

# Visualisation-Multiresolution

RICM3 - Multimédia - Polytech-Grenoble

Damien ROHMER  
damien.rohmer@imag.fr

1er semestre 2008

# Visualisation-Multiresolution 1-Introduction

Polytech-Grenoble

1er semestre 2008

# Visualisation-Multiresolution

- Qu'est ce que la Visualisation ?

## Wikipedia dit

Visualization is any technique for creating images, diagrams, or animations to **communicate a message.**

## Visualisation de **données**

- Scientifiques (donnees abstraites ...)
- Physiques (fluides, ...)
- Medicales (Scanners rayons-X, IRM, ...)
- Techniques (Pieces mécaniques, ...)
- ...

# Quel type de données

## Types de données variés

- un champ scalaire (température, pression, ...)
- un champ vectoriel (vitesse, orientation, ...)
- un champ tensoriel (contraintes mécaniques, courbure, ...)

Définit on les données sur une surface, un volume ?

## Difficultés de leur représentation

- Données complexes (tenseurs, ...)
- Données trop nombreuses (paysage. ...)
- Bruit (données médicales, ...)

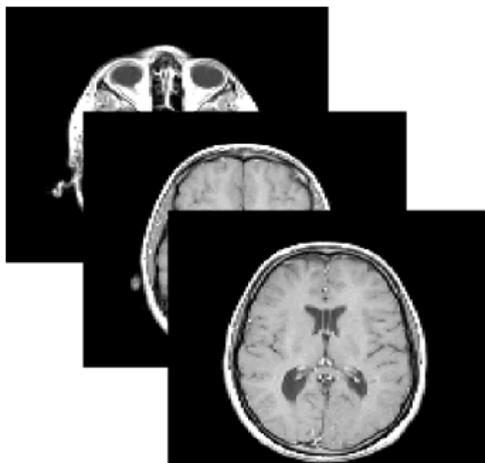
# Champ scalaire

Surface du domaine ou caracteristiques internes



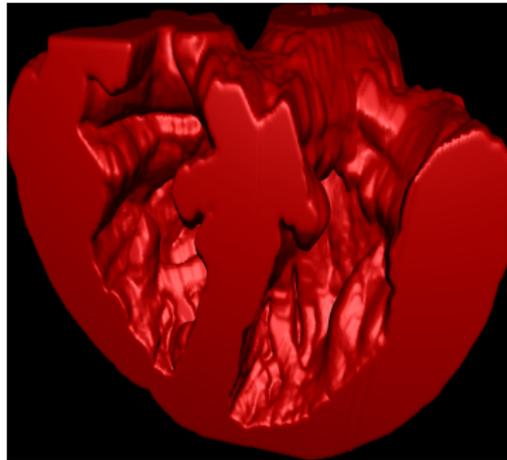
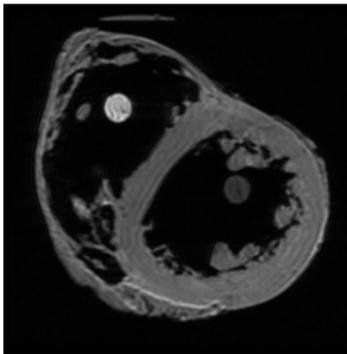
# Champ scalaire

Section 2D ou vue Volumique (isosurfaces, textures volumiques, ...)



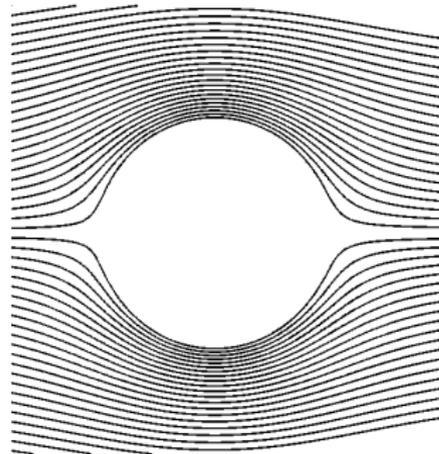
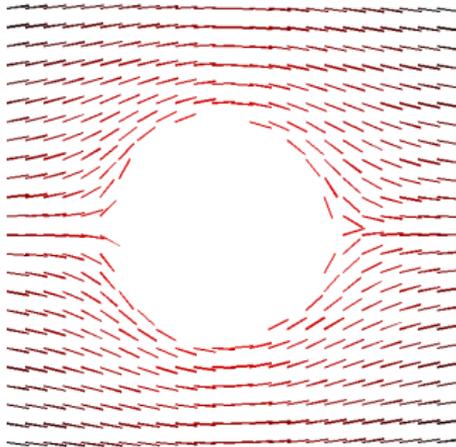
# Champ scalaire

Section 2D ou isosurfaces 3D



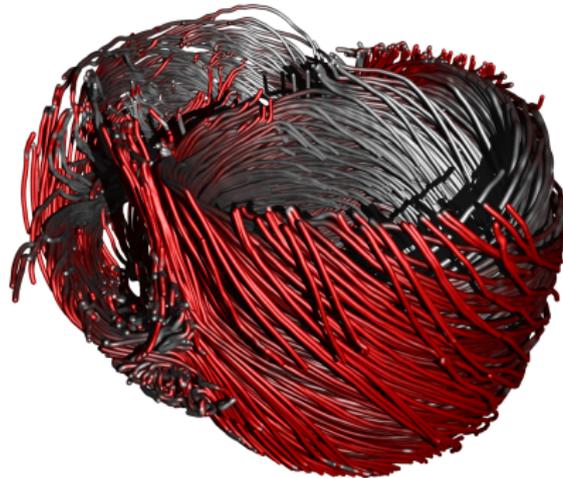
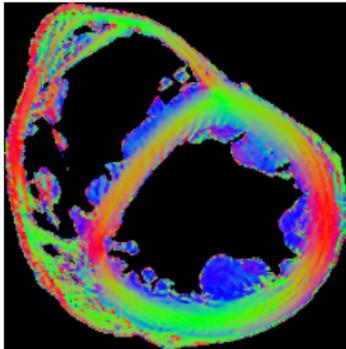
# Champ Vectoriel

## Vecteurs ou Trajectoire



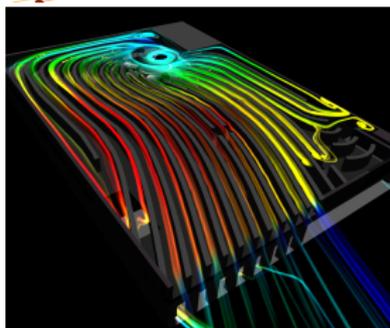
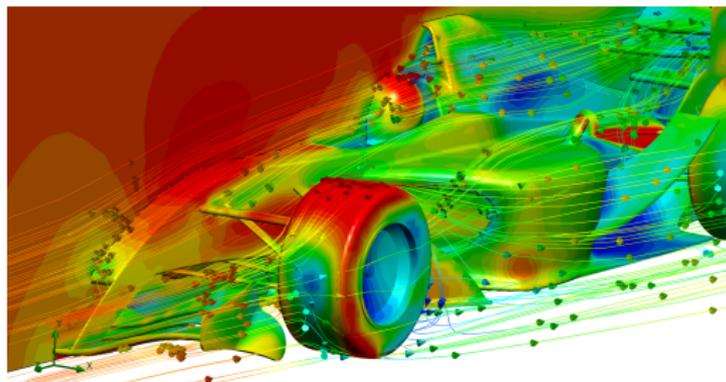
# Champ Vectoriel

Vecteurs ou Trajectoire (les lignes de flux peuvent être un objet réel)



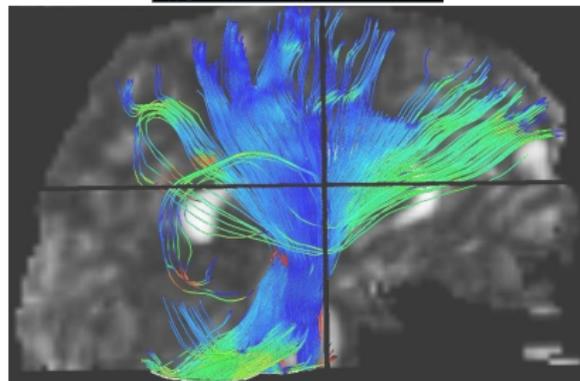
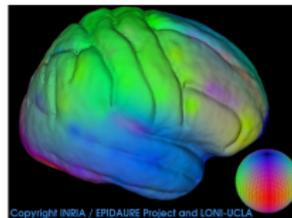
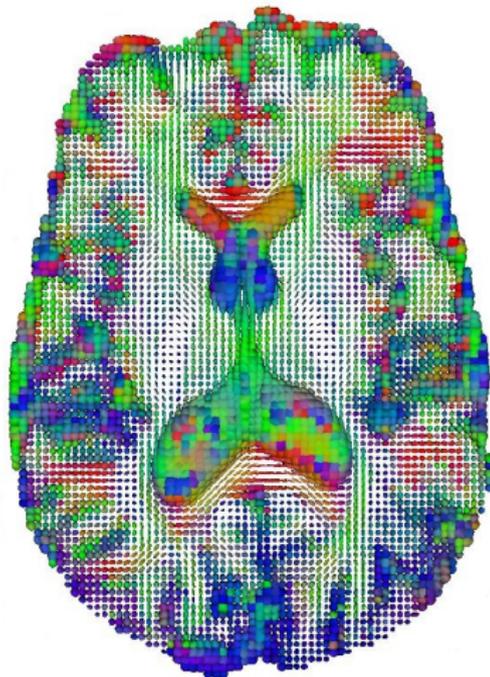
# Champ Vectoriel

Simulations physiques complexes (streamlines, hyperstreamlines, ...)



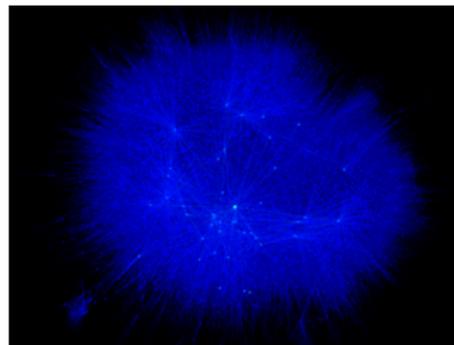
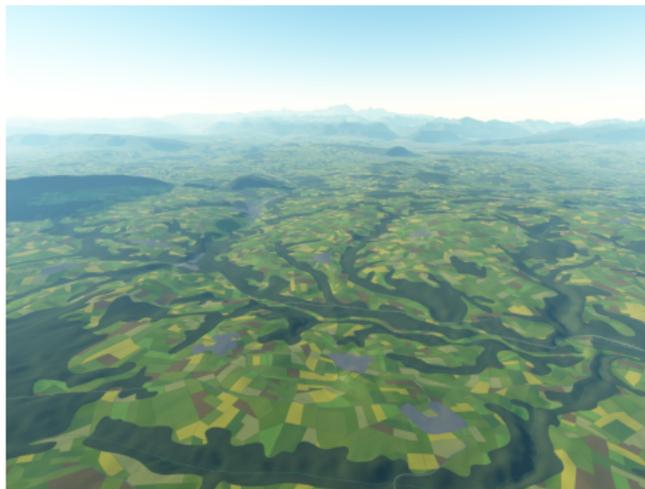
# Champ Tensoriel

Matrices symétriques  $3 \times 3$ . (Ellipsoïds, glyphs, orientation, fiber-tracking, ...)



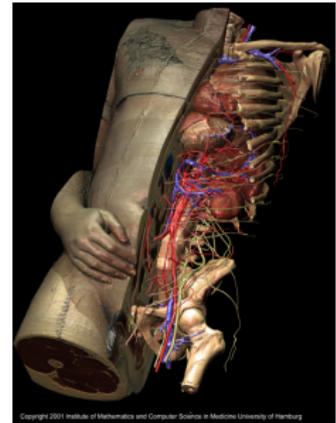
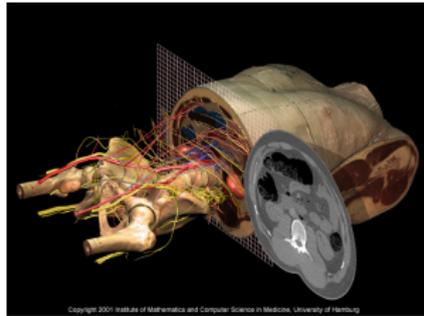
# Grand ensemble de Données

Les données physiques acquises sont souvent trop nombreuses !  
(Cartographies, Reseaux, ...)



# Grand ensembles de Données

## Visible Human Project 40GB (0.33mm)



# Difficultés

## Nécessité d'une multiresolution

- Affichage Progressif.
- Transmission progressive.
- Niveau de details (LOD).
- Controle hierarchique.

