

---

## Déformation de courbes multirésolution sous contraintes

Stefanie Hahmann\* — Georges-Pierre Bonneau\*\* — Basile Sauvage\*

\* Laboratoire LMC-IMAG

BP.53, F-38041 Grenoble cedex 9, France

Stefanie.Hahmann@imag.fr, Basile.Sauvage@imag.fr

\*\* Laboratoire GRAVIR, INRIA Rhône-Alpes

655, avenue de l'Europe, F-38330 Montbonnot, France

Georges-Pierre.Bonneau@imag.fr

---

*RÉSUMÉ.* Nous décrivons ici une méthode de déformation multirésolution de courbes planes fermées qui permet d'assurer la conservation de l'aire délimitée par la courbe. Les courbes vivent dans un espace de dimension finie muni d'une base d'ondelettes. Dans cette structure multirésolution les courbes sont représentées par un nombre fini de points de contrôle à chaque niveau de résolution, que l'on peut modifier pour déformer la courbe à n'importe quel niveau. A ce processus, connu sous le nom d'édition multirésolution, nous ajoutons la contrainte de conservation de l'aire. La contribution principale de cet article est de fournir une méthode de calcul efficace de l'aire à partir de la décomposition en ondelettes. Nous proposons en outre un algorithme de déformation basé sur différentes linéarisations de la contrainte d'aire, initialement quadratique. Ces deux apports nous permettent de déformer interactivement et sur des PC standards des courbes multirésolution complexes.

*ABSTRACT.* We describe a method for multiresolution deformation of closed planar curves that keeps the enclosed area constant. We use a wavelet based multiresolution representation of the curves that are represented by a finite number of control points at each level of resolution. A deformation can then be applied to the curve by modifying one or more control points at any level of resolution. This process is generally known as multiresolution editing to which we add the constraint of constant area. The key contribution of this paper is the efficient computation of the area in the wavelet decomposition form. Furthermore different linearizations of the quadratic area constraint are developed. These two contributions allow a real time multiresolution deformation of complex curves on standard PCs.

*MOTS-CLÉS :* analyse multirésolution, déformation, contrainte d'aire, courbes de subdivision, ondelettes, aire constante, courbes fermées.

*KEYWORDS:* multiresolution analysis, deformation, area constraint, subdivision curves, wavelets, constant area, closed curves.

---

## 1. Introduction

L'analyse multirésolution a reçu une attention particulière ces dernières années dans de nombreux domaines de la modélisation géométrique et de la visualisation. Elle apporte un outil puissant pour représenter efficacement des fonctions à différents niveaux de détail. Dans ce cadre, une fonction est représentée par une "tendance grossière" qui contient l'information basse résolution, couplée avec une série de coefficients de détails qui permettent la reconstruction exacte du signal original.

Des représentations multirésolution basées sur les ondelettes ont été développées pour les courbes paramétriques [CHU 92, FIN 94] et peuvent être généralisées aux surfaces produit tensoriel, aux surfaces de topologie arbitraire [LOU 97], aux données sphériques [SCH 95] et volumiques [CIG 97]. D'autres modélisations multirésolution (MR) existent pour des données définies sur des maillages irréguliers [BON 98] et pour des maillages arbitraires [ZOR 97, KOB 98, ECK 95, HOP 96].

Toutes ces techniques montrent les nombreux intérêts que présente la modélisation multirésolution. La déformation d'objets complexes avec une grande quantité de détails s'avère souvent difficile et coûteuse en calculs. Dans un cadre multirésolution, de tels objets peuvent être édités à une échelle précise, ce qui permet principalement deux effets : premièrement la modification de points de contrôle à basse résolution suivie de la réintégration des détails modifie la forme globale de l'objet sans en changer les détails caractéristiques. Deuxièmement, la modification de détails fins modifie le caractère de la courbe sans changer sa forme globale.

L'édition multirésolution et les techniques de déformations de courbes paramétriques ont été étudiées en détail par Finkelstein et Salesin [FIN 94] avec l'utilisation des B-splines. Mais l'édition multirésolution sous contraintes n'a fait l'objet que de peu de recherches, bien que de nombreux domaines tels que la CAD/CAM ou l'informatique graphique aient besoin de tels outils. Des travaux utilisant l'intégration numérique d'équations différentielles physiques ont été effectués, mais dans les cas où les temps de calculs sont prioritaires sur le réalisme, l'utilisation de contraintes géométriques est un avantage certain.

