Module : **CBRDF**

Année universitaire : 2019/2020

Date : 07/10/2020

Université Mohamed Khider – Biskra

Faculté des sciences exactes et des sciences de la nature et de la vie

Département : Informatique.

Examen

Exercice 1 : (6pts)

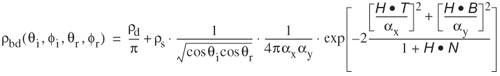
1. Donner la définition des termes suivante : Répartition spectrale d'énergie ; saturation ; rendu des couleurs ; rayonnement visible.
2. On superpose sur un écran blanc plusieurs faisceaux colorés.
   1. Quel type de synthèse réalise-t-on ?
   2. Dire quelle est la couleur observée lorsqu’on superpose:
   3. V + R ⇒ ………… V + C ⇒ B + R ⇒
   4. V + M ⇒………………….. V + B ⇒ R + J ⇒

Exercice 2 : (7pts)

1. Quels sont les différents paramètres intervenant dans le calcul de la quantité de lumière réfléchie vers une position d'observation particulière ?
2. Quel est le rôle des caractéristiques de surface ?
3. Quel est le reflet généré par les matériaux brillants ; les matériaux bruts, les métaux ont des reflets spéculaires ?
4. Quelle est la différence entre les modèles BRDF théoriques et les modèles BRDF empiriques ?

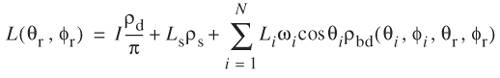
Exercice 3 : (7pts)

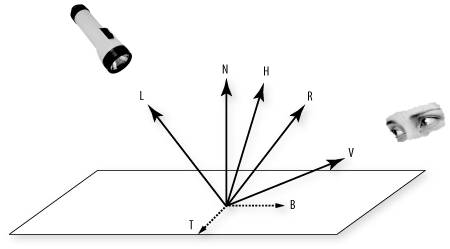
1. Donner le Vertex et le fragment shader du modèle Ward définit par :



* N : est la normale à la surface de l'unité.
* L : est le vecteur unitaire dans la direction de la source lumineuse simulée.
* V : est le vecteur unitaire dans la direction du spectateur.
* R : est le vecteur unitaire dans la direction de réflexion de la source lumineuse simulée.
* H : est la bissectrice angulaire unitaire de V et L (parfois appelée vecteur à mi-chemin).
* T : est un vecteur unitaire dans le plan de la surface qui est perpendiculaire à N (c'est-à-dire la tangente).
* B : est un vecteur unitaire dans le plan de la surface qui est perpendiculaire à la fois à N et à T (c'est-à-dire le binormal).

Le résultat de ce BRDF est inclus dans l'équation globale de l'éclairage suivante :





Exercice 1 :

1. Termes :

* Répartition spectrale d'énergie : Représentation des qualités spectrales d'un rayonnement. Courbe représentant une grandeur énergétique caractéristique quelconque en fonction de la longueur d'onde l .
* Saturation : Sensation visuelle permettant d'évaluer la proportion de couleur pure dans la sensation totale.
* Rendu des couleurs : Effet d'une source lumineuse sur l'aspect chromatique des objets, en comparaison avec l'aspect chro-matique de ces mêmes objets éclairés par un illuminant de référence dans des condi-tions déterminées.
* Rayonnement Visible : rayonnement polychromatique limité pratiquement par les longueurs d'onde de 400 nm à 750 nm.

b.1 : C’est une synthèse ADDITIVE

b.2 : V + R ⇒JAUNE V + C ⇒VERT BLEUTE B + R ⇒MAGENTA

V + M ⇒BLANC V + B ⇒CYAN R + J ⇒ORANGE

Exo 2 4

1. Quels sont les differents paramétres intervenant dans le calcul de la quantité de lumière réfléchie vers une position d'observation particulière ?

- la position de la lumière - de la position de l'observateur - de la normale - de la tangente de la surface. caractéristiques de surface

1. Quel est le rôle des caractéristiques de surface ?

lumière peuvent être réfléchies, transmises ou absorbées, en fonction des propriétés physiques du matériau.