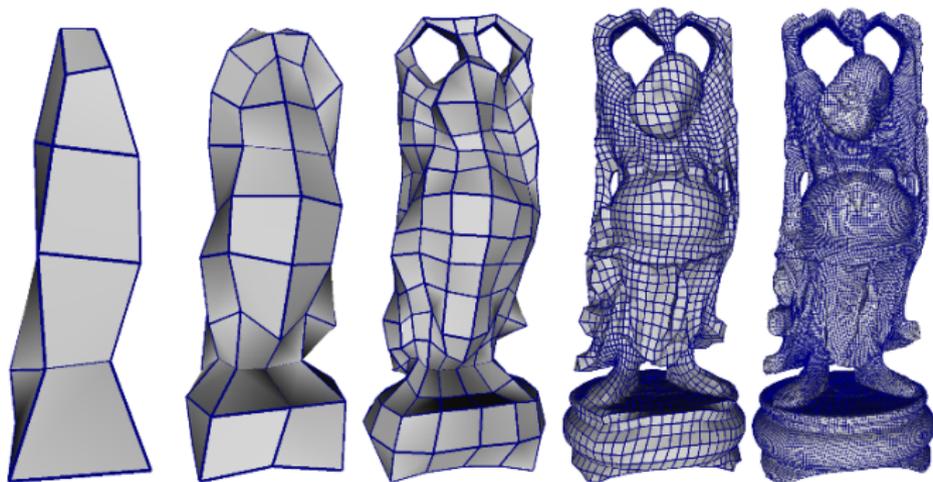
# Affichage Progressif



# Transmission Client-Serveur

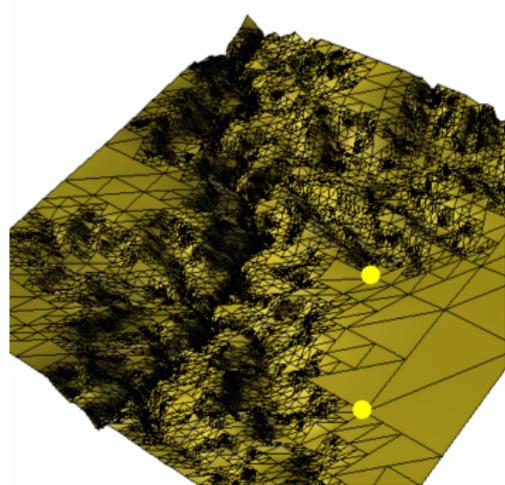
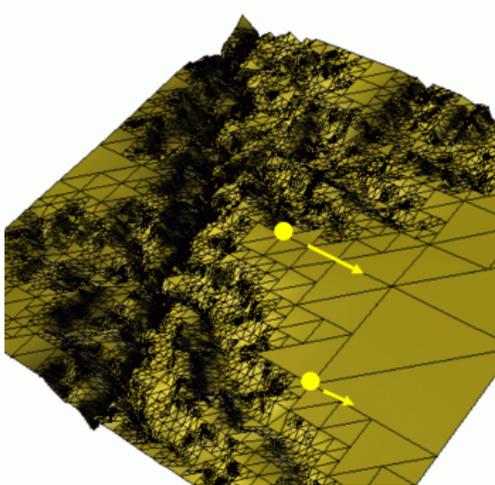
On envoi que les corrections au fur et à mesure.

- Chaque quadrilatere est divisé en quatre.



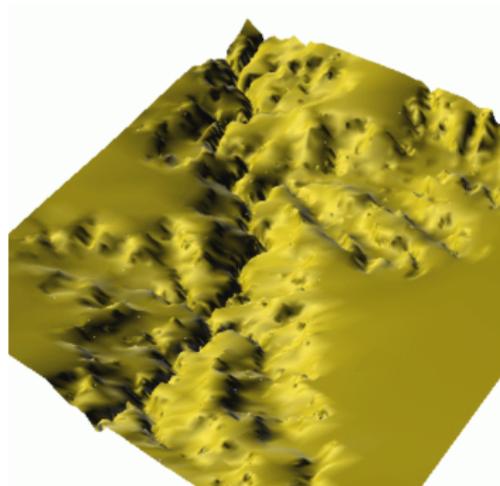
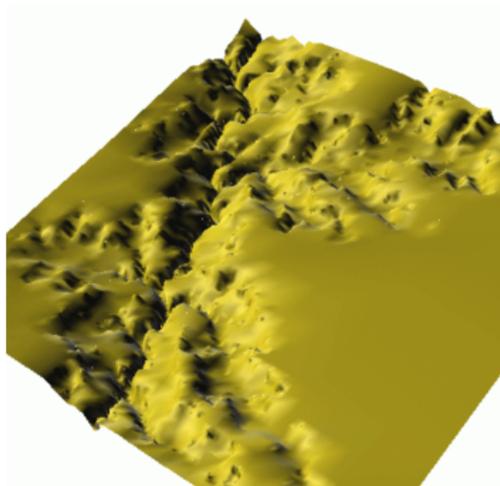
# Edition Hierarchique

Edition hierarchique de la forme en preservant les details.



# Edition Hierarchique

Edition hierarchique de la forme en preservant les details.



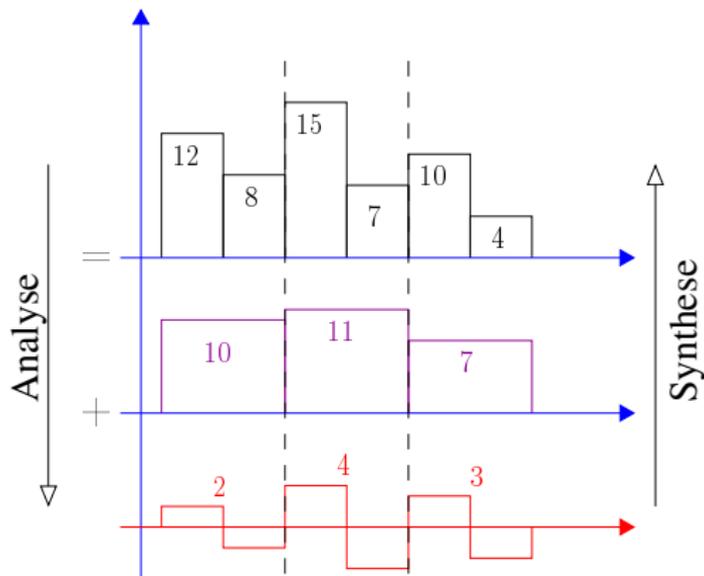
# Visualisation-Multiresolution 2-Ondelettes de Haar

Polytech-Grenoble

1er semestre 2008

# Principe

- Une étape descendante : Analyse, decomposition
- Une étape montante : Synthèse, reconstruction



# Principe

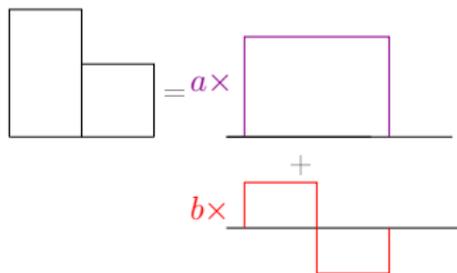
## ■ Action recursive

original	12	8	15	7	10	4	8	2
moyenne 1	10		11		7		5	
detail 1	2	-2	4	-4	3	-3	3	-3

original	10	11	7	5
moyenne 2	10.5		6	
detail 2	-0.5	0.5	1	-1

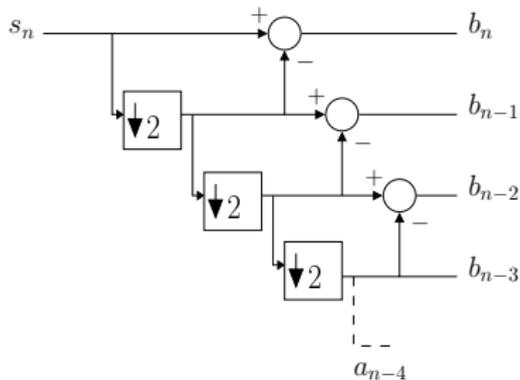
- $\text{original}_n = \text{moyenne}_{n-1} + \text{detail}_{n-1}$ .  
 $\Rightarrow \text{original}_n = \text{moyenne}_0 + \sum_{k=0}^{n-1} \text{detail}_k$

# Ondelette de Haar



$$s_n(k) = a_{n-1} m_{n-1} + b_n d_n$$

$$s_n(k) = a_0 m_0 + \sum_{i=0}^n b_i d_n$$



# Caractéristiques de la multiresolution

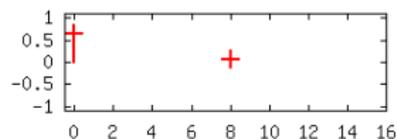
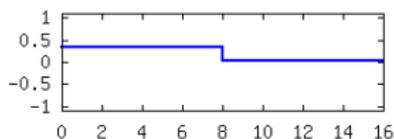
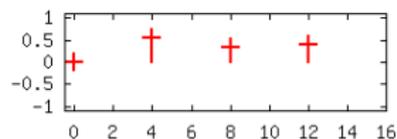
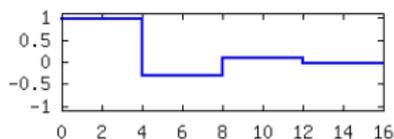
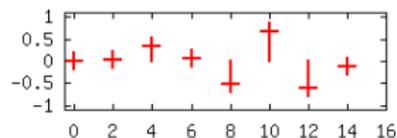
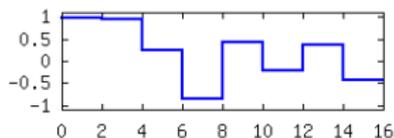
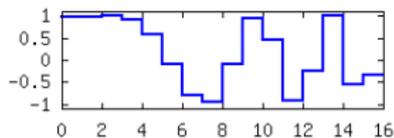
## Avantages :

- Temps de calcul lineaire  $\mathcal{O}(N)$  (plus rapide qu'une FFT ou un tri)
- Cout en mémoire constant

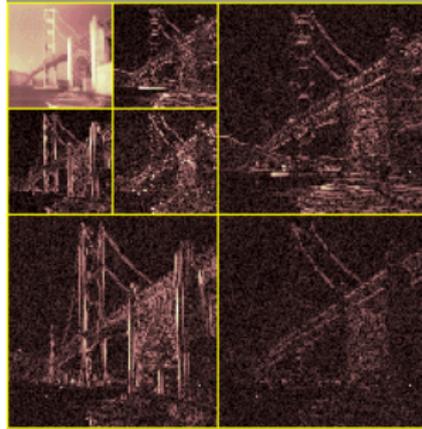
## Application type :

- Transmission progressive :
  - Serveur décompose et envoi les détails.
  - Client reconstruit à partir des détails de niveaux croissants.
- Compression (elimination des détails de faible amplitude (jpeg2000))

# Signal unidimensionnel



# Image 2D



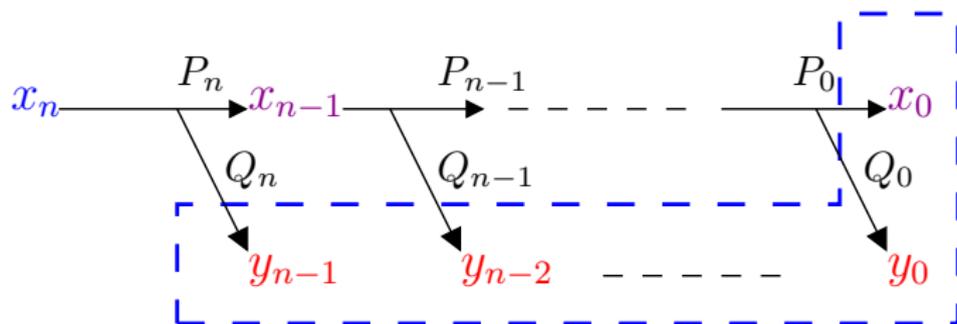
# Visualisation-Multiresolution 3-Multiresolution Chaikin

Polytech-Grenoble

1er semestre 2008

# Principe Général

■  $x_{n+1} = P_n(x_n) + Q_n(y_n)$



# Subdivision de Chaikin

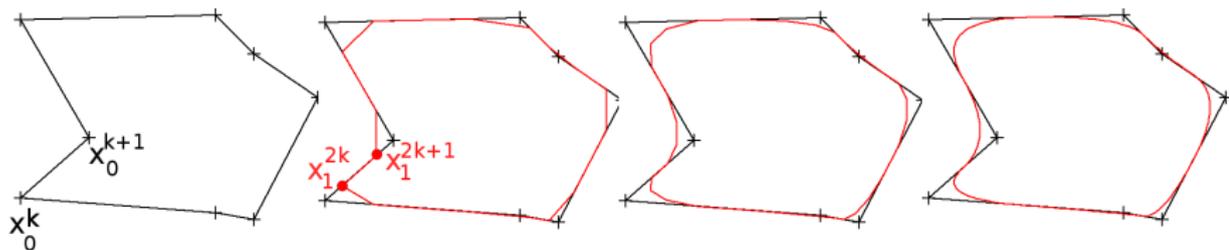
- Subdivision pour obtenir une courbe lisse

$$x_n^k = \begin{cases} \frac{3}{4}x_{n-1}^{\frac{k}{2}} + \frac{1}{4}x_{n-1}^{\frac{k}{2}+1} & k \text{ pair} \\ \frac{1}{4}x_{n-1}^{\frac{k-1}{2}} + \frac{3}{4}x_{n-1}^{\frac{k-1}{2}+1} & k \text{ impair} \end{cases}$$

Ou bien

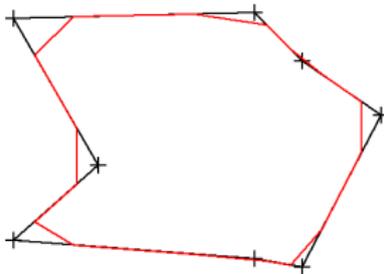
## Subdivision de Chaikin

$$\begin{cases} x_n^{2k} & = & \frac{3}{4}x_{n-1}^k + \frac{1}{4}x_{n-1}^{k+1} \\ x_n^{2k+1} & = & \frac{1}{4}x_{n-1}^k + \frac{3}{4}x_{n-1}^{k+1} \end{cases}$$



# Decomposition Multiresolution - Méthode de Chaikin

- Peut on faire l'inverse ?
- On a une courbe fine de  $2^n$  points.  
On recherche le "polygone grossier" ainsi que ces détails associés.



