3D Shape Modelling through a Constrained Estimation of a Bicubic B-spline Surface

Xinquan Shen and Michael Spann
School of Electronics and Electrical Engineering
The University of Birmingham, B15 2TT, UK
[shenx|spannm]@eee.bham.ac.uk

Abstract

This paper presents a new method to extract the 3D shape of objects from 3D gray level images using a bicubic B-spline surface model. Extraction of object shape is achieved through a hierarchical surface fitting by exploiting the multi-scale representation of the model. A strategy for converting the surface estimation into curve estimations is devised. The model surface is estimated by successively computing a set of cubic B-spline curves consisting of a coordinate curve net defining the surface. A regularising component is incorporated into the curve estimation to encourage the generation of an orthogonal coordinate curve net, preventing the creation of unwanted creases. Experimental results are presented for extracting the 3D shape of objects from real 3D images.

1 Introduction

The interpretation of 3D images often needs the shape information of objects in the image. A set of unmodelled 3D structures derived from local low level operations (e.g. edge detection) [10] can hardly ever be directly used as shape information since 3D images such as medical and geological images are often very noisy and objects in the image often have a complex shape. An alternative way is to detect all the surface elements belonging to the same object and integrating them into a shape model describing the object surface.

Basically, the mathematical form of the shape model describing the target object surface determines the technique for fusing detected surface elements (e.g. 3D points) into the model. Physically-based/active surface models [15, 3, 11] describe the object surface through the equilibrium of internal and external forces acting on the model. The extraction of the object surface is achieved through the deformation of the model by imposing external forces derived from the detected 3D positions on the target object surface to attract the model surface towards the object surface. The stiffness of the model is determined by material parameters which should be carefully chosen in order to obtain correct result. The final result is often dependent on the initial position/shape of the model. Topologically adaptable models [6, 5, 2, 14] describe the object surface using a set of 3D points defining a surface mesh (e.g. a triangular mesh). The surface of an object with arbitrary unknown topology can be estimated through a model deformation followed by a re-parameterisation [6], an oriented dynamical motion of the defining points of the model [14], or a surface evolution implemented by level set approaches [5, 2]. The method is

BMVC 1998 doi:10.5244/C.12.28

often difficult in controlling the balance between the tracking of shape details and the resistance to the noise effect. It is unlikely to obtain satisfactory result when the object in the image presents a uncontinuous (broken) boundary due to the noise involved in the imaging procedure.

In many real applications, the topology of objects to be modelled is known *a priori*. The operation of the shape estimation can thus be associated with a confined solution space so that it only seeks an object shape with some expected properties (e.g. a closed smooth surface). In the work reported in [13], a method is proposed for extracting the 3D shape of objects from 3D gray level images. A bicubic B-spline surface is used as a shape model with a scale determined by the number of control points defining the spline surface. The extraction of the object shape is achieved through a hierarchical processing in which the model at different scales is successively used to estimate the object surface. At each scale level, the object surface is estimated by iteratively fitting the surface model to a set of data points which are believed to be located on the boundary of the object and detected by searching, within automatically controlled search radii, voxels with the strongest edge response along directions normal to the model surface. However, the method is not implemented in an efficient way. The resulting surface may contain creases produced due to the irregularity of the coordinate curve net defining the surface (see Fig. 1(c)). These creases are not responsible for any shape details, and local geometrical structures (e.g. curvatures) in these areas are often not reliable.

In this paper, we present a substantial improvement to the work reported in [13]. By exploring the properties of bicubic B-spline surface, the computation of the surface fitting at each iteration is implemented in an efficient way through estimating a set of cubic B-spline curves which consists of a coordinate curve net defining the surface. A constraint term is then introduced into the curve estimation to encourage the generation of an orthogonal coordinate curve net and therefore to prevent the creation of unwanted creases. The result is a C^2 continuous surface approximating the object surface to be modelled. Compared to the work reported in [13], the new method is more reliable in shape estimation and produces better results.

