©1998, CIVIL-COMP Ltd., Edinburgh, Scotland
B.H.V. Topping, (Editor),
Advances in Engineering Computational Technology,
Civil-Comp Press, Edinburgh, 301-310

THREE DIMENSIONAL ACQUISITION AND NURBS BASED GEOMETRIC MODELLING OF NATURAL OBJECTS

E. Dimas*, E. Psarakis+, D. Briassoulis* and G. Moustakides+
***Department of Agricultural Engineering, Agricultural University of Athens, Greece**
+Computer Technology Institute, Patras, Greece

Abstract

This paper presents a recently developed Image Acquisition-Geometric Modelling system for the construction of geometric models of natural 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 digitised 3-D points, an approximating surface, a geometric model of the object of interest, is computed. 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.

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.

Highly sophisticated finite element packages have been commercially available, for analysis purposes. The construction of finite element models is fairly automated once a description of the geometry of the object of interest, i.e. a geometric model, is available. In some cases such models do exist. Mechanical components, for example, are often designed in front of a computer, using advanced drawing packages which have the ability to translate the drawing into a meaningful geometric model, which can be imported to a finite element analysis software package. However, constructing such models for biological objects (e.g. fruits, human body parts etc.), is a more complicated task, since they are not the product of a design process. Thus, the description of the geometry of natural objects often requires an advanced weaponry of mathematical tools and techniques.

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 three-dimensional 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 natural objects. This system is capable of obtaining 3D information from scanned images taken with a single camera, in 3D-point form. These points are then fitted in order to produce a geometric model (an equation) which describes the surface of the object of interest. 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[1].

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. monuments for which plans do not exist, fruits, biological objects etc.).

An image acquisition system has been developed for the creation of three-dimensional geometric data. The corresponding "3-D 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 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

[1]This work was supported by Human Capital and Mobility Research Network ERBCHRXCT930386

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 [6][7][8][9]. 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.

The technique used in the system organizes 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.

In section 2 of this paper the image acquisition technique is described. In section 3 the NURBS fitting techniques are described. Illustrative examples are presented in section 4.

2 Image Acquisition

The stereoscopic approach 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 Figure 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 φ are taken. Each pair is used to compute the 3-D coordinates of the corresponding sideview.

Notice that the acquisition system shown in Figure 1 is equivalent to a stereo model composed by two optical sensors, denoted by S_L and S_R in Figure 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 we use the lateral model [1] which is one of the simplest imaging models. The stereo camera arrangement of such a model is presented in Figure 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 $<i_L>$ and $<i_R>$ 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 $<i_L>$ and $<i_R>$ respectively, by using simple geometry we can easily relate the world coordinates to the image coordinates as follows:

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

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

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

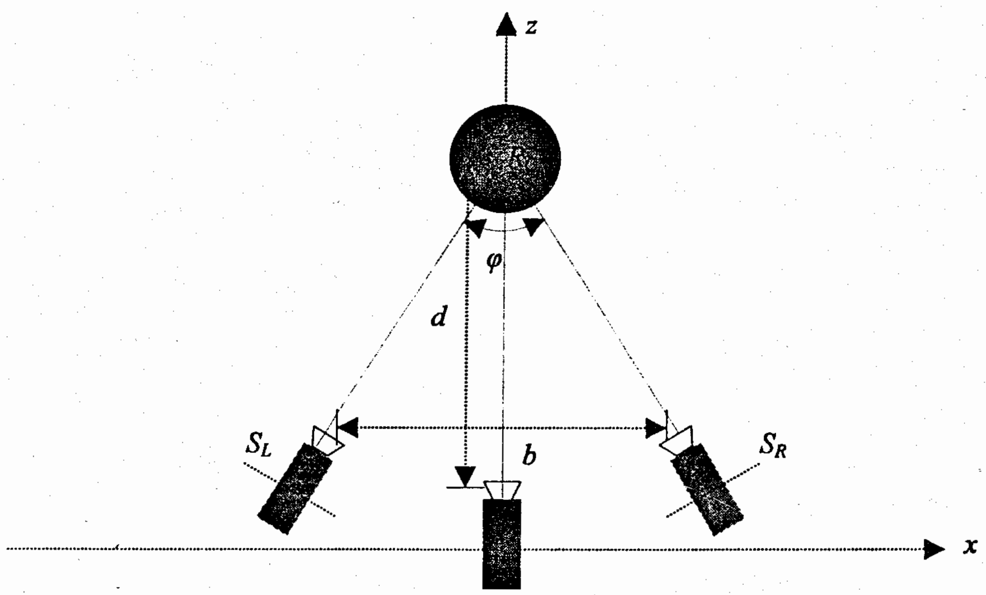

Figure 1. Top view of the existing, single camera (S), acquisition system and the equivalent model composed by the two cameras (S_L, S_R).

