

# Détection de Similarités de Surfaces Paramétriques

Quoc-Viet Dang, Sandrine Mouysset, Géraldine Morin

IRIT - Université de Toulouse

---

## Résumé

*Notre travail détermine des similarités locales sur des surfaces paramétriques, en particulier, pour des surfaces de B-Splines ou NURBS. Des parties de la surface similaires à une isométrie près sont identifiées. La détection de similarités dans des objets 3D maillés a été récemment étudiée dans la littérature en vue d'applications telles que l'alignement, la segmentation, l'édition de forme, ou encore la complétion de parties cachées. D'abord, nous proposons une orientation des signatures locales indépendante de la paramétrisation. De plus, une méthode de classification des paires de points dans un espace de transformation est utilisée. Nous adaptons cette approche, en améliorant la classification par une méthode spectrale. Nous illustrons les améliorations des résultats de la classification et l'identification de symétries et rotations.*

---

**Mots clé :** Modélisation géométrique, isométrie, détection de similarités, classification spectrale, édition de forme

## 1. Introduction

La similarité est un phénomène fréquent dans la nature, dans les objets synthétiques ou dans l'architecture. C'est une caractéristique essentielle à laquelle les artistes s'attachent dans leurs oeuvres, que les concepteurs de forme 3D doivent régler dans leurs conceptions ; en effet, la similarité crée une cohérence à laquelle le visuel humain est sensible en particulier pour une appréciation esthétique. Détecter ces similarités dans des modèles existants est donc une tâche importante, et conduit à de très nombreuses applications pour l'édition de formes, la complétion de parties manquantes ou encore la compression de modèles 3D. Par exemple, Chaouch et al. [CVB08] ont utilisé la réflexion comme une caractéristique pour aligner les modèles 3D, Li et al. [LYW\*11] proposent un système de complétion de données manquantes d'un modèle de crâne basé sur la symétrie, Mitra et al. [MGP06] produisent les modèles 3D édités en déformant les parties similaires à partir des originaux de ces modèles. En effet, la détection de similarités dans des objets 3D est un axe de recherche actuel et largement abordé [MGP06, LE06, PSG\*06, LCDF10]. Mitra et al. [MPWC12] proposent un état de l'art de ce domaine. D'autre part, les surfaces à pôles, et en particulier les B-splines ou plus généralement les NURBS sont des mod-

èles importants et passés dans les standards. Leur forme paramétrique permet d'accéder naturellement à des informations différentielles sur la surface. De plus, leur représentation par des points de contrôle permet un contrôle intuitif de ces surfaces. La détection de similarité sur des surfaces NURBS peut conduire aux mêmes applications que la détection proposée pour les maillages 3D, par exemple la compression ou la complétion. Les surfaces NURBS étant particulièrement attractives pour la conception, la détection de similarités locales peut être aussi intéressante dans le cadre du *reverse engineering* permettant d'analyser un modèle donné 3D, pour ensuite l'éditer de façon cohérente avec les similarités trouvées.

Cet article présente une méthode permettant d'identifier des parties d'une surface NURBS similaires à une isométrie près. Les contributions de ce travail sont les suivantes. Tout d'abord, pour orienter les vecteurs des repères caractéristiques sur un point de la surface, nous proposons une méthode simple basée sur l'analyse du voisinage. Nous distinguons les isométries directes et indirectes. Les isométries indirectes sont ramenées au cas direct afin de simplifier le traitement des données et d'obtenir des résultats plus cohérents. De plus, contrairement aux approches de l'état de l'art, notre approche de classification utilise une méthode spectrale, une heuristique non supervisée ayant la capacité de grouper automatiquement des classes sans réglage des paramètres globaux nécessaires à l'algorithme de *Mean Shift*. A notre connaissance, il n'y a aucune recherche dédiée

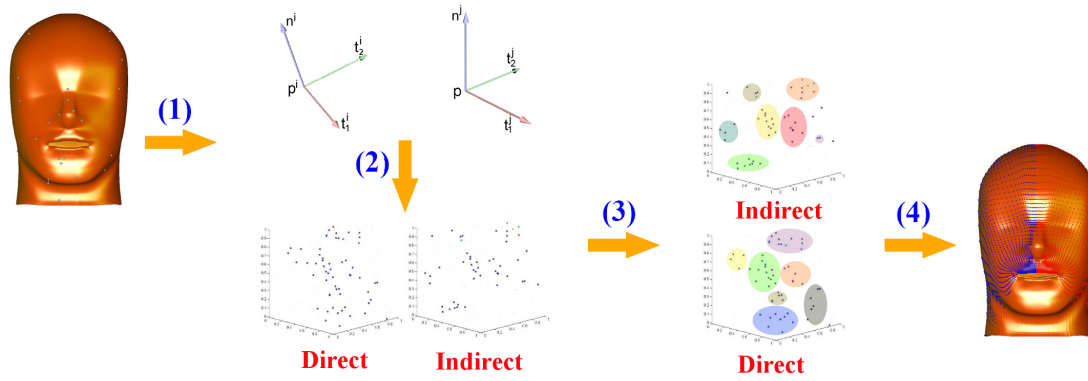


Figure 1: Chaîne de traitement – (1) échantillonnage et signatures, (2) passage dans l’espace des transformations, (3) classification, (4) validation.

à la détection de similarités dans les modèles paramétriques comme les B-Splines ou les NURBS.

