Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Moteurs Temps Réel
Textures et complexité visuelle

Guillaume Gilet
Guillaume.Gilet@unilim.fr

2012-2013

master ISICG limoges

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Réalisme et complexité

Réalisme et complexité

Réalisme

- Apparence très complexe
- Détails à plusieurs échelles
- Peu de surfaces totalement lisses
- Peu de couleur unie

Complexité en représentation discrète

- Des attributs par sommet
- Echantillonnage fin pour la géométrie ?
- Echantillonnage fin pour les attributs/détails ?

Réalisme et complexité

Réalisme et complexité
Principe
Interpolation et filtrage
Un peu plus loin
En pratique

Réalisme

- Apparence très complexe
- Détails à plusieurs échelles
- Peu de surfaces totalement lisses
- Peu de couleur unie

Complexité en représentation discrète

- Des attributs par sommet
- Echantillonnage fin pour la géométrie ?
- Echantillonnage fin pour les attributs/détails ?

Représentation géométrique

- Coût du maillage polygonal(stockage, calcul...)

Réalisme et complexité

Réalisme

- Apparence très complexe
- Détails à plusieurs échelles
- Peu de surfaces totalement lisses
- Peu de couleur unie

Complexité en représentation discrète

- Des attributs par sommet
- Echantillonnage fin pour la géométrie ?
- Echantillonnage fin pour les attributs/détails ?

Représentation géométrique

- Coût du maillage polygonal(stockage, calcul...)

Idée : Représenter de l'information **entre** les sommets

Informations entre les sommets

Différentes informations

- Couleur
- Paramètres de BRDF
- Normale
- Interaction avec la scène (reflets, illumination...)
- Modification de géométrie
- Ce qu'on veut...

Informations entre les sommets

Représentation des informations

Informations entre les sommets

Représentation des informations

- Par une fonction
  - Compact
  - Paramétrable
  - Très difficile à créer pour un effet
  - Très difficile à simuler (modèles physiques)
  - Complexe à calculer

Informations entre les sommets

Représentation des informations

- Par une fonction
  - Compact
  - Paramétrable
  - Très difficile à créer pour un effet
  - Très difficile à simuler (modèles physiques)
  - Complexe à calculer
- Par des données discrètes (tableau de données)
  - Facile à utiliser
  - Assez facile à obtenir (mesures, photographies...)
  - Stockage des données

Informations entre les sommets

Représentation des informations

- Par une fonction
  - Compact
  - Paramétrable
  - Très difficile à créer pour un effet
  - Très difficile à simuler (modèles physiques)
  - Complexe à calculer
- Par des données discrètes (tableau de données)
  - Facile à utiliser
  - Assez facile à obtenir (mesures, photographies...)
  - Stockage des données

En synthèse d'image

- Des textures
- 1D, 2D, 3D, 3D+t...
- Notion de **texel** (*texture element*)

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Textures
Principe

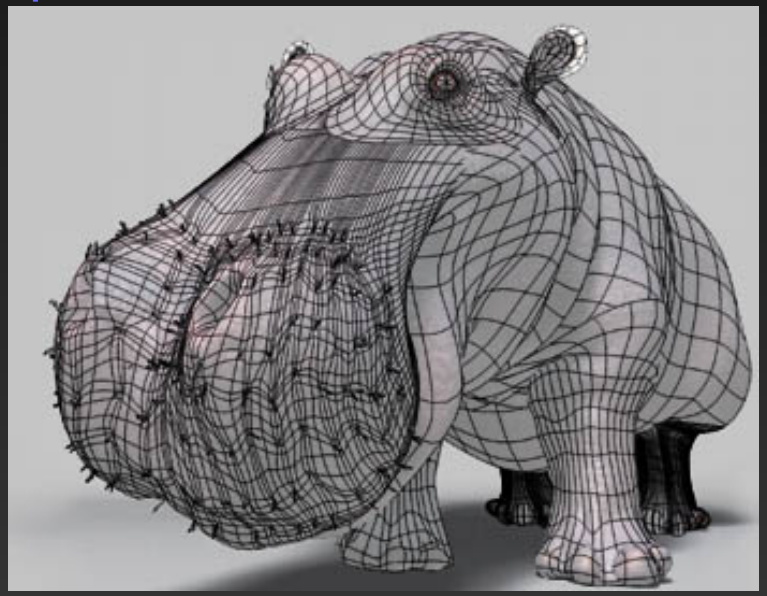
Textures

Utilisation dans un moteur 3D

- Souvent des images 2D
- Informations de couleur
- Apparence fine des surfaces à base d'images

Principe

- Géométrie simple



master ISICG limoges

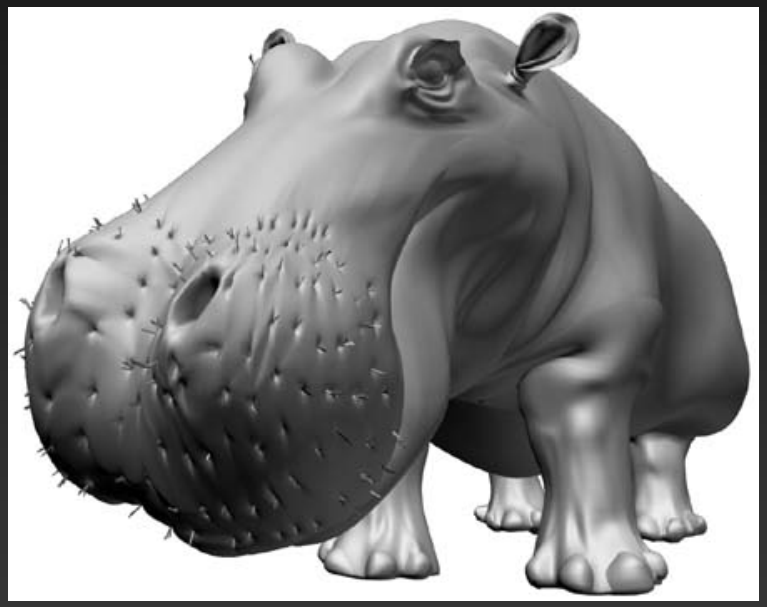
Textures

Utilisation dans un moteur 3D

- Souvent des images 2D
- Informations de couleur
- Apparence fine des surfaces à base d'images

Principe

- Géométrie simple



Textures

Utilisation dans un moteur 3D

- Souvent des images 2D
- Informations de couleur
- Apparence fine des surfaces à base d'images

Principe

- Géométrie simple
- *Habillage* de la géométrie

Textures

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Utilisation dans un moteur 3D

- Souvent des images 2D
- Informations de couleur
- Apparence fine des surfaces à base d'images

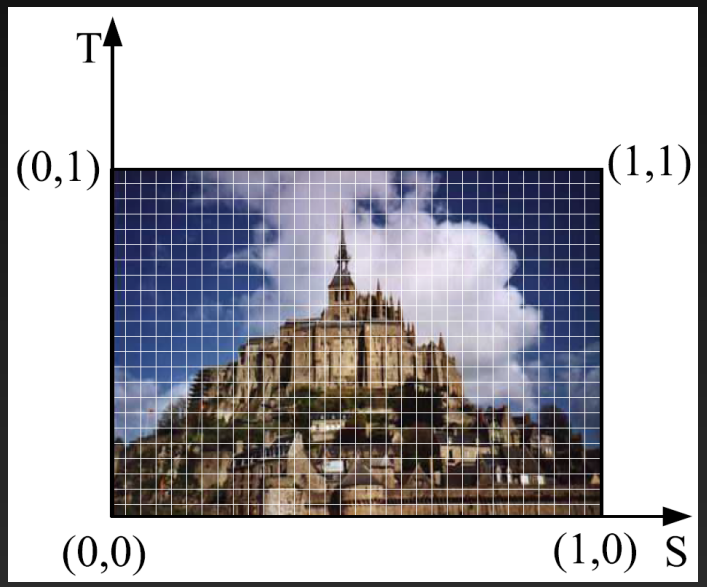
Principe

- Géométrie simple
- *Habillage* de la géométrie
- Placage de texture

Texture

Représentation (2D)

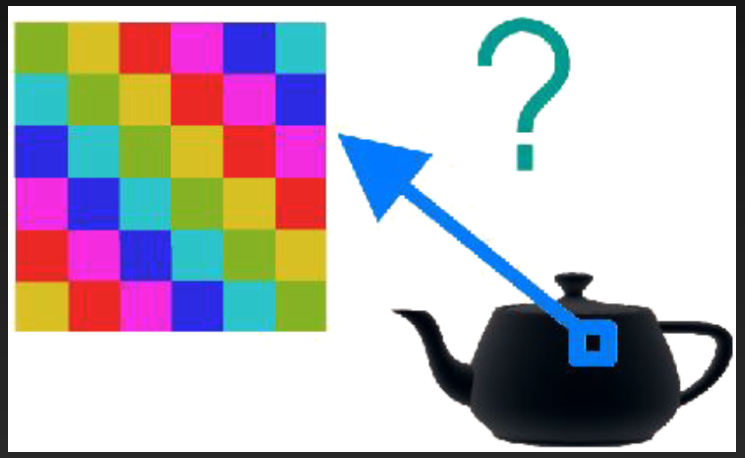
- Une texture est une « image », c'est à dire une grille de texels.
- Chaque texel est localisé dans la texture par ses coordonnées s et t.
- Toute l'image de la texture est décrite par $s \in [0, 1]$ et $t \in [0, 1]$



