HAL open science

Morphing multirésolution de courbes

Mélanie Cornillac

To cite this version:

Mélanie Cornillac. Morphing multirésolution de courbes. Modélisation et simulation. Université de Grenoble, 2010. Français. NNT : . tel-00581474

HAL Id: tel-00581474

https://theses.hal.science/tel-00581474

Submitted on 30 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : "Mathématiques Appliquées"

Arrêté ministériel : 7 août 2006

préparée au sein du Laboratoire Jean Kuntzmann
dans le cadre de **l'École Doctorale**
"Mathématiques, Sciences et Technologies de l'Information, Informatique"

présentée et soutenue publiquement

par

Mélanie CORNILLAC

le 9 décembre 2010

Morphing multirésolution de courbes

Directrice de thèse : Stefanie HAHMANN

JURY

Bernard LACOLLE	Professeur - Université de Grenoble	Président
Marc DANIEL	Professeur - Université de la Méditerranée	Rapporteur
Marc NEVEU	Professeur - Université de Bourgogne	Rapporteur
Stefanie HAHMANN	Professeur - Université de Grenoble	Examinateur
Jean-Claude LEON	Professeur - Université de Grenoble	Examinateur
Basile SAUVAGE	Maître de Conférences - Université de Strasbourg	Examinateur

Remerciements

Merci à Stefanie Hahmann pour m'avoir proposé ce sujet, et pour avoir encadré ma thèse.

Merci à Marc Daniel et Marc Neveu pour avoir accepté de relire ma thèse, et pour leurs remarques constructives. Merci à Bernard Lacolle de m'avoir fait l'honneur de présider mon jury. Merci à Jean-Claude Léon et Basile Sauvage pour leur participation à mon jury, et pour leur intérêt dans mon travail.

Merci à Hans Hagen et à toute son équipe pour m'avoir chaleureusement accueillie à TU Kaiserslautern pendant 6 mois.

Merci à toutes les personnes avec qui j'ai réalisé mes enseignements à l'Ensimag. Un merci particulier à François Bérard.

Merci à mes collègues de la tour IRMA, à Damien, Emilie, Boris, et à tous les autres qui se reconnaîtront même s'ils ne sont pas cités ici.

Merci à mes parents, qui me soutiennent avec beaucoup de constance depuis toujours.
Et enfin, merci Maxime pour ta présence et ton soutien dans les bons et moins bons moments.

Table des matières

I Introduction **9**

1 Introduction **11**

1.1 Morphing 11

1.1.1 Applications 12

1.1.2 Evaluation 14

1.2 Etat de l'art 14

1.2.1 Appariement des sommets 15

1.2.2 Morphing de courbes 17

1.2.3 Morphing de surfaces 24

1.3 Analyse multirésolution 25

1.3.1 Théorie de l'analyse multirésolution 26

1.3.2 Ondelette *paresseuse* 29

1.4 Contributions 30

II Morphing multirésolution de courbes planes **33**

2 Représentation intrinsèque d'une courbe 2D **37**

3 Analyse Multiresolution basée sur la représentation angle-longueur **39**

3.1 L'Analyse 40

3.2 La Synthèse 42

3.3 Les cas spéciaux 43

4 Morphing multirésolution **45**

4.1 Algorithme 46

4.1.1 Interpolation des coefficients grossiers 46

4.1.2 Interpolation des coefficients de détail 51
4.2 Résultats et discussion 52
4.2.1 Préparation des données 52
4.2.2 Représentation multirésolution de toutes les courbes 56
4.2.3 Morphing MR aussi-rigide-que-possible 56
4.2.4 Un exemple délicat 58

5 Etude de la stabilité de notre représentation intrinsèque MR 61
5.1 Stabilité liée au changement de coordonnées 61
5.2 Stabilité liée à l'analyse MR intrinsèque 64
5.3 Conclusion 67

III Morphing multirésolution de courbes 3D 71

6 Représentation intrinsèque d'une courbe 3D 75
6.1 Coordonnées sphériques locales 75
6.2 Représentation (l, θ, φ) d'une courbe 3D 77

7 Représentation multirésolution longueur-angles 79
7.1 Notations 80
7.1.1 Principe d'une analyse MR en 3D 80
7.1.2 Matrices de changements de base 83
7.2 Analyse 87
7.2.1 Coefficients grossiers 87
7.2.2 Coefficients de détail 89
7.3 Synthèse 93
7.3.1 Cas général 93
7.3.2 Cas particuliers 96

8 Morphing multirésolution de Courbes 3D 97
8.1 Algorithme 97
8.1.1 Interpolation des coefficients grossiers 98
8.1.2 Interpolation des coefficients de détail 99
8.2 Résultats et discussion 99
8.2.1 Intérêt par rapport à l'interpolation linéaire 100

8.2.2 Conservation des détails 102
8.2.3 Un exemple compliqué 102
8.2.4 Comparaison avec/sans MR 103

IV Bilan et perspectives 107

9 Bilan et perspectives 109

Première partie

Introduction

Chapitre 1

Introduction

1.1 Morphing

Le *morphing* (ou métamorphose) est la transformation progressive et lisse d'un modèle en un autre par interpolation. Ce modèle peut être en dimension 2 une image ou une courbe, ou en 3D une surface ou un volume. Le problème est de créer une transition entre deux formes qui soit esthétique et intuitive. Les formes intermédiaires doivent préserver l'apparence et les propriétés des formes en entrée. Quand nous parlons de *forme*, nous faisons habituellement référence à une propriété qui ne change pas avec l'orientation et la position d'un objet. En ce sens, préserver la forme signifie que l'objet peut voir son orientation et sa position modifiées, mais ni son échelle ni son étirement.

La simple interpolation linéaire entre les deux formes s'avère souvent inadaptée, comme illustré dans la Figure 1.1, car les formes intermédiaires ont tendance à beaucoup varier en volume ou encore à voir les proportions de leurs formes caractéristiques changer. Un autre effet négatif est que des détails géométriques peuvent disparaître pour réapparaître plus tard.

Le processus de *morphing* se décompose en deux problèmes : *le problème du couplage des sommets* (trouver une correspondance entre les caractéristiques géométriques des objets source et destination) et *le problème de la trajectoire des sommets* (trouver la trajectoire suivie par deux éléments correspondants au cours de la métamorphose). Ces deux problèmes suscitent toujours beaucoup d'intérêt en recherche, puisqu'il n'existe pas à ce jour de définition formelle d'une solution satisfaisante.

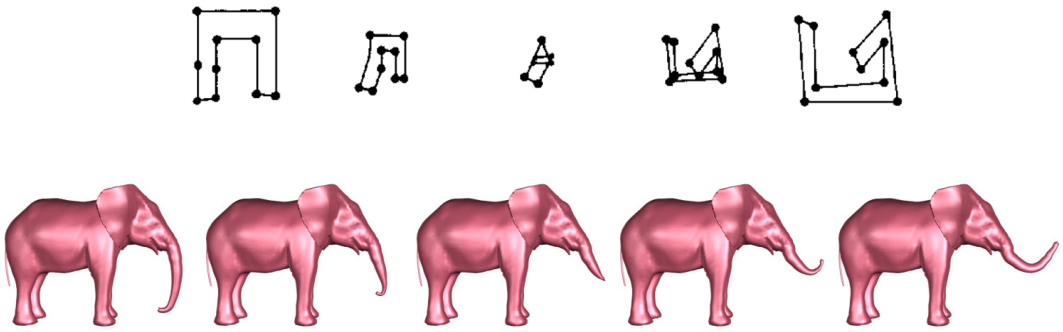


FIGURE 1.1 – Interpolation linéaire

1.1.1 Applications

Le morphing a de nombreuses applications dans différents domaines de l'informatique graphique, comme l'animation ou le design industriel. En animation, l'animateur principal dessine des positions clés (voir Figure 1.2) et différents intervalistes dessinent d'abord des étapes clés intermédiaires, puis les intervalles nécessaires à l'animation complète [TJ95]. C'est un travail fastidieux, encore parfois réalisé à la main (par exemple pour le long métrage *Persepolis* en 2007), qui peut être réalisé de façon automatisée par des méthodes de morphing.

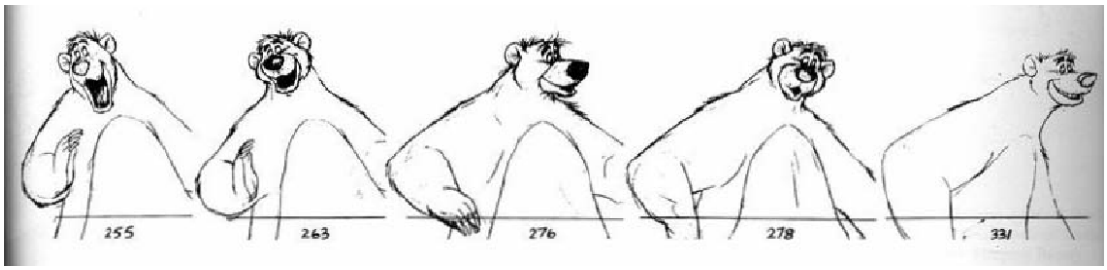


FIGURE 1.2 – Animation à partir d'images clés, exemple Disney, extrait de [TJ95]

Le morphing est aussi utilisé pour la création de nouvelles formes. Le designer peut générer des formes ayant les mêmes caractéristiques que des formes existantes. En imagerie médicale, la reconstruction du modèle 3D à partir des couches issues du scanner est aussi une variante de morphing, appelée l'*interpolation de couches*.

Le morphing est aussi communément utilisé pour réaliser des effets spéciaux dans les films ou à la télévision, par exemple le générique de l'émission *Thalassa* ou le clip de Mickael Jackson

Black or White en 1991. Les artistes contemporains allemands van Lawick et Müller ont travaillé avec le morphing de visages, voir la Figure 1.3 pour un exemple issu de leur série *Folie à Deux*, 1996.

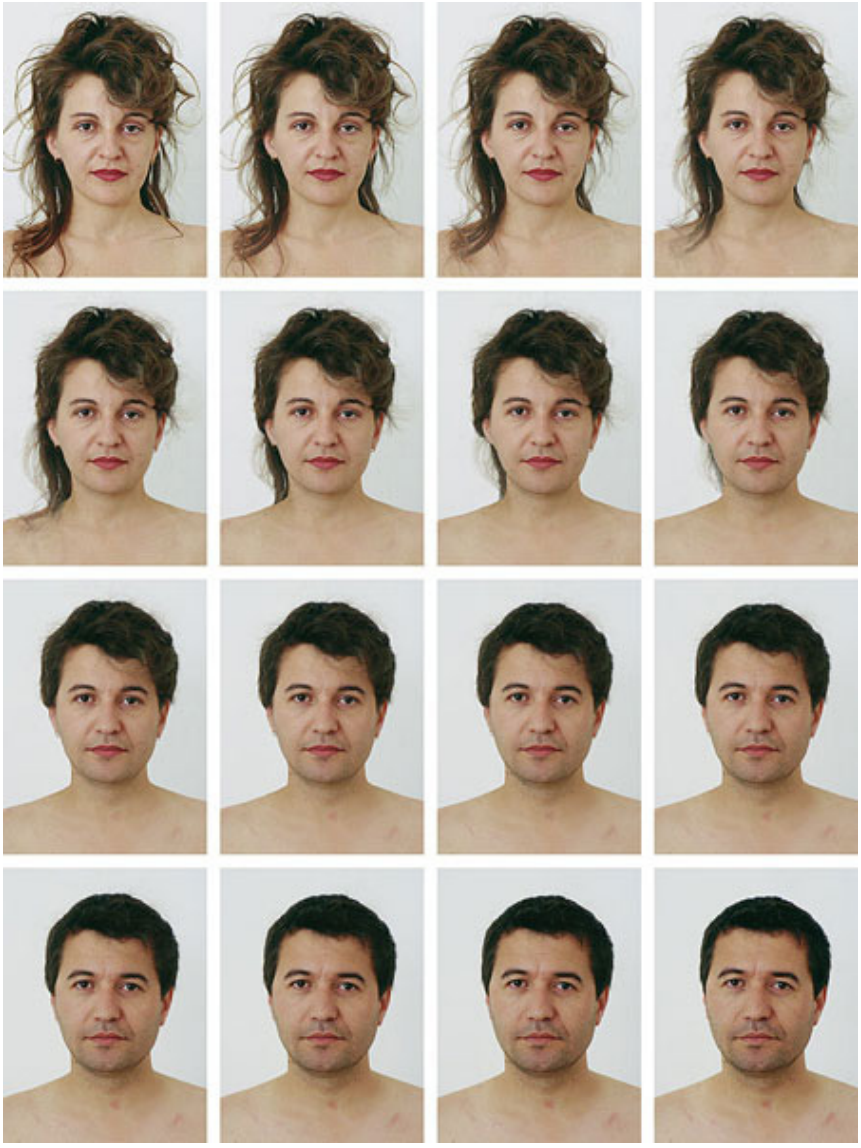


FIGURE 1.3 – Morphing de visages, par les artistes van Lawick et Müller.

Dans la plupart des cas, ces effets de morphing sont générés en utilisant des techniques basées sur les images où soit la géométrie des objets source et destination sont assez similaires (comme les visages de van Lawick et Müller ou ceux du clip de M. Jackson), soit le processus a requis un important travail de la part des graphistes (comme le générique de *Thalassa*). Dans cette thèse,

nous nous concentrons sur la création d'algorithmes capables de traiter des formes source et destination qui ont des géométries significativement différentes, avec une intervention minime de l'utilisateur, et de telle façon que les formes intermédiaires conservent autant que possible l'apparence et les propriétés de ces formes. Pour plus de détails sur les techniques de morphing d'images, le lecteur peut se référer aux états de l'art proposés par Wolberg [Wol98] et à celui proposé par Ruprech et al. [RM95]. Dans le reste de ce document, nous parlerons uniquement de morphing d'objets géométriques (courbes, surfaces maillées, ...).

1.1.2 Evaluation

Etant données deux formes, il existe une infinité de possibilités pour transformer l'une en l'autre. Quels critères nous permettent d'évaluer une de ces transformations ? Pour l'humain, il est facile et intuitif de juger si un morphing entre deux formes est agréable. Des critères esthétiques entrent bien sûr en ligne de compte, et sont fortement liés au contexte d'utilisation. Nous pouvons cependant dégager des critères mesurables qui assurent un bon morphing. Par exemple, les auto-intersections dans les formes intermédiaires doivent être évitées. Elles doivent aussi éviter les déformations superflues, qu'elles soient globales ou locales. De façon générale, les formes intermédiaires doivent conserver autant que possible les propriétés des formes en entrée, comme la longueur, l'aire ou la rugosité. B. Whited et J. Rossignac [WR09] proposent des mesures pour comparer les méthodes de morphing : la distance parcourue, la distorsion, l'étirement, l'accélération locale, l'aire, la courbure moyenne et la courbure maximale.

1.2 Etat de l'art

Comme rappelé dans l'introduction, Le processus de *morphing* se décompose en deux sous problèmes : *le problème du couplage des sommets* (trouver une correspondance entre les caractéristiques géométriques des objets source et destination) et *le problème de la trajectoire des sommets* (trouver la trajectoire suivie par deux éléments correspondants au cours de la métamorphose). Ces deux problèmes suscitent toujours beaucoup d'intérêt en recherche, puisqu'il n'existe pas à ce jour de définition formelle d'une solution satisfaisante.

Deux états de l'art ont été publiés sur le morphing [LV98, Ale02], le lecteur pourra s'y référer pour des études plus exhaustives que celle proposée ici. Le premier a été proposé par Lazarus et Verroust [LV98]. Les auteurs proposent une classification des méthodes de morphing selon leurs caractéritiques basiques : le type des données (polygones, maillages, données volumiques, ...),

l'interaction utilisateur nécessaire (l'utilisateur doit spécifier la correspondance, donner la trajectoire de certains sommets, ...), ainsi que le problème résolu (correspondance et/ou interpolation). Ils montrent l'importance de la représentation choisie pour les formes dans la qualité du morphing. L'état de l'art plus récent proposé par Alexa [Ale02], s'attache au morphing de surfaces maillées, mais évoque tout de même les méthodes de morphing de courbes notamment. Alexa étudie d'abord en détail les méthodes de mise en correspondance des formes, puis les différentes représentations de maillages, avant de détailler les méthodes utilisées pour la trajectoire des sommets. Il évoque aussi l'extension de ces méthodes à du morphing de plus de deux formes.

Dans cette section, nous évoquons d'abord le premier problème d'appariement des sommets, puis nous développons les méthodes proposées à ce jour pour le second problème de la trajectoire des sommets. Nous donnons un aperçu des méthodes de morphing de *courbes*, puis celles de morphing de *surfaces*. En modélisation géométrique, ces deux types de représentations : courbes et surfaces, sont les plus répandues. Les représentations volumiques sont plus anecdotiques, principalement à cause des coûts de calcul induits, nous choisissons donc de ne pas les traiter ici.

1.2.1 Appariement des sommets

La première étape du morphing, la mise en correspondance des sommets, est essentielle pour obtenir de bons résultats. La nécessité de faire correspondre les caractéristiques des formes est illustrée dans la Figure 1.4, extraite de [Ale02]. Alexa interpole un jeune cochon en un cochon adulte, une première fois sans mise en correspondance de caractéristiques (ligne du haut), une seconde fois avec une mise en correspondance manuelle (ligne du bas). Dans le premier cas, le résultat n'est pas acceptable, les modèles intermédiaires ont par exemple 8 pattes au lieu de 4. Dans le second cas, certaines caractéristiques sont mises en correspondance : les pattes, les oreilles, les yeux et la queue. Le résultat est bien meilleur.

De nombreuses méthodes ont été proposées pour accomplir automatiquement l'appariement des sommets, c'est un domaine de recherche très actif. Ce problème se rapproche de celui de la reconnaissance de forme [BCGJ95], mais avec la spécificité de chercher à obtenir une correspondance sommet à sommet des formes données en entrée, et pas seulement de détecter des formes similaires. De nombreuses approches différentes ont été publiées pour la mise en correspondance de formes 2D. Par exemple, Sederberg et Greenwood [SG92] proposent une approche physique qui minimise une fonction coût pour déterminer la correspondance. Semblablement, Zhang [Zha96] maximise une fonction similarité entre les sommets. Mortara et Spagnuolo [MS01] mettent eux en correspondance des squelettes décrivants les formes pour établir la cor-

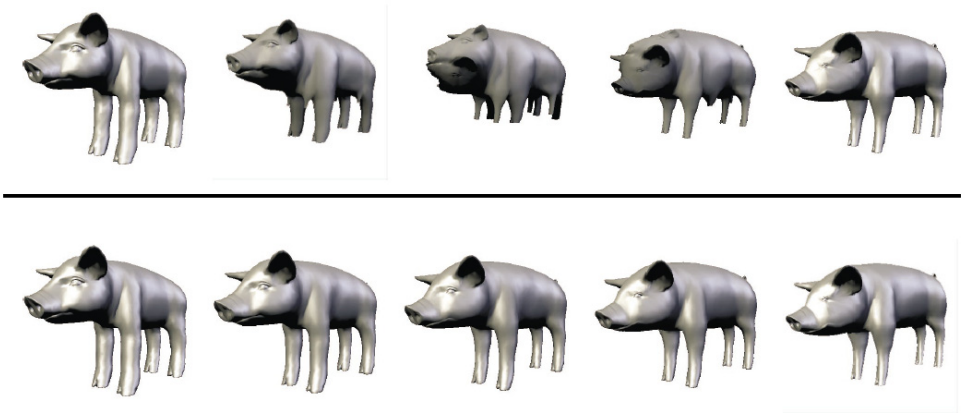


FIGURE 1.4 – Interpolation d'un jeune cochon en un cochon adulte. En haut : sans mise en correspondance des caractéristiques. En bas : avec une mise en correspondance de certaines caractéristiques : les pattes, les oreilles, les yeux et la queue.

respondance des sommets (voir Figure 1.5 haut). Liu et al. [LWZ+04] identifient d'abord les points caractéristiques des deux formes, puis la correspondance optimale est trouvée grâce à un programme dynamique efficace (voir Figure 1.5 bas). Notons que des méthodes de couplage automatiques sont aussi proposées pour les formes 3D, nous renvoyons le lecteur à l'état de l'art d'Alexa [Ale02] pour un aperçu des différentes approches possibles.

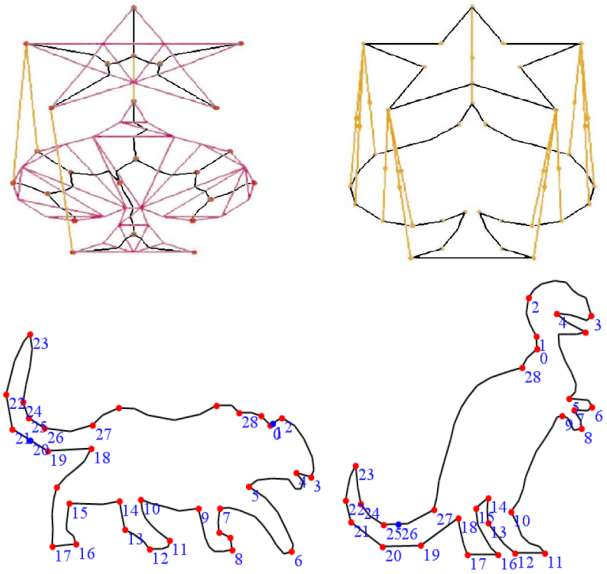


FIGURE 1.5 – Exemples de méthodes de mise en correspondance des sommets, extraits de [MS01] et [LWZ +04].

Nous venons donc de voir que de nombreuses méthodes de couplage automatiques sont proposées, aussi bien pour des formes 2D que 3D. Cependant, un bon couplage est sémantique, et pas nécessairement géométrique. Comme nous avons pû le constater sur l'exemple des cochons, nous souhaitons aparier des parties sémantiques des formes (les yeux, les oreilles,...), qui ne correspondent pas toujours à des caractéristiques géométriques de ces formes. Par exemple, si nous interpolons un chat vers un éléphant, nous souhaiterons faire correspondre le nez du chat avec la trompe de l'éléphant, qui sont très différents d'un point de vue géométrique. Selon nous, l'intervention humaine est donc nécessaire pour apparier les formes, en fonction de l'application et du résultat esthétique souhaité. L'appariement automatique reste réservé à des applications très particulières. Il serait intéressant de développer une méthode d'appariement des sommets qui fonctionne avec des entrées quelconques et des contraintes utilisateur quelconques, mais une telle méthode n'existe pas à ce jour. Certains travaux requièrent spécifiquement l'intervention de l'utilisateur pour définir la correspondance des sommets. Par exemple, Zöckler, Stalling et Hege [ScH00] proposent à l'utilisateur de sélectionner des groupes de sommets à mettre en correspondance (comme la zone du nez sur un visage ou celle de la bouche).

Dans de nombreux travaux, cette partie du travail est contournée soit en utilisant le même modèle dans différentes poses, soit en utilisant des modèles avec une topologie (connectivité) identique mais des géométries différentes.

Une fois le problème de l'appariement des sommets résolu, il reste à définir la trajectoire suivie par les sommets. Nous donnons ci-dessous un état de l'art des méthodes qui ont été proposées à ce jour, pour les courbes ainsi que pour les surfaces. Dans la plupart des cas, le problème de l'appariement des sommets est considéré comme résolu au préalable.

1.2.2 Morphing de courbes

Nous nous intéressons pour commencer aux méthodes de morphing de courbes. Deux types d'approches ont été proposées : certaines méthodes travaillent avec une représentation du *contour* de la forme, et d'autres travaillent avec une représentation de l'*intérieur* de la forme.

Interpolation linéaire

La méthode naïve pour déterminer la trajectoire des sommets consiste à interpoler le long d'un segment entre les positions des points couplés. En notant P_S et P_T les polygones source et destination, et $P_S^i = (x_S^i, y_S^i)$ (respectivement P_T^i) $\in \mathbb{R}^2$, $i = 0, \ldots, N-1$ leurs coordonnées

cartésiennes, les polygones intermédiaires P_t, avec $t \in [0,1]$ sont obtenus par interpolation linéaire de ces coordonnées :

$$\begin{aligned} x_t^i &= (1-t)x_S^j + t x_T^j, \\ y_t^i &= (1-t)y_S^j + t y_T^j, \end{aligned}$$

$$\text{pour} \quad 0 \leq i \leq N-1 \quad \text{et} \quad t \in [0,1].$$

Cette simple interpolation linéaire des sommets est connue pour modifier de manière inconsistante de nombreuses propriétés géométriques, telles que les longueurs, les angles et l'aire au cours du morphing, voir Figure 1.1 et Figure 1.6.

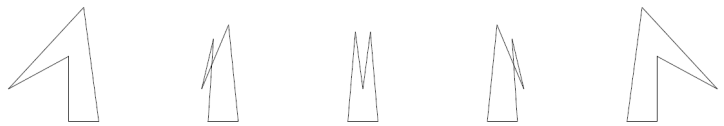


FIGURE 1.6 – Interpolation linéaire, observez les auto-intersections.

Interpolation longueur-angle

Sederberg et al. [SGWM93] ont présenté une méthode améliorée d'interpolation. Au lieu d'interpoler les positions des points, l'algorithme interpole des entités d'une définition intrinsèque du polygone : les angles et les longueurs des arêtes. En notant P_S et P_T les polygones source et destination, $\theta_S^i = \angle(\overrightarrow{P_S^{i-1}P_S^i}, \overrightarrow{P_S^iP_S^{i+1}})$ (resp. θ_T^i) l'angle entre deux arêtes consécutives et $l_S^i = \|\overrightarrow{P_S^iP_S^{i+1}}\|$ (resp. l_T^i), $i = 0,\ldots,N-2$ la longueur des arêtes. Les longueurs et angles décrivant les polygones interpolés P_t sont calculés en interpolant linéairement ces valeurs :

$$\begin{aligned} l_t^i &= (1-t)l_S^j + t l_T^j, \\ \theta_t^i &= (1-t)\theta_S^j + t\theta_T^j, \end{aligned}$$

$$\text{pour} \quad 0 \leq i \leq N-1 \quad \text{et} \quad t \in [0,1].$$

Une telle représentation des polygones est invariante par mouvement rigide. Il en résulte que le morphing de deux caractéristiques se comporte naturellement. Il s'avère aussi que les proportions de toutes les caractéristiques géométriques sont préservées. L'intérêt de ce morphing intrinsèque est particulièrement visible pour les trajectoires non linéaires, comme le bras du danseur de la Figure 1.7.

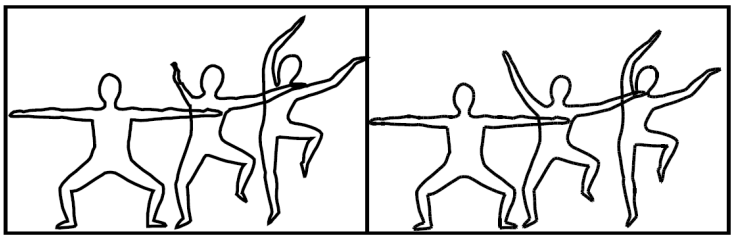


FIGURE 1.7 – Sederberg et al. [SGWM93]. A gauche : interpolation linéaire. A droite : interpolation des longueurs et angles.

Interpolation d'une représentation "squelette-étoilé"

De bons résultats sont généralement obtenus en choisissant d'abord une représentation alternative de la courbe avant de l'interpoler. Shapira et Rappoport [SR95] utilisent une représentation "squelette-étoilé". L'intérieur de la forme est décomposé en parties convexes, à chaque partie convexe est associé un squelette étoilé, reliant un point central aux sommets. Les différents points centraux sont reliés entre eux. Pour le morphing, les coordonnées polaires des sommets des squelettes sont interpolées linéairement. Cette méthode utilise l'information de l'intérieur de la courbe en plus de celle du contour, ce qui permet une meilleure préservation de la forme au cours du morphing (moins d'auto-intersections, moins de distorsion du volume), voir la Figure 1.8.

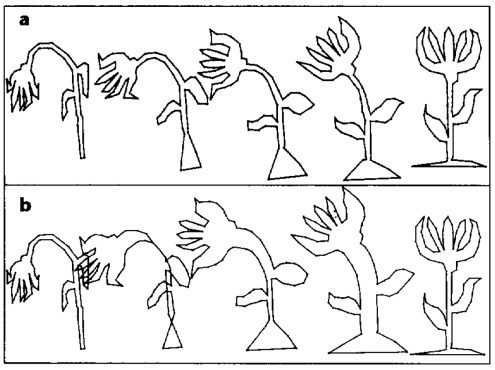


FIGURE 1.8 – (a) Interpolation d'une représentation squelette étoilé de la courbe [SR95] (b) Interpolation longueur-angle [SGWM93].

Interpolation d'une représentation multirésolution de la courbe

Goldstein et Gotsman [GG95] utilisent une représentation multirésolution basée sur l'évolution de la courbe. Le polygone P dont ses sommets sont notés $P^i = (x^i, y^i)$ est simplifié en modifiant la position de chaque sommet dans la direction inverse à la normale locale, d'une distance dépendant de la courbure locale. Les dérivées discrètes, sont
$\dot{x}^i = \frac{x^{i+1}-x^{i-1}}{s^{i+1}-s^{i-1}}$ et $\dot{y}^i = \frac{y^{i+1}-y^{i-1}}{s^{i+1}-s^{i-1}}$,
$\dot{x}^i_{suiv} = \frac{x^i-x^{i-1}}{s^i-s^{i-1}}$ et $\dot{x}^i_{prec} = \frac{x^{i+1}-x^i}{s^{i+1}-s^i}$,
$\ddot{x}^i = \frac{\dot{x}^i_{suiv}-\dot{x}^i_{prec}}{s^{i+1}-s^{i-1}}$ et $\ddot{y}^i = \frac{\dot{y}^i_{suiv}-\dot{y}^i_{prec}}{s^{i+1}-s^{i-1}}$,
avec $s^i = \sum_{j=0}^{i-1} ||p^{j+1} - p^j||$.
La courbure discrète locale k^i en P^i est $k^i = \frac{\dot{x}^i\ddot{y}^i - \ddot{x}^i\dot{y}^i}{((\dot{x}^i)^2+(\dot{y}^i)^2)^{3/2}}$. Cette étape de simplification est répétée jusqu'à obtenir un polygone qui soit convexe (voir Figure1.9 pour une illustration).

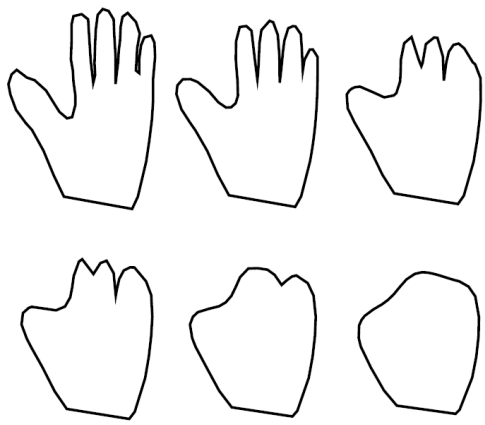


FIGURE 1.9 – Etapes de simplification d'une courbe selon [GG95]

La représentation multirésolution interpolée est constituée des trajectoires de chacun des sommets au cours de ces étapes. Cette méthode donne une qualité de résultat comparable à celle de Shapira et Rappoport [SR95], voir la Figure 1.10.

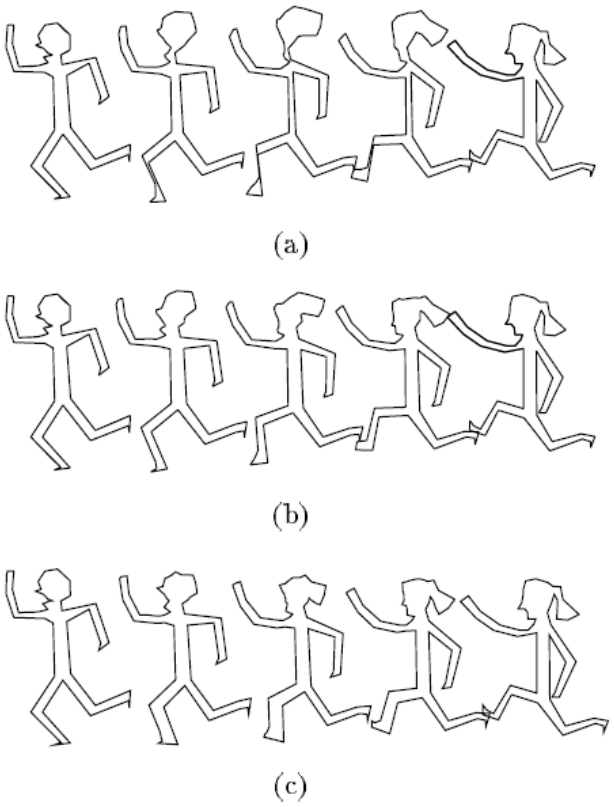


FIGURE 1.10 – (a) Interpolation longueur-angle [SGWM93]. (b) Interpolation d'une représentation squelette étoilé de la courbe [SR95]. (c) Interpolation d'une représentation multirésolution basée sur l'évolution de la courbe [GG95].