Une première section détaille l’état de l’art et la section 3 suivante fixe les notations et détaille l’approche de Mitra et al. [MGP06] que nous avons suivie et modifiée. La section 4 détaille le calcul de la signature et l’orientation des vecteurs du repère correspondant. La section 5 présente la méthode de classification originale proposée. Enfin, les sections 6 et 7 présentent les résultats obtenus et notre application à l’édition de forme.

## 2. État de l’art

La détection de similarités est un sujet étudié largement dans le domaine d’images numériques en 2D mais aussi en 3D pour des objets représentés par un maillage. Zabrodsky et al. [ZPA95] ont proposé la *distance symétrique* comme une métrique mesurant la symétrie présente dans des objets 2D ou 3D. Cette distance est définie par le carré de la distance moyenne nécessaire au déplacement des points de l’objet original pour obtenir l’objet symétrique correspondant. Sun et al. [SS97] convertissent le problème de détection de symétries par corrélation avec l’image Gaussienne, dans laquelle les symétries rotationnelle et bilatérale sont déterminées en utilisant l’histogramme des orientations. Kazhdan et al. [KCD\*03] introduisent un descripteur de symétrie qui représente les symétries d’un objet 3D par rapport à tous les plans possibles passant par le barycentre de l’objet. Podolak et al. [PSG\*06] généralisent cette approche pour rechercher les symétries d’un objet 3D associées à un plan arbitraire. L’ensemble de ces plans est utilisé pour définir deux notions : les axes principaux et le centre de symétrie. Les axes principaux de symétrie sont l’ensemble des normales des meilleurs plans de symétrie. Le centre de symétrie est l’intersection de ces trois plans. D’autres travaux récents [Low04, LE06, MGP06, PMW\*08] utilisent une nouvelle approche dans la détection de symétrie. Cette approche utilise un ensemble de points de

la surface. A chaque point de l’objet, ils associent des caractéristiques. Les points de mêmes caractéristiques sont associés ; puis, à chacune de ces paires de points est associée la transformation entre ses deux points. Ces transformations sont groupées ; les groupes dominants (qui contiennent un grand nombre de transformations proches, c’est à dire un grand nombre de votes) représentent potentiellement des couples de régions similaires. Cette approche, efficace et robuste, est le point de départ de notre recherche.

## 3. Recherche de similarité : pipeline

Considérons  $\chi$ , une surface NURBS en produit tensoriel de bidegré  $(p, q)$  associée aux vecteurs de noeuds  $\mathbf{u} = \{u_0, \dots, u_n\}$  et  $\mathbf{v} = \{v_0, \dots, v_m\}$  et un réseau de points de contrôle  $\mathbf{P}$  de poids  $w$ , défini par l’équation :

$$\chi(u, v) = \frac{\sum_{i=0}^{n-p} \sum_{j=0}^{m-q} w_{ij} P_{ij} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^{n-p} \sum_{j=0}^{m-q} w_{ij} N_{i,p}(u) N_{j,q}(v)}. \quad (1)$$

Notre travail vise à identifier des régions de cette surface similaires à une isométrie près. Les isométries peuvent être directes ou indirectes mais nous ne considérons pas les homothéties, ni plus généralement les similitudes. Nous suivons et modifierons la chaîne de traitement de la méthode proposée par Mitra et al. dans [MGP06] pour les objets 3D maillés. Cette section consiste à rappeler les différentes étapes de leur algorithme basées sur un *système de votes*. Cet algorithme est divisé en quatre étapes illustrées par la figure 1.

1. **Echantillonnage et signatures** A chaque point de la surface peut être associée une signature caractérisée par les propriétés différentielles locales de la surface (détaillées en section 4.1). Dans les travaux de Mitra et al. [MGP06], les points considérés sont les sommets du maillage. Dans notre cas, une grille de points de la surface est obtenue en échantillonnant les paramètres du modèle en  $u$  et  $v$ . Pour

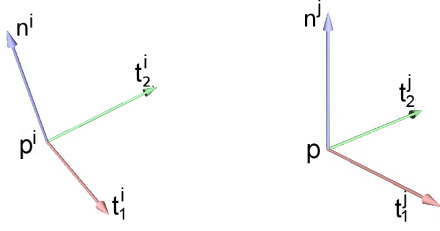


Figure 2: Deux repères locaux des deux points similaires  $p_i$  et  $p_j$  conformes à la règle de la main droite.

limiter le nombre de points, nous sélectionnons aléatoirement un nombre prédéfini de points sur cette grille comme points d'échantillonnage.

2. **Passage dans l'espace des transformations** Parmi les points échantillonnés sur la surface, les paires de points ayant des valeurs de courbure locales similaires sont mis en correspondance pour former une paire. On considère en chaque point échantillonné un repère orthonormé direct local  $(t_1, t_2, n)$ , composé de deux vecteurs tangents et d'un vecteur normal, dépendant des propriétés différentielles locales de la surface. Étant donnée une paire de points  $(p_i, p_j)$  de la surface paramétrique, la transformation de  $p_i$  à  $p_j$  est définie par  $T_{ij} \in \mathbb{R}^6$ , dont  $T_{ij} = (R_{ij}^x, R_{ij}^y, R_{ij}^z, t_{ij}^x, t_{ij}^y, t_{ij}^z)$ , où  $[R_{ij}^x, R_{ij}^y, R_{ij}^z]$  sont les trois angles d'Euler dérivés de la matrice de rotation  $R_{ij}$  et  $[t_{ij}^x, t_{ij}^y, t_{ij}^z]$  sont les trois composantes de la translation  $t_{ij} = p_j - R_{ij} * p_i$ . La matrice de rotation  $R_{ij}$  est calculée à partir des deux repères orthonormaux conformément à la règle de la main droite (voir la figure 2) :

$$R_{ij} = \begin{bmatrix} n^i \cdot n^j & n^i \cdot t_1^j & n^i \cdot t_2^j \\ t_1^i \cdot n^j & t_1^i \cdot t_1^j & t_1^i \cdot t_2^j \\ t_2^i \cdot n^j & t_2^i \cdot t_1^j & t_2^i \cdot t_2^j \end{bmatrix} \quad (2)$$

Chacune des paires de points correspond à une isométrie  $T_{ij}$  ; on considère ensuite l'espace  $\Gamma$ , de dimension 6, contenant ces isométries.

