Smooth reconstruction of branching objects with B-splines

Fabrice Jaillet — Behzad Shariat — Denis Vandorpe

LIGIM, Bâtiment 710 Université Lyon I, 43 bd du 1 nov. 1918
F-69622 Villeurbanne Cedex

ABSTRACT. *This paper addresses the problem of reconstructing objects defined by a set of 2D cross-sections. Each contour could be characterized by a set of vertices along a closed polygon. We use a skinning method: each contour is first modeled with a parametric B-spline curve, next a surface is stretched on these curves. We propose an extension to this method to obtain a closed surface that defines a volume. The case of branching objects is also handled by an original solution with Coons patches.*

RÉSUMÉ. *Cet article s'intéresse au problème de la reconstruction d'objets définis à partir d'un ensemble de contours 2D. Chaque contour peut être caractérisé par un ensemble de points décrivant un polygone fermé. Nous utilisons une méthode dite de "skinning" : chaque contour est d'abord modelisé par une courbe paramétrique de type B-spline, puis une surface est tendue sur ces courbes. Nous proposons une extension de cette méthode pour obtenir des surfaces fermées qui définissent un volume. Le cas des objets comportant des branches est lui aussi pris en compte avec une solution originale utilisant des carreaux de Coons.*

KEYWORDS: *surface modeling, branching objects, cross-sections, surface skinning, B-splines.*

MOTS-CLÉS : *modélisation de surface, objets à branches, sections planaires, skinning de surface, B-splines.*

1. Introduction

In the medical area, great improvements have been done in digital imagery techniques. Thus, more and more, the anatomic objects are described as a set of 2D parallel contours. Here, we are interested in reconstructing surfaces or volumes over such data. This can be achieved by first extracting from each section a set of closed contours. Numerous methods have been presented, some of them based on “snakes” [COH 90].

Next, one important difficulty is the correspondence problem: which contours should be connected by the surface [EKO 91]? It means that the shape has to be subdivided into different branches. Then, the external shape has to be fitted with either an interpolation or an approximation surface.

Many methods have been proposed for this surface fitting problem. The most widespread one is the polygonal approach. A set of triangular facets is generated between adjacent contours [FUC 77, MEY 92]. Since these methods produce a great number of triangles, some re-tiling methods have been developed [TUR 92]. Some authors have proposed to use the triangulation as a guide to construct a tensor product smooth surface [JOH 95].

Skinning algorithms have been previously proposed to construct a piecewise rectangular surface, but they present some limitations. In [PAR 96], the section curves are interpolated, this leads to a great number of control points to ensure the accuracy, while in [PIE 96] the contours are approximated but they only handle the case of open contours. Both methods are time consuming in order to make all the curves compatible for the skinning.

The handling of branching consists in connecting different branches (independently reconstructed) to give a global representation. When necessary, joining surfaces have to be constructed in branching areas. Eventually, a smoothing step may be performed to produce a continuous model. Proposed solutions can be classified into two classes: those using a polygonal [MEY 92, OLI 96] or polyhedral tiling [BOI 92]; and those based on parametric patches [AST 99, HAN 94, PAR 96].

The organization of the paper is as follows. In section 2, some necessary B-spline definitions are briefly reminded. In section 3, we describe our methodology to reconstruct open surfaces overcoming the problems linked to skinning techniques. In section 4 and 5 we present a solution to ordinary and branching closed surfaces. Finally, the paper ends with a conclusion and prospective section.

2. B-spline formulas and curve reconstruction

In this section, we present the main formulas necessary to fit a closed curve to a set of data points disposed along a contour.

2.1. *Curve approximation*

Let $\{\vec{Q}_k\}$, $k = 0, \dots, r-1$ be the set of r data points that should be approximated with a B-spline curve of degree $p \geq 1$. We search for the control points set $\{\vec{P}_i\}$, $i = 0, \dots, n-1$ of the curve that admits the equation:

$$\vec{C}(u) = \sum_{i=0}^{n-1} \vec{P}_i N_i^p(u) \text{ for } u \in [0,1] \qquad [1]$$

It is possible to rewrite equation 1 in a matrix form $[Q] = [N]\,[P]$, where $[Q]$ is a vector of dimension r representing initial data points, $[P]$ is the vector of dimension n containing the researched control points, and $[N]$ is the matrix of dimension $r \times n$ that could be written:

$$[N] = \begin{bmatrix} N_0^p(\bar{u}_0) & \dots & N_{n-1}^p(\bar{u}_0) \\ \vdots & \ddots & \vdots \\ N_0^p(\bar{u}_{r-1}) & \dots & N_{n-1}^p(\bar{u}_{r-1}) \end{bmatrix} \qquad [2]$$

with:

$$\begin{aligned} N_i^0(u) &= \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_i^p(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u) \end{aligned} \qquad [3]$$

Generally, the matrix $[N]$ is not square ($r > n$), and the system should be solved with the standard least squares approximation method [PIE 95]. That leads to the following constraints:

$$\sum_{k=0}^{r-1} \left| \vec{Q}_k - \vec{C}(\bar{u}_k) \right|^2 \text{ is minimal} \qquad [4]$$

Most of the time, the curve does not go through the points $\vec{Q}_k$, and $\vec{C}(\bar{u}_k)$ is not the closest point of $\vec{Q}_k$. Then, the new system could be written as:

$$[P] = \Big[[N]^T\,[N] \Big]^{-1} [N]^T\,[Q] \qquad [5]$$

(Equation 5 is sometimes written $[P] = \Big[[N]^T\,[N] \Big]^{-1} [R]$, with $[R] = [N]^T\,[Q]$.)

