

Modele de “shading”

Prof. univ. dr. ing. Florica Moldoveanu

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

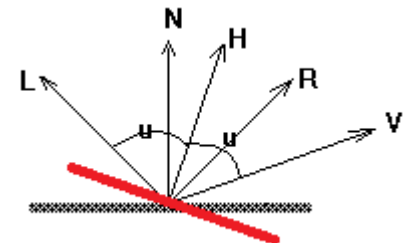
Calculul culorii fragmentelor la redarea suprafețelor 3D (Shading models) (1)

- În vederea vizualizării, suprafețele 3D sunt descompuse în fațete poligonale (triunghiulare) conectate. Poligoanele procesate în banda grafică pot fi fețele unui corp poliedral (fețe plate) sau pot proveni din descompunerea unei suprafețe oarecare, de ex. o sferă (fețele sunt petice de suprafață curba, approximate prin fațete plane).
- La momentul rasterizării unui poligon nu se cunoaște din ce suprafață face parte.

Modelul LAMBERT(1) (Iluminare constantă sau plată)

- Toate fragmentele unui poligon sunt afișate cu aceeași culoare, calculată folosind modelul de iluminare locală, de regulă considerând numai componenta difuză:

$$I_d \lambda = I_a \lambda * k_a \lambda + f_a t * I_{sursa \lambda} * [k_d \lambda * \max((N_u \cdot L_u), 0) + I_{um} * \max((N_u \cdot H_u)^n, 0)]$$



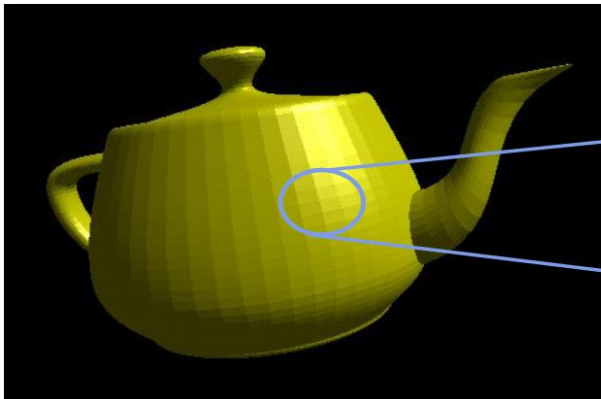
Modelul LAMBERT(2)

Modelul se bazează pe următoarele presupuneri:

1. Poligonul nu este o aproximare a unui petic de suprafață curbă: **vectorul N_u este același în orice punct al poligonului.**
 2. **Sursa de lumină este la infinit:** produsul scalar $(N_u \bullet L_u)$ este atunci constant pe întreaga suprafață a poligonului (aceeași reflexie difuză).
 3. **Observatorul este la infinit:** $(N_u \bullet H_u)$ este constant pe suprafața poligonului (aceeași reflexie speculară).
- Dacă ultimele două cerințe nu sunt satisfăcute, se poate adopta o convenție de calcul al vectorilor L și V pentru un întreg poligon. De exemplu, cei doi vectori pot fi calculați în centrul poligonului.
 - Dacă prima cerință nu este îndeplinită, aproximarea unei suprafețe curbe prin rețeaua de fațete poligonale este vizibilă deoarece intensitățile calculate pentru fațete adiacente cu orientare diferită sunt diferite.

Efectul de banda Mach

Efectul de banda Mach (descoperit de Mach în 1865) este o iluzie optica datorata filtrarii componentei de luminozitate a imaginii formate pe retina, de catre sistemul vizual uman.



▪ Efectul mărește contrastul dintre zone adiacente cu intensități ușor diferite, la frontiera dintre zone, declanșând detecția frontierei în sistemul vizual uman.

- Atunci când o suprafață curba (de exemplu o sferă) este descompusă în fațete poligonale, este de dorit ca laturile fațetelor să nu fie vizibile în imaginea suprafeței.
 - În cazul folosirii modelului Lambert, culorile fațetelor adiacente cu orientări ușor diferite vor fi ușor diferite, iar efectul de banda Mach va mări percepția schimbării de culoare pe laturile fațetelor adiacente.
- În concluzie, modelul Lambert este adecvat vizualizării obiectelor poliedrale.

