The University of Bradford Institutional Repository

Author(s): Ugail, H. and Sourin, A.

Title: Partial differential equations for function based geometry modelling within visual cyberworlds.

Publication year: 2008

Conference title: Cyberworlds. International Conference on Cyberworlds, 2008.

ISBN: 978-0-7695-3381-0

Publisher: IEEE Computer Society.

Original publication is available at *http://www2.computer.org/portal/web/csdl/proceedings/cc#5*

Citation: Ugail, H. and Sourin, A. (2008) Partial differential equations for function based geometry modelling within visual cyberworlds. In: International Conference on Cyberworlds, 2008. IEEE Computer Society. pp.224-231.

Ugail H and Sourin A (2008): "Partial Differential Equations for Function based Geometry Modelling within Visual Cyberworlds", ***Cyberworlds 2008*****, IEEE Computer Society ISBN: 978-0-7695-3381-0, pp. 224-231.**

Partial Differential Equations for Function based Geometry Modelling within Visual Cyberworlds

Hassan Ugail
School of Informatics
University of Bradford
Bradford BD7 1DP, UK
Email: h.ugail@bradford.ac.uk

Alexei Sourin
School of Computer Engineering
Nanyang Technological University
Singapore
Email: assourin@ntu.edu.sg

Abstract

We propose the use of Partial Differential Equations (PDEs) for shape modelling within visual cyberworlds. PDEs, especially those that are elliptic in nature, enable surface modelling to be defined as boundary-value problems. Here we show how the PDE based on the Biharmonic equation subject to suitable boundary conditions can be used for shape modelling within visual cyberworlds. We discuss an analytic solution formulation for the Biharmonic equation which allows us to define a function based geometry whereby the resulting geometry can be visualised efficiently at arbitrary levels of shape resolutions. In particular, we discuss how function based PDE surfaces can be readily integrated within VRML and X3D environments.

1. Introduction

Cyberworlds within the cyberspace can be considered as entities where information modelling can be undertaken through the integration of both spatial and temporal models. There is no doubt that visualisation within cyberworlds is an important aspect which requires special attention. As the gap between the cyberworlds and the real world gets narrower, the demand placed for realism within these shared virtual worlds, both in terms of the complexity of geometric models as well as their appearance, is ever increasing.

3D web visualization of cyberworlds can be achieved through several ways. For a strong server and weak client configuration *image transmissions* can be used. In this case the 3D rendering is performed at the server side and the clients receive only streamed images which they can use for interactive navigation through the scene and communication with other clients. This mode is quite typical of grid-based visualization when either a very powerful server or a cluster of networked computers are used for rendering. Its disadvantage is in a limited resolution of the image and slow update rate due to bandwidth limitations. Another scenario assumes that all the models are preloaded to client computers and only *even transmission* is performed between the clients and an optional server. This mode is quite common for MMORPG games and assumes downloading of an advance rendering engine and a large model database which may have gigabytes of information stored. Though this method provides very photorealistic rendering, the disadvantage is in difficulties which result when a new model is to be introduced into the scene, i.e. it has to be somehow delivered to all the participating clients to update the model databases. A compromise *model transmission* method assumes that the model of the scene is progressively downloaded on to client computers while it is being rendered. The rendering engines can be lightweighted like plug-ins to web browsers (e.g. VRML and X3D plug-ins to MS Internet Explorer and Mozilla Firefox), as well as standalone applications (viewers) with some preloaded textures and models (e.g. SecondLife viewer). This method is considered in our research.

An important aspect of the model transmission based visualisation of cyberworlds involves reduction of the models since large sizes prohibit them to be readily downloaded or transmitted within reasonable time frames. There has been a number of attempts, in the past, to address this problem. Some of the solutions proposed include mesh compression methods [8, 13, 26] and progressive data transmission [7, 11, 21]. Although many methods for geometry compression and progressive data transmission have been proposed, they have not yet come to the required level of maturity for them to be commonly utilised for practical purposes, such as real-time interactions with complex geometry within visual cyberworlds. Therefore, the most promising practical solution as we see it to reduce the amount of transmitted data, i.e. to represent the underlying geometry of objects in an efficient manner.

To this end, function based shape modelling is an avenue which would provide solutions. The basic idea behind function based shape modelling is to minimise the geometric data of a complex object through its geometric representation using mathematical functions. Compared to mesh based models, function based modelling has the advantage that the functions represent the geometry in a compact form and has the ability to display the model at unlimited level of detail.

