

Texture et transparence

Alexandre Meyer
Florence Zara
<http://licence-info.univ-lyon1.fr/LIFO41>

Introduction aux textures

- Objectif
 - Réalisme
 - Simplicité
- Image
 - Unique primitive
 - Complexe comme des millions
- Vocabulaire
 - Pixel de l'image (texture)
 = TEXEL pour TEXTure ELEMent

Cours de synthèse d'images 2

Textures : en TP

- Activation de la gestion des textures 2D


```
glEnable(GL_TEXTURE_2D);
```
- Création des textures
 - Donne n numéros de textures stockés dans texID

```
//glGenTextures(n, &texID);
m_quad_texture = read_texture(0, "data/papillon.png");
```
- Choix de la texture courante pour le rendu


```
//glBindTexture(GL_TEXTURE_2D, texID[0]);
gl.texture(m_quad_texture);
gl.model( Tquad );
gl.draw(m_quad);
```

Cours de synthèse d'images 3

Plaquage de la texture

- A chaque sommet de la face
 - Coordonnées textures (u,v)
 - (u,v) correspond à une position dans l'image (dans la texture)

Cours de synthèse d'images 4

Plaquage de la texture : en TP

- Spécification des coordonnées de textures pour chaque sommet


```
m_quad.normal( 0, 0, 1 );
m_quad.texcoord(0,0 );
m_quad.vertex(-1, -1, 0 );
```

Cours de synthèse d'images 5

Coordonnées de texture

- Les coordonnées de texture vont généralement de 0.0 à 1.0
- On peut mettre des coordonnées supérieures, si on souhaite répéter la texture (activer GL_REPEAT)

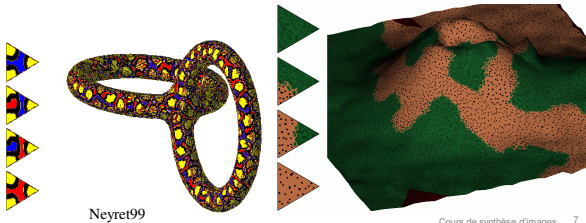
Valeurs entre 0.0 et 1.0

Valeurs entre 0.0 et 10.0 :
texture répétée 10 fois en longueur et largeur

Cours de synthèse d'images 6

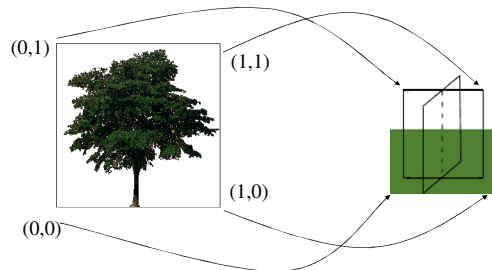
Répétition de texture

- Utile pour représenter des petits détails
 - crépi, routes, etc.
- Problème :
 - Textures répétitives : remarque la régularité
 - Solution : utiliser n textures avec des raccords continus et les combiner



Exemple d'utilisation : les billboards

- 1 quad toujours tourné face à la caméra
- Ou simplement en croiser 2 (en TP)

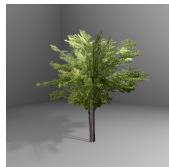


Plus généralement les arbres

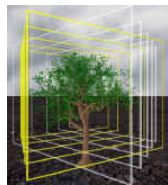


Nombres de triangles par arbre (>100)
et peu réaliste

16 couches de quad par arbre



2 quads par arbre



images 9

Plus généralement les arbres

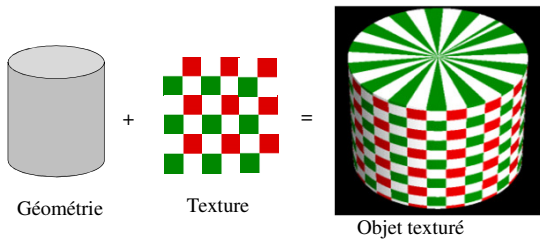
- Hierarchie de billboard
- 1 billboard par grande branche
 - 1 billboard par petite branche
 - 1 billboard par arbre



s de synthèse d'images 10

Problème de la paramétrisation

- Comment décider où chaque texel de la texture se projette sur la géométrie?