Solving this equation gives the control points of the curve. The computation of $[N]$ (Eq. 2) requires a knot vector $U = \{u_0, \dots, u_{n+p}\}$ and parameters $\{\bar{u}_k\}$ connected to $\{Q_k\}$. Their choice greatly affects the shape and parameterization of the curve. The cord-length method is the most used, and is generally well adapted even when data are non uniformly distributed.

$$\begin{aligned} &\bar{u}_0 = 0 \text{ and } \bar{u}_{n-1} = 1 \\ &\bar{u}_k = \bar{u}_{k-1} + \frac{|\vec{Q}_k - \vec{Q}_{k-1}|}{d} \text{ for } k = 1, \dots, n-2 \end{aligned} \qquad [6]$$

$$\text{with } d = \sum_{k=1}^{n-1} \left| \vec{Q}_k - \vec{Q}_{k-1} \right|$$

Some optimization methods exist to parameterize a set of non-uniformly distributed points along a curve [MA 95, SAR 91].

Knots should also reflect the data distribution. They are defined as follows:

$$\begin{array}{l} u_0 = \ldots = u_p = 0 \text{ and } u_n = \ldots = u_{n+p} = 1 \\ u_{i+p} = (1-\alpha)\bar{u}_j + \alpha \bar{u}_{j+1} \text{ for } i = 1, \ldots, n-p-1 \end{array} \quad [7]$$

where $\alpha = j * dist - i$ with $dist = r/(n-p)$ and j is the integer part of $(i * dist)$. This guarantees that each knot span contains at least one parameter $\bar{u}_k$, and it has been proved that the equation 5 is positive definite and well conditioned [BOO 78]. It is then resolvable with a Gaussian method without pivot.

2.2. *Curve interpolation*

When the curve should fit exactly each point, an interpolation is required. As in the previous subsection concerning approximation, a parameterization $\{\bar{u}_k\}$ and a knot vector $U = \{u_0, \ldots, u_{n+p}\}$ are needed. Equation 1 could be rewritten for each point $\vec{Q}_k$, leading to a linear system with n equations. That could be reformulated in a matrix form $[Q] = [N][P]$. Once more, the choice of the parameterization and knot vector is sensitive. The parameterization is defined using equation 6. The knot sequence could be obtained by an averaging technique:

$$\begin{aligned} u_0 &= \ldots = u_p = 0 \text{ and } u_n = \ldots = u_{n+p} = 1 \\ u_{i+p} &= \frac{1}{p} \sum_{j=i}^{i+p-1} \bar{u}_j \text{ for } i = 1, \ldots, n-p-1 \end{aligned} \quad [8]$$

Thus, knots are defined according to the distribution of parameters $\bar{u}_j$ along the curve.

2.3. *Closed curve*

When the initial data set describes a closed curve, it is difficult to obtain a high continuity at the extremities with an open curve. For example, a C^1 continuity at the joint of a 2 degree curve requires at least three aligned control points.

The use of periodic B-spline allows the reconstruction of closed curves of degree p with a C^{p-1} continuity. The periodicity is obtained by transforming the knot vector. The knot sequence for an open curve:

$$\underbrace{u_0 = \ldots = u_p}_{\text{multiplicity } (p+1)}, u_{p+1}, \ldots, u_{n-1}, \underbrace{u_n = \ldots = u_{n+p}}_{\text{multiplicity } (p+1)}$$

is transformed into a knot sequence of periodicity $(u_n - u_p)$, every knot being of multiplicity equal to 1, according to the following equations:

$$\begin{cases} u'_{p-1-i} & = u_p - (u_n - u_{n-1-i}) \quad \text{for } i = 0, \ldots, p-1 \\ u'_{i+p} & = u_{i+p} \quad \text{for } i = 0, \ldots, n-p \\ u'_{n+1+i} & = u_n + (u_{i+p+1} - u_p) \quad \text{for } i = 0, \ldots, p-1 \end{cases} \qquad [9]$$

This sequence describes the closure constraints of the curve, but it is necessary to add p equations on the control points to ensure a C^{p-1} continuity:

$$\vec{P}_i = \vec{P}_{i+n-p} \text{ for } i = 0, \ldots, p-1$$

Next, these additional equations are inserted into the system to obtain the control points.

3. Surface reconstruction

3.1. *Skinning method*

The skinning method permits to approximate or interpolate a set s of curves $\{\vec{C}_k(u)$, for $k = 0, \ldots, s-1\}$. The result is a B-spline surface. In this technique we assume that all $\vec{C}_k(u)$ are defined in the same basis:

– have common degree p

– contain the same control points number n

– be defined on the same knot vector U

If this is not the case, the curves should be brought to a common basis. This is imposed by the definition of the B-spline surface with a tensor product (Eq. 10):

$$S(u,v) = \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} \vec{P}_{i,j} N_i^p(u) N_j^q(v) \text{ for } u \in [0,1] \text{ and } v \in [0,1] \qquad [10]$$

where $\vec{P}_{i,j}$ represents the vertices of the control net, and $N_i^p(u)$, $N_j^q(v)$ are the B-spline basis functions in the two directions. The u direction is for the curves to be reconstructed, and the v is the longitudinal direction of degree q.

