Modélisation interactive des formes complexes à partir des esquisses

A. Alexe, V. Gaildrat, L. Barthe

Institut de Recherche en Informatique de Toulouse
118, route de Narbonne 31062 Toulouse Cedex

alexe@irit.fr

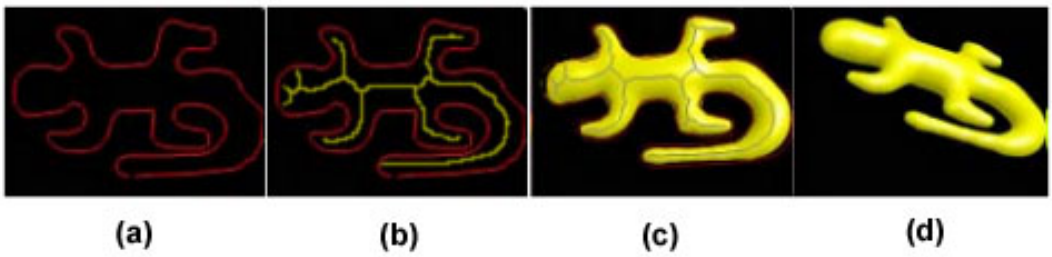

Fig. 1. Forme tridimensionnelle reconstruite avec notre système, à l'aide de son profil esquissé.

Nous présentons une méthode qui permet de reconstruire un objet tridimensionnel à partir d'une esquisse en deux dimensions dessinée par l'utilisateur. La méthode de reconstruction doit pouvoir être utilisée dans la modélisation interactive, et par conséquent elle répond à des critères de rapidité et de précision. La surface produite est lisse et reconstruite à partir d'un squelette extrait d'un contour 2D. Elle a un aspect naturel et peut être retouchée de manière intuitive, en manipulant des parties de son squelette.

Mots clés : modélisation géométrique, esquisse, surfaces implicites.

1.Introduction

Afin de produire un modèle 3D, l'utilisateur part en général d'une représentation mentale de son objet (une idée), avant de réaliser la maquette tridimensionnelle qui reproduit au mieux cet objet. Pendant le processus de conception, son idée se clarifie et prend forme progressivement. Il s'en suit la nécessité d'une longue session de retouches composée de multiples essais. Souvent plusieurs prototypes sont nécessaires afin d'aboutir à la forme finale. Cette forme sera ensuite raffinée dans ses plus petits détails.
Aujourd'hui il existe de nombreuses méthodes pour raffiner les formes tridimensionnelles une fois les premières ébauches produites. Ceci est fait généralement à partir des structures multi-résolution, telles que les surfaces de subdivision [ZSS97], les *Normal Meshes* [GVS00], etc. La recherche a su apporter pour cet aspect des réponses déjà satisfaisantes.
Mais en ce qui concerne la conception, les outils de modélisation actuels offrent peu d'assistance à l'utilisateur.
Ils disposent certes de moyens fortement avancés, qui une fois bien maîtrisés permettent d'aller très loin dans la réalisation réaliste des images de synthèse, mais toutefois ils ont un certain nombre de désavantages qui leurs sont communs:

- la présence de modèles théoriques complexes et non intuitifs (les carreaux paramétriques, la CSG, etc.). L'utilisateur doit passer par un processus fastidieux de familiarisation avec ces techniques qui est absolument nécessaire pour leur utilisation ;
- la manipulation de notions abstraites et loin du monde réel (primitive géométriques, paramètres numériques, fonctions paramétriques, points clés, etc.) ;
- la complexité des interfaces graphiques de type WIMP (*Windows Icons Menus Pointer*), qui sont difficiles à utiliser et souvent complexes, et qui alourdissent la tâche utilisateur. Les experts ont montré [Nor86, Suc87] que l'utilisateur passe beaucoup plus de temps à chercher parmi les commandes disponibles et les menus imbriqués qu'à se concentrer effectivement sur la tâche de modélisation.

Des questions essentielles qui restent encore non résolues sont : Comment réaliser rapidement des prototypes ou des maquettes ? Comment repenser les interfaces actuelles de manière à les rendre intuitives et naturelles ? Et enfin, comment surpasser le niveau trop bas d'abstraction inhérent aux modeleurs classiques ?