# Decomposition Multiresolution - Méthode de Chaikin

## ■ Inverser le système

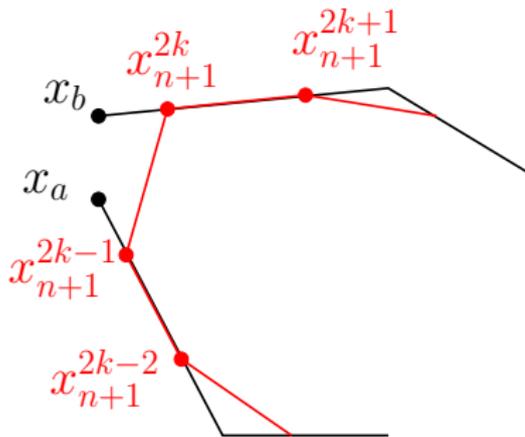
$$\begin{cases} x_n^{2k} &= \frac{3}{4}x_{n-1}^k + \frac{1}{4}x_{n-1}^{k+1} \\ x_n^{2k+1} &= \frac{1}{4}x_{n-1}^k + \frac{3}{4}x_{n-1}^{k+1} \end{cases}$$

Fournis 2 solutions possible  
pour  $x_n^k$

$$\begin{cases} x_a &= -\frac{1}{2}x_{n+1}^{2k-2} + \frac{3}{2}x_{n+1}^{2k-1} \\ x_b &= \frac{3}{2}x_{n+1}^{2k} - \frac{1}{2}x_{n+1}^{2k+1} \end{cases}$$

## ■ On prend la moyenne et on encode l'erreur

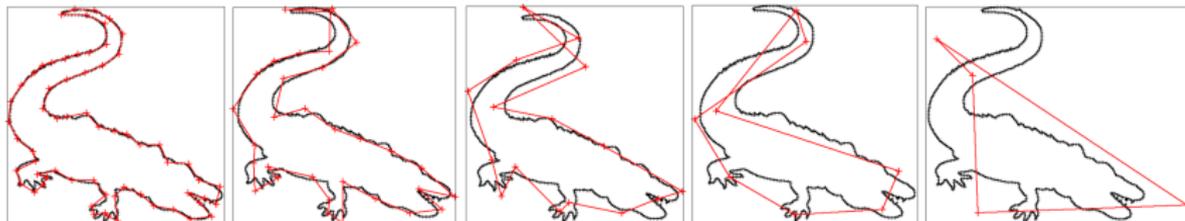
$$x_n^k = \frac{x_a + x_b}{2} \text{ et } y_n^k = \frac{x_b - x_a}{2}$$



# Decomposition Multiresolution - Méthode de Chaikin

## Décomposition

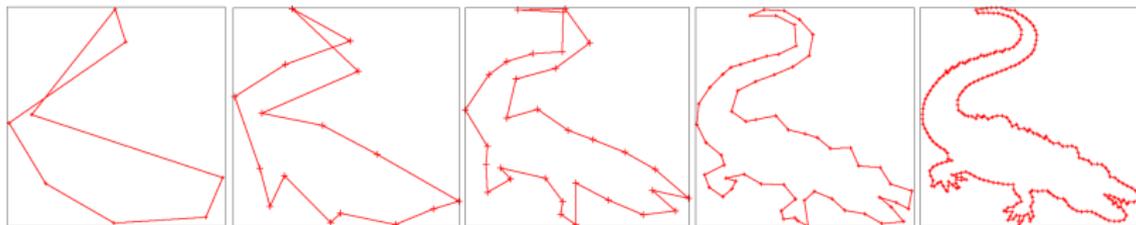
$$\begin{cases} x_n^k = \frac{1}{4} \left( -x_{n+1}^{2k-2} + 3x_{n+1}^{2k-1} + 3x_{n+1}^{2k} - x_{n+1}^{2k+1} \right) \\ y_n^k = \frac{1}{4} \left( x_{n+1}^{2k-2} - 3x_{n+1}^{2k-1} + 3x_{n+1}^{2k} - x_{n+1}^{2k+1} \right) \end{cases}$$



# Decomposition Multiresolution - Méthode de Chaikin

## Reconstruction + details

$$\begin{cases} x_{n+1}^{2k} &= \frac{3}{4} \left( x_n^k + y_n^k \right) + \frac{1}{4} \left( x_n^{k+1} - y_n^{k+1} \right) \\ x_{n+1}^{2k+1} &= \frac{1}{4} \left( x_n^k + y_n^k \right) + \frac{3}{4} \left( x_n^{k+1} - y_n^{k+1} \right) \end{cases}$$



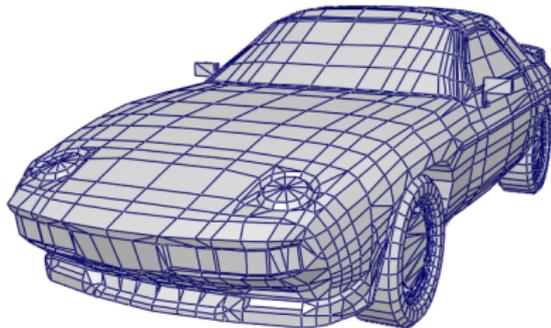
# Visualisation-Multiresolution 4-Courbes Splines

Polytech-Grenoble

1er semestre 2008

# CAD

- **Computer Aided Design**
  - Pierre Bézier, Renault
  - Paul de Casteljau, Citroën

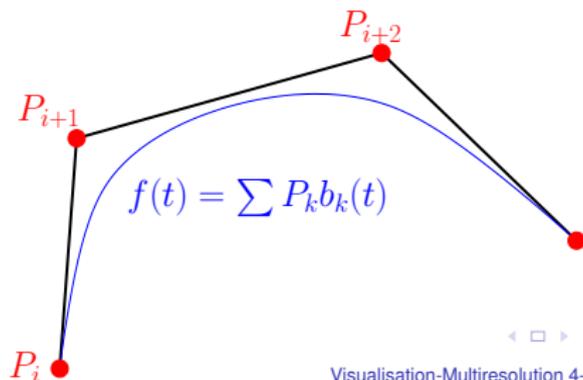


# Expression Analytique

- Etant donné un ensemble de positions discrètes, obtenir une expression analytique d'une courbe lisse.
- On recherche une base de fonction  $b_i$

## Expression analytique

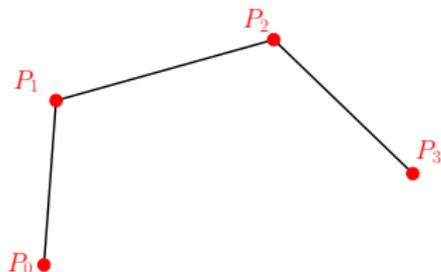
$$f(t) = \sum_k P_k b_k(t)$$



# Expression Analytique - base de fonctions

Exemples de fonctions pour  $N$  points ( $t \in [0, N]$ ).

- $b_k(t) = 1$
- $b_k(t) = \frac{1}{N}$
- $b_k(t) = \frac{1}{2} \prod(t - k)$
- $b_k(t) = \wedge(t - k)$



- Importance de la régularité des fonctions de bases.  
 $b_k \in \mathcal{C}^n \Rightarrow f \in \mathcal{C}^n$ .

# Bézier

- Bézier Propose les polynomes de Bernstein

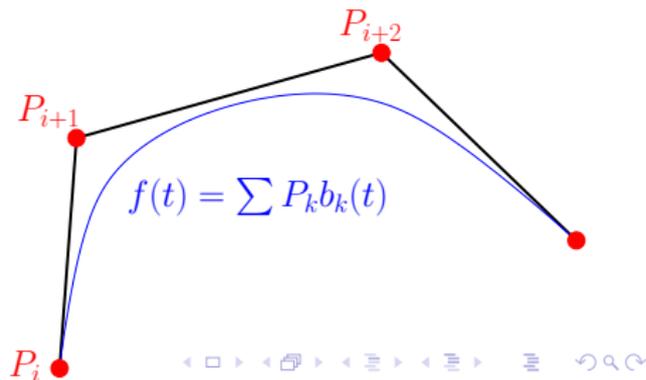
## Bernstein

$$\forall t \in [0, 1], b_k(t) = C_k^n t^k (1-t)^{n-k}$$

ex. Pour 4 points :

$$f(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3$$

- $f(0) = P_0$
- $f(1) = P_3$
- $f'(0) = 3\overrightarrow{P_0P_1}$
- $f'(1) = 3\overrightarrow{P_2P_3}$



# Théorème de Stone-Weierstrass

## Approximation de Stone-Weierstrass

$$\forall f \in C^0_{[a,b]}, \forall \epsilon > 0, \exists P \in \mathbb{P}; \forall x \in [a, b], |f(x) - P(x)| < \epsilon$$

Pour démontrer cela, on se place sur  $[0, 1]$ , et on utilise pour  $N$  suffisamment grand

$$P(x) = \sum_{k=0}^N C_k^N x^k (1-x)^{N-k} f\left(\frac{k}{N}\right)$$



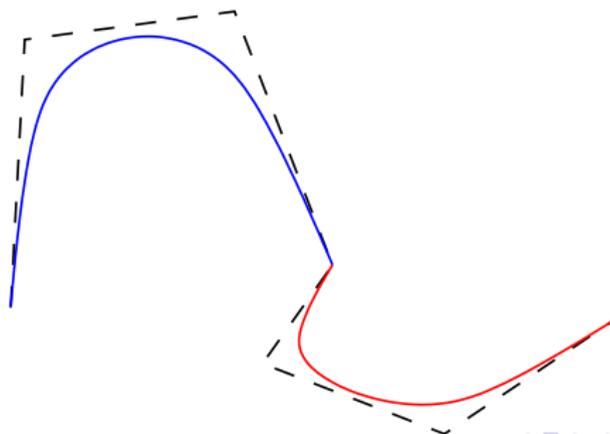
# Bézier

## Caractéristiques :

- $\forall x, \sum_k b_k(x) = 1$  invariance
- $\forall x, \forall k, b_k(x) \geq 0$  polygon convexe

## Défauts

- Le controle n'est pas locale
- Le raccord de courbes n'est pas trivial



# B-Spline

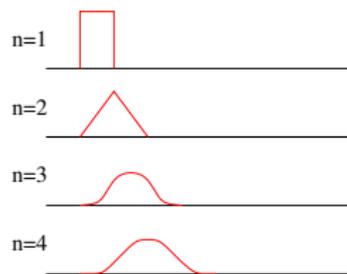
But :

Avoir une courbe  $C^k$ , avec un controle locale

Un vector de noeud  $[t_0, \dots, t_m]$

$$b_{j,0}(t) = \begin{cases} 1 & t_j \leq t < t_{j+1} \\ 0 & \text{sinon} \end{cases}$$

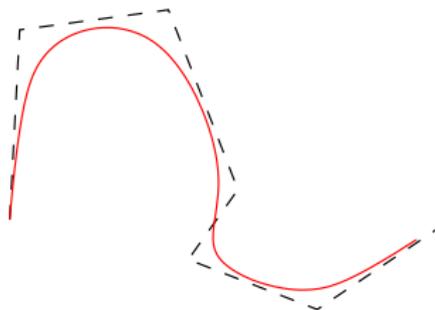
$$b_{j,n}(t) = \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t)$$



# B-Spline

Interet :

- Controle locale de la courbe.
- Raccord de courbe  $C^2$  pour des splines de degrés 3.

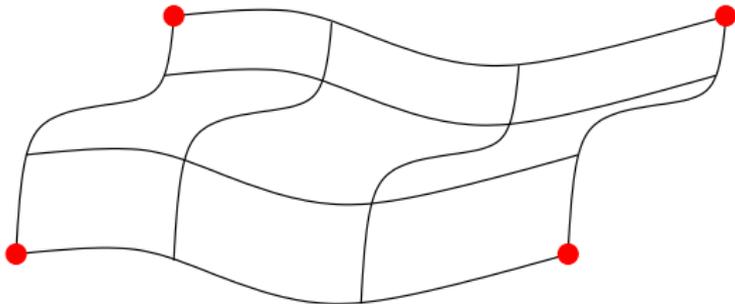


# Surfaces

- Fonctionnement 2D pour des surfaces  $\Rightarrow$  Produit tensoriel
- Grille de controles  $P_{jk}$

## Produit tensoriel

$$f(u, v) = \sum_j \sum_k b_j(u)b_k(v)P_{jk}$$



# Surfaces

On utilise un degré 3 pour une continuité  $C^2$ .

- Mise sous forme Matricielle

$$f(u, v) = (u^3 u^2 u 1) M [P_{jk}] M^T (v^3 v^2 v 1)^T$$

- Avec, pour une B-spline dont le vecteur de noeud est uniforme

$$M = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$



# Visualisation-Multiresolution 5-Surfaces de Subdivision

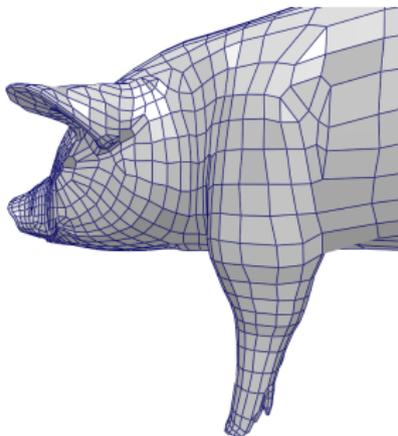
Polytech-Grenoble

1er semestre 2008

# Propriété du maillage

But : On possède un maillage, peut-on le subdiviser, avoir une multiresolution ?

- Propriétés du maillage
  - Variété ?
  - Face régulières ? Valence des sommets ?



# Topologie

## topologie

Topologie = Étude des déformations spatiale par des transformations continues

- 2 surfaces sont topologiquement équivalentes si on peut transformer l'une en l'autre par des transformations continues (pas de déchirures ni soudures).



# Variétés

- Une surface est une **2-variété** si tout point de la surface possèdent un voisinage homéomorphe à un disque (demi-disque).

