

# Numérisation 3D

La tête dans les nuages (de points)

---

Franck Hétroy-Wheeler

Montpellier, 30 mars 2018



- ▶ Franck Hétroy-Wheeler, [hetroywheeler@unistra.fr](mailto:hetroywheeler@unistra.fr)
- ▶ Enseignant-chercheur en **informatique**
- ▶ Professeur à l'**université de Strasbourg** (laboratoire ICube, équipe IGG) depuis 2017
  - ▶ Chercheur associé dans l'équipe MOSAIC (**Inria**, Lyon)
  - ▶ Auparavant : maître de conférences à Grenoble INP - Ensimag (2004-2017), laboratoire Jean Kuntzmann, équipes EVASION puis Morpheo

- ▶ Franck Hétroy-Wheeler, [hetroywheeler@unistra.fr](mailto:hetroywheeler@unistra.fr)
- ▶ Enseignant-chercheur en **informatique**
- ▶ Professeur à l'**université de Strasbourg** (laboratoire ICube, équipe IGG) depuis 2017
  - ▶ Chercheur associé dans l'équipe MOSAIC (**Inria**, Lyon)
  - ▶ Auparavant : maître de conférences à Grenoble INP - Ensimag (2004-2017), laboratoire Jean Kuntzmann, équipes EVASION puis Morpheo
- ▶ Thématiques de recherche :
  - ▶ **Analyse**/compréhension de **formes 3D**
  - ▶ Calcul/traitement numérique de la **géométrie** et de la **topologie**
  - ▶ Application aux données **botaniques** (arbres, plantes)

### Environnement virtuel :

représentation informatique du monde réel (et plus!)

- ▶ Terrain/paysage
- ▶ Personnages
- ▶ Objets et accessoires
- ▶ ...



Le monde de Rama, (c) Éric Bruneton

## Multimédia :

- ▶ Jeux vidéo
- ▶ Films d'animation
- ▶ Effets spéciaux
- ▶ Réalité virtuelle
- ▶ Infographie (web, logos, ...)



Big Buck Bunny, Blender Foundation



(c) Lucian Stanculescu

Autres industries :

- ▶ **Transport** : simulateurs de conduite
- ▶ **Architecture**
- ▶ **Médecine** : diagnostic via imagerie, simulation chirurgicale
- ▶ **Patrimoine**, musées
- ▶ **Objets** manufacturés : design, prototypage



(c) Corys

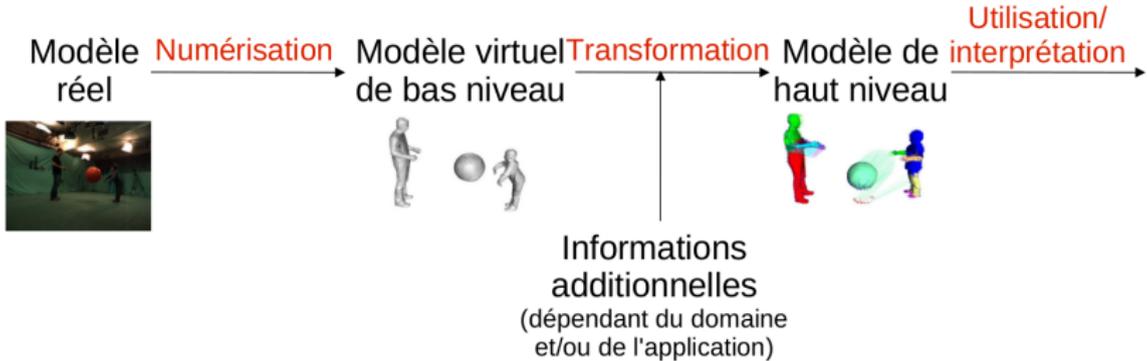


Portique du monastère de Ripoll

- ▶ Grâce à un **logiciel** adapté (Blender, Maya, ...)
  - ▶ **Artiste**
  - ▶ + : Pas forcément réaliste
  - ▶ - : Long, fastidieux, limité

- ▶ Grâce à un **logiciel** adapté (Blender, Maya, ...)
  - ▶ **Artiste**
    - ▶ + : Pas forcément réaliste
    - ▶ - : Long, fastidieux, limité
  - ▶ De manière **procédurale**
    - ▶ + : Rapide, niveaux de détails
    - ▶ - : Précision, pas toujours possible

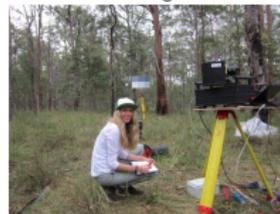
- ▶ Grâce à un **logiciel** adapté (Blender, Maya, ...)
  - ▶ **Artiste**
    - ▶ + : Pas forcément réaliste
    - ▶ - : Long, fastidieux, limité
  - ▶ De manière **procédurale**
    - ▶ + : Rapide, niveaux de détails
    - ▶ - : Précision, pas toujours possible
  - ▶ Par **numérisation** d'un modèle réel



- ▶ Objectif : récupérer un **ensemble de points** sur la **surface** du modèle
- ▶ Principe : envoi d'un rayon laser et retour
  - ▶ Balayage
- ▶ Un exemple :  
<http://www.youtube.com/watch?v=3oDp33mwPkE>



Scanner laser (petit modèle)  
NextEngine



Scanner laser (grand modèle)  
SALCA, (c) Salford University

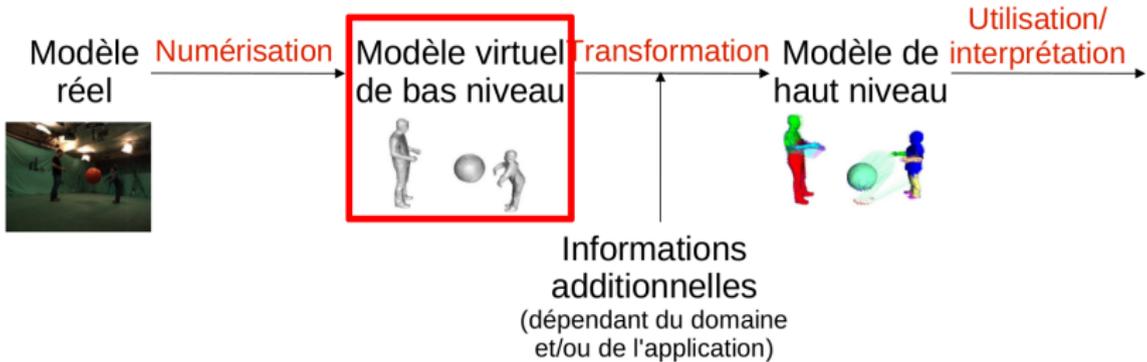
- ▶ Objectif : récupérer une **image 2D ou 3D** (un volume) permettant de distinguer le modèle de son environnement
- ▶ Principes : caméras, capteurs, ...
- ▶ Exemples :
  - ▶ Systèmes multi-caméras (images ou vidéos)
  - ▶ Caméra + capteurs (profondeur, ...)
  - ▶ Imagerie par résonance magnétique
  - ▶ Tomographie (rayons X)
  - ▶ Echographie (ultrasons)
  - ▶ Etc.



