COMPLEX 3D FEATURE REGISTRATION USING A MARCHING TEMPLATE

Joris S.M. vergeest, Sander Spanjaard, Chensheng Wang, Yu Song

Delft University of Technology

Faculty of Industrial Design Enginering

Landbergstraat 15, NL-2628 CE Delft, The Netherlands

http://www.dynash.org

ABSTRACT

The registration of 3D form features is essential to the supporting of reverse shape design processes. Extracting an editable shape feature from unordered data points is notoriously hard. We present a new method to extract a shape instance plus design parameters from a ridge structure contained in a freeform surface, where the ridge has no predefined path, and can have varying height and width as a function of cumulative arc-length. This complex feature is uncovered by a match-and-crawl algorithm based on partial template matching and shape dissimilarity computation. The (varying) longitudinal direction of the ridge is explicitly registered as a function of arc-length, as are its width and height. The latter two quantities are yielded as independent parameters for subsequent interactive shape modeling processes.

Keywords

Freeform shape, complex freeform features, 3D registration, shape matching

1. INTRODUCTION

3D scanning of objects is increasingly applied for the conceptualization of new products. Especially for consumer products, for which aesthetical appearance and ergonomic factors are of dominant importance, the styling and design of new shapes should happen quickly and intuitively. The designer needs many rough shape alternatives for evaluation and further development. Sometimes it is faster to manually produce a clay model of a shape, perform 3D scanning and import the data into a CAD system, instead of constructing the shape in CAD from the very start. In the motion picture industry the former workflow is increasingly often observed, and it is being more and more applied in new product design too. Techniques borrowed from reverse engineering partly support the data acquisition, registration and visual display of the reconstructed surfaces. However, these techniques primarily aim at the numerical inspection and comparison of manufactured parts against nominal CAD data.

To support conceptual shape design or redesign, it is not sufficient just to handle static models as the designer may intend to modify the shape. Simple operations, such as scaling, rotation and translation will always be possible. Local shape modifications are also feasible, as long as they are within the scope of the particular CAD system into which the model was imported. For example, if the shape was reconstructed as a NURBS surface, or as a set of NURBS surfaces, then tweaking based on control point editing will be facilitated by current systems. It turns out, however, that in general the intended shape modifications cannot be achieved directly, in a natural and interactive way, but only along a series of modeling steps, as a kind of work-around.

An important class of shape modifications is what can be achieved by the control of shape feature parameters. If a scanned object contains noticeable surface features such as holes, bumps, depressions or ridges, then the designer might want to change one of their "parameters", *e.g.* the diameter of a hole, the width of a ridge etc. Although these controls seem very obvious to the designer, they are actually absent in the model. In order to bring them in place some kind of recognition would be needed. The model must be augmented with feature information, and the augmented model must be interactively controllable by the designer. In the domain of regular, analytical shapes, this procedure has been successfully applied. However, in the domain of freeform shapes no practical solutions are available yet. The main challenge here is that the type of feature intended by (or understood by) the designer is generally unpredictable. Not until the designer points at a

