# Welding and Pinching Spline Surfaces: New Methods for Interactive Creation of Complex Objects and Automatic Fleshing of Skeletons

Marie-Paule Gascuel

## To cite this version:

Marie-Paule Gascuel. Welding and Pinching Spline Surfaces: New Methods for Interactive Creation of Complex Objects and Automatic Fleshing of Skeletons. Graphics Interface'89, 1989, London, Ontario, Canada. hal-01962411

## HAL Id: hal-01962411

## https://inria.hal.science/hal-01962411

Submitted on 21 Dec 2018

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

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

# Welding and Pinching Spline Surfaces: New Methods for Interactive Creation of Complex Objects and Automatic Fleshing of Skeletons

Marie-Paule Gascuel
LIENS, Ecole Normale Supérieure
CNRS URA 1327,
45 rue d'Ulm, 75230 PARIS Cédex 05, FRANCE

## Abstract

We present three new building primitives for surface modeling systems using splines. The first two, welding and pinching, are used to interactively create more complex objects by combining and by distorting spline surfaces. The welding automatically adjusts the edges to be welded, while the pinching creates continuous junctions of tubular surfaces. The last modeling primitive builds on the first two to "flesh" the skeleton of an object with a regular "skin".

**KEYWORDS:** Splines, free-form surfaces, refinement, deformation.

## 1 Introduction

Modeling complex objects with spline surfaces leads to numerous conceptual problems, topological as well as geometric. These problems are all the more difficult to solve since standard systems generally do not allow us to construct complex surfaces piece by piece, for lack of global primitives to put spline surfaces together (see [4]). On the other hand, in the case of solid modeling, complex objects are often created from simpler ones: We can cite the CSG-tree modeling technique, and the hierarchical operations incorporated by Barr [2] into traditional CAD/CAM solid modeling. This approach is very convenient and helps to retain a global view of the objects to be modeled. Therefore, it is important to adapt it to free-form surface modeling.

The new high level primitives presented in this article allow the user to create a complex surface through a sequence of simple steps: the different parts of the surfaces can be designed independently, and then deformed and combined by the operations of welding and pinching.

The techniques we propose can be used with all tensor-product spline surfaces, provided that the surface supports a refinement algorithm. This family includes, but is not restricted to, surfaces constructed from M-splines or from NURBS. To be easier to understand, the underlying ideas will be developed using the matrix spline notation which will be briefly reviewed in Section 2.

Thanks to the welding primitive, described in Section 3, the user can define the most complex objects in terms of simpler surfaces, cutting him off from technical problems such as "good positioning of the objects", and "adjustment of the edges to be welded". The resulting surface is as smooth as the original pieces.

The pinching of surfaces is described in Section 4. This is a local deformation primitive that allows us to create smooth junctions of tubular surfaces. The different options proposed turn pinching into a very flexible tool. The joint use of pinching and welding is useful, for instance, in the modeling of a tree branch and the creation of the limbs of a figure.

The automatic fleshing of skeletons, described in Section 5, is a direct application of the previous ideas, and answers a need which is being felt more and more: methods have been established to animate articulated rigid objects, or "skeletons", for instance obeying physical laws ([1,14,12,11]). Our aim is to "flesh" these articulated objects with a smooth skin in a purely automatic way. So, relatively realistic animation films may be produced, while avoiding the frame by frame manual intervention of the animator. The fleshing method is also a convenient conceptual tool, well adapted to the creation of objects with a fairly regular shape, such as a fork or a chesspiece.

A further strength of the system is that its interactive capability and its fleshing technique can be used together. This allows us, for instance, to combine regular objects built by fleshing with more irregular objects created interactively. Overall, this provides a powerful and convenient design tool. In addition, the system's flexibility allows it to be rapidly incorporated into many existing modeling systems.

## 2 Terminology and Notations

The primitives described in this article may be used with all the tensor-product spline surfaces supporting a refinement algorithm. The underlying ideas will be developed in the notations of M-splines ($G^1$ translation invariant uniform cubic splines supporting a matrix spline notation, which include for instance B-splines, Beta-splines, C-splines), but could be generalized to the non-uniform case, in particular to NURBS (see [3]).

The matrix spline notation [8] associated with the M-splines can be expressed in the following way: Let $Q(u)$ be a spline, $M$ its matrix (which characterizes the type of spline) and $(V_i)_{i=0..n-1}$ its control vertices. Each cubic segment of the curve can be defined by:

$$Q_i(u) = [u].M.[V]^t \tag{1}$$

$$\begin{aligned}[u] &= (u^3\ u^2\ u\ 1)\ ,\quad u \in [0,1] \\ [V] &= (V_{i-1}\ V_i\ V_{i+1}\ V_{i+2})\end{aligned}$$

Here are two examples of matrices (see also [8]):

$$M_{Cardinal(a)} = \begin{pmatrix} -a & 2-a & -2+a & a \\ 2a & -3+a & 3-2a & -a \\ -a & 0 & a & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$M_{B-spline} = \frac{1}{6}\begin{pmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix}$$

The matrix notation (1) is easy to generalize to tensor-product surfaces: A patch $Q_{i,j}(u,v)$ is defined by:

$$Q_{i,j}(u,v) = [u].M.[V_{i,j}].M^t.[v]^t \tag{2}$$

where $[u] = (u^3\ u^2\ u\ 1)$, $[v] = (v^3\ v^2\ v\ 1)$, $(u,v) \in [0,1]^2$, and where $[V_{i,j}]$ is a matrix of the sixteen control points.

