

AN EVOLUTION MODEL OF B-SPLINES PARAMETRIC SURFACE

Manuel González-hidalgo, Arnau Mir and Gabriel Nicolau-Bestard

Computer Graphics, Vision and Artificial Intelligence Group.

Maths. and Computer Science Department.

University of the Balearic Islands. Spain

email: manuel.gonzalez@uib.es, arnau.mir@uib.es, gabriel.nicolau@gmail.com

KEYWORDS

B-splines, Finite Elements, Variational Formulation, Dynamic Deformation Model.

ABSTRACT

The study of the motion of deformable objects is one of the most important topics in computer animation, computer modelling, and so on. This paper presents an evolution model which deforms a parametric surface with a linear complexity and nearly stability. The deformation model is based on an associated energy to one surface, that it checks the shape of it. The associated variational formulation to the problem of minimizing the energy functional is solved using the finite element method based on B-splines. The spatial discretization where these finite elements are defined and computed shows as a reduced number of control points is deformed instead of all the surface points, obtaining an efficient numerical scheme.

INTRODUCTION

The deformation models include a large number of applications, and they have been used in fields as edge detection, computer animation, geometric modelling, and so on. In this work, a deformation model will be introduced that uses B-splines as finite elements. The model includes deformation equation, numerical method to solve it, examples of deformations, computational cost and study of stability.

The use of B-splines and their properties as finite elements was introduced by Höllig in Höllig (2003). The model of deformation is based on the model used in Cohen (1992) where the finite elements are classical finite elements (triangles and squares). We choose B-splines as finite elements to solve our deformation equation.

The use of classical finite elements makes a mosaic that it fills all space with a large computational cost if we want to solve our problem because a big data structure is needed. Otherwise, the B-splines finite elements combine the computational advantage of B-splines and standard mesh-based elements because the obtained data structure is smaller than the data obtained using classical finite elements.

This work can be considered an extended version

of González-Hidalgo et al. (2006) presented in CompIM-AGE'06 Conference and it's organized as follows: in section 2, we introduce B-splines finite elements and their properties; in section 3, the model of surface deformation, the numerical resolution and the computational cost are developed. In section 4, we show some illustrative examples. Section 5 is devoted to analyze the stability of our numerical method; in sections 6 and 7, the influence of first and second order parameters are studied. Finally, in section 8, some conclusions are exposed. More details about sections 2, 3 and 4 can be found in González-Hidalgo et al. (2006).

B-SPLINES

The B-splines are piecewise polynomial functions with a good local approximation for smooth functions Piegler (1997). We have chosen B-splines as 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 de Boor (1978) Farin (1997) Piegler (1997) Höllig (2003). In this work we have taken the definition given by Höllig in Höllig (2003), which it will be described next.

Definition 0.1 Höllig (2003) *An uniform B-spline of degree n , b^n , is defined by*

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

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

To evaluate the B-splines in a simple form and fast computationally, we use the following recurrence equation (De Boor de Boor (1978) and Cox Cox (1972)).

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

In order to construct the finite elements 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 0.2 The transformation for $h > 0$ and $k \in \mathbb{Z}$ is $b_{k,h}^n(x) = b^n(\frac{x}{h} - k)$. The support of this function is $[k, k + n + 1)h$

The first order derivative of the transformed B-spline is given by

$$\frac{d}{dx}b_{k,h}^n(x) = h^{-1}(b_{k,h}^{n-1}(x) - b_{k+1,h}^{n-1}(x)) \quad (2)$$

Höllig (2003). Higher order derivatives can be computed as a linear combination of lower degree B-splines:

Theorem 0.1 The m^{th} derivative of a degree n transformed B-spline following the definition 0.2 is given by the recurrence relation

$$\frac{d^m}{dx^m}b_{k,h}^n(x) = h^{-m} \sum_{i=0}^m (-1)^i \binom{m}{i} b_{k+i,h}^{n-m}(x) \quad (3)$$

Obviously, this equation has sense if $m \leq n$ since in others cases the derivative is 0.

The generalization of one dimensional B-splines is described in the following construction. The N -variate B-spline of degree $\mathbf{n} = (n_1, \dots, n_N)$, of index $\mathbf{k} = (k_1, \dots, k_N)$ and the space discretization $\mathbf{h} = (h_1, \dots, h_N)$ is defined as

$$B_{\mathbf{k},\mathbf{h}}^{\mathbf{n}}(\mathbf{x}) = \prod_{i=1}^N b_{k_i,h_i}^{n_i}(x_i). \quad (4)$$

The support of this function is $\prod_{i=1}^N [k_i, k_i + n_i + 1)h_i$. Applying basic properties of differential calculus and applying theorem 0.1, a compact expression for any partial derivative of multivariate B-spline can be obtained. Using theorem 0.1, the derivatives can be evaluated and they are obtained with a smaller computational cost, since less recurrences are applied.