The idea of function based modelling involves the shape functions f_i which can be defined analytically or procedurally. These functions then define the geometry of the object along with other underlying properties such as colour and texture. Then, the functions describing the shape geometry and its properties are merged to form a single shape function. This process of merging can be performed in various ways depending on the underlying function models used for defining the geometry and other relevant properties.

The bulk of function based modelling techniques involve the use of parametric, implicit and explicit mathematical functions. As far as parametric functions are concerned the main dominant methods are based on Splines [10] such as the use of Coons surfaces, Bézier surfaces and Non-Uniform Rational B-Splines (NURBS). As far as function based implicit modelling is concerned, skeleton based modelling [2, 24] and the use of algebraic methods are notable [1]. Examples of explicit function based modelling include adaptive sampled distance fields (ADFs) [12] and the method of Function Representation (FRep) [20].

It is clear that function based modelling has promising prospects, for defining and reducing the amount of data used to represent geometric models and their underlying properties. One of the bottlenecks, however, faced by these methods are their inherent limitation for efficiently describing the geometry and other relevant properties. For example, in the case of spline based parametric models, the underlying geometry representation requires the storage and transmission of hundreds of control points and weights associated with the polynomial functions. Therefore, in order to address this problem, it is necessary to look for methods that can efficiently represent the geometry in terms of basic mathematical functions and yet be able to describe and parameterise complex geometry.

In this paper we are primarily concerned with the use of parametric PDEs to efficiently describe complex geometry within visual cyberworlds. From a traditional point of view PDEs, especially those that are elliptic in nature, have been mainly utilised in solving engineering related problems such as electromagnetism, stress/strain in physical structures and fluid flows. However, from a geometric design point view, nowadays PDE based techniques are increasingly becoming popular in many applications area. These include, for example, computer simulation of natural phenomena and animation [9], variational fairing [22], image inpainting [3], and Biharmonic polynomial surfaces [19]. For a more detailed discussions of the use PDEs in geometric design the reader is also referred to [6].

The work discussed in this paper centres around the pioneering work of Bloor and Wilson on the so called PDE method [4, 5, 25]. The particular approach used here treats shape modelling as a boundary-value problem. Hence a surface is characterised by defining a number of boundary conditions corresponding to the edges of the surface along with the associated derivative information which define how the surface propagates into its interior. The chosen PDE is usually low order and is elliptic in nature such as the Biharmonic equation.

The paper is organized as follows. Section 2 discusses the general idea of geometric modelling using PDEs whereby the idea of surface modelling using the Biharmonic equations is discussed. An analytic solution formulation for the chosen PDE is also presented. Further discussions on the geometric behaviour surfaces resulting from low order elliptic PDEs are also presented in this section. In Section 3 we discuss how the PDE geometry modelling can be utilised for function based modelling within visual cyberworlds. In particular, we discuss how PDE surfaces can be readily integrated with VRML and X3D environments. In this section we also present a number of examples demonstrating the use of PDE geometry and their potential use to create complex environments efficiently within visual cyberworlds. Finally in Section 4 we conclude the paper and indicate possible future directions of this work.

2 Partial Differential Equations for Geometry Modelling

We assume that a parametric surface patch is defined by a function $\mathbf{X}(u, v)$ such that,

$$\mathbf{X}(u, v) = (x(u, v), y(u, v), z(u, v)),$$

where u and v are parameters defining a finite two-dimensional region Ω which map onto a point in physical space (x, y, z). In order to generate a surface one seeks as a solution to an equation of the type,

$$\sum_{n=0}^{2r} \alpha_n(u, v) \frac{\partial^n}{\partial u^l \partial v^m} \mathbf{X}(u, v) = 0. \qquad (1)$$

Equation (1) represents a linear elliptic PDE of order $2r$, where $l, m, n \geq 0$ and $l + m = n$. Note that in general terms PDEs can be classified as to whether they are elliptic, hyperbolic, parabolic or mixed type depending on the boundary and/or initial conditions that are necessary for a particular problem to be well-posed. In this case the type of PDE given in Equation (1) is elliptic so as the resulting solution gives rise to smooth surfaces and are subject to boundary conditions. The boundary conditions required to solve Equation (1) can be specified as variations of the function and/or its normal derivatives along the edges of the domain over which Equation (1) is solved. Thus, the order of the PDE determines the number of unknown functions that must be specified at the boundary conditions.

