Ugail H (2004): "Time-Dependent Shape Parameterisation of Complex Geometry using PDE Surfaces", *Geometric Modelling and Computing,* , M.L. Lucian and M. Neamtu (eds.), Nashboro Press ISBN: 0-0-9728482-3-1, pp. 501-512.

Shape Parameterization of Complex Time-Dependent Geometry using PDE Surfaces

Hassan Ugail

Abstract. Here the problem of efficient shape parameterization for modeling complex time-dependent geometry is considered. The geometry is created using PDE surfaces where shape generation is treated as boundary value problems. By means of implementing a closed-form analytic solution to the chosen PDE the system allows real time generation of complex geometry. It is shown how the geometry of the object in question can be created interactively and then the shape can be parameterized intuitively where the chosen shape parameters can then be made time dependent. The technique is demonstrated by way of practical examples such as the simulations of a beating heart and undulatory swimming motion of a dolphin.

§1. Introduction

The computer modeling of realistic time dependent geometry for simulation of real world scenarios is a challenging task. Essentially one could think that the computer generated time dependent geometry, otherwise known as computer animation, relies upon the ability for the three dimensional geometry of an object to deform into a series of other objects through time. The bulk of Computer-Aided Design (CAD) techniques involve the use of polylines and polygonal faceted representation of objects. Therefore the modeling of the time-dependent geometry within the computer animation community has mostly revolved around the idea of time based interpolation, starting from a given base geometry to the target geometry. For example, a commonly used technique known as key-framing (e.g. [10]), is based on this idea where the animator can specify the extents

xxx and xxx (eds.), pp. 1–4.

ISBN 0-9728482-x-x

of the motion and then the computer with the aid of an interpolation algorithm fill in the 'in betweens' to create a smooth animation. Realizing the difficulties involved in creating realistic animation using such techniques has made researchers to motivate towards other approaches.

One such approach is the use of morphing where a seamless correspondence between an initial and a target geometry is generated, for example by using blending functions (e.g. [8]). Other approaches include the use of finite-difference and finite elements for the integration of energy-based Lagrange equations (e.g. [11]). These and similar techniques have successfully been used to model time-dependent deformable bodies.

In existing systems for generating time dependent geometry of complex shapes, one of the difficulties a user would face is that the techniques often do not allow direct control of the dynamic motion of the object. This is due to the fact that there exist a wider gap between the geometry generation tools and those used for animating such geometry. One reason for this is the lack of efficient techniques for parameterizing both the static and time-dependent geometry. The notion of parametric design here implies that the 'shape' or the 'design' can be described by a set of variables or parameters and can therefore be manipulated by varying these parameters in a controlled way. An important point noteworthy here is that such a design parameter space must allow scope for suitable variation of a given design whilst the design space should not be too large prohibiting intuitive shape manipulation.

The focus of this paper is on the use of PDE surfaces generating shape parameterizations for efficient simulation of time-dependent geometry. PDE surfaces are generated as solutions to elliptic Partial Differential Equations (PDEs), where the problem of surface generation is treated as a boundary-value problem with boundary conditions imposed around the edges of the surface patch [3]. PDE surfaces can be thought to be a powerful shape modeling technique [6, 13, 14]. For example, it has been demonstrated how a designer sitting in front of a workstation is able to create and manipulate complex geometry interactively in real time [13]. Furthermore, it has been shown that complex geometry can be efficiently parameterized both for intuitive shape manipulation [14] and for design optimization [5]. Moreover, recently Du and Qin (e.g. [6, 7]) have made a number of important contributions to this work. For example, they were able to integrate PDE surfaces with physics-based modeling techniques that enable the interactive sculpting and local control of complex geometry. They have, furthermore, extended their PDE techniques to encompass solid modeling and provide users with a set of direct editing toolkits to model real-world objects with interior material distribution.

The essential aim of this paper is to demonstrate how PDE surfaces can be used to create efficient time-dependent shape parameterization of complex geometry by allowing the associated design parameters to be func-

tions of time. In other words the focus here is to create time-dependent geometric models whose motion can be controlled using a handful of design parameters. The work presented here differs from the previous work on shape parameterization (e.g. [14, 16]) in that the design parameters are taken to be functions of time. To demonstrate the ideas presented here, practical examples involving complex time dependent geometry are discussed.