Modelul Gouraud (1)

- ❑ La momentul rasterizării unui poligon, fiecărui vârf al poligonului îi este asociată o culoare.
- ❑ Procesorul de rasterizare calculează culoarea fiecărui fragment al poligonului astfel:
 - prin interpolare între culorile varfurilor, pentru fragmentele de pe laturi
 - prin interpolare între culorile din capetele segmentului interior pe care se afla fragmentul.

Implementare

1. In **Vertex shader** se calculează culoarea varfului, aplicând modelul de iluminare locală.

Se folosesc: normala varfului (N), poz. sursei și poz. observatorului în coord globale.

Culoarea varfului este ieșire din vertex shader: **out vec3 color**

2. **Fragment shader** primește ca intrare culoarea fragmentului calculată de procesorul de rasterizare (in **vec3 color**).

❖ Normala unui vârf, necesară în vertex shader:

- poate fi calculată exact, folosind ecuația analitică a suprafeței pe care se afla varful
- poate fi aproximată ca medie aritmetică a normalelor la fețele adiacente în varf

Modelul Gouraud - implementare

//vertex shader

```
layout(location = 0) in vec3 v_position; //pozitia varfului  
layout(location = 1) in vec3 v_normal; //normala varfului (N)  
uniform vec3 light; // pozitia sursei de lumina  
uniform vec3 eye; //pozitia observatorului
```

```
out vec3 color; // culoarea varfului
```

```
void main() {
```

```
//calculeaza culoarea varfului aplicand modelul de iluminare locala  
// (N,light,eye)
```

```
.....
```

```
color = cemisiva + ambient + diffuse + specular;
```

```
}
```

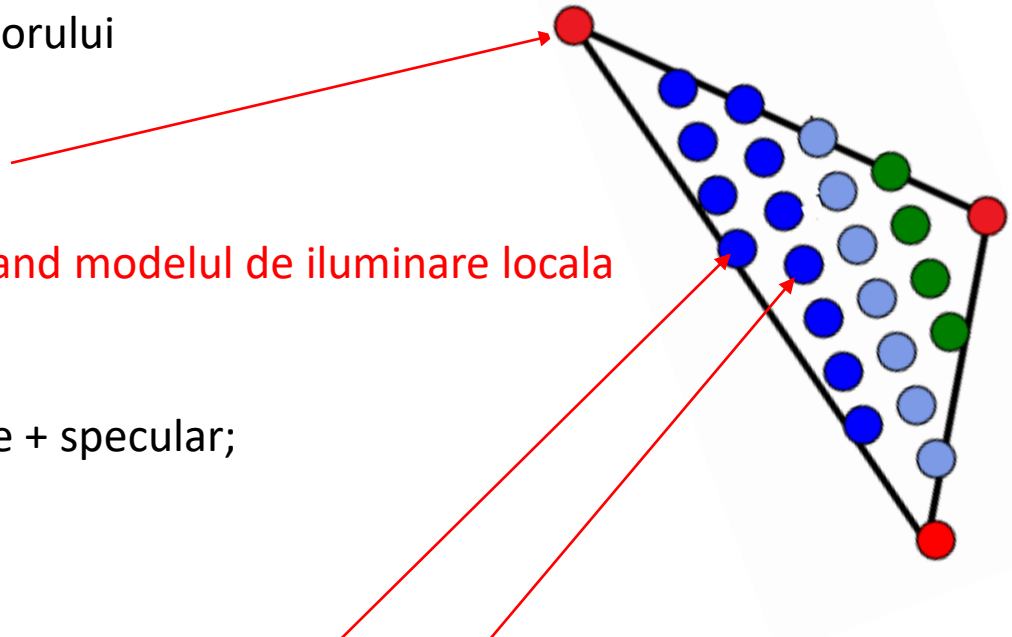
//fragment shader

```
in vec3 color; // culoarea fragmentului obtinuta prin interpolarea culorilor varfurilor
```

```
void main() {
```

```
gl_FragColor = color;
```

```
}
```



Modelul LAMBERT - implementare

Se ataseaza tuturor varfurilor unui poligon (triunghi) aceeasi normala: normala poligonului.