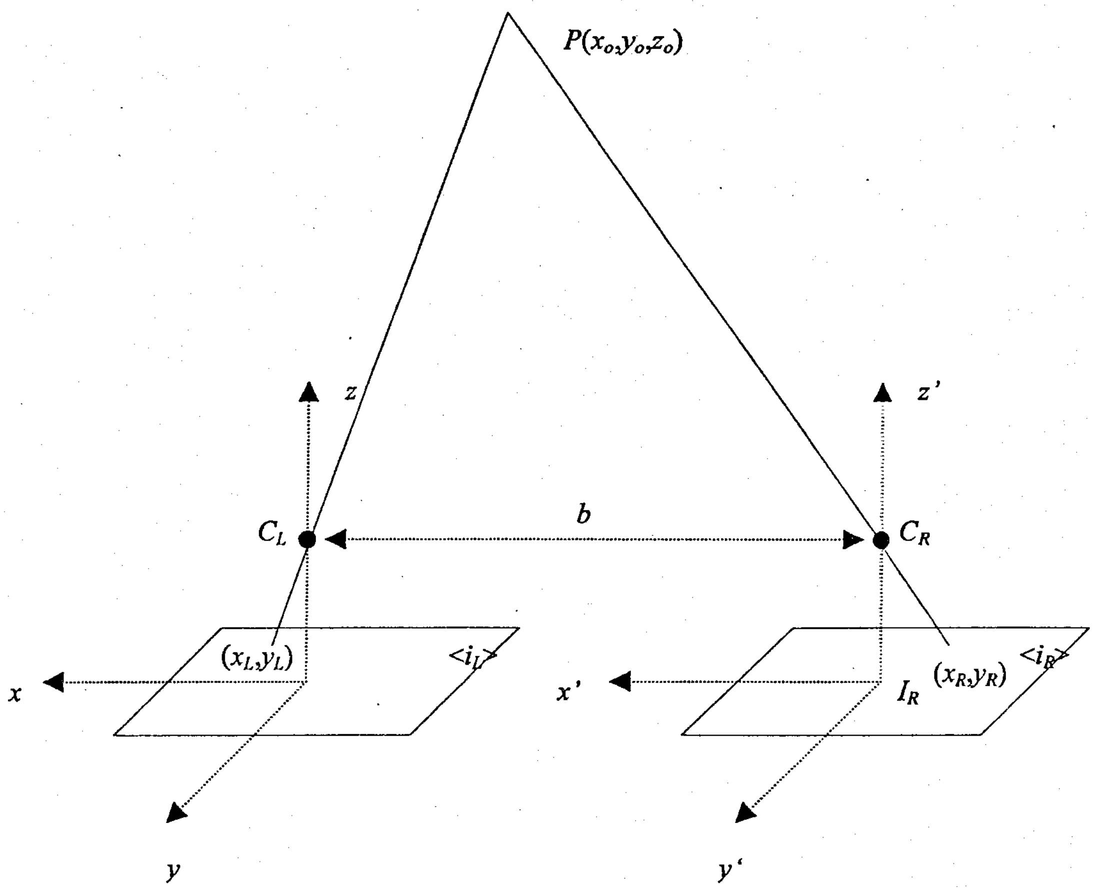

Figure 2. The lateral stereo camera model.

