ELSEVIER

Computer-Aided Design 34 (2002) 173–183

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Measurement-based modification of NURBS surfaces

Djordje Brujic, Mihailo Ristic*, Iain Ainsworth

Imperial College of Science, Technology and Medicine, Department of Mechanical Engineering, London SW7 2BX, UK

Received 27 September 2000; revised 15 December 2000; accepted 5 January 2001

Abstract

A frequent requirement in computer aided design and manufacture is to update or refine an existing CAD model using measured data. Least squares surface fitting is known to suffer from stability problems, caused by an insufficient measurement density in some regions. This is particularly evident in situations involving local surface updating and when knot insertion is applied for local surface refinement. This paper presents a new method to update the CAD model consisting of NURBS surfaces, trimmed or untrimmed, based on a set of unorganised measured points in three-dimensional space. The proposed method overcomes the fundamental problem of singular or ill-conditioned matrices resulting from incomplete data sets. This was achieved by introducing additional fitting criteria in the minimisation functional, which constrain the fitted surface in the regions with insufficient number of data points. Two main benefits were realised by this approach. First, local surface updating can be performed by treating the surface as a whole, without the need to specially identify the regions with insufficient data, nor to re-measure those regions. Second, the quality of the unmeasured regions may be controlled to suit specific needs. The results were found to be highly encouraging and the method was found to be especially useful in situations involving knot insertion and large surface deformations.

Keywords: Surface fitting; Least Squares; NURBs; Regularisation

1. Introduction

In today's industrial practice many products are designed using free-form surfaces. The shape of the object is dictated by numerous considerations relating to the product function, its aesthetic effect and the ease of manufacture. Examples can be readily found among the automotive products (exterior body panels, interior car trim), aerospace (airfoil shapes) and also in a wide range of consumer goods. The design and development process of such products is often an iterative one. It involves manufacture of a series of prototypes or models, which are progressively updated and refined until the design objective has been finally achieved [14].

In this process, the development of the desired physical object and the development of its CAD model mostly go hand-in-hand. In the case of consumer goods, for example, it is typical that the designer (stylist) creates a clay model of the product, which is then digitised using a contact or a non-contact measuring system to produce the data from which the CAD model can be generated. Design iterations then involve repeated CNC machining of new clay models from CAD, manual model modification, digitising and CAD model updating. In the case of engineering components, such as aeroengine turbine blades, it is important to correlate the shape of the nominal CAD model and the performance predicted by simulations with the shape of the manufactured part and its actual performance evaluated through experiments. In this situation it is highly desirable to update the CAD model using measurements, such that it provides a faithful representation of the realised (manufactured) shape.

The main modelling entity in the modern CAD systems are NURBS. Trimmed NURBS are also widely used, because they largely overcome the limitations imposed by the strictly rectangular domain of tensor product surfaces and provide additional flexibility for the designer. With trimmed NURBS however, only the untrimmed portion of a CAD surface exists in the physical part and is available for measurement. Thus if the CAD surface is to be updated using measured data then the updating process is a localised one. Furthermore, it is usually more practical to digitise only those regions of the object that are of interest and these may represent only a portion of a CAD entity. Surface updating can be seen as an estimation of the shape from an incomplete data set and an approximate model [7].

A number of geometric modelling techniques have been

* Corresponding author. Tel.: +44-20-7594-7048; fax: +44-20-7584-7239.

E-mail address: m.ristic@ic.ac.uk (M. Ristic).

0010-4485/01/$ - see front matter
PII: S0010-4485(01)00060-4

proposed for modifying shape. At the lowest level, the designer repositions the control points, adjusts weights and modifies the knot vectors. More sophisticated methods for modifying shape are presented in [15], such as warping, flattening, bending and constraint based modification. It is also important to remember that knot insertion is an invaluable tool whenever shape modification is to be applied. It introduces additional control points in order to provide the required local flexibility for achieving the desired shape.

In the case of scattered data fitting, deformable surfaces may serve as effective fitting tools. Terzopoulos and Metaxas [24] proposed a computational physics framework for shape modelling in which globally deformed superquadrics model coarse shape and local deformations add fine detail. Further, Qin and Terzopoulos [17] proposed using dynamic NURBS swung surfaces, but computational overheads mean that this method cannot be easily applied to problems with many control and data points. Also, blended deformable models [4,23] and generative modelling of Ramamoorthi and Arvo [18] might be seen as tools for surface fitting. They use the measured data cloud with a user defined class of models which are a generalisation of swept surfaces. A shortcoming of this approach is that it is limited in representing local detail.

Curve and surface fitting using least squares is a general technique and a good introduction is provided in [9]. In terms of CAD surface updating using least squares, the main work was done by Ma, Kruth and He [10–12], who present methods for fitting of B-splines, or NURBS, to unorganised points. It is well known that methods based on least squares fitting have a potential problem with rank deficient matrices which is a direct result of an insufficient coverage of certain regions by the measurements. In addition, the measurement distribution may be such that the least squares problem is non-singular, but it is ill-conditioned due to small parameter values [2]. The main idea proposed by Ma and He [10] is to avoid the singularity problem by excluding from fitting those control points for which the position is not defined by the data. These are identified in a pre-processing step and the relevant matrices are re-arranged to allow only a subset of the control points to be affected by the fitting. After applying this solution it is still possible that the system is ill-conditioned due to sparsity of data in some regions, so it was suggested that those regions should be re-measured. However, this is not very practical for shape refinement since fitting, knot insertion and measuring may need to be reiterated a number of times.

The work presented in this paper addresses the problems with ill-conditioned and singular matrices that arise during least squares fitting, using NURBS base surfaces and measured data. The proposed method also eliminates the need to identify those regions that are unaffected by the measurements in order to exclude them from fitting. Instead, the surface is always treated as a whole, irrespective of the measurement distribution, and there is no explicit distinction made between local and global modification.

The paper is organised as follows. In order to avoid notational confusion, the next section will briefly define NURBS surfaces, while Section 3 describes least squares fitting applied to NURBS and discusses the problem of rank deficiency. Section 4 describes regularisation as a method to overcome the rank deficiency problems and presents the solution proposed in this paper. Section 5 analyses the performance of the implemented method and provides some illustrative practical examples.

2. NURBS Definition

A NURBS surface of degree p in the u direction and degree q in the v direction is a bivariate vector-valued piecewise rational function of the form:

$$\mathbf{S}(u,v) = \big(x_s(u,v), y_s(u,v), z_s(u,v)\big) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \quad 0 \le u,v \le 1 \tag{1}$$

where the control points $\{\mathbf{P}_{i,j}\}$form a bi-directional control net with n points in the u direction and m points in the v direction ($n \times m = N$), while $\{w_{i,j}\}$ are control point weights. The functions $\{N_{i,p}(u)\}$ and $\{N_{j,q}(v)\}$ are the non-rational B-spline basis functions defined on the knot vectors

$$\begin{aligned} U &= \{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{r-p-1},\underbrace{1,\ldots,1}_{p+1}\} \\ V &= \{\underbrace{0,\ldots,0}_{q+1}, v_{q+1},\ldots,v_{s-q-1},\underbrace{1,\ldots,1}_{q+1}\} \end{aligned} \tag{2}$$

where $r =$ (number of knots in the u-direction) $-$ 1 and $s =$ (number of knots in the v-direction) $-$ 1.

