Géométrie numérique

Séance 5 – Reconstruction de surfaces

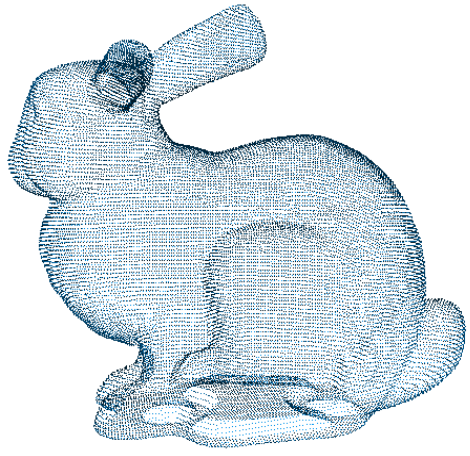
Franck Hétroy-Wheeler

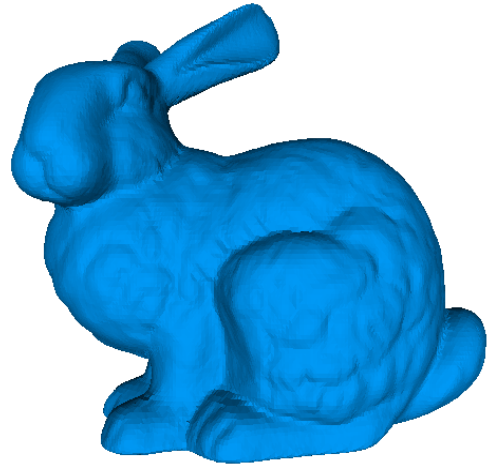
M1 ISI – 2017-2018

UNIVERSITÉ DE STRASBOURG

Motivation

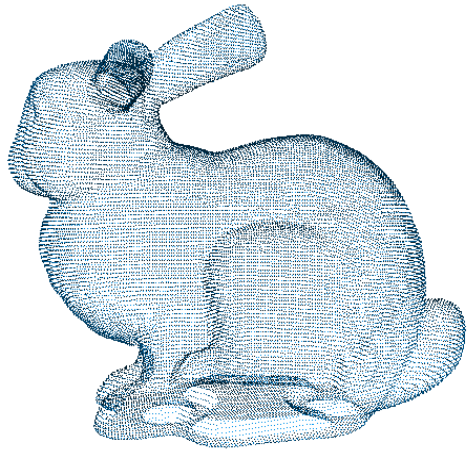
D'un **nuage de points** 3D à une surface **maillée**

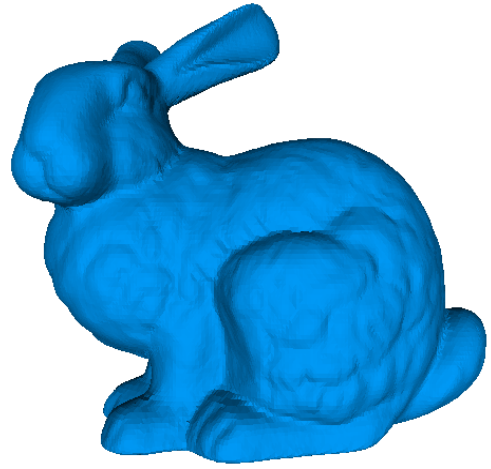




Motivation

D'un **nuage de points** 3D à une surface **maillée**



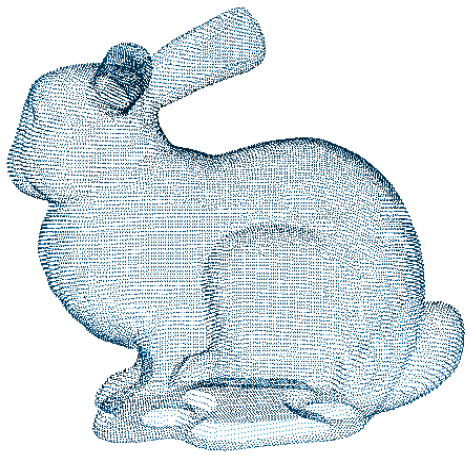


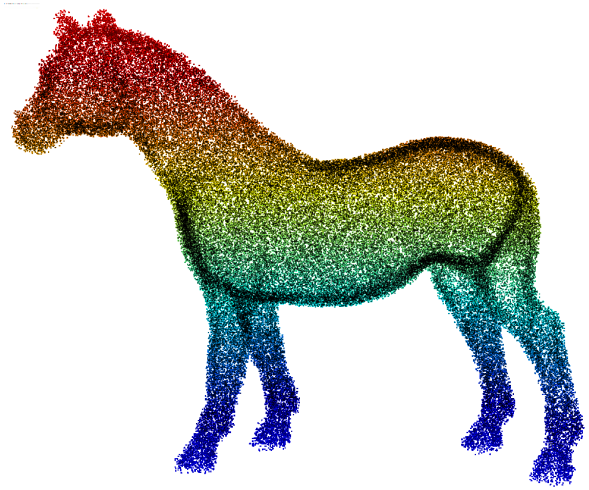
- 2-**variété**
- Sans **bord** (*watertight*)
- **Approchant** les points (plutôt qu'interpolant)