In principle, based on Equation (1), one can choose an elliptic PDE of any order to generate a surfaces. For the work described in this paper we utilise the Biharmonic equation. Thus, the generating equation in this case is based on the Laplace equation whereby the PDE is in the form,

$$\nabla^4 \mathbf{X} = \frac{\partial^4 \mathbf{X}}{\partial u^4} + 2\frac{\partial^4 \mathbf{X}}{\partial u^2 \partial v^2} + \frac{\partial^4 \mathbf{X}}{\partial v^4} = 0. \tag{2}$$

2.1 Analytic Solution Method

The solution of the Biharmonic equation given in (2) is a well studied problem and therefore there exist a variety of techniques to solve Equation (2). These include Eigenfunction expansions, integral transforms, Greens functions and numerical techniques such as finite difference and finite element method. The main point of this work is to ensure we have complex geometry at our disposal which can be described using functions that are in analytic form so that they can readily be integrated within function based modelling environments. Therefore, here we seek an analytic form of the solution to Equation (2).

Given that the functions representing the boundary conditions are defined as function $\mathbf{X}$ and normal $\mathbf{X_u}$, we take the (u, v) parameter space Ω to be the region $\{u, v : 0 \leq u \leq 1; 0 \leq v \leq 2\pi\}$. Thus, we assume that all the boundary conditions are periodic in v in the sense $\mathbf{X}(0) = \mathbf{X}(2\pi)$ and $\mathbf{X}_u(0) = \mathbf{X}_u(2\pi)$. We represent these boundary conditions as,

$$\begin{aligned} \mathbf{X}(0, v) &= P_0(v), \\ \mathbf{X}(1, v) &= P_1(v), \\ \mathbf{X}_u(0, v) &= d_0(v), \\ \mathbf{X}_u(1, v) &= d_1(v). \end{aligned} \tag{3}$$

We further assume that all the boundary conditions are continuous functions within the domain of Ω.

With the above assumptions on the boundary conditions we can utilise the method of separation of variables to write the analytic solution of Equation (2) as,

$$\mathbf{X}(u, v) = \mathbf{X}(u)\cos(nv) + \mathbf{X}(u)\sin(nv), \tag{4}$$

where n is a positive integer.

Substituting the terms in Equation (4) into Equation (2) yields a linear homogenous ordinary differential equaiton of the form,

$$\frac{d^4 \mathbf{X}}{du^4} - 2n^2 \frac{d^2 \mathbf{X}}{du^2} + n^4 \mathbf{X} = 0, \tag{5}$$

Using Equation (5) the form of $\mathbf{X}$ subject to appropriate boundary conditions can be given as,

$$\mathbf{X}(u) = c_1 e^{nu} + c_2 u e^{nu} + c_3 e^{-nu} + c_4 u e^{-nu}, \tag{6}$$

where c_1, c_2, c_3 and c_4 are given by,

$$\begin{aligned} c_1 = [(&-P_0(2n^2 e^{2n} + 2ne^{2n} + e^{2n} - 1) \\ &+P_1(ne^{3n} + e^{3n} + ne^n - e^n) \\ &-2d_0 ne^{2n} - d_1(e^{3n} - e^n)]/r, \end{aligned} \tag{7}$$

$$\begin{aligned} c_2 &= [(P_0(2n^2 e^{2n} + ne^{2n} - n) \\ -P_1(ne^{3n} + 2n^2 e^n - ne^n) &- d_0(2ne^{2n} - e^{2n} + 1) \\ &+d_1(e^{3n} - 2ne^n - e^n)]/r, \end{aligned} \tag{8}$$

$$\begin{aligned} c_3 = [(&P_0(e^{4n} - 2n^2 e^{2n} + 2ne^{2n} - e^{2n}) \\ &-P_1(ne^{3n} + e^{3n} + ne^n - e^n) \\ &+2d_0 ne^{2n} + d_1(e^{3n} - e^n)]/r, \end{aligned} \tag{9}$$

$$\begin{aligned} c_4 &= [(P_0(ne^{4n} + 2n^2 e^{2n} + ne^{2n}) \\ -P_1(2n^2 e^{3n} + ne^{3n} - ne^n) &+ d_0(e^{4n} - 2ne^{2n} - e^{2n}) \\ &+d_1(2ne^{3n} - e^{3n} + e^n)]/r, \end{aligned} \tag{10}$$

