INTRODUCEREA CETII IN IMAGINI

Prof. univ. dr. ing. Florica Moldoveanu

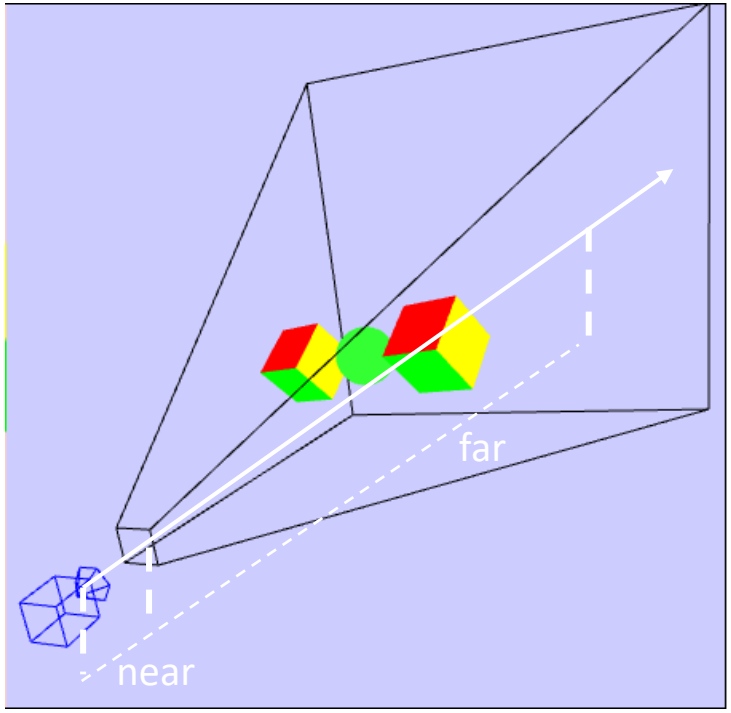
Curs *Elemente de Grafică pe Calculator* – UPB, Automatică şi Calculatoare
2021-2022

MODELAREA CEȚII(1)

Ceaţa adaugă realism imaginii şi ajută în eliminarea defectelor produse de decuparea la nivelul planului din spate al volumului vizual:

❑ Fără ceaţă, pe măsură ce observatorul se îndepărtează de un obiect, obiectul se aproprie de planul din spate şi este decupat atunci când planul din spate îl intersectează.

❑ Cu ceaţă, dacă densitatea ceţii creşte proporţional cu distanţa de la observator, se crează efectul de distanţă pentru obiectele îndepărtate, iar obiectele decupate la nivelul planului din spate nu mai sunt vizibile.



MODELAREA CETII(2)

Fie: Cfog – culoarea de ceaţă

Clocal – culoarea calculată într-un punct din scena 3D folosind modelul de iluminare locala

0<= Φ <= 1 – factorul de ceaţă, proporţional cu distanţa de la observator la punct

(Φ=1 : ceata este opacă)

Culoarea punctului, afectata de ceata: **Cloc_fog = (1-Φ)*Clocal + Φ*Cfog**

Fie dist distanta de la observator la punctul in care se calculeaza culoarea afectata de ceata.