By introducing the piecewise rational basis functions

$$R_{i,j}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m} N_{k,p}(u)N_{l,q}(v)w_{k,l}} \tag{3}$$

the surface Eq. (1) can be written as:

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)\mathbf{P}_{i,j} \tag{4}$$

Eqs. (1)–(4) define the evaluation of a point on a NURBS surface, the basic implementation of which is outlined in [15].

2.1. Greville abscissae

By setting

$$u_i^* = \frac{1}{p}\left(u_i + \cdots + u_{i+p}\right), v_j^* = \frac{1}{q}\left(v_j + \cdots + v_{j+q}\right) \quad (5)$$

one can associate a point in the surface $\mathbf{S}(u_i^*, v_j^*)$ with each control point $\mathbf{P}_{i,j}$ and link the geometry to the parametric values. This is the generalised association in the Schoenberg variation diminishing spline. The values u_i^*, v_j^* are called the *nodes* of the knot vectors [8] or Greville abscissae [6]. Generally, $\mathbf{S}(u_i^*, v_j^*)$ may be regarded as the points on the surface where their corresponding control points exercise the most influence [6,13].

3. NURBS surface fitting

Least squares fitting of NURBS through a set of points would lead to a non-linear optimisation problem if the unknowns are the control points, parameters (u, v), knots, and the weights. A method for adjusting the weights was proposed by Ma and Kruth [12]. However, the non-linear nature of the problem can be avoided and the optimisation can be greatly simplified, if the weights and the knot vector are set a priori. Since the nominal CAD model is already available as a good approximation of the actual object shape, we propose using the weights and knot vectors obtained from the nominal model and then optimising only the positions of control points.. If required, the accuracy of the fitted surface can be further increased using knot insertion. This approach has been shown to achieve both accuracy and reasonable computational speed [20]. The following explanation of NURBS surface fitting is an extension of that presented by [15] for use with NURBS curves.

By denoting the measured points as $\mathbf{Q}_1,\ldots,\mathbf{Q}_M$, we set up and solve the linear least squares problem for the unknown control points. The functional to be minimised is:

$$f = \sum_{k=1}^{M} |\mathbf{Q}_k - \mathbf{S}(u_k, v_k)|^2 \quad (6)$$

where u_k and v_k are the parametric co-ordinates corresponding to each measured point. The assignment of these coordinates is crucial because parameterisation has a strong effect on the shape of the fitted surface. A number of methods to parameterise measured points have been published, but the majority of this work makes the assumption that the data is ordered. Since our work aims to deal with both ordered and unordered data, an alternative method was found following the suggestion by Ma and Kruth [11], where the parameterisation can be achieved by projecting the points onto a base surface, from which the u_k and v_k values are obtained. In this work, the required base surface is readily provided by the entities of the existing CAD model and the required parameterisation is obtained as a result of registration.

Registration is the process of establishing point correspondences between the data and the model and it may also involve alignment of the two entities. The need for alignment arises from the fact that the data and the model are usually provided with respect to different co-ordinate systems. Following our experience with registration of NURBS models [19,21], we adopted the Iterative Closest Point (ICP) registration method [1]. As the name suggests, the ICP method minimises at each iteration step the collective square distances between the measured points and their closest points on the surface, defined by the functional:

$$f = \sum_{i=1}^{M} |\mathbf{S}_i - \mathbf{R}\mathbf{Q}_i - \mathbf{t}|^2 \quad (7)$$

where $\mathbf{t}$ is the translation matrix, $\mathbf{R}$ is the rotation matrix, $\mathbf{Q}_i$ is the ith measurement point and $\mathbf{S}_i$ is the closest point on the surface. In our implementation the search for the nearest point is performed using multidimensional simplex method [21].

Following this, the least squares fitting problem involves solving the set of normal equations

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{a} = \mathbf{A}^{\mathrm{T}}\mathbf{b} \quad (8)$$

where $\mathbf{A}$ is $M \times N$ matrix:

$$\mathbf{A} = \begin{bmatrix} R_{0,0}(u_0, v_0) & \ldots & \ldots & R_{n,m}(u_0, v_0) \\ R_{0,0}(u_1, v_1) & & & R_{n,m}(u_1, v_1) \\ \vdots & & & \vdots \\ R_{0,0}(u_M, v_M) & \ldots & \ldots & R_{n,m}(u_M, v_M) \end{bmatrix}$$

and

$$\mathbf{a} = \begin{bmatrix} \mathbf{P}_{0,0} & \ldots & \ldots & \mathbf{P}_{n,m} \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{Q}_1 & \ldots & \ldots & \mathbf{Q}_M \end{bmatrix}^{\mathrm{T}}$$

The system of Eq. (8) has to be solved a total of three times in order to calculate $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$ for all $\mathbf{P}_{i,j}$. Here we assume that matrix $\mathbf{A}$ is full column rank and defer treatment of rank deficient problems to Section 3.1. Eq. (8) can be solved using a number of methods. Cholesky factorisation [2] belongs to the group of direct methods and gives satisfactory results for problems of reasonable size. However, surface fitting problems can be large, sometimes involving over a million data points and modelling entities having over a thousand of control points, making the memory storage requirements imposed by the direct methods prohibitive. This is because the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is sparse, but its sparsity is destroyed by the Cholesky factorisation as it produces the triangular matrix. Iterative methods on the other hand, start from an initial approximation, which is successively improved until a sufficiently accurate solution is obtained. Importantly, basic iterative methods work only

with the original form of the matrix $\mathbf{A}^T\mathbf{A}$ and can readily exploit its sparse structure by storing only its non-zero elements. Furthermore, since $\mathbf{A}^T\mathbf{A}$ is involved only in terms of matrix by vector products, the rows of this matrix may be generated when required so the storage requirements can be reduced even further, but at the expense of additional computations. Thus for very large problems iterative methods may be the only feasible ones and they are used most often for the solution of very large sparse systems. Finally, we note that Singular Value Decomposition (SVD) may be an attractive solution method, especially in view of its stability in the presence of rank deficiency of $\mathbf{A}$, but SVD cannot take advantage of the matrix sparsity in order to speed up the calculations [5].

For the purposes of the work presented in this paper, we implemented the iterative method of Gauss–Seidel, as well as Cholesky decomposition.

