Background Surface Estimation for Reverse Engineering of Reliefs

Shenglan Liu [1,2] Ralph R. Martin [1] Frank C. Langbein [1] Paul L. Rosin [1]

[1] School of Computer Science, Cardiff University, UK
[2] CMEE, Nanjing University of Aeronautics and Astronautics, China
{Shenglan.Liu, Ralph, F.C.Langbein, Paul.Rosin}@cs.cf.ac.uk

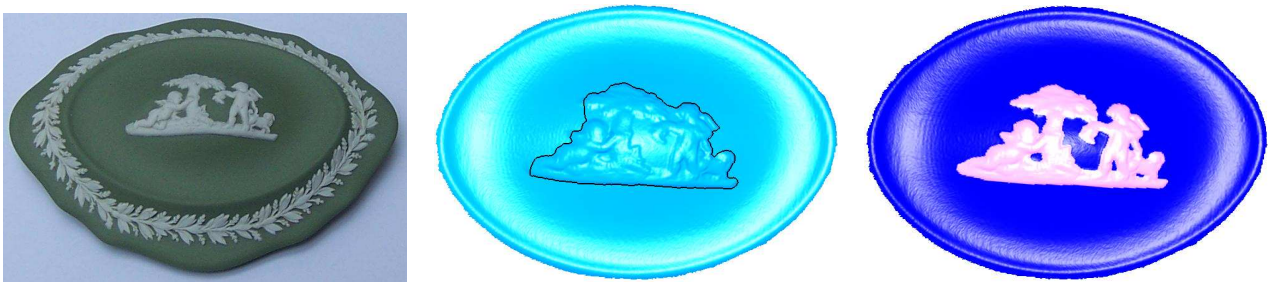


Figure 1. Relief segmentation for a porcelain box lid; relief segmented by earlier snake-based method; refined result using the method in this paper.

Abstract

Reverse engineering of reliefs *aims to turn an existing relief superimposed on an underlying surface into a geometric model which may be applied to a different base surface. Steps in this process include segmenting the relief from the background, and describing it as an offset height field relative to the underlying surface. We have previously considered relief segmentation using a geometric* snake. *Here, we show how to use this initial segmentation to estimate the background surface lying under the relief, which can be used (i) to refine the segmentation and (ii) to express the relief as an offset field.*

Our approach fits a B-spline surface patch to the measured background data surrounding the relief, while tension terms ensure this background surface smoothly continues underneath the relief where there are no measured background data points to fit. After making an initial estimate of relief offset height everywhere within the patch, we use a support vector machine *to refine the segmentation.*

Tests demonstrate that this approach can accurately model the background surface where it underlies the relief, providing more accurate segmentation, as well as relief height field estimation. In particular, this approach provides significant improvements for relief concavities with narrow mouths and can segment reliefs with small internal holes.

Keywords: *relief, segmentation, surface fitting, reverse engineering, background surface.*

1 Introduction

The goal of *reverse engineering of reliefs* is to turn a 3D scan of an existing relief superimposed on some underlying surface into an appropriate geometric model. An example relief on the lid of a porcelain box is shown in Fig. 1(left). The relief in this case is the slightly raised white motif applied to the coloured background. (We are interested in the central relief for now, not the white frieze around the edge).

Reverse engineering of reliefs allows them to be analysed, edited, and ultimately, reapplied to CAD models of various different objects. The need for such capability occurs in various industries such as sign-making, packaging, porcelain design and antique reproduction. For example, in the packaging industry, the demand for diverse, high-quality packaging designs is growing, to better attract the attention of consumers. Packages with reliefs appear more noticeable than plain packages. However, it is time consuming and expensive to design a new decorative relief. To meet the needs of low cost and short time-to-market, it is useful to be able to reverse engineer previously designed and manufactured reliefs (which were produced without the aid of CAD). Another example can be seen in replicating antique porcelain, or extending an existing, long-standing range of porcelain. At present, sculptors have to hand-copy the decorative reliefs, a process that requires a high level of skill, yet is also tedious and time consuming. Relief reverse engineering can free sculptors from such work, and allow the company to rapidly and automatically re-use reliefs.

A *relief* can be defined as an area of a surface with sculpted features different from those of the underlying surface, and which is raised by a small height; this height is typically larger than the characteristic size of features on the background. We can thus consider the relief to be a small offset relative to an underlying smooth, gently curved (relative to relief height) surface.

Reverse engineering of reliefs involves several steps including 3D scanning the object bearing the relief, constructing a triangle mesh from that point cloud data, segmenting the triangle mesh to extract the relief from the underlying background, representing the relief as an offset height field, and flattening the relief's underlying surface onto a plane, to prepare the relief for easy application to a new background surface.

We have previously addressed various aspects of relief segmentation problems such as separating an isolated relief from a smooth and slowly varying background [14], segmenting periodic reliefs and extracting a single basic repeat unit of such reliefs [16], and segmenting reliefs from more complex textured backgrounds [15]. All these methods segment the relief from the background using an active contour (a *snake*). The snake starts from an initial contour chosen by joining a few user-specified points, and is driven towards the outer relief boundary by energy terms based on characteristic relief properties: either the raised step at the relief boundary, or the difference in surface properties between the relief and the background surface. The evolution of the snake occurs in several phases which, firstly make it approach to the relief boundary quickly by a coarse contour, and then refine it and drive it as deep as possible into any concavities.

The snake methods generally produce reasonable segmentation results even for many reliefs with deep concavities. However, as we have pointed out in [14], this snake-based approach cannot locate relief boundaries for small internal holes in a relief, and for concavities with narrow mouths. Both of these frequently occur in real reliefs. For example, a typical snake-based segmentation can be seen in Fig. 1(middle), where the concavity between the boy's head and the tree (top centre of the relief), and several internal holes, have not been separated.

In this paper, we estimate the underlying smooth surface upon which the relief is assumed to be placed, which allows us to (i) find relief boundaries in the tricky cases mentioned above, (ii) find more accurate relief boundaries than our previous segmentation methods could achieve alone, and (iii) to model the relief as a normal offset field relative to this underlying smooth surface, a necessary prerequisite to relief flattening. The method presented in this paper assumes that the surface surrounding the relief is smooth, not textured; consideration of textured backgrounds is planned in our future work.

