

# Proposition projets 5ETI 2011/2012 - CPE

## Table des matières

1	Coordonnées différentielles	2
2	Interpolations de transformations	4
3	Synthèse de texture	6
4	Reconstruction de surfaces : Moving Least Squares	9
5	Déformation par coordonnées barycentriques complexes	12
6	Textures 3D procédurales	14
7	Visualisation de données scientifiques	16
8	Lancé de rayons avancé : Algorithme du Path Tracing	18
9	Surfaces de convolutions	20

# 1 Coordonnées différentielles

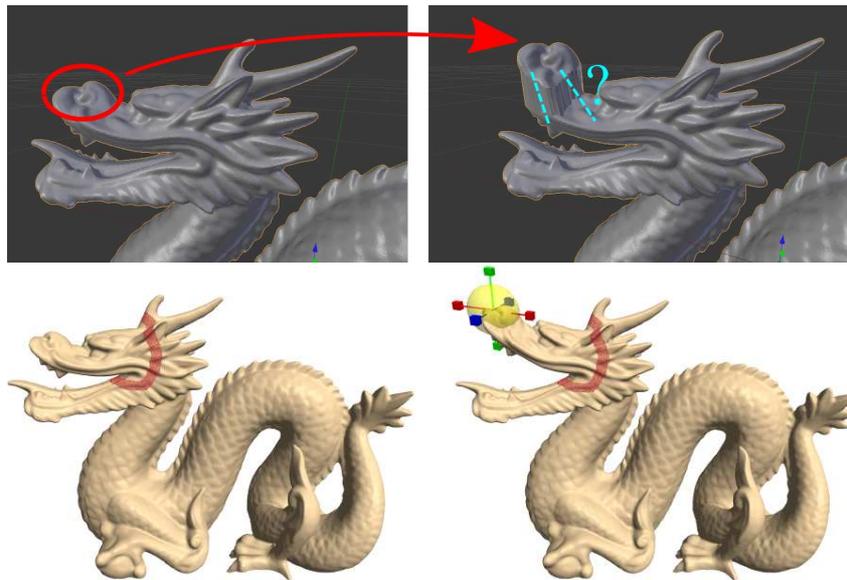


FIGURE 1 – Haut : La déformation rigide d'une partie du modèle n'entraîne pas de déformation locale du reste du maillage et n'offre pas une apparence réaliste finale. Bas (image tirée de [Sor+04]) : L'utilisation de l'encodage par coordonnées Laplaciennes permet de déformer localement la surface en préservant ses caractéristiques.

## 1.1 Problématique

Soit un maillage formé de  $N$  sommets donnés par le vecteur de coordonnées  $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{R}^{3N}$ . Nous souhaitons déformer ce maillage de manière interactive.

Cependant, déplacer un sommet unique  $\mathbf{x}_i$  manuellement ne permet pas d'obtenir une déformation correcte comme montrée en fig. 1 Haut. Supposons que le sommet  $i$  soit déplacé manuellement, nous souhaiterions en fait déplacer également les sommets voisins dans une certaine proportion afin de garder une forme globalement similaire, mais satisfaisant à la contrainte de position sur le sommet  $i$ . Mais dans quelle direction et de quelle amplitude ?

## 1.2 Principe

L'idée consiste à encoder non plus les sommets dans le système de coordonnées global cartésien  $(x, y, z)$ , mais à encoder, localement pour chaque sommet, des coordonnées dites *différentielles* [Lip+04 ; Sor06]. L'une des idées classiques consiste à définir ces coordonnées locales par la valeur du Laplacien associé aux coordonnées de la surface sous-jacente [Sor+04 ; Sor05]. Chaque sommet  $i$  se voit alors attribué une coordonnée différentielle  $\delta_i$  donnée par l'estimation de la valeur du Laplacien [WMKG07 ; AW11] en ce point de la surface. Ce Laplacien est obtenue par l'application d'un opérateur linéaire  $M$  sur les coordonnées cartésienne :  $\delta = Mx$ . Notons que deux surfaces  $S_1$  et  $S_2$  sont supposées être visuellement proche si leur valeur du Laplacien  $\Delta S_1$  et  $\Delta S_2$  l'est également. L'algorithme de déformation est alors le suivant :

1. Calculer  $\delta$  le Laplacien sur la surface non déformée  $S$ .
2. Déformer la surface  $S$  suivant les contraintes strictes de positions définies par l'utilisateur sur certains sommets.

3. Calculer la nouvelle surface  $S'$  telle que ses nouveaux sommets  $\mathbf{x}'$  satisfassent au sens des moindres carrées (voir fig. 1 Bas)

$$\delta' = M\mathbf{x}' = \delta_0 .$$

### 1.3 Travail demandé

- Comprendre et expliquer le principe de l'encodage suivant les coordonnées différentielles.
- Réalisez une étude bibliographique de la déformation Laplacienne et des autres encodage différentiels. Comparez leurs avantages et inconvénients.
- Réalisez une étude bibliographique des méthodes d'estimation du Laplacien sur un maillage. Comparez leurs avantages et inconvénients.
- En vous inspirant de [Sor+04], implémentez une déformation 2D ou 3D par déformation Laplacienne sur un maillage.
- + Implémentation et comparaison des différentes méthodes de calculs.
- + Réalisation d'un outil d'édition interactif.
- + Prise en compte des rotations.

### 1.4 Questions spécifiques

- Le Laplacien classique  $\Delta$  se calcule sur une fonction dans un espace Cartésien par la somme des dérivées partielles suivant  $(x, y, z)$  en 3D ou  $(x, y)$  en 2D. Nous sommes ici sur une surface, une variété, que cela change-il ? Quelle est la fonction considérée, dans quel espace varie-t-elle ? Comment se nomme cet opérateur ?
- Il existe plusieurs estimations du Laplacien discret sur un maillage, d'où proviennent les formules, expliquez.
- La matrice du Laplacien  $M$  possède des particularités propres, quels en sont les avantages.
- Ce type de déformation possède des limitations, mettez les avant. Que faudrait-il faire pour améliorer cette méthode.

### Références

- [AW11] Marc ALEXA et Max WARDETZKY. « Discrete Laplacians on General Polygonal Meshes ». Dans : *ACM SIGGRAPH, ACM Transaction on Graphics (TOG)* 30.4 (2011).
- [Lip+04] Yaron LIPMAN, Olga SORKINE, Daniel COHEN-OR, David LEVIN, Chrisitan ROSSL et Hans-Peter SEIDEL. « Differential Coordinates for Interactive Mesh Editing ». Dans : *Shape Modeling International (SMI)* (2004).
- [Sor05] Olga SORKINE. « Laplacien Mesh Processing ». Dans : *Eurographics, State of the Art Report* (2005).
- [Sor06] Olga SORKINE. « Differential Representations for Mesh Processing ». Dans : *Computer Graphics Forum (CGF)* 25.4 (2006), p. 789–807.
- [Sor+04] Olga SORKINE, Daniel COHEN-OR, Yaron LIPMAN, Marc ALEXA, Christian ROESSL et Hans-Peter SEIDEL. « Laplacian Surface Editing ». Dans : *Symposium on Geometry Processing (SGP)* (2004).
- [WMKG07] Max WARDETZKY, Saurabh MATHUR, Felix KALBERER et Eitan GRINSPUN. « Discrete Laplace Operators: No free lunch ». Dans : *Symposium on Geometry Processing (SGP)* (2007).