2 Surface model

The object surface $\boldsymbol{S}(u,v) = [x(u,v)\, y(u,v)\, z(u,v)]^T$ is assumed to be parametrically described by a uniform bicubic B-spline surface expressed as follows [1]:

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \boldsymbol{\alpha}_{ij} b_i(u) b_j(v) \tag{1}$$

where, $b_i(u)$ and $b_j(v)$ are B-spline basis functions of order 4 defined on parameters u and v respectively, $\boldsymbol{\alpha}_{ij} = [X_{ij}\, Y_{ij}\, Z_{ij}]^T$ are control points defining the surface.

The number of control points used for defining the spline surface can be considered as a scale associated with the model. Thus, a multi-scale representation of the model can be obtained by choosing M and N as the function of a scale parameter k ($k \geq 0$). In our implementation, the multi-scale of the model is defined as follows:

$$\begin{aligned} M(k) &= M_0 + ck \\ N(k) &= N_0 + ck \end{aligned} \tag{2}$$

where, M_0 and N_0 determine the model at the base scale ($k = 0$), and c is a constant specifying the scale increment.

The type of the spline surface and boundary conditions associated with the model are specified according to the prior information about the object surface. In our experiments, the object surface is a closed surface topological to a sphere. The spline model is chosen so that it is open in the u parameter and closed in the v parameter [1], and has the following boundary conditions:

$$\begin{array}{l} \boldsymbol{\alpha}_{0j} = \boldsymbol{\alpha}_{00} \\ \boldsymbol{\alpha}_{M-1,j} = \boldsymbol{\alpha}_{M-1,0} \end{array} , \quad 1 \leq j \leq N-1$$

The problem of the extraction of object shape now becomes the estimation of the control points $\boldsymbol{\alpha}_{ij}$ from a set of image points which are considered to be located on the object surface.

3 Model estimation

Given a set of data points $\boldsymbol{S}'_{st}$, $s = 0, 1, \ldots, P-1$, and $t = 0, 1, \ldots, Q-1$, the control points of the model can be estimated using the least squares method by minimising the follow error:

$$\text{Error} = \sum_{s=0}^{P-1} \sum_{t=0}^{Q-1} \left\| \boldsymbol{S}'_{st} - \boldsymbol{S}(u_s, v_t) \right\|^2 \tag{3}$$

P and Q are chosen to satisfy the conditions: $P \geq M$, and $Q \geq N$. It has been shown [13] that the minimisation of (3) leads to a matrix equation involving matrices of order of $PQ \times MN$. Solving the control points $\boldsymbol{\alpha}_{ij}$ from this matrix equation requires a computational complexity of $O(3MNPQ)$. As the model scale increases, the computation of the model becomes expensive.

3.1 A strategy of fast estimation

In this work, we present a fast method for estimating the model by exploring the fact that the B-spline surface is bicubic. The surface estimation problem is converted into the problem of curve estimation. Re-write (1) as

$$\boldsymbol{S}(u, v) = \sum_{i=0}^{M-1} b_i(u) \boldsymbol{\beta}_i(v) \tag{4}$$

where,

$$\begin{aligned} \boldsymbol{\beta}_i(v) &= [x_i(v)\, y_i(v)\, z_i(v)]^T \\ &= \sum_{j=0}^{N-1} b_j(v) \boldsymbol{\alpha}_{ij} \end{aligned} \tag{5}$$

For a fixed value of the parameter v, equation (4) describes a cubic B-spline curve (called *u-curve*) defined by control points $\boldsymbol{\beta}_i(v)$, $0 \leq i < M$. From the given data set $\boldsymbol{S}'_{st}$,

