

# Synthèse et plaquage de textures

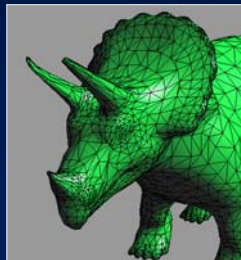
Cours de DEA  
Jeudi 25 Janvier 2007  
Cyril Soler

## Rappels sur les textures

- ✓ Pourquoi texturer ?
  - Ajout d'information visuelle à petit prix
  - Support hardware
    - » interpolation des coordonnées de texture
    - » interpolation des valeurs de couleur
    - » filtrage multi-résolution (*mip-mapping*)
    - » modulation
    - » *texture shaders* (NVidia, ATI)



X



=

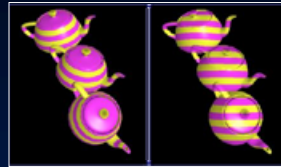


## Rappels sur les textures

- ✓ Implémentation:  $u, (u,v)$ , ou  $(u,v,w)$  par vertex



Surfacique (2D)



Volumique (1D/2D/3D)

- ✓ Techniques reliées:



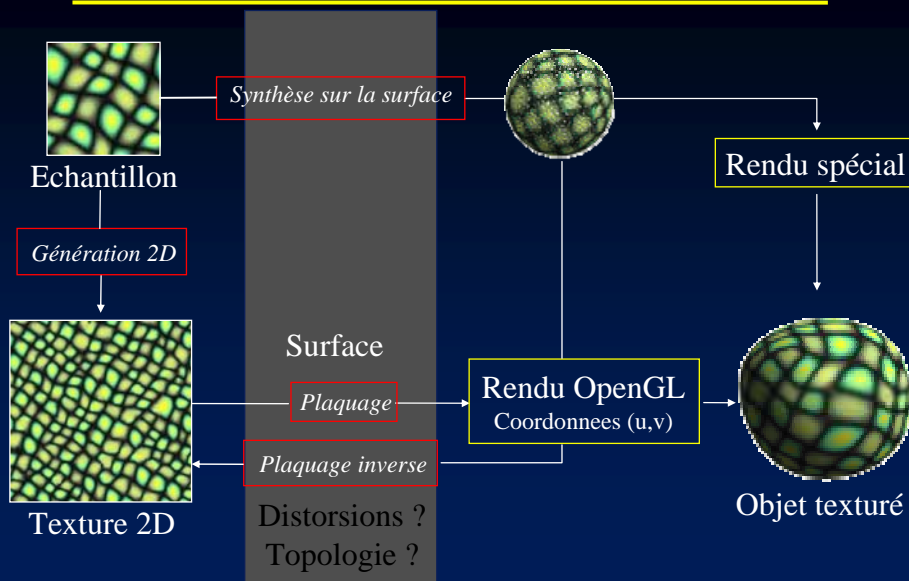
Bump mapping



Environment mapping

Introduction

## De l'échantillon à la surface texturée



Introduction

## Plan

---

- ✓ Synthèse 2D à partir d'un échantillon
- ✓ Plaquage sans distorsions (?)
- ✓ Synthèse sur surfaces
- ✓ Plaquage inverse

## Synthèse 2D à partir d'échantillons

---

- ✓ But: Synthétiser une nouvelle texture
  - Conserver le même `aspect` en évitant les répétitions