# 3 The Welding Primitive

This section provides us with a new and pleasant design tool: The parts of the objects are independently created and then welded together with automatic adjustment if their edges don't coincide. Furthermore, preprocessings allow two surfaces to be welded even if they are built with distinct types of M-splines, or if they do not have the same number of patches along the edges to be welded.

Let $S_1$ and $S_2$ be two surfaces previously selected. Suppose that the user has chosen the edges to be welded, and has rotated the surfaces in the desired directions. Our aim is to create a single object as regular as the initial surfaces (see Figure 1).

## 3.1 Adjustment of the Edges

First, suppose that the two surfaces have the same number of control points along the edges to be welded.

**First step:** We want to minimize the distance between these edges, by translating $S_2$. Let $(b1_i)_{i=1..n}$ and $(b2_i)_{i=1..n}$ be, respectively, the control point rows of $S_1$ and $S_2$ defining the edges to be welded. We apply to $S_2$ the translation-vector:

$$\vec{v} = \frac{1}{n}\sum_{i=2}^{n-1}(b1_i - b2_i) \tag{3}$$

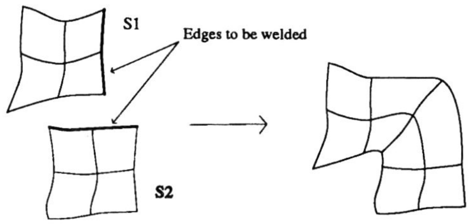

Figure 1: Unexpected directions of the edges to be welded.

**Second Step:** Now, the aim is to deform locally one of the surfaces, or both of them, to form a smooth object. To be perfectly welded, $S_1$ and $S_2$ must have three common rows of control points (see Figure 2). The basic idea is to calculate these common rows as linear combinations of $S_1$'s and $S_2$'s rows.

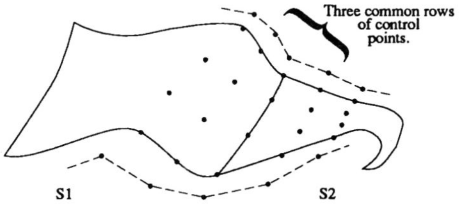

Figure 2: Control graphs of $S_1$ and $S_2$.

Method 1: We distort $S_2$ alone by using the coefficients: $[(1,0),(1,0),(1,0)]$ for the linear combination. The deformed area of $S_2$ is three patches wide. The welding edge coincides exactly with the former edge of $S_1$. See Figure 3.

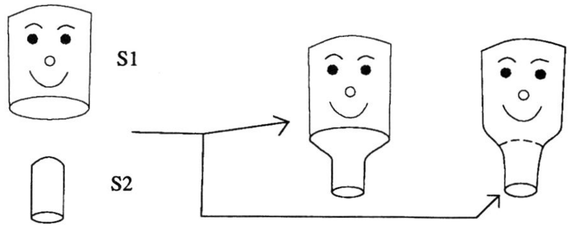

Figure 3: Welding options.

Method 2: $S_2$'s deformation will be less apparent if a symmetric processing is used for both surfaces (see Figure 3). We adjust the smoothness of the welding by controling the coefficients which define the three common control point rows: If we use $\left[(1,0),(\frac{1}{2},\frac{1}{2}),(0,1)\right]$, two patch rows of both $S_1$ and $S_2$ are deformed. With $\left[(\frac{2}{3},\frac{1}{3}),(\frac{1}{2},\frac{1}{2}),(\frac{1}{3},\frac{2}{3})\right]$, for instance, the deformed area is larger (three patches wide).

A welding with the deformation of both surfaces often leads to a more natural result, since $S_2$'s distortion is less appearent. Nevertheless, welding an additional piece without altering an already designed object can be very useful (an example will occur in Section 4).

**Note:** Our method is easy to generalize to $G^1$ or $G^2$ weldings, by adjusting the surfaces in order to achieve the underlying constraints. Then, control parameters (such as bias, continuity, or tension) are available along the welded edges (see [6,7]).

