Division of Informatics, University of Edinburgh

Institute of Perception, Action and Behaviour

Constrained Object Reconstruction Incorporating Free-form Surfaces

by

Robert Fisher, Craig Robertson, N. Werghi

Informatics Research Report EDI-INF-RR-0075

Division of Informatics **May 2001**
http://www.informatics.ed.ac.uk/

Constrained Object Reconstruction Incorporating Free-form Surfaces

Robert Fisher, Craig Robertson, N. Werghi

Informatics Research Report EDI-INF-RR-0075

DIVISION *of* INFORMATICS
Institute of Perception, Action and Behaviour

May 2001

Proc. IX Spanish Symposium on Pattern Recognition and Image Analysis, Benicassim, Spain, pp 273-280

Abstract :

Keywords :

Constrained Object Reconstruction Incorporating Free-form Surfaces

N. Werghi †, R.B. Fisher ‡, C. Robertson §
† University of Glasgow. Glasgow G128QQ, UK
‡ University of Edinburgh. Edinburgh, EH1 2QL
§ University of Edinburgh. Edinburgh, EH1 2QL

Abstract

This paper addresses the problem of integrating free-form surfaces in a constrained object reconstruction. The objects of interest here have both simple shaped surfaces (e.g. planes and quadrics) and free-from surfaces. The goal is to ensure a global optimization of the object shape where neighbourhood relationships between free-from surfaces and analytic surfaces are also satisfied. The scheme is validated with experiments made on synthetic and real object

Keywords: 3D range data, Object reconstruction, constrained optimization

1 Introduction

In reverse engineering, having an object reconstruction satisfying geometrical constraints is a fundamental requirement to ensure shape consistency and to reproduce human-intended relationships. In previous work we have developed an approach for imposing geometric constraints on objects having regular surfaces, e.g. planar [9] and second order surfaces [10, 11]. Although these categories covers a large range of manufactured objects, there are still a lot of objects that contain both simple shaped and free-form surfaces. For this category, imposing constraints only on simple surfaces while leaving the free-form surfaces free may cause undesirable effects, such as loss of continuity between the free-form surfaces and simple surfaces, and decrease of the smoothness of the object shape as whole. The aim of this work is to investigate how to impose constraints on the regular surfaces while maintaining their continuity or more generally the shape properties at the point of their intersection with the free-form surfaces which remain constrained by proximity to the data.

2 Constrained object surface reconstruction

Given sets of 3D measurement points representing surfaces belonging to a certain object, we want to estimate the different surface parameters, taking into account the geometric relationships between these surfaces and the specific shapes of surfaces as well. A state vector $\vec{p}$ is associated to the object, which includes all parameters related to the different object surfaces. The vector $\vec{p}$ has to best fit the data while satisfying the constraints. Consider $F(\vec{p})$ to be an objective function defining the relationship between the measured data points and the parameters. Such a function is generally a minimization criterion, for instance, the sum of the least squares residuals. Using an algebraic representation of the surfaces, this function can be put into a quadratic form: $F(\vec{p}) = \vec{p}^T \mathcal{H} \vec{p}$ where $\mathcal{H}$ is a data matrix. The constraints will also to be represented by a quadratic vector function: $\vec{p}^T A \vec{p} + B^T \vec{p} + C$ where A, B and C are respectively appropriate matrix, a vector and a scalar.

Consider $C_k(\vec{p})$, $k = 1..M$, the set of constraint functions defining the geometric constraints where $C_k(\vec{p})$ is a vector function associated with constraint k. The problem can be

then stated as follows: minimize $F(\vec{p})$ subject to the constraints $C_k(\vec{p}) = 0, \quad k = 1..M$ The problem is a constrained optimization problem. We suggested an approach in the framework of sequential unconstrained minimization where we consider the optimization function

$$E(\vec{p}) = F(\vec{p}) + \sum_{k=1}^{M} \lambda_k (C_k(\vec{p}))^2, \ \lambda_k > 0 \tag{1}$$

The scheme is to initialize the parameter vector $\vec{p}$, then increment the set of weighting values λ_k iteratively, at each step (1) is minimized by a standard non-constrained technique, and solution $\vec{p}$ is updated. The process continues until the constraints are satisfied. Figure 1.A illustrates a simplified version of the algorithm where the constraints have been assigned an equal weighting value. For more details about the technique and the convergence conditions of the algorithm we refer the readers to [11].