L'esquisse est un moyen d'expression simple et à la portée de tous. Elle sert à matérialiser rapidement les idées et à les partager avec les autres membres de l'équipe de travail. Elle est un excellent stimulant pour la créativité et l'innovation. Pour ces raisons, des travaux scientifiques ont essayé de reproduire la création des formes 3D à partir de l'esquisse.
Le premier à imaginer la modélisation à partir des esquisses a été Zeleznik [ZHH96]. Le modeleur proposé, SKETCH, se limite à des objets comportant des angles droits, alignés selon les axes de la scène. L'idée a été développée plus tard par T. Igarashi [IMT99] et son modeleur Teddy a démontré avec succès qu'il est possible de produire des objets de forme libre à partir d'esquisses. Cependant Teddy reste assez limité. Le maillage engendré comporte souvent des artefacts et il est non uniforme. Il nécessite d'être retouché si on veut l'utiliser pour produire une forme raffinée. Un autre inconvénient est l'impossibilité de réarranger les parties de l'objet les unes par rapport aux autres, ce qui signifie que l'objet ne peut pas être retouché de manière simple une fois créé, et de ce fait il ne répond pas entièrement au critère d'édition en conception. Récemment Karpenko et al. [KHR02] ont proposé un système qui reconstruit des objets tridimensionnels à partir d'esquisses en utilisant les surfaces implicites variationnelles. Cependant, ce système fonctionne uniquement avec des primitives en forme d'ellipse, il reste donc assez limité. Un autre désavantage est lié à l'épaisseur de l'objet produit qui est constante et fixée auparavant. [ZS03] reprend la même démarche et enrichit l'objet 3D avec des traits 2D dessinés entièrement sur la surface ou en partie sur la surface et en partie « dans l'air » pour plus d'expressivité de la représentation.
Ces travaux montrent que malgré l'intérêt des scientifiques pour la modélisation à partir d'esquisses, les résultats actuels sont encore loin de fournir des réponses vraiment satisfaisantes. Les prototypes existants demandent des améliorations tant au niveau des formes produites qu'à celui de retouche après création de l'objet.
Quant à la reconstruction de la surface, notre problème peut se poser de la manière suivante : étant donné un ensemble de points approximant un contour 2D (Fig. 1. (a)) trouver une surface 3D fermée qui approche aux mieux cet ensemble de points (Fig. 1. (c)). Formulé ainsi il s'inscrit dans le cas plus générique de la reconstruction à partir d'un nuage de points tridimensionnels. Dans notre cas il s'agit d'un nuage 2D.
Un nombre important des travaux scientifiques ont exploré ce domaine, les premiers étant ceux de Muraki [Mur91] qui emploient des fonctions exponentielles. [TBC95] utilise une méthode semblable mais accélérée qui malheureusement n'est pas automatique. [BTC95] exploite l'axe médian pour une recherche plus efficace des positions des centres des sphères implicites à placer. Enfin une approche plus générique [CBC01] emploie les RBF (Radial Basis Functions) qui sont des fonctions implicites polynomiales. Cette dernière approche, ainsi que les méthodes de reconstruction à partir des surfaces paramétriques nous semblent peu adaptées à la reconstruction d'un ensemble des points en deux dimensions, car il est difficile de gérer l'épaisseur de la surface.
Les autres méthodes peuvent être adaptées à notre étude en deux dimensions, mais elles sont malheureusement trop lentes pour se prêter à une modélisation interactive.
Pour répondre à nos attentes nous nous sommes donc orientés vers une solution différente dont voici les étapes clé.
L'utilisateur saisit d'abord un contour de forme libre non auto-intersectant (à l'aide d'une tablette graphique, par exemple) Fig. 1. (a) . Ce contour est discrétisé et transformé en un polygone fermé avec des arêtes de longueur uniforme. Ensuite on extrait le squelette du polygone (Fig 1. (b)) et on place sur le squelette des sphères implicites. Ces sphères sont ajustées de manière à ce que la surface engendrée par leur mélange épouse au mieux le contour dessiné par l'utilisateur Fig. 1. (c). On dispose ainsi de deux niveaux de représentation : le contour de la surface, qui peut être travaillé en deux dimensions par l'utilisateur, et le squelette de la surface, qui permet une manipulation simple et intuitive, mais cette fois des parties de l'objet (ex. bras, jambes, etc.).
La section 2. présente l'algorithme d'extraction du squelette à partir du contour fourni par l'utilisateur. La section 3. décrit en détail la construction d'une surface implicite à partir du contour 2D, et enfin la section 4. synthétise les opérations de manipulations du squelette implémentées jusqu'à présent.

