

# GPGPU

## Illumination avancée

### CPE

5ETI IMI

## 1 But

L'objectif de ce TP est de mettre en oeuvre des techniques de rendu sur GPU pour obtenir des images plus réalistes. Il permettra :

- d'observer une mise en oeuvre de l'illumination de Phong,
- d'utiliser plusieurs textures dans un même shader,
- d'utiliser des cartes d'occlusion ambiante,
- d'utiliser des cartes d'élévation pour modifier la géométrie,
- d'utiliser l'espace tangent des objets:
  - pour du normal mapping
  - pour du parallax mapping (en optionnel)

Ce TP sera noté.

## 2 Prise en main de l'environnement

### 2.1 Compilation

**Question 1** Compilez le code, assurez-vous de voir une sphère avec une texture de couleur roche.

**Question 2** Observez les fichiers du projet (hors dossier extern) afin de comprendre le fonctionnement du projet.

**Question 3** Observez les effets des appuis claviers dans `keyboard_callback(...)`.

## 3 Illumination de Phong

**Question 4** Observez l'illumination de Phong dans le shader `texture.frag`.

**Question 5** Observer l'importance du nombre de point sur la sphère.

## 4 Utilisation de multiples textures

On peut utiliser dans un shader plusieurs textures. Il faut pour cela assigner une "texture unit" à chaque texture.

- Récupérer l'identifiant de la texture dans le programme grâce à son nom : `glGetUniformLocation(...)`
- Définir le numéro de la texture unit, en affectant une variable uniforme : `glUniform1i(...)`
- Définir la texture unit active : `glActiveTexture(GL_TEXTUREi)` (ou `GL_TEXTURE0+i`)
- Choisir la texture à placer dans le texture unit : `glBindTexture(...)`

Attention, la fonction `glHelper::load_texture(...)` place la texture dans la texture unit active.

## 5 Ambient occlusion

L'illumination de Phong suppose que l'illumination globale éclaire de partout de la même manière (composante ambiante). Cependant certains endroits sont moins accessibles que d'autres et donc devraient moins éclairés. On appelle ce manque d'accessibilité l'occlusion ambiante. Les cartes d'occlusion permettent de définir les zones moins accessibles. Elles peuvent être défini dans l'espace écran à la volé (screen space ambiante occlusion SSAO) ou bien calculé pour l'objet. C'est ce dernier cas qui nous intéresse. Plus la carte d'AO est sombre, plus il faut atténuer la couleur, plus elle est claire, moins il faut l'atténuer.

**Question 6** Chargez deux images, la couleur et la carte d'occlusion, utilisez cette dernière pour assembler la composante ambiante dans les zones de fortes occlusions.

## 6 Mapping

La géométrie étant définie de manière non ordonnée, il est difficile de créer du détail géométrique à partir des triangulations. Pour ajouter de l'information à la géométrie, il est courant d'utiliser des textures représentant des caractéristiques de l'objet en tout point. Cette information peut contenir de l'information sur la position de l'objet, ses dérivées... On parle de carte de déplacement, de carte de normale... À partir de ces textures, des techniques ont été développées pour utiliser ces informations lors de l'affichage et mimer le comportement de la lumière. Parmi ces méthodes, on peut citer : le parallax mapping, bump mapping, normal mapping, ...

## 7 Modification de la géométrie avec une carte de hauteur

**Question 7** Modifier la hauteur des points suivant leur normale (issue de la displacement map), le coefficient utilisé pour la hauteur est de 0.2.

## 8 Modification des informations géométriques

### 8.1 Espace tangent

Pour que l'information à ajouter soit indépendante de l'objet, les modifications sont faites dans ce que l'on appelle l'espace tangent. Cet espace permet de lier la direction de l'objet à la carte de texture. Les textures sont sur le plan  $xy$  et appartiennent au triangle, la normale du triangle doit correspondre au  $z$  de l'image. Afin que l'espace tangent soit défini de manière lisse sur

tout le maillage, on définit le repère sur tous les sommets que l'on somme et que l'on normalise à la fin.

Plus d'information sont disponibles dans les sources, ce travail est déjà implémenté et la matrice de passage est disponible dans le vertex shader.

## 8.2 Normal mapping

Le normal mapping est une technique qui stocke la perturbation des normales dans la texture, avec  $r = n_x$ ,  $g = n_y$ ,  $b = n_z$ . Les normales map sont dans le plan tangent. Il faut replacer les valeurs  $[0, 1]$  entre  $[-1, 1]$ . Les normales étant principalement orienté vers  $z$ , la couleur majoritaire est 0.5, 0.5, 1..

**Question 8** Implémentez le normal mapping en plaçant l'ensemble des éléments dans l'espace tangent.

## 9 Pour aller plus loin

**Question 9** Augmentez le nombre de lumière, passez les en paramètre depuis le programme principale.

**Question 10** Ajoutez les étapes de tessellation pour affiner la géométrie de la sphère à l'affichage.

**Question 11** Implémentez la méthode de parallax Mapping

**Question 12** Effectuer le normal mapping sur des objets complexes.

**Question 13** Ajoutez un décors en utilisant une texture cube.

**Question 14** Ajoutez des boutons sur l'interface graphique (à faire à la main pour l'affichage et le click)

## 10 Sources

<http://foundationsofgameenginedev.com/>

<https://learnopengl.com/Advanced-Lighting/Parallax-Mapping>

OpenGL 4 Shading Language Cookbook, packt : disponible sur <http://univ.scholarvox.com/>



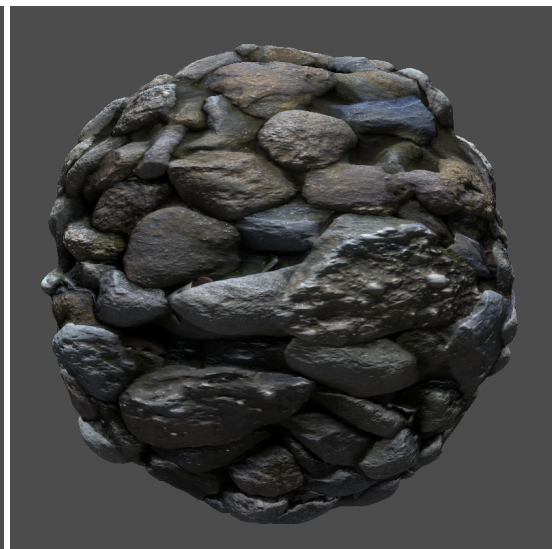
(a) Illumination de Phong sur une sphère



(b) Illumination avec de l'occlusion ambiante, utilisation de deux textures



(c) Modification de la géométrie suivant une carte de déplacement



(d) Modification des normales en tout point de l'objet

Figure 1: Images attendues pour chaque étape, jusqu'à obtenir de jolis cailloux mouillés