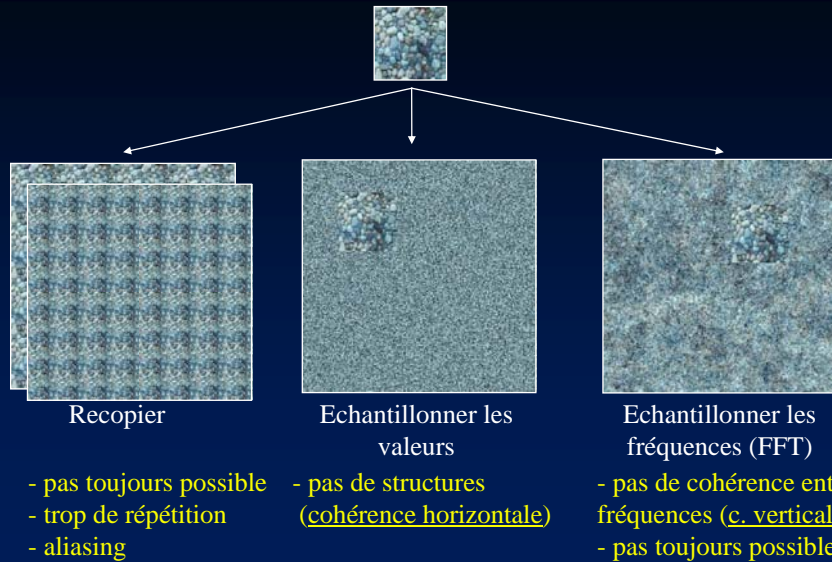


Entrée



Sortie

## Les trucs qui ne marchent pas



Synthèse 2D

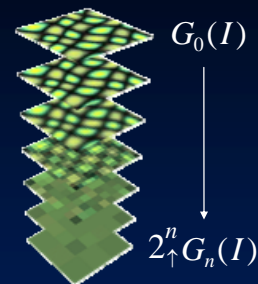
## Pyramides de Gauss et de Laplace

### ✓ 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

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

Synthèse 2D

## Méthode Debonet (Debonet'97) (1/2)

- ✓ 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.

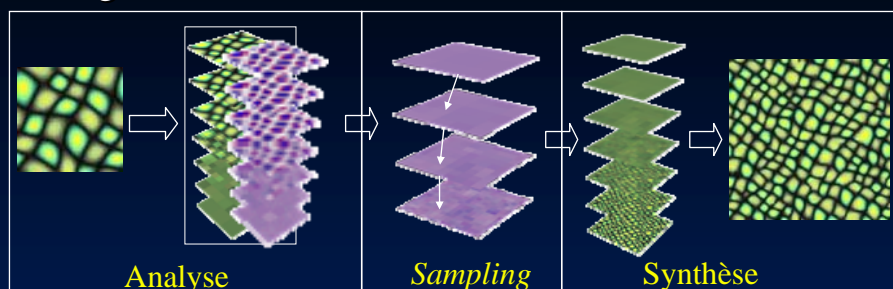


- ✓ Synthèse:
  - construire une nouvelle pyramide de Laplace en imitant celle de l'échantillon
  - introduire de l'aléatoire dans le placement des structures.
  - re transformer la pyramide construite en image

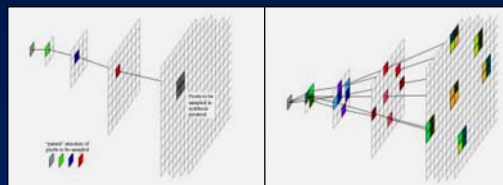
Synthèse 2D

## Méthode Debonet (Debonet'97) (2/2)

- ✓ Algorithme



- ✓ Sampling



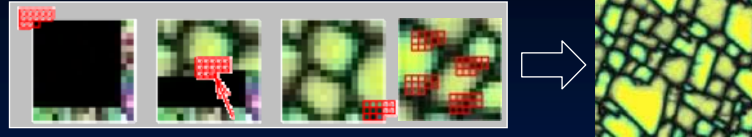
Pyramide de synthèse

Pyramide d'analyse

Synthèse 2D

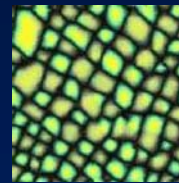
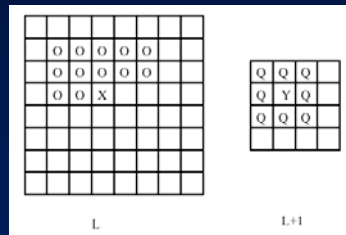
## Méthode Wei (Wei'00) (1/2)

- ✓ Synthèse pixel par pixel (cohérence horizontale)

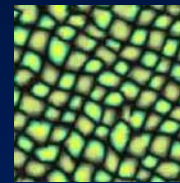


- ✓ Version multi-résolutions (cohérence verticale)