§2. PDE Surfaces

A PDE surface is a parametric surface patch $\underline{X}(u,v)$, defined as a function of two parameters u and v on a finite domain $\Omega \subset R^2$, by specifying boundary data around the edge region of $\partial\Omega$ and by regarding the coordinates of a point u and v as a mapping from that point in Ω to a point in the physical space. The boundary data are usually specified in the form of function boundary conditions at the edges of $\underline{X}(u,v)$ and a number of its derivatives on $\partial\Omega$. The most widely used PDE is based on the elliptic biharmonic equation namely,

$$\left(\frac{\partial^2}{\partial u^2} + a^2\frac{\partial^2}{\partial v^2}\right)^2 \underline{X}(u,v) = 0, \tag{1}$$

where the parameter a is a special design parameter [3]. In short one could think that the PDE method generates a smooth surface patch by solving an elliptic PDE such as Equation (1) subject to a set of boundary conditions that are imposed at the edges of the surface patch.

Note that the parameter a produces a 'waist' effect within the interior of the surface patch where the higher the value of 'a' the more waist it produces. This technique has been utlised in an interactive setting to control the interior of the surface as discussed in detail in [13].

2.1. Solution of the PDE

For the work described here, by restricting ourselves to periodic boundary conditions the closed form analytic solution of Equation (1) is utilized. Choosing the parametric region to be $0 \le u \le 1$ and $0 \le v \le 2\pi$, the periodic boundary conditions can be expressed as, $\underline{X}(0,v) = \underline{P}_0(v)$, $\underline{X}(1,v) = \underline{P}_1(v)$, $\underline{X}_u(0,v) = \underline{d}_0(v)$, and $\underline{X}_u(1,v) = \underline{d}_1(v)$. Here the boundary conditions $\underline{P}_0(v)$ and $\underline{P}_1(v)$ define the edges of the surface patch at $u = 0$ and $u = 1$ respectively. Using the method of separation of variables and spectral approximations [4], the approximate analytic solution of Equation (1) can be written as,

$$\underline{X}(u,v) \simeq \underline{A}_0(u) + \sum_{n=1}^{N}[\underline{A}_n(u)\cos(nv) + \underline{B}_n(u)\sin(nv)] + \underline{R}(u,v), \tag{2}$$

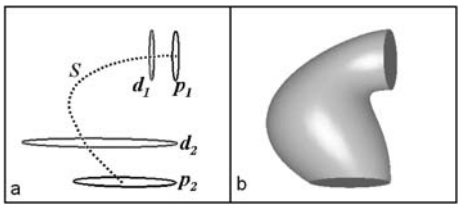

Fig. 1. A typical PDE surface patch. (**a**) The boundary curves and the spine (shown as the dotted line). (**b**) The corresponding PDE surface patch.

where

$$\underline{A}_0 = \underline{a}_{00} + \underline{a}_{01}u + \underline{a}_{02}u^2 + \underline{a}_{03}u^3, \tag{3}$$

$$\underline{A}_n = \underline{a}_{n1}e^{anu} + \underline{a}_{n2}ue^{anu} + \underline{a}_{n3}e^{-anu} + \underline{a}_{n4}ue^{-anu}, \tag{4}$$

$$\underline{B}_n = \underline{b}_{n1}e^{anu} + \underline{b}_{n2}ue^{anu} + \underline{b}_{n3}e^{-anu} + \underline{b}_{n4}ue^{-anu}, \tag{5}$$

where $\underline{a}_{00}, \underline{a}_{01}, \underline{a}_{02}, \underline{a}_{03}$ $\underline{a}_{n1}$, $\underline{a}_{n2}$, $\underline{a}_{n3}$, $\underline{a}_{n4}$, $\underline{b}_{n1}$ $\underline{b}_{n2}$, $\underline{b}_{n3}$ and $\underline{b}_{n4}$ are vector constants, whose values are determined by the imposed boundary conditions at $u = 0$ and $u = 1$. Note that the terms $\underline{A}_0$, $\underline{A}_n$ and $\underline{B}_n$ are not by choice but are by the nature of this particular analytic solution scheme.