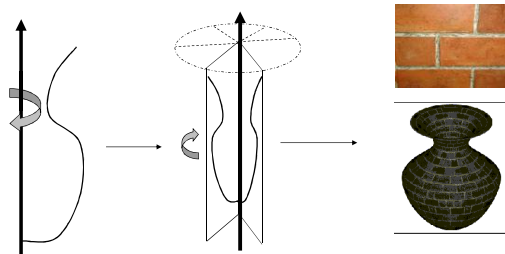
Géométrie

Texture

Objet texturé

Cours de synthèse d'images 11

Exemple du vase créé par révolution

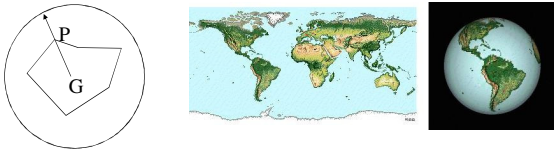


Exercice : définir (u,v) pour chaque sommet

Cours de synthèse d'images 12

Exemple de la sphère

- Projection sphérique
 - G = centre de gravité
 - Coordonnée angulaire du vecteur GP = (α, β) sont utilisées comme coordonnées de texture
- Très efficace pour les objets quasi-sphériques

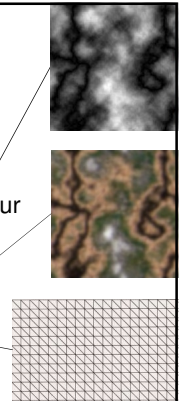
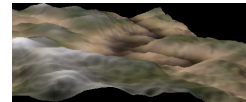


Exercice : définir (u,v) pour chaque sommet de la sphère

Cours de synthèse d'images 13

Exemple du terrain

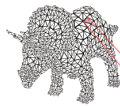
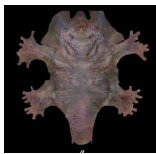
- Représentation d'un terrain
 - Carte de hauteur (niveau de gris)
 - Texture pour la couleur
 - Triangulation de la carte de hauteur



Exercice : définir (u,v) pour chaque sommet du terrain

Cours de synthèse d'images 14

Problème de la paramétrisation



For each triangle in the model establish a corresponding region in the phototexture



Un artiste peut créer une texture et paramétrer son modèle à la main

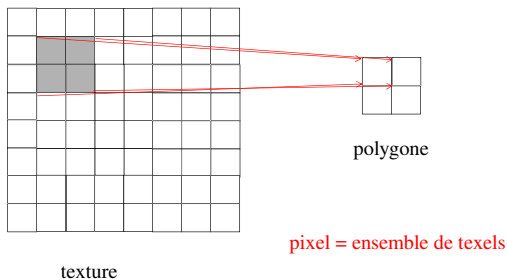
Cours de synthèse d'images 15

Correspondance pixel / texel

- Taille de la texture correspond rarement à la taille du polygone sur l'écran
- Si la taille texture > taille polygone sur écran
 - Réduction de la texture = **minification**
 - Pixel = ensemble de texels
 - `glTexParameter(..., GL_TEXTURE_MIN_FILTER, ...);`
- Si la taille texture < taille polygone sur écran
 - Agrandissement de la texture = **magnification**
 - Pixel = portion de texel
 - `glTexParameter(..., GL_TEXTURE_MAG_FILTER, ...);`

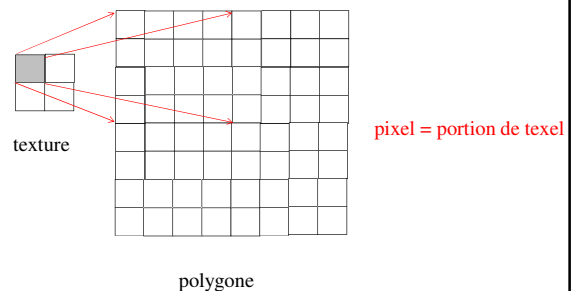
Cours de synthèse d'images 16

Filtre de texture : minification



Cours de synthèse d'images 17

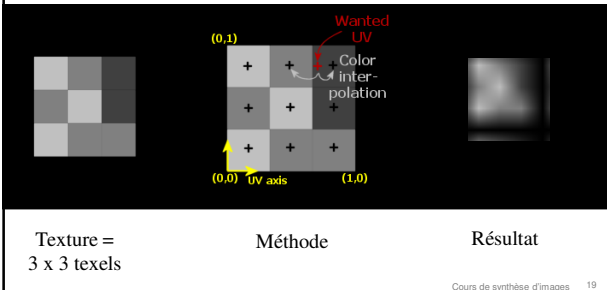
Filtre de texture : magnification



Cours de synthèse d'images 18

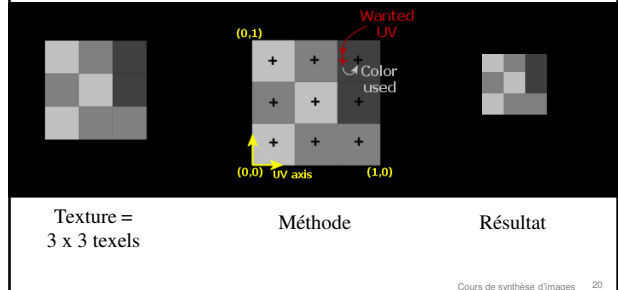
Valeurs possibles du filtre OpenGL

- `glTexParameter(..., ..., GL_LINEAR, ...)` : interpolation linéaire en faisant la moyenne des 2 texels les plus proches du centre du pixel