A B-spline parametric surface is given by $S : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ where

$$S(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} P_{\mathbf{k}} B_{\mathbf{k},\mathbf{h}}^{\mathbf{n}}(\mathbf{x}) \quad (5)$$

(see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007)) where $P_{\mathbf{k}}$ are the so called the *control points* and they are the elements that determine the B-spline surface.

DEFORMATION MODEL AND NUMERICAL RESOLUTION

The deformation model is based on an associated energy to one surface, that it checks the shape of it.

The energy function is a non convex function with a local minimum. The goal is to achieve this minimum using an evolution model. This minimum depends on the initial surface and the used evolution model.

The associated energy functional, $E : \Phi(S) \rightarrow \mathbb{R}$, $S \mapsto E(S)$, is defined as

$$E(S) = \int_{\Omega} \omega_{10} \left| \frac{\partial S}{\partial u} \right|^2 + \omega_{01} \left| \frac{\partial S}{\partial v} \right|^2 + \omega_{11} \left| \frac{\partial S}{\partial u \partial v} \right|^2 + \omega_{20} \left| \frac{\partial^2 S}{\partial u^2} \right|^2 + \omega_{02} \left| \frac{\partial^2 S}{\partial v^2} \right|^2 + \mathcal{P}(S(u, v)) dudv$$

Terzopoulos (1986), Cohen (1992), Montagnat et al. (2001), where \mathcal{P} is a potential of the forces that works on the surface. Using the equations of Euler-Lagrange, it can be proved Cohen (1992) that an energy local minimum must satisfy:

$$-\omega_{10} \frac{\partial^2 S}{\partial u^2} - \omega_{01} \frac{\partial^2 S}{\partial v^2} + 2\omega_{11} \frac{\partial^4 S}{\partial u^2 \partial v^2} + \omega_{20} \frac{\partial^4 S}{\partial u^4} + \omega_{02} \frac{\partial^4 S}{\partial v^4} = -\nabla \mathcal{P}(S(u, v)) + \text{boundary conditions} \quad (6)$$

The surface domain is $\Omega = [0, 1]^2$ and the boundary conditions are: $S(u, 0) = (u, 0, 0)$, $S(u, 1) = (u, 1, 0)$, $S(0, v) = (0, v, 0)$, $S(1, v) = (1, v, 0)$.

VARIATIONAL FORMULATION

With the purpose of establishing the variational formulation of the boundary value problem done by (6), we recall the definition of a Sobolev Space of order two $H^2(\Omega)$,

$$H^2(\Omega) = \{S \in L^2(\Omega) : \frac{\partial^{(\alpha_1 + \alpha_2)} S}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}} \in L^2(\Omega), 0 \leq \alpha_1 + \alpha_2 \leq 2, \alpha_1, \alpha_2 \in \mathbb{N}\}$$

We will consider the set of functions $(H^2(\Omega))^3$ satisfying the previous boundary conditions. We will denote this set by \mathcal{H} . It can be proved Cohen (1992) Raviart (1992) that solving equation (6) is equivalent to find an element $S \in \mathcal{H}$, such that $a(S, T) = L(T)$ for all $T \in \mathcal{H}$, where $a(\cdot, \cdot)$ is a bilinear form defined as

$$a(S, T) = \int_{\Omega} \left(\omega_{10} \frac{\partial S}{\partial u} \frac{\partial T}{\partial u} + \omega_{01} \frac{\partial S}{\partial v} \frac{\partial T}{\partial v} + 2\omega_{11} \frac{\partial^2 S}{\partial u \partial v} \frac{\partial^2 T}{\partial u \partial v} + \omega_{20} \frac{\partial^2 S}{\partial u^2} \frac{\partial^2 T}{\partial u^2} + \omega_{02} \frac{\partial^2 S}{\partial v^2} \frac{\partial^2 T}{\partial v^2} \right) dudv \quad (7)$$