The term $\underline{R}(u, v)$ is a truncation term given as $\underline{R}(u, v) = \underline{r}_1(v)e^{wu} + \underline{r}_2(v)e^{wu} + \underline{r}_3(v)e^{-wu} + \underline{r}_4(v)e^{-wu}$ where $\underline{r}_1(v)$, $\underline{r}_2(v)$, $\underline{r}_3$ and $\underline{r}_4(v)$ are vector constants and w is taken to be $N + 1$. For a given set of boundary conditions, the value of N is usually chosen so as to provide a close approximate Fourier series of the boundary conditions. The vector constants $\underline{r}_i(v), i = 1...4$ in $\underline{R}(u, v)$ are then chosen by computing the difference between the original boundary conditions and its approximation based on the chosen Fourier series, thus ensuring that the surface shape, although is approximate to the Equation (1), does indeed satisfy exactly at the boundaries. More details on this solution scheme can found in [4].

Drawing our attention to the term $\underline{A}_0$ in the Equation (3), we note that it takes the form of cubic polynomial, Using the solution technique described in Equation (2) one could therefore observe that a surface point $\underline{X}(u, v)$ may be regarded as being composed of sum of a vector $\underline{A}_0$ giving the position on the 'spine' of the surface and a radius vector defined by the term $\sum_{n=1}^{N}[\underline{A}_n(u)\cos(nv)+\underline{B}_n(u)\sin(nv)]+\underline{R}(u, v)$ providing the position of $\underline{X}(u, v)$ relative to the spine. Thus, the $\underline{A}_0$ term describes the spine or the skeleton of a PDE surface.

2.2. Shape Parameterization for Efficient Shape Manipulation

In previous works (e.g. [13, 14]) it has been shown how a user can interactively create a complex piece of geometry and then manipulate it using an interactively defined set of shape parameters. Figure 1 illustrates

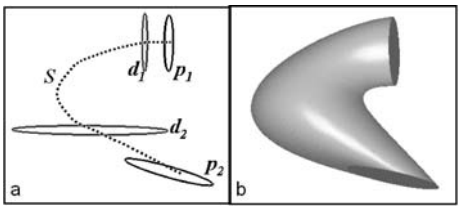

Fig. 2. An interactively manipulated PDE surface starting from the shape shown in Figure 1. (**a**) The boundary curves and the corresponding spine (shown as the dotted line). (**b**) The resulting PDE surface patch.

the basic idea behind this technique, where the boundary curves shown in Figure 1(a) are interactively created and the corresponding shape shown in Figure 1(b) is then generated in real time by solving Equation (1). Note that the curves P_1 and P_2 (known as 'position curves') define the edges of the surface shape and the curves d_1 and d_2 (known as 'derivative curves') define the derivative conditions. Here the derivative conditions are defined by means of creating a vector field corresponding to the difference between the points on the curves marked P_1 and P_2 and those marked d_1 and d_2 respectively, corresponding to the boundary conditions on the function $\frac{\partial \mathbf{X}}{\partial n}$ such that, $\frac{\partial \mathbf{X}}{\partial n} = [\mathbf{p}(v) - \mathbf{d}(v)]\, s$, where s is a scalar.

Figure 2(b) shows the shape of a resulting surface after the geometry has been manipulated though an interactively defined parameterization. This manipulation has been carried out by using the interactive PDE shape parameter model discussed in [14], where the parameterized boundary curves are used to define the shape of the surface. Essentially, this parameterization is defined in such a way that linear transformations, such as translation, rotation and dilation, of the boundary curves can be carried out interactively by the designer. To achieve this the designer is presented with a design interface which enables to modify the associated boundary curves in an intuitive manner [13, 14].