Nous présentons ici un environnement d'édition et de déformation de courbes multirésolution qui permet la conservation de l'aire délimitée par une courbe plane périodique, qui est une contrainte non linéaire. Des travaux ont déjà été faits dans ce sens par Elber [ELB 01a] concernant la linéarisation de la contrainte d'aire par décomposition du déplacement en de nombreux pas en  $x$  et en  $y$ . Dans [ELB 01b], Elber a mis au point une version multirésolution de cet algorithme, qui permet à l'utilisateur d'éditer la courbe à une échelle quelconque. Tous les détails plus fins que le niveau d'édition sont conservés. Contrairement à ce que nous présentons ici, l'algorithme de [ELB 01b] n'est pas basé sur une analyse multi-échelle de l'information, et l'échelle d'édition est déterminée par le support de la déformation. Dans notre cadre, la courbe est d'abord décomposée sous une forme multirésolution jusqu'à une échelle adaptée à la manipulation, puis elle est éditée par modification de cette décomposition à une ou plusieurs échelles, sur un voisinage quelconque. Dès lors nous pouvons bénéficier de

tous les avantages de la décomposition multirésolution, comme la compression, l’affichage avec niveaux de détail (LOD pour *Level of Detail*), la transmission progressive ou l’édition LOD, alors que [ELB 01b] a pour seul objectif l’édition à une échelle donnée.

Les formules que nous présentons peuvent s’appliquer à tous les schémas multirésolution basés sur des ondelettes. Pour la mise en pratique nous utilisons le schéma de subdivision de Chaïkin, identique aux ondelettes B-spline quadratiques. Notons enfin que tous nos algorithmes ont été développés dans un souci constant de simplicité et d’efficacité pour en faire un outil interactif.

Cet article est organisé comme suit : la section 2 rappelle les idées de base de la multirésolution. La section 3 explique comment calculer efficacement l’aire délimitée par une courbe définie dans une base multirésolution. La section 4 présente notre méthode de déformation de courbes sous contrainte d’aire. La section 5 conclut et donne des ouvertures sur des travaux futurs.

## 2. Courbes Multirésolution

Durant les dernières années les ondelettes ont trouvé des applications dans des domaines très variés : analyse numérique [BEY 91], analyse du signal [MER 99], traitement d’images [MAL 89], visualisation et informatique graphique. Le schéma multirésolution basé sur des ondelettes que nous allons présenter brièvement dans cette section est légèrement plus général que celui introduit dans [MAL 89]. Pour plus de détails, voir [STO 96, FIN 94].

Soient  $E$  un espace fonctionnel et  $V^j \subset E$  des *espaces d’approximation* linéaires imbriqués avec  $V^0 \subset V^1 \subset \dots \subset V^n$ . Dans le cadre de cet article nous allons nous intéresser aux courbes fermées, et ces espaces seront de dimension finie. Soit  $V^j$  engendré par un ensemble de fonctions de base  $(\varphi_i^j)_i$ , appelées *fonctions d’échelle*. L’espace  $W^j$ , complémentaire de  $V^j$  dans  $V^{j+1}$ , est appelé *espace de détails*. Ses fonctions de base  $(\psi_i^j)_i$  couplées avec les  $\varphi_i^j$  forment une base de  $V^{j+1}$ . Les fonctions  $\psi_i^j$  sont appelées *ondelettes*. L’espace  $V^n$  peut donc être décomposé comme suit :

$$V^n = V^{n-1} \oplus W^{n-1} = V^{n-2} \bigoplus_{j=n-2}^{n-1} W^j = \dots = V^0 \bigoplus_{j=0}^{n-1} W^j. \quad [1]$$

La condition 1 implique que les fonctions d’échelle admettent un schéma de subdivision, i.e. pour tout  $j \in [1, n]$  il doit exister des matrices  $P^j$  et  $Q^j$  telles que les *équations de subdivision* suivantes sont vérifiées (où  $\varphi^j$  et  $\psi^j$  sont les vecteurs colonne de fonctions de base)

$$\begin{aligned}\varphi^{j-1} &= (P^j)^T \varphi^j \\ \psi^{j-1} &= (Q^j)^T \varphi^j\end{aligned}\quad [2]$$

D'autre part, les fonctions d'échelle "fines"  $\varphi^j$  peuvent être construites à partir des fonctions d'échelles "grossières" et des ondelettes à l'aide de matrices  $A^j$  et  $B^j$  :

$$\varphi^j = (A^j)^T \varphi^{j-1} + (B^j)^T \psi^{j-1}. \quad [3]$$

Notons que  $[P^j \mid Q^j]$  et  $\begin{bmatrix} A^j \\ B^j \end{bmatrix}$  sont des matrices carrées, et que

$$\begin{bmatrix} P^j & Q^j \end{bmatrix} \begin{bmatrix} A^j \\ B^j \end{bmatrix} = I. \quad [4]$$

4 est généralement appelée *condition de reconstruction*. Le choix des fonctions d'échelle détermine la structure des matrices  $P^j$ ,  $Q^j$ ,  $A^j$ , et  $B^j$ . Pour la plupart des applications il est préférable que ces matrices soient creuses, ce qui est vrai pour les ondelettes à support compact.

