

# Fast Generation of 3-D Deformable Moving Surfaces

Lihua You and Jian J. Zhang

**Abstract**—Dynamic surface modeling is an important subject of geometric modeling due to their extensive applications in engineering design, entertainment and medical visualization. Many deformable objects in the real world are dynamic objects as their shapes change over time. Traditional geometric modeling methods are mainly concerned with static problems, therefore unsuitable for the representation of dynamic objects. Apart from the definition of a dynamic modeling problem, another key issue is how to solve the problem. Because of the complexity of the representations, currently the finite element method or finite difference method is usually used. Their major shortcoming is the excessive computational cost, hence not ideal for applications requiring real-time performance. In this paper, we propose a representation of dynamic surface modeling with a set of fourth order dynamic partial differential equations (PDEs). To solve these dynamic PDEs accurately and efficiently, we also develop an effective resolution method. This method is further extended to achieve local deformation and produce  $n$ -sided patches. It is demonstrated that this new method is almost as fast and accurate as the analytical closed form resolution method and much more efficient and accurate than the numerical methods.

**Index Terms**—Dynamic partial differential equations, dynamic surface modeling, 3-D deformable moving surfaces.

## I. INTRODUCTION

THREE-DIMENSIONAL (3-D) objects are usually represented with surfaces. The popular modeling methods for parametric surfaces have Bézier, B-spline and nonuniform rational B-splines (NURBS) [1]. These methods are effective for the modeling of static surfaces. In the real world, however, there are many deformable moving objects such as human and animal characters in motion, draping cloth and deforming metals. Currently, these objects are modeled mainly with the static methods combined with computer animation techniques. Since the inherent nature describing the deformation and motion of objects is not incorporated into the modeling methods, the realism of the modeled objects depends primarily on the perception of the designer. In contrast, dynamic modeling, taking into account physics laws, has a great potential in generating more realistic and visually appealing artifacts. Due to the difficulty of dynamic modeling, only a small number of references have discussed this issue. In [2], Terzopoulos *et al.* discussed the shape and motion of deformable curves, surfaces and solids using the theory of elasticity. This model was further extended in their later works [3] and [4] to include viscoelasticity, plasticity and

fracture. In order to achieve local deformation, Terzopoulos and Metaxas developed a 3-D dynamic model whose global deformation captures the gross shape features and the local deformation parameters reconstruct the details of the complex shapes that the global abstraction misses [5]. Qin and Terzopoulos represented a dynamic NURBS swung surface [6] by introducing the time parameter into its original static formulation. By minimizing an energy functional subject to user controlled geometric constraints and loads, Celniker and Gossard proposed a curve and surface finite element method for free-form surface generation [7]. Following the theory of pure elasticity, Gündükbay and Özgüç implemented a primal formulation and a hybrid formulation into a physically based modeling method for animating deformable objects [8]. Physics-based modeling has also been combined with geometric subdivision methodology to develop an integrated technique by Mandal *et al.* [9]. However, these physically based geometric modeling approaches rely on numerical methods, mainly the finite element method and finite difference method. Therefore, they are computationally expensive and unsuitable for real-time applications.

Partial differential equation (PDE) based surface modeling is another physics-based geometric modeling approach. The static PDE modeling was firstly proposed by Bloor and Wilson [10]. In their work, they mainly used a biharmonic partial differential equation with a shape parameter. Since the shape parameter has a strong effect on surface shapes, we proposed a more general fourth order PDE. It has three vector-valued shape parameters and provides more user handles for surface shape manipulation [11]. In addition, we have also discussed the capacity and efficiency of surface generation using the solution to PDEs with different orders [12]. Recently, the PDE method was also used in the parameterization for the reconstruction of 3-D free-form objects [13] and the generation of complex 3-D free form surfaces such as human hearts [14] and three-armed starfish vesicles [15]. The most important and challenging issue of the PDE approach in static geometric modeling is the resolution of the partial differential equations. Since the closed form resolution method can only deal with a limited number of applications with simple boundary conditions, numerical methods are again the dominating means. The finite difference method was proposed by Cheng *et al.* [16], and Du and Qin [17]. The finite element method was given by Brown *et al.* [18], and Li and Chang [19]–[21]. The method of collocation points was discussed by Bloor and Wilson [22]. These methods are effective. However, they are computationally expensive, and not ideal for interactive computer graphics applications. To achieve a higher computational efficiency, Bloor and Wilson developed a Fourier series solution [23]. For the cases where the boundary conditions consist of simple periodic functions, this solution is efficient. But if the boundary conditions do not consist of trigonometric functions, the boundary conditions cannot be satisfied exactly which

Manuscript received November 22, 2001; revised January 17, 2003. This work was supported by the Arts and Humanities Research Board (AHRB) under the Grant B/RG/AN5263/ABPN12727. This paper was recommended by Guest Editor H. Zha.

The authors are with the National Centre for Computer Animation, Bournemouth University, Talbot Campus, Poole BH12 5BB, U.K. (e-mail: lyou@bournemouth.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2003.814283

creates a significant setback for certain computer graphics applications, such as surface blending, and consequently the solution is inaccurate. In order to meet the boundary conditions exactly, Bloor and Wilson introduced a remainder function to modify the Fourier series solution [24]. This method was later employed to develop the interactive surface design techniques [25] and [26]. After such a modification, the boundary conditions can be satisfied. But the PDE is unfortunately violated, which inevitably brings errors. Therefore, the fast and accurate resolution of PDE still remains an unsolved problem. In addition, the existing PDE based geometric modeling methods only suit for static modeling problems. In the reported literature we have not yet seen any attempts in the modeling of 3-D deformable moving surfaces with dynamic partial differential equations.

In this paper, we will propose a set of fourth order dynamic partial differential equations. The effects of density and damping on deformable moving surfaces are considered. In order to solve the dynamic PDEs, we will develop a fast and accurate resolution method. The solution consists of trial functions and basic boundary functions. The unknown constants in the solution are determined by exactly satisfying the boundary conditions and minimizing the residual functions of the proposed PDEs. Then a comparison is made between the developed method, the closed form resolution method, and other existing analytical methods. Finally, we will discuss surface manipulation techniques, local deformations and the generation of  $n$ -sided patches, and demonstrate the applications of the proposed method with a number of examples.

## II. DYNAMIC PDE AND SOLUTION

A static 3-D surface can be regarded as the solution to a partial differential equation subject to boundary conditions. This idea is now extended to dynamic geometric modeling. For dynamic modeling, 3-D deformable moving surfaces are time dependent. In addition, density and damping of the surfaces have an influence on the surfaces [27] and [28]. Taking these factors into account, we propose the following dynamic fourth-order partial differential equations

$$\left\{ b_i(u, v, t) \frac{\partial^4}{\partial u^4} + c_i(u, v, t) \frac{\partial^4}{\partial u^2 \partial v^2} + d_i(u, v, t) \times \frac{\partial^4}{\partial v^4} + \rho \frac{\partial^2}{\partial t^2} + \eta \frac{\partial}{\partial t} \right\} x_i(u, v, t) = 0 \quad (i = 1, 2, 3) \quad (1)$$

where  $b_i(u, v, t), c_i(u, v, t), d_i(u, v, t), (i = 1, 2, 3)$  are the shape functions,  $\rho$  and  $\eta$  are the density and the damping coefficient of the surface, respectively,  $x_i(u, v, t) (i = 1, 2, 3)$  are the three positional functions representing the three position components of the surface,  $u, v$  are the parametric variables, and  $t$  is the time variable.

