

# CompuGe: A Computational Geometry Learning Environment

Wilson Rivera

Electrical and Computer Engineering Department  
University of Puerto Rico at Mayaguez  
P.O.Box 9042, Mayaguez, Puerto Rico 00681, USA

## ABSTRACT

CompuGe is an user-friendly interface which has been used for educational purposes in computational geometry courses. This computer-assisted learning software is intended to assist students at graduate and undergraduate levels in learning and visualizing basic concepts behind Computer Aided Geometry Design. CompuGe is an example of how using cognitive tools we may achieve a more effective learning process. The user-friendly interface not only provides students the opportunity to learn rapidly but also enables them to explore a wide range of abstract concepts with a significant control of the learning process. This paper presents the efforts in implementing CompuGe and applying it in the learning process.

## KEY WORDS

educational software, interactive learning environment, human-computer interface, computational geometry.

## 1 Introduction

Computational geometry modeling has played a fundamental role in the development of the modern engineering design technology. Applications of computational geometry include aerodynamic and automotive design, solid and structure mechanics, and grid generation. As a consequence, understanding of computational geometry concepts is important for engineering and applied mathematics students at graduate and undergraduate levels.

In order to model complex geometries using computers it is necessary to generate a computer-compatible description of the geometries. Parametric representation of curves and surfaces emerges then as an ideal methodology of representation. A curve can be represented, using a parameter  $t \in [0, 1]$ , as

$$c(t) = [x(t), y(t), z(t)], \quad (1)$$

where  $x(t)$ ,  $y(t)$  and  $z(t)$  are polynomials.

In the late 1950s, the exploration of the use of parametric curves gave as a result Bezier Curves. Paul de Casteljaou [1, 2] and Bezier [3, 4] independently developed Bezier curves, Schoenberg [5] introduced B-splines and de Boor [6] made important contributions to the theory of B-splines and Bezier curves. All progresses in this arena are included into a general theory referred to as Computer

Aided Geometry Design (CAGD) which is itself a subdiscipline within computational geometry.

CompuGe is a computational geometry learning environment in which basic concepts in CAGD can be visualized. The algorithms were implemented using C and the interface was designed using *GuiDe* (Graphical User Interface Designer); Newcomb & Gaither [7]. In this paper we present our efforts in implementing and applying CompuGe. This paper is organized as follows. Section 2 discusses some computational geometry issues in order to show the nature of the concepts that students have to deal with. Section 3 deals with educational issues including discussion of objectives, visualization of curves and surfaces obtained using CompuGe, and the computer user interface implementation. Finally, section 4 gives the conclusions.

## 2 Computational Background

The development of Bezier curves has had a relevant impact in the parametric representation of curves and surfaces.

### 2.1 Bezier Curves

**Definition 2.1** A Bezier curve of degree  $n$  is defined as

$$b^n(t) = \sum_{i=0}^n p_i B_i^n(t), \quad (2)$$

where  $t \in [0, 1]$ ,  $p_i \in \mathbf{R}^3$ ,  $i = 0, \dots, n$ , are called control points and  $B_i^n(t)$ ,  $i = 0, \dots, n$ , are the Bernstein polynomials of degree  $n$  defined by

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n. \quad (3)$$

Bernstein polynomials also satisfy the following recursive formula

$$\begin{aligned} B_0^n(t) &= 1, \\ B_i^n(t) &= 0, \quad \text{if } i \notin \{0, 1, 2, \dots, n\}, \\ B_i^n(t) &= (1-t)B_{i-1}^{n-1}(t) + tB_{i+1}^{n-1}(t). \end{aligned} \quad (4)$$

### 2.2 B-spline Curves

Some shortcomings appear when Bezier curves are used to model complex geometries. First, a high degree is required to appropriately model complicate geometries, but

high degree curves usually are inefficient and numerically unstable. Second, the control achieved using Bezier curves is not sufficiently local. Thus, if a control points is moved, the whole Bezier curve is affected. The solution of these shortcomings is to use a piecewise polynomial representation.

Let  $C(t)$  a curve defined on the interval  $[a, b]$ , and a set of parameter values  $\{t_i\}_{i=0}^L$  such that  $t_0 = a < t_1 < t_2 < \dots < t_L = b$ . Each interval  $[t_i, t_{i+1}]$  is mapped onto a polynomial curve segment so that the entire curve is represented by segments  $C_i(t)$  with  $t_i < t < t_{i+1}$ , and  $i = 0, 1, \dots, L - 1$ . Let  $C_i^{(j)}$  denote the  $j$ th derivative of  $C_i$ . The sequence  $\{t_i\}_{i=0}^L$  is called knot sequence.