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972

WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

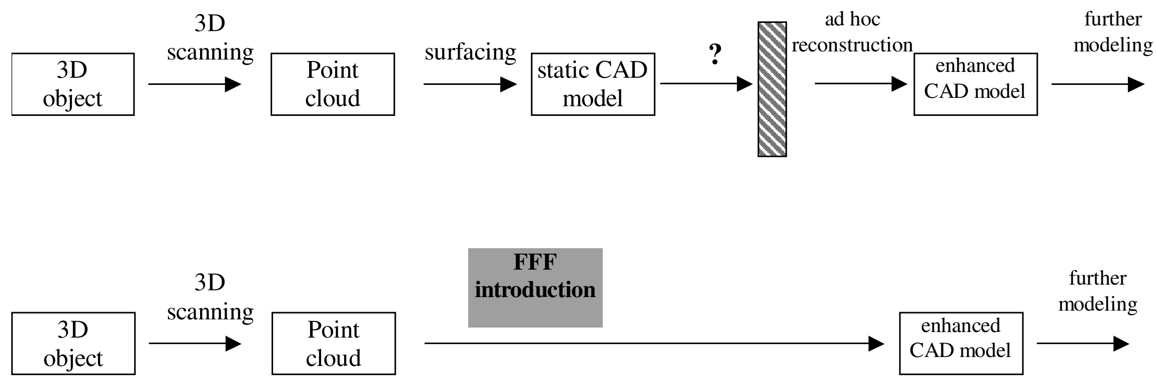

Fig. 1. Shape redesign from scanned objects using existing reverse engineering software is hampered after the generation of the static CAD model. This blocking is canceled if freeform features are explicitly recognized and embedded in the shape model.

place on the imported shape, he/she defines a region of interest (ROI) and he/she may specify in addition which type of control is needed, which implicitly specifies the type of freeform feature (FFF) that should be associated to the ROI.

Based on this information (shape, ROI in the shape, and type of feature requested for the ROI), the system should be able to put the required controls in place. If this is not provided by the system, then redesign from 3D scanned objects cannot be practical, as illustrated in Fig. 1.

The problem just described and related topics are investigated in the Dynash project at the TU Delft, see http://www.dynash.org.

This paper focuses on the introduction of FFFs to unordered set of points in 3D. Definitions and problem statement will be presented in Section 2. The solution principle will be outlined in Section 3. In Section 4 the method is applied to a relatively complex type of feature, namely the generalized ridge in surface. Conclusions are provided in Section 5.

2. PROBLEM STATEMENT AND APPROACH

We formally define the representations of the registered object, of the parameterized freeform feature and dissimilarity measures of shape in three-dimensional space $\mathbb{R}^3$.

2.1 Definitions

A shape $S \subset \mathbb{R}^3$ is a (portion of the) boundary of a compact subset of $\mathbb{R}^3$. Throughout this paper we assume that S is a 2-manifold in $\mathbb{R}^3$. Further, we assume that S is topologically equivalent to a sphere in case S covers the entire boundary of the compact subset; S is topologically equivalent to a square, otherwise. Likewise, $T(p)$ is a 2-manifold in $\mathbb{R}^3$ and $p \in P$ is called a parameter value from the parameter domain P, which can be any set. Informally, S and $T(p)$ are called 3D shapes, and $T(p)$ represents a freeform feature (FFF). The mapping $T : P \rightarrow 2^{\mathbb{R}^3}$, where $2^{\mathbb{R}^3}$ denotes the power set (the set of subsets) of $\mathbb{R}^3$, determines a family of shapes $\{T(p) \in 2^{\mathbb{R}^3} \mid p \in P\}$. T is sometimes called a FFF type.

The *Hausdorff distance* $D(T(p),S)$ between the shapes $T(p)$ and S is defined by

$$D(T(p), S) = \max\left(H(T(p), S),\ H(S, T(p))\right),$$

where $H(T(p),S)$ is the *directed Hausdorff distance* from $T(p)$ to S defined as

$$H(T(p), S) = \sup_{t \in T(p)} \left(\inf_{s \in S} |t - s| \right), \qquad (1)$$

where $|t - s|$ denotes the Euclidean distance between the points s and t. The directed Hausdorff distance $D(S,T(p))$ is defined similarly.

The *mean directed Hausdorff distance* $M(T(p), S)$ is

$$M(T(p), S) = \frac{1}{Area(T(p))} \iint_{T(p)} \inf_{s \in S} |t - s| \, dA, \qquad (2)$$

where the integration is over the surface of $T(p)$, normalized by the surface area of $T(p)$. The mean directed Hausdorff distance is sometimes preferred over the directed Hausdorff distance, as the latter is more sensitive to noise and inaccuracies in the shape data. Indeed, $H(T(p), S)$ is a worst case selection, of the point on $T(p)$ which is farthest from S, whereas $M(T(p), S)$ averages over the distances.

2.2 Problem statement

The problem can be stated in two parts:

(a) For given shape S and mapping T find the parameter value $p = p_{opt} \in P$ that minimizes the dissimilarity between $T(p)$ and any portion of S, where S is represented by a finite number of unordered points in 3D.