Vertex shader si fragment shader ca in modelul Gouraud

//vertex shader

```
layout(location = 0) in vec3 v_position; //pozitia varfului
layout(location = 1) in vec3 v_normal; //normala poligonului (N)
uniform vec3 light; // pozitia sursei de lumina
uniform vec3 eye; //pozitia observatorului

out vec3 color; // culoarea varfului calculata folosind normala poligonului
void main() {
    //calculeaza culoarea varfului aplicand modelul de iluminare locala (N,light,eye)
    .....
    color = ambient + diffuse + specular;
}
```

//fragment shader

```
in vec3 color; // culoarea fragmentului obtinuta prin interpolarea culorilor varfurilor
void main() {
    gl_FragColor = color;
}
```

Modelul LAMBERT - implementare

Nu se interpoleaza culorile varfurilor

//vertex shader

layout(location = 0) in vec3 v_position;

layout(location = 1) in vec3 v_normal;

uniform vec3 light; // directia sursei de lumina

uniform vec3 eye; // directia observatorului

flat out vec3 color; // culoarea varfului este intrare pt fragment shader

void main() {

//calculeaza light_dir, eye_dir și culoarea, în pozitia varfului, aplicand modelul de iluminare

.....

color = ambient + diffuse + specular;

}

//fragment shader

flat in vec3 color; // culoarea transmisa de vertex shader (neinterpolata)

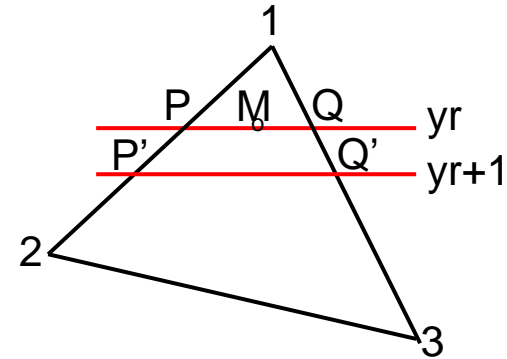
void main() {

gl_FragColor = color;

}

Modelul Gouraud (2)

Interpolarea liniara a culorilor



$$y = y_1 + t (y_2 - y_1) \rightarrow t_P = (y_r - y_1) / (y_2 - y_1)$$

$$I_P = I_1 + t_P (I_2 - I_1) = I_1 + (I_2 - I_1) (y_r - y_1) / (y_2 - y_1) = I_1 + (y_r - y_1) * C_{1-2}$$

$$C_{1-2} = (I_2 - I_1) / (y_2 - y_1) - \text{constanta pentru latura 1-2}$$

$$I_Q = I_1 + (I_3 - I_1) (y_r - y_1) / (y_3 - y_1) = I_1 + (y_r - y_1) * C_{1-3}$$

$$C_{1-3} = (I_3 - I_1) / (y_3 - y_1) - \text{constanta pentru latura 1-3}$$

$$x = x_P + t(x_Q - x_P) \rightarrow t_M = (x_M - x_P) / (x_Q - x_P)$$

$$I_M = I_P + t_M (I_Q - I_P) = I_P + (I_Q - I_P) (x_M - x_P) / (x_Q - x_P) = I_P + (x_M - x_P) * C_{P-Q}$$

$$C_{P-Q} = (I_Q - I_P) / (x_Q - x_P) - \text{constanta pentru segmentul P-Q}$$

Calculul incremental al culorilor

$$I_{P'} = I_1 + (y_{r+1} - y_1) * C_{1-2} = I_1 + (y_r - y_1) * C_{1-2} + C_{1-2} = I_P + C_{1-2}$$

$$I_{P'} = I_P + C_{1-2} - \text{analog, } I_{Q'} = I_Q + C_{1-3}$$

$$M' (x_M + 1, y_s)$$

$$I_{M'} = I_P + (x_M + 1 - x_P) * C_{P-Q} \rightarrow I_{M'} = I_M + C_{P-Q}$$

Modelul Gouraud (3)

Aprecieri asupra modelului Gouraud

- Calculul culorilor fragmentelor este integrat în algoritmul de rasterizare a poligoanelor
- Calcule simple pentru culoarea unui fragment (calcul incremental) → rapid
- Nu permite redarea luminii reflectate specular în fragmentele interioare unui poligon (intensitățile fragmentelor interioare nu pot fi mai mari decât cele din varfuri)
- Reduce efectul de banda Mach, dar nu îl elimină:
 - Culoarele pe latura de adiacență a 2 poligoane sunt aceleași
 - Nu ține cont de tangentele la fețele adiacente pe o latură (culorile de pe laturi se calculează folosind numai culorile din varfuri)
 - Efectul de banda Mach poate fi observat în vecinătatea siluetei unei suprafețe 3D și a zonelor de curbura mare.