3 Incorporating B-spline surfaces in constrained object reconstruction

The objective is to reconstruct the object taking into account the geometric relationships (referred as a regular constraints) between the simple surfaces while maintaining the boundary relationships between them and the free-form surfaces, these relationships are referred here by the boundary constraints. One possible approach to solve this problem is to apply the constrained surface fitting scheme of Section 2 on the regular surfaces then modifying afterwards the free-form surfaces so that they join the optimized analytic surfaces with the desired joining constraints. Such modifications can be performed manually using CAD packages. However, this way may affect the smoothness of the surface and introduce wrinkles. It may also lead to solutions with larger overall errors when later imposing the boundary constraints. Besides, achieving this adjustment manually is a tedious task.

Kruth *et al* [2] treats the case of objects containing only free-form surfaces which are joined with boundary conditions, namely, positional and tangential continuity. They adopted a NURBS representation of the surfaces (NURBS is a more elaborated version of the B-spline where the control points coefficients are rational and assigned a weighting values). In their approach, the NURBS surface is modified locally to fit an adjacent surface subject to the boundary conditions. The parameters of the adjacent surface are supposed to be already fixed.

Our work differs from [2] in three aspects. First we treat a category of objects composed of free-form surfaces and simple surfaces. Second, it follows from the first aspect that our constrained optimization scheme considers additionally to the geometric relationships between the simple surfaces, the boundary constraints inferred by the adjacency between the free-form surfaces and the simple surfaces. Third, most importantly, all the surfaces are fitted simultaneously within an automated constrained surface fitting scheme incorporating the two types of constraints. This ensures a global adjustment of all the surfaces' parameters rather than modifying some to fit already fixed ones and leads therefore to a homogeneous adjustment of the object surfaces' shape and spreading of the errors.

The whole constrained reconstruction process is described as follows: First we start from a cloud of points acquired with a laser scanner. The measurement points are treated with the *rangeseg* program [1] to produce two sets of surfaces, namely simple surfaces with known types and free-from surfaces. The simple surfaces include planes and quadric surfaces that can be defined by algebraic functions. The linearity with respect to the parameters of the algebraic functions allows a closed-from solution when fitting these surfaces with a Least Squares technique [8]. Whereas simple surfaces can be represented with algebraic

