THE TRIMMED NURBS AGE *

G. CASCIOLA † AND S. MORIGI ‡

Abstract. In this work we report our experience in the geometric modeling field, using trimmed NURBS surfaces and we describe how these surfaces can be used both for solid and surface modeling purposes. Our experience is based on the design and implementation of *xcmodel*, an interactive graphics system merging new and well-known methods for modeling.

Key words. trimmed NURBS, rendering, surface modeling, solid modeling.

AMS subject classifications. 65D17, 65Y25, 65Y20

1. Introduction. In geometric modeling there are two fundamental representation schemes which have been successful and have been incorporated in the most popular CAD/CAM software packages: solid modeling and surface modeling. A solid modeler represents an object unambiguously by describing its surface boundary and by topologically orienting it, so that we can tell, at each surface point, on which side the solid interior lies. Whereas, a surface modeler gives only a geometrical description of the object boundary without any topological information.

The Non-Uniform Rational B-Splines (NURBS) is a well-established tool for geometric design. NURBS have become the de facto industry standards for the representation, design, and data exchange of geometric information. NURBS have been added to several international standards, and many packages include NURBS as the primitive for designing simple and free form curves and surfaces. However, the most useful NURBS paradigm is limited by the requirement that the surfaces are defined over rectangular regions and this leads to topological rectangular patches. A generalization for an arbitrary topology can be obtained by collapsing some of the control mesh edges, but this creates surfaces with ambiguous surface normal and degenerated parametrization.

The introduction of a 'trimmed surface' data type in the description of free form objects or parts of solids has provided greater power and flexibility to these representational schemes. A trimmed surface is an ordinary tensor product surface that has a restricted parameter domain, thus overcoming the limit of tensor product surfaces defined over rectangular regions, and allowing for arbitrary domains.

This work is not intended as an exhaustive survey on trimmed NURBS surfaces, but as a description of our experience of geometric modeling using trimmed NURBS surfaces. Our experimental testbed is *xcmodel*, a system developed at the University of Bologna [9] that is based on trimmed NURBS surfaces, which tests new modeling ideas and where techniques can easily be experimented and evaluated. This system is the result of our experience of over 10 years in NURBS research and development and it combines new and well-tried approaches. In fact, our work methods are well expressed by the following sentence from Voelker and Requicha [44]:

"It is important to do both theoretical research and experimental system building. They are synergistic, and the exclusive pursuit of either can lead to sterile theory or quirky, opaque systems."

*This work was supported by MURST, Cofin97, Numerical Analysis: Methods and Mathematical Software.

†Dept. of Mathematics, University of Bologna, Italy; E-mail: casciola@dm.unibo.it.

‡Dept. of Mathematics, University of Bologna, Italy; E-mail: morigi@dm.unibo.it.

Everyone agrees with this idea, although, in the academic world, it is hardly ever rewarded.

The paper is structured as follows: in Section 2 the basic definitions of trimmed NURBS surface are given. In Section 3 we address the problem of representing the trimmed regions, while Section 4 deals with several approaches used for rendering trimmed surfaces. In Section 5 and 6 we describe how these surfaces can be used for solid and surface modeling purposes.

2. Trimmed NURBS surfaces. A trimmed NURBS surface can be defined by a tensor product NURBS surface and a set of trimming curves in the parametric space of the surface.

A NURBS surface of order (m, n) defined in the parametric domain $U \times V = [0,1] \times [0,1]$, can be represented as:

$$\mathbf{r}(u,v) = \frac{\sum_{i=1}^{n+H} \sum_{j=1}^{m+K} w_{ij} \mathbf{P}_{ij} N_{i,n}(u) N_{j,m}(v)}{\sum_{i=1}^{n+H} \sum_{j=1}^{m+K} w_{ij} N_{i,n}(u) N_{j,m}(v)} \tag{2.1}$$

where w_{ij} are the weights, $\mathbf{P}_{ij}$ are the control points, and $N_{i,n}(u), N_{j,m}(v)$ are the order n and m B-splines, respectively, defined over the knot vectors

$$U = (\underbrace{0, \cdots, 0}_{n}, u_{n+1}, \cdots, u_{n+H}, \underbrace{1, \cdots, 1}_{n})$$

$$V = (\underbrace{0, \cdots, 0}_{m}, v_{m+1}, \cdots, v_{m+K}, \underbrace{1, \cdots, 1}_{m}).$$

Associated with a NURBS surface is a set of planar, closed, non-intersecting curves that can be conveniently represented as NURBS curves defined in the parameter domain $[0,1]$ by

$$\mathbf{c}_k(t) = (x_k(t), y_k(t)) = \frac{\sum_{i=1}^{p_k+L_k} w_i \mathbf{P}_i N_{i,p_k}(t)}{\sum_{i=1}^{p_k+L_k} w_i N_{i,p_k}(t)}, \quad k = 1, \cdots, M, \tag{2.2}$$

with the knot vectors

$$T_k = (\underbrace{0, \cdots, 0}_{p_k}, t_{p_k+1}, \cdots, t_{p_k+L_k}, \underbrace{1, \cdots, 1}_{p_k}).$$

