

Plaquage et synthèse de textures

Cours de Master 2:
Techniques de Rendu pour la Synthèse d'Images

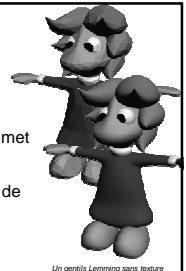
Basé sur les cours de: Cyril Soler, Joëlle Thollot, Elmar Eisemann, Frédo Durand, Barbara Cuttler, E. Boyer, H. Briceno, N. Holzschuch, Alex Meyer, Xavier Granier (oui tout ce monde là !)



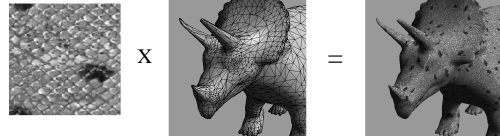
Cyril Crassin – Octobre 2007

Introduction

- Motivation
 - Sans texture:
 - Couleurs et transparence en chaque sommet
 - Problème:
 - On ne peut pas tout modéliser à l'échelle de la géométrie
 - Coût mémoire
 - Ajout d'information visuelle à petit prix



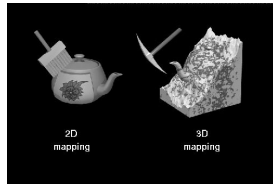
Un gentil Lemming sans texture



Un méchant rhino, mais texture

Types de textures

- Dimensionnalité:
 - 1D (u), 2D (u,v), ou 3D (u,v,w)
- Mode de génération:
 - Manuelle/Photographie
 - Procédurale
- Utilisation:
 - Couleurs
 - Relief
 - Réflexions
 - ...



Bump mapping



Environment mapping

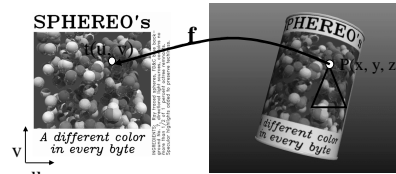
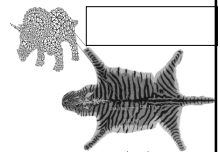


Plan

- Plaquage de textures 2D
- L'application de texture lors de la rasterisation
- Filtrage de textures
- Textures 3D et génération procédurale
- Synthèse de textures
- Synthèse sur surfaces

Plaquage 2D

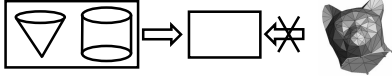
- Texture: Image plane I (u,v)
- Modèle 3D :
 - sommets, normales, coordonnées de texture (u,v)
- Problème: **Lier une image 2D avec un objet 3D quelconque**
 - Fonction de plaquage (mapping) $f : P(x,y,z) \rightarrow (u,v)$



Fonction de plaquage

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

● Équivalent à déplier la surface



● Problèmes:

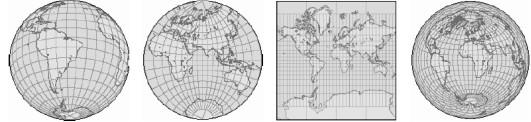
- Pas toujours possible :
 - Problème de topologie
 - Surface développable
 - Homéomorphisme à un disque topologique.
- Ex: sphère

Fonction de plaquage

● Sacrifices: distorsions

- Locales ou globales

● Problème connu en cartographie



⇒ Distorsion globale ou locale

⇒ Choix de conserver angles, distances, ...

Fonction de placage

● Définition d'une forme de placage

- Englobant simple pour l'objet
 - On sait le paramétrer
- Projection sur cette forme
- Définit la manière dont la texture sera appliquée

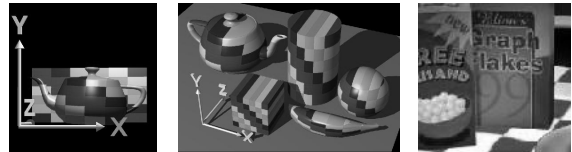
● Définition d'une fonction de conversion

- Fonction de projection sur la forme de placage

Formes de placage

● Planaire

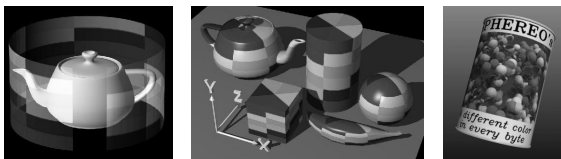
- Projection simple
- $f(x,y,z) = (||x||, ||y||)$



Formes de placage

● Cylindrique

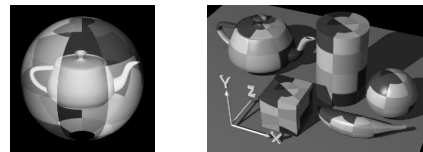
- $(\theta, z) \rightarrow (u, v)$
- Quel cylindre ?



