

Textures

Modélisation de la « matière »

- **Problème** : ne pas tout modéliser à l'échelle de la géométrie !

On veut garder une seule face, mais plusieurs couleurs

⇒ **Texture de couleurs**



Des micro-polygones seraient nécessaires

⇒ **Texture de normales**

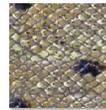


Les textures

- Une texture est un **champ scalaire discret** (*lookup table*) sur la surface
 - Défini sur un domaine linéaire (1D) / rectangulaire (2D) / cubique (3D)
 - Mappé sur la surface par des **coordonnées** de texture
 - Spécifiées en chaque sommet
 - Interpolées pour chaque fragment

Les textures

- Ajout d'information visuelle à **petit prix**
- **Support hardware**
 - interpolation des coordonnées de texture
 - interpolation des valeurs de couleur
 - filtrage multi-résolution (*mip-mapping*)



x

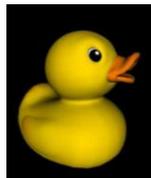


=



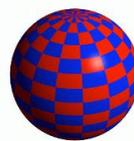
Les textures

- Peuvent être utilisées pour spécifier :
 - **La couleur ambiante / diffuse**

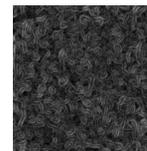


Les textures

- Peuvent être utilisées pour spécifier :
 - La couleur ambiante / diffuse
 - **Les normales** (*bump / normal mapping*)



Texture Diffuse mappée sur la sphère



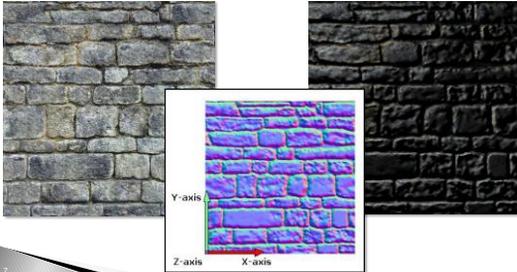
Bump Map



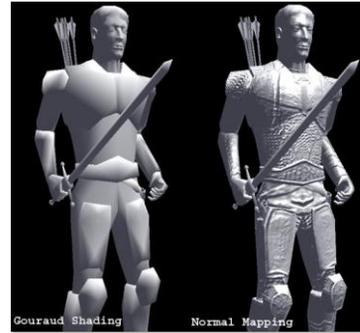
Textures combinées

Perturbation des normales

- Donnée par une texture de normale (*normal map*)



Perturbation des normales



Bump vs. Normal Map

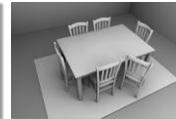
- *Bump mapping*
 - Stockage de la **hauteur de la surface**
 - Calcul des dérivées pour obtenir le déplacement dans le plan tangent à la surface
- *Normal mapping*
 - Stockage direct des **déplacements dans le plan tangent**
- Extensions :
 - *Parallax mapping* (déplacement des coordonnées de texture)
 - *Displacement mapping* (déplacement de la géométrie)

Les textures

- Peuvent être utilisées pour spécifier :
 - La couleur ambiante / diffuse
 - Les normales (*bump / normal mapping*)
 - Une **illumination pré-calculée** (*light mapping*)



Texture diffuse mappée sur la scène



Light Map mappée sur la scène



Textures Combinées

Les textures

- Peuvent être utilisées pour spécifier :
 - La couleur ambiante / diffuse
 - Les normales (*bump / normal mapping*)
 - Une illumination pré-calculée (*light mapping*)
 - **Les réflexions** (*environment mapping*)

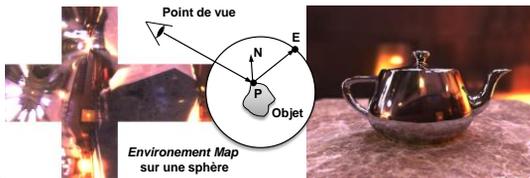
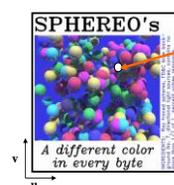