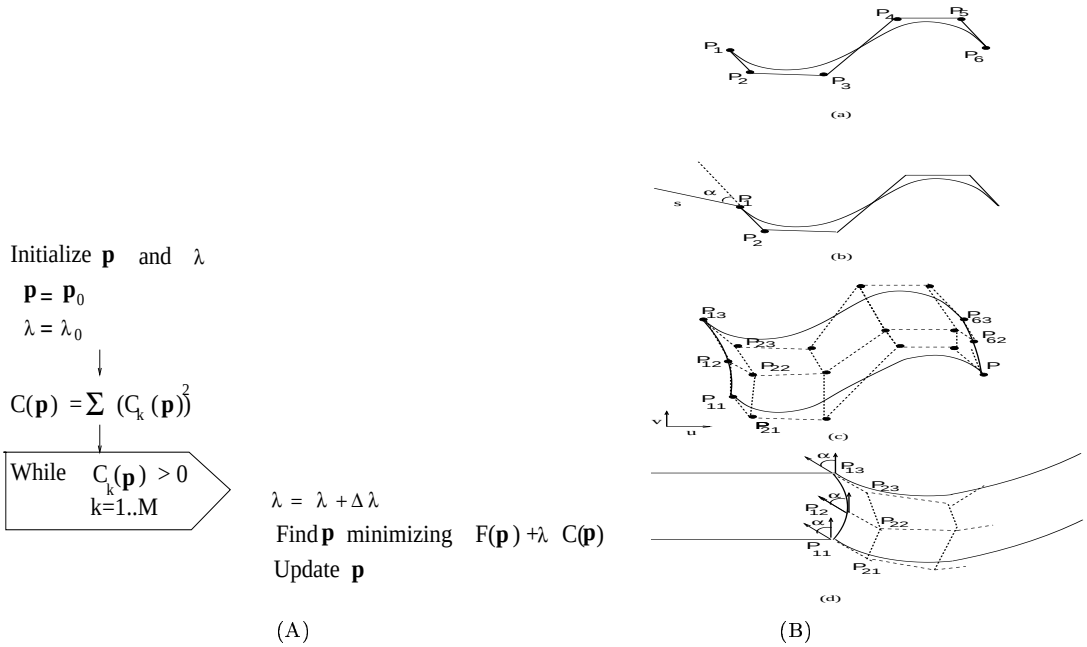

Figure 1: (a) a A B-spline curve with 6 control points. (c) a B-spline surface having 6 and 3 control points in the u, v directions respectively. (b) The end control point $\mathbf{P_1}$ is set on the segment $\mathbf{s}$, so the B-spline curve is continuous with the segment $\mathbf{s}$. By setting the relative orientation between $\mathbf{P_1 P_2}$ and $\mathbf{s}$ to angle α, the B-spline curve meets $\mathbf{s}$ with the same angle. (d) The first two control points in each column (v direction) of the control point matrix make a fixed angle with the plane surface, so the B-spline surface keeps the same orientation along the joining edge.

functions, it is not the case for free-form surfaces. For these surfaces we adopted The B-spline representation. The surfaces are fitted separately and the resulting parameters $\vec{p_{0f}}$ and $\vec{p_{0a}}$ of respectively the free-form surfaces and the simple surfaces are grouped into the parameter vector $\vec{p_0}$ defining an initial parameterization of the object. The regular constraints, the boundary constraints and the initial vector $\vec{p_0}$ constitute the input of the constrained optimization algorithm.

3.1 B-spline surface fitting

A B-spline surface is defined by the following equation:

$$\mathbf{C}(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{m} B_i(u,T_u)B_j(v,T_v)\mathbf{P}_{i,j} \quad (u,v) \in [0,1]$$

where (u,v) are the location parameters that uniquely define the location of the point C on the surface, n and m are the number of control points in the u and v directions, $B_i(u,T_u)$ and $B_j(u,T_v)$ are the normalized B-spline function. $B_i(u,T_u)$ is defined uniquely by the order k_u (degree + 1) of the polynomial and the knot sequence T_u consisting of a set of $n+k_u$ non-decreasing constants $\{u_0,\cdots,u_{n+k_u-1}\}$ non-uniformly sub-dividing the interval $[0,1]$. There is an analogous definition for $B_j(u,T_v)$ with respect to the order k_v and the knot sequence T_u:$\{v_0,\cdots,v_{n+k_v-1}\}$. $(\mathbf{P}_{i,j})$ are the control points controlling the shape of the surface. The control points are organized as an $n*m$ rectangular grid (Fig.1.b) where (n,m) are the numbers of control points in u and v directions respectively. Given a set of measurement points $\mathbf{X}_{k=1}^{N}$, a B-spline surface fitting is performed by determining the set of control points minimizing the least squares function: $\sum_{k=1}^{N}(\sum_{i=1}^{n}\sum_{j=1}^{m} B_i(u_k,T_u)B_j(v_k,T_v)\mathbf{P}_{i,j} - \mathbf{X}_k)^2$ This

is a non-linear problem since (u_k, v_k, T_u, T_v) defining the parametrization of the projection of the data points X_k on the B-spline surface have to be known.