– On utilise une pyramide de Gauss



3 niveaux

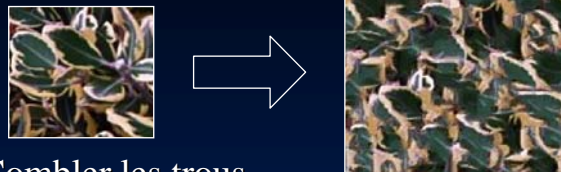


4 niveaux

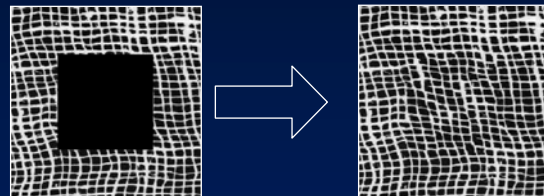
Synthèse 2D

## Wei'00 (2/2) Applications

- ✓ Exemples



- ✓ Comblent les trous



- ✓ Optimisations

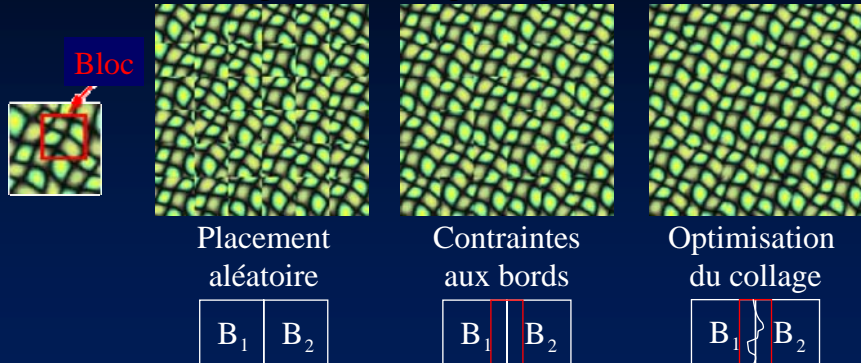
– quantisation de vecteurs

Synthèse 2D

## Image Quilting (Efros'01) (1/2)

### ✓ Principe:

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

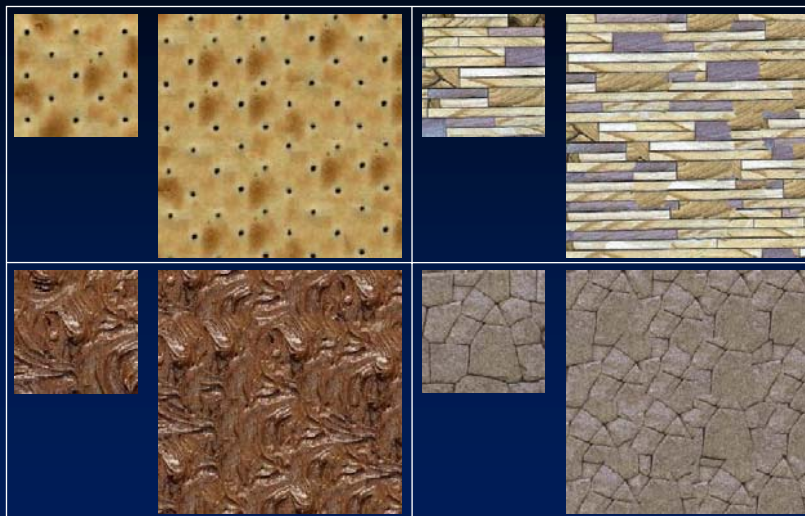


- L'échantillon n'a pas besoin d'être torique

Synthèse 2D

## Image Quilting (Efros'01) (2/2)

### ✓ Résultats



Synthèse 2D

## Plan

---

- ✓ Synthèse 2D à partir d'un échantillon
- ✓ **Plaquage sans distorsions (?)**
- ✓ Synthèse sur surfaces
- ✓ Plaquage inverse

## Plaquage sans distorsion

---

- ✓ Equivalent à un dépliage

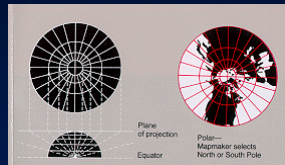


- Pas toujours possible (la surface doit être developpable)
- Problème local

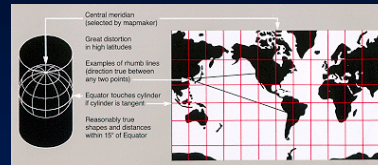


## Plaquage sans distorsion

- ✓ Vérification locale des contraintes:
  - Conservent les angles *ou* les distances
  - Exemple: Projections cartographiques spécialisées (En géographie)