3.1. Rank deficiency and ill-conditioning of the least squares problem

It is quite easy for the matrix $\mathbf{A}^T\mathbf{A}$ in the normal equations to be rank deficient. As the system increases it is also likely that the set of equations will become ill-conditioned. The problem of rank deficiency arises from the local character of the basis functions and its detailed presentation is provided in [5]. For the univariate case (curves), rank deficiency can be readily detected by examining validity of the Shoenberg–Whitney conditions [5], but such simple test does not exist for the bivariate case (surfaces). Instead, De Boor [3] shows that $\mathbf{A}^T\mathbf{A}$ is *positive definite* and *well-conditioned* if there is at least one data point assigned to every knot span. Since this condition can be readily examined, it is most often taken as the criterion to set up a well-posed system. In the context of NURBS surface updating however, it is difficult to guarantee this condition and the main reasons for this are:

- incomplete data sets due to inaccessibility (most often close to the edges);
- incomplete data sets due to the use of trimmed NURBS in modelling;
- knot insertion;
- re-parameterisation.

Some of these problems may be illustrated through the example in Fig. 1, when a base surface is updated using unevenly distributed measurements. The measurements, Fig. 1(a), are considerably more dense in the central region which involves a large deformation, while they are relatively sparse in the outer regions which are kept flat. The results of surface fitting in Fig. 1(b) show that the available number of control points provide insufficient flexibility of the surface, causing two distinct problems. First, in the central region, the fitted surface is incapable of following the prescribed shape with sufficient accuracy. Second, the strong pull exercised by the dense measurements in the central region causes waviness of the sparsely measured outer regions, which appears in order to accommodate the tight fitting of the central region. This is a clear case for employing knot insertion which, as Fig. 1(c) shows, provides additional flexibility and produces considerably better overall results. Repeated knot insertion provides an even greater surface flexibility and better corresponding fitting accuracy in the central region. Unfortunately, this quickly leads to a situation shown in Figs. 1(d) and (e), when empty knot segments start to appear in the outer regions, causing ill-conditioning or rank deficiency of the $\mathbf{A}^T\mathbf{A}$ matrix, bringing instability into the system.

A difficult problem arises when the system is ill-conditioned, in which case the diagonal elements are small. One way of dealing with this problem, as suggested by Dierckx [5], is to compare the diagonal elements to a pre-set threshold value ϵ and if smaller, then to set those elements to zero. However, this is not an entirely satisfactory solution because an appropriate threshold value should be determined by considering the relative magnitudes of the coefficients in $\mathbf{A}^T\mathbf{A}$ and this is difficult. Also, it is generally too crude to simply exclude control points from the updating scheme on this basis.

4. Regularisation

In order to overcome the problems outlined in the previous section, it is necessary to regularise the linear system. The principle of regularisation [2] is the prescription to expand the functional to be minimised as follows:

$$f(\mathbf{a}) = \|\mathbf{A}\mathbf{a} - \mathbf{b}\|_2^2 + \lambda^2\|\mathbf{C}\mathbf{a} - \mathbf{d}\|_2^2 \tag{9}$$

where $\|\mathbf{C}\mathbf{a} - \mathbf{d}\|_2^2$ defines some other fitting criterion as a *constraint* and the constant $\lambda > 0$ provides a trade-off between the two criteria. This is equivalent to the least squares problem:

$$\min_a \left\| \begin{pmatrix} \mathbf{A} \\ \lambda\mathbf{C} \end{pmatrix} \mathbf{a} - \begin{pmatrix} \mathbf{b} \\ \lambda\mathbf{d} \end{pmatrix} \right\|_2 \tag{10}$$

A necessary condition for a minimum is that the gradient $\partial f/\partial a = 0$, which then gives a set of generalised normal equations

$$\left(\mathbf{A}^T\mathbf{A} + \lambda^2\mathbf{C}^T\mathbf{C}\right)\mathbf{a} = \mathbf{A}^T\mathbf{b} + \lambda\mathbf{C}^T\mathbf{d} \tag{11}$$

The dimensions of vectors and matrices in Eqs. (8) and (11) are identical and both are solved using identical methods. Furthermore, the minimising functional and the generalised normal equations may be expanded in this fashion to include additional criteria. Importantly, when a quadratic minimisation principle is combined with a quadratic constraint, and both are positive, only one of the two need be non-degenerate in order to make the overall problem well-posed. It is also worth noting that the two parts of the minimisation will have

