

Géométrie numérique

Séance 6 – Traitement des nuages de points

Franck Hétroy-Wheeler

M1 I3D et IIRVIJ – 2018-2019



Rappel

Principe

La surface est définie par un ensemble de points supposés en faire partie.

Très répandu grâce aux scanners laser et caméras RGBD (Kinect)



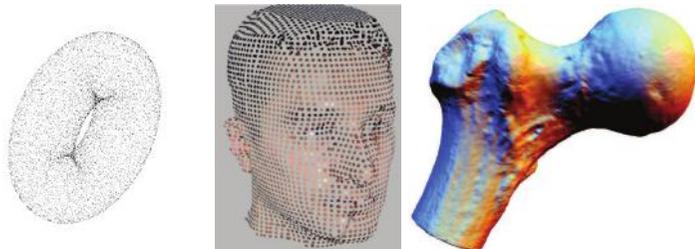
Par rapport aux autres représentations

Avantages :

- ▶ Donnée **simple** et compacte
- ▶ **Visualisation** rapide

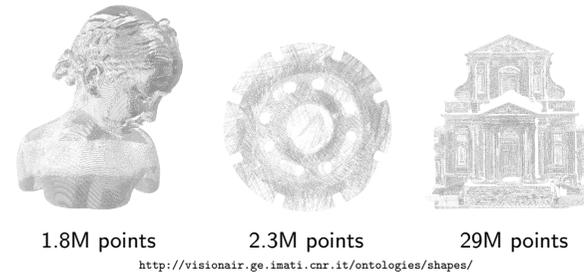
Inconvénients :

- ▶ Pas de **structure** / d'information topologique (voisinages)
- ▶ Densité nécessaire



Intérêt

- ▶ Modèles scannés : habituellement des **millions** voire des milliards de points



- ▶ **Reconstruction** de maillage trop coûteuse en temps/place mémoire
- ▶ Chaque **face** d'un maillage ne serait projetée que sur **ou deux pixel(s)** de l'image

Contexte

Représentation par points

Géométrie différentielle discrète

Descripteurs de forme

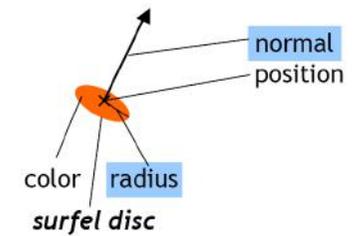
Conclusion

Surfel

- ▶ Proposé par H. Pfister et al. à SIGGRAPH 2000 ("Surfels : Surface Elements as Rendering Primitives")
- ▶ Définition : élément de **surface** (analogie pixel/voxel)
- ▶ En fait un **disque**

Attributs :

- ▶ Position 3D
- ▶ Couleur
- ▶ **Normale**
- ▶ **Rayon**



Intérêt

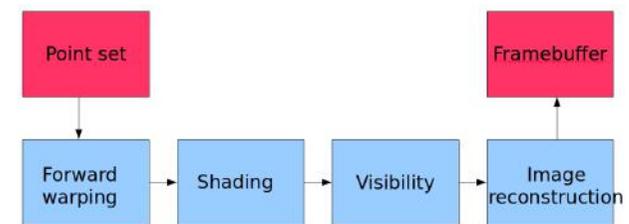
▶ Rendu

- ▶ Objectif d'origine
- ▶ L'utilisateur veut voir une **surface** (pas de trou)
- ▶ Tout un pipeline de calcul (cf. slide suivant)



- ▶ Objet en **mouvement** avec notamment changement de topologie
 - ▶ Fluide

Pipeline de rendu

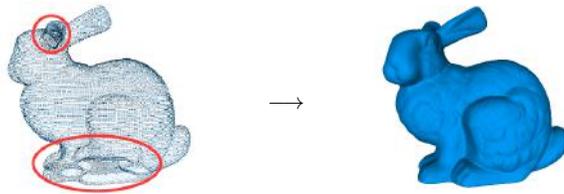


[M. Zwicker et al. 2002]

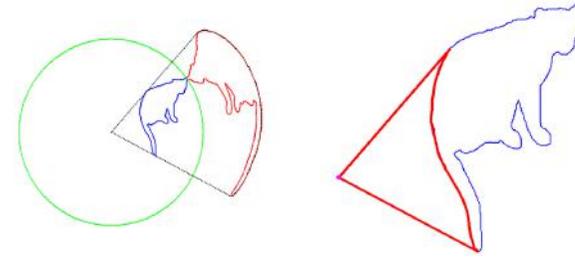
Deux premières étapes classiques (cf. UE Programmation graphique 3D) :

1. Projection perspective dans le plan image
2. Phong, etc.

- ▶ Calculées simultanément
- ▶ Objectif 1 : éliminer les points non vus depuis la caméra
- ▶ Objectif 2 : reconstruire une surface continue à partir des points vus



- ▶ Opérateur "Hidden Point Removal" (HPR) [Katz et al. 2007]



- ▶ Autre solution : Z-buffer après calcul des surfels

Exercice

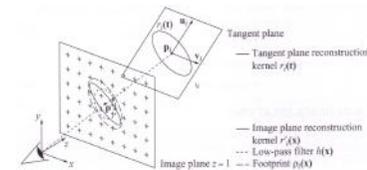
Écrire l'algorithme de calcul de la visibilité par HPR en pseudo-langage.