A trimmed NURBS surface is given by the restriction of $\mathbf{r}(u,v)$ to a subdomain $D \subset U \times V$, of the parametric space, named *trimmed region*. This domain D is defined as the set of regions on $U \times V$ whose boundaries are specified by the trimming curves and a given criteria. This criteria allows us to identify the part of the surface that remains when discarding all the holes defined by the trimming curves. The trimmed surface boundaries are obtained by mapping the 2D trimming curves onto the surface, that is,

$$\mathbf{r}(x_k(t), y_k(t)), \quad k = 1, \cdots, M.$$

Trimmed NURBS surfaces have been adopted by the CAD/CAM industry, and included in graphics standards. As such, trimmed NURBS surfaces are provided as modeling primitives in several geometric modeling software systems, and the rendering of trimmed NURBS surfaces is supported by international standards, such as PHIGS+, as well as graphics programming interfaces, such as Iris-GL and OpenGL (Silicon Graphics, Inc.), Starbase (Hewlett-Packard Corp.), and Renderman (Pixar).

Trimmed surfaces are essential for the modeling of non-regular boundary objects, generated by trimming away part of the rectangular patch. Trimming patches also play a fundamental role in the boundary description of solid models where the trimmed surfaces can give a complete representation of the boundary of a sculptured solid primitive by means of the union of surfaces restricted to suitable domains. They are the result of Boolean operations on solid objects bounded by NURBS surfaces [17].

3. Representing of trimmed NURBS surfaces. We focus now on the problem of representing the trimmed region D, providing three different representations that define the same domain D.

Definition 1: handedness rule of trimming

The curves $\mathbf{c}_k(t)$ are considered all properly oriented and joined to form a $N < M$ numbers of outer or inner loops. Outer loops are oriented counter-clockwise, whereas inner loops are oriented clockwise. The domain of the trimmed surface is defined as the common region within the outer boundary (corresponding to the outer loops) and outside the inner boundaries (corresponding to inner loops), including boundary curves [32]. Figure 3.1a presents an example of trimmed region obtained by this rule; the area that is part of the trimmed region is shaded.

Definition 2: winding rule of trimming

The curves $\mathbf{c}_k(t)$ are considered all properly joined to form $N < M$ closed loops (see Figure 3.1b). The set of contours divides the supporting surface into an inner or retained region and an outer or discarded region based on the odd-winding rule. According to this rule, the region of the surface that is enclosed by an even number of loops is trimmed out [29].

Definition 3: 2D CSG tree

The trimmed domain is represented by the 2D equivalent of a CSG (Constructive Solid Geometry) tree [5], which, like its 3D counterpart, is a collection of half-spaces and Boolean operation symbols [17, 5]. Each leaf of the CSG tree structure is an embedded non-intersecting closed curve. Curves at the same level are disjoint to each other.

The trimmed region that is kept can be determined by classifying a single point. Conventions can be assumed to consider alternate levels of the tree to be outer and inner loops, starting from an inner/outer top level. In Figure 3.1c the 2D CSG tree representing the trimmed region illustrated in Figure3.1a is shown with the top level set to be outer.

The trimmed region looks like a set of islands and lakes, where the islands represent part of the trimmed region, while the lakes are the holes in it. The algorithm used to build a 2D CSG tree $\mathcal{T}$ for a given trimmed region forms a union of islands (R_i), and subtracts from each island all of its lakes (S_{ij}):

$$\mathcal{T} = \bigcup_{i=1}^{I} (R_i \bigcap (\bigcup_{j=1}^{L_i} S_{ij})),$$

where I is the number of islands and L_i is the number of the lakes S_{ij} children of island i.

Another approach for representing a trimmed surface is based on the idea of decomposing the domain D of the parameter space using a set of planar subregions, whose union defines the entire domain D (see Figure 3.1d). This approach does not follow an intuitive idea of trimming out the part of the entire patch bounded by the trimming curves, that is preserved somehow in the previous definitions, but it allows us to manage a set of surfaces, defined on an irregular domain, that do not contain any trimming curves. This can represent an advantage during the evaluation/rendering phase, as well as in the mesh generation phase, because it allows for a representation of a trimmed tensor product patch as a collection of untrimmed patches. In [22] the authors consider this approach by defining a trimmed surface as the union of a set of planar, ruled (four-sided) subregions, addressing the problem of data exchange between systems. More generally, this approach can be applied using S-patches [28, 45]. S-patches are rational generalizations of Bezier-surfaces that admit any number of boundary curves.