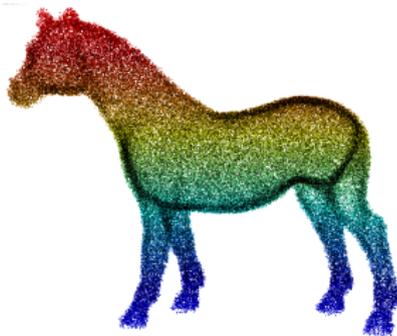
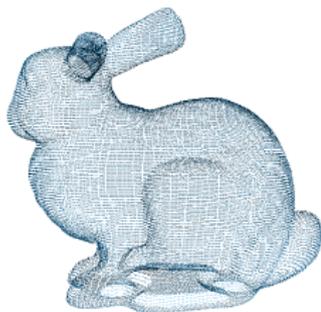
Système multi-caméras  
Plateforme Kinovis, (c) Inria



I.R.M.  
(source : Wikipedia)



- ▶ Un **point** = un ensemble de coordonnées  $(x,y,z)$ , dans un référentiel connu
- ▶ Parfois d'autres informations :
  - ▶ Couleur
  - ▶ Vecteur normal à la surface



## Contexte

### **Géométrie différentielle discrète**

Voisinage

Normale et plan tangent

Opérateur Laplacien

Descripteurs

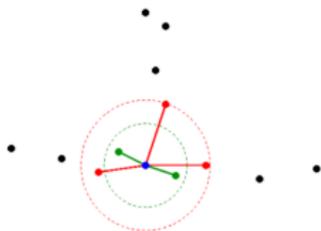
## Reconstruction de maillages

## Application aux arbres forestiers

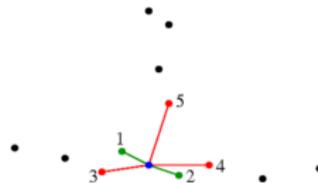
## Conclusion

- ▶ Pas d'information explicite de voisinage (connectivité)  
⇒ **distance Euclidienne** seule possibilité
- ▶ Hypothèse : nuage de points suffisamment **dense**

- ▶ Pas d'information explicite de voisinage (connectivité)  
⇒ **distance Euclidienne** seule possibilité
- ▶ Hypothèse : nuage de points suffisamment **dense**
- ▶ Méthodes :

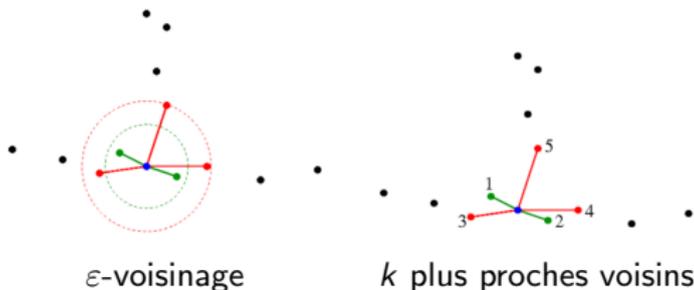


$\varepsilon$ -voisinage



$k$  plus proches voisins

- ▶ Pas d'information explicite de voisinage (connectivité)  
⇒ **distance Euclidienne** seule possibilité
- ▶ Hypothèse : nuage de points suffisamment **dense**
- ▶ Méthodes :



- ▶ Supposent une distribution **isotropique** des points
- ▶ Calcul rapide avec des structures **hiérarchiques** (octree, ...)

- ▶ Objectif : estimation **rapide** en tout point  $x_i$  :
  - ▶ du plan tangent
  - ▶ de la normale à la surface
  - ▶ des courbures

- ▶ Objectif : estimation **rapide** en tout point  $x_i$  :
  - ▶ du plan tangent
  - ▶ de la normale à la surface
  - ▶ des courbures
- ▶ Utilisation du **voisinage**  $N(x_i)$  de  $x_i$  tel que défini précédemment
- ▶ Outil : analyse de **covariance**

- ▶  $x_i$  point du nuage,  $\bar{x}$  **centroïde** (isobarycentre) du voisinage  $N(x_i)$  de  $x_i$  :

$$\bar{x} = \frac{1}{|N(x_i)|} \sum_{x_j \in N(x_i)} x_j$$

- ▶ **Matrice de covariance** :

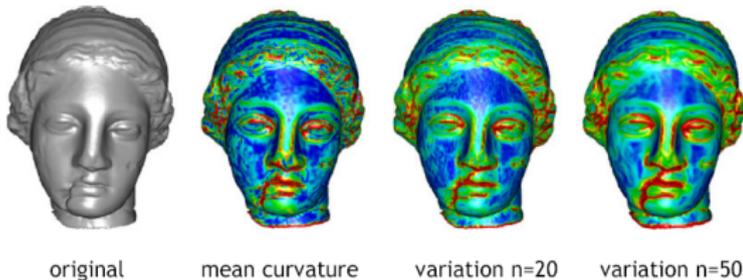
$$C = \begin{pmatrix} x_{j_1} - \bar{x} \\ \vdots \\ x_{j_k} - \bar{x} \end{pmatrix}^T \cdot \begin{pmatrix} x_{j_1} - \bar{x} \\ \vdots \\ x_{j_k} - \bar{x} \end{pmatrix}$$

- ▶ Valeurs propres  $\lambda_0 \leq \lambda_1 \leq \lambda_2$ , vecteurs propres associés  $v_0$ ,  $v_1$  et  $v_2$
- ▶ Chaque  $\lambda_j$  mesure la **variation** de  $N(x_i)$  le long de  $v_j$
- ▶ Variation totale :

$$\lambda_0 + \lambda_1 + \lambda_2 = \sum_{x_j \in N(x_i)} \|x_j - \bar{x}\|^2$$

- ▶ **Normale** à la surface :  $v_0$
- ▶ **Plan tangent** : plan défini par  $v_1$  et  $v_2$  passant par  $x_i$
- ▶ Axe de plus grande variation :  $v_2$

- ▶ **Normale** à la surface :  $v_0$
- ▶ **Plan tangent** : plan défini par  $v_1$  et  $v_2$  passant par  $x_i$
- ▶ Axe de plus grande variation :  $v_2$
- ▶ **Variation de surface** :  $\sigma(x_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$ 
  - ▶  $\sigma(x_i) = 0$  : voisinage plan
  - ▶  $\sigma(x_i) = 1/3$  : distribution isotropique sur une sphère
  - ▶ Approximation de la **courbure moyenne**



$n$  = nombre de points dans le voisinage [Pauly et al. 2002]

- ▶ **Inconvénient** calcul plan tangent : cas des arêtes vives
- ▶ Autres approches :
  - ▶ **Triangulation** locale
  - ▶ Calcul **quadrique** approchante (fonction de hauteur)
- ▶ Besoin de spécifier le **voisinage**

- ▶ Matrice de covariance des **normales** aux points :  $C' = \sum_{x_j \in N(x_i)} n_j^T \cdot n_j$
- ▶ Valeurs propres  $\lambda'_0 \leq \lambda'_1 \leq \lambda'_2$ , vecteurs propres  $v'_0, v'_1$  et  $v'_2$

- ▶ Matrice de covariance des **normales** aux points :  $C' = \sum_{x_j \in N(x_i)} n_j^T \cdot n_j$
- ▶ Valeurs propres  $\lambda'_0 \leq \lambda'_1 \leq \lambda'_2$ , vecteurs propres  $v'_0, v'_1$  et  $v'_2$
- ▶ Variation de normale :  $\sigma'(x_i) = \lambda'_1$
- ▶ [Garland 1999] : si la surface sous-jacente est lisse alors, quand l'échantillonnage  $\rightarrow \infty$ ,
  - ▶  $v'_2$  converge vers la normale moyenne
  - ▶  $v'_1$  converge vers la **direction de courbure maximum**
  - ▶  $v'_0$  converge vers la **direction de courbure minimum**

▶ **Opérateur de Laplace** (Laplacien) :

- ▶  $f$  fonction différentiable de deux variables dans un espace euclidien
- ▶  $\Delta f = \operatorname{div}(\operatorname{grad} f) = f_{uu} + f_{vv}$