3. **Classification** Dans l'espace  $\Gamma$ , chacune de ces isométries  $T_{ij}$  représente une transformation entre deux points. Les paires de points de même transformation peuvent potentiellement correspondre à deux régions de la surface similaires à une isométrie près. Les méthodes de classification classiques comme K-means ou les recherches des  $K$  plus proches voisins sont utilisées pour résoudre ce problème. Cependant, nous ne disposons a priori d'aucune connaissance sur le nombre de régions similaires, ni sur le nombre de points proches d'un point donné dans l'espace de transformation. Nous proposons donc d'utiliser une méthode non supervisée, la classification spectrale automatique, qui permet de déterminer automatiquement les classes de façon stable sans avoir

à connaître le nombre de classes recherchées. Nous détaillerons cet algorithme dans la section 5.

4. **Validation** Chacune des classes obtenues par la classification se compose de points qui appartiennent idéalement à deux régions similaires à une isométrie près. Néanmoins, la cohérence spatiale entre les deux paires de points n'est pas assurée : c'est le but de cette dernière étape, dite de validation. La vérification est effectuée par un processus d'expansion de régions. Étant donnée  $C_k$  une classe de points dans l'espace des transformations, une paire de points  $(p_i, p_j)$  est sélectionnée aléatoirement. L'isométrie  $T_{ij}$  est appliquée à huit points proches de  $p_i$  et leurs images sont comparées à  $p_j$ . Le processus est appliqué itérativement pour les voisins dont l'image est suffisamment proche de  $p_i$ . Ce processus est répété jusqu'à ce que tous les points de la région aient été visités. Nous conservons cette étape de validation.

La transformation entre deux régions similaires correspond alors à une isométrie choisie arbitrairement parmi les points d'une même classe dans l'espace des transformations. Le calcul des points d'une région depuis une autre en appliquant cette transformation n'est donc pas exact. Une étape supplémentaire est mise en place pour rectifier l'isométrie initiale par une isométrie complémentaire. Nous utilisons l'algorithme ICP (Iterative Closest Points) proposée dans [RL01] pour calculer cette transformation complémentaire. L'isométrie finale entre les deux régions similaires est donc la composée de deux isométries successives.

La prochaine section détaille le calcul de la signature en un point pour les surfaces paramétriques considérées.

#### 4. Calcul de signature et orientation

Dans notre étude, la signature en un point de la surface paramétrique est un repère défini par la normale et les directions principales.

##### 4.1. Cas des B-Spline et NURBS

Le calcul des signatures d'un point spécifique sur une surface B-Spline ou NURBS est basé sur les propriétés locales différentielles ; leur calcul pour les surfaces NURBS est décrit par Farin [Far92]. La *première forme fondamentale* est définie par :

$$ds^2 = E du^2 + 2F du dv + G dv^2 \quad (3)$$

$$\text{où } \begin{cases} E = E(u, v) = \chi_u \chi_u \\ F = F(u, v) = \chi_u \chi_v \\ G = G(u, v) = \chi_v \chi_v \end{cases}$$

Le discriminant est défini par :

$$D = \|\chi_u \wedge \chi_v\| = \sqrt{EG - F^2} \quad (4)$$

Dans l'équation (3), les dérivées partielles  $\chi_u$  et  $\chi_v$  en un point  $\mathbf{x}$  engendrent le plan tangent à la surface en  $\mathbf{x}$ . Alors,

le vecteur unitaire normal

$$n = \frac{\chi_u \wedge \chi_v}{\|\chi_u \wedge \chi_v\|} = \frac{1}{D} (\chi_u \wedge \chi_v) \quad (5)$$

avec les vecteurs non normalisés  $\chi_u, \chi_v$ , forme un repère affine d'origine  $\mathbf{x}$ .

La courbure d'une région de la surface est basée sur la *deuxième forme fondamentale* :

$$\kappa \cos \phi ds^2 = Ldu^2 + 2Mdudv + Ndv^2 \quad (6)$$

$$\text{avec } \begin{cases} L = \chi_{uu} \cdot n \\ M = \chi_{uv} \cdot n \\ N = \chi_{vv} \cdot n \end{cases}$$

où  $\chi_{uu}, \chi_{uv}$  et  $\chi_{vv}$  sont les dérivées partielles secondes en un point  $\mathbf{x}$ .

L'équation (6) signifie que, pour une direction donnée  $du/dv$  dans le plan  $u, v$  et pour un angle donné  $\phi$ , la *deuxième forme fondamentale*, avec la *première forme fondamentale*, nous permettent de calculer la courbure  $\kappa$  d'une courbe tracée sur une surface et dont la tangente a cette direction.