To determine the control net, the equation 10 is rewritten for each intermediate curve $\vec{C}_k(u)$:

$$\begin{aligned} \vec{C}_k(u) &= \vec{S}(u, v_k) \\ &= \sum_{i=0}^{n-1} \left\{ \sum_{j=0}^{m-1} \vec{P}_{i,j} N_j^q(v_k) \right\} N_i^p(u) \\ &= \sum_{i=0}^{n-1} \left(\vec{P'}_k \right)_i N_i^p(u) \text{ for } k = 0, \ldots, N-1 \end{aligned} \qquad [11]$$

where $\left(\vec{P'}_k\right)_i$ for $i = 0, \ldots, n-1$ are known and define the control polygon of each curve. It follows:

$$\sum_{j=0}^{m-1} \vec{P}_{i,j} N_j^q(v_k) = \left(\vec{P'}_k\right)_i \text{ for } k = 0, \ldots, s-1; i = 0, \ldots, n-1 \qquad [12]$$

To determine the set of control points $\vec{P}_{i,j}$, it is sufficient to reconstruct a curve on each column $i = 0, \ldots, n-1$ of control points $\left(\vec{P'}_k\right)_i$ of intermediate curves. This problem can be solved either by approximation (cf. section 2.1) or interpolation (cf. section 2.2). The complete control net defining the surface is simply obtained by grouping the results from each column (the curves in the v direction should also respect the three previous conditions on degree, number of control points and knot vector).

3.2. *Setting compatibility between curves*

The skinning method is general since it allows to fit a surface to curves with different degrees and knot vectors. However, this is done at high cost because these curves should be brought to a common basis by degree raising and knot refinement. For a unique curve, placing knots is relatively easy. For a set of curve it is more difficult since each value affects all the curves. A simple way could consist in distributing knots uniformly but the results are not very satisfying. Ideally, the knot vector should reflect the data distribution along the curve while preventing the formation of empty spans. Good values are obtained by an averaging process. The knot vector is computed for each curve (Eq. 7 or 8), and next, the average is taken:

$$u_i = \frac{1}{s} \sum_{j=0}^{s-1} u_i^j \text{ for } i = 0, \ldots, n+p \qquad [13]$$

where s is the number of contours and $(n + p + 1)$ the number of knots. Empty spans could appear but the problem is avoided by sampling new points in this part. The last step is to set n, the number of control points. The minimal number is very difficult to find, however the following algorithm (Alg. 1) applies dichotomy to reduce it [PAR 96]. Even if the number of points is not optimal it is very close for a cost greatly inferior. In some cases, a simple averaging could be sufficient, especially when data are distributed on quite similar contours. For the compatibility step in the v direction, control points are similarly placed along each column. Once more, an averaging is sufficient and it is not worth to employ time consuming methods as is knot insertion.

Algorithm 1 Reducing the control point number n

1 Initialize the range for n

$\triangleright$ select the contour containing the greatest number of points n_{max}

$\triangleright$ n_{min} is the number of control points to reconstruct this contour with a given error

$\triangleright$ $n_{inf} \leftarrow n_{min} - 1$, $n_{sup} \leftarrow n_{max}$, $n \leftarrow n_{min}$

2 Compute a knot vector according to n

3 Reconstruct the contour with this knot vector

if curve satisfies the given error

then goto next contour

else goto step 5

if every curve satisfies the error

then goto step 4

4 Success

if $n - n_{inf} \leq 1$

then END

else $n_{sup} \leftarrow n$, $n \leftarrow (n_{inf} + n_{sup})/2$

goto 3

5 Failure

if $n_{sup} - n \leq 1$

then END, the best value has been found

else $n_{inf} \leftarrow n$, $n \leftarrow (n_{inf} + n_{sup})/2$

goto step 3

3.3. *Compatibility problems*

The previous method of averaging is no more recommended when curves are very different (number of points between two contours, shape, non-uniform data distribution could be some factors). An evaluation of the complexity of a contour has been proposed [PAR 96]. The contour is compared to its convex hull according to its number of points, area and perimeter. These values are interesting. Indeed, when they are different, contours have certainly different shapes. Unfortunately, the contrary is not true, two different contours could have the same complexity value.

When the set of contours of an object does not fit the complexity criteria, it is necessary to compute the exact knot vector instead of evaluating it by a simple mean value. The idea is to merge together the knot vectors computed for each contour. A refining algorithm derived from the knot insertion is used [BOE 80]. For each merging step, the number of knots increases. Consequently, for a great number of contours, if knots are different from one contour to another, that number could be dramatically high. A method has been proposed to reduce the number of control points generated by the approximation process issued from a large knot vector. It consists in computing the common knot vector, and suppressing as much knots as possible while preserving the deviation of the curve within an error tolerance [PIE 96]. The algorithm is essentially derived from knot removal except that the knot is suppressed simultaneously on all the curves. It could also be used in the longitudinal direction, but in this case the error should be computed with respect to the surface (instead of the curve). This is worth only for a great number of contours.

The main contribution of our method in comparison with the existing ones that use the skinning, is that it produces a small number of control points within a short computation time. Actually, we reduce greatly the number of knots during the compatibility step, thus we reduce the number of control points. However the accuracy is always respected, preserving the validity of the geometric model.

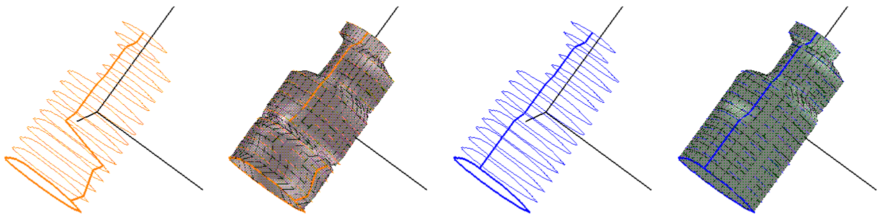
4. Evolution to closed surfaces

The tensor product representation imposed by the choice of the B-spline model introduces some problems. Shapes to be modeled often represent volumes. Thus, some modifications of the classic B-spline model are required to achieve the closure of the objects. In the u direction (the one of the contours), we use the periodic B-splines to reconstruct closed curves. However, the surface remains open in the v direction (on top and bottom in fact). Technically, this is very difficult to use periodic curves in this other direction. The idea that we have introduced is to add some fictitious contours at the extremities. Then, we impose the surface to pass through these contours while imposing continuity constraints.