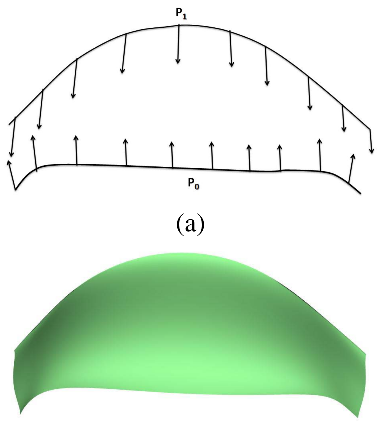
where $r = e^{4n} - 4n^2 e^{2n} - 2e^{2n} + 1$.

Assuming the boundary conditions are continuous functions which are also periodic in v, we can represent each of the boundary conditions in terms of a Fourier series such that,

$$\mathbf{f}(v) = \mathbf{A}_0 + \sum_{n=1}^{\infty} [\mathbf{C}_n(v)\cos(nv) + \mathbf{S}_n(v)\sin(nv)]. \tag{11}$$

This formulation then gives rise to a linear system involving the unknowns c_1, c_2, c_3 and c_4 which can be then determined by solving the system using standard techniques.

It should be stressed here that in order to take advantage of the solution formulation given in (4), it is necessary that the boundary conditions (i.e. the edges of a given surface patch and how these edges propagate into the interior of the surface patch) can be expressed in terms of a finite Fourier series. Although this may seem to be a restriction on the type of geometry that can be generated through this solution scheme a surprisingly large variety of geometry can indeed



(a) (b)

Figure 1. A surface patch generated using the Biharmonic PDE. (a) Illustration of the boundary conditions. (b) The resulting surface in the region $0 \leq v \leq \pi$.

be generated through this scheme as described later in this paper.

As an example consider the surface shown in Fig. 1(b). The boundary conditions to generate this surface are,

$$\begin{aligned} P_0 &= (\cos(\pi v) + \cos(2\pi v), \frac{1}{2}\sin(2\pi v), \frac{4}{5}), \\ P_1 &= (\cos(2\pi v), \sin(2\pi v) + \sin(\pi v), 1 - \alpha), \\ d_0 &= (\cos(2\pi v), \sin(2\pi v), \sin^2(2\pi v)/\alpha), \\ d_1 &= (\cos(2\pi v), \sin(\pi v), -\sin^2(\pi v)/\alpha), \end{aligned} \tag{12}$$

where $\alpha = \sqrt{\cos^2(2\pi v) - 1}$. These boundary conditions are illustrated in Fig. 1(a) where the curves P_0 and P_1 illustrate the boundary conditions defined at $\mathbf{X}(0, v)$ and $\mathbf{X}(1, v)$ respectively where $0 \leq v \leq \pi$. The arrows marked at the boundary curves define the normal boundary conditions where the size and the direction of the arrow illustrates the variation in these boundary at the edges of the surface patch which then determines the internal shape of the surface patch. The above boundary conditions which can be represented as a finite Fourier series is utilised to formulate a linear system involving the unknowns c_1, c_2, c_3 and c_4. The resulting surface is obtained purely in analytic function form which allows us to compute the surface efficiently at any display resolution.

2.2 Geometric Properties of the Biharmonic PDE

As discussed above, the Biharmonic operator can be seen to act as a smoothing operator which enables to produce an interpolating surface for a given set of boundary data. The resulting surface in the above case is provided as an analytic expression and is infinitely differentiable. An important point to highlight here is that since we are treating surface generation as a boundary-value problem the resulting surface is entirely dependent on the boundary conditions and hence the boundary conditions can be utilised as a surface manipulation tool [25].

Common parametric surface generation methods, such as those based on spline techniques, have attractive geometric properties through which the behaviour of the surface subject to changes in the relating control points are somewhat intuitive. For example, in the case of Bézier surfaces, the convex hull property guarantees that the resulting surface is entirely bounded within the convex hull of the control polygon which determine the shape of the surface. In the case of surfaces generated as solutions of PDEs, in particular low order elliptic PDEs, similar geometric properties can be identified. Thus, in the case of PDE surfaces discussed here one can also show that the resulting surface behaves in an intuitive fashion subject to changes in the boundary data.