Entrée

Le nuage de points peut être :

- **Organisé** (issu d'un scan en lignes/colonnes) ou non
- **Orienté** (normale associée à chaque point) ou non
- De **densité uniforme** ou non
- **Bruité** ou non





UNIVERSITÉ DE STRASBOURG

Méthodes type Delaunay

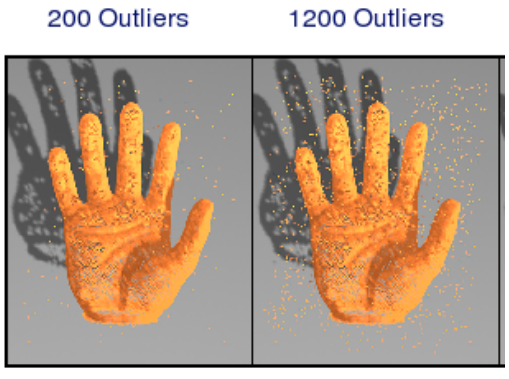
- **Triangulation de Delaunay** : cf. cours “Modélisation et algorithmique géométrique” S3
  - Définition : aucun sommet dans le cercle circonscrit à un triangle
  - Propriété fondamentale : maximise l'angle minimum

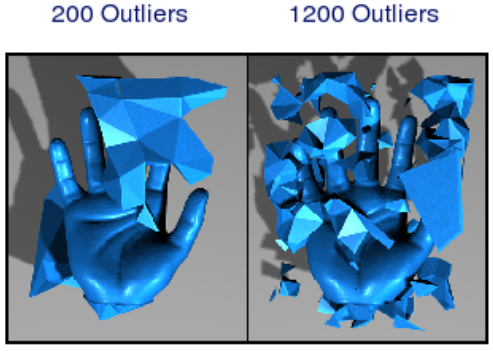
Méthodes type Delaunay

- **Triangulation de Delaunay** : cf. cours “Modélisation et algorithmique géométrique” S3
  - Définition : aucun sommet dans le cercle circonscrit à un triangle
  - Propriété fondamentale : maximise l'angle minimum
- Grande partie des méthodes existantes
- Intérêt : preuves théoriques de **validité**
  - Topologie, distance Euclidenne aux échantillons, distance des normales
  - Sous **conditions** (pas/peu de bruit, . . .)
- `http://interstices.info/display.jsp?id=c_12845`

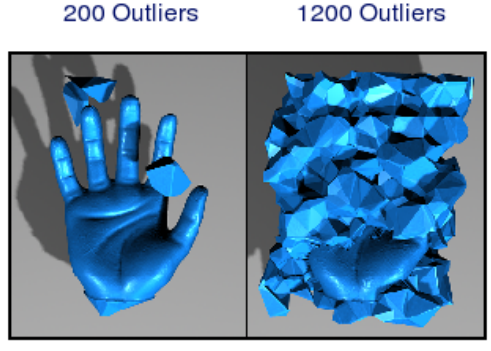
Avantages et inconvénients

- **Taille** du maillage résultat $\sim$ taille de l'échantillon
  - Pas/peu d'ajout de points
- **Echantillonnage** connu et uniforme $\Rightarrow$ très **précis**
- Bruit ou **points aberrants** $\Rightarrow$ mauvais résultats





Tight Cocone



Powercrust

UNIVERSITÉ DE STRASBOURG

Fitting de surface implicite

- Idée :
  1. Définir une **surface implicite lisse** qui approche la surface réelle sous-jacente
  2. **Projeter** ou générer les/des points sur cette surface et trianguler

UNIVERSITÉ DE STRASBOURG

Fitting de surface implicite

- Idée :
  1. Définir une **surface implicite lisse** qui approche la surface réelle sous-jacente
  2. **Projeter** ou générer les/des points sur cette surface et trianguler
- Problème principal : **définition** de la surface implicite
  - Beaucoup de possibilités : fonction de distance, MLS, RBF, ...

Hoppe et al. SIGGRAPH 1992

- **Entrée** :
  - Points avec coordonnées 3D, pas de normale
  - **Topologie**/bords : **arbitraires**, inférés depuis le nuage de points
  - **Densité** d'échantillonnage : supposée **uniforme**, donnée en paramètre
  - Erreur max par rapport à la surface sous-jacente (bruit) : donnée en paramètre

Hoppe et al. SIGGRAPH 1992

- **Entrée** :
  - Points avec coordonnées 3D, pas de normale
  - **Topologie**/bords : **arbitraires**, inférés depuis le nuage de points
  - **Densité** d'échantillonnage : supposée **uniforme**, donnée en paramètre
  - Erreur max par rapport à la surface sous-jacente (bruit) : donnée en paramètre
- **Sortie** :
  - Surface **maillée**
  - Pas nécessairement triangulée
  - **2-variété** connexe et orientable