## 2 Interpolations de transformations

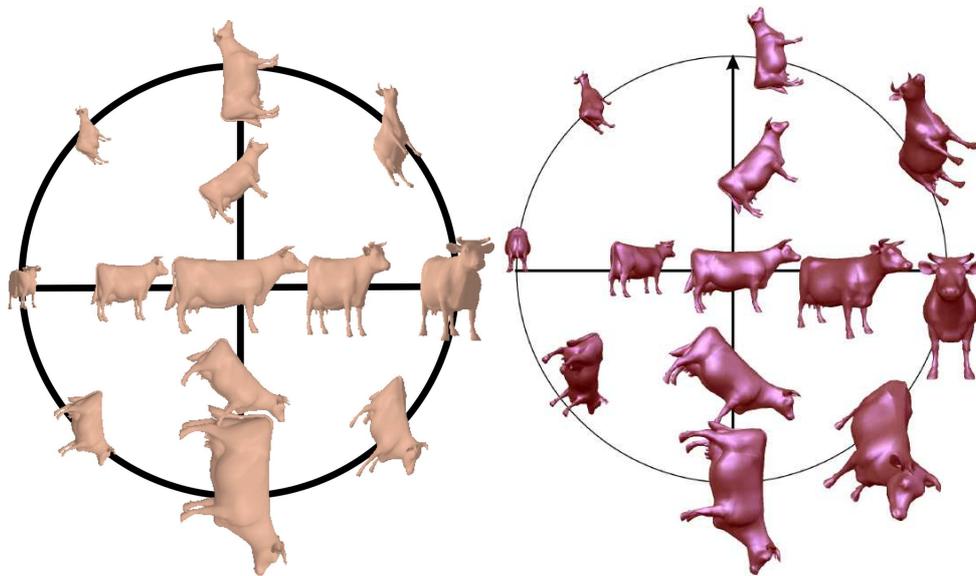


FIGURE 2 – Gauche : L’interpolation linéaire des coefficients des matrices de transformations pour les formes intermédiaires génère des déformations de cisaillements non souhaitées. Droite (tirée de [Ale02]) : L’utilisation de méthodes d’interpolations matricielles spécialisées permet d’interpoler correctement les transformations.

### 2.1 Problématique

Les transformations linéaires de l’espace peuvent être représentées par une matrice carrée  $4 \times 4$ . L’application d’une transformation  $M$  à un sommet de l’espace  $x$  peut donc se réaliser par le produit matriciel  $x' = Mx$ .

La composition de ces transformations est cependant moins aisée. Considérer la moitié d’une transformation ou interpoler linéairement deux transformations  $M_1$  et  $M_2$  n’est pas directe. En effet, la matrice  $M$  peut contenir des translations, rotations et homothéties. Or l’espace des rotations ne forme pas un espace vectoriel. Ainsi, interpoler linéairement les coefficients de deux matrices de rotations  $R_1$  et  $R_2$  n’aboutit généralement pas à une matrice de rotation

$$\alpha R_1 + (1 - \alpha) R_2 \neq SO(3).$$

Typiquement, un effet de cisaillement apparaît (voir fig. 2).

### 2.2 Principe

L’idée consiste à définir un espace particulier dans lequel les opérations de multiplication par un scalaire et d’addition restent closes dans l’espace. Ces espaces particuliers sont dénommés *groupes de Lie*.

Considérons la variété définie par l’ensemble des transformations possibles. Dans cet espace de Lie, une transformation peut être vue comme un vecteur défini dans l’espace tangent de cette variété. L’interpolation entre deux vecteurs se réalise alors en comparant ces vecteurs dans leur espace tangent respectif. Cette interpolation généralisée se nomme *application exponentielle* [RM94 ; EM98 ; PFA06] (exponential map) et revient, dans notre cas, à l’évaluation d’exponentielles et logarithmes matriciels.

Il devient alors possible de définir des opérations particulières  $\odot$  tel que  $\alpha \odot M$  permette de calculer la part  $\alpha$  d'une déformation tout en préservant ces caractéristiques [Ale02].

### 2.3 Travail demandé

- Comprendre et expliquer la problématique d'interpolation.
- Comprendre et expliquer l'interpolation par l'application exponentielle.
- Réaliser une étude bibliographique des différents travaux de calculs d'interpolation dans des variétés
- En vous basant sur [Ale02], implémentez l'interpolation de déformations 3D par l'application exponentielle.
- + Implémenter d'autres interpolations spécifiques et comparez les résultats.
- + Implémenter l'interpolation sur variété dans le cadre de la mécanique.

### 2.4 Questions spécifiques

- Une méthode triviale d'interpolation sur variété consiste à interpoler linéairement des composantes et à projeter sur la contrainte. Quelles sont les limitations de cette approche ? En quoi l'application exponentielle résout-elle cela ?
- Quel rapport il y a t'il entre l'espace tangent, l'application exponentielle et l'interpolation ?
- Il existe d'autres cas de groupes, voir d'algèbres, de Lie rependues. Donnez des exemples, quels sont les méthodes utilisées dans ce cas ?
- Quels algorithmes sont utilisés pour calculer l'exponentielle de matrices ?

### Références

- [Ale02] Marc ALEXA. « Linear Combination of Transformations ». Dans : *ACM SIG-GRAPH, ACM Transaction on Graphics (TOG)* 21.3 (2002).
- [EM98] Practical Parameterization of Rotations Using the EXPONENTIAL MAP. « Sebastian Grassia ». Dans : *The Journal of Graphics Tools* (1998).
- [PFA06] Xavier PENNEC, Pierre FILLARD et Nicholas AYACHE. « A Riemannian Framework for Tensor Computing ». Dans : *International Journal of Computer Vision* 66.1 (2006).
- [RM94] Shankar Sastry RICHARD MURRAY Zexiang Li. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