Image by Henrik Wann Jensen - Environment map by Paul Debevec

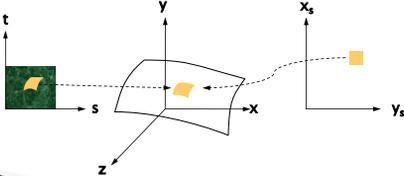
Les textures 2D

- Image plane $I(u,v)$
- Fonction de placage (mapping)
 $f: P(x,y,z) \rightarrow (u,v)$
- Modèle 3D : points ou faces, normales, coordonnées de texture (u,v)



Rendu

- ▶ Modèle + texture \Rightarrow pixel
- ▶ Pour chaque pixel utiliser le mapping inverse pour calculer (u,v) à partir de (x,y)
 - Peut-être fait incrémentalement, codé en hard
- ▶ Afficher le pixel le plus proche de (u,v)



13

Problèmes



14

Problèmes

- ▶ Aliassage
- ▶ Plaquer une texture sans distorsion
- ▶ Synthétiser une texture (cf. cours génération de contenu)

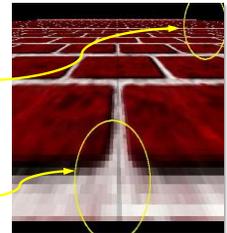
15

Problèmes

- ▶ Aliassage
- ▶ Plaquer une texture sans distorsion

Plusieurs couleurs pour un pixels

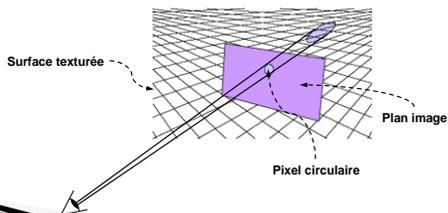
Pixels visibles



16

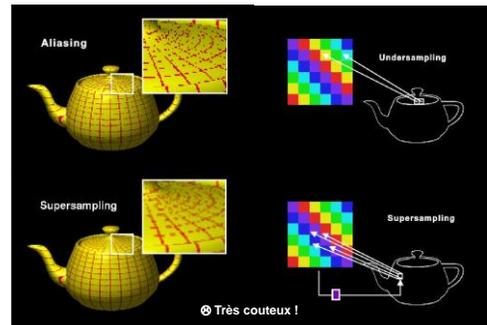
Aliassage

- ▶ Problème d'échantillonnage d'un signal
 - Particulièrement visible durant une animation
 - Apparition/disparition brusque de détails
- ▶ Comment associer une unique valeur à la surface vue à travers un pixel du plan image ?



17

Solution 1 : calcul à haute résolution



18

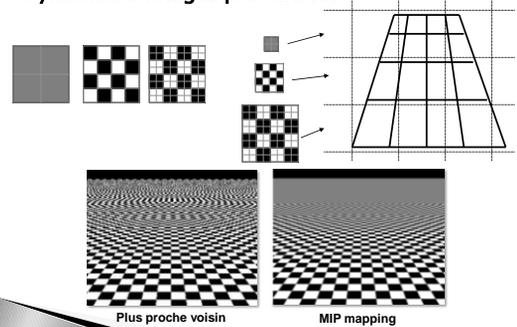
Solution 2: filtrage spatial

- › Supprimer les hautes fréquences qui créent les artefacts lors de la minification
- › Intégration (convolution) **variant spatialement** en chaque pixel
⇒ trop coûteux
- › Pré-calcul d'une **approximation** possible

19

Solution 2 : MIP Mapping

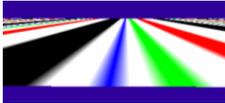
- › **Pyramide d'images pré-filtrées**



20

Solution 2 : MIP Mapping

- › **Problème** : ne tient pas compte de l'anisotropie
 - L'approximation carrée de la MIP-map devient mauvaise



Plus proche voisin



MIP Mapping

- › **Solutions** (plus coûteuses)
 - Filtrage anisotrope
 - *Elliptical weighted average*