## 3.2 Welding Surfaces of Different Types

Suppose that $S_1$ and $S_2$ are defined by two distinct matrices $M_1$ and $M_2$, and assume first that we only want to modify $S_2$. We can't assign directly to $S_2$, whose matrix is $M_2$, three control point rows of $S_1$. Therefore, we need to work out the imaginary control points matrix $[X]$ that would define a patch $Q_1(u, v)$ of $S_1$ if its type was $M_2$. The unknown matrix $[X]$ satisfies the equation:

$$Q_1(u, v) = [u].M_2.[X].M_2^t.[v]^t \tag{4}$$

By identifying terms of the same degree in $u$ and $v$ in (4) and in the matrix equation defining $Q_1$, the equation for $[X]$ can be rewritted as:

$$M_1.[V1].M_1^t = M_2.[X].M_2^t \tag{5}$$

$$[X] = M_2^{-1}.M_1.[V1].M_1^t.(M_2^{-1})^t \tag{6}$$

If the two surfaces must be distorted, twice as much work must be done: We also need to calculate which control points would define $S_2$ if its matrix was $M_1$.

## 3.3 Different Numbers of Patches Along the Edges to be Welded

The basic idea is to use the spline refinement property (see [8]) to repeatedly halve patches of $S_1$ (or $S_2$) providing control points to add along the edge to be welded (see Figure 4). Then a standard welding is executed.

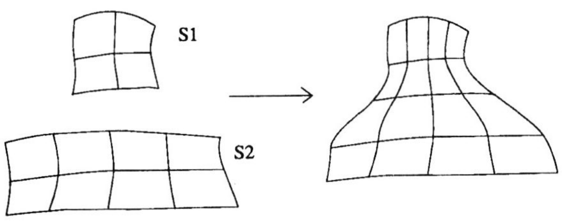

Figure 4: Welding with $S_1$'s refinement.

Let $Q_{i,j}(u, v)$ be a patch, defined by a matrix $M$ and a control graph $[V]$. We want to divide it into two patches along the curve $v = \frac{1}{2}$. Let $T(u, v)$ be the refined surface:

$$\begin{aligned} T_{i,2j}(u, v) &= Q_{i,j}(u, v/2) &(7)\\ T_{i,2j+1}(u, v) &= Q_{i,j}(u, (v+1)/2) &(8) \end{aligned}$$

The control point matrix $[X_{i,2j}]$ of $T_{i,2j}$ is then given by:

$$[u].M.[X_{i,2j}].M^t.[v]^t = [u].M.[V].M^t.[v/2]^t \tag{9}$$

where:

$$\left[\frac{v}{2}\right] = \left(\left(\frac{v}{2}\right)^3 \left(\frac{v}{2}\right)^2 \frac{v}{2}\, 1\right) \tag{10}$$

Then, we have:

$$[u].M.[X_{i,2j}].M^t.[v]^t = [u].M.[V].M^t.M_{left}.[v]^t \tag{11}$$

where:

$$M_{left} = \begin{pmatrix} \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

By identifying terms of same degree in $u$ and $v$ in (11), it can be rewritted as:

$$[X_{i,2j}] = [V].M^t.M_{left}.(M^{-1})^t \tag{13}$$

In the same way, if $[X_{i,2j+1}]$ is the control point matrix of $T_{i,2j+1}$, we have:

$$[X_{i,2j+1}] = [V].M^t.M_{right}.(M^{-1})^t \tag{14}$$

$$M_{right} = \frac{1}{8}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 3 & 4 & 4 & 0 \\ 1 & 2 & 4 & 8 \end{pmatrix} \tag{15}$$

The control point matrices corresponding to a splitting of the patches along the curve $u = 1/2$ are given in [10].

**Note:** The splitting of some patches of a surface, but not of others, may cause an irregular welding. This problem can be solved by refining in such a way that all the patchs of $S_1$ - respectively of $S_2$ - are split the same number of times: The final number of patches along the common edge is then the smallest common multiple of the original numbers of patches of the two surfaces. Nevertheless, this technique is useful only if all the patches of a surface have about the same size, and may unecessarily increase the number of control points.

## 3.4 Welding Tubular Surfaces

A tubular surface can be created, for instance, by sweeping a plane section along a skew path [5,10]. We can close the extremities of such surfaces while preserving the tangent smoothness (see [10]).

To weld two tubular surfaces, further preprocessing is needed after an eventual refinement: The edges to be welded are now closed curves, so we must find the control points numbering that produces the most natural welding (the points with equal numbers are merged together). A circular permutation of $S_2$'s points is executed to minimize the quantity:

$$A = \sum_{i=1}^{n} |b1_i - b2_i| \tag{16}$$

where $(b1)_{i=1..n}$ and $(b2)_{i=1..n}$ are the control point rows of the edges to be welded.

Welding tubular surfaces allows us to distort continuously the cross-sections of the objects. Using standard sweeping techniques, we are now able to construct the same objects as those swept with a profile curve giving a scale factor along a path (see [5] and Figure 5). Furthermore, a class of objects with continuously changing shape cross sections can be created by welding (see Figure 6). This could not be done with the profile curve method.