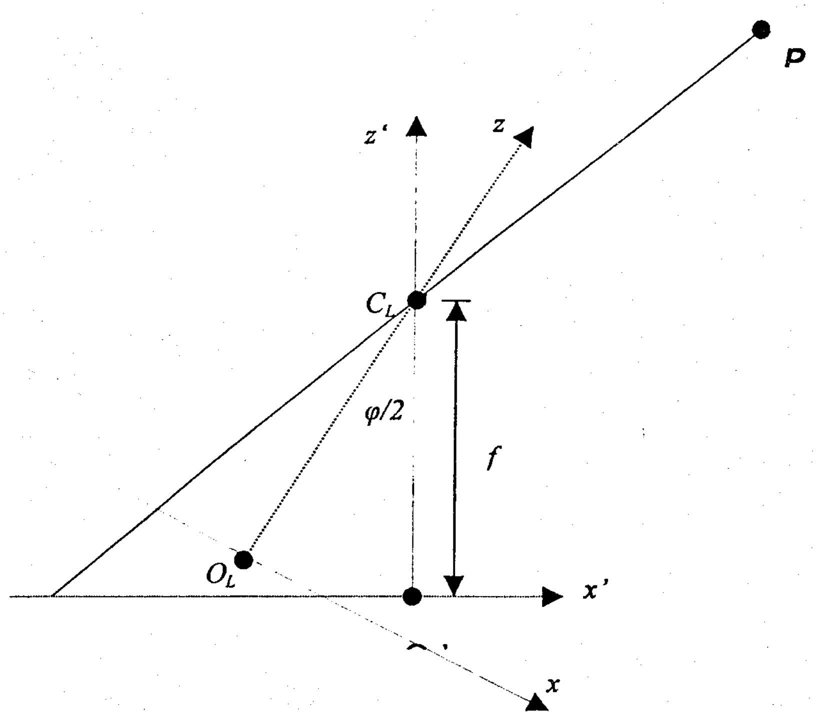

Figure 3. Geometry for the computation of the normal image.

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

The most difficult task in applying Equations (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 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 φ 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 and similarly we can find the corresponding transformation for the right. Let xO_Lz and $x'O_L'z'$ be the coordinate systems of the original and the normal image. Also let O_LA and $O_L'A'$ be the x-coordinates of the projections of the scene point P onto these two image planes as shown in Figure 3. Then, from the orthogonal triangles O_LC_LA and $O_L'C_LA'$ by using simple geometry we obtain the following relation for the x_L' coordinate for the left normal image:

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

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') co-ordinates, of the right normal image will be given by the following relations:

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

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

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

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 3-D information.

Stereo systems can be broadly classified into two categories [2]. 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) [3].

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 [4]. 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 [2].

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 the original non-linear optimisation problem to reduce it to a linear one without the need of any form of linearisation or any a priory knowledge of the nominal parameters. The estimates of the desired geometric and radiometric parameters are, as in the classical method, obtained though 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 [5].

The output of the image acquisition and the image matching algorithm is a cloud of 3D points on the surface of the object of interest

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 form.

The fitting method implemented is a combination of a curve approximation technique and a cross-sectional design technique. It is applied in a two-step fashion:

1. 3-D data points are organized in cross-sectional data, according to their distance from some user specified planes. The resulting data points (''filtered'' points) are approximated, thus creating cross-sectional curves, in NURBS form.
2. The NURBS cross-sectional curves are ''skinned'', in order to construct a NURBS surface.

The modelling procedure is described in the rest of this section. Subsection 3.1 defines, in summary, the B-Spline basis functions and the NURBS curve and surfaces resulting from this definition. The curve fitting algorithm is described in subsection 3.2, while the cross-sectional design technique in subsection 3.3.

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* $\mathbf{P}_i=[x_i, y_i, z_i]^T$, $i=1,n$, a set of *weights* w_i $i=1,\ldots,n$, a non-decreasing sequence of real numbers u_i $i=0,\ldots,n+p$ 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:

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

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

The curve itself is defined by the following formula:

$$C(s)=\frac{\sum_{i=0}^{n} w_i \mathbf{P}_i N_i^p(s)}{\sum_{i=0}^{n} w_i N_i^p(s)} \tag{9}$$

Similarly, a NURBS surface is defined by

$$S(s,t)=\frac{\sum_{i=0}^{k}\sum_{j=0}^{n} w_{ij}\mathbf{P}_{ij}N_i^p(s)N_j^q(t)}{\sum_{i=0}^{k}\sum_{j=0}^{n} w_{ij}N_i^p(s)N_j^q(t)} \tag{10}$$

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

3.2 Curve Fitting for 3D Points

The first step in the modelling procedure is to organize the data points in cross-sections. This is achieved according to their distance from a number of parallel user-specified planes (in a fixed distance between them), which cut through the computed data points of the object. All points, which are within some given distance from a specified plane, are retained, while the rest are ignored. Thus, the originally computed data points have now been "filtered", leaving only cross-sectional data to be taken into account for further processing. In this way, the number of points is significantly reduced, which speeds up the modelling procedure. Furthermore, there is no significant geometric information loss, because the number of cross-sections can be increased, according to the desired accuracy.

