

# 3D image acquisition and NURBS based geometry modelling of natural objects

G. Moustakides<sup>a</sup>, D. Briassoulis<sup>b,\*</sup>, E. Psarakis<sup>a</sup>, E. Dimas<sup>b</sup>

<sup>a</sup>Computer Technology Institute, Patras, Greece

<sup>b</sup>Department of Agricultural Engineering, Agricultural University of Athens, 75 Iera Odos, 118 55, Athens, Greece

Received 19 October 1998; accepted 17 July 1999

## Abstract

This paper presents a recently developed Image Acquisition–Geometric Modelling system for the construction of geometric models of natural or artificial objects. This construction is achieved in a two-step procedure. First, the object of interest is digitised, i.e. points on the surface of the object are computed. Then, based on the cloud of digitised 3D points, an approximating surface, a geometric model of the object of interest, is constructed. Coupled with a mesh generation package, the system presented allows for further processing and analysis of the resulting model instead of the real object itself. © 2000 Civil-Comp Ltd. and Elsevier Science Ltd. All rights reserved.

*Keywords:* Image acquisition; Geometry modelling; Non-uniform rational B-splines

## 1. Introduction

Recent developments in the fields of Computer Aided Design (CAD) and Finite Element (FE) technology and the emergence of modern, more powerful computers, have made possible the substitution of real models of products with computer models. This evolution has many advantages in cutting down the cost and the time for a complete design–analysis–creation cycle of a variety of products in a wide range of fields and applications.

The construction of finite element models for the purpose of analysis is in itself a laborious task especially for biological objects (e.g. fruits, human body parts etc.) which often have a more complicated geometry than mechanical components. From this point of view the major bottleneck in the complete modelling–analysis cycle is the definition of the geometry of the object under consideration (i.e. modelling part of the cycle). However, emerging technology in image processing techniques allows for the reconstruction of scanned objects (in the form of 3D points on the surface of the object) from image data (photographs, video recordings, CT, NMR etc.). These points could be automatically processed and fitted so that a geometric model of the scanned object is produced. This model could then be supplied to a finite element analysis system for further processing and analysis or be otherwise utilised.

The aim of this paper is to present an integrated computer

*image-acquisition/modelling* system which allows for the automatic creation of accurate geometric models from scanned images of objects. Such a system coupled with a finite element mesh generator and analysis package can be used in applications in many fields of science and humanities such as agricultural, civil and structural engineering, archaeology, and bioengineering.

The advantages in terms of cost efficiency and reduction of the duration of the design/analysis cycle of products are enormous. This is especially true in cases where the objects do exist but any attempt of modelling them accurately by conventional means would be inefficient or even impossible (e.g. biological objects etc.).

An image acquisition system has been developed for the creation of 3D geometric data. The corresponding “3D object reconstruction problem” has been met with the development of a technique for the generation of normal images. The principal idea is to replace the images in two planes by images in one plane, by using the fact that any perspective projection is a projective projection. Then, as the key to a stereo system is a method for determining which point in one image corresponds to a given point in the other image, the problem of image matching had to be solved. Using the image matching model, the parameters of the mapping functions of the model had to be determined. The differential matching method was used assuming that approximate values of the parameters are known and replacing the non-linear problem by a linear one. Then, the values of the desired parameters result from the minimisation of energy

\* Corresponding author.

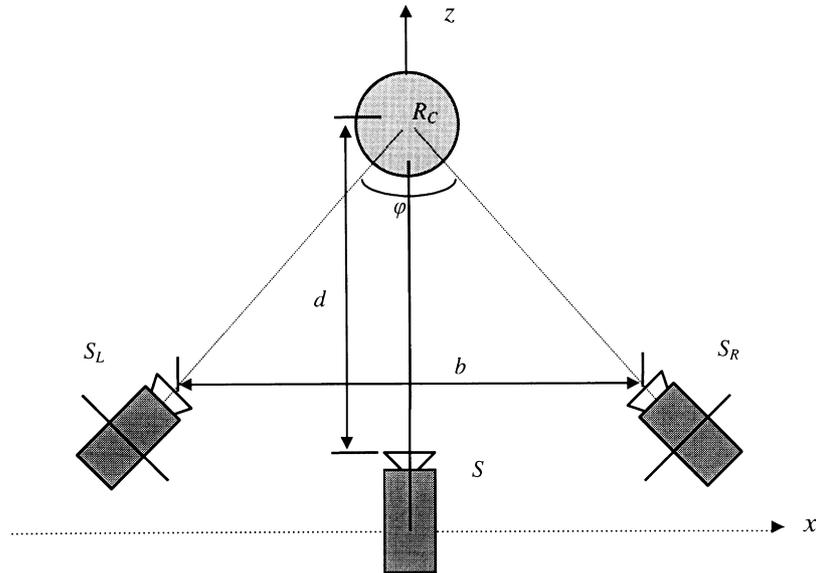


Fig. 1. Top view of the existing, single camera ( $S$ ) acquisition system and the equivalent model composed of two cameras ( $S_L$ ,  $S_R$ ).

of the observation noises with respect to the parameters of the problem.

The output of the image acquisition stage is a cloud of 3D points on the surface of the object of interest. These points are then converted to a mathematically expressed geometric model (i.e. an equation) of the object suitable for further processing and analysis. The underlying mathematical representation of curves and surfaces implemented by the geometric modelling software is the well-known Non-Uniform Rational B-Splines (NURBS) representation. NURBS curves and surfaces theory and applications have been thoroughly investigated and reported in the literature [13–16]. They have become a de facto industry standard mainly because they can represent both free-form shapes and commonly used analytical shapes such as conic curves. Currently, there are two techniques implemented for the construction of NURBS surfaces. The first builds a surface taking into account all the data points that can be approximated within a given tolerance or interpolated if there are sufficient degrees of freedom available. This method is somewhat slow (but sufficiently accurate) because of the usually large amount of data. The second technique organises the data points into cross-sectional data which are then interpolated or approximated within a given tolerance (thus creating cross-section curves). The cross-section curves are then “skinned” so as to produce a surface model. The latter technique is faster than the former because of the reduced amount of points that are fitted, but this can sometimes lead to less accurate models. However, it is well suited for cross-sectional data, for example tomography data.

In Section 2 of this paper the image acquisition technique developed is described. In Section 3 the NURBS fitting techniques developed are described and problems identified during the experimental stage of the system are presented. Illustrative examples are presented in Section 4.

## 2. Image acquisition

A stereoscopic approach, like the one described herein, is characterised by the following two steps:

*Step 1.* Image acquisition.

*Step 2.* Stereo matching.

These two steps play an important role in the design of a stereo system, but the success of the approach greatly depends on its ability to solve the stereo matching or correspondence problem. Most of the existing stereoscopic systems consist of either one optical sensor, which can be moved so that its relative positions at different times are known, or two optical sensors always maintaining the same known position with respect to each other.

A top view of the image acquisition system that we are proposing is illustrated in Fig. 1. As we can see the system is based on a single camera combined with a turning disk whose centre  $R_c$  is placed at a distance  $d$  from the optical centre of the camera  $S$ .

In a typical experiment the object is placed on the disk and turned at various angles to obtain different sideviews. For each sideview two snapshots differing by a small angle  $\varphi$  are taken. Each pair is used to compute the 3D coordinates of the corresponding sideview.

Notice now that the acquisition system shown in Fig. 1 is equivalent to a stereo model composed by two optical sensors, denoted by  $S_L$  and  $S_R$  in Fig. 1, whose *baseline* (distance between the optical centres of sensors) is  $b$ .

### 2.1. Generating normal images

It is well known that the stereo matching (also known as correspondence problem) heavily depends on the stereo camera modelling and it can be significantly simplified if

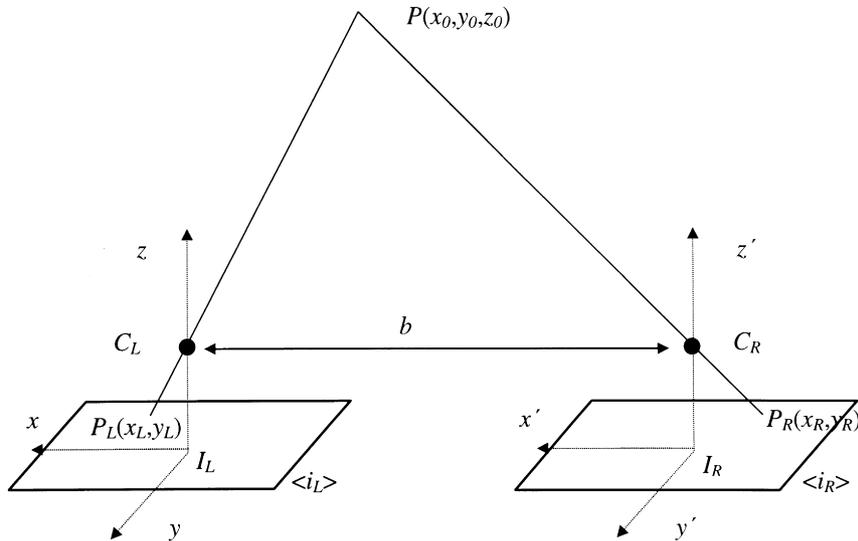


Fig. 2. The lateral stereo camera model.

we use the lateral model [13] which is one of the simplest imaging models. The stereo camera arrangement of such a model is presented in Fig. 2. Notice that the optical centres  $C_L$  and  $C_R$  of the two cameras are separated only by a translation  $b$  in the  $x$ -direction and that their optical axes are parallel. A consequence of this last property is that *epipolar* lines are parallel to the *baseline* and therefore any scene point is projected onto the two image planes at points having the same  $y$ -coordinate.