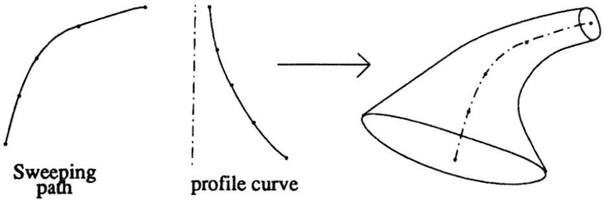

Figure 5: Sweeping with a profile curve.

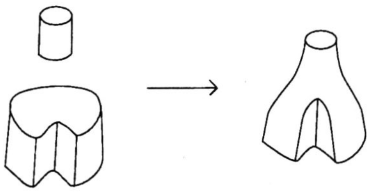

Figure 6: Objects with continuously changing cross sections.

# 4 Pinching

The welding primitive has provided a method for continuously changing the cross-section shape along tubular objects. Now, we want to create smooth junctions of tubular surfaces. Our method allows us to weld several tubes to a single bigger tube, while preserving as much of the regularity of the resulting object as possible.

### Simple Pinching

Our aim is to distort locally the extremity of an initial tubular surface to create two folds where two other tubes will be welded (see Figure 7). To do this, two points of the last visible section of the tube must be selected, and then merged at the "pinching point" (which can be, for instance, midway between the selected points).

If we use interpolating splines, the two points can be selected from among the control points. For approximating splines, we have to work out where to put the control polygon for the section to have a double point. To achieve this, in the case of B-splines, we use the edit-points manupulation method presented in [9].

As soon as the pinched edge of the tube presents a double point, the surface is split into two open surfaces which have a closed $C^0$ section at their pinched extremity (see Figure 7). The last control point row of the initial tube must be distorted as well to align the tangent vectors with those of the small tubes to be welded. Then, the two open surfaces are used as the welding basis for the small tubes. The former are not distorted during the welding, in order to preserve the regularity along their common edge.

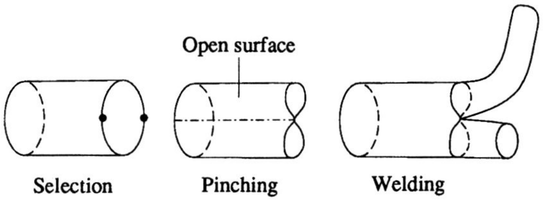

Figure 7: Simple pinching.

A direct welding of a small tube (which has a closed section) to an open surface (undeformed during the welding) would cause an overlapping of the two patches of the small tube which are adjacent to the pinching point (see Figure 8). Moreover, tangent discontinuities would appear between two patches of the second patches' row. Thus we process further processing, according to the type of spline used, to minimize these discontinuities. For instance, in the case of regularizing splines such as B-splines, the overlapping can be avoided by also pinching the initial tube's third section (see Figure 8): Then, the edges of the two patches which overlapped are defined by the same four control points, and thus the edges are merged. The tangent discontinuities of the next patches' row disappear if we move pairs of neighbour points of the initial section to the pinching point.

So, our pinching primitive preserves the initial regularity of the small welded tubes, except along a curve segment passing through the pinching point, where the tubes's surfaces are only $C^0$. These local tangent discontinuites cannot be avoided, since the extremity of the opened surface used for the welding is only $C^0$ at the pinching point, and is not distorted. In the real world, these tangent discontinuities occur, for instance, between the fingers of a hand.

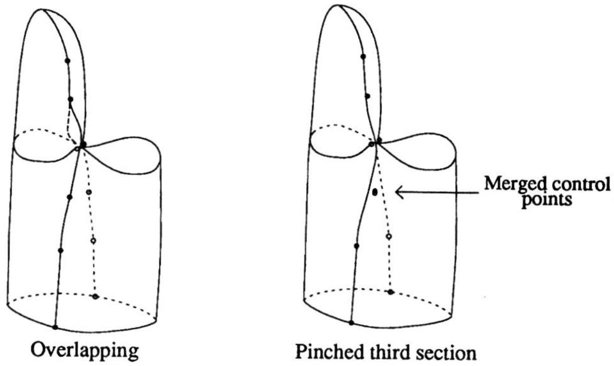

Figure 8: Overlapping problems.

**Non Symmetric Pinchings:** All kinds of non symetric pinchings can be produced by modifying the repartition of the selected points around the section of the initial tube.

Such a pinching is used, for example, to weld the thumb during the design of a hand.

### Central Pinching with $n$ Branches

The previous method is easy to generalize to the case of multiple junctions. To weld $n$ tubes to an initial tube, $n$ points are selected and merged at the pinching point. The initial tube is then splitted into $n$ open surfaces with a closed $C^0$ section. Smaller tubes can then be welded to these surfaces, by the previous method.