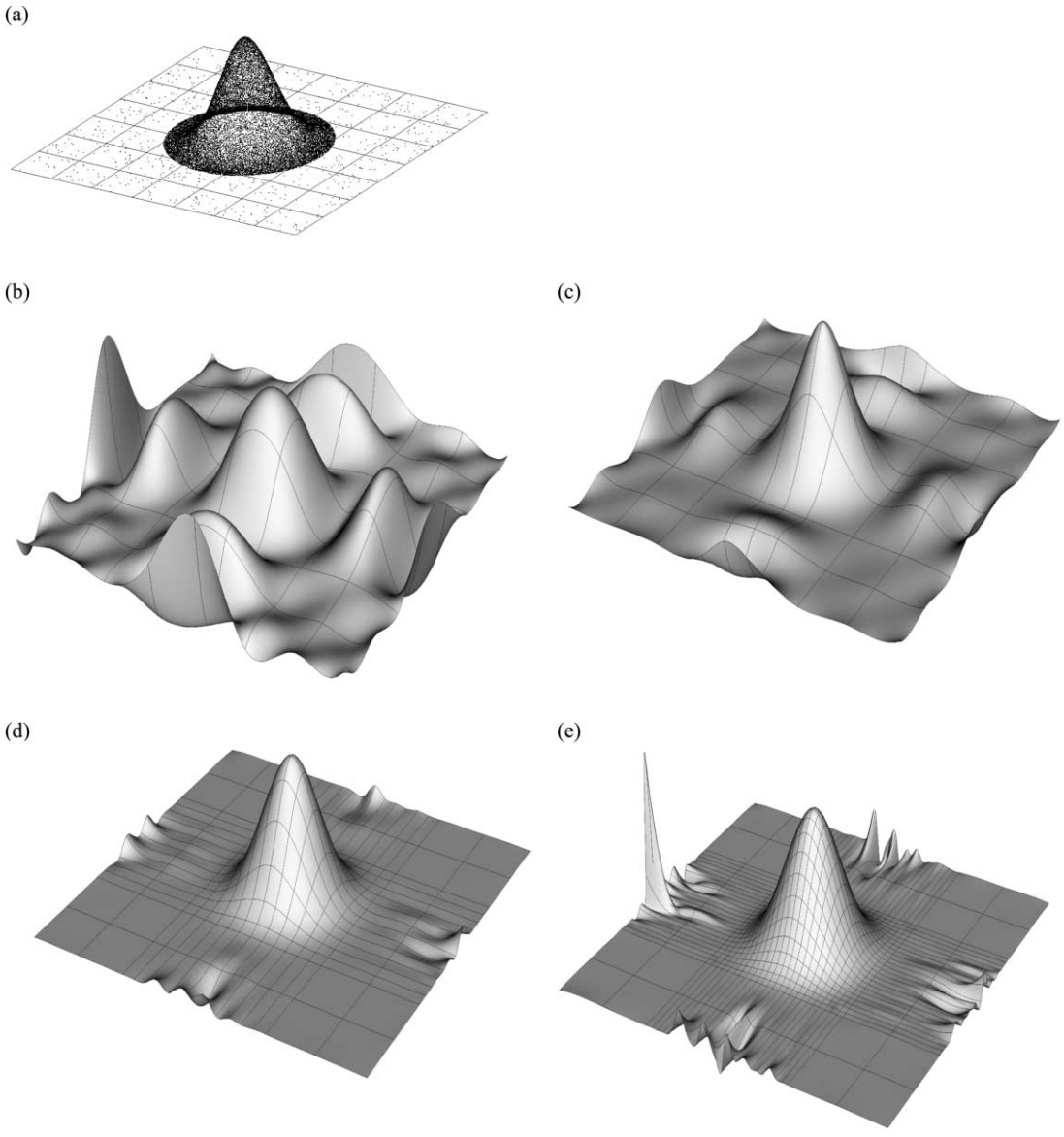

Fig. 1. Surface fitting and knot insertion. (a) Initial surface and unevenly distributed measurement cloud; (b) fitted surface; (c) fitting results are improved following knot insertion; (d) repeated knot insertion improves fitting results but instability starts to appear at the sparsely measured outer regions; (e) further knot insertion causes further deterioration in the regions with sparse data.

comparable weights [16] by choosing:

$$\lambda^2 = \frac{Tr\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)}{Tr\left(\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)} \tag{12}$$

Thus in the knowledge that matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is possibly degenerate, a new criterion and a corresponding weight λ may be chosen to set up a well-posed problem. A number of such regularisation schemes have been proposed. One standard solution for dealing with singular systems is provided by the minimum norm criterion, defined by

$$\min_a \|\mathbf{a}\|_2, \quad \mathbf{Na} = \mathbf{c} \tag{13}$$

Indeed SVD is a special case of minimum norm regularisation [16], but it was found that in practice it does not give satisfactory solution for B-spline surface fitting, confirming the reports in [5]. As an alternative, Schmidth [22] proposes to couple neighbouring B-spline coefficients which are not properly defined by the data. This kind of regularisation is helpful in cases where the knot distribution is such that no explicit gap, but less data than local functions, appear over some subregion. Unfortunately, this regularisation may also change the shape of the surface in a strange manner.

4.1. Proposed solution

In developing the solution for the regularisation problem, it was noted that when the system becomes unstable, the control points associated with the areas with insufficient

data move in an uncontrollable fashion, away from the surface. Our main idea is based on the fact that control points do approximate the surface and it seems natural to keep them as close to the surface as possible by introducing an additional criterion. We therefore suggest minimising the sum of the squared distances between the control points and their corresponding points on the fitted surface. We expect this criterion to smooth the surface and to involve an equivalent of energy minimisation.

Mathematically, we expand the functional of Eq. (6) to include an additional '*criterion*', as follows:

$$f = \sum_{k=0}^{M-1} |\mathbf{Q}_k - \mathbf{S}(u_k, v_k)|^2 + \alpha \sum_{i=0}^{n} \sum_{j=0}^{m} \left|\mathbf{P}_{i,j} - \mathbf{S}(u_{i,j}, v_{i,j})\right|^2 \quad (14)$$

where $\mathbf{P}_{i,\,j}$ are the control points and $\mathbf{S}(u_{i,j}, v_{i,j})$ are their corresponding points on the surface, while coefficient $\alpha \geq 0$ provides the required trade-off flexibility

Naturally, the question arises as to how to define the corresponding surface points. We adopted a solution using Greville abscissae (Section 2.1), because they provide the most regular matrix, as they are obtained using knot averaging (Eq. (5)). For example, in the case of a cubic B-spline, no two adjacent knot segments will be without Greville points.

After implementing this idea, we conducted extensive experiments with a variety of shapes and concluded that inclusion of the *α-criterion* is indeed highly beneficial, especially in situations involving large deformations of the original model. The experiments have also shown that the *α-criterion* can produce the effect of flattening the fitted surface, especially in the regions where the data points are sparse and the control points are relatively far from the base surface. This effect is in fact often desirable. The regions with no, or with few, data points are generally re-shaped to accommodate the deformation of the regions with dense data, such that a gentle transition between the two regions is obtained. Furthermore, the flattening effect may be controlled and significantly reduced by knot insertion, making the control polygon approximate the surface more closely.

However, as it will be shown below, the inclusion of the *α-criterion* does not guarantee that the overall problem is well posed. Furthermore, it was also recognised that there are many situations when it is desired that the shape of the unmeasured, or sparsely measured, regions is preserved after updating. For this reason an additional constraint, we call it '*β-criterion*', was introduced, which attempts to limit the displacement of the control points relative to their original positions. The overall minimisation problem still remains linear and the cost function to be minimised becomes:

$$f = \sum_{k=0}^{M-1} |\mathbf{Q}_k - \mathbf{S}(u_k, v_k)|^2 + \alpha \sum_{i=0}^{n} \sum_{j=0}^{m} \left|\mathbf{P}_{i,j} - \mathbf{S}(u_{i,j}, v_{i,j})\right|^2 + \beta \sum_{i=0}^{n} \sum_{j=0}^{m} \left|\mathbf{P}_{i,j} - \mathbf{P}^0_{i,j}\right|^2 \quad (15)$$

where $\mathbf{P}^0_{i,\,j}$ is the original position of the control point $\mathbf{P}_{i,\,j}$, and $\beta \geq 0$ is a weighting factor.

Substituting Eq. (4) into Eq. (15) gives:

$$\begin{aligned} f = & \sum_{k=0}^{M-1} \left|\mathbf{Q}_k - \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_k, v_k)\mathbf{P}_{i,j}\right|^2 \\ & + \alpha \sum_{k=0}^{n} \sum_{l=0}^{m} \left|\mathbf{P}_{k,l} - \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_{k,l}, v_{k,l})\mathbf{P}_{i,j}\right|^2 + \beta \sum_{i=0}^{n} \sum_{j=0}^{m} \left|\mathbf{P}_{i,j} - \mathbf{P}^0_{i,j}\right|^2 \\ = & \sum_{k=0}^{M-1} \left[\mathbf{Q}_k \cdot \mathbf{Q}_k - 2\mathbf{Q}_k \cdot \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_k, v_k)\mathbf{P}_{i,j} \right. \\ & \left. + \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_k, v_k)\mathbf{P}_{i,j} \cdot \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_k, v_k)\mathbf{P}_{i,j} \right] \\ & + \alpha \sum_{k=0}^{n} \sum_{l=0}^{m} \left[\mathbf{P}_{k,l} \cdot \mathbf{P}_{k,l} - 2\mathbf{P}_{k,l} \cdot \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_{k,l}, v_{k,l})\mathbf{P}_{i,j} \right. \\ & \left. + \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_{k,l}, v_{k,l})\mathbf{P}_{i,j} \cdot \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_{k,l}, v_{k,l})\mathbf{P}_{i,j} \right] \\ & + \beta \sum_{i=0}^{n} \sum_{j=0}^{m} \left[\mathbf{P}_{i,j} \cdot \mathbf{P}_{i,j} - 2\mathbf{P}_{i,j} \cdot \mathbf{P}^0_{i,j} + \mathbf{P}^0_{i,j} \cdot \mathbf{P}^0_{i,j} \right] \end{aligned} \quad (16)$$

where f is a scalar-valued function of $N = (m + 1)(n + 1)$ variables $\mathbf{P}_{i,j}$. We now apply the standard technique of linear least squares fitting, whilst minimising f by setting the derivatives of f with respect to $\mathbf{P}_{r,s}$ equal to zero. The r,s-th derivative is:

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{P}_{r,s}} = & \sum_{k=0}^{M-1} \left(-2\mathbf{Q}_k \cdot R_{r,s}(u_k, v_k) \right. \\ & \left. + 2R_{r,s}(u_k, v_k) \cdot \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,j}(u_k, v_k)\mathbf{P}_{i,j} \right) \\ & + 2\alpha \left(\mathbf{P}_{r,s} - \sum_{i=0}^{n} \sum_{j=0}^{m} (R_{i,j}(u_{r,s}, v_{r,s})\mathbf{P}_{i,j} + R_{r,s}(u_{i,j}, v_{i,j})\mathbf{P}_{i,j} \right. \\ & \left. + \sum_{i=0}^{n} \sum_{j=0}^{m} \left(R_{r,s}(u_{i,j}, v_{i,j}) \sum_{k=0}^{n} \sum_{l=0}^{m} R_{i,j}(u_{i,j}, v_{i,j})\mathbf{P}_{k,l} \right) \right) \\ & + 2\beta(\mathbf{P}_{r,s} - \mathbf{P}^0_{r,s}) \end{aligned} \quad (17)$$

which implies that:

$$\sum_{k=0}^{M-1}\left(R_{r,s}(u_k,v_k)\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}R_{i,j}(u_k,v_k)\mathbf{P}_{i,j}\right)+\beta\cdot\mathbf{P}_{r,s}+\alpha\cdot\mathbf{P}_{r,s}$$
$$-\alpha\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}R_{i,j}(u_{r,s},v_{r,s})\mathbf{P}_{i,j}-\alpha\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}R_{r,s}(u_{i,j},v_{i,j})\mathbf{P}_{i,j}$$
$$+\alpha\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}\left(R_{r,s}(u_{i,j},v_{i,j})\sum_{k=0}^{n}\sum_{l=0}^{m}R_{i,j}(u_{i,j},v_{i,j})\mathbf{P}_{k,l}\right)$$
$$=\sum_{k=0}^{M-1}\mathbf{Q}_k\cdot R_{r,s}(u_k,v_k)+\beta\cdot\mathbf{P}^0_{r,s} \quad (18)$$

Interchanging the order of summation gives:

$$\sum_{i=0}^{n}\sum_{j=0}^{m}\mathbf{P}_{i,j}\left(\sum_{k=0}^{M-1}R_{r,s}(u_k,v_k)R_{i,j}(u_k,v_k)\right)+\beta\cdot\mathbf{P}_{r,s}+\alpha\cdot\mathbf{P}_{r,s}$$
$$-\alpha\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}R_{i,j}(u_{r,s},v_{r,s})\mathbf{P}_{i,j}-\alpha\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}R_{r,s}(u_{i,j},v_{i,j})\mathbf{P}_{i,j}$$
$$+\alpha\cdot\sum_{i=0}^{n}\sum_{j=0}^{m}\left(\mathbf{P}_{i,j}\sum_{k=0}^{n}\sum_{l=0}^{m}R_{r,s}(u_{k,l},v_{k,l})R_{i,j}(u_{k,l},v_{k,l})\right)$$
$$=\sum_{k=0}^{M-1}\mathbf{Q}_k\cdot R_{r,s}(u_k,v_k)+\beta\cdot\mathbf{P}^0_{r,s} \quad (19)$$

Eq. (19) represents one linear equation in the unknowns $\mathbf{P}_{0,0},\ldots,\mathbf{P}_{n,m}$. Letting $r=0,\ldots,n$ and $s=0,\ldots,m$ yields a system of $(n+1)\times(m+1)$ equations in $(n+1)\times(m+1)$ unknowns and it can be presented in matrix notation as:

$$\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}+\alpha\left(\mathbf{I}+\mathbf{B}^{\mathrm{T}}\mathbf{B}-\mathbf{B}^{\mathrm{T}}-\mathbf{B}\right)+\beta\mathbf{I}\right]\mathbf{a}=\mathbf{A}^{\mathrm{T}}\mathbf{b}+\beta\mathbf{a}^0 \quad (20)$$

or:

$$\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}+\alpha(\mathbf{B}-\mathbf{I})^{\mathrm{T}}(\mathbf{B}-\mathbf{I})+\beta\mathbf{I}^{\mathrm{T}}\mathbf{I}\right]\mathbf{a}=\mathbf{A}^{\mathrm{T}}\mathbf{b}+\beta\mathbf{a}^0 \quad (21)$$

where $\mathbf{A}$, $\mathbf{a}$ and $\mathbf{b}$ are as in Eq. (8), while:

$$\mathbf{B}=\begin{bmatrix}R_{0,0}(u_{0,0},v_{0,0}) & \ldots & \ldots & R_{n,m}(u_{0,0},v_{0,0})\\ R_{0,0}(u_{0,1},v_{0,1}) & & & R_{n,m}(u_{0,1},v_{0,1})\\ \vdots & & & \vdots\\ R_{0,0}(u_{n,m},v_{n,m}) & \ldots & \ldots & R_{n,m}(u_{n,m},v_{n,m})\end{bmatrix}$$

and

$$\mathbf{a}^0=\begin{bmatrix}\mathbf{P}^0_{0,0} & \ldots & \ldots & \mathbf{P}^0_{n,m}\end{bmatrix}^{\mathrm{T}}$$

This represents a set of generalised normal equations which are then solved using the same methods as those used for solving Eq. (8).

By examining Eq. (21) it can be seen that all elements of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ are positive and that $\beta\mathbf{I}$ is a positive diagonal matrix for $\beta > 0$. This clearly means that any non-negative value of β will guarantee that the system is stable. However, it can also be seen that inclusion of the α-criterion does not guarantee stability because the matrix $(\mathbf{B}-\mathbf{I})^{\mathrm{T}}(\mathbf{B}-\mathbf{I})$ cannot be guaranteed to be non-singular.

5. Performance analysis and results

Fitting complex surfaces through large measurement data sets can be expensive in terms of both computational time and memory, a significant proportion of which can be attributed to the matrix multiplication $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. The time and memory requirements can be drastically reduced by exploiting the sparse and banded nature of the matrices $\mathbf{A}$. and $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, meaning that only the non-zero matrix elements are stored and directly multiplied. As a result it is possible to reduce both the number of computations and the memory requirements.