The boundary conditions for a dynamic geometric modeling problem can be generalized in the following form which takes into account the effect of the boundary tangent on the surface shape

$$\begin{aligned} u = 0 \quad x_i &= G_{i1}(v, t) \quad \frac{\partial x_i}{\partial u} = G_{i2}(v, t) \\ u = 1 \quad x_i &= G_{i3}(v, t) \quad \frac{\partial x_i}{\partial u} = G_{i4}(v, t) \quad (i = 1, 2, 3). \end{aligned} \quad (2)$$

The boundary functions of (2) can always be written as a sum of some nonpolynomial basic functions of parametric variable  $v$  and the time variable  $t$ . With this treatment, the boundary conditions are transformed into

$$\begin{aligned} u = 0 \quad x_i &= \sum_{j=1}^J a_{ij1} g_{ij}(v, t) \quad \frac{\partial x_i}{\partial u} = \sum_{j=1}^J a_{ij2} g_{ij}(v, t) \\ u = 1 \quad x_i &= \sum_{j=1}^J a_{ij3} g_{ij}(v, t) \quad \frac{\partial x_i}{\partial u} = \sum_{j=1}^J a_{ij4} g_{ij}(v, t) \\ & \quad (i = 1, 2, 3). \end{aligned} \quad (3)$$

In order to solve dynamic PDEs (1), we construct the solution functions which consist of the basic functions in (3) and trial functions of variable  $u$  as follows.

$$x_i(u, v, t) = \sum_{j=1}^J \sum_{n=0}^N p_{ijn} u^n g_{ij}(v, t) \quad (i = 1, 2, 3). \quad (4)$$

Substituting (4) into (3), and zeroing the sum of the coefficients of each basic function  $g_{ij}(v, t)$ , we obtain four linear algebraic equations for each basic function  $g_{ij}(v, t)$ . Solving these four linear algebraic equations for all basic functions, the unknown constants  $p_{ijn} (i = 1, 2, 3; j = 1, 2, \dots, J; n = 0, 1, 2, 3)$  are determined. Then substituting the expressions of these constants back to (4), the position functions are changed into

$$\begin{aligned} x_i(u, v, t) &= \sum_{j=1}^J \left\{ a_{ij1}(1 - 3u^2 + 2u^3) + a_{ij2}(u - 2u^2 + u^3) \right. \\ & \quad + a_{ij3}(3u^2 - 2u^3) + a_{ij4}(-u^2 + u^3) \\ & \quad \left. + \sum_{n=4}^N [(n-3)u^2 - (n-2)u^3 + u^n] p_{ijn} \right\} g_{ij}(v, t) \\ & \quad (i = 1, 2, 3). \end{aligned} \quad (5)$$

In order to determine the rest unknown constants in the above equation, we substitute (5) into dynamic PDE (1) where  $b_i = b_i(u, v, t), c_i = c_i(u, v, t)$ , and  $d_i = d_i(u, v, t)$ . We now obtain the following residual functions:

$$R_i(u, v, t) = \sum_{j=1}^J \left[ \sum_{n=4}^N B_{ijn}(u, v, t) p_{ijn} + C_{ij}(u, v, t) \right] \quad (i = 1, 2, 3) \quad (6)$$

where

$$\begin{aligned} B_{ijn}(u, v, t) &= n(n-1)(n-2)(n-3)u^{n-4} b_i g_{ij}(v, t) \\ & \quad + [2(n-3) - 6(n-2)u + n(n-1)u^{n-2}] \\ & \quad \times c_i \frac{\partial^2 g_{ij}(v, t)}{\partial v^2} + [(n-3)u^2 - (n-2)u^3 + u^n] \\ & \quad \times \left[ d_i \frac{\partial^4}{\partial v^4} + \rho \frac{\partial^2}{\partial t^2} + \eta \frac{\partial}{\partial t} \right] g_{ij}(v, t) \end{aligned} \quad (7)$$

and

$$\begin{aligned}
c_{ij}(u, v, t) &= 2[-3a_{ij1}(1-2u) - a_{ij2}(2-3u) + 3a_{ij3}(1-2u) \\
&\quad - a_{ij4}(1-3u)]c_i \frac{\partial^2 g_{ij}(v, t)}{\partial v^2} \\
&\quad + [a_{ij1}(1-3u^2 + 2u^3) + a_{ij2}(u-2u^2 + u^3) \\
&\quad \times [+a_{ij3}(3u^2 - 2u^3) + a_{ij4}(-u^2 + u^3) \\
&\quad \times \left[ d_i \frac{\partial^4}{\partial v^4} + \rho \frac{\partial^2}{\partial t^2} + \eta \frac{\partial}{\partial t} \right] g_{ij}(v, t)]. \quad (8)
\end{aligned}$$

At an instant of motion  $t = t_0$ , within the resolution region  $\{0 \leq u \leq 1; v_0 \leq v \leq v_1\}$ , by uniformly choosing  $M$  collocation points and substituting the values of parametric variables  $u$  and  $v$  at these collocation points into (6), (7), and (8), we obtain the residual values from residual functions (6), which consist of  $M$  linear algebraic equations for every position function  $x_i$  as follows [29]

$$\begin{aligned}
R_i(u_m, v_m, t_0) &= \sum_{j=1}^J \left[ \sum_{n=4}^N B_{ijn}(u_m, v_m, t_0) p_{ijn} \right. \\
&\quad \left. + C_{ij}(u_m, v_m, t_0) \right] \quad (i = 1, 2, 3; m = 1, 2, \dots, M). \quad (9)
\end{aligned}$$

If the size of the equations is equal to the number of the unknown constants of each position function, i.e.,  $M = J \times (N - 3)$ , (9) can be solved directly to obtain all unknown constants of (6). Alternatively, we can choose more collocation points than the number of the unknown constants and find the square sum of all the residual values. In doing this, we rewrite (9) in the following matrix form

$$\mathbf{R}_i = \mathbf{B}_i \mathbf{P}_i - \mathbf{C}_i \quad (i = 1, 2, 3) \quad (10)$$

where  $\mathbf{R}_i$ ,  $\mathbf{P}_i$ , and  $\mathbf{C}_i$  are column vectors of  $M$  elements consisting of residual values  $R_i(u_m, v_m, t_0)$  ( $m = 1, 2, \dots, M$ ), unknown constants  $p_{ijn}$  ( $j = 1, 2, \dots, J; n = 4, 5, \dots, N$ ), and known constants  $\sum_{j=1}^J C_{ij}(u_m, v_m, t_0)$ , respectively, and  $\mathbf{B}_i$  is a  $M \times [J \times (N - 3)]$  matrix consisting of the coefficients of unknown constants  $p_{ijn}$  ( $j = 1, 2, \dots, J; n = 4, 5, \dots, N$ ).

Thus the square sum of all the residual values for each position function is given by

$$\mathbf{I}_i = \mathbf{R}_i^T \mathbf{R}_i \quad (i = 1, 2, 3) \quad (11)$$

By minimizing the square sum, we obtain the following  $J \times (N - 3)$  linear algebraic equations

$$\mathbf{B}_i^T \mathbf{B}_i \mathbf{P}_i = \mathbf{B}_i^T \mathbf{C}_i \quad (i = 1, 2, 3). \quad (12)$$

The resolution of (12) will determine the rest of the unknown constants. Thus various 3-D deformable moving surfaces can be created with (5).

The developed method is easily understood and programmed. The resolution of the linear algebraic equations (12), which can be achieved with many standard linear equation solving

methods, will result in all the unknown constants to be obtained. Since only a small number of unknown constants are involved in the equations, the developed method is computationally very efficient.