For instance, for the Laplace equation $\frac{\partial^2 \mathbf{X}}{\partial u^2} + \frac{\partial^2 \mathbf{X}}{\partial v^2} = 0$ one can show (through the min/max principle) that the minimum/maximum of the interpolating function occurs at the boundaries of the surface patch. Whilst this desirable property holds for the Laplace equation, the surfaces generated by it are somewhat limited since there are only two boundary conditions available for the user to define a surface patch. The Biharmonic equation on the other hand is preferable since the user can impose four boundary conditions, two defining the edges of the surface patch and the two defining the rate of change of these edges which determine the interior of the surface patch. Though there is no minimum/maximum principle for the Biharmonic equation, one can still find a priori estimates on the bounds of the interpolating function resulting from the Biharmonic equation. This can be undertaken by applying the minimum/maximum principle for the quantity $\| \nabla \mathbf{X} \|^2 - \mathbf{X}\nabla^2\mathbf{X}$ and applying the maximum modulus theorem [23] so that

$$\| \mathbf{X} \| \leq K(\| \mathbf{X}_0 \| + \| \partial \mathbf{X}/\partial n)_0 \|, \tag{13}$$

where Equation (2) is solved over the region Ω subject to the conditions $\mathbf{X} = \mathbf{X}_0$ and $\partial \mathbf{X}/\partial n = (\partial \mathbf{X}/\partial n)_0$ on the boundary of Ω. Here K is a constant which depends only on the Biharmonic equation and the geometric shape of Ω. Equation (13) indicates that the resulting surface will be of order corresponding to the maximum dimensions of the boundary conditions summed with the maximum rate of change of distance with which the surface moves away from the boundary.

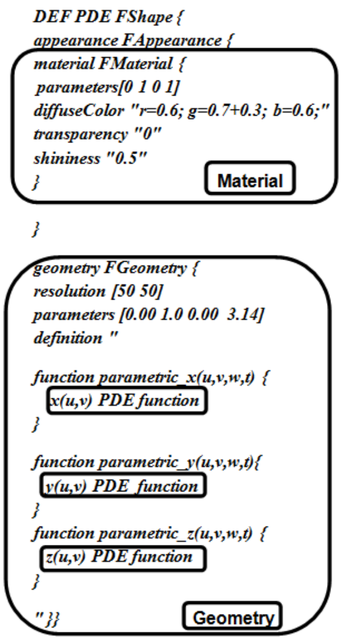


Figure 2. An example of modelling PDE geometry within FVRML and FX3D virtual environment.

Figure 3. Shape of a cup and a vase. In each case we use two surface patches with common boundary conditions.

3 Modelling Framework and Examples

Here we discuss how the PDE geometry can be implemented to enhance the function based modelling within FVRML and FX3D which are extensions of VRML and X3D respectively. The FVRML/FX3D was originally introduced in [14] and subsequent work on this theme have been discussed in [16, 17, 15]. This framework enables modelling of function based shapes and their metamorphosis in visual cyberworlds in an efficient and neat way. It enables to define complex mathematical functions through combinations of basic functions. Hence, individual expressions corresponding to complex shapes, their appearances can be directly defined within the source code through script like mathematical language.

The scripting is based on a subset of JavaScript for both individual formulas and function scripts. In addition to existing built-in simple mathematical fucntions (such as *exp(x)*, *log(x)*, *sin(x)*, *cos(x)* etc) additional flow control operators such as *for loops*, *while loops* and *if-else* operations are also implemented.

Further to the standard VRML/X3D nodes, the implementation of FVRML/FX3D contains ten additional nodes, which are *FShape*, *FGeometry*, *FAppearance*, *FMaterial*, *FTexture3D*, *FPhysics*, *FDensity*, *FFriction*, *FForce* and *FTransform*. These nodes can be used together as well as with the standard VRML and X3D nodes. The *FShape* node is a container for the *FGeometry* or any standard geometry node, and the *FAppearance* or the standard *Appearance* node. These nodes define the geometry and the appearance of the shape, respectively, as illustrated in Fig.2. *FShape* may be called from the *FTransform* node or from the standard *Transform* node. The *FGeometry* node is designed to define a geometry using implicit, explicit or parametric functions defined straight in the code or in the external library. The *FAppearance* node may contain the *FMaterial* or the standard *Material* nodes, as well as the standard color *Texture* and the *FTexture3D* nodes. In the *FMaterial* node, the components of the illumination model are defined with mathematical functions. The *FTexture3D* node contains displacement functions for the geometry defined in the *FGeometry* node. The *FPhysics* node is used for defining physical properties associated with the shape's geometry. It contains the *FDensity*, *FFriction* and *FForse* nodes. With reference to the topic of the paper, parametric functions in FVRML and FX3D define surfaces, solid objects, force vector coordinates and colors as x, y, z Cartesian coordinates and r, g, b values of colors. Parametric functions can be functions of parameters u, v and w and the time t.

