

TOON SHADING

Prof. univ. dr. ing. Florica Moldoveanu

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

Afișarea în 2 nuanțe

Redare non-fotorealistă

- Tehnici și stiluri de redare care crează imagini diferite de cele fotografice

Toon shading (cel shading)

- Tehnici de redare non-fotorealistă utilizate pentru a obține aparența de desen manual al unui obiect tri-dimensional.

Pot fi implementate în fragment shader sau în vertex shader. Exemple:

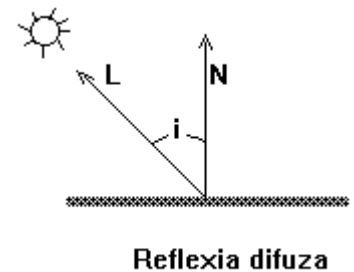
1. Afișarea în 2 nuanțe: culoare cu lumină și fără lumină

```
vec3 color = emissive_light + ambient_light; // fără lumină
```

```
if (attenuationFactor*dot(N, L) > 0.1) // cu lumină
```

```
    color = light_intensity * emissive_light;
```

```
out_color = color; //culoare fragment
```



Iluminare difuză în trepte

```
NL = dot(N, L);
```

```
//// în 4 trepte //////////
```

```
if(0 <= NL && NL <= 0.25) NL =0;
```

```
if(0.25 <= NL && NL <= 0.5) NL =0.25;
```

```
if(0.5 <= NL && NL <= 0.75) NL =0.5;
```

```
if(0.75 <= NL && NL <= 1) NL =0.75;
```

```
//// în 2 trepte //////////
```

```
if(0 <= NL && NL <= 0.5) NL =0;
```

```
if(0.5 < NL && NL <= 1) NL =0.5;
```

Exemplu:

////////// **Fragment shader: Phong shading** //////////

.....

uniform int diffuse_Toon;

uniform int two_Tone;

layout(location = 0) out vec3 out_color;

void main()

{

vec3 L = normalize(light_position - world_position);

vec3 N = normalize(world_normal);

float attenuationFactor = 1 / (kc + kl * distance(light_position, world_position) + kq *
pow(distance(light_position, world_position), 2));

float NL = dot(N, L);

if(two_Tone ==1)

{ vec3 color = emissive_light + ambient_light;// neluminat

if (attenuationFactor*dot(N, L) > 0.1)//luminat

color = light_intensity * emissive_light;

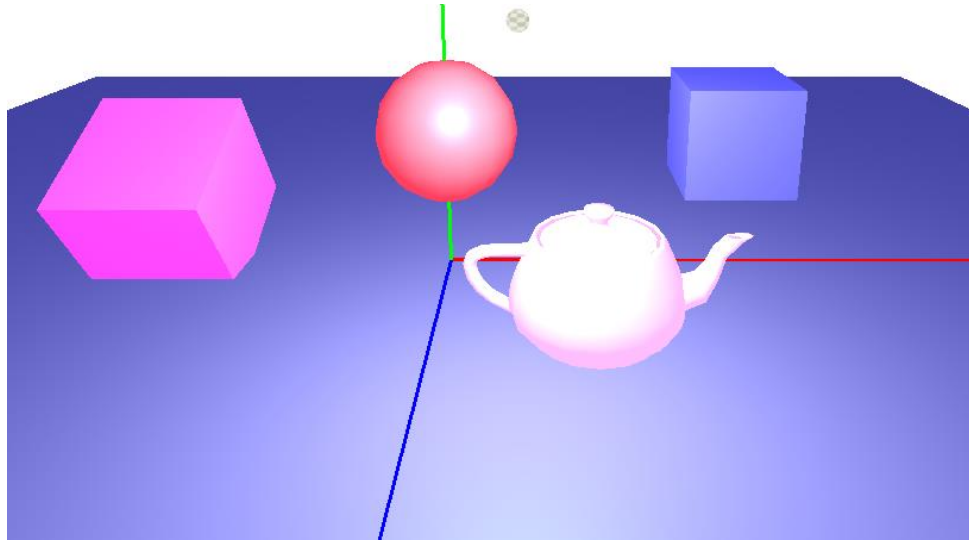
out_color = color;