► **Opérateur de Laplace** (Laplacien) :

- ▶  $f$  fonction différentiable de deux variables dans un espace euclidien
- ▶  $\Delta f = \operatorname{div}(\operatorname{grad} f) = f_{uu} + f_{vv}$

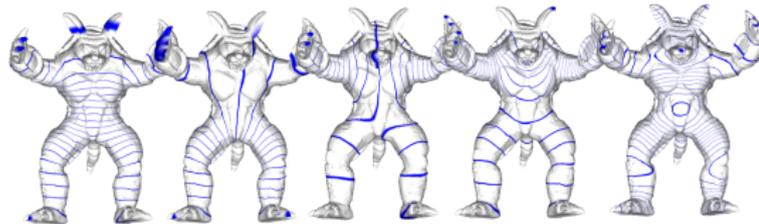
► **Opérateur de Laplace-Beltrami** :

- ▶  $f$  définie sur une surface  $\mathbf{X}$  variété lisse
- ▶  $\Delta_{\mathbf{X}} f = \operatorname{div}_{\mathbf{X}}(\operatorname{grad}_{\mathbf{X}} f)$
- ▶ Opérateur intrinsèque
- ▶ Pour  $\mathbf{u}$  la fonction qui associe à tout point de  $\mathbf{X}$  ses coordonnées :

$$\Delta_{\mathbf{X}} \mathbf{u} = -2H\mathbf{n}$$

- ▶ **Variation locale** de la surface
  - ▶ Applications : édition, transfert de détails, interpolation, ...

- ▶ **Variation locale** de la surface
  - ▶ Applications : édition, transfert de détails, interpolation, ...
- ▶ **Décomposition spectrale** : directions “intrinsèques” principales de la forme



[Lévy 2006]

- ▶ Descripteur de forme (“Shape DNA” [Reuter et al. 2006])
- ▶ Filtrage, débruitage
- ▶ Compression, segmentation, ...

- ▶ S'exprime généralement sous la forme

$$\Delta f(x_i) = \frac{1}{W_i} \sum_{x_j \in N(x_i)} w_{ij} (f(x_j) - f(x_i))$$

- ▶ S'exprime généralement sous la forme

$$\Delta f(x_i) = \frac{1}{W_i} \sum_{x_j \in N(x_i)} w_{ij} (f(x_j) - f(x_i))$$

- ▶ [M. Belkin, J. Sun and Y. Wang SoDA 2009] :
  1. Calcul plan tangent  $\Pi_x$  en tout point  $x$  (minimise distance de Hausdorff au  $\varepsilon$ -voisinage)
  2. Calcul triangulation de Delaunay  $T_x$  du voisinage projeté sur  $\Pi_x$
  3.  $w_{ij} =$  aire faces incidentes au projeté de  $x_j$  dans  $T_x \times$  poids gaussien (distance  $x_j$  à  $x_i$ )

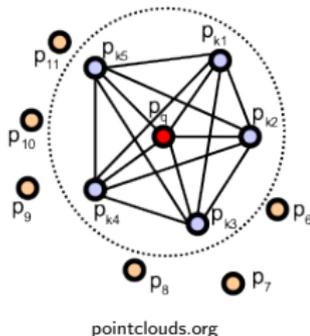
- ▶ S'exprime généralement sous la forme

$$\Delta f(x_i) = \frac{1}{W_i} \sum_{x_j \in N(x_i)} w_{ij} (f(x_j) - f(x_i))$$

- ▶ [M. Belkin, J. Sun and Y. Wang SoDA 2009] :
  1. Calcul plan tangent  $\Pi_x$  en tout point  $x$  (minimise distance de Hausdorff au  $\varepsilon$ -voisinage)
  2. Calcul triangulation de Delaunay  $T_x$  du voisinage projeté sur  $\Pi_x$
  3.  $w_{ij} =$  aire faces incidentes au projeté de  $x_j$  dans  $T_x \times$  poids gaussien (distance  $x_j$  à  $x_i$ )
- ▶ **Converge** vers l'opérateur lisse quand le nuage de points devient de plus en plus dense
- ▶ Extensions : [Liu et al. 2012] (symétrique), [Petronetto et al. 2013] (sans maillage), [Qin et al. 2018] (arêtes vives)

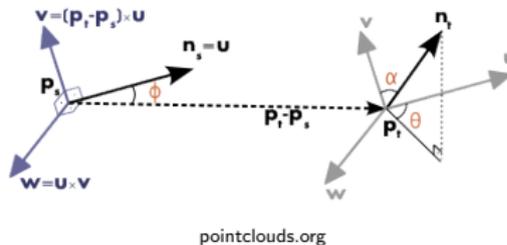
- ▶ Limitation normales et courbures : approximation **grossière** du voisinage d'un point
  - ▶ Beaucoup de points peuvent avoir la même normale ou courbure

- ▶ Limitation normales et courbures : approximation **grossière** du voisinage d'un point
  - ▶ Beaucoup de points peuvent avoir la même normale ou courbure
- ▶ Objectif : analyser **toutes** les relations entre les  $k$  plus proches voisins d'un point  $p$ 
  - ▶ Complexité  $k^2$



B. Rusu et al., "Aligning Point Cloud Views using Persistent Feature Histograms", IROS 2008

- ▶ Différence entre les **normales** à deux points  $p_s$  et  $p_t$  :



- ▶  $\alpha = v \cdot n_t$ ,  $\phi = u \cdot \frac{p_t - p_s}{\|p_t - p_s\|}$ ,  $\theta = \arctan(w \cdot n_t, u \cdot n_t)$
- ▶ Le quadruplet  $(\alpha, \phi, \theta, \|p_t - p_s\|)$  représente l'estimateur de différence entre  $p_s$  et  $p_t$
- ▶ Les quadruplets des couples de voisins d'un point  $p$  sont mis sous la forme d'un **histogramme**

Contexte

Géométrie différentielle discrète

## **Reconstruction de maillages**

Types d'approches

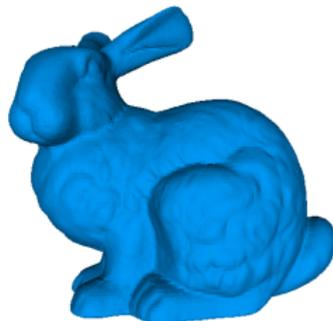
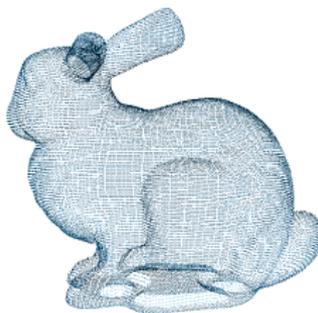
Focus : méthodes par surface implicite

Conclusion

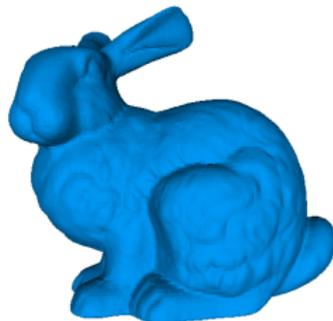
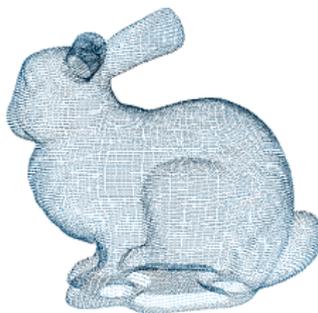
Application aux arbres forestiers