1. Quel est le reflet généré par Les matériaux brillants ; Les matériaux bruts, Les métaux ont des reflets spéculaires

* Les matériaux brillants ont des reflets spéculaires concentrés, proches d'un miroir.
* Les matériaux bruts ont des reflets spéculaires plus étalés.
* Les métaux ont des reflets spéculaires qui sont la couleur du métal plutôt que la couleur de la source lumineuse.

1. Quel est la différence entre Les modèles BRDF théoriques et Les modèles BRDF empiriques.

* Les modèles théoriques tentent de modéliser la physique de la lumière et des matériaux afin de reproduire les propriétés de réflectance observées.
* un modèle empirique est une fonction avec des paramètres ajustables qui est conçue pour ajuster les données de réflectance mesurées pour une certaine classe de matériaux.
* Le volume de données mesurées interdit généralement leur utilisation directe dans un environnement d'infographie, et ces données sont souvent imparfaites ou incomplètes. D'une manière ou d'une autre, les données mesurées doivent être réduites à quelques valeurs utiles qui peuvent être insérées dans une formule ou utilisées pour créer des textures accessibles pendant le rendu. Une variété de méthodes pour réduire les données mesurées a été développée.

**Vertex shader for rendering with Ward's BRDF model**

|  |
| --- |
| attribute vec3 Tangent;  attribute vec3 Binormal;  uniform vec3 LightDir; // Light direction in eye coordinates  uniform vec4 ViewPosition;  varying vec3 N, L, H, R, T, B;  void main()  {  vec3 V, eyeDir;  vec4 pos;  pos = gl\_ModelViewMatrix \* gl\_Vertex;  eyeDir = pos.xyz;  N = normalize(gl\_NormalMatrix \* gl\_Normal);  L = normalize(LightDir);  V = normalize((gl\_ModelViewMatrix \* ViewPosition).xyz - pos.xyz);  H = normalize(L + V);  R = normalize(reflect(eyeDir, N));  T = normalize(gl\_NormalMatrix \* Tangent);  B = normalize(gl\_NormalMatrix \* Binormal);  gl\_Position = ftransform();  } |

**Fragment shader for rendering with Ward's BRDF model**

|  |
| --- |
| const float PI = 3.14159;  const float ONE\_OVER\_PI = 1.0 / PI;  uniform vec4 SurfaceColor; // Base color of surface  uniform vec2 P; // Diffuse (x) and specular reflectance (y)  uniform vec2 A; // Slope distribution in x and y  uniform vec3 Scale; // Scale factors for intensity computation  varying vec3 N, L, H, R, T, B;  void main()  {  float e1, e2, E, cosThetaI, cosThetaR, brdf, intensity;  e1 = dot(H, T) / A.x;  e2 = dot(H, B) / A.y;  E = -2.0 \* ((e1 \* e1 + e2 \* e2) / (1.0 + dot(H, N)));  cosThetaI = dot(N, L); cosThetaR = dot(N, R);  brdf = P.x \* ONE\_OVER\_PI +  P.y \* (1.0 / sqrt(cosThetaI \* cosThetaR)) \*  (1.0 / (4.0 \* PI \* A.x \* A.y)) \* exp(E);  intensity = Scale[0] \* P.x \* ONE\_OVER\_PI +  Scale[1] \* P.y \* cosThetaI \* brdf +  Scale[2] \* dot(H, N) \* P.y;  vec3 color = intensity \* SurfaceColor.rgb;  gl\_FragColor = vec4(color, 1.0);  } |

Module : **CBRDF**

Année universitaire : 2019/2020

Date : 22/10/2020

Université Mohamed Khider – Biskra

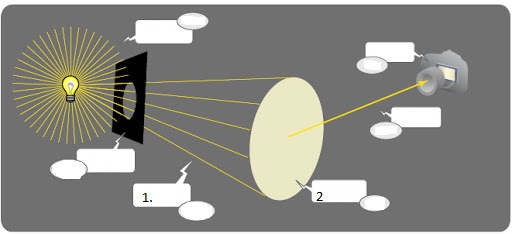
Faculté des sciences exactes et des sciences de la nature et de la vie

Département : Informatique.

Rattrapage

Exercice 1 : (6pts)

1. Compléter le schéma suivant avec les gradeurs photométrique et leur unité de mesure.
2. Expliquer les grandeurs N° 1 et 2.