Nous venons de décrire des méthodes basées sur des représentation du contour des formes. Le morphing est aussi appliqué à des éléments décrivant l'intérieur de la forme 2D, comme des triangles.

Interpolation de coordonnées barycentriques

Ainsi, Floater et Gotsman [FG99] ont utilisé des coordonnées barycentriques pour métamorphoser des triangulations compatibles. Cette approche nécessite que non seulement l'intérieur mais aussi l'extérieur de la forme soit triangulé (voir Figure 1.11). Chaque sommet intérieur v_i est représenté par ses coordonnées barycentriques fonction des sommets de son 1-voisinage $V(i)$: $v_i = \sum_{j \in V(i)} \lambda_{i,j} v_j$, avec $\sum_{j \in V(i)} \lambda_{i,j} = 1$ et $\lambda_{i,j} > 0$. Chacune des 2 formes est entièrement décrite par sa matrice des pondérations $\Lambda = \lambda_{i,j}$. En notant Λ_S et Λ_T les matrices associées aux formes source et destination, les matrices de pondération des formes intermédiaires sont obtenue par interpolation linéaire :

$$\Lambda_t = (1-t)\Lambda_S + t\Lambda_T, \quad t \in [0,1]$$

Surazshky et Gotsman [SG03] proposent une amélioration de cette méthode, en utilisant une interpolation entre des entités intrinsèques. Ils interpolent la valeur moyenne des coordonnées barycentriques des triangles en utilisant une description par longueurs et angles des 1-voisinages de chaque sommet. Cette méthode a l'avantage par rapport aux autres que nous présentons de pouvoir garantir l'absence d'auto-intersections dans les courbes intermédiaires.

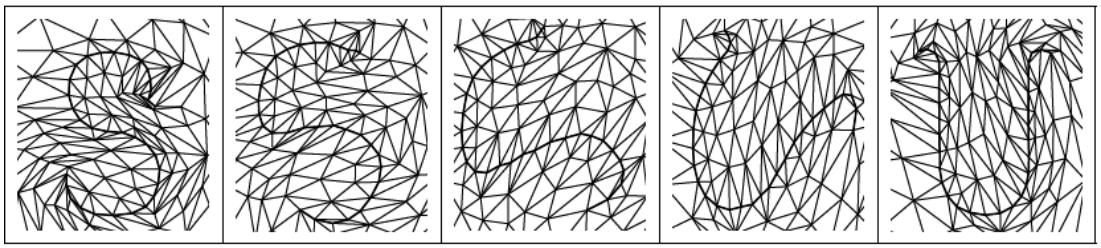


FIGURE 1.11 – Interpolation de la lettre S en la lettre U [SG03].

Interpolation aussi-rigide-que-possible

Alexa et al. [ACOL00] effectuent un morphing entre des triangulations compatibles en utilisant des applications locales qui minimisent les distorsions. Pour chaque triangle de la triangulation, la transformation affine A entre les triangles source et destination est décomposée en

une matrice de rotation R_γ et une matrice symétrique S en utilisant la décomposition en valeurs singulières : $A = R_\gamma S$. Les triangles intermédiaires sont ensuite obtenus en interpolant les composantes de cette décomposition : $A(t) = R_{t\gamma}((1-t)I + tS)$. Cette interpolation entre deux triangles est dite "aussi-rigide-que-possible" car les angles varient de manière pratiquement linéaire. Reste ensuite une étape d'optimisation pour que les triangles interpolés séparément forment toujours une triangulation. la Figure 1.12 donne un exemple de résultat obtenu avec cet algorithme. Nous reviendrons en détail sur cette méthode puisque nous en utilisons une partie dans une étape de notre algorithme de morphing (voir 4.1.1).

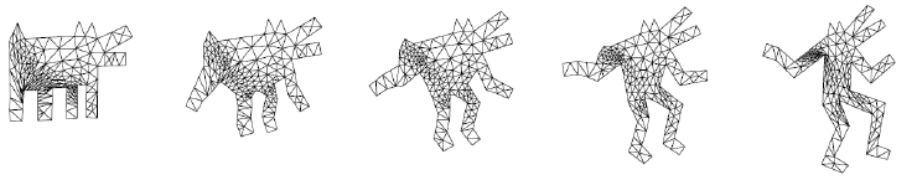


FIGURE 1.12 – Interpolation aussi-rigide-que-possible de Alexa et al. [ACOL00].

L'intérêt d'appliquer le morphing à des triangulations compatibles est qu'en général, les auto-intersections peuvent être évitées. Cependant, le calcul de triangulations compatibles ainsi que de représentations alternatives de l'intérieur des polygones [SR95] est généralement difficile et coûteux.

Conclusion

Au vu des méthodes proposées pour le morphing de courbes, la clé d'une méthode réussie semble être l'utilisation d'une représentation basée sur des propriétés intrinsèques à la géométrie des objets de façon à ce que l'interpolation de ses éléments donne automatiquement un résultat satisfaisant.

Remarquons que dans les résultats proposés dans les différents articles décrits ci-dessus, les formes proposées comportent peu de détails fins. En effet, éviter les auto-intersections et préserver les propriétés géométriques des formes intermédiaires est un problème particulièrement délicat dans le cas de formes très détaillées. Dans cette thèse, nous proposons un algorithme de morphing dont l'objectif est de traiter les formes comprenant de nombreux détails fins.

1.2.3 Morphing de surfaces

Nous nous intéressons maintenant aux travaux sur le morphing de surfaces maillées. Le travail de Lipman et al. [LSLCO05] peut être vu comme une sorte de généralisation de la technique d'interpolation 2D de Sederberg à l'interpolation de maillages 3D. Ils introduisent une représentation des surfaces invariante par mouvement rigide basée sur les formes fondamentales discrètes intrinsèques, et proposent d'interpoler les coefficients de ces formes. Sheffer et al. [SK04a] interpolent eux aussi une représentation locale invariante par rotation, basée sur des longueurs et angles, mais cette représentation nécessite la résolution d'un système non linéaire. Xu et al. [XZWB05] proposent d'interpoler non-linéairement le champ de gradient des surfaces, ce qui permet une bonne conservation des arêtes vives.

Mentionnons aussi une technique de morphing multirésolution de maillages par Lee et al. [LDSS99a], dont un résultat est présenté en Figure 1.13. Cette méthode a l'intérêt de travailler avec des maillages de connectivités différentes, mais induit de gros coûts de calcul.

Une approche rigoureuse est proposée par Kilian et al. [KMP07], qui calculent des géodésiques dans l'espace de toutes les formes intermédiaires possibles (géométrie riemannienne), cependant les temps de calcul ne sont pas évoqués, et les résultats proposés sont des modèles identiques dans différentes poses, ce qui rend difficile l'évaluation de cette méthode.

Les coordonnées laplaciennes sont populaires pour le traitement des maillages, nous pourrions donc imaginer les utiliser pour le morphing. Cependant, ces coordonnées sont une combinaison linéaire des coordonnées cartésiennes, les interpoler linéairement entraîne donc les mêmes problèmes qu'avec l'interpolation linéaire (voir Figure 1.1). Une interpolation plus complexe doit donc être envisagée pour utiliser les coordonnées laplaciennes pour le morphing [Sor06].

Remarquons que certains travaux requièrent l'interaction de l'utilisateur pour définir les trajectoires des sommets. Michikawa, Kanai et al. [MKFC01] utilisent une représentation multirésolution basée sur un remaillage semi-régulier de leurs modèles, l'interpolation des sommets grossier est ensuite définie à la main par l'utilisateur, puis propagée aux niveaux plus fins.

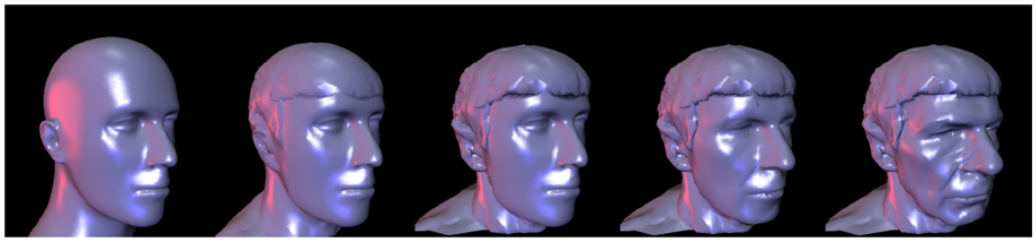


FIGURE 1.13 – Multiresolution Mesh Morphing - Lee et al. Siggraph'99 [LDSS99b]

1.3 Analyse multirésolution

L'analyse multirésolution a fait l'objet de beaucoup d'attention ces dernières années dans la recherche en informatique graphique, modélisation géométrique et visualisation [SDS96]. C'est un outil puissant pour représenter efficacement des fonctions à différents niveaux de détails (NdD), ce qui induit de nombreux avantages comme la compression, l'affichage des NdD et l'édition de NdD.

L'analyse multirésolution, introduite par S. Mallat [Mal89], est la combinaison d'un processus d'analyse et de synthèse. Le but d'une étape d'analyse est de générer une approximation grossière de basse résolution, ainsi qu'un ensemble de coefficients de détail, nécessaires pour reconstruire la fonction initiale. Cette étape peut être répétée jusqu'à obtenir une approximation suffisamment grossière. En partant du niveau le plus grossier, nous pouvons itérativement reconstruire les niveaux plus fins en réappliquant les détails. Une représentation bien choisie des approximations grossières et des détails doit permettre d'éditer la forme à l'échelle choisie avec deux effets principaux illustrés par la Figure 1.14 extraite de [FS94]. D'abord, modifier des points de contrôle basse-résolution puis réappliquer les détails modifie la forme globale de l'objet. Ensuite, modifier des coefficients de détail fins modifie le caractère de l'objet sans en affecter la forme globale.

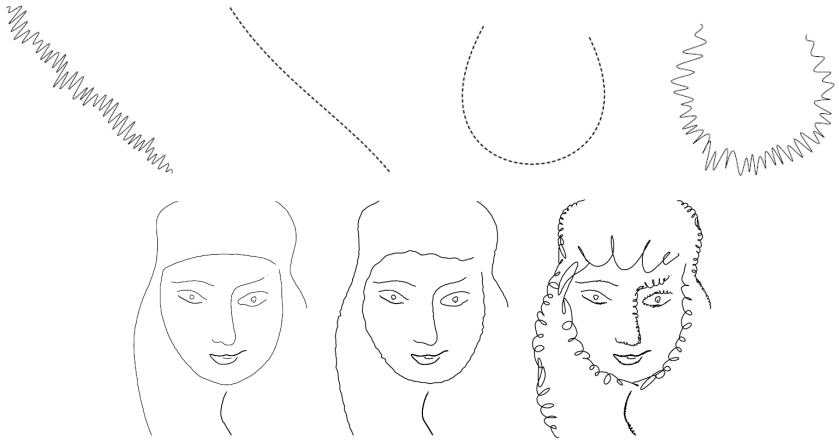


FIGURE 1.14 – Ligne du haut : effet de la modification de coefficients grossiers. Ligne du bas : effet de la modification de coefficients de détail.

Dans la littérature, le terme multirésolution (MR) est employé dans différents contextes, incluant la multirésolution en base d'ondelettes, la subdivision, les hiérarchies ou multigrilles. Des représentations multirésolution en bases d'ondelettes ont été développées pour des courbes pa-

ramétriques [CQ92, LM92, FS94], et peuvent être généralisées à des surfaces produit tensoriel [KE97, FS94], à des surfaces de topologie arbitraire [LRW97], à des données sphériques [SS95], et à des données volumiques [CMPS97]. D'autres représentations multirésolution existent pour les surfaces produit tensoriel, appelées B-slines hiérarchiques [FB88], pour des surfaces maillées [EDD+95, ZSS97, GVSS00], mais elles ne proposent pas toujours une reconstruction exacte, pour des données volumiques représentées par des fonctions à trois variables [RE00], ou pour des surfaces nuages de points, en utilisant le sous-échantillonnage et le sur-échantillonnage [DBP08, ZYC05].

Les ondelettes ont l'avantage de procurer un cadre mathématique unifié rigoureux (orthogonalité des fonctions de base, moments nuls, meilleure approximation, ...). Cependant, les ondelettes ne conviennent pas pour l'édition de niveaux de détails et la déformation de formes géométriques, en particulier dans le cas de déformations de grande échelle. En effet, lorsque une déformation de grande échelle est appliquée à une approximation grossière d'une forme de telle façon que sa forme générale est modifiée, il est attendu d'une analyse MR que les détails suivent naturellement la forme globale. Ce n'est pas le cas en utilisant les ondelettes B-splines par exemple [FS94]. Les coefficients de détail étant encodés dans un repère global, ils gardent une orientation fixe. Finkelstein et Salesin [FS94] proposent de résoudre ce problème en encodant les détails dans un repère local relatif à l'approximation grossière, de façon similaire à [FB88]. Des schémas multirésolution basés sur l'évolution de la courbe [GG95] ou sur une représentation intrinsèque de la courbure [HBCC07] ont montré leurs bonnes performances pour l'application particulière du morphing. Toutes ces méthodes pour les courbes 2D ont des filtres d'analyse et de synthèse non linéaires. Cependant, une expression explicite des coefficients grossiers et de détails est donnée, de telle façon que l'évaluation reste linéaire selon le nombre de points de contrôle pour chaque niveau de résolution. Les représentations multirésolution de courbes 2D et 3D que nous présentons dans cette thèse présentent les même caractéristiques.

1.3.1 Théorie de l'analyse multirésolution

Considérons un signal discret C^n, exprimé comme un vecteur colonne d'échantillons $[c_1^n, \ldots, c_m^n]^T$. Dans l'application aux courbes étudiée dans cette thèse, les échantillons c_i^n peuvent être vus comme les points de contrôle de notre courbe.

Nous souhaitons créer une version basse résolution C^{n-1} de C^n, avec un plus petit nombre d'échantillons m'. L'approche standard pour créer les m' échantillons de C^{n-1} est d'utiliser un filtrage ou un sous-échantillonnage sur les m échantillons de C^n. Nous pouvons exprimer ce

processus matriciellement :

$$C^{n-1} = A^n C^n$$

où A^n est une matrice $m' \times m$.

Puisque C^{n-1} contient moins d'échantillons que C^n, il est clair qu'une certaine quantité de détails est perdue dans cette étape de filtrage. Si A^n est bien choisie, il est possible de représenter cette perte de détails par un autre signal D^{n-1}, calculé ainsi :

$$D^{n-1} = B^n C^n$$

où B^n est une matrice $(m - m') \times m$, dépendante de A^n. Le couple de matrices A^n et B^n est appelé *filtres d'analyse*. A^n est un filtre passe-bas, B^n est un filtre passe-haut. Le processus de séparer un signal C^n en une partie basse-résolution C^{n-1} et une partie de détails D^{n-1} est appelé *décomposition*.

Si A^n et B^n sont bien choisies, le signal original C^n peut être retrouvé à partir de C^{n-1} et D^{n-1}, en utilisant un autre couple de matrices P^n et Q^n, appelés les *filtres de synthèse*, de la façon suivante :

$$C^n = P^n C^{n-1} + Q^n D^{n-1}$$

Retrouver C^n en utilisant C^{n-1} et D^{n-1} est appelé la *reconstruction*. Ce processus peut être appliqué récursivement, ainsi le signal original C^n peut être exprimé comme une hiérarchie de signaux de plus basse résolution $C^0, \ldots, C^{n-1}$ ainsi que de détails $D^0, \ldots, D^{n-1}$. Ce processus récursif est appelé *banc de filtres* [Mal89], voir la Figure 1.15.

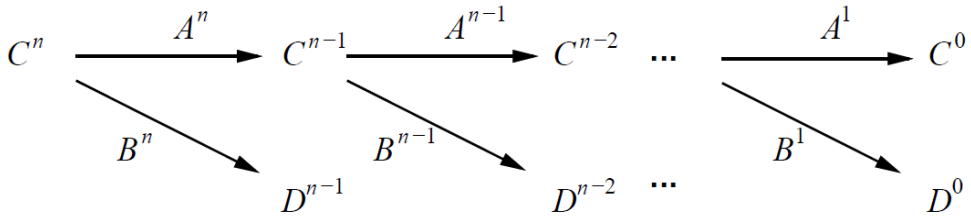


FIGURE 1.15 – Banc de filtres

Puisque le signal original C^n peut être retrouvé à partir de la séquence $C^0, D^0, \ldots, D^{n-1}$, cette séquence peut être vue comme une transformée du signal original, appelée *tranformée en ondelette*. Notons que la taille totale de la séquence $C^0, D^0, \ldots, D^{n-1}$ est exactement la même que celle du signal original C^n.

Tout ce dont nous avons besoin pour calculer la transformée en ondelette est d'un ensemble de filtres d'analyse et de synthèse A^j, B^j, P^j et Q^j. Pour voir comment construire ces filtres, nous associons à chaque signal C^n une fonction $f^n(u)$, avec $u \in [0,1]$, et $f^n : \mathbb{R} \to \mathbb{R}$.

$$f^n(u) = \Phi^n(u)C^n$$

où $\Phi^n(u)$ est une matrice ligne des fonctions de base $\left[\phi_1^n(u), \ldots, \phi_m^n(u)\right]$, appelées *fonctions d'échelle*.

Les fonctions de base sont supposées raffinables, i.e. pour tout $j \in [1,n]$, il doit exister une matrice P^j telle que

$$\Phi^{j-1} = \Phi^j P^j$$

Comme suggéré par la notation, la matrice de raffinement est en fait le filtre de synthèse P^j.

Ensuite, soit V^j l'espace défini par les fonctions d'échelle Φ^j. La condition de raffinement sur Φ^j implique que ces espaces linéaires sont imbriqués : $V^0 \subset V^1 \subset \ldots \subset V^n$. Les *espaces de détails* W^j sont les complémentaires de V^j dans V^{j+1}, c'est à dire que l'espace W^j, dont les fonctions de base $\Psi^j = \left[\psi_1^j(u), \ldots, \psi_{m-m'}^j(u)\right]$ sont telles que Φ^j et Ψ^j forment une base de V^{j+1}, et chaque $\psi_i^j(u)$ est orthogonal à chaque $\phi_i^j(u)$. Les fonctions de base $\psi_i^j(u)$ sont appelées *ondelettes*.

Nous pouvons maintenant construire le filtre de synthèse Q^j comme la matrice qui vérifie :

$$\Psi^{j-1} = \Phi^j Q^j$$

Nous obtenons, en concaténant les matrices :

$$\left[\begin{array}{c|c} \Phi^{j-1} & \Psi^{j-1} \end{array}\right] = \Phi^j \left[\begin{array}{c|c} P^j & Q^j \end{array}\right]$$

Finalement, les filtres d'analyse A^j et B^j sont formés par les matrices qui satisfont la relation inverse :

$$\left[\begin{array}{c|c} \Phi^{j-1} & \Psi^{j-1} \end{array}\right] \left[\frac{A^j}{B^j}\right] = \Phi^j$$

Remarquons que $\left[\begin{array}{c|c} P^j & Q^j \end{array}\right]$ et $\left[\begin{array}{c|c} A^j & B^j \end{array}\right]^T$ sont des matrices carrées, qui vérifient la condition de reconstruction :

$$\left[\begin{array}{c|c} P^j & Q^j \end{array}\right] \left[\frac{A^j}{B^j}\right] = I$$

Voici un exemple en utilisant les ondelettes de Chaïkin. Elles sont basées sur le schéma de subdivision de Chaïkin [Cha74] et sont également connues comme ondelettes B-spline quadratiques. La décomposition d'une courbe à $D2^n$ points de contrôle $c_0^n, \ldots, c_{D2^n-1}^n$ est donnée par

$$\begin{aligned} c_i^{n-1} &= \tfrac{1}{4}(-c_{2i-2}^n + 3c_{2i-1}^n + 3c_{2i}^n - c_{2i+1}^n) \\ d_i^{n-1} &= \tfrac{1}{4}(c_{2i-2}^n - 3c_{2i-1}^n + 3c_{2i}^n - c_{2i+1}^n), \end{aligned} \tag{1.1}$$

où tous les indices sont pris modulo le nombre de points de contrôle. La formule de construction pour Chaikin est :

$$\begin{aligned} c_{2i}^n &= \tfrac{3}{4}(c_i^{n-1} + d_i^{n-1}) + \tfrac{1}{4}(c_{i+1}^{n-1} - d_{i+1}^{n-1}) \\ c_{2i+1}^n &= \tfrac{1}{4}(c_i^{n-1} + d_i^{n-1}) + \tfrac{3}{4}(c_{i+1}^{n-1} + d_{i+1}^{n-1}). \end{aligned}$$

La figure 1.16 illustre les approximations successives d'une courbe de subdivision de Chaikin obtenues par la décomposition 1.1. La figure 1.17 montre l'effet d'une déformation du polygone grossier. Cet exemple a été généré avec l'application associée à la thèse de B. Sauvage [Sau05]. Le polygone fin reconstruit à partir du polygone grossier déformé et des détails originaux est montré à droite. Nous constatons que l'orientation des détails ne suit pas la déformation, ce qui induit un comportement peu naturel des détails fins. En particulier, les pics situés près de la tête du hérisson gardent une orientation verticale, provoquant des auto-intersections des pics. Nous verrons par la suite que la représentation MR intrinsèque que nous proposons dans cette thèse résoud ce problème d'orientation des détails.

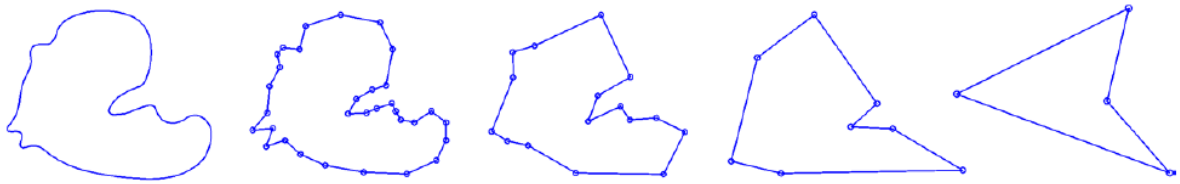


FIGURE 1.16 – De gauche à droite : une courbe de Chaikin définie par 32 points de contrôle, son polygone de contrôle (niveau 5), puis les polygones de contrôle des approximations correspondant au niveaux 4, 3 et 2.

1.3.2 Ondelette *paresseuse*

L'ondelette *paresseuse* (*lazy wavelet*) est encore plus simple que l'ondelette de Haar. Elle est utilisée pour une étape du *lifting scheme* développé par Sweldens [Swe97]. Cette ondelette est dite *paresseuse* car elle ne fait rien d'autre que de sous-échantillonner le signal.

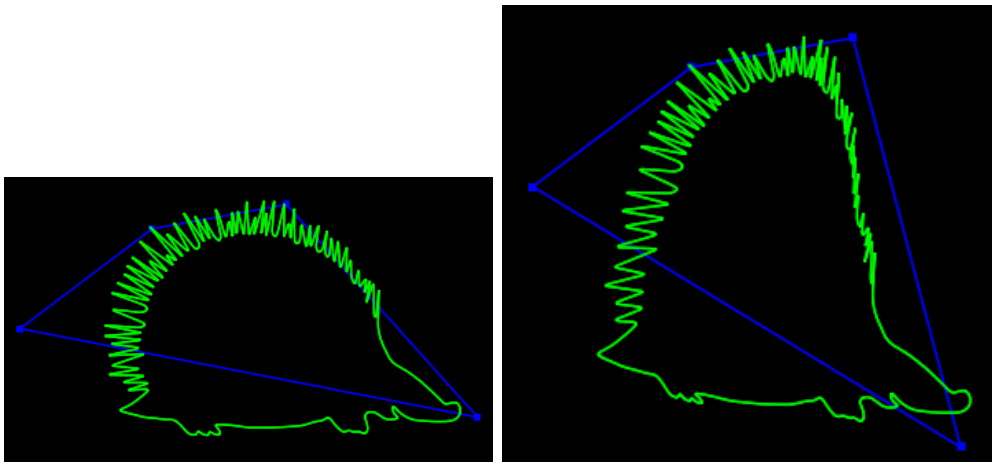


FIGURE 1.17 – A gauche : une courbe de Chaikin définie par 512 points de contrôle (en vert) et son polygône grossier (en bleu). A droite, une déformation est appliquée au polygone grossier avant la reconstruction.

Le banc de filtres correspondant aux filtres *paresseux* se contente de séparer les échantillons pairs des échantillons impairs dans un signal, sans filtrage supplémentaire. Cette séparation triviale est en soi déjà une analyse multirésolution. Les ondelettes et fonctions d'échelles *paresseuses* correspondant à ces filtres sont les Diracs $\phi(u) = \delta(u)$ et $\psi(u) = \delta(u - 1/2)$.

Appliqué au cas d'un signal qui est une courbe discrète, une analyse MR basée sur l'ondelette paresseuse va donc consister à ne conserver qu'un sommet sur deux de notre courbe à chaque étape d'analyse. Dans cette thèse, nous développons une analyse MR pour les courbes discrètes 2D et 3D, basée sur l'ondelette paresseuse.

1.4 Contributions

Nous avons vu en étudiant l'état de l'art que de bons résultats sont généralement obtenus en choisissant d'abord une représentation alternative de la courbe ou surface avant de l'interpoler. La clé d'une méthode réussie semble être l'utilisation d'une représentation basée sur des propriétés intrinsèques à la géométrie des objets de façon à ce que l'interpolation de ses éléments donne automatiquement un résultat satisfaisant.

Ensuite, nous souhaitons traiter le problème particulier du morphing de formes comprenant de nombreux détails. Ce problème n'a pas vraiment été traité à ce jour, les algorithmes publiés montrent des résultats sur des formes comprenant peu de détails fins. Nous choisissons d'utiliser une représentation multirésolution de nos courbes pour pouvoir traiter séparément le morphing

de leur formes grossières, et celui de leurs détails. Ainsi, nous pouvons obtenir de bons résultats sur des formes très détaillées. La représentation MR de courbes que nous introduisons dans cette thèse est basée sur des quantités intrinsèques : les longueurs et angles décrivant nos courbes discrètes. Notre analyse MR est basée sur les ondelettes paresseuses que nous avons présenté plus haut (Section 1.3.2). Nous introduisons ensuite notre algorithme de morphing, qui interpole les coefficients de notre représentation MR intrinsèque.

Nous présentons donc dans cette thèse un algorithme de morphing multirésolution intrinsèque de courbes, qui a l'avantage de pouvoir traiter des formes comprenant de nombreux détails fins. Nous déclinons notre représentation MR intrinsèque et notre algorithme de morphing pour les courbes 2D (Partie II) puis pour les courbes 3D (Partie III). Enfin, nous concluons et donnons les perpectives de nos travaux (Partie IV).

Deuxième partie

Morphing multirésolution de courbes planes

Résumé

Nous présentons une nouvelle méthode de *morphing* de courbes planes utilisant une combinaison entre une interpolation ”aussi-rigide-que-possible” et une décomposition multirésolution intrinsèque. Cette nouvelle représentation multirésolution est basée sur les angles et longueurs, et présente l’avantage que l’orientation des détails suit naturellement n’importe quelle déformation. Le principe du *morphing* multirésolution est de transformer séparément les coefficients grossiers et ceux de détail issus de la décomposition multirésolution. Ainsi, toutes les manipulations des niveaux de détail (NdD) comme l’affichage de NdD, la compression, l’édition, ... peuvent être appliquées directement aux métamorphoses sans calcul supplémentaire. De plus, nous montrons la robustesse de notre algorithme sur des polygones de grande taille avec de nombreux détails. Les polygones intermédiaires se comportent naturellement et leur distorsion est minimale grâce à la représentation multirésolution intrinsèque que nous avons choisie.

Problématique

La méthode de *morphing* que nous présentons est basée sur une nouvelle représentation multirésolution. Dans la modélisation de formes complexes comportant de nombreux détails géométriques qui sont donc difficiles à manipuler, l’analyse multirésolution offre un outil puissant pour représenter et déformer efficacement des objets à différents niveaux de résolution. Ainsi, une forme complexe est décomposée en une approximation grossière ainsi qu’un ensemble de coefficients de détail, permettant à tout moment la reconstruction du modèle initial. La déformation d’objets complexes comportant de nombreux détails peut être délicate et coûteuse à calculer. Dans un cadre multirésolution, même les objets complexes peuvent être édités à une échelle choisie avec deux principaux effets. Premièrement, modifier des points de contrôle basse-résolution modifie la forme globale de l’objet sans en affecter les détails. Deuxièmement, modifier des coefficients de détail fins modifie le caractère local de l’objet sans affecter sa forme globale.

Le *morphing* multirésolution décompose donc la forme source et la forme destination en une approximation grossière et un ensemble de coefficients de détail, il calcule séparément la suite de formes grossières intermédiaires et celle de détails, et enfin reconstruit toutes les formes intermédiaires. Le choix de la représentation multirésolution est déterminant pour la qualité des formes générées. En modélisation géométrique assistée par ordinateur (CAGD) ou en informatique graphique, les courbes ont généralement une représentation multirésolution (MR) basée

sur les B-splines [FS94, CQ92, EG95, Ber05, HBS05] et sont définies pour des applications particulières. Dans le cas des déformations, et le *morphing* appartient à cette application, l'orientation des détails est un problème majeur. Par exemple, une analyse multirésolution basée sur une ondelette ne préserve pas l'orientation des détails au cours de la transformation. En effet, les détails sont encodés dans un système global de coordonnées. Une représentation multirésolution qui encode les détails en utilisant des repères locaux comme [FS94] résout ce problème mais n'est pas appropriée pour le *morphing*. En effet, les polygones grossiers peuvent tirer profit du *morphing* intrinsèque de Sederberg. Mais les détails sont encodés sous forme de vecteurs, qui ne sont pas des quantités intrinsèques, donc une simple interpolation linéaire des détails souffrirait du même manque de réalisme qu'un *morphing* linéaire des sommets. Nous proposons une méthode de morphing multirésolution qui préserve l'orientation des détails et est appropriée pour le morphing.

Dans cet partie, nous définissons pour commencer notre représentation intrinsèque de courbes 2D dans le Chapitre 2. Nous introduisons ensuite une représentation multirésolution pour les courbes 2D polygonales basée sur la courbure dans le Chapitre 3. Tous les coefficients de détail seront représentés intrinsèquement par angles et longueurs. Nous montrons que, comme les repères locaux, la représentation multirésolution préserve l'orientation des détails au cours de la déformation. En outre, nous étudions son utilisation pour le *morphing* dans le Chapitre 4, en la combinant avec la technique de *morphing* "aussi-rigide-que-possible" de Alexa et al. [ACOL00] qui s'est avérée fournir des métamorphoses de bonne qualité. Nous présentons donc un algorithme de *morphing* multirésolution intrinsèque aussi-rigide-que-possible. Enfin, nous étudions la stabilité de notre représentation dans le Chapitre 5, avant de conclure.

Chapitre 2

Représentation intrinsèque d'une courbe 2D

Notons les sommets du polygone à décomposer $P_i = (x_i, y_i) \in \mathbb{R}^2$, $i = 0, \ldots, N-1$. Nous transformons alors la représentation en coordonnées (x, y) du polygone initial en une représentation en coordonnées (θ, l), où $\theta_i = \angle(\overrightarrow{P_{i-1}P_i}, \overrightarrow{P_iP_{i+1}})$ est l'angle défini en sens trigonométrique de deux segments consécutifs de polygone au point P_i et $l_i = \|\overrightarrow{P_iP_{i+1}}\|$, $i = 0, \ldots, N-2$, voir Figure 2.1. La représentation (θ, l) définit un polygone à une isométrie près (ie. à une rotation et une translation près). Les coordonnées (x, y) des points de contrôle P_i peuvent être retrouvées exactement en utilisant un point initial et une direction fixée comme segment initial (déterminant ainsi la translation et la rotation). Nous détaillons ci-dessous les formules de changement de coordonnées.