4.1. *Alignment of contours*

One of the problems that appears during the skinning process on closed curves is the shift of origins between successive contours. So, before any modeling task, the initial data need to be preprocessed. To obtain a smooth surface without twist, it is necessary to align original contours. That means that an adequate starting point had to be found on each contour. The line connecting all these points is called the baseline. Intuitively, this line should be as short and as rectilinear as possible. It should also take into account the relative shape of the contours. Ideally, we should search among all the possible paths the one that best fits the previous criteria. We propose a method that permits to find an acceptable baseline at low cost by testing a restricted number of paths. The key idea is to generate a baseline from each point of one of the contours. The candidate baselines grow by connecting the closest point of the next contour in both directions. Among all these baselines, we select the shortest one. The problem of this method is that although the result is correct locally, it does not reflect the whole shape of the contours. Thus, we are considering several lines (uniformly distributed along one of the contours) at the same time. This permits not to privilege a part of the shape. Experimentally, three or four baselines are sufficient to choose a visually correct set of starting points. Then, the points are simply renumbered.

Figure 1 shows how badly aligned points (a) could lead to twisted parts on the final surface (b). On the contrary, when points are correctly aligned (c), this default disappears (d).



(a) Initial contours (b) Twisted surface (c) Initial Contours (d) Smooth surface

Figure 1. *Reconstruction from (a-b) misaligned and (c-d) correctly aligned contours*

4.2. *Closure of surfaces*

4.2.1. *Extension to one contour*

The proposed idea is to extend the surface in such a way that all the iso-parametric curves in the v direction are convergent to a single vertex. This permits to model the whole object with a single surface.

The shape at the extremities relies on the choice of the vertex position. We propose to use the following equation (Fig. 2):

$$\overrightarrow{FG} = \alpha \cdot \frac{S}{S'} \cdot \overrightarrow{GG'}$$

Thus the summit F lies on the line connecting G and G' the gravity centers of the two last contours. The use of the ratio of the areas S et S' covered by the two last contours allows not to privilege any part of the contours. When the contours are of the same size, the summit is far from the last contour. On the contrary, the summit is near when the size of the last contour is small compared to the size of the other contour. α permits to modulate the shape according to the required global appearance: when α is close to 1, the resulting surface may be stretched, and when it is near 0 the surface may seem to be flat.

Once the summit vertex has been chosen, it should be integrated in the other curves data set. The skinning process requires compatibility: a new B-spline curve (reduced to one point) is then created with same degree and knot vector as the others. Control points are the summit vertex with the necessary multiplicity.

This method gives good results as long as the gravity centers are inside the contours. When the contours are more or less concave, the convergent vertex position could be erroneous and introduce folding on the surface.

4.2.2. *Tangency at extremities*

To ensure continuity at the extremities, additional constraints are introduced when modeling curves in the v direction. A G^1 continuity is sufficient in most cases. Thus we present the method for order 1, but we explain how to obtain higher order continuities.

The reconstruction process is essentially the same that is used for curves without closure, except that each derivative (for G^1 continuity) adds one point and one knot and consequently one more equation in the linear system to be solved. The derivative of a B-spline curve is also a B-spline curve, but with degree $p-1$:

$$\vec{C}'(u) = \sum_{i=0}^{n-2} \frac{(\vec{P}_{i+1} - \vec{P}_i)}{u_{i+p+1} - u_{i+1}} N_i^{p-1}(u) \text{ for } u \in [0,1] \quad [14]$$

It is defined on the same knot sequence from which the first and last knots are removed. We want to impose derivatives at both extremities, that leads to add two supplementary equations to obtain the $n+2$ control points. For example for the interpolation, the new system could be:

$$\begin{aligned} \vec{Q}_k &= \sum_{i=0}^{n+1} \vec{P}_i N_i^p(\bar{u}_k) \\ -\vec{P}_0 + \vec{P}_1 &= \frac{u_{p+1}}{p} \vec{C}'(0) = \vec{D}_0 \\ -\vec{P}_n + \vec{P}_{n+1} &= \frac{u_{m+p-2}}{p} \vec{C}'(1) = \vec{D}_1 \end{aligned} \quad [15]$$

These last two equations are inserted in second and last but one positions to obtain a bandwidth linear system that could be solved by interpolation or approximation as described previously.

The following diagram 2 shows how the derivatives are determined on the top of a curve. If F is the converging point, then the derivative is defined by $\vec{D}_1 = \gamma \overrightarrow{FC'}$ where C' is the intersection of the line $(\vec{Q}_{m-2}\vec{F}_{m-1})$ with the plane containing F and orthogonal to $(\vec{G}_{m-2}\vec{G}_{m-1})$. γ is a coefficient needed to obtain more or less flat shapes. Thus we obtain G^1 continuity as the tangent plane is always the same for each curve.