Soit une *courbe multirésolution*  $c(t)$ ,  $t \in I$  définie comme élément d'un tel espace fonctionnel  $V^n$  de dimension  $D2^n$ , avec  $D = \dim(V^0)$ . À chaque niveau  $j$  du schéma multirésolution les deux sous-espaces  $V^j$  et  $W^j$  seront dans notre cas de même dimension égale à  $D2^j$ .  $c(t)$  peut être représentée comme suit :

$$c(t) = \sum_{i=0}^{D2^n-1} x_i^n \varphi_i^n = (\mathbf{x}^n)^T (\varphi^n), \quad [5]$$

où  $\mathbf{x}^n$  est un vecteur colonne de  $D2^n$  points de contrôle  $x_0^n, \dots, x_{D2^n-1}^n \in \mathbb{R}^2$ . De manière générale le vecteur  $\mathbf{x}^n$  peut être vu comme un signal avec  $D2^n$  échantillons.

Les relations 2 et 3 permettent maintenant de créer un signal basse résolution  $\mathbf{x}^{n-1}$  avec moins d'échantillons en utilisant le filtre *passé bas*  $A^n$  :

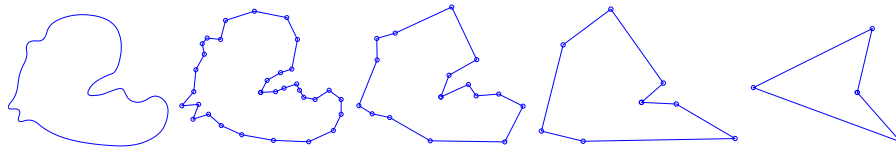
$$\mathbf{x}^{n-1} = A^n \mathbf{x}^n,$$

où  $A^n$  est de taille  $(D2^{n-1} \times D2^n)$ . Les détails qui ont été perdus dans ce filtrage peuvent être récupérés dans un autre signal  $\mathbf{d}^{n-1}$  en utilisant un filtre *passé haut*  $B^n$  :

$$\mathbf{d}^{n-1} = B^n \mathbf{x}^n,$$

où  $B^n$  est une matrice  $(D2^{n-1} \times D2^n)$ . Ce processus de séparation d'un signal  $\mathbf{x}^n$  en un signal grossier  $\mathbf{x}^{n-1}$  et des détails  $\mathbf{d}^{n-1}$  est appelé **décomposition** ou *analyse*. Les matrices  $A^n$  et  $B^n$  s'appellent *filtres de décomposition* (*d'analyse*). La décomposition





**Figure 2.** 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

### 3. Calcul de l'aire délimitée par une courbe multirésolution

Nous allons dans cette section développer un outil de déformation préservant l'aire intérieure d'une courbe multirésolution plane périodique. Pour cela il est nécessaire de pouvoir évaluer l'aire quel que soit le niveau de résolution de la courbe. La section (3.1) présente les formules multirésolution de la fonction d'aire. La section (3.2) explique comment le calcul peut être effectué de manière efficace en développant une relation de récurrence.

#### 3.1. L'aire délimitée par une courbe multirésolution

Soit  $c(t) = (x(t), y(t))$  une courbe paramétrique plane et périodique sans auto-intersections et suffisamment régulière pour que toutes les dérivées nécessaires puissent être définies. L'aire incluse dans  $c(t)$  est donnée par le théorème de Green [ELB 01a, EBE 91] :

$$\mathcal{A} = \frac{1}{2} \oint x(t)y'(t) - x'(t)y(t)dt. \quad [8]$$

Quelques notations sont nécessaires pour la suite. Introduisons la forme bilinéaire

$$I(\varphi, \psi) := \left( \begin{array}{c} I(\varphi_i, \psi_j)_{ij} \end{array} \right), \quad [9]$$

où  $\varphi$  et  $\psi$  sont deux vecteurs de fonctions et  $I(\varphi_i, \psi_j) = \oint \varphi_i(t)\psi_j'(t) - \varphi_i'(t)\psi_j(t)dt$ .