**Definition 2.2** Given a knot sequence  $t_0 < t_1 < t_2 < \dots < t_L$ . The  $i$ th B-spline basis function of  $k$ th order denoted by  $N_i^k(t)$  is defined as

$$N_i^1(t) = \begin{cases} 1 & \text{if } t_i \leq t \leq t_{i+1}, \\ 0 & \text{elsewhere} \end{cases}, \quad (5)$$

and

$$N_i^k(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1}^{k-1}(t), \quad (6)$$

for  $k > 1$  and  $i = 0, 1, \dots, n$ .

**Definition 2.3** Given  $n$  control points  $d_0, d_1, \dots, d_n$  and a knot sequence  $\{t_i\}_{i=0}^{n+k}$ , then

$$Bs(t) = \sum_{i=0}^n d_i N_i^k(t), \quad t \in [t_{k-1}, t_{n+1}], \quad n \geq k - 1 \quad (7)$$

is called a B-spline curve of order  $k$  with knot sequence  $\{t_i\}_{i=0}^{n+k}$ .

**Definition 2.4** Given  $n$  control points  $d_0, d_1, \dots, d_n$ , a knot sequence  $\{t_i\}_{i=0}^{n+k}$ , and a weight sequence  $\{w_i\}_{i=0}^n$ , a Nonuniform Rational B-spline (NURBS) curve of order  $k$  is defined as

$$NURBS(t) = \frac{\sum_{i=0}^n w_i d_i N_i^k(t)}{\sum_{i=0}^n w_i N_i^k(t)}. \quad (8)$$

## 2.3 Surfaces

A parametric based surface can be defined as

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \Phi_i(u) \Psi_j(v) \Gamma_{ij}, \quad (9)$$

where  $\Phi_i$ ,  $i = 0, \dots, n$ , are basis functions pertinent to  $u$  parametric direction,  $\Psi_j$ ,  $j = 0, \dots, m$ , are basis functions related to  $v$  parametric direction, and  $\Gamma_{ij}$  is the network of control points. As a consequence, all concepts discussed before may be easily extended to surface modeling.

## 3 Interactive Learning

The previous section gives us an insight of the complex nature of the concepts behind parametric representation of curves and surfaces. It is due to the complexity of this particular study content that traditional teacher-student interactions cannot be mapped directly to interactive computer-delivered learning. In other words, technology may not necessarily be suitable as a presenter of instructional material[8]. Thus, CompuGe might be thought as an additional resource for learning instead of an instructional strategy to replace the traditional classroom interaction.

The educational goal of CompuGe relies on two basic elements. First, the underlying principles of interactive learning environments contend that students learn best by constructing their own knowledge [9]. This requires that students must have exclusive or at least significant control of the learning process. Second, when a computer-assisted learning software strengthens the students intrinsic ambition to understand the study content we may call it a cognitive tool. By using cognitive tools we may reach more effective learning process [10]. This is particularly true on computational geometry where exploration is essential for understanding the abstract concepts behind.

CompuGe is an effort in these directions. The software has been designed to allow students to explore the behavior of different parametric representations so that they can adjust parameters of a particular type of representation in order to see possible changes in curves and surfaces. CompuGe is a flexible system so that students can learn from the software both during lectures with specific examples provided by teachers and after lectures as a explorative learning tool. Teachers can design their own examples in a file format easy to understand by students. These examples can be used in the classroom to introduce the instructional material systematically, and students can use posteriorly these files to explore and create their own examples. In addition, students can use the interactive input mode to explore the behavior of curves and surfaces (see Figure 1).

### 3.1 The User-friendly Interface

Most contemporary graphical user interface design tools are staggeringly complex. Creating a graphical user interface (GUI) using these tools requires a priori knowledge of widget design. These tools are excellent for GUI developers who are interested in speeding up the design process, but are a poor medium for educators and researchers with no GUI design experience who are looking for a way to incorporate a simple working interface with preexisting computational procedures. The goal of the Graphical User Interface Designer (*GuiDe*) is to provide scientists, engineers, and educators with a quick and effective method to design simple, resource file based GUIs.

CompuGe is a powerful example of how *GuiDe* is a cost-effective method to produce an user-friendly computer-assisted learning package. The most time-

consuming part of the implementation has to do with plotting options included in the control display. Once these options were developed, few hours were only necessary to complete the GUI.