The next step is to construct NURBS curves, approximating the "filtered" cross-sectional data points. The curve fitting problem to be solved here can be stated as follows: Given a set of data points $\mathbf{Q}_i$, $i=1,2,\ldots,m$, find a NURBS curve 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 $\mathbf{Q}_i$, $i=1,\ldots,m$,
- the knot vector $\mathbf{U}$,
- The control points $\mathbf{P}_i$, and weights w_i, $i=1,\ldots,n$.

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 are fixed beforehand. The location parameters and the knot vector are also determined beforehand according to the distribution of the data points (this procedure is called data parameterization and it is described in section 3.2.2). Therefore, the fitting procedure results to the minimization of a distance criterion, based on the l_2 norm, with respect to the unknown weights and control points:

$$\sum_{l=1}^{m}\left\|C(s_l)-\mathbf{Q}_l\right\|_2^2=\sum_{l=1}^{m}\left\|\frac{\sum_{i=0}^{n} w_i\mathbf{P}_i N_i^p(s_l)}{\sum_{i=0}^{n} w_i N_i^p(s_l)}-\mathbf{Q}_l\right\|_2^2 \tag{11}$$

where s_i, $i=1,\ldots,m$ are location parameters of the data points.

3.2.1 Fitting Algorithm

A typical outline of the curve-fitting algorithm adopted is (described in [10]):

1. Input: 2-D or 3-D data points.
2. Assign initial parameter values to the data points.
3. Assume an initial knots distribution.
4. Go through an approximation procedure to obtain the weights and control points of the NURBS curve, i.e. obtain the fitting curve.
5. Optimise the parameter distribution if necessary to obtain a better fit [13]. Go through steps 3 to 5 until a satisfactory accuracy has been achieved.
6. Output: NURBS curve.

The first three steps of the algorithm are described in section 3.2.2. 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 significant gain. In the rest of this section the minimisation of Equation (11) with respect to the unknown control points and weights is briefly described.

The minimisation of the distance problem defined by equation (11) for curve fitting is equivalent to solving, in a least square sense, the linear systems

$$\begin{aligned}\mathbf{N}_{m\times r}\cdot\mathbf{P}_\mathrm{X}-\mathbf{Q}_\mathrm{X}\cdot\mathbf{N}_{m\times r}\cdot\mathbf{w}&=\mathbf{0}\\ \mathbf{N}_{m\times r}\cdot\mathbf{P}_\mathrm{y}-\mathbf{Q}_\mathrm{y}\cdot\mathbf{N}_{m\times r}\cdot\mathbf{w}&=\mathbf{0}\\ \mathbf{N}_{m\times r}\cdot\mathbf{P}_\mathrm{Z}-\mathbf{Q}_\mathrm{Z}\cdot\mathbf{N}_{m\times r}\cdot\mathbf{w}&=\mathbf{0}\end{aligned} \tag{12}$$

where $r=n+1$, $\mathbf{N} = \lfloor N_i(s_j) \rfloor$ for $i=1,\ldots,r$, $j=1,\ldots,m$, $\mathbf{w}$ is the vector of the unknown weights, $\mathbf{P_x}$, $\mathbf{P_y}$, $\mathbf{P_z}$ are vectors containing the unknown coordinates of the control points multiplied by their respective weight, i.e.

$$\begin{bmatrix}\mathbf{P_x} & \mathbf{P_y} & \mathbf{P_z}\end{bmatrix} = \begin{bmatrix} w_0x_0 & w_0y_0 & w_0z_0 \\ w_1x_1 & w_1y_1 & w_1z_1 \\ \vdots & \vdots & \vdots \\ w_nx_n & w_ny_n & w_nz_n \end{bmatrix} \quad (13)$$

and

$$\begin{aligned} \mathbf{Q_x} &= diag(X_1 \;\ldots\; X_m), \\ \mathbf{Q_y} &= diag(Y_1 \;\ldots\; Y_m), \\ \mathbf{Q_z} &= diag(Z_1 \;\ldots\; Z_m) \end{aligned} \quad (14)$$

are diagonal matrices which contain the coordinates of the data points. Equation (12) can be rearranged in matrix form

$$\begin{bmatrix} \mathbf{N} & \mathbf{0} & \mathbf{0} & -\mathbf{Q_x}\cdot\mathbf{N} \\ \mathbf{0} & \mathbf{N} & \mathbf{0} & -\mathbf{Q_y}\cdot\mathbf{N} \\ \mathbf{0} & \mathbf{0} & \mathbf{N} & -\mathbf{Q_z}\cdot\mathbf{N} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P_x} \\ \mathbf{P_y} \\ \mathbf{P_z} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (15)$$