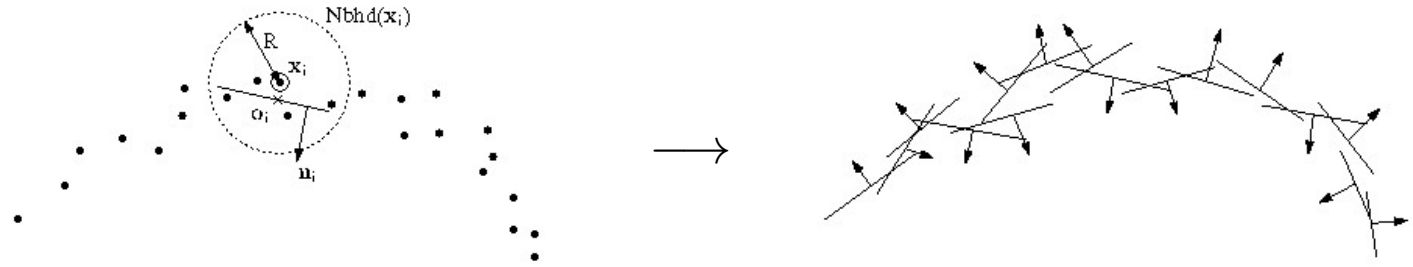
Algorithme

- 4 étapes :
  1. Estimation du **plan tangent** à la surface sous-jacente S en chaque point x_i du nuage
     - Approximation localement linéaire de S
  2. **Orientation** consistante de chacun des plans par rapport à ses voisins
  3. Fonction implicite : **distance Euclidienne signée** à un plan proche et échantillonnage sur une **grille de voxels**
  4. Extraction d'une **isosurface**

UNIVERSITÉ DE STRASBOURG

Etape 1 : plan tangent

- Définition : **centre** o_i et **normale** n_i
- Approche par **plus proches voisins**
  - Calcul de o_i par approximation aux **moindres carrés** des k points les plus proches de x_i
  - Calcul de n_i par ACP
  - k = paramètre fixé par l'utilisateur



Etape 2 : orientation des plans

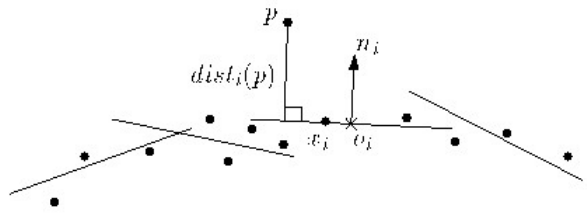
- **Optimisation de graphe**
  - Nœuds = plans, arête P_iP_j si o_i et o_j proches (graphe riemannien)
  - Poids d'une arête $= 1 - n_i.n_j$
  - Objectif : minimisation du poids total du graphe
  - Problème NP-complet $\Rightarrow$ approximation : arbre couvrant minimal

Etape 3 : fonction implicite

- Distance Euclidienne signée à un **plan proche** :

