XCMODEL 3.0 — AN ACADEMIC MODELING/RENDERING SYSTEM

Giulio Casciola
Dipartimento di Matematica, University of Bologna - P.zza di Porta S. Donato 5 – Bologna, Italy
Phone: +39 51 2094439; Fax: +39 51 2094490; Email: casciola@dm.unibo.it

Roberto Sottile
Dipartimento di Scienze dell'Informazione, University of Bologna - Mura Anteo Zamboni 7 – Bologna, Italy
Email: sottile@cs.unibo.it

Abstract:

Non-Uniform Rational B-Splines (NURBS) is a well-established tool for geometric design having become the de facto industry standards for the representation, design, and data exchange of geometric information. 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 the standard NURBS representational schemes, overcoming the limit of tensor product surfaces defined over rectangular regions, and allowing for arbitrary domains.*

xcmodel is a modeling/rendering system realized and usable in an academic environment which summarises our knowledge and experience in geometric modeling and NURBS curves and surfaces acquired over ten years of research. xcmodel and its subsystems were designed to represent a research and teaching laboratory to experiment and learn, so it is an ideal environment to develop, perfect and compare methods and algorithms in geometric modeling and graphics visualization, assuming trimmed NURBS as the only shape primitive for its purposes. In this paper we present the third version of the system, describing its functionalities, enhancements and add-ons with respect to the previous versions.

Keywords: Trimmed NURBS, curves and surfaces modeling, real-time rendering, ray-tracing

1 Introduction

xcmodel is a software package entirely developed at the University of Bologna and in rapid and continuous evolution. The system has been developed and improved with the contribution of many graduates and undergraduates in mathematics and computer science, and it has been adopted in many university courses in geometric modeling and computer graphics as a laboratory to experiment and learn the well-known methods and algorithms for modeling and graphics visualization [1].

One of the educational peculiarities of *xcmodel* is that the user is free to interact with any kind of shape, by modifying every single parameter involved in its NURBS description, without any software constraints; this aspect may be extremely important and appreciable for mathematics researchers, students or even the simple enthusiast, also because this *total level of freedom* can not be found in any of the professional or commercial packages based on NURBS. Thanks to its expressive power, *xcmodel* can be seen as an ideal environment to learn, create, test, or simply experiment new modeling ideas and where graphics techniques can easily be implemented and evaluated. This is why the system assumes the user to be familiar with the mathematics behind the representations used in the modeling systems.

A brief description of *xcmodel* internals, subsystems and modules is presented in Section 2. In Section 3 some basic definitions about the modeling primitives involved in the system are given. Section 4 considers some basic and advanced geometric modeling tools included in the system, along with the main rendering techniques adopted in *xcmodel.* In Section 5 we show the last enhancements and add-ons from the previous version of the system, then a brief history and a preview of future work is presented in Section 6.

2 The *xcmodel* system

At the present time, the *xcmodel* system runs on UNIX platforms such as Sun SPARC (Solaris), SGI (Irix), and Intel (Linux), but we expect problems to be minimal under other environments. The system is a collection of modules and libraries entirely implemented in C ANSI programming language.
System modules: the *xcmodel* system is based on four independent graphic subsystems: *xccurv*, the 2D NURBS curves modeler [4]; *xcsurf*, the 3D NURBS curves and surfaces modeler [5]; *xcbool*, the boolean object composer [6]; *xcrayt*, the 3D scene descriptor and renderer [7].
System libraries: the Graphical User Interface of the environment is realized using the *xtools* library [11], built on the top of Xlib, the lowest level library of the XWindow System. The graphic icons in the user interface make use of the XPM (XPixMap) library which allows *xtools* to deal with pixmaps. Together with the *xtools* library, other three libraries have been implemented in the system: the *MATRIX* library [8], for vector and matrix computation, the *descriptor* library [9], for scene setting, and the *trim* library [10] for the real time visualization of trimmed NURBS surfaces. The scene descriptor and rendering *xcrayt* package makes use of two modules: *xhrayt/hrayt* for ray-tracing and *xframe* for visualizing the rendered scene. The solid composer *xcbool* module makes use of two independent subsystems: *xcssi*, for surface/surface intersection and *xcdbe*, a conversion tool for trimmed surfaces. Animation is managed by the *xmovie* tool [7] that allows for the visualization of images and image sequences. The *xcmodel* software package is available [3] and freely distributed versions as well as user's guides, data, models and images, can be downloaded from the following URL:

`http://www.dm.unibo.it/~casciola/html/xcmodel.html`

3 Modeling primitives

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:

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

where $w_{ij} > 0$ 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

$$U = (\underbrace{0, \cdots, 0}_{n}, u_{n+1}, \cdots, u_{n+H}, \underbrace{1, \cdots, 1}_{n}) \quad , \quad V = (\underbrace{0, \cdots, 0}_{m}, v_{m+1}, \cdots, v_{m+K}, \underbrace{1, \cdots, 1}_{m}) \tag{2}$$