$0 \leq s < P$ and $0 \leq t < Q$, we compute the control points $\boldsymbol{\beta}_i(v_t)$ by minimising the following error:

$$\text{Error} = \sum_{s=0}^{P-1} \|\boldsymbol{S}'_{st} - \sum_{i=0}^{M-1} b_i(u_s)\boldsymbol{\beta}_i(v_t)\|^2 \tag{6}$$

By vanishing the first order derivatives with respect to the $\boldsymbol{\beta}_i(v_t)$, and collecting the data points and control points into vectors, $\boldsymbol{\beta}_i(v_t)$ can be computed by solving the following matrix equation:

$$\boldsymbol{G} = \left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1} \boldsymbol{B}^T\boldsymbol{D} \tag{7}$$

where $\boldsymbol{B}$, $\boldsymbol{D}$, and $\boldsymbol{G}$ have following forms:

$$\boldsymbol{B} = \begin{bmatrix} b_0(u_0) & \dots & b_{M-1}(u_0) \\ \vdots & \vdots & \vdots \\ b_0(u_{P-1}) & \dots & b_{M-1}(u_{P-1}) \end{bmatrix}, \boldsymbol{D} = \begin{bmatrix} \boldsymbol{S'}^T_{0t} \\ \vdots \\ \boldsymbol{S'}^T_{P-1,t} \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} \boldsymbol{\beta}^T_0(v_t) \\ \vdots \\ \boldsymbol{\beta}^T_{M-1}(v_t) \end{bmatrix}$$

A correspondence between a data point $\boldsymbol{S}'_{st}$ and the parameter values u_s and v_t must be established in order to compute the matrix $\boldsymbol{B}$. A usual way to achieve this is to determine the u_s and v_t based on the chord length between two successive data points [1, 9]. For an irregular data point net, the matrix $\boldsymbol{B}^T\boldsymbol{B}$ obtained from the determined parameter values may be singular. To ensure a valid computation, data points are first adjusted by applying a linear interpolation so that they are equally spaced in u and v [12]. Parameter values corresponding to a data point $\boldsymbol{S}'_{st}$ are then determined as follows:

$$\begin{aligned} u_s &= \frac{s}{P-1} \cdot u_{max} \\ v_t &= \frac{t}{Q-1} \cdot v_{max} \end{aligned} \tag{8}$$

where, u_{max} and v_{max} are the maximum parameter values which are uniquely determined by the type of the spline surface [9].

Based on (8), the matrix $\boldsymbol{B}^T$ is full ranked, and therefore, $\boldsymbol{B}^T\boldsymbol{B}$ is non-singular. Also, for the fixed value of M, N, and P, the matrix $(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$ is fixed. Noting that the matrix $(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$ only needs to be computed once for estimating all Q u-curves, the computational complexity of the equation (7) can be considered as $O(3MP)$. The complexity for computing all Q u-curves leading to MQ control points[1] $\boldsymbol{\beta}_{it}$, $0 \leq i < M$ and $0 \leq t < Q$, is $O(3MPQ)$.

On the other hand, equation (5) describes a cubic B-spline curve (called *v-curve*) defined by control points $\boldsymbol{\alpha}_{ij}$, $0 \leq j < N$. In a similar way to the estimation of u-curves, the MN control points $\boldsymbol{\alpha}_{ij}$, $0 \leq i < M$ and $0 \leq j < N$, can be estimated through computing M v-curves using already computed points $\boldsymbol{\beta}_{it}$, $0 \leq i < M$ and $0 \leq t < Q$. The complexity of this computation is $O(3MNQ)$.

In summary, the B-spline surface model defined in (1) can be estimated from a set of PQ data points by first estimating Q u-curves each defined by M control points and then estimating M v-curves each defined by N control points. The complexity of this

[1]Since parameter values u_s and v_t are uniquely determined by indices s and t, we simply note $\beta_i(v_t)$ as β_{it}

computation will be $O(3MQ(N+P))$. Since $N+P < NP$, for $N > 2$ and $P > 2$, this method for surface fitting is much more efficient than the previous one which has a complexity of $O(3MNPQ)$.