As we assume the relief background surface is smooth and slowly varying, it is appropriate to represent it by a B-spline surface as typically used in CAD. We define a rectangular region surrounding the relief, including some of the surrounding background surface (see e.g. Fig. 4(c)) and fit a suitable B-spline surface modelling this background surface. However, note that part of the area inside our rectangular boundary is covered by the relief: we can only fit the B-spline to those data points which are not covered by the relief. Thus, we use our previous segmentation method [14] (with a small safety margin) to choose those parts of the mesh inside the rectangle which are definitely part of the background surface and should be included in the fitting process. We must fit a surface to data with a hole, or an *ignore area*, to use Varady's terminology [5]. We follow Varady's approach, and fit this surface in such a way that where there *is* background data, the B-spline approximates it well, and where there is *no* background data, tension terms used in the surface fitting process smoothly extend the B-spline surface over the hole in a natural manner.

Once the background surface has been found, in principle, parts of the mesh with a height of zero above this surface belong to the background, and parts with positive height belong to the relief. In practice, due to noise, and errors in fitting the background surface, such a simplistic classification does not work well. Instead, for robustness, we employ a *support vector machine* (SVM) [21], which is designed for binary classification, to re-classify mesh vertices as background or relief. The idea of SVM is to map the input feature vectors into a high dimensional feature space through some non-linear mapping

(chosen a priori), and to construct an optimal separating hyperplane in this space using some training data, and then the whole data set can be set apart by the hyperplane. Here in our case we use heights above the background *and* other properties as the input features, and the training data is automatically chosen near the boundary of the relief. This classification is then used to improve the existing relief boundary, both globally and locally: as shown in Fig. 1 (right), this approach can detect areas within the outer relief boundary which belong to the background (our previous segmentation method only detects the outer relief boundary), and it can also properly determine the extents of concavities with narrow mouths (again, a problem for our earlier segmentation approach). It also generally provides a relief boundary which more accurately follows local details.

Finally, having estimated the background surface, and found accurate relief boundaries, the raised relief can then be represented as an offset relative to the background surface.

In Section 2 we review related work: B-spline surface fitting with 'ignore areas', hole filling and surface detail transferring, and the support vector machine method. Section 3 addresses background surface estimation from data with an 'ignore area' using a B-spline surface, while Section 4 describes relief offset field representation and relief segmentation refinement utilising the SVM classification method. Experiments and results are provided in 5. Section 6 concludes the paper and suggests future work.

2 Related Work

2.1 B-spline Surface Fitting with 'Ignore Areas'

It is well known that B-spline surface fitting for evenly distributed measurement data can be posed as a least-squares fitting problem, where a functional describes the error between the data and the surface to be fitted. However, when the data contains large 'ignore areas', or holes, the least-squares system may become singular. This problem has been studied in several papers, e.g. [5, 17, 4, 22]. Following [5], the basic approach used to overcome this problem is to add into the functional a second term representing *tension*, which can prevent singularity of the system matrix and which also ensures that the surface generated propagates smoothly across the hole, without unwanted ripples. A weight is used to adjust the relative importance of goodness of fit to the data outside the hole, and tension, or smoothness, of the surface produced. This approach is based on the assumption that the background surface itself does nothing unexpected in the region for which no data is present, and hence a smooth continuation is the most natural solution.

Martin et al. [17] give an approach to estimate the volume of a flesh wound from scanned data. To estimate the shape of the original skin, a surface is fitted to a region around the wound, with an 'ignore area' where the wound is present. The surface is simply treated as a Monge patch: $z = f(x,y)$; cubic Bézier or single-patch second order B-spline basis functions are used. This method is simple, fast and easy to implement, but the number of control points is limited, and such a simple approach cannot adequately capture the necessary detail required for reverse engineering: its goal is to provide *coarse* estimates of wound volumes.

Dietz [4] represents the tension term in a simplified form using quadratic functionals of the parametric derivatives, enabling the linearity of the functional to be preserved.

A thorough investigation into surface fitting with irregularly distributed data and holes can be found in [22], where various techniques are proposed. The method firstly computes a reference surface to provide an initial parametrisation, and then performs accurate fitting with a gradually tuned tension weight. The knot vector and tension weight are automatically adjusted until the prescribed distance tolerances are reached. Various strategies are proposed for handling of the weakly defined control points which are related to 'ignore areas', and for shape dependent knot insertion.

We follow the basic principles used in [4] and [22], allowing the fitted B-spline surface to achieve high accuracy where background data is present and to smoothly cross the 'ignore areas'. However, in the particular case of relief background surface fitting, the most important task is to accurately approximate the original designed background surface in the area of the relief. To achieve this goal, we have performed tests on various models giving ground-truth, and used them to tune the parameters such as the number of control points and the smoothness weight. We take advantage of the background being slowly curved and being relatively evenly sampled, to allow a straightforward approach to parametrisation and choice of knot vector.