(b) For given p_{opt} find a new mapping $T' : Q \to 2^{\mathbb{R}^3}$ such that $T'(q) = T(p_{opt})$, or $T'(q) \approx T(p_{opt})$ where $q \in Q$ is an application-dependent parameter.

Part (a) can be regarded as an optimization problem, whereas part (b) can be classified as a representation conversion from a feature template to a user-editable feature.

2.3 Approach

An algorithm is presented that maximizes a goodness of fit of $T(p)$ to data S. Since the complexity of $T(p)$ can be very high (p can have more than 100 components, almost each influencing the intrinsic shape of template $T(p)$), the fitting procedure is subdivided according to a marching straight ridge scheme. At any time in the fitting process, maximally 8 independent parameters are non-fixed. The union of straight ridges thus found provide an initial representation of $T(p_{opt})$. Finally $T(p_{opt})$ is approximated by a smooth surface $T'(q)$ which acts as a complex FFF, furnished with high-level intrinsic controls of its height and width. The mapping T' is constructed as a NURBS representation, where its control polygon is editable by a function that implements the target application variables.

Before describing the solution in more detail we briefly review other work in this area.

2.4 Other research

Several approaches to feature extraction have been published (Loncaric 1998). The choice of approach depends heavily on the representation form of S.

If both S and T are available as boundary representations (B-Rep), *i.e.* a set of surfaces with topological and connectivity information, then graph matching techniques would be a proper way of recognizing feature T inside S. In some studies, a feature type is defined as a subgraph structure (for example blind hole, through hole, protrusion etc.). Then, matching the subgraph to a portion of S's graph is sufficient for the feature recognition process (Subrahmanyam 1995). If, besides the topological structure, geometric properties are also part of the definition of a feature, then a geometric match will be needed after the graph match was successful. In our application we are dealing with freeform shapes with either little connectivity information (in case S would be represented by a triangulation) or no connectivity information at all. Therefore, we cannot apply graph matching techniques.

Another approach to detect features in freeform surfaces is based on an analysis of local geometric properties. For example, in Várady (1997), S would be scanned for regions of lower and higher surface curvature. Then, based on a threshold criterion, it is possible to identify smooth regions surrounded by higher curvature regions. An analysis of convexity/concavity might suggest the presence of a protrusion, depression, step, ridge or bump feature contained in a smooth main surface. This method is extremely efficient and robust if the data (S) would be available as an accurate point set of high density without loss. In applications such as part manufacturing verification these conditions are met. However, if S has been measured with a low-cost scanner of limited accuracy and if noise and missing data cannot be avoided, then the analysis of local surface geometry is deemed to fail. In the application that we aim to support, also the source of S (the physical part to be scanned) may have just a preliminary shape; for example it could be made by manual claying. Therefore, surface features may already be perceptually obvious to a human observer, but not easily derived from an algorithm based on local surface properties.

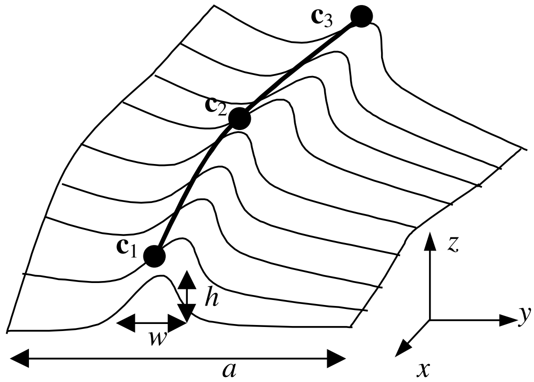

Fig. 2. Direct matching of a freeform feature template works for templates with up to 8 or 10 degrees of freedom.