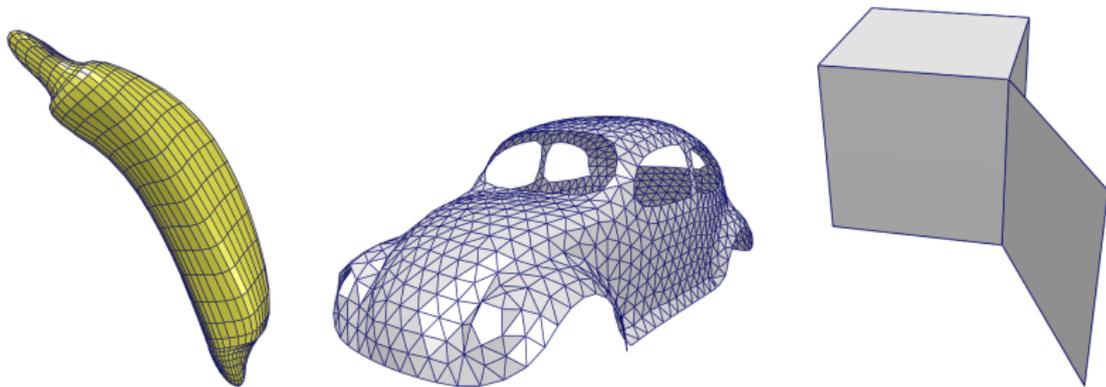


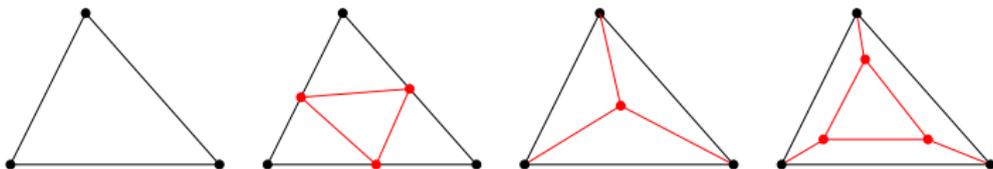
FIG.: Variété — Variété à bord — Non-variété

# Structure de données adaptée

- Pour les variétés : *voir chap.3 - Geometric Modeling Based on Polygonal Meshes - Botsch, Pauly, Kobbelt, Alliez, Lévy, Bischoff, Rössl - Siggraph course note, 2007*

# Possibilités de subdivision

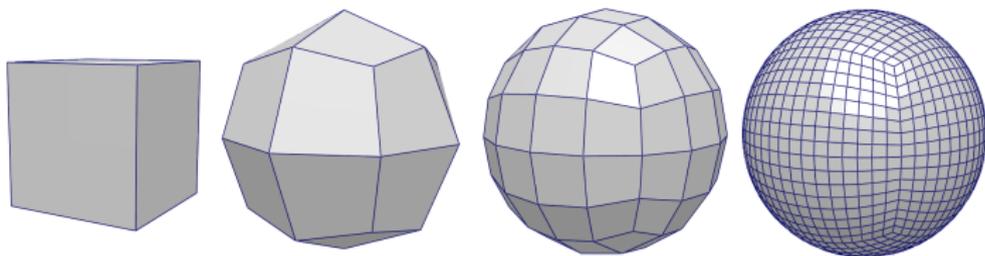
- Grille non structurée  $\Rightarrow$  Multiples possibilités de subdivision.



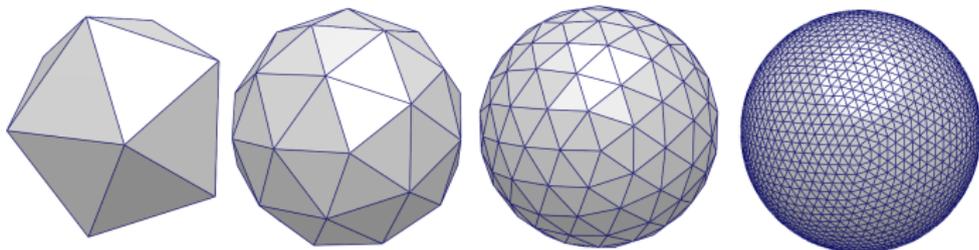
- Présence de quads, polygones quelconques ...

# Application : Construction d'une Sphère par projection

- On débute d'un cube



- On débute d'un icosaèdre



## Application au cas du lissage

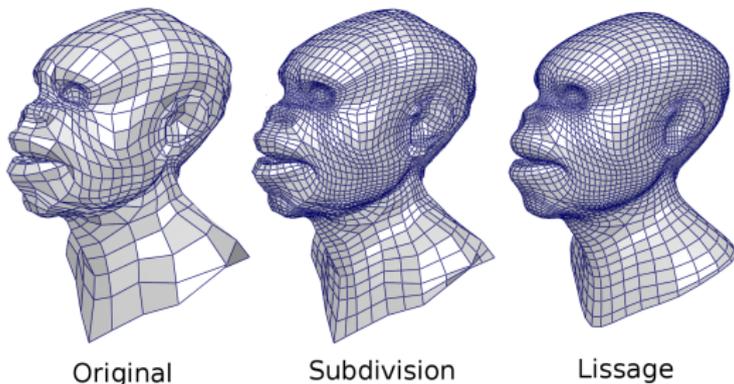
- On recherche à lisser la surface  $\mathbf{c}$  suivant

$$\frac{\partial \mathbf{c}}{\partial t} = \lambda \Delta \mathbf{c}$$

- En discretisant temporellement

$$\mathbf{c}^{i+1} = \mathbf{c}^i + \Delta t \lambda \Delta \mathbf{c}^i$$

- Problème : Calcul du Laplacien sur un maillage.



# Calcul du Laplacien

- Laplacien :

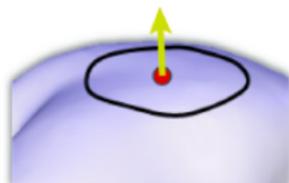
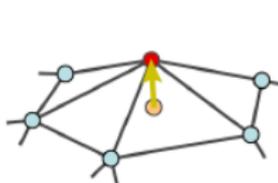
$$\Delta f(x, y, z) = \operatorname{div} \operatorname{grad}(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$

- Sur une variété :

$$\Delta c = \lim_{r \rightarrow 0} \frac{\nabla A}{A} = 2 \kappa \mathbf{n}$$

- Sur un maillage :

$$\Delta(\mathbf{x}_i) \simeq \frac{1}{N} \sum_{j \in \mathcal{V}_i} (\mathbf{x}_j - \mathbf{x}_i) = \text{bar} - \mathbf{x}_i$$

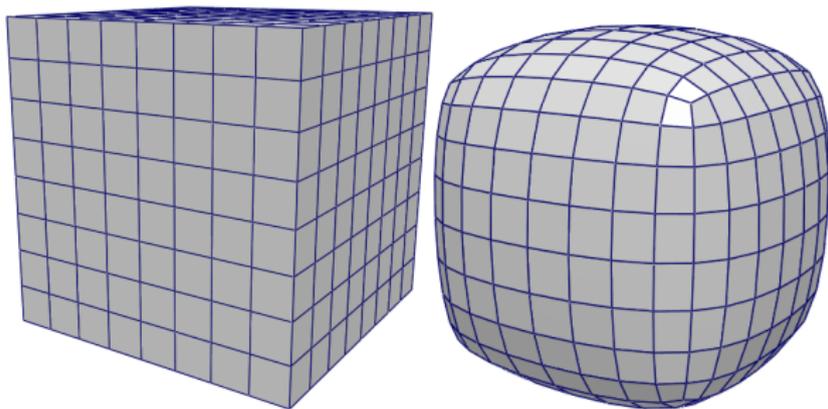


# Lissage

- Au final : Filtrage d'un maillage de la surface  $\mathbf{c}$  :

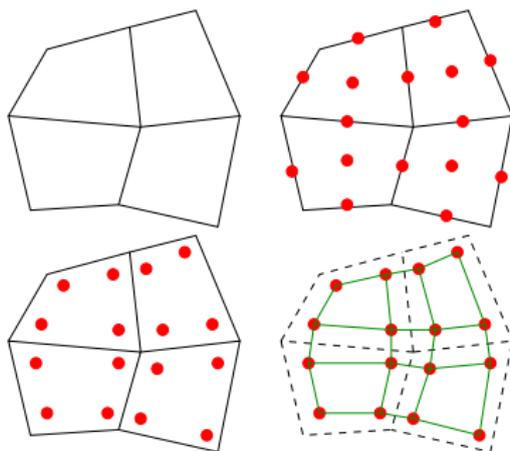
$$\frac{\partial \mathbf{c}}{\partial t} = \lambda \Delta \mathbf{c}$$

$$\Rightarrow x_i^{k+1} = (1 - \mu) x_i^k + \mu \text{bar}$$



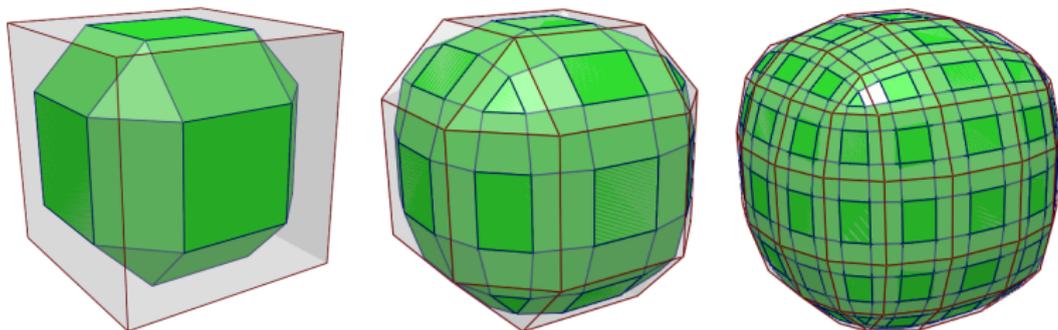
# Corner-cutting

- Peut on simuler le corner cutting pour un maillage ? Doo-Sabin
  - Étant donnée une face formée par les sommets  $(x_i)_{i=[0,N]}$
  - Construire les sommets milieu  $m_i = \frac{x_i+x_{i+1}}{2}$
  - Et le barycentre de la face  $b = \frac{1}{N} \sum_{i=1}^{i=N} x_i$
  - Les nouveaux sommets sont donnés par  $n_i = \frac{x_i+m_i+m_{i-1}+b}{4}$



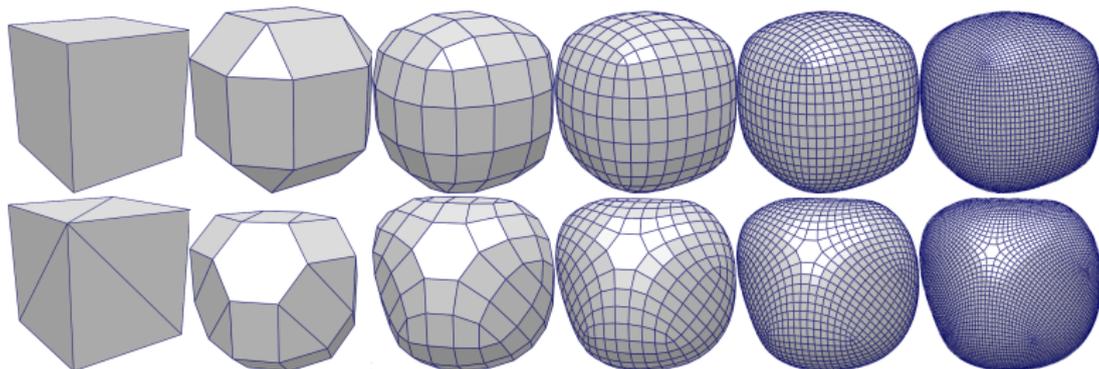
# Corner-Cutting

- Application en 3D sur des quads



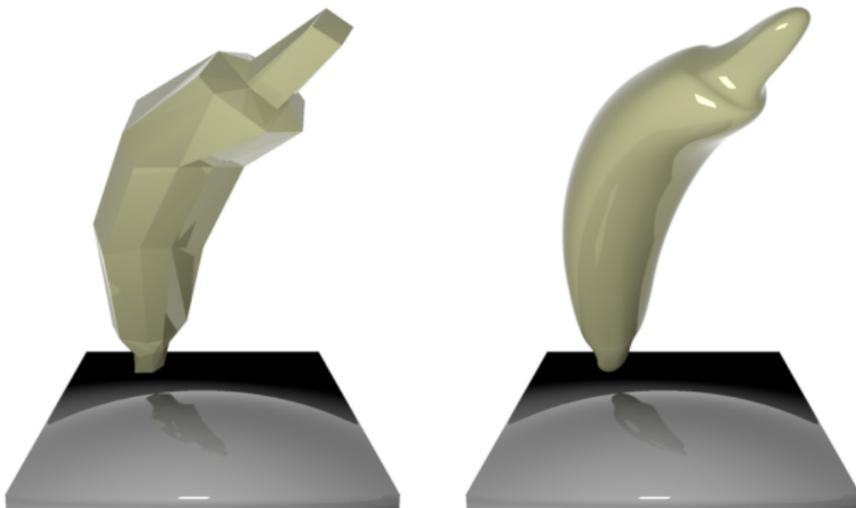
# Corner-Cutting

- L'aspect final dépend fortement du maillage du polygone de controle.