Let us denote the image planes of  $C_L$  and  $C_R$  by  $\langle i_L \rangle$  and  $\langle i_R \rangle$ , respectively, and by  $I_L, I_R$  the centres of the two images. Let us also assume that  $I_L$  is the origin of the  $(x, y, z)$  world coordinate system. Then, if  $P_L(x_L, y_L)$  and  $P_R(x_R, y_R)$  are the projections of the scene point  $P(x_o, y_o, z_o)$  onto the image

planes  $\langle i_L \rangle$  and  $\langle i_R \rangle$ , respectively, by using simple geometry we can easily relate the world coordinates to the image coordinates as follows:

$$x_o = \frac{2d \sin(\varphi/2)x_L}{x_L - x_R} \tag{1}$$

$$y_o = \frac{2d \sin(\varphi/2)y_L}{x_L - x_R} \tag{2}$$

$$z_o = \frac{2d \sin(\varphi/2)f}{x_L - x_R} \tag{3}$$

where  $d$  and  $f$  can be obtained through a camera calibration process.

The most difficult task in applying Eqs. (1–3) is in fact the determination of points in the two images that correspond to the same scene point (matching or correspondence problem). As it was also stated above, this problem can be significantly simplified if the two images are normal corresponding to parallel optical axes. Consequently next we are going to present the necessary equations that can transform the two images taken by the proposed system to two normal images. In other words we will assume that we have a pair of images (left and right) taken by rotating the disk by a small angle  $\varphi$  and we are going to see how this pair can be transformed into a normal pair. As the common projection plane for the two normal images we are going to consider the plane that passes through the optical centres of the left and right sensors.

Let us consider the transformation of the left image (similarly we can find the corresponding transformation for the right image). Let  $x_{O_L}z$  and  $x'O'_Lz'$  be the coordinate systems of the original and the normal image. Also let  $O_LA$  and  $O'_LA'$  be the  $x$ -coordinates of the projections of the scene point  $P$  onto these two image planes as shown in Fig. 3. Then, from the orthogonal triangles  $O_LC_LA$  and  $O'_LC_LA'$  by using

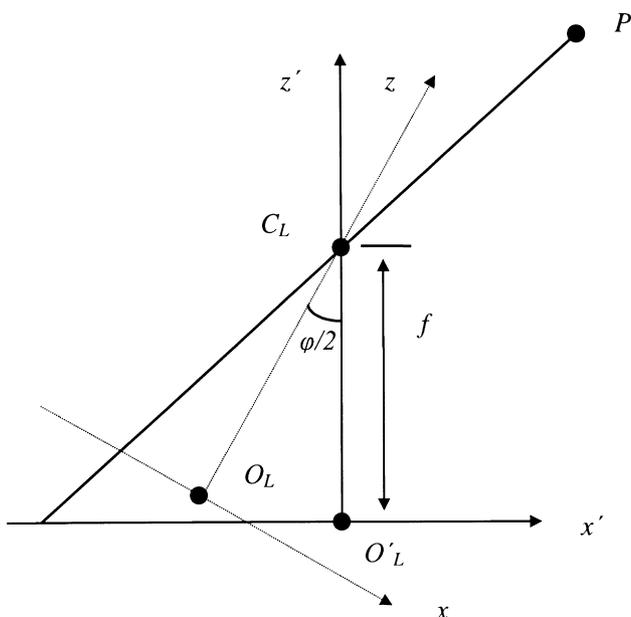


Fig. 3. Geometry for the computation of the normal image.

simple geometry we obtain the following relation for the  $x'_L$ , coordinate for the left normal image:

$$x'_L = f \frac{x_L \cos(\phi/2) - f \sin(\phi/2)}{f \cos(\phi/2) + x_L \sin(\phi/2)} \quad (4)$$

where we recall that  $f$  is the *focal length* of the camera and  $(x_L, y_L)$  are the coordinates (in pixels) of the projection of the scene point  $P$ .

Following similar steps, we can prove that the  $y'_L$  coordinate of the left normal image as well as  $(x'_R, y'_R)$  coordinates of the right normal image will be given by the following relations:

$$y'_L = \frac{y_L f}{f \cos(\phi/2) + x_L \sin(\phi/2)} \quad (5)$$

$$x'_R = f \frac{f \sin(\phi/2) + x_R \cos(\phi/2)}{f \cos(\phi/2) - x_R \sin(\phi/2)} \quad (6)$$

$$y'_R = \frac{y_R f}{f \cos(\phi/2) - x_R \sin(\phi/2)} \quad (7)$$

Notice that in general the coordinates of the original images are taken from a uniform sampling. Unfortunately the corresponding  $x', y'$  coordinates of the normal images do not enjoy this property. Therefore some type of interpolation is needed in order to compensate for this drawback.

### 2.2. The Image matching problem

The image matching problem, between a pair of images, is characterised by the following steps:

*Step 1.* A feature of interest is selected in one image.

*Step 2.* The same feature is identified (usually through processing) in the second image.

*Step 3.* The disparity between the two image features is recorded and can be used for the computation of 3D information.

Stereo systems can be broadly classified into two categories [14]. The first includes techniques that match sparse and irregularly distributed features, as edges and contours, whereas the second includes matching of dense features, such as grey levels. As far as the second category is concerned, which is the one that is of interest to us, correspondence is typically established using a cross-correlation like measure. The most well-known technique in this category, is the Differential Matching Technique (DMT) [15].

With the DMT we attempt to compensate intensity differences, appearing in the image pair, using combinations of geometric and radiometric transformations. More precisely a geometric transformation is used to describe the geometric relation between corresponding points in the two images, whereas intensity changes, due to the different viewing directions, are compensated with a radiometric transformation. Although both transformations are linear their combination produces a non-linear transformation. This in turn

requires the solution of a non-linear optimisation problem for the estimation of the necessary transformation parameters. If the variation of the radiometric parameters is small and we have an a priori knowledge of their nominal values then the non-linear problem can be easily reduced to a linear one. For such a case the resulting optimisation is well defined and easily solved through least squares. Under the above assumptions the performance of the DMT is known to be satisfactory [16]. However for cases where the variation of the radiometric parameters is significant or the a priori knowledge of their nominal values is not available the method behaves poorly [14].

In our system we have alleviated this drawback by a proper modification of the classical DMT. Key characteristic of the proposed implementation is the fact that we were able to reduce the original non-linear optimisation problem to a linear one without the need of any form of linearisation or any a priori knowledge of the nominal parameters. The estimates of the desired geometric and radiometric parameters are, as in the classical method, obtained through the solution of a well-defined least squares minimisation problem and turn out to be reliable even for large variations of the radiometric parameters. A detailed description of the implemented matching algorithm can be found in Ref. [17].

### 3. Geometric model construction

The 3D points on the surface of the object of interest are the input to a geometric modelling system. The underlying mathematical representation for curves and surfaces of this system is the well-known NURBS representation. Parametric representations and especially NURBS curves and surfaces theory and applications have been thoroughly investigated and reported in the literature [1–4,8,9,18]. They have become a de facto industry standard mainly because they can represent both free-form shapes and commonly used analytical shapes such as conic curves and they offer a common mathematical form for the widely used parametric curves and surfaces such as Bézier and rational Bézier curves and surfaces. Therefore, the NURBS mathematical representation was chosen to be implemented in the current system.

Two fitting techniques are employed in the geometric modelling part of the system presented herein:

- A 3D point surface approximation/interpolation technique which creates a surface fitted to 3D data points (described in Section 3.2).
- Cross-sectional design technique (“skinning”) which creates a surface from cross-sectional curves given in NURBS form (described in Section 3.3). Alternatively, NURBS cross-sectional curves can be created from 3D point data by a 3D point curve approximation/interpolation technique similar to the surface fitting technique mentioned above and be subsequently skinned in order to produce a NURBS surface.