2.Extraction du squelette à partir d'un contour dessiné par l'utilisateur

Tour d'abord l'utilisateur esquisse un profil non intersectant. Ensuite le contour est éventuellement fermé puis approximé par un polygone uniforme (avec les arêtes de la même longueur). Ceci assure un placement uniforme des primitives implicites qui vont servir à reconstruire la forme.
L'étape suivante est la construction d'une triangulation de Delaunay [Aur91] contrainte à l'intérieur du polygone (voir la Fig. 2. (b)). Ensuite on calcule le squelette du polygone. Pour cela on utilise le *chordal axis* [Pra97]. Ceci est une transformation proche de l'axe médial [CCM97] mais définie localement et elle permet de supprimer les branches inutiles du squelette.
L'algorithme d'extraction du squelette est basé sur celui utilisé dans [IMT99] mais nous allons le décrire en détail pour plus de complétude.
La triangulation produit trois types de triangles : les triangles *normaux* qui ont deux arêtes internes et une arête externe, les triangles *externes* qui ont deux arêtes extérieures et une arête intérieure, et les triangle *internes*, qui ont toutes les arêtes à l'intérieur. Le calcul du squelette est un algorithme récursif qui commence à une extrémité

(triangle *externe*). Lorsqu'on rencontre un triangle *normal*, on joint ensemble les points médians de ses deux arêtes internes, et on appelle l'algorithme pour son (unique) voisin. Lorsqu'on rencontre un triangle *interne*, on crée un point de branchement en son centre de gravité et on appelle l'algorithme pour ses deux voisins. Si on rencontre un triangle *externe* l'algorithme s'arrête et ferme la branche (voir la Fig. 2. (c)).

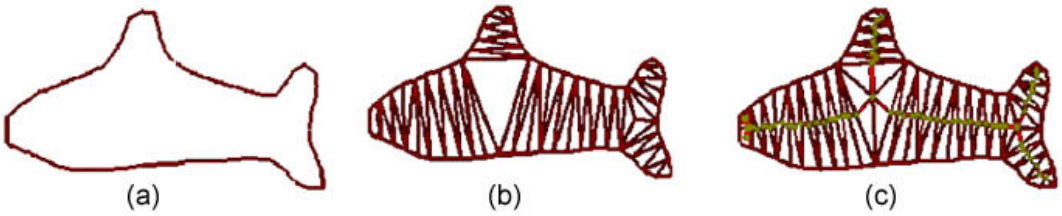

Fig. 2. (a) Contour initial. (b) Triangulation de Delaunay. (c) Elimination des triangles qui comportent des arêtes à l'extérieur de la forme. (d) Calcul du squelette.

3.Extraction de la surface implicite à partir d'un contour

3.1.Considérations théoriques

Les primitives auxquelles on va s'intéresser par la suite sont des primitives sphériques, définies par une fonction $f_i : R^3 \rightarrow R$ et par un point c_i qui est le squelette de la primitive. Ensuite on choisit une norme d, en général la norme euclidienne, et on note $r_i = d(p,c_i)$, la distance d'un point quelconque p au squelette. Alors f_i est une fonction de r_i. Les primitives peuvent être définies également avec un rayon d'influence R_i. Dans ce cas, lorsque $r_i \geq R_i$, la valeur de $f_i(r_i)$ est nulle (primitive bornées) ou presque nulle ($f_i(r_i) < \varepsilon, \forall\, r_i \geq R_i$ avec ε négligeable). La fonction décroît de manière lisse suivant une courbe semblable à une gaussienne entre 1 et 0 lorsque r_i varie de 0 à R_i, comme illustré dans la Fig. 3.
La surface est alors définie par l'ensemble des points $p \in R^3$ tels que $d(p,c_i)=r_0$ et $f_i(r_0) = C$, où C est une constance dans l'intervalle $(0,1)$, en général choisie à $0,5$.