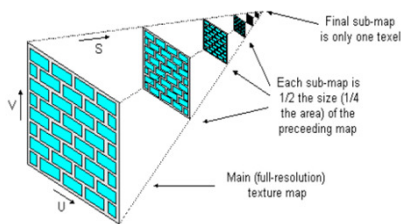
Valeurs possibles du filtre OpenGL

- `glTexParameter(..., ..., GL_NEAREST, ...)` : utilise le texel dont les coordonnées sont les plus proches du centre du pixel



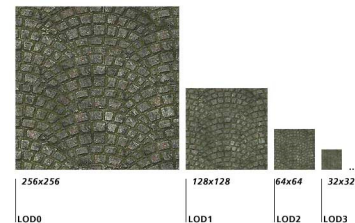
Mipmapping

- Contrôle du niveau de détail des textures avec les mipmaps
 - L'idée est de construire une pyramide d'image à différentes résolutions



Mipmapping

- Texture de taille initiale 256 x 256 avec différents niveaux de détail



Mipmapping

- OpenGL détermine l'image de texture à utiliser en fonction de la résolution du polygone
- Polygone proche de la caméra
 - gd nb de pixels sur écran = gde résolution
 - Utilisation de la texture de niveau de détail le + élevé
- Polygone loin de la caméra
 - Moins de pixels à l'écran
 - Utilisation de la texture de niveau de détail moins élevé

Cours de synthèse d'images 23

Mipmapping : OpenGL

- `glTexImage2d(..., level, ...)`
 - Texture normale
 - Level = 0
 - Mipmapping :
 - Level = 0 : texture de + gdr résolution
 - Level = 1 : résolution divisée par 2
 - Level = n : résolution de la texture (n-1) divisée par 2
 - Il faut spécifier les textures jusqu'à la résolution 1x1
 - Si texture initiale 128x128 (level 0), il faut définir 64x64 (level1), 32x32 (level 2), ..., 1x1 (level 7)

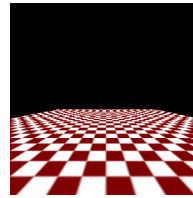
Cours de synthèse d'images 24

Génération des niveaux de détail

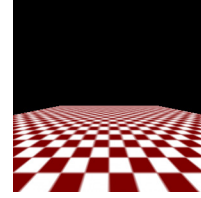
- Génération manuel
 - On fournit les textures des différentes résolutions avec `glTexImage2d()`
- Génération automatique
 - On fournit la texture de plus grande résolution avec `gluBuild2DMipmaps()`

Cours de synthèse d'images 25

Mipmapping : exemple



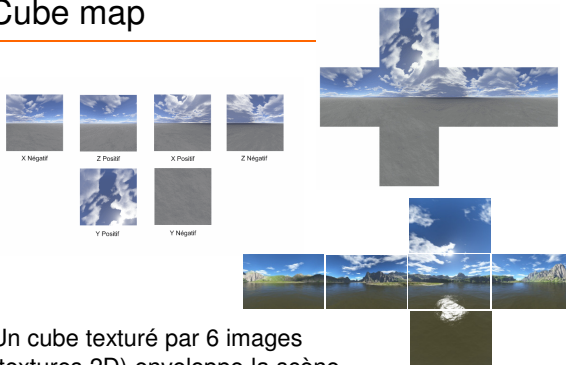
Sans mipmapping



Avec mipmapping

Cours de synthèse d'images 26

Cube map



- Un cube texturé par 6 images (textures 2D) enveloppe la scène
 - Impression d'espace

Cours de synthèse d'images 27

Cube map

- Assigne une valeur (couleur) pour une direction donnée depuis la position de la caméra
 - Coordonnées de textures dépendent de la position de la caméra
 - OpenGL : mode automatique de calcul des coordonnées de textures

Cours de synthèse d'images 28

Cube map

- Permet de visualiser les reflets de l'environnement sur l'objet

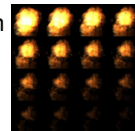


Cours de synthèse d'images 29

Texture animée



- N images côte à côte : effet d'animation
 - Feu, personnage marche, etc.
 - Mouvement doit être cyclique
- Changer coord texture (u,v) en fonction du temps
 - `glTexCoord(U(t), V(t))` avec t le temps



Exercice : coordonnées UV pour afficher le feu ci-dessus

Cours de synthèse d'images 30

Où trouver des textures ?

