Computer-Aided Design 45 (2013) 168–179

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

# A tool for analytical simulation of B-splines surface deformation

Manuel González-Hidalgo, Antoni Jaume-i-Capó*, Arnau Mir, Gabriel Nicolau-Bestard

*Computer Graphics, Vision and Artificial Intelligence Group. Maths. and Computer Science Department, University of the Balearic Islands, Spain*

ARTICLE INFO

*Article history:*
Received 7 October 2011
Accepted 5 August 2012

*Keywords:*
Computer graphics
Surface deformation
Finite elements
B-splines

ABSTRACT

A non-planar surface deformation model based on B-splines as finite elements is presented here. The model includes the variational formulation, the system of ordinary differential equations derived from it and its analytical solution. The model has been checked for a variety of surfaces such as tiles, half spheres, planes, etc. Furthermore, we are able to solve the system analytically by only moving a reduced number of control points to deform the surface. This makes the method faster, since numerical methods are no longer necessary.



## 1. Introduction

Deformation models include a large number of applications that have been used in fields such as edge detection and image segmentation [1,2], computer animation and graphics [3,4], geometric modeling [5,6], cloth and material simulation [7–12], medical simulation [13,1], face synthesis [14,15,3,13], and so on. Due to the applications that derive from object deformation, it is one of the most active topics in computer graphics today.

A wide range of applications are covered by deformable models, which were introduced in 1987 by Kass et al. in 2D as explicit deformable contours [16] and generalized to 3D by Terzopoulos et al. [17]. Since then, different approaches have been used in order to represent them and their dynamic simulation. An excellent early survey on modeling of deformable objects in computer graphics can be found in [18] and on Free-Form Deformation (FFD) techniques in [19]. The geometric methods [5,6] are the most commonly used in FFD, a term first used by Sederberg and Parry in [20]. In [21], a review on deformable surface representation and deformable models evolution can be seen. Recent advances in mesh deformation and editing techniques are presented by Botsch and Sorkine in [22].

However, despite the existence of large numbers of techniques to represent deformable objects and solve the evolution equations associated with the simulations, the most used techniques remain finite element methods [23–25,12,22]. Many finite element methods have been used in the modeling of deformable objects, as can be seen in the Botsch survey: from the Lagrange finite elements [26,4], to the finite elements of Bogner–Fox–Schmit [27] or the recently introduced finite element based on B-spline functions [28,29].

Höllig [28] was the first to introduce the use of uniform B-splines and their properties as finite elements in order to solve partial differential equations. B-splines are piecewise polynomial functions with good local approximation properties for smooth functions and with local support [30]. B-splines were introduced across the Bézier polynomial functions; Carl deBoor [31], Kauss Höllig et al. [32] and Risler [33] studied the spline surfaces. Later, Piegl and Tiller wrote *The NURBS Book* [30], where B-splines are presented as the basis of NURBS functions. The definition of Uniform B-splines was presented in different contexts in [31,34,30,28]. Using B-splines, Isogeometric Analysis (IgA) [35] is introduced with the purpose of accelerating numerical analysis and to closely link Computer Aided Design (CAD) and Computer Aided Engineering (CAE) [36]. With this [37] shows that using IgA an object can be obtained regardless of how coarse the discretization and the mesh refinement is. This makes it possible to eliminate the need to communicate with the CAD geometry. The basic idea of IgA is to use CAD basic functions, for example NURBS [38], in a numerical analysis context. Other authors have used splines (B-splines, Uniform B-splines, web-splines, NURBS) in a numerical analysis context (see [28,21]). Höllig used web-spline functions as the basis of a finite element space [28], Awanou et al. in [39] used finite elements based on B-splines to fit scattered data using numerical solutions of partial differential equations and Nguyen-Thanh in [36], shows that NURBS-based isogeometric analysis is inefficient in refinement and introduces the Bézier and B-spline functions in the IgA.

There are many ways to define B-splines and uniform B-splines and obtain their properties. It can be achieved by using divided differences of the truncated power function (see [31]), by blossoming or polar form that can be found in [40], or by using the recurrence relations. The recurrence relations were first used by Gordon and Riesenfeld in 1974 and can be found, for

* Correspondence to: Ed. Anselm Turmeda, 07122 Palma, Spain. Tel.: +34 971 25 9721; fax: +34 971 17 3003.

*E-mail addresses:* manuel.gonzalez@uib.es (M. González-Hidalgo), antoni.jaume@uib.es (A. Jaume-i-Capó), arnau.mir@uib.es (A. Mir), gabriel.nicolau@uib.es (G. Nicolau-Bestard).

0010-4485/$ – see front matter 
doi:10.1016/j.cad.2012.08.003

instance, in [34,30]. For its computational simplicity, this last method is the most used in CAD, CAGD (Computer Aided Geometric Design), CAE or CAM (Computer Aided Manufacturing). In this work we consider only uniform knots and uniform B-splines, then we use the recurrence definition given by Höllig in [28], which is appropriate for multivariate approximation and also simple, from a computational point of view.

