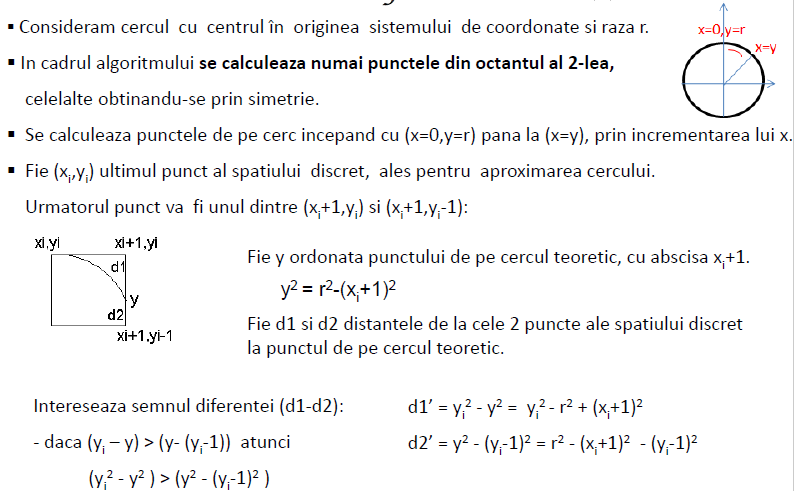
- La momentul rasterizarii, fiecarui vȃrf al unui poligon îi este asociata normala în vȃrf , care poate fi calculata la fel ca pentru modelul Gouraud. Normalele sunt calculate si asociate varfurilor în programul de aplicatie.  
- Pentru fiecare fragment rezultat din rasterizarea unui poligon, GPU calculează o normala prin interpolare liniara intre normalele varfurilor.  
- Culoarea pentru fiecare fragment interior poligonului se obtine pe baza normalei interpolate, folosind un model de iluminare local.  
- La rasterizarea unui poligon se calculeaza o normala pentru fiecare fragment astfel:  
a) prin interpolarea liniara a normalelor varfurilor, pentru fragmentele de pe laturi (la fel ca in modelul Gouraud, pentru culori);  
b) prin interpolare liniara intre normalele capetelor fiecarui segment interior, pentru fragmentele interioare poligonului (la fel ca in modelul Gouraud, pentru culori).

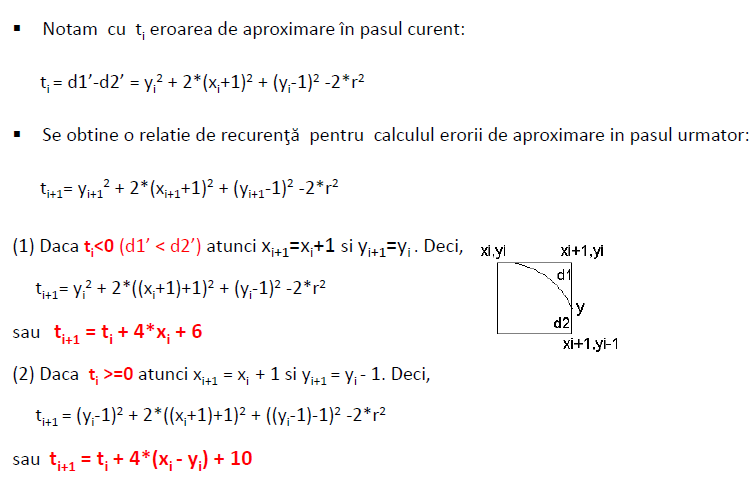
1. **Care dintre urmatoarele NU este o transformare de proiectie afina? (afina = conserva paralelismul liniilor)**
2. **Proiectia oblica**
3. **Proiectia perspectiva**
4. **Proiectia axonometrica**
5. **Proiectia ortografica**
6. **Calculul culorilor fragmentelor la afisarea suprafetelor 3D folosind modelul Gourand. Avantajele si dezavantajele modelului Gourand in comparatie cu alte modele.**

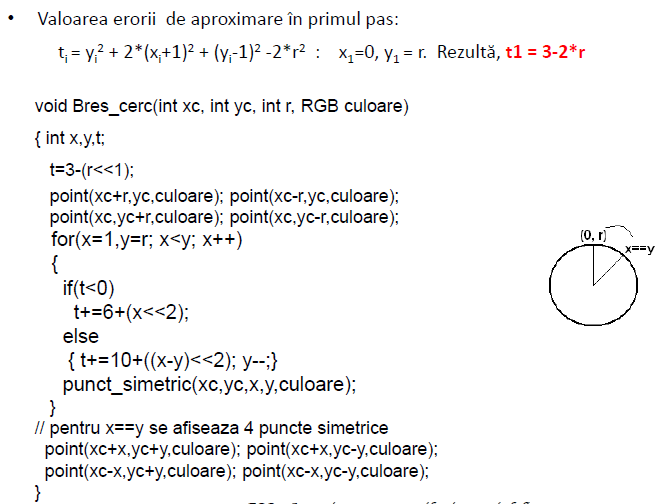
****

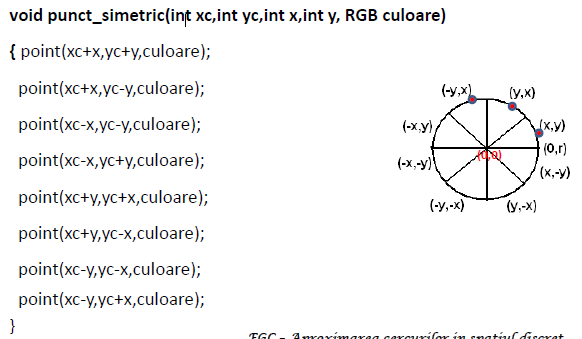


1. **Algoritmul Bresenham pentru rasterizarea cercurilor: alegerea punctelor in spatiul discret, implementare in C, functia de afisare a punctelor de pe cerc**

****







1. **Sa se scrie produsul matriceal prin care se poate exprima rotatia unui punct din spatiul 3D, in jurul unei drepte oarecare. Se va preciza efectul fiecarei matrici.**

Se considera dreapta data printr-un punct (xd, yd, zd) si directia sa, D[a, b, c].  
1. Translatie care face ca dreapta sa treaca prin origine: T(-xd, -yd, -zd)  
2. Alinierea dreptei cu una dintre axele principale, de ex. cu axa OZ:  
2.1. Rotatie in jurul axei OX, cu un unghi ux, prin care dreapta ajunge in planul XOZ: Rox(ux)  
2.2. Rotatie in jurul axei OY, cu un unghi uy, prin care dreapta se suprapune pe axa OZ: Roy(uy)  
3. Rotatia cu unghiul dat, u, in jurul axei pe care s-a aliniat dreapta: rotatie in jurul axei OZ : Roz(u)  
4. Transformarea inversa celei din pasul 2:  
4.1. Rotatie in jurul axei OY, cu unghiul –uy: Roy(-uy)  
4.2. Rotatie in jurul axei OX, cu unghiul –ux: Rox(-ux)  
5. Transformarea inversa celei de la pasul 1: T(xd, yd, zd)