There are two alternatives for the solution, the first is to proceed with a non-linear minimization technique as in [7]. The other alternative is to use an iterative algorithm which consists of: 1) Assign initial parameter (u_k, v_k) to the data points and deduce from them (T_u, T_v). 2) Solve the linear least squares equation (3.1) which by restructuring the control points into a single vector $\vec{p_f} = [P_{11_x}, P_{12_x}, .., P_{nm_x}, P_{11_y}, .., P_{nm_y}.., P_{11_z}, .., P_{nm_z}]^T$ can be rewriten as the vector function: $(H_f\vec{p_f} + \vec{h_f})^2$ where H_f and $\vec{h_f}$ are a matrix and a vector whose elements depend on the normalized B-spline function values and the measurement data. 3) If the fit is not acceptable then recompute the parameters (u_k, v_k, T_u, T_v) for the fixed control points and go to 2).

We have chosen the second alternative. The determination of the projection parameters was performed using the methods described in [3]. Prior to the determination of the projection parameters, the scattered measured data points need to be organized into a regular grid structure. This can be performed semi-automatically, for instance [3] or automatically, for instance [4].

3.2 The boundary constraints on a B-spline surface

The free-form surface has to be continuous with the adjacent simple surface and keep a fixed angle all along its boundary with that surface. To set the equations related to these constraints we need to examine first the properties of B-spline curve defined by $\mathbf{C}(u) = \sum_{i=1}^{n} B_i(u, T_u)\mathbf{P}_i$

Property 1: A B-spline curve interpolates its first and last control points (Figure 1.a) , that is $\mathbf{C}(0) = \mathbf{P}_0$ and $\mathbf{C}(1) = \mathbf{P}_1$

Property 2: At its ends, a B-spline curve is tangent to the segments connecting the first two and, respectively the last two control points, that is:

$$\frac{\partial C(u)}{\partial u}|_{u=0} = K(\mathbf{P_1} - \mathbf{P_2}), \quad \frac{\partial C(u)}{\partial u}|_{u=1} = K'(\mathbf{P_n} - \mathbf{P_{n-1}}) \tag{2}$$

where K and K' are constants. From these properties we deduce that a B-spline curve is continuous with a segment if the end control point is set to lie on that segment and that by setting the segment made by the last two end control points so that it makes given angle with the joined segment, the tangent to the B-spline curve at the joining point will make the same angle with the joined segment (Fig.1.c).

Extending these properties to surfaces leads to saying that the first order continuity of a B-spline surface with an simple surface is satisfied by setting the control points of the end row or column (depending on which side the regular surface meets the B-spline surface) to belong to the simple surface. Similarly making the tangent plane of the B-spline surface have a fixed angle with the tangent plane of the regular surface is imposed by setting each of the segments formed by the two last end control points of each row (or column) so that they make the same angle with the tangent plane of the regular surface at their joining edge (Fig.1.d).

Consider the case of a plane as the simple surface and consider a B-spline surface having n and m control points in the u and v direction respectively, and which joins a plane surface having as orientation the normal $\vec{n}$ and d as offset parameter. The B-spline surface joins the plane so that their orientations at the edge make an angle α. Let us suppose without loss of generality that the B-spline surface joins the plane along the v direction (as in Fig.1.d). Then the continuity constraints and the orientation constraints on the B-spline surface at its boundary with the plane can be defined by the equations (3), (4) and (5)

$$\vec{n}^T\mathbf{P_{1,j}} + d = 0, \quad j = 1..m; \quad (3)$$

$$\vec{n}^T(\mathbf{P_{1,j}} - \mathbf{P_{2,j}}) = K_j cos(\alpha), \quad j = 1..m; \quad (4)$$