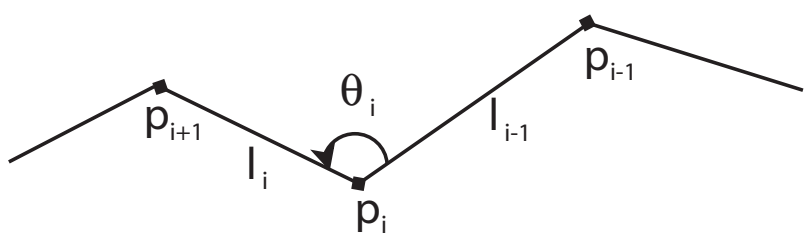


FIGURE 2.1 – Des coordonnées (x, y) aux coordonnées (θ, l) (l_i, θ_i en fonction de P_{i-1},P_i , P_{i+1})

Des coordonnées (x, y) aux coordonnées (l, θ)

$$\begin{cases} l_i &= \left\|\overrightarrow{P_iP_{i+1}}\right\| \\ \theta_i &= \arccos\left(\frac{\overrightarrow{P_{i-1}P_i}\cdot\overrightarrow{P_iP_{i+1}}}{\left\|\overrightarrow{P_{i-1}P_i}\right\|\left\|\overrightarrow{P_iP_{i+1}}\right\|}\right) \end{cases}, \textit{pour } i = 1,\ldots,N-2 \tag{2.1}$$

Pour l'initialisation (calcul de l_0 et θ_0), nous fixons une orientation de référence $\overrightarrow{P_{-1}P_0} = (-1,0)$.

Remarquons aussi qu'en passant en coordonnées (l,θ), nous perdons l'information de position du polygone dans l'espace. Pour être capable de retrouver exactement les coordonnées (x,y) d'un polygone à partir de ses coordonnées (l,θ), nous avons donc besoin de stocker une position de référence. Pour cela, nous stockons la position de P_0 : $(x_{init}, y_{init}) = (x_0, y_0)$.

Des coordonnées (l,θ) aux coordonnées (x,y)

Le premier sommet est égal au sommet initial stocké : $(x_0,y_0) = (x_{init}, y_{init})$. Nous utilisons aussi la direction de référence $\overrightarrow{P_{-1}P_0} = (-1,0)$ pour l'initialisation. Les coordonnées (x,y) des sommets suivants sont ensuite calculés de proche en proche, voir Figure 2.2 :

$$\begin{cases} x_i &= l_{i-1}\cos\left(\theta_0 + \sum_{k=1}^{i-1}(\theta_k - \pi)\right) + x_{i-1} \\ y_i &= l_{i-1}\sin\left(\theta_0 + \sum_{k=1}^{i-1}(\theta_k - \pi)\right) + y_{i-1} \end{cases}, \textit{pour } i = 1,\ldots,N-1 \tag{2.2}$$

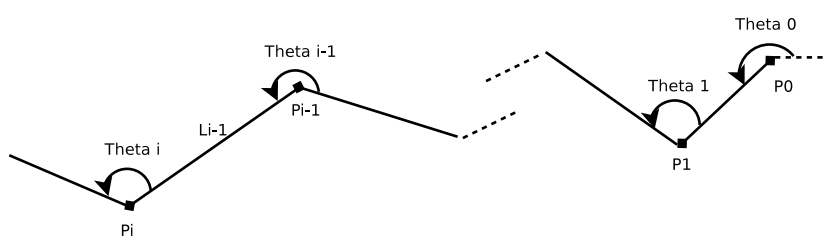


FIGURE 2.2 – Des coordonnées (l,θ) aux coordonnées (x,y) (P_i fonction de P_{i-1} et θ_0, ...,θ_{i-1})

Chapitre 3

Analyse Multiresolution basée sur la représentation angle-longueur

L'analyse multirésolution divise un vecteur de 2^n coefficients c^n en une partie grossière c^{n-1} comprenant 2^{n-1} coefficients et une partie de détail d^{n-1} comprenant elle aussi 2^{n-1} coefficients. Cette division peut-être appliquée récursivement au nouveau vecteur c^{n-1}. Ainsi le vecteur original des coefficients peut-être décomposé en une hiérarchie de vecteurs dits de basse-résolution $c^{n-1}, \ldots, c^0$ et de vecteurs de détails $d^{n-1}, \ldots, d^0$. Ce procédé est appelé *analyse ou décomposition*. Le vecteur des coefficients originaux peut-être retrouvé depuis n'importe quelle séquence $c^j, d^j, d^{j+1}, \ldots, d^n$, $j = 0, \ldots, n-1$. Ce procédé est appelé *synthèse ou reconstruction*.

Comme nous l'avons expliqué dans l'introduction, les déformations d'une représentation MR basée sur des ondelettes souffrent généralement d'un manque de réalisme car l'orientation des détails reste fixe lors de la déformation. C'est pourquoi nous introduisons dans cet article une autre méthode d'analyse MR pour des courbes linéaires par morceaux (polygones 2D). Dans le cadre applicatif du *morphing*, notre méthode possède deux propriétés importantes : l'orientation des détails suit la déformation naturellement (voir Figure 3.3) et la représentation MR est définie intrinsèquement.

Nous savons, grâce à la géometrie différentielle, qu'une courbe paramétrique est déterminée de manière unique, à isométrie près, par sa fonction courbure [Car76]. La courbure est une propriété intrinsèque d'une courbe. Dans le contexte des courbes planes et linéaires par morceaux, les angles et longueurs définissent intrinsèquement une courbe, à isométrie près [Ada75, SGWM93]. Nous allons utiliser ces quantités intrinsèques pour définir les coefficients grossiers et de détail d'une analyse MR.

La *représentation MR angle-longueur* que nous introduisons maintenant fonctionne exclusivement avec les coordonnées (l, θ). A l'instar des ondelettes *paresseuses* [Swe97], nous choisissons de ne garder que la moitié des points de contrôle à chaque étape d'analyse afin d'obtenir le polygone grossier.

3.1 L'Analyse

L'analyse (décomposition) est une procédure récursive consistant à diviser un vecteur de coefficients d'un polygone (θ^{n+1}, l^{n+1}) en un vecteur de coefficients grossiers (correspondant au polygone de basse-résolution) (θ^n, l^n) et en un vecteur de coefficients de détail. L'indice n dénote alors le niveau de résolution. A partir d'un polygone à 2^{n+1} segments, une étape d'analyse crée un polygone à 2^n segments et 2^n coefficients de détail, lesquels sont représentés par deux vecteurs bidimensionnels de la forme :

$$\begin{array}{ccc} (\theta^{n+1}, \mathbf{l}^{n+1}) & \rightarrow & (\theta^n, \mathbf{l}^n) \\ & \searrow & \\ & & (\alpha^n, \beta^n), \end{array}$$

où $\theta^n = (\theta_0^n, \ldots, \theta_{2^n-1}^n)$, de même pour $\mathbf{l}^n, \alpha^n, \beta^n$. A partir d'un polygone à D segments, $2^n \leq D < 2^{n+1}$, on peut donc effectuer jusqu'à n étapes d'analyse.

Il est important de noter que les deux types de coefficients (de détail et grossiers) au niveau n doivent être déduits uniquement de (θ^{n+1}, l^{n+1}) et vice-versa. Ainsi les coefficients d'un niveau de résolution ne dépendent que du niveau plus fin.

En utilisant les règles basiques de trigonométrie pour les triangles, nous déduisons les formules d'analyse, illustrées dans la Figure 3.1. A chaque étape de l'analyse, le polygone fin (θ^{n+1}, l^{n+1}) est décomposé en son approximation (θ^n, l^n), appelé polygone grossier, ainsi qu'un ensemble de détails (α^n, β^n) permettant la reconstruction du polygone fin (voir Partie 3.2). Les détails sont des angles.

Considérons la situation comme décrite dans la Figure 3.1. La longueur l_i^n est calculée en appliquant le théorème d'Al-Kashi dans le triangle $P_i^n P_{i+1}^n P_{2i+1}^{n+1}$. Ce théorème relie dans un triangle quelconque la longueur d'un côté à celles des deux autres et au cosinus de l'angle formé par ces deux côtés, généralisant ainsi le théorème de Pythagore aux triangles non rectangles. Nous connaissons un angle : $2\pi - \theta_{2i+1}^{n+1}$ et ses 2 côtés adjacents l_{2i}^{n+1} et l_{2i+1}^{n+1}. Notons que $\cos(2\pi - \theta_{2i+1}^{n+1}) = \cos(-\theta_{2i+1}^{n+1}) = \cos(\theta_{2i+1}^{n+1})$. Nous en déduisons le troisième côté du triangle $l_i^n = \sqrt{\left(l_{2i+1}^{n+1}\right)^2 + \left(l_{2i}^{n+1}\right)^2 - 2 \cdot l_{2i+1}^{n+1} \cdot l_{2i}^{n+1} \cos\left(\theta_{2i+1}^{n+1}\right)}$. D'autre part, nous avons de façon triviale $\theta_i^n = \beta_{i-1}^n + \theta_{2i}^{n+1} + \alpha_i^n$. En appliquant Al-Kashi dans le triangle $P_i^n P_{i+1}^n P_{2i+1}^{n+1}$, nous obtenons directement $\alpha_i^n = \arccos\left(\frac{\left(l_{2i}^{n+1}\right)^2 + \left(l_i^n\right)^2 - \left(l_{2i+1}^{n+1}\right)^2}{2\, l_i^n\, l_{2i}^{n+1}}\right)$, puis β_i^n est déduit en utilisant le fait que la somme des angles de ce triangle est égale à π : $\beta_i^n + \alpha_i^n + 2\pi - \theta_{2i+1}^{n+1} = \pi$, d'où $\beta_i^n = \theta_{2i+1}^{n+1} - \alpha_i^n - \pi$.

Au final, nous obtenons les résultats suivants :

Coefficients grossiers

$$\theta_i^n = \beta_{i-1}^n + \theta_{2i}^{n+1} + \alpha_i^n,$$

$$l_i^n = \sqrt{\left(l_{2i+1}^{n+1}\right)^2 + \left(l_{2i}^{n+1}\right)^2 - 2 \cdot l_{2i+1}^{n+1} \cdot l_{2i}^{n+1} \cos\left(\theta_{2i+1}^{n+1}\right)}, \tag{3.1}$$

$$\text{pour } \; i = 0, \ldots, 2^n - 1.$$

Coefficients de détail

$$\alpha_i^n = \arccos\left(\frac{\left(l_{2i}^{n+1}\right)^2 + \left(l_i^n\right)^2 - \left(l_{2i+1}^{n+1}\right)^2}{2\, l_i^n\, l_{2i}^{n+1}}\right)$$

$$\beta_i^n = \theta_{2i+1}^{n+1} - \alpha_i^n - \pi, \qquad \text{pour } \; i = 0, \ldots, 2^n - 1. \tag{3.2}$$

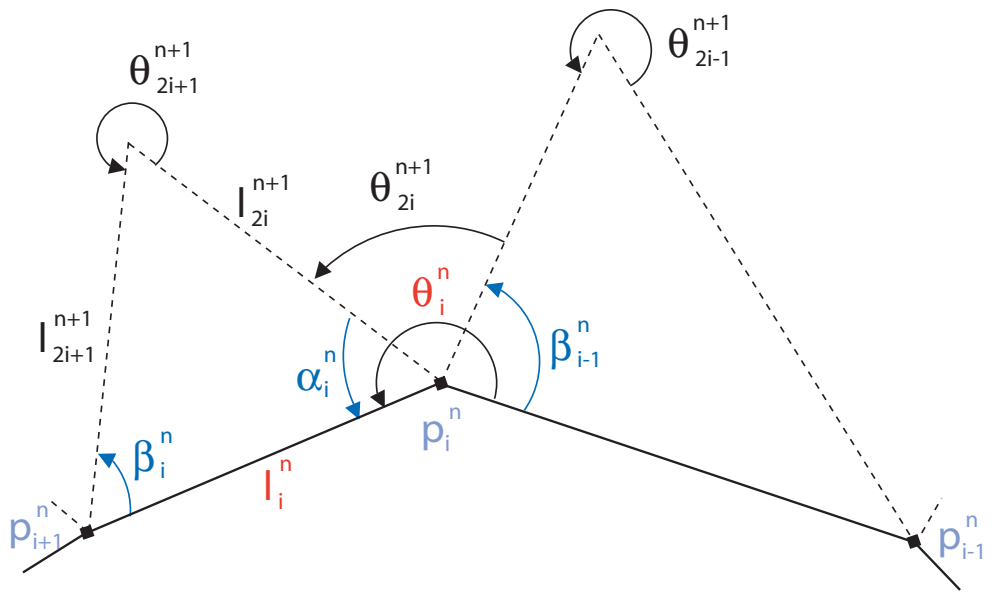


FIGURE 3.1 – L'analyse. Le polygone en pointillés appartient au niveau de résolution $n+1$, le polygone en traits pleins appartient au niveau n.

3.2 La Synthèse

L'analyse fournit un polygone grossier $(\theta^0, \mathbf{l}^0)$ (une forme géometrique simple) et un ensemble de coefficients de détail $(\alpha^0, \beta^0), \ldots, (\alpha^n, \beta^n)$. Le nombre de segments grossiers dépend du nombre d'étapes d'analyse. La synthèse doit reconstruire le polygone original. Nous travaillons encore en coordonnées (θ, l). Ainsi, l'algorithme consiste à calculer les longueurs l_{2i}^{n+1}, l_{2i+1}^{n+1} et les angles θ_{2i}^{n+1}, θ_{2i+1}^{n+1} au niveau $n+1$ uniquement grâce aux coefficients grossiers et de détail au niveau n. Ce procédé correspond à une étape de synthèse. Nous pouvons ensuite itérer l'algorithme sur la forme reconstruite avec les détails.

Les formules de synthèse, comme celles d'analyse, sont obtenues géométriquement (voir Figure 3.1 pour une illustration).

Pour exprimer les formules de reconstruction des longueurs, nous nous plaçons dans le triangle $P_i^n P_{i+1}^n P_{2i+1}^{n+1}$. Nous considérons la hauteur issue de P_{2i+1}^{n+1}, dont nous notons H la base, et h la longueur. Nous introduisons $l_1 = ||P_{i+1}^n H||$ et $l_2 = ||HP_i^n||$, avec $l_1 + l_2 = l_i^n$. Voir Figure 3.2 pour une illustration.

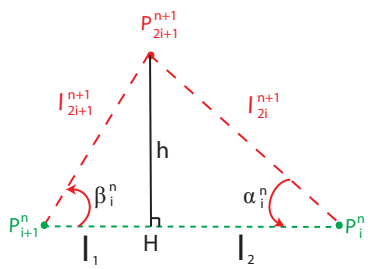


FIGURE 3.2 – Illustration pour le calcul de la reconstruction des longueurs.

Ces notations nous permettent d'écrire $\sin\beta_i^n = \frac{h}{l_{2i+1}^{n+1}}$, $\cos\beta_i^n = \frac{l_1}{l_{2i+1}^{n+1}}$, $\sin\alpha_i^n = \frac{h}{l_{2i}^{n+1}}$, $\cos\alpha_i^n = \frac{l_2}{l_{2i}^{n+1}}$. Ainsi,

$$\begin{aligned}
\sin(\alpha_i^n + \beta_i^n) &= \sin\alpha_i^n \cos\beta_i^n + \cos\alpha_i^n \sin\beta_i^n \\
&= \frac{h}{l_{2i}^{n+1}} \frac{l_1}{l_{2i+1}^{n+1}} + \frac{l_2}{l_{2i}^{n+1}} \frac{h}{l_{2i+1}^{n+1}} \\
&= \frac{(l_1+l_2)h}{l_{2i}^{n+1} l_{2i+1}^{n+1}} \\
&= \frac{l_i^n h}{l_{2i}^{n+1} l_{2i+1}^{n+1}}
\end{aligned}$$

Ainsi, $\frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n)} = \frac{l_{2i}^{n+1}}{l_i^n}$, d'où $l_{2i}^{n+1} = l_i^n \frac{\sin(\beta_i^n)}{\sin(\alpha_i^n+\beta_i^n)}$.
Et $\frac{\sin\alpha_i^n}{\sin(\alpha_i^n+\beta_i^n)} = \frac{l_{2i+1}^{n+1}}{l_i^n}$, d'où $l_{2i+1}^{n+1} = l_i^n \frac{\sin(\alpha_i^n)}{\sin(\alpha_i^n+\beta_i^n)}$.

Pour ce qui est de la reconstruction des angles fins, nous avons par définition de ces angles, et symétriquement à la formule d'analyse 3.1 $\theta_{2i}^{n+1} = \theta_i^n - \alpha_i^n - \beta_{i-1}^n$. De la même façon que dans la formule d'analyse pour calculer β_i^n 3.2, nous avons $\theta_{2i+1}^{n+1} = \pi + \alpha_i^n + \beta_i^n$.

Reconstruction des longueurs

$$\begin{aligned} l_{2i}^{n+1} &= l_i^n . \frac{\sin(\beta_i^n)}{\sin(\alpha_i^n + \beta_i^n)} \\ l_{2i+1}^{n+1} &= l_i^n . \frac{\sin(\alpha_i^n)}{\sin(\alpha_i^n + \beta_i^n)} . \end{aligned} \tag{3.3}$$

Reconstruction des angles

$$\begin{aligned} \theta_{2i}^{n+1} &= \theta_i^n - \alpha_i^n - \beta_{i-1}^n \\ \theta_{2i+1}^{n+1} &= \pi + \alpha_i^n + \beta_i^n . \end{aligned} \tag{3.4}$$

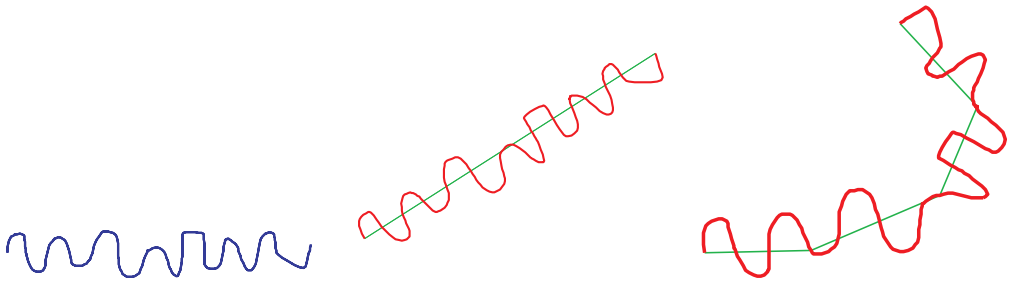


FIGURE 3.3 – A gauche : courbe initiale avec 256 points de contrôle. Au centre : déformation du polygone grossier au niveau 6. A droite : déformation du polygone grossier au niveau 8. L'orientation des détails suit naturellement l'orientation du polygone grossier lors de la déformation.

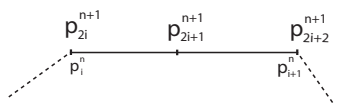
3.3 Les cas spéciaux

Trois cas spéciaux nécessitant un traitement particulier sont à prendre en considération. Ils correspondent aux cas où les coefficients de détail (α_i, β_i) prennent les valeurs $(0,0)$, $(0,\pi)$ et $(\pi,0)$. Géométriquement, ces cas correspondent à des triangles dégénérés i.e. où p_{2i}^{n+1}, p_{2i+1}^{n+1} et p_{2i+2}^{n+1} sont colinéaires. Lorsque nous stockons les valeurs des détails lors de l'analyse, nous ne pouvons pas retrouver les longueurs correctes l_{2i}^{n+1}, l_{2i+1}^{n+1} au moment de la synthèse car ces

valeurs reportées dans les formules précédentes (3.3) provoquent une division par 0, et donc $l_{n+1}^{2i} = \infty$ et $l_{n+1}^{2i+1} = \infty$.

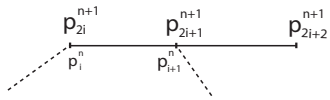
Ainsi à l'apparition d'un de ces cas, nous procédons de la manière suivante : β_i est assigné à une valeur spécifique au cas rencontré. Cette valeur nous permettra de savoir quel traitement appliquer lors de la synthèse, et nous assignons à α_i le rapport des longueurs caractéristiques du triangle plat. Ainsi la reconstruction peut être effectuée sans les aberrations numériques mentionnées précédemment.

Cas particulier $(\alpha_i = 0, \beta_i = 0)$



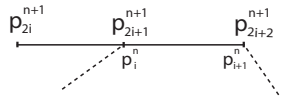
Lors de l'analyse, nous stockons le ratio des longueurs $r = \frac{l_{2i}^{n+1}}{l_{2i+1}^{n+1} + l_{2i}^{n+1}}$. Ainsi, au moment de la synthèse, nous pouvons retrouver les longueurs fines en utilisant ce ratio : $l_i^n = l_{2i+1}^{n+1} + l_{2i}^{n+1}$ $\Longrightarrow \quad l_{2i}^{n+1} = l_i^n \times r$ et $l_{2i+1}^{n+1} = l_i^n.(1-r)$

Cas particulier $(\alpha_i = 0, \beta_i = \pi)$



Lors de l'analyse, nous stockons le ratio $r = \frac{l_{2i}^{n+1}}{l_{2i}^{n+1} - l_{2i+1}^{n+1}}$. Puis nous l'utilisons au moment de la synthèse : $l_i^n = l_{2i}^{n+1} - l_{2i+1}^{n+1} \quad \Longrightarrow \quad l_{2i}^{n+1} = l_i^n \times r$ et $l_{2i+1}^{n+1} = l_i^n.(r-1)$.

Cas particulier $(\alpha_i = \pi, \beta_i = 0)$



Lors de l'analyse, nous stockons le ratio $r = \frac{l_{2i}^{n+1}}{l_{2i+1}^{n+1} - l_{2i}^{n+1}}$, que nous utilisons au moment de la synthèse : $l_i^n = l_{2i+1}^{n+1} - l_{2i}^{n+1} \quad \Longrightarrow \quad l_{2i}^{n+1} = l_i^n \times r$ et $l_{2i+1}^{n+1} = l_i^n.(1+r)$

Chapitre 4

Morphing multirésolution

En entrée, nous avons deux polygones P_S et P_T avec le même nombre de points de contrôle, appelés polygones source P_S et destination P_T. Nous cherchons à construire les polygones intermédiaires P_t qui transforment graduellement P_S en P_T, pour $t \in [0,1]$, où $P_S = P_0$ et $P_T = P_1$. Il existe de nombreux interpolants possibles pour P_t et il est particulièrement difficile de caractériser mathématiquement de ”bons” algorithmes de *morphing*, puisque le jugement dépend à la fois du domaine d'application et de la perception humaine de la géométrie.

L'algorithme que nous présentons dans cette partie fait partie de la classe des algorithmes basés sur des représentations intrinsèques du contour. Les polygones en entrée sont représentés par la représentation multirésolution intrinsèque que nous avons introduite dans le Chapitre 3. La nature multirésolution de cette représentation a l'avantage de générer les polygones intermédiaires directement à tous les niveaux de détails sans avoir à en calculer une représentation MR. Voici les caractéristiques de notre algorithme intrinsèque MR qui le rendent particulièrement approprié pour le *morphing* :

- L'orientation des détails suit la déformation naturellement.
- La représentation MR est définie intrinsèquement.
- Les formes intermédiaires interpolent les polygones source et destination de façon ”aussi-rigide-que-possible” comme dans [ACOL00].
- Les formes intermédiaires peuvent être visualisées à tous les niveaux de résolution grâce au *morphing* MR qui calcule séparément l'interpolation des coefficients grossiers et de détail.

4.1 Algorithme

Une fois la représentation MR de P_S et P_T calculée, nous disposons de deux ensembles de coefficients grossiers $(\theta_S^0, \mathbf{l}_S^0)$, $(\theta_T^0, \mathbf{l}_T^0)$ et de deux ensembles de coefficients de détail $(\alpha_S^0, \beta_S^0), \ldots,$ (α_S^n, β_S^n), et $(\alpha_T^0, \beta_T^0), \ldots, (\alpha_T^n, \beta_T^n)$. Les formes intermédiaires sont maintenant générées en interpolant les coefficients de cette représentation intrinsèque.

4.1.1 Interpolation des coefficients grossiers

A. Interpolation linéaire des coefficients grossiers

Dans le cas de polygones fins à $D \cdot 2^n$ sommets, les polygones grossiers obtenus après n étapes d'analyse ont exactement D sommets. Nous verrons dans la Partie 4.2 comment nous obtenons des courbes de $D \cdot 2^n$ sommets à partir d'une courbe avec un nombre quelconque de sommets. Nous choisissons D petit (3 ou 4 sommets dans nos exemples), afin que le polygone grossier obtenu représente la forme grossière du polygone donné en entrée, et que la plus grande partie de l'information de forme soit contenue dans les coefficients de détail. Ainsi, nous tirons pleinement parti de notre représentation multirésolution.

Nous avons donc 2 polygones grossiers P_S^0 et P_T^0 de 3 ou 4 sommets, à interpoler. Puisqu'ils sont très simples, et ont la même orientation (voir Partie 4.2), une simple interpolation linéaire de leurs coordonnées cartésienne pourrait dans la plupart des cas donner des résultats satisfaisants. Cependant, si nous avons une rotation entre ces deux polygones, bien que simples, nous risquons d'avoir des auto-intersections dans les formes intermédiaires. Nous préférons donc interpoler les coordonnées (l, θ) de ces polygones grossiers, ce qui revient à appliquer l'algorithme de morphing de Sederberg à nos polygones grossiers [SGWM93]. Nous interpolons donc linéairement les coefficients grossiers de notre représentation multirésolution :

$$\begin{aligned} l_t^j &= (1-t)l_S^j + t l_T^j, \\ \theta_t^j &= (1-t)\theta_S^j + t\theta_T^j, \end{aligned} \tag{4.1}$$

$$\text{pour} \quad 0 \leq j \leq n \quad \text{et} \quad t \in [0,1].$$

Dans le cas de courbes fermées, nous avons cependant le problème de non fermeture des courbes intermédiaires à traiter, comme décrit par Sedeberg [SGWM93]. La différence importante avec l'algorithme de Sederberg, et que nous ne rencontrons ce problème de non-fermeture que sur nos polygones grossiers, qui sont extrêmement simples. Et du fait de notre représentation

MR, appliquer une fermeture affine aux polygones grossiers assure une fermeture des polygones fins reconstruits.

B. Interpolation aussi-rigide-que-possible des coefficients grossiers

Dans le cas particulier de polygones fermés à $3 \cdot 2^n$ sommets, les polygones grossiers après n étapes d'analyse P_S^0 et P_T^0 sont des triangles. Dans ce cas, au lieu d'appliquer l'algorithme de Sederberg aux polygones grossiers, ce qui nécessite une étape supplémentaire de fermeture, nous proposons d'interpoler ces deux triangles avec l'interpolation "aussi-rigide-que-possible" d'Alexa [ACOL00], détaillée ci-dessous. Cette interpolation, en plus d'assurer la fermeture des polygones intermédiaires, assure un morphing rigide (le triangle ne s'écrase pas au cours du morphing). La transformation affine A entre P_S^0 et P_T^0 est décomposée en une matrice de rotation R_γ et une matrice symétrique S en utilisant la décomposition en valeurs singulières (SVD) : $A = R_\gamma S$. Les triangles intermédiaires sont ensuite obtenus en interpolant les composantes de cette décomposition : $A(t) = R_{t\gamma}((1-t)I + tS)$. Cette interpolation entre deux triangles est dite "aussi-rigide-que-possible" car les angles varient de manière pratiquement linéaire. Remarquons qu'interpoler linéairement les longueurs et angles représentant un polygone assure aussi une interpolation rigide. Ici c'est l'intérieur de la forme (le triangle) qui est pris en compte et non simplement son contour, ce qui permet un contôle plus fin de la rigidité de la forme.

Estimation du mouvement entre deux triangles Nous avons deux triangles source et destination. Nous souhaitons estimer le mouvement entre ces deux triangles afin de l'interpoler. Nous décrivons ici la méthode que nous utilisons, proposée par Alexa, Cohen-Or et Levin [ACOL00].

Soient les sommets du triangle source $P = (\overrightarrow{p_1}, \overrightarrow{p_2}, \overrightarrow{p_3})$, et ceux du triangles destination $Q = (\overrightarrow{q_1}, \overrightarrow{q_2}, \overrightarrow{q_3})$, où les sommets de même indice se correspondent. La transformation affine transformant P en Q est :

$$A\overrightarrow{p_i} + \overrightarrow{l} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \overrightarrow{p_i} + \begin{pmatrix} l_x \\ l_y \end{pmatrix} = \overrightarrow{q_i}, \quad i \in \{1,2,3\}$$

La translation $\overrightarrow{l}$ n'est pas prise en compte, puisque qu'elle décrit uniquement l'emplacement du triangle dans le plan, mais ne donne aucune information sur sa forme. Nous souhaitons plutôt définir les triangles intermédiaire en faisant varier les composantes affines contenues dans A.

Nous devons donc trouver la matrice A, qui estime le mouvement entre nos deux triangles. Pour la calculer, nous écrivons le système de façon à ce que a_1, a_2, a_3, a_4 en soit la solution.

Soient $\overrightarrow{p_i} = (x_i, y_i)$ et $\overrightarrow{q_i} = (x'_i, y'_i)$ les coordonnées cartésiennes des sommets des triangles, nous avons le système :

$$\begin{pmatrix} x_1 & y_1 & 0 & 0 \\ x_2 & y_2 & 0 & 0 \\ x_3 & y_3 & 0 & 0 \\ 0 & 0 & y_1 & x_1 \\ 0 & 0 & y_2 & x_2 \\ 0 & 0 & y_3 & x_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \\ y'_1 \\ y'_2 \\ y'_3 \end{pmatrix}$$

Pour résoudre ce système, nous utilisons la méthode des moindres carrés et une résolution de Cholesky. Le système précédent s'écrit :

$$Mx = b$$

Il se trouve qu'un vecteur x qui minimise $\|M \cdot x - b\|^2 = 0$ résoud aussi l'équation normale

$$M^t M x = M^t b$$

Nous choisissons de résoudre ce système avec la méthode de Cholesky : décomposition et résolution, ce qui nous donne $x = (a_1, a_2, a_3, a_4)^t$. Nous avons donc calculé la matrice A qui estime le mouvement entre nos deux triangles à interpoler. Reste à utiliser cette matrice A pour définir les triangles intermédiaires de notre interpolation. Les triangles intermédiaires $V = (\overrightarrow{v_1}(t), \overrightarrow{v_2}(t), \overrightarrow{v_3}(t))$ sont décrits comme $V(t) = A(t)P$, $t \in [0;1]$. Il nous reste à définir la façon de calculer$A(t)$.

Interpolation de la matrice d'estimation du mouvement La première solution envisagée par Alexa et al. est de simplement interpoler linéairement la matrice A pour obtenir $A(t)$:

$$A(t) = (1-t)I + tA$$

Cependant, des propriétés pour $A(t)$ sont souhaitables pour obtenir une interpolation plus élaborée de nos triangles : la transformation doit être symétrique selon t, les angles et échelle doivent changer linéairement, le triangle doit conserver son orientation (i.e. ne doit pas se refléter), la trajectoire des sommets obtenue doit être simple.

L'idée de base pour parvenir à ces propriétés est de décomposer A en matrices de rotations et

parties affines. La méthode retenue est basée sur la décomposition en valeurs singulières (SVD) de A :

$$A = U \cdot \Sigma \cdot V^t$$

où U et V sont unitaires, $U^t U = Id$ et $V^t V = Id$. La matrice U décrit les lignes de A selon les vecteurs de base associés aux valeurs singulières. La matrice V décrit les colonnes de A selon les vecteurs de base associés aux valeurs singulières. La matrice Σ contient les valeurs singulières.

Les auteurs apportent ensuite une amélioration en décomposant A en une seule rotation et une matrice symétrique (i.e. une décomposition polaire [SD92]). Par expérimentation, les auteurs ont trouvé que cette factorisation est celle qui donne visuellement les meilleurs résultats. Cette amélioration est déduite de la SVD :

$$\begin{aligned} A &= U\Sigma V^t = U\,(V^t V)\,\Sigma V^t \\ &= (UV^t)\,(V\Sigma V^t) \\ &= R_\gamma\, S \end{aligned}$$

En se basant sur cette décomposition, $A(t)$ est calculée en interpolant linéairement le paramètre γ de la factorisation. C'est cette interpolation que nous utilisons pour interpoler nos polygones grossiers dans le cas particulier où ils sont réduits à des triangles :

$$A(t) = R_{t\gamma}((1-t)I + tS)$$

avec les matrices de rotations usuelles $R_\gamma = \begin{pmatrix} cos\gamma & -sin\gamma \\ sin\gamma & cos\gamma \end{pmatrix}$ et $R_{t\gamma} = \begin{pmatrix} cos\gamma t & -sin\gamma t \\ sin\gamma t & cos\gamma t \end{pmatrix}$.