### 3.1. NURBS curve/surface definition

A NURBS curve  $C(s)$  of degree  $p$  is a parametric piecewise polynomial curve of degree  $p$ , defined by a set of control points  $\mathbf{P}_i = [x_i, y_i, z_i]^T$ ,  $i = 1, \dots, n$ , a set of weights  $w_i$ ,  $i = 1, \dots, n$ , a non-decreasing sequence of real numbers  $u_i$   $i = 0, \dots, n + p$  which is called knot vector (and is in effect a partition of the parameter domain) and a set of B-Spline basis functions  $N_i^p(s)$  defined recursively by:

$$N_i^p(s) = \frac{s - u_i}{u_{i+p} - u_i} N_i^{p-1}(s) + \frac{u_{i+p+1} - s}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(s) \quad (8a)$$

$$N_i^0(s) = \begin{cases} 1, & \text{if } u_i \leq s \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (8b)$$

The curve itself is defined by the following formula:

$$C(s) = \frac{\sum_{i=0}^n w_i P_i N_i^p(s)}{\sum_{i=0}^n w_i N_i^p(s)} \quad (9)$$

where  $w_i$  are the weights,  $\mathbf{P}_i = [x_i, y_i, z_i]^T$  the control points and  $N_i^p(s)$  are the B-Spline basis functions defined over the knot vector  $\{u_0, u_1, \dots, u_{n+p}\}$ . The parameter domain is  $[u_{p+1}, u_n]$ . The curve can also be written in a matrix form

$$C(s) = \frac{\mathbf{b}^T(s) \cdot \mathbf{P}}{\mathbf{b}^T(s) \cdot \mathbf{w}} = \frac{\mathbf{b}^T(s) \cdot [\mathbf{P}_x \ \mathbf{P}_y \ \mathbf{P}_z]}{\mathbf{b}^T(s) \cdot \mathbf{w}} \quad (10)$$

where  $\mathbf{b}^T(s) = [b_i(s)] = [N_0^p(s), N_1^p(s), \dots, N_n^p(s)]$ ,  $\mathbf{w} = [w_0, w_1, \dots, w_n]^T$  and  $\mathbf{P}$  is a matrix defined as

$$\mathbf{P} = [\mathbf{P}_x \ \mathbf{P}_y \ \mathbf{P}_z] = \begin{bmatrix} w_0 x_0 & w_0 y_0 & w_0 z_0 \\ w_1 x_1 & w_1 y_1 & w_1 z_1 \\ \vdots & \vdots & \vdots \\ w_n x_n & w_n y_n & w_n z_n \end{bmatrix} \quad (11)$$

Similarly, a NURBS surface is defined by

$$S(s, t) = \frac{\sum_{i=0}^k \sum_{j=0}^n w_{ij} \mathbf{P}_{ij} N_i^p(s) N_j^q(t)}{\sum_{i=0}^k \sum_{j=0}^n w_{ij} N_i^p(s) N_j^q(t)} \quad (12)$$

over a grid of control points, weights and a knot vector for each parametric direction. In a matrix form

$$S(s, t) = \frac{\mathbf{b}^T(s, t) \cdot \mathbf{P}}{\mathbf{b}^T(s, t) \cdot \mathbf{w}} = \frac{\mathbf{b}^T(s, t) \cdot [\mathbf{P}_x \ \mathbf{P}_y \ \mathbf{P}_z]}{\mathbf{b}^T(s, t) \cdot \mathbf{w}} \quad (13)$$

where

$$\begin{aligned} \mathbf{b}^T(s, t) &= [b_i(s, t)] \\ &= [N_0^p(s)N_0^q(t), \dots, N_0^p(s)N_n^q(t), N_1^p(s)N_0^q(t), \dots, \\ &\quad N_1^p(s)N_n^q(t), \dots, N_n^p(s)N_0^q(t), \dots, N_n^p(s)N_n^q(t)] \end{aligned}$$

$$\mathbf{w} = [w_{00}, w_{01}, \dots, w_{0n}, w_{10}, \dots, w_{1n}, \dots, w_{k0}, \dots, w_{kn}]^T$$

and  $\mathbf{P}$  is a matrix defined by

$$\mathbf{P} = [\mathbf{P}_x \ \mathbf{P}_y \ \mathbf{P}_z] = \begin{bmatrix} w_{00}x_{00} & w_{00}y_{00} & w_{00}z_{00} \\ \vdots & \vdots & \vdots \\ w_{0n}x_{0n} & w_{0n}y_{0n} & w_{0n}z_{0n} \\ w_{10}x_{10} & w_{10}y_{10} & w_{10}z_{10} \\ \vdots & \vdots & \vdots \\ w_{1n}x_{1n} & w_{1n}y_{1n} & w_{1n}z_{1n} \\ \vdots & \vdots & \vdots \\ w_{k0}x_{k0} & w_{k0}y_{k0} & w_{k0}z_{k0} \\ \vdots & \vdots & \vdots \\ w_{kn}x_{kn} & w_{kn}y_{kn} & w_{kn}z_{kn} \end{bmatrix} \quad (14)$$

where  $x_{ij}, y_{ij}, z_{ij}$  are the coordinates of the control points.

### 3.2. Curve/surface fitting to 3D points

The first method employed in the current system for solving the fitting problem is a technique which fits a curve/surface to a cloud of 3D data points. The curve/surface fitting problem to be solved here can be stated as follows: Given a set of data points  $\mathbf{Q}_i$ ,  $i = 1, 2, \dots, m$  (obtained from the image acquisition system described earlier), find a NURBS curve or surface that fits the data according to some specified criterion. In general, computing the desired curve amounts to computing:

- the degree  $p$  of the basis functions, the number of control points and weights  $n$ ;
- the location parameter values  $s_i$  corresponding to data points  $\mathbf{Q}_i$ ,  $i = 1, \dots, m$ ;
- the knot vector  $\mathbf{U}$ ;
- the control points  $\mathbf{P}_i$ , and weights  $w_i$ ,  $i = 1, \dots, n$ .

For surface fitting the following have to be computed:

- The degrees  $p, q$  of the basis functions and the number of control points and weights  $n$  and  $k$  in the  $s$  and  $t$  parametric directions, respectively. These are usually user specified.
- The location parameters  $(s_i, t_i)$   $i = 1, \dots, m$  corresponding to the given data points.

- Suitable knot vectors  $\mathbf{U}$ ,  $\mathbf{V}$  in parametric direction  $s$ ,  $t$  taking into account the parameter distribution.
- The control points  $\mathbf{P}_{ij}$ , and weights  $w_{ij}$ ,  $i = 1, \dots, k$ ,  $j = 1, \dots, n$ .

Solving for all the above-mentioned unknowns at once is a highly complicated problem. In practice, the degree of the basis functions and the number of control points and weights is fixed beforehand (refer to Section 3.2.1). The location parameters and the knot vector are also determined beforehand according to the distribution of the data points (this procedure is called data parameterisation, see Sections 3.2.1 and 3.3.1). Therefore, the fitting procedure results to the minimisation of an appropriate distance criterion with respect to the unknown weights and control points.

3.2.1. Data parameterisation for curve/surface fitting

In the fitting algorithm it is required that the degree of the curve, the number of control points and weights, the location parameter and the knot vectors are specified or are algorithmically inferred from the data points beforehand. The procedure of computing these parameters is usually called data parameterisation. Data parameterisation is usually a three-step procedure:

1. The degree  $p$  of the basis functions in the  $s$  direction (and degree  $q$  in the  $t$  direction for surfaces) and the number of control points and weights  $n$  (and  $k$  in the other parametric direction for surfaces) are determined. These are usually user specified. The user is allowed to specify any degree of the basis functions and number of control points as long as the number of knots is made compatible afterwards. Ideally, there is an optimum degree for a specified number of control points or an optimum number of control points for a specified degree. However, what one means optimum in this case is not easy to define and most of the time depends on the particular application; for example, if speed is more important than accuracy choosing low degree basis functions with a low number of weights and control points is more appropriate. Furthermore, there is always a trade off between the degree and number of control points and computational cost: the higher the degree or the number of control points the more costly the method. However, keeping the balance between speed and accuracy by choosing the degree of the basis functions and the number of control points is a rather intuitive process. In most of the applications run on the current system a good balance between accuracy and speed is achieved by using cubic basis functions.
2. The location parameters  $s_i$ ,  $i = 1, \dots, m$  corresponding to data points  $((s_i, t_i)$  for surfaces) are computed.
3. A suitable knot vector  $\mathbf{U}$ , ( $\mathbf{U}$ ,  $\mathbf{V}$  for surface fitting) in parametric direction  $s$ , (and  $t$  for surface fitting) is determined taking into account the parameter distribution.

3.2.1.1. Computation of location parameters. There are several methods used in the current system in order to compute location parameters from the data points.

- Uniform:

$$s_i = \frac{i - 1}{m - 1} \quad i = 1, \dots, m \tag{15}$$

for curve fitting of  $m$  data points, and

$$s_i = \frac{i - 1}{m_s - 1} \quad i = 1, \dots, m_s \tag{16}$$

$$t_j = \frac{j - 1}{m_t - 1} \quad j = 1, \dots, m_t$$

for surface fitting of a grid of  $m_s m_t$  data points. This method assigns equidistant parameter values to the data points, hence it is used only in ideal situations where the data points are nearly equally distributed in space.