A full discussion of the algorithm implementation is beyond the scope of this paper, and it will be reported in a separate publication. Nevertheless, the results in Table 1 were included in order to provide an indication of the realised performance, obtained using a Pentium III 400 MHz personal computer with 128 Mb RAM. The storage requirements of the algorithm to compute $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is $O(N(2p+1)(2q+1))$, while the computational complexity is $O(M((p+1)(q+1))^2)$. For solving the system using

Table 1
Computational performance of the implemented least squares fitting algorithms on a Pentium personal computer. Time in seconds and required memory in kilobytes are given for different numbers of data points and the control points of the fitted surface

Number of data points	Number of control points							
	10		100		1000		10,000	
100	0.031 s	42.6 Kb	0.094 s	145.9 Kb				
1000	0.078 s	42.6 Kb	0.141 s	145.9 Kb	0.781 s	1054 Kb		
10,000	0.625 s	42.6 Kb	0.688 s	145.9 Kb	1.391 s	1054 Kb	7.281 s	9126 Kb
100,000	6.172 s	42.6 Kb	6.203 s	145.9 Kb	6.968 s	1054 Kb	12.969 s	9126 Kb
1,000,000	61.781 s	42.6 Kb	61.515 s	145.9 Kb	63.062 s	1054 Kb	69.250 s	9126 Kb

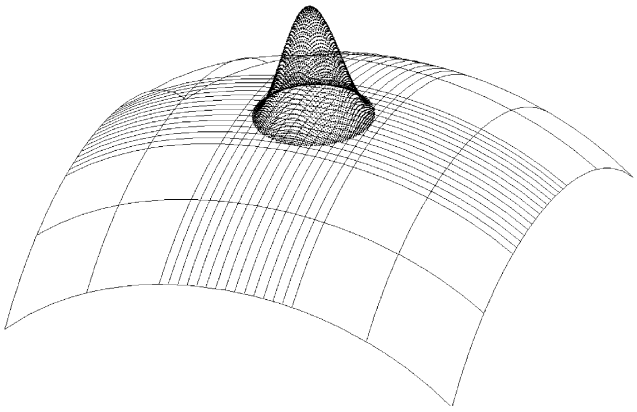

Fig. 2. Base surface and data points.

Gauss–Seidel method, the computational complexity in has dropped significantly, from $O(N^2)$ to $O(N)$. Thus for a problem involving, say, $M = 10^5$ measured points, $N = 10^3$ control points and surface degree $p = q = 3$, the storage requirement is $O(49 \times 10^3)$ doubles. Computational complexity for this case is $O(256 \times 10^5)$ operations (multiplications and additions) for matrix multiplication and $O(10^3)$ operations for the solution phase. This shows that fitting surfaces with many control points to large data sets is achievable at nearly interactive speeds and with only modest memory requirements by today's standards. The achieved computational efficiency is particularly important in view of the suggestions by a number of authors that optimal fitting results may be obtained by repeated knot insertion, re-parameterisation and fitting.

The choice of parameter values α and β deserves attention. As explained previously, both criteria were, in principle, included in order to stabilise the system, but only the β-criterion can guarantee the stability (for $\beta \geq 0$). On the other hand, from the point of view of the resulting surface shape, the two criteria have very different effects and surface quality may be controlled as a trade-off, by the choice of the weights α and β.

The influence of the α-criterion is that it flattens the surface by bringing the control points and the surface closer together, and this effect is stronger in the sparsely measured and/or highly curved regions, where the distance between the control points and the surface is larger. It is also worth noting that the flattening effect of the α-criterion may be reduced in any region through knot insertion, because a finer control mesh is a better approximation of the surface. The overall effect of the α-criterion is that it realises a gentle transition between the measured and unmeasured regions, which may be especially desirable in the case of large deformations.

In contrast, the effect of the β-criterion on the resulting shape is quite different, since it acts to preserve the shape of the unmeasured regions. Therefore it may produce a sharp transition in order to accommodate a large deformation. The shape preserving effect of the β-criterion may also be reduced, through iterative fitting.

In order to obtain a better insight into the effects produced by the choice of α and β, we analysed the fitting results for different values of these parameters. Fig. 2 shows the base surface and the data points used for the analysis. The data set is relatively dense, but it corresponds only to the central region of the surface. The number and distribution of the control points were chosen to be compatible with the desire to accurately represent the deformation prescribed by the data.

Fig. 3(a–p) shows the fitting results arranged in a tabular form, with ascending values of α and β. It should be noted that the indicated parameter values have been set according to Eq. (12), meaning that they are relative weights (for example $\alpha = 0.5$ means that the total effect of the α-criterion is half that of the measured points).

From these results it may be concluded both α and β may be chosen to be sufficiently small such that their effect on the measured regions is negligible, while their relative values may be set by the user to simultaneously produce desirable results in the unmeasured regions. In our work, we experimented with many different shapes and concluded that the effect produced by the α-criterion is generally more desirable than simple preservation of the shape of the unmeasured regions. We found that optimal setting for the parameter α is 0.1, meaning that its overall effect should be about 10 times smaller than that of the measured points. We also found that it is sufficient if the parameter β is set to a small positive value, such that the stability of the system is guaranteed. Thus we found that setting $\beta = 10^{-9}$ produced perfectly adequate results. The results presented in Figs. 4 and 5 were obtained using these settings for α and β.

In order to demonstrate the quality of the results that have been produced using the described approach, we present the example in Fig. 4(a–f), involving physical measurement and reconstruction of a face mask, measuring approximately 250×150 mm. The physical mask, Fig. 4(a), was measured using a coordinate measuring machine to produce a relatively sparse cloud of 3000 data points, Fig. 4(b), and this was then used to fit a pre-defined surface of degree $p = q = 3$, with 43×18 control points, Fig. 4(c) The fitting results are shown in Fig. 4(d). Subsequently, the regions around the eyes, nose and mouth were re-measured to produce a more detailed local data set consisting of 87,500 points, shown in Fig. 4(e). Knot insertion was applied to the previously generated surface, in order to provide the required flexibility in those areas, giving a total of $84 \times 83 = 6972$ control points. The final fitting results are shown in Fig. 4(f). The performance in terms of the computational speed and fitting error statistics are provided in Table 2.

Finally, Fig. 5 presents an example of a car windscreen, measuring approximately 1500×700 mm and involving trimmed NURBS with 20×30 control points. Fig. 5(a) shows the initial shape, where the grey area represents the untrimmed region which was digitised using 20,000 data points. The modified shape after fitting is shown in Fig. 5(b). This example demonstrates the situation where the