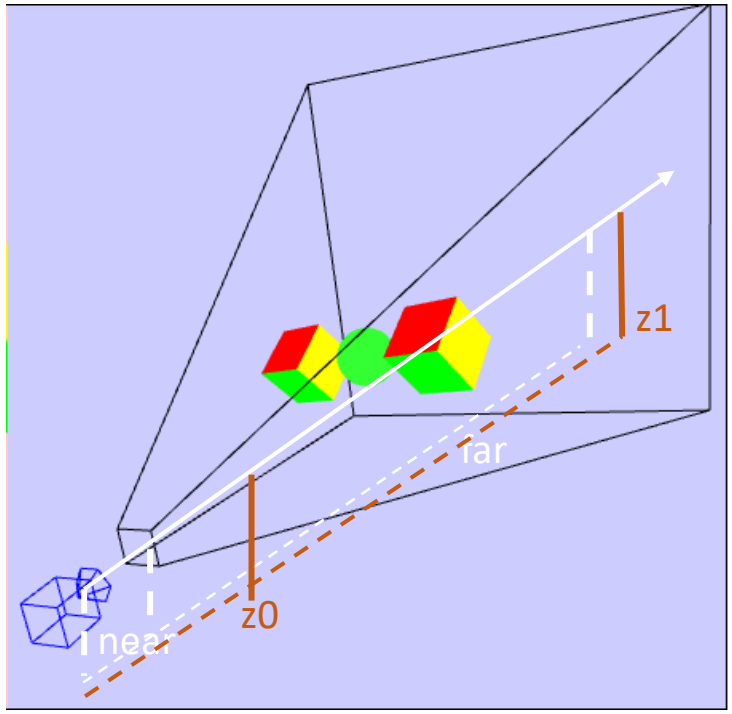
Factorul de ceaţă liniar

-Actioneaza intre 2 distante definite in sistemul coordonatelor observator, z0 si z1, masurate pe directia in care priveste observatorul (axa z).

Φ = 0 pentru dist < z0

Φ = (dist- z0) / (z1- z0) pentru z0 <= dist <= z1

Φ = 1 pentru dist > z1



MODELAREA CETII(3)

Factorii de ceaţă exponentiali

$\Phi = 1 - e^f$

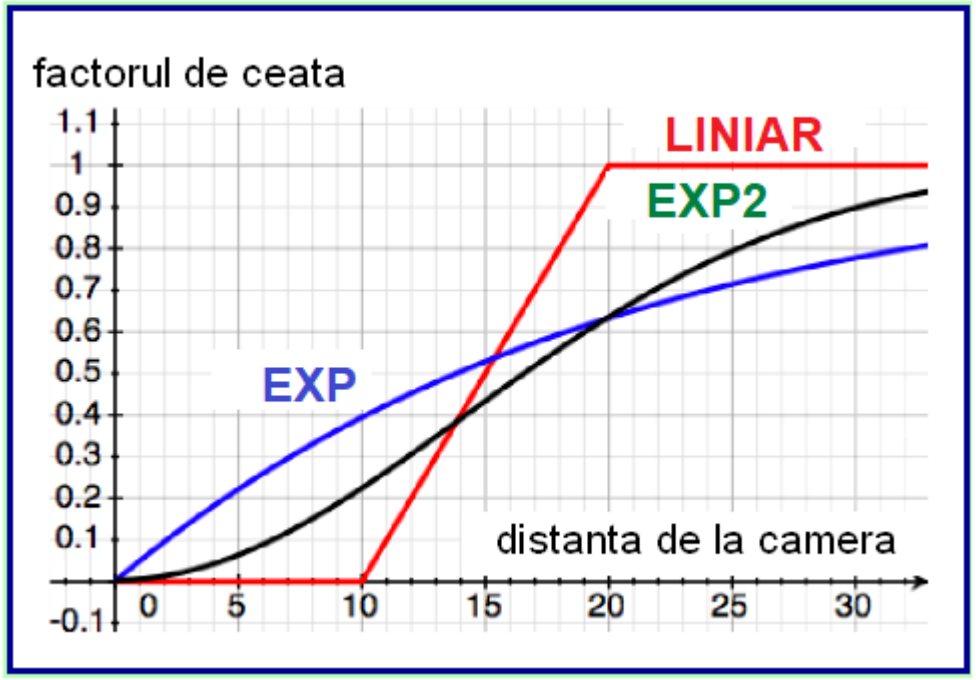
EXP: f = (- fog_density*dist)