L'aire 8 pour une courbe multirésolution 6 peut maintenant être évaluée directement à n'importe quel niveau de résolution  $L$ . On obtient l'équation bilinéaire

$$2\mathcal{A} = (\mathbf{X}^L) \left[ \begin{array}{c} M^L \end{array} \right] (\mathbf{Y}^L)^T, \quad \forall L \in \{0, \dots, n\}, \quad [10]$$

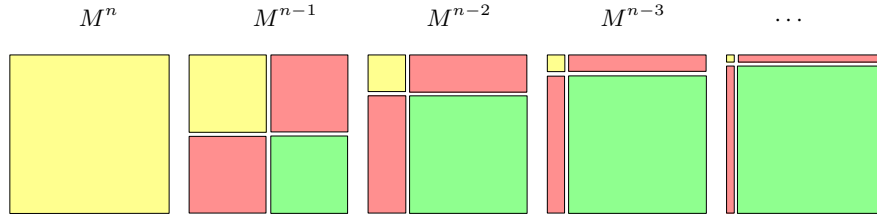
où  $X^L$  et  $Y^L$  sont des vecteurs ligne contenant respectivement les coordonnées  $x$  et  $y$  des  $D2^n$  coefficients de la représentation multirésolution de la courbe, i.e.

$$\begin{pmatrix} X^L \\ Y^L \end{pmatrix} = (x^L, d^L, d^{L+1}, \dots, d^{n-1}),$$

et

$$M^L = \begin{bmatrix} I(\varphi^L, \varphi^L) & I(\varphi^L, \psi^l)_{l=L}^{n-1} \\ I(\psi^k, \varphi^L)_{k=L}^{n-1} & I(\psi^k, \psi^l)_{k,l=L}^{n-1} \end{bmatrix} = \begin{bmatrix} A & B \\ -B^T & C \end{bmatrix}.$$

La matrice d'aire  $M^L$  est une matrice anti-symétrique de taille  $D2^n \times D2^n$  contenant 4 blocs principaux. Le bloc  $A$  (jaune en fig. 3) est de taille  $D2^L \times D2^L$ , le bloc  $B$  (rouge) est de taille  $D2^L \times (D2^n - D2^L)$ , et le bloc  $C$  (vert) est de taille  $(D2^n - D2^L) \times (D2^n - D2^L)$ , voir fig. 3.



**Figure 3.** Les matrices d'aire  $M^L$  pour  $L \in \{0, \dots, n\}$  ont la même taille  $D2^n \times D2^n$  mais leur décomposition en quatre blocs est différente selon le niveau de résolution  $L$

### 3.2. Calcul efficace des matrices d'aire

Les matrices d'aire doivent être pré-calculées pour tous les niveaux de résolution. Chaque élément de ces matrices est l'intégrale d'un produit de fonctions 8. Nous allons montrer dans cette section que dans le cas de courbes multirésolution, *seule* l'évaluation des éléments de la matrice "mère"  $M^n$  est nécessaire. Par bilinéarité de la forme 9, les matrices des niveaux  $n - 1, \dots, L$  peuvent ensuite être obtenues en utilisant récursivement les équations de décomposition 2.

Dans le cas où un argument de la forme bilinéaire  $I$  reste fixe (indice  $k$ ), on obtient les relations récursives suivantes. Nous les appellerons filtres (P) et (Q) en fonction du filtre de décomposition utilisé :

$$\text{(P) - filter :} \quad I(\varphi^{L-1}, \psi^k) = (P^L)^T I(\varphi^L, \psi^k)$$

$$\text{(Q) - filter :} \quad I(\psi^{L-1}, \psi^k) = (Q^L)^T I(\varphi^L, \psi^k).$$

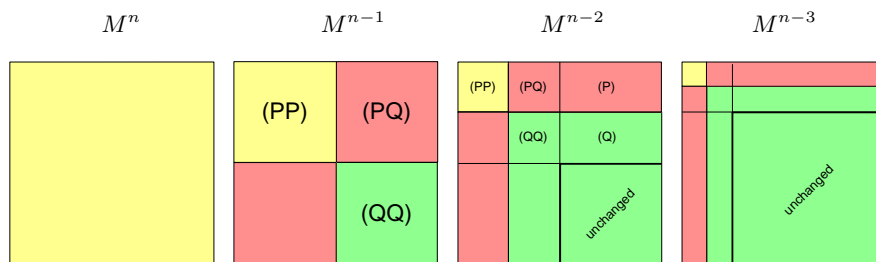
Par symétrie on obtient des formules analogues dans le cas où le premier argument est fixe. Dans le cas où les deux arguments ont besoin d'être calculés de manière récursive, les équations de décomposition seront utilisées sous forme produit tensoriel :