Nous introduisons deux matrices symétriques :

$$\mathcal{F}_1 = \begin{pmatrix} E & F \\ F & G \end{pmatrix} \text{ et } \mathcal{F}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} \quad (7)$$

Puisque  $\chi_u$  et  $\chi_v$  sont linéairement indépendantes,  $\mathcal{F}_1$  est toujours inversible. De plus, la matrice  $\mathcal{F}_1^{-1}\mathcal{F}_2$  possède toujours des valeurs propres réelles. Les deux valeurs propres  $\kappa_1, \kappa_2$  sont les deux courbures principales et les deux vecteurs propres  $V_1 = (\xi_1, \eta_1)^T, V_2 = (\xi_2, \eta_2)^T$  définissent les deux directions principales  $t_1 = \xi_1\chi_u + \eta_1\chi_v, t_2 = \xi_2\chi_u + \eta_2\chi_v$ .

**Théorème 4.1** Etant données  $\kappa_1$  et  $\kappa_2$  les courbures principales en un point  $\mathbf{p}$  d'une région de la surface  $\chi$ . Alors :

- $\kappa_1, \kappa_2 \in \mathbb{R}$
- si  $\kappa_1 = \kappa_2 = \kappa$ , toutes les tangentes à  $p$  sont des vecteurs principaux, et le point  $p$  est appelé *point ombilic*.
- si  $\kappa_1 \neq \kappa_2$ , les deux vecteurs principaux correspondants sont perpendiculaires l'un à l'autre.

La signature en chaque point correspond à un repère orthonormé affine dont l'origine est ce point et les vecteurs sont le vecteur normal et les deux vecteurs principaux. Deux de ces repères orthonormés permettent de caractériser l'isométrie faisant passer de l'un à l'autre. Dans la suite de cet article, nous appellerons surfaces similaires deux régions de la surface similaires à une isométrie près. Dans le cas des points ombilics, les directions des axes sont mal définies, et donc nous ne considérerons pas ces points particuliers. Pour les autres points (non ombilics), les directions du repère sont bien définies par la paramétrisation. Dans le paragraphe suivant, nous expliquons comment nous déterminons le sens des vecteurs et le calcul de la transformation quelque peu différents de celui proposé par Mitra et al. [MGP06].

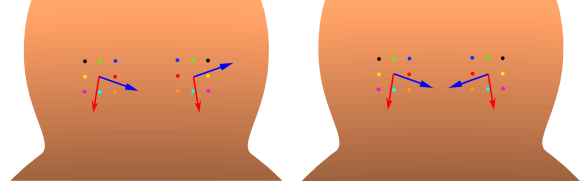


Figure 3: Surface paramétrique offrant une symétrie planaire (plan vertical entre les deux repères). Avant modification (à gauche) le sens des vecteurs est donné par la paramétrisation, après modification (à droite) le sens des vecteurs est déterminé par le voisinage.

## 4.2. Orientation du voisinage

Considérons deux points  $p_i$  et  $p_j$  du modèle paramétrique et leurs repères associés. Les directions principales sont caractérisées comme expliqué dans le paragraphe précédent, cependant le sens des vecteurs est déterminé par la paramétrisation; ce n'est donc pas une propriété géométrique de la surface. Pour cette raison, les repères locaux ne sont pas cohérents pour certaines isométries, les isométries indirectes en particulier. Nous proposons alors une méthode simple pour régler ce problème.

Supposons que le repère du point  $p_i$  fixé, c'est-à-dire, nous fixons arbitrairement le sens des vecteurs  $t_1^i$  et  $t_2^i$ . Considérons maintenant le repère en  $p_j$ . Nous pouvons observer qu'il existe quatre orientations différentes possibles pour le sens des vecteurs principaux (tangents) au point  $p_j$ . Notre méthode a pour but de déterminer l'orientation de ces vecteurs en  $p_j$  la plus cohérente par rapport à celle des vecteurs tangents en  $p_i$ . Cette méthode se compose des étapes suivantes :

1. Aux points  $p_i$  et  $p_j$ , déterminer ses huit points voisins au sens de la paramétrisation, c'est-à-dire sur la grille d'échantillonnage, et calculer les valeurs des courbures principales en chaque voisin.
2. Projeter les voisins de  $p_i$  (resp.  $p_j$ ) dans le plan tangent généré par les vecteurs  $(t_1^i, t_2^i)$  (resp.  $(t_1^j, t_2^j)$ ). Les points sont ordonnés, le premier point choisi est le plus proche du vecteur  $t_1$ . Les suivants sont choisis en tournant autour de  $p_i$  depuis  $t_1^i$  vers  $t_2^i$ . De même les voisins autour de  $p_j$  sont ordonnés. Donc, chaque configuration des tangentes principales donne une suite de courbures différente.
3. Chacune des quatre orientations de vecteurs tangents au point  $p_j$  est testée et sa suite des courbures correspondante est comparée avec celle de  $p_i$ ; l'orientation choisie est alors celle qui minimise la somme des valeurs absolues des différences entre sa suite et celle de  $p_i$ .

La figure 3 représente un modèle NURBS avec deux points symétriques dont les plans tangents se composent d'une tangente maximale (en bleue) et une tangente minimale (en rouge), et dont les points voisins sont indiqués en couleur. Chaque couleur correspond à une valeur de courbures; les points de même couleur ont les mêmes valeurs de

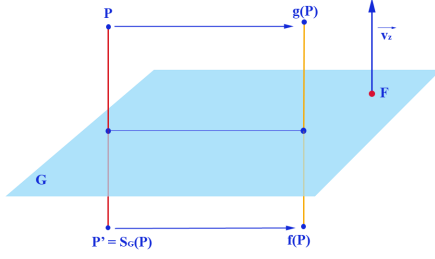


Figure 4: Description d'une isométrie indirecte  $f$  du point  $P$  par rapport au plan  $G$  et une transformation  $g$ .

courbure. Dans la figure de gauche, le sens de ces vecteurs est donné par la paramétrisation, nous observons que les tangentes minimales s'orientent de façon cohérente (entre le point cyan et le point violet), alors que les tangentes maximales s'orientent dans des sens différents (il faudrait être entre le point rouge et le point orange). La figure de droite représente le résultat final de notre méthode qui modifie l'orientation pour qu'elle soit cohérente avec le voisinage des points similaires.