### III. COMPARISON BETWEEN DIFFERENT MODELING METHODS

Since the developed method exactly satisfies all the boundary conditions and minimizes the errors of the surface functions, the developed method is expected to generate 3-D deformable moving surfaces quickly and accurately. In order to demonstrate the advantages of the developed method over existing geometric modeling methods, in this section, we compare the developed method with the closed form resolution method and some existing analytical methods. Since no PDE based method is available for the problem of dynamic surface modeling, we will make a comparison only over a static problem.

In order to obtain a closed form solution, we choose the following boundary conditions with which a free form surface can be generated using the solution to PDE (1)

$$\begin{aligned}
u = 0 \quad x_1 &= a + b \sin v + ce^{0.1v} \\
\frac{\partial x_1}{\partial u} &= a' + b' \sin v + c'e^{0.1v} \\
x_2 &= d + ev + f \cos v \quad \frac{\partial x_2}{\partial u} = d' + e'v + f' \cos v \\
x_3 &= h_0 - h_1 v \quad \frac{\partial x_3}{\partial u} = h'_0 - h'_1 v \\
u = 1 \quad x_1 &= r \sin v \quad \frac{\partial x_1}{\partial u} = r' \sin v \\
x_2 &= -r \cos v \quad \frac{\partial x_2}{\partial u} = -r' \cos v \\
x_3 &= 0 \quad \frac{\partial x_3}{\partial u} = 0. \quad (13)
\end{aligned}$$

According to the above boundary conditions, the closed form solution of PDEs (1) can be expressed as the following:

$$\begin{aligned}
x_1 &= f_{11}(u) + f_{12}(u)e^{0.1v} + f_{13}(u) \sin v \\
x_2 &= f_{21}(u) + f_{22}(u)v + f_{23}(u) \cos v \\
x_3 &= f_{31}(u) + f_{32}(u)v. \quad (14)
\end{aligned}$$

Substituting the above equations into PDEs (1) and taking  $b_i = 1, c_i = 2a_i^2, d_i = a_i^4$  ( $i = 1, 2, 3$ ), we obtain the unknown functions  $f_{ij}(u)$ , which are given by

$$\begin{aligned}
f_{ij}(u) &= c_{ij0} + c_{ij1}u + c_{ij2}u^2 + c_{ij3}u^3 \\
&\text{for } f_{i1}(u) \quad (i = 1, 2, 3) \quad \text{and} \quad f_{i2}(u) \quad (i = 2, 3) \quad (15)
\end{aligned}$$

$$\begin{aligned}
f_{12}(u) &= (c_{120} \cos 0.1a_i u + c_{121} \sin 0.1a_i u) \\
&\quad + u(c_{122} \cos 0.1a_i u + c_{123} \sin 0.1a_i u) \quad (16)
\end{aligned}$$

and

$$\begin{aligned}
f_{i3}(u) &= (c_{i30} + c_{i31}u)e^{a_i u} + (c_{i32} + c_{i33}u)e^{-a_i u} \\
&\text{for } f_{i3}(u) \quad (i = 1, 2). \quad (17)
\end{aligned}$$

The unknown constants in the above equations can be determined by substituting them into boundary conditions (13).

Taking  $a_i = 3$  ( $i = 1, 2, 3$ ) and  $a = 4.763, b = 0.017, c = -3.395, d = d' = -f = -f' = 0.475, e = e' = 0.0159, r = r' = 1, h_0 = -h'_0 = 3, h_1 = -h'_1 = 0.1, a' = b' = c' = 0$ , the surface which is defined with the closed form solution (14)

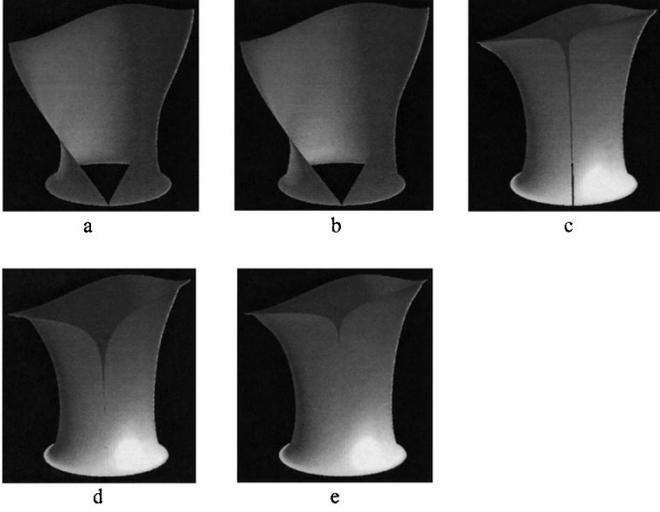


Fig. 1. Surface generated with different methods.

of PDEs (1) subject to (13), is depicted in Fig. 1(a). It is the accurate surface of this problem.

Considering the basic functions in boundary conditions (13), the solution of PDEs (1) subject to boundary conditions (13) can be written as follows according to the solution function (4)

$$\begin{aligned}
 x_1 &= \sum_{n=0}^N p_{11n} u^n + \sum_{n=0}^N p_{12n} u^n e^{0.1v} + \sum_{n=0}^N p_{13n} u^n \sin v \\
 x_2 &= \sum_{n=0}^N p_{21n} u^n + \sum_{n=0}^N p_{22n} u^n v + \sum_{n=0}^N p_{23n} u^n \cos v \\
 x_3 &= \sum_{n=0}^N p_{31n} u^n + \sum_{n=0}^N p_{32n} u^n v.
 \end{aligned} \quad (18)$$

With the above-developed method, we can determine all the unknown constants. Using the same shape parameters and geometric parameters, taking the total collocation points to be  $M = 36$  and the terms  $N = 7$ , the obtained surface for this problem is given in Fig. 1(b). There is no visible difference between Fig. 1(a) and (b). It indicates that the accuracy of the developed method is very close to that of the closed form solution method.

For the Fourier series based method [23], the basic functions  $v$  and  $e^{0.1v}$  in boundary conditions (13) must be first expanded into the following Fourier series, respectively

$$\begin{aligned}
 v &= \pi - \sum_{n=1}^N \frac{2}{n} \sin nv \\
 e^{0.1v} &= \frac{e^{0.2\pi} - 1}{\pi} \\
 &\quad \times \left[ 5 + \sum_{n=1}^N \frac{1}{0.1^2 + n^2} (0.1 \cos nv - n \sin nv) \right].
 \end{aligned} \quad (19)$$

Then, the solution of PDEs (1) under boundary conditions (13) can be approximated with the following Fourier series

$$x_i = A_{i0}(u) + \sum_{n=1}^N [A_{in}(u) \cos nv + B_{in}(u) \sin nv] \quad (i = 1, 2, 3) \quad (21)$$

where the unknown functions  $A_{i0}(u)$ ,  $A_{in}(u)$ , and  $B_{in}(u)$  can be found in [23] and the unknown constants in these unknown functions are determined by solving a series of linear algebraic equations of order four which are obtained from boundary conditions (13) with the consideration of (19) and (20).

With the same shape parameters and geometric parameters and taking  $N = 20$ , we obtain the surface generated from approximate solution (21) as shown in Fig. 1(c). Since the Fourier series cannot accurately fit the function at the two ends of the upper boundary curve [30], a small gap goes through the bottom boundary curve. Clearly, big discrepancy exists between the Fourier series solution and the above two solutions because boundary conditions (13) are not satisfied exactly by the Fourier series solution. We have taken many terms ( $N = 50$ ) of the Fourier series. However, the surface cannot be further improved.