- Cumulative chord length:

$$s_i = s_{i-1} + \frac{\|\mathbf{Q}_i - \mathbf{Q}_{i-1}\|}{\sum_{j=2}^m \|\mathbf{Q}_j - \mathbf{Q}_{j-1}\|} \quad i = 2, \dots, m, \quad s_1 = 0. \tag{17}$$

in the case of curve fitting or

$$s_{ij} = \frac{\sum_{k=2}^i \|\mathbf{Q}_{kj} - \mathbf{Q}_{k-1,j}\|}{\sum_{k=2}^{m_s} \|\mathbf{Q}_{kj} - \mathbf{Q}_{k-1,j}\|} \tag{18}$$

$$i = 2, \dots, m_s, \quad j = 1, \dots, m_t \quad s_{1j} = 0$$

$$t_{ij} = \frac{\sum_{k=2}^j \|\mathbf{Q}_{ik} - \mathbf{Q}_{i,k-1}\|}{\sum_{k=2}^{m_t} \|\mathbf{Q}_{ik} - \mathbf{Q}_{i,k-1}\|}$$

$$i = 1, \dots, m_s, \quad j = 2, \dots, m_t \quad t_{i1} = 0$$

for surface fitting of an  $m_s m_t$  grid of data points. This method assigns a parameter value to a data point according to the length of all the line segments formed between successive points, starting from the first and ending to the current point. It is obvious from the definition of the method that some sort of ordering of the data points already exists and predefines the curve topology. This means that as the parameter value of the curve sweeps the parameter interval, say  $[0,1]$ , a point on the curve is “traveling” along the data points in the order they are specified, i.e. from the first to the last. (However, the points computed during the image acquisition phase are not necessarily ordered so as to produce the correct object topology.

- Centripetal method for curve fitting:

$$s_i = s_{i-1} + \frac{\|Q_i - Q_{i-1}\|^{1/2}}{\sum_{j=2}^m \|Q_j - Q_{j-1}\|^{1/2}} \quad i = 2, \dots, m, \quad s_1 = 0. \quad (19)$$

This method observes the changing curvature of the underlying curve.

- Base curve/surface parametrisation: this method is due to Ma [5]. Each data point is associated with a point on a simple known underlying curve or surface (for example by minimising the distance of the data point from the selected primitive curve or surface with respect to the parameter values). The parameter values of the associated point on the underlying curve or surface (base curve/surface) are the location parameter values for the data point. This method can be applied recursively starting with a simple curve/surface, improving its shape and repeating the parameter computation until a satisfactory result is obtained. For more on the curve/surface parametrisation method refer to Ref. [5].

All of the above-mentioned methods are available in the current system and, theoretically, they could all be used as alternatives (with different results in terms of accuracy). However, in all of the above methods except the base curve/surface parametrisation method the data points are assumed to be chain or grid distributed. In the last method (i.e. base/surface parametrisation), however, this is not required and therefore it can be used to order the data points in case they are randomly distributed which is often the case for measured points obtained by a scanning device.

At this point, the problems encountered in the location parameter computation stage of the algorithm should be mentioned. Owing to the fact that the data points come from multiple views of the object, there are very often cases where there is an overlap between points coming from different views. Therefore, there is no ordering in the data points. Ordering the data points is, in fact, a procedure directly connected to the topology of the object under construction. It is better to be tackled in the image acquisition stage of the algorithm, as it is not really a geometric procedure. It is also essential for computing the knot vectors. However, in the current system, the lack of ordering in the data points is overcome by adopting and using the base surface parametrisation method described above. The points can be ordered by ordering the  $s$  location parameters first and the  $t$  location parameters afterwards. Experience shows that in the case of relatively simple objects such as fruits, a simple starting base surface such as a sphere gives sufficiently accurate results. However, for more complicated objects, for example objects which include holes or branches, topological information (i.e. connectivity of the data points) of the object of interest should be included in the set of input data to the geometric modelling software.

3.2.1.2. *Knot distribution.* A suitable knot vector  $\mathbf{U}$  (or suitable knot vectors  $\mathbf{U}$ ,  $\mathbf{V}$  for surface fitting) in parametric direction  $s$ , (and  $t$  for surface fitting) is determined by taking into account the parameter distribution. Commonly used methods for knot values allocation are [5,12,18].

- Simple knots setting:

$$u_i = \begin{cases} 0 & i = 0, \dots, p \\ \frac{i-p}{n-p+1} & i = p+1, \dots, n \\ 1 & i = n+1, n+p \end{cases} \quad (20)$$

for curve fitting and

$$u_i = \begin{cases} 0 & i = 0, \dots, p \\ \frac{i-p}{n-p+1} & i = p+1, \dots, n \\ 1 & i = n+1, n+p \end{cases} \quad (21a)$$

$$v_j = \begin{cases} 0 & j = 0, \dots, q \\ \frac{j-q}{k-q+1} & j = q+1, \dots, k \\ 1 & j = k+1, \dots, k+q \end{cases} \quad (21b)$$

for surface fitting.

- Averaging methods: in averaging methods the knot distribution is varying according to the distribution of the data points by taking into account the distribution of the parameter values in the parameter domain. For more on averaging methods see Ref. [5].

In the system presented here two averaging methods are used because, although they are more expensive computationally, they lead to more accurate models. The first one is a simple averaging method used for curve fitting [12] and the other one is applicable both in curve and surface fitting [5]. They both produce a curve or surface which is, in general, more accurate than a curve or surface with knots computed by a conventional method because more basis functions (and therefore more control points) are assigned to areas where the distribution of the data points is denser.

### 3.2.2. Formulation of the curve fitting problem

In the approach adopted by the system developed and presented here, the fitting criterion is the minimisation of

a distance function based on the  $l^2$  norm:

$$\sum_{l=1}^m \|C(s_l) - \mathbf{Q}_l\|_2^2 = \sum_{l=1}^m \left\| \frac{\sum_{i=0}^n w_i \mathbf{P}_i N_i^p(s_l)}{\sum_{i=0}^n w_i N_i^p(s_l)} - \mathbf{Q}_l \right\|_2^2$$

$$= \sum_{l=1}^m \left\| \frac{\mathbf{b}^T(s_l) \cdot [\mathbf{P}_x \quad \mathbf{P}_y \quad \mathbf{P}_z]}{\mathbf{b}^T(s_l) \cdot \mathbf{w}} - \mathbf{Q}_l \right\|_2^2 \quad (22)$$

with respect to the unknown control points and weights, where  $s_l, l = 1, \dots, m$  are location parameters of the data points.

### 3.2.3. Formulation of the surface fitting problem

Similarly to the curve fitting problem the surface fitting problem can be stated as follows: Given a set of data points  $\mathbf{Q}_l, l = 1, \dots, m$  find the control points  $\mathbf{P}_{ij}$  and weights  $w_{ij}$  of the NURBS surface so that some approximation criterion is minimised. Again, the quantity to be minimised is based on the  $l^2$  norm

$$\sum_{l=1}^m \|S(s_l, t_l) - \mathbf{Q}_l\|_2^2 = \sum_{l=1}^m \left\| \frac{\sum_{i=0}^n \sum_{j=0}^k w_{ij} \mathbf{P}_{ij} N_i^p(s_l) N_j^q(t_l)}{\sum_{i=0}^n \sum_{j=0}^k w_{ij} N_i^p(s_l) N_j^q(t_l)} - \mathbf{Q}_l \right\|_2^2$$

$$= \sum_{l=1}^m \left\| \frac{\mathbf{b}^T(s_l, t_l) \cdot [\mathbf{P}_x \quad \mathbf{P}_y \quad \mathbf{P}_z]}{\mathbf{b}^T(s_l, t_l) \cdot \mathbf{w}} - \mathbf{Q}_l \right\|_2^2 \quad (23)$$

with respect to the unknown control points and weights, where  $(s_l, t_l), l = 1, \dots, m$  are location parameters of the data points. Again, it is assumed that the degrees of the surface and the location parameters and knot vectors have been determined beforehand.

### 3.2.4. Fitting algorithm

A typical outline of the fitting algorithm adopted is (described in Ref. [5]).

- Input: 2D or 3D data points.
- Assign initial parameter values to the data points.
- Assume an initial knots distribution.
- Go through an interpolation or approximation procedure to obtain the weights and control points of the NURBS curve or surface, i.e. obtain the fitting curve or surface.
- Optimise the parameter distribution if necessary to obtain a better fit [7].
- Output: NURBS surface.

The first two steps of the algorithm were described in Section 3.2.1. Parameter optimisation schemes are not implemented in the current system. This is because in most of the modelling applications the model constructed by the fitting technique described in this paper is sufficiently

accurate. Therefore, it is not necessary to further load the modelling procedure with time consuming processes without a significant gain. In the rest of this section the minimisation of Eq. (22) or (23) with respect to the unknown control points and weights is briefly described.