}

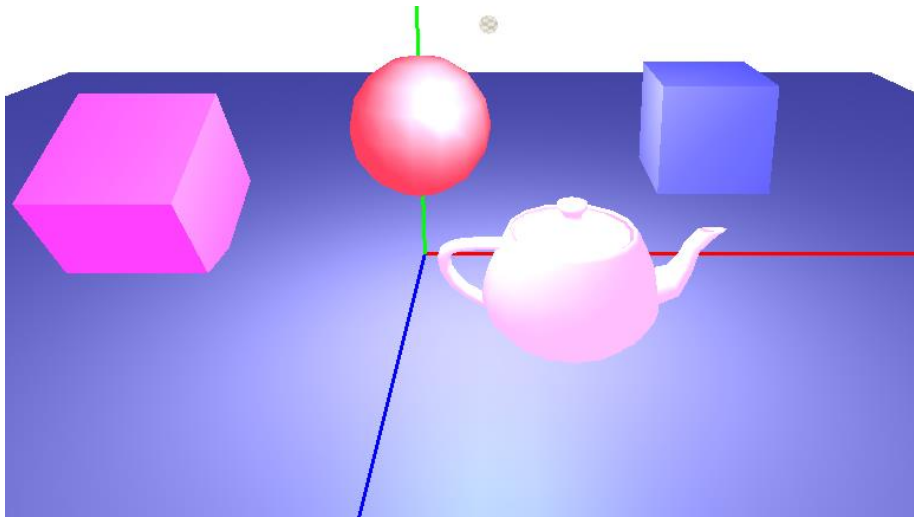
```

else
{
    // Discretizeaza nivelurile de gri
    if(diffuse_Toon == 4 )
    {
        if(0 <= NL && NL <= 0.25) NL =0;
        if(0.25 <= NL && NL <= 0.5) NL =0.25;
        if(0.5 <= NL && NL <= 0.75) NL =0.5;
        if(0.75 <= NL && NL <= 1) NL =0.75;
    }
    else
        if(diffuse_Toon == 2)
        {
            if(0 <= NL && NL <= 0.5) NL =0;
            if(0.5 < NL && NL <= 1) NL =0.5;
        }
    vec3 diffuse_light = material_kd * max(NL,0);
    vec3 specular_light = vec3(0);
    if (NL > 0)
        specular_light = .....;
    .....
    out_color = emissive_light + ambient_light +
        attenuationFactor * light_intensity *(diffuse_light + specular_light);
}
}