To avoid or reduce the influence of noise and of missing data, template matching was proposed for feature detection. Good results were achieved for regular shaped features (Thomson 1999) and also for freeform features (Vergeest 2001). The detection is based on a macroscopic analysis of the surface data, *i.e.* by fitting a (range of) surface templates against portions of S. In this way, both the portion of S that actually contains the feature can be determined (by varying the location and position of the template), as well as the optimal intrinsic shape that the template should assume. If, in a given problem, the intrinsic shapes never change, then the problem is reduced to a 6 DOF alignment of 3D shapes. We refer to Besl (1992) for a discussion of methods to solve such registration issues. In our application, intrinsic parameters have to be included. It turned out that in practical situations, the total number of parameters should be limited to about 10. Depending on the fitting strategies (in which parameters were alternatively kept fixed or released) ridges, holes and bumps could be effectively detected in freeform surfaces. (Song 2002). However, the limitation to 10 parameters practically implies that at most 4 parameters would be left for the intrinsic shape variation of the feature, as 6 DOFs are already needed for the placement of the template in 3D. In Figure 2 this is depicted for the ridge feature. Besides the 6 DOF for its placement, the template has 9 additional variables: its width (w), height (*h*) and breadth (*a*) and the locations in 3D of the control points c_2 and c_3 (Vergeest 2001). Therefore, for features exceeding the complexity of a straight ridge (for example), direct template matching ceases to be robust.

In our paper we present an extension of the template matching procedure that enables the detection and registration of complex ridges.

3. THE MARCHING TEMPLATE ALGORITHM

The registration of the feature occurs through a concatenation of relatively simple templates $T(p_k)$, each matched to the surface *S*. The templates all belong to the same type, but each of the templates (indexed *k*) has its own placement and its own intrinsic shape. In Figure 3 this principle is schematically depicted for feature templates of type ridge.

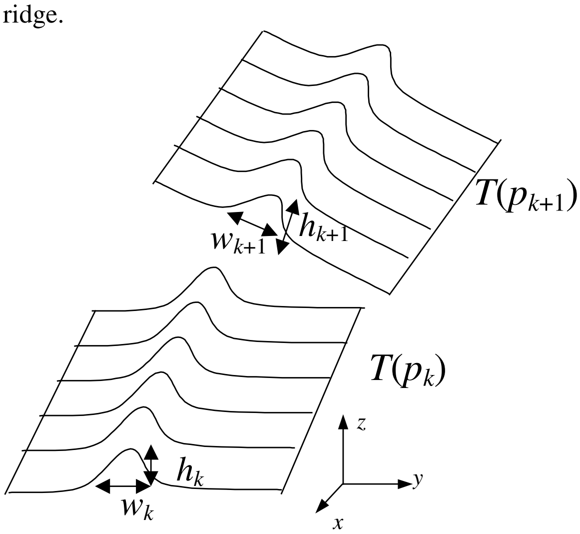

Fig. 3. Template $T(p_{k+1})$ receives its starting location and shape after $T(p_k)$ has been matched to the data.

Each of the templates $T(p_k)$ has 8 parameters; 6 DOF for placement and width w_k and height h_k. Typically, template $T(p_{k+1})$ is a follow-up of template $T(p_k)$. The position of $T(p_{k+1})$ will be approximately one feature-length further along the ridge in *S*, and the orientation of $T(p_{k+1})$ will adapt to the local direction of the ridge in the data. However, if the ridge happens to have a changing height and/or width as it develops in the surface *S*, the values h_{k+1} and w_{k+1} will be set accordingly. The process is continued until some stop criterion is reached. If the ridge in *S* has been covered by *d* templates $T(p_k)$, $k=1,...,d$, then the total structure is defined by 8*d* parameters. For *d* typically being in the order of 20, the number 8*d* is too large for an interactive conceptual shape design application. Therefore, the *d* templates are being approximated with a NURBS surface *T'*(*q*), controllable with very few parameters, down to two.

The marching ridge algorithm is outlined in Figure 4.

```
MARCHING-3D-RIDGE (S, ε)
1     Obtain the points s_j representing S
2     k ← 1
3     p_k ← GUESS
4     while not T(p_k) is flat
5     do   e ← LARGE
6          while  e > ε
7          do  Obtain n sample points t_i from
                 T(p_k)
8              e ← 0
9              for each t_i
10             do  Find s_j closest to t_i
11                   e ← e + | t_i – s_j |
12             e ← e/n
13             IMPROVE(p_k)
14         p_{k+1} ← APPEND(T(p_k))
15         k ← k+1
16    T', q, k-1 ← CONSTRUCT
17    return T', q
```

Fig. 4. Marching ridge algorithm to extract concatenated templates $T(p_k)$ and its approximation *T*(*q*).