$$f(x) = (x - o_i).n_i$$

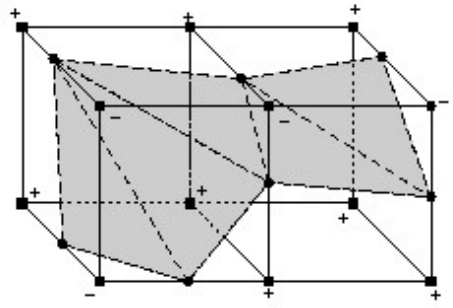
avec o_i centre le plus proche de x

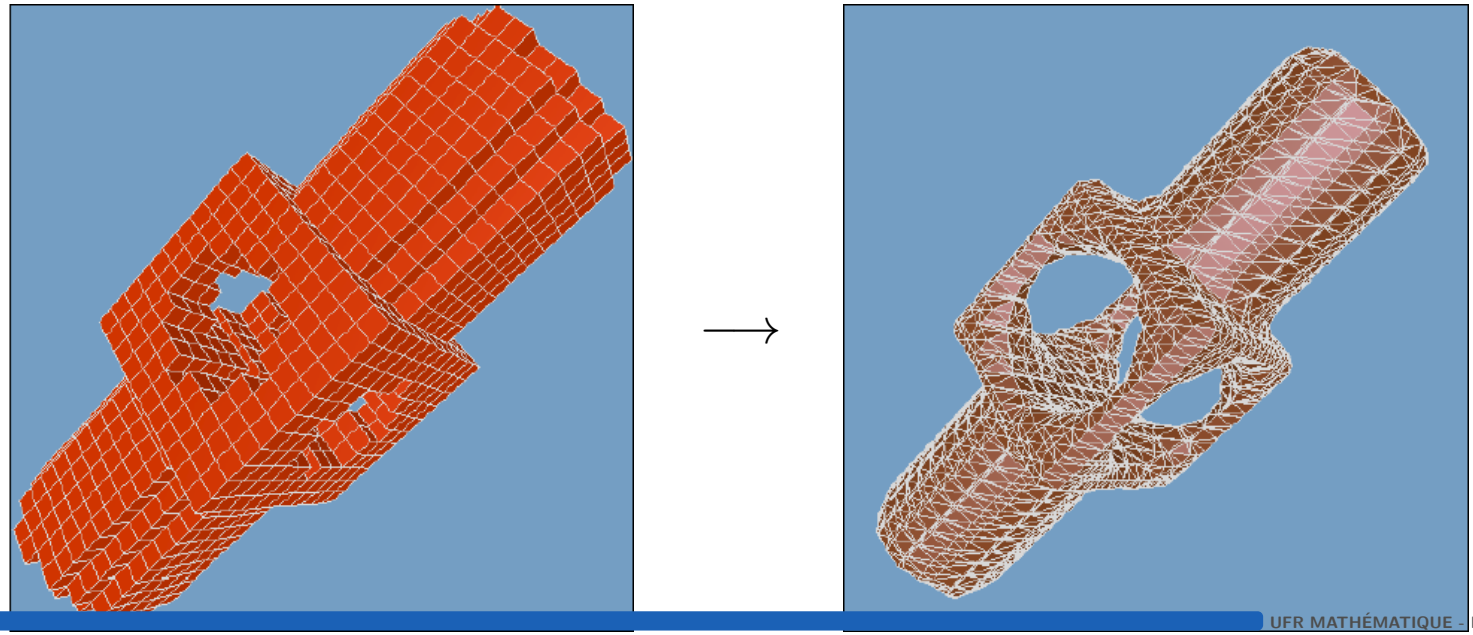


- Surfaces à bord : les points trop éloignés se voient assignés une distance infinie
- Distance échantillonnée aux sommets d'une **grille de voxels**

Etape 4 : isosurface

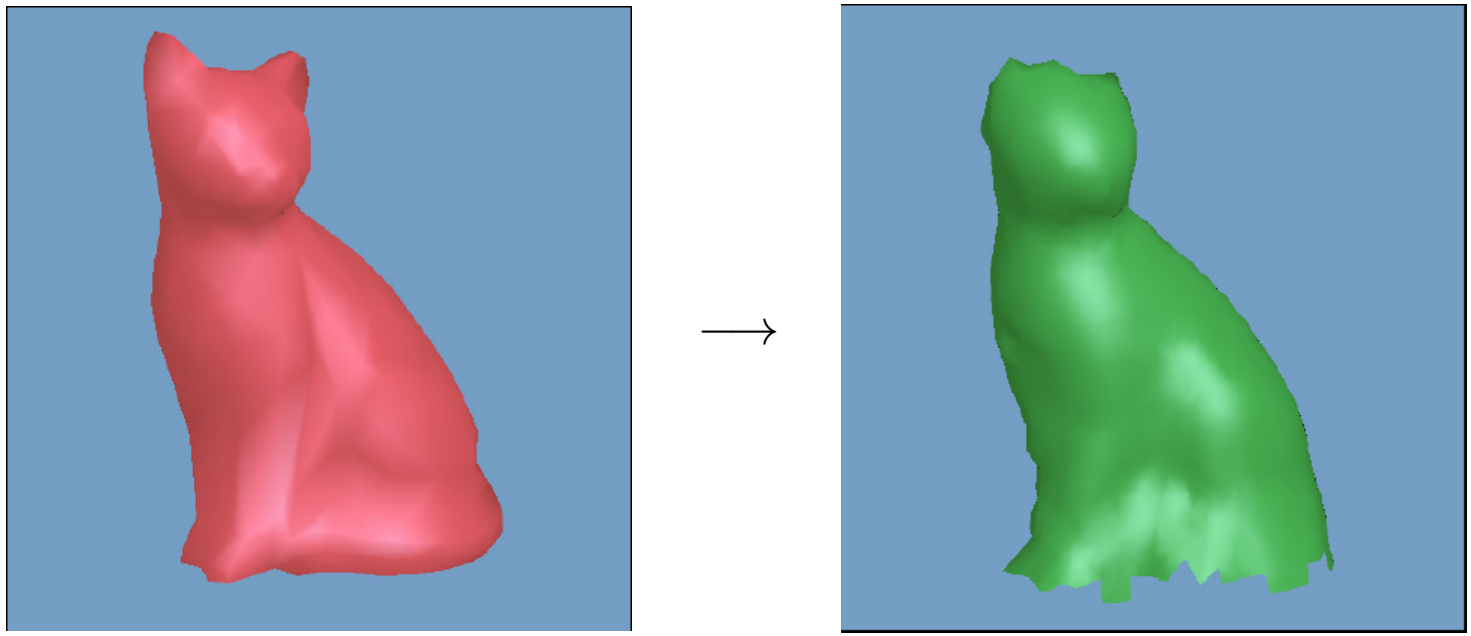
- Algorithme des **Marching Cubes** [Lorensen and Cline, SIGGRAPH 1987]





Discussion

- Complexité en temps et place mémoire : $O(n \log n + m)$
  - n = nombre de points, m^3 = taille de la grille
- Problème si densité non uniforme : besoin d'une **taille de voisinage adaptative**
- Résultats impressionnants . . . pour l'époque (1992) !