```

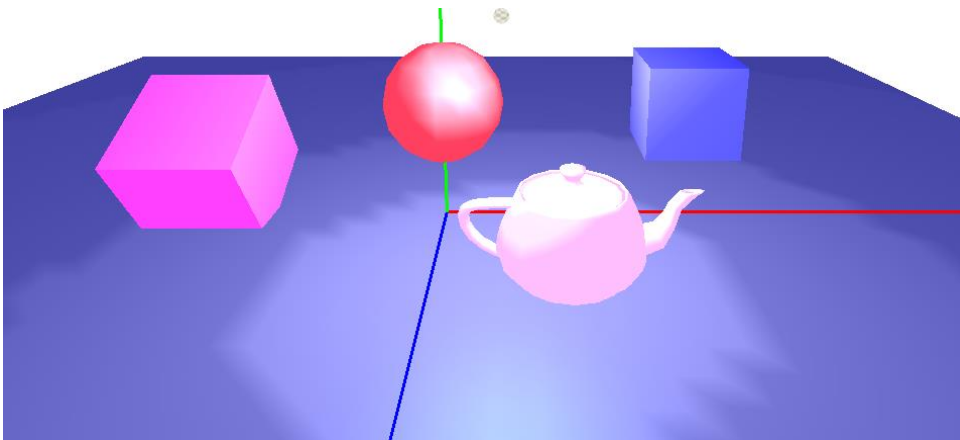
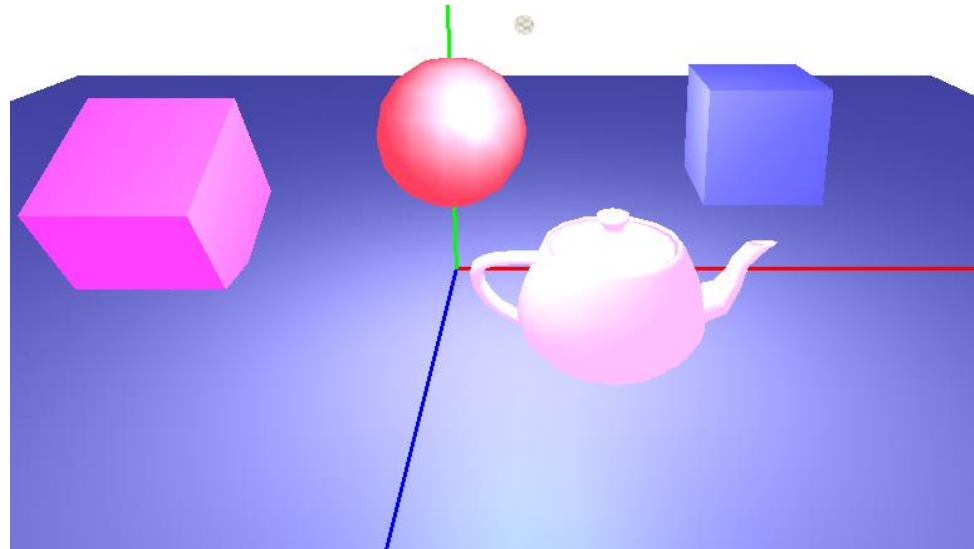