Placage de texture

Problématique

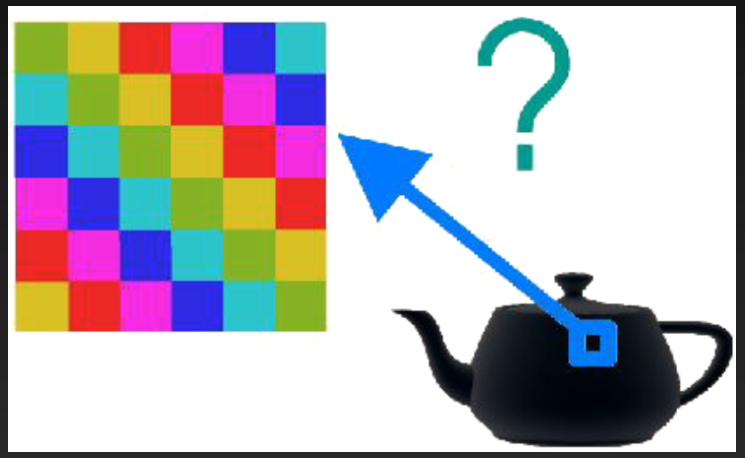
- Consiste à associer à chaque point 3D un texel (donne au point sa couleur).
- L'association se fait simplement en indiquant (ou en calculant) les coordonnées (s, t) du texel voulu



Placage de texture

Problématique

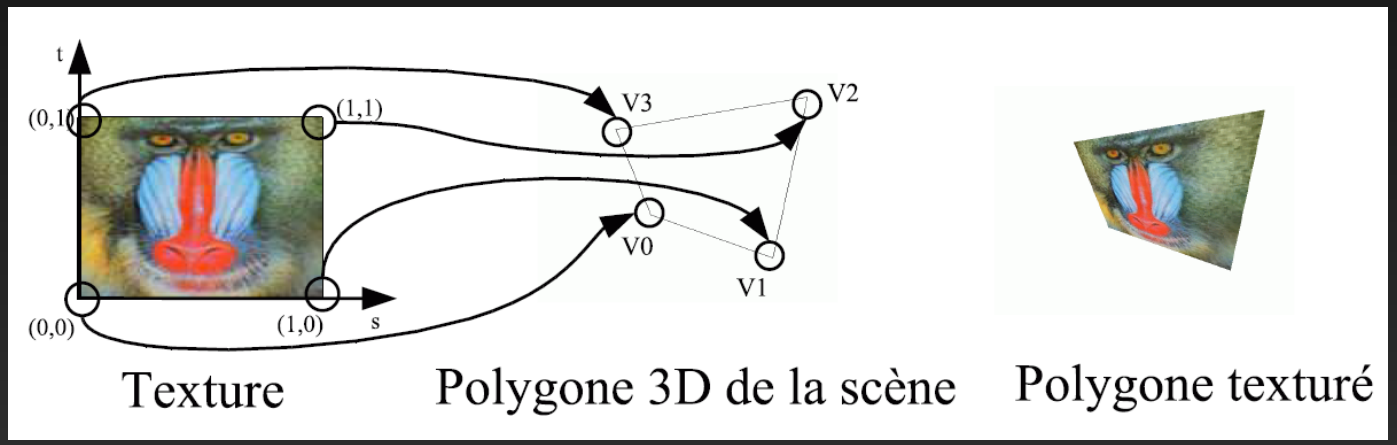
- Consiste à associer à chaque point 3D un texel (donne au point sa couleur).
- L'association se fait simplement en indiquant (ou en calculant) les coordonnées (s, t) du texel voulu
- Transformation de $\Re^3 \longmapsto \Re^2$



Placage de texture

Pour simplifier

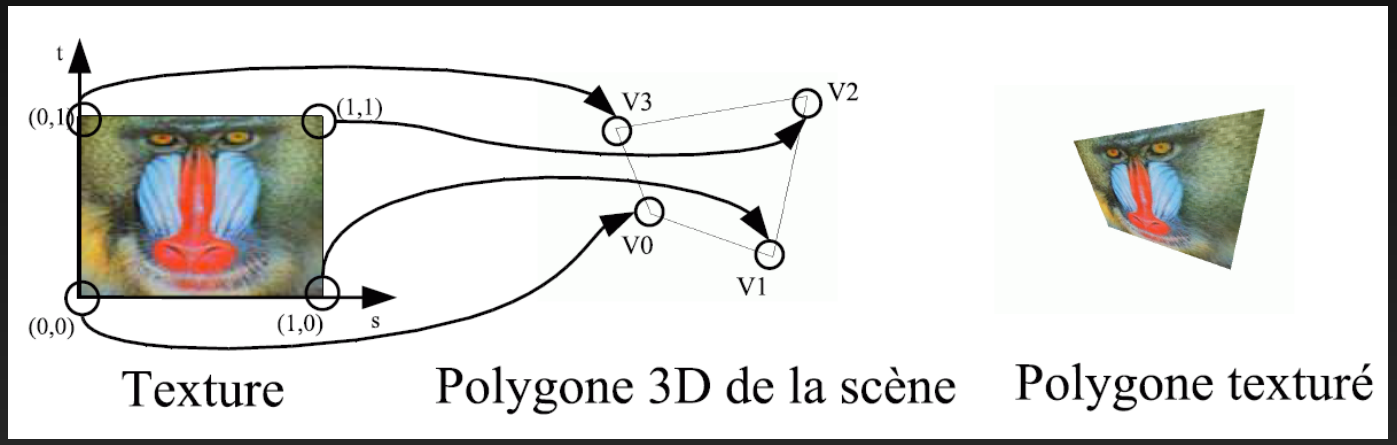
- Association uniquement aux sommets
- Interpolation des coordonnées de textures



Placage de texture

Pour simplifier

- Association uniquement aux sommets
- Interpolation des coordonnées de textures



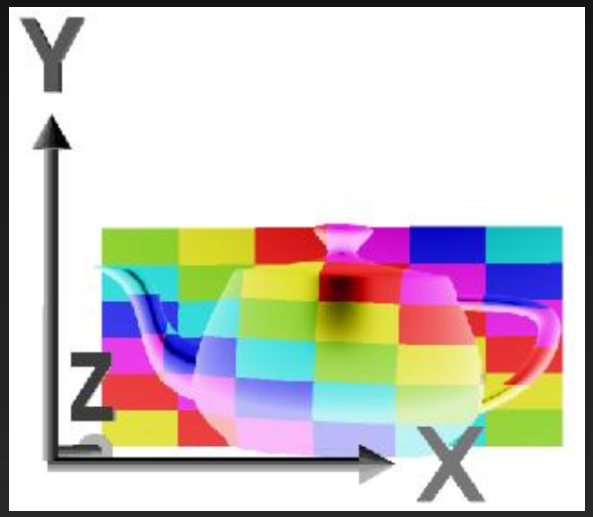
Comment calculer les coordonnées ?

- Calculer automatiquement ces coordonnées ?
  - Problème difficile pour des objets quelconques
- Spécification à la main des coordonnées ?

Placage de texture

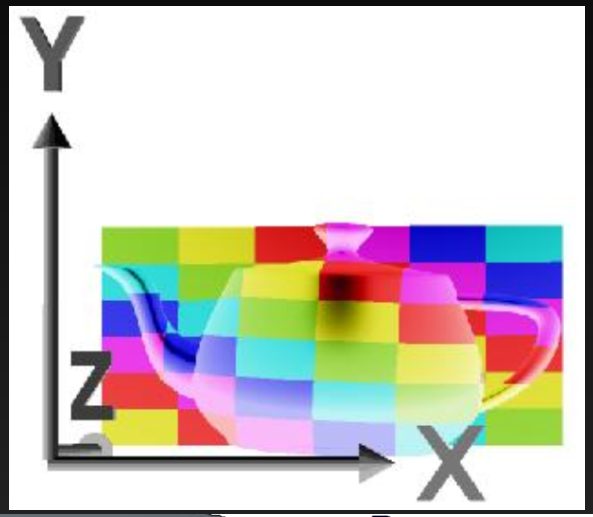
Habillage Plan

- Projection plane
- Choix du plan de projection

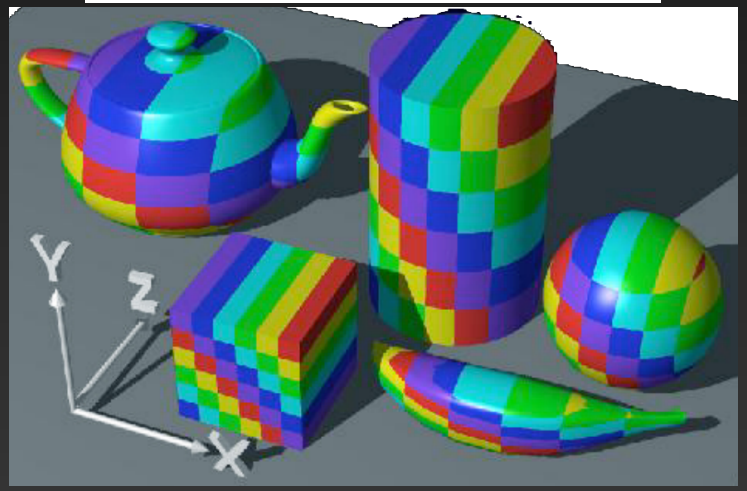


Placage de texture

Habillage Plan



- Projection plane
- Choix du plan de projection



Placage de texture

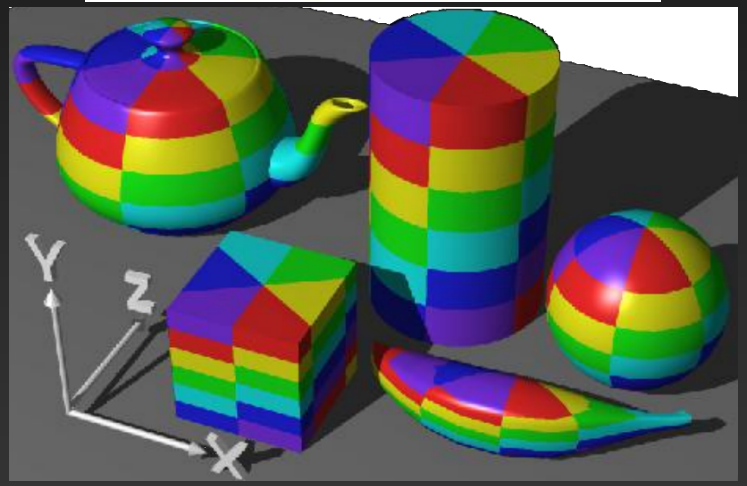
Habillage cylindrique

- Projection sur un cylindre
- Cylindre englobant orienté
- Paramétrisation (r, θ, h)
- $(\theta, h) \longmapsto (u, v)$