For each *k* the template $T(p_k)$ is an instance derived from a geometric feature class or, more generally, $T(p_k)$ is a 2-manifold in some geometric representation form. In the domain of freeform shapes the most common representation forms are surfaces (including B-spline surfaces) or triangulations or surface meshes. *S* (or an ROI in *S*) may be obtained from a measurement and be represented by a set of points. However, *S* can also originate from a CAD system and hence be available in a geometric representation form.

We decided to base the dissimilarity computation (lines 7 to 13 of the algorithm) on the point sets $\{t_i \in T(p_k), i=1,m\}$ and $\{s_i \in S, j=1,n\}$, for some *m* and *n*. The distance in equation (2) has been implemented in lines 8 to 12 of the algorithm. It is the mean directed Hausdorff distance, which has been proven less sensitive to noise than the directed Hausdorff distance (Spanjaard 2001). The approximation reads

$$M(T(p_k), S) = \frac{1}{m} \sum_{i=1,m} \left(\min_{j=1,n} |t_i - s_j| \right), \quad (3)$$

where t_i and s_j are the points in $T(p_k)$ and *S*, respectively.

A matching procedure is required to obtain the best fitting template $T(p_k)$ under variation of the multi-component parameter p_k. This involves the search for $p_{opt} \in P$ defined as

$$P_{opt} = \operatorname{Arg} \min_{p_k \in P} M(T(p_k), S). \qquad (4)$$

The search, or optimization process, is denoted IMPROVE in the algorithm and performed by a general purpose optimization software IMSL.

The APPEND function generates a starting position for $T(p_{k+1})$ from the template $T(p_k)$ obtained in the previous step. For every value of k, $T(p_k)$ is represented as a simple NURBS surface containing the straight ridge (see Fig. 3). If $T(p_k)$ is determined, then the orientation and position of $T(p_{k+1})$ are chosen such that its first row of control points (the control points in a plane perpendicular to the ridge direction) coincide with the last row of control points of $T(p_k)$. The width w_{k+1} and height h_{k+1} of $T(p_{k+1})$ are initially set to the width w_k and height h_k found for $T(p_k)$. With this initial value of the multi-component parameter p_{k+1} the loop in lines 5 to 15 is reentered. If for some k a ridge is no longer detected (*i.e.* the data S is consistent with zero height), then no more templates will be considered. Finally, the k-1 templates are unified to a single NURBS surface $T'(q)$, constructed from the control points of the individual templates. The control points that determine the shape of the ridge itself (*i.e.* the control points in the center part of each row) can be proportionally shifted in such a way that the height and the width of the ridge in $T(q)$ is locally increased or decreased with a particular rate. This latter function is included for shape modeling purposes.

4. IMPLEMENTATION AND RESULTS

The first numerical experiment was for data S obtained from a synthetic model, see Fig. 5. The surface contains a spirally formed ridge with changing height and width. Using a sampling algorithm, points on S were generated as input to the algorithm. As mentioned, the algorithm was designed for unordered data points, so the sampling order in S had no effect on the outcome of the algorithm. The fitting process was triggered by the user who points with the cursor to S near the starting point of the ridge.