and $L(\cdot)$ is the following linear form

$$L(T) = - \int_{\Omega} \nabla \mathcal{P}(S) T dudv.$$

DISCRETIZATION

We want to find a function $S \in \mathcal{H}$ such that

$$a(S, T) = L(T), \quad \forall T \in \mathcal{H}. \quad (8)$$

In order to do this, the surface domain will be discretized. The spatial discretization is given by $h_1 \mathbb{Z} \times h_2 \mathbb{Z}$ where $h_1 = \frac{1}{N_1+n_x-1}$ and $h_2 = \frac{1}{N_2+n_y-1}$, where $N_1 \times N_2$ are the control points of the B-spline surface that we will to evolve (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007) for details). This spatial discretization will fix the control points that are not zero. The index $\mathbf{k} = (k_1, k_2)$ belongs to the set $\{-n_x, \dots, N_1 + n_x - 1\} \times \{-n_y, \dots, N_2 + n_y - 1\}$. So, the B-spline surface will come determined by the relevant B-splines, and they are specified by the following equation

$$S(u, v) = \sum_{k_1=-n_x}^{N_1+n_x-1} \sum_{k_2=-n_y}^{N_2+n_y-1} P_{(k_1, k_2)} B_{(k_1, k_2)\mathbf{h}}^{\mathbf{n}}(u, v). \quad (9)$$

The B-spline bases are determined by the following set of finite elements of finite dimension: $V_{\mathbf{h}}^{\mathbf{n}} = \langle \{(B_{\mathbf{k}, \mathbf{h}}^{\mathbf{n}}(u, v), 0, 0) : \mathbf{k} \in \mathcal{I}\} \cup \{(0, B_{\mathbf{k}, \mathbf{h}}^{\mathbf{n}}(u, v), 0) : \mathbf{k} \in \mathcal{I}\} \cup \{(0, 0, B_{\mathbf{k}, \mathbf{h}}^{\mathbf{n}}(u, v)) : \mathbf{k} \in \mathcal{I}\} \rangle$ where $\mathcal{I} = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\}$. Thus, taking into account the boundary conditions, the control points $P_{\mathbf{k}}$ associated to B-splines belonging to the set $V_{\mathbf{h}}^{\mathbf{n}}$ are the unique ones that are computed using the equations (10) and (11) (see below).

Using equations (8) and (9) we can obtain three linear systems, one for each coordinate:

$$AP_i = L_i, \quad i = 1, 2, 3, \quad (10)$$

where A is a square matrix and their elements are:

$$a((B_{k, h}^n, 0, 0), (B_{j, h}^n, 0, 0))_{(k, j) \in \mathcal{I} \times \mathcal{I}},$$

P_i is a vector of component i of each control point and L_i is a vector with components $L_1 = L((B_{k, h}^n, 0, 0))_{k \in \mathcal{I}}$, $L_2 = L((0, B_{k, h}^n, 0))_{k \in \mathcal{I}}$, $L_3 = L((0, 0, B_{k, h}^n))_{k \in \mathcal{I}}$.

The computational cost of A is constant and the computational cost of L_i , $i = 1, 2, 3$ is $O(N_1 \times N_2)$ (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007)).

Dynamic evolution model

The classical dynamical model of evolution has been applied Cohen (1992), Qin (1997), Montagnat et al. (2001), González et al. (2001), Mascaró (2002) Mascaró et al. (2002). In our dynamic model, the surface depends on time. So, we have $S(u, v, t)$. Nevertheless, this dependency only affects to the control points which is an advantage since in each iteration we do not have to calculate all the surface. Therefore, we only must calculate the new control points. Thus, our dynamic model of evolution comes determined by the equations

$$M \frac{d^2 P_i}{dt^2} + C \frac{d P_i}{dt} + AP_i = L_i, \quad i = 1, 2, 3. \quad (11)$$

where M and C are the mass and damping matrices respectively and they are diagonal matrices.

The dynamic system (11) has been discretized in time using the standard finite-difference approximation operators. A scheme based on central finite-differences has been used:

$$M \frac{P_i(t + \Delta t) - 2P_i(t) + P_i(t - \Delta t)}{\Delta t^2} + C \frac{P_i(t + \Delta t) - P_i(t - \Delta t)}{2\Delta t} + AP_i(t) = L_i, \quad i = 1, 2, 3. \quad (12)$$

So, to solve equation of the scheme, we have to solve the following three linear system of equations for $P_i(t + \Delta t)$, associated to an explicit integration procedure for the ordinary differential equation:

$$G \cdot P_i(t + \Delta t) = H_i, \quad i = 1, 2, 3, \quad (13)$$

where $G = 2M + \Delta t C$ and $H_i = 2\Delta t L_i - (2\Delta t^2 A - 4M) P_i(t) - (2M - \Delta t C) P_i(t - \Delta t)$ (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007) for details).