Gnomonique



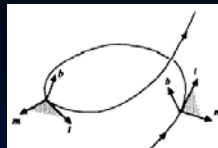
Mercator

- ✓ Dépliage autour d'une courbe
- ✓ Répartition globale des contraintes:
  - Techniques d'optimisation

Distorsion

## Courbure (Rappels)

$$X(s) = \begin{bmatrix} x(s) \\ y(s) \\ z(s) \end{bmatrix}$$



$s$  : abscisse curviligne  
 $t, m, b$ : repere de Frenet

$$t = X' = \frac{dX}{ds}$$

$$m = \frac{X''}{\|X''\|}$$

$$b = t \wedge m$$

$$k(s) = \|t'(s)\| \quad \text{Courbure principale}$$

$$\tau(s) = \|b'(s)\| \quad \text{Torsion}$$

$$k = 0 \implies \text{Courbe rectiligne}$$

$$\tau = 0 \implies \text{Courbe plane}$$

Distorsion

## Courbes géodésiques sur une surface

### ✓ Courbes géodésiques

Surface  $S(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}$

$$N(u, v) = \frac{dS}{du} \wedge \frac{dS}{dv}$$

$$n = \frac{N}{\|N\|} \quad \text{Vecteur normal à } S$$

Courbe sur la surface

$$C_s = (s \rightarrow X(s))$$

$$t' = (t' \cdot n)n + d$$

$\|d\|$  Courbure géodésique de la courbe

$d=0$  : La courbe est une **géodésique**

### ✓ En d'autres termes:

*La courbure géodésique en X d'une courbe sur une surface est égale à la courbure (2D) de la projection de cette courbe sur le plan tangent à la surface en X*

On en déduit une méthode pour déplier une surface:

*On mesure et on conserve la courbure géodésique le long de courbes sur la surface.*

Distorsion

## Développement local (Bennis'01)

### ✓ Surface échantillonnée $\Rightarrow$ courbes isoparamétriques

### ✓ Pour chaque point M:

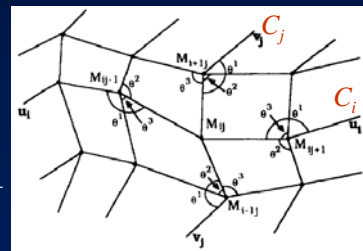
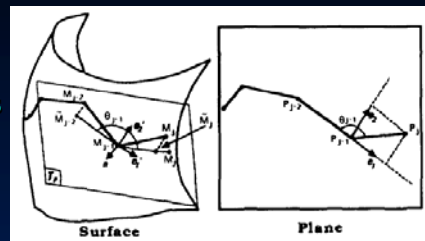
- Projection sur  $T_p(M)$
- Mesure des angles et distances
- Report sur la courbe dépliée

### ✓ Pareil pour les courbes transverses

### ✓ Courbes parallèles:

- On ne peut pas tout conserver
- Exemple de mesure de l'erreur le long d'une courbe:

$$E(C_i) = \frac{1}{N-1} \sum_{j=0}^{N-1} \frac{|d(M_{ij}, M_{ij+1}) - d(P_{ij} - P_{ij+1})|}{d(M_{ij}, M_{ij+1})}$$



Distorsion

## Relaxation

- ✓ Intégration de la courbure dans les deux directions: relaxation

$P_{\theta_{i,j}^r}^n$  : operateur de conservation des angles

$$M_{ij}^n = \frac{1}{12} \sum_{r=1}^3 \left( P_{\theta_{i-1,j}^r}^{n-1} + P_{\theta_{i,j-1}^r}^{n-1} + P_{\theta_{i+1,j}^r}^{n-1} + P_{\theta_{i,j+1}^r}^{n-1} \right) (M_{ij}^{n+1})$$

- ✓ Algorithme:

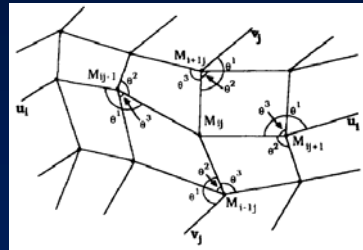
– Itérer jusqu'a ce que

$$\sum_{i,j} D_{sp}(P_{ij}) < \varepsilon$$

avec:

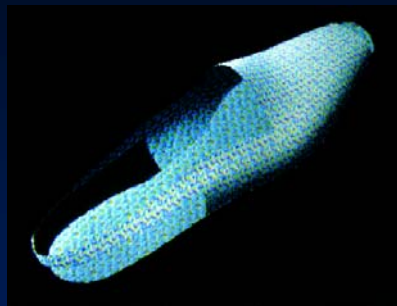
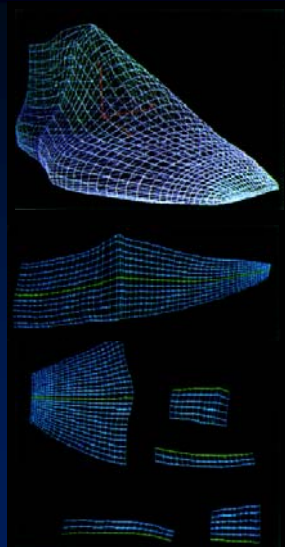
$$D_{sp}(P_{ij}) = \frac{1}{12} \sum_{r=1}^3 d(P_{ij}^n - P_{ij}^{n-1})^2 + \dots$$

(Dispersion)



Distorsion

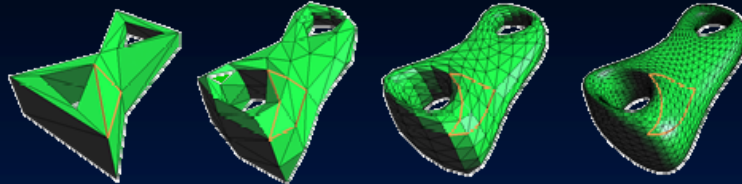
## Exemples



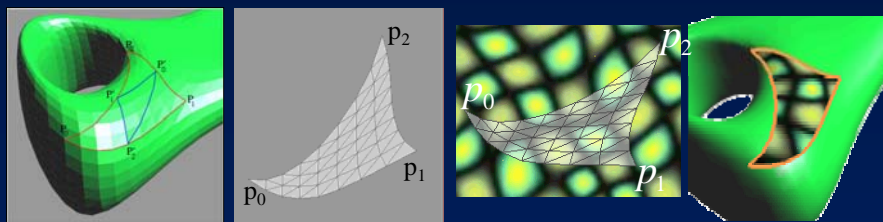
Distorsion

## Extension aux surfaces de subdivision

- ✓ On se base sur la définition récursive de la surface



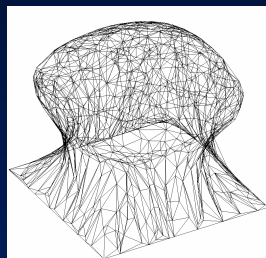
- ✓ Les points d'un niveau sont positionnés par rapport à leur parent dans l'espace texture.



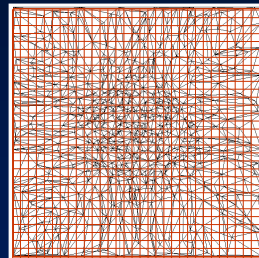
Distorsion

## Repartition globale des contraintes

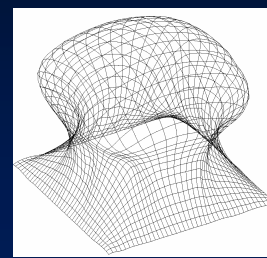
- ✓ Algorithme:
  - Trouver le meilleur plaquage du graphe des sommets qui minimise uniformément la distorsion
  - Les coordonnées de texture de chaque triangle sont celles des points transformés



Entrée  
(Espace 3D)



Mapping  
(Espace texture)



Affichage des courbes  
isoparametriques (texture)

Distorsion

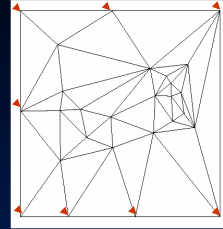
## Definition barycentrique (Floater'97)