In order to remedy this weakness, Bloor and Wilson suggested a remainder function  $R_i(u, v)$  ( $i = 1, 2, 3$ ), which has the form of [24]

$$\begin{aligned}
 R_i &= R_{i1}(v) e^{\omega u} + R_{i2}(v) u e^{\omega u} \\
 &\quad + R_{i3}(v) e^{-\omega u} + R_{i4}(v) u e^{-\omega u} \quad (i = 1, 2, 3).
 \end{aligned} \quad (22)$$

Adding the remainder function (22) to (21), they obtained a modified approximation solution below, which was called the pseudospectral solution

$$\bar{x}_i = x_i + R_i \quad (i = 1, 2, 3). \quad (23)$$

Unknown functions  $R_{ij}(v)$  ( $i = 1, 2, 3; j = 1, 2, 3, 4$ ) are determined by making solution (23) satisfy boundary conditions (13) exactly.

With the pseudospectral solution (23) and taking  $N = 5$  in (21), the surface produced is shown in Fig. 1(d). Although the surface is slightly improved by the pseudospectral method, it is still different from the accurate surface. When the term number of the Fourier series was increased to  $N = 10$ , the surface obtained in Fig. 1(e) actually becomes poorer. It indicates that with the pseudospectral method, the accuracy of the solution cannot be improved by increasing the number of Fourier series terms. This is because after the introduction of the remainder function, although the boundary conditions are satisfied exactly, PDEs (1) itself will not be satisfied any more.

This example shows that among all the methods, the method proposed in this paper is closest to the closed form resolution method. The other two are not accurate enough.

Since our developed method obtains the unknown constants by solving a small set of linear equations, it is computationally very efficient. We have timed the process to determine all unknown constants of the above four solutions. Using both the closed form resolution method and the method developed in this paper, it took less than  $10^{-6}$  of a second on an 800 MHz PC. For the Fourier series solution, when taking Fourier series terms to be  $N = 20$ , the time determining all unknown constants is also less than  $10^{-6}$  of a second. However, it is 20 times slower than the developed method in the generation of the surface. For the pseudospectral method, when the number of Fourier series terms was taken to be  $N = 5$ , it took 0.047 seconds to determine all the unknown constants and the values of the unknown functions  $R_{ij}(v)$  at 100 points of the boundary curves which are

used to generate the surface. If the unknown functions  $R_{ij}(v)$  are not determined numerically, tedious manual operations have to be performed to obtain their analytical expressions. Therefore, the method developed in this paper is more efficient than the Fourier series based method and the pseudospectral method. In fact, it can generate surfaces almost as fast as the closed form resolution method in addition to its ability to deal with complex dynamic surface modeling which cannot be solved by the closed form solution method. Since the numerical methods are impossible to generate surfaces as quickly and accurately as the closed form resolution method, the developed method has much better accuracy and higher computational efficiency than the reported numerical methods.

#### IV. DYNAMIC GENERATION OF 3-D DEFORMABLE MOVING SURFACES

The developed method provides us with a fast and accurate generation method of 3-D deformable moving surfaces. In this section, we will use two examples to demonstrate this point.

In the first example, we will indicate how a glass-like object is generated by dynamically deforming its upper circular opening into a pedal-like object. The boundary conditions for the upper opening are given by

$$\begin{aligned}
 u = 0 \quad x_1 &= a \cos v + bt \sin 5v \\
 \frac{\partial x_1}{\partial u} &= a' \cos v + b't \sin 5v \\
 x_2 &= c \sin v + bt \cos 5v \\
 \frac{\partial x_2}{\partial u} &= c' \sin v + b't \cos 5v \\
 x_3 &= h_0 + h_{01}t \sin 5v \\
 \frac{\partial x_3}{\partial u} &= h'_0 + h'_{01}t \sin 5v
 \end{aligned} \quad (24a)$$

and those for the other boundaries have the forms of

$$\begin{aligned}
 u = i \quad x_1 &= r_i \cos v \quad \frac{\partial x_1}{\partial u} = r'_i \cos v \\
 x_2 &= r_i \sin v \quad \frac{\partial x_2}{\partial u} = r'_i \sin v \quad (i = 1, 2, 3) \\
 x_3 &= h_i \quad \frac{\partial x_3}{\partial u} = h'_i.
 \end{aligned} \quad (24b)$$

Since  $t$  is a time variable, PDEs (1) subject to the basic functions  $t \sin 5v$  and  $t \cos 5v$  have no closed form solutions. The method developed in this paper will be used to solve this problem.

According to the above boundary conditions (24a), (24b) and (4), the solution functions for the upper surface are given as

$$\begin{aligned}
 x_1 &= \sum_{n=1}^N p_{11n} u^n \cos v + \sum_{n=1}^N p_{12n} u^n t \sin 5v \\
 x_2 &= \sum_{n=1}^N p_{21n} u^n \sin v + \sum_{n=1}^N p_{22n} u^n t \cos 5v \\
 x_3 &= \sum_{n=1}^N p_{31n} u^n + \sum_{n=1}^N p_{32n} u^n t \sin 5v
 \end{aligned} \quad (25)$$

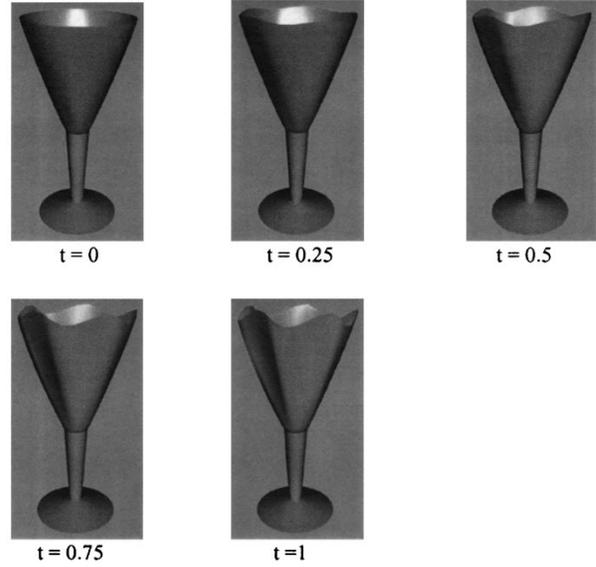


Fig. 2. Dynamic generation of a glass-like 3-D surface.

and those for the middle and bottom surfaces have the following unified forms

$$\begin{aligned}
 x_1 &= \sum_{n=1}^N \bar{p}_{11n} u^n \cos v \\
 x_2 &= \sum_{n=1}^N \bar{p}_{21n} u^n \sin v \\
 x_3 &= \sum_{n=1}^N \bar{p}_{31n} u^n.
 \end{aligned} \quad (26)$$

With the above-developed method, we can determine all the unknown constants in (25) and (26). Specifying the values of the shape parameters, density and damping coefficient in PDEs (1), and the geometric parameters in boundary conditions (24a) and (24b), fixing them and only changing the value of time  $t$ , we generate a series of surfaces dynamically. In Fig. 2, we give the surfaces at 5 different instants. The upper circular opening was consecutively deformed into a pedal-like shape.