UNIVERSITÉ DE STRASBOURG

Fonctions de base radiales (RBF)

- **Fonction radiale** : ne dépend que de la **distance à un point** x_i

$$\Phi(x, x_i) = \phi(\|x - x_i\|)$$

  - Invariante pour toute rotation autour de x_i
  - x_i appelé **centre** de la fonction radiale

Fonctions de base radiales (RBF)

- **Fonction radiale** : ne dépend que de la **distance à un point** x_i

$$\Phi(x, x_i) = \phi(\|x - x_i\|)$$

  - Invariante pour toute rotation autour de x_i
  - x_i appelé **centre** de la fonction radiale

- Fonctions de base radiales classiques :
  - Gaussiennes : $\phi(r) = e^{-\alpha r^2}$
  - Multiquadriques : $\phi(r) = \sqrt{r^2 + \alpha^2}$
  - Splines à plaque mince : $\phi(r) = r^2 \log r$
  - Monomiales : $\phi(r) = r^n$

Décomposition d'une fonction implicite en RBF

- $f(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + P(x)$
  - On suppose f connue en les x_i, les λ_i inconnus
  - P polynôme, dépend du choix des RBF, nécessaire pour trouver les λ_i

Décomposition d'une fonction implicite en RBF

- $f(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + P(x)$
  - On suppose f connue en les x_i, les λ_i inconnus
  - P polynôme, dépend du choix des RBF, nécessaire pour trouver les λ_i
- S'écrit sous forme matricielle :

$$\begin{pmatrix} \Phi & X \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \end{pmatrix}$$

avec :

- $\Phi(i,j) = \phi(\|x_i - x_j\|)$
- $X(i,.) = (1 \ x_i \ y_i \ z_i)$
- λ vecteur des λ_i
- c vecteur des coefficients de P
- f vecteur des $f(x_i)$

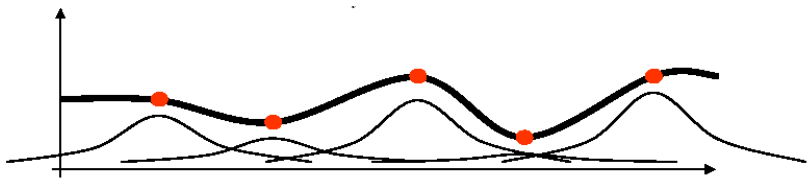


Figure par Marc Alexa

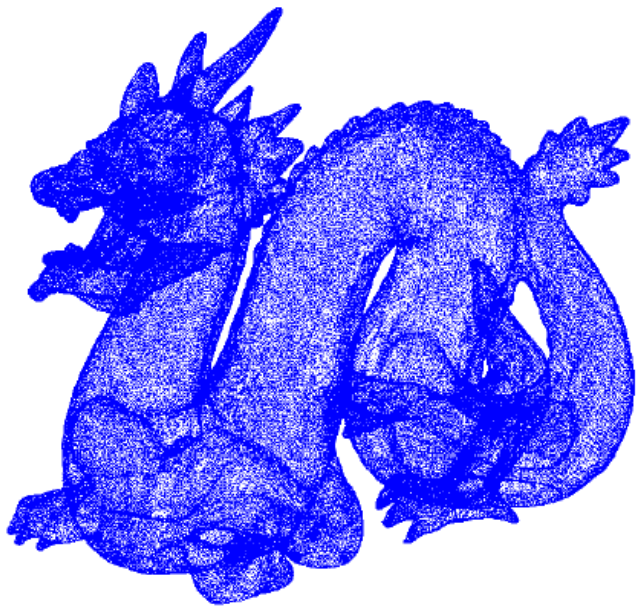
UNIVERSITÉ DE STRASBOURG

Comparaison par rapport à Hoppe

- Système inversible d'équations linéaires : pas besoin de **grille**
- Choix des fonctions de base : plus de **contrôle** qu'avec une fonction de distance
- Inconvénient : besoin de quelques **normales** (peuvent être estimées par la méthode de Hoppe) pour éviter la solution triviale $f(x) = 0$
- Temps de calcul : long mais astuces pour accélérer le temps de calcul