The previous system is easy to solve because matrix G is diagonal. Then, the original nonlinear differential equation (11) has been reduced to a sequence of diagonal linear algebraic systems done by (13). And, taking into account the set of finite elements that determines our B-spline bases, this algebraic system of equations is solved only for a reduced number of control points.

The used numerical scheme in the dynamic model depends on two previous iterations $P^{t-\Delta t}$ and P^t , taking the same P^0 and P^1 , where $P^0 = S(u, v, 0)$.

This dynamic model requires $O(N_1 \times N_2)$ for each time step in order to solve the three linear systems (see González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007) for details)

EXAMPLES

This section shows several examples of deformations obtained applying our dynamical model. All the experiments displayed in this section has been made using $\omega_{10} = \omega_{01} = 0.1$ and $\omega_{11} = \omega_{20} = \omega_{02} = 0.01$ and $\Delta t = 0.1$.

In Figure 1, we show several iterations obtained using the dynamic model with bicubic B-splines, a force $(0, 100, 0)$ and $N_1 \times N_2 = 25$, The force is applied only at the central point of the surface.

Not only we can apply forces in vertical directions. Also, we can apply forces in other directions as we can see in Figure 2, where we have applied over all the surface domain an oblique force in the direction $(1, 4, 1)$, with module 106, taking bicubic B-splines and $N_1 \times N_2 = 49$.

Figure 3 shows several iterations obtained using the vertical force $(0, 200, 0)$ over all the surface, using bicubic

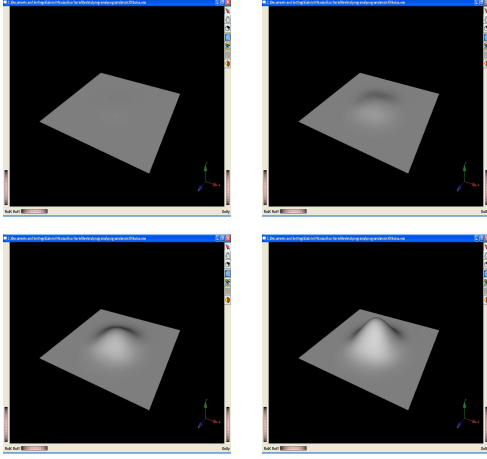


Figure 1: Dynamic simulation of a plane deformation using a positive constant force $(0, 100, 0)$.

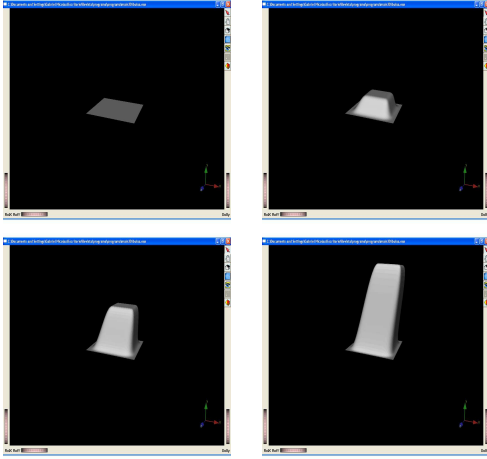


Figure 2: Dynamic simulation of a plane deformation using a force in the direction $(1, 4, 1)$

B-splines and $N_1 \times N_2 = 36$. We can compare this deformation with the obtained one in Figure 1. More examples can be seen in González-Hidalgo et al. (2006) and González-Hidalgo et al. (2007).

STUDY OF STABILITY

In order to study the stability of our numerical method (12), first of all, we need to write it as:

$$\begin{pmatrix} M_1 & M_2 \\ \text{Id} & 0 \end{pmatrix} \begin{pmatrix} P_i(t) \\ P_i(t - \Delta t) \end{pmatrix} + \begin{pmatrix} 2L_i\Delta t^2 \\ 0 \end{pmatrix}, \quad (14)$$

where M_1 and M_2 are $M_1 = (2M + \Delta tC)^{-1}(4M - 2A\Delta t^2)$, $M_2 = (2M + \Delta tC)^{-1}(\Delta tC - 2M)$.