$$\text{(PP) - filter : } I(\varphi^{L-1}, \varphi^{L-1}) = (P^L)^T I(\varphi^L, \varphi^L) (P^L)$$

$$\text{(PQ) - filter : } I(\varphi^{L-1}, \psi^{L-1}) = (P^L)^T I(\varphi^L, \varphi^L) (Q^L)$$

$$\text{(QQ) - filter : } I(\psi^{L-1}, \psi^{L-1}) = (Q^L)^T I(\varphi^L, \varphi^L) (Q^L)$$

L'algorithme qui découle de ces formules, pour calculer successivement les matrices  $M^{n-1}, M^{n-2}, \dots, M^L$ , est schématisé en fig. 4. Par exemple, au départ on applique les filtres (PP), (PQ) et (QQ) à la matrice  $M^n$  pour obtenir les quatre blocs de la matrice symétrique  $M^{n-1}$ .



**Figure 4.** Calcul récursif de la matrice d'aire :  $M^{n-1}$  à partir de  $M^n$ ,  $M^{n-2}$  à partir de  $M^{n-1}$  etc.

#### 4. Déformation multirésolution sous contrainte d'aire

Des contraintes linéaires peuvent facilement être incorporées dans un système de design interactif de courbes et surfaces NURBS, e.g. positions, tangentes, orthogonalité. Des contraintes non-linéaires comme l'aire, le volume, la convexité ou des contraintes d'origine physique sont difficiles à traiter, en particulier à cause d'un coût numérique considérable.

Dans cette section, nous présentons une méthode de déformation de courbes multirésolution planes et fermées à n'importe quel niveau de résolution tout en préservant l'aire intérieure. De manière interactive il va être possible de :

- préserver les détails en modifiant globalement la forme de la courbe, et
  - déformer la courbe à différents niveaux de résolution
- tout en conservant l'aire délimitée par la courbe.

#### 4.1. Principe

La méthode de déformation est composée de trois étapes principales :

– **décomposition** : La courbe est décomposée jusqu'à une échelle qui a été définie par l'utilisateur. La courbe est alors représentée par un polygone de contrôle et un ensemble de coefficients de détail (voir formule 6).

– **déformation** : L'utilisateur définit une déformation en déplaçant un point du polygone de contrôle.

– **conservation d'aire** : A chaque mouvement de souris, une nouvelle courbe est calculée et affichée suivant la déformation définie par l'utilisateur et en préservant l'aire intérieure de la courbe initiale. Cette étape constitue le problème que nous nous proposons de résoudre.

Etant donné une courbe multirésolution  $c(t)$  et un niveau de décomposition  $L$  auquel est appliquée la déformation, prenons les notations suivantes, en négligeant les indices  $L$  et  $n$  :

- $\mathcal{A}_{ref}$  est l'aire de référence que nous voulons conserver,
- $\begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$  sont les coordonnées après déformation du polygone de contrôle, et avant les corrections pour conserver l'aire,
- $\mathcal{A}_0 = \frac{1}{2}X_0MY_0^T$  est l'aire correspondante qui doit être corrigée,
- $\begin{pmatrix} X \\ Y \end{pmatrix}$  sont les coordonnées après correction,
- $\mathcal{A} = \frac{1}{2}XMY^T$  est la nouvelle aire, telle que  $\mathcal{A} = \mathcal{A}_{ref}$ .

Dans ces termes, nous pouvons reformuler les trois étapes :

- exprimer  $c(t)$  dans une base multirésolution de  $V^n$ ,
- déformer le polygone de contrôle pour obtenir  $X_0$  et  $Y_0$ ,
- calculer  $X$  et  $Y$  "proches de"  $X_0$  et  $Y_0$  tels que  $\mathcal{A} = \mathcal{A}_{ref}$ .

#### 4.2. Critère de lissage

Pour obtenir une courbe "lisse", ayant une forme visuelle plaisante, la minimisation d'une version linéarisée de l'énergie de tension, classique en design variationnel, est utilisée :

$$\mathcal{E} = \int |c''(t)|^2 dt = \int x''(t)^2 + y''(t)^2 dt.$$

Dans le cadre particulier d'une courbe multirésolution, nous pouvons l'exprimer de façon similaire à la formule 10 pour un niveau  $L \in \{0, \dots, n\}$  :

$$\mathcal{E}(X, Y) = \frac{1}{2}X^L H^L (X^L)^T + \frac{1}{2}Y^L H^L (Y^L)^T$$

où  $H^L$  est une matrice symétrique calculée par le même processus que la matrice  $M^L$ .