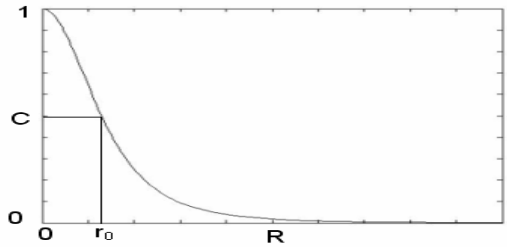

Fig. 3. La fonction implicite appelée « quasi-Cauchy » [She00] qui est utilisée dans notre approche.

Le volume de la surface est défini par l'ensemble des points de R^3 tels que $f_i(r_i) \geq C$. Ces surfaces ont une propriété importante qui est leur capacité à se mélanger automatiquement lorsque l'on somme leurs potentiels. Ainsi, le mélange des N primitives s'exprime en additionnant leurs contributions d'après la relation qui suit.

$$F(p) = \sum_{i=0}^{N-1} f_i\,(p) \qquad (3.1)$$

Le premier type de fonction implicite qui a été proposé dans la littérature est le *Blobby Model* de Blinn [Blin82], qui est à base de fonctions exponentielles. Ce modèle est trop coûteux en temps de calcul, et de ce fait les chercheurs ont essayé d'approximer la gaussienne par des fonctions polynomiales, ce qui a donné naissance aux *Soft Objects* [WMW87] et aux *MetaBalls* [NHK85]. Ensuite il y a eu les modèles issus de besoins spécifiques, par exemple pour les logiciels de rendu (la « polynomiale en W » [POV]), ou pour les surfaces de convolution (la fonction « pseudo-Cauchy » [She99]). Il existe également d'autres modèles polynomiaux [TC94, Sto96].
Dans ce qui suit nous allons faire référence à toutes les fonctions f_i dont la définition satisfait les caractéristiques énoncées, sous le nom générique des « *blobs* ».

3.2.Algorithme de création d'une surface implicite à partir de l'esquisse.

3.2.1.Calcul des valeur initiales pour les paramètres de la fonction

Une fois le squelette calculé, des blobs sont placés en chaque point du squelette et leur somme produit une première approximation du profil 2D par une surface lisse. La fonction utilisée pour produire les objets présentés dans ce papier s'écrit comme suit [She00]:

$$f_i\ (p) = \frac{1}{\left(1 + s_i^2 r_i^2\right)^2} \tag{3.2}$$

r est la distance euclidienne entre le centre du blob et le point p considéré : $r = d(p, c_i)$.
Cette fonction et plus rapidement évaluée qu'une exponentielle, mais on peut également utiliser des fonctions polynomiales à la place.
En général on va essayer d'optimiser les paramètres de chaque primitive pour faire converger la surface vers le contour. Dans un point quelconque de l'espace, la fonction qui caractérise la surface globale s'écrit en additionnant les contributions de tous les blobs :

$$F(p) = \sum_{i-0}^{N-1} f_i\ (p) = \sum_{i-0}^{N-1} \frac{1}{\left(1 + s_i^2 r^2\right)^2} = C \tag{3.3}$$

L'équation comporte un paramètre s_i à déterminer pour chaque blob. La condition pour déterminer ce paramètre est que la surface engendrée épouse au mieux le contour dessiné par l'utilisateur. On ne peut pas déterminer directement les valeurs de ces paramètres, mais par contre on est capable de calculer une première approximation pour ceux ci, qui est ensuite suivie d'un processus d'ajustement pour faire converger la surface plus près du contour. Pour cela on considère tout d'abord chaque blob comme étant isolé. La surface engendrée par un seul blob de rayon e_i s'écrit:

$$f_i\ (p) = \frac{1}{\left(1 + s_i^2 e_i^2\right)^2} = C \tag{3.4}$$

C étant la même constante utilisée pour la surface dans la formule (3.3). e_i est une valeur fixe et représente le rayon en isolation du blob i. Initialement on considère une approximation de e_i comme étant la moyenne des distances aux points voisins qui se trouvent sur le contour (voir la Fig. 4.). Cette valeur de e_i, remplacée dans l'équation (3.4) permet de déterminer la valeur initiale de s_i.
Avec les valeurs des paramètres s_i calculés ainsi, on peut remarquer que la surface prend un aspect « gonflé » par rapport au contour, comme dans la Fig. 5. (a). Ce gonflement est dû au fait que l'on a calculé le rayon initial indépendamment des autres blobs. Lorsque les blobs voisins sont additionnés pour former la surface, leurs contributions viennent se rajouter et elles provoquent le gonflement. Il résulte la nécessité d'ajuster les valeurs des paramètres s_i afin que la surface vienne épouser le contour.
Comme dans les approches précédentes [Mur91], l'ajustement est réalisé en minimisant une fonction d'énergie qui caractérise la distance entre la surface et les points du contour :

$$E = \sum_{j-0}^{M-1} \left(F(p_j) - C\right)^2 \tag{3.5}$$

M étant le nombre des points du contour.
Pour cela on utilise une technique heuristique de type « *Greedy* » qui est rapide à implémenter, et qui présente des résultats déjà satisfaisants. Cette méthode est décrite ci dessous.

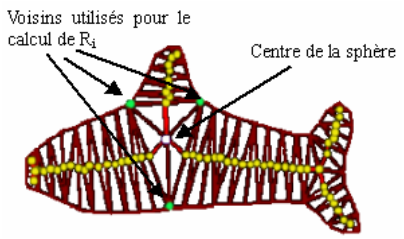

Fig. 4. Calcul des valeurs initiales pour les rayons d'influence R_i.

3.2.2.Ajustement des valeurs des paramètres de la fonction

Tout d'abord on sélectionne le point du contour pour lequel la valeur du potentiel est maximale :

$$p_j, \text{ tel que } F(p_j) = max\left|\sum_{i=0}^{N-1} f_i\ (p_j) - C\right| \tag{3.6}$$

Ce point correspond à la zone dans laquelle la reconstruction a la plus faible qualité. Ensuite on sélectionne parmi les blobs celui dont l'influence en ce point est maximale. Le rayon de ce blob est ajusté par recherche

dichotomique de manière à minimiser la fonction F en p_j. Ensuite on sélectionne un nouveau point donné par (3.6.) et l'algorithme est réitéré jusqu'à l'atteinte d'un seuil prédéfini pour $F(p_j)$, c.à.d. la valeur maximale de potentiel pour un point du contour ou lorsque l'erreur ne peut plus être réduite. Plus la valeur de F décroît, plus E décroît et plus la surface converge vers le contour. Les résultats au niveau visuel sont satisfaisants et les temps de calcul sont de l'ordre de la seconde. Cependant nous pensons qu'une exploration plus approfondie des méthodes numériques de minimisation, afin de faire converger la surface vers le contour d'une manière plus fiable, serait souhaitable pour l'avenir. Le but est de mettre en balance la précision de la forme générée par rapport au temps de calcul afin de choisir la méthode la plus adaptée.
Enfin la surface est polygonisée et affichée, en utilisant un algorithme de polygonisation classique de type Marching Cubes [Blo94].
La Fig. 7. compare le maillage obtenu par notre algorithme de reconstruction et celui obtenu par l'algorithme de [IMT99], et on peut voir une surface nettement plus lisse dans le premier cas, ce qui est dû à la fonction implicite.

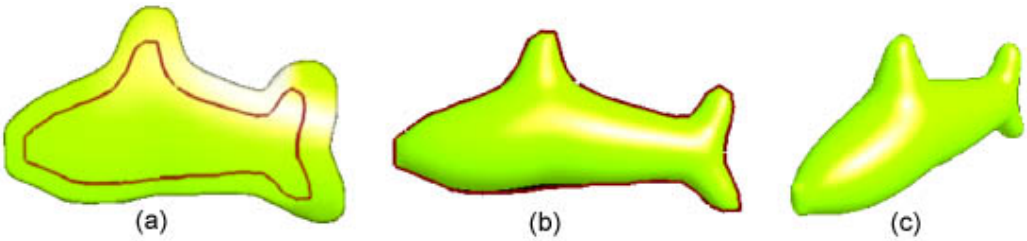

Fig. 5. (a) Surface initiale produite. (b) Surface après ajustement. (c) Le même objet vu d'un angle différent.