In Figures 1(a) and 2(a) the dotted lines show the shape of the spine of the corresponding PDE surface patches. Note that the spine can also be utilized as an intuitive shape manipulation tool [15]. One way of doing this is to imagine the vector valued constants $\underline{a}_{00}, \underline{a}_{01}, \underline{a}_{02}, \underline{a}_{03}$ in the $\underline{A}_0$ as a set of design parameters whose values can be interactively changed. Other methods such as defining the spine in terms of a spline control curve are also possible. It is important to bear in mind that the constants $\underline{a}_{00}, \underline{a}_{01}, \underline{a}_{02}, \underline{a}_{03}$ in the $\underline{A}_0$ are dependent on the boundary conditions chosen. Therefore, when these parameters are interactively changed in order to manipulate the surface shape, the boundary conditions will indeed change accordingly. Hence the user has a choice of using boundary

conditions or the spine or even a combination of both to manipulate the shape.

§3. Shape Parameterization of Time-Dependent Geometry

From the above discussion one can clearly see that the effect of varying the chosen parameters, in terms of the boundary curves defining the shape and via the use of the spine, on the surface shape is easy even for a novice user to appreciate. Due to the very nature of PDE surfaces, which are generated by specifying the boundary curves and which essentially reflect the key features of the shape to be modeled, only a small number of design parameters are required since the shape is entirely determined by the information specified around the boundary curves. This enables efficient shape parameterization of time dependent geometry by taking design parameters to be functions of time thus enabling a user to create time-dependent geometric models whose motion can be controlled using a handful of design parameters.

Another important point noteworthy here is that the use of semi-analytic solution technique described above enables fast generation of the geometry, which usually takes a fraction of a second, thus enabling real-time animations. For the work described here, we take an initial geometry and introduce shape parameters on the boundary conditions and on the spine, where the parameters are made time-dependent in order to simulate the animation process that is of interest. Thus, given an initial parameterization the geometry is re-generated for each time step of the animation process where the time-dependent parameters are calculated at each stage and the corresponding PDE surfaces are generated. Noticeably, this process has the advantage over existing animation techniques such as key framing, in that the animation process is essentially controlled by the animator via the time dependent shape parameters.

To illustrate the methodology we have chosen to discuss examples of practical nature. In particular, we present two examples, namely the modeling of a ventricle of a human heart where the beating of the heart is simulated, and the modeling of a dolphin whose swimming is simulated. For both the examples, the generic geometry of the object is created using a set of smoothly connected PDE surfaces and a set of time-dependent shape parameters, based on the boundary curves and the spine (where necessary) are introduced to generate animations.

3.3. Time-Dependent Geometry of a Beating Heart

As a first example for illustrating the modeling of time-dependent shape parameterization using PDE surfaces we create the left ventricle of the human heart shown in Figure 3. This particular geometry involves four separate PDE surface patches which are appropriately joined with common boundaries. Figure 3(a) shows the position boundary curves defining

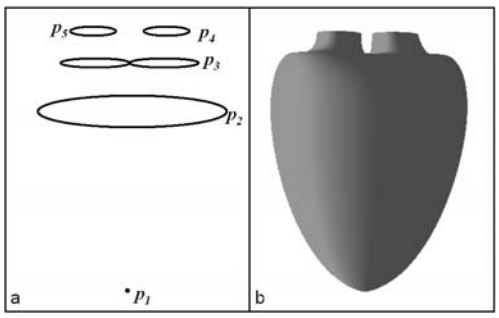

Fig. 3. The generic shape of a left ventricle of a human heart. (**a**) The position boundary curves. (**b**) The resulting shape of the left ventricle generated using a combination of four separate PDE surface patches.

the entire ventricle shape. For the sake of bringing clarity to the illustration the position boundary curves are only shown here thus omitting the corresponding derivative curves from the Figure. As discussed previously, the derivative boundary curves are generated by means of transforming the corresponding position curves, and in the cases when the surface patches meet each other, the derivative curves are taken in a manner such that tangent plane continuity is ensured between the patches. Thus, as shown in Figure 3(b) , the four surface patches which make up the left ventricle are created using separate surface patches between the curves P_1 and P_2, P_2 and P_3, P_3 and P_4 and P_3 and P_5. Note that the curve P_1 was initially taken to be a circle where its radius was then effectively reduced to zero in order to create a point in 3-space. Note also that the curve P_3 is a single curve formed by means of joining two ellipses where the left ellipse and the curve P_5 and the right ellipse and the curve P_4 form two separate surface patches.