3.2 Applying constraints

u-curves and v-curves form a curve net (called *coordinate curve net* [4]) on the model surface. In the estimation method presented in the previous section, each u-curve and v-curve is estimated independently, and no constraint is applied to the configuration of the coordinate curve net. Although the result is a C^2 continuous surface, the corresponding coordinate curve net may be very irregular in some local area to form some creases (see Fig. 1(c)) which are not responsible for any shape details. In this section, we further modify the estimation method to prevent the creation of creases. The idea is to introduce a constraint into the curve estimation to encourage the generation of an orthogonal coordinate curve net.

For given PQ data points, all u-curves described by (4) are first estimated using the least squares method. Let $\boldsymbol{\beta}_{it} = [x_{it}\, y_{it}\, z_{it}]^T$, $0 \le i < M$ and $0 \le t < Q$, be the obtained MQ control points defining Q u-curves, and $\tilde{u}_i$, $0 \le i < M$, be the corresponding u parameter values determined in a similar fashion using (8). For each obtained u-curve, its direction (unit tangent vector) can be computed. Let $\boldsymbol{\delta}_u(i) = [a_x^{(t)}(\tilde{u}_i)\, a_y^{(t)}(\tilde{u}_i)\, a_z^{(t)}(\tilde{u}_i)]^T$ be the direction of the tth u-curve at the position $u = \tilde{u}_i$. According to the estimation scheme presented in the previous section, the model surface can then be estimated by computing M v-curves described by (5). To encourage the generation of an orthogonal coordinate curve net, each v-curve (say, ith v-curve) is estimated by minimising the following error:

$$\text{Error} = \sum_{t=0}^{Q-1} \|\boldsymbol{\beta}_{it} - \sum_{j=0}^{N-1} \boldsymbol{\alpha}_{ij} b_j(v_t)\|^2 + \frac{\lambda}{Q} \sum_{t=0}^{Q-1} \left[\boldsymbol{\delta}_u^T(i)\boldsymbol{\delta}_v(t)\right]^2 \qquad (9)$$

where, λ is a weighting parameter, and

$$\boldsymbol{\delta}_v(t) = \sum_{j=0}^{N-1} \boldsymbol{\alpha}_{ij} \left.\frac{\partial b_j(v)}{\partial v}\right|_{v=v_t} \qquad (10)$$

Obviously, $\boldsymbol{\delta}_v(t)$ describes the direction of the v-curve at the position $v = v_t$. The second term on the right of (9) counts the average of dot products between the direction of the v-curve to be estimated and the directions of already computed u-curves. It acts as a regularising component [7] in the minimisation operation to encourage the selection of a v-curve which tends to be orthogonal to u-curves. The strength of the regularisation is controlled by the weighting parameter λ. When $\lambda = 0$, the problem is reduced to the standard least squares estimation.

By vanishing the first order derivatives with respect to $\boldsymbol{\alpha}_{ij}$, $0 \le j < N$, and expressing the result using matrix notation, $\boldsymbol{\alpha}_{ij}$ can be computed by solving the following matrix equation:

$$\begin{bmatrix} \boldsymbol{C}^T & & \\ & \boldsymbol{C}^T & \\ & & \boldsymbol{C}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & & \\ & \boldsymbol{C}^T\boldsymbol{C} & \\ & & \boldsymbol{C}^T\boldsymbol{C} \end{bmatrix} - \right.$$

$$\frac{\lambda}{Q}\begin{bmatrix} \tilde{\boldsymbol{C}}^T\boldsymbol{W}_1\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^T\boldsymbol{W}_2\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^T\boldsymbol{W}_3\tilde{\boldsymbol{C}} \\ \tilde{\boldsymbol{C}}^T\boldsymbol{W}_2\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^T\boldsymbol{W}_4\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^T\boldsymbol{W}_5\tilde{\boldsymbol{C}} \\ \tilde{\boldsymbol{C}}^T\boldsymbol{W}_3\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^T\boldsymbol{W}_5\tilde{\boldsymbol{C}} & \tilde{\boldsymbol{C}}^T\boldsymbol{W}_6\tilde{\boldsymbol{C}} \end{bmatrix}\Bigg)\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} \tag{11}$$