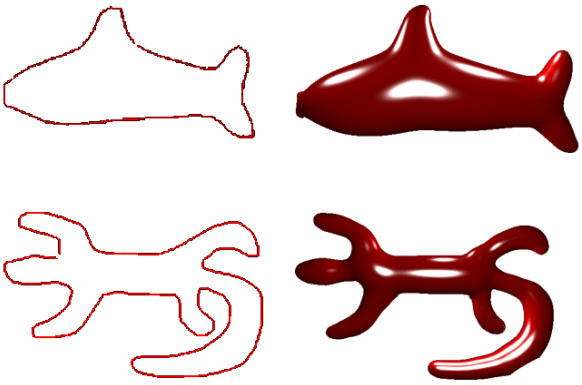

Fig. 6. Images sont rendues avec un logiciel de Lancer de Rayons afin de permettre une meilleure perception de la surface.

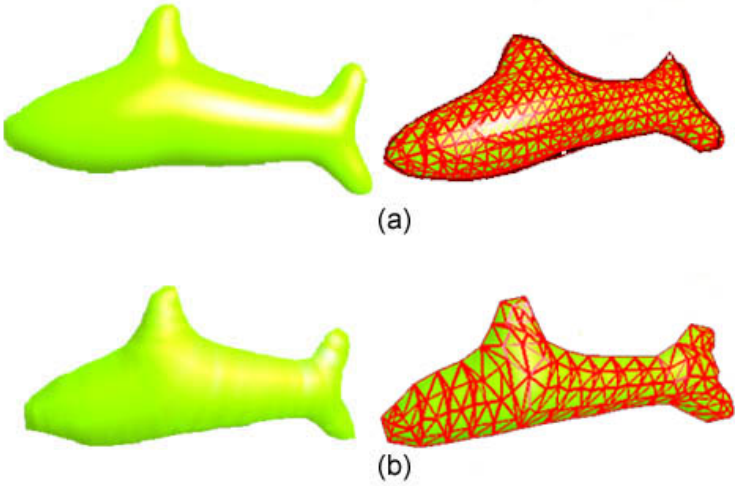

Fig. 7. (a) Surface implicite produite par notre algorithme de reconstruction. (b) Maillage produit par l'algorithme décrit dans [IMT99]. La surface résultante est beaucoup plus lisse lorsqu'on utilise les surfaces implicites.

4.Manipulation du squelette

Le fait de disposer du squelette de la surface est un avantage important, car cela permet de réaliser des opérations simples et intuitives directement sur les points qui le constituent. La figure suivante synthétise toutes les opérations qui sont actuellement implémentées.

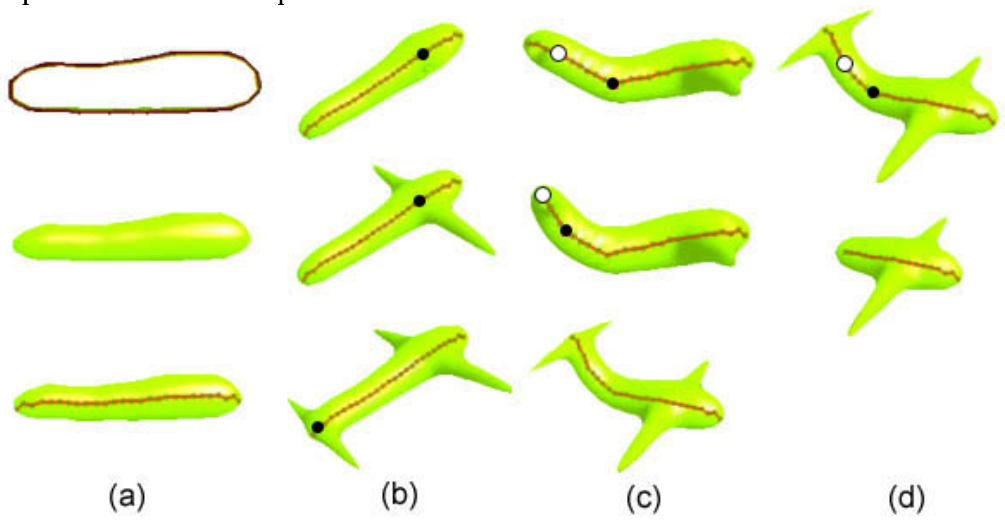

Fig. 8. Opérations possibles avec notre système : (a) Création d'une forme à partir de son profil. (b) Changement d'épaisseur d'un blob. (c) Rotation autour d'une articulation. (d) Suppression d'une partie du squelette.