Note that it is often necessary to refine the initial cross-section a little before pinching to produce larger folds (see Figure 9).

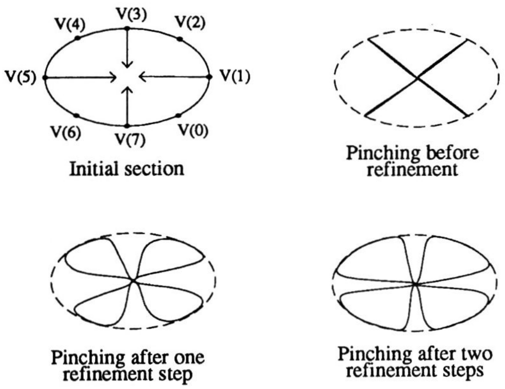
V(4)
V(3)
V(2)
V(5)
V(1)
V(6)
V(7)
V(0)
Initial section
Pinching before refinement
Pinching after one refinement step
Pinching after two refinement steps

Figure 9: Central pinching with four branches.

### Multiple Pinchings with the Same Axis

If we want, for instance, to model the chest of a figure by pinching the shoulder into three parts, in order to weld the arms and the body, a double pinching with the same axis must be used (see Figure 10). This kind of pinching is more difficult to perform than a central pinching: In the latter case, the open surfaces created by pinching were piecewise surfaces. Here, the central surface is composed of two disjoint pieces. Thus, the usual welding operation cannot be used: A tube consisting of the same number of patches as the central surface along the edge to be welded is defined by fewer control points (even including the last three points of its section, which coincide with the first three) because all its patches are contiguous. It is thus impossible to merge these points to those of the central open surface (see Figure 10).

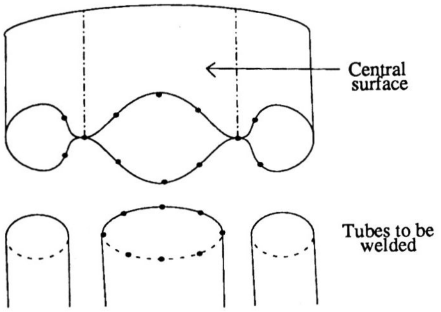

Figure 10: Double pinching with the same axis.

This problem can be solved by moving pairs of neighbour points during the pinching of the section. The quadruple points that appear along the pinched section create degenerate patches. It is then possible to weld the central open surface to a tube with an equal number of patches along its section.

## 4.1 Comparison with Other Modeling Techniques using Patches

We saw previously that the welding primitive applied to tubular surfaces provides a great improvement compared with classical sweeping techniques [5], by allowing us to create smooth objects with sections of varying shapes.

The combined use of multiple pinching and the closure of the ends of tubes can yield the same kind of objects as those created by hierarchical spline refinement [9]. Nevertheless, a single junction direction is provided by our method, since pinching can occur only at the extremities of the tubes. Thus, it would be interesting to introduce welding methods elsewhere than on the edges of surfaces.

On the other hand, it is very natural to express the design of some objects (a hand for instance) in terms of successive weldings and pinchings, objects which are much more harder to design by hierarchical refinement. Furthermore, for topological reasons it is impossible to use refinement to create closed surfaces or objects with holes like those of Figure 11.

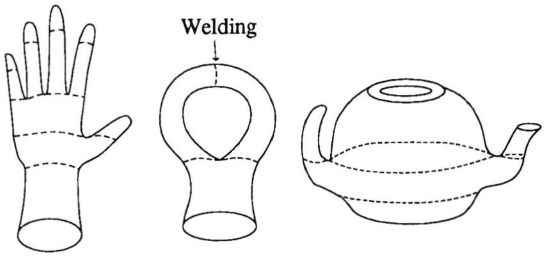

Figure 11: Examples.

# 5 An Application: The Automatic Fleshing of Skeletons

To convey the story of an animated film, we need only to specify the parameters controlling the movement of the objects, or their deformations. Thus, a model of the articulations of the objects (a "skeleton") is often used. A walking figure, for instance, can be animated independently of its skin's surface by applying physical forces to its bones (see

[1,14]), by using purely cinematic methods (see [11]), or finally by combining these two approaches (see [12]).

Our fleshing method builds on both welding and pinching to automatically create an object's "skin" from a description of its skeleton including the flesh thicknesses desired. The flexibility of this automatic construction method should reduce the animator's work during the creation of a film. The fleshing method is also a rapid modeling tool, very convenient for creating objects with a fairly regular shape.

## 5.1 Basic Ideas

The skeleton can come from a system which animates articulated objects or be designed interactively. In the latter case the user must specify, during the creation of each bone, the desired flesh thickness. He may even give several "thickness vectors" in various directions. The data is stored in a tree structure: Each node describes a bone, characterized by its extremities and its thickness vectors (see Figure 12). The basic idea is to produce, during a traversal of this tree, a sequence of elementary orders such as "create a skin tube", "weld", "pinch"...