Formes de placage

● Sphérique

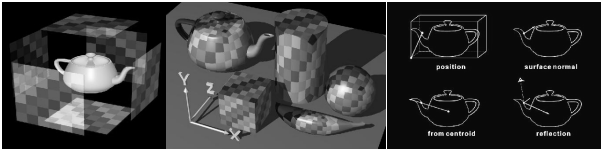
- $(\theta, \phi) \rightarrow (u, v)$



Formes de placage

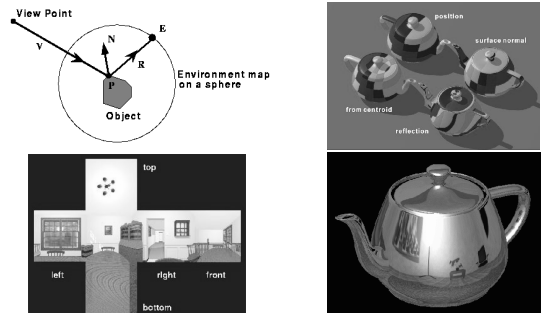
● Cubique

- Similaire au placage plan
- Définition de 6 plans (6 « cas »)



Environment mapping

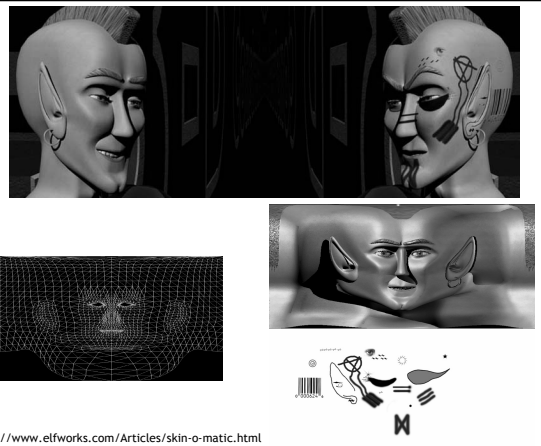
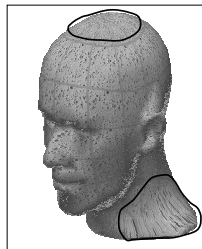
● Fonction de conversion : reflection



Problèmes dans le cas général

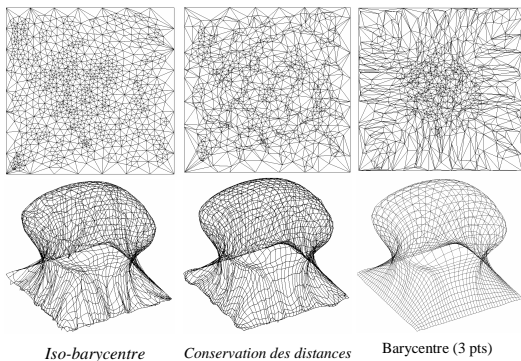
- Distortions
- Poles

- ⇒ Paramétrisation de la surface qui minimise la distortion
- ⇒ Pb d'optimisation
- ⇒ **Fait à la main par les graphistes**



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

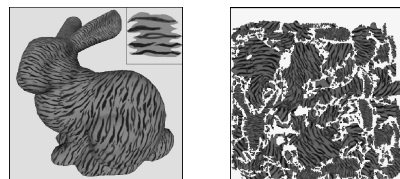
Exemples d'optimisations automatiques



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

Plaquage inverse: Atlas de textures

- Usage:
 - Pour rendre efficace stockage et affichage après synthèse
- Algorithme:
 - Découper la surface en *patches* P à topologie de disque
 - Trouver un plaquage local P → P' qui minimise la distortion
 - Remplir une image au mieux avec les patches P'