4.1.Contrôle de l'épaisseur d'un blob

Pour modifier l'épaisseur d'un blob l'utilisateur sélectionne son centre (point foncé sur la Fig. 8. (b)) et déplace la souris ou le stylo tactile en haut ou en bas, suivant l'effet qu'il désire : aplatissement ou bosse. Pour effectuer une mise à l'échelle il suffit d'introduire un facteur dans le calcul de la distance euclidienne. Par exemple, pour le calcul d'une mise à l'échelle sur l'axe *0z*, la distance $r^2=x^2+y^2+z^2$ sera remplacée par $r^2=x^2+y^2+\gamma z^2$, γ étant le facteur de mise à l'échelle. Le blob sphérique implicite devient une ellipse implicite. Deux exemples sont illustrés dans la Fig. 8. (b). Il faut noter que cette opération n'affecte pas l'approximation du contour 2D par la surface.

4.2.Rotation autour d'une articulation

L'utilisateur sélectionne d'abord le point du squelette autour duquel il désire effectuer la rotation, (le point foncé sur la Fig. 8. (c)). Puisque le squelette est un graphe connexe, il existe au moins deux possibilités de sélection pour la partie à laquelle on appliquera la rotation. Le deuxième point sélectionné (le point clair sur la Fig. 8. (c)), sert à identifier la partie à tourner autour de l'articulation. Le « membre » ainsi sélectionné est orienté à l'aide de la souris selon la direction désirée. Tous les blobs qui se trouvent dans la partie sélectionnée vont subir la même transformation de rotation. La propriété de mélange de la surface implicite garantit que l'aspect lisse de la surface sera préservé après transformation. Ici il existe le risque d'apparition du mélange non désiré (ex. un bras amené trop près du corps va se mélanger avec le corps) et nous envisageons de résoudre ce problème en employant un graphe de mélange [WW89]).

4.3.Suppression d'une partie du squelette

Pour l'opération de suppression, la sélection s'effectue de la même manière que pour la rotation : on sélectionne un point de départ (foncé sur la Fig. 8. (d)) et un deuxième point (le point clair sur la Fig. 8. (c)) qui identifie toute la partie à enlever. Tous les blobs qui se trouvent dans cette partie vont être supprimés, y compris celui situé au point de départ.

5.Conclusion et perspectives

Nous avons décrit un système capable de reconstruire des surfaces tridimensionnelles à partir d'un profil esquissé par l'utilisateur. Les fonctions implicites et le squelette offrent l'avantage d'une surface lisse et facilement manipulable, qui peut être stockée de manière compacte et polygonizée avec la résolution désirée pour l'affichage ou pour l'exportation dans d'autres modeleurs classiques ou il est possible de rajouter de détails plus fins.

Le principal désavantage est lié à la difficulté de représentation des arêtes franches.

Les perspectives comprennent : l'exploration des méthodes numériques afin de mettre en balance une convergence plus fiable vers le contour initial et le temps de calcul interactif, l'implémentation des opérateurs de

type copier /coller sur l'objet, retouche de l'objet par des esquisses (à partir de l'objet 3D), retouche local d'une partie du contour après reconstruction, et on peut imaginer encore beaucoup d'autres opérations. L'exploitation d'une tablette graphique doit offrir un meilleur confort à l'utilisateur habitué aux dessins au crayon.
Un aspect important pour améliorer l'interactivité de la modélisation est également l'implémentation d'un algorithme adaptatif et incrémental de polygonisation, afin de mettre à jour uniquement les parties de la surface qui sont affectées par la transformation en cours [GA98,FCG00].

Références

[Aur91] F Aurenhammer, "Voronoi diagrams - A survey of a fundamental geometric data structure", *ACM Computing Surveys*, 1991, 23:345-405.

[Bli82] J. F. Blinn "A Generalisation of Algebraic Surface Drawing", *ACM Trans Graphics*, Vol 1, No. 3 , July 1982, p 135-256.

[Blo94] J.Bloomenthal, "An Implicit Surface Polygonizer", *Graphics Gems IV* (P. Heckbert, ed.), Academic Press, New York, 1994.