Placage de texture

Habillage cylindrique

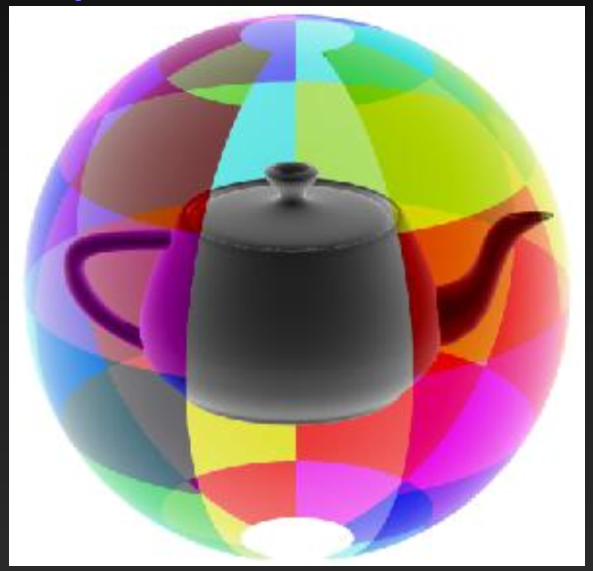
- Projection sur un cylindre
- Cylindre englobant orienté
- Paramétrisation (r, θ, h)
- $(\theta, h) \longmapsto (u, v)$



Placage de texture

Habillage sphérique

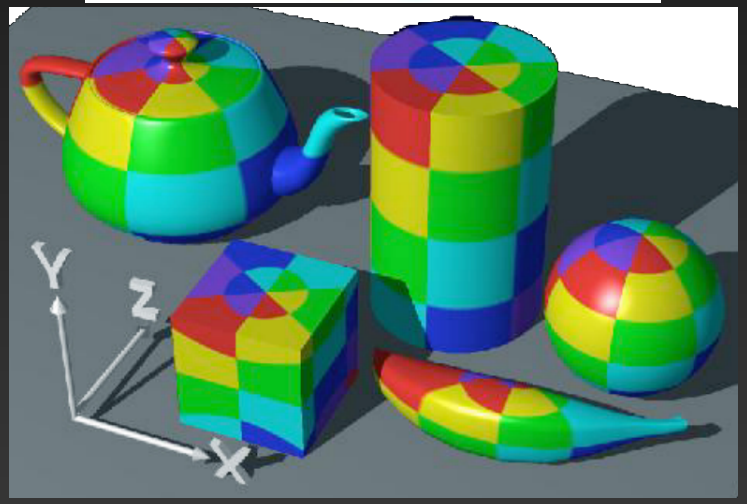
- Projection sur une sphère englobante
- Paramétrisation (r, θ, ϕ)
- $(\theta, \phi) \longmapsto (u, v)$



Placage de texture

Habillage sphérique

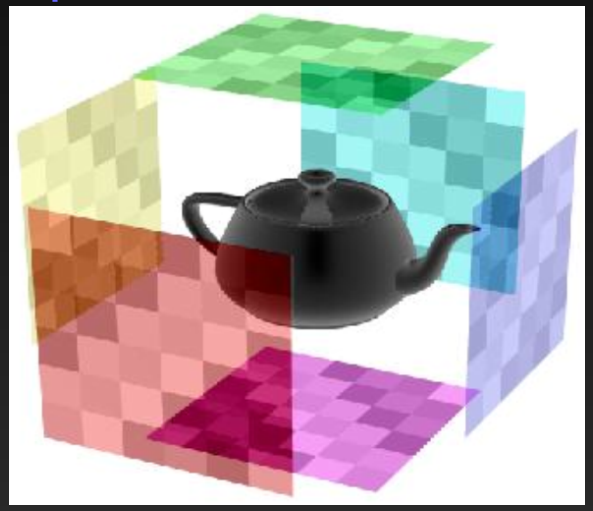
- Projection sur une sphère englobante
- Paramétrisation (r, θ, ϕ)
- $(\theta, \phi) \longmapsto (u, v)$



Placage de texture

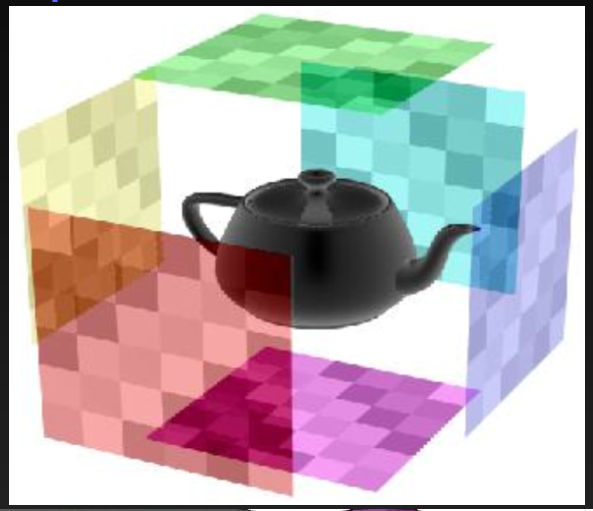
Habillage cubique

- Projection sur un cube englobant
- Paramétrisation plane par face

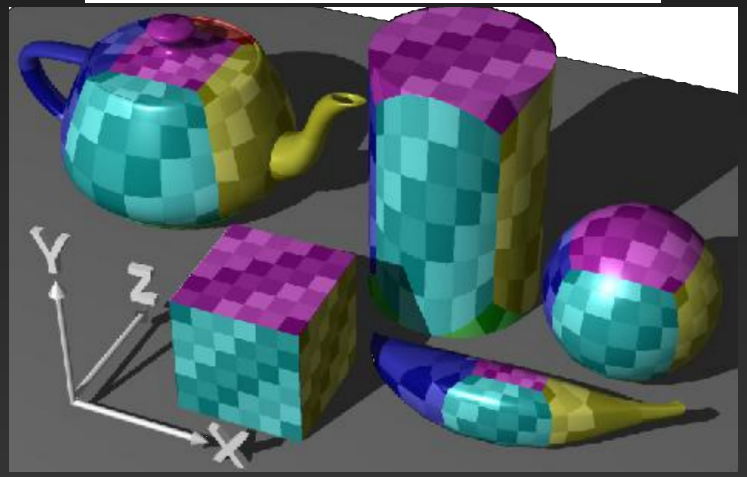


Placage de texture

Habillage cubique



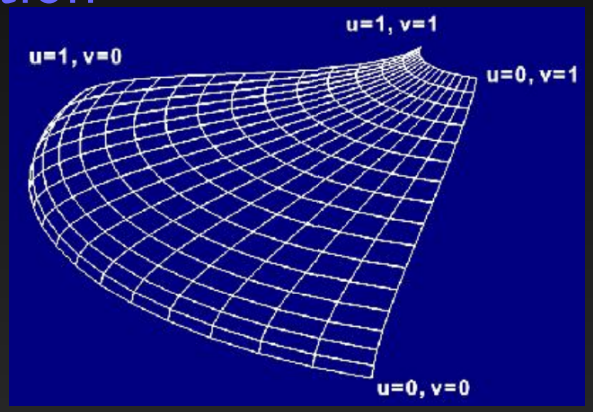
- Projection sur un cube englobant
- Paramétrisation plane par face



Placage de texture

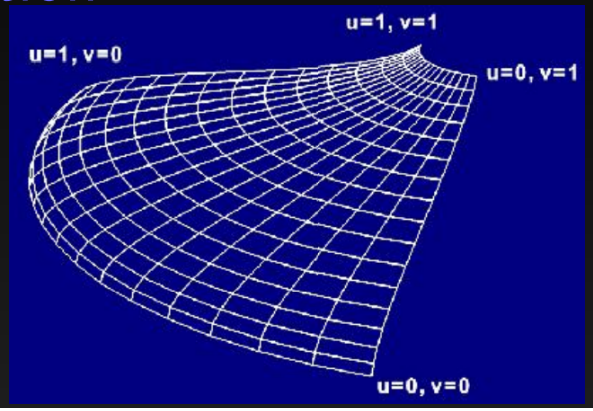
Paramétrisation

- Paramétrisation du maillage
- Représentation explicite de la projection
- Coordonnées de texture aux sommets du maillage

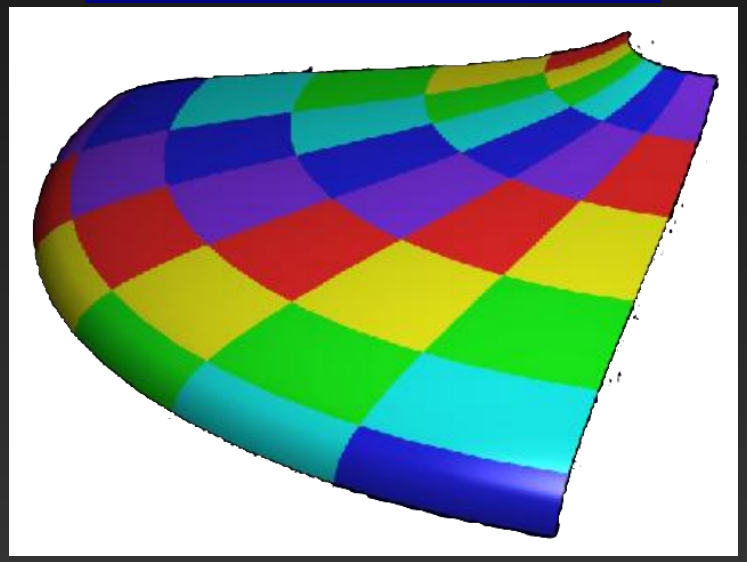


Placage de texture

Paramétrisation



- Paramétrisation du maillage
- Représentation explicite de la projection
- Coordonnées de texture aux sommets du maillage