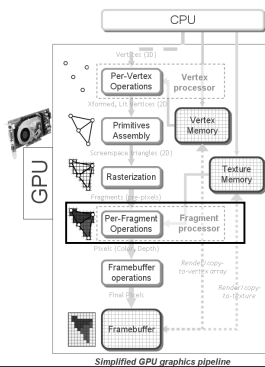
L'application de texture dans le pipeline graphique

Textures et rasterisation

- Modèle projectif:
 - Paramétrisation de la surface:
 - Coordonnées de textures (u, v) associées aux sommets
 - Projection des sommets:
 - 3D model space \rightarrow 2D screen space
 - Rasterisation :
 - Remplissage du triangle projeté
 - Interpolation en espace image des (u, v)

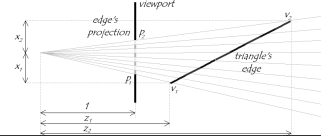
Accélération matérielle

- Lors de la rasterisation:
 - Interpolation des coordonnées de texture (*varying*)
 - Correction perspective
- Lors de l'accès aux textures
 - Interpolation des valeurs de couleur (filtrage linéaire)
 - Filtrage multi-résolution (*mip-mapping*)
- Fragment Shader
 - Modulation, transformation, accès dépendants...



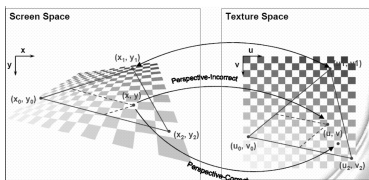
Correction perspective (1/2)

- Problème:
 - Interpoler en espace écran != interpoler en espace objet
- Le mappage qui varie linéairement dans l'espace ne varie pas linéairement sur l'écran !



Correction perspective (2/2)

- Solution:
 - $u/z, v/z$ et $1/z$ varient linéairement !
 - On interpole ($u/z, v/z, *, 1/z$)
 - Puis par fragment on divise u/z et v/z interpolés par $1/z$ interpolé



- Réalisé automatiquement par le hardware lors de la rasterisation

<http://groups.csail.mit.edu/graphics/classes/6.837/F98/Lecture21/Slide14.html>
<http://easynet.co.uk/%7Emmeanie/imap/imap.htm>
<http://www.lysator.liu.se/~mikaelt/doc/perspectiveoftexture/>

Filtrage

Problèmes de filtrage

● Les texels ne correspondent jamais exactement aux pixels

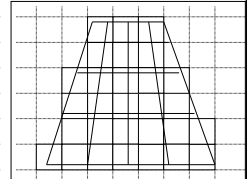
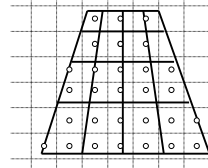
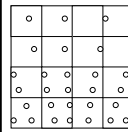
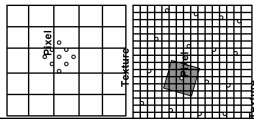
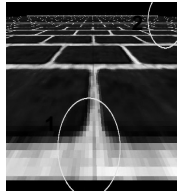
● Deux cas:

■ Magnification (1)

- Peu de texels sur beaucoup de pixels
 - Blocs
- Nearest
- Bilinear interpolation

■ Minification (2)

- Plusieurs texels sur un pixel
 - Aliasing
- Intégration de la couleur du pixel à partir de l'ensemble des texels



Texture map

Polygon far from the viewer in perspective projection

Rasterized and textured

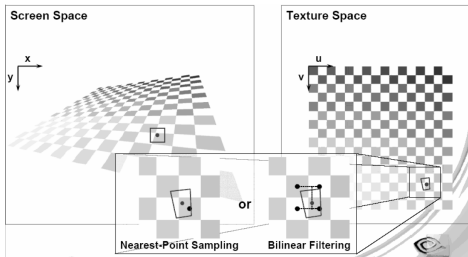
● Devant, on voit les pixels

● Au fond

- plusieurs couleurs à afficher dans un pixel
 - On aimerait une moyenne!
- basses fréquences parasites : les « alias »

Filtrage: Magnification

● Point le plus proche vs interpolation bilinéaire



Filtrage: Minification

● Problème:

- L'empreinte d'un pixel varie dans une même primitive
- Calculer l'intégrale pour un pixel est trop coûteux