The implementation of FVRML and FX3D is done as plug-ins for blaxxun Contact (http://www.blaxxun.com) and Bitmanagement BS Contact VRML/X3D (http://www.bitmanagement.com) browsers which enable to build visual cyberworlds using VRML or X3D and their function-based extensions FVRML/FX3D.

3.1 Examples

In order to demonstrate the capability of the PDE surfaces within the FVRML/FX3D framework, here we show a series of examples whereby we show how it is possible to model function based complex geometry.

Fig 3 show the geometry of a cup like shape and a vase like shape. Each of these two shapes are generated using two PDE surface patches with common boundary condi-

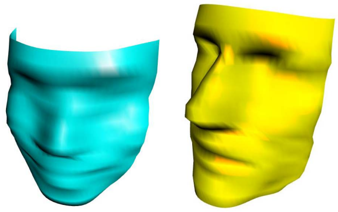


Figure 4. Shape of faces. In each case we use eight surface patches with common boundary conditions.

tions. As usual, the boundary conditions are defined in terms of a finite Fourier series which are utilised to generate the function definition of the surfaces through the PDE. In each case where the surface patches meet each other at the common boundaries, the derivative boundary conditions were chosen to ensure that the surfaces are blended together with tangent continuity. Thus, the geometry definition for the blended two surface patches for each case of the two examples are defined as,

```
geometry FGeometry {
⋮
parameters [0.00001 2.0 0.0001 6.28]
⋮
function parametric_x(u,v,w,t){
if(u<1) return (x component of surface patch 1)
if(u>1) {
u = u-1.0;
return (x component of surface patch 2)
}
}
⋮
}
```

In the next example shown in Fig. 4 we show how complex geometry such as the shape of human faces can be generated via function based geometry through the use PDE surfaces.

Similar to the previous examples, the geometry of the faces is generated by means of utilising a series of connected PDE surface defined through appropriate boundary conditions with common boundary conditions where the surfaces meet each other. In this particular case we have utilised 3D human facial scan data to extract a series of profile curves through which we define the necessary boundary conditions. Thus, from a given human facial 3D data scan (which provides us with a dense point cloud) we automatically extracted an ordered set of points by means of defining a series of horizontal planes through which the facial data intersect. For this purpose we assume the facial data is normalized and appropriately aligned within a Cartesian coordinate system. Note, there exist several techniques which enable the automatic pre-processing of data for this purpose. For example the use of Principle Component Analysis (PCA) algorithms are common in normalizing and aligning facial data. For more details the interested reader is referred, for example, to [18].

Once an appropriate number of horizontal profile data is available, Fourier analysis is then performed on the discrete data sets in order to determine the boundary conditions suitable for generating the analytic functions describing the face through the PDE equation. In this particualar case of the two faces shown in Fig. 4 we have utilised a series of nine horizontal planes which define the position boundary condition for eight surfaces patches. Again the derivative boundary conditions for each blending surface patch is defined so as to ensure there is tangent plane continuity between the patches. In each case it was found that a total of five Fourier modes are enough to adequately represent each face. Note in this case the analytic solution domain is valid for the region $0 \leq v \leq \pi$.

An important point that needs to be highlighted here is that the original scan data for each face is over 1MB whilst the PDE version of the face within FVRML file requires a mere 45kb, thus showing a substantial reduction in the size of the data utilised.

In the next example we show how a series of airplane shapes can be efficiently generated and stored using the PDE formulation for use within a visual cyberworld environment. Fig. 5 shows the shape of a delta airplane generated using six surface patches with common boundary conditions where appropriate. In this case we have generated the fuselage shape using a single surface patch to which the wing surface patches are blended. In a similar fashion the surface patches corresponding to the tail part of the airplane is generated.