Phong shading

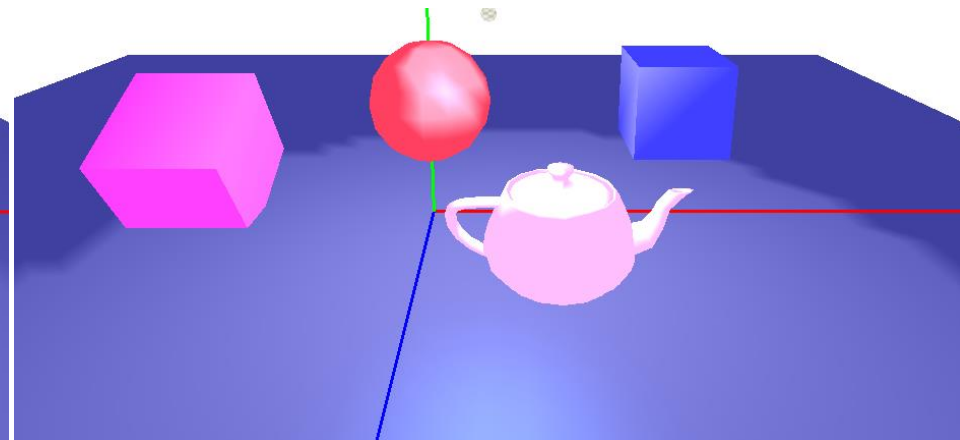


Gouraud shading

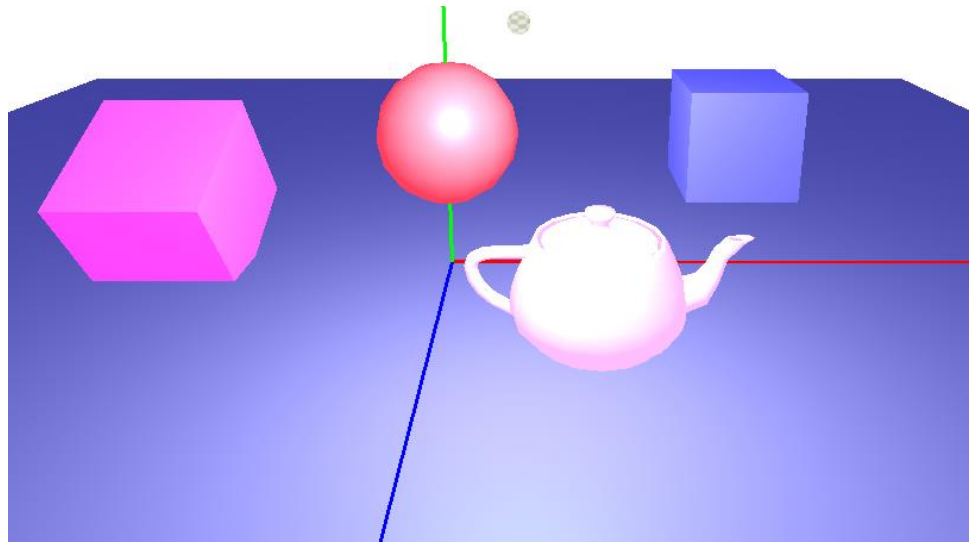
Gouraud



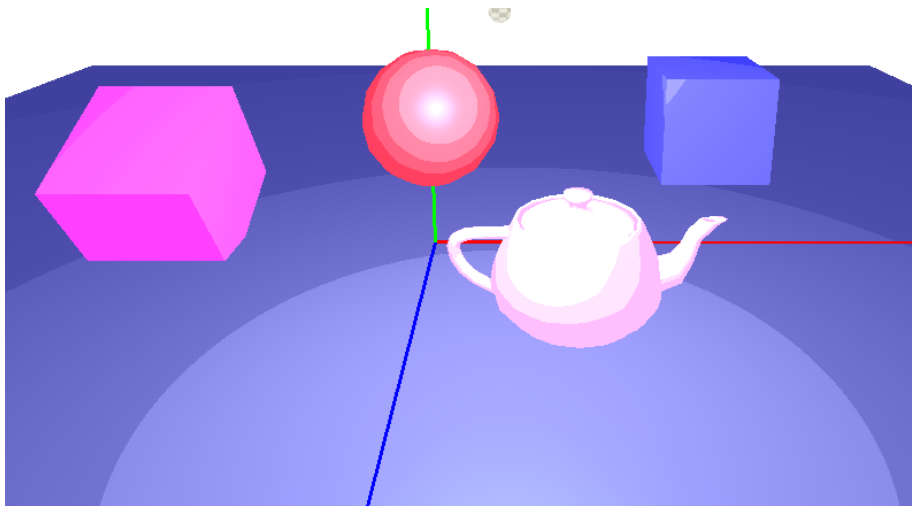
Gouraud – iluminare difuză în 4 trepte