Pour en savoir plus

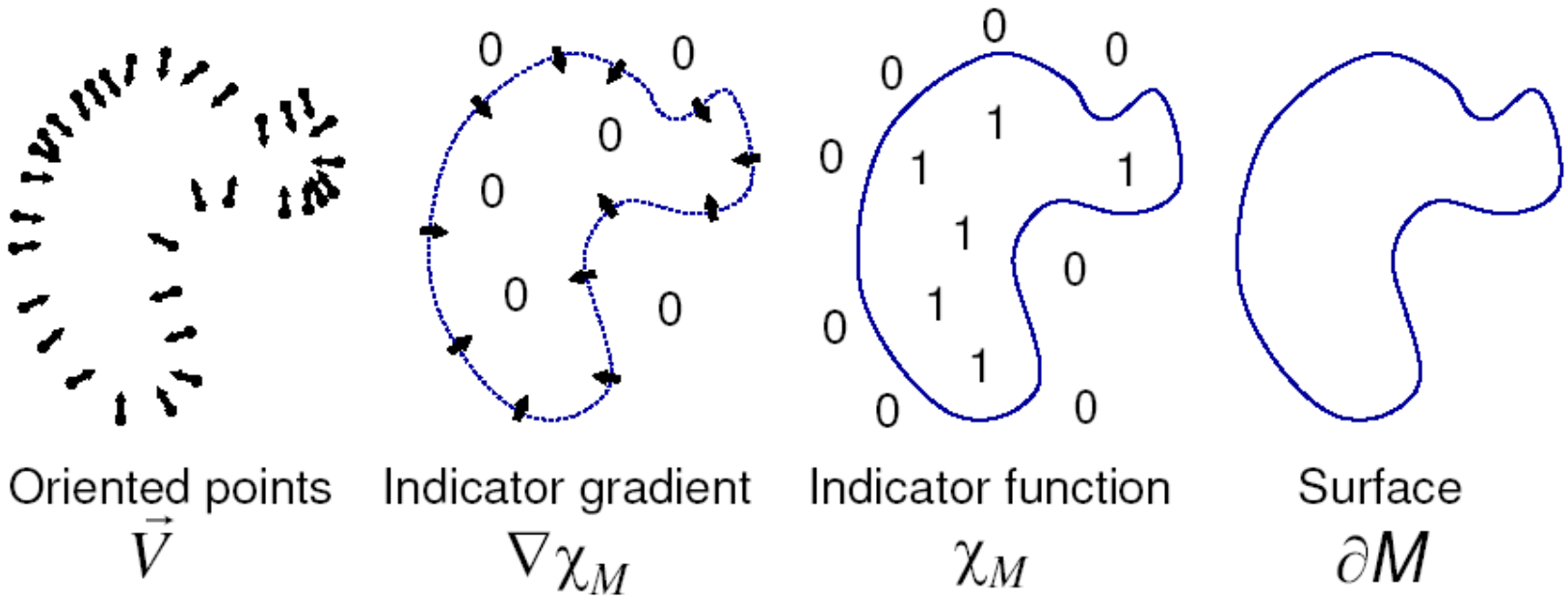
J. C. Carr et al., “Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2003



Nouvelle modélisation du problème

Modélisation du problème :

- Entrée : points **munis de leurs normales**
- **Fonction indicatrice** sur R^3 : 1 à l'intérieur de l'objet, 0 en dehors
- $\Rightarrow$ **Gradient** : 0 partout sauf au voisinage de la surface



UNIVERSITÉ DE STRASBOURG

Reconstruction de Poisson

- Entrée : **gradient** échantillonné
  - V = champ de gradient défini aux points = **normales** des points x_i
- Problème : trouver la **fonction indicatrice** χ qui minimise $\|\nabla\chi - V\|$

Reconstruction de Poisson

- Entrée : **gradient** échantillonné
  - V = champ de gradient défini aux points = **normales** des points x_i
- Problème : trouver la **fonction indicatrice** χ qui minimise $\|\nabla\chi - V\|$
- Se reformule comme une **équation de Poisson**

$$\triangle\chi = \nabla.\nabla\chi = \nabla V$$

  - Equation de diffusion : $\triangle\chi = 0$

Implémentation

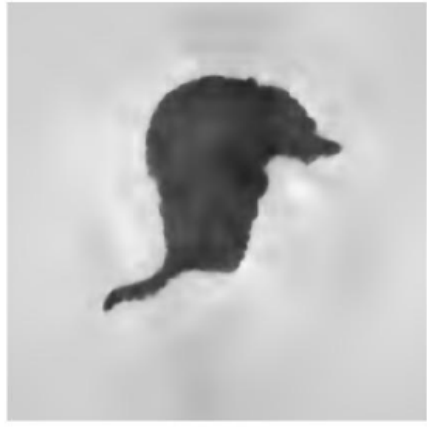
- Implémentation **efficace** possible
  - χ décomposée dans une base de fonctions **à support compact** $\Rightarrow$ système matriciel creux
  - Discrétisation de l'espace adaptative : **octree** plutôt qu'une grille uniforme
- **Code** disponible
  - `http://www.cs.jhu.edu/~misha/Code/PoissonRecon`
  - Trois méthodes

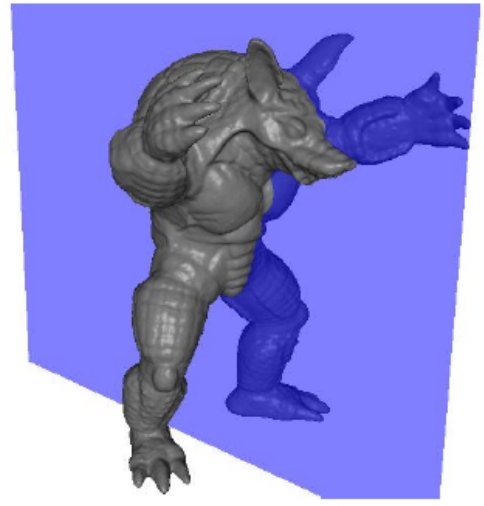
Implémentation

- Implémentation **efficace** possible
  - χ décomposée dans une base de fonctions **à support compact** $\Rightarrow$ système matriciel creux
  - Discrétisation de l'espace adaptative : **octree** plutôt qu'une grille uniforme
- **Code** disponible
  - `http://www.cs.jhu.edu/~misha/Code/PoissonRecon`
  - Trois méthodes
- Pour en savoir plus :
  - M. Kazhdan, M. Bolitho and H. Hoppe, "Poisson Surface Reconstruction", Symposium on Geometry Processing 2006