- ✓ Expression des contraintes par combinaison convexe

$u_i$  Fixe pour  $i = n+1, \dots, N$

$$u_i = \sum_{j=1}^N \lambda_{ij} u_j \quad i = 1, \dots, n$$

$$\sum_{j=1}^N \lambda_{ij} = 1 \quad \text{Combinaison convexe} \quad \lambda_{ii} = 0$$



- On choisit les  $\lambda_{ij}$  pour minimiser la déformation
- On résout le système

$$u_i - \sum_{j=1}^n \lambda_{ij} u_j = \sum_{j=n+1}^N \lambda_{ij} u_j \quad (I - A)u = U$$

» Matrice  $I-A$  à diagonale dominante

» Toujours inversible ! ( $\rho(A) < 1$ ). On peut donc aussi résoudre itérativement:

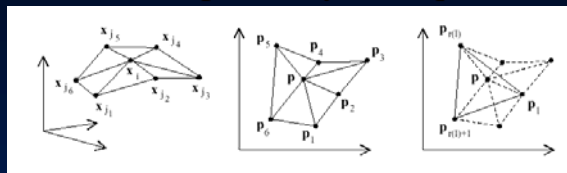
$$u^{n+1} = Au^n + U \quad u^0 = [0, \dots, 0]$$

Distorsion

## Definition barycentrique (Floater'97)

- ✓ Choix des coefficients

- Projection sur le plan tangent
- Choix d'un repère barycentrique



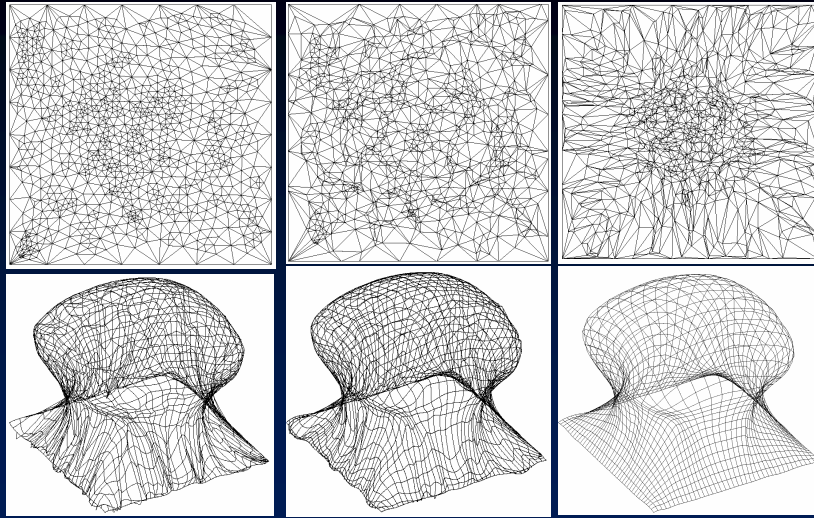
- Calcul des coordonnées: barycentre de 3 points

$$\lambda_{i,j_2} = \frac{\text{Aire}(p, p_1, p_3)}{\text{Aire}(p_1, p_2, p_3)} \quad \lambda_{i,j_1} = \frac{\text{Aire}(p, p_2, p_3)}{\text{Aire}(p_1, p_2, p_3)}$$

$$\lambda_{i,j_3} = \frac{\text{Aire}(p, p_1, p_2)}{\text{Aire}(p_1, p_2, p_3)} \quad \lambda_{i,j} = 0 \quad j \notin \{j_1, j_2, j_3\}$$

Distorsion

## Exemples divers



Iso barycentre

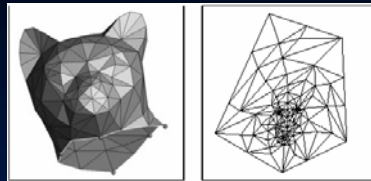
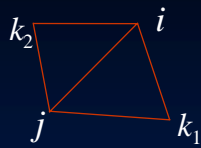
Conservation des distances

Barycentre (3 pts)