In order to simulate the time-dependent motion of the left ventricle we utilize the information provided in Smith and Kampine [12] describing the changes in the ventricular shape in a heart beat. Essentially a heat beat involves three components of motion, namely, axial contraction, radial contraction and twisting motion. To model these three modes of motion with realistic changes in volume and dimension of the heart we define the time dependent parameters based on the boundary conditions shown in Figure 3(a). Thus, taking t to be the time the analytic forms of these boundary conditions are given as follows.

$$p_2(v,t) = (x_2 + r_2(t)\cos(v+\alpha(t)), y_2 + r_2(t)\sin(v+\alpha(t)), z_2 - z(t)), \quad (6)$$

$$\begin{aligned} p_3(v,t) = (&x_3 + r_3(t)\cos(v+\alpha(t))\sin(v-\alpha(t)),\\ &y_3 + r_3(t)\sin(v+\alpha(t)), z_3 - z(t)), \end{aligned} \quad (7)$$

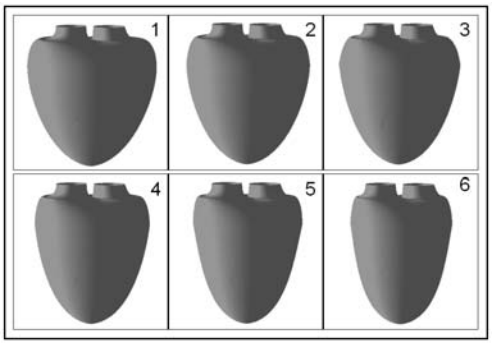

Fig. 4. A sequence of shapes showing a typical cardiac cycle simulated using the time-dependent shape parameters introduced on the boundary curves of the corresponding PDE surface patches.

$$p_4(v,t) = (x_4 + r_4\cos(v+\alpha(t)), y_4 + r_4\sin(v+\alpha(t)), z_{top} - z(t)), \quad (8)$$

$$p_5(v,t) = (x_5 + r_5\cos(v+\alpha(t)), y_5 + r_5\sin(v+\alpha(t)), z_{top} - z(t)), \quad (9)$$

where

$$z(t) = z_{ch}\sin^2(t\pi/t_m), \quad (10)$$

$$r_2(t) = r_2(1 - r_{ch}\sin^2(t\pi/t_m), \quad (11)$$

$$r_3(t) = r_3(1 - r_{ch}\sin^2(t\pi/t_m), \quad (12)$$

and

$$\alpha(t) = \alpha_{ch}\sin^2(t\pi/t_m). \quad (13)$$

The parameter t_m is the time period of a typical cardiac cycle which was taken to be 0.8. The function $z(t)$ controls the axial contraction where the parameter r_{ch} was taken to be 0.7. Similarly, the function $r(t)$ controls the radial contraction where the parameter r_{ch} was taken to be 0.3. Finally the twisting motion is generated via the parameter $\alpha(t)$ with α_{ch} accounting for the amount of twist in the ventricle. Here the value of α_{ch} was taken to be $\pi/4$.

With the above parameters describing the dynamics of heart it is seen that realistic animations accounting for the changes in the volume and dimension of the ventricle for a typical cardiac cycle can be generated in real time. Figure 4 shows a series of images to illustrate this. Note that the twisting motion although incorporated in the parameterization, does not allow prominent changes in the geometry and therefore may not be observed in the images shown Figure 4.

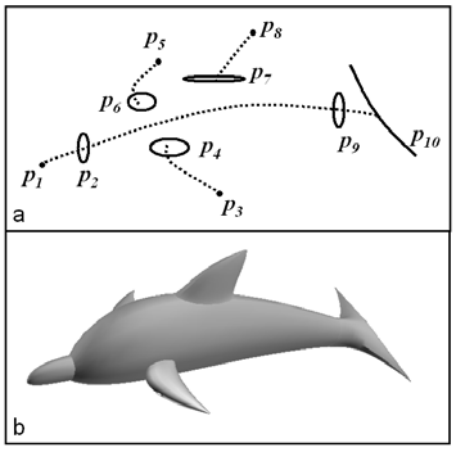

Fig. 5. A PDE based dolphin shape. (**a**) The boundary curves and the corresponding spines (marked as dotted lines). (**b**) The corresponding generic shape of the dolphin generated using a combination of six separate PDE surface patches.