4.2.3. *Results*

Figure 3 illustrates the accuracy of our closing method. The parameter γ defined at the end of the previous subsection is set to 1, thus the curvature is constant in the

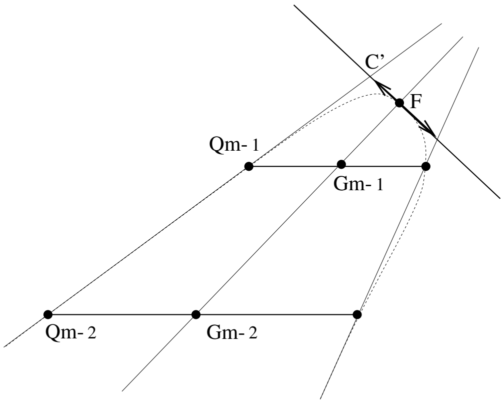
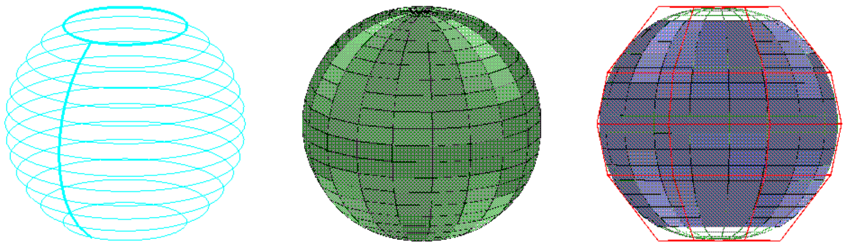


Figure 2. *Summit derivative determination*



(a) Initial contours

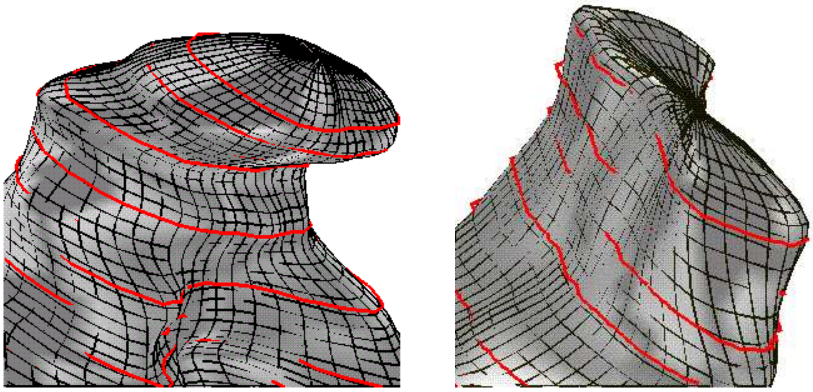
(b) Closed B-spline surface

(c) Closed and open surfaces superposition

Figure 3. *The closing method at extremities*

closing area. In the subfigure 3(c), the control points net permits to verify that the tangent planes' equality is respected.

The figure 4 shows some examples of closure shapes representing volumes. When the last contours are concave, folds may appear on the surface. A solution could be to extend the surface to a curve (for example extracted from the skeleton) and no more to a single vertex. Another solution could be to cap the open surface with another surface, but some joining problems may arise.



(a) Object 1 (b) Object 2

Figure 4. *Examples with real data*

5. Branching objects

The reconstruction of surfaces comprising branches is greatly difficult to solve in the general case. We present a method that handles the bifurcation case (ie. restricted to two branches). In our solution, the trunk and the two branches are reconstructed first, using our skinning method. Then, these three surfaces are connected together with the help of joining Coons patches. We distinguish several cases according to the branching configuration.

To obtain satisfying results and simplify the branching process, we need the use of several patches as described in figure 5. The first two patches make the joint between the body and the branches. The third one fills the space between the two previous ones to provide a smooth surface that lies on existing contours.

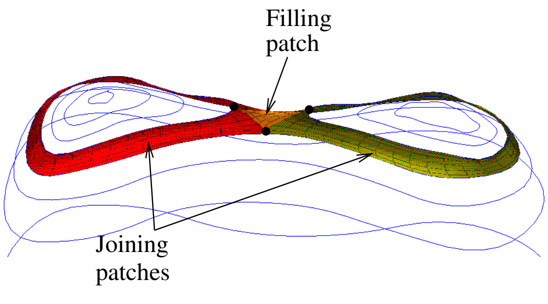


Figure 5. *Joining and filling patches*

5.1. *Creating Coons patch*

Coons patches are bi-parametric surfaces that permit to fill the interior of a patch defined with help of four boundary curves as follows:

$$
\begin{aligned}
C_k(u) &= \sum_{i=0}^{m} N_i^p(u) P_{ki} \quad k = 0,1 \;\; u \in [0,1] \\
C_l(v) &= \sum_{j=0} N_j^q(v) P_{lj} \quad l = 0,1 \;\; v \in [0,1]
\end{aligned} \qquad [16]
$$

These curves must satisfy some compatibility, especially at their extremities: $C_0(u = 0) = C_0(v = 0)$, $C_0(u = 1) = C_1(v = 0)$, $C_1(u = 0) = C_0(v = 1)$ and $C_1(u = 1) = C_1(v = 1)$. The position of a vertex inside the patch could be computed with equation 17 :

$$
\begin{aligned}
S(u,v) \;\; = \;\; & C_0(u).g_0(v) + C_1(u).g_1(v) \\
& + C_0(v).f_0(u) + C_1(v).f_1(u) \\
& - S(0,0).f_0(u).g_0(v) - S(0,1).f_0(u).g_1(v) \\
& - S(1,0).f_1(u).g_0(v) - S(1,1).f_1(u).g_1(v)
\end{aligned} \qquad [17]
$$

where f_0, f_1 and g_0, g_1 are the blending functions with $f_0(t) = g_0(t) = t$ and $f_1(t) = g_1(t) = 1 - t$.