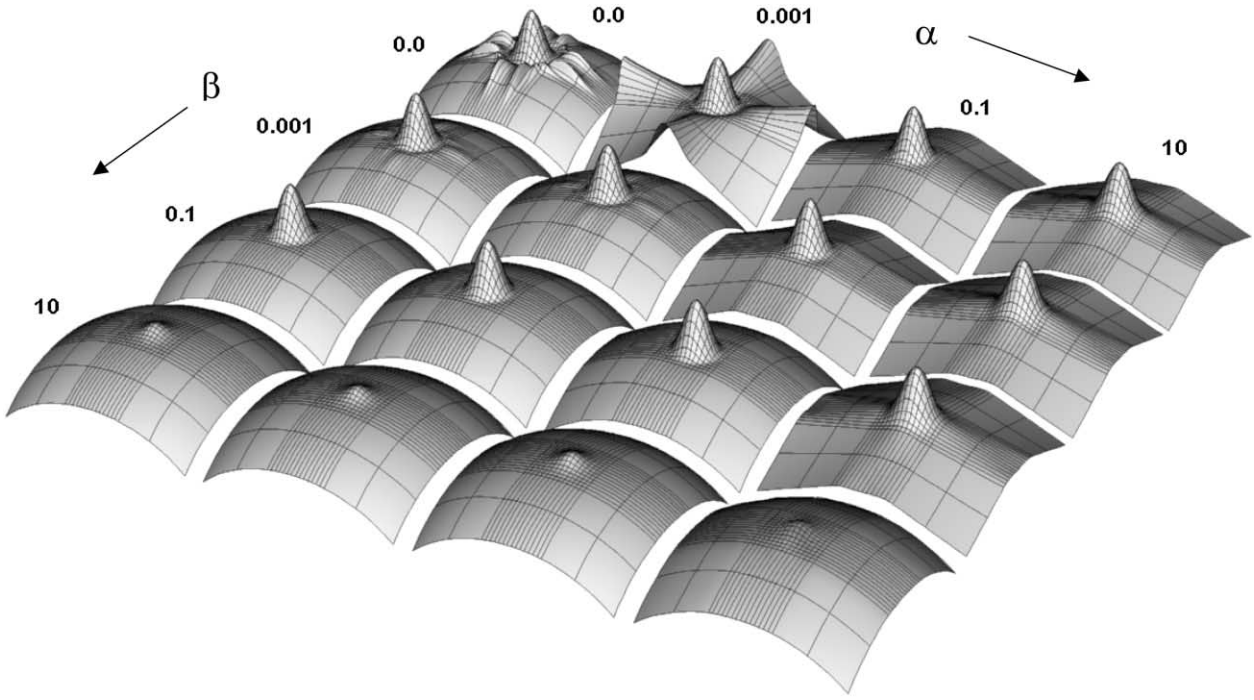

Fig. 3. Effect of varying weights α and β on the fitting results

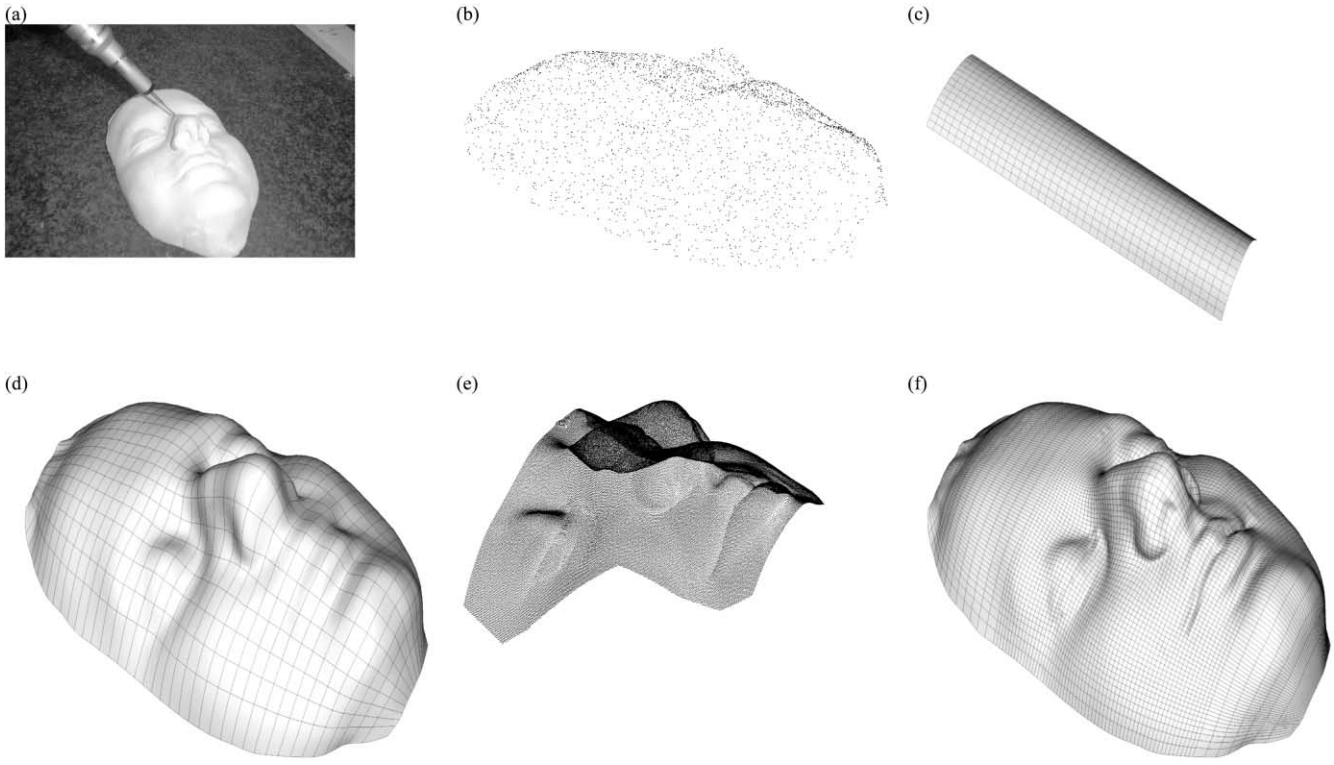

Fig. 4. Surface fitting to measured data. (a) Plaster mask digitised using a coordinate measuring machine; (b) initial measured data cloud; (c) base surface for initial fitting generated by the user; (d) fitted surface; (e) dense point cloud corresponding to the eyes, nose and mouth regions; (f) final result after knot insertion and surface updating using the dense data set only.