Fig. 6 shows the shape of a fighter airplane generated using six surface patches with common boundary conditions where appropriate. The fuselage shape in this case is defined using two surface patches which are blended. The wing surface patches are blended to the rear surface patch corresponding to the fuselage. In addition, four separate surfaces are then generated which corresponding to the four engine shapes shown.

Fig. 7 shows the shape of a B17 airplane generated using eight surface patches with common boundary conditions where appropriate. The fuselage shape in this case is generated using a single surface patch to which the wing surface patches are blended. In a similar fashion, surface patches corresponding to the tail part of the airplane are generated. In addition, two separate surfaces are then generated corresponding to the two engine shapes shown. Note in this

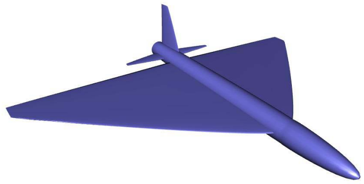


Figure 5. Shape of a delta airplane surface generated using six surface patches with common boundary conditions where appropriate.

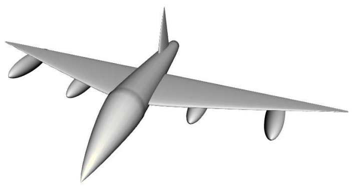


Figure 6. Shape of a fighter airplane surface generated using six surface patches with common boundary conditions where appropriate

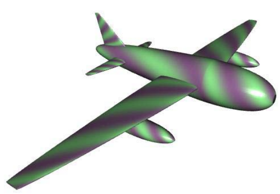


Figure 7. Shape of a B17 airplane surface generated using six surface patches with common boundary conditions where appropriate.

case that we have utilised a simple material function within FVRML in order to enhance the appearance of the B17.

Since, compared to the polygon-based VRML and X3D models, the function-based PDE models are small in size, it is possible to perform their rapid exchange across the Internet for making collaborative interactive modifications with concurrent synchronous visualization at each client computer with any required level of detail. It is also possible to use the function-based PDE models together with large data sets from 3D scanners. In that case only the modifications to the models can be exchanged while the original data are kept on the client computers or shared on a web server.

As a final example we show how function based animation can be carried out. Fig. 8 shows a morphing sequence between the two faces shown in Fig. 4. In this case the morphing is undertaken using a time dependent function of the form $\mathbf{X}(u, v, t) = \mathbf{X}_1(u, v) + t(\mathbf{X}_2(u, v) - \mathbf{X}_1(u, v))$ where $\mathbf{X}_1(u, v)$ and $\mathbf{X}_2(u, v)$ are the PDE functions defining the faces and t is the time parameter where $0 \leq t \leq 1$.

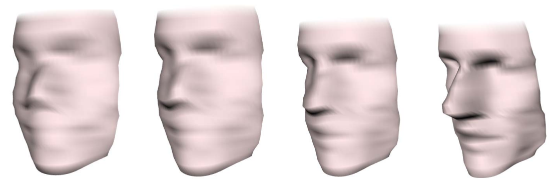


Figure 8. Function based animation using morphing between two faces.

4 Conclusion

The discussion of this paper centres around the use of PDEs for geometry modelling for function based modelling within visual cyberworlds. The proposed solution enables surface modelling to be defined as a boundary-value problem. In this paper we have presented an analytic solution formulation for the chosen PDE, i.e. the Biharmonic equation which allows us to define a function based surface enabling the geometry to be visualised efficiently for arbitrary surface resolutions. Details of how the PDE geometry modelling can be utilised for function based modelling within FVRML and FX3D has been presented.

We can think of several extensions to this work which can be undertaken in the future. First this paper clearly demonstrates that the concept of the PDE based shape modelling is a viable solution for use within visual cyberworlds where fast solutions and real time data exchange of complex geometry is required. In this work we have only demonstrated static objects which are defined as surfaces. The work can be extended for generation and parameterisation of solid objects. It can also be extended to generate time dependent geometry enabling complex animations within cyberworlds. This work can be also extended to study other various properties associated with the geometry, such as texture defined within function based modelling framework where functions arising as solutions to PDE equations can be applied. All these extensions will require further studies on efficient analytic solution methods in order to be compatible with the existing functions based modelling and visualisation frameworks.

References

[1] C. Bajaj and I. Ihm, Algebraic Surface Design with Hermite Interpolation, *ACM Transactions on Graphics*, *11*(1), 61-91, 1992.