21

Problèmes

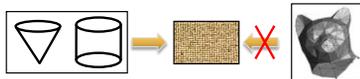
- › Aliasing
- › **Plaquer une texture sans distorsion**



22

Fonction de placage

- › Equivalent à un **dépliage**

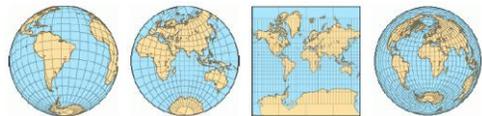


- › Pas toujours possible : **surface développable**
- › Problème **local**

23

Fonction de placage

- › Problème connu en cartographie



- ⇒ **Distorsion** globale ou locale
- ⇒ Choix de conserver angles, distances, ...

24

Solutions simples

$f: (x,y,z) \rightarrow [0,1] \times [0,1]$

- ▶ **Planaire** : projection
 $f(x,y,z) = (||x||, ||y||)$
- ▶ **Cylindrique**
 $f(\theta, z) = (\theta/2\pi, z)$
- ▶ **Sphérique**
 $f(\theta, z) = (\theta/2\pi, z)$

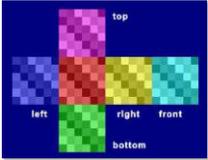
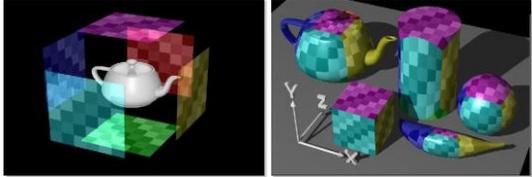


25

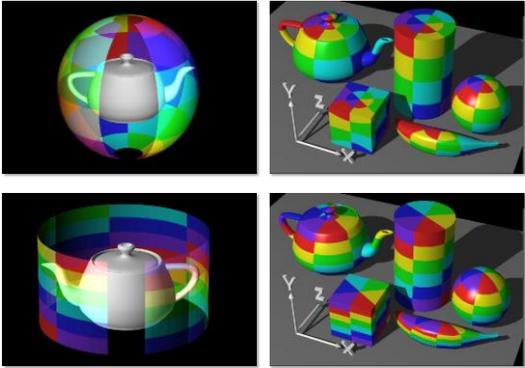
Solutions simples

▶ Passer par un objet simple intermédiaire :

1. Dépliage
2. Projection

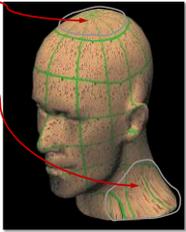
26



27

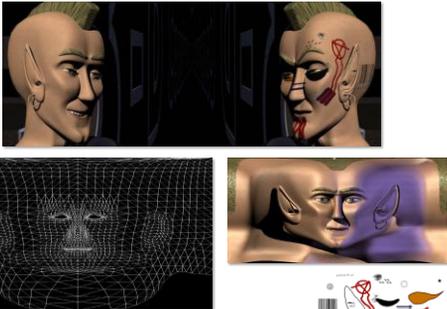
Cas général

- ▶ Problème d'optimisation de la paramétrisation pour minimiser
 - Les **singularités** (pôles)
 - Les **distorsions**
- ▶ Réalisé à la main par les artistes
- ▶ Techniques automatiques



28

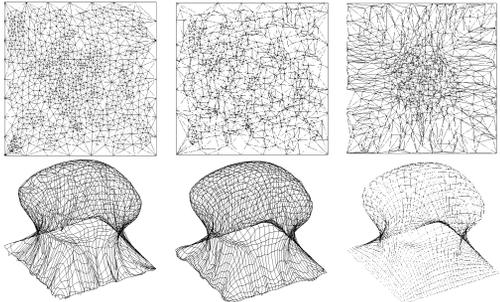
Paramétrisation manuelle



<http://www.elfworks.com/Articles/skin-o-matic.html>

29

Exemples d'optimisations automatiques



Iso-barycentre **Conservation des distances** **Barycentre (3 pts)**

cf. : <http://www2.in.tu-clausthal.de/~hormann/parameterization/index.html>

30