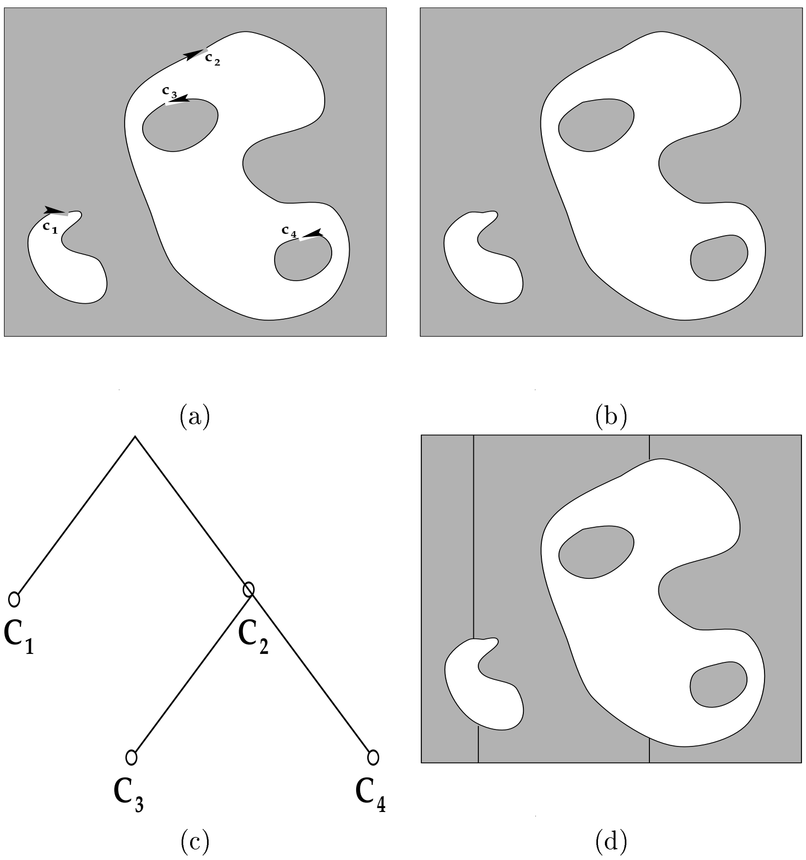

FIG. 3.1. *Example of trimmed region obtained by: (a) definition 1, (b) definition 2, (c) definition 3, and by untrimmed regions approach (d).*

4. Rendering of trimmed NURBS surfaces. This section deals with the problem of visualizing the trimmed NURBS surfaces. As we know, the visualization algorithms differ in rendering quality with a computational cost that increases as the quality improves. A modeling system needs both medium/low-level quality algorithms, allowing for real-time visualization (such as depth cueing, hidden-line or z-buffer), and high quality algorithms for a realistic final rendering (such as ray-tracing).

Medium/low-level quality algorithms use a piecewise planar approximation (tessellation or triangulation) of the trimmed surface within a given tolerance. After the triangulation phase a surface can be rendered in real time using the triangle rendering capabilities common in current graphics systems. Several algorithms have been developed in the past to obtain a triangulation of a trimmed NURBS surface [26, 32, 35, 40, 29]. These methods are parametrization dependent and use the same tolerance to polygonize the trimming curves and to triangulate the surface. Only Piegl and Tiller in [33] present an algorithm, which is not parametrization dependent, that solves the problem with different tolerances for the trimming curves and the surface. The triangulation algorithm implemented in *xcmodel* was inspired by the latter proposal and consists in the following steps:

- *grid surface*; a rectangular grid in the entire domain is created adaptively. By splitting each rectangular into two triangles, we obtain a piecewise linear approximation of the surface. The goal is to approximate curved regions using more triangles than for the flat regions, while ensuring that the approximation satisfies the requirements of a given tolerance [1];
- *polygonize trimming curves*; consists in obtaining a piecewise linear approximation of the trimming curves within a given tolerance.
- *merge trimming polygon and rectangular grid*; given a rectangle in the grid of the parameter domain, along with its status and a set of trimming polygons, the goal is to compute a set of closed polygons that bound the trimmed domain lying inside the given rectangle.
- *triangulating domain polygons*; this problem has been solved in a simple and heuristic way. Our approach distinguishes two cases: the points on the rectangular grid may or may not be consecutive polygon vertices. In the first case, the triangles open fanwise from one of the grid points; in the second case, where two grid points are opposite, the triangles form a seam.

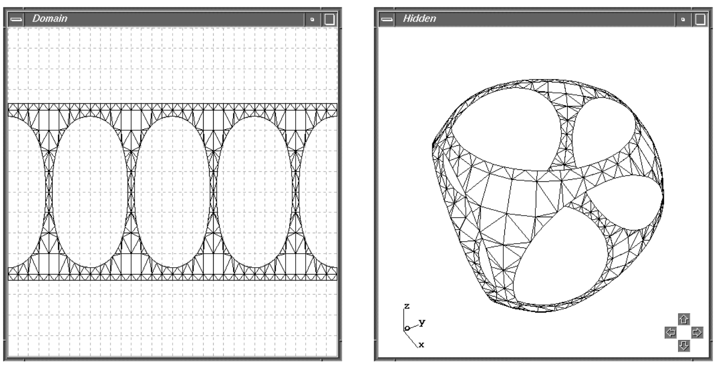

FIG. 4.1. *Rendering of trimmed NURBS sphere with a hidden-line algorithm; a triangulation of the trimmed domain (left), the rendered trimmed surface (right).*

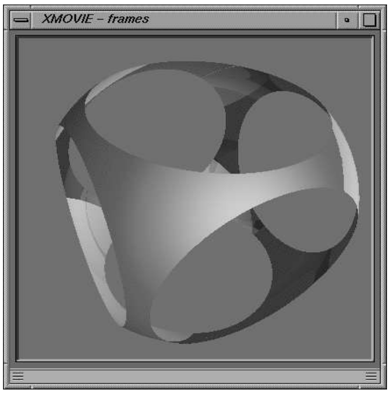

FIG. 4.2. *Rendering of trimmed NURBS sphere with a ray-tracing algorithm.*

Figure 4.1 illustrates an example of a trimmed NURBS sphere rendered by a hidden-line algorithm obtained using *xcmodel*.