Conclusion

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



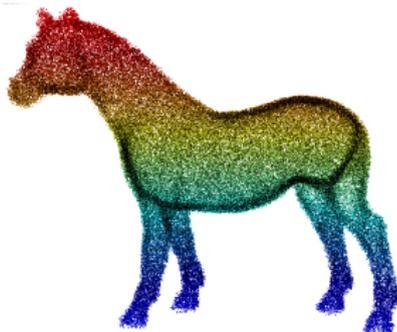
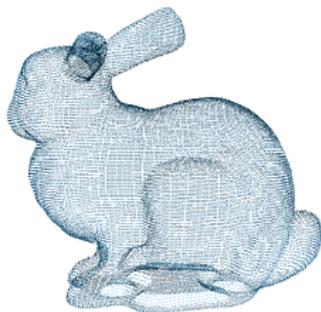
D'un **nuage de points** 3D à une surface **maillée**



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

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



## ▶ Triangulation de Delaunay

- ▶ Définition : aucun sommet dans le cercle circonscrit à un triangle
- ▶ Propriété fondamentale : maximise l'angle minimum

- ▶ **Triangulation de Delaunay**

- ▶ Définition : aucun sommet dans le cercle circonscrit à un triangle
- ▶ Propriété fondamentale : maximise l'angle minimum

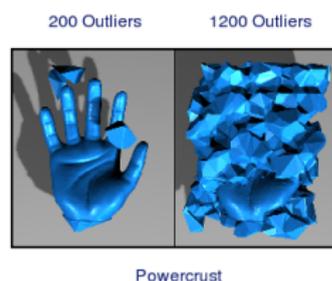
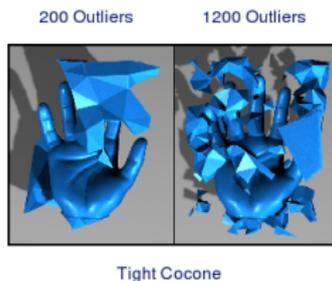
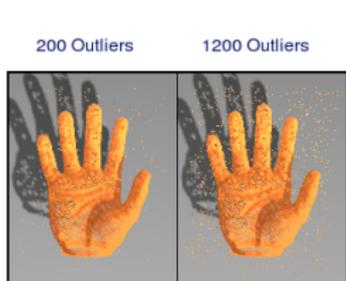
- ▶ Grande partie des méthodes existantes

- ▶ Intérêt : preuves théoriques de **validité**

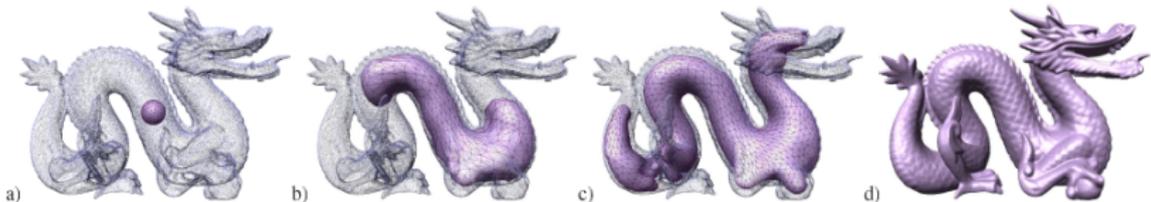
- ▶ Topologie, distance Euclidienne aux échantillons, distance des normales
- ▶ Sous **conditions** (pas/peu de bruit, ...)

- ▶ [http://interstices.info/display.jsp?id=c\\_12845](http://interstices.info/display.jsp?id=c_12845)

- ▶ **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

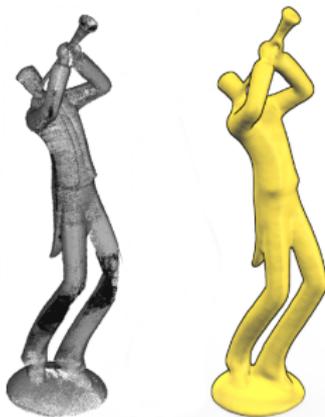


- ▶ Idée : partir d'une forme simple et petite (sphère par exemple) à l'intérieur de l'objet et la **déformer** pour qu'elle aille plaquer la surface
- ▶ Difficulté : établir les équations de déformation
- ▶ Suppose la surface fermée (**watertight**)
- ▶ Robuste au données manquantes
- ▶ Exemple : A.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



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 aux données manquantes
- ▶ Exemple : A. Tagliasacchi et al., "VASE : Volume-Aware Surface Evolution for Surface Reconstruction from Incomplete Point Clouds", Symposium on Geometry Processing 2011



► 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

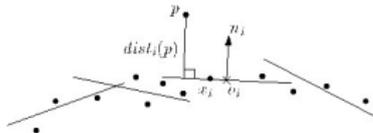
- ▶ Idée :
  1. Définir une **surface implicite lisse** qui approche la surface réelle sous-jacente
  2. **Projeter** ou générer les/des points sur cette surface et trianguler
- ▶ Problème principal : **définition** de la surface implicite
  - ▶ Beaucoup de possibilités : fonction de distance, MLS, RBF,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

- ▶ <http://hhoppe.com/proj/recon/>

- ▶ <http://hhoppe.com/proj/recon/>
- ▶ **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

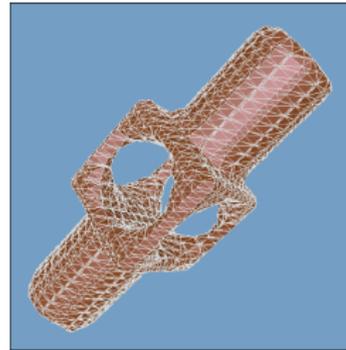
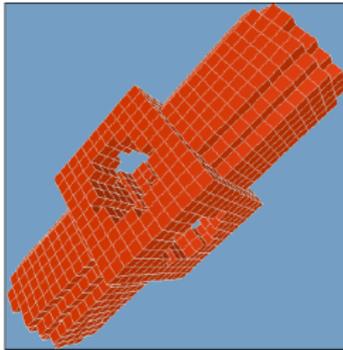
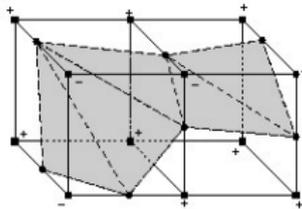
- ▶ <http://hhoppe.com/proj/recon/>
- ▶ **Entrée** :
  - ▶ Points avec coordonnées 3D, pas de normale
  - ▶ **Topologie**/bords : **arbitraires**, inférés depuis le nuage de points
  - ▶ **Densité** d'échantillonnage : supposée **uniforme**, donnée en paramètre
  - ▶ Erreur max par rapport à la surface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

- ▶ 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**

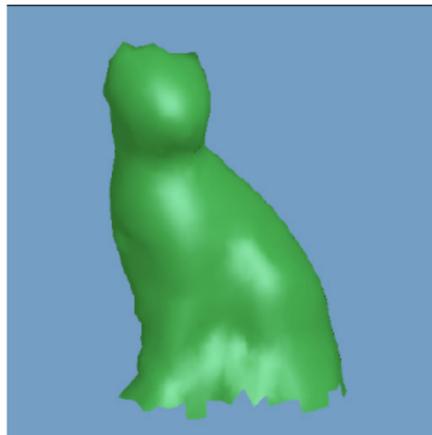
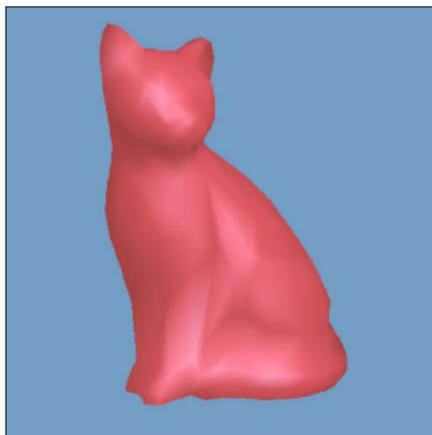


4. Extraction d'une **isosurface**

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



- ▶ 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)!