For many objects, the boundary curves defining these objects change with the motion of the objects. Human limbs in motion are such an example. In the example below, we will demonstrate the dynamic generation of a vase-like 3-D surface by moving its one boundary curve and deforming its surface. The boundary conditions for this example are defined by

$$\begin{aligned}
 u = 0 \quad x_1 &= \lambda t + r_1 \cos v \quad \frac{\partial x_1}{\partial u} = r'_1 t^3 \cos v \\
 x_2 &= \lambda t + r_1 \sin v \quad \frac{\partial x_2}{\partial u} = r'_1 t^3 \sin v \\
 x_3 &= h_1 \quad \frac{\partial x_3}{\partial u} = h'_1 t^2 \\
 u = 1 \quad x_1 &= r_0 \cos v \quad \frac{\partial x_1}{\partial u} = r'_0 t^2 \cos v \\
 x_2 &= r_0 \sin v \quad \frac{\partial x_2}{\partial u} = r'_0 t^2 \sin v \\
 x_3 &= h_0 \quad \frac{\partial x_3}{\partial u} = h'_0 t^3.
 \end{aligned} \quad (27)$$

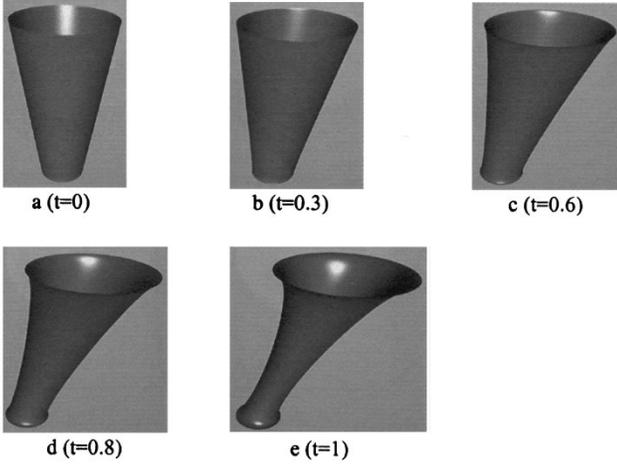


Fig. 3. Dynamic generation of a vase-like 3-D surface.

Same as above, PDEs (1) have no closed form solutions for the basic functions  $t^2 \sin v$ ,  $t^2 \cos v$ ,  $t^3 \sin v$  and  $t^3 \cos v$  in the boundary conditions (27).

According to (4) and the basic functions in (27), the solution functions for this problem can be written as the following forms:

$$\begin{aligned}
 x_1 &= \sum_{n=0}^N p_{11n} u^n t + \sum_{n=0}^N p_{12n} u^n \cos v + \sum_{n=0}^N p_{13n} u^n t^2 \cos v \\
 &\quad + \sum_{n=0}^N p_{14n} u^n t^3 \cos v \\
 x_2 &= \sum_{n=0}^N p_{21n} u^n t + \sum_{n=0}^N p_{22n} u^n \sin v + \sum_{n=0}^N p_{23n} u^n t^2 \sin v \\
 &\quad + \sum_{n=0}^N p_{24n} u^n t^3 \sin v \\
 x_3 &= \sum_{n=0}^N p_{31n} u^n + \sum_{n=0}^N p_{32n} u^n t^2 + \sum_{n=0}^N p_{33n} u^n t^3. \quad (28)
 \end{aligned}$$

The shape parameters in PDEs (1) are taken to be  $b_i = d_i = 1$ ,  $c_i = 2(i = 1, 2, 3)$ . The density and damping coefficient are set to  $\rho = \eta = 1$ . And the geometric parameters in boundary conditions (27) are specified as  $\lambda = 1$ ,  $r_0 = 0.4$ ,  $r_1 = 0.9$ ,  $h_0 = 0$ ,  $h_1 = 3$ ,  $r'_0 = 0.77$ ,  $r'_1 = -1.47$ , and  $h'_0 = -h'_1 = 0.3$ . Taking  $t = 0, 0.3, 0.6, 0.8$  and  $1$ , we obtain a sequence of the images of the deformed surface, as seen in Fig. 3, where the upper boundary curve is consecutively moved to new positions and the surface is also deformed accordingly.

The boundary conditions given in the above two examples are only for the purpose of illustration. Since the boundary constraints given in (2) and (3) are in a general form, more flexible and general boundary constraints can also be considered with the developed method. It is applicable to any complicated boundary conditions.

## V. DYNAMIC MANIPULATION TECHNIQUES

In the previous section, we have shown that when time variable  $t$  takes a series of values, the developed method can be used to animate 3-D deformable moving surfaces. At a given instant

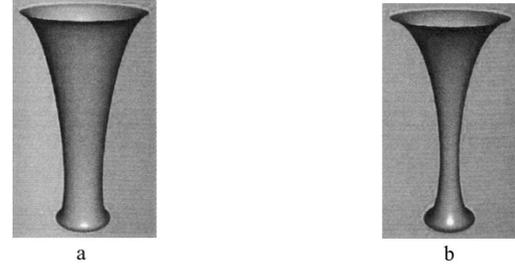


Fig. 4. Surface manipulation with the boundary tangent functions.

$t$ , the surface can be further manipulated by changing the other factors of PDEs (1) and boundary conditions (3). Their combination with the time variable provides us with effective dynamic manipulating tools. In this section, we will examine these shape manipulation techniques.

### A. Boundary Curve and Tangent Functions

Changing the coefficients of the boundary curve functions can scale the size of the boundary curves and using different boundary curve functions can create different boundary curves. Both methods will result in a different surface shape.

The boundary tangent functions provide another powerful shape manipulation means. Still using the same boundary conditions (27) and the same shape parameters and geometric parameters defining Fig. 3(a), but setting  $\lambda = 0$ , fixing  $t = 1$ , and taking  $r'_0 = 0.845$ ,  $r'_1 = -1.29$ ,  $h'_0 = 0.192$ ,  $h'_1 = -0.154$ , the surface in Fig. 4(a) is created. Then, setting  $r'_0 = 1.19$ ,  $r'_1 = -1.84$ ,  $h'_0 = 0.271$ ,  $h'_1 = -0.257$ , the surface is changed to that in Fig. 4(b).

### B. Shape Functions

In our previous examples, the shape functions in PDEs (1) were taken as constants. In general, the shape functions can be functions of parametric variables  $u, v$  and time variable  $t$ . When doing so, these shape functions become more powerful in shape manipulation. Here we use the following shape functions to demonstrate it.

$$\begin{aligned}
 b_i &= b_{i0}(1-t)u(u-1) \\
 c_i &= c_{i0}(1+t)u(u-1) \quad (i = 1, 2, 3) \\
 d_i &= d_{i0}(1+t^2)u(u-1). \quad (29)
 \end{aligned}$$

The geometric parameters were taken to be those that define Fig. 3(a), but  $\lambda$  is set to 0. Keeping  $t = 0.5$ ,  $r'_0 = 1.32$ ,  $r'_1 = -2.52$ , and  $h'_0 = -h'_1 = 0.3$  unchanged. When taking  $b_{10} = b_{20} = b_{30} = 1$ ,  $c_{10} = c_{20} = c_{30} = 2$ ,  $d_{10} = d_{20} = d_{30} = -1300$ , we obtain the surface shape in Fig. 5(a). When  $d_{i0}$  ( $i = 1, 2, 3$ ) are changed to  $-9000$ , the surface shape in Fig. 5(b) is produced. Clearly, shape functions exert a great influence on the surface shape.