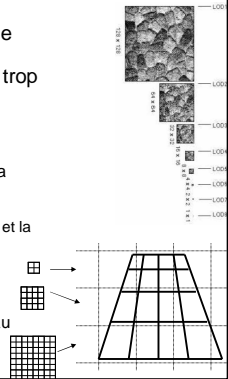
● Solution:

■ Mip-Mapping

- On pré-calcule une pyramide filtrée de la texture (LOD)
- On choisit la résolution adaptée
 - Fonction de l'angle entre le vecteur vue et la surface

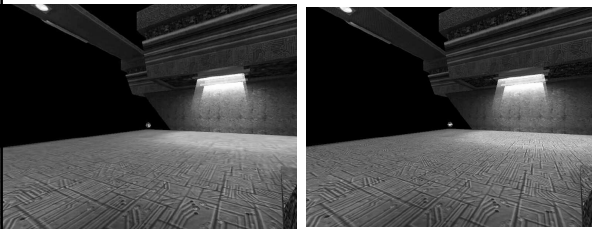
■ Filtrage trilineaire

- Utilisation des deux niveaux de LOD les plus proches
- Interpolation bilinéaire sur chaque niveau
- Interpolation linéaire entre niveaux



Effet dans Quake

● Quake 3:



Mais il y a encore des problèmes !

● Toujours une approximation de l'intégrale

- Echantillonnage carré (isotrope) de la texture
 - Taille des pixels en espace texture identique dans chaque direction

● Flou excessif

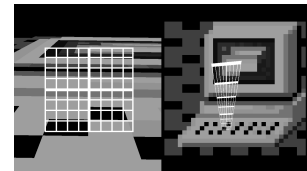


Figure 5. Anisotropic footprints are very common.

● Solution:

- Filtrage anisotrope
 - Prend en compte la forme de l'empreinte des pixels
 - Techniques multiples
- Exemple

Effet pas dans Quake



Textures 3D

Les textures 3D

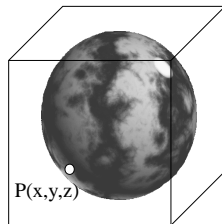
Volume de matière où l'objet est plongé

- bois, marbre...

+ plaquage trivial !

$u = x, v = y, w = z$

- coût mémoire
(ou coût de calcul)

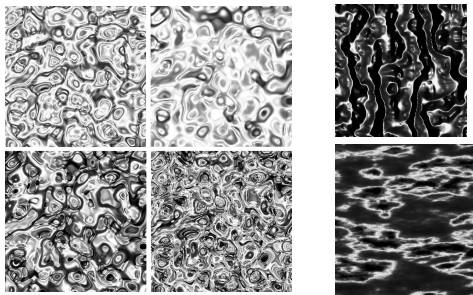


Construction de textures 3D

- Photo impossible
- Synthèse par l'exemple
 - Difficile en 3D
- Textures procédurales
 - Marche en 2D et 3D

Textures de Perlin

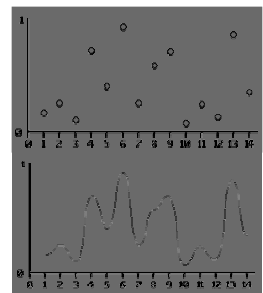
- Bruit fractal continu



Bruit de Perlin

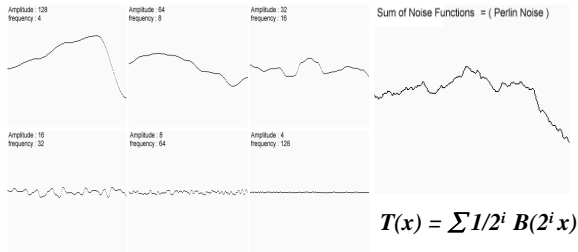
Fonction de base (1D)

- $B(x)$ = interpolation de valeurs aléatoires, en des points régulièrement espacés
- Pré-calcul des valeurs (tableau 1D)
- Pour un bruit moins lisse $B'(x) = |2B(x) - 1|$



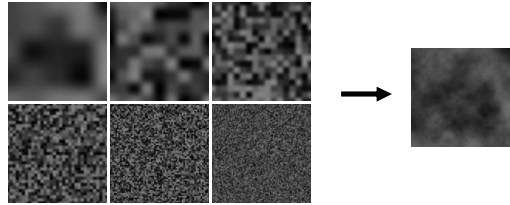
Bruit de Perlin

Bruit turbulent : sommer des copies de B à différentes échelles



Textures de Perlin

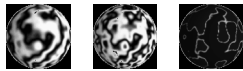
Extension des bruits de Perlin en 2D ou 3D