Modélisation

Réalisme et complexité

Principe

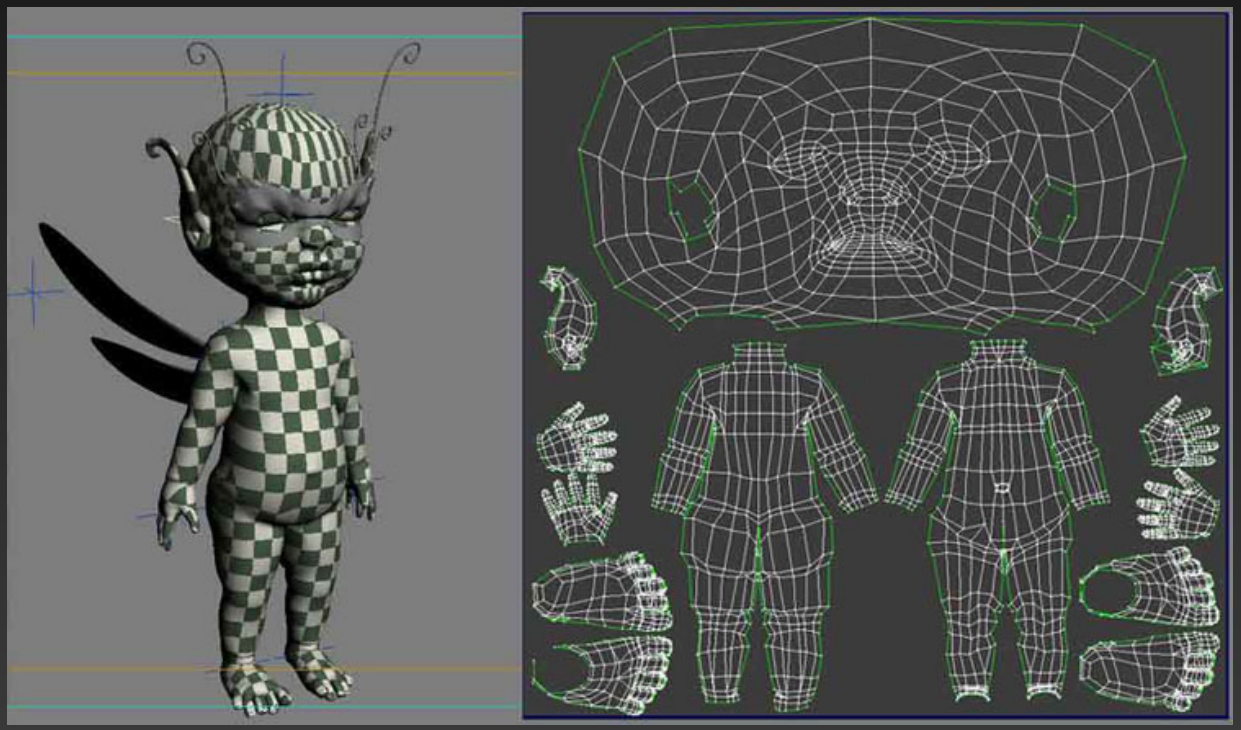
Interpolation et filtrage

Un peu plus loin

En pratique

Paramétrisation du maillage

- Méthode prépondérante dans l'industrie (semi-automatique)
- *Découpage* d'un modèle
- Mise à plat et distorsion
- Problème encore difficile selon les objets
- Correspondance image(photo ?) / texture



Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Textures
Interpolation et filtrage

master ISICG limoges

Interpolation et filtrage

Problème d'aliassage

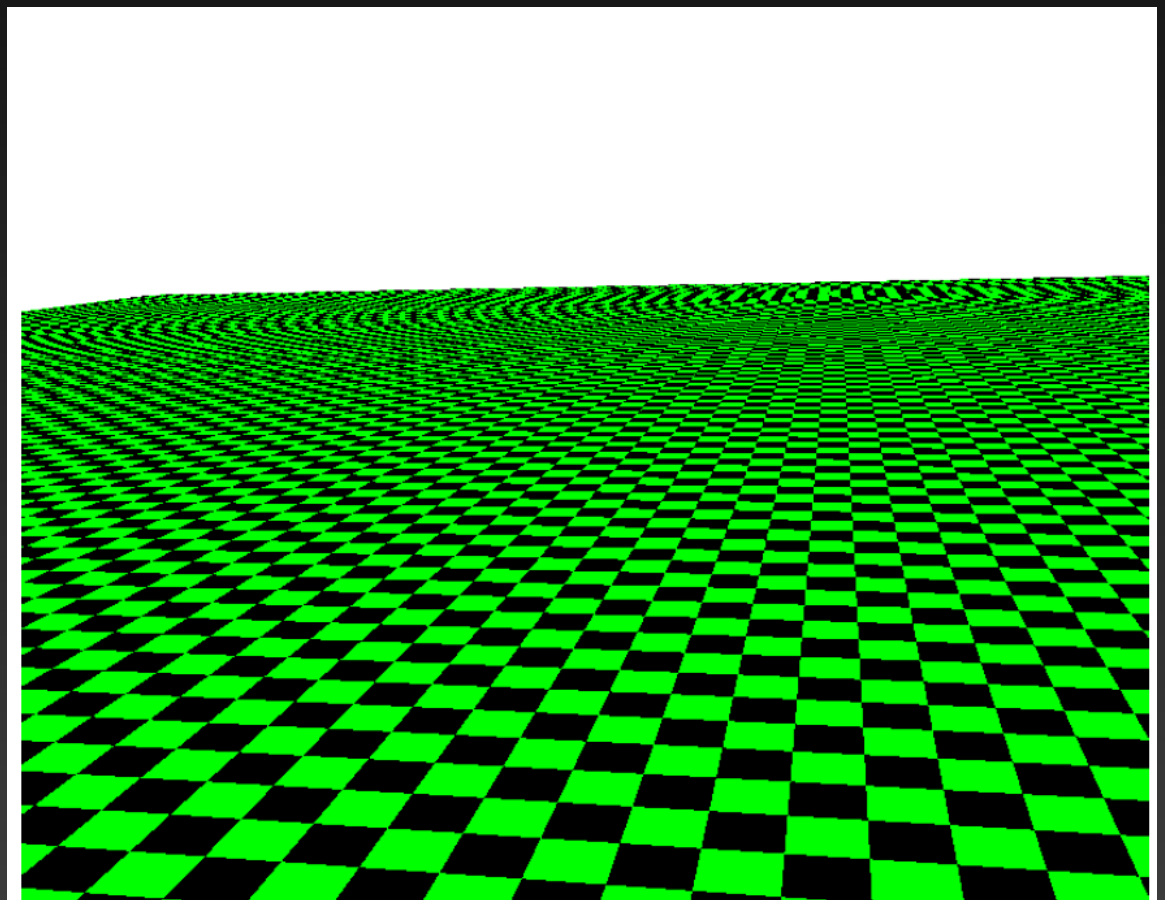
- Problème de correspondance texel/pixel

master ISICG limoges

Interpolation et filtrage

Problème d'aliassage

- Problème de correspondance texel/pixel
- 1 pixel = plusieurs texels (magnification) = perte d'information.



master ISICG limoges

Interpolation et filtrage

Problème d'aliassage

- Problème de correspondance texel/pixel
- 1 pixel = plusieurs texels (magnification) = perte d'information.
- 1 texel =plusieurs pixels (minification) = crénelage

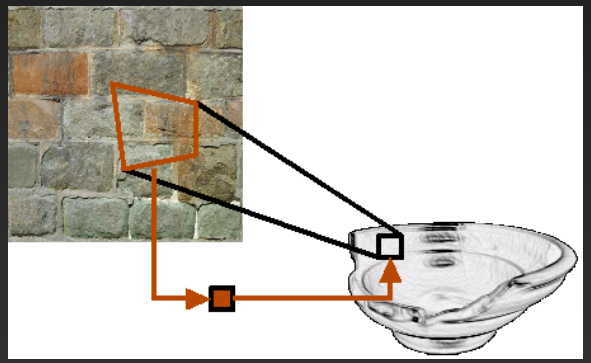
Interpolation et filtrage

Problème d'aliassage

- Problème de correspondance texel/pixel
- 1 pixel = plusieurs texels (magnification) = perte d'information.
- 1 texel =plusieurs pixels (minification) = crénelage

Dans l'idéal

- Projeter le pixel dans l'espace texture et intégrer
- Couteux...