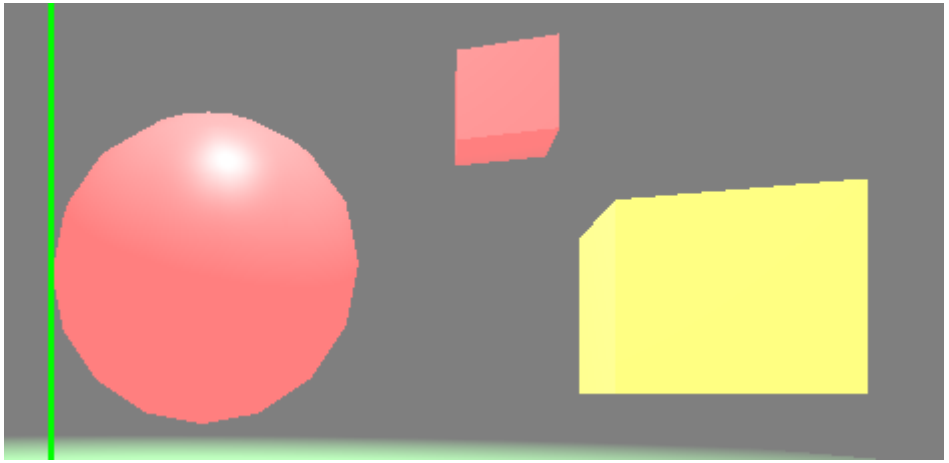
EXP2: f = (- fog_density*dist)^2

- fog_density: densitatea cetii (un numar real, de ex. 0.05) - modeleaza atenuarea luminii datorata cetii.

z0 = 10; z1 = 20;

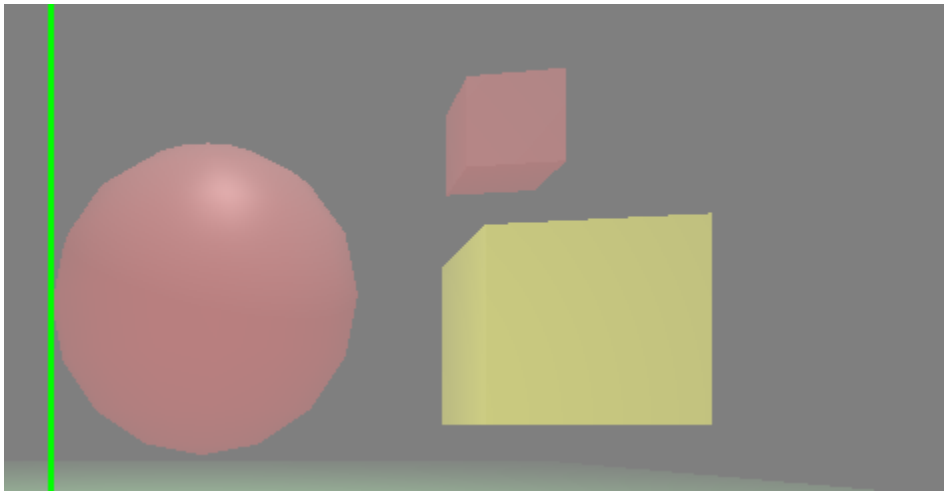
fog_density = 0.3





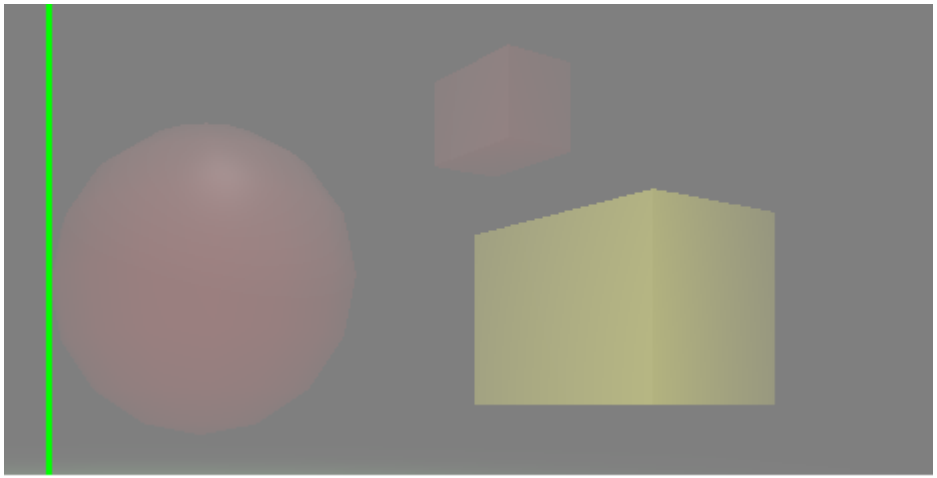
Factorul de ceata liniar

```
float z0 = 3, z1 = 9;
// z1 = far_distance -1
```