Comparaison aux méthodes précédentes

- Théoriquement très **simple**
- Fonction χ **lisse** $\Rightarrow$ bons résultats même en présence de bruit
- Surface fermée, sans arêtes vives
- Besoin de l'information de **normales** partout, et consistantes





Autres méthodes par surface implicite

- **Moving Least Squares** (MLS)
  - Fonctions de base : fonctions quadratiques approchant localement la surface
  - Plus de détails demain
  - Article : C. Shen, J.F. O'Brien, J.R. Shewchuk, "Interpolating and Approximating Implicit Surfaces from Polygon Soup", SIGGRAPH 2004

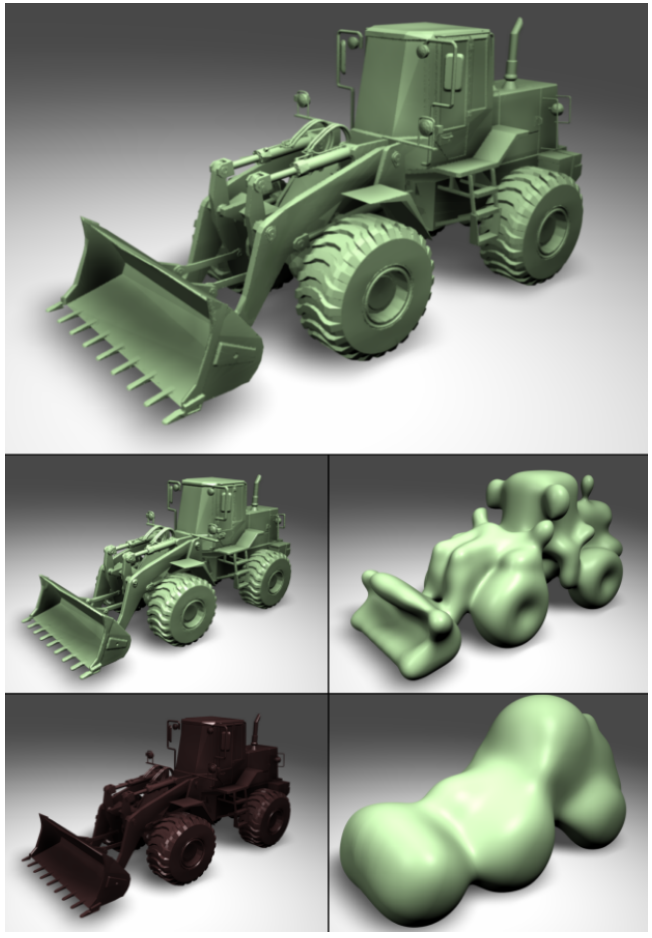
Autres méthodes par surface implicite

- **Moving Least Squares** (MLS)
  - Fonctions de base : fonctions quadratiques approchant localement la surface
  - Plus de détails demain
  - Article : C. Shen, J.F. O'Brien, J.R. Shewchuk, "Interpolating and Approximating Implicit Surfaces from Polygon Soup", SIGGRAPH 2004
- **Multi-Level Partition of Unity** (MPU)
  - Fonctions de base : trois types de quadriques différentes suivant la forme locale de la surface (lisse, arêtes vives)
  - Combinées à des **fonctions de poids** bien choisies
  - Article : Y. Ohtake et al., "Multi-Level Partition of Unity Implicits", SIGGRAPH 2003

Autres méthodes par surface implicite

- **Moving Least Squares** (MLS)
  - Fonctions de base : fonctions quadratiques approchant localement la surface
  - Plus de détails demain
  - Article : C. Shen, J.F. O'Brien, J.R. Shewchuk, "Interpolating and Approximating Implicit Surfaces from Polygon Soup", SIGGRAPH 2004
- **Multi-Level Partition of Unity** (MPU)
  - Fonctions de base : trois types de quadriques différentes suivant la forme locale de la surface (lisse, arêtes vives)
  - Combinées à des **fonctions de poids** bien choisies
  - Article : Y. Ohtake et al., "Multi-Level Partition of Unity Implicits", SIGGRAPH 2003
- Ces deux méthodes supposent la surface lisse **localement**, contrairement à RBF et Poisson
  - Arêtes vives possibles

Résultats



MLS

MPU

Segmentation volumique

Idée :

- **Etiqueter** directement chaque sommet d'une **grille volumique** comme **intérieur** ou **extérieur**
- Appliquer une méthode type Marching Cubes pour générer la surface

Exemple : A. Hornung and L. Kobbelt, "Robust reconstruction of 3D models from point clouds", Eurographics 2006