### 4.3. Conservation de l'aire

Afin d'assurer que  $X$  et  $Y$  seront proches de  $X_0$  de  $Y_0$ , on introduit un critère de distance, et notre problème se reformule ainsi en un problème d'optimisation convexe sous contrainte :

$$\begin{aligned} \min_{X,Y} \quad & (1 - \alpha)\left(\frac{1}{2}XHX^T + \frac{1}{2}YHY^T\right) + \alpha(\|X - X_0\|^2 + \|Y - Y_0\|^2) \\ \text{sous} \quad & \mathcal{A} - \mathcal{A}_{ref} = 0. \end{aligned} \tag{11}$$

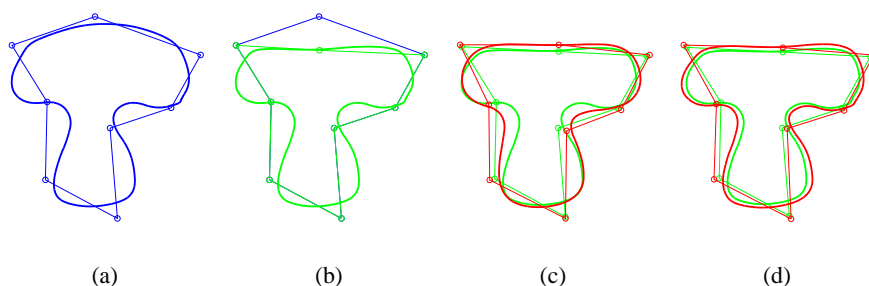
$0 \leq \alpha \leq 1$  est un scalaire qui permet d'équilibrer entre une courbe plus lisse ou une courbe qui respecte mieux la déformation définie par l'utilisateur.

Pour résoudre ce problème, nous pouvons soit adapter une méthode de Elber [ELB 01a] au système 11, soit utiliser une linéarisation de la contrainte d'aire et les multiplicateurs de Lagrange, ce qui nous ramène à trouver le point stationnaire d'une fonction quadratique, c'est à dire à résoudre un système linéaire. Le système est codé sous forme de matrices creuses et résolu itérativement par la méthode du gradient bi-conjugué (à l'aide des bibliothèques SparseLib++ et Impl++). Pour des courbes de l'ordre de 1000 points, la résolution du système (de taille  $2000 \times 2000$ ) nécessite quelques millisecondes, comparé à quoi le temps CPU utilisé pour l'interface et pour les étapes d'analyses et de synthèse est faible. Cela nous permet d'avoir un outil d'édition intégrant la contrainte en temps réel.

### 4.4. Exemples

Dans toutes les illustrations nous utiliserons un code de couleurs qui nous permettra d'identifier la courbe initiale (en bleu), la courbe déformée (en vert), et la courbe finale vérifiant la contrainte d'aire (en rouge). Les polygones de contrôle présents sur la plupart des figures sont les coefficients d'échelle au niveau  $L$ , donc les polygones que peut manier l'utilisateur.

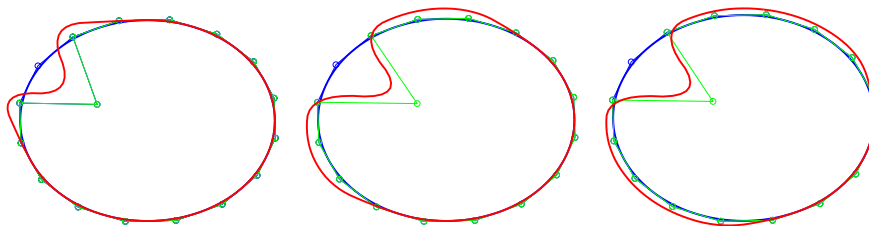
La figure 5 illustre les 3 étapes de notre algorithme, et montre le rôle du coefficient  $\alpha$  dans la formule 11. La courbe originale a 32 points de contrôle ( $n = 5$ ). Après deux décompositions on obtient 8 coefficients d'échelle qui forment le polygone de contrôle au niveau  $L = 3$  et 24 coefficients de détail. La déformation est définie en déplaçant un point de contrôle, ce qui, grâce à la représentation multirésolution, déforme la courbe sur une large portion. La reconstruction à ce moment donnerait une courbe qui ne vérifie pas la contrainte d'aire. La courbe finale est calculée par la formule 11 dans son expression multirésolution puis reconstruite. Nous constatons que le choix d'une petite valeur pour  $\alpha$  favorise une courbe régulière mais qui s'éloigne de la courbe déformée (en vert), alors qu'une valeur plus grande donne une courbe plus conforme aux souhaits de l'utilisateur.