The Coons patch could also be defined as the composition of 3 surfaces:

$$S(u,v) = R_1(u,v) + R_2(u,v) - T(u,v)$$

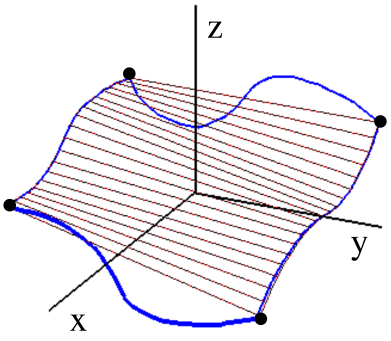
where $R_1(u,v)$ and $R_2(u,v)$ are ruled surfaces between the curves $C_k(u)$ and the curves $C_l(v)$, respectively. $T(u,v)$ is the surface obtained from the following tensor product:

$$T(u,v) = \begin{bmatrix} 1 & u \end{bmatrix} \begin{bmatrix} S_{00} & S_{01} \\ S_{10} & S_{11} \end{bmatrix} \begin{bmatrix} 1 \\ v \end{bmatrix} \qquad [18]$$

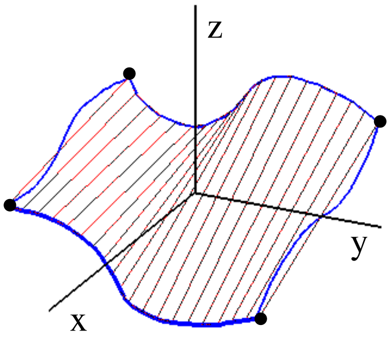
Figure 6 shows the different steps of the construction of the Coons patch.

We apply this model for the joining. We possess two curves at the extremities of the surfaces to be joint. We need next to define the other two boundary curves (Fig. 7). We employ biquadratic B-splines curves that allow to ensure G^1 continuity at least along these curves. Control points are defined as follows (Eq. 19 and Fig. 8).

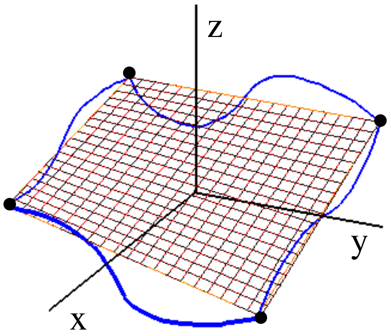
$$
\begin{aligned}
C_{10} &= T(0,1) & C_{20} &= T(1,1) \\
C_{11} &= \alpha_1.\frac{dT(0,1)}{dv_1} & C_{21} &= \alpha_2.\frac{dT(1,1)}{dv_1} \\
C_{12} &= \beta_1.\frac{dB(0,0)}{dv_2} & C_{22} &= \beta_2.\frac{dB(1,0)}{dv_2} \\
C_{13} &= B(0,0) & C_{23} &= B(1,0)
\end{aligned} \qquad [19]
$$



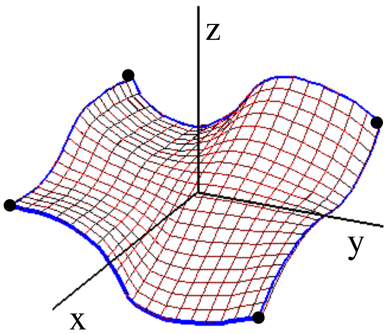
(a) $R_1(u,v)$: ruled surface u



(b) $R_2(u,v)$: ruled surface in v



(c) $T(u,v)$: bilinear surface



(d) $S(u,v)$: final surface

Figure 6. *Reconstruction of a Coons patch*

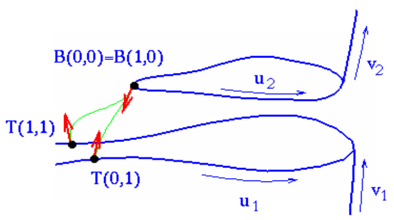


Figure 7. *Boundary curves*

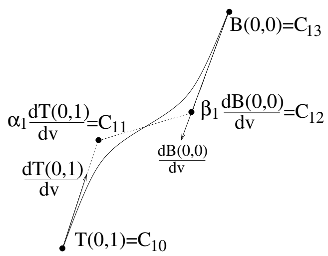


Figure 8. *Control points' determination*

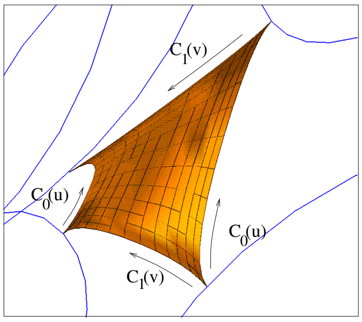


Figure 9. *Filling patch*

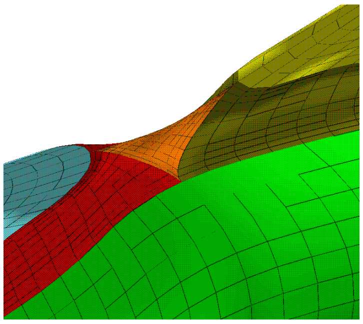


Figure 10. *Close view on branching area*

Parameters α and β are chosen to ensure a uniform repartition of control points along the curve when it is possible.

Once the joints between the body and the branches have been done, there remains a space to be filled. Here also we use the Coons patch model that allows to obtain easily a continuity between adjacent patches. Boundary curves of this patch have already been defined as presented in this section. It is then sufficient to interpolate these four curves to obtain a smooth complete surface (Fig. 9). The final result is shown in figure 10.