Au final, la Figure 4.1, extraite de [ACOL00], illustre l'intérêt de cette interpolation aussi-rigide-que-possible de triangles par rapport à une simple interpolation linéaire de leurs sommets. Le mouvement du triangle est bien plus rigide, les angles varient de façon proche de linéaire.

Interpolation aussi-rigide-que-possible de courbes Nous avons vu comment interpoler deux triangles, c'est cette interpolation que nous utilisons pour nos polygones grossiers quand ce sont des triangles. Voyons toutefois rapidement comment Alexa et al. utilisent leur interpolation de triangles pour interpoler des triangulations complètes. La première étape est de calculer des triangulations compatibles pour les polygones que nous souhaitons interpoler. Une fois le mouvement de chacun des triangles calculé comme décrit ci-dessus, les auteurs appliquent une optimisation pour que les triangulations intermédiaires soient bien des triangulations. Cette étape est indispen-

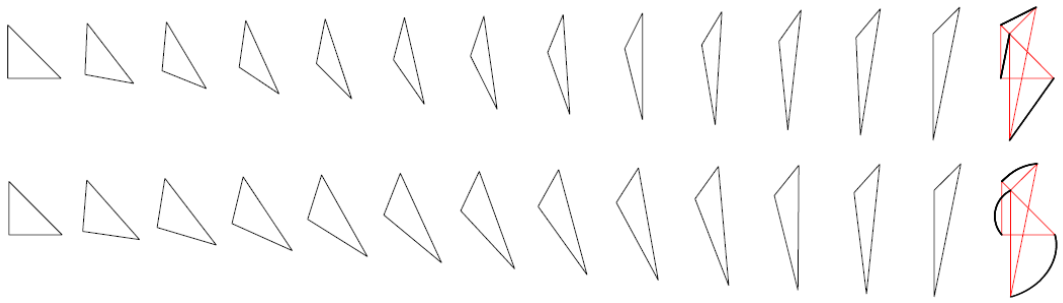


FIGURE 4.1 – En haut : interpolation linéaire des sommets des triangles. En bas : interpolation aussi-rigide-que-possible des triangles. La colonne de droite montre la trajectoire des sommets.

sable car les triangles sont interpolés séparément, et n'ont donc aucune raison de rester adjacents au cours de l'interpolation. Rappelons que dans le cas de l'interpolation de deux triangles, les étapes coûteuses décrites ici ne sont pas nécessaires.

C. Comparaison de nos deux méthodes d'interpolation de triangles grossiers

Dans le cas où nos polygones grossiers sont réduits à des triangles, comparons l'utilisation de l'une ou l'autre de nos méthodes d'interpolation. Rappelons que la première méthode, dite aussi-rigide-que-possible utilise une estimation du mouvement entre les 2 triangles, voir la Figure 4.2. La seconde méthode interpole linéairement les coefficients grossiers (l, θ), voir la Figure 4.3. Dans ces deux figures, les triangles grossiers source et destination se trouvent repectivement en début de première ligne, et en fin de seconde ligne. Notons que les triangles utilisés pour cette comparaison sont les triangles grossiers associés au hérisson et à l'hippocampe de la Figure 4.5.

Nous constatons comme attendu que ces deux méthodes donnent de bien meilleurs résultats qu'une simple interpolation linéaire des coordonnées cartésiennes des sommets. Ensuite, il est difficile de départager ces deux méthodes, elles donnent toutes deux un comportement rigide au morphing, leurs résultats sont similaires.

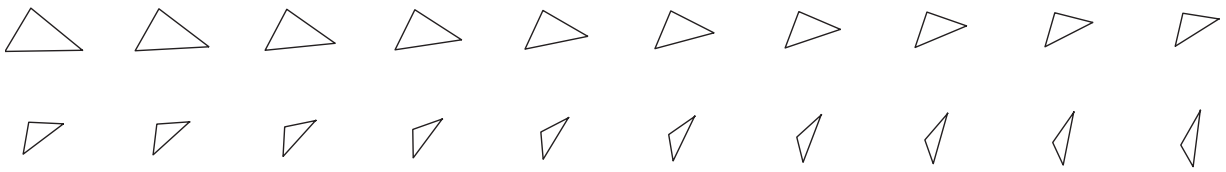


FIGURE 4.2 – Interpolation aussi-rigide-que-possible des triangles grossiers

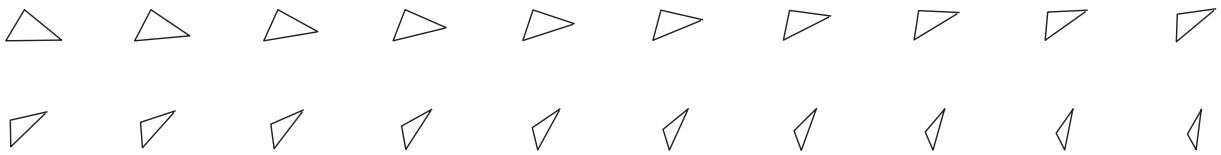


FIGURE 4.3 – Interpolation linéaire des coefficients grossiers

4.1.2 Interpolation des coefficients de détail

Les coefficients de détail représentent intrinsèquement les triangles qui sont ajoutés au polygone grossier pour l'amener à un niveau de résolution plus élevé. Chaque couple d'angles (α, β) représente un triangle. L'intérêt d'avoir choisi de représenter les détails par des angles est qu'il suffit maintenant de les interpoler linéairement pour obtenir un comportement rigide de notre interpolation. Les coefficients de détail intermédiaires de niveau j sont obtenus par interpolation linéaire :

$$\begin{aligned} \alpha_t^j &= (1-t)\alpha_S^j + t\alpha_T^j, \\ \beta_t^j &= (1-t)\beta_S^j + t\beta_T^j, \end{aligned} \tag{4.2}$$

$$\text{pour} \quad 0 \leq j \leq n \quad \text{et} \quad t \in [0,1].$$

L'interpolation peut générer des cas particuliers, correspondant à des triangles dégénérés. Nous procédons alors comme décrit dans la Partie 3.3 pour permettre la recontruction au moment de la synthèse.

Les formes intermédiaires sont les déformations "aussi-rigides-que-possible" des formes source et destination. En effet, la reconstruction de toutes les métamorphoses intermédiaires se fait par l'ajout de triangles aux polygones de plus basse résolution à chaque niveau de résolution. De plus, ces triangles sont obtenus en interpolant linéairement les angles, voir l'équation 4.2. Notre méthode peut donc être considérée comme un *morphing* "aussi-rigide-que-possible" de courbes multirésolutions, selon la définition donnée par Alexa.

Un autre intérêt de notre méthode est que nous obtenons la représentation MR complète de toutes les courbes intermédiaires ce qui nous permet de les visualiser à différents niveaux de résolution.

4.2 Résultats et discussion

Contrairement à la plupart des algorithmes de *morphing* existants, notre méthode est conçue pour les polygones de très grande taille avec beaucoup de détails locaux. Cependant, elle peut aussi bien traiter tous les autres polygones simples et lisses. Par conséquent nous avons seulement choisi des exemples comportant de nombreux détails fins, qui sont difficiles à métamorphoser.

Dans cette section, nous commençons par expliquer notre façon de préparer les données en vue du morphing, puis nous présentons différents résultats illustrant les propriétés de notre méthode de morphing, en la comparant à d'autres approches.

4.2.1 Préparation des données

Nos données sont des polygones 2D fermés ou ouverts. L'application de notre algorithme de morphing multirésolution induit différentes contraintes sur les données utilisées. Tout d'abord, en vue de faire du morphing, nous avons besoin de deux polygones comportant exactement le même nombre de sommets. Ensuite, notre algorithme utilise une représentation multirésolution, dans laquelle l'étape de simplification consiste à ne garder qu'un sommet sur deux du polygone fin. La taille de notre polygone est divisée par deux à chaque étape d'analyse. Il est donc plus commode d'avoir un nombre de sommets initial qui soit une puissance de deux (2^p), ou bien un multiple d'une puissance de deux ($D.2^p$). Enfin, il nous faut traiter le premier problème du morphing, la correspondance des points, et fixer une correspondance point à point de nos polygones avant de leur appliquer notre algorithme de morphing qui calcule la trajectoire des sommets. Nous décrivons ici ces différentes étape de préparation des données.

Création des polygones à partir d'images

La plupart de nos exemples ont été produits en extrayant le contour d'une image (tous sauf ceux de la Figure 4.9 qui ont été construits mathématiquement). A partir de l'image d'une forme fermée (voir par exemple la Figure 4.4), nous souhaitons extraite un polygone représentant son contour, en ayant la possibilité de choisir le nombre de points de ce polygone, et en assurant une répartition régulière des sommets. Nous commençons par passer l'image en binaire : blanc sur fond noir (avec l'intérieur plein). Nous extrayons ensuite le contour d'épaisseur 1 pixel de la partie blanche de notre image, ce qui nous donne un ensemble de M pixels. Les coordonnées $X(i), Y(i)$ des M points sont ensuite lissées en utilisant une spline de lissage. A chaque point $(X(i), Y(i))$ du bord est associé le paramètre $t(i) = i$, pour $i = 0, \ldots, M-1$. Pour obtenir un contour final avec N points, nous choisissons N noeuds $t'(k)$, pour $k = 0, \ldots, N-1$, équirépartis

dans $[0, M-1]$, et nous calculons la sline cubique de lissage correspondante. En général, nous choisissons $N = 2^p$. Ce programme a été développé dans l'équipe par Nicolas Szafran.

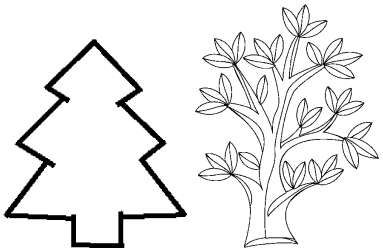


FIGURE 4.4 – Images originales du sapin et de l'arbre à fleur, dont sont extraits les polygones de la Figure 4.10.

Correspondance des points

La correspondance des points influence la qualité du *morphing*, mais comme nous l'avons précisé dans l'introduction, c'est un problème à part entière que nous ne prétendons pas traiter ici. Nous décrivons néanmoins la méthode que nous utilisons pour créer nos exemples.

Pour le couplage des sommets, nous définissons manuellement un point de départ pour le polygone source et destination. Tous les sommets restants correspondent alors automatiquement par leurs indices. Afin d'éviter des auto-intersections des *morphings*, les sommets doivent être orientés dans le même sens, dans le sens trigonométrique par exemple. De plus, nous avons constaté que les sommets de départ doivent soit correspondre à un caractère particulier des courbes, soit se correspondre géographiquement, par exemple être les sommets d'abscisses minimales. Dans la Figure 4.5 nous choisissons le nez des animaux comme point de départ.

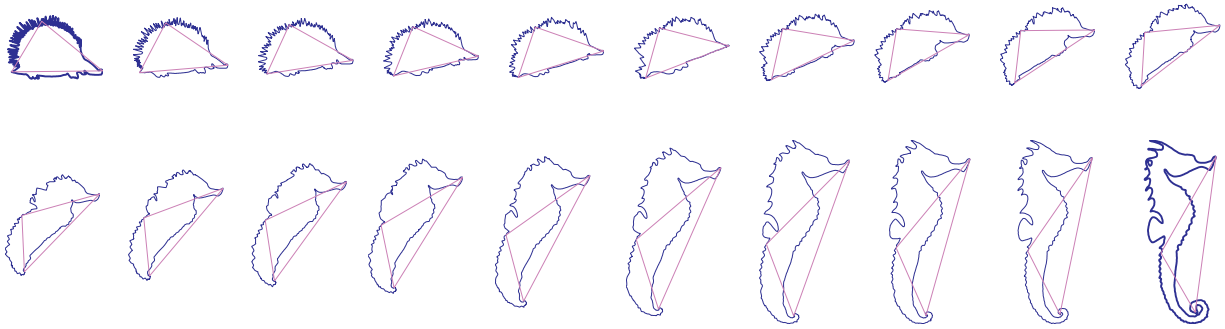


FIGURE 4.5 – Hérisson - Hippocampe avec un appariement bien choisi (1536 points)

Le choix de plus d'un sommet à coupler est possible avec notre méthode. Ceci permet à l'utilisateur de mettre en correspondance les caractéristiques des formes qui lui semblent les plus pertinentes, en fonction du résultat qu'il vise, en sélectionnant les sommets du polygone grossier de la façon suivante. Une analyse multiresolution est tout d'abord exécutée selon les algorithmes décrits dans la Section 4.1. Puisque notre analyse MR est basée sur le sous-échantillonage (nous conservons un sommet sur deux à chaque étape de la décomposition), les sommets du polygone grossier sont aussi des sommets du polygone fin et sont séparés par exactement $2^n - 1$ sommets. Ces relations particulières entre les polygones fins et grossiers peuvent maintenant être utilisées pour fixer manuellement tous les sommets des courbes source et destination grossières en tant que points de couplage pour le *morphing*.

L'utilisateur choisit manuellement trois indices de sommets à mettre en correspondance : i_0^S, i_1^S, i_2^S pour le polygone source, i_0^T, i_1^T, i_2^T pour le polygone destination, avec $i_0^S = i_0^T = 0$. Dans notre exemple avec le hérisson (source) et l'hippocampe (destination), nous avons choisi i_0^S et i_0^T les indices correspondants aux nez des animaux, i_1^S et i_1^T ceux du milieu de leur dos, i_2^S et i_2^T ceux du bout de leur queue. Nous souhaitons ensuite rééchantillonner nos polygones fins, de façon à ce que leurs polygones grossiers soient constitués de ces sommets choisis pour être mis en correspondance. Les polygones fins sont rééchantillonés de sorte qu'une puissance de deux du nombre de sommets se trouvent entre eux.

Pour rééchantillonner, nous pourrions à la fois ajouter et supprimer des sommets, afin de modifier les indices de nos sommets fixés. Cependant, afin de respecter la forme de notre polygone initial, nous choisissons de ne pas supprimer de sommets. Nous pouvons donc juste ajouter des sommets pour modifier les indices.

Dans l'exemple du hérisson et de l'hippocampe, nos données originales sont des polygones de 512 points. Pour fixer nos 2 points de correspondance (le milieu du dos, et le bout de la queue), nous ajoutons 1024 points aux polygones (les polygones produits ont donc 2056 sommets). Nous ajoutons des points avant, aprés et entre ceux d'indices i_1^S et i_2^S (resp. i_1^T et i_2^T), de façon à ce que les sommets d'indices i_1^S et i_2^S (resp. i_1^T et i_2^T) du polygone original, soient désormais les sommets d'indices 512 et 1024 du nouveau polygone à 2056 points. Les polygones ainsi générés ont $2056 = 3 \times 512$ sommets. Rappelons que pour un polygone fin de 3×2^n sommets, le polygone grossier après n étapes d'analyse est un triangle (composé des points d'indices 0, 512 et 1024 du polygone fin). Donc pour nos polygones hérisson et hippocampe à $2056 = 3 \times 512$ sommets, les polyonges grossiers aprés 9 étapes d'analyse sont des triangles, dont les sommets sont ceux que nous avons choisi de mettre en correspondance.

Notons que le même principe peut être appliqué pour fixer plus de 3 points de correspon-

dance, en travaillant avec des polygones grossiers de plus de 3 sommets.

Les nouveaux points sont ajoutés entre les anciens de façon à respecter la forme originale. Pour cela, nous appliquons une interpolation quadratique d'Hermite plutôt qu'une simple interpolation linéaire. La Figure 4.6 illustre ce choix. L'interpolation linéaire aurait de plus le désavantage de créer de nombreux cas particuliers pour notre représentation longueurs-angles (voir Section 3.3), en introduisant des points parfaitement alignés.

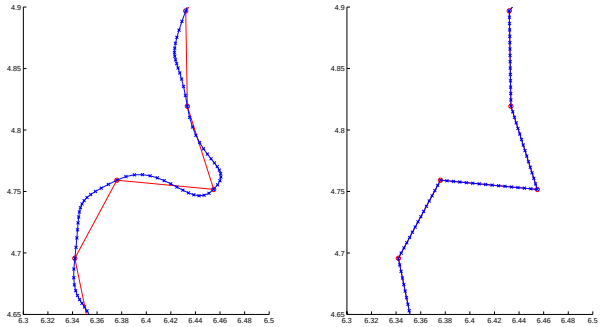


FIGURE 4.6 – Les nouveaux points (-x-) sont ajoutés entre les anciens points (-o-) en utilisant l'interpolation d'Hermite à gauche, l'interpolation linéaire à droite.

C'est ainsi que nous procédons pour apparier les formes dans les exemples présentés dans les Figures 4.5, 4.10 et 4.12. Pour les autres Figures 4.8 et 4.9, nous avons simplement fixé le sommet d'abscisse minimale comme point de départ.

Remarquons que dans le cas d'une partie très saillante et même enroulée sur elle-même comme la queue de l'hippocampe, il est quasiment impossible d'éviter les auto-intersections si nous ne choisissons pas l'extrémité de la queue comme point d'appariement, voir la Figure 4.7 pour une illustration. Cet exemple illustre bien l'importance du choix de couplage des sommets pour l'obtention d'un bon morphing.

FIGURE 4.7 – Hérisson - Hippocampe avec un appariement des sommets mal choisi (1024 points). Observez les auto-intersections au niveau de la queue des animaux.

Maintenant que nous avons vu comment sont préparées nos données, et comment nous procédons pour apparier nos polygones, nous allons détailler les résultats de notre algorithme de morphing multirésolution à travers différents exemples. La Figure 4.5 présentée plus haut illustre bien la conservation des détails au cours du morphing, avec des exemples comportants de nombreux détails fins, cette importante propriété de notre algorithme est vérifiable dans tous les exemples que nous présentons.

4.2.2 Représentation multirésolution de toutes les courbes

L'algorithme de *morphing* interpole les représentations intrinsèques du polygone grossier et les coefficients de détail. Nous obtenons donc en sortie une représentation multirésolution complète de chaque courbe intermédiaire. Ainsi, toutes les manipulations des niveaux de détails comme l'affichage, la compression, l'édition, ... peuvent être appliquées directement aux métamorphoses sans aucun calcul supplémentaire.

La Figure 4.8 montre sur la première ligne le *morphing* MR entre les cartes de France et d'Allemagne ainsi que les *morphings* des polygones grossiers. Les courbes de résolution complète (8192 sommets) sont obtenues par une synthèse complète (Partie 3.2). Les quatre autres lignes correspondent aux métamorphoses de résolutions inférieures obtenues par une synthèse partielle.

L'utilisateur peut en fait contrôler deux paramètres de l'interpolation. Il peut choisir d'interpoler une représentation plus ou moins grossière des polygones (voir Figure 4.12). Il peut aussi choisir une synthèse complète ou partielle pour visualiser les formes intermédiaires.

4.2.3 Morphing MR aussi-rigide-que-possible

Nous avons pu observer dans tous nos tests que les métamorphoses ont un comportement naturel et que les transformations sont de distorsion minimale. En effet, la manière dont nous interpolons les polygones grossiers et les coefficients de détail rend notre méthode d'interpolation de formes "aussi-rigide-que-possible". Pour l'interpolation des polygones grossiers réduits à des triangles, nous utilisons la méthode de Alexa et al. [ACOL00] qui calcule l'interpolation de distorsion minimale par optimisation. Cette méthode, appliquée aux polygones fins, exigerait le calcul de triangulations compatibles pour les polygones source et destination, puis une étape d'optimisation après le calcul des trajectoires individuelles de chaque triangle. Ces deux étapes coûteuses sont omises dans notre cas, puisque nous l'appliquons aux polygones grossiers qui sont réduits à des triangles. Nous utilisons uniquement l'interpolation aussi-rigide-que-possible entre deux

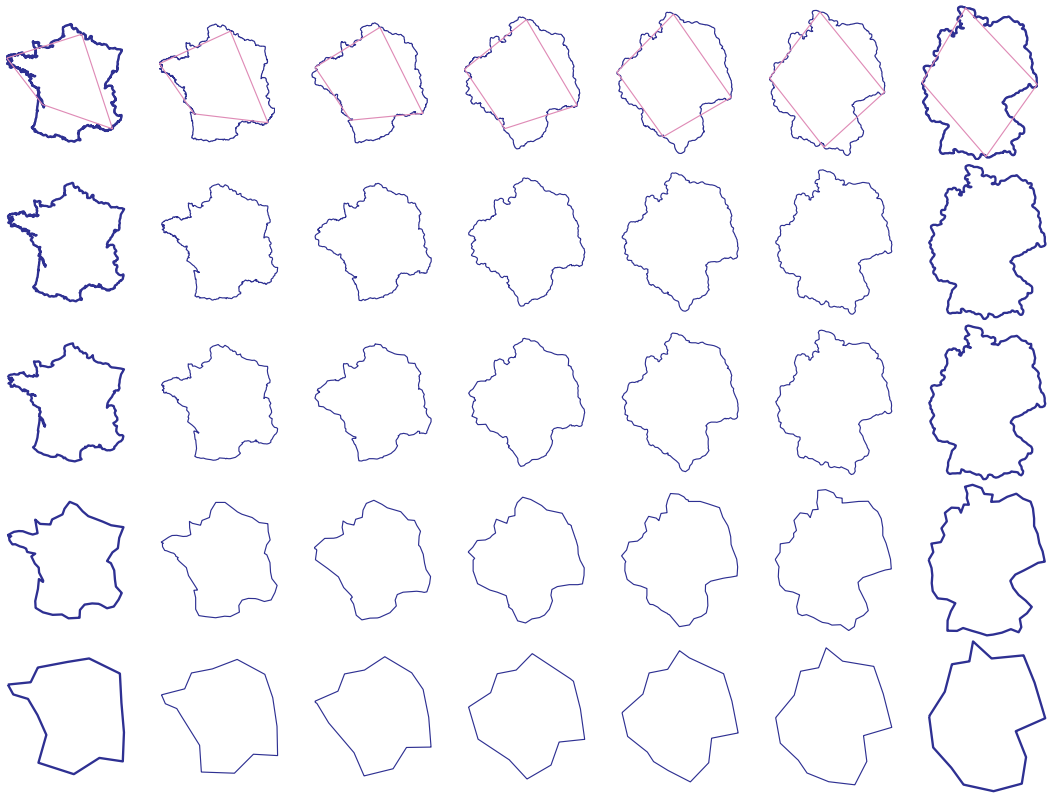


FIGURE 4.8 – France - Allemagne (8192 points) à 5 niveaux de résolution n=13 (niveau le plus fin), 10, 8, 6, 4

triangles grossiers. Dans le cas de polygone grossiers de plus de trois sommets, nous interpolons linéairement leurs longueurs et angles, ce qui leur confère aussi un comportement rigide. En outre, les détails sont transformés en interpolant linéairement leurs représentations intrinsèques. Ici aussi, l'interpolation est de distorsion minimale, puisque les angles varient linéairement. Rappelons que l'essentiel de l'information de la forme de départ est contenu dans les détails, la qualité de notre morphing repose donc principalement sur l'interpolation de ces coefficients de détail. Ces coefficients étant des angles que nous interpolons linéairement, nous assurons automatiquement à notre morphing un comportement rigide. En résumé, notre méthode de *morphing* MR se comporte de façon "aussi-rigide-que-possible". Toutes les figures témoignent de cette importante propriété. En particulier, une comparaison entre le *morphing* MR et le *morphing* intrinsèque classique [SGWM93] représentée sur la Figure 4.9 soutient clairement notre affirmation. Une simple interpolation linéaire des longueurs et angles (ligne du bas) entraîne une grosse variation de volume dans les courbes intermédiaires. Avec le morphing intrinsèque MR (ligne du haut), les courbes intermédiaires conservent un volume comparable à celui des courbes source et destination.

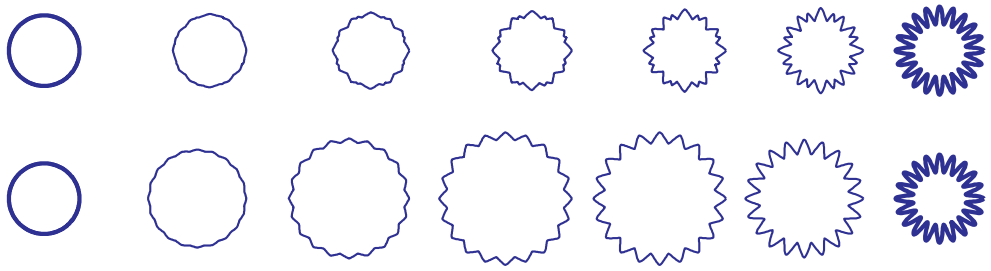


FIGURE 4.9 – Comparaison *morphing* MR (ligne du haut) - *morphing* intrinsèque sans MR (513 points) (ligne du bas)

4.2.4 Un exemple délicat

Bien que la méthode n'empêche pas explicitement les auto-intersections, il est néanmoins rare d'en observer. En effet, la manière dont nous interpolons les polygones grossiers garantit des transformations sans auto-intersection, mais seulement pour les polygones grossiers. Cependant, pour tester la robustesse de notre méthode sur cet aspect, nous employons un exemple très délicat représenté sur la Figure 4.10. Aucune auto-intersection ne se produit, contrairement à un *morphing* intrinsèque sans multirésolution, voir la Figure 4.11.

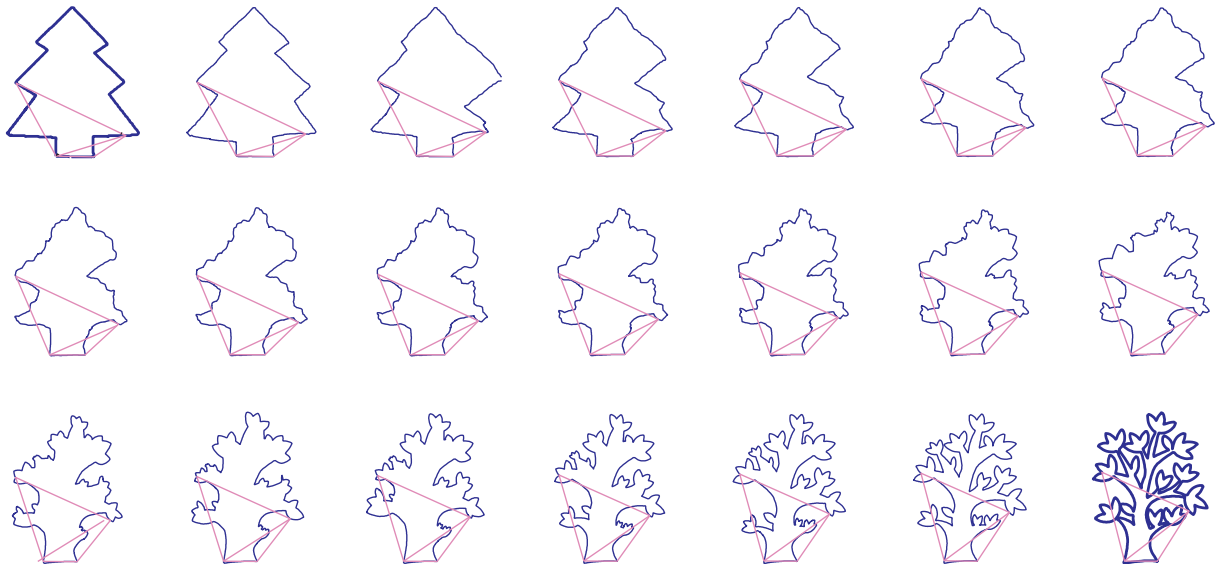


FIGURE 4.10 – Sapin de Noël - Arbre à fleurs (2048 points)

Un autre problème du *morphing* intrinsèque est la non-fermeture des polygones intermédiaires. Dans [SGWM93], il a été énoncé que les polygones intermédiaires sont toujours proches. Ceci pourrait être vrai pour des *morphings* entre des formes similaires, comme illustré dans [SGWM93]. Mais dans les cas où les formes sont très différentes et compliquées avec beaucoup de détails, la non-fermeture est un réel problème puisque de grandes parties des métamorphoses doivent être corrigées, voir la Figure 4.11. Notre méthode ne présente pas ce problème, puisque les polygones au niveau le plus grossier sont interpolés selon [ACOL00] lorque ce sont de simples triangles. Dans le cas où les polygones grossiers ont plus de 3 sommets, comme ici, nous leur appliquons la méthode d'interpolation de Sederberg, suivie d'une fermeture affine. Cependant, puisque la fermeture est appliquée au polygone grossier et fin, nous observons de bien meilleurs résultats, et bien moins de problèmes d'auto-intersections, bien que nous n'en garantissons pas l'absence. Rappelons que la majorité de l'information est contenue dans les coefficients de détail, qui sont des angles que nous interpolons linéairement, et non dans les coefficients grossiers.

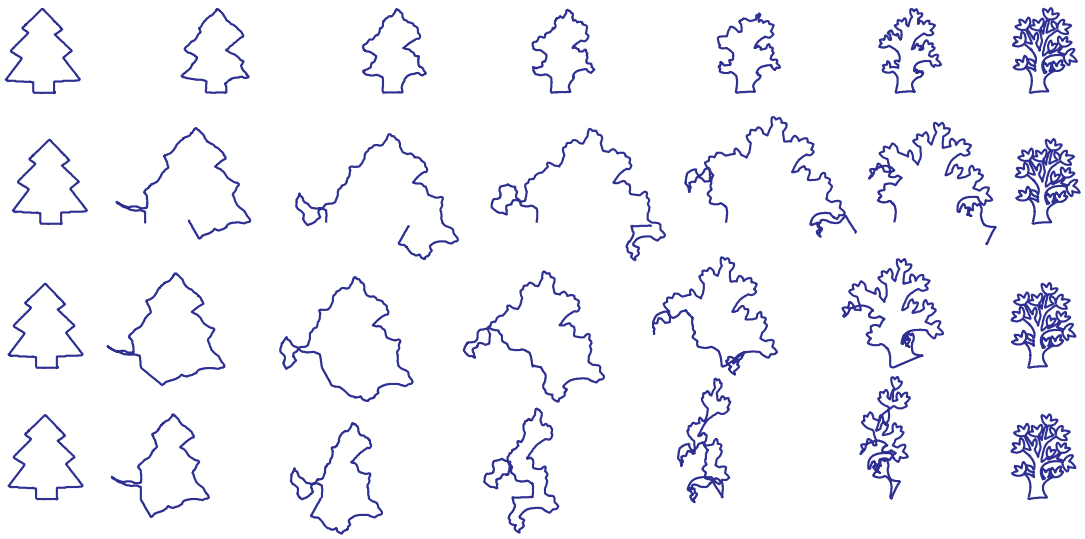


FIGURE 4.11 – Comparaison de 4 méthodes de *morphing* intrinsèque. Première ligne : *morphing* MR. Deuxième ligne : *morphing* intrinsèque des coordonnées (θ, l) des polygones. Les polygones des deux dernières lignes sont obtenus en fermant les polygones de la ligne deux de deux façons différentes. Troisième ligne : fermeture affine. Quatrième ligne : 'mise au point des arêtes' selon [SGWM93].

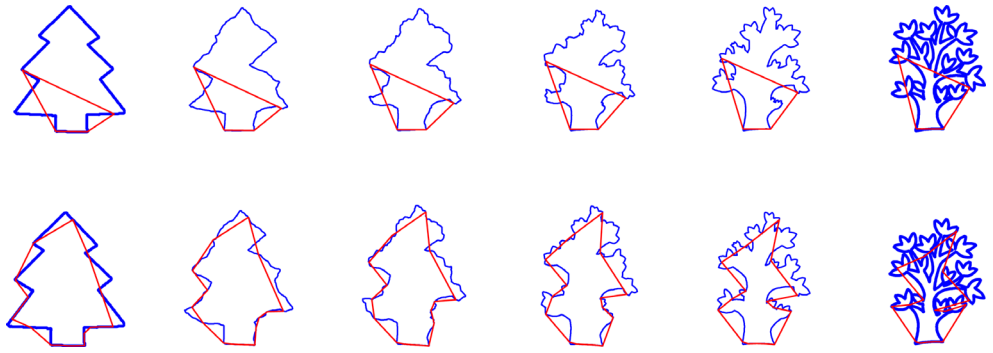


FIGURE 4.12 – Comparaison du morphing MR avec un polygone grossier à 4 et 16 sommets respectivement (polygones de 2048 points).