### 3 Synthèse de texture

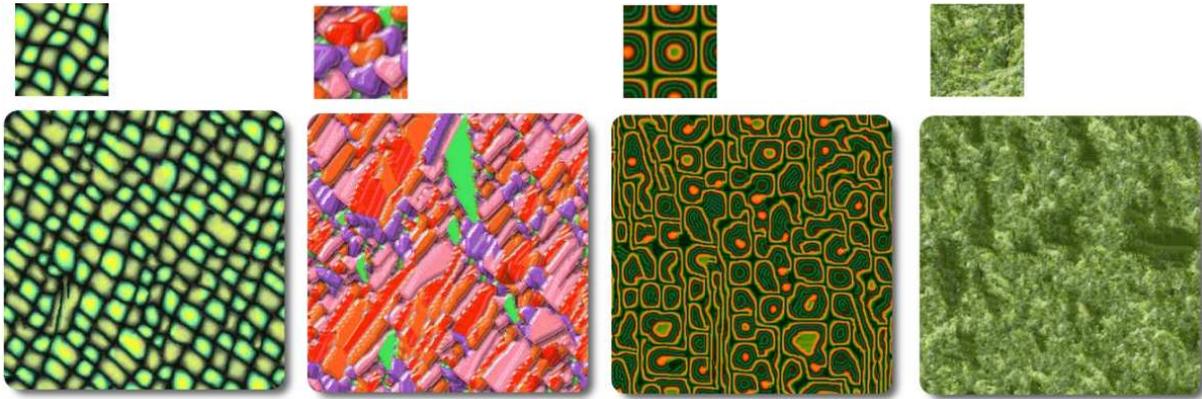


FIGURE 3 – Texture originale en haut, et synthèse d’un patch plus grand préservant en dessous. Images obtenues avec la méthode présentée en [EL99].

#### 3.1 Problématique

Soit une l’image d’un échantillon de texture. Il est généralement commode de vouloir étendre celle-ci afin de couvrir un domaine plus grand. Dupliquer des morceaux de texture permet de préserver les caractéristiques de celle-ci, par contre l’oeuil humain percevra directement un effet de répétition. D’un autre côté, considérer des accolements de morceaux aléatoires permet d’éviter l’impression de répétition, mais ne préserve pas les caractéristiques de la texture.

#### 3.2 Principe

Soit une image source contenant l’échantillon de texture  $I_0$ . Soit l’image de destination  $I_1$  initialement initialisé par des pixels de couleur aléatoire.

L’idée consiste à parcourir  $I_1$  de proche en proche (par ex. de gauche à droite et de haut en bas) et d’assigner au pixel de coordonnées  $(x, y)$  la couleur du pixel de l’image  $I_0$  dont le voisinage ressemble le plus à celui de  $I_1(x, y)$  [EL99 ; WL00 ; Wei01].

En formalisant : Soit  $v_{(x_0, y_0)}$  le voisinage centré en  $(x_0, y_0)$  (par exemple un masque 5x5 ou 9x9). Appelons  $v_{(x_0, y_0)}^k$  la  $k$ ème coordonnée du masque  $v$  contenant au total  $N$  entrées. Pour assigner la valeur du pixel  $I_1(x_1, y_1)$ , on cherche alors les coordonnées  $(x_0, y_0)$  dans l’image  $I_0$  telles que

$$\sum_{k=0}^{k=N-1} \|I_1(v_{(x_1, y_1)}^k) - I_0(v_{(x_0, y_0)}^k)\| \text{ soit minimal.}$$

L'algorithme est alors le suivant :

```
I0<- image source
I1<- image destination
Pour tout (x1,y1)

  Chercher (x0,y0) tel que
    ||I1(voisins(x1,x1))-I0(voisins(x0,y0))|| soit min

  I1(x1,y1)=I0(x0,y0)
  (x0,y0)=pixel suivant

Fin pour
```

### 3.3 Travail demandé

- Réaliser une étude bibliographique des travaux de synthèse de textures et leurs applications. On prendra soin de comparer les approches ([WL00 ; EF01 ; Kwa+03 ; MWT11], etc).
- Comprendre et expliquer le principe décrit dans [EL99]. Notamment expliquer ses fondements théoriques.
- Implémenter la synthèse de texture suivant l'approche de [EL99] sur divers échantillons. Le mettre en oeuvre sur de la suppression de partie d'une image (voir fig. 4).
- Expliquez les limitations et améliorations possibles : Contrôle de la synthèse, prise en compte des bords francs, calcul efficace.
- + Implémentation efficace de l'algorithme.
- + Implémentation d'une méthode prenant en compte les bords francs [Ash01].
- + Contrôle de la synthèse, prise en compte de sources multiples.
- + Textures 3D / animée.

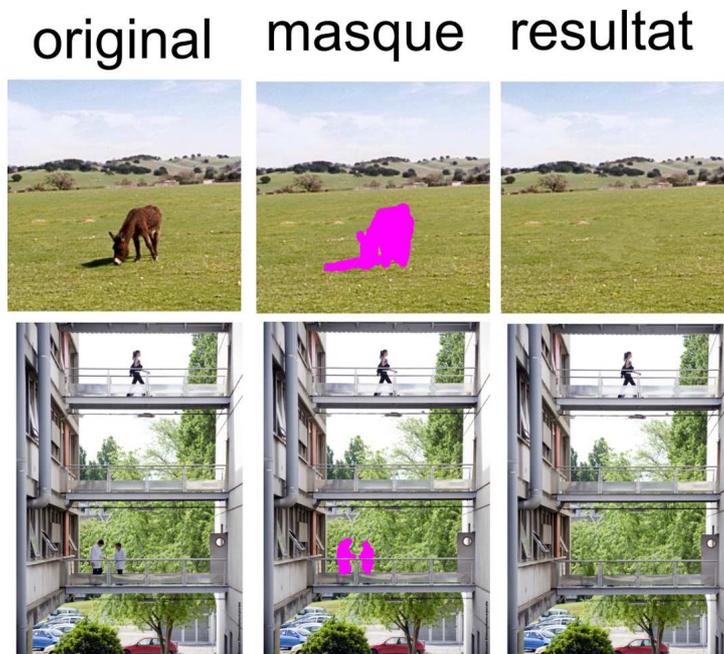


FIGURE 4 – Exemple de suppression de parties d'une image. De gauche à droite : Image originale ; masque de suppression ; résultat après synthèse de texture.

### 3.4 Questions spécifiques

- Il est noté que le masque du voisinage doit être causale, pourquoi ? Que donne le résultat lorsqu'il ne l'est pas.
- Quel est le lien entre les chaînes de Markov et la méthode de synthèse de texture ?

#### Références

- [Ash01] Michael ASHIKHMIN. « Synthesizing Natural Textures ». Dans : *Symposium on Interactive 3D Graphics (I3D)* (2001).
- [EF01] Alexei EFROS et William FREEMAN. « Image quilting for texture synthesis and transfer ». Dans : *ACM SIGGRAPH* (2001).
- [EL99] Alexei EFROS et Thomas LEUNG. « Texture Synthesis by Non-Parametric Sampling ». Dans : *IEEE International Conference on Computer Vision (ICCV)* 2 (1999).
- [Kwa+03] Vivek KWATRA, Arno SCHODL, Irfan ESSA, Greg TURK et Aaron BOBICK. « Graphcut textures: image and video synthesis using graph cuts ». Dans : *ACM SIGGRAPH, ACM Transaction on Graphics (TOG)* 22.3 (2003).
- [MWT11] Chongyan MA, Li-Yi WEI et Xin TONG. « Discrete Element Textures ». Dans : *ACM SIGGRAPH, ACM Transaction on Graphics (TOG)* 30.4 (2011).
- [Wei01] Li-Yi WEI. « Texture Synthesis by Fixed Neighborhood Searching ». Thèse de doct. Stanford, 2001.
- [WL00] Li-Yi WEI et Marc LEVOY. « Fast Texture Synthesis using Tree-structured Vector Quantization ». Dans : *ACM SIGGRAPH* (2000).