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

$$\mathbf{c}_k(t) = (x_k(t), y_k(t)) = \frac{\sum_{i=1}^{p_k+L_k} w_i \mathbf{P}_i N_{i,p_k}(t)}{\sum_{i=1}^{p_k+L_k} w_i N_{i,p_k}(t)} \qquad where \;\; k = 1, \cdots, M \tag{3}$$

with the knot vectors

$$T_k = (\underbrace{0, \cdots, 0}_{p_k}, t_{p_k+1}, \cdots, t_{p_k+L_k}, \underbrace{1, \cdots, 1}_{p_k}) \tag{4}$$

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:

$$\mathbf{r}(x_k(t), y_k(t)) \qquad where \;\; k = 1, \cdots, M \tag{5}$$

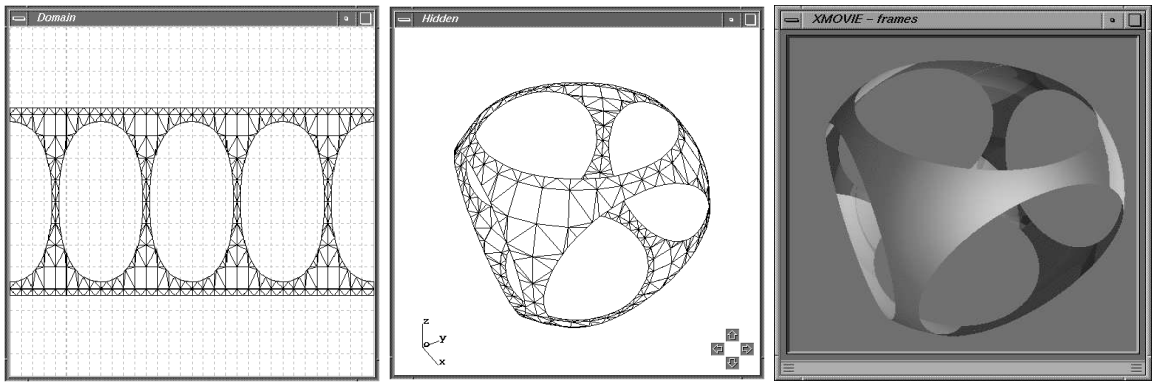

Figure 1: *A trimmed NURBS sphere: tessellation of the trimmed domain (left); the sphere rendered with a hidden-line algorithm (center); the same sphere rendered with a raytracing algorithm (right)*

Figure 1 shows a simple example of a trimmed NURBS surface.

Trimmed NURBS surfaces are 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.), Renderman (Pixar), and many others.

For a detailed discussion of the properties of NURBS curves and trimmed NURBS surfaces we refer the reader to [2] and [13].

4 Modeling tools and Rendering techniques

4.1 Basic geometric tools

In order to manipulate NURBS curves and surfaces, *xcmodel* provides the following fundamental operators: **knot insertion**, the process of expressing a NURBS curve in terms of a related spline representation which has one or more (knot vector *refinement*) additional knots, a fundamental tool to evaluate, subdivide, add control points, reparametrize, and perform a NURBS polygonal approximation; **knot removal**, the reverse process of knot insertion, useful to adjust knot vector after control point insertion/displacement, and to join different curves to form an unique one; **degree elevation**, to represent a spline curve as a curve with an increased degree, used in the "surface from curves" modeling technique to make compatible section curves, or to obtain bi-p-ic parametric surfaces (where e.g. $p = 4$ for bicubic) where the order of the profile and trajectory curves may be required to be the same; **reparametrization**, the process of obtaining a new curve from a given one, which is geometrically identical to the original one, but with a different parametrization. It is used mainly to speed up and enhance the numerical and rendering processes, because they are directly affected by parametrization in terms of computational complexity and numerical stability.

4.2 Curve creation

In *xcmodel* a new NURBS curve can be created by **shape approximation** with *equally spaced*, *periodic*, *uniform*, *chord length* and *manual* extended knot partition; by **interpolation** with *Lagrange*, *Hermite*, *periodic*, *rational cubic C^1 or C^2 (with tension)* methods, choosing from *uniform*, *chord length*, *centripetal*, or *exponential* parametrization with *equally* or *manually* selected NURBS weights; by **weighted-norm approximation** with *standard*, *constrained* or *periodic* least square methods, *uniform*, *chord length*, *centripetal*, *exponential* parametrization, *automatic* or *manual* extended knot partition, and *equally* or *manually* selected NURBS weights. Many analytic tools are included to study the resulting curve, such as *curve length computation*, *equally spaced draw points*, *span highlighting*, test functions such as *slope*, *speed*, *curvature*, and modifying operators such as *geometric transformations* or *geometric constraints*. Also, a new facility (yet in a prototypal stage), is represented by *a multiresolution curve representation* based on wavelets (for details and descriptions refer to [4]).