# Corner-Cutting

- Exemple de lissage + rendu
- Intérêt pour les films d'animations



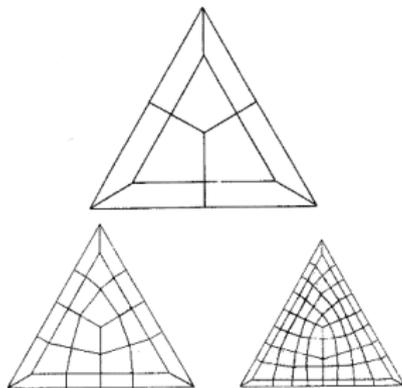
# Classements

- Subdivision Primale = Subdivision de faces
- Subdivision Duale = Subdivision de sommets

# Catmull-Clark

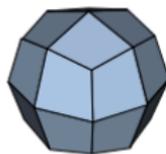
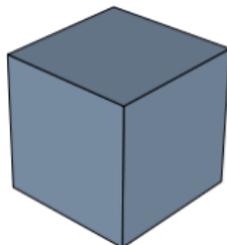
- Un sommet de face = barycentre de l'ancienne face
- Un sommet d'arête = moyenne entre le milieu des anciens sommets et le milieu des faces partageant l'arête
- Nouvelle position du sommet =  $\frac{Q+2R+S(n-3)}{n}$

- Q : Moyenne des sommets de face adjacents
- R : Moyenne des points milieux des arêtes incidentes
- S : Anciennes coordonnées
- n : valence



# Catmull-Clark Resultats

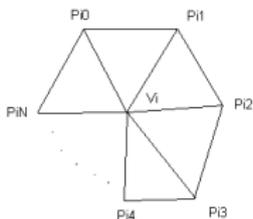
- $C^2$  sauf au sommets extra-ordinaires
- Schema d'approximation
- Subdivision de face
- Basé de préférence sur des quads



# Loop

## Maillage Triangulaire

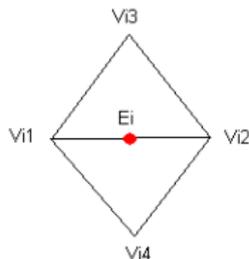
### ■ Nouvelle Position



$$V^{i+1} = (1 - n\alpha)V^i + \alpha \sum_{k=0}^n P_k$$

$$\alpha = \frac{1}{n} \left( \frac{5}{8} - \left( \frac{3}{8} - \frac{1}{4} \cos \left( \frac{2\pi}{n} \right) \right)^2 \right)$$

### ■ Sommets d'arêtes



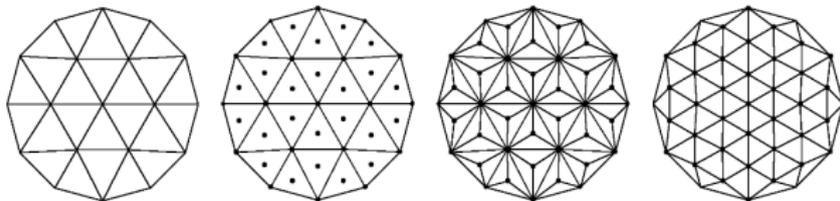
$$E^{i+1} = \frac{3}{8}(V_1 + V_2) + \frac{1}{8}(V_3 + V_4)$$

# Loop

- $C^2$  sauf au sommets extra-ordinaires
- Schema d'approximation
- Subdivision de face
- Subdivision de triangles

# $\sqrt{3}$ -Kobbelt

## Maillage Triangulaire



- Nouveaux sommets : barycentre de l'ancienne face
- Nouvelle position du sommet  $i$  :

$$(1 - \alpha_n)p_i + \frac{\alpha_n}{n} \sum_{j \in \mathcal{V}_i} p_j \quad (n=\text{valence}, \mathcal{V}=\text{voisinage})$$

$$\text{et } \alpha_n = \frac{4 - 2 \cos\left(\frac{2\pi}{n}\right)}{9}$$

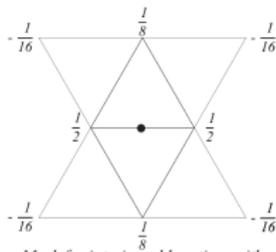
# $\sqrt{3}$ -Kobbelt

- $C^2$  sauf au sommets extra-ordinaires
- Schema d'approximation
- Subdivision de face
- Subdivision de triangles

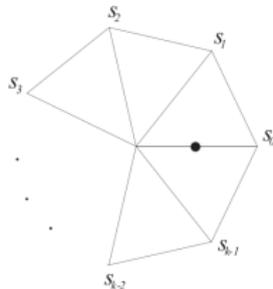
# Butterfly

## Maillage Triangulaire

### ■ Sommet régulier



### ■ Sommet non régulier



### ■ Ajout d'un point par coté

### ■ Dans le cas d'un sommet non régulier :

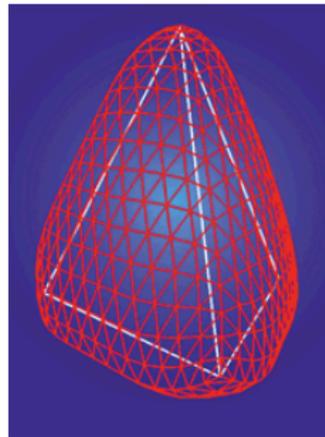
$$s_i = \frac{1}{k} \left( \frac{1}{4} + \cos \left( \frac{2\pi}{k} \right) + \frac{1}{2} \cos \left( \frac{4\pi}{k} \right) \right) \quad \text{pour } k > 5$$

$$s_0 = \frac{5}{12}, s_{1,2} = -\frac{1}{12} \quad \text{pour } k = 3$$

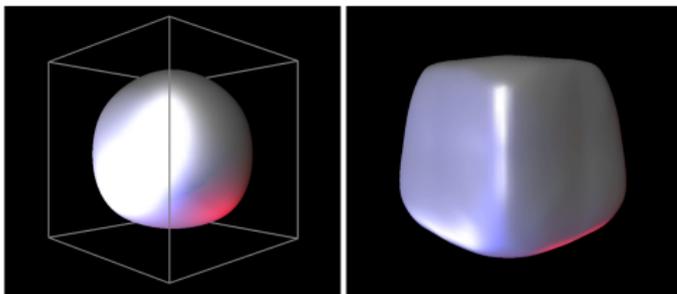
$$s_0 = \frac{3}{8}, s_2 = -\frac{1}{8}, s_{1,3} = 0 \quad \text{pour } k = 4.$$

# Butterfly

- $C^1$  sauf au sommets extra-ordinaires
- Schema d'interpolation
- Subdivision de face
- Subdivision de triangles

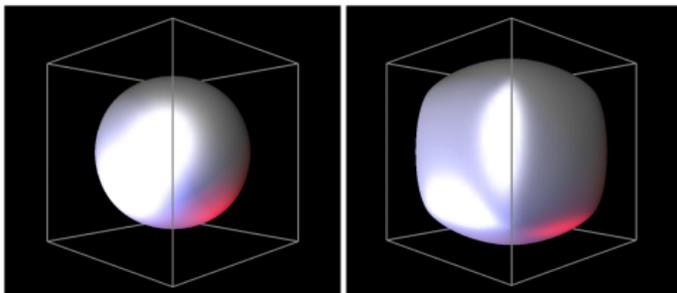


# Comparaison



*Loop*

*Butterfly*



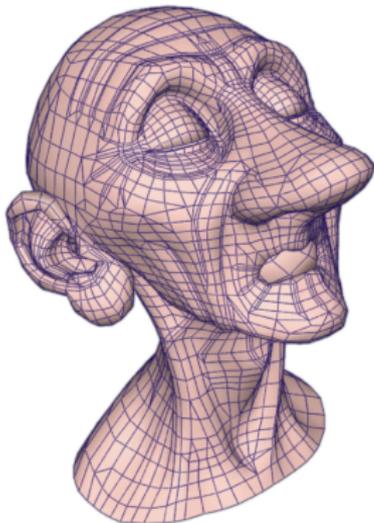
*Catmull-Clark*

*Doo-Sabin*

- voir - *Subdivision Surfaces in character animation - DeRose, Kobbelt, Levin, Sweldens - Siggraph course note 2000*

# Application dans des films d'animations

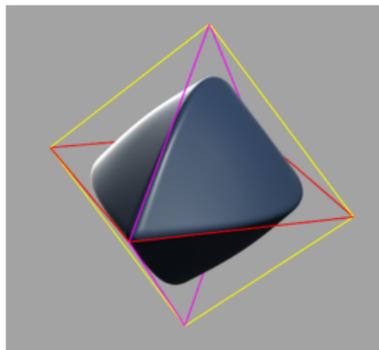
## ■ Geri's game (Pixar)



■ voir - *Subdivision Surfaces in character animation - De Rose, Kass, Truong - Siggraph 2008*

# Arêtes vives

- Nécessité de garder certaines arêtes vives.



# Multiresolution

- Ne s'applique pas directement pour des maillages quelconques
- Multiresolution "manuelle"
  - Duplication de données.
  - Intérêt pour l'edition hiérarchique

# Visualisation-Multiresolution 6-Visualisation, Introduction

Polytech-Grenoble

1er semestre 2008

- D'une façon générale, on visualise une fonction  $d$  **dimensionnelle**, dépendant de  $v$  **variables**, **plongé** dans un espace à  $n$  dimensions ( $n \geq v$ ).

$$\vec{f}(\vec{u}) = \begin{pmatrix} f_1(u_1, u_2, \dots, u_v) \\ f_2(u_1, u_2, \dots, u_v) \\ \vdots \\ f_d(u_1, u_2, \dots, u_n) \end{pmatrix}$$

Où  $u$  sont les coordonnées paramétrique d'un objet  $O$   $v$ -dimensionnel (courbe, surface, domaine volumique, ...)

Dans notre cas, on s'intéressera typiquement à  $n = 3$  ou  $2$  et où  $O$  est une variété.

## ■ Type de champ

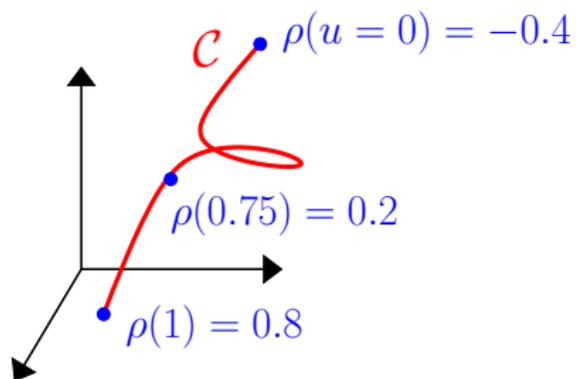
- Si  $d = 1$  : On parle de **champ scalaire** (densité, ...)
- Si  $d > 1$  : On parle de **champ vectoriel**
- rem.  $f$  peut être une matrice : par ex. champs de transformations, ou les champs de tenseurs.

## ■ Variables

- Si  $v = 1$  : On parle de **champ linéique** (ou unidimensionnel).
- Si  $v = 2$  : On parle de **champ surfacique**.
- Si  $v = 3$  : On parle de **champ volumique**.

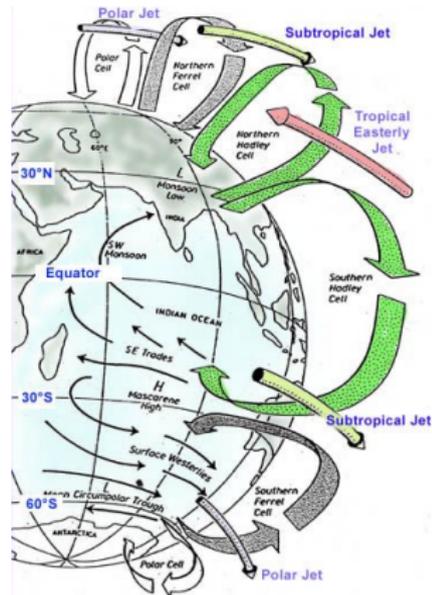
# Type de données (exemple I)

- Une densité de charge électrique sur un câble définie dans l'espace = Champ scalaire ( $d=1$ ) linéique ( $v=1$ ) plongé dans  $\mathbb{R}^3$ . On définit donc la densité par  $\rho(u)$  sur une courbe  $\mathcal{C} \subset \mathbb{R}^3$



# Type de données (exemple II)

- La direction du vent défini sur le globe terrestre à une hauteur de 50m au dessus du sol = Champ vectoriel ( $d=3$ ) surfacique ( $v=2$ ) plongé dans  $\mathbb{R}^3$ .
- La direction du vent défini sur le globe terrestre en dessous de 10000m = Champ vectoriel ( $d=3$ ) volumique ( $v=3$ ) plongé dans  $\mathbb{R}^3$ .



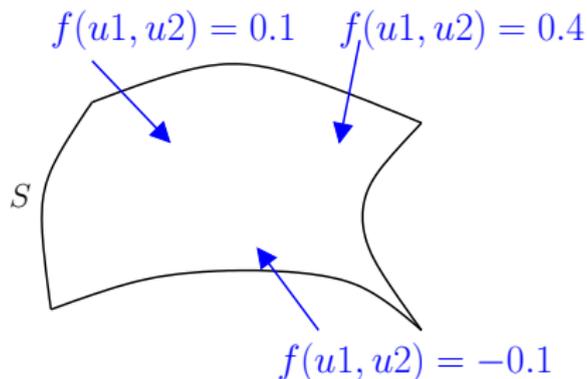
# Visualisation-Multiresolution 7-Données Scalaires Surfaceutiques

Polytech-Grenoble

1er semestre 2008

# Données scalaires surfaciques

- On désigne par  $S$  la surface sur laquelle est défini le champ paramétrée par  $(u_1, u_2)$ .
- On peut être plongé dans  $\mathbb{R}^3$ , ou dans  $\mathbb{R}^2$  ( $S$ =domaine).
- Le champ est du type  $f : (u_1, u_2) \mapsto f(u_1, u_2)$ .



# Données scalaires surfaciques

Visualisation classique :

- On affecte un **niveau de gris**
  - Que faire des données négatives
  - Nécessite une échelle
  - ...
- On affecte une “hauteur” (bump-mapping)
  - On déforme la surface sous-jacente
  - ...