**Figure 5.** Influence du coefficient  $\alpha$  (a) courbe initiale et son polygone de contrôle (bleu)  $n=5$ ,  $L=3$  (b) polygone de contrôle déformé et courbe reconstruite sans conservation d'aire (vert)  $L=3$  (c) polygone de contrôle corrigé et courbe vérifiant la contrainte d'aire (rouge)  $\alpha = 0,5$  (d)  $\alpha = 0,99$

Remarquons que cette méthode est une optimisation globale qui peut d'une part s'avérer coûteuse malgré un système creux, et qui d'autre part ne permet pas un contrôle précis du résultat : on voudrait pouvoir déformer la courbe plus ou moins localement, et utiliser pleinement les fonctionnalités de l'analyse multirésolution en préservant les fins détails de la courbe.

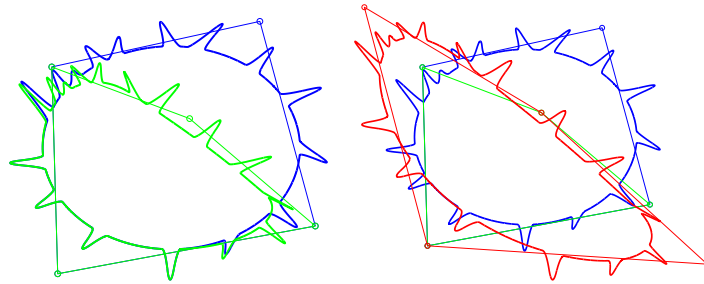
Ce double but est atteint facilement en effectuant la minimisation 11 seulement sur certains coefficients (d'échelle et/ou de détails). La figure 6 illustre une déformation pour laquelle l'aire a été compensée seulement sur les points voisins, pour différentes étendues. On peut ainsi borner la déformation sur un voisinage facile à contrôler.



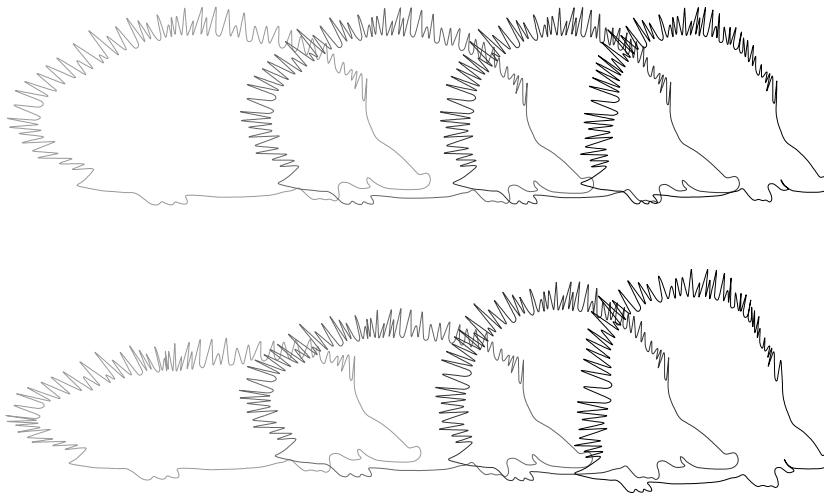
**Figure 6.** Contrôle local :  $n = 7$  et  $L = 4$ ; De gauche à droite : 1, 2 et 5 voisins

Une utilisation très intéressante de la formulation en base multirésolution est mise en valeur par la figure 7 : l'aire est adaptée en ne corrigeant que les deux voisins directs sur le polygone de contrôle, et sans toucher aux détails. On remarque que la forme générale est modifiée pour l'adaptation d'aire, mais que tous les détails qui caractérisent la courbe sont soigneusement conservés. En d'autres termes, l'expression du problème 11 dans la base multirésolution nous permet de corriger l'aire de la courbe avec seule-

ment des coefficients d'échelle, ce qui a le double avantage de permettre un contrôle du résultat et d'alléger considérablement les calculs.