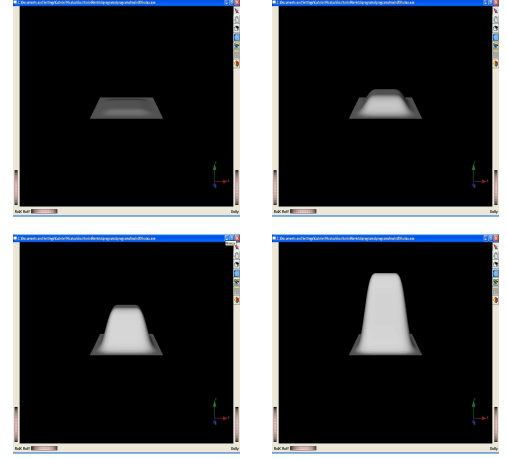


Figure 3: Several iterations of a plane deformation using the vertical force $(0, 200, 0)$ over all the surface.

Following Bathe (1982), the previous system will be stable if the spectral radius of the matrix

$$M = \begin{pmatrix} M_1 & M_2 \\ \text{Id} & 0 \end{pmatrix},$$

is less than 1.

In order to study the stability, the expansion of eigenvalues λ of M as a function of Δt will be found. In this way, the behaviour of eigenvalues λ of M for Δt small will be studied.

Taking into account that $M = m \text{Id}$ and $C = c \text{Id}$ are diagonal matrices, M_2 is a diagonal matrix too. It can be written as: $M_2 = \frac{c\Delta t - 2m}{c\Delta t + 2m} \text{Id}$. Let k be the constant $k = \frac{c\Delta t - 2m}{c\Delta t + 2m}$. So, $M_2 = k \text{Id}$.

Let λ be an eigenvalue of M . It exists a vector x such that $Mx = \lambda x$. The last equation can be written as:

$$\begin{pmatrix} M_1 & k\text{Id} \\ \text{Id} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} M_1x_1 + kx_2 \\ x_1 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

where $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

The previous system has a non-trivial solution if the determinant of the matrix $\lambda M_1 + (k - \lambda^2) \text{Id}$ is zero. That is, $\frac{k - \lambda^2}{\lambda}$ is an eigenvalue of M_1 . (The case $\lambda = 0$ corresponds to take $\Delta t = \frac{2m}{c}$. So, for the experiments values $m = 1$ and $c = 0.3$, $\Delta t = 6.67$ is too large and this case cannot be taken into account.)

So, for every λ eigenvalue of M , there exists μ eigenvalue of M_1 such that

$$\lambda = \frac{\mu \pm \sqrt{\mu^2 + 4k}}{2}. \quad (15)$$

Taking into account that $M_1 = (2M + \Delta tC)^{-1}(4M - 2A\Delta t^2)$, for every eigenvalue μ of M_1 , there exists an eigenvalue α of A such that $\mu = \frac{4m - 2\alpha\Delta t^2}{2m + c\Delta t}$.

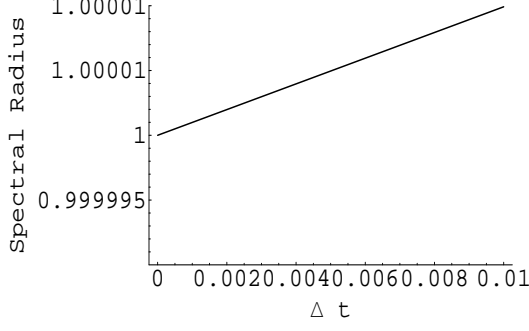


Figure 4: This graph shows the spectral radius of matrix M as a function of Δt

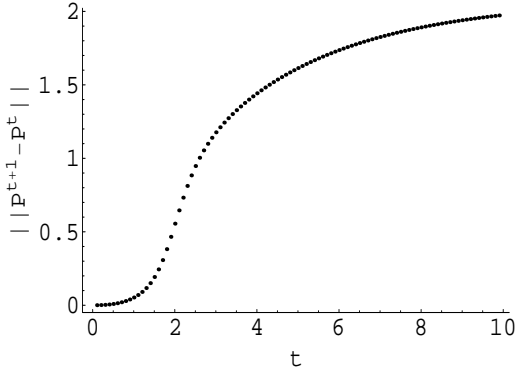


Figure 5: This graph shows the difference norm between two consecutive deformation control points from $t = 0$ to $t = 10$ seconds.

The expansion of eigenvalues μ of M_1 as a function of Δt is: $\mu = 2 - \frac{c}{m}\Delta t + \frac{1}{m^2} \left(\frac{c^2}{2} - \alpha m \right) \Delta t^2 + O(\Delta t^3)$. If we substitute this expression into equation (15), we obtain the expansion of eigenvalues λ of M :

$$\lambda = 1 - \left(c \mp \sqrt{c^2 - 4\alpha m} \right) \frac{\Delta t}{2m} + O(\Delta t^2).$$

So, if $\alpha < 0$ (α eigenvalue of A), our integration method is not stable. The computation of eigenvalues of A shows us that there exists negative eigenvalues of matrix A . Our system is not stable but the spectral radius is close to 1 as it can be seen if figure 4.