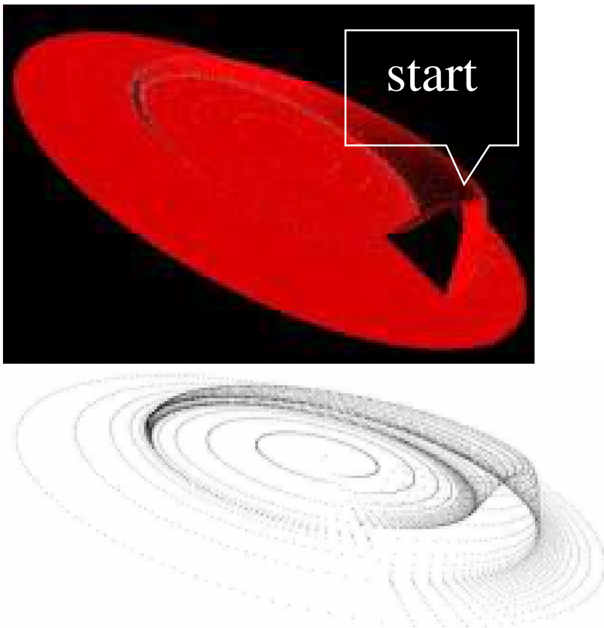

Fig. 5. A ridge (original surface and sampled points) along a circular path with monotonically decreasing height and width.

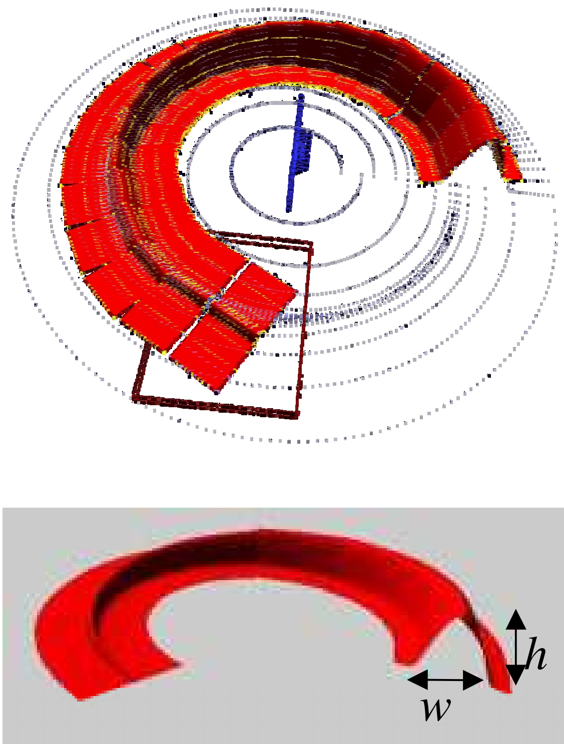

Fig. 6. Top: The series of 20 straight ridge templates making up the total feature structure. Bottom: The 19 features are merged as to form a single FFF, controllable with two design parameters, *h* and *w*.

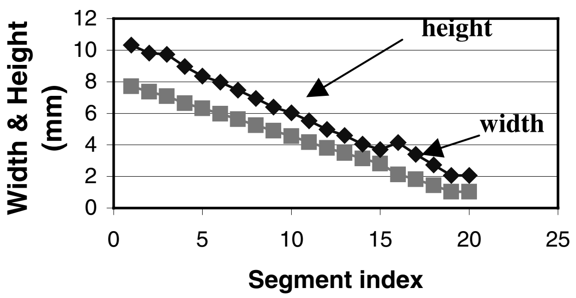

Fig 7. Feature parameters height and width as a function of segment index.

This procedure is motivated by the target application of the feature registration; it is the user who decides that a particular aspect (feature) in *S* is relevant for later reuse, so it can be assumed that he/she points at that feature. Once the initial position and orientation of $T(p_k)$, for k=1 is reached, the Marching-3D-Ridge Algorithm operates autonomously, and finds all subsequent templates making up the total ridge. However, in some runs, the algorithm did not detect the ending of the ridge, so the user had to force termination. The results of the fits are displayed in Figs. 6.

In the following experiment *S* was obtained by 3D scanning of an existing part (Fig. 8) using a mechanical coordinate measuring machine. The point set obtained was relatively accurate and dense. The algorithm yielded a series of templates (Fig. 9) from which a single NURBS surface could be constructed (Fig. 10). The measured height and width of the ridge as a function of cumulative arc-length is consistent with being constant (Fig. 11), in accordance with the shape of the feature on the physical part. As mentioned, the height and width of the entire ridge become editable by the designer.

Fig. 8. A physical part containing ridges in the upper surface.

Fig. 9. One of the ridges, as detected by the algorithm.