Paramètres : amplitude et fréquence de chaque octave

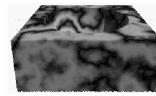
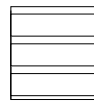
Utilisation

- Utilisation directe
 - Éventuellement seuillée
- Modification d'une image ou d'une fonction simple
 - Image : $I(x,y)$ remplacée par $I(x+T_1(x,y), y+T_2(x,y))$



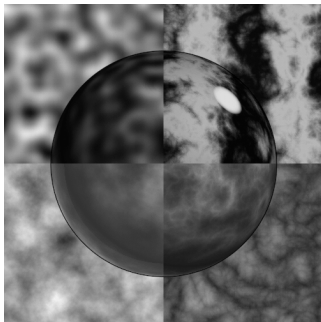
Utilisation

- Modification d'une image ou d'une fonction simple
 - Fonction



$$I(x,y,z) = \cos(x + T(x,y,z))$$

Bruit B $\sin(x + T)$

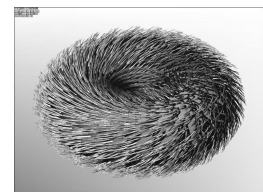
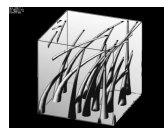


$$T = \sum 1/f(B)$$

$$\sum 1/f(|B|)$$

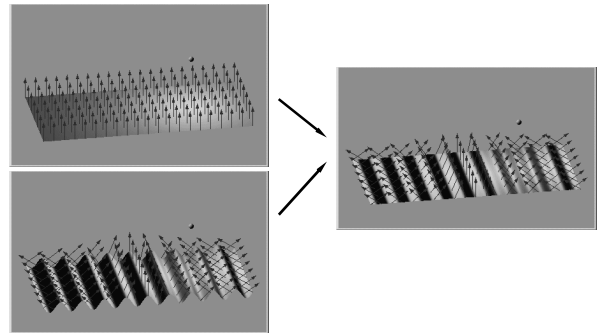
Texels : mélange de 2D et 3D

- Element de texture en relief, multi-résolution
- On les plaque le long d'une surface.



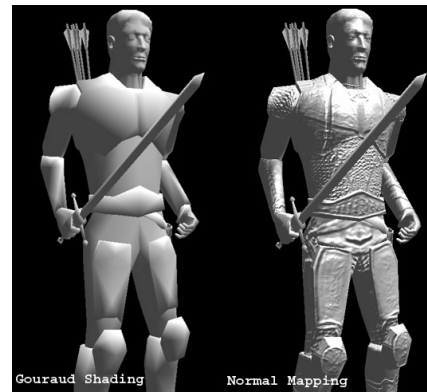
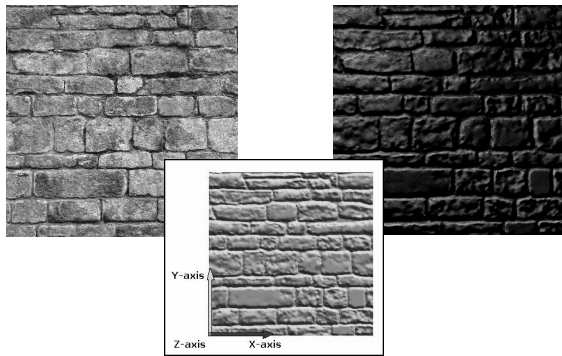
Textures d'autre chose

Textures de normales (bump mapping)



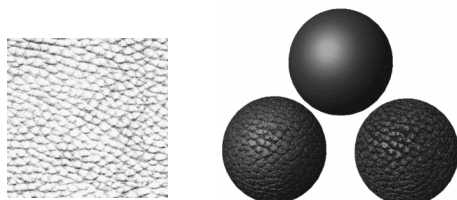
Perturbation des normales

- Donnée par une texture de normale (normal map)