$$(\mathbf{P_{1,j}} - \mathbf{P_{2,j}})^T(\mathbf{P_{1,j}} - \mathbf{P_{2,j}}) = K_j^2 \quad (5)$$

$$\vec{p}^T A_j \vec{p} + \vec{B}^T \vec{p} = 0, \quad j = 1..m; \quad (6)$$

$$\vec{p}^T B_j \vec{p} - K_j cos(\alpha) = 0, \quad j = 1..m; \quad (7)$$

$$\vec{p}^T C_j \vec{p} - K_j = 0, \quad j = 1..m; \quad (8)$$

The equation (5) constrains the length of each segment $P_{1,j}P_{2,j}$ to a fixed value. We added this constraint in order to simplify the implementation of the angle constraint. From these relations we extract the quadratic vector functions (6),(7) and (8). where $\vec{p}$ is the whole parameter vector $\vec{p}$ grouping the B-spline surface parameters $\vec{p_f}$ and the parameters of the simple surfaces $\vec{p_a}$, A_j, B, B_j and C_j are appropriate matrices and vectors. Under this form the integration of the boundary constraint into the constrained surface optimization algorithm described in section 2 is straightforward.

3.3 Recapitulation of the algorithm

The constrained reconstruction algorithm for objects having simple and free-form surfaces contains the following steps: 1) Fit the simple surfaces (planes and quadrics) and compute the related initial parameters $p_{a_0}^{\rightarrow}$ using for instance the technique in [8]. 2) Fit the B-spline surface using the method mentioned in Section 3.1 and assign to the B-spline surface a parameter vector $p_{f_0}^{\rightarrow}$ initialized by the control points coordinates. 3) Set the object parameter vector to $\vec{p}$ and initialise it to the vector $\vec{p_0}$ grouping $p_{a_0}^{\rightarrow}$ and $p_{f_0}^{\rightarrow}$ and define the objective function as $F(\vec{p}) = \vec{p}^T\mathcal{H}_a\vec{p} + (\mathcal{H}_f\vec{p} + \vec{h_f})^2$ where $\mathcal{H}_a, \mathcal{H}_f$ and $\vec{h_f}$ are matrices and vector deduced from the initial surface fitting of the simple and the B-spline surfaces. 4) Set the equations of the regular constraints (**??**) and the boundary constraints (6) and (7). 5) Apply the constrained optimization algorithm and determine $\vec{p}$.

4 Experiments

The experiments were carried out on synthetic and real parts containing planar and free-form surfaces. The aim was to study the impact of the constrained reconstruction on the free-form surfaces in terms of satisfaction to the boundary constraints imposed by the geometric relationships with adjacent planar surfaces. Because of space limit only one object will be presented in this paper.

The experiment considered a set of surfaces from the object shown in Figure.2.(a). The free-form surface S_f is bounded by the three planes. The number of control points assigned to S_f are $n = m = 5$ in the u and v directions respectively. The plane S_1 is orthogonal to S_2 and S_3, these two last make a fixed angle between their orientation. The free-form surface has then a positional continuity constraint with each plane. This is encoded by the equations (9), (10) and (11)

$$\sum_{j=1}^{m}(\vec{p}^T A_{(S_f S_1)_j}\vec{p} + B_{S_1}^T\vec{p})^2 = 0 \quad (9)$$

$$\sum_{j=1}^{n}(\vec{p}^T A_{(S_f S_2)_j}\vec{p} + B_{S_2}^T\vec{p})^2 = 0 \quad (10)$$

$$\sum_{j=1}^{n}(\vec{p}^T A_{(S_f S_3)_j}\vec{p} + B_{S_3}^T\vec{p})^2 = 0 \quad (11)$$

$$\sum_{j=1}^{m}(\vec{p}^T B_{(S_f S_1)_j}\vec{p} - K_{(S_1)_j})^2 = 0 \quad (12)$$

$$\sum_{j=1}^{n}(\vec{p}^T B_{(S_f S_2)_j}\vec{p})^2 = 0 \quad (13)$$

$$\sum_{j=1}^{n}(\vec{p}^T B_{(S_f S_3)_j}\vec{p})^2 = 0 \quad (14)$$

The equations related to the orthogonality of S_f to S_1 and the tangency of S_f to S_2 and S_3 are obtained by setting the angle α in (4) to 0 and $\pi/2$, we get the equations (12), (13), (14). We notice that the value $\pi/2$ of the angle α makes the norm value K_j not involved