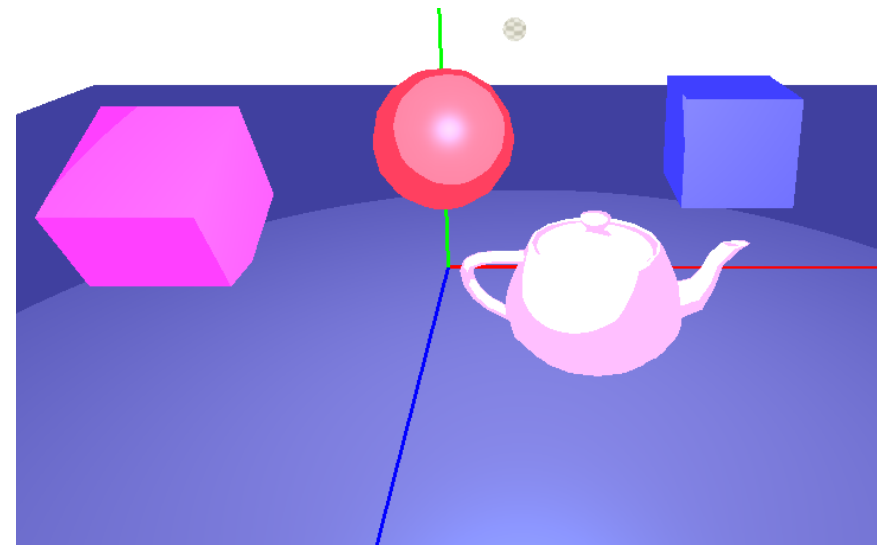
Gouraud – iluminare difuză în 2 trepte



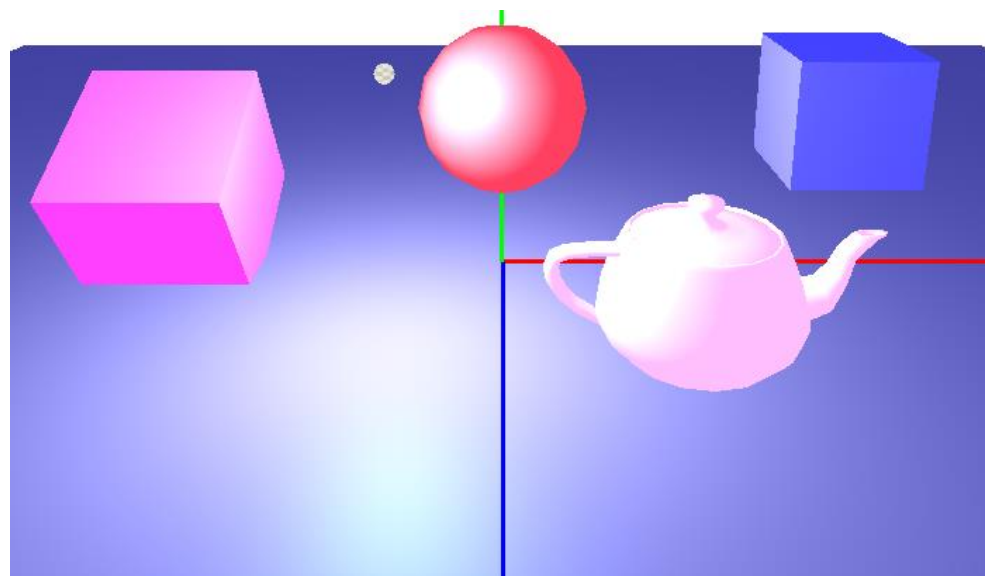
Phong



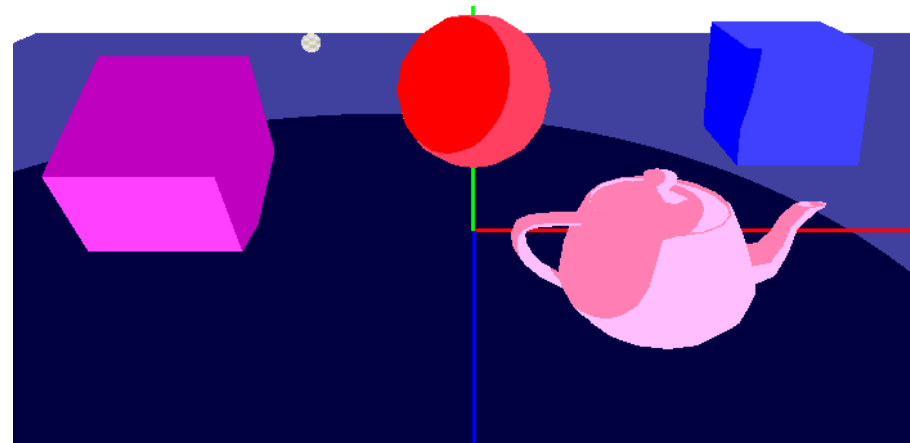
Phong – iluminare difuză în 4 trepte