Chapitre 5

Etude de la stabilité de notre représentation intrinsèque MR

Ce chapitre étudie la stabilité de notre représentation MR de courbes, c'est à dire la sensibilité de notre représentation à l'introduction d'une erreur dans un de ses coefficients. Il est légitime de poser cette question car avec notre représentation, les sommets sont définis de proche en proche, par des longueurs et angles, ce qui implique une propagation des erreurs sur un sommet à tous les sommets suivant.

5.1 Stabilité liée au changement de coordonnées

Dans notre algorithme de morphing MR, nous travaillons uniquement en coordonnées (l, θ), qui sont aussi celles que Sedeberg interpole linéairement dans sa méthode de morphing [SGWM93]. Ces coordonnées ont la particularité de définir la position des sommets de proche en proche. Ainsi, la position d'un sommet P_i dépend de l'ensemble des (l_k, θ_k), $k = 0, \ldots, i-1$. Du fait de cette méthode itérative, une erreur introduite dans une longueur ou un angle se propage à tous les sommets suivants. En travaillant uniquement en coordonnées cartésiennes, ce problème n'est pas rencontré, puisque qu'une erreur sur un sommet n'influe pas les sommets voisins.

Nous étudions ici la propagation d'une erreur sur une longueur ou un angle à l'ensemble du polygone. Nous noterons P'_k le k-ème sommet reconstruit à partir des données erronnées. L'erreur introduite sera notée ε, elle sera considérée petite.

Erreur sur l Nous étudions l'influence d'une erreur commise sur une longueur. Soit $l'_i = l_i + \varepsilon$, toutes les autres données restent inchangées : $l'_k = l_k \ \forall k \neq i$ et $\theta'_k = \theta_k \ \forall k$.

Les sommets du polygone situés avant l'erreur commise en i sont retrouvés exactement,

$P'_k = P_k$, $\forall k \leq i$. Par contre, les sommets suivants comportent tous une erreur. Nous avons $||\overrightarrow{P_iP'_i}|| = \varepsilon$, voir Figure 5.1. Ensuite, puisque les angles ne sont pas modifiés, nous avons $\overrightarrow{P_{k-1}P_k}$ colinéaire à $\overrightarrow{P'_{k-1}P'_k}$, $\forall k \geq i$. Seule la longueur l_i est modifiée, les autres longueurs restent inchangées, donc $||\overrightarrow{P_{k-1}P_k}|| = ||\overrightarrow{P'_{k-1}P'_k}|| = l_{k-1}$, $\forall k \geq i$. Au final, nous avons donc $\overrightarrow{P_{k-1}P_k} = \overrightarrow{P'_{k-1}P'_k}$, ce qui nous permet de déduire que $\overrightarrow{P_kP'_k} = \overrightarrow{P_{k-1}P'_{k-1}} = \ldots = \overrightarrow{P_{i+1}P'_{i+1}}$. D'où aussi $||\overrightarrow{P_kP'_k}|| = ||\overrightarrow{P_{k-1}P'_{k-1}}|| = \ldots = ||\overrightarrow{P_iP'_i}|| = \varepsilon$. Ainsi, nous avons $||\overrightarrow{P_kP'_k}|| = \varepsilon$, $\forall k > i$. La Figure 5.3 (milieu) illustre ce résultat, la modification d'une longueur translate de ε tous les sommets suivants le segment modifié.

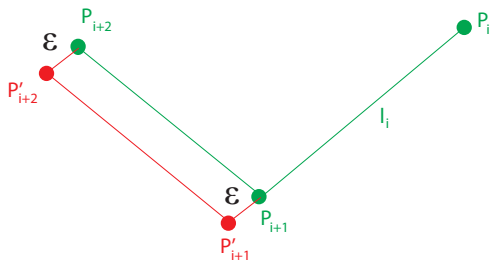


FIGURE 5.1 – Erreur sur l.

Erreur sur θ Etudions maintenant le propagation d'une erreur commise sur un angle. Soit $\theta'_i = \theta_i + \varepsilon$, toutes les autres données restent inchangées : $\theta'_k = \theta_k$ $\forall k \neq i$ et $l'_k = l_k$ $\forall k$.

Là encore, les sommets situés avant P_i sont recalculés de façon exacte, $P'_k = P_k$, $\forall k \leq i$. Faire $\theta'_i = \theta_i + \varepsilon$ revient à appliquer au polygone $P_i \ \ldots \ P_{N-1}$ une rotation de centre P_i et d'angle ε. Aussi, les positions de tous les sommets $P_{i+1}, \ldots, P_{N-1}$ sont modifiées. Et la norme de $||\overrightarrow{P_kP'_k}||$, avec $k > i$, va être en $O(C_k\ \varepsilon)$ avec la constante C_k fonction de toutes les longueurs $l_{i+1}, \ldots, l_k$. C_k sera de plus en plus élevée quand la distance de P_k au centre de la rotation, P_i, sera importante. $C_{i+1} = l_{i+1}$, $C_{i+2} = \sqrt{{l_{i+1}}^2 + {l_{i+2}}^2 - 2l_{i+1}l_{i+2}\cos\theta_{i+2}}$, ... Notons que, bien que les positions des P'_k sont modifiées, la forme du polygone $P'_i \ \ldots \ P'_{N-1}$ reste la même que celle de $P_i \ \ldots \ P_{N-1}$, puisque nous appliquons une rotation qui est une transformation rigide. Ceci est illustré par la Figure 5.3 (droite), la modification d'un angle provoque une rotation d'angle ε de tous les sommets suivants l'angle modifié.

Conclusion Comme attendu, la représentation (l, θ) n'est pas très stable. La modification d'un seul de ses coefficients peut avoir des répercutions importantes sur tout le polygone. C'est principalement la modification d'un angle qui peut provoquer de grosses erreurs, puisque une erreur de ε sur un angle provoque des erreurs non bornées sur les sommets suivants, dépendantes des

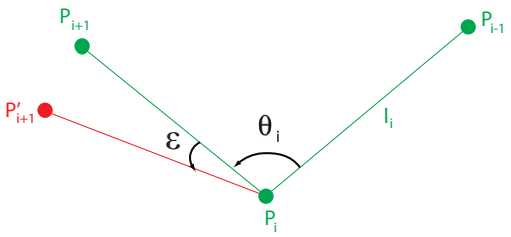


FIGURE 5.2 – Erreur sur θ.

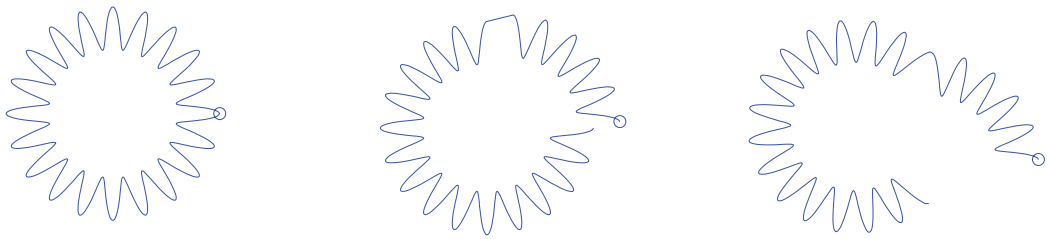


FIGURE 5.3 – A gauche, le polygone reconstruit sans erreur (512 sommets). P_0 est marqué par un cercle. Au milieu : avec une erreur sur une longueur $l'_{128} = l_{128} + 5\, l_{max}$. A droite : avec une erreur sur un angle $\theta'_{128} = 0.8\, \theta_{128}$.

longueurs. Nous avons pû constater en pratique cette instabilité dans le cas de l'utilisation de ces coordonnées pour le morphing (voir Figure 4.11).

5.2 Stabilité liée à l'analyse MR intrinsèque

Nous étudions ici la propagation d'une erreur sur un des coefficients de la représentation MR. Une erreur sur un coefficient grossier ou de détail va s'étendre à plusieurs coefficients fins, voir la Figure 3.1 pour un rappel des liens entre coefficients fins, grossiers, et de détail. La modification d'un coefficient grossier, i.e. d'une longueur l_i^n ou d'un angle θ_i^n du polygone grossier, va entraîner des modifications sur toutes les longueurs et angles reconstruits à partir de ces coefficients grossiers. La modification d'un coefficient de détail α_i^n ou β_i^n va elle aussi modifier les longueurs et angles qu'ils servent à reconstruire. Ainsi, la modification de l_i^n, θ_i^n, α_i^n ou β_i^n va impacter au niveau $n+1$ les 2 couples de coefficients fins $(l_j^{n+1}, \theta_j^{n+1})$ avec $j = 2i, 2i+1$, puis au niveau $n+2$ les 4 couples $(l_j^{n+2}, \theta_j^{n+2})$ avec $j = 4i, 4i+3$. Après k étapes de synthèse, les coefficients fins impactés sont les 2^k couples $(l_j^{n+k}, \theta_j^{n+k})$ avec $j = 2^k i, \ldots, 2^k i + 2^k - 1$.

Nous utilisons les formules de synthèse 3.3 et 3.4 pour exprimer la propagation de l'erreur à un niveau n vers un niveau de résolution plus fin $n+1$. Nous étudions successivement l'impact de la modification des coefficients grossiers l_i^n, θ_i^n, puis des coefficients de détail α_i^n, β_i^n.

Erreur sur l Nous modifions une longueur du polygone grossier : $l_i^{n\prime} = l_i^n + \varepsilon$. Les autres coefficients de la représentation MR restent inchangés. Seules les deux longueurs reconstruites comportent une erreur, les angles reconstruits restent identiques.

$$\begin{aligned}
l_{2i}^{n+1\prime} &= (l_i^n + \varepsilon)\,\frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n)} &&= l_{2i}^{n+1} + \frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n)}\,\varepsilon \\
l_{2i+1}^{n+1\prime} &= (l_i^n + \varepsilon)\,\frac{\sin\alpha_i^n}{\sin(\alpha_i^n+\beta_i^n)} &&= l_{2i+1}^{n+1} + \frac{\sin\alpha_i^n}{\sin(\alpha_i^n+\beta_i^n)}\,\varepsilon \\
\theta_{2i}^{n+1\prime} &= \theta_{2i}^{n+1} \\
\theta_{2i+1}^{n+1\prime} &= \theta_{2i+1}^{n+1}
\end{aligned} \tag{5.1}$$

Après plusieurs étapes de synthèse, seules les longueurs du polygone fin sont modifiées, ses angles restent identiques. Plus précisemment, une erreur sur l_i^n entraine une modification au niveau $n+k$ des l_j^{n+k}, avec $j = 2^k i, \ldots, 2^k i + 2^k - 1$. La Figure 5.8 (b) illustre ceci, en modifiant une longueur du polygone grossier, nous voyons que les segments fins correspondants au segment grossier modifié voient leurs longueurs modifiées, mais pas leurs angles. Les sommets fins correspondants aux segments grossiers non modifiés sont juste translatés, leur représentation longueur-angle étant resté identique.

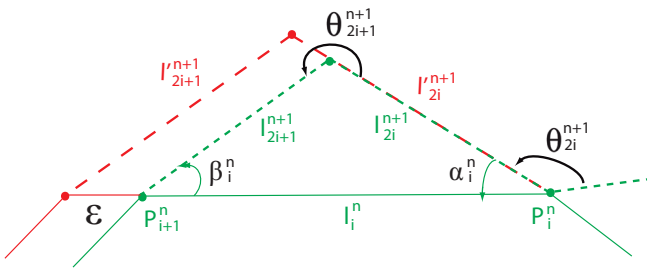


FIGURE 5.4 – Erreur sur une longueur l_i^n du polygone grossier

Erreur sur θ Nous modifions maintenant un angle du polygone grossier : $\theta_i^{n\prime} = \theta_i^n + \varepsilon$. Seul un des angles fins est impacté par cette erreur, les longueurs ainsi que l'autre angle fin restent inchangés. En fait, modifier l'angle grossier θ_i^n revient à appliquer une rotation de centre $P_i^n = P_{2i}^{n+1}$ et d'angle ε à toute la suite du polygone fin $P_{2i}^{n+1} \dots P_{N-1}^{n+1}$:

$$\begin{aligned} l_{2i}^{n+1\prime} &= l_{2i}^{n+1} \\ l_{2i+1}^{n+1\,\prime} &= l_{2i+1}^{n+1} \\ \\ \theta_{2i}^{n+1\prime} &= \theta_{2i}^{n+1} + \varepsilon \\ \theta_{2i+1}^{n+1\,\prime} &= \theta_{2i+1}^{n+1} \end{aligned} \tag{5.2}$$

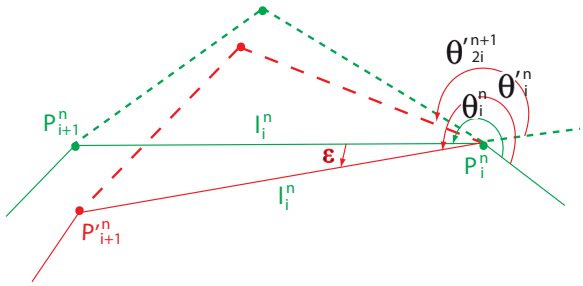


FIGURE 5.5 – Erreur sur un angle θ_i^n du polygone grossier

Aprés plusieurs étapes de synthèse, le polygone fin reconstruit subit juste une rotation d'angle ε, pour sa partie situé aprés l'angle modifié, comme illustré par la Figure 5.8 (c). Plus précisem-

ment, faire $\theta_i^{n\prime} = \theta_i^n + \varepsilon$ entraîne au niveau $n+k$: $\theta_{2^k i}^{n+k\prime} = \theta_{2^k i}^{n+k} + \varepsilon$, ce qui revient à appliquer une rotation d'angle ε et de centre $P_{2^k i}^{n+k}$ à tous les sommets suivants. Au final, avec ou sans MR, l'effet de la modification d'un angle θ est exactement le même.

Erreur sur α L'impact de la modification d'un des coefficients de détail : $\alpha_i^{n\prime} = \alpha_i^n + \varepsilon$ sur les coefficients du niveau plus fin $n+1$ est le suivant :

$$\begin{aligned}
l_{2i}^{n+1\prime} &= l_i^n \frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n+\varepsilon)} = l_i^n \frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n)+\varepsilon\cos(\alpha_i^n+\beta_i^n)} \\
l_{2i+1}^{n+1\prime} &= l_i^n \frac{\sin(\alpha_i^n+\varepsilon)}{\sin(\alpha_i^n+\beta_i^n+\varepsilon)} = l_i^n \frac{\sin\alpha_i^n+\varepsilon\cos\alpha_i^n}{\sin(\alpha_i^n+\beta_i^n)+\varepsilon\cos(\alpha_i^n+\beta_i^n)} \\
\theta_{2i}^{n+1\prime} &= \theta_{2i}^{n+1} - \varepsilon \\
\theta_{2i+1}^{n+1\prime} &= \theta_{2i+1}^{n+1} + \varepsilon
\end{aligned} \tag{5.3}$$

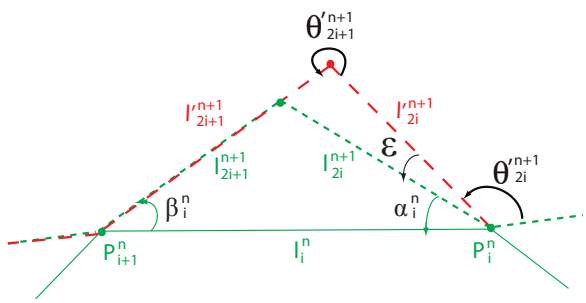


FIGURE 5.6 – Erreur sur un coefficient de détail α_i^n

Erreur sur β L'impact de la modification de l'angle de détail : $\beta_i^{n\prime} = \beta_i^n + \varepsilon$ sur les coefficients de niveau plus fin $n+1$ est le suivant :

$$\begin{aligned}
l_{2i}^{n+1\prime} &= l_i^n \frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n+\varepsilon)} = l_i^n \frac{\sin\beta_i^n}{\sin(\alpha_i^n+\beta_i^n)+\varepsilon\cos(\alpha_i^n+\beta_i^n)} \\
l_{2i+1}^{n+1\prime} &= l_i^n \frac{\sin(\alpha_i^n+\varepsilon)}{\sin(\alpha_i^n+\beta_i^n+\varepsilon)} = l_i^n \frac{\sin\alpha_i^n+\varepsilon\cos\alpha_i^n}{\sin(\alpha_i^n+\beta_i^n)+\varepsilon\cos(\alpha_i^n+\beta_i^n)} \\
\theta_{2i+1}^{n+1\prime} &= \theta_{2i+1}^{n+1} + \varepsilon \\
\theta_{2i+2}^{n+1\prime} &= \theta_{2i+2}^{n+1} - \varepsilon
\end{aligned} \tag{5.4}$$

L'effet de la modification d'un angle de détail α ou β après plusieurs étapes de synthèse est illustré dans la Figure 5.8 (d) et (e). La modification d'un coefficient de détail de la représentation

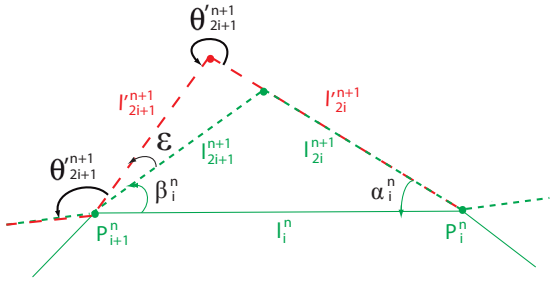


FIGURE 5.7 – Erreur sur un coefficient de détail β_i^n

MR du polygone induit la modification à la fois des longueurs et angles du polygone fin reconstruit. La déformation intervient uniquement pour les segments du polygone fin correspondants au segment grossier concerné par la modification, c'est à dire sur les 2^k couples $(l_j^{n+k}, \theta_j^{n+k})$ avec $j = 2^k i, \ldots, 2^k i + 2^k - 1$. Remarquons qu'à la différence de la modification d'un coefficient grossier, la modification d'un coefficient de détail sur un polygone fermé n'entraîne pas son ouverture. Ainsi, si le polygone grossier est fermé, le polygone fin reconstruit sera toujours fermé, quels que soient les coefficients de détail.

5.3 Conclusion

Au vu de notre étude de stabilité, nous pouvons pour commencer affirmer que la représentation multirésolution n'amplifie pas la propagation des erreurs par rapport à la simple représentation longueur-angle sans MR. Aussi, la représentation MR présente l'avantage d'assurer un polygone fin fermé si le polygone grossier est fermé. L'autre avantage d'utiliser la représentation MR est que l'influence de l'erreur au niveau n se limite au niveau $n+k$ aux 2^k couples de coefficients fins $(l_j^{n+k}, \theta_j^{n+k})$ avec $j = 2^k i, \ldots, 2^k i + 2^k - 1$. Ceci explique en partie les bons résultats que nous obtenons en pratique avec notre algorithme de morphing basé sur notre représentation MR intrinsèque, présentés dans la Section 4.2.

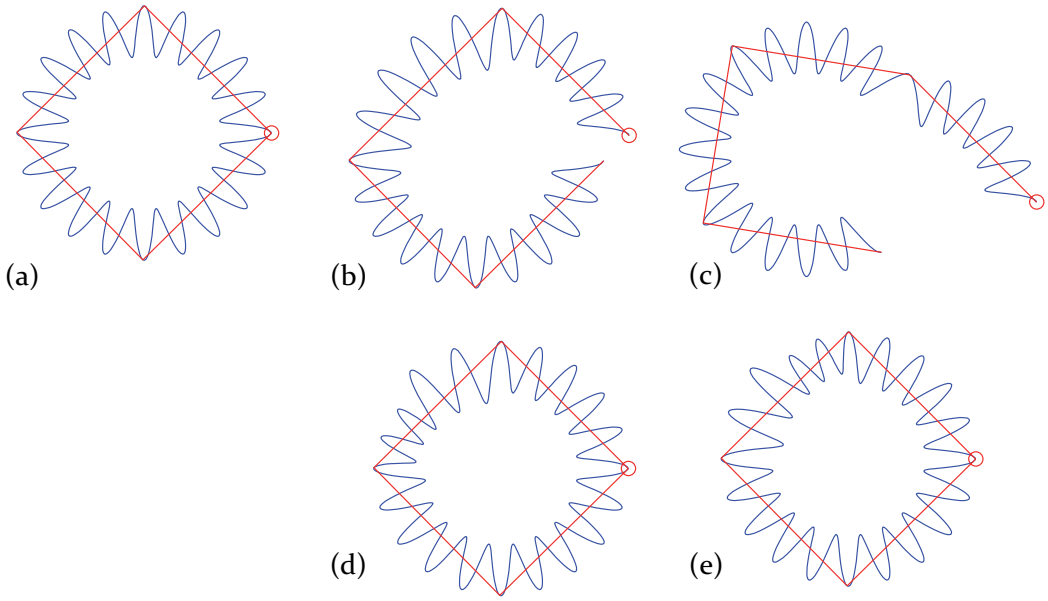


FIGURE 5.8 – *Ligne du haut* – (a) Le polygone reconstruit sans erreur (512 sommets). P_0 est marqué par un cercle. Le polygone grossier, issu de 7 étapes d'analyse, est représenté en rouge. (b) Avec une erreur sur une longueur du polygone grossier $l'_1 = 1.2\ l_1$. (c) Avec une erreur sur un angle du polygone grossier $\theta'_1 = 0.8\ \theta_1$. *Ligne du bas* – Modification de coefficients de détail de niveau le plus grossier (d) $\alpha'_1 = 1.5\ \alpha_1$ (e) $\beta'_1 = 1.5\ \beta_1$.

Conclusion

Nous avons présenté une nouvelle représentation multirésolution de formes polygonales, basée sur les angles et longueurs. Combinée à l'interpolation "aussi rigide que possible" de Alexa et al. [ACOL00], cette représentation est utilisée pour définir une nouvelle méthode de *morphing*.

L'intérêt d'une représentation intrinsèque, telle que celle que nous décrivons, est que l'orientation des détails suit naturellement n'importe quelle déformation. Puisque nous interpolons les représentations multirésolutions des courbes, nous obtenons sans calcul supplémentaire la représentation MR de toutes les courbes intermédiaires. Nous interpolons les polygones grossiers de façon aussi rigide que possible, et les angles de détails sont interpolés linéairement. Notre méthode a donc un comportement globalement aussi rigide que possible. Dans le cas de polygones fermés, contrairement à la méthode de Sederberg, notre méthode ne rencontre pas le problème de non fermeture des formes intermédiaires. Comme toutes les méthodes de *morphing* interpolant les contours des formes et non une représentation de l'intérieur des formes, notre méthode ne peut pas garantir l'absence d'auto intersections. Cependant, nous montrons au travers de nombreux exemples qu'il est possible de les éviter en pratique.

Troisième partie

Morphing multirésolution de courbes 3D

Résumé

Dans cette partie, nous introduisons un nouvel algorithme de morphing multirésolution de courbes de l'espace. Notre méthode repose sur une représentation multirésolution intrinsèque des courbes, basée sur des coordonnées sphériques locales. Les étapes d'analyse et de synthèse multirésolution sont données explicitement, sans système d'équations à résoudre. De plus, l'analyse multirésolution s'effectue à espace mémoire constant, et permet une reconstruction exacte de la courbe initiale. L'avantage par rapport à l'utilisation d'une analyse multirésolution basée sur des ondelettes est que l'orientation des détails suit intuitivement la déformation au cours du morphing. Nous démontrons l'efficacité de notre morphing MR en interpolant des courbes comprenant de nombreux détails.

Problématique

Suite aux bons résultats obtenus pour le morphing de courbes 2D, nous souhaitons développer une méthode similaire pour les courbes 3D, présentant les mêmes propriétés : un morphing intrinsèque rigide préservant les longueurs, qui conserve bien les détails au cours de l'interpolation et évite les auto intersections. Cependant, nous allons voir que nous ne pouvons pas directement transposer nos résultats sur les courbes 2D aux courbes 3D. Dans cette partie, nous introduisons une représentation multirésolution intrinsèque pour les courbes 3D polygonales. Elle est basée sur les coordonnées sphériques locales.

Dans le cas des courbes planaires, des règles trigonométriques basiques sur les triangles sont suffisantes pour expliciter les expressions de passage des coefficients fins aux grossiers et vice versa. Dans le cas 3D, la non-planarité des courbes impose d'introduire des repères locaux. Les formules d'analyse et de synthèse multirésolution sont ensuite obtenues en utilisant des transitions entre repères adjacents et en recalculant de nouveaux repères. Il est important de noter que toutes nos formules d'analyse et de synthèse sont au final entièrement exprimées dans notre représentation intrinsèque. Cela signifie que la courbe initiale aussi bien que ses représentations multirésolution sont entièrement et exclusivement exprimées en coordonnées sphériques locales. Les coefficients grossiers et de détail d'un niveau donné sont calculés à partir du niveau plus fin en utilisant uniquement la représentation intrinsèque. Cette nouvelle représentation est ensuite utilisée pour l'interpolation multirésolution de formes.

Comme nous l'avons vu dans l'introduction, le morphing de courbes 2D et en particulier la trajectoire des sommets au cours du morphing de courbes 2D a été beaucoup étudié en recherche.

A notre connaissance, nous sommes les premiers à présenter un algorithme qui détermine la trajectoire des points au cours du morphing de courbes 3D.

Cette partie est organisée comme suit. Nous introduisons d'abord la représentation intrinsèque de courbe basée sur des coordonnées sphériques locales, et explicitons le passage des coordonnées cartésiennes à cette représentation dans le Chapitre 6. Une représentation multirésolution basée sur ces coordonnées est développée dans le Chapitre 7. Ensuite, nous présentons notre algorithme de morphing MR et en discutons les résultats dans le Chapitre 8, avant de conclure.

Chapitre 6

Représentation intrinsèque d'une courbe 3D

Un critère essentiel pour évaluer une méthode d'édition est sa capacité à préserver les propriétés globales et locales du modèle original. L'édition multirésolution, grâce au processus d'analyse et de synthèse, conserve bien les propriétés des formes à différents niveaux de résolution. Cependant, nous devons faire particulièrement attention au comportement des détails fins. Comme mentionné dans l'introduction, les schémas multirésolution basés sur les ondelettes échouent généralement à conserver les détails fins, puisque les détails sont encodés dans un système de coordonnées global et que l'orientation des détails reste donc toujours fixe. Les limitations de ces techniques utilisant des coordonnées absolues ont conduit à l'introduction de techniques de déformation et de morphing basées sur des coordonnées locales [SK04b, LSLCO05, Sor06] et des représentations intrinsèques des formes [Ale02, SGWM93, HBCC07].

Comme rappelé dans la partie sur les courbes planaires, les fonctions courbure et torsion d'une courbe paramétrique C^3 de l'espace déterminent la courbe de façon unique, à une isométrie près [Car76]. La courbure et la torsion sont des propriétés intrinsèques d'une courbe lisse de l'espace. Dans le cas de courbes linéaires par morceaux, les longueurs et les angles définissent une courbe intrinsèquement, à une isométrie près [Ada75, SGWM93]. Notre analyse multirésolution est construite sur les coordonnées sphériques locales. Elles définissent les angles et la longueur de façon unique relativement à un sommet de son voisinage immédiat.

6.1 Coordonnées sphériques locales

Le système de coordonnées sphériques [Arf94] est un système de coordonnées qui permet de représenter des figures géométriques en trois dimensions en utilisant trois coordonnéees (l, θ, φ). Il existe différentes conventions pour représenter ces trois coordonnées. Nous utilisons les notations suivantes qui sont les plus appropriées dans notre travail, voir la Figure 6.1 (partie de

gauche) pour une illustration :

- l est la distance entre l'origine et un point donné P = (x,y,z).
- θ est l'angle entre l'axe des abscisses et la ligne entre l'origine et la projection de P sur le plan xy.
- φ est l'angle entre le plan xy et la ligne entre l'origine et P.

Les coordonnées sphériques (l,θ,φ) peuvent être déduites des coordonnées cartésiennes (x,y,z) de P :

$$\begin{aligned} l &= \sqrt{x^2+y^2+z^2} \\ \theta &= atan2(y,x) := \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{si } x>0 \\ \pi - \arctan\left(\frac{y}{x}\right) & \text{si } x<0 \\ sign(y)\cdot\frac{\pi}{2} & \text{si } x=0 \end{cases} \\ \varphi &= \arcsin\left(\frac{z}{\sqrt{x+y+z}}\right), \end{aligned} \tag{6.1}$$

avec $l \geq 0$, $\theta \in [-\pi,\pi]$, et $\varphi \in [-\pi/2,\pi/2]$. Remarquons que θ peut être simplement calculé avec la fonction C $atan2$.

Réciproquement, les coordonnées sphériques peuvent être calculées à partir des coordonnées cartésiennes :

$$\begin{aligned} x &= l\cos\theta\cos\varphi \\ y &= l\sin\theta\cos\varphi \\ z &= l\sin\varphi. \end{aligned} \tag{6.2}$$

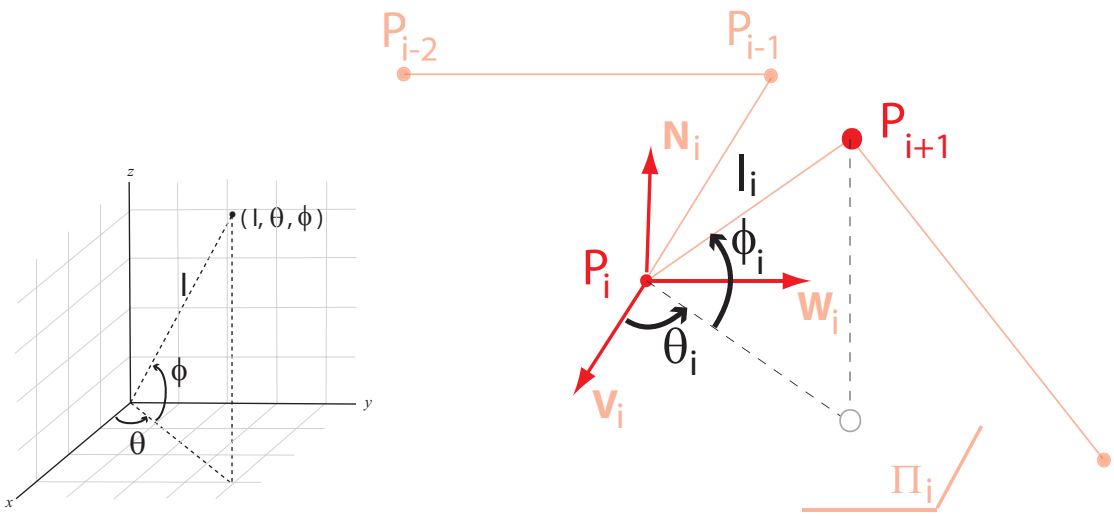


FIGURE 6.1 – Gauche : Coordonnées sphériques. Droite : Les coordonnées longueur-angles locales (l_i,θ_i,φ_i) décrivent la position de P_{i+1} par rapport à P_i.

6.2 Représentation (l,θ,φ) d'une courbe 3D

Nous introduisons maintenant une représentation longueurs-angles (l,θ,φ) pour une ligne polygonale de l'espace. Les sommets sont notés $P_i = [x_i, y_i, z_i]^T \in \mathbb{R}^3$, $i = 0,..,N-1$, avec N le nombre de sommets. Une courbe de l'espace peut être représentée en coordonnées-(l,θ,φ) en utilisant les coordonnées sphériques locales (l_i,θ_i,φ_i) qui décrivent de façon unique la position du sommet P_{i+1} par rapport à P_i, dans un repère orthonormal ayant pour origine le sommet précédent P_i, voir la Figure 6.1 (partie de droite).

Le repère orthonormal local $B_i = (P_i; \overrightarrow{V_i}, \overrightarrow{W_i}, \overrightarrow{N_i})$ est défini comme suit :

$$\begin{aligned}
\overrightarrow{V_i} &= \frac{\overrightarrow{P_{i-1}P_i}}{\left\|\overrightarrow{P_{i-1}P_i}\right\|} \\
\overrightarrow{N_i} &= \frac{\overrightarrow{P_{i-2}P_{i-1}} \wedge \overrightarrow{P_{i-1}P_i}}{\left\|\overrightarrow{P_{i-2}P_{i-1}} \wedge \overrightarrow{P_{i-1}P_i}\right\|} \quad , \quad \text{pour} \quad i = 0,\ldots,N-1 \\
\overrightarrow{W_i} &= \overrightarrow{N_i} \wedge \overrightarrow{V_i}
\end{aligned} \tag{6.3}$$

avec $\overrightarrow{V_i}$ le vecteur unitaire colinéaire à $\overrightarrow{P_{i-1}P_i}$, $\overrightarrow{N_i}$ la normale unitaire au plan Π_i contenant P_{i-2}, P_{i-1} et P_i, et $\overrightarrow{W_i}$ le vecteur tel que $(\overrightarrow{V_i}, \overrightarrow{W_i}, \overrightarrow{N_i})$ est une base orthonormale directe.