The CAD/CAE concept is hardly a new one. It was not started by Höllig [28] but has its origins in the early 1970s. As pointed out, the solution to problems governed by differential equations, such as the development of equations for deformable models, is addressed by finite element methods (FEM). Conventional FEM uses standard shape functions [41–43], that often require high degrees of freedom (DOF) for a specified accuracy, thus causing a considerable increase in complexity and, consequently, a delay in design tasks that require repeated computations [44]. In order to reduce the number of DOFs required, competitive methods have been designed. A relevant review of such methods and their relation with CAD/CAE integration tasks can be found in [44]. The main drawback of FEM techniques is their high computational costs. In this work, we present a mathematical technique that is able to deform non-planar parametric surfaces with a low computational cost; see [29,45,46] for previous work. Although the methodology used here is similar, the deformed surfaces and the numerical techniques that have been implemented are different. Previously [45] we used Mathematica in order to obtain the numerical examples, but in some cases we found that Mathematica was not able to find a solution to our problem and results were unsatisfactory. Also, a detailed study of the numerical cost and a comparison between the numerical and analytical solutions is presented. The numerical techniques used allow for more complex and higher resolution deformed surfaces. The mathematical model is similar to the model used by Cohen in [27] and the generalization developed by Terzopoulos in [47]. In [27] the classical Bogner–Fox–Schmidt finite elements are used in order to solve the proposed model, and in [47] finite differences are used. In our work, we use B-spline finite elements instead. Classical finite elements are commonly used to solve models that involve partial differential equations, but this requires big data structures. On the other hand, the use of B-splines as finite elements reduces the data structure of the model since only a reduced number of control points of the surface are moved in order to compute its deformation. Moreover, our model has the advantage that it can be solved analytically.

This work is organized as follows. In Section 2, we define the uniform B-splines used to introduce finite elements and present their properties. Section 3 is devoted to the model of surface deformation. First, the static model is presented showing the variational formulation and the space discretization. The dynamic evolution model is next in this section and the equation associated to the model is solved, taking into account some considerations on mass and damping matrices. Finally, the computation of the control points of the initial surface is explained. Section 4 is devoted to analyze the computational cost of the described method and is compared to the analytical and numerical solutions. The description of the programming tools developed in order to display the simulations and the representation of B-spline surfaces is addressed in Section 5. Also, several computed numerical deformations are displayed using the evolution model, with different surfaces and forces. The last section is devoted to conclusions and future work.

## 2. B-splines and parametric surfaces with B-splines

B-splines are piecewise polynomial functions. It has been verified, with other function approximation techniques [30] that polynomials provide a good local approximation for smooth functions. However, if large intervals are used, the approximation accuracy can be very low, the accuracy of the approach could be very low and local changes have a global influence. Therefore, it is natural to use piecewise polynomials, defined on a fine partition of the function domain. We have chosen B-splines as a piecewise polynomial approximation because of its local support. This property reduces the computational cost of the model.

Uniform B-splines can be defined in several ways [31,34,30,28]. In this work we have taken the definition given by Höllig in [28], which is described next.

**Definition 1.** A uniform B-spline of degree $n$, $b^n$, is defined by the following recurrence formula:

$$b^n(x) = \int_{x-1}^{x} b^{n-1}(t)dt$$

starting with $b^0(x) = \begin{cases} 1, & x \in [0, 1[, \\ 0, & \text{otherwise.} \end{cases}$

A uniform B-spline of degree $n, b^n$, is positive on $(0, n+1)$ and vanishes outside this interval, is $(n-1)$-times continuously differentiable with discontinuities of the $n$th derivative at the break points $0, \dots, n+1$. More, it has a piecewise polynomial structure, that is, $b^n$ is a polynomial of degree $n$ on each interval $[k, k+1]$, $k = 0, \dots, n$. Finally, two qualitative properties are noted: the B-spline of degree $n$ is symmetric, i.e., $b^n(x) = b^n(n+1-x)$ and strictly monotone on $[0, (n+1)/2]$ and $[(n+1)/2, n+1]$ (see [28] for a detailed account of these properties).

The previous definition is not well adapted for numerical evaluations. In order to evaluate B-splines in a simple and computationally fast manner, the recurrence equation can be used. This equation was given by De Boor [31] and Cox [48], and it is a linear combination of smaller degree B-splines:

$$b^n(x) = \frac{x}{n} b^{n-1}(x) + \frac{(n+1-x)}{n} b^{n-1}(x-1). \quad (1)$$

In order to construct the finite element bases, we will use a scaled and translated uniform B-spline. They are defined by transforming the standard uniform B-spline, $b^n$, to the grid $h\mathbb{Z} = \{\dots, -2h, h, 0, h, 2h, \dots\}$, where $h$ is the scaled step.

**Definition 2.** For $h > 0$ and $k \in \mathbb{Z}$, the scaled and translated B-spline of degree $n$, $b^n_{k,h}$, is defined by $b^n_{k,h}(x) = b^n(\frac{x}{h} - k)$.

As we can see, $b^n_{k,h}$ are the B-splines on the grid $h\mathbb{Z}$. Their linear combinations are called *cardinal splines* of degree less than or equal to $n$ with grid width $h$. The support[1] of this function is $[k, k+n+1]\,h$. Moreover, on each grid interval $Q = [l, l+1]\,h$ exactly $n+1$ B-splines are nonzero.

From Definition 1 we obtain that the first order derivative of a degree $n$ B-spline is given by

$$\frac{d}{dx} b^n(x) = b^{n-1}(x) - b^{n-1}(x) \quad (2)$$

with $b^n(0) = 0$ [28]. If we apply the transformation given in Definition 2, the first order derivative of the transformed B-spline is given by

$$\frac{d}{dx} b^n_{k,h}(x) = h^{-1}(b^{n-1}_{k,h}(x) - b^{n-1}_{k+1,h}(x)). \quad (3)$$

[28]. We also need the derivatives of any order. Higher order derivatives can be computed as a linear combination of lower degree B-splines. The differentiation formula can be expressed in a compact form as follows.

1 The support of a function $f$ is the closure of the set where the function $f$ is not zero.