Comme le sens du vecteur normal est inchangé après la modification, certains repères orthonormaux sont indirects (non conformes à la règle de la main droite). Il est donc impossible d'appliquer l'équation (2) pour calculer la matrice de rotation. Nous appliquons alors une symétrie planaire à l'isométrie indirecte pour retrouver le cas de l'isométrie directe.

### Isométrie indirecte

Soit  $X$  un espace affine de dimension 3 et  $f \in Is(X)$ . Supposons que  $f$  est indirecte (voir la figure 4),  $f = s_G \circ g = g \circ s_G$  [Tis88], alors :

- $g$  est soit une translation (pas de points fixes) soit une rotation (1 seul point fixe).
- $s_G$  : symétrie par rapport au plan  $G$  de vecteur normal  $\vec{v}_z$  qui est le vecteur propre associé à la valeur propre  $-1$  de  $\vec{f}$ ,  $\vec{f}$  étant la transformation vectorielle associée à  $f$ .

Avec cette approche, une transformation entre deux points similaires se représente par une isométrie soit directe soit indirecte. Supposons que  $p_j$  est un point similaire au point  $p_i$ , nous définissons alors les nouvelles formules pour ces transformations avec les isométries :

- Directe :  $p_j = T_{ij}(p_i)$ ,
- Indirecte :  $p_j = T_{ij} \circ s_G(p_i)$ .

Les cas d'isométries directes ou indirectes sont traités séparément. La section suivante expose la méthode de classification utilisée séparément sur l'ensemble des transformations directes, et sur l'ensemble des transformations indirectes issues d'une transformation indirecte.

## 5. Classification spectrale

L'étape de classification des paires de points dans un espace de transformation  $\Gamma$  est basée sur une approche spectrale, à savoir la classification spectrale. Le but est de regrouper les points/paires correspondants à des transformations proches (étape 3 de la section 3).

### 5.1. Méthode

Introduite dans le domaine de l'apprentissage automatique par *Shi et al.* [SM00], *Ng et al.* [NJW02] and *Meila et Shi* [MS01], la méthode de classification spectrale consiste à extraire les vecteurs propres associés aux plus grandes valeurs propres d'une matrice affinité gaussienne normalisée [VL07]. Ces vecteurs propres constituent un espace propre de dimension réduite dans lequel les données projetées sont regroupées en classes. L'algorithme associé à cette méthode de classification non supervisée est décrit ci-dessous (Algorithme 1). Nous rappelons que pour mesurer l'affinité entre deux transformations dans (9), la norme  $\|\cdot\|$  de la transformation  $T \in \Gamma$  est définie par [MGP06] :

$$\|\mathbf{T}\|^2 = \beta_1 \|R\|_2^2 + \beta_2 \|t\|_2^2, \forall T \in \Gamma, \quad (8)$$

où  $\|\cdot\|_2$  est la norme euclidienne,  $R \in \mathbb{R}^3$  est le vecteur des trois angles d'Euler,  $t \in \mathbb{R}^3$  est le vecteur des translations et  $\beta_i$  pour  $i = \{1, 2\}$  sont des poids servant à ajuster l'influence relative de chaque composante dans la transformation. En pratique, pour  $\beta_1 = 1/2\pi$ ,  $\beta_2$  est inversement proportionnel à la plus grande diagonal de la boite englobante de l'objet 3D.

---

#### Algorithm 1 Classification spectrale

---

Input : Ensemble  $S = \{T_{ij}\}_{i,j=1..N_T} \in \Gamma$ , nombre de classes  $k$ .

1. Construction de la matrice affinité  $A \in \mathbb{R}^{N_T \times N_T}$  définie par :

$$A_{il} = \begin{cases} \exp\left(-\frac{\|T_{ij}-T_{lm}\|^2}{(\sigma/2)^2}\right) & \text{si } (ij) \neq (lm), \\ 0 & \text{sinon.} \end{cases} \quad (9)$$

où  $\|\mathbf{T}\|$  est définie par (8).

2. Construction de la matrice normalisée :  $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$  avec  $D_{i,i} = \sum_{r=1}^{N_T} A_{ir}, \forall i \in \{1, ..N_T\}$ .
  3. Assembler la matrice  $X = [X_1 X_2 .. X_k] \in \mathbb{R}^{N_T \times k}$  formée à partir des  $k$  plus grands vecteurs propres de  $L$ .
  4. Construction de la matrice  $Y$  formée en normalisant les lignes de  $X$ .
  5. Traiter chaque ligne de  $Y$  comme un point de  $\mathbb{R}^k$  et les classer en  $k$  classes via la méthode *K-means*.
  6. Assigner le point original  $T_{ij}$  à la classe  $\theta$  si et seulement si la ligne  $i$  de la matrice  $Y$  est assignée à la classe  $\theta$ .
- 