Comportement des textures

Réalisme et complexité

Principe

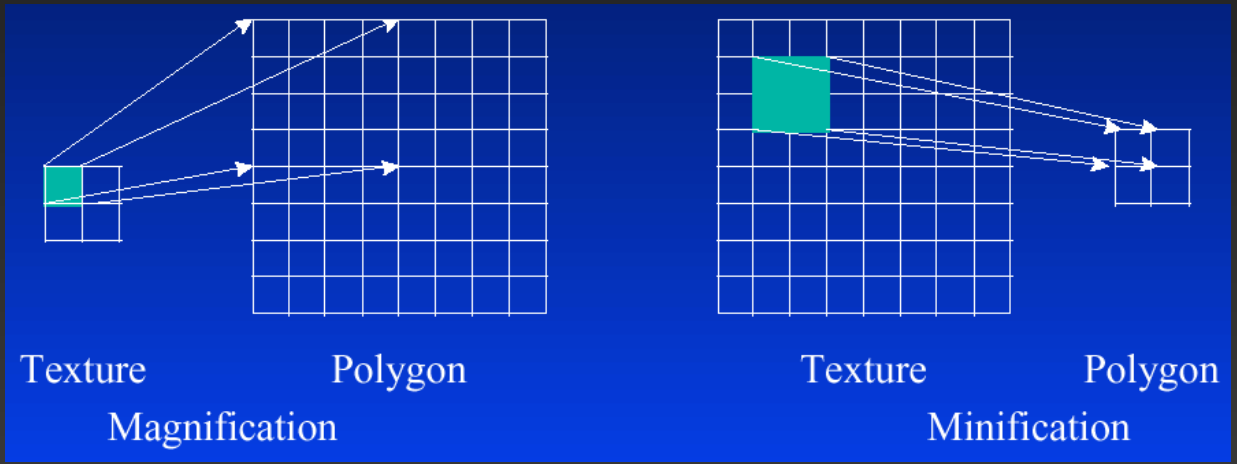
Interpolation et filtrage

Un peu plus loin

En pratique

Minification et magnification

- Deux cas possibles
- glTexParameteri(GLenum target,GLenum pname,GLenum param)
  - pname : type de filtrage
  - GL_TEXTURE_MAG_FILTER (agrandissement)
  - GL_TEXTURE_MIN_FILTER (réduction)
  - param : fonction de filtrage
  - GL_LINEAR (interpolation linéaire)
  - GL_NEAREST (le plus proche)



master ISICG limoges

Mip-Mapping

Mip-mapping

- Stocker plusieurs résolutions d'une même texture
- Changer de résolution de texture selon le nombre de texels qui correspondent à un pixel.
- Permet de nuancer les pertes d'informations (filtre d'aliassage) si les différentes résolutions sont construites par moyennage.

Mip-Mapping : Comparaison

Sans Mip-Mapping

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

master ISICG limoges

Mip-Mapping : Comparaison

Avec Mip-Mapping

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

master ISICG limoges

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Textures
Un peu plus loin

Light Map

Utiliser la texture pour représenter la lumière

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Scene

source : http ://www.ipcode.net/

Light Map

Utiliser la texture pour représenter la lumière

Scene + eclairage GL

source : http ://www.ipcode.net/

Réalisme et complexité

Principe

Interpolation et filtrage

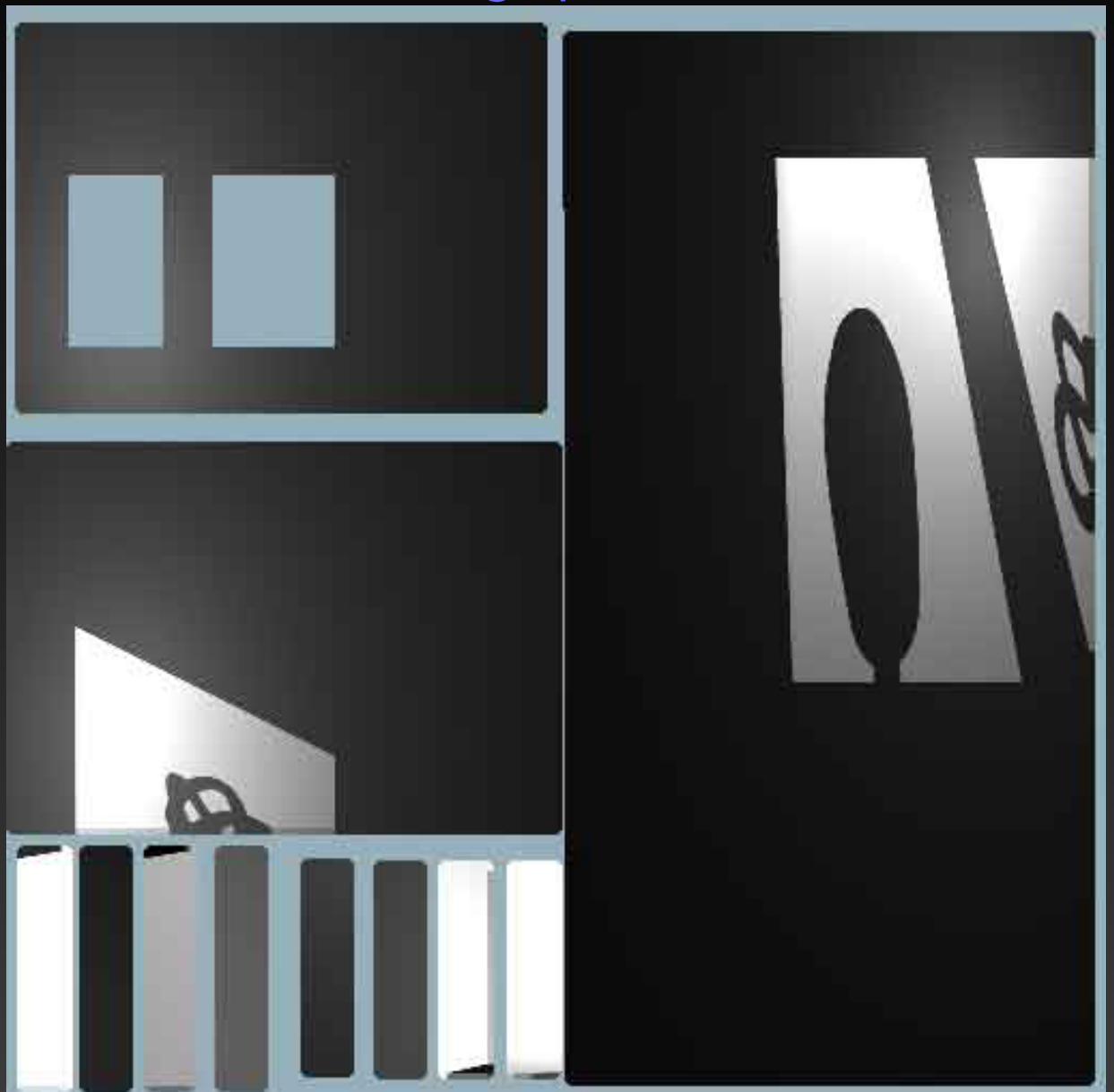
Un peu plus loin

En pratique

Light Map

Utiliser la texture pour représenter la lumière

Eclairage precalculé



master ISICG limoges

Light Map

Utiliser la texture pour représenter la lumière

Scene + texture (lightMap)