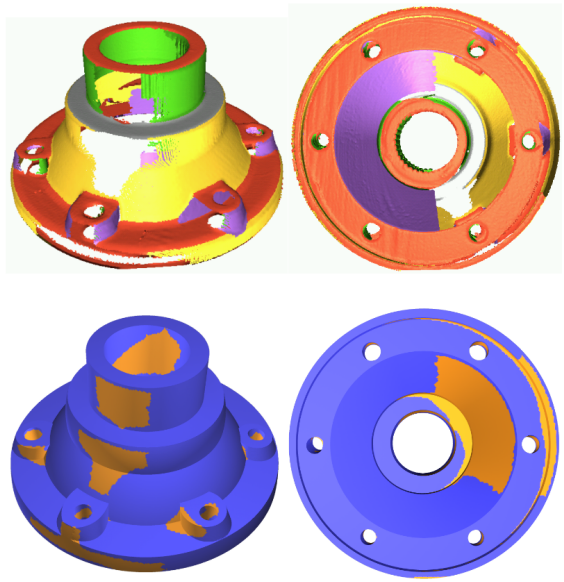
- Etiquetage : méthode par coupure de graphe
- Très robuste : bruit, échantillonnage non uniforme, données manquantes

Fitting de primitives

Idée :

- Approcher **localement** la surface par une **primitive géométrique** simple
  - **Plan**, sphère, cône, cylindre, ...
- Problème à résoudre ensuite : **couture** des primitives voisines

- Particulièrement adapté aux **modèles industriels** (CAO)
- Robuste au bruit et données manquantes
- Exemple : R. Schnabel et al., "Completion and Reconstruction with Primitive Shapes", Eurographics 2009

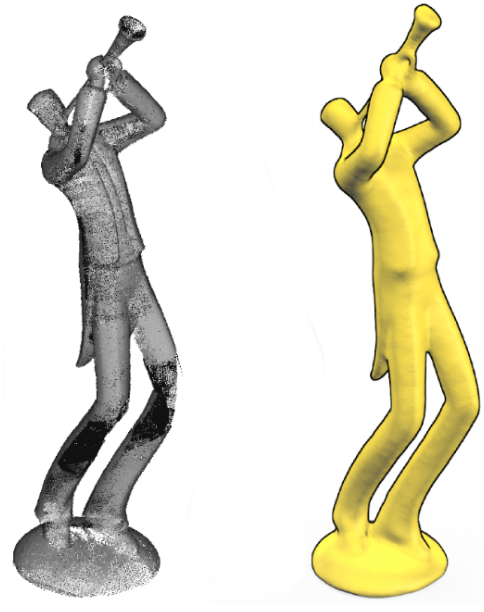


Utilisation d'un squelette

Idée :

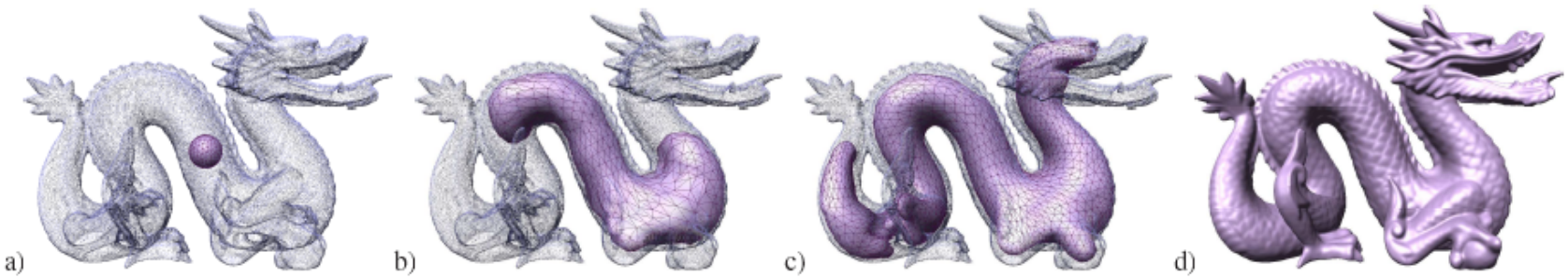
- Calcul d'une **courbe représentative** de l'objet appelée **squelette** à l'intérieur de celui-ci
- Grossissement local anisotropique de cette courbe

- Particulièrement adapté aux **modèles tubulaires** ou à axe de révolution (vases, ...)
- Robuste au données manquantes
- Exemple : A. Tagliasacchi et al., "VASE : Volume-Aware Surface Evolution for Surface Reconstruction from Incomplete Point Clouds", Symposium on Geometry Processing 2011



Modèles déformables

- Idée : partir d'une forme simple et petite (sphère par exemple) à l'intérieur de l'objet et la **déformer** pour qu'elle aille plaquer la surface
- Difficulté : établir les équations de déformation
- Suppose la surface fermée (**watertight**)
- Robuste au données manquantes
- Exemple : A. Sharf et al., "Competing Fronts for Coarse-to-Fine Surface Reconstruction", Eurographics 2006



UNIVERSITÉ DE STRASBOURG

Bilan

- **Très** nombreuses approches
  - Etat de l'art récent : M. Berger et al., "A Survey of Surface Reconstruction from Point Clouds", Computer Graphics Forum 2016
- Idée classique : interpoler ou approcher les points par une **surface implicite**
  - Surface lisse, la plupart du temps fermée
- Points critiques :
  - Approche locale ou globale ?
  - Utilisation des **normales** aux points ou pas ? Orientées ou pas ?
  - **Robustesse** : bruit, densité non uniforme, données manquantes

Merci