High quality rendering algorithms use an exact representation. The history, theory and capabilities of ray-tracing algorithms are well known and documented [20]. The basic problems in ray-tracing trimmed NURBS surfaces are the following:

- ray/patch intersection
- determining if the intersection points belong to a trimmed patch.

If the trimming is caused by a Boolean operation involving solid geometric models, the latter step can be performed using conventional CSG methods [37]. Our *xcmodel* system manages boundary representations for solids by using trimmed surfaces and therefore the second step determines wether an intersection point lies inside or outside a trimmed region using a 2D PMC (Point Membership Classification) algorithm (see section 5).

Solutions to the ray/patch intersection problem can be categorized as being based on subdivision, numerical or hybrid techniques. Subdivision approaches are described by Whitted [46], Rogers [36] and Woodward [47]. These algorithms are based on the convex-hull properties of NURBS surfaces. If the ray does not intersect the convex-hull of the control points, it does not intersect the patch. By recursively subdividing the patch and checking convex-hulls, the intersection points can be computed at a linear convergent rate amounting to a binary search. Whitted's algorithm operates in three dimensions, whereas Rogers and Woodward map the problem in two dimensions.

Numerical solutions of the ray/patch intersection problem include those developed by Toth [43], Sweeny and Bartels [42] and Joy and Bhetanabhotla [24]. Toth's algorithm is based on interval Newton iterations. It works robustly on any parametric surface for which bounds onto the surface and its first derivative can be obtained. Sweeny and Bartels proposed refining the control mesh using the Oslo algorithm [15] until the mesh closely approximates the surface. The ray intersection is then computed by intersecting the control mesh with the ray, and using that intersection point as a starting point for Newton iteration. Joy and Bhetanabhotla's algorithm uses quasi Newton optimization to capture the point/s on the patch nearest the ray, including the intersection points.

By hybrid solutions, we refer to Bezier clipping [31]. This algorithm uses the convex-hull property in a more powerful manner, by determining parameter ranges which guarantee that they do not include points of intersection. Bezier clipping has the flavor of a geometrically based interval Newton method, and thus may be categorized as partly a subdivision based algorithm and partly a numerical method.

From a numerical point of view all these methods find the ray/patch intersections by solving a non-linear system of equations. The above methods are global and when several solutions occur, they split the problem. The only one which uses a local convergence method after applying an initial global localization step is Toth's algorithm. However, the localization step is very expensive from a computational point of view, as well as the fact that, in the case of near solutions in the parametric domain this localization step can fail, requiring the application of a binary subdivision method of the function.

In our system we implemented first the Toth's proposal, then Bezier clipping and recently we added a new algorithm following one of our ideas. This new approach was called "Toth speed". It is a combination of the Toth and Bezier clipping algorithms that outperforms their performances [30, 41]. The idea is to reduce the application of the interval Newton iterations of the Toth method, trying to find the solution using a simple Newton, in the knowledge that, if a solution exists, it is unique. If this fails, an interval Newton iteration is applied. Moreover, when the Toth algorithm is forced to use binary subdivision, our method uses Bezier clipping.

All these algorithms are included in the system and can be compared on complex ray traced scenes. Note that a ray-tracing algorithm exploits a certain number of optimization techniques in order to speed up performance, such as subdividing a 3D scene in voxels, subdividing the NURBS surface in rational Bezier patches, further subdividing each rational Bezier patch until a given flat tolerance is reached [19]. These optimizations can improve a ray/patch intersection in different ways.

The optimizations mentioned in the case of trimmed surfaces, introduce a new problem: the subdivision of a patch implies the redefinition of the trimmed domain. The solution to this problem may be more or less complex, depending on the type of representation used for the trimmed region.

Figure 4.2 shows an example of a trimmed NURBS sphere rendered by a ray-tracing algorithm obtained by *xcmodel* .

5. Solid modeling. Following the B-rep (Boundary representation) of a solid, we can define as 'primitive solid' any solid that has, as its boundary, a single surface that is simply closed in order to separate the space into two parts, one of which will be enclosed. According to convention, the boundary surface must be parametrized in such a way that the surface normal vector indicates the area outside the solid. For example, spheres, cylinders, etc. are included in this definition, as well as closed sculptured solids. Primitive solids can also be combined by Boolean operations, such as intersection, union and difference, in order to create a complex solid. The difficulties posed by Boolean combinations of free form objects have been overcome through the use of trimmed surfaces.

The boundary of a solid object S is defined as follows

$$bS = \bigcup_{i=1}^{k} \mathbf{r}_i(D_i) \tag{5.1}$$

where D_i is the trimmed region associated with the surface $\mathbf{r}_i$. A primitive solid is then defined by $k = 1$ and $D_1 = U \times V$. The boundary of the solid S is a closed surface.

Given two solids A and B, respectively defined by their boundary surfaces, the Boolean operation problem consists in determining the boundaries of the solids $A \bigcup B$,

$A \bigcap B$, $A - B$ starting from bA and bB. We will refer to this process as *the set operation algorithm*.