From PDE (1) there are three shape functions for each position function, making a total of nine shape functions. Each of them affects the shapes of the generated surfaces. These nine shape functions can have many different combinations and their effects on surface shapes are profound. However, if we set  $b_1 = b_2 = b_3 = \xi$ ,  $c_1 = c_2 = c_3 = \zeta$  and  $d_1 = d_2 = d_3 = \beta$ , the

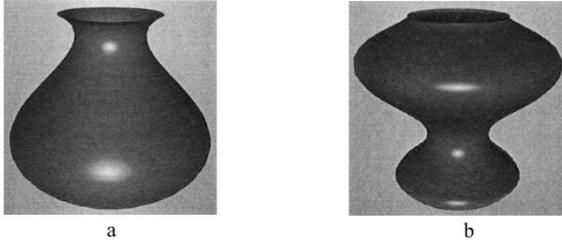


Fig. 5. Surface manipulation with shape functions.

smaller the absolute values of  $\xi$  and  $\zeta$ , or the larger that of  $\beta$ , the more concave the generated surfaces become.

### C. Density and Damping Coefficient

Any objects and surfaces moving or deforming with a high speed are subject to the inertia and damping forces according to the laws of physics. In order to describe their appearance more realistically, these forces must be considered.

In the previous examples, we fixed the density and damping coefficient to be  $\rho = \eta = 1$ . Now let us change their values and see how the surface shape is affected. Keeping the shape parameters, boundary conditions and the value of the time variable defining the surface in Fig. 5(a) unchanged, when taking the density to be  $\rho = 8$  and fixing the damping coefficient  $\eta = 1$ , we change the surface in Fig. 5(a) into that in Fig. 6(a). Then, changing the density and damping coefficient to  $\rho = 1$  and  $\eta = 9$ , respectively, the surface in Fig. 6(a) is deformed to Fig. 6(b).

Unlike the effect of the shape parameters, the influence of the density and damping coefficient on surface shapes is simple. From the above example, it is clear that the larger the density, the more concave the surface; the bigger the damping coefficient, the more convex the surface.

In fact, if we do not regard the density and damping coefficient as the physical properties of the surfaces, but as factors which manipulate the surfaces, we can also take the density and damping coefficient as functions of position variables  $u, v$  and time variable  $t$ . They again are powerful surface manipulation tools.

## VI. LOCALIZED GEOMETRIC OPERATIONS AND MULTIPLE-SURFACE MODELING

Most of the above examples are modeled with a single surface patch. Although this is able to demonstrate the advantages of the PDE based dynamic modeling method, one has to realize that more powerful modeling measures have to be introduced for complex shapes. In the following we first discuss how local deformations are achieved so that a surface can be sculpted to have more complex surface features. Then, we introduce the construction of three and four-sided patches. And finally, we give an example of the modeling of a more complex object.

### A. Local Deformation Technique

To locally deform a region of a surface with the developed method is to treat this region as a sub-patch within the original surface. The first step is to specify the boundaries of the deformed region. These local boundary curves are then repre-

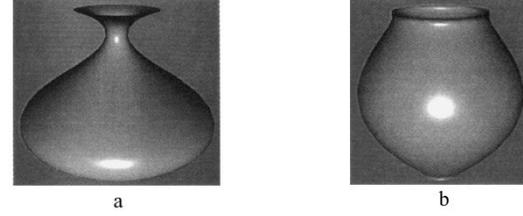


Fig. 6. Surface manipulation with density and damping coefficient.

sented through reparametrization. If the tangential continuity is also required between the local and the global surfaces, the local tangent boundary conditions can be determined according to the mathematical expressions of the local boundary curves.

To perform local deformation, one can use one of the three techniques: adjusting tangential boundary conditions of the local region; setting different forms of the shape functions and other parameters in PDE (1); and adding an additional deforming term to the solution (5). Among them, the first two techniques require the resolution of PDE (1) in the local region whereas the last does not. Therefore the last technique is the simplest and here we introduce this technique.

If we intend to perform a localized geometric operation in a local region of  $u_1 < u < u_2$  and  $v_1 < v < v_2$  of a given surface, we first define two new parameters  $\bar{u}$  and  $\bar{v}$  in the range of  $0 \leq \bar{u} \leq 1$  and  $0 \leq \bar{v} \leq 1$ . Next, the local position boundary conditions and tangent boundary conditions are determined from (5). If the boundaries of the local region are not along the original  $u$  and  $v$  parametric directions, it is always possible to obtain the position and tangent values of some points at the local boundaries from the global surface (5). Then, the new local boundary conditions are formulated from these points. Similar to (5), the mathematical equations of the local surface can be written in the following form after the introduction of the additional term

$$\begin{aligned}
 x_i(\bar{u}, \bar{v}, t) &= \sum_{j=1}^J \left\{ a_{ij1}(1 - 3\bar{u}^2 + 2\bar{u}^3) + a_{ij2}(\bar{u} - 2\bar{u}^2 + \bar{u}^3) \right. \\
 &\quad + a_{ij3}(3\bar{u}^2 - 2\bar{u}^3) + a_{ij4}(-\bar{u}^2 + \bar{u}^3) \\
 &\quad \left. + \sum_{n=4}^N [(n-3)\bar{u}^2 - (n-2)\bar{u}^3 + \bar{u}^n] p_{ijn} \right\} \\
 &\quad \times g_{ij}(\bar{v}, t) + \bar{x}_i \quad (i = 1, 2, 3) \quad (30)
 \end{aligned}$$

where  $\bar{u} = u_1 + \bar{u}(u_2 - u_1)$ ,  $\bar{v} = v_1 + \bar{v}(v_2 - v_1)$ , and  $\bar{x}_i$  is the additional term. Depends on whether the tangential boundary continuity is required, the additional term  $\bar{x}_i$  has two different forms.

When only the positional continuity is required,  $\bar{x}_i$  can be written as

$$\bar{x}_i = (\bar{u}\bar{v} - \bar{u}^2\bar{v} - \bar{u}\bar{v}^2 + \bar{u}^2\bar{v}^2) p_i(\bar{u}, \bar{v}) \quad (i = 1, 2, 3). \quad (31)$$

When the tangential continuity is also required, the above additional term takes the following form

$$\bar{x}_i = (\bar{u}\bar{v} - \bar{u}^2\bar{v} - \bar{u}\bar{v}^2 + \bar{u}^2\bar{v}^2)^2 p_i(\bar{u}, \bar{v}) \quad (i = 1, 2, 3) \quad (32)$$

where  $p_i(\bar{u}, \bar{v})$  ( $i = 1, 2, 3$ ) can be an arbitrary function whose forms determine the local deformation.

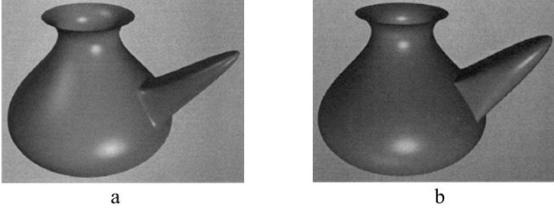


Fig. 7. Local deformation with different degrees of blending smoothness.

As an application of the above method, let us carry out a local geometric operation on the surface given in Fig. 5(a). The chosen local region is  $0.4 < u < 0.5$  and  $5.03 < v < 5.65$ . The unknown functions of the additional term are taken to be  $p_1(\bar{u}, \bar{v}) = \zeta \cos[v_1 + \bar{v}(v_2 - v_1)]$  for the  $x_1$  component,  $p_2(\bar{u}, \bar{v}) = \zeta \sin[v_1 + \bar{v}(v_2 - v_1)]$  for  $x_2$  component and  $p_3(\bar{u}, \bar{v}) = \zeta$  for  $x_3$  component. Using the additional term (32) and setting  $\zeta = 300$ , we have achieved the local deformation depicted in Fig. 7(a). Changing the additional term to (31) and setting  $\zeta = 19$ , the surface shape of the local deformation is changed to that in Fig. 7(b). In Fig. 7(a), both positional and tangential continuities at the local boundaries are guaranteed. As a result, a smooth transition surface appears between the original and the local surfaces. However, in Fig. 7(b), the tangential continuity at the local boundaries is not satisfied and only the position continuity is maintained. A sharp edge separates both surfaces, as the positional continuity only gives  $G^0$  smoothness. With this method, complex surfaces can be effectively created with a controlled blending smoothness ( $G^0$  or  $G^1$ ).

