

INTRODUCEREA CETII IN IMAGINI

Prof. univ. dr. ing. Florica Moldoveanu

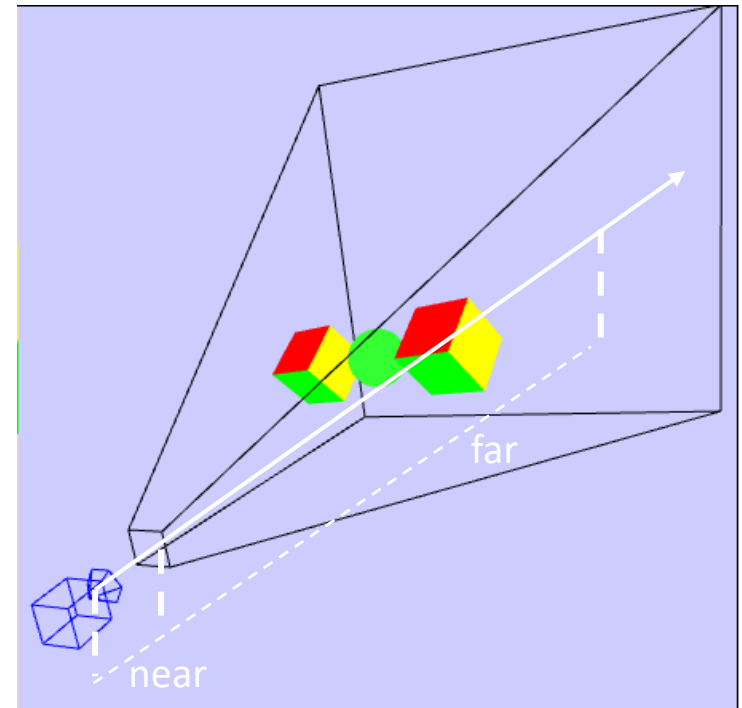
Curs Elemente de Grafică pe Calculator – UPB, Automatică și Calculatoare
2020-2021

MODELAREA CETII(1)

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

❑ Fără ceață, pe măsură ce observatorul se îndepărtează de un obiect, obiectul se apropie de planul din spate și este decupat atunci când planul din spate îl intersectează.

❑ Cu ceață, dacă intensitatea ceții crește proporțional cu distanța de la poziția observatorului, se crează efectul de distanță pentru obiectele îndepărtate, iar obiectele decupate la nivelul planului din spate nu mai sunt vizibile.



MODELAREA CETII(2)

Fie: Cfog – culoarea de ceață

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

$0 \leq \Phi \leq 1$ – factorul de ceață, proporțional cu distanța de la observator la punct

($\Phi=1$: ceata este opacă)

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

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

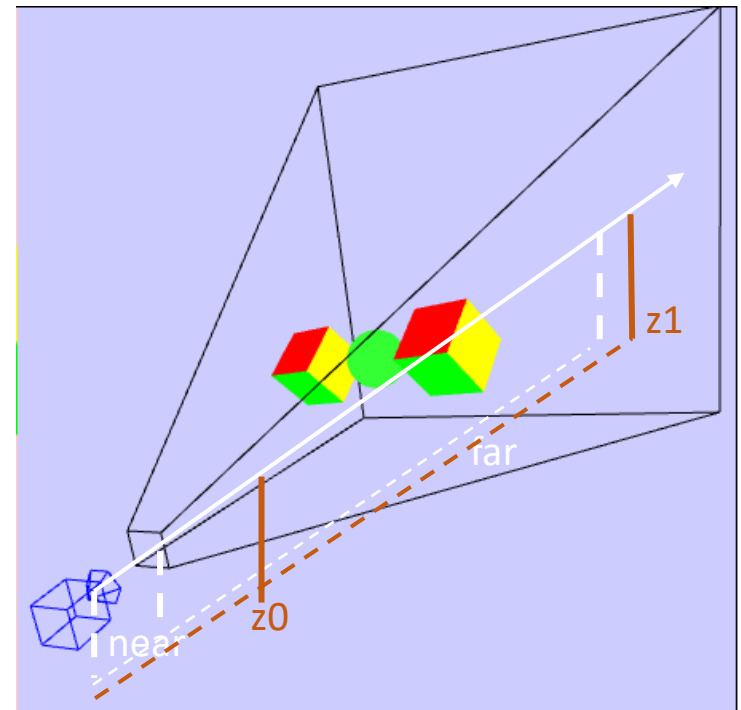
Factorul de ceață liniar

-Actioneaza intre 2 distante definite in sistemul coordonatelor observator, z_0 si z_1 , masurate pe directia in care priveste observatorul (axa z).