source : http ://www.ipcode.net/

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Light Map

Utiliser la texture pour représenter la lumière

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

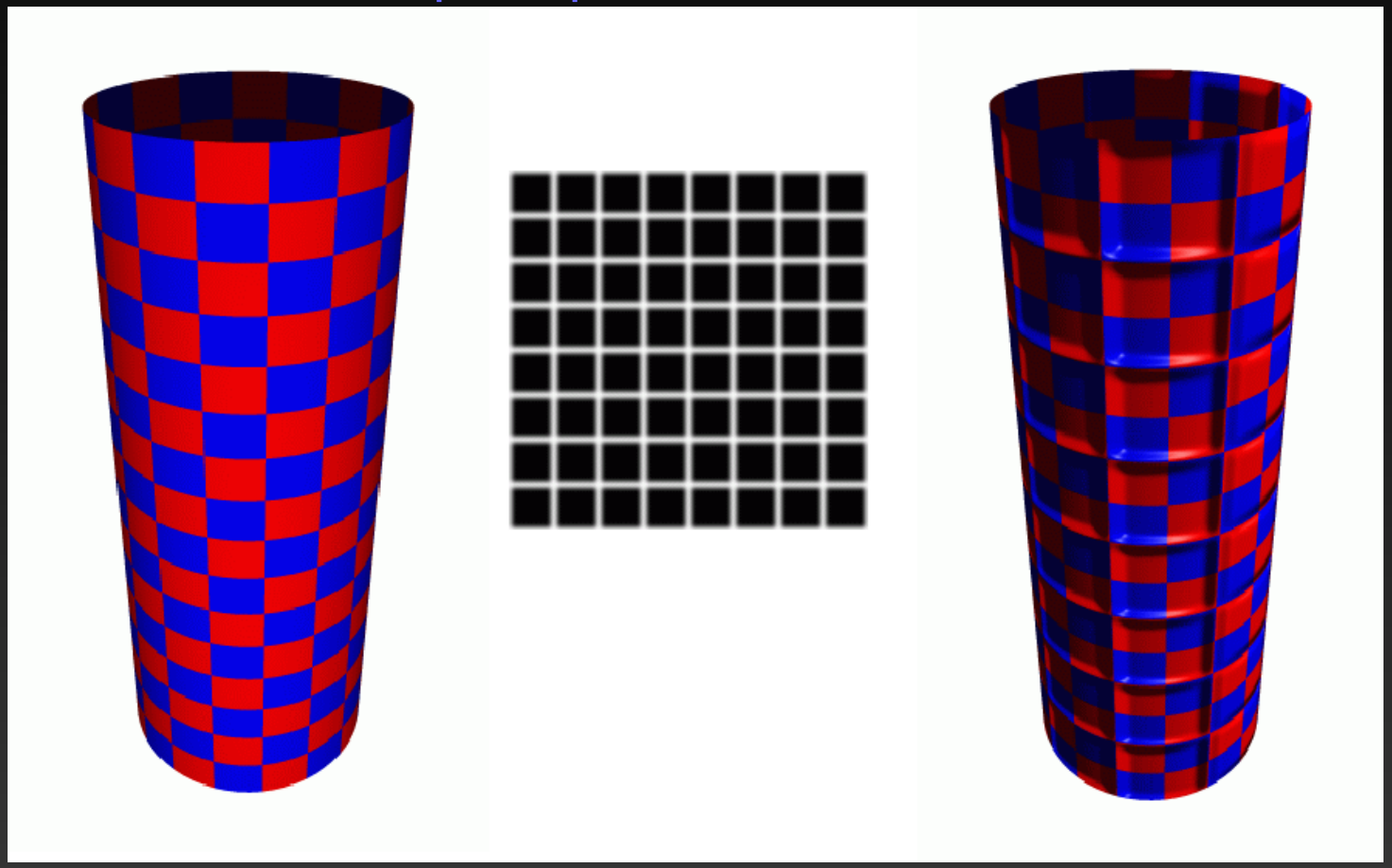
Scene finale

source : http ://www.ipcode.net/

Bump Mapping

Principe

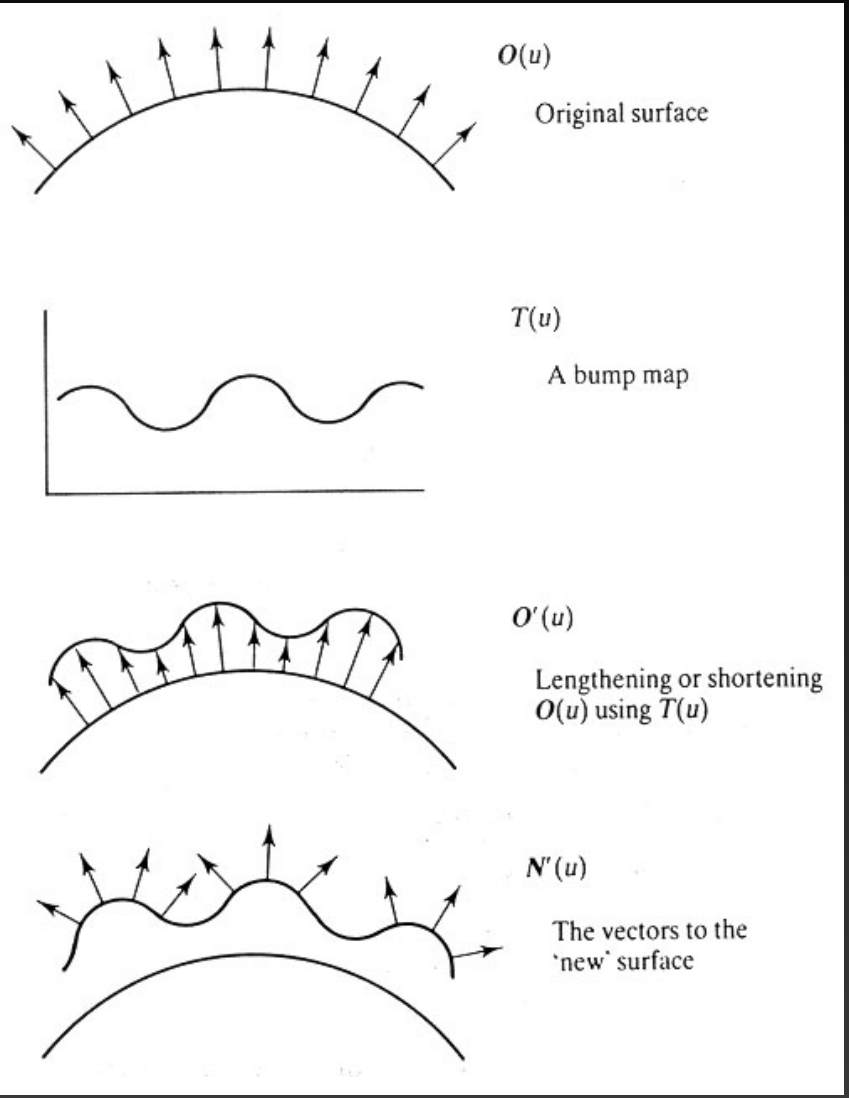
Utiliser la texture pour perturber la normale



Normal Mapping

Principe

Utiliser la texture pour remplacer la normale



Normal Mapping

Réalisme et complexité

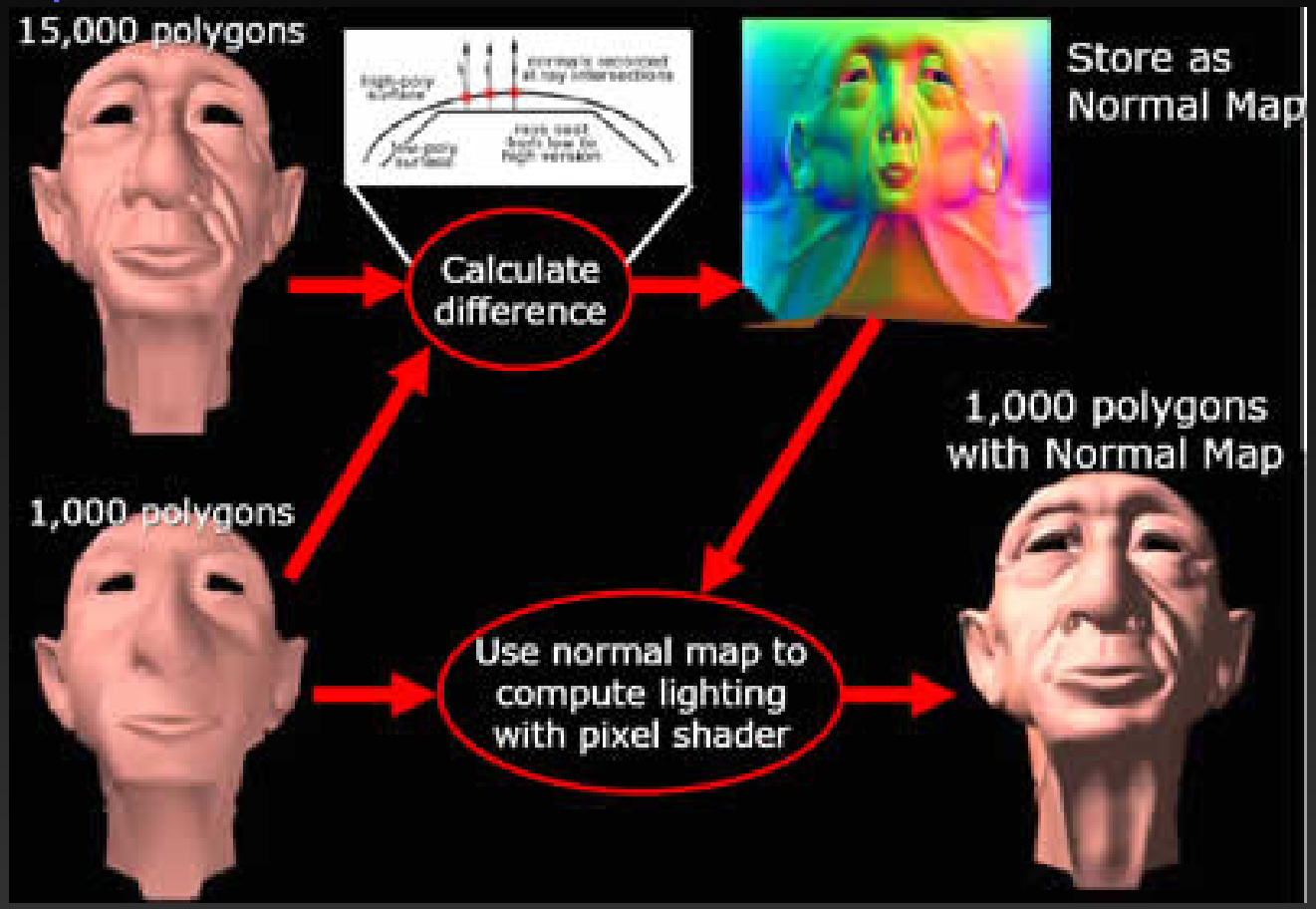
Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Principe de création