Les coordonnées-(x,y,z) des points de contrôle P_i peuvent être retrouvées directement à une rotation et une translation près. Cette rotation et cette translation peuvent être définies en choisissant par exemple le sommet P_0 comme point de référence et la base canonique $B_0 = \left(P_0; [1,0,0]^T, [0,1,0]^T, [0,0,1]^T\right)$ comme repère de référence.

Les coordonnées locales $(\overline{x}_{i+1}, \overline{y}_{i+1}, \overline{z}_{i+1})$ de P_{i+1} dans le repère local B_i sont calculées ainsi :

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{bmatrix}_{B_0} = M_{B_iB_0} * \begin{bmatrix} \overline{x}_{i+1} \\ \overline{y}_{i+1} \\ \overline{z}_{i+1} \end{bmatrix}_{B_i} + \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_{B_0} \quad , \quad \text{avec} \quad M_{B_iB_0} = \begin{pmatrix} \overrightarrow{V_i} & \overrightarrow{W_i} & \overrightarrow{N_i} \end{pmatrix}. \tag{6.4}$$

où $M_{B_iB_0}$ est la matrice de rotation de B_i vers B_0.

Dans le cas de points de contrôle alignés, les repères locaux ne peuvent être déterminés de façon unique. Nous prenons alors la convention d'utiliser le premier repère précédent qui a pû être déterminé.

Chapitre 7

Représentation multirésolution longueur-angles

L'analyse multirésolution est la combinaison d'étapes d'analyse et de synthèse, appelé l'algorithme banc de filtre (Figure 7.1 et Section 1.3). Rappelons qu'une étape d'analyse sur une courbe polygonale consiste en sa séparation en un ensemble de coefficients grossiers de basse résolution et de coefficients de détail nécessaires pour retrouver le polygone original. Les étapes d'analyse peuvent être répétées successivement jusqu'à obtenir une approximation suffisamment grossière. Nous pouvons donc décomposer le polygone de départ en une hiérarchie de polygones basse résolution et de coefficients de détail. En partant du polygone le plus grossier, nous pouvons reconstruire (synthétiser) itérativement les polygones fins en réappliquant les détails. Comme rappelé plus haut, la préservation des propriétés locales des formes, comme ses petits détails, est un point important lorsqu'on utilise une représentation multirésolution d'un objet pour l'éditer ou le déformer. Des représentations intrinsèques de la forme et des coordonnées locales sont les plus appropriées dans ce cas. La représentation longueur-angles d'un polygone basée sur ses coordonnées sphériques locales, présentée dans le Chapitre 6, sont de telles quantités intrinsèques.

Nous présentons dans ce chapitre une analyse multirésolution pour les courbes 3D, entièrement basée sur cette représentation intrinsèque. Comme dans le cas des courbes 2D, nous utilisons le sous-échantillonage pour déterminer la forme grossière, de telle façon qu'à chaque étape d'analyse, un point sur deux est supprimé, similairement aux ondelettes paresseuses [Swe97]. Le polygone ainsi que tous les polygones de plus basse résolution sont exclusivement exprimés en représentation MR longueur-angles, en coordonnées (l, θ, φ). De plus, nous proposons une représentation elle aussi intrinsèque pour les détails supprimés, comme dans notre représentation

MR en 2D (voir Chapitre 3), les détails seront représentés par des angles.

7.1 Notations

L'indice supérieur n indique le niveau de résolution. Nous notons P_i^n le i-ème point de la courbe à la résolution n. $(l_i^n, \theta_i^n, \varphi_i^n)$ décrivent la position de P_{i+1}^n relativement à P_i^n dans le repère local B_i^n, en coordonnées sphériques (voir Figure 6.1).

La base orthonormale $B_i^n = (P_i^n; \overrightarrow{V_i^n}, \overrightarrow{W_i^n}, \overrightarrow{N_i^n})$ est calculée à partir de P_{i-2}^n,P_{i-1}^n et P_i^n, de façon analogue à B_i dans la Section 6.2 (voir les équations (6.3)).

Dans les sections suivantes, les formules d'analyse et de synthèse pour la MR sont développées. Bien que tous les coefficients grossiers et de détail sont exprimés entièrement en coordonnées (l, θ, φ), nous utiliserons les coordonnées cartésiennes dans nos calculs intermédiaires pour décrire ces relations. Pour bien distinguer les coordonnées cartésiennes des coordonnées sphériques, nous utiliserons des crochets pour les coordonnées cartésiennes, et des parenthèses pour les coordonnées sphériques. Les coordonnées cartésiennes que nous utilisons sont généralement des coordonnées locales dans un repère local donné. L'indice B indique le repère local correspondant, par exemple

$$P_{i+1}^n = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{B_i^n} \quad \text{ou} \quad P_{i+1}^n = \begin{pmatrix} l_i^n \\ \theta_i^n \\ \varphi_i^n \end{pmatrix} = \begin{bmatrix} l_i^n \cos\theta_i^n \cos\varphi_i^n \\ l_i^n \sin\theta_i^n \cos\varphi_i^n \\ l_i^n \sin\varphi_i^n \end{bmatrix}_{B_i^n}.$$

Les coordonnées sphériques d'un sommet seront toujours exprimées dans le repère local centré sur le sommet précédent, nous ne donnerons donc pas d'indication de repère.

7.1.1 Principe d'une analyse MR en 3D

L'analyse (décomposition) est une procédure récursive consistant à diviser un vecteur de coefficients d'un polygone $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$ en un vecteur de coefficients grossiers (correspondant au polygone de basse-résolution) $(l^n, \theta^n, \varphi^n)$ et en un vecteur de coefficients de détail. L'indice n dénote alors le niveau de résolution. A partir d'un polygone à 2^{n+1} segments, une étape d'analyse crée un polygone à 2^n segments et 2^n coefficients de détail, lesquels sont représentés par deux vecteurs de la forme :

$$\begin{array}{ccc} (\theta^{n+1}, \mathbf{l}^{n+1}, \varphi^{n+1}) & \rightarrow & (\theta^n, \mathbf{l}^n, \varphi^n) \\ & \searrow & \\ & & (\alpha^n, \beta^n, \gamma^n), \end{array}$$

où $\theta^n = (\theta_0^n, \ldots, \theta_{2^n-1}^n)$, de même pour $\mathbf{l}^n, \varphi^n, \alpha^n, \beta^n, \gamma^n$. La Figure 7.1 illustre l'algorithme banc de filtres pour nos courbes 3D.

Dans notre représentation MR, les coefficients grossiers sont représentés en coordonnées sphériques locales (l, θ, φ) décrites dans le chapitre précédent. Les coefficients de détail représentent les détails perdus au cours d'une étape d'analyse, c'est-à-dire l'erreur entre deux niveaux successifs de résolution. Les coefficients de détail de degré n associés aux coefficients grossiers de degré n doivent permettre la reconstruction exacte du niveau fin suivant $n+1$. Dans l'analyse multirésolution que nous proposons, les détails sont exprimés sous forme d'angles. En 2D, nous avions besoin de deux angles pour retrouver exactement un sommet supprimé, en 3D nous avons logiquement besoin d'un troisième angle. Les coefficients de détail sont donc un ensemble de trois angles, que nous notons (α, β, γ), illustrés dans la Figure 7.5. Ces angles sont utilisés pour reconstruire les sommets supprimés. Les angles sont des caractéristiques intrinsèques du polygone, l'analyse multirésolution que nous proposons est donc construite sur une représentation intrinsèque.

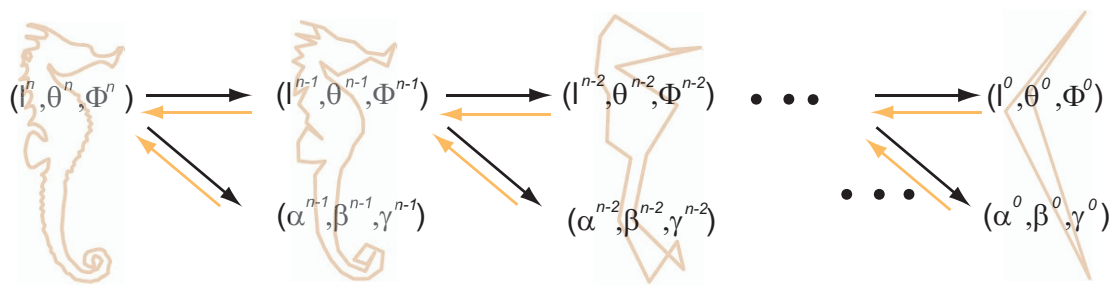


FIGURE 7.1 – L'algorithme banc de filtres : analyse et synthèse d'une courbe polygonale en coordonnées (l, θ, φ).

La Figure 7.2 présente deux niveaux de résolution. Sont représentées la courbe fine de niveau $n+1$, et en pointillés la courbe grossière de niveau n. Deux repères locaux sont alors attachés au même sommet $P_i^n = P_{2i}^{n+1}$, le repère grossier $B_i^n = (V_i^n, W_i^n, N_i^n)$ et le repère fin $B_{2i}^{n+1} = (V_{2i}^{n+1}, W_{2i}^{n+1}, N_{2i}^{n+1})$. Ces deux repères ne sont pas identiques, puisqu'ils ne sont pas définis à partir des mêmes sommets. En particulier, les normales N_{2i}^{n+1} et N_i^n engendrent deux plans différents, notés Π_i et Π_{2i}, engendrés respectivement par $P_{i-2}^n, P_{i-1}^n, P_i^n$ et $P_{2i-2}^{n+1}, P_{2i-1}^{n+1}, P_{2i}^{n+1}$ (voir la définition des vecteurs de base 6.3).

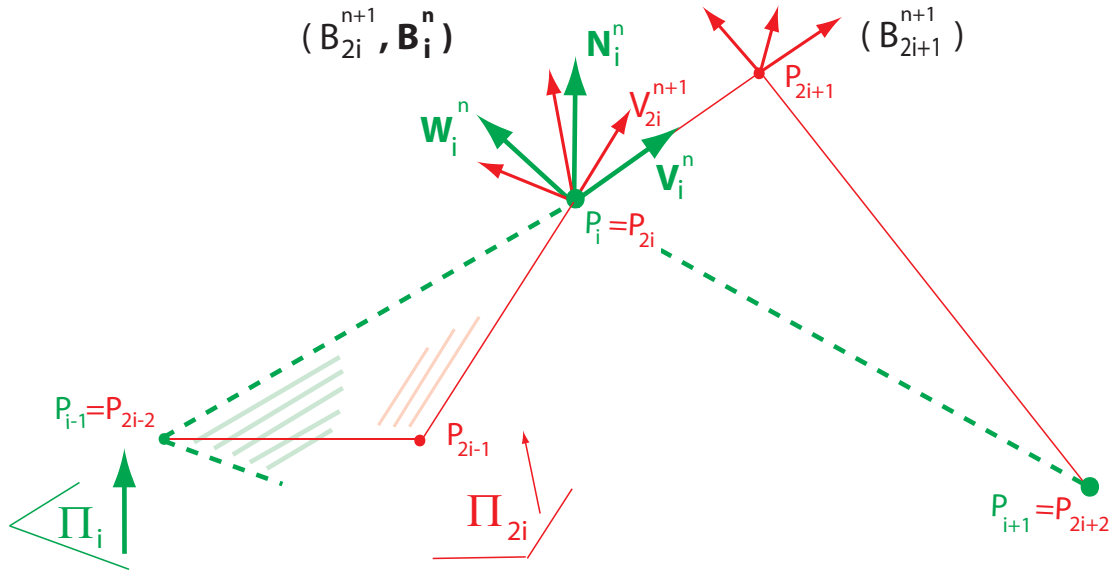


FIGURE 7.2 – Analyse multirésolution. Polygone à deux niveaux de résolution différents : n (vert) et $n+1$ (rouge). Les repères locaux B_{2i}^{n+1} et B_i^n ont leur origine en le même sommet $P_{2i}^{n+1} = P_i^n$.

La Figure 7.3 illustre la définition des coefficients de détail α_i^n, β_i^n, qui à partir du polygone grossier permettent de reconstruire le sommet P_{2i+1}^{n+1}, à une rotation près. α_i^n est l'angle orienté $\angle(\overrightarrow{P_{2i}^{n+1}P_{2i+2}^{n+1}}, \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}})$. β_i^n est l'angle orienté $\angle(\overrightarrow{P_{2i+2}^{n+1}P_{2i}^{n+1}}, \overrightarrow{P_{2i+2}^{n+1}P_{2i+1}^{n+1}})$. Le troisième coefficient de détail γ_i^n fixe la rotation du triangle $P_{2i}^{n+1}P_{2i+1}^{n+1}P_{2i+2}^{n+1}$ autour de l'axe $\overrightarrow{P_{2i}^{n+1}P_{2i+2}^{n+1}}$, voir Figure 7.3. γ_i^n désigne l'angle orienté entre les normales des repères B_{i+1}^n et B_{2i+2}^{n+1}, tous deux centrés en $P_{i+1}^n = P_{2i+2}^{n+1}$.

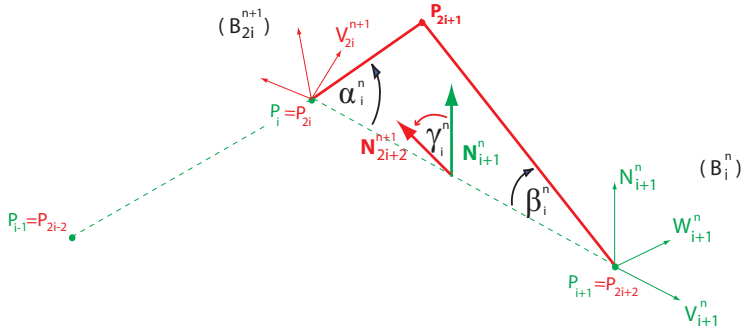


FIGURE 7.3 – coefficients de détail : angles $\alpha_i^n, \beta_i^n, \gamma_i^n$.

7.1.2 Matrices de changements de base

Nous travaillons dans des repères locaux. Pour pouvoir exprimer un même vecteur dans différents repères locaux, nous avons besoin de matrices de changement de base. Nous définissons deux matrices. Ces matrices de passage travaillent en coordonnées cartésiennes, et dépendent uniquement des coordonnées sphériques et des coefficients de détail. La première matrice $M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}$ permet de passer d'une base à sa voisine au même niveau de résolution. La seconde matrice $M_{B_{2i}^{n+1}B_{i}^{n}}$ permet de passer d'une base fine à une base grossière localisées sur le même sommet. Nous développons ci dessous les calculs de ces deux matrices. Nous les utiliserons ensuite dans nos calculs intermédiaires.

Matrice de passage $M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}$

Pour calculer la matrice de passage $M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}$, nous exprimons les vecteurs V_{2i+1}^{n+1}, W_{2i+1}^{n+1}, N_{2i+1}^{n+1} de la base B_{2i+1}^{n+1} dans le repère B_{2i}^{n+1}. La Figure 7.4 illustre ce calcul, notez bien que nous sommes en 3D, donc les deux triangles représentés ne sont pas dans le même plan.

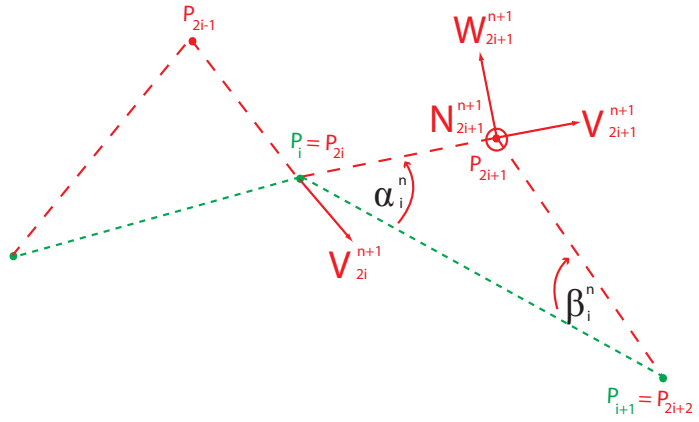


FIGURE 7.4 – Illustration pour le calcul du changement de base de B_{2i+1}^{n+1} vers B_{2i}^{n+1}. Le polygone grossier est représenté en pointillés.

Pour développer nos calculs, nous revenons à la définition des vecteurs de base (6.3) :

$$
\begin{aligned}
\left[V_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}} &= \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}/\|\,.\,\| = \begin{pmatrix} 1 \\ \theta_{2i}^{n+1} \\ \varphi_{2i}^{n+1} \end{pmatrix} = \begin{bmatrix} \cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ \sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ \sin\varphi_{2i}^{n+1} \end{bmatrix}_{B_{2i}^{n+1}} \\
\left[N_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}} &= (\overrightarrow{P_{2i-1}^{n+1}P_{2i}^{n+1}} \times \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}})/\|\,.\,\| \\
&= V_{2i}^{n+1} \times V_{2i+1}^{n+1} \\
&= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{B_{2i}^{n+1}} \times \left[V_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}} = \begin{bmatrix} 0 \\ -\sin\varphi_{2i}^{n+1} \\ \sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \end{bmatrix}_{B_{2i}^{n+1}}
\end{aligned}
$$

$$
\left[W_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}} = \left[N_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}} \times \left[V_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}} = \begin{bmatrix} -\sin^2\varphi_{2i}^{n+1} - \sin^2\theta_{2i}^{n+1}\cos^2\varphi_{2i}^{n+1} \\ \sin\theta_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos^2\varphi_{2i}^{n+1} \\ \sin\varphi_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \end{bmatrix}_{B_{2i}^{n+1}},
$$

avec "$/\|\,.\,\|$" qui indique une normalisation de vecteur.
Nous obtenons donc la matrice de passage :

$$
\begin{aligned}
M_{B_{2i+1}^{n+1}B_{2i}^{n+1}} &= \left(\left[V_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}}, \left[W_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}}, \left[N_{2i+1}^{n+1}\right]_{B_{2i}^{n+1}}\right) \\
&= \begin{pmatrix} \cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} & \sin^2\varphi_{2i}^{n+1} + \sin^2\theta_{2i}^{n+1}\cos^2\varphi_{2i}^{n+1} & 0 \\ \sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} & \sin\theta_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos^2\varphi_{2i}^{n+1} & -\sin\varphi_{2i}^{n+1} \\ \sin\varphi_{2i}^{n+1} & -\sin\varphi_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} & \sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \end{pmatrix}
\end{aligned} \tag{7.1}
$$

Matrice de passage $M_{B_{2i}^{n+1}B_i^n}$

Pour calculer la matrice de passage $M_{B_{2i}^{n+1}B_i^n}$, nous devons exprimer les coordonnées des vecteurs composants la base B_{2i}^{n+1} dans le repère B_i^n.
L'angle β_{i-1}^n est l'angle orienté $\angle(V_i^n, V_{2i}^{n+1})$. L'angle γ_{i-1}^n, est quant à lui l'angle orienté $\angle(N_i^n, N_{2i}^{n+1})$ (voir Figure 7.5). Nous reviendrons plus en détail sur le mode de calcul des angles de détails dans la section 7.2.2. Le changement de base de B_i^n vers B_{2i}^{n+1} peut donc être exprimé géométriquement comme la composition de deux rotations, voir la Figure 7.5 pour une illustration :

- une rotation d'angle β_{i-1}^n selon l'axe N_i^n, puis

– une rotation d'angle γ_{i-1}^n selon l'axe V_i^n.

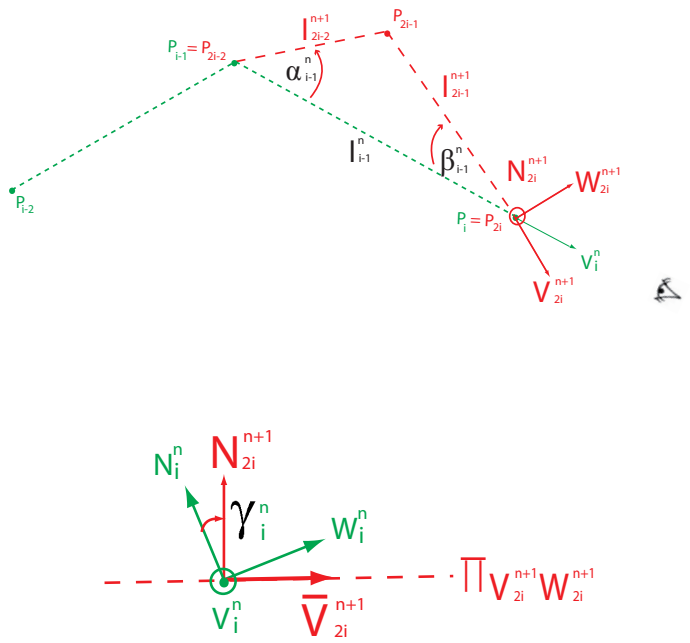


FIGURE 7.5 – Illustration pour le calcul du changement de base de B_{2i}^{n+1} vers B_i^n. Le polygone grossier est représenté en pointillés. L'oeil à droite indique le point de vue adopté pour la seconde figure. $\Pi_{V_{2i}^{n+1} W_{2i}^{n+1}}$ est le plan défini par V_{2i}^{n+1} et W_{2i}^{n+1}, il est évoqué en pointillés.

Les matrices de rotation correspondantes, notées respectivement $R_{N_i^n}(\beta_{i-1}^n)$ et $R_{V_i^n}(\gamma_{i-1}^n)$, ont une expression simple car l'axe de la rotation est un des vecteurs de la base B_i^n dans laquelle nous travaillons.

Dans B_i^n, nous avons :

$$R_{N_i^n}(\beta_{i-1}^n) = \begin{pmatrix} \cos\beta_{i-1}^n & -\sin\beta_{i-1}^n & 0 \\ \sin\beta_{i-1}^n & \cos\beta_{i-1}^n & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

et

$$R_{V_i^n}(\gamma_{i-1}^n) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma_{i-1}^n & -\sin\gamma_{i-1}^n \\ 0 & \sin\gamma_{i-1}^n & \cos\gamma_{i-1}^n \end{pmatrix}$$

Nous travaillons maintenant uniquement dans B_i^n, dans laquelle sont exprimées nos matrices de

rotation. Les coordonnées des vecteurs composants B_i^n, exprimées dans B_i^n, sont simplement les coordonnées de la base canonique. Nous leur appliquons successivement les deux rotations pour obtenir les coordonnées des vecteurs de B_{2i}^{n+1}, toujours dans la base B_i^n. Nous obtenons ainsi l'expression des coordonnées des vecteurs de B_i^n exprimées dans B_i^n :

$$\begin{aligned}
\left[V_{2i}^{n+1}\right]_{B_i^n} &= R_{V_i^n}(\gamma_{i-1}^n)\, R_{N_i^n}(\beta_{i-1}^n)\, [V_i^n]_{B_i^n} = R_{V_i^n}(\gamma_{i-1}^n)\, R_{N_i^n}(\beta_{i-1}^n) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{B_i^n} \\
&= \begin{bmatrix} \cos\beta_{i-1}^n \\ \sin\beta_{i-1}^n \cos\gamma_{i-1}^n \\ \sin\beta_{i-1}^n \sin\gamma_{i-1}^n \end{bmatrix}_{B_i^n} \\
\left[W_{2i}^{n+1}\right]_{B_i^n} &= R_{V_i^n}(\gamma_{i-1}^n)\, R_{N_i^n}(\beta_{i-1}^n)\, [W_i^n]_{B_i^n} = R_{V_i^n}(\gamma_{i-1}^n)\, R_{N_i^n}(\beta_{i-1}^n) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_{B_i^n} \\
&= \begin{bmatrix} -\sin\beta_{i-1}^n \\ \cos\beta_{i-1}^n \cos\gamma_{i-1}^n \\ \cos\beta_{i-1}^n \sin\gamma_{i-1}^n \end{bmatrix}_{B_i^n} \\
\left[N_{2i}^{n+1}\right]_{B_i^n} &= R_{V_i^n}(\gamma_{i-1}^n)\, R_{N_i^n}(\beta_{i-1}^n)\, [N_i^n]_{B_i^n} = R_{V_i^n}(\gamma_{i-1}^n)\, R_{N_i^n}(\beta_{i-1}^n) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{B_i^n} \\
&= \begin{bmatrix} 0 \\ -\sin\gamma_{i-1}^n \\ \cos\gamma_{i-1}^n \end{bmatrix}_{B_i^n}
\end{aligned}$$

Nous obtenons donc la matrice de passage :

$$\begin{aligned}
M_{B_{2i}^{n+1} B_i^n} &= \left(\left[V_{2i}^{n+1}\right]_{B_i^n}, \left[W_{2i}^{n+1}\right]_{B_i^n}, \left[N_{2i}^{n+1}\right]_{B_i^n} \right) \\
&= \begin{pmatrix} \cos\beta_{i-1}^n & -\sin\beta_{i-1}^n & 0 \\ \sin\beta_{i-1}^n \cos\gamma_{i-1}^n & \cos\beta_{i-1}^n \cos\gamma_{i-1}^n & -\sin\gamma_{i-1}^n \\ \sin\beta_{i-1}^n \sin\gamma_{i-1}^n & \cos\beta_{i-1}^n \sin\gamma_{i-1}^n & \cos\gamma_{i-1}^n \end{pmatrix}.
\end{aligned} \tag{7.2}$$

7.2 Analyse

La procédure récursive qui sépare les 2^{n+1} coefficients d'un polygone $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$ en un vecteur de 2^n coefficients grossiers d'un polygone de plus basse résolution $(l^n, \theta^n, \varphi^n)$ et un vecteur de 2^n coefficients de détail $(\alpha^n, \beta^n, \gamma^n)$ est appelée analyse (décomposition). Les étapes d'analyse sont illustrées dans la Figure 7.1 par les flèches noires. Nous allons maintenant introduire le calcul des coefficients grossiers et de détail de niveau n à partir des coefficients fins $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$.

7.2.1 Coefficients grossiers

Les coefficients grossiers $(l_i^n, \theta_i^n, \varphi_i^n)$ sont les coordonnées sphériques locales du sommet grossier P_{i+1}^n dans le repère local grossier $B_i^n = (P_i^n; V_i^n, W_i^n, N_i^n)$, voir Figure 7.2. Comme indiqué précédemment, nous utilisons pour nos calculs intermédiaires les coordonnées cartésiennes locales des points de contrôle. Cela nous permet d'utiliser les formules classiques de changements de base en coordonnées cartésiennes. Mais nous obtenons au final une expression des coefficents grossiers ainsi que des coefficients de détail qui dépend uniquement des coefficients fins $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$.

Notons que puisque nous travaillons avec des repères locaux, les coordonnées du point P_{i+1}^n dans le repère B_i^n sont identiques aux coordonnées du vecteur $\overrightarrow{P_i^n P_{i+1}^n}$ dans la même base, puisque P_i^n est l'origine de B_i^n. De plus, puisque le passage du niveau n à $n+1$ consiste à supprimer un sommet sur deux, donc le vecteur grossier $\overrightarrow{P_i^n P_{i+1}^n}$ peut être simplement exprimé à partir de points de contrôle fins :

$$\overrightarrow{P_i^n P_{i+1}^n} = \overrightarrow{P_{2i}^{n+1} P_{2i+2}^{n+1}} = \overrightarrow{P_{2i}^{n+1} P_{2i+1}^{n+1}} + \overrightarrow{P_{2i+1}^{n+1} P_{2i+2}^{n+1}}. \tag{7.3}$$

Les coefficients grossiers $(l_i^n, \theta_i^n, \varphi_i^n)$ sont les coordonnées sphériques locales de $\overrightarrow{P_i^n P_{i+1}^n}$ dans B_i^n. Nous souhaitons les exprimer en fonction des coefficients fins $(l_\cdot^{n+1}, \theta_\cdot^{n+1}, \varphi_\cdot^{n+1})$. Pour cela, nous nous servons de la relation entre les vecteurs grossiers et fins (7.3). Nous avonc donc besoin d'exprimer $\overrightarrow{P_{2i}^{n+1} P_{2i+1}^{n+1}}$ et $\overrightarrow{P_{2i+1}^{n+1} P_{2i+2}^{n+1}}$ dans le repère local grossier B_i^n. La seule information disponible pour chaque vecteur fin sont ses coordonnées (l, θ, φ) dans le repère local associé au sommet le précédant au même niveau de résolution. Nous avons donc besoin de calculer des changements de base pour les exprimer dans le repère grossier B_i^n.

Nous avons détaillé plus haut le calcul des matrices de changement de base. Nous avons à notre disposition une matrice de changement de base entre deux repères voisins de même niveau de résolution (changement de base fin-fin ou grossier-grossier), et une matrice de changement

de base entre deux repères de niveaux de résolutions différents mais centrés sur la même origine (changement de base fin-grossier ou grossier-fin), voir leur expression respective (7.1) et (7.2).

Nous structurons notre calcul en trois étapes :

1. Exprimer les coordonnées de $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ dans B_{2i}^{n+1} et de $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans B_{2i+1}^{n+1}.
2. Exprimer les coordonnées de $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ et $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans la même base grossière B_i^n.
3. En déduire les coordonnées intrinsèques grossières $(l_i^n, \theta_i^n, \varphi_i^n)$.

Etape 1 – Exprimer $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ dans B_{2i}^{n+1} et $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans B_{2i+1}^{n+1}

Nous commençons par exprimer les coordonnées cartésiennes locales de $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ dans B_{2i}^{n+1} et de $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans B_{2i+1}^{n+1}. En utilisant les équations de changement de coordonnées (6.2), nous obtenons directement :

$$\begin{aligned}
\left[\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}\right]_{B_{2i}^{n+1}} &= \begin{pmatrix} l_{2i}^{n+1} \\ \theta_{2i}^{n+1} \\ \varphi_{2i}^{n+1} \end{pmatrix} = \begin{bmatrix} l_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ l_{2i}^{n+1}\sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ l_{2i}^{n+1}\sin\varphi_{2i}^{n+1} \end{bmatrix}_{B_{2i}^{n+1}} \\
\left[\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}\right]_{B_{2i+1}^{n+1}} &= \begin{pmatrix} l_{2i+1}^{n+1} \\ \theta_{2i+1}^{n+1} \\ \varphi_{2i+1}^{n+1} \end{pmatrix} = \begin{bmatrix} l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\varphi_{2i+1}^{n+1} \end{bmatrix}_{B_{2i+1}^{n+1}}
\end{aligned} \tag{7.4}$$

Etape 2 – Calculer $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ et $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans la même base grossière B_i^n

Les coordonnées de $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ et $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ peuvent ensuite être exprimées dans le repère grossier B_i^n en utilisant les matrices de changement de base. Pour exprimer $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ dans B_i^n, un seul changement de base est nécessaire : de B_{2i}^{n+1} vers B_i^n qui ont le même sommet pour origine $P_{2i}^{n+1} = P_i^n$. Pour exprimer $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans B_i^n, deux changements de base sont nécessaires : de B_{2i+1}^{n+1} vers B_{2i}^{n+1} puis de B_{2i}^{n+1} vers B_i^n.

$$\begin{aligned}
\left[\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}\right]_{B_i^n} &= [M_{B_{2i}^{n+1}B_i^n}] \begin{bmatrix} l_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ l_{2i}^{n+1}\sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ l_{2i}^{n+1}\sin\varphi_{2i}^{n+1} \end{bmatrix}_{B_{2i}^{n+1}} \\
\left[\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}\right]_{B_i^n} &= [M_{B_{2i}^{n+1}B_i^n}][M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}] \begin{bmatrix} l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\varphi_{2i+1}^{n+1} \end{bmatrix}_{B_{2i+1}^{n+1}}
\end{aligned} \tag{7.5}$$

Etape 3 – Calculer$(l_i^n, \theta_i^n, \varphi_i^n)$

Nous avons donc les coordonnées cartésiennes de $\overrightarrow{P_i^nP_{i+1}^n} = \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}} + \overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans B_i^n, nous les notons v, w, n. Elles dépendent uniquement des coordonnées (l, θ, φ) de niveau fin, ainsi que des β_{i-1}^n et γ_{i-1}^n qui interviennent dans $M_{B_{2i}^{n+1}B_i^n}$. Nous avons :

$$\overrightarrow{P_i^nP_{i+1}^n} = \begin{bmatrix} v = v(l^{n+1}, \theta^{n+1}, \varphi^{n+1}) \\ w = w(l^{n+1}, \theta^{n+1}, \varphi^{n+1}) \\ n = n(l^{n+1}, \theta^{n+1}, \varphi^{n+1}) \end{bmatrix}_{B_i^n}.$$

Nous obtenons finalement l'expression des coefficients grossiers $(l_i^n, \theta_i^n, \varphi_i^n)$ en fonction des coefficients fins $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$, en appliquant la formule de changement de coordonnées (6.1) :

$$\begin{aligned}
l_i^n &= \sqrt{v^2+w^2+n^2} = \sqrt{(l_{2i}^{n+1})^2 + (l_{2i+1}^{n+1})^2 - 2l_{2i}^{n+1}l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1}} \\
\theta_i^n &= \operatorname{atan2}(w, v) \\
\varphi_i^n &= \arcsin\left(\frac{n}{l_i^n}\right).
\end{aligned} \tag{7.6}$$

7.2.2 Coefficients de détail

Les coefficients de détail représentent les détails perdus au cours d'une étape d'analyse, c'est-à-dire l'erreur entre deux niveaux successifs de résolution. Dans l'analyse multirésolution que nous proposons, les détails sont exprimés sous forme d'angles. Les coefficients de détail sont

un ensemble de trois angles $(\alpha^n, \beta^n, \gamma^n)$. Ces angles sont utilisés pour reconstruire les sommets supprimés.