In solving the homogeneous linear system (15) in a least square sense, special care is needed for the computation of weights because they need to be positive. In [10] 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

$$\mathbf{M}_{m\times(n+1)} \cdot \mathbf{w} = \mathbf{0} \quad (16)$$

where $\mathbf{M}$ is a symmetric nonnegative matrix of dimension $m\cdot(n+1)$. The unknown weights can be found by minimizing the Rayleigh quotient of matrix $\mathbf{M}$ under suitable constraints for the weights. Interpolating solutions are found if rank(M)<(n+1) [10].

It was shown in [10] that the same method can be easily extended for surface fitting of 3D data points.

3.2.2 Data Parameterization

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

1. The degree p of the basis functions in the s direction and the number of control points and weights n 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, $i=1,\ldots,m$ corresponding to data points are computed.
3. A suitable knot vector $\mathbf{U}$, in parametric direction s, is determined, usually taking into account the parameter distribution.

Computation of location parameters: There are several methods available, in order to compute location parameters from the data points:

- *Uniform:* 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

$$s_i = \frac{i-1}{m-1} \qquad i = 1,\ldots,m \quad (17)$$

for curve fitting of m data points.

- *Cumulative Chord Length:* 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:

$$s_i = s_{i-1} + \frac{\left\|\mathbf{Q}_i - \mathbf{Q}_{i-1}\right\|}{\sum_{j=2}^{m}\left\|\mathbf{Q}_j - \mathbf{Q}_{j-1}\right\|} \qquad i = 2,\ldots,m, s_1 = 0. \quad (18)$$

- *Centripetal:* This method observes the changing curvature of the underlying curve

$$s_i = s_{i-1} + \frac{\left\|\mathbf{Q}_i - \mathbf{Q}_{i-1}\right\|^{\frac{1}{2}}}{\sum_{j=2}^{m}\left\|\mathbf{Q}_j - \mathbf{Q}_{j-1}\right\|^{\frac{1}{2}}}, i = 2,\ldots,m, s_1 = 0. \quad (19)$$

- *Base curve parametrization:* This method is due to Ma [10]. Each data point is associated with a point on a simple known underlying curve (for example by minimising the distance of the data point to the curve with respect to the parameter values). The parameter values of the associated point on the underlying curve (base curve) are the location parameter values for the data point. This method can be applied recursively starting with a simple curve, improving its shape and repeating the parameter computation until a satisfactory result is obtained. For more on the curve parameterization method see [10].

All of the above methods are available in the current system and, theoretically, they could all be used (with different results in terms of accuracy). However, in all of the above methods except the base curve parameterization method the data points are assumed to be chain or grid distributed.

In the last method 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. Therefore, at this point, the problems encountered in the location parameter computation stage of the algorithm should be mentioned. Due 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 essential for computing the knot vectors and affects the shape of the resulting curve. This problem is overcome by adopting and using the base curve parameterization method described above. The data points can be ordered by ordering the s location parameters of their respective associated points.

The base curve technique can be also generalised for surface fitting of 3D data points [10]

Knot vector computation: A suitable knot vector $\mathbf{U}$ in parametric direction s, is usually computed by taking into account the parameter distribution. Commonly used methods for knot values allocation are:

- *Simple Knots setting*

$$u_i = \begin{cases} 0 & i = 0,\ldots,p \\ \dfrac{i-p-1}{n-p+11} & i = p+1,\ldots,n \\ 1 & i = n+1, n+p \end{cases} \qquad (20)$$

- *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. Hence, good parameterization of the data points is essential. For more on averaging methods see [10].

In the system presented here averaging methods are used because, although they are more expensive computationally they lead to more accurate models. The resulting curve or surface is more accurate because more basis functions (and therefore more control points) are assigned to areas were the distribution of the data points is denser.

3.3 Cross-sectional Design

The fitting technique employed in the current system is based on a cross-sectional design technique, which fits a NURBS surface to a sequence of cross-sectional NURBS curves. For more on cross-sectional design see [15][16]. A NURBS surface is fitted to the cross-section curves that have been computed with the method described in section 3.2. The same method can be also applied, in the case where the cross-sections are given or are otherwise computed. However, before the "skinning" of the curves, certain parameters have to be computed before hand (data parameterization). The data parameterization procedure is described in the section 3.3.2.