where,

$$\boldsymbol{C} = \begin{bmatrix} b_0(v_0) & \dots & b_{N-1}(v_0) \\ \vdots & \vdots & \vdots \\ b_0(v_{Q-1}) & \dots & b_{N-1}(v_{Q-1}) \end{bmatrix}, \tilde{\boldsymbol{C}} = \begin{bmatrix} \dot{b}_0(v_0) & \dots & \dot{b}_{N-1}(v_0) \\ \vdots & \vdots & \vdots \\ \dot{b}_0(v_{Q-1}) & \dots & \dot{b}_{N-1}(v_{Q-1}) \end{bmatrix}$$

where $\dot{b}_j(v_t) = \left.\frac{\partial b_j(v_t)}{\partial v}\right|_{v=v_t}$. $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are $Q \times 1$ vectors, and

$$[\boldsymbol{x}\,\boldsymbol{y}\,\boldsymbol{z}] = [\boldsymbol{\beta}_{i0}\,\boldsymbol{\beta}_{i1}\ \dots\ \boldsymbol{\beta}_{i,Q-1}]^T \tag{12}$$

$\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ are $N \times 1$ vectors, and

$$[\boldsymbol{X}\,\boldsymbol{Y}\,\boldsymbol{Z}] = [\boldsymbol{\alpha}_{i0}\,\boldsymbol{\alpha}_{i1}\ \dots\ \boldsymbol{\alpha}_{i,N-1}]^T \tag{13}$$

$\boldsymbol{W}_n$, $1 \leq n \leq 6$, are $Q \times Q$ diagonal matrices whose (t,t)th elements are ${a_x^{(t)}}^2(\tilde{u}_i)$, $a_x^{(t)}(\tilde{u}_i)a_y^{(t)}(\tilde{u}_i)$, $a_x^{(t)}(\tilde{u}_i)a_z^{(t)}(\tilde{u}_i)$, ${a_y^{(t)}}^2(\tilde{u}_i)$, $a_y^{(t)}(\tilde{u}_i)a_z^{(t)}(\tilde{u}_i)$, and ${a_z^{(t)}}^2(\tilde{u}_i)$.

The model surface can then be estimated by computing M such v-curves. Since $\boldsymbol{C}^T\boldsymbol{C}$ is non-singular, the result of the term within the parenthesis on the right side of (11) is a non-singular matrix of size $3N \times 3N$. Computing the inverse of this matrix requires $O(27N^3)$ computation [8]. Thus, the estimation of the v-curve requires $O(27N^3+3NQ)$ computations. Considering the computational complexity for estimating u-curves, the complexity for estimating the surface is $O(3M(9N^3+NQ+PQ))$.

It should be noted that an estimation scheme which first estimates v-curves and then estimates u-curves can be similarly obtained.

4 Hierarchical shape estimation

Data points required for estimating object shape are detected by searching, within automatically controlled search radii, voxels with the strongest edge response along directions normal to the current model surface [13]. The response of the model surface to the data points in the surface fitting is dominated by the number of control points defining the model. A model with a small number of control points is inclined to have a stiff surface in response to the data points, leading to an over-smoothed estimation. As the number of control points increases, the "stiffness" of the model decreases, allowing a response to local details.