Exercice 2 : (6pts)

On interpose sur le trajet d’un faisceau de lumière blanche un ou plusieurs filtres colorés.

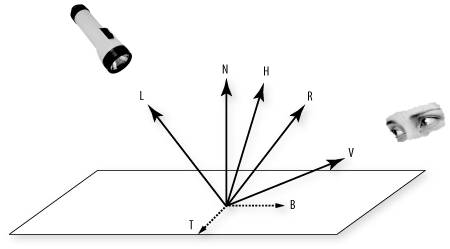
1. Quel type de synthèse réalise-t-on?
2. Dire quelle est la couleur observée lorsque les filtres interposés sont:

J + M ⇒……. C + M⇒…… J + R ⇒……

J + C ⇒…… C + M + J ⇒…. R + V ⇒….

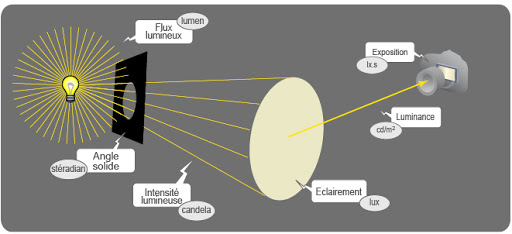
Exercice 3 : (7pts)

1. Donner le Vertex et le fragment shader du modèle Blinn-Phong définit par :

Pbd=Id\*Kd\*cos(θi) + Is\*Ks\* cos(ϕ)n

* N : est la normale à la surface de l'unité.
* L : est le vecteur unitaire dans la direction de la source lumineuse simulée.
* V : est le vecteur unitaire dans la direction du spectateur.
* R : est le vecteur unitaire dans la direction de réflexion de la source lumineuse simulée.
* H : est la bissectrice angulaire unitaire de V et L (parfois appelée vecteur à mi-chemin).

Exo 1:



1. Voir cours

Exo 2

EXERCICE 2: On interpose sur le trajet d’un faisceau de lumire blanche un ou plusieurs filtres colors.

1.Quel type de synthse ralise-t-on?C’est une synthse SOUSTRACTIVE

2.Dire quelle est la couleur observe lorsque les filtres interposs sont:J + M ⇒ROUGEC + M⇒BLEUJ + R ⇒ROUGEJ + C ⇒VERTC + M + J ⇒NOIRR + V ⇒NOIR

Exo3

precision mediump float;

in vec3 normalInterp;

in vec3 vertPos;

uniform int mode;

const vec3 lightPos = vec3(1.0, 1.0, 1.0);

const vec3 lightColor = vec3(1.0, 1.0, 1.0);

const float lightPower = 40.0;

const vec3 ambientColor = vec3(0.1, 0.0, 0.0);

const vec3 diffuseColor = vec3(0.5, 0.0, 0.0);

const vec3 specColor = vec3(1.0, 1.0, 1.0);

const float shininess = 16.0;

const float screenGamma = 2.2; // Assume the monitor is calibrated to the sRGB color space

void main() {

vec3 normal = normalize(normalInterp);

vec3 lightDir = lightPos - vertPos;

float distance = length(lightDir);

distance = distance \* distance;

lightDir = normalize(lightDir);

float lambertian = max(dot(lightDir, normal), 0.0);

float specular = 0.0;

if (lambertian > 0.0) {

vec3 viewDir = normalize(-vertPos);

// this is blinn phong

vec3 halfDir = normalize(lightDir + viewDir);

float specAngle = max(dot(halfDir, normal), 0.0);

specular = pow(specAngle, shininess);

// this is phong (for comparison)

if (mode == 2) {

vec3 reflectDir = reflect(-lightDir, normal);

specAngle = max(dot(reflectDir, viewDir), 0.0);

// note that the exponent is different here

specular = pow(specAngle, shininess/4.0);

}

}

vec3 colorLinear = ambientColor +

diffuseColor \* lambertian \* lightColor \* lightPower / distance +

specColor \* specular \* lightColor \* lightPower / distance;

// apply gamma correction (assume ambientColor, diffuseColor and specColor

// have been linearized, i.e. have no gamma correction in them)

vec3 colorGammaCorrected = pow(colorLinear, vec3(1.0 / screenGamma));

// use the gamma corrected color in the fragment

gl\_FragColor = vec4(colorGammaCorrected, 1.0);

}