Distorsion

## Harmonic maps (C.f Eck'95)

✓ Harmonic maps



– On minimise:

$$E_{\text{harm}} = \frac{1}{2} \sum_{\text{edge}=\{i,j\}} \kappa_{ij} d(h(P_i), h(P_j))^2 \quad (\text{Energie de distorsion})$$

avec:

$$\kappa_{ij} = \frac{L_{i,k_1}^2 + L_{j,k_1}^2 - L_{i,j}^2}{\text{Aire}(i, j, k_1)} + \frac{L_{i,k_2}^2 + L_{j,k_2}^2 - L_{i,j}^2}{\text{Aire}(i, j, k_2)}$$

Plaquage inverse

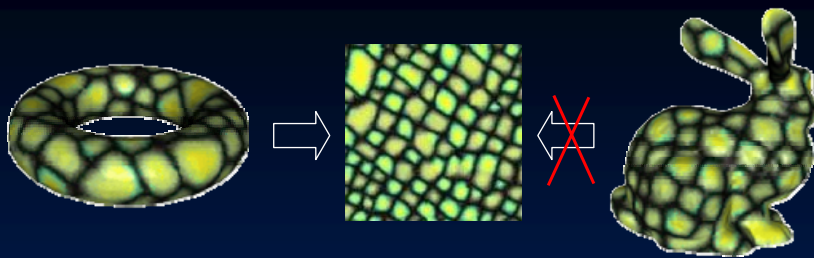
## Plan

---

- ✓ Synthèse 2D à partir d'un échantillon
- ✓ Plaquage sans distorsions (?)
- ✓ Synthèse sur surfaces
- ✓ Plaquage inverse

## Synthèse sur surfaces

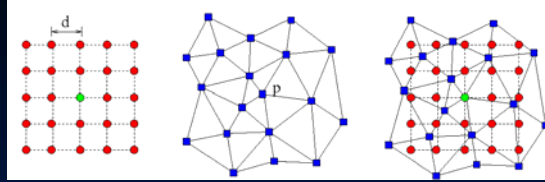
---



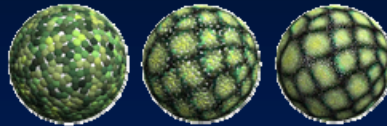
- ✓ Même si on sait minimiser localement les déformations...  
...la topologie n'est pas toujours compatible
- ✓ C'est un problème global (pas d'homéomorphisme)
- ✓ Solutions:
  - Synthèse de la texture *in situ* point par point
  - *Quilting* (collage)
  - *Pattern-based texturing* (précalcul des contraintes de collage)

## Synthèse pyramidale (1/2) (Wei'01,Turk'01)

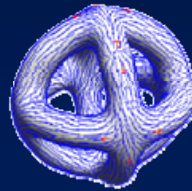
- ✓ Utilisation d'une
  - Pyramide de Gauss
  - Hiérarchie de points



- ✓ Produit une hiérarchie de points colorés



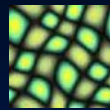
- ✓ Le remplissage utilise un champ de vecteurs
  - Création assistée
  - Parcours le long du champ



Synthèse surfacique

## Synthèse pyramidale (2/2) (Wei'01,Turk'01)

- ✓ Résultats



- ✓ Inconvénients
  - Ne produit pas une texture exportable

⇒ Nécessite un plaquage inverse ou un rendu special

Synthèse surfacique



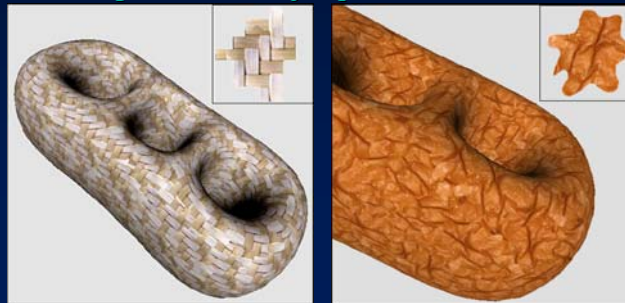
## 3D quilting : Lapped textures (Praun'00)

### ✓ Entrée:

- Une série de textures pré- découpées selon les discontinuités (utilisateur)
- Un maillage + un champ de vecteurs

### ✓ Algo:

- Des morceaux de texture sont collés dans la direction du champ de vecteurs jusqu'à recouvrement