Extraction of the object shape is achieved through a hierarchical surface fitting by exploiting the multi-scale representation of the model. First, the model at the lowest scale ($k = 0$) is used to achieve a coarse estimation of the object surface. This will prevent the model surface from being tied to positions where the image is contaminated by noise and produces strong edge responses. The result is then successively refined by increasing the

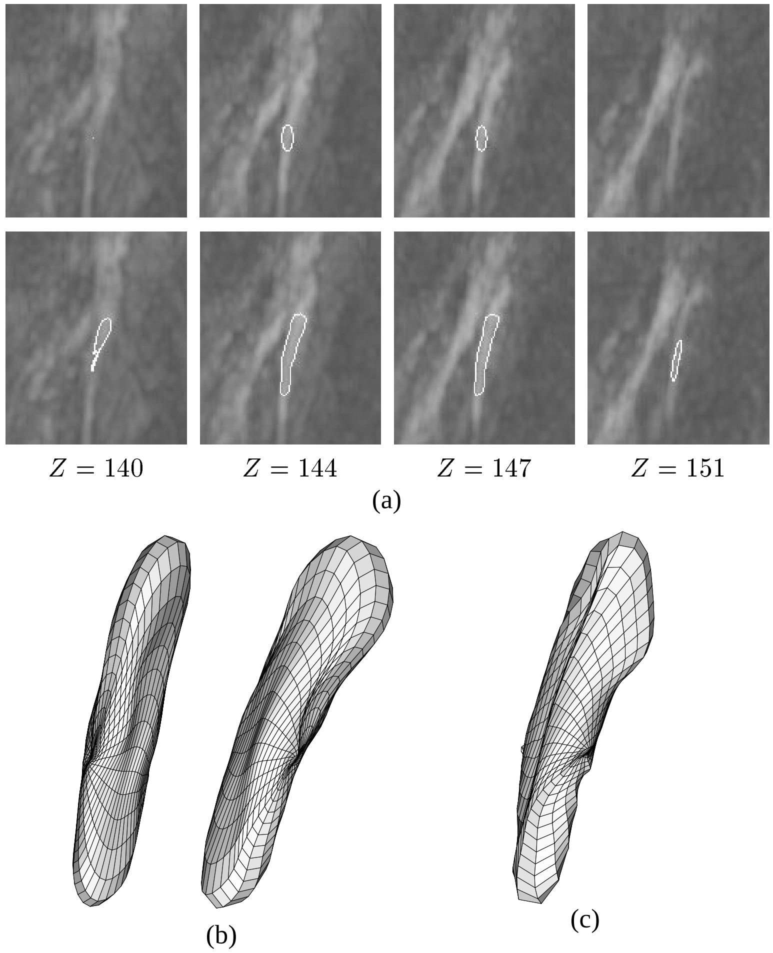


Figure 1: Experiments on a $128 \times 150 \times 201$ 3D seismic (variance filtered) data set. (a) The initial model (the first row) and resultant model (the second row) overlapped to the slices. (b) Two visualisations of the obtained shape model. (c) A shape model obtained by choosing $\lambda = 0$ in the shape estimation.

scale of the model until the desired scale level ($k = K$) is reached. At each scale, the model is iteratively estimated until

$$\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|\boldsymbol{\alpha}_{ij}^{(n+1)} - \boldsymbol{\alpha}_{ij}^{(n)}\| < \epsilon \tag{14}$$

where, $\boldsymbol{\alpha}_{ij}^{(n)}$ are estimated control points from the nth iteration and ϵ is a threshold.