The minimisation of the distance problem defined by Eq. (22) or (23) for curve or surface fitting, respectively, is equivalent to solving, in a least square sense, the linear systems

$$\mathbf{N}_{m \times r} \cdot \mathbf{P}_x - \mathbf{Q}_x \cdot \mathbf{N}_{m \times r} \cdot \mathbf{w} = 0 \quad \mathbf{N}_{m \times r} \cdot \mathbf{P}_y - \mathbf{Q}_y \cdot \mathbf{N}_{m \times r} \cdot \mathbf{w} = 0$$

$$\mathbf{N}_{m \times r} \cdot \mathbf{P}_z - \mathbf{Q}_z \cdot \mathbf{N}_{m \times r} \cdot \mathbf{w} = 0 \quad (24)$$

where  $r = n + 1$  for curve fitting or  $r = (k + 1)(n + 1)$  for surface fitting,  $\mathbf{N} = [b_i(s_j)] \quad i = 1, \dots, r, \quad j = 1, \dots, m$  for curve fitting or  $\mathbf{N} = [n_{ij}] = [b_i(s_j, t_j)] \quad i = 0, \dots, r, \quad j = 1, \dots, m$  for surface fitting, and

$$\mathbf{Q}_x = \text{diag}(X_1 \quad \dots \quad X_m), \quad \mathbf{Q}_y = \text{diag}(Y_1 \quad \dots \quad Y_m),$$

$$\mathbf{Q}_z = \text{diag}(Z_1 \quad \dots \quad Z_m)$$

are diagonal matrices which contain the coordinates of the data points (computed in the image acquisition stage). Eq. (24) can be rearranged in matrix form

$$\begin{bmatrix} \mathbf{N} & 0 & 0 & -\mathbf{Q}_x \cdot \mathbf{N} \\ 0 & \mathbf{N} & 0 & -\mathbf{Q}_y \cdot \mathbf{N} \\ 0 & 0 & \mathbf{N} & -\mathbf{Q}_z \cdot \mathbf{N} \end{bmatrix}_{3m \times 4(k+1)(n+1)} \begin{bmatrix} \mathbf{P}_x \\ \mathbf{P}_y \\ \mathbf{P}_z \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (25)$$

In solving the homogeneous linear system (25) in a least square sense, special care is needed for the computation of weights because they need to be positive. In Ref. [5] Ma shows how to compute the weights and control points in a two step linear fashion by separating the weights and control points in a linear system of the form

$$\mathbf{M}_{m \times (k+1)(n+1)} \cdot \mathbf{w} = 0 \quad (26)$$

where  $\mathbf{M}$  is a symmetric non-negative matrix of dimension  $(k + 1)(n + 1)$ . The unknown weights can be found by minimising the Rayleigh quotient of matrix  $\mathbf{M}$  under suitable constraints for the weights. Interpolating solutions are found if  $\text{rank}(\mathbf{M}) < (k + 1)(n + 1)$ .

### 3.3. Cross-sectional design

The second method employed in the current system for solving the fitting problem is based on a cross-sectional design technique which fits a NURBS surface to a sequence of NURBS curves. For more on cross-sectional design see Refs. [10,11]. In the case where the cross-sections are not given, but only the 3D data points are available, this technique can still be used by filtering the data points in such a way as to create cross-section data. This means that those points which are within a given tolerance from user

specified planes are classified as belonging to the same cross-section curve. The rest of the points are discarded. Each set of cross-section data points is then fitted with a NURBS curve by the algorithm described in the previous section. The NURBS curves are then “skinned” using the algorithm presented in Section 3.3.3. However, before the “skinning” of the curves certain parameters have to be computed before hand (data parametrisation). The data parametrisation procedure is described in the next section.

### 3.3.1. Data parametrisation for cross-sectional design

In the case of cross-sectional design the following have to be determined before the “skinning” procedure:

1. Degree of the cross-sectional curves (it must be the same for every curve; it is the same as the degree of the surface under construction in the direction of the cross-sections) and the degree of the surface in the other parametric direction, as well as the number of weights and control points in each parametric direction, i.e.:
  - Degree  $p$  of the surface in the direction of the cross-sectional curves (say in the  $s$ -direction).
  - Degree  $q$  of the surface in the other (“longitudinal”) parametric direction.
  - Number of weights and control points for each cross-sectional curve.
  - Number of weights and control points in the “longitudinal” direction.

The degree and number of control points for the cross-section curves are given when the input are the curves themselves. If the input is data points they are user specified along with the degree and number of control points and weights of the longitudinal direction. The same rules and restrictions mentioned in Section 3.2.1 for point fitting apply here as well.
2. Parameter values corresponding to data points for each cross-sectional curve. These values can be computed by any of the methods described for curve fitting. This step is applicable if the cross-section curves are given in terms of data points on the curves instead of the curves themselves.
3. Knot vectors for each cross-sectional curve. These can also be computed by any of the methods described for curve or surface fitting. This step is applicable if the cross-section curves are given in terms of data points on the curves instead of the curves themselves.
4. Location parameters for the control points of the cross-section curves in the longitudinal direction. These can be computed by parametrising (by any of the methods for curve fitting) the control points of the given cross-section curves in the longitudinal direction.
5. Knot vector for the control points of the cross-section curves in the longitudinal direction. These can be computed by any of the methods described for

curve fitting once the parameter values have been determined.

### 3.3.2. Formulation of the fitting problem

In the cross-sectional design a surface has to be constructed from a collection of given curves. The cross-sectional design problem can be stated as follows: A collection of NURBS curves

$$C^l(s) = \frac{\sum_{i=0}^k w_i^l P_i^l N_i^p(s)}{\sum_{i=0}^k w_i^l N_i^p(s)} \quad l = 1, \dots, n \quad (27)$$

is given. From this collection of curves a NURBS surface (defined by Eq. (13)) that interpolates them has to be constructed. Therefore, the surface under construction needs to interpolate the given curves at certain parameter values, i.e. it should have the following interpolation property

$$S(s, t_l) = C^l(s) \quad l = 1, \dots, n \quad (28)$$

It should be emphasised that the cross-section curves defined by Eq. (27) are compatible. This means that they have the same degree, the same number of weights and control points and are defined over the same knot vector. In case the input cross-section curves are not compatible, degree raising and knot insertion algorithms like the Boehm or the Oslo algorithm [6] can be employed to force compatibility, without changing the curves’ geometry.

### 3.3.3. Fitting algorithm

The fitting algorithm adopted in the system presented here is as follows.

- Input: cross-sectional curves or 3D data points. In the case that the data is in point form, organise them in sets of cross-sectional data and fit a NURBS curve to each set by the algorithm presented for curve fitting.
- Make all curves compatible (if they are not already).
- Go through a skinning technique like the one presented below.
- Output: NURBS surface.

The problem of curve fitting from 2D or 3D data points was investigated in the previous section. Therefore, in the following it is assumed that the cross-section curves are known or they have been already computed. Thus the skinning algorithm used in the geometric modeller is presented in the following section.

### 3.3.4. Skinning method

It is known that a NURBS surface  $S(s, t)$  of degrees  $p, q$  at a fixed parameter value  $t = a$  and  $s$  ranging in  $[0, 1]$ , is a

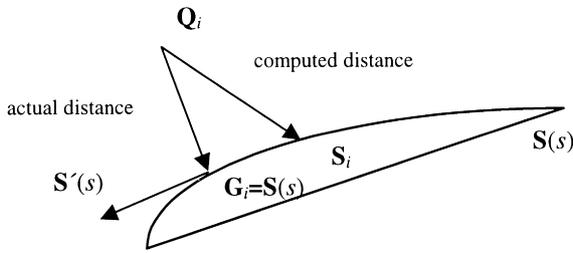


Fig. 4. Error computation techniques.

NURBS curve of degree  $p$  with weights [18]

$$w_i = \sum_{j=0}^n w_{ij} N_j^q(a) \tag{29}$$

and control points

$$\mathbf{P}_i = \sum_{j=0}^n \frac{w_{ij} N_j^q(a)}{w_i} \mathbf{P}_{ij} \tag{30}$$

The cross-section curve defined by Eq. (27) should be identical to the curve defined by the control points and weights defined by Eqs. (29) and (30) at parameter value  $t = t_i$ . Therefore,

$$w_i = w_i^l \text{ and } \mathbf{P}_i = \mathbf{P}_i^l, \quad i = 0, \dots, k \tag{31}$$

This leads to the following system of linear equations with respect to the unknown weights  $w_{ij}$  and control points  $\mathbf{P}_{ij}$  of the desired surface

$$\sum_{j=0}^n N_j^q(t_l) w_{ij} = w_i^l, \quad i = 0, \dots, k, \quad l = 0, \dots, n \tag{32}$$

$$\sum_{j=0}^n w_{ij} N_j^q(t_l) \mathbf{P}_{ij} = w_i^l \mathbf{P}_i^l, \quad i = 0, \dots, k, \quad l = 0, \dots, n \tag{33}$$

Solving Eqs. (32) and (33) guarantees interpolation of the section curves by the NURBS surface. The solution is computed in a two-step fashion; first solving Eq. (32) for the unknown weights and then solving Eq. (33) for the unknown control points.