3.3.1 Fitting Algorithm

The fitting algorithm adopted in the system presented here is:

1. Input: Crossectional NURBS curves.
2. Make all curves compatible (if they are not already).
3. Go through a skinning technique.
4. Output: NURBS surface.

The skinning algorithm used in the geometric modeller is presented in the following.

The fitting problem can be mathematically stated as follows: A collection of NURBS curves

$$C^l(s) = \frac{\sum_{i=0}^{k} w_i^l \mathbf{P}_i^l N_i^p(s)}{\sum_{i=0}^{k} w_i^l N_i^p(s)} \qquad l=1,\ldots,n \qquad (21)$$

is given. Then, a NURBS surface (defined by Equation (10)) that interpolates the curves 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

$$S(s,t_l) = C^l(s) \qquad l=1,\ldots,n \qquad (22)$$

It should be emphasised that the crossection curves defined by equation (21) 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. If they are not, degree raising and knot insertion algorithms like the Boehm or the Oslo algorithm [12] can be employed to force compatibility (step 2 of the algorithm).

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 NURBS curve of degree p with weights [12]

$$w_i = \sum_{j=0}^{n} w_{ij} N_j^q(a) \qquad (23)$$

and control points

$$\mathbf{P}_i = \sum_{j=0}^{n} \frac{w_{ij} N_j^q(a)}{w_i} \mathbf{P}_{ij} \qquad (24)$$

The cross-section curve defined by equation (21) should be identical to the curve defined by the control points and weights defined by equations (23-24) at parameter value $t=t_l$. Therefore,

$$w_i = w_i^l \quad \text{and} \quad \mathbf{P}_i = \mathbf{P}_i^l,\ i = 0,..,k \qquad (25)$$

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

$$\sum_{j=0}^{n} N_j^q(t_l) w_{ij} = w_i^l, \quad i = 0,\ldots,k, l = 0,\ldots,n \qquad (26)$$

$$\sum_{j=0}^{n} w_{ij} N_j^q(t_l) \mathbf{P}_{ij} = w_i^l \mathbf{P}_i^l, \quad i = 0,\ldots,k, l = 0,\ldots,n \qquad (27)$$

Solving equations (26-27) guarantees interpolation of the section curves by the NURBS surface. The solution is computed in a two step fashion; first solving Equation (26) for the unknown weights and then solving Equation (27) for the unknown control points.

Assume that the control points of the cross-section curves form a matrix (in fact three matrices of x, y, z coordinates), in which each row of control points consists of the control points of a specific cross-section curve. Intuitively, the skinning procedure amounts to interpolating each column of this matrix by a NURBS curve. The control points of the resulting interpolating curves are the control points of the computed NURBS surface.

3.3.2 Data Parameterization

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

1. Degrees of the cross-sectional and longitudinal curves as well as the number of weights and control points in each parametric direction, i.e.
   - Degree p of the cross-sectional curves (say in the s-direction).
   - Degree q of the longitudinal curves.
   - Number of weights and control points for each cross-sectional curve.
   - Number of weights and control points for each longitudinal curve.

   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 curves.
2. Location parameters of the longitudinal curves. These can be computed by parameterizing (by any of the methods for curve fitting) the control points of the given curves in the longitudinal direction.
3. Knot vector for longitudinal curves. These can be computed by any of the methods described for curve fitting once the parameter values have been determined.

Note that the longitudinal curves must be compatible, in the same way the cross-sections curves are compatible.

4 Illustrative Examples

In this section illustrative examples are presented in order to demonstrate the techniques employed in the image acquisition-geometric modelling system. For this purpose a peach and a pear were scanned and modelled. The peach example, illustrates modelling of an open surface, while the pear is modelled by a closed surface. Points on the surface of the peach and the pear were computed by using the image acquisition method described in section 2. The cloud of 3D points for the peach and the pear are shown in Figures 4 and 8 respectively. The data points were fitted by using the techniques described in sections 3.2 and 3.3 of this paper:

1. Normally, the raw 3D data points are filtered according to their distance from user specified planes. All points within some tolerance from a particular plane are classified as belonging to the same cross-section curve. However, in the peach example, there was no need for "filtering", because the data points were organised in cross-sections beforehand (during the image acquisition phase). In the pear example the original 3D data points had to be "filtered". The cross-sectional data points of the peach and the pear are shown in Figures 5 and 9 respectively. Notice that the cross-sectional data of the peach do not determine cross-sections in the classical sense of the term. However, the fitting method works well.
2. The points defining each curve were parameterized by a base curve parameterization method and the knot vectors were computed and made compatible.
3. The cross-sectional data were subsequently fitted with cubic NURBS curves, with 20 control points each, using the curve fitting technique described in section 3.2. This means that the weights w_i^l and control points $\mathbf{P}_i^l$ of Equation (21) were computed. The resulting curves are shown in Figure 6 and 10 for the peach and the pear respectively.
4. Then, the weights and control points of the resulting NURBS surface were computed using Equations (26) and (27). The peach surface is a bicubic, while the pear surface is of degree 3 in the cross-sectional direction and of degree 2 in the longitudinal direction.

Although the peach model is smooth and can be automatically imported into a finite element package for further processing, any analysis result is of little importance since the model is incomplete, i.e. a fruit is a closed surface and not open as in the surface of Figure 7.

The pear model, however, was constructed in such a way that it is closed in both parametric directions, in order to illustrate that the method works well for closed surfaces. The modelling procedure was the same as in the case of the peach, but the cross-sections curves of the pear are smoothly closed. In order to achieve that special care is needed in order to satisfy smoothness conditions by the knots and control points of each cross-section curve [10].

5 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, 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.

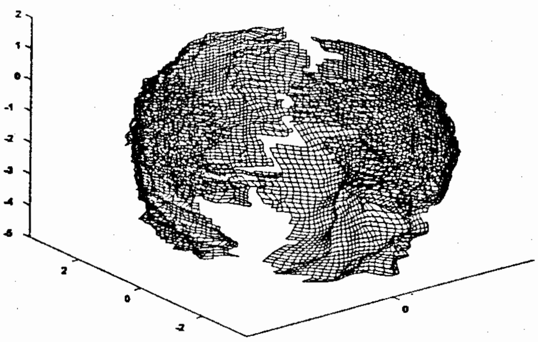

Figure 4. Raw data on the surface of a peach.

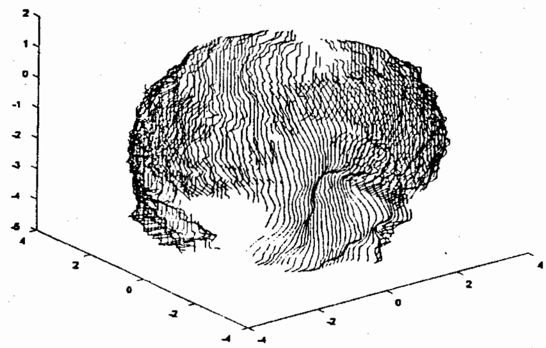

Figure 5. Cross sectional data on the surface of the peach.

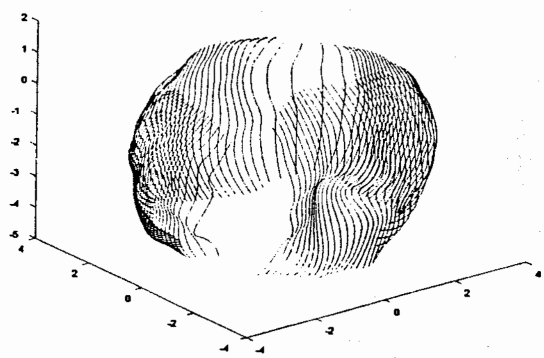

Figure 6. Cross-sectional NURBS curves for the data points of Figure 5.

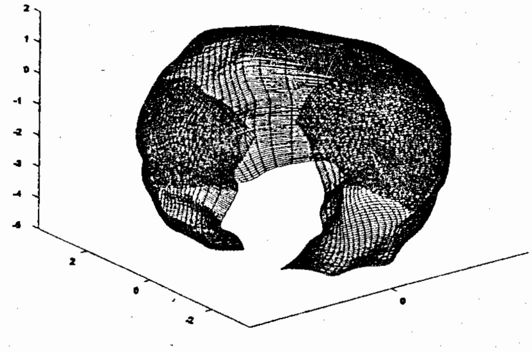

Figure 7. NURBS surface for the peach of Figure 4.