The boundary of the resulting solid is given by the equations known as the boundary formula [34]:

$$
\begin{aligned}
b(A \bigcup B) &= (bA \bigcap cB) \bigcup (bB \bigcap cA) \\
b(A \bigcap B) &= (bA \bigcap iB) \bigcup (bB \bigcap iA) \\
b(A - B) &= (bA \bigcap cB) \bigcup (bB \bigcap iA)
\end{aligned} \tag{5.2}
$$

where iX and cX represent, respectively, the interior and the complement of the solid X.

The trimmed patches of the resulting solid are the same as those of bA and bB, except for the portion of the patches that have been trimmed away (see Figure 5.1).

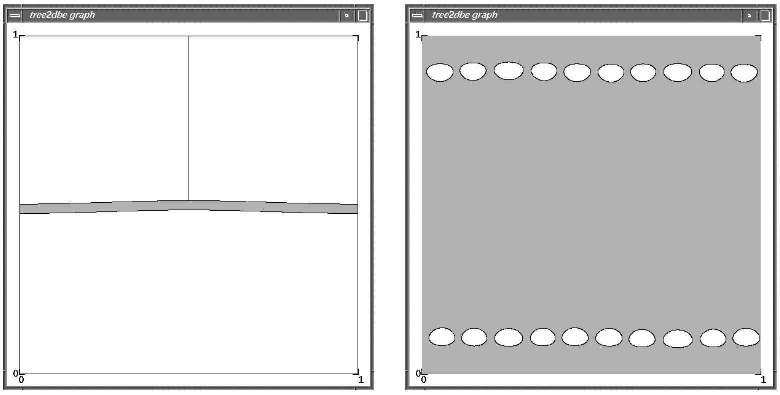

FIG. 5.1. *Solid modeling; trimmed domains for the solid finger disk in Fig.5.2; the cylinder part of the solid (left), the disk part of the solid (right).*

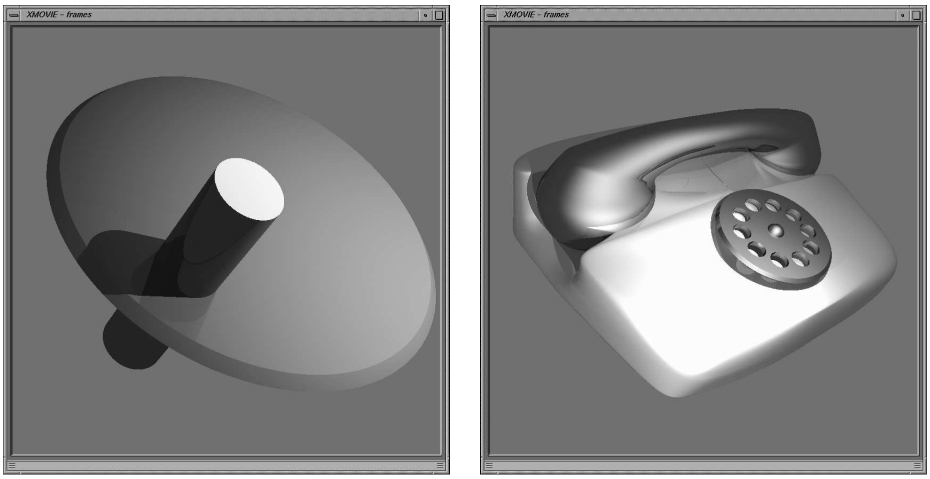

FIG. 5.2. *Solid modeling; primitive solids to compose the finger disk by difference (left), the resulting scene (right).*

In [5] and [6], a set operation algorithm is given, that operates on solids modelled with trimmed patches. In [17], the trimmed patch is represented by a dual representation, i.e. the trimmed patch is given both by an implicit surface and a parametric patch, and a basic theory is presented for a set operation algorithm, which operates on the dual representation. In [7] the proposal in [5] is taken up, adapted for NURBS surfaces and implemented in the *xcmodel* system. A complete description and implementation of a set operation algorithm is technically complex. The basic idea in [7] consists in avoiding the intersections between trimmed patches by considering the intersections between the patches over the whole domain. After this, the trimmed regions (2D CSG tree) resulting from the intersection between the trimmed regions of the solids and the trimmed regions obtained by the patch/patch intersection operations, are determined. In the following, we will give a brief description of untrimmed surface/surface intersection (SSI), 2D and 3D PMC and 2D curve/curve intersection (CCI), which represent the basic steps involved in this process.