# Données scalaires surfaciques, exemple

ex,  $S$  est un tore ( $v=2$ , plongement dans  $\mathbb{R}^3$ ) :

$$S : \begin{cases} S_x(u_1, u_2) = (R+r \cos(2\pi u_1)) \cos(2\pi u_2) \\ S_y(u_1, u_2) = (R+r \cos(2\pi u_1)) \sin(2\pi u_2) \\ S_z(u_1, u_2) = r \sin(2\pi u_1) \end{cases}, (u_1, u_2) \in [0, 1]^2$$

$$f(u_1, u_2) = \frac{\sin(2\pi u_1) + 1}{2} \cos^2(16\pi u_2)$$

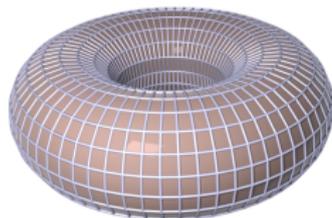
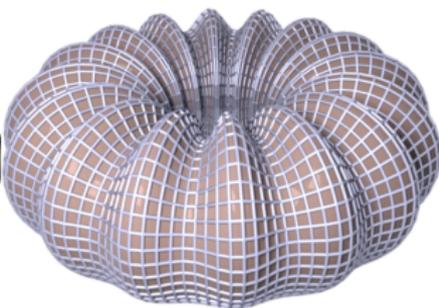
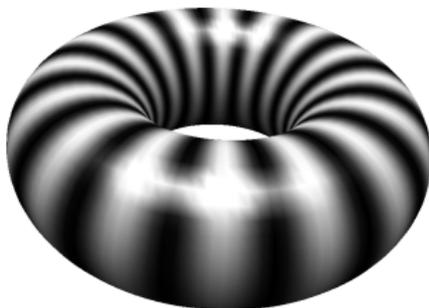


FIG.: Niveau de gris / déplacement normale

# Cas Particulier : Images

- Dans le cas où l'on se place dans  $R^2$ , et où  $S$  est un domaine rectangulaire, on parle d'**images** à niveaux de gris.
- Si on affecte une hauteur  $z$ , on parle de **height-map** (montagne).
- Si  $v = 3$ , on peut traiter scalairement en tant qu'**image couleur**.



FIG.: Image à niveau de gris / Image à trois composantes (R,G,B)

# Cas Particulier : Images.

- Exemple d'application d'un champ d'une image sur une 2-variété (tore)

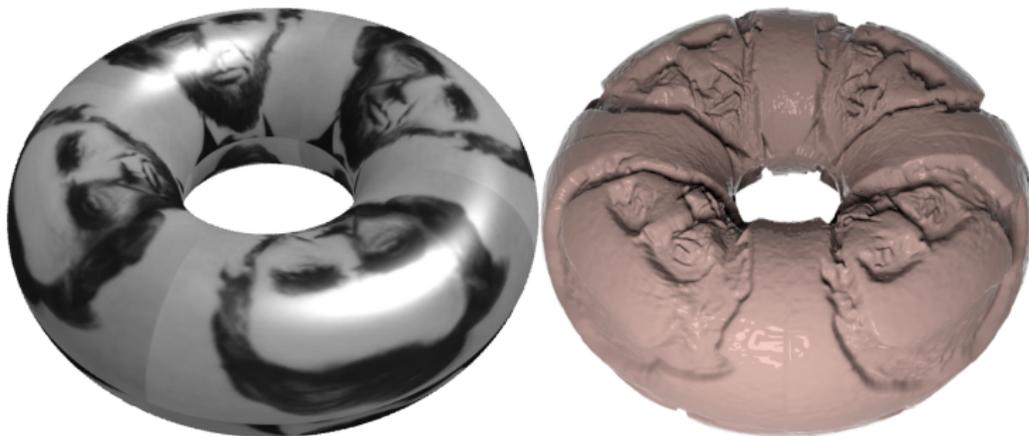


FIG.: Niveau de gris / Déplacement normale

# Cas Particulier : Texture

- Champ scalaire d'une image appliqué sur une surface = **Texture mapping**
- Directement intégré dans les cartes graphiques
- (déplacement = bump/displacement mapping)

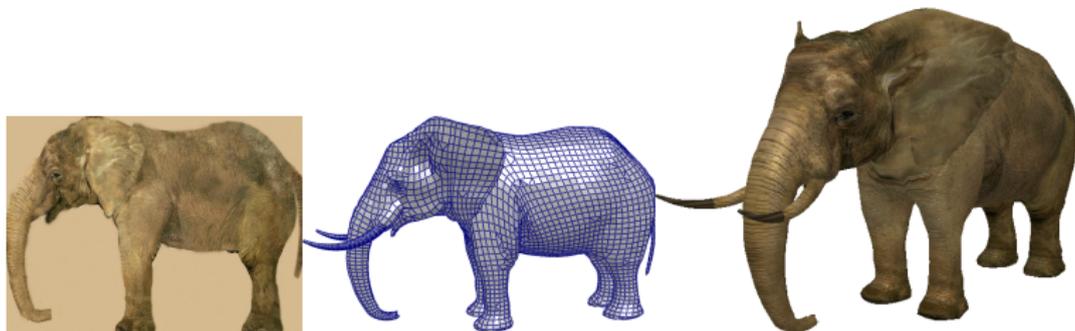


FIG.: Texture / surface  $S$  / objet texturé

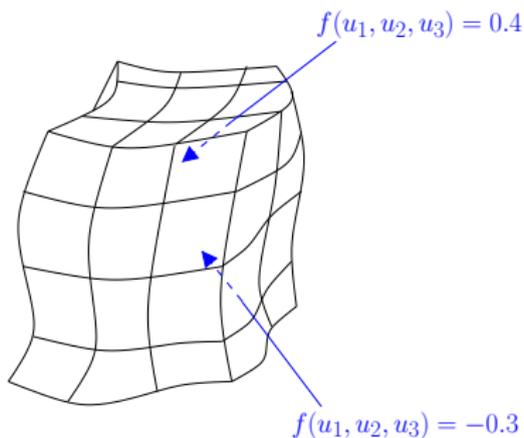
# Visualisation-Multiresolution 8-Données Scalaires Volumiques

Polytech-Grenoble

1er semestre 2008

# Données scalaires volumiques

- On désigne par  $V$  le volume sur lequel est définie le champ  $f$  dépendant de  $(u_1, u_2, u_3)$  (souvent une grille rectangulaire  $(u_1, u_2, u_3) = (x, y, z)$ ).
- On est plongé dans  $\mathbb{R}^3$ .



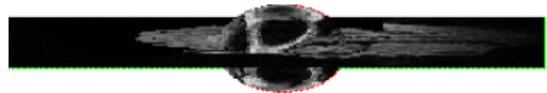
# Slicing

- Idée : On découpe des “tranches” de surfaces prédéfinie dans  $V$ .
- On colore la **densité** rencontré (niveau de gris, texture, ...)
- On affiche  $f(u_1 = \text{const}, u_2, u_3)$ ,  $f(u_1, u_2 = \text{const}, u_3)$ ,  
 $f(u_1, u_2, u_3 = \text{const})$ .



# Rendu sur variété

- On peut considérer des 2-variétés quelconques
- Question : Comment choisir la surface



# Isosurface

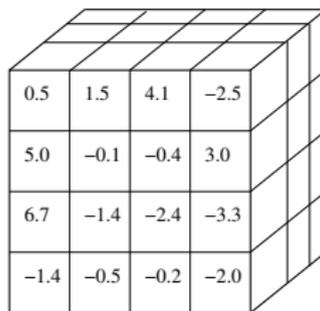
- Une surface particulière souvent utilisée : l'**Isosurface**  
Isosurface d'isovaleur  $a$  de la fonction  $f$  est

$$\{(x, y, z) \in \mathbb{R}^3 \mid f(x, y, z) = a\}$$

- On fait évoluer  $a$  pour obtenir différentes surfaces
- Comment construire une surface triangulée ?

# Marching-Cube : Introduction

- But : Construire une surface triangulée à partir d'un champ volume discret donnée par  $f(x, y, z) - a$ .
- Premier brevet logiciel en infographie en 1985 par Lorensen and Cline.
- Données d'entrées : Grille 3D suivant  $(x, y, z)$  de  $(N_i, N_j, N_k)$  sommets.



0.5	1.5	4.1	-2.5
5.0	-0.1	-0.4	3.0
6.7	-1.4	-2.4	-3.3
-1.4	-0.5	-0.2	-2.0

# Marching-Cube : Principe

- On parcourt cube à cube
- On calcule le signe de  $f(x_i, y_j, z_k) - a$
- On considère les différents cas possibles
- La valeur 0 est obtenue par interpolation

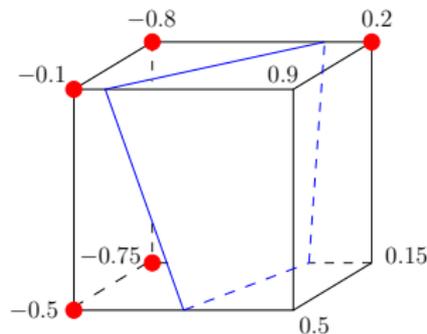
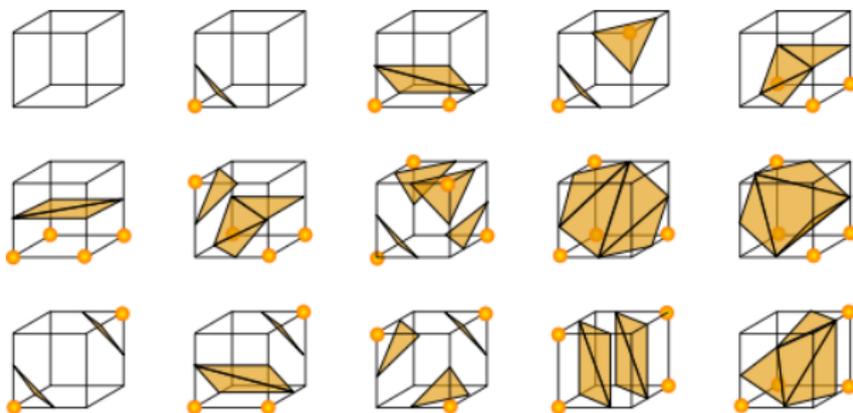


FIG.: Exemple d'un cas

# Marching-Cube : Différents Cas

- En tout : 256 cas possible
- Se ramène à 15 cas de bases (on retrouve les 256 par rotation)

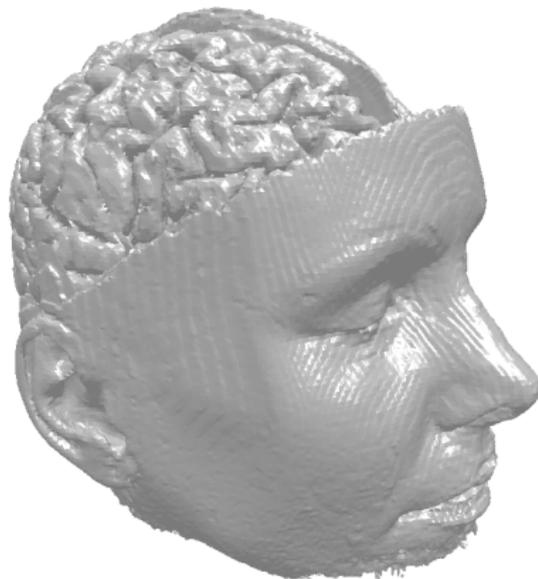
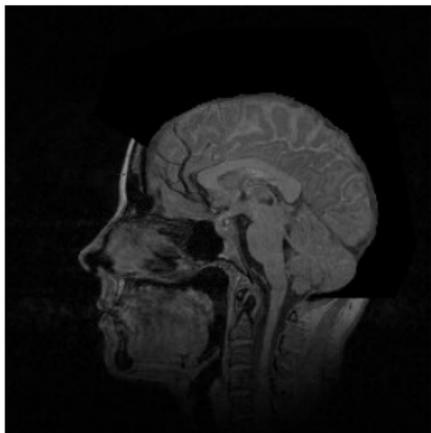


# Marching-cube : Avantage-Inconvénients

- ⊕ Rapidité d'exécution
- ⊖ Aspect cubique
  - Lissage du volume
  - Lissage de la surface finale
  - Adéquation médicale ?
- ⊖ Cas litigieux

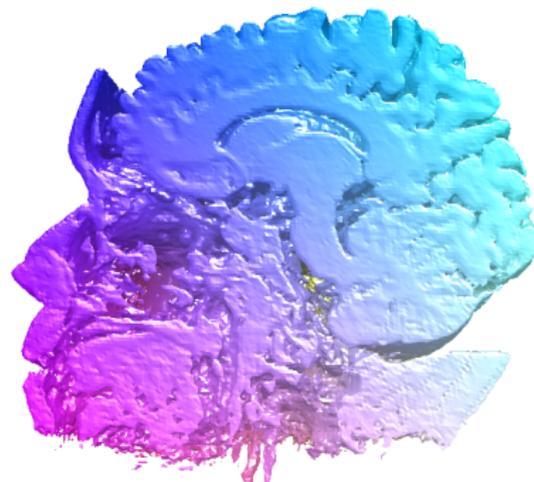
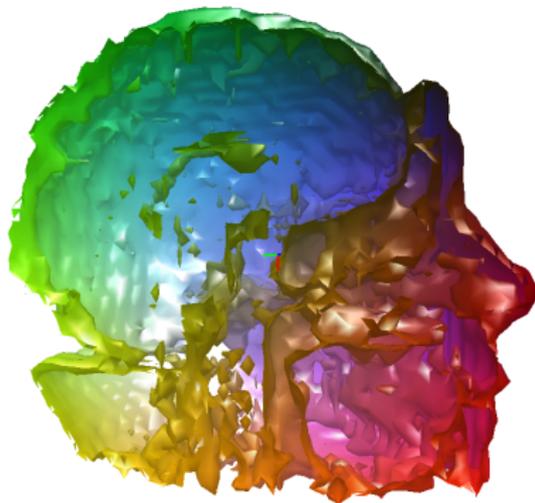
# Exemple d'isosurface : IRM