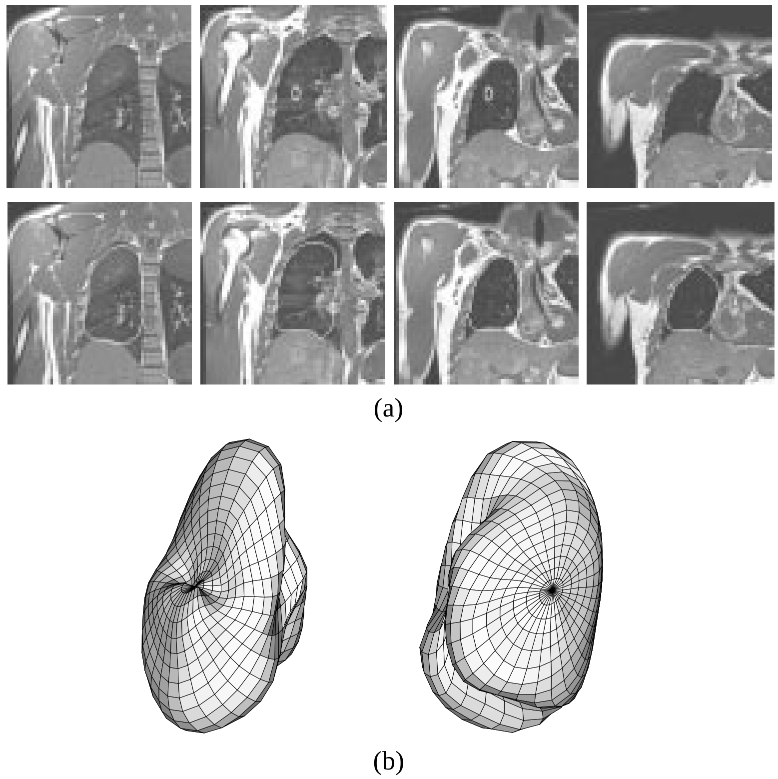
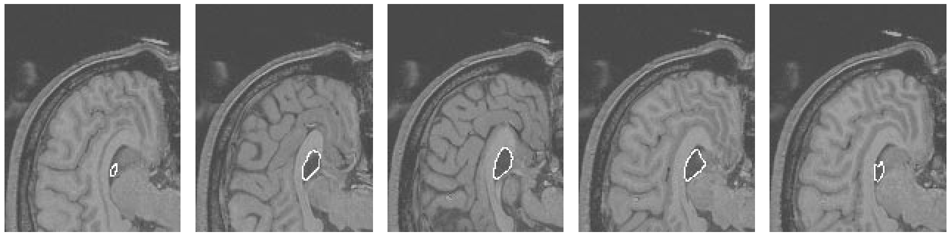


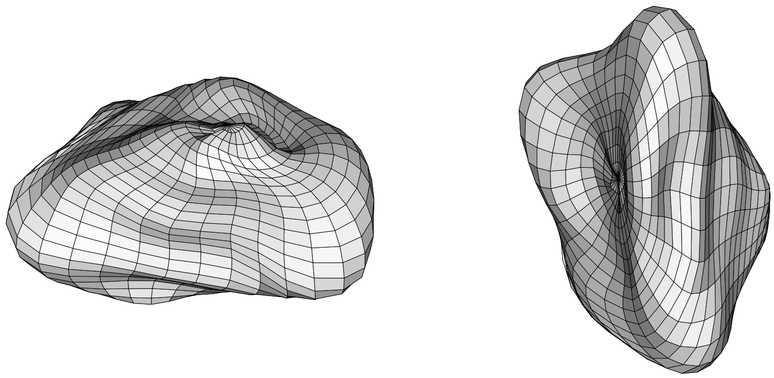
Figure 2: Experiments on a 3D medical image. (a) Initial model (first row) and resultant model (second row) overlapped to X-Y slices of the image. (b) Visualisations of obtained shape model.