[2] J. F. Blinn, A Generalization of Algebraic Surface Drawing, *ACM Transactions on Graphics*, *1*(3),235-258, 1982.

[3] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, Image Inpainting. In *SIGGRAPH 2000*, 417-424, 2000.

[4] M.I.G. Bloor and M.J. Wilson. Using Partial Differential Equations to Generate Freeform Surfaces, *Computer Aided Design*, *22*(4),202-211, (1990).

[5] M.I.G. Bloor and M.J. Wilson. Functionality in Solids Obtained from Partial Differential Equations, *Computing*, *8*, 21-42, (1993).

[6] Castro G, H Ugail, P. Willis and I. Palmer, A Survey of Partial Differential Equations in Geometric Design, *The Visual Computer*, *24*(3), 213-225, 2008.

[7] B. Y. Chen, T. Nishita, Multiresolution Streaming Mesh with Shape Preserving and QoS-like Controlling, *Proceedings of the Seventh International Conference on 3D Web Technology*, 35-42, 2002.

[8] M. Deering, Geometry compression, in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 13-21, 1995.

[9] D.S. Ebert, F.K. Musgrave, P. Prusinkiewicz, J. Stam, and J. Tessendorf, Simulating Nature: From Theory to Practice, *SIGGRAPH 2000 Course Notes 25*, 2000.

[10] G. Farin J. Hoschek M.-S. Kim, *Handbook of Computer Aided Geometric Design*, North Holland, 2002.

[11] E. Fogel, D. Cohen-Or R. Ironi, T. Zvi, A web Architecture for Progressive Delivery of 3D Content, *Proceedings of Sixth International Conference on 3D Web Technology* , 35-41, 2001.

[12] S. F. Frisken, R. N. Perry, A. P. Rockwood, T. R. Jones, Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics, in *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques* , 249-254, 2000.

[13] B. Kronrod and C Gotsman, Efficient Coding of Non-Triangular Mesh Connectivity, in *Proceedings of Pacific Conference on Graphics and Applications*, 235-242, 2000.

[14] F. M. Lai, and A Sourin, Function-Defined Shape Node for VRML, in *Eurographics 2002, Short Presentations*, 207215, 2002.

[15] K. Levinski and A. Sourin, Interactive function-based shape modelling, *Computers & Graphics 31*(1), 66-76, (2007).

[16] Q. Liu and A. Sourin, Function-Based Representation of Complex Geometry Appearance, in *Proceedings of ACM Web3D '05*, 123134, 2005.

[17] Q. Liu and A. Sourin, Function-Defined Shape Metamorphoses in Visual Cyberworlds, *The Visual Computer*, *22*, 977-990, 2006.

[18] S. Malassiotis and M.G.Strintzis, Robust Real-time 3D Head Pose Estimation from Range Data, *Pattern Recognition*, *38*(8), 1153-1165, 2005.

[19] J. Monterde and H. Ugail, A General 4th-Order PDE Method to Generate Bézier Surfaces from the Boundary, *Computer Aided Geometric Design*, *23*(2), 208-225, 2006.

[20] A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko, "Function representation in geometric modeling: concepts, implementation and applications", *The Visual Computer*, *11*(8), 429-446, 1995.

[21] S. Rusinkiewicz and M. Levoy, Streaming QSplat: A Viewer for Networked Visualisation of Large, Dense Models, *Proceedings of 2001 Symposium on Interactive 3D Graphics*, 63-68, 2001.

[22] R. Schneider and L. Kobbelt, Geometric Fairing of Irregular Meshes for Free-Form Surface Design, *Computer Aided Geometric Design*, 18(4), 359-379, 2001.

[23] R. Sperb, *Maximum Priciples and their Applications*, Academic Press, 1981.

[24] D. W. Storti, G. M. Turkiyyah, M. A. Ganter, C. T. Lim and D. M. Stal, Skeleton-Based Modeling Operations on Solids, in *Proceedings of the fourth ACM symposium on Solid modeling and applications*, 141 - 154, 1997.

[25] H. Ugail, M.I.G. Bloor, and M.J. Wilson, Techniques for Interactive Design Using the PDE Method, *ACM Transactions on Graphics*, *18*(2), 195-212, (1999).

[26] X. Wu and D. Li, Triangle mesh compression and simplification, *Machine Graphics & Vision*, *11*(4),475-488, 2000.