Ng et al. [NJW02] justifie cet algorithme en considérant

un cas géométrique idéal avec trois classes bien séparées (Figure 5 (a)). En partant de l'hypothèse que les points sont déjà ordonnés consécutivement par classe, la matrice affinité bien que dense, a alors une structure numérique proche de celle d'une matrice bloc-diagonale (Figure 5 (b)). Ainsi, la plus grande valeur propre de la matrice affinité normalisée est 1 avec un ordre de multiplicité égal à 3. Les lignes normalisées de la matrice constituée des vecteurs propres associés aux plus grandes valeurs propres sont donc constantes par morceaux. Dans l'espace défini par les  $k = 3$  plus grands vecteurs propres, il est facile alors d'identifier les 3 points compacts séparés qui correspondent aux 3 constantes par morceaux des vecteurs propres (Figure 5 (c)). Ainsi en appliquant la méthode *K-means* pour  $k = 3$ , la classification est réalisée dans l'espace de projection spectrale : chaque point correspond à une classe. Enfin, par équivalence, la classification dans l'espace original est directement obtenue : chaque point correspond à un bloc (Figure 5 (d)).

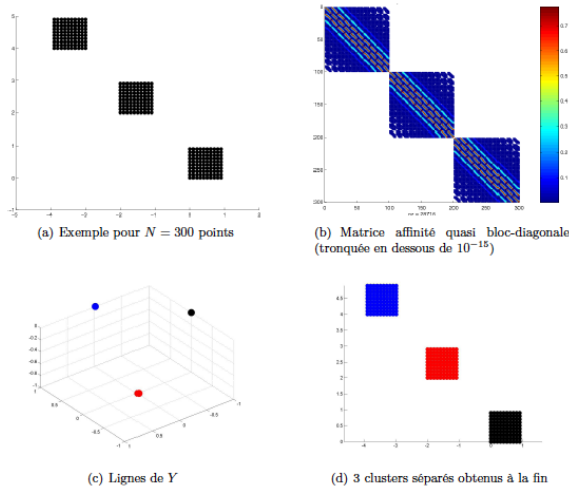


Figure 5: Illustration des étapes de la classification spectrale.

Cette méthode est principalement basée sur la mesure d'affinité gaussienne, son paramètre et ses éléments spectraux. Elle présente l'intérêt d'utiliser les propriétés inhérentes aux noyaux de Mercer (ici, le noyau gaussien) permettant de projeter implicitement les données dans un espace de grande dimension dans lequel les données seront linéairement séparables. Ainsi, des classes de formes arbitraires (notamment des domaines non convexes) peuvent être définies [VL07].

L'algorithme associé comprend peu d'étapes et peut être facilement codé sur Matlab. De plus, l'algorithme dépend uniquement de deux paramètres à savoir le paramètre d'affinité gaussienne et le nombre de classes  $k$ . Pour rendre cette méthode totalement non supervisée, nous utilisons une heuristique pour définir chaque paramètre [MNRG11].

## 5.2. Paramètre d'affinité

L'expression de l'affinité gaussienne, définie par l'équation (9), dépend donc d'un paramètre  $\sigma$ . Or,  $\sigma$  influe sur la séparabilité des données dans l'espace de projection spectrale. En effet, d'après Ng et al. [NJW02],  $\sigma$  contrôle la similarité entre les données et conditionne la qualité des résultats. Pour garder l'efficacité de la méthode, une approche globale est privilégiée. Comme le paramètre  $\sigma$  sert de seuil sur les distances des transformations entre les paires  $(p_i, p_j)$  de la surface paramétrique, ce dernier doit être alors fonction des distances entre les points et de la dimension du problème. On se base pour le définir sur le cas d'une distribution uniforme où tous les points sont équidistants les uns des autres. Soit  $S = \{T_{ij}\}_{i,j=1..N_T}$  un ensemble de  $N_T$  paires de  $(p_i, p_j)$  de la surface paramétrique dont la distribution est isotropique. Les éléments de  $S$  sont inclus dans une boîte carrée englobante de dimension 6 et de longueur de coté  $D_{max} = \max_{(ij) \neq (lm)} \|T_{ij} - T_{lm}\|$  avec  $\|T\|$  défini par (8). En divisant cette boîte en  $N_T$  volumes identiques, une distance uniforme, notée  $D_{unif}$ , est définie :

$$D_{unif} = \frac{\max_{(ij) \neq (lm)} \|T_{ij} - T_{lm}\|}{N_T^{1/6}}. \quad (10)$$

Donc on peut considérer que s'il existe des classes alors il existe des points qui sont séparés d'une distance inférieure à  $D_{unif}$ . Ainsi, le paramètre gaussien  $\sigma$  est pris égal à une fraction de la distance  $D_{unif}$  :

$$\sigma = \frac{D_{unif}}{2}. \quad (11)$$

Cette heuristique intègre donc la notion de densité de points dans l'ensemble des données 6-dimensionnelles et l'équation (11) donne un seuil à partir duquel des points sont considérés comme proches.

## 5.3. Nombre de classes

Le problème du choix du nombre de classes est un problème général pour les algorithmes de classification non supervisée. Pour déterminer ce nombre de classes  $k$ , la matrice d'affinité gaussienne  $A$  est à nouveau exploitée et une mesure de qualité basée sur des ratios de normes de Frobenius est définie.