5 Experimental results

Experiments have been applied to real 3D images. In all experiments, a model with scale $M(k) = 13+4k$, $N(k) = 14+4k$, and $K = 5$ is used. The number of data points used for estimating the surface is chosen as $P(k) = 15 + 4k$ and $Q(k) = 16 + 4k$. The weighting parameter λ in (9) is chosen as 5.0. Fig. 1 gives the result of an experiment applied to a 3D ($128 \times 150 \times 201$) seismic (variance filtered) data set. Only 4 X-Y slices containing the object region of interest are shown. Fig. 1(a) shows the initial model (an ellipsoid) and the resultant model overlapped to the slices. Fig 1(b) gives two visualisations of the obtained shape model. Fig 1(c) shows the shape model obtained without applying regularisation (i.e. $\lambda = 0$) in the shape estimation. It can be seen that a crease is produced due to the irregularity of coordinate curve net. Fig. 2 shows an experiment applied to a 3D medical image (visual human data). Fig. 2(a) shows the initial model and resultant model overlapped to X-Y slices of the image. The obtained 3D shape model is shown in Fig. 2(b). It can be seen that some parts of the object boundary have not been correctly extracted. This is mainly due to the uneven sampling involved in the generation of the image (the sampling ratio in X, Y, and Z coordinate is 1:1:4). Applying the method to the interpolated image is currently under investigation. Fig. 3 gives the experimental result



(a)



(b)

Figure 3: Experiments on a MRI brain image. (a) Estimated model overlapped to the image. (b) Visualisations of obtained shape model.

applied to a MRI brain image. Fig. 3(a) shows the resultant model overlapped to the image, and Fig. 3(b) shows the obtained shape model.

6 Conclusion

We have presented a new method for extracting the 3D shape of objects from 3D gray level images using a bicubic B-spline surface model. Extraction of object shape is achieved through a hierarchical surface fitting by exploiting the multi-scale representation of the model. A strategy for converting the surface estimation into curve estimations is proposed to achieve an efficient surface fitting and allow the incorporation of constraints applied to the coordinate curve net defining the surface. A regularising component is introduced into the curve estimation to encourage the generation of an orthogonal coordinate curve net and therefore to prevent the creation of unwanted creases. Experiments have been applied to real 3D images, and satisfactory results have been obtained.

Acknowledgement

This work was funded by the EPSRC IMV initiative (grant of GR/K 95987) and carried out in collaboration with Shell KSEPL.

References

[1] R. H. Bartels, J. C. Beatty, and B. A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modelling*. Morgan Kaufmann, 1987.

[2] V. Caselles, R. Kimmmel, G. Sapiro, and C. Sbert. Minimal Surfaces Based Object Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):394–398, April 1997.

[3] I. Cohen, L. D. Cohen, and N. Ayache. Using Deformable Surfaces to Segment 3-D Images and Infer Differential Structures. *Computer Vision, Graphics and Image Processing: Image Understanding*, 56(2):242–263, September 1992.

[4] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.

[5] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape Modelling with Front Propagation: A Level Set Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[6] T. McInerney and D. Terzopoulos. Medical Images Segmentation Using Topologically Adaptable Surfaces. In *Proc. CVMed'97*, Grenoble, France, March 1997.

[7] T. Poggio, V. Torre, and C. Koch. Computational Vision and Regularization Theory. *Nature*, 317:314–319, 1985.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[9] D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill Publishing Company, 2nd edition, 1990.

[10] P. T. Sander and S. W. Zucker. Inferring Surface Trace and Differential Structure from 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):833–854, September 1990.

[11] X. Shen and D. Hogg. 3-D Shape Recovery Using A Deformable Model. *Image and Vision Computing*, 13(5):377–383, 1995.

[12] X. Shen and D. Hogg. Generic 3-D Shape Model: Acquisition and Applications. In V Hlavac and R Sara, editors, *6th International Conference on Computer Analysis of Images and Patterns*, pages 98–105. Springer-Verlag, 1995.

[13] X. Shen and M. Spann. 3D Shape Modelling Using A Multi-scale Surface Model. In *IEEE International Conference on Image Processing*, volume 2, pages 478–481, Santa Barbara, California, U.S.A., 1997.

[14] R. Szeliski and D. Tonnesen. Surface Modelling with Oriented Particle System. *Computer Graphics*, 26(2):185–194, July 1992.

[15] D. Terzopoulos and D. Metaxas. Dynamic 3D Model with Local and Global Deformations: Deformable Superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.