Factorul de ceață EXP

```
vec3 fogColor = vec3(0.5, 0.5, 0.5);
float fogDensity = 0.3;
```



Factorul de ceață EXP2

```
vec3 fogColor = vec3(0.5, 0.5, 0.5);
float fogDensity = 0.3;
```

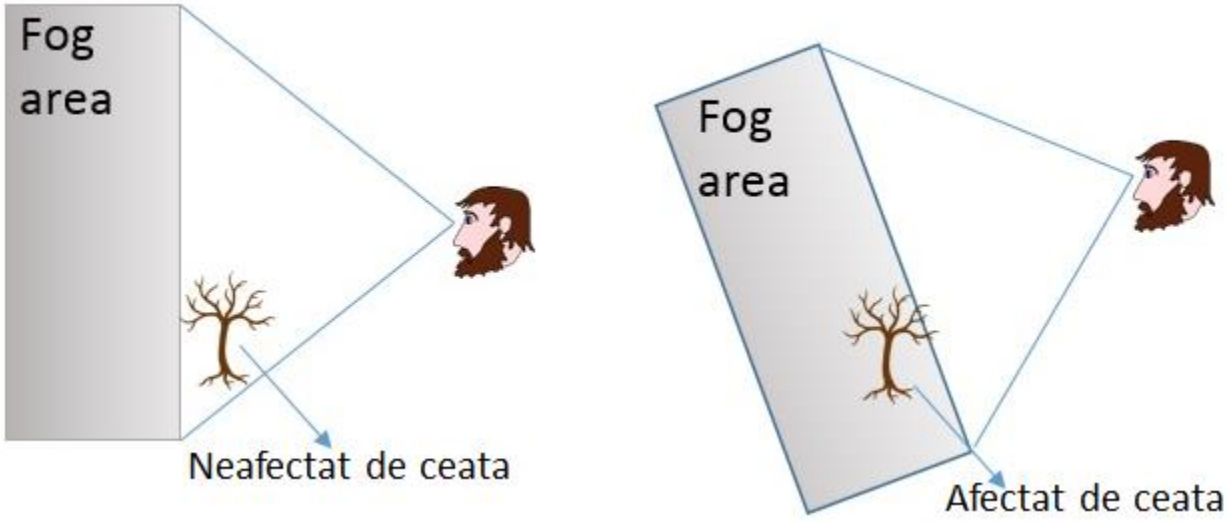
DISTANTA - dist

Distanta fata de observator a punctului in care se calculeaza culoarea:

- Punctul este in coordonate observator

1. **dist** este coordonata z a punctului, la fel ca distantele z0 si z1

Daca se roteste camera, fara a-i modifica pozitia, coord. z a punctului se modifica → un obiect aflat la o distanta d de observator poate sa intre/iasa din zona de ceata:



2. **dist** este lungimea vectorului de la observator (0,0,0) la punct.

Varianta 1 este mai ieftina computational si preferata atunci cand nu este rotita camera.