3.4. Time-Dependent Geometry of a Swimming Dolphin

In this example we show how it is possible to parameterize the shape of a dolphin so as to simulate the challenging task of animating its swimming motion. Drawing from the pioneering work of the English mathematician Sir James Lighthill on the locomotion of aquatic animals we know that such motion is generated by the undulatory body motions that impart momentum to the surrounding [9]. Here the exception being the evolutionarily less successful propulsion methods such as beating cilia and jet reactions. Such aquatic body deformation essentially involves three modes namely steady swimming, rapid starting, and turning.

Here we discuss how the shape of the dolphin created and parameterized to simulate its steady swimming motion. For the purpose of simulating the swimming motion we mainly use the spine of the PDE surface generating the main body shape of the dolphin. Figure 5(a) shows the position boundary conditions that generate a generic shape of a dolphin using six separate PDE surface patches. Again, like the previous example of the ventricle shape, for the sake of clarity the position boundary curves are only shown here as it is intuitive to imagine the corresponding shapes of the derivative curves. The curves P_1, P_3, P_5 and P_8 were initially chosen to be circles whose radii were then effectively reduced to zero so as to form points in the 3-space. The curves P_2 and P_9 defining the main body shape of the dolphin were appropriately chosen ellipses in 3-space. The curve P_{10} representing the tail flukes were created using a cubic B-spline

curve whose shape was determined by using the flukes measurements for the Common Dolphins [1]. The curves P_4, P_6 and P_7, which form the body attachment for the lateral fins and the dorsal fin respectively, lie on the main body surface where the portions have been trimmed off from the main body surface. These trimming processes were performed via the use of the (u, v) parameter space using the techniques outlined in [13]. The broken lines in Figure 5 (a) show the spines of the associated PDE surface patches. As can be observed, the spines representing the beak, main body and the flukes forms a continuous curve which we shall refer to as $S_d(u)$. This continuity in the spines forming curve $S_d(u)$ is asserted by means of ensuring that tangent continuity is always maintained at the points where the curves meet.

To model the dynamic body deformation for the steady swimming motion of the dolphin we assume the curve $S_d(u)$ for the main body of the dolphin represents the backbone of the dolphin. Since the backbone of the dolphin is flexible we model $S_d(u)$ to be a time dependent piecewise linear curve $B(u, t)$ consisting of n segments with length L where t is the time. Thus, taking the shape of the rest position of the dolphin's backbone spine to be $S_d(u)$, the polynomial $B(u, t)$ interpolates $n+1$ control points $B_i(u, t), i = 1, ..., n+1$ where the control points are constrained so as the Euclidean distance between the adjacent points is L/n. The flexibility along the backbone is controlled by associating an angle θ_i for the first n control points of $B_i(u, t)$ such that the difference between adjacent angles is defined as a control parameter. In order to model the steady undulatory motion the angle θ_i is modulated along the spine according to a general traveling sine wave given as,

$$B_i(u, t) = \Omega S_d(u) \sin\left(\frac{2\pi}{\lambda}(S_d(u) - \alpha t)\right), \tag{14}$$

with

$$\theta_i = \tan^{-1}\left(\frac{\partial B_i(u, t)}{\partial u}\right). \tag{15}$$

Similar steady undulatory motion models for slender bodied aquatic animals are discussed elsewhere. e.g. ([17, 2, 9]).

With the above formulation the steady motion of the dolphin illustrated in Figure 5 was simulated by taking $\Omega = 2.1$, $\lambda = 1.1$, $\alpha = 0.5$ and taking the rest position of the combined spine $S_d(u)$ to lie along the y-axis of the 3-space. Figure 6 shows a series of images illustrating the motion of the dolphin.