## 4 Reconstruction de surfaces : Moving Least Squares

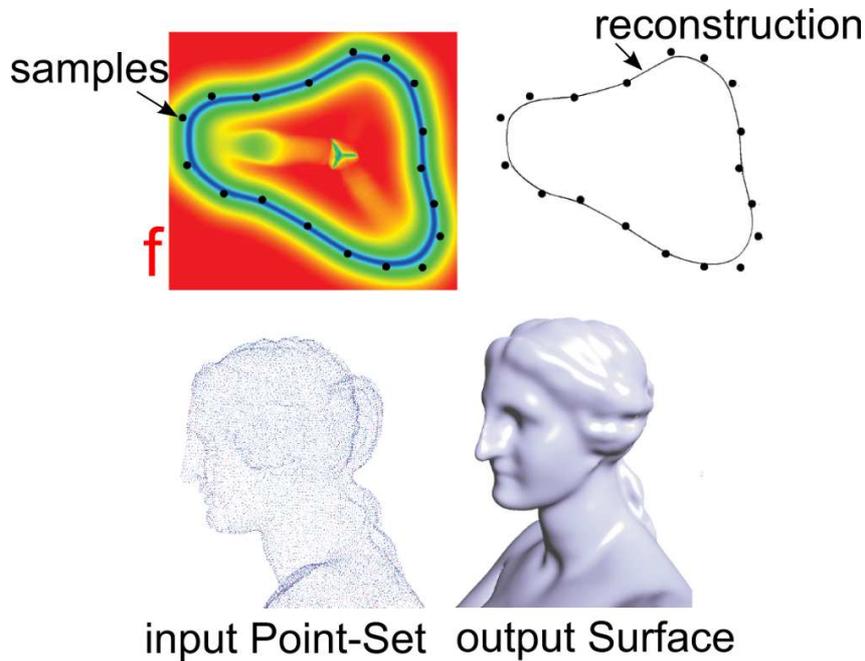


FIGURE 5 – Exemples de résultats de reconstruction de courbes en 2D (haut) et surfaces 3D (bas) à partir de points+normales. Images tirées de [Ale+03 ; AA09].

### 4.1 Problématique

Les scanners lasers ne permettent pas la capture d'un objet réelle sous forme de surface directement. En effet, chaque échantillon mesuré fournit une information de position spatiale ainsi que la normale en cette position. Elle ne permet cependant pas de relier les points afin de modéliser une surface polygonale par exemple. De plus, ce type de mesure est potentiellement sujet au bruit. Il est donc nécessaire de définir des algorithmes permettant de reconstruire une surface polygonale de manière robuste à partir de l'information de points et de normales.

### 4.2 Principe

Une technique de reconstruction de surface à partir de tels échantillons consiste à estimer une fonction volumique  $f$  telle que l'isovaleur 0 se trouve sur la surface finale que l'on cherche [Ale+03]. La surface polygonale est alors obtenue par extraction d'une isosurface suivant des algorithmes standards.

Dénommons par  $S$  la surface lisse que l'on cherche. Soit  $\mathbf{c}$  un point de  $S$  et  $\mathbf{p}$  une position de l'espace quelconque. Si  $\mathbf{p}$  appartient au plan tangent de  $S$  au voisinage de  $\mathbf{c}$ , alors

$$\langle \mathbf{n}, \mathbf{p} - \mathbf{c} \rangle = 0,$$

avec  $\mathbf{n}$  la normale à  $S$  en  $\mathbf{c}$ .

Nous définissons ainsi  $f$  par le produit scalaire

$$\forall \mathbf{p} \in \mathbb{R}^3, f(\mathbf{p}) = \langle \mathbf{n}(\mathbf{p}), \mathbf{p} - \mathbf{c}(\mathbf{p}) \rangle,$$

avec  $\mathbf{c}(\mathbf{p})$  obtenu comme la projection de  $\mathbf{p}$  sur une estimation de  $S$ . De même,  $\mathbf{n}(\mathbf{p})$  est une estimation du vecteur normale de  $S$  en  $\mathbf{c}(\mathbf{p})$ . On a alors  $\mathbf{p} \in S \Rightarrow f(\mathbf{p}) = 0$ .

Ces estimateurs  $\mathbf{c}$  et  $\mathbf{n}$  sont calculés comme des valeurs pondérées des positions et normales locales. En suivant une approche inspirée des *Moving Least Squares* [AA04; AK04], on peut alors définir

$$\mathbf{c}(\mathbf{p}) = \frac{\sum_k h(\|\mathbf{p} - \mathbf{p}_k\|) \mathbf{p}_k}{\sum_k h(\|\mathbf{p} - \mathbf{p}_k\|)},$$

$$\mathbf{n}(\mathbf{p}) = \frac{\sum_k h(\|\mathbf{p} - \mathbf{p}_k\|) \mathbf{n}_k}{\sum_k h(\|\mathbf{p} - \mathbf{p}_k\|)},$$

où les  $\mathbf{p}_k$  et  $\mathbf{n}_k$  sont les échantillons des positions et normales que l'on cherche à interpoler. Et  $h$  est une fonction lisse décroissante de  $\mathbb{R}^+$ .

### 4.3 Travail demandé

- Réaliser une étude bibliographique de la reconstruction de surface à partir de points + normales. Placer les *Moving Least Squares* dans ce contexte.
- Comprendre et expliquer le rapport entre la notion de *Moving Least Squares* et cette approche.
- En se basant sur [Ale+03], implémenter une méthode de reconstruction de surface en 2D dans un premier temps. Puis passer à une généralisation en 3D.
- + Mise en place d'une reconstruction 3D complète.
- + Comparaison avec des améliorations telles que l'approche algébrique, etc ([GG07; OGG09; AA09], etc).

### 4.4 Questions spécifiques

- Pourquoi peut on considérer que  $f(\mathbf{p}) = 0 \Rightarrow \mathbf{p} \in S$  ?
- Comment passer de la valeur échantillonnée de  $f$  à une surface polygonale définie par  $f = 0$  ? Quels sont les algorithmes s'appliquant ?
- Quels importances ont les fonctions noyaux  $h$  ? Comparez leurs rôles sur la surface obtenue.
- Quels sont les autres approches de reconstruction 3D, quelles sont leurs spécificités (ex. [KBH06]).