Fig. 10. The set of straight ridge templates merged into one NURBS surface.

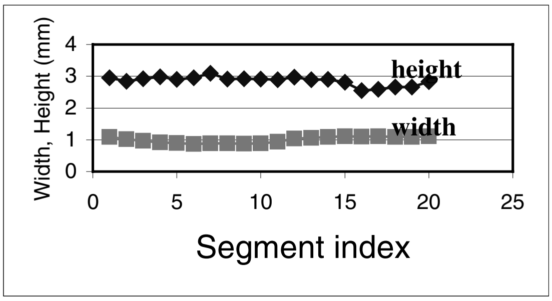

Fig 11. The width and height of the ridge are consistent with being constant as a function of segment index.

Finally, we tested the algorithm with data obtained from a manually made object (Fig. 12).

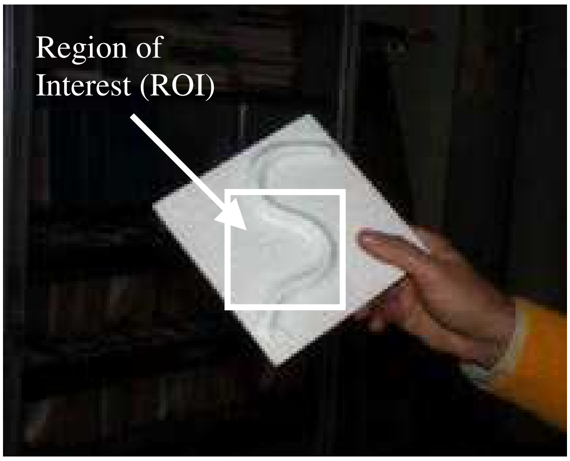

Fig. 12. A manually made clay model of a ridge in a flat surface.

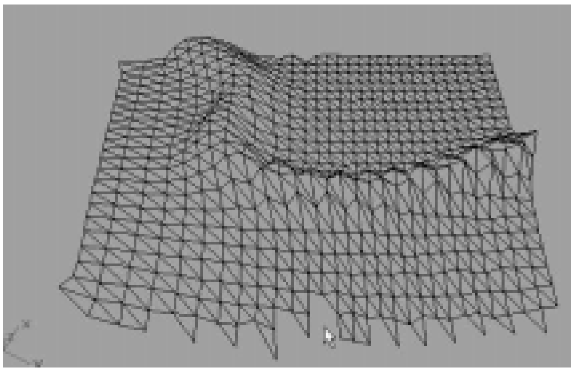

Fig. 13. Data points obtained with the desktop laser scanner from the object depicted in Fig. 12.

Both the object itself and the measuring method introduced significant inaccuracies. The 3D scanner was a low-cost desktop device, which was fast (appr. 20 min. for a scan) but of low resolution, appr. 1 mm, where the diameter of the ROI was about 100mm. Despite the sparseness of the data points (Fig. 13), the algorithm was able to capture the ridge and to register the feature attributes as a function of arc-length (Fig. 14).

Typically the computation time of a ridge match takes about 50 seconds, which amounts to about 10 to 20 minutes for the entire ridge, if the ridge is registered with 10 to 20 straight ridges.

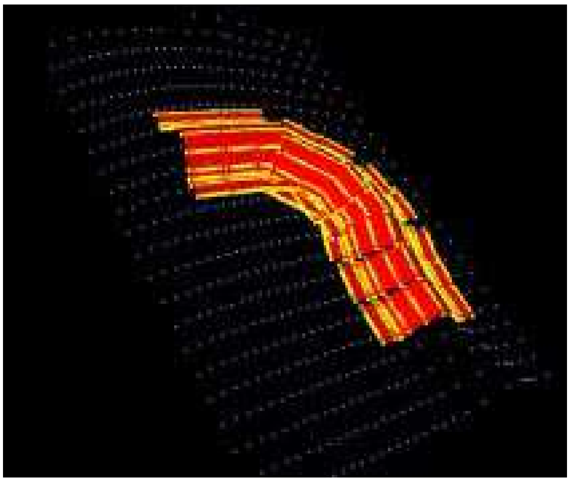

Fig. 14. Straight ridges marching along the ridge feature in the ROI indicated in Fig. 12.