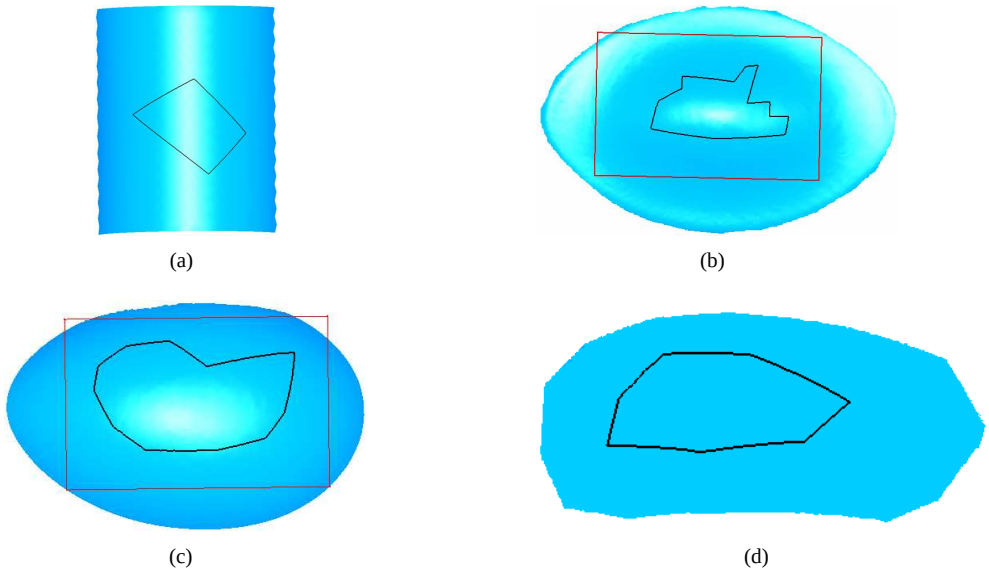


Figure 2. Ground-truth models for surface fitting with 'ignore area': (a) a synthetic cylinder, (b)–(d) real relief backgrounds.

2.2 Hole Filling and Surface Detail Transferring

Hole filling for scanned data, especially for triangular meshes, is also an active research topic. The main aim is to plausibly complete small gaps in a mesh caused by e.g. occlusion during scanning. However, in certain cases, large parts of a surface may need completion, e.g. if access to all sides of the object is not possible with the scanner.

Many methods for hole filling have been proposed. Representative approaches include implicit surface construction using radial basis functions [2], the use of mesh smoothing or mesh deformation operations [13, 19], and volumetric diffusion [3].In general, the main requirements of such hole filling methods are either the production of a water tight surface, or the production of an aesthetically pleasing, plausible surface, while generating geometry with no obvious join at the hole boundary.

To transfer surface details between meshes generally incur the problem of separating a surface into base and detail, which shares some similarity to our work. The detail surface can be represented as a scalar displacement along the normal [1] or as local differential coordinates [20] over a base surface, e.g. a multiresolution subdivision surface. The base surface is extracted by mesh smoothing method or by its low-frequency subdivision surface, which are good enough to meet the visual and aesthetical requirement.

Generally, in both hole filling and surface detail transferring cases, *accurate* replication of some assumed correct geometry in not an important consideration. Indeed, such methods are generally not accurate enough for estimating the surface underlying a relief, where small errors in position can lead to large estimated errors in relief height.

2.3 Support Vector Machine Method

Support vector machines (SVMs), introduced by Vapnik [21], solve the *two-class* pattern recognition problem, based on structural risk minimisation. A vector of properties is used to describe each individual in a set, and each individual is allocated to one of the two classes according to the values of the properties. (In our problem, the individuals are mesh points, the classes are *relief* and *background*, and the properties are geometric quantities). The approach used is to map the input feature vectors into a high dimensional feature space through some non-linear mapping (chosen *a priori*), and then to construct an optimal separating hyper-plane in this space.

SVMs have proven successful for various tasks such as text categorisation [9], face detection [18], image texture classification [10, 12], and relief segmentation on a textured background [15]. Encouraged by the successful use of an SVM in our earlier work, we also use an SVM in this paper to classify points, enabling us to find interior holes in the relief, and narrow-mouthed concavities, and to refine the relief contour. However, the features used are different from those used in our earlier work, and in particular the offset height from the mesh to the estimated background surface, is used.

3 Background Surface Estimation

We first find an initial segmentation of the relief, using a snake, by our previous method [14]. This step is necessary so that we can fit a background surface, which must be done using mesh points surrounding the relief which are known not to belong to the relief. We next estimate the background surface, using B-spline surface fitting with 'ignore areas'.

We next explain the basic functional used for fitting, and then describe in detail how we use the functional in practice.

3.1 Basic Functional

We assume here that the relief lies on a smooth and slowly varying background. However, the shape of the background surface is not restricted to any special class, and could be, for example, a cylinder, a surface of revolution, or indeed any simple free-form surface. In our implementation, we approximate the background surface using a single bi-cubic tensor product B-spline surface written in the form

$$\mathbf{S}(u,v) = \sum_{j=0}^{n_u-1} \sum_{k=0}^{n_v-1} \mathbf{s}_{jk} N_j(u) N_k(v), \tag{1}$$

where the surface has $n_u \times n_v$ control points $\mathbf{s}_{jk}$ and the $N_\cdot(\cdot)$ are B-spline basis functions of degree 3.

We assume that a pair of parameter values (u_i, v_i) (determined as explained in Section 3.2) is associated with each data point $\mathbf{p}_i$ ($i = 0, \ldots, m-1$). We use an open uniform knot vector scheme for simplicity: it is adequate for a slowly varying background surface. Then for every pair j, k, the task of finding the control point $\mathbf{s}_{jk}$ can be solved by minimising the linear functional [4, 22]:

$$\sum_{i=0}^{m-1} \|\mathbf{S}(u_i, v_i) - \mathbf{p}_i\|^2 + \gamma \int\int \mathbf{S}_{uu}^2 + 2\mathbf{S}_{uv}^2 + \mathbf{S}_{vv}^2 \, du\, dv, \tag{2}$$

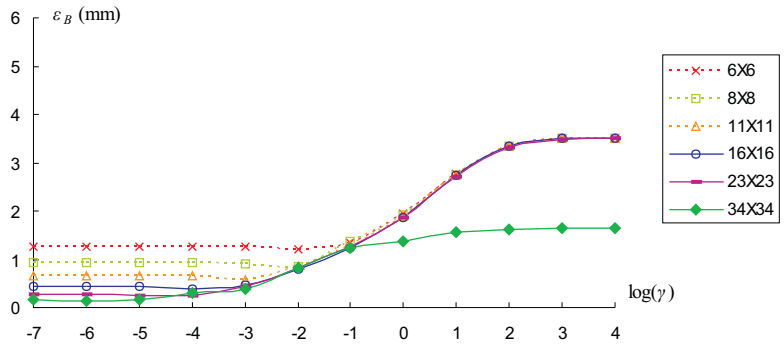
where γ is a tension weight adjusting the relative goodness of fit to the background data points, and smoothness of the whole background surface.