### Références

- [AA04] Marc ALEXA et Anders ADAMSON. « On normals and Projection Operators for Surfaces Defined by Point Sets ». Dans : *Symposium on Point-Based Graphics* (2004).
- [AA09] Marc ALEXA et Anders ADAMSON. « Interpolatory Point Set Surfaces - Convexity and Hermite Data ». Dans : *ACM Transaction on Graphics (TOG)* 28.2 (2009).
- [Ale+03] Marc ALEXA, Johannes BEHR, Daniel COHEN-OR, Shachar FLEISHMAN, David LEVIN et Claudio SILVA. « Computing and Rendering Point Set Surfaces ». Dans : *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 9.1 (2003).
- [AK04] Nina AMENTA et Yong KIL. « Defining Point-Set Surfaces ». Dans : *ACM SIGGRAPH, ACM Transaction on Graphics (TOG)* 23.3 (2004).
- [GG07] Gael GUENNEBAUD et Markus GROSS. « Algebraic Point Set Surfaces ». Dans : *ACM SIGGRAPH, ACM Transaction on Graphics (TOG)* 26.3 (2007).
- [KBH06] Michael KAZHDAN, Matthew BOLITHO et Hughes HOPPE. « Poisson Surface Reconstruction ». Dans : *Symposium on Geometry Processing (SGP)* (2006).

[OGG09] Cengiz OZTIRELI, Gael GUENNEBAUD et Markus GROSS. « Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression ». Dans : *Eurographics, Computer Graphics Forum (CGF)* 28.2 (2009).

## 5 Déformation par coordonnées barycentriques complexes

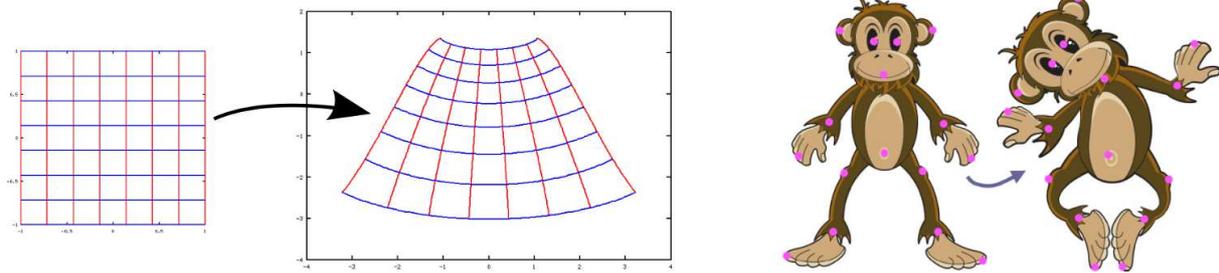


FIGURE 6 – Gauche : Déformation du carré unitaire, notez que la déformation préserve les angles en tous points du domaine. Droite (Image tirée de [WBCG09]) : Application à la déformation d'un personnage 2D.

### 5.1 Problématique

Une déformation d'une image ou d'un objet vectoriel du plan nous apparait naturelle lorsque cette déformation préserve certaines propriétés telles que les angles. De plus, il est commode de *plonger* l'objet à déformer dans un polygone de contrôle prenant en compte sa morphologie afin de permettre sa manipulation aisé. La déformation de ce polygone étant alors répercuté le domaine ainsi défini par coordonnées barycentriques généralisées.

Trouver une telle déformation se confronte à deux problèmes :

1. Les coordonnées barycentriques linéaires d'interpolation standard sont définies pour des triangles uniquement, et ne s'appliquent pas à des polygones généraux, potentiellement non convexes.
2. Les fonctions de déformations ainsi définies ne préservent pas les angles et introduisent un effet de cisaillement.

### 5.2 Principe

L'idée consiste à généraliser des coordonnées barycentriques à des polygones quelconques du plan et de forcer la préservation d'angle en exprimant ceux-ci à l'aide de fonctions holomorphes [WBCG09 ; Web10 ; WBCGH11]. Ces fonctions sont aisément exprimables à l'aide de la variable complexe. Il devient alors possible d'exprimer un schéma d'interpolation d'une coordonnée  $z$  linéairement en fonction des sommets  $b_k$  du polygone de contrôle déformé

$$z' = \sum_k f_k(z) b_k ,$$

où  $z = x + iy$  est la coordonnée complexe initiale d'une position du polygone originale, et  $z'$  est la coordonnée complexe interpolée après déformation.

### 5.3 Travail demandé

- Réalisez une étude bibliographique des déformations du plan.
- Comprendre et expliquer la méthode de déformation proposée dans ce contexte.
- En se basant sur [WBCG09], implémentez la déformation d'une image par coordonnées de Cauchy-Green et par coordonnées de Szegö. On considèrera spécifiquement des polygones de contrôles non convexes (voir fig. 6).

- Discutez des avantages et inconvénient de cette approche par rapport à d'autres méthodes.
- + Réaliser un outil de déformation interactive de déformation point à point.
- + Discutez et présentez des extensions à la 3D.

#### 5.4 Questions spécifiques

- Expliquez le principe générale de la déformation par polygone de contrôle.
- Pourquoi est-il important de pouvoir traiter des polygones de contrôles non convexes ?
- Pourquoi ne pas utiliser une interpolation barycentrique classique sur un polygone triangulé ?
- Pourquoi l'utilisation de fonctions holomorphes implique la conformité ?
- Expliquez les formules proposées.

#### Références

- [Web10] Ofir WEBER. « Hybrid Methods for Interactive Shape Manipulation ». Thèse de doct. Technion, 2010.
- [WBCG09] Ofir WEBER, Mirela BEN-CHEN et Craig GOTSMAN. « Complex Barycentric Coordinates with Applications to Planar Shape Deformation ». Dans : *Eurographics, Computer Graphics Forum (CGF)* 28.2 (2009).
- [WBCGH11] Ofir WEBER, Mirela BEN-CHEN, Craig GOTSMAN et Kai HORMANN. « A complex view of barycentric mappings ». Dans : *Symposium on Geometry Processing (SGP)* (2011).

## 6 Textures 3D procédurales

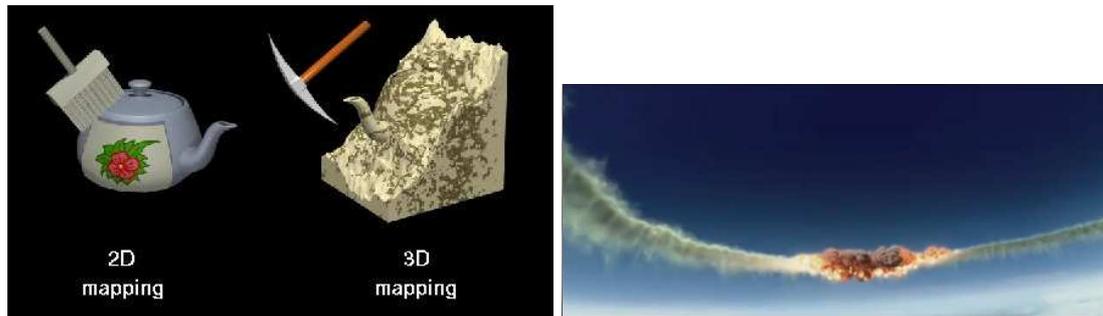


FIGURE 7 – Gauche : Différence entre texture 2D et texture 3D (image de Pascal Vuylstecker). Droite : Un exemple d'utilisation de texture procédurale dans un film d'animation (©Stealth, Digital Domain).