- ▶ **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

- ▶ **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 :  $\phi(r) = r^n$

$$\blacktriangleright f(x) = \sum_{i=1}^n \lambda_i \phi(\|x - x_i\|) + P(x)$$

- ▶ On suppose  $f$  connue en les  $x_i$ , les  $\lambda_i$  inconnus
- ▶  $P$  polynôme, dépend du choix des RBF, nécessaire pour trouver les  $\lambda_i$

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

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

avec :

- ▶  $\Phi(i, j) = \phi(\|x_i - x_j\|)$
- ▶  $X(i, \cdot) = (1 \ x_i \ y_i \ z_i)$
- ▶  $\lambda$  vecteur des  $\lambda_i$
- ▶  $c$  vecteur des coefficients de  $P$
- ▶  $f$  vecteur des  $f(x_i)$

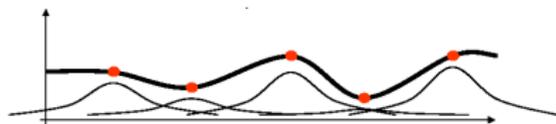
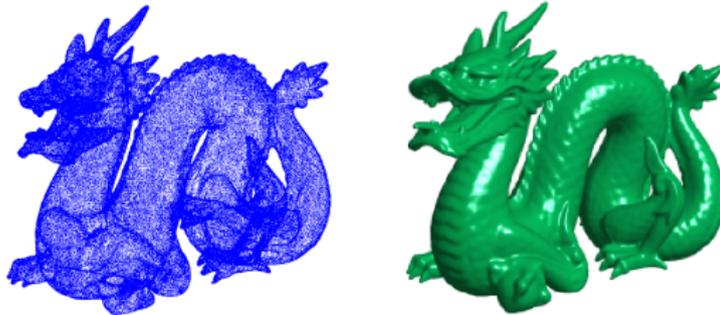


Figure par Marc Alexa

- ▶ 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



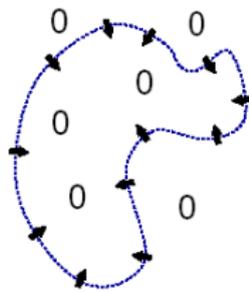
[Carr et al. 2003]

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



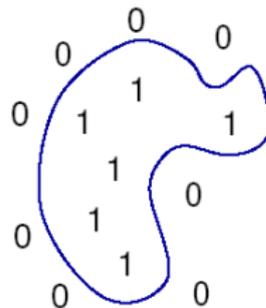
Oriented points

$$\vec{V}$$



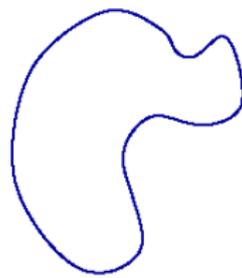
Indicator gradient

$$\nabla \chi_M$$



Indicator function

$$\chi_M$$



Surface

$$\partial M$$

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

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

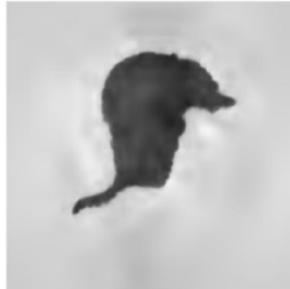
$$\Delta\chi = \nabla \cdot \nabla\chi = \nabla V$$

- ▶ Equation de diffusion :  $\Delta\chi = 0$

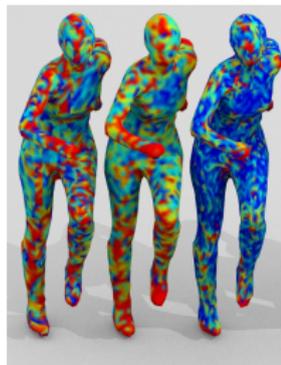
- ▶ Implémentation **efficace** possible
  - ▶  $\chi$  décomposée dans une base de fonctions **à support compact**  $\Rightarrow$  système matriciel creux
  - ▶ Discrétisation de l'espace adaptative : **octree** plutôt qu'une grille uniforme
- ▶ **Code** disponible
  - ▶ <http://www.cs.jhu.edu/~misha/Code/PoissonRecon>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

- ▶ Implémentation **efficace** possible
  - ▶  $\chi$  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

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

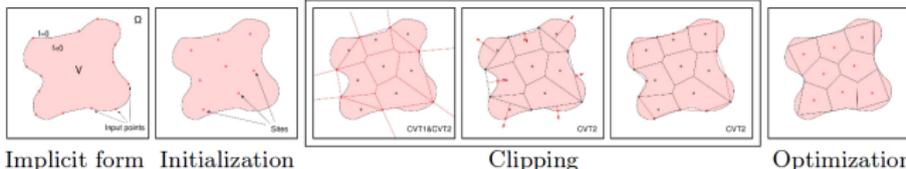


- ▶ **Centroidal Voronoi Tessellation** : itérations calcul Voronoï/déplacement des germes
- ▶ Calcul dans le volume et coupure (**clipping**) au niveau de la surface implicite
- ▶ Frontière CVT = **meilleure approximation** de la surface implicite que Marching Cubes



(MC) (Del) (CVT1)

[Wang, HW, Boyer 2016]



- ▶ **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

Contexte

Géométrie différentielle discrète

Reconstruction de maillages

### **Application aux arbres forestiers**

Besoins et applications

“Amélioration” du nuage de points

Perspectives

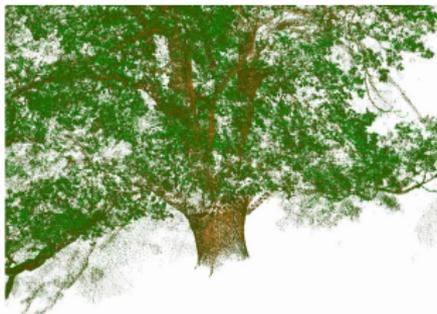
Conclusion

- ▶ **Collecte** et **évaluation** de données forestières
  - ▶ Distribution des espèces, qualité du bois, santé, diamètre à hauteur de poitrine, ...
  - ▶ Mesure de **biomasse** : volume de bois et indice de surface foliaire (LAI)



Projet PlantScan3D

- ▶ **Gestion durable**
  - ▶ Qualité du bois, carte de risque de départ de feu, suivi des maladies/nuisibles, ...
- ▶ Surveillance du **changement climatique**
  - ▶ Effet de serre
  - ▶ Flux de carbone



(c) Eric Casella

<https://www.youtube.com/watch?v=1pEFIXSgBvM>

Contexte

Géométrie différentielle discrète

Reconstruction de maillages

## **Application aux arbres forestiers**

Besoins et applications

“Amélioration” du nuage de points

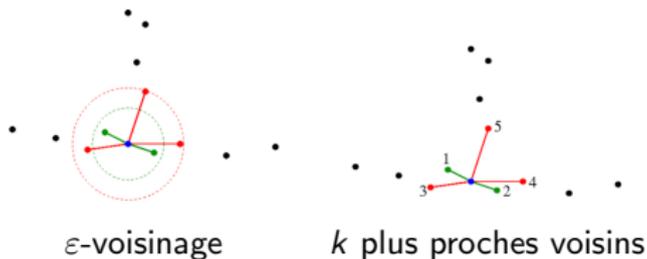
Débruitage

Segmentation

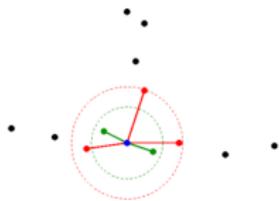
Fitting de primitives

Perspectives

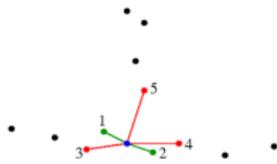
Conclusion



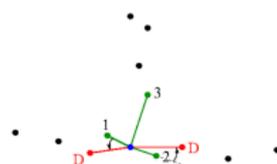
- ▶ Pas assez de voisins : manque d'information
- ▶ Trop de voisins : lent