- Photographie
- Artistes
- Bibliothèque de textures sur le Web
- Techniques de génération automatique (non abordées en L2)

Cours de synthèse d'images 31

Pour résumer

- Texture
 - Outil très puissant
 - Simple
 - Réaliste
- Egalement de nombreux problèmes
 - Paramétrisation
 - Génération de textures
 - Proche : ne remplace pas la géométrie

Cours de synthèse d'images 32

Transparence

33

Opacité ou transparence

- Transparence pour la couleur d'un polygone ou les pixels d'une texture
- RGB → RGBA
 - Paramètre A (alpha) = opacité
 - 0.0 = transparent
 - ... = semi-opaque
 - 1.0 = opaque
- Effet de transparence pour simuler les vitres, l'eau, etc.

Cours de synthèse d'images 34

Transparence

- Formule
 - Calcul pendant la rasterization
 - Fusion (*blending*) des valeurs chromatiques sources et des valeurs chromatiques de destination
 - $C_buffer = A \times C_polygone + (1-A) \times C_buffer$
- `glEnable(GL_BLENDING)`

Cours de synthèse d'images 35

Transparence

- Exemple les billboards
 - Les pixels blancs doivent être transparents



Cours de synthèse d'images 36

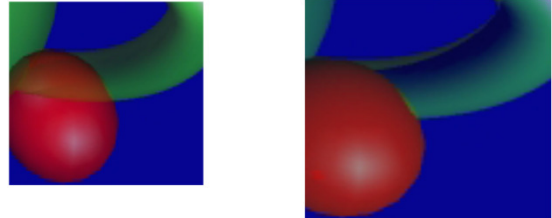
Transparence

- $C_buffer = A \times C_polygone + (1-A) \times C_buffer$
- Attention l'ordre a de l'importance
 - Car le Z-buffer ne stocke qu'une seule valeur
 - Si objet semi-transparent
 - problème si l'ordre des objets n'est pas de derrière à devant
 - Si objet tout transparent ou tout opaque
 - pas de problème

Cours de synthèse d'images 37

Transparence

- Comment faire pour dessiner 2 objets semi-transparent l'un derrière l'autre ?
 - Il faut trier pour afficher les objets de derrière à devant



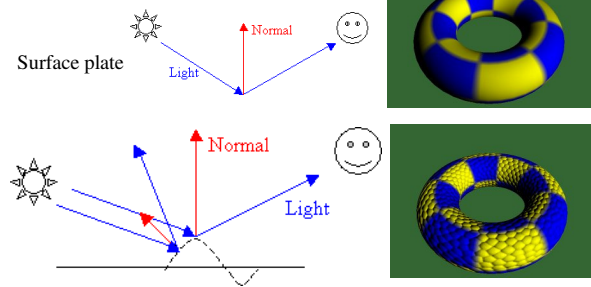
Cours de synthèse d'images 38

Textures Bonus

39

Bump Maps

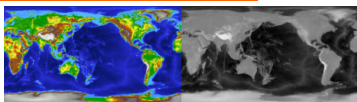
- Utilisation d'une carte de normales pour donner une impression de relief



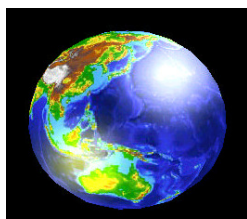
Cours de synthèse d'images 40

Bump Maps

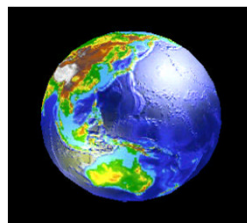
Texture des normales



Texture des couleurs



Rendu "classique"

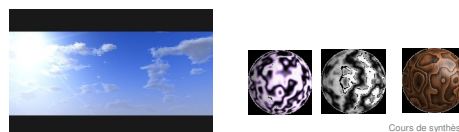


Rendu avec bump-mapping

Cours de synthèse d'images 41

Génération automatique de textures

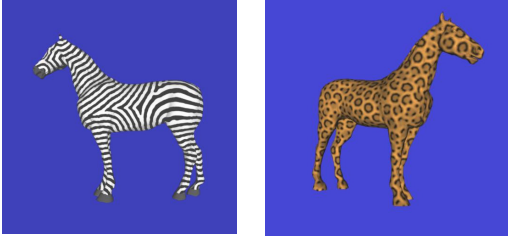
- Random : bruit de Perlin
 - Valeurs de pixel tirées au hasard
 - Interpolation entre ces pixels
 - Ceci à différente résolution
 - Combinaison



Cours de synthèse d'images 42

Génération automatique de textures

- Reaction-Diffusion : simulation des interactions entre cellules lors de la croissance



Cours de synthèse d'images 43

Génération automatique de textures

- Cellular Texture Generation (Fleischer '95)



Cours de synthèse d'images 44