1. Placer l'origine C du repère au point de vue (translation des points)
2. Calculer la plus grande distance R entre C et le nuage de points
3. **Spherical flipping** de tout point p_i : $f(p_i) = p_i + 2(R - \|p_i\|) \frac{p_i}{\|p_i\|}$
4. Calculer l'**enveloppe convexe** de $C \cup \{f(p_i)\}$

Remarques :

- ▶ Rapide : $O(n \log n)$
- ▶ 8 lignes en Matlab
 - ▶ www.mathworks.com/matlabcentral/fileexchange/16581-hidden-point-removal

- ▶ Surface **splatting** [Zwicker et al. 2001]



- ▶ Calcul **normale** : cf. [Hoppe et al. 1992]
- ▶ Calcul **rayon** : de manière à couvrir la surface (pas de trou)
- ▶ **Splat** : **floutage** de chaque surfel autour du point selon un noyau gaussien



Contexte

Représentation par points

Géométrie différentielle discrète

- Voisinage
- Normales et courbures
- Opérateur de Laplace-Beltrami
- Approximation locale de la surface

Descripteurs de forme

Conclusion

On suppose que la **surface sous-jacente** est une **2-variété lisse**

1. Approximation du **voisinage** d'un point
2. Approximation locale du **plan tangent** et des courbures : **ACP**
 - ▶ Extension : approximation du **Laplacien**
3. Approximation locale de la **surface** autour d'un point : **MLS**

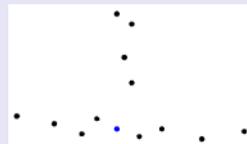
Voisinage

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

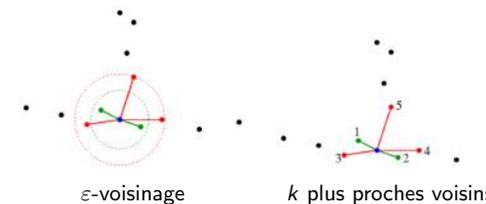
Exercice

Dessiner les voisins du point bleu pour :

- ▶ ϵ = distance au 1er point à droite puis ϵ = distance au 1er point au-dessus
- ▶ $k = 2$ puis $k = 5$



Voisinage



- ▶ 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
- ▶ 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** (taille 3×3) :

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

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

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

Exercice

Soit un carré échantillonné avec 12 points (i.e. 4 par côté), dont les coordonnées vont de (0, 0) à (3, 3).

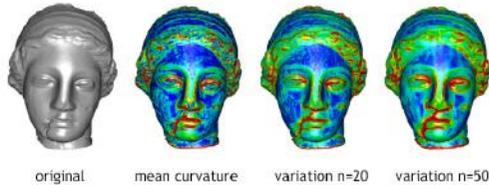
1. Calculer la matrice de covariance C de chacun des points, en considérant uniquement 2 voisins.
2. Montrer que les valeurs propres de C sont 0 et soit 1 soit 2.
3. Trouver les vecteurs propres associés. Que représentent-ils ?

▶ Point "intérieur" : $C = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$, $\lambda_0 = 0$, $v_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\lambda_1 = 2$, $v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

▶ Coin : $C = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$, $\lambda_0 = 0$, $v_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\lambda_1 = 1$, $v_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

▶ $v_0 =$ normale, $v_1 =$ tangente

- ▶ 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 : $\sigma(x_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$
 - ▶ $\lambda(x_i) = 0$: voisinage plan
 - ▶ $\lambda(x_i) = 1/3$: 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
- ▶ 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

- ▶ Rappel : 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 Π_x en tout point x (minimise distance de Hausdorff au ε -voisinage)
 2. Calcul triangulation de Delaunay T_x du voisinage projeté sur Π_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)

Contexte

Représentation par points

Géométrie différentielle discrète

- Voisinage
- Normales et courbures
- Opérateur de Laplace-Beltrami
- Approximation locale de la surface

Descripteurs de forme

Conclusion

Moindres carrés mobiles (Moving Least Squares)

- ▶ Approximation par moindres carrés pondérés **adaptée** autour du point auquel on se trouve

$$f_{MLS} : x \mapsto f_{WLS,x}(x)$$

- ▶ Théorème [Levin 1998] : f_{MLS} est C^1 si et seulement si θ est C^1

- ▶ Pour en savoir plus :

<http://www.nealen.com/projects/mls/asapmls.pdf>

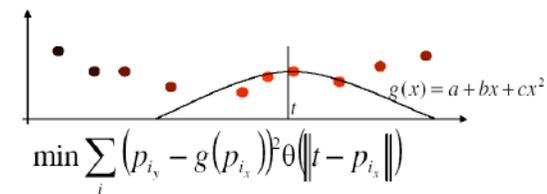


Figure : Marc Alexa

Moindres carrés pondérés (Weighted Least Squares)

$$f_{WLS,\bar{x}} = \operatorname{argmin} \sum_i \theta(\|\bar{x} - x_i\|) \|f(x_i) - f_i\|^2$$