5.2. *Continuity problems*

The use of Coons patches generated by polynomial blend gives a C^0 continuity. To obtain a higher continuity, the use of another kind of Coons patch is required: the idea is to employ surfaces of higher order and their derivative fields (Fig. 11).

$$
\begin{aligned}
D_k(u) &= \sum_{i=0}^{m} N_i^p(u) Q_{ki} \quad k = 0,1 \quad u \in [0,1] \\
D_l(v) &= \sum_{j=0} N_j^q(v) Q_{lj} \quad l = 0,1 \quad v \in [0,1]
\end{aligned}
\qquad [20]
$$

The boundary curves should satisfy compatibility criteria at their extremities as presented previously and also satisfy the following twist vectors conditions:

$$
\begin{aligned}
T_{00} &= \frac{dD_0(u=0)}{du} = \frac{dD_0(v=0)}{dv} \\
T_{01} &= \frac{dD_0(u=1)}{du} = \frac{dD_1(v=0)}{dv} \\
T_{10} &= \frac{dD_1(u=0)}{du} = \frac{dD_0(v=1)}{dv} \\
T_{11} &= \frac{dD_1(u=1)}{du} = \frac{dD_1(v=1)}{dv}
\end{aligned}
\qquad [21]
$$

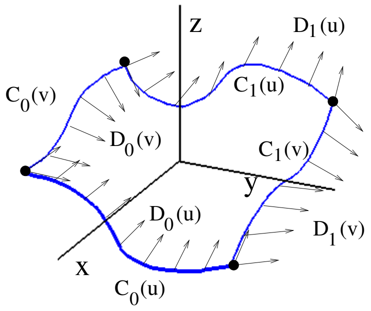


Figure 11. *Boundary curves and derivative fields*

As previously, the Coons surface is defined with help of three surfaces:

$$S(u,v) = S_1(u,v) + S_2(u,v) - T(u,v) \quad [22]$$

But this time, $S_1(u,v)$ and $S_2(u,v)$ are cubic Bézier surfaces in direction u and v respectively, which allows to obtain G^1 continuity. S_1 is defined from $C_k(u)$ and $D_k(u)$, the four rows of control points are computed with the following equations:

$$\begin{aligned} P^1_{i0} &= P_{0i} \\ P^1_{i1} &= P_{0i} + \frac{1}{3}Q_{0i} \\ P^1_{i2} &= P_{1i} - \frac{1}{3}Q_{1i} \\ P^1_{i3} &= P_{1i} \qquad\qquad i = 0, \ldots, m \end{aligned} \quad [23]$$

S_2 is defined in the same way from $C_l(u)$ and $D_l(u)$. $T(u,v)$ is a bicubic Bézier surface. Its control points in the neighbourhood of S_{00} could be computed as shown in equation 24. The other twelve control points are obtained in an analogous way.

$$\begin{aligned} P^T_{00} &= S_{00} \\ P^T_{10} &= \frac{1}{3}D_0(v=0) + P^T_{00} \\ P^T_{01} &= \frac{1}{3}D_0(u=0) + P^T_{00} \\ P^T_{11} &= \frac{1}{9}T_{00} + P^T_{10} + P^T_{01} - P^T_{00} \end{aligned} \quad [24]$$

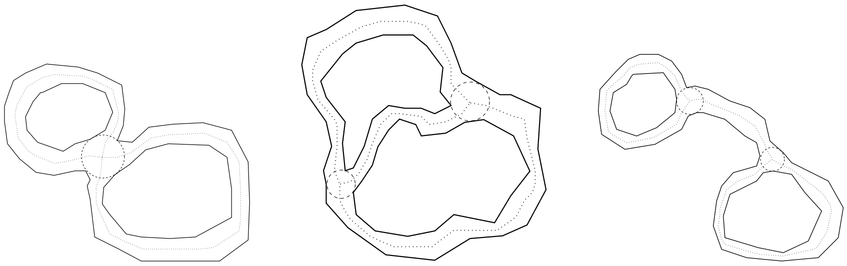
Next, the surfaces S_1, S_2 and T are made compatible (same degree and same knot vectors in the same directions). At last, control points of the final Coons surface could be computed by:

$$P_{ij} = P^1_{ij} + P^2_{ij} - P^T_{ij}$$

In practice, it is difficult to construct such a bicubic Coons surface. In our case, the derivative fields are not supplied for all the directions, thus we use a simplified version which does not take into account all the derivatives.

5.3. *Choice of boundary curves*

Classic branching methods produce poor results as they do not take into account the geometry of the contours before and after the branching area. For simple bifurcation cases, three classes have to be distinguished. More complex branching could also be treated with these terms. The classes are determined by studying the medial axis of the contours' projection on a same plane. One property of the medial axis is to contain as much loops as there are branches. The relative disposition of the loops allows to determine one of the three branching classes (Fig. 12). This allows the definition of



(a) Saddle (b) Canyon (c) Ridge

Figure 12. *Different branching classes*

boundary curves of the Coons patches used to fill up the branching area. We apply a different reconstruction process according to the class of branching that has been found. For a saddle, one single patch is sufficient, while for a canyon or a ridge, three Coons patches are necessary.

5.4. *Results*

The following figures illustrate the quality of our automatic branching reconstruction method on objects containing a saddle (Fig. 13) or a ridge (Fig. 14).

An important contribution of this paper is the branching method. The use of Coons surface permits to interpolate boundary curves and to add continuity constraints on borders. This leads to an interesting model to fill the space between trunk and branches. Each branch is reconstructed independently, then the joint is done taking into account the geometric disposition of surrounding contours.