### 6.1 Problématique

De nombreux objets possèdent des structures volumiques caractéristiques et continues spatialement. On pourra citer par exemple des matières comme le bois, le marbre, ou des objets flous volumiques tels que les nuages (voir fig. 7). La modélisation de ces objets nécessite de plaquer des textures particulières. Cependant, le placage de texture surfacique standard sur de tels objets ne donne pas de résultats visuels convaincants de par la déformation liée à la paramétrisation. De plus, la structure volumique nécessaire lors de la coupe de tels objets n'est pas direction accessible.

L'utilisation de textures 3D résout ce problème en plongeant alors la surface des objets dans cette image volumique. Lorsque l'objet est creusé, la texture reste cohérente. Cependant, l'obtention de texture volumique est, contrairement au cas des images, difficile à obtenir et excessivement lourde à stocker en mémoire.

### 6.2 Principe

L'idée consiste à générer des textures de manière procédurale, c'est à dire à l'aide d'un sous ensemble de paramètres permettant de générer différentes variations. Ces textures pouvant potentiellement être générées à la volée, elles ne nécessitent pas d'être stockée entièrement en mémoire.

La méthode la plus répandue de génération de texture 3D aléatoire, mais continue spatialement, est celle du *bruit de Perlin* [Per85 ; PH89]. Les déclinaisons de ce *bruit* sont très nombreuses et il s'agit d'un élément primordial de nombreux effets spéciaux.

### 6.3 Travail demandé

- Réalisez une étude bibliographique de l'utilisation du *bruit de Perlin* en s'attachant à montrer l'étendue de leur utilisation.
- Réalisez une étude bibliographique sur la génération de texture 3D, quelles autres approches existent, comparez leurs avantages et inconvénients (ex. [Lew89 ; Tur91 ; Ney03 ; Pea85 ; Miy90 ; CN02], etc).
- Implémentez une génération de texture 3D, visualisez celle-ci par coupe ou placage sur un objet 3D.
- Mettez en place des paramètres de contrôle sur votre texture.

- + Implémentation efficace de votre outil de génération de texture.
- + Réalisez des déformations volumiques de votre objet (creusement, etc).
- + Visualisez des textures volumiques transparentes (ex. nuages, etc).
- + Implémentez la génération d'autre type de bruits.
- + Implémentez des textures animées.

## 6.4 Questions spécifiques

- Pourquoi le placage de texture 2D implique des distorsions ?
- Quels problèmes apparaissent lors de l'utilisation de ce bruit en 2D et en 3D par rapport à la 1D ? Quelles solutions sont proposées ? Expliquez.

## Références

- [CN02] Jean COMBAZ et Fabrice NEYRET. « Painting Folds Using Expansion Textures ». Dans : *Pacific Graphics (PG)* (2002).
- [Lew89] J. P. LEWIS. « Algorithms for solid noise synthesis ». Dans : *ACM SIGGRAPH* (1989).
- [Miy90] Kazunori MIYATA. « A method of generating stone wall patterns ». Dans : *ACM SIGGRAPH* (1990).
- [Ney03] Fabrice NEYRET. « Advected textures ». Dans : *Symposium on Computer Animation (SCA)* (2003).
- [Pea85] Darwyn PEACHEY. « Solid texturing of complex surfaces ». Dans : *ACM SIGGRAPH* (1985).
- [Per85] Ken PERLIN. « An Image Synthesizer ». Dans : *ACM SIGGRAPH* (1985).
- [PH89] Ken PERLIN et Eric HOFFERT. « Hypertexture ». Dans : *ACM SIGGRAPH* (1989).
- [Tur91] Greg TURK. « Generating textures on arbitrary surfaces using reaction-diffusion ». Dans : *ACM SIGGRAPH* (1991).

## 7 Visualisation de données scientifiques

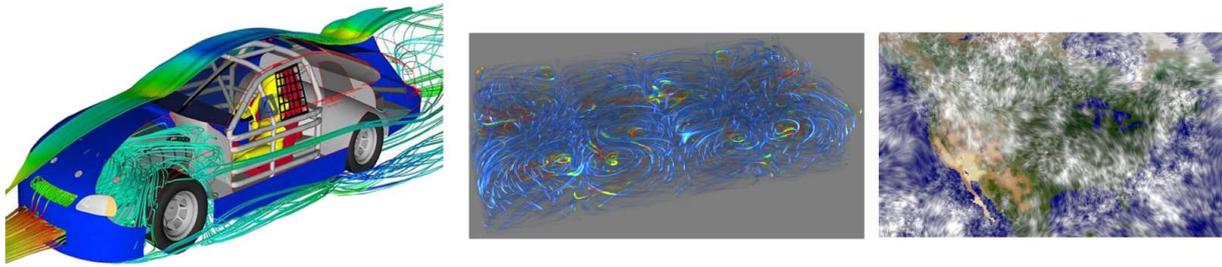


FIGURE 8 – Différents exemples de visualisations scientifiques (on pourra y remarquer la présence de : lignes de flux, isosurface, encodage couleur, LIC, rendu volumique). Images tirées respectivement de : Computational Engineering International, [FW08][CL93].

### 7.1 Problématique

Les données issues de simulation numérique fournissent souvent des informations de surfaces ou de volume difficilement visualisable. Par ex. la vitesse d'écoulement d'air/de fluide dans l'espace, des contraintes tensorielles sur une surface ou un volume, des densités volumiques (voir fig. 8).

De manière générale, on peut considérer que ces données sont représentées par une fonction

$$f : \mathbf{x} \in \mathbb{R}^i \mapsto f(\mathbf{x}) \in \mathbb{R}^o ,$$

où  $i$  vaut 2 ou 3 lorsque l'on se trouve respectivement sur une surface ou un volume, et  $o$  valant 1 pour des données scalaires, ou plus pour des données vectorielles ou tensorielles.

La problématique est alors d'arriver à visualiser spatialement les informations utiles afin de comprendre le phénomène porté par la simulation.

### 7.2 Principe

Différentes approches permettent de visualiser des informations en fonction des caractéristiques recherchées.