$\Phi = 0$ pentru $dist < z_0$

$\Phi = (dist - z_0) / (z_1 - z_0)$ pentru $z_0 \leq dist \leq z_1$

$\Phi = 1$ pentru $dist > z_1$



MODELAREA CETII(3)

Factorii de ceață exponențiali

$$\Phi = 1 - e^f$$

$$\text{EXP: } f = (- \text{fog_density} * \text{dist})$$

$$\text{EXP2: } f = (- \text{fog_density} * \text{dist})^2$$

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

$$z0 = 10; z1 = 20;$$

$$\text{fog_density} = 0.3$$



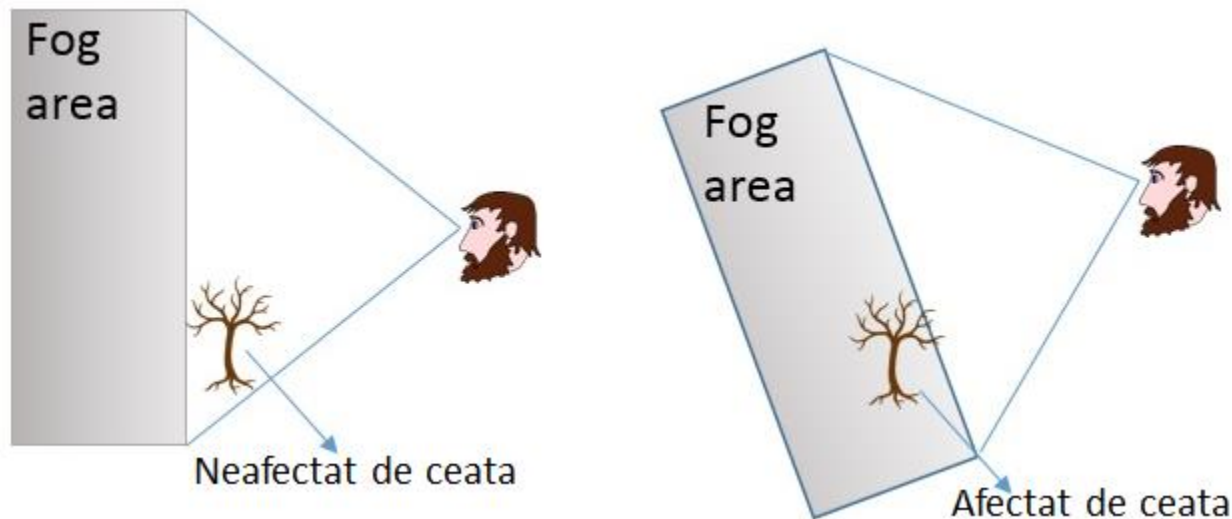
DISTANTA - dist

Distanța față de observator a punctului în care se calculează culoarea:

- Punctul este în coordonate observator

1. **dist** este coordonata z a punctului, la fel ca distanțele z_0 și z_1

Dacă se rotește camera, fără a-i modifica poziția, coord. z a punctului se modifică → un obiect aflat la o distanță d de observator poate să intre/iasă din zona de ceață:



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

Varianta 1 este mai ieftină computațional și preferată atunci când nu este rotită 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

// iesire pentru fragment shader

out vec3 Cloc_fog;

void main()

{ vec3 world_pos = (Model * vec4(v_position,1)).xyz;

vec4 view_pos = (View * Model * vec4(v_position,1));

dist = length(view_pos); //lungimea vectorului (0,0,0) → view_pos

if(fogSelector == 0)//linear fog

{ fogFactor = (dist - 20)/(80 - 20); // z0 =20; z1=80

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

Cloc_fog = (1-fogFactor)*color + fogFactor*Cfog;

} gl_Position = Projection * view_pos;

Se calculeaza culoarea afectata de ceata,
Cloc_fog in vertex shader (**modelul
Gouraud**):

Culoarea de ceață interpolata, **Cloc_fog**,
este intrare pentru fragment shader.

CALCULUL CULORII DE CEATA (2)

In fragment shader(modelul Phong):

Fragment shader primește pozitia fragmentului în coordonate observator

- pozitia varfului în coordonate observator este iesire din Vertex shader
- pozitiile varfurilor in coordonate observator sunt interpolate de GPU la nivel de fragmente
- pozitia fragmentului in coordonate observator este intrare pt Fragment shader
- în Fragment shader se calculeaza Φ folosind distanta (**dist**) de la observator la pozitia fragmentului în coordonate observator

INTRODUCEREA CETII IN IMAGINI (2)

// Calculul culorii 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;

}


```
//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
```

```
void main(){
```

```
//Calculeaza Cfragm aplicand modelul de iluminare locala in pozitia world_pos a fragmentului
```

```
    Cfragm = .....
```

```
//Calculeaza distanta de la pozitia view_pos a fragmentului la observator
```

```
dist = length(view_pos); //lungimea vectorului (0,0,0) → view_pos
```

```
if(fogSelector == 0)//linear fog
```

```
{ fogFactor = (dist - 20)/(80 - 20); // z0 =20; z1=80
```

```
    fogFactor = clamp( fogFactor, 0.0, 1.0 );//restrange fogFactor la intervalul 0-1
```

```
else .....
```

```
gl_FragColor = (1-fogFactor)*Cfragm + fogFactor*Cfog;
```

```
}
```