[BTC95] E. Bittar , N. Tsingos, M.-P. Cani., "Automatic Reconstruction of Unstructured 3D Data: Combining Medial Axis and Implicit Surfaces", *Eurographics*, September 1995.

[CBC01] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, "Reconstruction and representation of 3D objects with radial basis functions," *Computer Graphics (SIGGRAPH 2001 proceedings), pp. 67-76*, August 2001.

[CCM97] H. I. Choi, S. W. Choi, H. P. Moon "Mathematical Theory of Medial Axis Transform" *Pacific Journal of Mathematics*, Vol. 181, No. 1, pp. 57-88, November 1997.

[FCG00] E. Ferley, M.-P. Cani, J.-D. Gascuel, Practical Volumetric Sculpting, *The Visual Computer* no. 8 vol. 16 pp 469--480 , décembre 2000.

[GA98] E. Galin, S. Akkouche, "Incremental Techniques for Implicit Surface Modeling", *Computer Graphics International'98* 312-321, June 1998.

[GVS00] I. Guskov, K. Vidimce, W. Sweldens, P. Schröder, "Normal Meshes", *Computer Graphics Proceedings (SIGGRAPH 2000)*, pp 95-102, 2000.

[IMT99] T. Igarashi, S. Matsuoka, H. Tanaka. "Teddy: A Sketching Interface for 3D Freeform Design", *ACM SIGGRAPH 1999.*

[KHR02] O. Karpenko, J. F. Hughes et R. Raskar, "Free-form Sketching With Varational Implicit Surfaces", *Eurographics 2002*, TR2002-27, June 2002.

[Mur91] S. Muraki, "Volumetric shape description of range data using blobby model", *Computer Graphics*, 25(4): 227-235, July 1991.

[NHK85] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura, "Object Modelling by Distribution Function and a Method For Image Generation", *The Transaction for the Institute of Electronics and Communication for Engineers of Japan*, 1985, Vol J68-D, Part 4, p 718-725.

[Nor86] D. A. Norman, "User Centred System Design", *Lawrence Erlbaum Associates, Inc.*, 1986.

[POV] POV-Ray , http://www.povray.org/

[Pra97] L. Prasad, "Morphological analysis of shapes", *CNLS Newsletter*, 139: 1-18, July 1997.

[She00] A.Shersyuk, "Convolution surfaces in Computer Graphics" *Ph.D. dissertation, School of Computer Sciences and Software Engineering Monosh University*, Australia 2000.

[Sto96] N. Stolte, *Espaces Discrets de Haute Résolutions: Une Nouvelle Approche pour la Modélisation et le Rendu d'Images Réalistes)*, thèse de doctorat, Université Paul Sabatier - Toulouse - France, April 1996.

[Suc87] S.A. Suchman, "Plans and Situated Actions: The Problem of Human-Machine Communication", *Cambridge Univ. Press. 1987.*

[TBC95] N. Tsingos, E. Bittar , M.-P. Cani, "Semi-automatic reconstruction of implicit surfaces for medical applications", *Computer Graphics International*, June 1995.

[TC94] N. Tsingos, M.-P. Cani, " Un modeleur interactif d'objets définis par des surfaces implicites", *Secondes Journées de l'AFIG* décembre 1994.

[WMW86] B. Wyvill, C. McPheeters, and G. Wyvill., "Data Structure for Soft Objects", *The Visual Computer*, Vol.2, No. 4 August 1986a, p 227-234.

[WW89] B. Wyvill and G. Wyvill. "Field functions for implicit surfaces", *The Visual Computer*, 5:75, December 1989.

[ZHH96] R. Zeleznik, K. Herndon and J. Hughes "Sketch: An Interface for Sketching 3D Scenes", *Proceedings of SIGGRAPH'96,* 1996, p. 163-170.

[ZS03] R. Zenka, P. Slavik "New Dimensions For Sketches*", Spring Conference on Computer Graphics 2003 Proceedings*, Budmerice, Slovak Republic, 2003.

[ZSS97] D. Zorin, P. Schröder, W. Sweldens, "Interactive Multiresolution Mesh Editiong", *Computer Graphics (SIGGRAPH '97 Proceedings),* pp. 256-268.=