**Figure 7.** déformation multirésolution :  $n = 7$ ,  $L = 2$ ; courbe initiale (bleu), courbe déformée au niveau 2 (vert), courbe finale conservant l'aire (rouge)



**Figure 8.** En haut : déformations sans contrainte; En bas : avec conservation de l'aire

La conservation d'aire est notée comme un effet visuel important pour le réalisme dans les jeux vidéo et les films d'animation [LAS 87]. La figure 8 montre un hérisson allongé puis rétréci avec (en bas) et sans (en haut) l'aire constante. On observe que les images du bas semblent venir du même animal déformé, alors que celles du haut en montrent plusieurs de tailles différentes.

## 5. Conclusion et perspectives

Nous avons présenté une méthode de déformation de courbes planes et périodiques intégrant la contrainte non linéaire d'aire. De plus, nous avons développé une expression multirésolution de la contrainte d'aire pour des courbes exprimées dans des bases d'ondelettes. Cette expression se calcule efficacement par une relation de récurrence qui assure un pré-calcul rapide. Enfin, la méthode d'optimisation employée permet des déformations en temps-réel.

Les travaux actuels concernent l'intégration d'autres contraintes non linéaires dans cet environnement d'édition multirésolution, telle que la conservation de la longueur. Une généralisation aux surfaces est aussi en préparation.

## 6. Bibliographie

- [BEY 91] BEYLKIN G., COIFMAN R. R., ROKHLIN V., « Fast wavelet transforms and numerical algorithms I. », *Comm. Pure and Appl. Math.*, vol. 44, 1991, p. 141-183.
- [BON 98] BONNEAU G.-P., « Multiresolution Analysis on Irregular Surface Meshes », *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, n° 4, 1998, p. 365-378.
- [CHA 74] CHAIKIN G., « An algorithm for high speed curve generation », *Computer Graphics and Image Processing*, vol. 3, 1974, p. 346-349.
- [CHU 92] CHUI C., QUAK E., « Wavelets on a Bounded Interval », BRAESS D., SCHUMAKER L. L., Eds., *Numerical Methods of Approximation Theory*, vol. 9, Birkhäuser Verlag, Basel, 1992, p. 53-75.
- [CIG 97] CIGNONI P., MONTANI C., PUPPO E., SCOPIGNO R., « Multiresolution Representation and Visualization of Volume Data », *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, n° 4, 1997, p. 352-369.
- [EBE 91] EBERLY D., LANCASTER J., « On Gray Scale Image Measurements I. Arc Length And Area », *GMIP*, vol. 53, n° 6, 1991, p. 538-549.
- [ECK 95] ECK M., DEROSE T., DUCHAMP T., HOPPE H., LOUNSBERRY M., STUETZLE W., « Multiresolution Analysis of Arbitrary Meshes », *Computer Graphics*, vol. 29, 1995, p. 173-182.
- [ELB 01a] ELBER G., « Linearizing the area and volume constraints », *Technical Report CIS-2000-04*, 2001.
- [ELB 01b] ELBER G., « Multiresolution curve editing with linear constraints », *6th ACM/IEEE Symposium on Solid Modeling and Applications*, Ann Arbor, Michigan, June 2001, p. 109-119.
- [FIN 94] FINKELSTEIN A., SALESIN D. H., « Multiresolution curves », *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, 1994, p. 261-268.
- [HOP 96] HOPPE H., « Progressive meshes », *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM Press, 1996, p. 99-108.
- [KOB 98] KOBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P., « Interactive Multi-Resolution Modeling on Arbitrary Meshes », *Computer Graphics*, vol. 32, n° Annual Conference Series, 1998, p. 105-114.

- [LAS 87] LASSETER J., « Principles of Traditional Animation Applied to 3D Computer Animation », *Computer Graphics Proceedings (SIGGRAPH 87)*, 1987, p. 35-44.
- [LOU 97] LOUNSBERY M., DEROSE T., WARREN J., « Multiresolution analysis for surfaces of arbitrary topological type », *ACM Transaction on Graphics*, vol. 16, n° 1, 1997, p. 34-73.
- [MAL 89] MALLAT S., « A theory for multiresolution signal decomposition : the wavelet representation », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, 1989, p. 674-693.
- [MER 99] MERTINS A., *Signal Analysis : Wavelets, Filter Banks, Time-Frequency Transforms and Applications*, John Wiley & Sons, Chichester, 1999.
- [SCH 95] SCHRÖDER P., SWELDENS W., « Spherical Wavelets : efficiently representing functions on the sphere », *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, 1995, p. 161-172.
- [STO 96] STOLLNITZ E. J., DEROSE T., SALESIN D. H., *Wavelets for Computer Graphics : Theory and Applications*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1996.
- [ZOR 97] ZORIN D., SCHRÖDER P., SWELDENS W., « Interactive multiresolution mesh editing », *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 1997, p. 259-268.