Together, Equ. 1 and 2 determine a bicubic uniform B-spline surface, given associated parameter values for each data point, the number of control points, and the smoothness weight. We next explain how to choose these values.

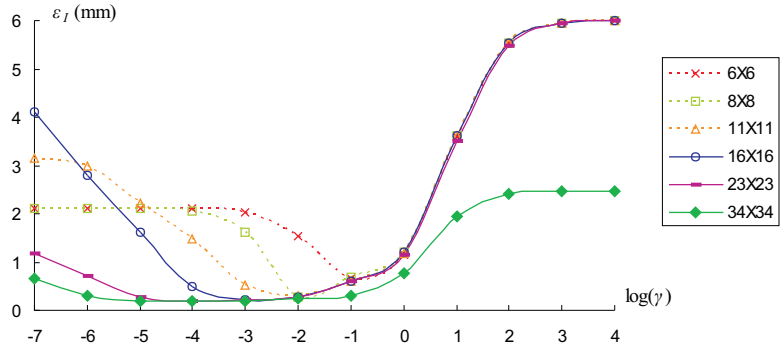
3.2 Parametrisation

To obtain the parameter values (u_i, v_i) for every data point $\mathbf{p}_i$, we do the parametrisation in two steps.

Firstly, an initial parameterisation is found by mapping the data points onto a base surface. For simple background surfaces like those shown in this paper, a plane is adequate. We obtain the plane via simple user interaction: the user selects three points on the model as shown in Fig. 4(c). A plane is constructed; the bottom two points determine one side of the rectangle in this plane, and the third point determines its height. For efficiency, this region should not be too much considerably larger than the area covered by the relief: any point data far away from the relief is unlikely to have much bearing on the shape of the background surface underneath the relief. On the other hand, an adequately large rectangular region should be used to provide sufficient points to model the geometry of the background surface. An appropriate compromise seems to be to use a rectangle with a margin of about 1/3 to 1/2 of the diameter of the relief on all sides of the relief. For a model with a small background area, the user can use all the background data if desired.



(a)



(b)

Figure 3. Accuracy of fitted background surface with varying smoothness weight and number of control points: (a) in background region, (b) in 'ignore area'.

Secondly, a 4×4 B-spline surface is fitted using the initial parametrisation, to give an improved base surface. Improved parameter values for the data points are obtained by finding and using the parameter values of the closest points on the fitted surface [6, 22].

3.3 Tension and Control Point Choice

In order to determine the number of control points, and the tension weight, we assessed some surfaces with ground-truth data, which also allowed us to see how sensitive our approach is to changes in these values. Various simulated and real scanned mesh surfaces were tested as shown in Fig. 2. Fig. 2(a) is a synthesised cylinder, and the other three test surfaces were obtained by scanning the *reverse* side of various objects with reliefs: the reverse surface is quite similar to the side with the relief, except, of course, it carries no relief. For example, the surface in Fig. 2(b) is the inner surface surface of the box lid shown in Fig. 1(a).

A (black) irregular polygon was drawn on each test surface to simulate a relief contour and was used to cut out an 'ignore area' from the test surface. The (red) rectangular regions in Figs. 2(b) and (c) define the portion of the background data used for surface fitting; in Figs. 2(a) and (d), all background data outside the simulated relief contour was used for surface fitting.

Fig. 3 illustrates the results produced using the test model shown in Fig. 2(b) for varying numbers of control points and tension

weight. The size of this model is about $80 \times 50 \times 12\text{mm}^3$ and the part inside the rectangle is $50 \times 32 \times 9\text{mm}^3$. Fig. 3(a) shows the accuracy of the fitted B-spline surface in the region used for surface fitting, while Fig. 3(b) shows the accuracy of the fitted B-spline surface in the 'ignore area'. The vertical axis shows the mean absolute distance between the data points and the fitted surface (denoted by ε_B and ε_I respectively for the background region and the ignored area). The horizontal axis shows the logarithm of the smoothness weight. The different curves represent differing numbers of control points (chosen to make knot span decrease by a factor of 1.5 for each successive choice).

Fig. 3(a) shows that using more control points and a lower smoothness weight produces a more accurate surface fit in the *background* region. Weiss [22] obtained very similar results and suggested fixing the smoothness weight γ so that the least-squares residual was 1.15 times that obtained when $\gamma = 0$ to avoid over-smoothing. Fig. 3(a) shows that as the smoothness weight γ decreases beyond a certain value, there is no effect on the accuracy.

However, if we consider the fit to the *ground-truth* data in the supposed *ignore area*, in Fig. 3(b) (which was not considered in [22]) different effects are observed. For a fixed number of control points, in the ignore area, the most accurate surface is not produced by choosing very small γ due to the lack of data and insufficient smoothing. Both under-smoothing and over-smoothing can occur if γ is chosen inappropriately. Generally, as we would expect, increasing the number of control points improves the fit, although past a certain number, little extra accuracy is obtained, while the computation time is increased. It is important to note that these plots are quite flat near their minima, indicating that the choice of γ is *not* critical to get good results. Also, as the different curves have similar minima, we can conclude that the number of control points to use is not critical either.

The concept of *weakness* of a control point (denoted as σ) is used by Weiss [22] to describe how well a control point is supported by the *measured* point data. A zero value for σ means the control point is totally determined by the smoothness term in Equ. 2. We considered how many control points had zero σ, the values being 0, 0, 0, 5, 23 and 57 for the surfaces with 6×6, 8×8, …, 34×34 control points in Fig. 3(b). However, there was no clear relationship between this zero count and the accuracy of the surface fitted in the ignore area.

Similar results were obtained using the other ground-truth models, even for the model in Fig. 2(c) which has a large ignore-area compared to the selected background. The porcelain objects in Figs. 2(b–d) have varying surface shape and were scanned using a Minolta VI-910 scanner. B-spline surfaces were fitted with an accuracy of 0.2–0.3mm in the 'ignore area' using a tension γ in the range 10^{-4} to 10^{-3} and at least 16×16 control points. This is an acceptable accuracy as the estimated roughness of the porcelain is about 0.08mm and the precision of the scanner is also about 0.08mm. For the cylinder model, with height 120mm and radius 50mm, an accuracy of 0.03mm ($10\times$ better than for the other models) was obtained in the 'ignore area', using similar parameters as for the other three models.