Perturbation des normales

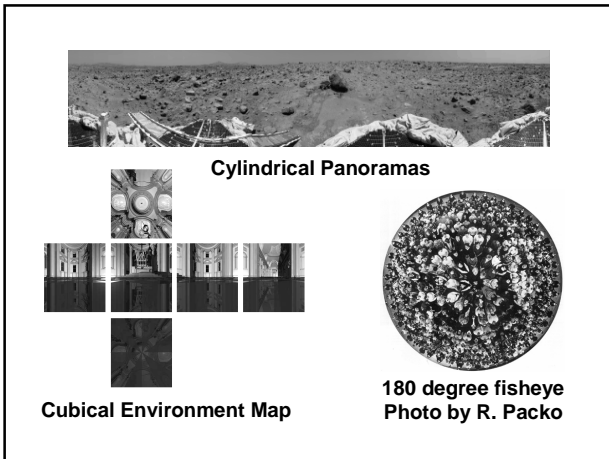
- Donnée par une texture de relief (bump map)



Environment map

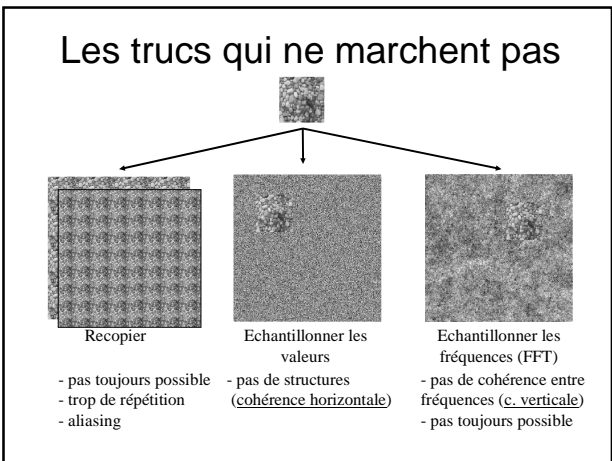
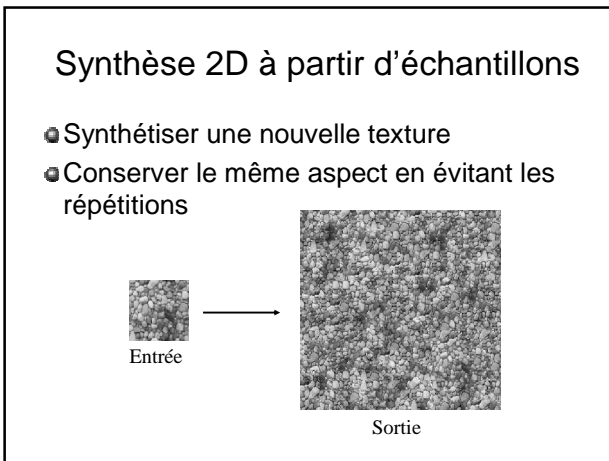
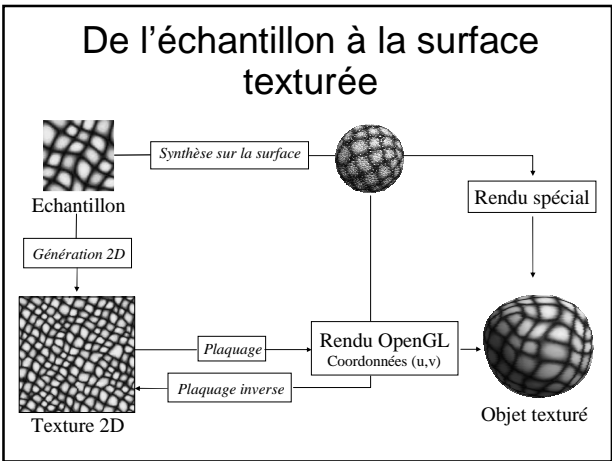


Miller and Hoffman, 1984



Synthèse de textures

- ## Problèmes
- Synthétiser une texture par l'exemple
 - Synthétiser sur une surface directement



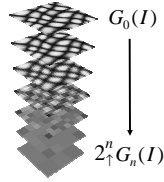
Pyramides de Gauss et de Laplace

● Pyramide de Gauss

$$G_i(I) = 2_{\downarrow} [G_{i-1}(I) \otimes g]$$

$$G_0(I) = I$$

g : noyau gaussien
 $2_{\uparrow}, 2_{\downarrow}$: opérateurs de sur/sous échantillonnage