4.3 Surface creation

A new NURBS surface can be obtained by a set of *basic primitives*, *from scratch*, *imported from file*, or *created starting from 2D/3D curves*. Operators include *piecewise curves extruding*, *skinning*, *sweeping*, *2D/3D curve revolution*, *swinging*, *tubular* and *quasi tubular*, *curves/surfaces inbetweening*, *trimming*, *free form deformations*, *3D scanner data reconstruction*, and many others (see [5]).

Obviously a fine grained modification of any shape is possible by operating on the single control points, knots, weights, u and v degrees, control polygon/net, or by hierarchically modifying a single portion of the shape without affecting the underlying NURBS.

4.4 Rendering

Real time rendering algorithms, such as *wire-frame*, *depth cueing*, *hidden line*, *flat shading*, *Gouraud shading*, *Phong shading* are provided by the *trim* library; the *hrayt* raytracer implements **pure raytracing for trimmed NURBS** basing on three different ray/patch intersection methods: *Toth*, *Bezier clipping* and our new approach, *Toth speed* [7].

Again, we want to stress on the fact that everything in *xcmodel* is user configurable, and where the few system defaults have to be bypassed, the code can be easily modified in order to reflect the user needs.

5 What's new in version 3.0

Version 3.0 differs from the previous versions for many enhancements and add-ons:

- *xccurv*: completely rewritten using the *xtools* library; added multiresolution curve representation; added IGES entity no.126 (2D NURBS) file format compatibility (importing/exporting); new user documentation;
- *xcsurf*: added reconstruction of models from a 3D scanner data acquisition; new functionalities in 3D curve/surface creation; free from deformation (FFD [12]) plugin included; added IGES entity no.126 (2D NURBS) and no.128 (3D NURBS) file format compatibility (importing/exporting);
- *xcrayt*: included new procedural and domain texture mapping features;
- *xtools library*: added nested GUI management and RWindows (Resizeable Windows) object; open source version;
- *trim library*: added flat shading; open source version;
- Bugs: many (but not all) bugs fixed.

6 *xcmodel project* history and future work

Since 1991, free form surfaces and geometric modeling are a matter of research at University of Bologna, Departments of Mathematics and Computer Science. Numerous course projects and master theses progressively converged to the first version of the *xcmodel* system.

In 1997, within the Cofin 98-00 "Analisi Numerica: Metodi e Software Matematico" National Project, the first beta version of the system was presented, and the *xcmodel project* started:

01/20/2000: The official public release date of *xcmodel version 1.0*
12/07/2000: The official public release date of *xcmodel version 2.0*
03/07/2002: The official public release date of *xcmodel version 3.0*

We intend to release the next version of the system (*xcmodel 4.0*) completely open source, with APIs and developer guides as soon as the testing phase will be completed.

References

[1] G. CASCIOLA - S. MORIGI, *xcmodel: an aCADedmic system*, Ann. Univ. Ferrara - Sez. VII - Sc.Mat., Supplemento al Vol.XLV, (2000)

[2] G. CASCIOLA - S. MORIGI, *The trimmed NURBS age*, Advances in Computation: Theory and Practice; Recent Trends in Numerical Analysis, Ed.D.Trigiante, Nova Science Publishers, Inc., (2000)

[3] G. CASCIOLA, *xcmodel: a system to model and render NURBS curves and surfaces, User's Guide - Version 3.0*, (2002)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[4] G. CASCIOLA, *xccurv: the 2D modeller of* xcmodel, User's Guide - Version 3.0, (2002)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[5] G. CASCIOLA, *xcsurf: the 3D modeller, User's Guide - Version 1.0*, Progetto MURST: "Analisi Numerica: Metodi e Software Matematico", Ferrara (2000)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[6] G. CASCIOLA - G. DEMARCO, *xcbool: the object composer, User's Guide - Version 1.0*, Progetto MURST: "Analisi Numerica: Metodi e Software Matematico", Ferrara (2000)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[7] G. CASCIOLA, *xcrayt: the scene descriptor, User's Guide - Version 1.0*, Progetto MURST: "Analisi Numerica: Metodi e Software Matematico", Ferrara (2000)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[8] G. CASCIOLA, *MATRIX library: Programming Guide - Version 1.0*, (1999)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[9] G. CASCIOLA - R. SOTTILE, *descriptor library: Programming Guide - Version 3.0*, (2002)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[10] G. CASCIOLA - G. DEMARCO, *trim library: Programming Guide - Version 1.0*, (1999)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[11] G. CASCIOLA - S. BONETTI, *xtools library: Programming Guide - Version 2.0*, (2002)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[12] G. CASCIOLA - E. TREVISAN, *Free Form Deformation:* xcsurf *(Version 2.0) plugin*, (2002)
http://www.dm.unibo.it/∼casciola/html/xcmodel.html

[13] L.A. PIEGL - W. TILLER, *The NURBS book*, Springer, (1995).