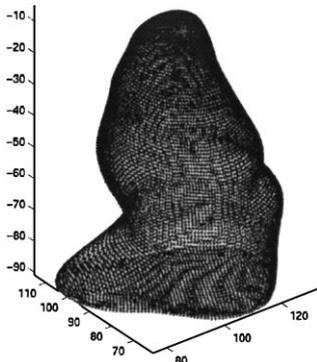


Fig. 5. Data points on the surface of the left ventricle of the heart.

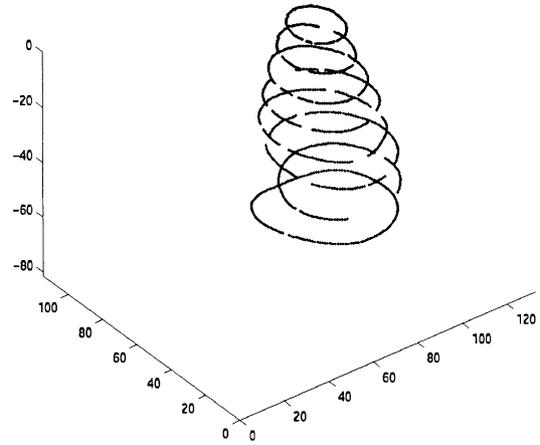


Fig. 6. Crosssectional data of the left ventricle of the heart.

#### 4. Error computation

In this section error computation techniques are presented.

The simplest way to compute the deviation of the measured points from the computed surface is to compute the distance  $d_i$  of each measured point  $\mathbf{Q}_i$  (which is associated with a parameter pair  $(s_i, t_i)$ ) from the point  $\mathbf{S}_i = \mathbf{S}(s_i, t_i)$  on the computed surface (or  $\mathbf{S}_i = \mathbf{S}(s_i)$  for curves). The total error can then be computed as

$$E_t = \sum_{i=1}^n d_i = \sum_{i=1}^n \|\mathbf{Q}_i - \mathbf{S}_i\|_2 \tag{34}$$

and the mean error, for  $n$  data points, can then be expressed as

$$E_m = \frac{1}{n} E_t = \frac{1}{n} \sum_{i=1}^n d_i = \frac{1}{n} \sum_{i=1}^n \|\mathbf{Q}_i - \mathbf{S}_i\|_2 \tag{35}$$

However, as Fig. 4 illustrates, the computed distance which determines the error measurement does not necessarily correspond to the actual distance of the measured point  $\mathbf{Q}_i$  from the computed curve or surface.

A more accurate approach would be to compute the perpendicular distance of the measured point  $\mathbf{Q}_i$  to the curve or surface, i.e the distance of  $\mathbf{Q}_i$  to the point  $\mathbf{G}_i$  on the curve  $\mathbf{S}(s)$ . For this purpose, the parameter value  $s$  corresponding to the point  $\mathbf{G}_i$  has to be computed. Since the vectors  $\mathbf{Q}_i - \mathbf{S}_i$  is orthogonal to the derivative vector  $\mathbf{S}'(s)$  of the curve, the inner product of the two vectors is 0, hence the following condition for the point  $\mathbf{G}_i = \mathbf{S}(s)$  is obtained:

$$(\mathbf{Q}_i - \mathbf{S}(s)) \cdot \mathbf{S}'(s) = 0 \tag{36}$$

Hence, solving Eq. (36) for  $s$  gives the location of the point  $\mathbf{G}_i$ . The total and mean errors can then be computed by Eqs. (34) and (35) substituting  $\mathbf{G}_i$  for  $\mathbf{S}_i$ .

Note that solving Eq. (36) is equivalent to the following

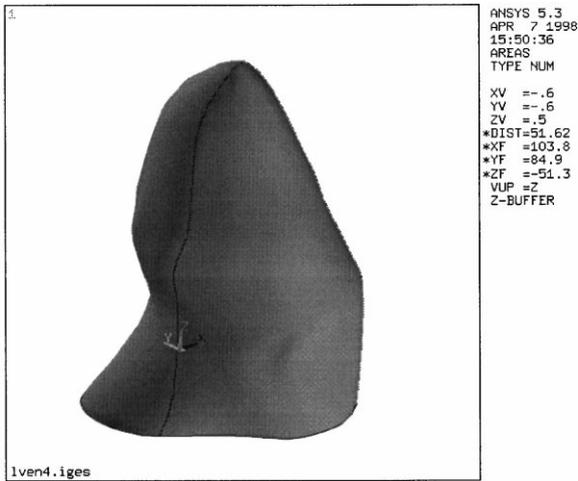


Fig. 7. Surface model of the left ventricle of the heart.

minimisation problem:

$$\min \|Q_i - S(s)\|_2^2 \quad (37)$$

with respect to the unknown parameter  $s$ .

A useful error measurement is the relative error. For closed curves it can be formulated as

$$E_r = \sum_{i=1}^n \frac{\|Q_i - S(s)\|_2}{R_i} \quad (38)$$

where  $R_i$  is the distance of the point  $G_i$  from the barycentre of the curve. The mean relative error is then

$$e_r = \frac{1}{n} E_r = \frac{1}{n} \sum_{i=1}^n \frac{\|Q_i - S(s)\|_2}{R_i} \quad (39)$$

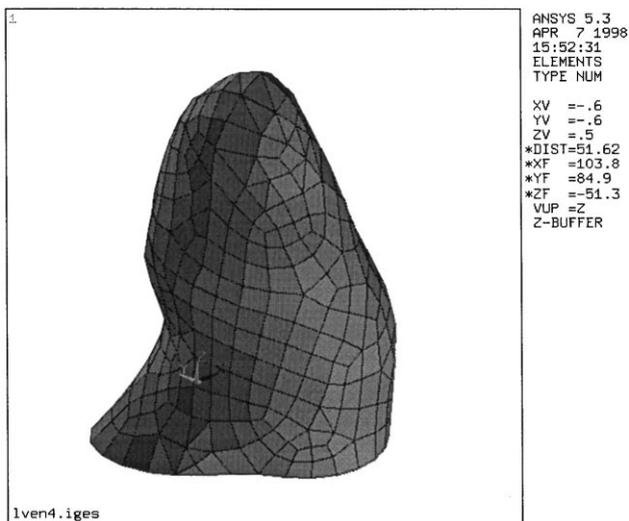


Fig. 8. Finite element mesh of the left ventricle.

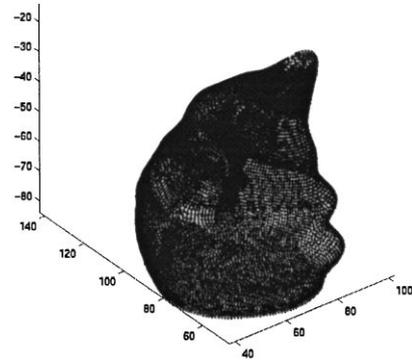


Fig. 9. Data points on the surface of the right ventricle of the heart.

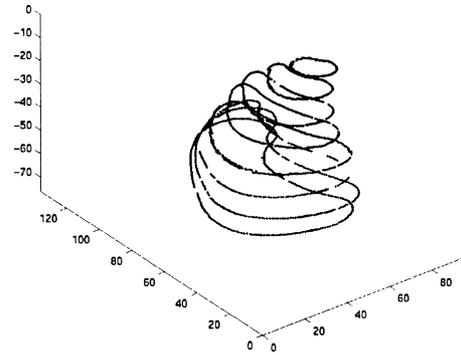


Fig. 10. Cross-sectional data of the right ventricle of the heart.

### 5. Illustrative examples

In this section several illustrative examples are presented to demonstrate the techniques employed in the image acquisition/geometric modelling system. The first experiment involves modelling a human heart and in particular three components of it, the right and left ventricles and the myocardium. Each component was modelled separately and then they were all pieced together to form a single

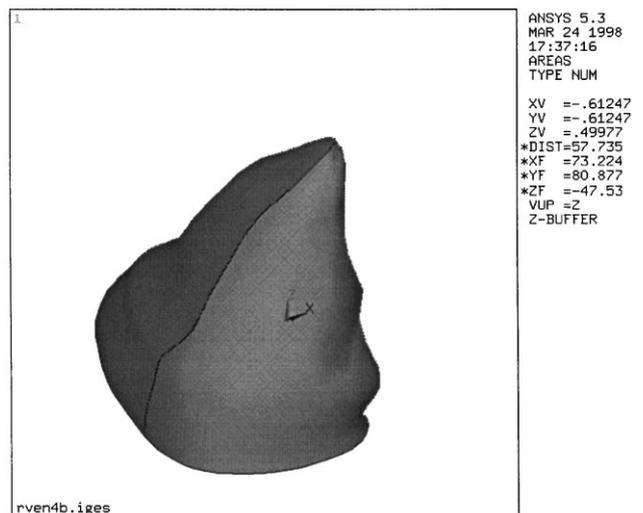


Fig. 11. Surface model of the right ventricle of the heart.