As noted above, our experiments suggest that neither the value of smoothness weight nor the number of control points is critical to success, and good results can be obtained over a wide range of both of these parameters. So we suggest that the user use $\gamma = 10^{-3}$ and 16×16 patches as generally suitable values. For other real relief models, these values produce quite reasonable results as shown in Figs. 4–7. Nevertheless, for reliefs with rather different characteristics, it might be appropriate for users to scan some background data of their own and carry out some ground-truth tests like those illustrated in Fig. 3, to select $n_u \times n_v$ and γ to obtain the best possible results.

4 Relief Segmentation Refinement

Having fitted the background surface, a support vector machine method is used to re-classify the mesh into relief and background areas, to refine the initial snake-based segmentation. As noted earlier, simply using the criterion that the relief height is positive relative to the background surface is too simplistic to give good results. Thus, the SVM is used for classification, which takes both the relief height and other factors into account.

4.1 Relief Offset Field

A complex relief surface can be regarded as a smooth underlying background surface with superimposed relief details. The complex surface is our original triangle mesh surface $\mathbf{M}$ and we have estimated the background surface $\mathbf{B}(u,v)$ by a B-spline

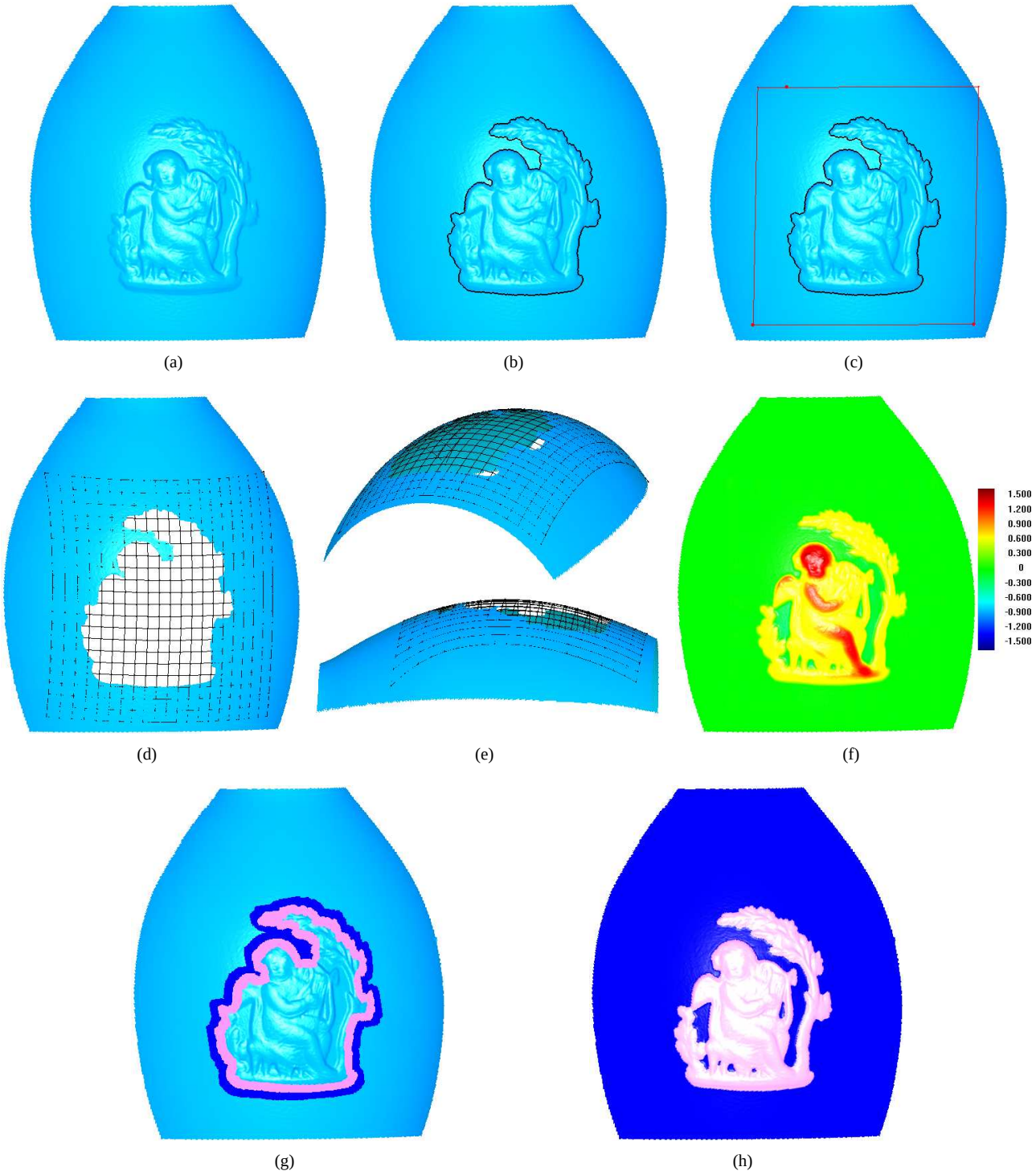


Figure 4. Background surface fitting and SVM re-classification for the 'Angel' model: (a) the mesh model, (b) the relief contour segmented by previous snake method, (c) parametrisation plane, (d) estimated B-spline surface, (e) another views for the fitted surface, (f) relief height, (g) training data for SVM, (h) SVM re-classification result.

surface in the previous section. The relief detail can be represented as a normal offset field $h(u,v)\mathbf{n}(u,v)$, where $\mathbf{n}(u,v)$ is the normal of $\mathbf{B}(u,v)$ and $h(u,v)$ is the offset, i.e. the relief height relative to this background surface:

$$\mathbf{M}(u,v) = \mathbf{B}(u,v) + h(u,v)\mathbf{n}(u,v). \tag{3}$$

For each mesh point $\mathbf{M}_i$, we find its foot-point $\mathbf{F}_i$ on the base surface to give the corresponding u_i and v_i, and we then estimate the relief height h_i as $|\mathbf{M}_i - \mathbf{F}_i|$, which is similar to the displacement map representation in [11].