§4. Conclusions

In this paper it is shown how the generic geometry of objects can be generated using PDE surfaces where upon time dependence can then be built

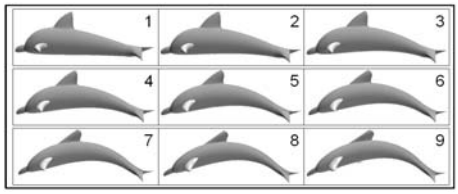

Fig. 6. A sequence of shapes showing the steady motion of the dolphin using the time-dependent shape parameters introduced on the spine curve $S_d(u)$.

into the geometry by means of a small set of shape parameters. The shape parameterization introduced here are essentially based on the boundary curves and the spine or the skeleton. Thus, it is shown that the shape parameters can be chosen intuitively without requiring prior knowledge on the part of the animator concerned on solving complex PDEs and how such PDEs behave upon the changes in their boundary conditions. To demonstrate the methodology we have discussed examples of time-dependent geometry of realistic objects such as the beating of a heart and the steady swimming motion of a dolphin. Again an important point to note here is the ease with which the time-dependent shape parameterization has been set up. i.e. in the case of the heart the parameters were introduced on the corresponding boundary conditions and in the case of the dolphin the parameters were simply introduced on the spine.

There are many possible applications of this work. For example, with the PDE based interactive design and manipulation tools (e.g. [15, 14, 6]) along with the time-dependent shape parameterization techniques presented here there is potential for digital artists and computer animators to take their work into a more interactive and controllable domain.

§5. References

1. Alpers, A., *A Book of Dolphins*, John Murray, London, 1960.
2. Blake, R. W., *Fish Locomotion*, Cambridge University Press, Cambridge, 1983.
3. Bloor, M. I. G., and M. J. Wilson, Using partial differential equations to generate freeform surfaces. Computer-Aided Design, **22** (1990), 202–212.
4. Bloor, M. I. G., and M. J. Wilson, Spectral approximations to PDE surfaces. Computer-Aided Design, **28** (1996), 145–152.

5. Bloor, M. I. G., and M. J. Wilson, Method for efficient shape parameterization of fluid membranes and vesicles. Physical Review E, **61**(4) (2000), 4218–4229.
6. Du, H., and H. Qin, Direct manipulation and interactive sculpting of PDE surfaces. Computer Graphics Forum (Proceedings of Eurographics 2000), **19**(3) (2000) 261-270.
7. Du, H., and H. Qin, Integrating physics-based modeling with PDE solids for geometric design, in *Pacific Graphics 2001*, Tokyo, Japan, (2001), 198–207.
8. Lee, S., G. Wolberg, and S. Y. Shin., Polymorph: morphing among multiple images, IEEE Computer Graphics and Applications. **1**(18) (1998), 58–71.
9. Lightlhill, M. J., Hydrodynamics of aquatic animal propulsion, Ann. Rev. Fluid Mech. **1** (1969), 413–446.
10. Reeves, W., Inbetweening for computer animation utilizing moving point constraints, Computer Graphics, **15**(3) (1981), 263–269.
11. Metaxas, D., and D. Terzopolous, Dynamic deformation of solid primitives with constraints, Computer Graphics, **26**(2) (1992), 309–312.
12. Smith, J. J., and J. P. Kampine *Circulatory Physiology, The Essentials*, Williams and Wilkins, Baltimore, 1990.
13. Ugail, H., M. I. G Bloor, and M. J. Wilson, Techniques for interactive design using the PDE method. ACM Transactions on Graphics, **18**(2) (1999), 195–212.
14. Ugail, H., M. I. G Bloor, and M. J. Wilson, Manipulations of PDE surfaces using an interactively sefined parameterisation. Computers and Graphics, **24**(3) (1999), 525–534.
15. Ugail, H ., On the spine of a PDE surface, in *Mathematics of Surfaces X*, M. J. Wilson and R. R. Martin (eds.), Springer, 2003, 366–376.
16. Ugail, H., M. J. Wilson, Efficient shape parametrisation for automatic design optimisation using a partial differential equation formulation, Computers and Structures, **81**(29) (2003), 2601–2609.
17. Weihs, D., The mechanics of rapid starting of slender fish, Biorheology **10** (1973), 343–350.

Hassan Ugail
School of Informatics
University of Bradford, Bradford BD7 1DP, UK
h.ugail@bradford.ac.uk
http://www.inf.brad.ac.uk/research/groups/dve/sbd/sbd.html