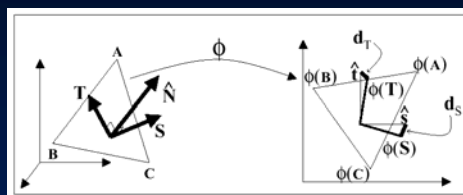


Synthèse surfacique

## 3D quilting : Lapped textures (Praun'00)

### ✓ Questions:

- **Plaquage local**
  - » Calcul du recouvrement
  - » Optimisation de la déformation



Système  
Linéaire  
creux

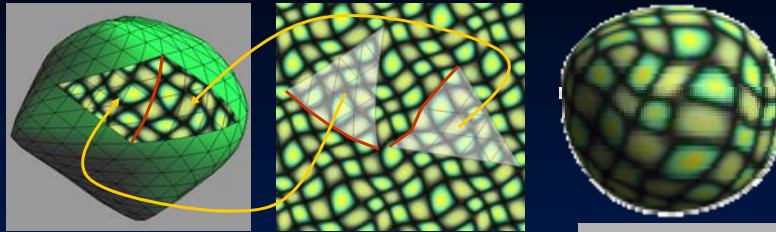
### - Rendu

- » En utilisant le alpha
- » Par plaquage inverse

Synthèse surfacique

## 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



Synthèse surfacique

## Pattern texture mapping (Neyret'99)

- ✓ Algorithme:

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



- ✓ Gestion des contraintes:



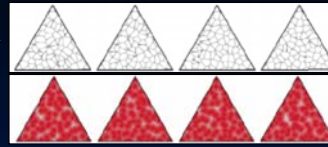
Cas général:

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

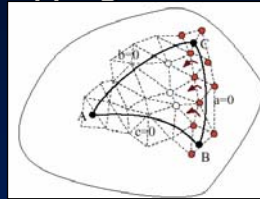
Synthèse surfacique

## Pattern texture mapping (Neyret'99)

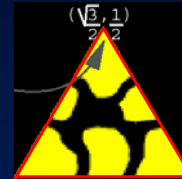
- ✓ Construction des tuiles
  - Algorithmes de Worley, de Perlin
  - Dessin à la main.
  - Transfert par pyramide ?



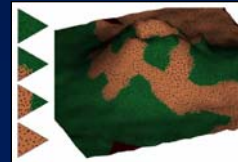
- ✓ Mapping



Coordonnées  
barycentriques



- ✓ Exemples
  - Deux conditions  
symétriques



Synthèse surfacique

## Plan

- ✓ Synthèse 2D à partir d'un échantillon
- ✓ Plaquage sans distorsions (?)
- ✓ Synthèse sur surfaces
- ✓ **Plaquage inverse**

## Plaquage inverse

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



Plaquage inverse

## Découpage en régions (Eck'95)

- ✓ Partitionnement
  - Choisir un ensemble de graines:
    - 1 - Les régions doivent être homéomorphes à un disque
    - 2 - La frontière entre deux régions est connexe
    - 3 - Une frontière touche 2 ou 3 régions
  - Calcul:
    - » Distance géodésique discrète



- » Recherche de plus court chemin multi-sources dans un graph → variante de l'algorithme de Dijkstra



1 - surface originale

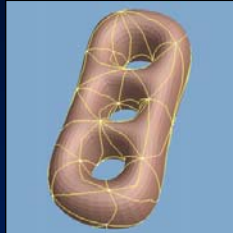


2 - diagramme de Voronoï

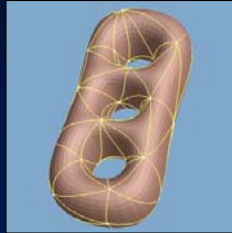
Plaquage inverse

## Découpage en régions (Eck'95)

- ✓ Dernière étape: triangulation de Delaunay
  - Utilisation des *harmonic maps* pour
    - » Calculer les trajets entre graines.
    - » Obtenir des trajets plus lisses



3 - triangulation de Delaunay primaire



4 - triangulation apres lissage



Plaquage inverse

## 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. Iglésias) 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

Les articles sont tous sur

<http://www-imagis.imag.fr/~Cyril.Soler/DEA/Textures/Papers/>