$\epsilon$ -voisinage



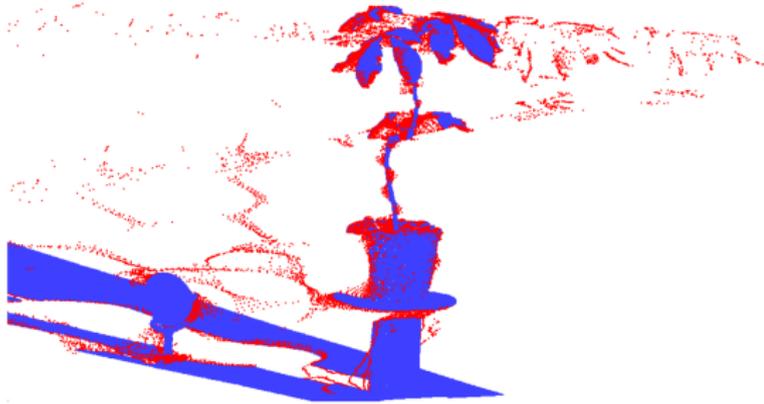
$k$  plus proches voisins



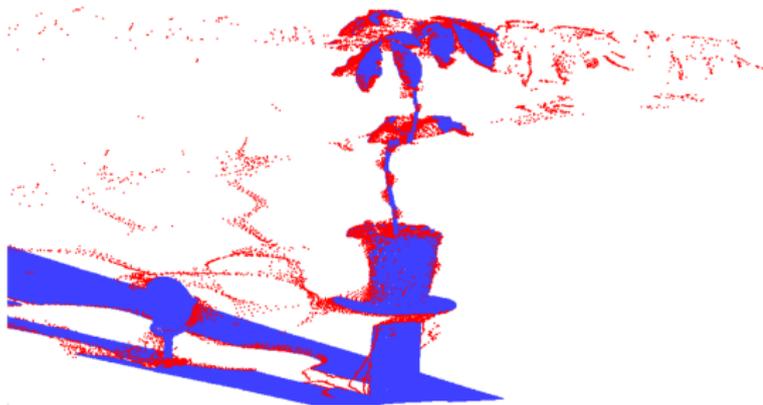
[HW et al. IJRS 2016]

- ▶ Pas assez de voisins : manque d'information
- ▶ Trop de voisins : lent
- ▶ Paramètre supplémentaire : **angle** minimum entre voisins
- ▶ Adapté aux **branches**

- ▶ Très présent dans le cas des arbres



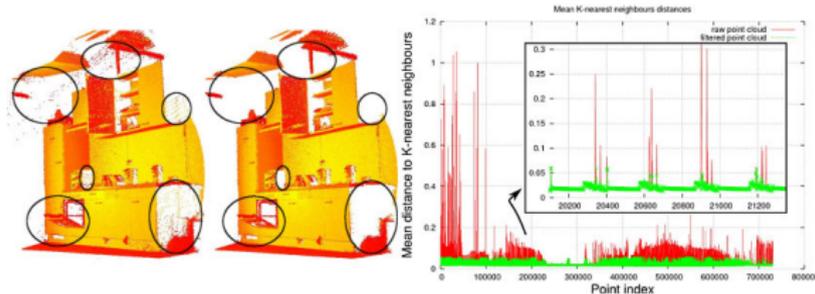
- ▶ Très présent dans le cas des arbres



- ▶ **Bruit de pixel mixte** [Hébert and Krovov 1992] : plusieurs contacts laser/surface
- ▶ **Bruit de ciel** (scanners à décalage de phase) : pas de retour

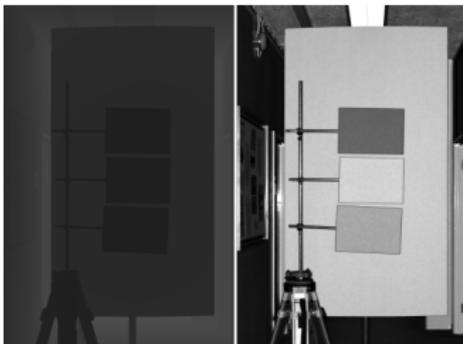
Suppression des points isolés (**outliers**) :

- ▶ Supprimer tous les points n'ayant pas au moins  $k$  voisins dans un rayon  $r$
- ▶ Ou : supprimer tous les points dont la distance moyenne aux  $k$  voisins n'est pas dans un intervalle de confiance

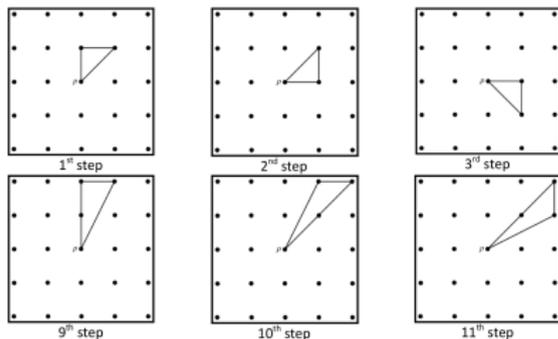


pointclouds.org

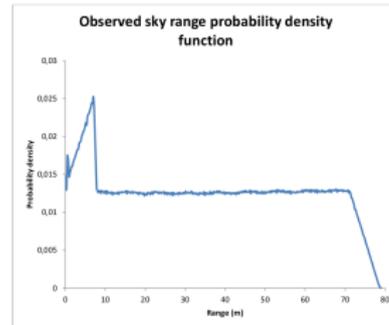
- ▶ Doctorat **Romain Rombourg** (2015-2018), co-encadrement **Eric Casella** (Forestry Commission, Royaume-Uni)
- ▶ Article en cours de rédaction
- ▶ Idée : travailler sur l'**image de profondeur** et l'**image d'intensité** issues du scan



- ▶ Image de **profondeur**
- ▶ **Angles** entre la direction du laser et les normales aux triangles ( $p$ , deux voisins)
- ▶  $p$  = bruit si plus de la moitié des angles au-dessus d'un seuil
- ▶ Exemple de résultat :



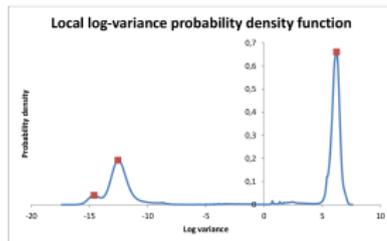
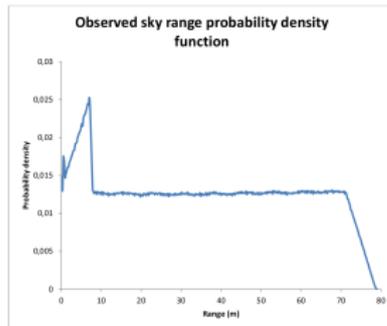
- ▶ Images d'**intensité** et de **profondeur**
- ▶ Ciel = faible intensité et grande variance de profondeur