5. CONCLUSIONS AND FURTHER WORK

We have presented an algorithm to register snake-shaped ridges in a freeform surface. The method is based on the matching of multiple 8 DOF straight ridges to 3D point data. The purpose of the algorithm is to extract shape feature information from existing physical or digital shape models for subsequent use in new designs. Even if the source data is very sparse and inaccurate, the characteristic shape can be retrieved, and be converted into a representation suited for further manipulations. In another research in the Dynash project a method has been developed to copy and paste the feature information into a new solid or surface model (Wang 2002). Once the feature has been inserted into the new model (Wang 2002), the parameters should still be operating in order to adapt the ridge to the new design.

Several improvements to the algorithm are currently being investigated. The performance of the mean directed Hausdorff distance computation (line 9 to 12 of the algorithm in Fig. 4) has been dramatically improved by limiting the search in line 10 to a rectangular region in S in the vicinity of t_i. Since in average the distance between subsequent test points t_{i+1} and t_i is small the rectangular region needs relatively few updates.

Another problem under investigation is occasional aliasing due to possible uniformity of the samples from S and $T(p_k)$. When one of the sampling frequencies is a multiple of the other sampling frequency, then $M(T(p_k), S)$ as a function of one of the components of p_k can be undulating, causing the minimizer to get trapped in a local minimum. This problem is being cured by taking the average of the two (or more) nearest points in S to t_i, rather than the nearest point.

Other related topics, currently under investigation, are the extension of the method to other feature types and to intersecting features.

6. ACKNOWLEDGEMENT

The authors thank Tim van Bentum and Vaniek Colenbrander for their excellent work and experiments.

7. REFERENCES

[Loncaric 1998] S. Loncaric, "A survey of shape analysis techniques", Pattern Recognition, Vol. 3, No. 8, 1998, pp. 983-1001.

[Besl 1992] P.J. Besl and N.D. McKay, "A method for registration of 3-D shapes". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp 239-255.

[Song 2002] Y. Song, J.S.M. Vergeest, I. Horváth, "Reconstructing freeform surface with parameterized features". Proc. of the 2002 DETC conference, DETC'02/DAC-34036, ASME, New York, 2002.

[Spanjaard 2001] S. Spanjaard and J.S.M. Vergeest, "Comparing different fitting strategies for matching two 3D point sets using a multivariable minimizer" Proc. of the 2001 Computers and Information in Engineering Conference, DETC'01/CIE-21242, ASME, New York, 2001.

[Subrahmanyam 1995] S. Subrahmanyam and M. Wozny, "An overview of automatic feature recognition techniques for computer-aided process planning". Computers in Industry, Vol. 26, 1995, pp 1-21.

[Thompson 1999] W.B. Thompson, J.C. Owen, H. James de St. Germain, S.R. Stark and T.C. Henderson, "Feature-based reverse engineering of mechanical parts", IEEE Tran. Robotics and Automation, Vol. 15, No 1, 1999, pp. 57-66.

[Váradi 1997] T. Váradi, R.R. Martin and J. Cox, "Reverse engineering of geometric models - an introduction", Computer-Aided Dseign, Vol. 29, No. 4, 1997, pp. 255-268.

[Vergeest 2001] J.S.M. Vergeest, S. Spanjaard, I. Horvath, J.J.O. Jelier, "Fitting Freeform Shape Patterns to Scanned 3D Objects". Journal of Computing and Information Science in Engineering, Transactions of the ASME, Vol. 1, Nr. 3, pp 218-224, 2001. IMA-CNR, IEEE, Piscataway, 2001, pp. 20-29. See also http://www.dynash.org.

[Wang 2002] C. Wang, J.S.M. Vergeest, I. Horváth, R. Dumitrescu, T. Wiegers, Y. Song, "Cross-model shape reuse: copying and pasting of freeform features". Proc. of the 2002 DETC'02/DAC-34064, ASME, New York, 2002.