- Données IRM ( $256 \times 256 \times 99$ )



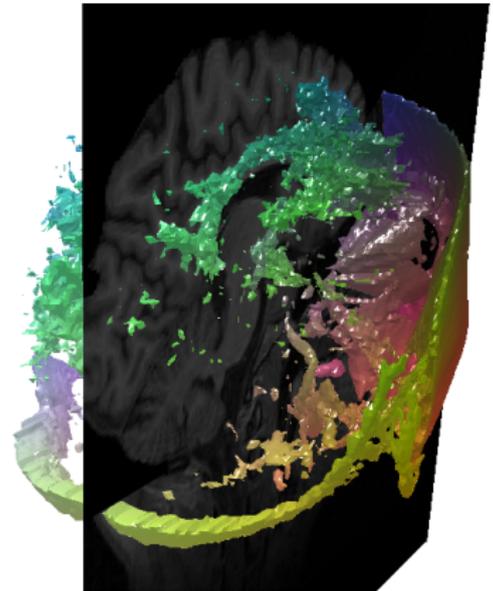
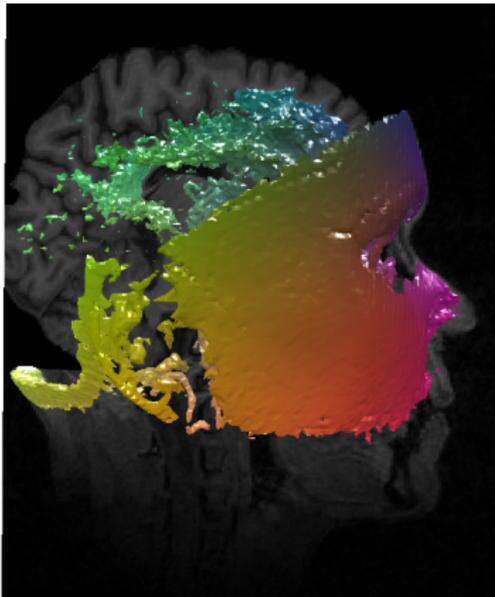
# Exemple d'isosurface : IRM

- Structures interne observable en coupant la surface.
- Valeurs aux frontières donne l'aspect du maillage.



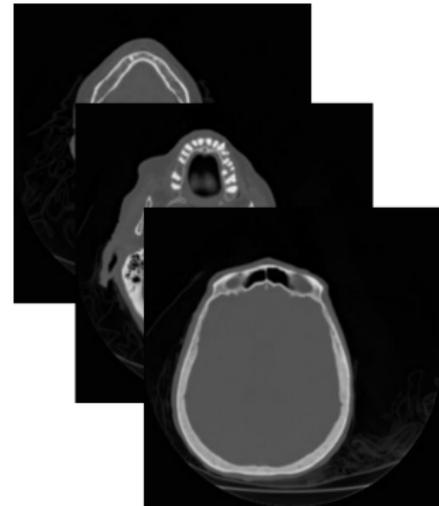
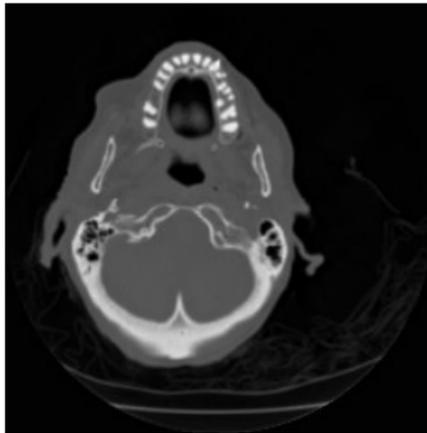
# Exemple d'isosurface : IRM

- Combine slicing + isosurface



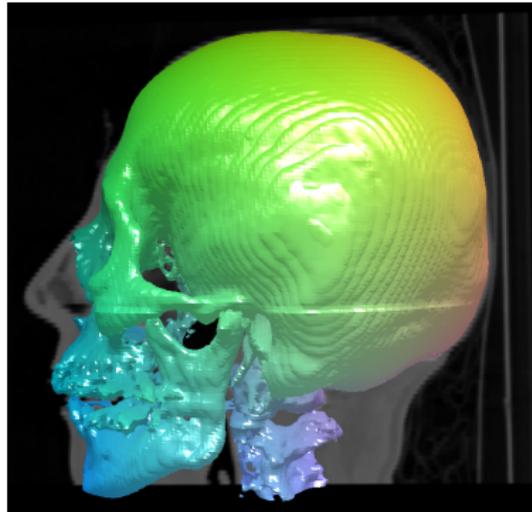
# Exemple d'isosurface : CT

- Donnée CT (Rayons X)
- Information morphologique : peau/os
- $(256 \times 256 \times 99)$



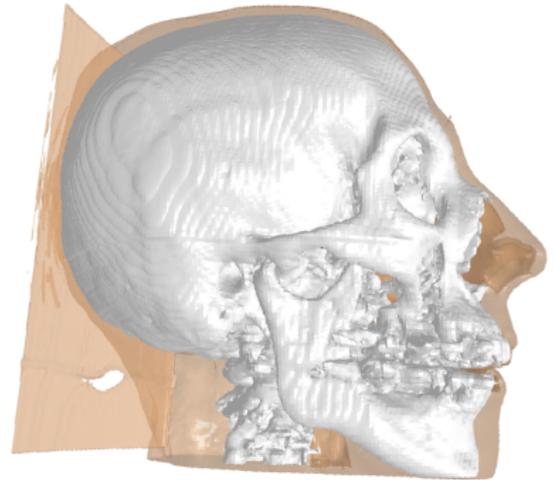
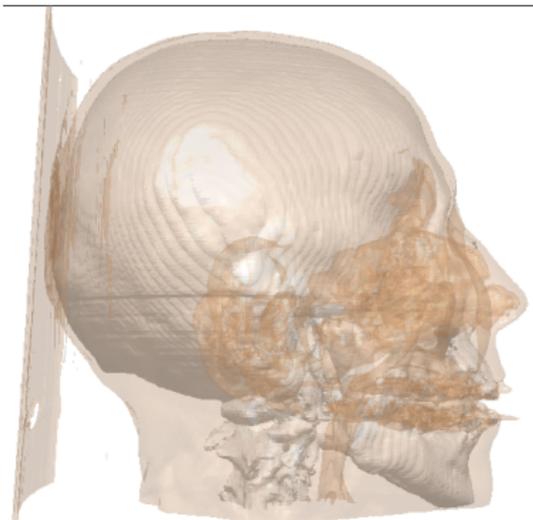
# Exemple d'isosurface : CT

- 2 Informations majeurs de peau + os
- Intérêt de la combinaison coupe + isosurface



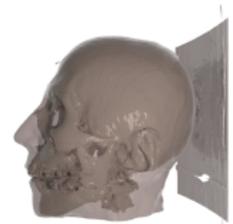
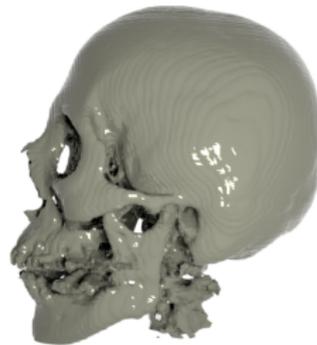
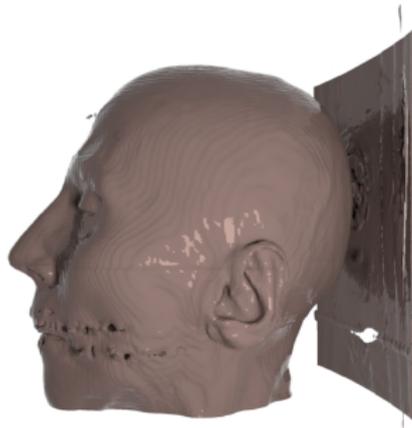
# Exemple d'isosurface : CT

- Possibilité de cumule d'informations surfacique par transparence



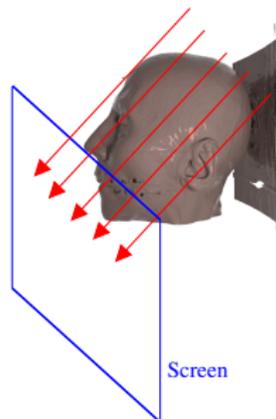
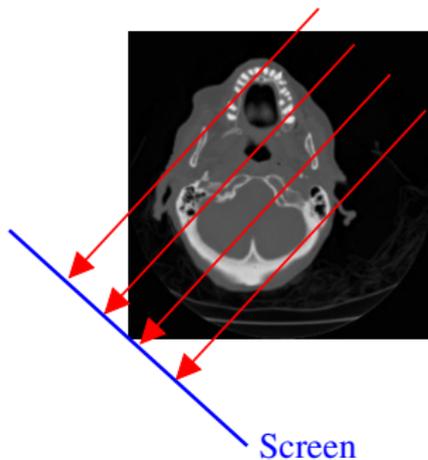
# Exemple d'isosurface : CT

- Ajout d'un rendu, visualisation morphologique



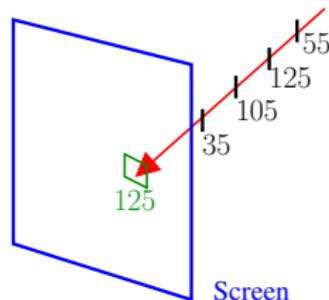
# Rendu Volumique

- Rendu direct des valeurs du volume projetées sur un plan.
- Par rotation, on a l'impression de 3D.
- On rend l'objet "transparent"
- *Principe général* : On lance des rayons et on affecte une couleur en fonction du trajet parcouru = **ray-casting**



# Maximum Intensity Projection (MIP)

- On lance une série de rayons perpendiculaires au plan de la caméra
  - On affecte au pixel correspondant l'intensité maximale rencontrée.
- ⊕ Très utilisé en imagerie-médicale  
⊖ Effet de profondeur inexistant  
⇒ nécessite une animation.



# Maximum Intensity Projection (MIP) : Théorie

- Plan  $\mathcal{P}$  définie par un point  $\mathbf{x}_0$  et une normale  $\mathbf{n}$ .

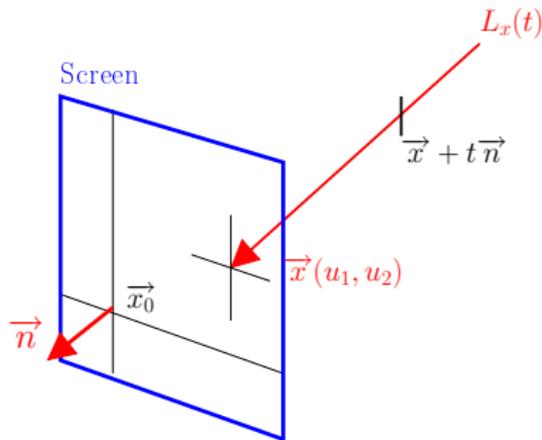
$$\mathbf{x}(u_1, u_2) = \mathcal{P} \Rightarrow (\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{n} = 0$$

- Ligne  $L$  telle que

$$L_{\mathbf{x}}(t) = \mathbf{x}(u_1, u_2) + t \mathbf{n}$$

- On affecte au pixel

$$I(u_1, u_2) = \max_t f(L_{\mathbf{x}}(t))$$



# Maximum Intensity Projection (MIP)

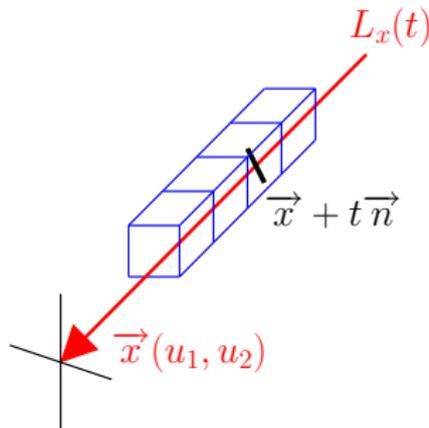
## ■ En discret :

pour tout  $(x[i], y[j])$  du plan

pour tout  $t$  de la droite d'integration

$$X = (x[i], y[j], z) + t \cdot \text{normal}$$

$$I(i, j) = \max(f(X), I(i, j))$$

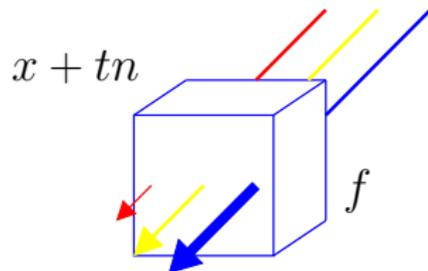


# Maximum Intensity Projection (MIP)



# Fonction de transfert

- Dans le cas général, on affecte une fonction lors de la progression.
- $I(i, j) = \mathcal{F}(t, \mathbf{x}, f_{\mathbf{x}}(\gamma(t)))$
- Chaque position possède
  - Une absorption/atténuation  $\alpha$
  - Une intensité/couleur  $C$