4.2 SVM Classification

The relief height h_i can in principle be used to re-classify the relief and background: a zero value for h_i means the data point belongs to the background, and a positive h_i means the data point belongs to relief which in this paper is assumed to be *extra* material added on top of the original surface. (There is nothing in principle to prevent our methods being applied to negative reliefs, i.e., embossing, too.) However, due to various sources of error (roughness of the scanned object, the scanning process, background surface estimation, etc.), points on the background do not have perfect zero h_i. We might try to set a small threshold to allow for such errors, but this does not work well: firstly, it is not easy to choose an appropriate threshold, and secondly, the same value of threshold may not be appropriate in all areas of the relief—for example, background surface estimation errors may be larger in some places than others.

Thus, we utilise a more robust classification method, a *support vector machine*, to solve our particular background-relief classification problem. It works as follows: given some training samples $(\mathbf{P}_i, C_i)$ where $\mathbf{P}_i$ is a feature vector of property values at a point, and C_i is the class label for the point (*relief* or *background*), the SVM is trained to give an SVM model which can reliably predict C_i from $\mathbf{P}_i$.

For our problem, we use as the feature vector $\mathbf{P}_i = \{u_i, v_i, h_i, l_i\}$ where l_i is the *planarity* of a point, defined to be the signed distance between the point and the average plane fitted to its 2-ring neighbours. This property describes how curved the surface is locally, and helps classifying points near the relief boundary—the step up to the relief from the background causes large values for planarity there. The coarse segmentation from the snake is used to automatically choose suitable training data, near to the coarse boundary as shown in Fig. 4. We have found using strips of data in a region of width just over 3 times of the relief height works well in practice.

Various software implementations of SVM are available. We used LIBSVM [8] which is an efficient open source SVM package written in C++, with a helpful guide for use by practitioners [7]. We use radial-basis functions (RBF) for the SVM kernel, which is a commonly recommend kernel for general usage and works well in practice for our problem. Other SVM parameters are left at the default values provided by LIBSVM.

Using the training data, the LIBSVM algorithm generates an SVM model which we use to classify all mesh points inside the rectangular patch region as background or relief. (This includes relabeling any original training points if necessary, to correct any outliers that were previously been assigned to the incorrect class). After SVM classification, the result is a refined segmentation which more accurately locates the relief boundaries, as well as clearing out any deep concavities in the relief and internal holes.

We now close the loop by using this improved classification to refine the background surface. Our new knowledge of background points in interior holes and deep concavities provides us with extra data for background surface fitting within what was previously part of the ignore area. The background surface is refitted, which in turn gives a better estimate of the relief height field. Indeed, it is also now possible to redo the SVM classification with this refined height field, which can also provide further, although usually minor, improvements to the segmentation.

5 Results

As well as the ground-truth tests performed for parameter determination, we have carried out various experiments on a variety of real scanned relief models.

Fig. 4 illustrates the steps of our algorithm for the ‘Angel’ relief model which has a very deep concavity and several internal holes. Fig. 4(a) shows the mesh model captured by a Minolta VI-910 3D scanner, and Fig. 4(b) shows the relief contour