CALCULUL CULORII DE CEATA (1)

```
layout(location = 0) in vec3 v_position; //pozitia varfului in coord obiect
layout(location = 1) in vec3 v_normal;
uniform mat4 Model;  uniform mat4 View; uniform mat4 Projection;
uniform vec3 light_position; uniform vec3 eye_position;
uniform float material_kd; uniform float material_ks; uniform int material_shininess;
uniform vec3 object_color; uniform vec3 LuminaSursa; uniform vec3 LuminaAmbient;
//uniforme pentru ceata: fogselector, z0, z1, Cfog
// Intrare pentru fragment shader
out vec3 colorWithFog;
void main()
{  vec3 world_pos = (Model * vec4(v_position,1)).xyz; //poz varf in coord globale pt calcul culoare
  vec4 view_pos = (View *Model * vec4(v_position,1)); // poz varf in coord observator
  dist = length(view_pos); //lungimea vectorului (0,0,0) → view_pos
  if(fogSelector == 0)//linear fog
{  fogFactor = (dist – z0)/(z1 – z0);
   fogFactor = clamp( fogFactor, 0.0, 1.0 );//restrange fogFactor la intervalul 0-1
   //calculeaza culoare varf: vec3 color = object_color + cul_ambient + cul_difuza + cul_spec
   colorWithFog  = (1-fogFactor)*color + fogFactor*Cfog;
}  gl_Position = Projection * view_pos;
```

Modelul Gouraud

Se calculeaza culoarea afectata de ceata, colorWithFog in vertex shader

CALCULUL CULORII DE CEATA (2)

Modelul Phong: calculul culorii afectate de ceata in fragment shader

```
//Vertex shader
layout(location = 0) in vec3 in_position;
layout(location = 1) in vec3 in_normal;
uniform mat4 Model; uniform mat4 View; uniform mat4 Projection;

out vec3 world_pos;// pozitia varfului in coord globale
out vec3 world_normal;//normala varfului transformata in coord globale
out vec4 view_pos; //pozitia varfului in coord observator

void main(){
  world_pos = (Model * vec4(in_position,1)).xyz;
  world_normal = normalize(mat3(Model) * in_normal);
  view_pos = View * Model * vec4(in_position,1);
  gl_Position = Projection * view_pos;
 }
```

CALCULUL CULORII DE CEATA (3)

Modelul Phong

Calculul culorii afectate de ceata in fragment shader

```
//fragment shader
uniform vec3 light_position;
uniform vec3 eye_position;
uniform int fogSelector; //0 linear; 1 exponential; 2 exponential square
uniform vec3 Cfog; //ex = vec3(0.5, 0.5,0.5);//culoarea cetii
……..
in vec3 world_pos;// pozitia interpolata a fragmentului, in coord globale
in vec3 world_normal;// normala interpolata a fragmentului, in coord globale
in vec4 view_pos;// pozitia interpolata a fragmentului, in coord observator
layout(location = 0) out vec3 out_color;
void main(){
//Calculeaza Cfragm aplicand modelul de iluminare locala in pozitia world_pos a fragmentului
……..
vec3 color = object_color + ambient_color + diffuse_color + specular_color;
if(fogOn == 0)//ceata dezactivata
   out_color = color;
else
```

CALCULUL CULORII DE CEATA (4)

```
{
dist = length(view_pos); //lungimea vectorului (0,0,0) → view_pos
if(fogSelector == 0)//linear fog
{  fogFactor = (dist - z0)/(z1 - z0); // z0 =20; z1=80
   fogFactor = clamp( fogFactor, 0.0, 1.0 );//restrange fogFactor la intervalul 0-1
 }
else if (fogSelector == 1) // EXP fog
          { fogFactor = exp(-dist * fogDensity);
            fogFactor = 1.0 - clamp(fogFactor, 0.0, 1.0);
          }
        else //  EXP2 fog
         { fogFactor = exp(-(dist * fogDensity)* (dist * fogDensity));
           fogFactor = 1.0 - clamp(fogFactor, 0.0, 1.0);
         }
out_color = (1 - fogFactor)*color + fogFactor * fogColor;
}
}
```