### B. Construction of $n$ -Sided Patches

With the above-developed local deformation technique, we can create a complex surface containing various surface details through a number of local deformations applied on the surface. Alternatively, a surface of a complex shape may be viewed as a collection of smaller and simpler patches connected together smoothly. The modeling of such a complex surface is thus transformed into the generation of simple surface patches with a set of constraints to stitch them together. Since any  $n$ -sided patches can be constructed from both four-sided and three-sided patches, in the following we will extend our method to generate these surface patches.

For a four-sided patch, when both the position and tangent constraints at its four sides are required, boundary conditions (2) should be extended to comprise those at the boundaries of  $v = 0$  and  $v = 1$ , i.e.,

$$\begin{aligned} v = 0 \quad x_i &= G_{i5}(u, t) & \frac{\partial x_i}{\partial u} &= G_{i6}(u, t) \\ v = 1 \quad x_i &= G_{i7}(u, t) & \frac{\partial x_i}{\partial u} &= G_{i8}(u, t) \\ & & & (i = 1, 2, 3). \end{aligned} \quad (33)$$

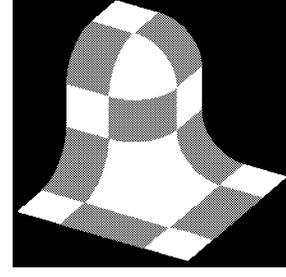


Fig. 8. Surface consisting of three- and four-sided PDE surface patches.

Transforming these boundary conditions into polynomial forms of the  $u$  parametric variable, (33) can be written as

$$\begin{aligned} v = 0 \quad x_i &= \sum_{n=0}^N b_{in1}(t)u^n & \frac{\partial x_i}{\partial u} &= \sum_{n=0}^N b_{in2}(t)u^n \\ v = 1 \quad x_i &= \sum_{n=0}^N b_{in3}(t)u^n & \frac{\partial x_i}{\partial u} &= \sum_{n=0}^N b_{in4}(t)u^n \\ & & & (i = 1, 2, 3) \end{aligned} \quad (34)$$

Setting  $v = 0$  and  $v = 1$  for (5) and its first partial derivatives with respect to the parametric variable  $v$ , we obtain the boundary functions of the surface (5) at the boundaries  $v = 0$  and  $v = 1$ . Equating these boundary functions to those of (34), we can obtain a set of linear equations. Taking the positional boundary conditions of  $v = 0, x_i = \sum_{n=0}^N b_{in1}(t)u^n$  as an example, the following linear equations are generated

$$\begin{aligned} b_{i01}(t) &= \sum_{j=1}^J a_{ij1}g_{ij}(0, t) & b_{i11}(t) &= \sum_{j=1}^J a_{ij2}g_{ij}(0, t) \\ b_{i21}(t) &= \sum_{j=1}^J \left[ -3a_{ij1} - 2a_{ij2} + 3a_{ij3} - a_{ij4} \right. \\ & \quad \left. + \sum_{n=4}^N (n-3)p_{ijn} \right] g_{ij}(0, t) \\ b_{i31}(t) &= \sum_{j=1}^J \left[ 2a_{ij1} + a_{ij2} - 2a_{ij3} + a_{ij4} \right. \\ & \quad \left. - \sum_{n=4}^N (n-2)p_{ijn} \right] g_{ij}(0, t) \\ b_{in1}(t) &= \sum_{j=1}^J p_{ijn}g_{ij}(0, t) \quad (n = 4, 5, \dots, N) \end{aligned} \quad (35)$$

Similar equations can also be obtained for other boundary conditions. By solving these linear equations, we can determine unknown constants  $p_{ijn}$  ( $i = 1, 2, 3; j = 1, 2, \dots, J; n = 4, 5, \dots, K < N$ ). Following the treatment given in (6)–(12), the rest unknown constants can be determined with (12). Then, the obtained surface functions  $x_i$  ( $i = 1, 2, 3$ ) can be used to generate the surface patch. Using this method, the four-sided patches in Fig. 8 are created.

For a three-sided patch, we take its 3 boundaries to be  $u = 0, v = 0$  and  $v = 1$ . The boundary conditions at the boundary



Fig. 9. Model of an aircraft.

$u = 0$  are still given by (3). With them, position functions (5) are modified to have the following form:

$$x_i(u, v, t) = \sum_{j=1}^J \left( a_{ij1} + a_{ij2}u + \sum_{n=2}^N p_{ijn}u^n \right) g_{ij}(v, t) \quad (i = 1, 2, 3). \quad (36)$$

Still taking the positional boundary condition of  $v = 0, x_i = \sum_{n=0}^N b_{in1}(t)u^n$  as an example, the resolution equations from this position boundary condition become

$$\begin{aligned} b_{i01}(t) &= \sum_{j=1}^J a_{ij1}g_{ij}(0, t) \\ b_{i11}(t) &= \sum_{j=1}^J a_{ij2}g_{ij}(0, t) \quad (n = 2, 3, \dots, N) \\ b_{in1}(t) &= \sum_{j=1}^J p_{ijn}g_{ij}(0, t). \end{aligned} \quad (37)$$

Similarly, the resolution equations from other boundary conditions can be obtained. Their resolution determines more unknown constants in (36). Then the mathematical operations given in (6)–(12) can be employed to determine the remaining unknown constants. In Fig. 8, a three-sided patch is produced.

It is well known that any  $n$ -sided patches can be decomposed into a number of four-sided patches or four-sided plus three-sided patches. Therefore, the above method can be used to construct any  $n$ -sided patches. In Fig. 8, a five-sided patch is generated.

### C. Modeling of a More Complex Object

Applying the above-developed method, we here generate the model of an aircraft. The aircraft is divided into four main parts: a fuselage, two wings, three tail components and four engine housings. These parts are further decomposed into a number of surface patches. The above-developed PDE based modeling method is employed to generate these surface patches. Then, they are assembled to produce the model of the aircraft as given in Fig. 9.

## VII. CONCLUSIONS AND FUTURE WORK

Dynamic modeling of 3-D deformable surfaces is an important issue in many computer graphics applications such as computer animation and virtual reality. However, due to the complexity of dynamic modeling, this problem has not been solved

satisfactorily. Existing dynamic modeling approaches rely on the use of numerical methods such as the finite element method and finite difference method, which are too slow for dynamic modeling, and therefore unsuitable for tasks requiring interactive or real-time performance.

In order to dynamically model 3-D deformable moving surfaces quickly and accurately, in this paper, we extended the PDE based static modeling approach to dynamic modeling. A set of fourth order dynamic partial differential equations have been introduced for this purpose. They consider the effects of inertia and damping on the motion and deformation of a surface. By constructing an effective solution function which consists of the trial functions and the basic functions in the boundary conditions, all the boundary conditions are satisfied exactly. In addition, the dynamic partial differential equations of the fourth order are solved by minimizing its residual functions.

A comparison has been made between this method, the closed form resolution method and the existing Fourier series method and pseudospectral method. It has been found that the developed method can generate 3-D surfaces almost as fast and accurately as the closed form resolution method, far more efficient and accurate than the existing analytical and numerical methods. Moreover, it can cope with the complicated problems of dynamic modeling of 3-D deformable moving surfaces which cannot be solved with the closed form resolution method.