● Pyramide de Laplace

- Isole les caractéristiques de chaque niveau

$$L_i(I) = G_i(I) - 2_{\uparrow} G_{i+1}(I)$$

Méthode Debonet (97)

- Utilise une pyramide de Laplace

● Hypothèse:

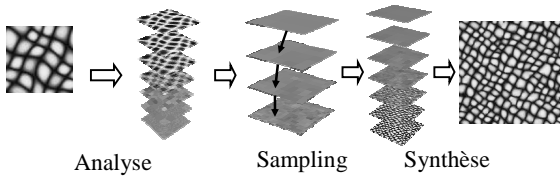
Aux résolutions les plus faibles, il existe des régions interchangeables. Seul l'aspect stochastique de la texture dépend de la position de ces régions.



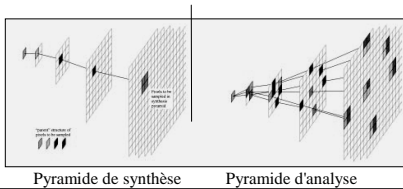
● Synthèse:

- construire une nouvelle pyramide de Laplace en imitant celle de l'échantillon
- introduire de l'aléatoire dans le placement des structures.
- re-transformer la pyramide construite en image

● Algorithme

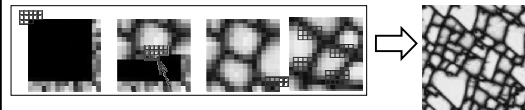


● Sampling

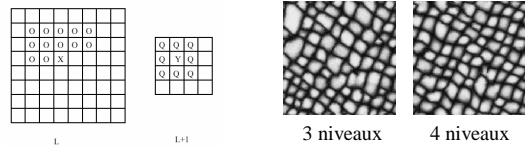


Méthode Wei (2000)

Synthèse pixel par pixel (cohérence horizontale)



Version multi-résolutions (cohérence verticale)
 On utilise une pyramide de Gauss



● Résultats

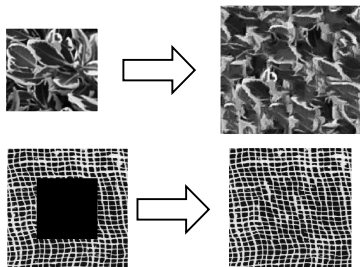
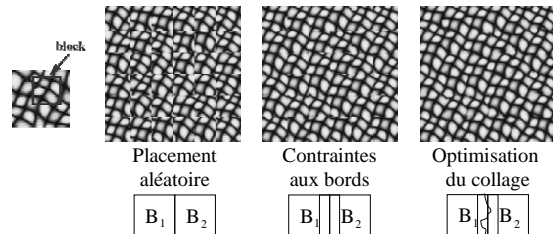


Image Quilting (Efros'01)

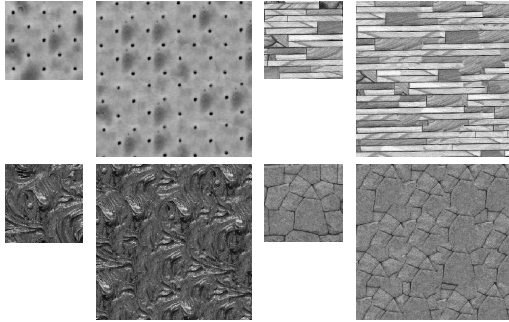
● Principe:

- Collage de blocs sélectionnés au hasard
- Recollement par une technique adaptée:



- L'échantillon n'a pas besoin d'être torique

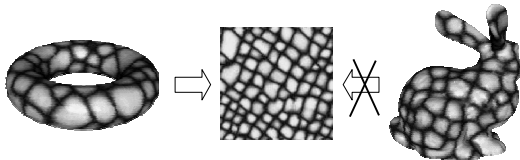
● Résultats



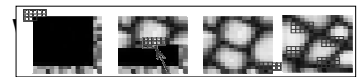
Problèmes

- Synthétiser une texture par l'exemple
- Synthétiser sur une surface directement