- Les algorithmes d'encodage couleur le long de coupes surfaciques sont aisément mise en place pour la visualisation de donnée scalaire volumique.
- Les algorithmes d'extraction d'isosurfaces permettent de visualiser la répartition spatiale d'une valeur particulière d'un champ scalaire [LC87 ; MW97].
- Le rendu volumique par transparence permet d'obtenir rapidement l'apparence globale d'un champ scalaire [Lev88].
- L'affichage de donnée vectorielle sous forme de vecteur est généralement trop complexe pour en tirer une interprétation, on peut alors se ramener à des champs scalaires en interprétant spécifiquement leur orientation ou leur norme.
- La visualisation de lignes de champs, ou de caractéristiques vectorielles telles que la divergence ou le rotationnel peut également se révéler porteur d'informations importantes [Hul92 ; HE06 ; PL09].
- Le filtrage de textures le long de champs vectoriels surfaciques permet également de visualiser certains champs de vecteurs [CL93 ; HA04].

### 7.3 Travail demandé

- Réaliser une étude bibliographique sur les différents types de données que l'on cherche à visualiser.
- Réaliser une étude bibliographique de différentes méthodes de visualisation de données scientifiques. On portera une attention particulière à expliciter l'intérêt et l'utilisation standard de chaque méthode présentée (et non une simple énumération).
- Implémentez certains algorithmes de visualisation sur des données surfaciques et volumiques.
- Montrez la complémentarité de différentes visualisations sur un même jeu de donnée.

### 7.4 Questions spécifiques

- Les données à visualiser sont-elles simulées ou scannées ? Quelles différences cela implique-t-il ?
- Qu'est ce qu'une donnée tensorielle ? Donnez des exemples. Comment visualise-t-on ce type de donnée ?
- Quelles contraintes supplémentaires implique la visualisation de données définies sur une surface ?
- Peut on visualiser des données de dimension supérieure ? Par ex. des données dont le domaine de définition est compris dans  $\mathbb{R}^4$ . Donnez des exemples.
- Exmpliquez l'approche de [VTK].

### Références

- [CL93] Brian CABRAL et Leith LEEDOM. « Imaging Vector Fields Using Line Integral Convolution ». Dans : *ACM SIGGRAPH* (1993).
- [FW08] Martin FALK et Daniel WEISKOPF. « Output-Sensitive 3D Line Integral Convolution ». Dans : *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 14.4 (2008).
- [HA04] Anders HELGELAND et Oyvind ANDREASSEN. « Visualization of Vector Fields Using Seed LIC and Volume Rendering ». Dans : *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 10.6 (2004).
- [HE06] Anders HELGELAND et Thomas ELBOTH. « High-Quality and Interactive Animations of 3D Time-Varying Vector Fields ». Dans : *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 12.6 (2006), p. 1535–1546.
- [Hul92] J. P. HULTQUIST. « Constructing stream surfaces in steady 3D vector fields ». Dans : *Conference on Visualization* (1992).
- [Lev88] Marc LEVOY. « Display of Surfaces from Volume Data ». Dans : *IEEE Computer Graphics and Applications (CG&A)* 8.3 (1988).
- [LC87] William LORENSEN et Harvey CLINE. « Marching cubes: A high resolution 3D surface construction algorithm ». Dans : *ACM SIGGRAPH* (1987).
- [MW97] Heinrich MULLER et Michael WEHLE. « Visualization of Implicit Surfaces Using Adaptive Tetrahedrizations ». Dans : *Conference on Scientific Visualization* (1997).
- [PL09] Zhenmin PENG et Robert LARAMEE. « Higher Dimensional Vector Field Visualization: A Survey ». Dans : *Theory and Practice of Computer Graphics* (2009).
- [VTK] Visualization Toolkit VTK. <http://www.vtk.org/>.

## 8 Lancé de rayons avancé : Algorithme du Path Tracing

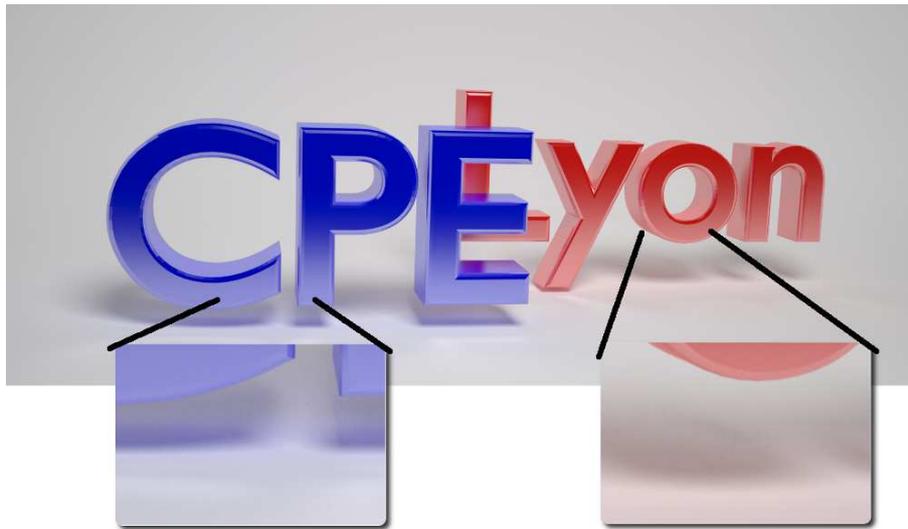


FIGURE 9 – Exemple d’un rendu modélisant les objets comme source de lumière secondaire. Les colorations bleues et rouges des objets se diffusent sur le voisinage blanc.

### 8.1 Problématique

L’algorithme de rendu par lancé de rayon [Whi80] permet de simuler des effets complexes tels que la transparence, la réfraction, les réflexions sur des objets quelconques. Pour rappel, l’implémentation standard du lancé de rayons consiste à :

- Lancer un rayon pour chaque pixel de l’image dans la direction du point de vue de la caméra.
- Calculer la première intersection (pour des objets opaques) avec un objet de la scène.
- Lancer un rayon partant de cette intersection en direction de/des source(s) de lumière de la scène afin de calculer l’illumination associée à cette intersection.

Ce type d’algorithme ne modélise cependant pas le comportement physique de la lumière. En effet, le comportement diffusif de la matière implique que chaque objet illuminé devient lui-même source de lumière secondaire de la scène. L’ensemble des interactions, une fois liée peut se modéliser par une équation intégrale infiniment récursive. Cette équation complexe ne possède pas de solution analytique connue.

À titre d’exemple, un objet rouge posé sur une table blanche viendra ainsi diffuser un éclairage rouge autour de sa position sur la table (voir 9). Suivant la matière considérée, cette diffusion peut, de plus, dépendre de l’orientation relative de la source de lumière et de la table par rapport à l’objet. Ce comportement d’illumination indirecte n’existe pas dans l’algorithme initial du lancé de rayons ce qui conduit souvent à une apparence *trop parfaite* des objets représentés.

### 8.2 Principe

Afin d’approximer cette équation intégrale de l’illumination, il est possible de passer par des méthodes numériques d’intégration en hautes dimensions dites de *Monte Carlo*.

En pratique, cela revient à lancer d’avantages de rayons à chaque intersection afin d’échantillonner plus finement le comportement d’un sous-ensemble de rayons lumineux.