Normal Mapping

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

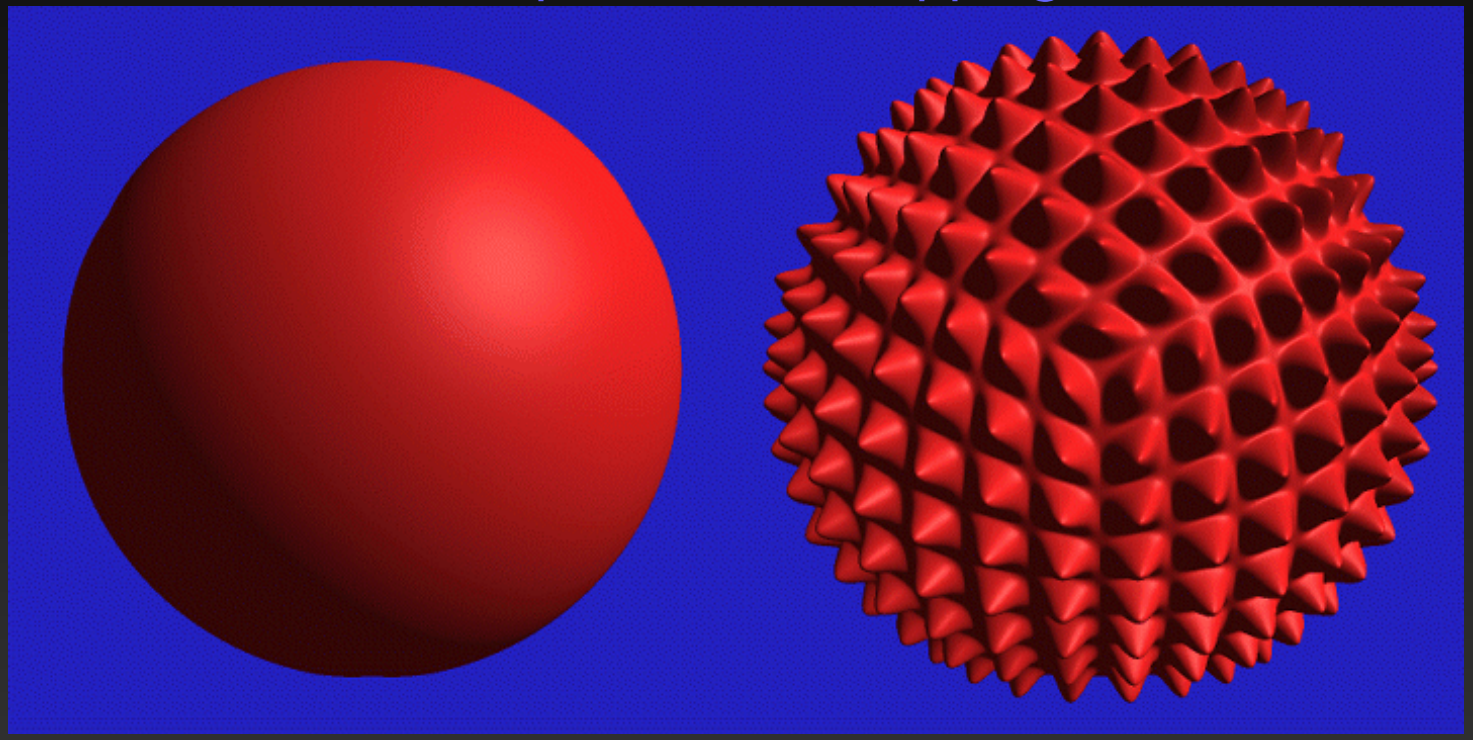
En pratique

Utilisation Intensive - Jeux Vidéos

Displacement Map

Utiliser la texture pour perturber/déplacer la géométrie

Displacement Mapping



Displacement Map

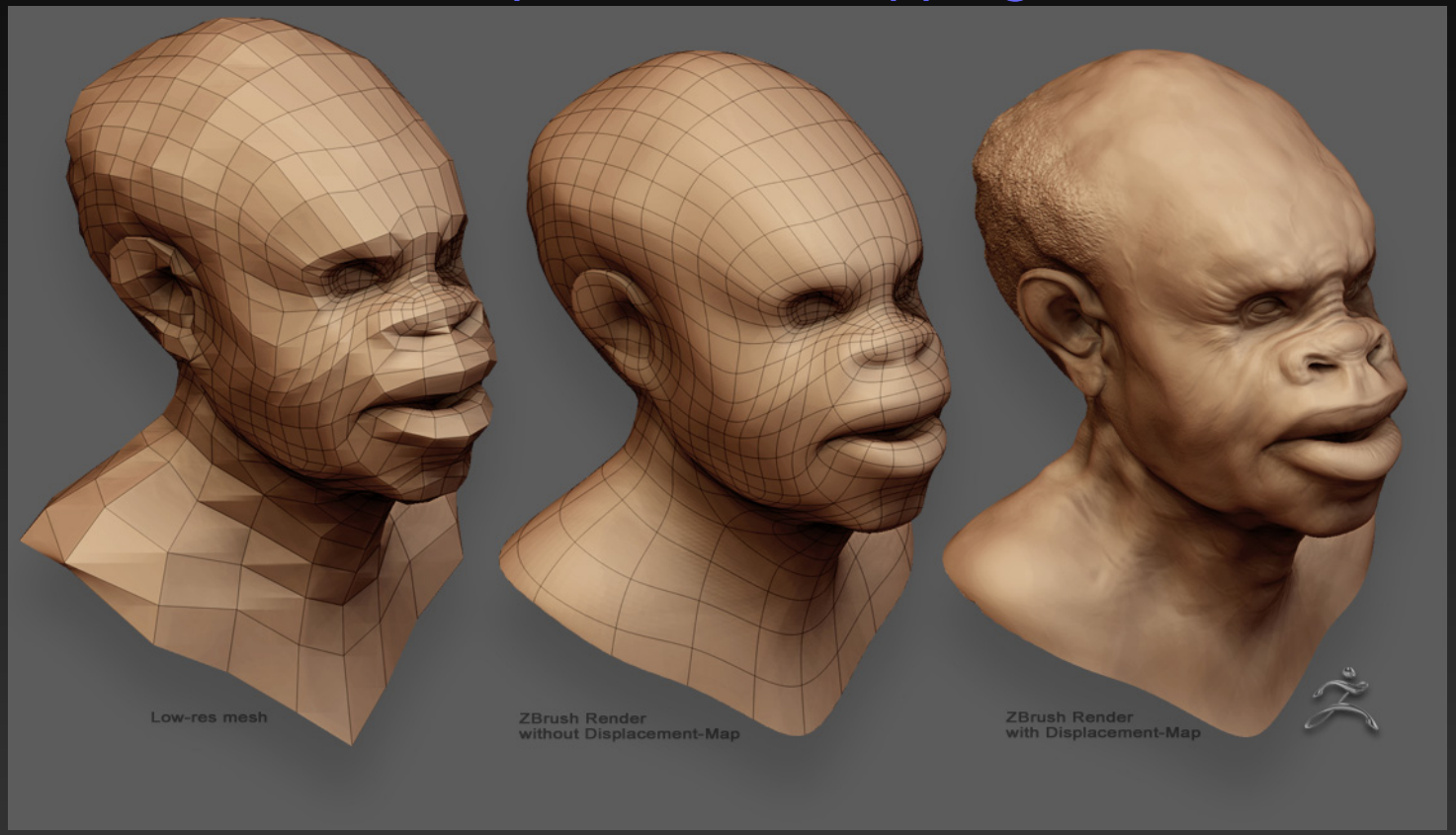
Utiliser la texture pour perturber/déplacer la géométrie