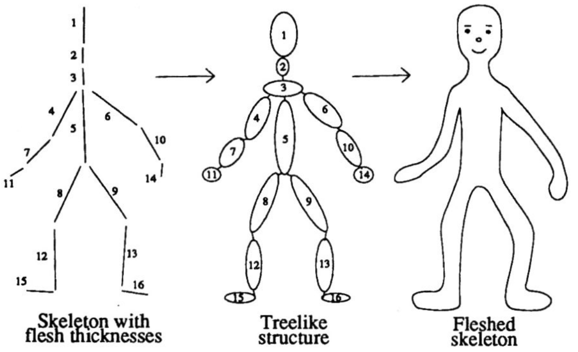

Figure 12: Steps of a figure's fleshing.

Let us describe more precisely a standard fleshing primitive which may be used, for instance, to flesh the figure in Figure 12:
- Two thickness vectors are specified for each bone.
- The treelike structure is covered level by level, starting at the root.
- For each bone:

- A tube is created by sweeping:
  - The sweeping path is defined by spacing six control points evenly along the bone segment.
  - An ellipsoidal cross-section is computed from the two thickness vectors.
- If the current node is the root of the structure
  - then the first extremity of the skin tube is closed;
  - otherwise, the tube is welded on to the free end of the surface corresponding to its parent. The latter, which may be an open surface created by a pinching, is not distorted. A welding without translation of the tubes is used because the bones are already in the proper position.
- If the current node is a leaf of the structure
  - then its last extremity is closed;
  - otherwise, the tube is pinched and split into a number of open surfaces equal to the number of its children in the structure. Those surfaces waiting for a son are kept in a list which is updated during each welding.

**Note:** More generally, a skeleton described by a graph containing loops can be fleshed.

## 5.2 Flexibility of the Method

The standard method presented in the last section is perfectly well adapted to the fleshing of articulated solid objects whose components are rectangular parallelepipeds (two directional thicknesses are known for the cross-section) and whose bones are well positioned (the translation at the begining of the welding is unnecessary). Another application field would have led to different choices. Below, we summarize the key elements of the fleshing algorithm, which provide a compromise between the simplicity of the data and the richness of the created objects.

**Construction of the Cross-Section:** The two thickness vectors that were used in the standard primitive of Section 5.1 allow us to create ellipsoidal cross-sections and provide a sufficient richness to design most objects. Nevertheless, it may be useful in some cases to use more than two vectors or to offer a choice between different models of sections (circular, square, with a star shape, etc).

**Construction of the Sweeping Path:** To create a tube of skin around a bone, the path used must correspond to the bone segment. Nevertheless, we have to choose the number of control points defining the path and their repartition along the bone. These choices will greatly influence the shape of the tube if it has a closed end, or the size of the deformed area during a welding. We must try to minimize the number of control points on the path, while preserving as much as is needed of the initial shape of the tubes during a welding.

**Welding Options:** For the simple weldings of one tube onto another one we can decide if the two tubes must be deformed or not, and adjust the smoothness of the welding by the method proposed in Section 3.1.

For prepositioned objects, such as the bones in a skeleton description, we can weld the objects without translating them. In addition, this kind of welding allows us to hollow out the objects by using the overlapping of a bone segment and its child (see, for instance, the chess queen in the picture on the last page).

**Ends Closure:** Many applications require the closure of the free ends of the objects. This is handled by introducing a new parameter in the description of the leaves or of the root of the structure, specifying if the corresponding tube must be closed or not.

**Pinching Axis, Dissymmetric Pinchings:** For a simple pinching, it is important to choose carefully the pinching axis defined by the straight line joining the selected points of the section. It would be difficult to specify this choice in the treelike structure describing the skeleton: This would impose too much thought about fleshing during the creation of the skeleton. It is often sufficient to pick a convention for fixing a pinching axis (it may be for instance the direction of one of the thickness vectors). A parameter specified in the structure may offer a choice between simple options of non-symetric pinchings, such as "pinching in the middle of the section", "pinching at the quarter"...

Another method consists in computing the points to select for each pinching according to the positions and the thicknesses of the small tubes to be welded (this includes the choice between the pinching options of Section 4: central pinching or multiple pinchings with the same axis). The best points are those which minimize the distortion of the small tubes during the welding.

# 6 Conclusion

In many modeling systems using splines, the only deformation tool available is still direct manipulation of the control points, with all the difficulties associated with this method: point capture in 3-dimensional space, solely local control requiring continual interventions, and resulting slowness and tediousness. With such systems, it is difficult to intuitively create complex objects as products designed by combining and distorting simpler surfaces. To retain a global vision of the objects being created, we need to use high level primitives such as those presented in this article. Welding and pinching provide convenient and efficient control of the surfaces. Of course, more classic approaches may be used for adjusting individual points.