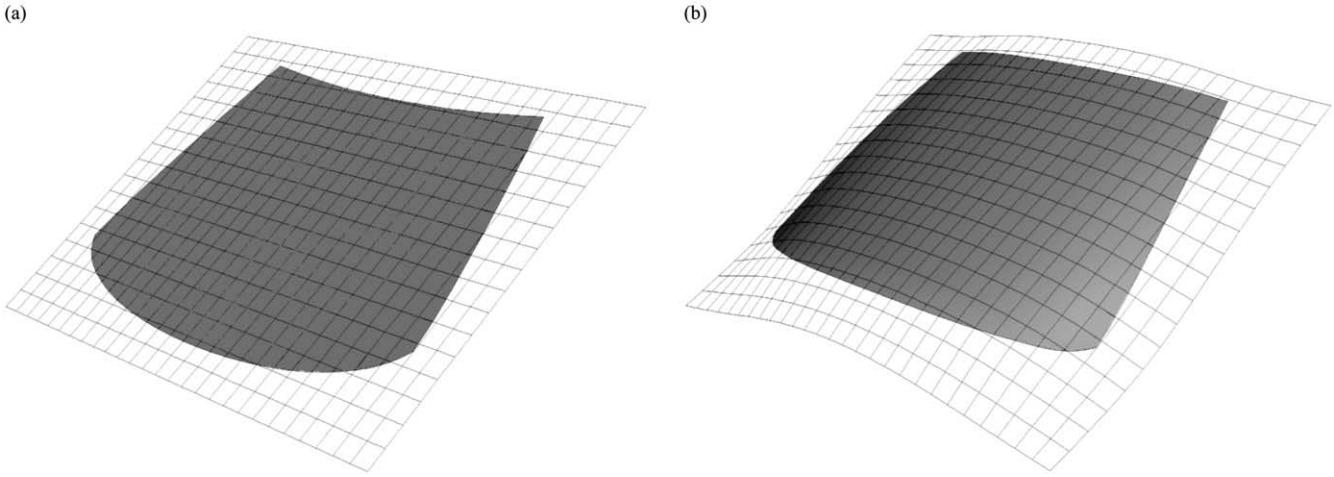

Fig. 5. Trimmed NURBS model of a car windscreen, where the grey area denotes the untrimmed region which was measured. (a) Initial model. (b) Model updated using measured data.

requirement is to minimise the energy of the unmeasured (trimmed) regions.

6. Conclusion

The paper has presented a new method for linear least squares surface updating, which overcomes the frequently encountered problems of rank deficient and ill-conditioned matrices, while also producing desirable results in the data deficient regions. This was achieved by regularisation of the least-squares problem, through the adoption of additional criteria in the minimisation functional. By setting the relevant weights, the user can control the quality of the result according to specific requirements. The proposed method works for unorganised measured points, possibly with highly non-uniform density and not covering all of the surface. The method assumes that an initial approximation of the object shape is available. In industrial applications this will be provided by a CAD model of the part, while in other cases a base surface generated by the user may be employed. The implemented algorithms utilise the sparse structure of the matrices and achieve a considerable improvement in computational speed and memory requirements and the details will be presented in a separate paper.

Table 2
Performance of the proposed method for the example in Fig. 4(f), using 400 MHz Pentium III Personal Computer, 128 Mb RAM

Computational performance	
Number of control points, N	6972
Number of measured points, M	87,500
Time for registration/ parameterisation	15.9 s
Time for multiplication $\mathbf{A}^T\mathbf{A}$	5.4 s
Time for final fitting stage	4.2 s
Total computing time	25.5 s
Surface fitting error statistics	
Standard deviation	0.0573 mm
Mean	0.0003 mm
Maximum	0.591 mm
Minimum	−0.313 mm

7. Notation

$\mathbf{P}_{i,j}$	control points
$\mathbf{Q}_i$	measured points
N	number of control points
M	number of measured points
x, y, z	Cartesian coordinates
u, v	parametric coordinate values
$N_{i,p}(u)$	non-rational B-spline basis functions
$R_{i,j}(u, v)$	piecewise rational basis function
$\mathbf{S}(u_k, v_k)$	Cartesian coordinates a point on a parametric surface
$\mathbf{I}$	identity matrix
$\mathbf{A}$	design matrix
$\mathbf{a}$	vector of unknown parameters
$\mathbf{b}$	vector of observed values
α, β, λ	constants
$\|\mathbf{p}\|_2$	Euclidean norm of vector $\mathbf{p}$
$\|\mathbf{p}\|_2^2$	sum of squares of the elements of vector $\mathbf{p}$
$\mathbf{p}\cdot\mathbf{q}$	dot product of vectors $\mathbf{p}$ and $\mathbf{q}$

References

[1] Besl PJ, McKay ND. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 1992;14(2).
[2] Bjorck A. Numerical methods for least squares problems. Philadelphia: Society for Industrial and Applied Mathematics, 1996.
[3] De Boor C. Gauss elimination by segments and multivariate polynomial interpolation. Purdue Conference on Approximation and Computation, 2–5 December 1993.

[4] DeCarlo D, Metaxas D. Shape evolution with structural and topological changes using blending. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998;20(11):1186–205.
[5] Dierckx P. Curve and surface fitting with splines. Oxford: Clarendon, 1993.
[6] Farin GE. Curves and surfaces for computer-aided geometric design: a practical guide. 4th ed. San Diego; London: Academic Press, 1997.
[7] Farin GE, Hamann B. Current trends in geometric modeling and selected computational applications. J of Computational Physics 1997;138:1–15.
[8] Gordon W, Riesenfield R. B-spline curves and surfaces. In: Barnhill R, Riesefield R, editors. Computer aided geometric design, New York: Academic Press, 1974. p. 95–125.
[9] Lancaster P, Salkauskas K. Curve and surface fitting: an introduction. Academic Press, 1986.
[10] Ma W, He PR. B-spline surface local updating with unorganised points. Computer Aided Design 1998;30(11):853–62.
[11] Ma W, Kruth JP. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. Computer-Aided Design 1995;27(9):663–75.
[12] Ma W, Kruth JP. NURBS curves and surface fitting for reverse engineering. The International Journal of Advanced Manufacturing Technology 1998;14(12):918–27.
[13] Milroy MJ, Bradley C, Vickers GW, Weir DJ. G(1) continuity of b-spline surface patches in reverse engineering. Computer Aided Design 1995;27(6):471–8.
[14] Pahk HJ, Jung MY, Hwnag SW, Kim YH, Hong YS, Kim SG. Integrated precision inspection system for manufacturing of moulds having CAD defined features. The International Journal of Advanced Manufacturing Technology 1995;10(3):98–207.
[15] Piegl LA, Tiller W. The NURBS book. 2nd ed. Springer, 1997.
[16] Press WH, Teukolsky SA, Vetterling SW, Flannery BP. Numerical recipes in C: the art of scientific computing. 2nd ed. Cambridge University Press, 1993.
[17] Qin H, Terzopoulos D. Dynamic nurbs swung surfaces for physics-based shape design. Computer Aided Design 1995;27(2):111–27.
[18] Ramamoorthi R, Arvo J. Creating generative models from range images. Computer Graphics Proceedings, 1999, p. 195–204.
[19] Ristic M, Brujic D. Analysis of free form surface registration. Proc Institution of Mechanical Engineers Part B 1997;211:605–17.
[20] Ristic M, Brujic D, Ainsworth I. Precision reconstruction of manufactured free-form components. 12th Annual International Symposium SPIE Electronic Imaging, San Jose, California, 22–28 January 2000.
[21] Ristic M, Brujic D. Efficient registration of NURBS geometry. International Journal of Image and Vision Computing 1997;15: 925–35.
[22] Schmidt RM. Fitting scattered surface data with large gaps. In: Barnhill R, Boehm W, editors. Surfaces in computer aided geometric design, North-Holland, 1983.
[23] Snyder J, Kajiya J. Generative modelling: a symbolic system for geometric modelling. Computer Graphics (SIGGRAPH 92 Proceedings), 1992, p. 369–78.
[24] Terzopoulos D, Metaxas D. Dynamic 3D models with local and global deformations: deformable superquadrics. IEEE PAMI 1991;13(7):703–14.