5.1. SSI. The methods available for intersecting patches can be divided into two categories: curve following and subdivision. Firstly, *the curve following method* finds some points of intersection. Then, the points on the intersection curve are detected using a numerical method [1, 2]. This method is especially useful for processing singular cases of intersection [3]. The *subdivision method* divides the problem into smaller problems by approximating the patch into simpler linear or quadratic subpatches [23, 6]. The patches are intersected, resulting in curves that are then refined by another numerical method. A modified version proposed in [1] has been implemented in the *xcmodel* system [8]. Here, the balance between robustness and efficiency, a problem for SSI algorithms, is solved by a "little trick". The four main steps are the following:

- *Adaptive mesh generation*; this consists in the approximation of each of the surfaces using an adaptive grid of isoparametric curves, approximated by piecewise linear curves within a given tolerance. From this grid we easily obtain a surface triangulation;
- *Initial intersection point generation*; the intersections between the grid segments of a surface and the triangles of the other surface are determined in order to obtain at least one starting point for each intersection curve;
- *Following an intersection curve*; with an initial point on the intersection curve, we move along the curve by steps. To do this, first of all we find an estimate for the next point on the intersection curve, and then we evaluate an exact intersection within a given tolerance;
- *Sorting*; finally, the curve intersection segments found in the preceding step are linked together in order to obtain complete intersection curves.

5.2. 2D and 3D PMC. Point Membership Classification is a function that takes, as its input, a point P and a closed set S, and which returns one of three possible outcomes: P is inside S, P is on the boundary of S, or P is outside S. The set operation algorithm makes extensive use of PMC in both two and three dimensions. During the construction of the 2D CSG tree, a single 3D point is classified with respect to the opposing solid. The result will be used for further 2D classifications in order to build a collection of half-spaces which, ultimately, become the leaves of the tree. A single concept is used for both 2D and 3D classification: a ray is extended from the point to be classified, and the number of intersections of the ray with the boundary of the set determines the membership status of the point. An even number of intersections means that the point is outside the set, while an odd number implies that the point is inside. While the concept is the same, its implementation is very different for two or three dimensions. In the 2D case, the ray is intersected with polygons or curved boundaries. In the case of curved boundaries all the roots in the parametric interval [0,1] of the scalar NURBS function resulting from the ray/curve intersection must be found. To solve this problem the methods in [21, 27, 4, 31] are explored and compared [41]. In the 3D case, the boundary consists of trimmed patches, and a ray/patch method is required (see section 4).

5.3. CCI. This problem can either be tackled geometrically, that is, using subdivision techniques for the curves, that exploit the convex-hull property of the NURBS curves [27, 25, 38], or numerically. The latter is done either by finding all the zeros of the function resulting from the system, when a curve has been implicitized [39] (this is profitably applicable only for low degree curves), or by finding the zeros of the vector function in two variables (2D surface), obtained from the difference between the curves [16]. The problem of finding the zeros of a 2D surface can be solved using the same methods as those used to determine the ray/patch intersections, (see section 4).

Figure 5.2 shows an example of solid modelling generated by the *xcmodel* system.

6. Surface modeling. The refinement technique for spline surfaces is a well known tool used to represent a surface defined by a finer control mesh with the feature that each single control point will have a more local effect on the surface. This allows us to model small patches and to modify their shape without affecting the rest. This idea led to the definition of Hierarchical Spline Surfaces (HSS) [18], proposed in 1988 by Forsey and Bartels, enabling a surface to be refined and details added, using a hierarchy, whose levels correspond to the varying levels of refinement of the surface patches. This technique enables us to restrict the influence of refinement to the relevant parts of the surface.

Our paper illustrates how the *xcmodel* system uses trimmed NURBS surfaces to represent HSS and enables modeling to be carried out with these surfaces. For this purpose *rectangular trimmed NURBS surfaces* are defined. They consists in a particular class of trimmed surface, whose trimmed away regions are defined as rectangular in the parametric domain.

We can represent an HSS as a NURBS surface $\mathbf{r}$ and a set of regions D_i as follows:

$$\bigcup_{i=1}^{n} \mathbf{r}(D_i)$$

where each region D_i represents the ith level of detail of an HSS and is defined by

$$D_i = A_{i-1} - A_i \quad \text{with} \quad A_0 = U \times V, \quad A_i = \bigcup_{j=1}^{m_i} \hat{A}_{ij}$$

and where $\hat{A}_{ij} = [u_{a_j}, u_{b_j}] \times [v_{c_j}, v_{d_j}]$ represents the jth rectangular trimmed region at level i, and is a subset of A_{i-1}. Note that $\bigcup_{i=1}^{n} D_i = U \times V$.

The modeling phase, by adding or editing details, consists in first selecting the rectangular patch on the surface to be modified, then determining the corresponding rectangular subdomain and trimming it so that the domain becomes restricted to the chosen patch, which now acts as an autonomous and untrimmed surface. The appropriate refinement is carried out and the control points that can be moved without altering their order of continuity with the original surface are determined. In the initial HSS work [18], the control points that cannot be altered in order to avoid losing the original continuity are already clearly specified. These form a frame $m-1$ wide and

$n-1$ high in the control mesh, where m and n are the NURBS surface orders. Any change in position and/or weight of all or only some of the control points within this frame results in a local modification of the patch, while maintaining the original continuity with the rest of the surface.

In Figure 6.1 we show an example of HSS modeling using trimmed NURBS.