The user-friendly interface allows even novice users to handle the learning environment without difficulties obtaining a fundamental understanding of the principles underlying computational geometry. For educational purposes, CompuGe not only gives students the opportunity to learn rapidly but also enable them to explore the different concepts due to the great flexibility to use the system according their needs.

### 3.2 CompuGe Visualization

Parametric representation of curves and surfaces is plenty of complex concepts which can be difficult to learn in a traditional classroom environment. The following discussion presents a few examples of how CompuGe can be used to improve the learning process.

Students can visualize three different parameterization of curves (Bezier, Bspline and NURBS curves). Different algorithms can be applied to a specific type of parameterization, the corresponding parameters of the algorithms can be modified, and interesting geometrical properties can be addressed (see Figure 2). For example, an arbitrary point  $b^n(t)$  on the Bezier curve, as long as  $t \in [0, 1]$ , is always inside the convex hull of the control points  $p_i$ ,  $i = 0, 1, 2 \dots, n$ . The convex hull of control points, also referred to as control polygon, is a polygon whose vertices are the control points. The convex hull property can be viewed through CompuGe because control points can be handle interactively. When a control point is added, moved or erased, the corresponding Bezier curve and control polygon are automatically plotted.

Plotting of Bezier, Bspline and NURBS curves for the same set of control points is possible as illustrated in Figure 3. In addition, it is possible simultaneously and interactively to visualize changes of the three different curves when a control point is added or when one of the control points in moved or erased. Parametric representation of surfaces is simply the generalization of the parametric representation of curves. Figure 4 shows how CompuGe can be used for modeling surfaces.

An interesting exercise is to provide students with the skeleton of CompuGe without including code for the basic algorithms so that students can insert their own code into CompuGe. It is possible due to the high modularity of the software. A *plotgraph* routine, is a black box including all the plotting options so that students do not have to deal with these options in their implementation of the basic algorithms. An advanced computational geometry course, taught while the author was in Mississippi State University, was implemented using CompuGe as a skeleton so that students implemented each of the algorithms through the coursework.

## 4 Conclusions

A visual environment for learning computational geometry has been presented. CompuGe can be used in lectures and laboratories to visualize the behavior of Bezier, B-spline and NURBS curves in an integrated environment. In addition, the software can be used by students after lectures to construct their own knowledge and generate feedback necessary to complete the learning process. The high modularity in the implementation of CompuGe also allows for programming exercises which contribute to understand the different algorithms. We believe that CompuGe is an important contribution to the computational geometry coursework as it assists well students in learning and understanding basic concepts behind parametric representation of curves and surfaces. We have received very positive feedback from students and faculty.

## References

- [1] de Casteljaou, P. (1959). Outillages méthodes calcul. Technical Report, A. Citroen, Paris.
- [2] de Casteljaou, P. (1963). Courbes et surfaces à poles. Technical Report, A. Citroen, Paris.
- [3] Bezier, P. (1966). Définition numérique des courbes et surfaces I. *Automatisme*, 11: 625-632.
- [4] Bezier, P. (1967). Définition numérique des courbes et surfaces II. *Automatisme*, 12: 17-21.
- [5] Schoenberg, I. (1946). Contributions to the problem of approximation of equidistant data by analytic functions. *Quart. Appl. Math.*, 4: 45-99.
- [6] de Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50-62.
- [7] Newcomb, G. & Gaither, K. (1995). Graphical User Interface Designer: A Guide for Research and Education. ERC-MSU Report.
- [8] Sims, R. (1997). Interactivity: A forgotten art? *Computers in Human Behavior*, 13(2): 157-180.
- [9] Larkin, J. H. & Chabay, R. W. (1992). *Computer Assisted Instruction and Intelligent Tutoring Systems*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- [10] Khan, J. & Yip, Y. J. (1996). Pedagogic principles of case-based CBL. *Journal of Computer Based Learning*, 12: 172-192.