Soit  $\alpha_k$  un nombre limite de classes à chercher. Pour une valeur  $k' \in [2, \alpha_k]$ , la matrice d'affinité est réordonnée suivant les  $k'$  classes. Une matrice par bloc est ainsi définie : les blocs hors diagonaux représentent les affinités entre les classes et les blocs diagonaux les affinités intra-classes. A partir de cette structure bloc, nous pouvons évaluer un ratio moyen, noté  $r_{k'}$ , entre tous les blocs hors diagonaux et les blocs diagonaux en norme de Frobenius. Ainsi, parmi les valeurs de  $k' \in [2, \alpha_k]$ , le minimum de ce ratio définit le nombre optimal de classes  $k$  :

$$k = \arg \min_{k' \in [2, \alpha_k]} r_{k'}. \quad (12)$$

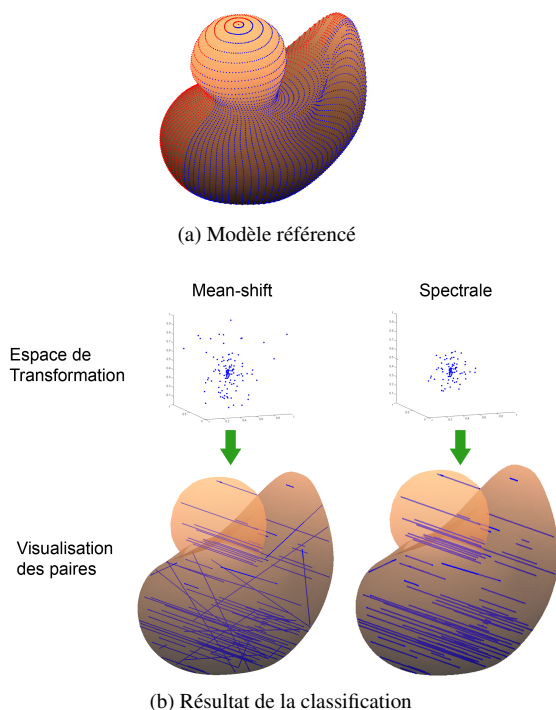


Figure 6: Comparaison d'une même classe définie par la méthode Mean-shift et la classification Spectrale.

En effet, ce minimum correspond au cas où l'affinité entre les classes est la plus faible et l'affinité au sein des classes est la plus forte.

## 6. Expérimentations

Nous avons implémenté la chaîne de traitement décrite dans la section 3 pour détecter les régions similaires de trois modèles NURBS.

La figure 6 permet de comparer les deux méthodes de classification : *Mean-shift* [MGP06] et notre approche *Spectrale Automatique* [MNRG11]. Dans la figure 6a, il existe une symétrie entre les deux parties qui correspond à une classe de transformation à déterminer. La figure 6b représente pour chaque méthode le résultat de recherche de cette classe de deux manières différentes : d'une part, les paires de points dans l'espace des transformations (ici en  $\mathbb{R}^3$ , selon les trois angles d'Euler) et, d'autre part, les paires de points associées sur la surface du modèle 3D. Chaque ligne sur la surface du modèle 3D correspond à un point dans l'espace de transformation qui relie donc deux points de même signature. Nous observons donc qu'il y a des points mal classés avec la méthode *Mean-shift*, alors que la méthode spectrale améliore ce problème. En effet, dans l'espace des transformations, la classe détectée par l'approche spectrale est plus compacte que celle de la méthode *Mean-shift*. Les points mal classés sont associés en appliquant l'approche spectrale à d'autres

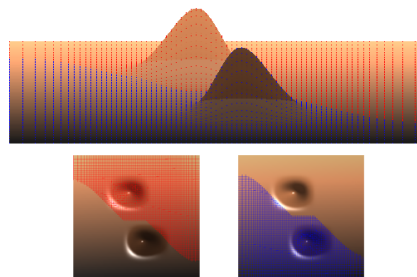


Figure 7: Modèle NURBS simple : deux régions similaires à une rotation de  $180^\circ$  et une translation près sont détectées.

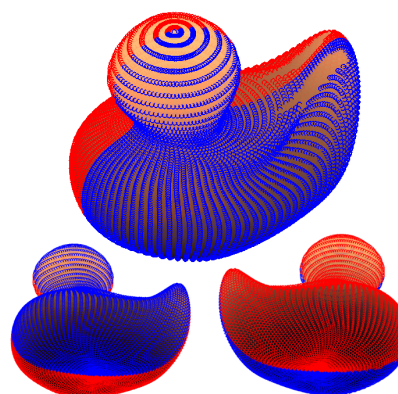


Figure 8: Modèle NURBS d'un canard : deux régions symétriques sont détectées.

classes qui contiennent des points de même transformation. Les figures 7, 8 et 9 représentent les couples de régions similaires à une isométrie près détectés sur les modèles, ces régions similaires sont colorées respectivement en bleu et en rouge. La figure 7 représente des données synthétiques fonctionnelles où les deux modes s'obtiennent par une rotation de  $180^\circ$  et une translation. Les figures 8 et 9 montrent les parties symétriques par rapport à un plan sur des modèles paramétriques.

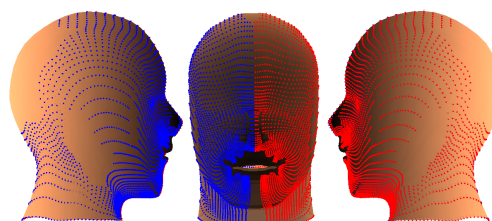


Figure 9: Modèle NURBS d'une tête humaine : deux régions symétriques sont détectées.



Figure 10: La translation d'un point de contrôle (en rouge) synchronise la translation du point de contrôle correspondant (en bleu) par rapport à la symétrie détectée.