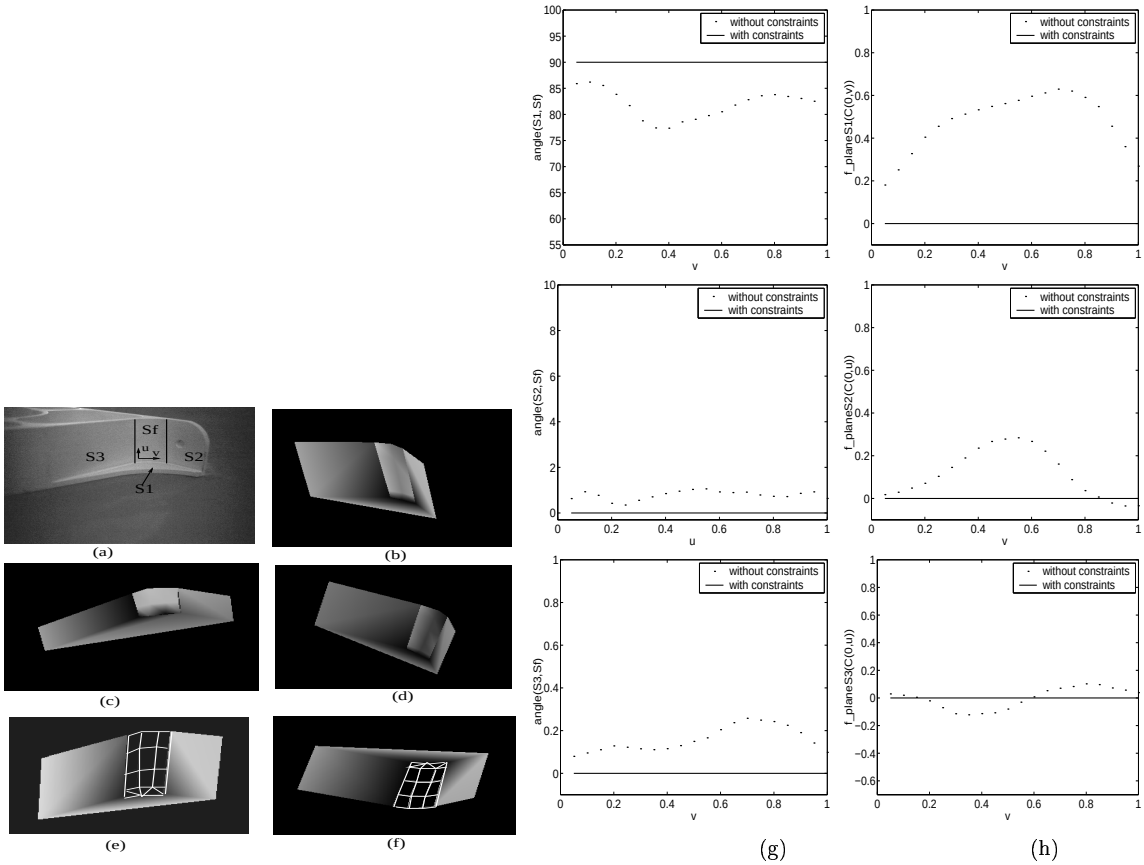

Figure 2: (a) view of the object. (b)..(f), views of the reconstructed surfaces (g)angle between the free-form surface normal and the plane normal of the adjacent surfaces. (h) plane function values at the edge points of the free-from surfaces S_f

in the tangency constraint between S_f and both S_2 and S_3. Therfore there is no need to impose the norm constraint (5) on the control points of the B-spline surface associated to S_f and located on the side of S_2 and S_3. For the control points on the side of S_1, the related norm constraints is $\sum_{j=1}^{n}(\vec{p}^T C_{(S_f S_2)_j}\vec{p} - K_{(S_1)_j})^2 = 0$