Calcul de α_i^n et β_i^n

α_i^n et β_i^n sont les angles intérieurs des triangles fins supprimés $\Delta(P_{2i}^{n+1}, P_{2i+1}^{n+1}, P_{2i+2}^{n+1})$ comme décrit dans la Figure 7.6 (gauche). γ_i^n est l'angle entre le plan contenant le triangle fin supprimé et le plan contenant le triangle de niveau grossier $\Delta(P_{i-1}, P_i, P_{i+1})$ comme illustré dans la Figure 7.6 (droite). α_i^n et β_i^n permettent de reconstruire P_{2i+1} à partir des coefficients grossiers, à une rotation près, autour de l'axe $\overline{P_i^n P_{i+1}^n}$. L'angle γ_i^n caractérise cette rotation de façon unique. Nous décrivons ci-dessous le calcul des détails $(\alpha^n, \beta^n, \gamma^n)$ à partir des coefficients de niveau fin $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$.

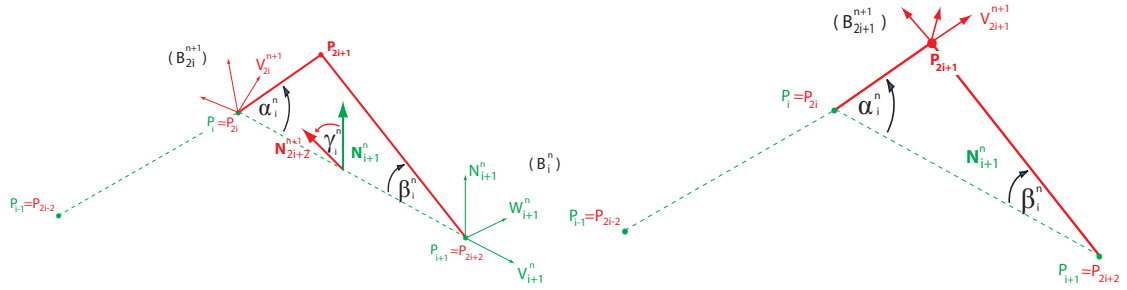


FIGURE 7.6 – Analyse multirésolution. Définition des 3 angles de détails α, β et γ

Comme décrit dans la Figure 7.6, nous avons :

$$\begin{aligned}
\alpha_i^n &= \angle(\overrightarrow{P_{2i}^{n+1}P_{2i+2}^{n+1}}, \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}) = \arccos\left(\frac{<\overrightarrow{P_{2i}^{n+1}P_{2i+2}^{n+1}}, \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}>}{\left\|\overrightarrow{P_{2i}^{n+1}P_{2i+2}^{n+1}}\right\| \left\|\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}\right\|}\right) \\
\beta_i^n &= \angle(\overrightarrow{P_{2i+2}^{n+1}P_{2i}^{n+1}}, \overrightarrow{P_{2i+2}^{n+1}P_{2i+1}^{n+1}}) = \arccos\left(\frac{<\overrightarrow{P_{2i+2}^{n+1}P_{2i}^{n+1}}, \overrightarrow{P_{2i+2}^{n+1}P_{2i+1}^{n+1}}>}{\left\|\overrightarrow{P_{2i+2}^{n+1}P_{2i}^{n+1}}\right\| \left\|\overrightarrow{P_{2i+2}^{n+1}P_{2i+1}^{n+1}}\right\|}\right).
\end{aligned} \tag{7.7}$$

Nous exprimons les trois sommets concernés en coordonnées cartésiennes locales dans la même base B_{2i+1}^{n+1}, voir Figure 7.6 (gauche)

$$P_{2i}^{n+1} = \begin{bmatrix} -l_{2i}^{n+1} \\ 0 \\ 0 \end{bmatrix}_{B_{2i+1}^{n+1}}, \quad P_{2i+1}^{n+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{B_{2i+1}^{n+1}}, \quad P_{2i+2}^{n+1} = \begin{bmatrix} l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\varphi_{2i+1}^{n+1} \end{bmatrix}_{B_{2i+1}^{n+1}},$$

et nous les insérons dans l'équation (7.7), ce qui nous donne le résultat suivant, en utilisant seulement les coefficients $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$:

$$\begin{aligned} \alpha_i^n &= \arccos\left(\frac{l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1}+l_{2i}^{n+1}}{\sqrt{(l_{2i}^{n+1})^2+(l_{2i+1}^{n+1})^2+2l_{2i}^{n+1}l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1}}}\right) \\ \beta_i^n &= \arccos\left(\frac{l_{2i}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1}+l_{2i+1}^{n+1}}{\sqrt{(l_{2i}^{n+1})^2+(l_{2i+1}^{n+1})^2+2l_{2i}^{n+1}l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1}}}\right), \end{aligned} \tag{7.8}$$

où $\alpha_i^n, \beta_i^n \in [0, \pi]$. Remarquons que le dénominateur est le coefficient grossier l_i^n déjà calculé précédemment.

Calcul de γ_i^n

Le coefficient de détail restant, γ_i^n, est défini comme l'angle $\angle(N_{i+1}^n, N_{2i+2}^{n+1})$ entre le plan contenant le triangle de niveau fin $\Delta(P_{2i}^{n+1}, P_{2i+1}^{n+1}, P_{2i+2}^{n+1})$ et le plan contenant le triangle de niveau grossier $\Delta(P_{i-1}^n, P_i^n, P_{i+1}^n)$, voir Figure 7.6 (droite). Pour calculer cet angle, nous devons exprimer ces deux normales dans le même repère local. Nous choisissons de les exprimer dans le repère B_{2i}^{n+1}, afin de minimiser le nombre de changements de base.

Nous avons directement les coordonnées du vecteur normal N_{2i+2}^{n+1} dans le repère local B_{2i+1}^{n+1} grâce à la formule (7.1.2) : $N_{2i+2}^{n+1} = [0, \sin\varphi_{2i+1}^{n+1}, -\sin\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1}]^T$. Reste à appliquer le changement de base de B_{2i+1}^{n+1} vers B_{2i}^{n+1}. N_{2i+2}^{n+1} est donc exprimé dans la base B_{2i}^{n+1} comme suit :

$$\left[N_{2i+2}^{n+1}\right]_{B_{2i}^{n+1}} = [\, M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}] \begin{bmatrix} 0 \\ \sin\varphi_{2i+1}^{n+1} \\ -\sin\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \end{bmatrix}_{B_{2i+1}^{n+1}} / \,\|\,.\,\|,$$

où "$/\|\,.\,\|$" dénote une normalisation du vecteur.

La normale de niveau grossier N_{i+1}^n est définie en terme de points de contrôle de niveau fin par le produit vectoriel suivant :

$$N_{i+1}^n = \left[\left(\overrightarrow{P_{2i-2}^{n+1}P_{2i-1}^{n+1}} + \overrightarrow{P_{2i-1}^{n+1}P_{2i}^{n+1}} \right) \times \left(\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}} + \overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}} \right) \right] / \| \cdot \|$$

Ces quatre vecteurs peuvent être exprimés dans le même repère local fin B_{2i}^{n+1} :

$$\left[\overrightarrow{P_{2i-2}^{n+1}P_{2i-1}^{n+1}}\right]_{B_{2i}^{n+1}} = [M_{B_{2i}^{n+1}B_{2i-1}^{n+1}}]^{-1} \begin{bmatrix} l_{2i-2}^{n+1} \\ 0 \\ 0 \end{bmatrix}_{B_{2i-1}^{n+1}}$$

$$\left[\overrightarrow{P_{2i-1}^{n+1}P_{2i}^{n+1}}\right]_{B_{2i}^{n+1}} = \begin{bmatrix} l_{2i-1}^{n+1} \\ 0 \\ 0 \end{bmatrix}_{B_{2i}^{n+1}}$$

$$\left[\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}\right]_{B_{2i}^{n+1}} = \begin{bmatrix} l_{2i}^{n+1}\cos\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ l_{2i}^{n+1}\sin\theta_{2i}^{n+1}\cos\varphi_{2i}^{n+1} \\ l_{2i}^{n+1}\sin\varphi_{2i}^{n+1} \end{bmatrix}_{B_{2i}^{n+1}}$$

$$\left[\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}\right]_{B_{2i}^{n+1}} = [M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}] \begin{bmatrix} l_{2i+1}^{n+1}\cos\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\theta_{2i+1}^{n+1}\cos\varphi_{2i+1}^{n+1} \\ l_{2i+1}^{n+1}\sin\varphi_{2i+1}^{n+1} \end{bmatrix}_{B_{2i+1}^{n+1}}$$

En combinant les résultats précédents, nous obtenons une expression du coefficient de détail γ_i^n fonction de coefficients de niveau fin $(l^{n+1}, \theta^{n+1}, \varphi^{n+1})$

$$\gamma_i^n = signe(|V_i^n N_{i+1}^n N_{2i+2}^{n+1}|) \arccos < N_{i+1}^n, N_{2i+2}^{n+1} >_{B_{2i}^{n+1}} . \tag{7.9}$$

où $\gamma_i^n \in [-\pi, \pi]$.

Pour $x \in [-1;1]$, nous avons $arccos(x) \in [0;\pi]$. Nous devons donc ajouter un signe pour que l'angle γ_i^n décrive toutes les rotations possibles autour de l'axe $\overline{P_i^n P_{i+1}^n}$. Ce signe est donné par le déterminant des trois vecteurs V_i^n, N_{i+1}^n et N_{2i+2}^{n+1}. Ainsi, nous avons bien $\gamma_i^n \in [-\pi, \pi]$. Nous avons déjà exprimé plus haut N_{i+1}^n et N_{2i+2}^{n+1} dans B_{2i}^{n+1}, reste à exprimer V_i^n dans cette même base locale. Pour cela, il suffit de noter que $V_i^n = \overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}} + \overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$.

7.3 Synthèse

Le résultat des étapes successives d'analyse est un polygone grossier de forme géométrique très simple (l^0, θ^0, ϕ^0) ainsi qu'un ensemble de coefficients de détail issus des différentes étapes d'analyse $(\alpha^0, \beta^0, \gamma^0)$, ..., $(\alpha^n, \beta^n, \gamma^n)$.

Une étape de synthèse consiste à reconstruire un polygone de niveau fin $(l^{n+1}, \theta^{n+1}, \phi^{n+1})$ à partir des coefficients grossiers (l^n, θ^n, ϕ^n) et des coefficients de détail $(\alpha^n, \beta^n, \gamma^n)$. Les étapes de synthèse sont illustrées dans la Figure 7.1 par les flèches oranges. Plusieurs étapes successives de synthèse permettent donc de reconstruire le polygone initial. Comme pour l'analyse, nous développons les formules de synthèse exclusivement en coordonnées (l, θ, ϕ). Nous gardons les notations introduites dans la section précédente.

7.3.1 Cas général

Dans le cas général, nous supposons $\beta_i^n \neq 0$ de telle façon que les P_{2i}^{n+1}, P_{2i+1}^{n+1} et P_{2i+2}^{n+1} reconstruits ne seront pas colinéaires. Les formules de synthèse des longueurs l_{2i}^{n+1} and l_{2i+1}^{n+1} peuvent être exprimées directement en utilisant les formules trigonométriques simples sur les triangles, exactement de la même façon que pour les formules de synthèse des longueurs en 2D (voir équation 3.3) :

$$l_{2i}^{n+1} = l_i^n \frac{\sin \beta_i^n}{\sin(\alpha_i^n + \beta_i^n)}, \qquad l_{2i+1}^{n+1} = l_i^n \frac{\sin \alpha_i^n}{\sin(\alpha_i^n + \beta_i^n)}. \tag{7.10}$$

Nous détaillons maintenant les formules de synthèse pour reconstruire les angles fins θ^{n+1} et φ^{n+1}. Pour commencer, un sommet $P' \in \Pi_{i+1}$, projection de P_{2i}^{n+1} sur Π_{i+1}, est reconstruit à partir des coefficients de détail α_i^n, β_i^n en utilisant les coordonnées polaires, puisque la composante normale est nulle dans le repère B_{i+1}^n, voir la Figure 7.7 (gauche).

Nous avons

$$\overrightarrow{P_{2i}^{n+1}P'} = \begin{bmatrix} l_{2i}^{n+1} \cos \alpha_i^n \\ l_{2i}^{n+1} \sin \alpha_i^n \\ 0 \end{bmatrix}_{B_{i+1}^n}$$

Pour calculer $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$, le troisième coefficient de détail γ_i^n est utilisé. Par définition, il représente l'angle de rotation entre P_{2i+1}^{n+1} et sa projection P' selon l'axe $\overline{P_{2i}^{n+1}P_{2i+2}^{n+1}}$. Voir la Figure 7.7 (droite), pour voir le passage de P' à P_{2i+1}^{n+1}. On a donc :

$$\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}} = R_{V_{i+1}^n}(\gamma_i^n)\left[\overrightarrow{P_{2i}^{n+1}P'}\right]_{B_{i+1}^n} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma_i^n & -\sin\gamma_i^n \\ 0 & \sin\gamma_i^n & \cos\gamma_i^n \end{pmatrix} \begin{bmatrix} l_{2i}^{n+1}\cos\alpha_i^n \\ l_{2i}^{n+1}\sin\alpha_i^n \\ 0 \end{bmatrix}_{B_{i+1}^n}$$

$$= \begin{bmatrix} l_{2i}^{n+1}\cos\alpha_i^n \\ -l_{2i}^{n+1}\sin\alpha_i^n\cos\gamma_i^n \\ -l_{2i}^{n+1}\sin\alpha_i^n\sin\gamma_i^n \end{bmatrix}_{B_{i+1}^n}$$

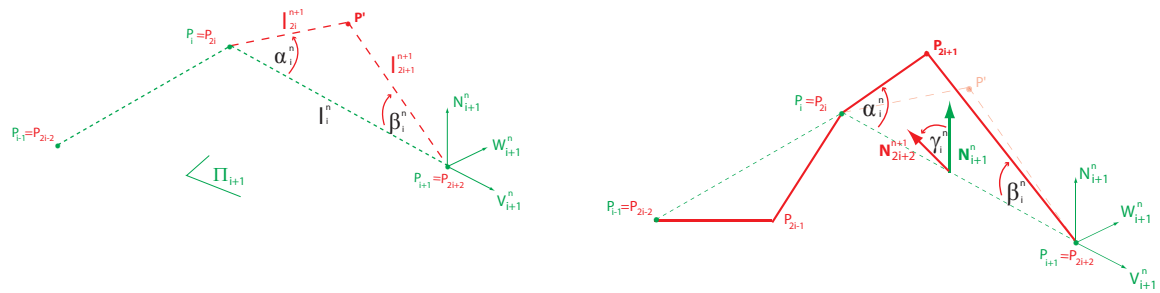


FIGURE 7.7 – Etape de synthèse du niveau n vers le niveau $n+1$.

Reste à exprimer $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ dans le repère local B_{2i}^{n+1}, afin d'en déduire ses coordonnées (l,θ,ϕ). Pour cela, deux changements de base sont nécessaires : de B_{i+1}^n à B_i^n puis de B_i^n à B_{2i}^{n+1}.

Nous obtenons donc les coordonnées cartésiennes de $\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}}$ qui dépendent uniquement des coefficients grossiers et de détail de niveau n :

$$\overrightarrow{P_{2i}^{n+1}P_{2i+1}^{n+1}} = [M_{B_{2i}^{n+1}B_i^n}]^{-1}[M_{B_{i+1}^nB_i^n}] \begin{bmatrix} l_{2i}^{n+1}\cos\alpha_i^n \\ -l_{2i}^{n+1}\sin\alpha_i^n\cos\gamma_i^n \\ -l_{2i}^{n+1}\sin\alpha_i^n\sin\gamma_i^n \end{bmatrix}_{B_{i+1}^n}$$

$$=: \begin{bmatrix} x_1 = x_1(l^n,\theta^n,\varphi^n,\alpha^n,\beta^n,\gamma^n) \\ y_1 = y_1(l^n,\theta^n,\varphi^n,\alpha^n,\beta^n,\gamma^n) \\ z_1 = z_1(l^n,\theta^n,\varphi^n,\alpha^n,\beta^n,\gamma^n) \end{bmatrix}_{B_{2i}^{n+1}}$$

Nous pouvons maintenant, grâce aux formules (6.1) et (7.10) exprimer les coordonnées sphériques de P_{2i+1}^{n+1}, ce qui revient à exprimer θ_{2i}^{n+1} et φ_{2i}^{n+1}, qui dépendent uniquement des coeffi-

cients de niveau n :

$$\begin{aligned} \theta_{2i}^{n+1} &= \operatorname{atan2}\left(y_1, x_1\right) \\ \varphi_{2i}^{n+1} &= \arcsin\left(\frac{z_1}{l_{2i}^{n+1}}\right). \end{aligned} \tag{7.11}$$

Nous procédons de la même façon pour calculer le second couple de coefficients fins θ_{2i+1}^{n+1} et φ_{2i+1}^{n+1}. Nous avons

$$\overrightarrow{P'P_{2i+2}^{n+1}} = \begin{bmatrix} l_{2i+1}^{n+1}\cos\beta_i^n \\ l_{2i+1}^{n+1}\sin\beta_i^n \\ 0 \end{bmatrix}_{B_{i+1}^n}$$

Nous utilisons γ_i^n pour retrouver le sommet fin :

$$\begin{aligned} \overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma_i^n & -\sin\gamma_i^n \\ 0 & \sin\gamma_i^n & \cos\gamma_i^n \end{pmatrix} \begin{bmatrix} l_{2i+1}^{n+1}\cos\beta_i^n \\ l_{2i+1}^{n+1}\sin\beta_i^n \\ 0 \end{bmatrix}_{B_{i+1}^n} \\ &= \begin{bmatrix} l_{2i+1}^{n+1}\cos\beta_i^n \\ l_{2i+1}^{n+1}\sin\beta_i^n\cos\gamma_i^n \\ l_{2i+1}^{n+1}\sin\beta_i^n\sin\gamma_i^n \end{bmatrix}_{B_{i+1}^n} \end{aligned}$$

Nous souhaitons maintenant exprimer $\overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}}$ dans B_{2i+1}^{n+1}, ce qui nécessite trois changements de base successifs :

$$\begin{aligned} \overrightarrow{P_{2i+1}^{n+1}P_{2i+2}^{n+1}} &= [\,M_{B_{2i+1}^{n+1}B_{2i}^{n+1}}]^{-1}[\,M_{B_{2i}^{n+1}B_i^n}]^{-1}[\,M_{B_{i+1}^nB_i^n}] \begin{bmatrix} l_{2i+1}^{n+1}\cos\beta_i^n \\ l_{2i+1}^{n+1}\sin\beta_i^n\cos\gamma_i^n \\ l_{2i+1}^{n+1}\sin\beta_i^n\sin\gamma_i^n \end{bmatrix}_{B_{i+1}^n} \\ &=: \begin{bmatrix} x_2 = x_2(l^n,\theta^n,\varphi^n,\alpha^n,\beta^n,\gamma^n) \\ y_2 = y_2(l^n,\theta^n,\varphi^n,\alpha^n,\beta^n,\gamma^n) \\ z_2 = z_2(l^n,\theta^n,\varphi^n,\alpha^n,\beta^n,\gamma^n) \end{bmatrix}_{B_{2i+1}^{n+1}} \end{aligned}$$

Et nous déduisons pour finir le second couple de coefficients d'angles fins, qui dépendent eux aussi uniquement des coefficients de niveau grossier n :

$$\begin{aligned} \theta_{2i+1}^{n+1} &= \operatorname{atan2}\left(y_2, x_2\right) \\ \varphi_{2i+1}^{n+1} &= \arcsin\left(\frac{z_2}{l_{2i+1}^{n+1}}\right). \end{aligned} \tag{7.12}$$

7.3.2 Cas particuliers

Quand un cas particulier $(\alpha_i^n, \beta_i^n) = (0,0)$, $(0,\pi)$ ou $(\pi,0)$ intervient, les formules de synthèse des longueurs (7.10) sont incorrectes, car nous avons une division par zéro. Géométriquement, ces cas correspondent aux situations où les trois sommets fins P_{2i}^{n+1}, P_{2i+1}^{n+1}, P_{2i+2}^{n+1} sont colinéaires. Ces situations sont valides, mais elles nécessitent un traitement particulier. Si un de ces cas est rencontré au cours de l'analyse, nous stockons dans la variable α_i^n une valeur particulière identifiant le cas dans lequel nous nous trouvons. Ceci nous permet au moment de la synthèse de détecter les cas particuliers à traiter. De plus, nous donnons à la variable β_i^n un ratio de longueurs R, qui permet de reconstruire les longeurs l_{2i}^{n+1}, l_{2i+1}^{n+1} correctement au moment de la synthèse, voir la Table 7.1.

(α_i, β_i)	Analyse	Synthèse
$(0,0)$	$R = \frac{l_{2i}^{n+1}}{l_{2i+1}^{n+1} + l_{2i}^{n+1}}$	$l_{2i}^{n+1} = l_i^n \cdot R$ $l_{2i+1}^{n+1} = l_i^n \cdot (1-R)$
$(0,\pi)$	$R = \frac{l_{2i}^{n+1}}{l_{2i}^{n+1} - l_{2i+1}^{n+1}}$	$l_{2i}^{n+1} = l_i^n \cdot R$ $l_{2i+1}^{n+1} = l_i^n \cdot (R-1)$
$(0,\pi)$	$R = \frac{l_{2i+1}^{n+1}}{l_{2i}^{n+1} - l_{2i}^{n+1}}$	$l_{2i}^{n+1} = l_i^n \cdot R$ $l_{2i+1}^{n+1} = l_i^n \cdot (1+R)$

TABLE 7.1 – Solution des cas particuliers de l'analyse et synthèse.

Chapitre 8

Morphing multirésolution de Courbes 3D

Nous utilisons notre représentation intrinsèque multirésolution de courbes 3D pour interpoler deux courbes. L'intérêt d'utiliser cette représentation pour le morphing est que l'étape d'interpolation est très simple. Une fois les représentations longueur-angles multirésolution calculées pour nos deux courbes source et destination, il suffit d'interpoler linéairement les coefficients de cette représentation pour obtenir un morphing rigide. A la différence d'autres méthodes qui travaillent avec une représentation *xyz* des courbes, et qui doivent donc utiliser un algorithme complexe d'interpolation pour pallier aux problèmes induits par cette représentation.

8.1 Algorithme

Nous considérons que nous travaillons en coordonnées longueur-angles. Le principe de l'algorithme de morphing multirésolution que nous proposons est le même qu'en 2D, et peut être décomposé en trois étapes illustrées dans Figure 8.1 :

1. Analyse - Calcul de la représentation multirésolution de chacune des deux courbes, en appliquant plusieurs étapes d'analyse.
2. Interpolation - Interpolation des représentations MR obtenues.
3. Synthèse - Reconstruction de la courbe intermédiaire, en appliquant plusieurs étapes de synthèse.

La première et la dernière étape, l'analyse et la synthèse, ont été détaillées dans le chapitre précédent. Reste à décrire l'interpolation des représentation longueur-angles multirésolution.

A l'issue de l'analyse, nous obtenons la représentation MR du polygone source ainsi que du polygone destination. Cette représentation consiste pour chacune de ces deux courbes en un

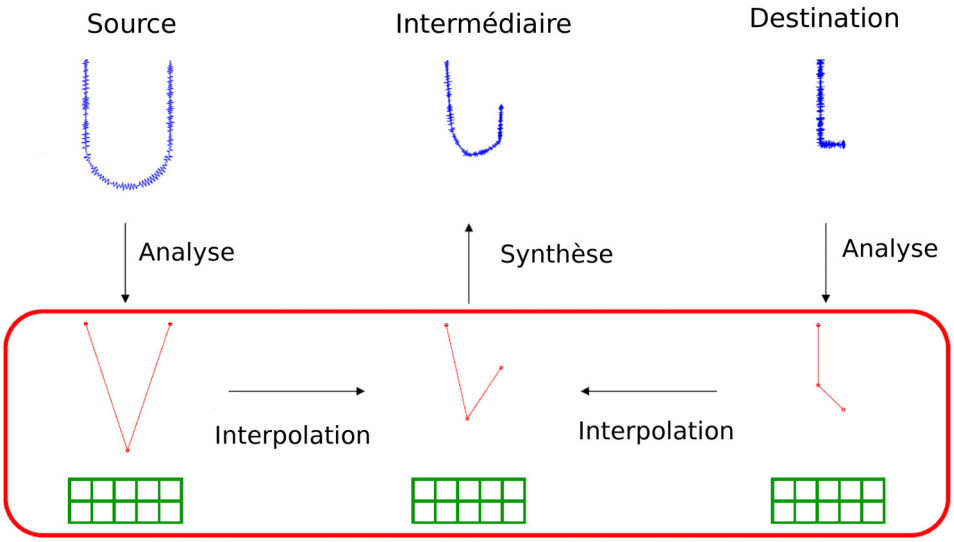


FIGURE 8.1 – Les étapes de notre algorithme de morphing multirésolution.

ensemble de coefficients grossiers et de détail, que nous allons maintenant interpoler pour obtenir les représentations MR des polygones intermédiaires. Nous traitons séparément les coefficients grossiers et les coefficients de détail.

8.1.1 Interpolation des coefficients grossiers

Les coefficients grossiers sont un ensemble de longueurs et d'angles décrivant le polygone grossier obtenu après les étapes d'analyse. Ce polygone a peu de sommets.

Interpolation linéaire

Nous pouvons simplement interpoler linéairement les coefficients grossiers obtenus. Puisque les polygones grossiers sont peu détaillés, les résultats sont satisfaisants.

$$\begin{aligned} l_t^j &= (1-t)l_S^j + t l_T^j \\ \theta_t^j &= (1-t)\theta_S^j + t\theta_T^j, \text{ pour } \quad 0 \leq j \leq n \quad \text{ et } \quad t \in [0,1]. \\ \phi_t^j &= (1-t)\phi_S^j + t\phi_T^j \end{aligned} \tag{8.1}$$

Interpolation aussi-rigide-que-possible

Dans le cas où les polygones grossiers sont réduits à des triangles, nous pouvons, de façon similaire à notre algorithme pour les courbes planaires, les interpoler avec la méthode aussi-rigide-que-possible que Sorkine et Alexa ont développé pour les surfaces discrètes [SA07]. Il suffit d'estimer le mouvement entre les deux triangles en 3D et non plus en 2D.

Cette possibilité n'a pas été mise en oeuvre dans le cadre de cette thèse.

8.1.2 Interpolation des coefficients de détail

Les coefficients de détail sont des angles. Nous les interpolons linéairement. Une évolution linéaire des angles assure un comportement rigide à notre morphing [ACOL00].

$$\begin{aligned} \alpha_t^j &= (1-t)\alpha_S^j + t\alpha_T^j, \\ \beta_t^j &= (1-t)\beta_S^j + t\beta_T^j, \text{ pour } \quad 0 \leq j \leq n \quad \text{ et } \quad t \in [0,1]. \\ \gamma_t^j &= (1-t)\gamma_S^j + t\beta_T^j, \end{aligned} \tag{8.2}$$

8.2 Résultats et discussion

Les polygones utilisés dans nos exemples ont été construits mathématiquement, il est donc aisé d'en choisir le nombre de sommets. Pour un polygone fin de $D2^n$ sommets, nous obtenons un polygone grossier de D sommets après n étapes de synthèse. En pratique, D est choisi petit, ainsi la majorité de l'information de la forme de départ est contenue dans les coefficients de détail qui sont des angles. Ces angles de détail étant interpolés linéairement, nous assurons un comportement rigide à notre morphing.

De la même façon que pour nos polygones 2D (voir Section 4.2.1), nous devons choisir un appariement des sommets pertinent pour assurer de bons résultats. Dans la construction mathématique de nos exemples, nous avons pris en compte cette contrainte, de façon à construire des couples de polygones bien appariés. Par exemple, les polygones de la Figure 8.4 sont tous deux construits à partir d'un morceau d'hélice lisse auquel sont ajoutés différents détails, ceux de la Figure 8.3 sont des courbes balle de tennis construites avec différents paramètres. La préparation des données est donc plus simple que pour nos polygones 2D qui étaient eux extraits de contours d'images.

La Figure 8.2 illustre l'intérêt de notre méthode de morphing par rapport à une simple interpolation linéaire. Nous interpolons deux polygones à seulement 3 sommets (représentés en rouge). Ces deux polygones ont deux sommets en commun, seul leur deuxième sommet diffère. En interpolant linéairement les coordonnées cartésiennes de ces deux polygones, le deuxième sommet

suit une trajectoire en ligne droite (ligne du bas). Au cours de l'interpolation, la longueur du polygone décroit donc pour ensuite augmenter de nouveau. En appliquant une interpolation linéaire des coordonnées longueur-angles de ces mêmes polygones, nous obervons que la trajectoire du deuxième sommet assure une interpolation rigide (ligne du haut). La longueur du polygone est préservée tout au long de l'interpolation.

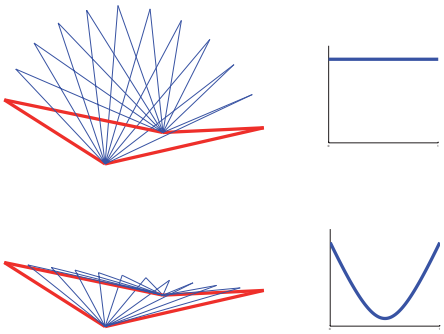


FIGURE 8.2 – Comparaison de notre méthode (ligne du haut) avec l'interpolation linéaire (ligne du bas) de deux polygones à 3 sommets (en rouge). A droite : évolution de la longueur du polygone au cours de chacune de ces interpolations.

Nous présentons dans cette partie les résultats obtenus en utilisant notre algorithme de morphing multirésolution longueur-angles. Les résultats présentés illustrent les différents apports de notre méthode, en la comparant à d'autres.

8.2.1 Intérêt par rapport à l'interpolation linéaire

La Figure 8.3 illustre la différence de notre méthode intrinsèque MR par rapport à une simple interpolation linéaire. La courbe source "couture de balle de tennis" est représentée en bleu clair, la courbe destination est représentée en magenta, c'est un cercle légèrement non plan. Les dix courbes intermédiaires sont représentée avec un dégradé de couleurs du bleu vers le magenta. La représentation sous forme de maillage permet une visualisation plus intuitive de l'évolution des courbes 3D au cours de l'interpolation, en liant les courbes successives les unes aux autres.

Dans le cas de l'interpolation linéaire (ligne du bas), la trajectoire des sommets est linéaire, alors que notre morphing MR induit des trajectoires courbes qui semblent plus naturelles et donnent une interpolation plus rigide (ligne du haut).

Cette différence apparait de façon quantifiée en comparant l'évolution de la longueur de la courbe au cours de l'interpolation. Avec l'interpolation linéaire, la longueur de la courbe diminue

puis augmente de nouveau au cours de l'interpolation. Avec notre méthode, l'évolution de la longueur est plus "linéaire".