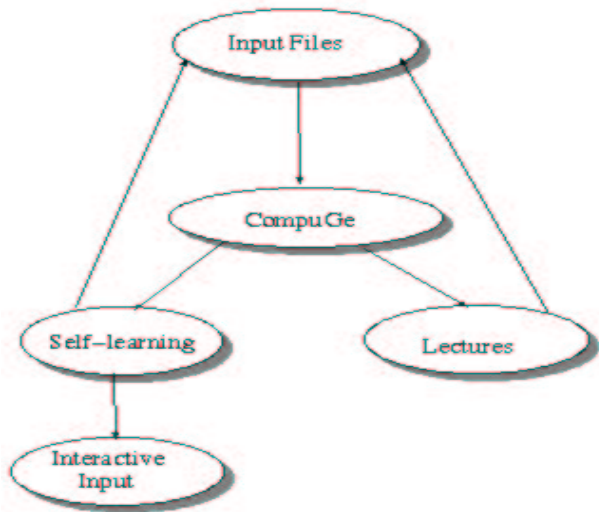


Figure 1. Learning Model

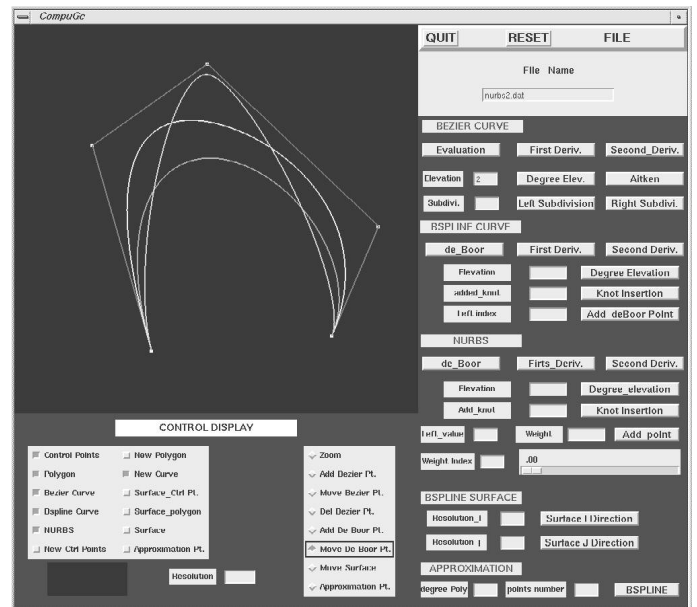


Figure 3. Bezier, B spline, and NURBS curves visualization

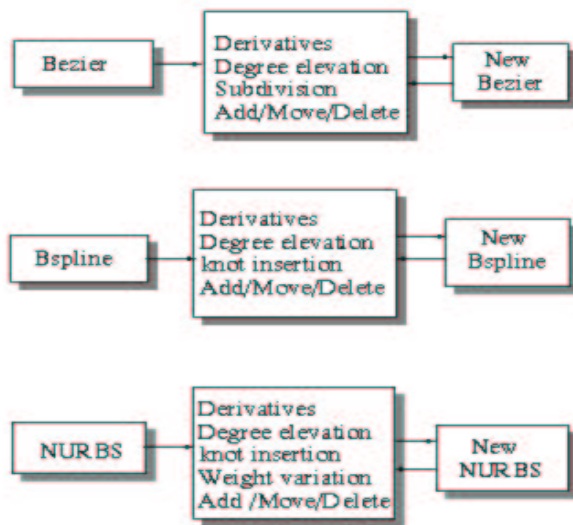


Figure 2. Example of visualization process

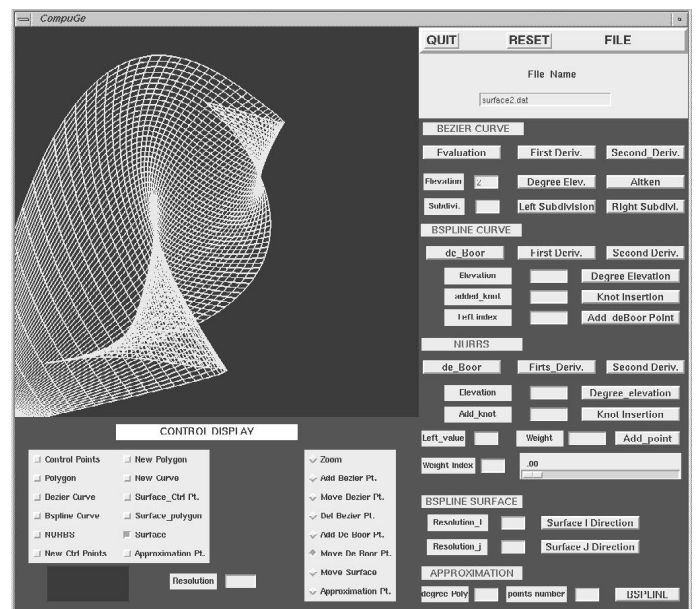


Figure 4. Surface visualization