Figure 2(b..f) shows some views of the reconstructed surfaces. The satisfaction of the continuity and the tangency of the B-splines surfaces with the bounding plane surfaces is illustrated in Figure 2(g,f).

5 Conclusion

In this paper we have described an approach for integrating free-form surfaces and their constrained relationships in a surface recovery scheme incorporating geometric constraints. The global aspect of the surface optimization produces a free-form surface approximation satisfying the constraints while having the desired smoothness. The experiments were carried out with planes as bounding surfaces, but the quadric surface case can be also used in this approach. The set of constraints between the free-form surfaces and the simple surfaces can be further extended to cover constraints other than the boundary ones. For example, a free-form surface can be imposed to have its local orientation (the local orientation at a given point of a free-from surface can be defined by the normal of the plane tangent to

the free-from surface at that point) orthogonal to (or set to a fixed angle with) an simple surface. This problem will be addressed in a future work. The constrained optimization was implemented in C++ language. The program was run on a 200 MHz Sun Ultrasparc workstation. The optimization time was about 17 min for the used object. Further improvement can be brought to the algorithm: at this stage, only the control points are updated by the optimization process. It is worth to investigating how to incorporate the other parameters of the B-spline surface, namely, the knots sequences, T_u and T_v.

References

[1] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, R. Fisher, "An Experimental Comparison of Range Segmentation Algorithms", *IEEE Trans. PAMI*, Vol.18, No.7, 673–689, July 1996.

[2] J.P. Kruth, A. Kersrens, "Reverse engineering modelling of free-from surfaces from point clouds subject to boundary conditions", *Journal of Materials Processing Technology*, No. 76, pp.120-127, 1998.

[3] W. Ma, J.P. Kruth, "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces", *Computer-Aided Design*, Vol.27, No.9, pp.663-675, 1995.

[4] I.K. Park, I.D. Yun, S.U. Lee, "Constructing NURBS Surface Model from Scattered and Unorganized Range Data", *Proc. 2nd International Conference on 3-D Digital Imaging and Modeling* . Ottawa, Canada, October 1999.

[5] L. Piegl, "On NURBS, A Survey", *Computer Graphics and Applications* , Vol.11, No.1, pp.35-71, January 1999.

[6] L. Piegl, W. Tiller, *The NURBS Book* , Springer-Verlag, 1997.

[7] B. Sarkar, C.H. Menq, "Parameter optimization in approximating curves and surfaces to measurement data", *Computer Aided Geometric Design*, Vol.8, pp. 267-290, 1991.

[8] G. Taubin, "Estimation of Planar Curves, Surfaces and Non planar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation", *IEEE Trans. PAMI* , Vol.13, No.11, November 1991

[9] N. Werghi, R.B.Fisher, C.Robertson, A.Ashbrook, "Improving Model Shape Acquisition by Incorporating Geoemetric constraints" *Proc. BMVC* , pp.530-539, Essex, September 1997.

[10] N.Werghi, R.B.Fisher, A.Ashbrook, C.Robertson, "Modelling Objects Having Quadric Surfaces Incorporating Geometric Constraints", *Proc. ECCV'98*, pp.185-201, Friburg, Germany, June 1998.

[11] N. Werghi, R.B. Fisher, C.Robertson, A. Ashbrook, "Object Modelling by Incorporating Geometric Constraints", *Computer-Aided Design* , Vol.3, No.6, pp.363-399, May 1999.