In addition, the design by welding and pinching reduces the data storage requirements: As in [2], the objects are stored in a hierarchical structure describing the history of their construction ("building-tree"). The leaves of this structure can be, for instance, tubes created by sweeping; then, only a path and a cross-section are needed to define them. Moreover, the use of a "building-tree" allows simple editing of objects (the different nodes of the structure can be modified independently). Many deformations, besides pinching and end-closing, can be incorporated into the structure. For instance, the operations presented in [13], in [2], or in [4] may be included.

Furthermore, an object can be automatically built from the "construction plan" provided by its skeleton. The articulated objects animation systems provide a very good application area for this idea: animated films may be produced using automatic fleshing of articulated skeletons, thereby, reducing the animator's work.

Finally, welding, pinching, and fleshing methods can be used with most current types of splines. Therefore, they are easy to incorporate into any existing system.

## Acknowledgments

Thanks to Claude Puech, Anne Verroust, Richard Cole, François Sillion, and Jean-Dominique Gascuel, who made many valuable suggestions which helped me to clarify and refine my ideas. Special thanks to Jean-Philippe Thirion who provided me with the ray-tracer.

# References

[1] W. Armstrong and M. Green. The Dynamics of Articulated Rigid Bodies for Purposes of Animation. *Graphics Interface 85*, pp. 407–415, 1985.

[2] Alan H. Barr. Global and Local Deformations of Solid Primitives. *Computer Graphics*, Volume 18, Number 3, pp. 21-30, 1984.

[3] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann Publishers, Palo Alto, California, 1987.

[4] E.S. Cobb. *Design of Sculptured Surfaces Using the B-spline Representation.* Ph.D. Thesis, University of Utah, Salt Lake City, Utah, June 1984.

[5] S. Coquillart. A Control Point Based Sweeping Technique. *IEEE Computer Graphics and Applications*, pp. 36–45, November 1987.

[6] Du and Shmitt. Bézier Patches with Local Shape Control Parameters. *Proceedings of EUROGRAPHICS'87*, Amsterdam, 24-28 August 1987, pp. 261-274, 1987.

[7] W.H. Du. *Etude sur la Representation des Surfaces Complexes: Application a la Reconstruction de Surfaces Echantillonnées.* Thèse, Ecole Nationale Supérieure des Télécommunications, Paris, France, Octobre 1988.

[8] T. Duff. Splines in Animation and Modeling. *Unpublished course notes. Cook, R. ed. State of the Art in Image Synthesis (SIGGRAPH'86 course notes Number 15, Dallas, TX)*, 1986.

[9] D.R. Forsey and R.H. Bartels. Hierarchical Bspline Refinement. *Computer Graphics*, Volume 22, Number 4, pp.205-212, 1988.

[10] M-P. Gascuel. Recollement et Pincement de Surfaces Splines: Nouvelles Méthodes pour la Création Interactive d'Objets Complexes et l'Habillage Automatique de Squelettes. *Rapport de Recherche du LIENS*, Number 88-12, Ecole Normale Supérieure, Paris, France, Novembre 1988.

[11] M. Girard and A.A. Maciejewski. Computational Modeling for the Computer Animation of Legged Figures. *Computer Graphics*, Volume 19, Number 3, pp. 263-270, 1985.

[12] P.M. Isaacs and M.F. Cohen. Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics. *Computer Graphics*, Volume 21, Number 4, pp. 215-224, 1987.

[13] T.W. Sedeberg and S.R. Parry. Free-form Deformations of Solid Geometric Models. *Computer Graphics*, Volume 20, Number 4, pp. 151-160, 1986.

[14] J. Wilheims. Using Dynamic Analysis for Realistic Animation of Articulated Bodies. *IEEE Computer Graphics and Applications*, pp. 12–27, 1987.

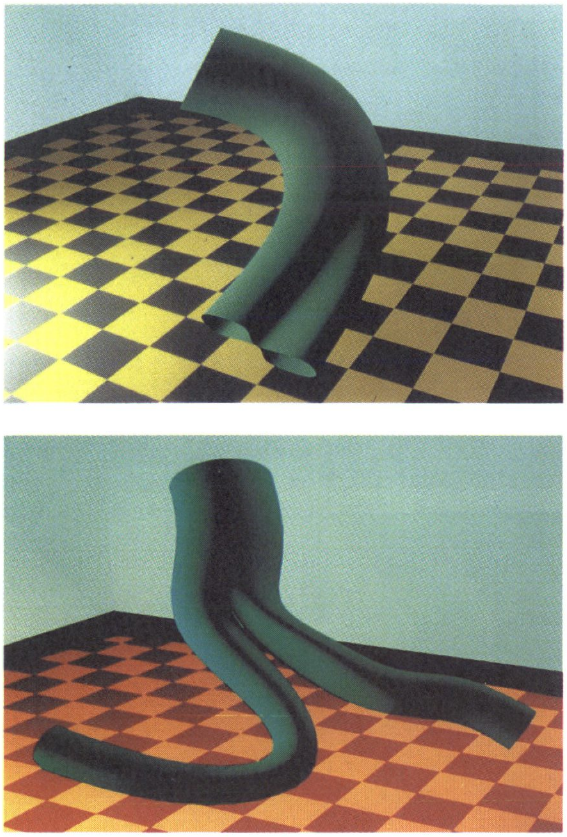

Figure 13: Pinching and welding.