Displacement Mapping

Displacement Map

Utiliser la texture pour perturber/déplacer la géométrie

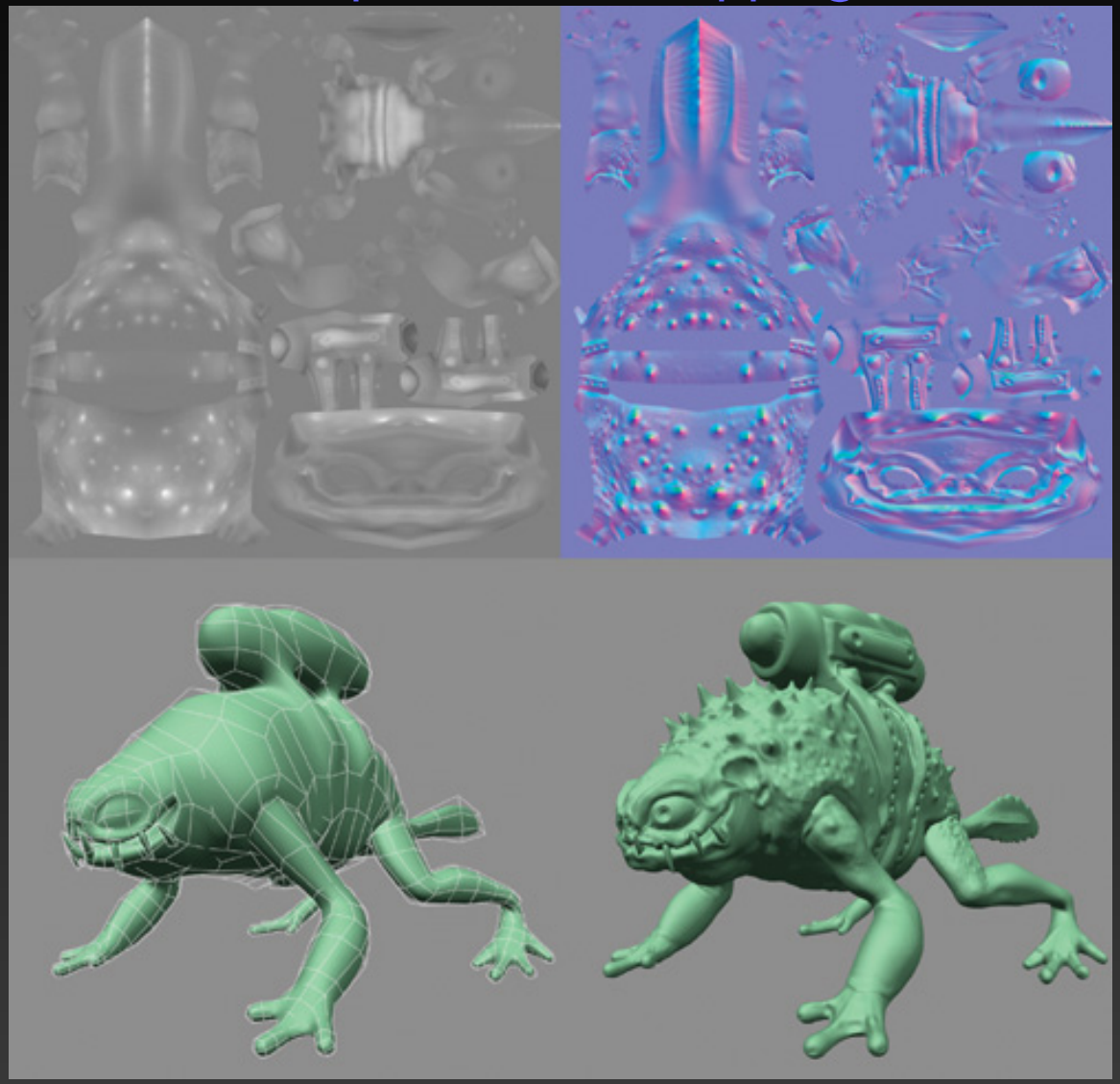
Displacement Mapping



Displacement Map

Utiliser la texture pour perturber/déplacer la géométrie

Displacement Mapping



Combinaison de cartes

Mélanger les effets

Réalisme et complexité

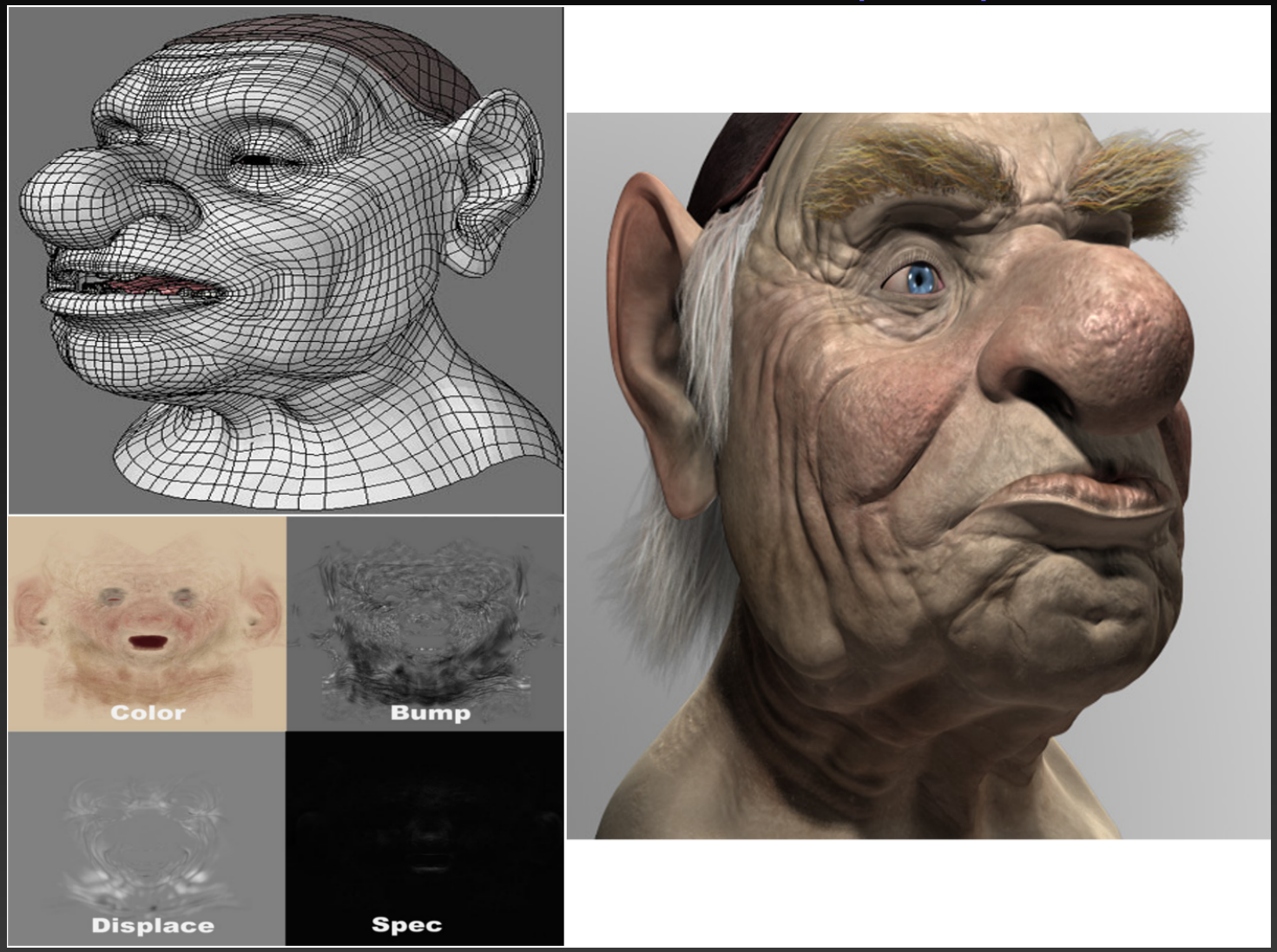
Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Dans les moteurs 3D : En pratique



http ://www.brilliant-creations.com/hoggletut.htm

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Textures
En pratique

Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

En OpenGL : Objet Texture

- Fonctionnement identique aux **Objets OpenGL**
- Un identifiant unique (GLuint)
- Représente un espace dans la mémoire graphique
- On manipule l'identifiant (création, chargement, utilisation...)

Chargement de texture

- Il faut d'abord avoir la texture en mémoire CPU (librairies...)
- Création de l'objet texture sur GPU : glGenTextures(1,&texId) ;
- Activation de l'objet texture : glBindTexture(GL_TEXTURE_2D,texId) ;
- Chargement dans la mémoire GPU : glTexImage 1|2|3D(...
- Des paramètres en pagaille (voir doc OpenGL)
- Taille, format des données, bordures...

Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

En OpenGL(>=3.3) : Objet Samplers

- Représente toutes les manières d'interpréter une texture
- Répétition de la texture, mipmapping, filtrage ?
- Fonctionnement identique aux **Objets OpenGL**
- Un identifiant unique (GLuint)
- On manipule l'identifiant
  - Creation : glGenSamplers(int nb,GLuint* id)
  - Definition : glSamplerParameter(...)
  - Utilisation : glBindSampler(GLuint texId,GLuint samplerId)

Textures - La pratique

Réalisme et complexité

Principe

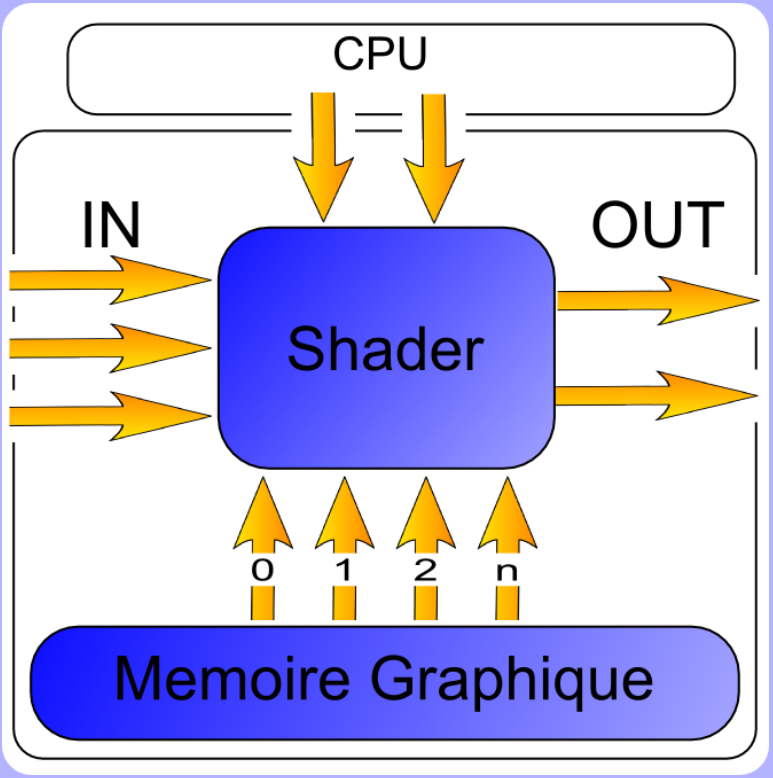
Interpolation et filtrage

Un peu plus loin

En pratique

OpenGL < 4.2

- Pas d'accès direct à la mémoire graphique
- Accès indirect : mécanisme d'unité de textures
- Le pipeline accède uniquement aux unités de textures



Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

OpenGL < 4.2

- Pas d'accès direct à la mémoire graphique
- Accès indirect : mécanisme d'unité de textures
- Le pipeline accède uniquement aux unités de textures

OpenGL >= 4.2

- Accès direct à la mémoire graphique
- Notion d'objet Image
- Changement dans la philosophie

Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

OpenGL - Interaction avec le pipeline

- Plusieurs unité de textures (8 minimum maintenant)
- Définition/Changement d'unité active **i**
  - glActiveTexture(GL_TEXTURE**i**)
- Opérations suivantes sur l'unité active **i**
- Exemple : Utilisation d'une texture
  - glBindTexture(GL_TEXTURE_2D,texId)
- Activer le pipeline
- Penser a libérer l'unité
  - glBindTexture(GL_TEXTURE_2D,0)

Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

GobLim : Objet GPUTexture

- Un objet **GPUTexture2D** (encapsule texture et sampler)
- Un constructeur : GPUTexture2D(string nomDeFichier)
- Une fonction pour récupérer l'adresse mémoire sur GPU de la texture : getHandle()

GobLim : Et avec le shader ?

- une extension a activer dans le shader : extension *GL_ARB_bindless_texture* : *enable*
- Une variable uniform de type sampler2D
- Transmettre depuis le CPU l'adresse de l'unité correspondante
- Récuperer un texel : vec4 texture(sampler2D myTex,vec* texCoords)

Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Exemple : Initialisation

- Créer et charger la **GPUTexture**
- Récupérer pointeur sur variable GLSL (**GPUSampler**)
- Transmettre a cette variable l'adresse mémoire GPU de la texture (getHandle())

Exemple : A chaque rendu

- Rien a faire, la texture reste de manière permanente en mémoire

Textures - La pratique

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

Exemple : Fragment Shader

```
#version 440 core

layout (std140) uniform CPU
{
 sampler2D maTexture;
{
in vec2 v_TexCoord;
out vec4 Color;
void main()
{
        Color = texture(maTexture,v_TexCoord);
}
```

Réalisme et complexité

Principe

Interpolation et filtrage

Un peu plus loin

En pratique

A vous de jouer !