## 7. Application

Nous souhaitons utiliser les résultats de la détection de similarités des modèles NURBS pour l'édition de forme de ces modèles en conservant les régions similaires. Supposons  $S_{orig}$  et  $S_{sim}$  les deux régions similaires à une isométrie près caractérisée par la transformation  $T$ . La région  $S_{sim}$  peut être calculée en appliquant la transformation  $T$  sur  $S_{orig}$  :

$$\begin{aligned} S_{sim}(u, v) &= T * S_{orig}(u, v) \\ &= T * \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} Q_{ij} N_i^k(u) N_j^l(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} N_i^k(u) N_j^l(v)} \\ &= \frac{\sum_{i=0}^n \sum_{j=0}^m N_i^k(u) N_j^l(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} N_i^k(u) N_j^l(v)} * \mathbf{T} * \mathbf{w}_{ij} * \mathbf{Q}_{ij} \end{aligned} \quad (13)$$

D'où  $Q_{ij}$  sont des points de contrôles qui définissent la région  $S_{orig}$ .

L'équation 13 montre que la transformation d'un modèle NURBS est identique qu'elle soit appliquée soit sur les points de contrôle, soit sur les points de la surface. De plus, il est possible de retrouver les points de contrôle correspondants à partir de deux régions similaires par un calcul inverse simple. Nous voudrions alors synchroniser la modification d'une région avec une modification cohérente de la région similaire en laissant les utilisateurs saisir les points de contrôle du modèle NURBS (voir figure 10). Nous pensons que cet outil d'édition cohérente peut être utile pour l'édition de modèles en 3D.

## 8. Conclusion

Nous avons présenté un algorithme de recherche de parties similaires sur des surfaces paramétriques B-splines ou NURBS. Cet algorithme suit l'approche proposée par Mitra et al. [MGP06]. Nous proposons une mise en correspondance des points originale, basée sur une étude du voisinage des points, permettant de trouver un sens cohérent des vecteurs caractéristiques. Nous traitons ainsi les transformations indirectes de façon à se ramener à un cas tout à fait similaire à celui des transformations directes. Enfin, l'étape de classification est une méthode spectrale non supervisée permettant d'éviter le choix de paramètres inhérents à la méth-

ode de K-means ou Mean Shift.

Par la suite, nous souhaitons généraliser cette classification pour unifier les étapes de classification et de validation. De plus, nous souhaitons améliorer le temps de calcul de notre système afin de pouvoir l'appliquer sur des modèles NURBS plus complexes, en particulier des modèles B-Rep contenant plusieurs surfaces NURBS.

## Références

- [CVB08] CHAOUCH M., VERRUST-BLONDET A. : *A Novel Method for Alignment of 3D Models*. Research Report RR-6408, INRIA, 2008.
- [Far92] FARIN G. : *Courbes et surfaces pour la CGAO - Conception Géométrique Assistée par Ordinateur*. Masson, 1992.
- [KCD\*03] KAZHDAN M. M., CHAZELLE B., DOBKIN D. P., FUNKHOUSER T. A., RUSINKIEWICZ S. : A reflective symmetry descriptor for 3d models. *Algorithmica*. Vol. 38, Num. 1 (2003), 201–225.
- [LCDF10] LIPMAN Y., CHEN X., DAUBECHIES I., FUNKHOUSER T. : Symmetry factored embedding and distance. *ACM Trans. Graph.* Vol. 29 (2010), 103 :1–103 :12.
- [LE06] LOY G., EKLUNDH J.-O. : Detecting symmetry and symmetric constellations of features. In *ECCV* (2006), pp. 508–521.
- [Low04] LOWE D. G. : Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*. Vol. 60, Num. 2 (novembre 2004), 91–110.
- [LYW\*11] LI X., YIN Z., WEI L., WAN S., YU W., LI M. : Cultural heritage : Symmetry and template guided completion of damaged skulls. *Comput. Graph.* Vol. 35 (August 2011), 885–893.
- [MGP06] MITRA N. J., GUIBAS L., PAULY M. : Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (SIGGRAPH)*. Vol. 25, Num. 3 (2006), 560–568.
- [MNRG11] MOUYSSET S., NOAILLES J., RUIZ D., GUIVARCH R. : On a strategy for spectral clustering with parallel computation. *High Performance Computing for Computational Science-VECPAR 2010* (2011), 408–420.
- [MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D. : Symmetry in 3d geometry : Extraction and applications. In *EUROGRAPHICS State-of-the-art Report* (2012).
- [MS01] MEILA M., SHI J. : A random walks view of spectral segmentation.
- [NJW02] NG A., JORDAN M., WEISS Y. : On spectral clustering : Analysis and an algorithm. *Advances in neural information processing systems*. Vol. 2 (2002), 849–856.



- [PMW\*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L. : Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*. Vol. 27, Num. 3 (2008).
- [PSG\*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T. : A planar reflective symmetry transform for 3d shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*. Vol. 25, Num. 3 (July 2006).
- [RL01] RUSINKIEWICZ S., LEVOY M. : Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)* (juin 2001).
- [SM00] SHI J., MALIK J. : Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. Vol. 22, Num. 8 (2000), 888–905.
- [SS97] SUN C., SHERRAH J. : 3d symmetry detection using the extended gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 19 (1997), 164–168.
- [Tis88] TISSERON C. : *Géométries Affine, Projective et Euclidienne*. HERMANN, 1988.
- [VL07] VON LUXBURG U. : A tutorial on spectral clustering. *Statistics and Computing*. Vol. 17, Num. 4 (2007), 395–416.
- [ZPA95] ZABRODSKY H., PELEG S., AVNIR D. : Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 17 (1995), 1154–1166.