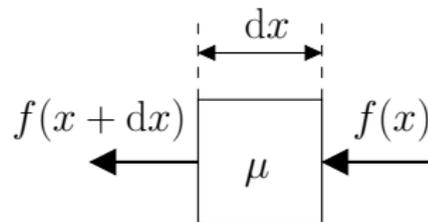


## Rappel sur l'atténuation

- Soit une atténuation  $\alpha(t)$  en un point de la droite paramétré par  $t$ .
- Soit une section  $dL$ .
- Une intensité lumineuse  $f$  arrive en  $L(t)$ .
- La perte d'intensité en  $L(t) + dt$  est proportionnelle à  $f$  incident, et à l'épaisseur  $dt$ .

$$\Rightarrow f(L(t) + dt) - f(L(t)) = -\alpha f(L(t))dt$$

$$\Rightarrow f(L(t)) = f(L(t_0))e^{\int_{t'=t_0}^t -\alpha t' dt'}$$



- L'intensité totale reçu sur le plan image  $\gamma(0)$  est donc

$$I(u_1, u_2) = \int_{t=-\infty}^0 f(L(t)) e^{-\int_{t'=t}^0 \alpha t' dt'}$$

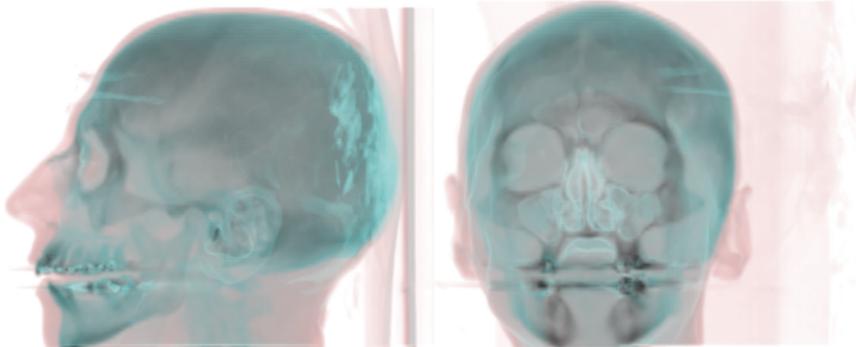
# Intégration discrète

- Classiquement on utilise des fonctions de “blending”
- Intensité de l'image en  $k = 0$  est donnée par  $C_0$  avec

$$C_{k-1} = \alpha C_{k+1} + (1 - \alpha)f_k$$

# Fonction de transfert

- Pour une valeur  $f_k$  donnée (temps donné), on peut affecter une couleur ( $R, G, B, A$ ).
- $A$  définit l'absorption.
- Tableau de correspondance de couleur / LUT / Fonction de transfert ...



# Visualisation-Multiresolution 9-Données Vectorielles

Polytech-Grenoble

1er semestre 2008

# Généralités

- On est sur une surface ( $v=2$ ) ou un volume ( $v=3$ ).
- On est plongé dans  $\mathbb{R}^2$  ou  $\mathbb{R}^3$ .
- On a typiquement  $d = 3$  (ou 2) composantes par positions
- Cas d'exemple général :

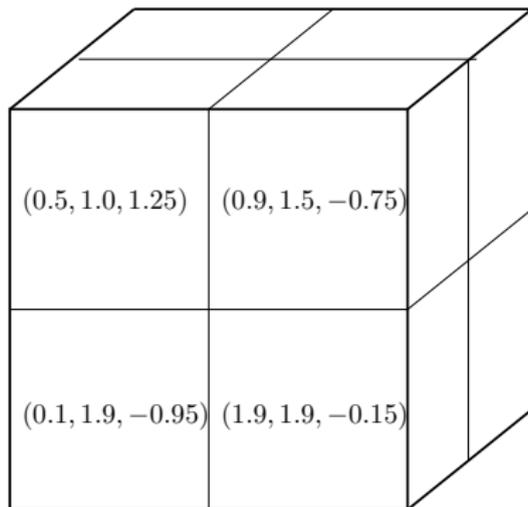
$$\vec{f}(\vec{u}) = \left( f_1(u_1, u_2, u_3), f_2(u_1, u_2, u_3), f_3(u_1, u_2, u_3) \right)$$

- Cas classique :

$$\vec{f} = \left( f_x(x, y, z), f_y(x, y, z), f_z(x, y, z) \right)$$

# Généralités

- En discret : Valeurs vectorielles sur chaque échantillon (voxel si suivant  $x,y,z$ )



## Affichage direct

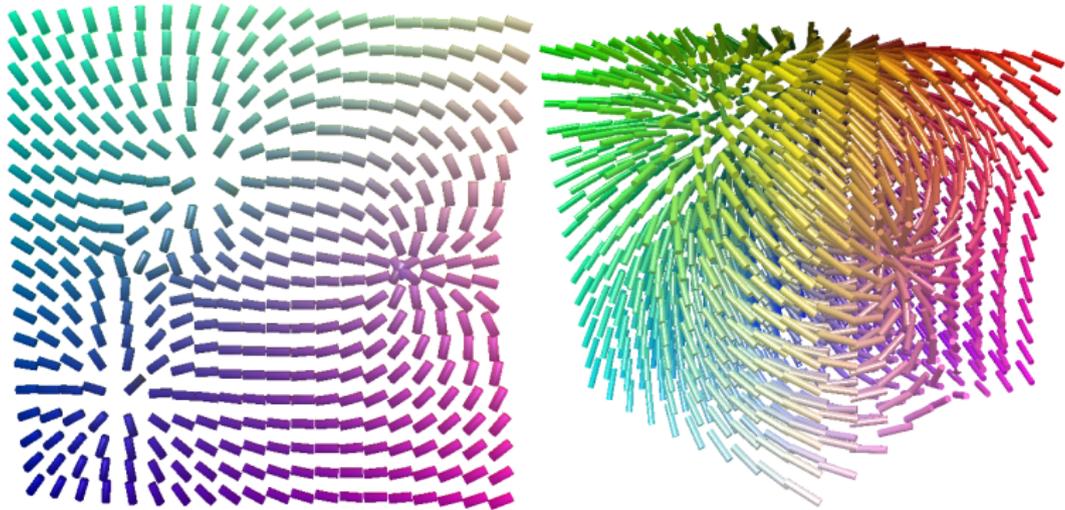
- Le plus simple : On affiche un vecteur pour chaque position
- Si on est sur un volume  $V$  de  $\mathbb{R}^3$ , on affiche alors le vecteur

$$V(u_1, u_2, u_3) + \lambda \vec{f}(u_1, u_2, u_3) .$$

Pour tout  $(x, y, z)$  de  $V$

Affiche segment  $[V(x, y, z), V(x, y, z) + a * f(x, y, z)]$

# Affichage direct



# Affichage direct

Avantages :

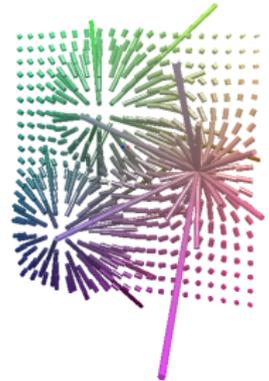
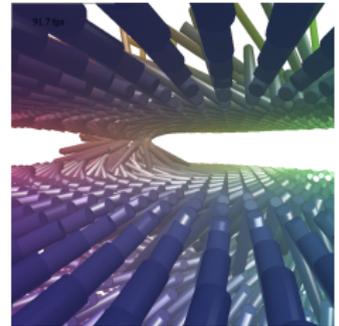
⊕ Visualisation locale de la direction et norme

Inconvénients :

⊖ Indication locale (on ne suis pas de trajet)

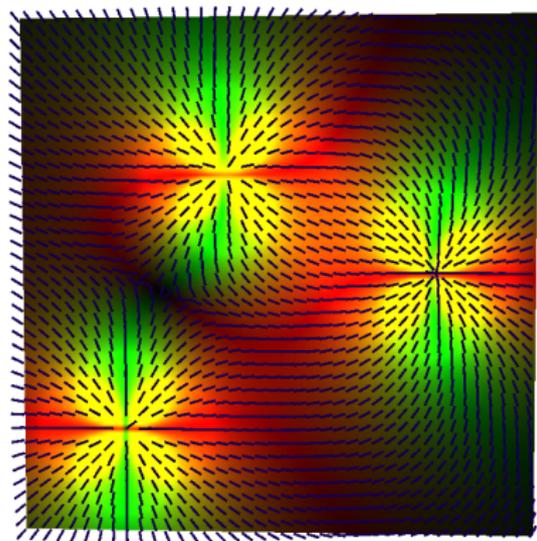
⊖ Mal adapté à une variation importante de la norme.

⊖ Rapidement confus (surtout en 3D)



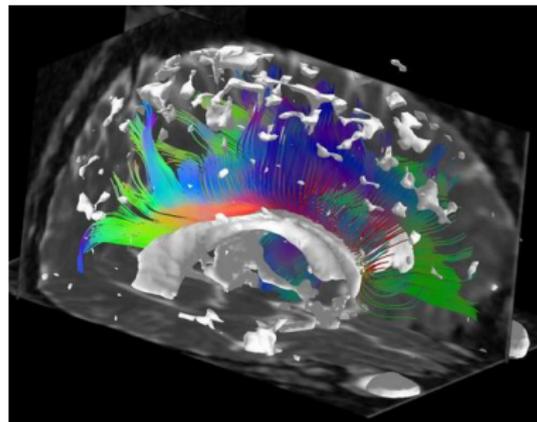
# Encodage couleur

- On applique une couleur en fonction de la direction.
  - L'amplitude est encodée par l'intensité.
  - Surtout pour le cas de surface 2D
- ⊕ Visualisation continue
- ⊖ Direction difficile à interpréter



# Streamlines

- On cherche à tracer la trajectoire de particules suivant le champ  $\vec{f}$ .
- ⊕ Information structurelle Globale
- La trajectoire peut être réelle ou non



# Streamlines

- Suivre un flux = intégrer en partant d'un point initial  $\vec{x}(0)$ .

$$\vec{x}(t) = \int_{t'=0}^{t'=t} \vec{f}(\vec{x}(t')) dt'$$

- Mis sous forme différentielle

$$\begin{cases} \vec{x}'(t) = \vec{f}(\vec{x}(t)) \\ \vec{x}(0) = \vec{x}_0 \end{cases}$$

- Trouver une trajectoire = résoudre une equation différentielle ordinaire (ODE)

# Streamlines : Discretisation

- Discretisation temporelle :

$$\vec{x}'(t) = \frac{\vec{x}(t + dt) - \vec{x}(t)}{dt} + O(dt) = \vec{f}(\vec{x}(t)) + O(dt)$$

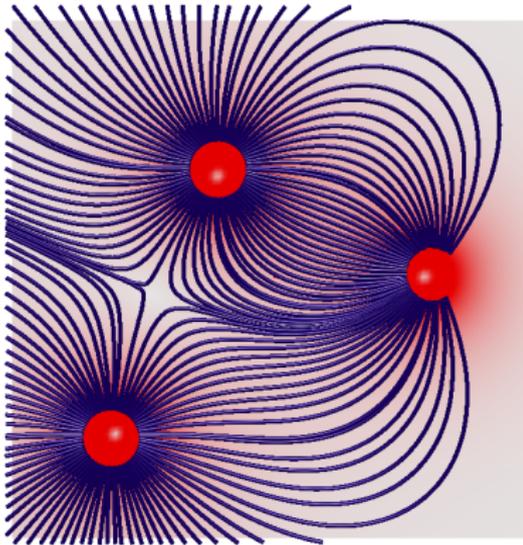
$$\Rightarrow \vec{x}(t + dt) = \vec{x}(t) + dt \vec{f}(\vec{x}(t)) + O(dt^2)$$

- En prenant  $k$  itérations de pas de temps  $\Delta t$  :

$$\vec{x}^{k+1} \simeq \vec{x}^k + \Delta t \vec{f}(\vec{x}^k)$$

- Méthode d'Euler explicite du premier ordre

# Streamlines : Exemples

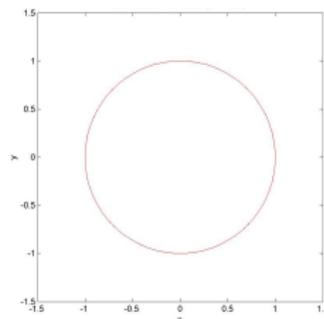
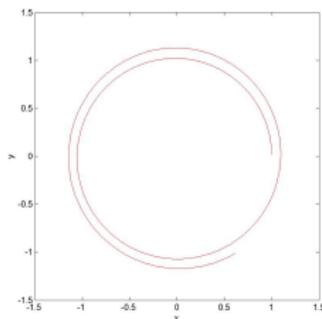


# Streamlines : Stabilité

- Problème de stabilité des méthodes explicites
- Fonctions raides (stiff)
- Méthode implicite

$$\vec{x}(t + dt) = \vec{x}(t) + \vec{f}(\vec{x}(t + dt)) dt$$

$$(\text{Id} - \Delta t \vec{f})(\vec{x}^{k+1}) = \vec{x}^k$$



# Streamlines : Précision (accuracy)

- Euler ordre 1 : À chaque pas de temps, on fait une erreur qui varie en  $O(\Delta t^2)$ .
- Ordre supérieur :
  - Approximer dérivée ordre  $n$  (complexe)
  - Runge-Kutta : (ex. RK4)

$$\left\{ \begin{array}{l} \vec{x}^{k+1} = \vec{x}^k + \frac{\Delta t}{6} (\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) \\ \vec{k}_1 = \vec{f}(\vec{x}^k) \\ \vec{k}_2 = \vec{f}(\vec{x}^k + \frac{\Delta t}{2} \vec{k}_1) \\ \vec{k}_3 = \vec{f}(\vec{x}^k + \frac{\Delta t}{2} \vec{k}_2) \\ \vec{k}_4 = \vec{f}(\vec{x}^k + \Delta t \vec{k}_3) \end{array} \right.$$