Lancer plusieurs rayons de manière récursive aboutit rapidement à des complexités algorithmiques exponentielles. La méthode dite de *path tracing* [Kaj86] propose de lancer un seul rayon supplémentaire à chaque intersection, mais d'échantillonner plusieurs fois un même pixel. L'algorithme est dit *non biaisé* car il converge asymptotiquement vers la vraie solution de l'équation intégrale de l'illumination [Laf96 ; PH10].

### 8.3 Travail demandé

- Comprendre et expliquer l'équation de la l'illumination que l'on cherche à résoudre.
- Placer et expliquer le lien avec l'algorithme du path tracing.
- Réaliser un travail bibliographique sur les méthodes de rendu approximant cette équation. Donnez les avantages et inconvénients propres à chacune. On prendra soin de séparer les approches biaisées et non biaisées.
- Implémentez un ray-traceur proposant un rendu par illumination directe, et par *path-tracing*.
- + Implémentation et comparaison avec d'autres approches non biaisées (MLT, etc)
- + Implémentation et comparaison avec d'autres approches biaisées (photon mapping, radiance, etc)
- + Prise en compte des BRDF.
- + Implémentation parallèle.
- + Implémentation d'un ray-traceur générique (primitives, etc).

### 8.4 Questions spécifiques

- D'où provient l'équation de l'illumination, qu'est ce qu'une BRDF ?
- Pourquoi utilise-t-on la méthode de Monte Carlo et non une intégration numérique de type Simpson ?
- Pourquoi la méthode du path-tracing converge ? à quelle vitesse ?

### Références

- [Kaj86] James KAJIYA. « The rendering equation ». Dans : *ACM SIGGRAPH* (1986).
- [Laf96] Eric LAFORTUNE. « Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering ». Thèse de doct. Leuven University, 1996.
- [PH10] Matt PHARR et Greg HUMPHREYS. *Physically Based Rendering: From Theory To Implementation*. Morgan Kaufmann, 2010.
- [Whi80] Turner WHITTED. « An improved illumination model for shaded display ». Dans : *Communications of the ACM (CACM)* 23.6 (1980).

## 9 Surfaces de convolutions



FIGURE 10 – Exemple de surface de convolutions. Image d’hippocampe tirée de [She99] : gauche squelette linéique, droite surface résultante. Image de personnages tirée de [BBCW10].

### 9.1 Problématique

Les surfaces implicites sont définies comme les isosurfaces d’une fonction scalaire  $f$ . C’est à dire par l’ensemble  $\{(x, y, z) \in \mathbb{R}^3 | f(x, y, z) = 0\}$  dans le cas où l’isovaleur est égale à 0.

Ces surfaces, de par leur définition implicite, permettent de gérer de manière transparente les changements de topologies (fusion d’éléments distincts, ou au contraire, séparation). Elles sont donc l’élément de choix pour les simulations d’éléments hautement déformables tels que les fluides.

Leur utilisation dans le contexte de la modélisation d’objets nécessite de travailler sur la fonction  $f$  qui n’est pas une chose aisée. Il est commode d’avoir recours à des éléments de bases appelés squelettes : tels que des points, des segments, des triangles qui vont venir définir des primitives implicites en générant un potentiel calculable dans leur espace environnant. En assemblant différents squelettes et en mélangeant leurs potentiels respectifs (généralement par une sommation), un objet 3D peut être modélisé comme l’isosurface du champ total.

La définition la plus classique de telles fonctions de champ revient à considérer des fonctions décroissantes de la distance la plus courte entre un point de l’espace et l’élément du squelette. Par exemple  $h(d) = 1/d^2$ , où  $d$  est la distance entre une position  $\mathbf{x} \in \mathbb{R}^3$  et le squelette. Si cette définition donne satisfaction dans le cas de squelettes ponctuels [Bli82 ; Mur91], elles génèrent des bosses et discontinuités de dérivabilité dans le cas de squelettes 1D et 2D.

### 9.2 Principe

Une définition permettant d’obtenir de meilleures propriétés de dérivabilités [Blo97] et de mélange consiste à modéliser le squelette comme une somme infinie de squelettes ponctuels. Le champ généré par le squelette est alors obtenu par l’intégrale d’une fonction noyaux  $h$  le long du squelette [MS98 ; JT02]. C’est à dire

$$f(\mathbf{x}) = \int_{\mathbf{p} \in S} h(\|\mathbf{x} - \mathbf{p}\|) d\mathbf{p} .$$

La surface résultante est alors appelée surface de convolution [MS98 ; BS91 ; She99] (voir fig. 10).

### 9.3 Travail demandé

- Réaliser une étude bibliographique de l’art de l’utilisation des fonctions implicites.
- Réaliser une étude bibliographique sur l’utilisation des surfaces implicite en modélisation. Placer les surfaces de convolution dans ce contexte.
- Implémenter une modélisation par surface de convolution le long de segments en 2D puis en 3D.
- Tester différents noyaux d’intégration et comparez à une fonction de distance la plus courte.
- + Réaliser un outil de modélisation interactif 3D permettant de visualiser le champ et la surface.
- + Implémentez une méthode d’intégration numérique de noyaux quelconque.

### 9.4 Question spécifiques

- Que sont les primitives implicites des squelettes de bases ?
- quand-est-ce que le champ  $f$  peut être calculé analytiquement ? Que peut-on faire dans le cas contraire ?
- Quel est le lien entre le degré de derivabilité de  $f$  et celui de  $h$  ?

### Références

- [BBCW10] Adrien BERNHARDT, Loic BARTHE, Marie-Paule CANI et Brian WYVILL. « Implicit Blending Revisited ». Dans : *Eurographics, Computer Graphics Forum (CGF)* 29.2 (2010), p. 367–375.
- [Bli82] James BLINN. « A Generalization of Algebraic Surface Drawing ». Dans : *ACM Transactions on Graphics (TOG)* 1.3 (1982).
- [Blo97] Jules BLOOMENTHAL. « Bulge Elimination in Convolution Surfaces ». Dans : *Computer Graphics Forum (CGF)* 16.1 (1997), p. 31–41.
- [BS91] Jules BLOOMENTHAL et Ken SHOEMAKE. « Convolution surfaces ». Dans : *ACM SIGGRAPH* (1991).
- [JT02] Xiaogang JIN et Chiew-Lan TAI. « Analytical methods for polynomial weighted convolution surfaces with various kernels ». Dans : *Computer and Graphics* 26.3 (2002), p. 437–447.
- [MS98] Jon MCCORMACK et Andrei SHERSTYUK. « Creating and Rendering Convolution Surfaces ». Dans : *Computer Graphics Forum (CGF)* 17.2 (1998), p. 113–120.
- [Mur91] Shigeru MURAKI. « Volumetric shape description of range data using “Blobby Model” ». Dans : *ACM SIGGRAPH* (1991).
- [She99] Andrei SHERSTYUK. « Convolution Surfaces in Computer Graphics ». Thèse de doct. Monash University, 1999.