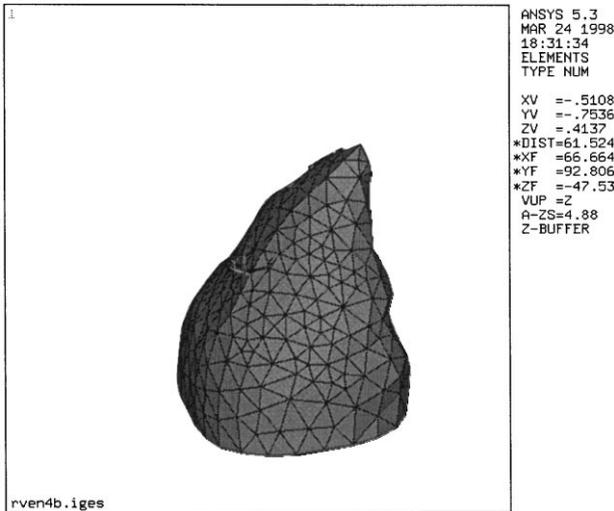


Fig. 12. Finite element mesh of the right ventricle.

model of the heart. The data points provided by the University of Ilmenau (Germany), were obtained from tomography data ([19,20]) and were fitted by using the cross-sectional design technique described in Section 3.3. The data points of the left and right ventricles and the myocardium are shown in Figs. 5, 9 and 13, respectively.

- First, the raw 3D data points were filtered according to their distance from user specified planes. This not only organises the raw 3D data into cross-sectional data but also significantly reduces the amount of input to the geometric modelling software, thus significantly increasing the speed of the whole modelling procedure. All points within some tolerance from a particular plane were classified as belonging to the same cross-section curve. The so-produced cross-sectional data points are shown in Figs. 6, 10 and 14, respectively.
- The points on each curve were parametrised by a base curve parametrisation method (using a circle as a base curve) and the knot vectors were computed and made compatible.
- The cross-sectional data were subsequently fitted with cubic curves, with 20 control points each, using the

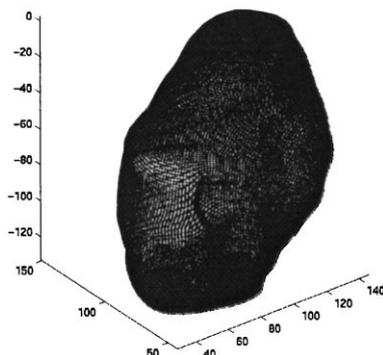


Fig. 13. Data points on the surface of the myocardium of a human heart.

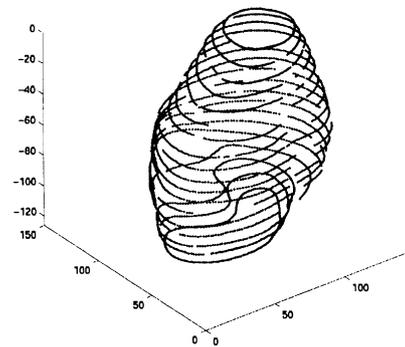


Fig. 14. Cross-sectional data of the myocardium.

curve fitting technique described in Section 3.2. This means that the weights  $w_i^j$  and control points  $P_i^j$  of Eqs. (32) and (33) were computed.

- Then, the weights ( $w_{ij}$ ) of the resulting surface were computed using Eq. (32).
- The control points ( $P_{ij}$ ) of the resulting NURBS surface were found using Eq. (33).

The resulting surfaces are shown in Figs. 7, 11 and 15. One may observe that the models follow smoothly the corresponding X, Y, Z data, allowing for a direct input to a finite element geometric mesh generator. Problems encountered during the geometric modelling procedure mainly concern:

- (a) the appropriate selection of the control points to avoid possible overlapping of the individual components of a composite object such as the model of a complete heart. Such problems may be due to inaccuracies in the image acquisition or the modelling stage of the system, in areas where two or more individual components are very close to each other;
- (b) to enable the creation of a smooth surface with no localised disturbances. Such problems may be due to an

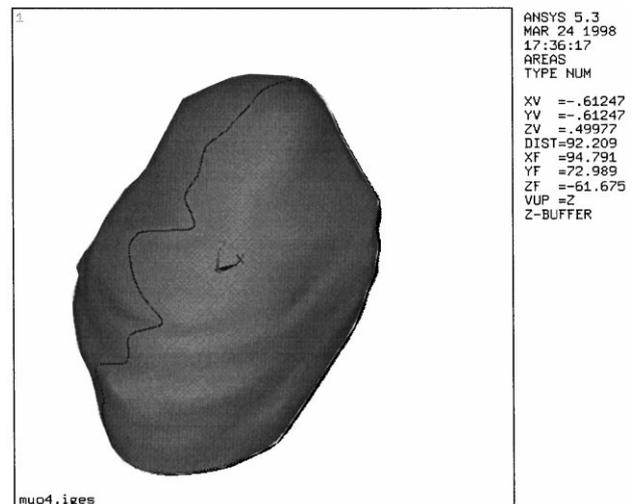


Fig. 15. Surface model of the myocardium of Fig. 9.

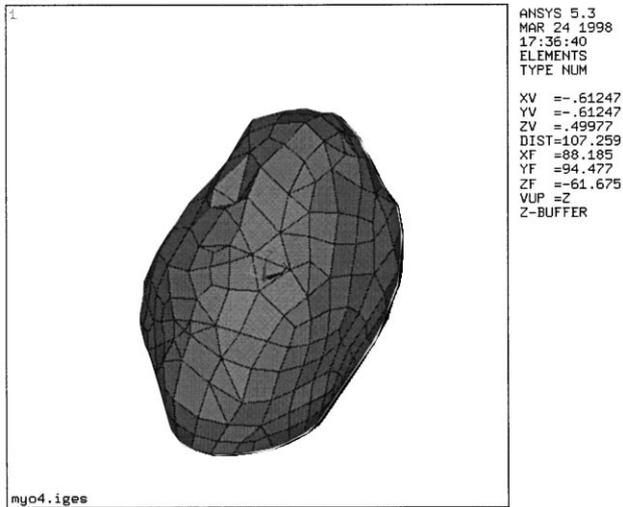


Fig. 16. Finite element mesh of the myocardium.

uneven distribution of the 3D data points computed in the image acquisition stage of the algorithm.

After the geometric models were created, it was possible to use them as input to a commercial F.E. mesh generator by using IGES interface with no need for manual manipulation at the stage of mesh generation. Thus, importing the smooth model to the ANSYS mesh generator, the F.E. models shown in Figs. 8, 12 and 16 were obtained.

Error estimation for the three examples mentioned above is given in the following table:

	Mean error (mm)	Mean relative error (%)
Left ventricle	2.5	1.22
Right ventricle	2.8	1.42
Myocard	1.9	0.90

The technique presented here is also suitable for other applications as well, for example in the field of agricultural engineering. Thus, the same technique was applied in order to model a pear. Geometric modelling of pears was used in the framework of non-destructive testing concerning the quality assessment of such fruits by employing system iden-

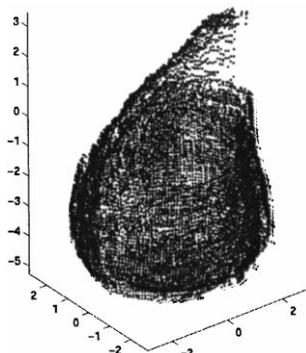


Fig. 17. Data points on the surface of a pear.

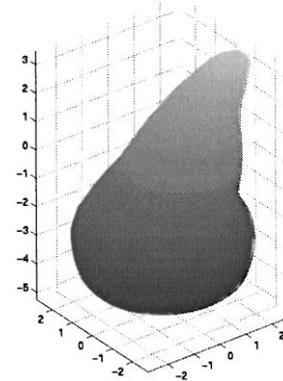


Fig. 18. NURBS surface of the pear.

tification techniques [2,21]. Figs. 17–19 show the 3D data obtained by the image acquisition technique described in Section 2, the resulting surface of the pear and its finite element model, respectively.

A major difficulty in order to produce a smooth surface of the desired object is related to the distribution of the 3D data point along the surface of object. If the data points resulting from the image acquisition phase are not uniformly distributed the parameterisation of the corresponding cross-section curves varies a lot between the curves. This may produce a rather twisted geometric surface. So, extra attention is needed in such cases in order to construct well parametrised cross-section curves during the first steps of the geometric modelling algorithm.