Modelul Phong (1)

- ❑ La momentul rasterizării unui poligon, fiecărui vârf îi este asociată normala vârfului.
- ❑ Procesorul de rasterizare calculează normala fiecărui fragment al poligonului astfel:
 - prin interpolare între normalele varfurilor, pentru fragmentele de pe laturi;
 - prin interpolare între normalele capetelor segmentului interior pe care se afla fragmentul (la fel ca în modelul Gouraud, pentru culori).

Implementare

- Normala vârfului este transmisă din programul de aplicație, ca și pentru modelul Gouraud.
- Normala și poziția vârfului, în coordonate globale, sunt ieșiri din vertex shader.
- Procesorul de rasterizare interpolează liniar pozițiile și normalele varfurilor, transmitând la fragment shader poziția și normala fragmentului.
- ❑ În fragment shader se calculează culoarea fragmentului aplicând modelul de iluminare locală în poziția fragmentului (N_{frag} , L ($poz_{sursa} - poz_{fragm}$), V ($poz_{obs} - poz_{fragm}$)).

Modelul Phong - implementare

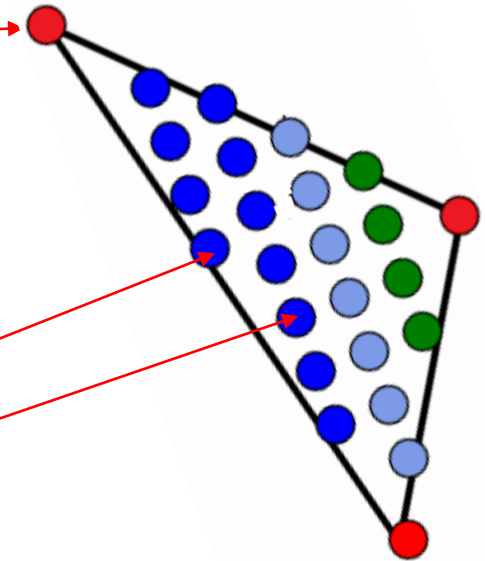
//vertex shader

```
layout(location = 0) in vec3 v_position; //pozitia varfului
layout(location = 1) in vec3 v_normal; //normala varfului (N)
uniform mat4 Model;
uniform mat4 View;
uniform mat4 Projection;
out vec3 world_position;
out vec3 world_normal;
void main() {
    world_position = (Model * vec4(v_position, 1)).xyz;
    world_normal = normalize(mat3(Model) * v_normal);
    gl_Position = Projection * View * Model * vec4(v_position, 1.0);
}
```

//fragment shader

```
uniform vec3 light_position;
uniform vec3 eye_position;
in vec3 world_position; //pozitie fragment
in vec3 world_normal; //normala fragment
void main() {
    * Calculeaza culoarea fragmentului folosind world_position, world_normal, light_position, eye_position
}
```

Calculate prin interpolare



Modelul Phong (2)

Dezavantaje fata de modelul Gouraud

❑ Componentele N_x , N_y , N_z ale normalei unui fragment se obțin (în algoritmul de rasterizare) printr-un calcul incremental analog cu cel folosit pentru calculul culorilor în modelul Gouraud dar, pentru folosirea în calculul culorii, **normala fragmentului trebuie să fie normalizată în fragment shader**:

$N_u = N / |N|$, unde $|N| = (N_x^2 + N_y^2 + N_z^2)^{0.5}$ se folosește funcția GLSL **normalize**

❑ De asemenea, vectorii L (catre sursa de lumina) și V (catre observator), folosiți în modelul de iluminare locală la nivel de fragment, sunt calculați și normalizați în programul fragment shader pentru a fi folosiți în calculul culorii fragmentului; ex: `vec3 Vu = normalize (poz_obs – poz_fragm)`

➤ **Calcule mai complexe la nivel de fragment decât în modelul Gouraud**

Avantaje

❑ **Permite redarea reflexiei speculare în orice fragment al unui poligon**

❑ **Reduce mult efectul de bandă Mach.**