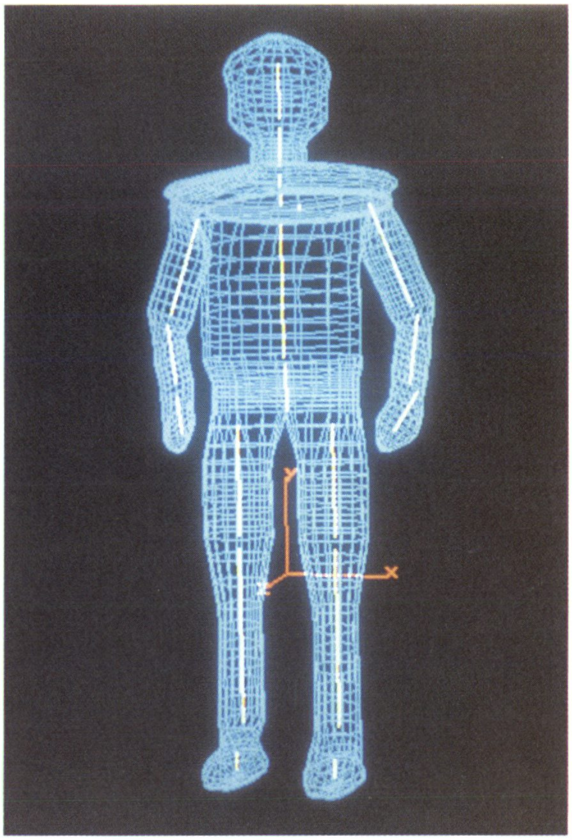

Figure 14: Automatic fleshing of a skeleton.

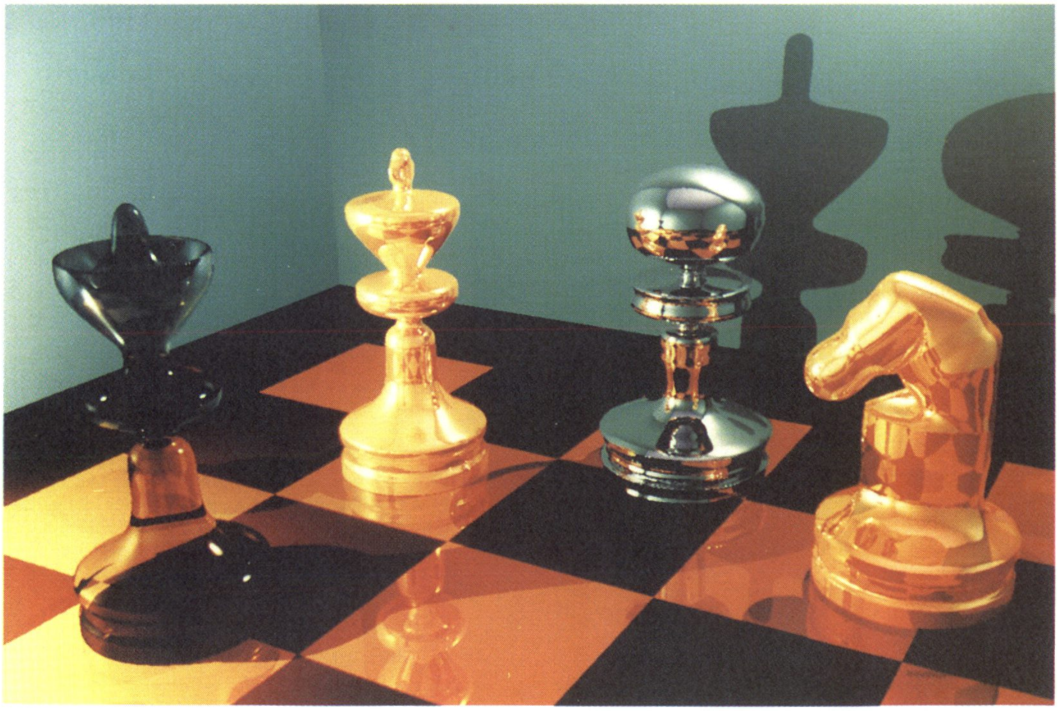

Figure 15: Chesspieces created by automatic fleshing of skeletons.