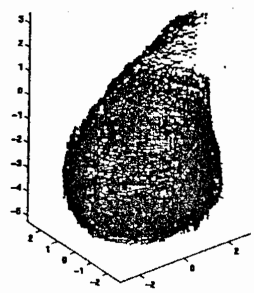

Figure 8. Raw data points on the surface of a pear.

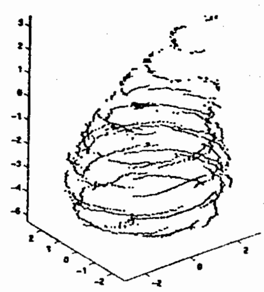

Figure 9. Cross-sectional data on the surface of the pear.

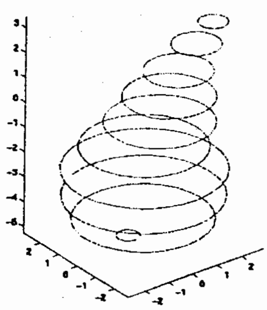

Figure 10. NURBS cross-section curves for the data points of Figure 9.

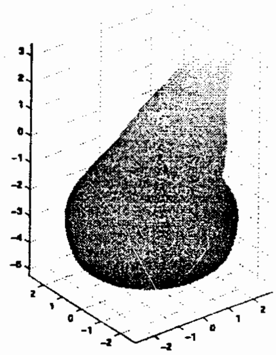

Figure 11. NURBS surface for the pear of Figure 8.

References

[1] N. Alvertos, D. Brzakovic and R.C. Gonzalez, "*Camera Geometries for Image Matching in 3-D Machine Vision*", IEEE trans. on Patt. Anal. Machine Intell., vol. 11, no. 6, pp. 897-914 (1989)

[2] S.B. Marapane and M.M. Trivedi, "*Multiprimitive hierarchical stereo analysis*", IEEE Trans. on Patt. Anal. Machine Intell., vol. 16, no. 3, pp. 227-240, (1994)

[3] R.M. Haralik and L.G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, New York (1993).

[4] J.L. Barron, D. J. Fleet and S.S. Beauchemin, "*Performance of optical flow techniques*", International Journal of Computer Vision, vol. 12, no. 1, pp. 43-47 (1994)

[5] E. Z. Psarakis, G. V. Moustakides and G. Henning, "*A modified Differential Method for the 3-D reconstruction Problem*" Accepted for presentation in *Eusipco-98*, Rhodes, Greece, (1998)

[6] Riesenfeld, "*Applications of B-Spline Approximation to Geometric Problems of Computer Aided Design*", PhD Thesis, Syracuse University, Syracuse, New York, USA (1973)

[7] Coons, "*Surfaces for Computer Aided Design of Space Forms*", Technical Report MIT MAC-TR-41, Cambridge, MA (1967)

[8] Vesprille, "*Computer Aided Design Applications of the Rational B-Spline Approximation Form*", PhD Thesis, Syracuse University, Syracuse, New York, USA (1975)

[9] Barnsky and D.P. Greenberg, "*Determining a Set of B-Spline Vertices to Generate an Interpolating Surface*", Computer Graphics and Image Processing, Vol. 14, No 3, pp. 203-226 (1980).

[10] W. Ma, "*NURBS-Based CAD Modelling from Measured Points of Physical Models*", PhD Thesis, Dept. of Mechanical Engineering, Catholic University of Leuven, Leuven, Belgium (1995).

[11] Boehm, "*Inserting new knots into B-Spline Curves*", CAD, Vol. 12, No 4, pp. 199-201 (1980).

[12] E. Dimas E. and D. Briassoulis, "*3-D Geometric Modelling Based on NURBS: A Review*", Proceedings of Third International Conference on Computational Structures Technology, CST 96, Budapest 21st-23rd August, Hungary (1996).

[13] Sarkar and C.H. Menq, "*Parameter Optimisation in Approximating Curves and Surfaces to Measurement Data*", CAGD, Vol. 8, pp. 267-290 (1991)

[14] Middleditch and E. Dimas, "*Triangular Algebraic Surface Patches*", Proceedings of CSG94: International Conference on CSG, (1994)

[15] C.D. Woodward, "*Skinning Techniques for Interactive B-Spline Surface Interpolation*", CAD, Vol. 20, No 8, pp. 441-451 (1988)

[16] L. Piegl and W. Tiller, "*Curve and Surface Constructions using Rational BSplines*", CAD, Vol. 19, No 9, pp. 485-498 (1987)