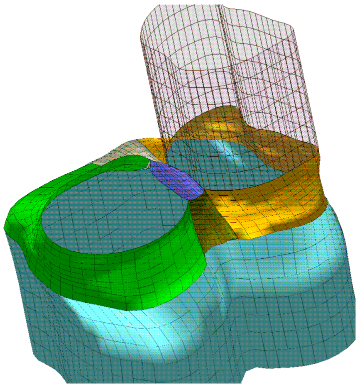


Figure 13. *Coons patches on a saddle*

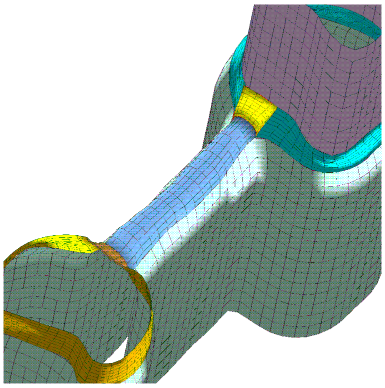


Figure 14. *Coons patches on a ridge*

6. Conclusion

In this paper, we have presented a reconstruction method for objects defined by 2D contours. First, we have focused our attention on simple shapes without branches, homeomorphic to cylinders or spheres. These shapes have been reconstructed with a skinning method. We have proposed a new technique for defining a small common knot vector while making the curves compatible. We never compute the exact knot sequence but estimate it by averaging. This avoids to generate a great number of control points on the final surface. The employed process is at the reverse of other methods, since we start with a simple surface that will be refined where the accuracy is not respected.

This paper also addresses the problem of objects comprising branches. Most of the existing methods do not care about the topology of the branching area, leading to incorrect surfaces. The use of Coons patches allows to obtain a reduced number of surfaces. Moreover, this parametric model permits to handle G^1 continuity.

The obtained results are very satisfactory despite some minor problems. The method works perfectly for relatively regular shapes. As soon as the contours differ greatly from a section to another, an exact reconstruction is required to satisfy error criteria. Even if we have presented an automatic method for bifurcation, the general case of multi-branches is not resolved and seems to be an open research area.

Acknowledgment

This work was supported in part by EC Framework IV, Biomed II, contract BMH4-CT95-0567, project INFOCUS

7. References

[AST 99] ASTART L., SEQUEIRA J., “Modélisation d’un embranchement : une paramétrisation intermédiaire cohérente”, *12e journées AFIG*, Reims, Dec. 1999, p. 342-351.

[BOE 80] BOEHM W., “Inserting new knots into B-spline curves”, *Computer Aided Design*, vol. 12, num. 4, 1980, p. 199-201.

[BOI 92] BOISSONNAT J.-D., GEIGER B., “Three dimensional reconstruction of complex shapes based on the Delaunay triangulation”, report num. 1697, April 1992, INRIA Sophia Antipolis, Programme 4: robotique, image et vision.

[BOO 78] BOOR C. D., *A practical guide to splines*, Springer-Verlag, 1978.

[COH 90] COHEN L. D., COHEN I., “A finite element method applied to new active contour models and 3D reconstruction from cross sections”, *Proceedings of Third International Conference on Computer Vision, Osaka, Japan*, 1990.

[EKO 91] EKOULE A. B., PEYRIN F. C., ODET C. L., “A triangulation algorithm from arbitrary shaped multiple planar contours”, *ACM Transactions on Graphics*, vol. 10, num. 2, 1991, p. 182-199.

[FUC 77] FUCHS H., KEDEM Z., USELTON S., “Optimal surface reconstruction from planar contours”, *Communications of the ACM*, vol. 20, num. 10, 1977, p. 693-702.

[HAN 94] HAN Y., LEE K., LEE S., KHANG G., “Reconstruction of 3D objects from 2D medical images”, *Proceedings of the 6th ICECGDG*, Tokyo, Jp, Aug. 1994, p. 380-384.

[JOH 95] JOHNSTONE J. K., SLOAN K. R., “Tensor product surfaces guided by minimal surface area triangulations”, *IEEE*, 1995.

[MA 95] MA W., KRUTH J. P., “Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces”, *Computer-Aided Design*, vol. 27, num. 9, 1995, p. 663-675.

[MEY 92] MEYERS D., SKINNER S., SLOAN K., “Surfaces from contours”, *ACM Transactions on Graphics*, vol. 11, num. 3, 1992, p. 228-258.

[OLI 96] OLIVA J.-M., PERRIN M., COQUILLART S., “3D reconstruction of complex polyhedral shapes from contours using a simplified generalized voronoï diagram”, *Eurographics’96, Poitiers (F)*, 1996, p. C397-C408, vol 15 (3).

[PAR 96] PARK H., KIM K., “Smooth surface approximation to serial cross-sections”, *Computer Aided Design*, vol. 28, num. 12, 1996, p. 995-1005.

[PIE 95] PIEGL L., TILLER W., *The NURBS book*, Springer, 1995.

[PIE 96] PIEGL L., TILLER W., “Algorithm for approximate NURBS skinning”, *Computer-Aided Design*, vol. 28, num. 9, 1996, p. 699-706.

[SAR 91] SARKAR B., MENQ C.-H., “Parameter optimization in approximation curves and surfaces to measurement data”, *Computer Aided Geometric Design*, vol. 8, num. 4, 1991, p. 267-290.

[TUR 92] TURK G., “Re-tiling polygonal surfaces”, *Proceedings of SIGGRAPH ’92*, Computer Graphics, Jul. 1992, p. 55-64.