With the developed method, we have demonstrated how to generate 3-D deformable moving surfaces dynamically with two examples. This method can also be employed as a dynamic manipulation means of 3-D surfaces. We have investigated the effects of the boundary curve and tangent functions, shape functions, density and damping coefficient on the surface shape, and demonstrated that they can be used as an effective tool for the dynamic manipulation of 3-D deformable moving surfaces.

We have also introduced an effective local deformation technique with which not only the positional continuity at the local boundaries, but also the tangential continuity can be satisfied. The local deformable patch can be connected to the original surface with a controllable degree of continuity. The developed method has been further extended to the construction of three-sided, four-sided and  $n$ -sided patches. An arbitrarily complex surface can be readily created as a collection of these basic surface patches.

In this paper, we have discussed the basic theory, the resolving method and some applications of PDE based dynamic modeling, some of which have not been discussed in existing literature. In order for this method to have a wider impact on the modeling of deformable moving surfaces, and to make this approach user-friendly for interactive applications, we will produce an interactive user-interface, which transforms the user interaction into mathematical expressions, and hide the implementation details from the user. This idea has been tested in another related project [11] and will be included in our future development.

## REFERENCES

- [1] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 4th ed. New York: Academic, 1997.
- [2] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proc. Computer Graphics*, vol. 21, 1987, pp. 205–214.

- [3] D. Terzopoulos and K. Fleischer, "Modeling inelastic deformation: Viscoelasticity, plasticity, fracture," *Comput. Graph.*, vol. 22, no. 4, pp. 269–278, 1988.
- [4] —, "Deformable models," *Vis. Comput.*, vol. 4, pp. 306–331, 1988.
- [5] D. Terzopoulos and D. Metaxas, "Dynamic 3-D models with local and global deformations: Deformable superquadrics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 703–714, 1991.
- [6] H. Qin and D. Terzopoulos, "Dynamic NURBS swung surfaces for physics-based shape design," *Comput. Aided Des.*, vol. 27, no. 2, pp. 111–127, 1995.
- [7] G. Celniker and D. Gossard, "Deformable curve and surface finite-elements for free-form shape design," *Comput. Graph.*, vol. 25, no. 4, pp. 257–266, 1991.
- [8] U. Gdkbay and B. zgc, "Animation of deformable models," *Comput. Aided Des.*, vol. 26, no. 12, pp. 868–875, 1994.
- [9] C. Mandal, H. Qin, and B. C. Vemuri, "Dynamic modeling of butterfly subdivision surfaces," *IEEE Trans. Visual. Comput. Graphics*, vol. 6, pp. 265–287, 2000.
- [10] M. I. G. Bloor and M. J. Wilson, "Generating blend surfaces using partial differential equations," *Comput. Aided Des.*, vol. 21, no. 3, pp. 165–171, 1989.
- [11] J. J. Zhang and L. H. You, "PDE based surface representation," *Comput. Graph.*, vol. 26, no. 1, pp. 89–98, 2002.
- [12] —, "Surface representation using second, fourth and mixed order partial differential equations," in *Proc. Int. Conf. Shape Modeling Applications*. Los Alamitos, CA, May 7–11, 2001, pp. 250–256.
- [13] J. Barhak and A. Fischer, "Parameterization for reconstruction of 3-D free-form objects from laser-scanned data based on a PDE method," *Vis. Comput.*, vol. 17, pp. 353–369, 2001.
- [14] C. J. Evans, M. I. G. Bloor, and M. J. Wilson, "Representing the time-dependent geometry of the heart for fluid dynamical analysis," in *Proc. 9th IMA Conf. Mathematics Surfaces*. New York, Sept. 4–7, 2000, pp. 459–469. Mathematics of Surfaces IX.
- [15] M. I. G. Bloor and M. J. Wilson, "Method for efficient shape parametrization of fluid membranes and vesicles," *Phys. Rev. E*, vol. 61, no. 4, pp. 4218–29, 2000.
- [16] S. Y. Cheng, M. I. G. Bloor, A. Saia, and M. J. Wilson, "Blending between quadric surfaces using partial differential equations," in *Proc. Advances Design Automation, Computer Computational Design*, vol. 1, B. Ravani, Ed., 1990, pp. 257–263.
- [17] H. Du and H. Qin, "Direct manipulation and interactive sculpting of PDE surfaces," in *Proc. Computer Graphics Forum*, vol. 19, 2000, pp. 261–270.
- [18] J. M. Brown, M. I. G. Bloor, M. S. Bloor, and M. J. Wilson, "The accuracy of B-spline finite element approximations to PDE surfaces," *Computer Methods Appl. Mech. Eng.*, vol. 158, no. 3–4, pp. 221–234, 1998.
- [19] Z. C. Li, "Boundary penalty finite element methods for blending surfaces, I. basic theory," *J. Comput. Math.*, vol. 16, pp. 457–480, 1998.
- [20] —, "Boundary penalty finite element methods for blending surfaces, II. Biharmonic equations," *J. Comput. Appl. Math.*, vol. 110, pp. 155–176, 1999.
- [21] Z. C. Li and C.-S. Chang, "Boundary penalty finite element methods for blending surfaces, III, superconvergence and stability and examples," *J. Comput. Appl. Math.*, vol. 110, pp. 241–270, 1999.
- [22] M. I. G. Bloor and M. J. Wilson, "Representing PDE surfaces in terms of B-splines," *Comput. Aided Des.*, vol. 22, no. 6, pp. 324–331, 1990.
- [23] —, "Using partial differential equations to generate free-form surfaces," *Comput. Aided Des.*, vol. 22, no. 4, pp. 202–212, 1990.
- [24] —, "Spectral approximations to PDE surfaces," *Comput. Aided Des.*, vol. 28, no. 2, pp. 145–152, 1996.
- [25] H. Ugail, M. I. G. Bloor, and M. J. Wilson, "Techniques for interactive design using the PDE method," *ACM Trans. Graph.*, vol. 18, no. 2, pp. 195–212, 1999.
- [26] —, "Manipulation of PDE surfaces using an interactively defined parameterization," *Comput. Graph.*, vol. 23, pp. 525–534, 1999.
- [27] L. H. You, J. J. Zhang, and P. Comninos, "Cloth deformation modeling using a plate bending model," in *Proc. 7th Int. Conf. Central Europe Computer Graphics, Visualization Interactive Digital Media*, 1999, pp. 485–491.
- [28] J. J. Zhang, L. H. You, and P. Comninos, "Computer simulation of flexible fabrics," in *Proc. 17th Eurographics Conf.*, 1999, pp. 27–35.
- [29] L. H. You, J. J. Zhang, and P. Comninos, "A volumetric deformable muscle model for computer animation using weighted residual method," *Comput. Meth. Appl. Mech. Eng.*, vol. 190, pp. 853–863, 2000.
- [30] K. A. Stroud, *Further Engineering Mathematics*. New York: Macmillan, 1996.



**Lihua You** received the Ph.D. degree in civil and structural engineering from Chongqing University of Architecture, Chongqing, China in 1990.

He is a Research Fellow at Bournemouth University, Bournemouth, U.K.



**Jian J. Zhang** received the Ph.D. degree in mechanical engineering from Chongqing University, Chongqing, China in 1987.

He is Professor of Computer Graphics at the National Centre for Computer Animation, Bournemouth University, Bournemouth, U.K. His research interests include computer graphics, computer animation, physics-based simulation, and geometric modeling.