Our simulations show good deformations because the instability of the numerical method is close to stability. In González-Hidalgo et al. (2006), it has been shown examples of good simulations using the numerical method. Moreover, the difference norm $\|P_{t+1} - P_t\|$ between two consecutive deformation control points isn't large as it can be seen in figure 5.

THE INFLUENCE OF FIRST ORDER PARAMETERS

In this section, we have studied how the simulations of the model can vary if the value of the parameters ω_{01} and ω_{10} are changed. To do this, we have made several deformations changing the value of the parameter ω_{01} first, and ω_{10} later. The other parameters and the applied force remain constant throughout the simulation. Let P^0 be the vector of control points of the initial surface; that is, the vector of control points of the surface before the deformation. The number of components of P^0 is $3 \times N_1 \times N_2$, three spatial components for each control point. So, if $P_1^0, \dots, P_{N_1 \times N_2}^0$ are the initial control points, P^0 can be written as:

$$P^0 = (P_{1,1}^0, P_{1,2}^0, P_{1,3}^0, \dots, P_{N_1 \times N_2, 1}^0, P_{N_1 \times N_2, 2}^0, P_{N_1 \times N_2, 3}^0)^\top.$$

Using the same notation, let P^f be the vector of control points of the deformed surface. The influence of the parameters has been measured using the following norm:

$$\|P^f - P^0\| = \sqrt{\sum_{i=1}^{N_1 \times N_2} \sum_{j=1}^3 (P_{i,j}^f - P_{i,j}^0)^2}. \quad (16)$$

With the purpose of study the influence of the first order parameter, we have made several simulations with different values of them. The values of the second order parameters have been $\omega_{11} = 0.01, \omega_{20} = 0.01, \omega_{02} = 0.01$. The applied force in the simulations is zero over all surface except in a small central square whose value is 100. Figure 6 displays the linear approximation of the values, where horizontal axis is ω_{10} or ω_{01} and the vertical axis is the quadratic norm (16). The parameters values are 0; 0.0001; 0.001; 0.01; 0.1; 0.3; 0.5; 1; 2; 5; 7; 10; 25; 50; 75; 100; 150 and 200.

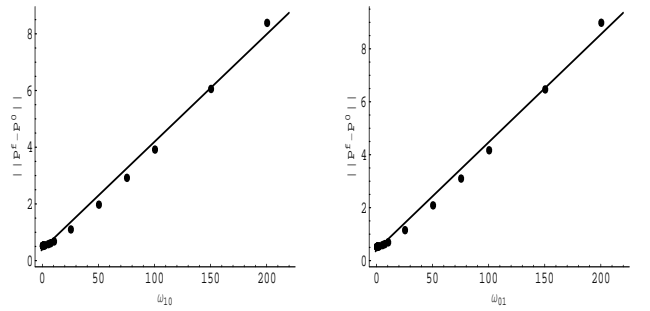


Figure 6: Left: graph of the norms (16) of the simulations vs ω_{10} . Right: graph of the norms (16) vs ω_{01} .

We can see in figure 6, as larger is the first order parameter, smaller is the resistance to the deformation. The relationship between the increase of the first order parameter and the decrease of the resistance to the deformation (16) is linear. The lines of regression are $\xi_1 = 0.4 + 0.038 \omega_{10}$ and $\xi_2 = 0.3924 + 0.0407 \omega_{01}$ where

ξ_1 and ξ_2 is the norm (16) with regression coefficient 0.992 for two parameters.

The effects made by this parameter can be seen in figure 7. This figure shows the effect of these parameters on a deformation made by a vertical force on a central square 0.2×0.2 and module 200. As the parameter increases, the length deformation increases too, following the direction of the applied force. So, we can conclude that the resistance to the length deformation decreases as the value of first order parameters increases.