- ▶ Les **poinds** θ dépendent de la **distance à un point** \bar{x}
- ▶ Décroissants et tendent vers 0
- ▶ Choix standard : exponentielle

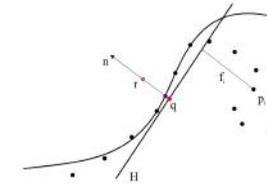
Application à la représentation par points

- ▶ M. Alexa et al., "Computing and Rendering Point Set Surfaces", TVCG 2003
- ▶ Objectif : surface S lisse **approchant** les points x_i
- ▶ Bruit



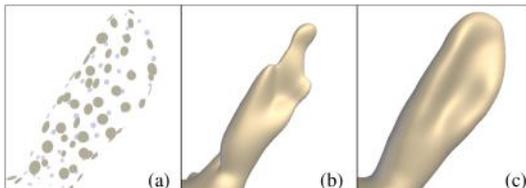
Méthode

1. Pour tout point r de l'espace, calcul d'un **plan** H par WLS
 - ▶ Ne passe pas nécessairement par r
2. Calcul de S (ici g) autour de r par **MLS**
 - ▶ Poids associés aux x_i (ici p_i) dépendent de la distance de leur **projection** sur H à la projection q de r sur H
3. Projection de r sur S (point bleu)



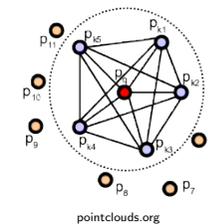
Extension : surfaces algébriques par points

- ▶ G. Guennebaud and M. Gross, "Algebraic Point Set Surfaces", SIGGRAPH 2007
- ▶ **Problème** approche Alexa : calcul du **plan** très **instable** dans les régions de courbure importante si échantillonnage insuffisant
- ▶ Idée : fitter plutôt une **sphère** (ou surface algébrique d'ordre supérieur)
- ▶ Également : algos plus robustes et plus rapides, prise en compte des arêtes vives



Caractéristiques de points (features)

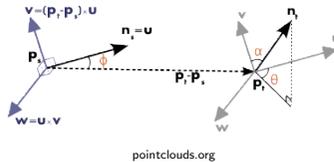
- ▶ Limitation normales et courbures : approximation **grossière** du voisinage d'un point
 - ▶ Beaucoup de points peuvent avoir la même normale ou courbure
- ▶ Objectif : analyser **toutes** les relations entre les k plus proches voisins d'un point p
 - ▶ Complexité k^2



Exemple : Point Feature Histogram (PFH)

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 :



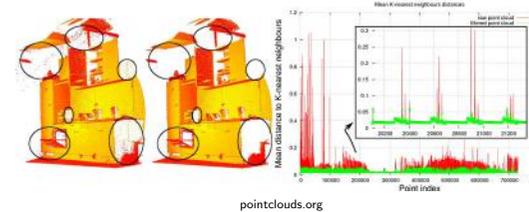
- ▶ $\alpha = v \cdot n_t$, $\phi = u \cdot \frac{p_t - p_s}{\|p_t - p_s\|}$, $\theta = \arctan(w \cdot n_t, u \cdot n_t)$

- ▶ 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**

Application : débruitage statistique

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

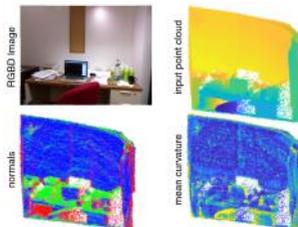
- ▶ Supprimer tous les points n'ayant pas au moins k voisins dans un rayon r
- ▶ Ou : supprimer tous les points dont la distance moyenne aux k voisins n'est pas dans un intervalle de confiance



pointclouds.org

Hot topic : deep learning

- ▶ Objectif : **apprendre** automatiquement des **propriétés géométriques** locales ou globales (segmentation, reconnaissance, ...)
- ▶ Idée : s'entraîner sur des **maillages** "propres" dont on connaît les propriétés pour apprendre sur des nuages de points **bruités**
- ▶ Méthodes récentes :
 - ▶ **Réseau de neurones** : PointNet/PointNet++ [Qi et al. 2017]
 - ▶ Calcul de **normales et courbures** : PCPNet [Guerrero et al. 2018]



[Guerrero et al. 2018]

Bilan

- ▶ Rendu par **surfels**
 - ▶ Normale, rayon
 - ▶ **Splatting**
- ▶ Géométrie locale
 - ▶ **Voisinage** d'un point
 - ▶ Approximation locale de la **surface sous-jacente** : **ACP** et **MLS**
- ▶ **Descripteurs** de forme
 - ▶ Analyse statistique
- ▶ Bibliothèques logicielles :
 - ▶ www.pointclouds.org
 - ▶ www.open3d.org



1. Constituer votre **équipe**
2. Choisir un **responsable de produit**
3. Préparer (en équipe) la première version du **carnet de produit**

Merci