Finally, in Figs. 20–22 we present the image acquisition–geometric modelling–finite element modelling cycle for a simple quadric surface, a cone. The results show that, using the technique described above, it is possible to model free-form objects and also surfaces, such as quadrics, that are often encountered in many modelling problems of artificial objects, such as mechanical components, archaeological objects etc.

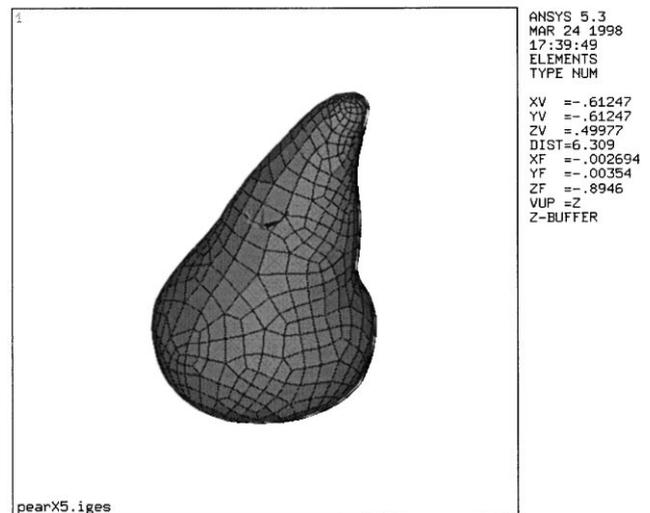


Fig. 19. Finite element mesh of the pear.

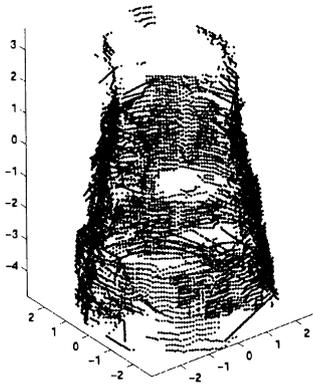


Fig. 20. Data points on the surface of a cone.

## 6. Conclusion

An image acquisition–geometric modelling system for the construction of geometric models of natural objects was presented. The system is able of scanning the desired object, obtain 3D information in the form of 3D data points and fit a surface to them, thus creating a mathematical description of the object.

In the image acquisition stage, a modified Differential Matching method is used in order to compute coordinates of points on the surface of the object.

Then, the computed 3D points are used in the geometric modelling part of the system. They are organised in cross-sectional data and a NURBS curve is fitted to each cross-section. The cross-section NURBS curves are then “skinned”, producing a NURBS surface approximating the object of interest.

The system presented is able to model both open and closed surfaces, with NURBS surfaces of arbitrary degree. In principle, objects with holes, slits etc. can be modelled, but somehow this information has to be obtained beforehand, probably in the image acquisition stage. In this case,

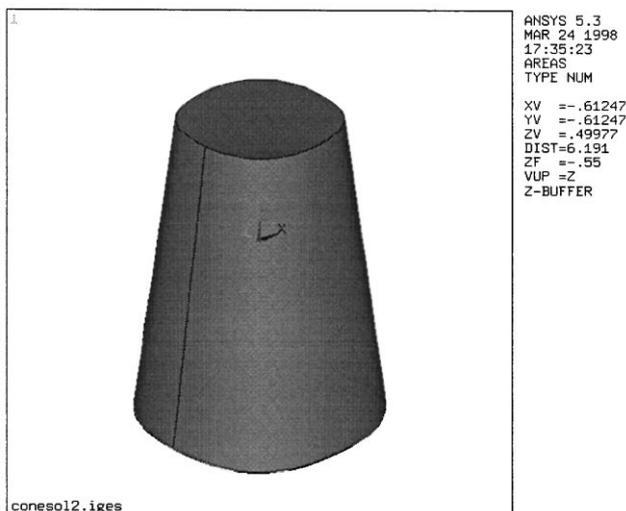


Fig. 21. NURBS model of the cone.

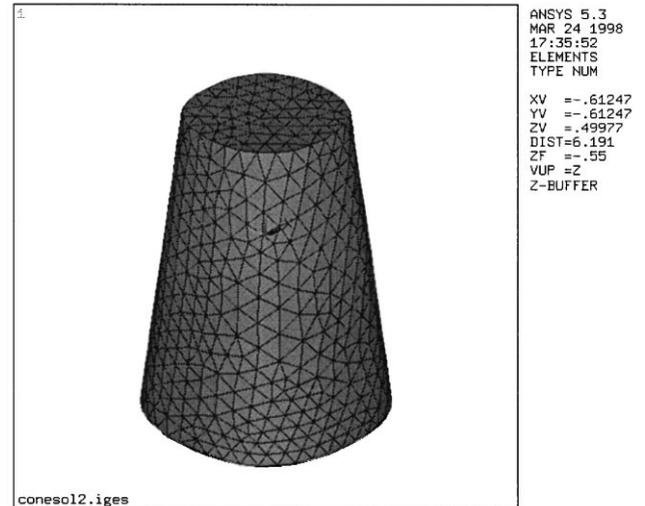


Fig. 22. Finite element mesh of the cone.

additional topological information must be added in the input to the geometric modelling system, i.e. some sort of neighbourhood connectivity information of the data points.

## Acknowledgements

This work was supported by Human Capital and Mobility Research Network ERBCHRXCT930386.

## References

- [1] Riesenfeld RF. Applications of B-Spline approximation to geometric problems of computer aided design. PhD Thesis, Syracuse University, Syracuse, New York, USA, 1973.
- [2] Coons SA. Surfaces for computer aided design of space forms. Technical Report MIT MAC-TR-41, Cambridge, MA, 1967.
- [3] Vesprille KJ. Computer aided design applications of the rational B-Spline approximation form. PhD Thesis, Syracuse University, Syracuse, New York, USA, 1975.
- [4] Barnsky BA, Greenberg DP. Determining a set of B-Spline vertices to generate an interpolating surface. *Computer Graphics and Image Processing* 1980;14(3):203–26.
- [5] Ma W. NURBS-based CAD modelling from measured points of physical models. PhD Thesis, Department of Mechanical Engineering, Catholic University of Leuven, Leuven, Belgium.
- [6] Boehm W. Inserting new knots into B-Spline curves. *CAD* 1980;12(4):199–201.
- [7] Sarkar B, Menq CH. Parameter optimisation in approximating curves and surfaces to measurement data. *CAGD* 1991;8:267–90.
- [8] Bajaj C, Ihm I, Warren JW. High order interpolation and least squares approximation using implicit algebraic Splines. *ACM Transactions on Graphics* 1993;12(4):327–47.
- [9] Middleditch A, Dimas E. Triangular algebraic surface patches. In: *Proceedings of CSG'94: International Conference on Constructive Solid Geometry*, 1994.
- [10] Woodward CD. Skinning techniques for interactive B-Spline surface interpolation. *CAD* 1988;20(8):441–51.
- [11] Piegl L, Tiller W. Curve and surface constructions using rational B-Splines. *CAD* 1987;19(9):485–98.
- [12] Piegl L. On NURBS: a survey. *IEEE Computer Graphics and Applications* 1991;January:55–71.

- [13] Alvertos, Brzakovic D, Gonzalez RC. Camera geometries for image matching in 3-D machine vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1989;11(6):897–914.
- [14] Marapane, Trivedi MM. Multiprimitive hierarchical stereo analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1994;16(3):227–40.
- [15] Haralik, Shapiro LG. *Computer and Robot Vision*. New York: Addison-Wesley, 1993.
- [16] Barron, Fleet DJ, Beauchemin SS. Performance of optical flow techniques. *International Journal of Computer Vision* 1994;12(1):43–7.
- [17] Psarakis M, Moustakides GV, Henning G. A modified differential method for the 3-D reconstruction problem accepted for presentation in Eusipco-98, Rhodes, Greece, September 1998.
- [18] Dimas E, Briassoulis D. 3-D geometric modelling based on NURBS: a review. In: *Proceedings of the Third International Conference on Computational Structures Technology, CST 96, Budapest 21st–23rd August, Hungary, 1996*.
- [19] Sienz J, Szarvasy I, Briassoulis D, Schulteiß B, Dimas E, Hinton E. NURBS based geometry modelling and automatic meshing of natural objects. In: Topping BHV, editor. *Proceedings of the 4th International Conference on Computational Structures Technology CST98, Edinburgh, August 1998*.
- [20] Schulteiß B, Briassoulis D, Dimas E, Psarakis E, Henning G. Use of advanced techniques to directly create finite element models of biological objects. In: *Proceedings of IEEE/EMBS Region 8, International Conference on Information Technology Applications in Biomedicine, Prague, 7–9 September 1997*. p. 98, 99.
- [21] Dewulf W, Janscók P, Nicolai, De Roeck G, Briassoulis D. Determining the firmness of a pear using finite element modal analysis. *Journal of Agricultural Engineering Research* 1999;74(3):217–24.