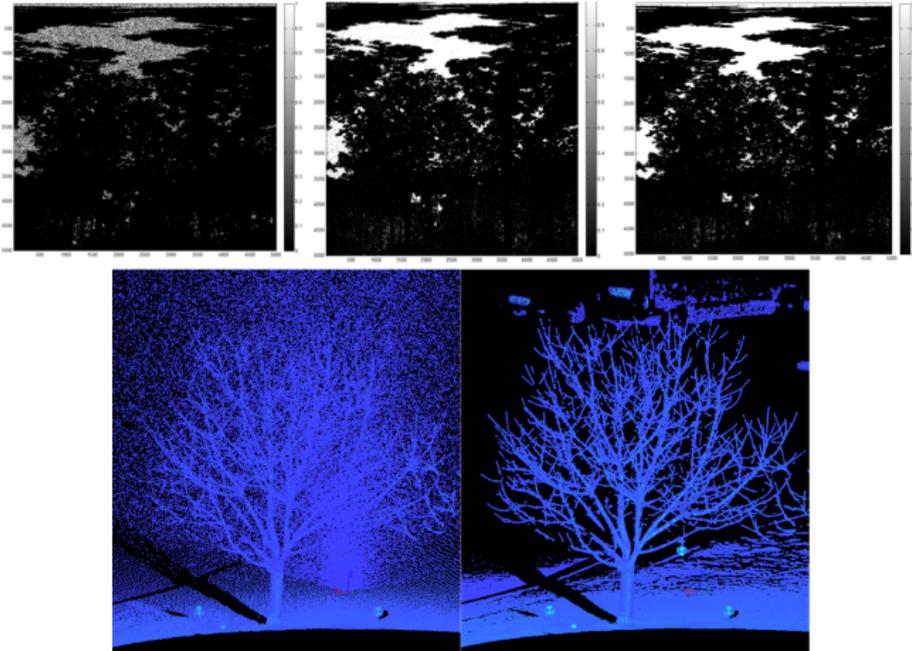
- ▶ Images d'**intensité** et de **profondeur**
- ▶ Ciel = faible intensité et grande variance de profondeur

$\rho$  = bruit si :

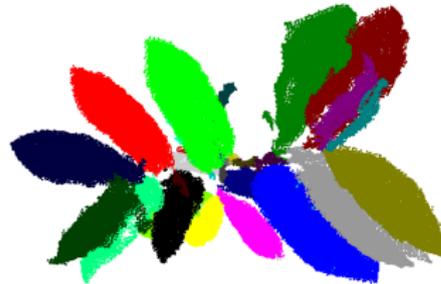
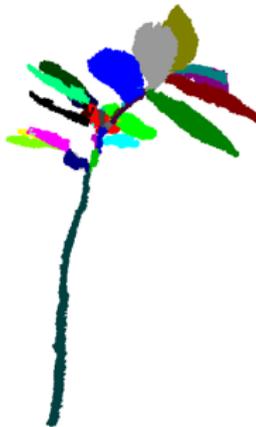
1. Variance de profondeur dans le plus grand mode
2. Intensité < seuil calculé à partir des points précédents
3. > 50% des voisins sont du bruit



## Quelques résultats



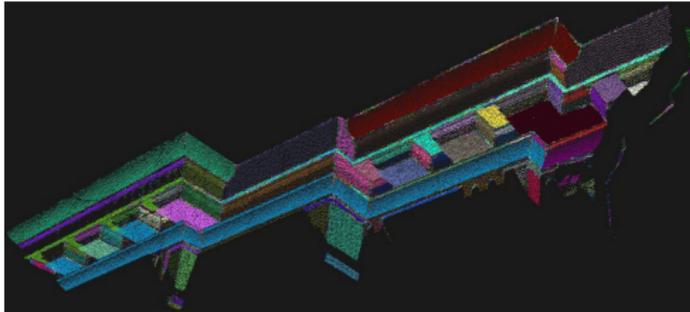
- ▶ **Décomposition** de la forme en régions sémantiques
  - ▶ Arbre : tronc, branches, feuilles, fruits bourgeons, ...
  - ▶ Forêt ou parcelle : arbres
- ▶ Intérêt : calculs spécifiques
  - ▶ Volume de bois, surface foliaire





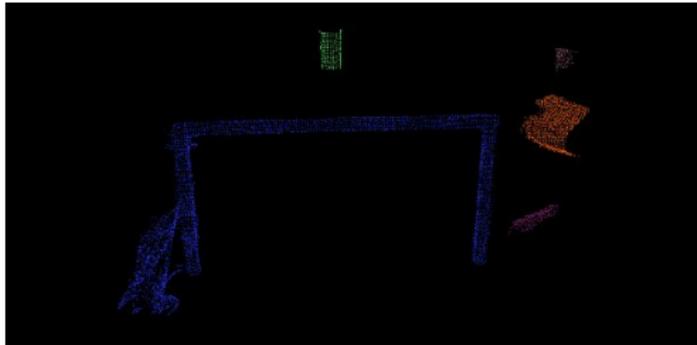
- ▶ Nombreuse !
- ▶ Overview : [Nguyen and Le ICRAM 2013]
- ▶ Quelques types d'approches :
  - ▶ Croissance de région
  - ▶ Extraction de cluster
  - ▶ Différence de normales

1. Choix d'une **graine**  $g$  parmi les points non traités
  - ▶ Point dans zone plane (courbure faible)
2. Nouvelle région  $R \leftarrow \{g\}$ , liste des graines  $G \leftarrow \{g\}$
3. Pour toute graine  $g'$  dans  $G$  :
  - 3.1 Pour tout voisin  $p$  de  $g'$  :
    - 3.1.1 Si normale de  $p$  proche de la normale de  $g'$  alors ajout de  $p$  à  $R$
    - 3.1.2 Si courbure de  $p$  faible alors ajout de  $p$  à  $G$
4. Recommencer avec une nouvelle région et une nouvelle graine



pointclouds.org

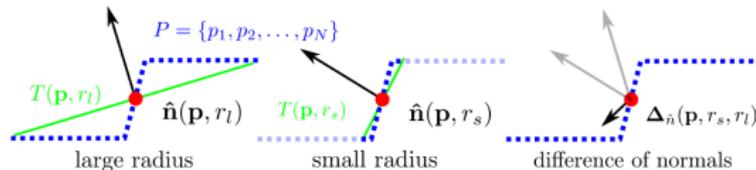
1. **File de priorité**  $Q$  sur les points (initialement vide)
2. Pour tout point  $p$  du nuage pas encore traité :
  - 2.1 Ajout de  $p$  à  $Q$
  - 2.2 Pour tout  $p'$  dans  $Q$  :
    - 2.2.1 Ajout des  $\varepsilon$ -voisins de  $p'$  dans  $Q$  (si pas déjà traités)
  - 2.3 Nouveau cluster  $\leftarrow Q$ , et  $Q \leftarrow \emptyset$



pointclouds.org

Y. Ioannou et al., "Difference of Normals as a Multi-Scale Operator in Unorganized Point Clouds", 3DIMPVT 2012

1. Calcul de la normale au point considéré à **deux échelles différentes**
2. Calcul du vecteur **différence** entre les deux



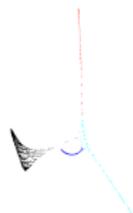
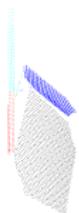
3. Segmentation (extraction de cluster) du champ de vecteurs résultant



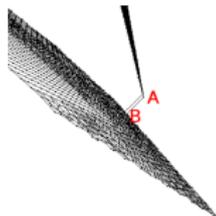
- ▶ Arbre : pas de grandes zones **planes** ...
- ▶ ... mais **directions privilégiées** (même pour les feuilles)
- ▶ Problème : ces directions sont “**courbes**”