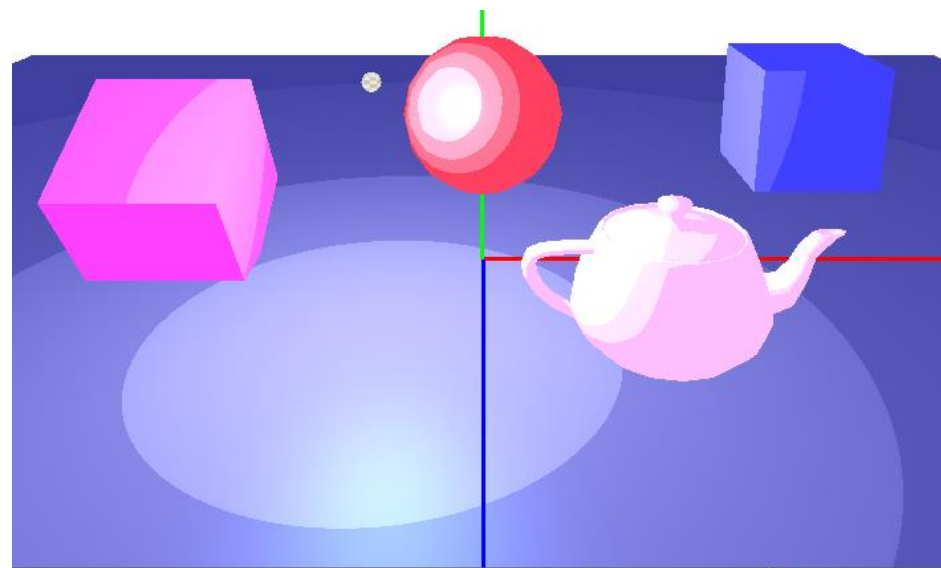
Phong – iluminare difuză în 2 trepte



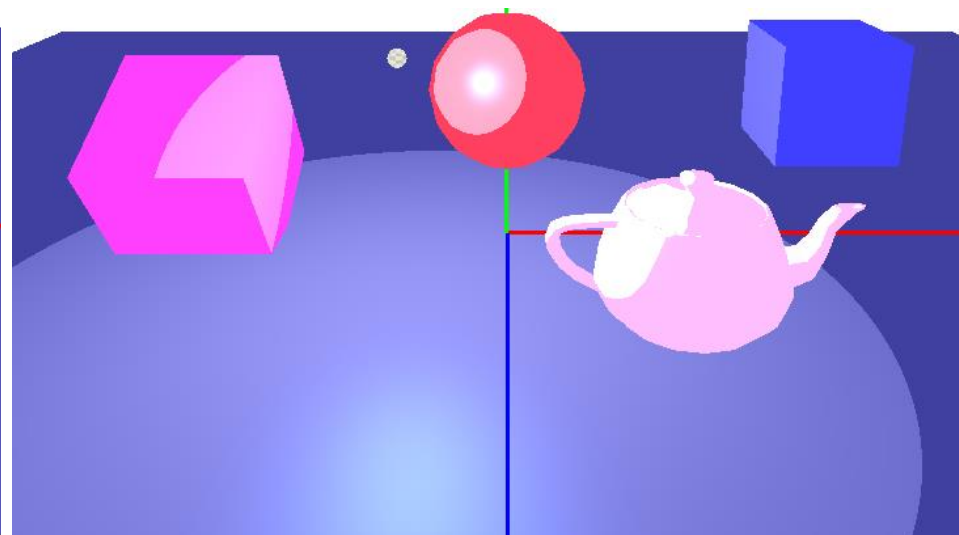
Phong



Phong – 2 nuanțe



Phong – iluminare difuză în 4 trepte



Phong – iluminare difuză în 2 trepte