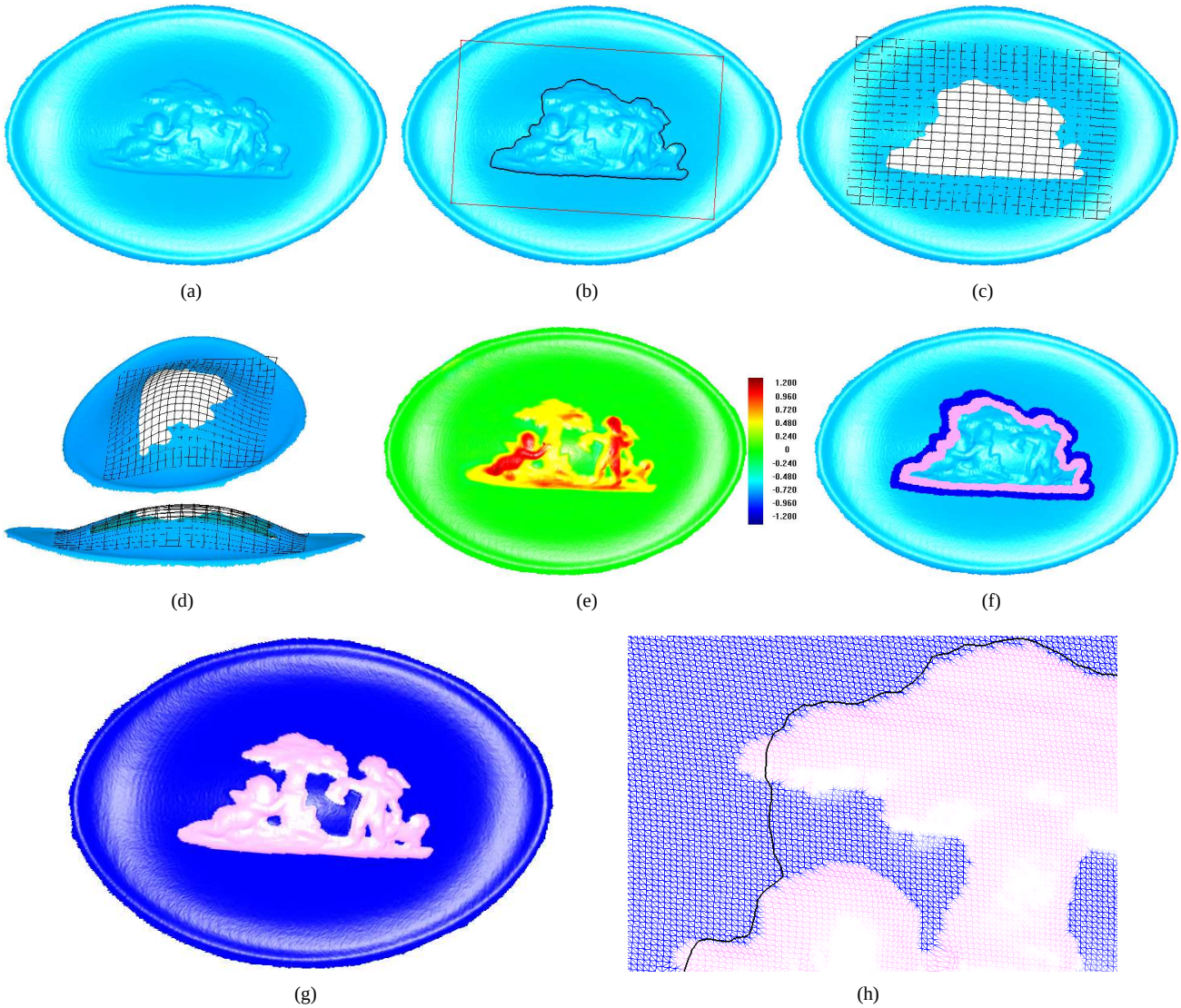
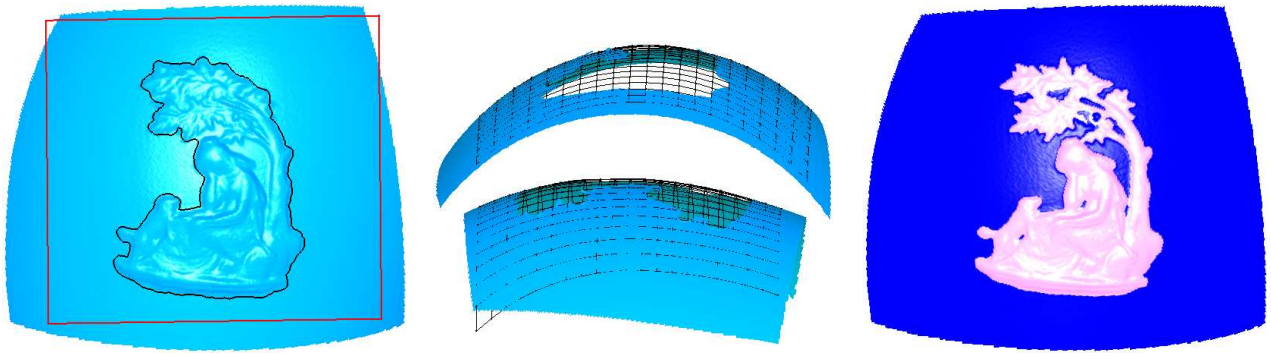


Figure 5. Background surface fitting and SVM re-classification for the 'Tree' Model: (a) the mesh model, (b) the relief contour and parametrisation plane, (c) estimated B-spline surface, (d)another views for the fitted surface, (e) relief height, (f) training data for SVM, (g) SVM re-classification result, (h) close-up of the snake contour and SVM refinement result.

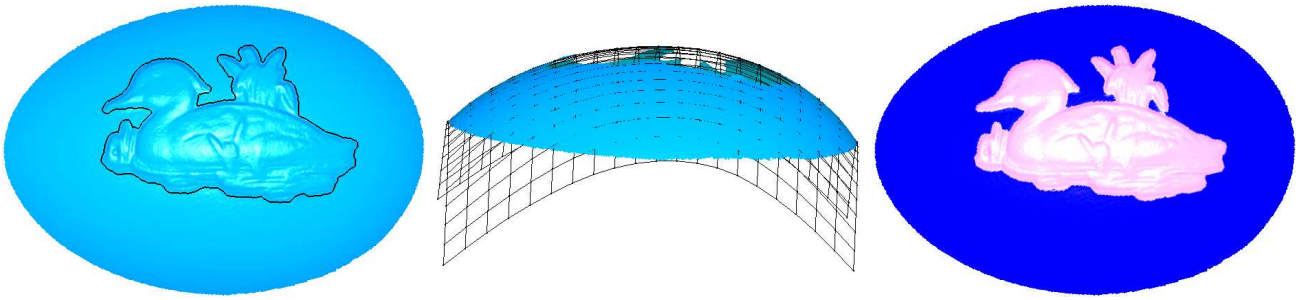
obtained by our previous snake method. The three dots in Fig. 4(c) were specified by the user to define the plane for initial parametrisation and to generate the rectangular region for background surface fitting. The grid lines in Figs. 4(d) and (e) show the fitted B-spline surface. The front of the mesh is coloured blue, and the rear of the mesh green, in this figure. Here we used 16×16 control points and $\gamma = 0.001$. We can see that the estimated background smoothly spans the 'ignore area' very well. Fig. 4(f) shows a colour map of the relief height relative to the background surface. Fig. 4(g) shows the training data used for the SVM, which in this case were the 8-ring neighbours of the snake. The SVM classification result is shown in Fig. 4(h). Note that, although there may be mislabeled points in the training data set, they are robustly corrected by the SVM classification. Note also that the deep concavity between the boy and tree, and several internal holes, are now correctly recognised as background, giving a much improved segmentation compared to the initial snake segmentation.

Fig. 5 shows the tests on the 'Tree' relief model. Note that the background surface has a more complex shape than the previous example, and turns up at the edge: see Fig. 5(d). Again, the B-spline background surface was fitted using $\gamma = 0.001$,



(a) (b) (c)

Figure 6. Background surface fitting and SVM re-classification for the 'Lady' Model: (a) the relief contour and parametrisation plane, (b) estimated B-spline surface, (c) SVM re-classification result.



(a) (b) (c)

Figure 7. Background surface fitting and SVM re-classification for the 'Duck' Model: (a) the relief contour and parametrisation plane, (b) estimated B-spline surface, (c) SVM re-classification result.