- ▶ Arbre : pas de grandes zones **planes** ...
- ▶ ... mais **directions privilégiées** (même pour les feuilles)
- ▶ Problème : ces directions sont “**courbes**”
- ▶  $\Rightarrow$  théorie du **regroupement spectral** (spectral clustering)

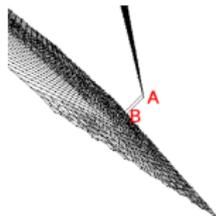
- ▶  $L$  matrice **Laplacienne** sur les sommets
  - ▶  $L = D - W$  matrice des degrés –  $W$  matrice d'adjacence
- ▶  $L'$  Laplacien normalisé =  $D^{-1}L$
- ▶ **Plongement** défini par les  $k$  premiers **vecteurs propres**  $u_1, \dots, u_k$  de  $L'$ 
  - ▶ Coordonnées du point numéro  $i$  = ligne  $i$  de la matrice  $U = (u_1, \dots, u_k)$
- ▶ Clustering : algorithme des  **$k$ -moyennes** sur les lignes de  $U$



- ▶ **Poids** des arêtes : distance **commute-time**
  - ▶  $\sim L^2$  dans l'espace spectral (pondéré par les valeurs propres)



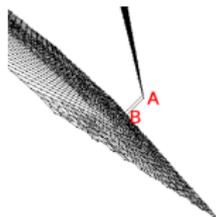
- ▶ **Poids** des arêtes : distance **commute-time**
  - ▶  $\sim L^2$  dans l'espace spectral (pondéré par les valeurs propres)



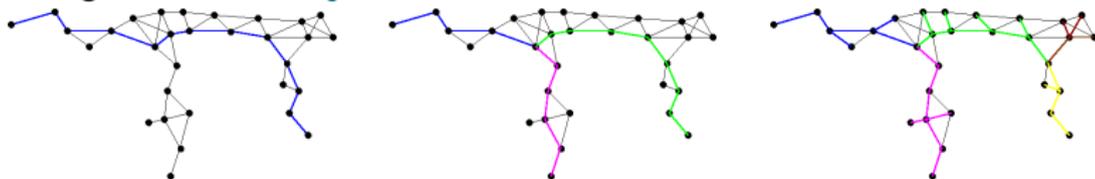
- ▶ Problème  $k$ -moyennes : partitionnement isotropique
- ▶  $\Rightarrow$  algorithme basé **Dijkstra**



- ▶ **Poids** des arêtes : distance **commute-time**
  - ▶  $\sim L^2$  dans l'espace spectral (pondéré par les valeurs propres)



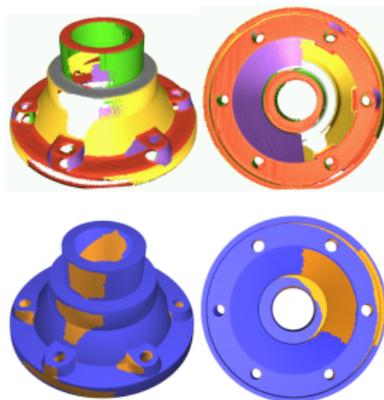
- ▶ Problème  $k$ -moyennes : partitionnement isotropique
- ▶  $\Rightarrow$  algorithme basé **Dijkstra**



- ▶ Plus de détails : voir [HW, Casella and Boltcheva IJRS 2016]

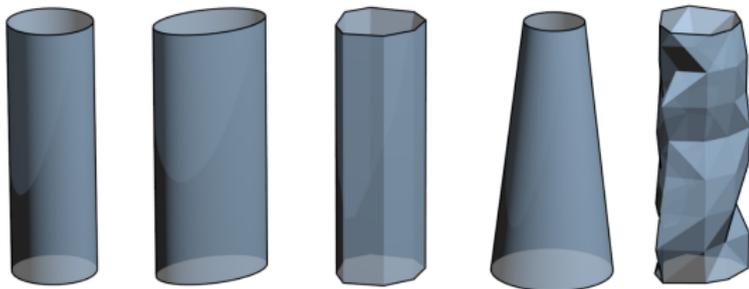
Idée :

- ▶ 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



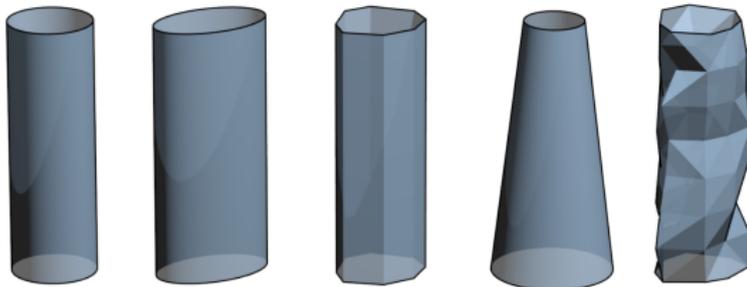
M. Åkerblom et al., "Analysis of Geometric Primitives in Quantitative Structure Models of Tree Stems", Remote Sensing 2015

- **Comparaison** primitives pour modélisation branches à partir de points



M. Åkerblom et al., "Analysis of Geometric Primitives in Quantitative Structure Models of Tree Stems", Remote Sensing 2015

- ▶ **Comparaison** primitives pour modélisation branches à partir de points



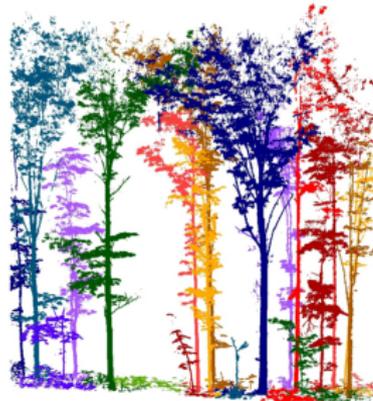
- ▶ **Cylindre circulaire** à privilégier
  - ▶ Simple : **robuste** au bruit et données manquantes
  - ▶ Erreur reste faible
- ▶ Solutions plus complexes pour les troncs

## ► Segmentation :

- D'un arbre en ses organes (branches, feuilles)
- D'un ensemble/d'une **placette forestière** en ses arbres
- **Heuristiques** botaniques, informations géométriques et d'intensité

## ► Calcul de **primitives** pour les **feuilles**

- Surfaces ouvertes, trous
- Maillage, spline/**NURBS**, ensemble de quadriques



(c) J. Ravaglia et al.

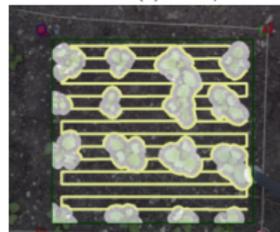


[Chaurasia/Beardsley 2017]

- ▶ **Céréales** (thèse CIFRE Mélinda Boukhana, Arvalis/INRA Avignon)
  - ▶ **Acquisition** en plein champ
  - ▶ **Identification** et caractérisation des **organes** (épis, feuilles, tiges)
- ▶ **Légumes** vs. mauvaises herbes (projet H2020 ROMI, Inria/CNRS Lyon/Sony/etc.)
  - ▶ **Segmentation** en organes
  - ▶ **Suivi** temporel



Phenomobile, (c) INRA/Arvalis



ROMI, (c) Sony CSL

Contexte

Géométrie différentielle discrète

Reconstruction de maillages

Application aux arbres forestiers

Conclusion

On peut travailler directement sur les nuages de points mais :

- ▶ Besoin d'information de **voisinage**
- ▶ Besoin d'information de **normales**
- ▶ **Maillage** utile pour des opérations globales ou évoluées (analyse de forme, modification, déformation, ...)

Joyeuses Pâques