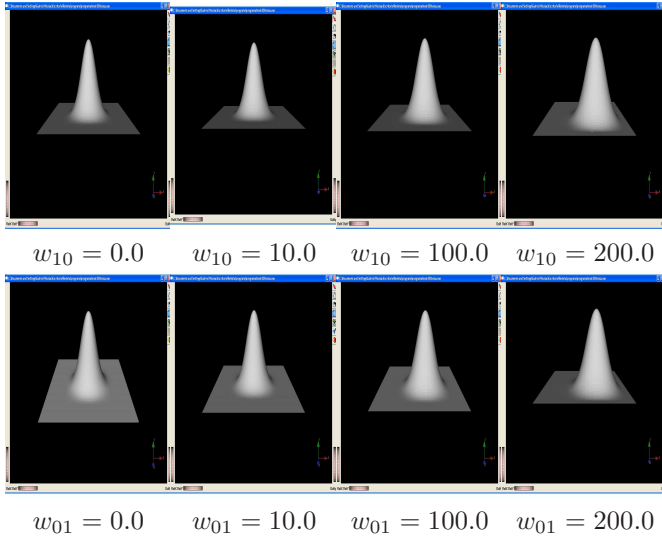


Figure 7: last iteration of deformation with $\Delta t = 0.1$ and $t=1$. Vertical Force $(0,200,0)$

THE INFLUENCE OF SECOND ORDER PARAMETERS

To study the influence of second order parameters ω_{11} , ω_{02} and ω_{20} , a similar development to the first order parameters has been made. We have made several simulations with different values for these parameters. The values of the others parameters remain constant with value 0.1. We have applied a vertical force $(0, 100, 0)$ on a central square 0.2×0.2 . In figure 8 the quadratic norms of the simulations with values $\omega_{ij} = 0; 0.001; 0.01; 0.1; 1; 2; 5; 10; 25; 50; 75; 100; 125; 150; 175; 200$ ($i + j = 2$) are exposed. The regression line is also displayed. Figure 8 shows the following: as larger is the parameter ω_{ij} ($i + j = 2$), smaller is the resistance to the deformation. The relationship between the increase of the resistance is linear. The regression lines are the following $\xi_3 = -0.00077475 + 0.00400753 \omega_{11}$ for the parameter ω_{11} , $\xi_4 = 0.0041346 \omega_{20} - 0.00122073$ for the parameter ω_{20} and $\xi_5 = 0.00410251 \omega_{20} - 0.000905363$ for the parameter ω_{02} where ξ_3, ξ_4 and ξ_5 are the norm. The regression coefficients are larger than 0.99 for the three second order parameters.

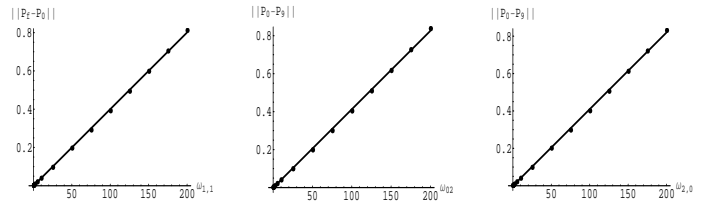


Figure 8: The quadratic norms and the regression line of the simulations vs the different second order parameters.

In figure 9 the last iteration for the parameter values 0, 25, 100 and 200 are displayed. The deformation is moved on plane XZ while the force is applied in direction Y .

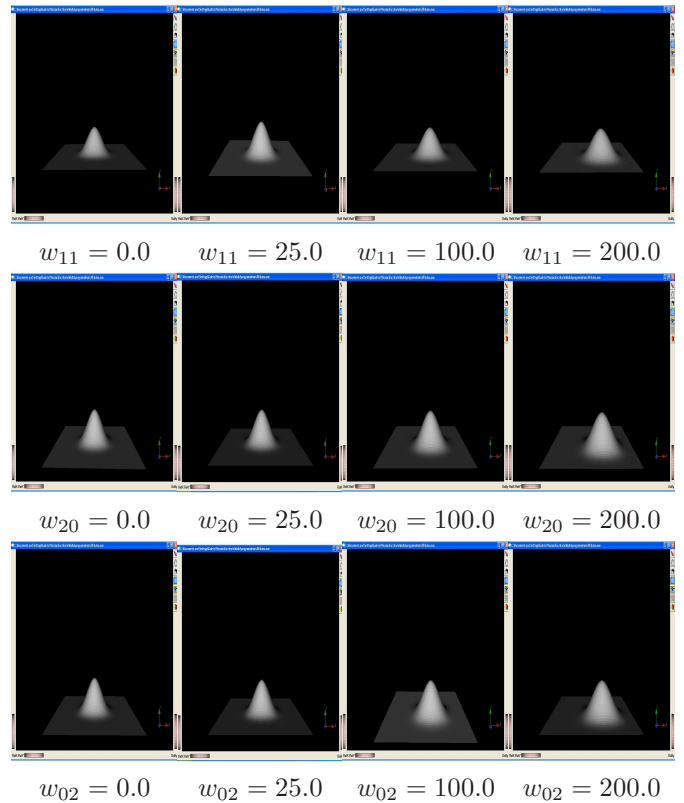


Figure 9: last iteration of deformation with $\Delta t = 0.1$ and $t=1$. vertical force $(0, 100, 0)$

CONCLUSIONS AND FUTURE WORK

We have developed a variational formulation of a model that allows us the deformation of a surface. Moreover, we have shown how to solve the variational equations numerically in a efficient way using B-splines as finite elements. More explicitly, since only a certain number of control points is evolved, the numerical scheme ob-

tained is very efficient.