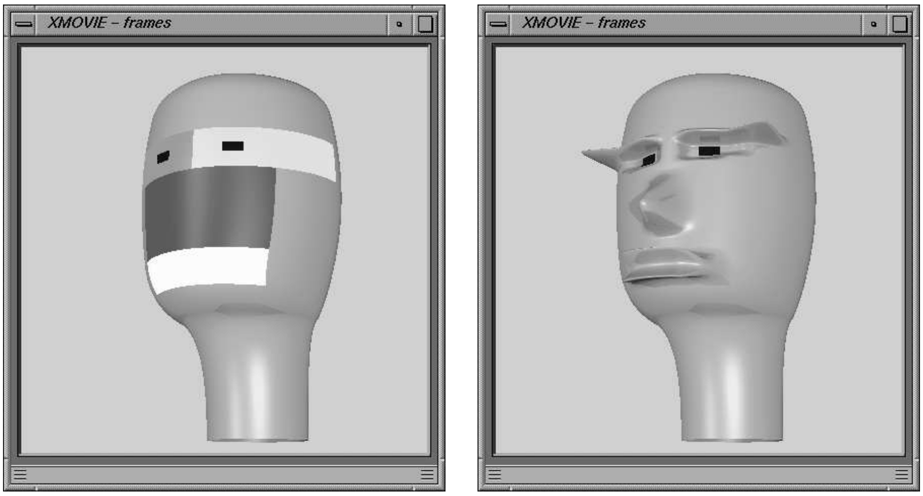

FIG. 6.1. *HSS modeling using trimmed NURBS; on the left three levels of rectangular patches are shown, on the right the modeled surface is given.*

Furthermore, in order to design multi-sided surfaces, it makes perfect sense to consider a modeling procedure that trims one surface by means of another surface, even if no set operation is indicated. To do this, the *xcmodel* system allows the user to model the 2D trimming curves and define a 2D CSG tree for a given surface.

7. Conclusions and new trends. In this paper we have discussed the usefulness of the trimmed NURBS paradigm for modeling.

Trimmed NURBS present at least two difficulties:

- trimming is expensive and prone to numerical errors;
- it becomes difficult to maintain smoothness at the seams of the patchwork, when the model is animated.

Recently, subdivision surfaces have been considered as a way of overcoming NURBS topological limitations without mesh degeneracy. They do not require trimming, and model smoothness is automatically guaranteed, even in animated models. The idea is to refine an irregular/regular mesh, by creating a new mesh that approximates the old one. By repeating this process, a smooth surface is formed as the limit of the process itself. Subdivision algorithms are quite simple, and are generally able to produce quite complex objects, although the basic theoretical modeling background has yet to be explored, and effective, intuitive user interface, designed to exploit the potential of the representation to the full, requires further development. Although trends in geometric modeling community research are moving towards an exhaustive exploration of the subdivision paradigm, at the moment, trimming modeling still remains the most complete and powerful means for free form object modeling.

REFERENCES

[1] R.E.Barnhill, G.Farin, M.Jordan, B.R.Piper, *Surface Surface Intersection*, Computer Aided Geometric Design, 4 (1987), pp. 3–16.

[2] R.E.Barnhill and S.N.Kersey, *A marching method for parametric surface/surface intersection*, Computer Aided Geometric Design, 7 (1990), pp. 257–280.

[3] C.L.Bajaj, J.E.Hoffman, R.E.Lynch, *Tracking surface intersection*, Computer Aided Geometric Design, 5 (1988), pp. 285–307.

[4] R.Brent, *Algorithm for minimization without derivatives*, Prentice-Hall, 1973.

[5] M.S. Casale, *Free-form solid modeling with trimmed surface patches*, IEEE Computer Graphics & Applications, Jan. (1987), pp. 33–43.

[6] M.S. Casale and J.E.Bobrow, *A set operation algorithm for sculptured solids modeled with trimmed patches*, Computer Aided Geometric Design, 6 (1989), pp. 235–247.

[7] G.Casciola and B.Quaquarelli, *Primitive solide, operazioni Booleane e NURBS*, Proceedings of ICO-GRAPHICS'90, Milano, (1990).

[8] G.Casciola and S.Morigi, *Il problema SSI nella modellazione solida con superfici NURBS*, Atti dell'Accademia delle Scienze dell'Istituto di Bologna, serie V, n.6, (1995).

[9] G.Casciola and S.Morigi, *xcmodel: an aCADemic system*, Proceedings of Cofin.97, Ferrara, January 19-21, (2000).

[10] G.Casciola, *xcmodel: a system to model and render NURBS curves and surfaces* User's Guide (1999), http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[11] G.Casciola, *xccurv: the 2D modeller* User's Guide (1999), http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[12] G.Casciola, *xcsurf: the 3D modeller* User's Guide (1999), http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[13] G.Casciola, *xcbool: the object composer* User's Guide (1999), http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[14] G.Casciola, *xcrayt: the scene descriptor* User's Guide (1999), http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[15] E.Cohen, T.Lyche, R.F.Riesenfeld, *Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics*, Computer Graphics and Image Processing, 14 (1980), pp. 87–111.

[16] T.Dokken, *Finding intersections of B-splines represented geometries using recursive subdivision techniques*, Computer Aided Geometric Design, 2 (1985), pp. 189–195.

[17] R.T.Farouki, *Trimmed-surface algorithms for the evaluation and interrogation of solid boundary representations*, IBM J.RES.DEVELOP. , 31 (1987), pp. 314–333.

[18] D.R.Forsey and R.H.Bartels, *Hierarchical B-spline refinement*, Proceeding SIGGRAPH'88, (1988). In Computer Graphics, 22 (1988), pp 205–212.