$n_u = n_v = 16$. Fig. 5(a) shows the mesh model, while Fig. 5(b) shows the snake contour and user specified rectangle. The estimated background surface is shown in Fig. 5(c) and (d) and the relief height is shown in Fig. 5(e). Fig. 5(f) and (g) show the SVM training data and classification result. Again, we can see that the concavities and the internal holes are correctly classified. As well as correcting the gross errors in the initial segmentation, the SVM classification provides a more accurate relief boundary in general, as shown in the close-up picture in Fig. 5(h).

Other tests on the 'Lady' and 'Duck' models are shown in Figs. 6 and 7. For the 'Lady' model, we used $\gamma = 0.001$, $n_u = n_v = 12$. The surface shown in Fig. 6(b) is the refitted background using the SVM classification result. Based on the refitted background, the improved segmentation shown in Fig. 6(c) is obtained. For the 'Duck' model, we used *all* the background data and set $\gamma = 0.001$, $n_u = n_v = 16$.

All algorithms were implemented in C++ and the computations were performed on a PC with a 2.4GHz AMD Athlon CPU and 1GB of RAM. In the whole process, the most time consuming part is the background surface fitting phase. For example, the 'Angel' model has 287446 triangles and 144564 points, of which 59587 points were used for surface fitting. For this model, the previous snake based segmentation method took about 2 minutes, background surface fitting with 16×16 control points took 35 minutes and height-field estimation and SVM classification took about 10 minutes. For other models, similar computation times were observed.

6 Conclusions and Future Work

Based on our previous snake-based segmentation results for relief reverse engineering, we have addressed the problem of background surface estimation, allowing us to represent the relief as a height field, and also to perform segmentation refinement.

We estimate the background surface via B-spline surface fitting with an 'ignore area'. Although this approach has been addressed in several papers, previous work has not reported how the accuracy of the fit within the 'ignore area' is dependent on the key parameters of smoothness weight and number of control points. Our tests on ground-truth models have shown that the method is in fact fairly robust to variations in both these parameters. Ground truth data has allowed us to find suitable values for these parameters.

Once the background surface has been found, we can describe the relief as a normal offset field relative to the background surface, giving the relief height for every point on the mesh. Using the relief height and other properties, we re-classify the mesh into relief and background using an SVM. This re-classification improves the segmentation in several ways: relief concavities with narrow mouths are now correctly identified, as are internal holes in the relief, and at the same time, the relief boundary is generally more accurately located.

Our method works best for simple background surfaces with lowish curvature, while it does not cope well with those are highly curved. To improve the fitting efficiency, the original mesh surface can be simplified although it may reduce accuracy of final relief height. When the background surface is curved around, for example, like a cylinder, then the parameterisation method in Section 3.2 should be improved to use a different base surface, which will be considered in our future work. And in the future, we intend to consider a further necessary step of relief reverse engineering: flattening the underlying surface while minimising distortion, to allow the relief to be readily reapplied to different base surfaces.

Acknowledgements

The authors wish to acknowledge the support of Delcam plc, including many helpful discussions with Richard Barratt and Steve Hobbs, and the support of EPSRC grant GR/T24425, for this work.

References

[1] H. Biermann, I. Martin, F. Bernardini, and Z. Zorin. Cut-and-paste editing of multiresolution surfaces. In *Proc. of ACM SIGGRAPH '02*, pages 312–321, 2002.

[2] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. SIGGRAPH*, pages 67–76, 2001.

[3] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proc. 3D Data Processing Visualization and Transmission*, pages 428–439, 2002.

[4] U. Dietz. Fair surface reconstruction from point clouds. In *Proc. Mathematical Methods for Curves and Surfaces II*, pages 79–86, 1998.

[5] T. Hermann, T. Varady, and Z. Kovacs. Surface fitting with ignore areas. In R. R. Martin, T. Varady, eds, RECCAD Deliverable Document 1 Copernicus Project No. 1068, Report on data acquisition, preprocessing and other tasks in 1995-1996. Report GML 1996/1, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest.

[6] J. Hoschek. Intrinsic parametrization for approximation. *Computer Aided Geometric Design*, 5(1):27–31, 1988.

[7] C. W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2003.

[8] C. W. Hsu, C.-C. Chang, and C.-J. Lin. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2004.

[9] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. European Conference on Machine Learning*, pages 137–142, 1998.

[10] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim. Support vector machines for texture classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(11):1542–1550, 2002.

[11] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proc. of SIGGRAPH '96*, pages 313–324, 1996.

[12] S. Li, J. T. Kwoka, H. Zhua, and Y. Wang. Texture classification using the support vector machines. *Pattern Recognition*, 36(12):2883–2893, 2003.

[13] P. Liepa. Filling holes in meshes. In *Proc. Eurographics/ACM SIGGRAPH Symp. on Geometry Processing (SGP'03)*, pages 200–205, 2003.

[14] S. Liu, R. R. Martin, F. C. Langbein, and P. L. Rosin. Segmenting reliefs on triangle meshes. In *Proc. ACM Symp. Solid and Physical Modeling*, pages 7–16, 2006.

[15] S. Liu, R. R. Martin, F. C. Langbein, and P. L. Rosin. Segmenting geometric reliefs from textured background surfaces. *Computer-Aided Design & Applications*, 4(5):565–583, 2007.

[16] S. Liu, R. R. Martin, F. C. Langbein, and P. L. Rosin. Segmenting periodic reliefs on triangle meshes. In *Mathematics of Surfaces XII,12th IMA International Conference, Sheffield, UK*, pages 290–306, 2007.

[17] R. R. Martin and M. M. Anguh. Estimating wound volumes from scanner data. *Machine Graphics and Vision*, 7(4):879–889, 1998.

[18] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[19] J. P. Pernot, G. F. Moraru, and P. Veron. Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Computers & Graphics*, 30(6):892–902, 2006.

[20] O. Sorkine, D. Cohen-Or, and M. Alexa. Laplacian surface editing. In *Proc. of Eurographics Symposium on Geometry Processing*, pages 175–184, 2004.

[21] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[22] V. Weiss, L. Andor, G. Renner, and T. Várady. Advanced surface fitting techniques. *Computer Aided Geometric Design*, 19(1):19–42, 2002.