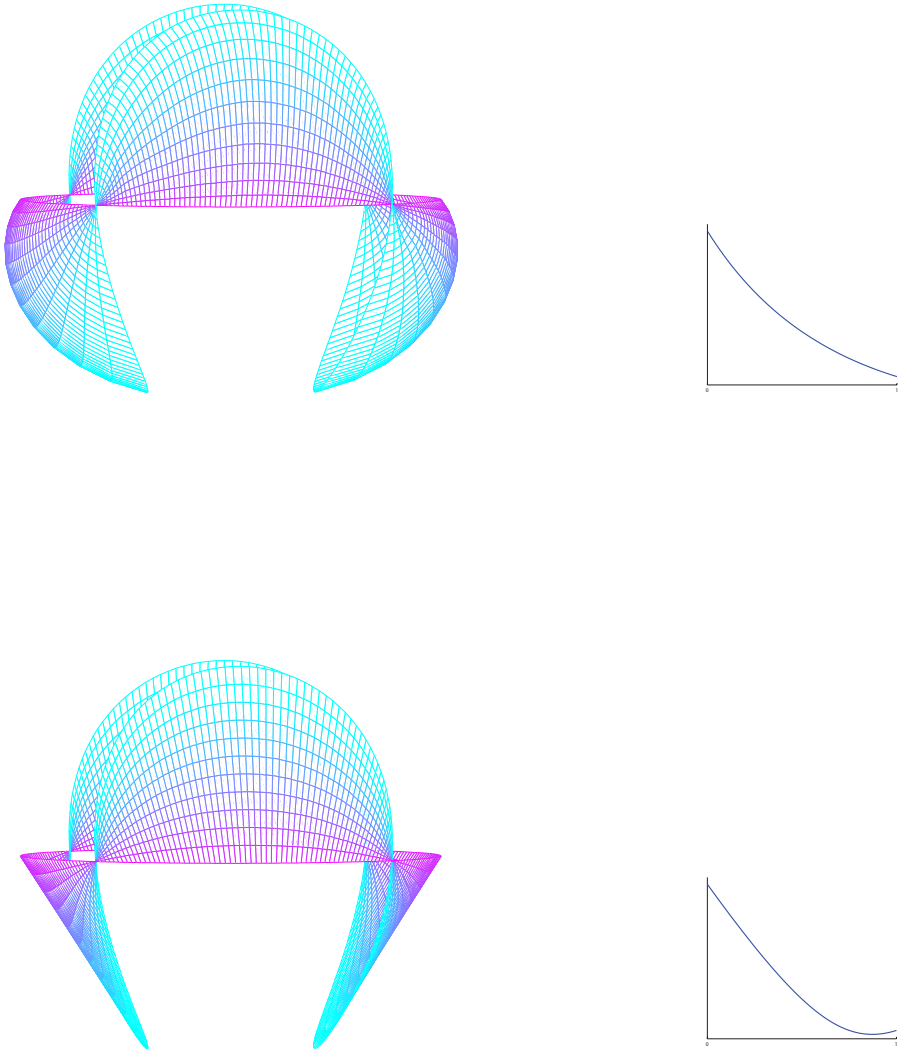


FIGURE 8.3 – Comparaison du morphing MR avec 6 niveaux de résolution (ligne du haut) avec l'interpolation linéaire (ligne du bas). A droite : évolution de la longueur de la courbe au cours de l'interpolaton. Polygones de 512 points.

8.2.2 Conservation des détails

Notre méthode permet de créer de nouvelles formes qui conservent les caractéristiques des formes données en entrée.

Sur la Figure 8.4, nous avons deux formes avec de nombreux détails en entrée (en rouge), ces courbes ne représentent rien de particulier. Les dix courbes intermédiaires obtenues par interpolation sont représentées en bleu, de gauche à droite, de façon similaire à l'affichage que nous avions choisi pour les courbes planaires (voir 4.2). Avec l'interpolation linéaire (lignes du bas), les courbes intermédiaires sont de plus en plus lisses et ne comportent pas de détails (les détails disparaissent puis réapparaissent au cours de l'interpolation). Par contre, avec notre morphing MR (lignes du haut), les courbes intermédiaires conservent les caractéristiques des courbes données en entrée, elles présentent elles aussi de nombreux détails.

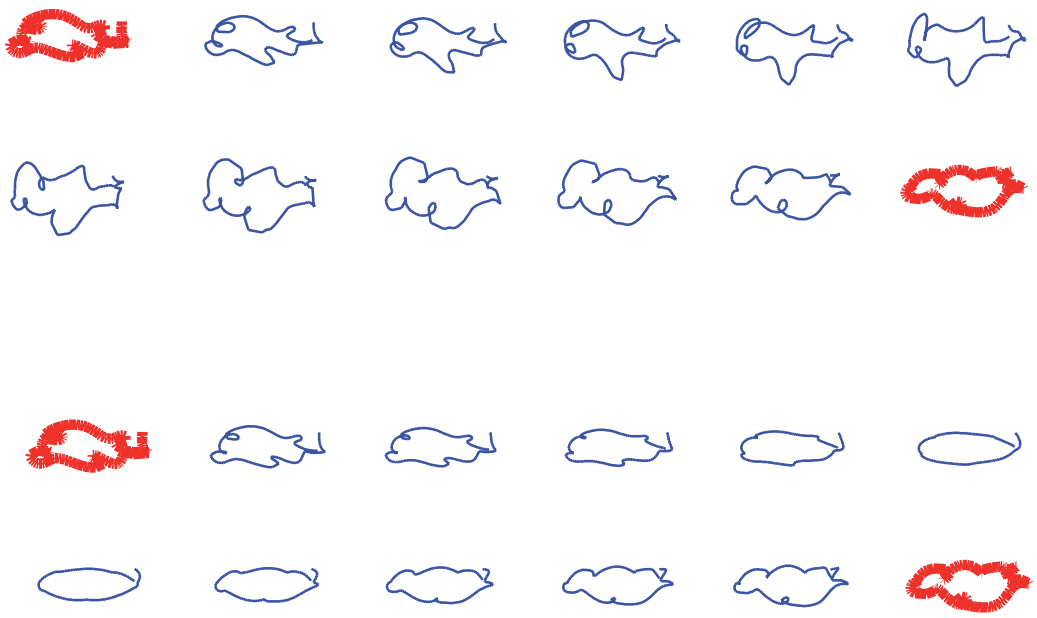


FIGURE 8.4 – Comparaison du morphing MR avec 6 niveaux de résolution (lignes du haut) avec l'interpolation linéaire (lignes du bas). Polygones de 256 points.

8.2.3 Un exemple compliqué

Nous démontrons la robustesse de notre méthode en interpolant des courbes comprenant de nombreux sommets et de nombreux détails. L'interpolation MR se comporte bien même dans

ces cas délicats, voir la Figure 8.5 pour une illustration. Dans cet exemple, 10 étapes d'analyse sont appliquées sur les polygones initiaux de 2048 points. Nous interpolons les représentations intrinsèques MR de nos polygones source et destination. Ces représentations sont composées de polygones grossiers à 4 sommets (l, θ, φ) et de 2044 ensembles de coefficients de détail (α, β, γ).

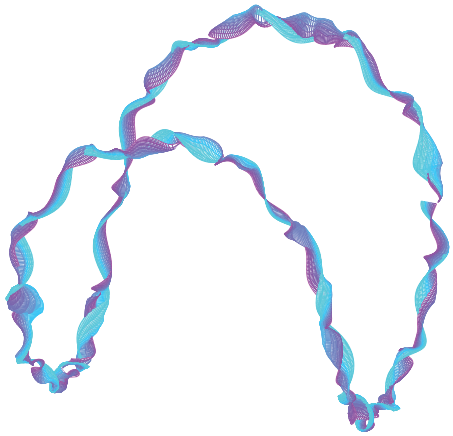


FIGURE 8.5 – Morphing MR avec 10 niveaux de résolution. Polygones de 2048 points.

8.2.4 Comparaison avec/sans MR

La méthode de Sederberg consiste en une simple interpolation linéaire entre les représentations longueur-angles des polygones source et destination (sans MR). Nous avions déjà vu que cette méthode provoque des problèmes de divergence avec les courbes planaires quand celles-ci comportent beaucoup de détails (voir 4.2.4). En 3D, ces problèmes interviennent même pour des polygones peu détaillés, comme illustré dans la Figure 8.6 (ligne du milieu). Avec cet exemple, nous voyons bien la divergence puisque à partir de courbes source et destination fermées, les courbes interpolées obtenues ne sont pas fermées. Pour pallier cette divergence, nous pouvons appliquer une fermeture affine sur les courbes interpolées, comme illustré dans la dernière ligne.

Cependant, le résultat n'est pas satisfaisant. Les courbes intermédiaires sont fermées, mais ne ressemblent pas aux courbes données en entrée, le morphing n'est pas naturel. Par contre, avec notre morphing MR (ligne du haut), nous ne rencontrons pas ce problème de non-fermeture, les courbes intermédiaires obtenues sont toutes fermées et l'interpolation est naturelle. Il nous suffit en effet d'appliquer une fermeture affine sur les polygones grossiers intermédiaires (à 4 sommets ici) pour que les courbes détaillées reconstruites soit elles aussi automatiquement fermées.

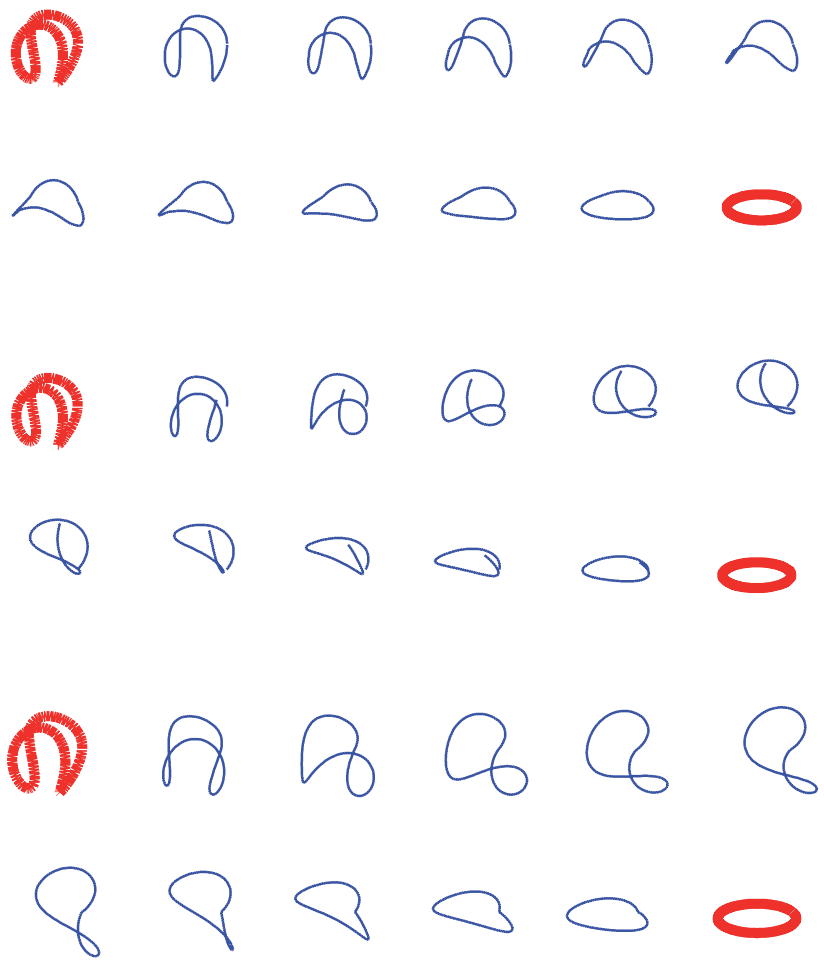


FIGURE 8.6 – Comparaison du morphing MR avec 6 niveaux de résolution (lignes du haut) avec l'interpolation de Sederberg (lignes du milieu), et Sederberg avec fermeture affine (lignes du bas).

Conclusion

Nous présentons un algorithme de morphing multirésolution de courbes 3D qui possède les même propriétés que notre algorithme pour les courbes 2D. Grâce à notre représentation intrinsèque des formes grossières, basée sur les coordonnées sphériques locales, ainsi que des coefficients de détail, notre morphing est rigide et en particulier les détails sont bien conservés au cours de l'interpolation. Nos formules d'analyse et de synthèse ne nécessitent pas de résolution de système, les coefficients sont tous exprimés directement dans notre représentation intrinsèque. Enfin, nous avons démontré l'intérêt de notre algorithme de morphing MR pour l'interpolation de formes détaillées, ainsi que sa robustesse sur des exemples comportant de nombreux sommets et de nombreux détails.

Quatrième partie

Bilan et perspectives

Chapitre 9

Bilan et perspectives

Dans cette thèse, nous nous sommes intéressés au problème de la trajectoire des sommets au cours du morphing de courbes, et plus particulièrement au comportement des détails fins au cours de cette transformation. Dans cet objectif, nous avons proposé une représentation multirésolution, ce qui permet de traiter séparément la forme grossière et les détails, et donc de mieux contrôler l'interpolation des détails. De plus, afin d'assurer un comportement rigide à notre morphing, nous avons choisi d'interpoler une représentation intrinsèque de nos courbes, basée sur les longueurs et angles.

Nous avons introduit une nouvelle représentation intrinsèque MR de courbes 2D, basée sur les longueurs et angles, puis de courbes 3D, basée sur des coordonnées sphériques locales. Nous avons ensuite utilisé ces nouvelles représentations pour développer nos algorithmes de morphing MR de courbes 2D, puis 3D. Les algorithmes MR que nous avons développés permettent une bonne conservation des détails au cours de l'interpolation. De plus, nous avons constaté que les auto-intersections, bien que possibles en théorie, sont rares en pratique grâce aux représentations intrinsèques MR que nous avons choisies. La particularité de nos algorithmes de morphing MR est qu'ils peuvent, à la différence des autres méthodes de l'état de l'art, travailler avec des modèles comportant de nombreux détails fins.

Perspectives

Comme nous l'avons rappelé dans l'introduction (voir section 1.2.1), la première phase du morphing – *l'appariement des courbes* – est difficile à automatiser, du fait de sa forte dépendance à l'application visée et de sa dimension esthétique. Nous pourrions donc proposer une méthode d'appariement des sommets intuitive, en s'appuyant sur notre représentation multirésolution de courbes. Notre représentation multirésolution permet d'imaginer un outil permettant à l'utilisa-

teur de définir des points d'appariement à différents niveaux de résolution. L'utlisateur pourrait ainsi fixer les points d'appariement grossiers, puis de plus en plus fin selon ses besoins. Il pourrait aussi bien fixer la forme globale (le museau du hérisson et de l'hippocampe, le bout de leur queue), qu'apparier des détails fins lui semblant importants (les pics du herisson avec les pics de l'hippocampe par exemple). Il suffirait ensuite d'appliquer à nos courbes un sur-échantillonnage adapté pour prendre en compte ces contraintes utilisateur.

Le principe de notre algorithme de morphing MR de courbes pourrait être étendu aux surfaces. Toutefois, différentes difficultés se présentent. Le passage des courbes 2D aux courbes 3D a déjà beaucoup complexifié nos calculs. Le passage aux surfaces serait encore plus difficile à réaliser. Nous avons cependant des idées de pistes à explorer, et avons identifié les difficultés qui seraient rencontrées dans la mise en oeuvre d'une méthode de morphing multirésolution de surfaces maillées.

Pour pouvoir développer une représentation MR d'une surface discrète, les maillages triangulaires sont les plus commodes. La première difficulté, liée au choix de travailler en multirésolution, est qu'il faut que la structure du maillage permette de passer simplement d'un niveau de résolution à l'autre. Pour cela, nous pourrions utiliser un *maillage semi-régulier*, c'est-à-dire un maillage avec une structure topologique particulière, où presque tous les sommets ont six voisins. Un maillage semi-régulier peut être obtenu à partir d'un maillage quelconque en utilisant un algorithme de remaillage, par exemple la reparamétrisation MAPS [LSS+98]. L'intérêt d'utiliser un maillage semi-régulier est que sa topologie se prête bien à une représentation multirésolution. Pour passer d'un niveau de résolution grossier vers un niveau de résolution plus fin, chaque triangle est subdivisé en quatre nouveaux triangles, les nouveaux sommets sont situés sur les arêtes du triangle grossier. Pour passer d'un niveau fin à un niveau grossier, le processus inverse est appliqué, en fusionnant les triangles quatre par quatre. La Figure 9.1, extraite de [SHB07], illustre les étapes d'analyse et se synthèse pour un maillage semi-régulier.

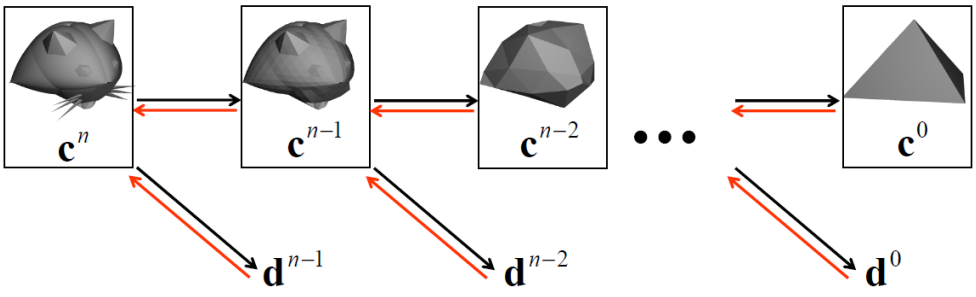


FIGURE 9.1 – Algorithme banc de filtres pour les maillages semi-réguliers. Flèches noires : analyse, flèches rouge : synthèse.

Se pose ensuite le problème de la préparation des données, de façon similaire à ce que nous avons évoqué pour les courbes dans la section 4.2.1. De la même façon que pour les courbes, il faut que nos maillages soient topologiquement identiques (triangulations compatibles), en plus d'être semi-réguliers. De plus, pour pouvoir contrôler les sommets mis en correspondance, nous avons besoin de pouvoir générer nos triangulations compatibles en respectant les contraintes d'appariement choisies par l'utilisateur. A notre connaissance, aucun algorithme n'a été développé à ce jour permettant de traiter ces trois contraintes à la fois : à partir de deux maillages quelconques, générer des maillages semi-réguliers, de même topologie, en intégrant des contraintes de mise en correspondance de sommets.

Puis, il nous faut choisir une représentation pour nos maillages. En étudiant les différentes représentations de surfaces récemment développées pour la déformation de surfaces maillées, nous en dégageons deux sur lesquelles nous pourrions nous baser pour développer une représentation intrinsèque MR de maillages : les *coordonnées pyramide* de Sheffer et al. [SK04b], et les *coordonnées invariantes par rotation* de Lipman et al. [LSLCO05]. Ces deux systèmes de coordonnées ont pour point commun de représenter un maillage par des quantités intrinsèques. Un maillage est défini par un ensemble de longueurs et d'angles représentant la position du 1-voisinage de chaque sommet par rapport à celui-ci. Les *coordonnées pyramide* nécessitent la résolution d'un système non-linéaire pour revenir aux coordonnées cartésiennes, par contre les *coordonnées invariantes par rotation* nécessitent seulement la résolution de deux systèmes linéaires.

Une des difficultés les plus importantes serait enfin d'exprimer les formules d'analyse MR en fonction seulement des coordonnées intrinsèques choisies. Nous avons vu dans notre travail sur les courbes 3D que ces formules ne sont pas triviales à obtenir, la difficulté serait encore augmentée en travaillant avec des surfaces maillées. Nous pourrions donc essayer de développer ces formules en nous basant sur les deux représentations intrinsèques pointées ci-dessus, et choisir celle avec laquelle les calculs seraient les plus simples. Similairement à nos algorithmes de morphing MR de courbes, une fois la représentation MR de chaque maillage calculée, il ne nous resterait plus qu'à les interpoler linéairement pour obtenir notre morphing intrinsèque MR de maillages.

En appliquant un algorithme tel que nous l'esquissons ici, nous pourrions espérer les mêmes avantages que pour nos algorithmes de morphing MR de courbes. Le morphing obtenu serait rigide, les détails seraient bien préservés au cours de l'interpolation, de façon analogue à nos algorithmes de morphing MR de courbes 2D et 3D. Rappelons toutefois que rien ne nous garantit que cette idée est possible à mettre en oeuvre, notamment à cause de la préparation des données qui semble difficile à mettre en oeuvre.

Bibliographie

[ACOL00] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[Ada75] J. Alan Adams. The intrinsic method for curve definition. *Computer Aided Design*, 7(4) :243–249, 1975.

[Ale02] Marc Alexa. Recent advances in mesh morphing. *Comput. Graph. Forum*, 21(2) :173–196, 2002.

[Arf94] G. Arfken. Spherical polar coordinates. In *Mathematical Methods for Physicists*, pages 102–111. Academic Press, 1994.

[BCGJ95] Ronen Basri, Luiz Costa, Davi Geiger, and David Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38 :135–143, 1995.

[Ber05] Martin Bertram. Single-knot wavelets for non-uniform b-splines. *Computer Aided Geometric Design*, 22(9) :848–864, 2005.

[Car76] M.P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[Cha74] George Merrill Chaikin. An algorithm for high-speed curve generation. *Computer Graphics and Image Processing*, 3(4) :346 – 349, 1974.

[CMPS97] P. Cignoni, C. Montani, E. Puppo, and R. Scopigno. Multiresolution representation and visualization of volume data. *IEEE Trans. on Visualization and Comp. Graph*, 3(4) :352–369, 1997.

[CQ92] C. Chui and E. Quak. Wavelets on a bounded interval. In D. Braess and L. Schumaker, editors, *Numerical Methods of Approximation Theory*, pages 1–24. Birkhäuser Verlag, Basel, 1992.

[DBP08] Francçois Duranleau, Philippe Beaudoin, and Pierre Poulin. Multiresolution point-set surfaces. In *GI '08 : Proceedings of graphics interface 2008*, pages 211–218. Canadian Information Processing Society, 2008.

[EDD+95] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, T. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Computer Graphics Proceedings (SIGGRAPH 95)*, pages 173–182, 1995.

[EG95] G. Elber and C. Gotsman. Multiresolution control for nonuniform bspline curve editing. In *The Third Pacific Graphics Conference on Computer Graphics and Applications, Seoul, Korea*, pages 267–278, [August 1995.

[FB88] D. Forsey and R. Bartels. Hierarchical b-spline refinement. *Proceedings of SIGGRAPH'88, ACM New York*, pages 205–212, 1988.

[FG99] Michael S. Floater and Craig Gotsman. How to morph tilings injectively. *J. Comput. Appl. Math.*, 101(1-2) :117–129, 1999.

[FS94] A. Finkelstein and D. H. Salesin. Multiresolution curves. *Computer Graphics Proceedings (SIGGRAPH 94)*, pages 261–268, 1994.

[GG95] E. Goldstein and Craig Gotsman. Polygon morphing using a multiresolution representation. In *Graphics Interface '95*, pages 247–254. Canadian Inf. Process. Soc., 1995.

[GVSS00] Igor Guskov, Kiril Vidimče, Wim Sweldens, and Peter Schröder. Normal meshes. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 95–102, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[HBCC07] S. Hahmann, G.-P. Bonneau, B. Caramiaux, and M. Cornillac. Multiresolution morphing for planar curves. *Computing*, 79(2) :197–209, 2007.

[HBS05] S. Hahmann, G.-P. Bonneau, and B. Sauvage. Area preserving deformation of multiresolution curves. *Computer Aided Geometric Design*, 22(4) :249–267, 2005.

[KE97] R. Kazinnik and G. Elber. Orthogonal decomposition of non-uniform bspline spaces using wavelets. *Computer Graphics forum*, 16(3) :27–38, September 1997.

[KMP07] Martin Kilian, Niloy J. Mitra, and Helmut Pottmann. Geometric modeling in shape space. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 64, New York, NY, USA, 2007. ACM.

[LDSS99a] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. *Computer Graphics Proceedings (SIGGRAPH 99)*, pages 343–350, 1999.

[LDSS99b] Aaron Lee, David Dobkin, Wim Sweldens, and Peter Schröder. Multiresolution mesh morphing. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 343–350, Los Angeles, 1999. Addison Wesley Longman.

[LM92] T. Lyche and K. Morken. Spline wavelets of minimal support. In D. Braess and L. Schumaker, editors, *Numerical Methods of Approximation Theory*, pages 177–194. Birkhäuser Verlag, Basel, 1992.

[LRW97] M. Lounsbery, T. De Rose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transaction on Graphics*, 16(1) :34–73, 1997.

[LSLCO05] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or. Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph.*, 24(3) :479–487, 2005.

[LSS+98] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. Maps : Multiresolution adaptive parameterization of surfaces. *Computer Graphics Proceedings (SIGGRAPH 98)*, pages 95–104, 1998.

[LV98] Francis Lazarus and Anne Verroust. Three-dimensional metamorphosis : a survey. *The Visual Computer*, 14 (8/9) :373–389, 1998.

[LWZ+04] Ligang Liu, Guopu Wang, Bo Zhang, Baining Guo, and Heung-Yeung Shum. Perceptually based approach for planar shape morphing. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, PG '04, pages 111–120, Washington, DC, USA, 2004. IEEE Computer Society.

[Mal89] S. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :674–693, 1989.

[MKFC01] Takashi Michikawa, Takashi Kanai, Masahiro Fujita, and Hiroaki Chiyokura. Multiresolution interpolation meshes. In *PG '01 : Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 60, Washington, DC, USA, 2001. IEEE Computer Society.

[MS01] Michela Mortara and Michela Spagnuolo. Similarity measures for blending polygonal shapes. *Computers & Graphics*, 25(1) :13–27, 2001.

[RE00] A. Raviv and G. Elber. Three dimensional freeform sculpting via zero sets of scalar trivariate functions. *CAD*, 32(8/9) :513–526, July/August 2000.

[RM95] Detlef Ruprecht and Heinrich Müller. Image warping with scattered data interpolation. *IEEE Comput. Graph. Appl.*, 15 :37–43, March 1995.

[SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, pages 109–116, 2007.

[Sau05] Basile Sauvage. *Déformation de courbes et surfaces multirésolution sous contraintes*. PhD thesis, Institut National Polytechnique de Grenoble, December 2005.

[ScH00] Detlev Stalling and Hans christian Hege. Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, 16 :241–253, 2000.

[SD92] Ken Shoemake and Tom Duff. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface '92*, pages 258–264, 1992.

[SDS96] E. Stollnitz, T. DeRose, and D. Salesin. *Wavelets for Computer Graphics : Theory and Applications*. Morgan-Kaufmann, 1996.

[SG92] Thomas W. Sederberg and Eugene Greenwood. A physical based approach to 2-d shape bending. *Computer Graphics,(SIGGRAPH 92 Proceedings)*, 26(2) :25–34, 1992.

[SG03] Vitaly Surazhsky and Craig Gotsman. Intrinsic morphing of compatible triangulations. *International Journal of Shape Modeling*, 9(2) :191–201, 2003.

[SGWM93] T.W. Sederberg, P. Gao, G. Wang, and H. Mu. 2-d shape blending : An intrinsic solution to the vertex path problem. *Computer Graphics,(SIGGRAPH 93 Proceedings)*, 27 :15–18, 1993.

[SHB07] Basile Sauvage, Stefanie Hahmann, and Georges-Pierre Bonneau. Volume preservation of multiresolution meshes. *Comput. Graph. Forum*, 26(3) :275–283, September 2007. Special Issue : Eurographics'2007.

[SK04a] Alla Sheffer and Vladislav Kraevoy. Pyramid coordinates for morphing and deformation. In *3DPVT '04 : Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 68–75, Washington, DC, USA, 2004. IEEE Computer Society.

[SK04b] Alla Sheffer and Vladislav Kraevoy. Pyramid coordinates for morphing and deformation. In *3DPVT '04 : Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 68–75, Washington, DC, USA, 2004. IEEE Computer Society.

[Sor06] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4) :789–807, 2006.

[SR95] Michal Shapira and Ari Rappoport. Shape blending using the star-skeleton representation. *IEEE Comput. Graph. Appl.*, 15(2) :44–50, 1995.

[SS95] P. Schröder and W. Sweldens. Spherical wavelets : Efficiently representing functions on the sphere. *Computer Graphics Proceedings (SIGGRAPH 95)*, pages 161–172, 1995.

[Swe97] W. Sweldens. The lifting scheme : A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2) :511–546, 1997.

[TJ95] F. Thomas and O. Johnston. *The Illusion of Life : Disney Animation*. Disney Editions, revised edition, 1995.

[Wol98] George Wolberg. Image morphing : a survey. *The Visual Computer*, 14 :360–372, 1998.

[WR09] Brian Whited and Jarek Rossignac. b-morphs between b-compatible curves in the plane. In *Symposium on Solid and Physical Modeling*, pages 187–198, 2009.

[XZWB05] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. Poisson shape interpolation. In *SPM '05 : Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 267–274, New York, NY, USA, 2005. ACM.

[Zha96] Yuefeng Zhang. A fuzzy approach to digital image warping. *IEEE Comput. Graph. Appl.*, 16 :34–41, July 1996.

[ZSS97] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. *Computer Graphics Proceedings (SIGGRAPH 97)*, pages 259–268, 1997.

[ZYC05] Dian-Hua Zhang, Tian Yue, and Yong-Hui Chen. Multi-scale surface representation of point-sampled geometry. In *ISPA'05 : Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 371– 376. IEEE, 2005.

Morphing multirésolution de courbes

Le *morphing* (ou métamorphose) est la transformation progressive et lisse d'un modèle en un autre par interpolation. Ce modèle peut être en dimension 2 une image ou une courbe, ou en 3D une surface ou un volume. Le problème est de créer une transition entre deux formes qui soit esthétique et intuitive. Les formes intermédiaires doivent préserver l'apparence et les propriétés des formes en entrée. Le processus de *morphing* se décompose en deux problèmes : *le problème du couplage des sommets* (trouver une correspondance entre les caractéristiques géométriques des objets source et destination) et *le problème de la trajectoire des sommets* (trouver la trajectoire suivie par deux éléments correspondants au cours de la métamorphose). Ces deux problèmes suscitent toujours beaucoup d'intérêt en recherche, puisqu'il n'existe pas à ce jour de définition formelle d'une solution satisfaisante.

Dans cette thèse, nous nous intéressons au *problème de la trajectoire des sommets* au cours du morphing. Nous présentons un nouvel algorithme de morphing de courbes utilisant une décomposition multirésolution intrinsèque que nous introduisons. Cette nouvelle représentation multirésolution est basée sur des quantités intrinsèques des courbes polygonales, les longueurs et angles. Elle présente l'avantage que l'orientation des détails suit naturellement n'importe quelle déformation. Le principe du morphing multirésolution est d'interpoler séparément les coefficients grossiers et ceux de détail issus de la décomposition multirésolution. Les polygones intermédiaires se comportent naturellement et leur distorsion est minimale grâce à la représentation multirésolution intrinsèque que nous avons développée. Nous montrons au travers de nombreux exemples la robustesse de notre algorithme sur des polygones de grande taille comportant de nombreux détails.

Nous déclinons notre morphing MR pour les courbes planaires, puis pour les courbes de l'espace.

Mots-clés : Modélisation géométrique, courbes, analyse multirésolution, morphing, interpolation, représentation intrinsèque.

Multiresolution morphing for curves

In computer graphics, morphing (or metamorphosis) is known as the smooth and progressive transformation of one shape into another. The shape can be an image or a planar curve in 2D space, or it can be a surface or a volume in 3D space. The problem is to create an aesthetic and intuitive transition between two shapes. The intermediate shapes should preserve the appearance and the properties of the input shapes. The morphing process consists of solving two problems : the vertex correspondence problem (finding the correspondence between the geometric features of the source and target object) and the vertex path problem (finding the trajectory two corresponding elements follow during the morphing). Both problems still attract much attention in research, since no formal definition of a successful solution exits. In this work we assume that the correspondence is given and only the vertex path problem is to be solved. The morphing method we introduce in this thesis is based on a new intrinsic multiresolution decomposition. This novel multiresolution representation is defined intrinsically by lengths and angles. We show that this intrinsic representation preserves details orientation during deformation.

Multiresolution morphing principle is to interpolate separately coarse and detail coefficients of the multiresolution decomposition. It can be observed in all tests we did, that the morphs behave natural and that the transformations are least-distorting.

We apply this intrinsic multirésolution morphing algorithm principle for both 2D and 3D curves.

Keywords : Geometric modeling, curves, multiresolution analysis, morphing, interpolation, intrinsic representation.