[19] A.Fujimoto, T. Tanaka, and K. Iwata, *ARTS: accelerated ray-tracing system*, IEEE Comp. Graphics and Appl., Apr. (1986), pp. 16–26.

[20] A.Glassner ed., *Introduction to ray tracing*, Academic Press, 1989.

[21] A.Grandine, *Computing zeroes of spline functions*, Computer-Aided Geometric Design, 6 (1989), pp 129–136.

[22] B.Hamann, P.-Y. Tsai, *A tessellation algorithm for the representation of trimmed NURBS surfaces with arbitrary trimming curves*, Computer-Aided Design, 28 (1996), pp. 461–472.

[23] E.G.Houghton, R.F.Emnett, J.D.Factor, C.L.Sabharwal, *Implementation of a divide-and-conquer method for intersection of parametric surfaces*, in R.E.Barnhill, W.Boehem, eds. Surfaces in Computer Aided Geometric Design (1984), North-Holland, Amsterdam.

[24] K.Joy, M.Bhetanabhotla, *Ray tracing parametric patches using numerical techniques and ray coherence*, Proceeding of SIGGRAPH'86, (1986). In Computer Graphics, 20 (1986), pp.279–285.

[25] P.A.Koparkar and S.P.Mudur, *A new call of algorithms for the processing of parametric curves*, Computer-Aided Design, 15 (1983), pp 35–46.

[26] S.Kumar and D.Manocha, *Efficient rendering of trimmed NURBS surfaces*, Computer-Aided Design, 27 (1995), pp. 509–521.

[27] J.M.Lane, R.F.Riesenfeld, *A theoretical development for the computer generation and display of piecewise polynomial surfaces*, IEEE Transaction on PAMI, 2 (1980), pp.35–46.

[28] C.T.Loop and T.D.DeRose, *A multisided generalization of Bezier surfaces*, ACM Transaction on Graphics, 8 (1989), pp.204–234.

[29] W.L.Luken, *Tessellation of trimmed NURBS surfaces*, Computer Aided Geometric Design, 13 (1996), pp 163–177.

[30] M.Marini, D.Grilli, *Un algoritmo di ray-tracing per superfici spline razionali trimmate*, Master

Thesis in Compuer Science, University of Bologna, (1995).
[31] T.Nishita, T.W.Sederberg and M. Kakimoto, *Ray tracing trimmed rational surface patches*, ACM Computer Graphics, 24 (1990), pp. 337–345.
[32] L.A.Piegl and A.M.Richard, *Tessellating trimmed NURBS surfaces*, Computer-Aided Design, 27 (1995), pp. 16–26.
[33] L.A.Piegl and W. Tiller, *Geometry-based triangulation of trimmed NURBS surfaces*, Computer-Aided Design, 30 (1998), pp. 11–18.
[34] A.A.G.Requicha, *Mathematical models of rigid solids*, Tech. Memo. 28, Production Automation Project, University of Rochester, (1977).
[35] A.Rockwood, K.Heaton and T.Davis, *Real time rendering of trimmed surfaces*, Proceeding of SIGGRAPH'89, (1989). In Computer Graphics, 23 (1989), pp. 107–116.
[36] D.Rogers, *Procedural elements for computer graphics*, McGraw-Hill, 1985.
[37] S.Roth, *Ray casting for solid modeling*, Computer Graphics and Image Processing, 18 (1982), pp. 109–144.
[38] T.W.Sederberg and T.Nishita, *Curve intersection using Bezier-clipping*, Computer-Aided Design, 24 (1990), pp. 538–549.
[39] T.W.Sederberg and Parry, *Comparison on three curve intersection algorithm*, Computer-Aided Design, 18 (1986), pp 58–63.
[40] X.Sheng and B.E.Hirsch, *Triangulation of trimmed surfaces in parametric space*, Computer-Aided Design, 24 (1992), pp. 437–444.
[41] S.Spagna, *Analisi, sviluppo e applicazione di metodi di intersezione per spline razionali*, Master Thesis in Compuer Science, University of Bologna, (1998).
[42] M.Sweeney and R.Bartels, *Ray tracing free-form B-spline surfaces*, IEEE Computer Graphics & Applications, 6 (1986) pp.41–49.
[43] D.Toth, *On ray tracing parametric surfaces*, Proceeding of SIGGRAPH'85, (1985). In Computer Graphics, 19 (1985) pp.171–179.
[44] H.B.Voelcker, A.G.Requicha, *Research in solid modelling at the University of Rochester: 1972-87*, in Fundamental developments of Computer-Aided Geometric Modeling, L.A.Piegl ed., London, Academic Press 1993.
[45] J.Warren, *Creating multisided rational Bezier surfaces using base points*, ACM Transaction on Graphics, 11 (1992) pp.127–139.
[46] T.Whitted, *An improved illumination model for shaded display*, Communication ACM, 23 (1980), pp. 96–102.
[47] C.Woodward, *Ray tracing parametric surfaces by subdivision in viewing plane*, in W.Strasser and H.-P.Seidel, eds. Theory and Practice of Geometric Modeling, Springer Verlag (1989), pp.273–290.