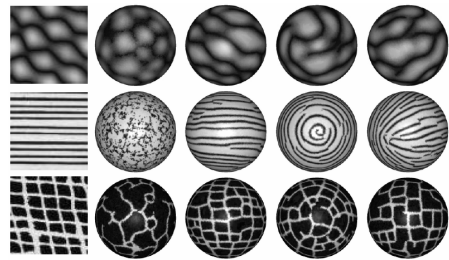
Passer à la surface



- Même si on sait minimiser localement les déformations...
- ...la topologie n'est pas toujours compatible
- Synthèse directe sur la surface à partir d'un échantillon

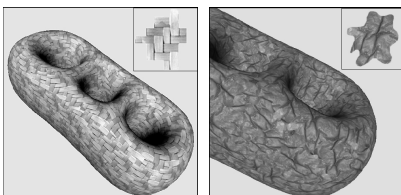


- Voisinage sur la surface
- Comment traiter les conflits topologiques ?



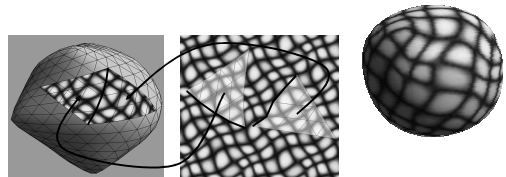
Lapped textures (Praun'00)

- Série de textures pré-découpées selon les discontinuités (utilisateur)
- Un maillage + un champ de vecteurs
- Morceaux de texture collés dans la direction du champ de vecteurs jusqu'à recouvrement



Hierarchical Pattern Mapping (Soler'02)

- Sélectionner des régions indépendantes dont les bords collent



● Difficultés:

- chercher les régions qui collent de manière efficace
- réduire les discontinuités

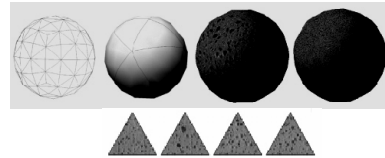
● Intérêts

- Préservation du maillage et de la texture originale
- Pas d'orientation privilégiée, topologie gérée automatiquement
- L'information calculée (coord. de textures) est portable



Pattern texture mapping
(Neyret'99)

- Découper en patches triangulaires *réguliers*
- Former des tuiles respectant toutes les contraintes possibles
- Plaquer chaque tuile



● Gestion des contraintes

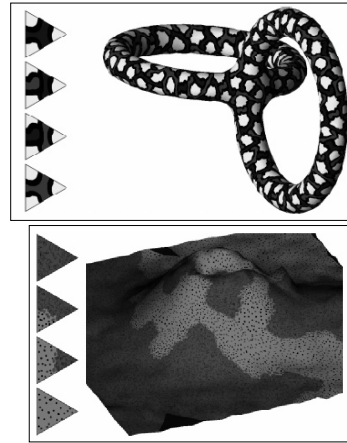
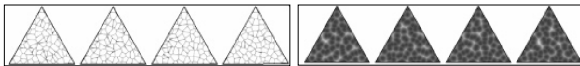


Cas général:

$n/2$ contraintes: $n + n(n-1) + n(n-1)(n-2)/3$ tuiles.

● Construction des tuiles

- Algorithmes de Worley, de Perlin
- Dessin à la main.
- Transfert par pyramide ?



Références

Synthèse 2D:

- *Multiresolution sampling procedure for the analysis and synthesis of texture images.* (J. Debonet.) Siggraph'97
- *Image quilting for texture synthesis and transfer.* (A.Efros, W.Freeman). Siggraph'00.

Synthèse sur surfaces:

- *Texture Synthesis over Arbitrary Manifold Surfaces* (Li-Yi Wei, Mar Levoy). Siggraph'01
- *Texture Synthesis on Surfaces* (Greg Turk). Siggraph'2001

Plaquage sans distorsions:

- *Piecewise surface flattening for non distorted texture mapping.* (Chakib Bennis, J.M. Vézien, G. Igiésias) Siggraph'91.
- *Parametrization and smooth approximation of surface triangulations.* (Michael S.Floater.)
- *Non distorted texture mapping for sheared triangulated meshes.* (B.Levy, J.L. Mallet). Siggraph'97

Plaquage inverse:

- *A Texture-Mapping Approach for the Compression Of Colored 3D triangulations.* (Soucy Marc, Guy Godinand, Marc Rioux), TheVisual Computer, Vol.12, No.10, 1996, pp.503-514.

Découpage en patches:

- *Multiresolution analysis of arbitrary meshes.* (M.Eck,...). Siggraph'95