Also, we have obtained a low computational cost, and we have shown that the model is nearly stable as it happens on the Terzopoulos model Palmer et al. (2000)

The study of behavior of first and second order parameters shows that the growth of these makes a linear decrease resistance of the surface, but in the case of second order parameters the decrease is slower than the first order parameters. In order to obtain a smooth deformation, the external force module must be small or the first order parameters must be smaller than 1. The implementation of this model has been made using C++ and Coin3D, a 3D modelling toolkit which simplifies visualization and scene composition tasks.

As future work, other numerical methods, as conjugate gradient or semi-implicit ones, will be applied. Moreover, other functionals will be taken into account in order to improve the deformation of the surface.

ACKNOWLEDGEMENTS

This work is supported by the project TIC-2004-07926-E, INEVAI3D, of the Spanish Government. The authors would like to thank to the department of Mathematics and Computer Science of University of the Balearic Islands.

REFERENCES

- Bathe K.J., 1982. *Finite element procedures in engineering analysis*. Prentice-Hall Engineering and Engineering Mechanics Series. Prentice-Hall, Englewood Cliffs, New Jersey.
- Cohen I., 1992. *Modèles Déformables 2-D et 3-D: Application à la Segmentation d'Images Médicales*. Ph.D. thesis, Université Paris IX, Dauphine.
- Cox M.G., 1972. *The numerical evaluation of B-splines*. *IMA Journal of Applied Mathematics*, 10, no. 2, 134–149.
- de Boor C., 1978. *A practical guide to splines*. Springer Verlag, New York.
- Farin G., 1997. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Academic Press.
- González M.; Mascaró M.; Mir A.; Palmer P.; and Perales F., 2001. *Modeling and animating deformable objects*. In *Proceedings of IX Spanish Symposium on Pattern Recognition and Image Analysis*. AERFAI Society, Benicasim, Castellón, Spain, 279–290.
- González-Hidalgo M.; Mir A.; and Nicolau G., 2006. *An evolution model of parametric surface deformation using finite elements based on B-splines*. In *Proceedings of CompImage'2006 Conference, Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*. Coimbra, Portugal.
- González-Hidalgo M.; Mir A.; and Nicolau G., 2007. *Dynamic parametric surface deformation using finite elements based on B-splines*. To appear in *International Journal of Computer vision and Biometrics*, 1, no. 1.
- Höllig K., 2003. *Finite element methods with B-Splines*. Frontiers in Applied Mathematics. SIAM, Philadelphia.
- Mascaró M., 2002. *Modelo de Simulación de Deformaciones de Objetos Basado en la Teoría de la Elasticidad*. Ph.D. thesis, Universitat de les Illes Balears.
- Mascaro M.; Mir A.; and Perales F., 2002. *P³DMA : A physical 3D deformable modelling and animations system*. *LNCS*, 2492, 68–79.
- Montagnat J.; Delingette H.; and Ayache N., 2001. *A review of deformable surfaces: topology, geometry and deformation*. *Image and Vision Computing*, 19, no. 14, 1023–1040. URL citeseer.nj.nec.com/498203.html.
- Palmer P.; González M.; and Mir, 2000. *Stability and complexity study of animated elastically deformable objects*. In *Proceedings of I Articulated Motion and Deformable Objects*. Palma de Mallorca Spain, 58–71.
- Piegl L. & Tiller W., 1997. *The NURBS book*. Springer Verlag, Berlin.
- Qin H & Terzopoulos D., 1997. *Triangular nurbs and their dynamic generalizations*. *Computer Aided Geometric Design*, 14, 325–347.
- Raviart P. A. & Thomas J.M., 1992. *Introduction à l'analyse numérique des équations aux dérivées partielles*. Masson, Paris.
- Terzopoulos D., 1986. *Regularization of inverse visual problems involving discontinuities*. *IEEE PAMI*, 8, no. 4, 413–424.