Automatic fitting and control of complex freeform shapes in 3-D

Y. Song
Delft University of Technology
2628CE, Delft, The Netherlands
y.song@io.tudelft.nl

J.S.M.Vergeest
Delft University of Technology
2628CE, Delft, The Netherlands
j.s.m.vergeest@io.tudelft.nl

C. Wang
Delft University of Technology
2628CE, Delft, The Netherlands
c.wang@io.tudelft.nl

ABSTRACT

In many computer graphics and computer-aided design problems, it is very common to find a smooth and well structured surface to fit a set of unstructured 3-dimensional data. Although general approaches of fitting give satisfactory results, the computation time and the complexity often prevent their further developments in more complex cases especially in reusing an existing design. In this paper, for a better control of existing freeform shapes, they are approximated by feature templates, with emphasis on extendable templates. By the advantage of the small number of intrinsic parameters in the feature based deformable templates, fitting procedures are faster and more robust. Three key types of simple freeform templates, the bump, the ridge and the hole, are introduced first. With distance measuring methods, a uniform optimization function is presented to achieve automatic feature recognition and fitting. By introducing the extendable template, the hole and the ridge template are further developed to match complex freeform shapes. Based on the approximated template, further shape manipulations can be conducted effectively using the shape intrinsic parameters. Numerical experiments are conducted in order to verify the proposed algorithms. It is also described how the matching technique can be applied in computer graphics and computer-aided design applications.

Keywords

Freeform shape, fitting, control, template, estimations

1. INTRODUCTION

Finding effective and efficient methods to fit and reconstruct a set of unstructured 3-Dimensional (3-D) data is one of the state-of-the-art topics in Computer Graphics (CG) and Computer-Aided Design (CAD). In particular, in the domain of 3-D scanning data reuse, one often needs to solve a fitting problem in order to build a smooth and well structured surface either to reconstruct or to get a better control of the shape, especially in the freeform area.

3-D shape matching to, possibly sparse, inaccurate or otherwise degraded, freeform shape data is known to be hard. Over the past two decades, many fitting problems have been formulated as the minimization of an optimization function corresponding to a shape model. Those applications include quadratic surface fitting [Chi93], B-Spline surface fitting [Sak91], rotational surface fitting [Mot94], loft surface fitting [Lin97] and sweep surface fitting [Uen98]. Many of those approaches dealt with problems in a particular case, where a large number of linear systems always were involved. Thus it cannot always yield satisfied solutions in many complex cases. Comparing to those methods, parametric deformable templates, such as superquardrics or hyperquardrics templates [Sol90] [Bar98], offer the advantage of a relative small number of parameters. For deformable template fitting, the minimization of the optimization function associated to the model is done in a reduced space of admitted solutions. Due to their small number, choosing the parameters of the template is a crucial issue of the goodness of the fitting.

In general, current fitting algorithms offer a good approximation of the model, but their application for reusing an existing design is still an unsolved problem especially in CAD applications. The key issue is the non-uniqueness of the types of parameters for a given geometric object. However, modifying higher level entities in a geometry surface

WSCG'2004, February 2-6, 2003, Plzen, Czech Republic.

is much easier than operating on geometric constituents such as points, curves and individual surfaces [Ver01]. But currently, if designers want to change some high-level parameters, such as the height of a bump in a reconstructed freeform shape, the whole shape would have to be regenerated first to make those parameters available. For solving this problem, the feature concept was recently introduced to the freeform shape area.

Generally, a feature is a generic shape of a product with which designers can associate certain attributes and knowledge useful for reasoning about the product. Feature offers the advantage of treating sets of elements as single entities, thus improving the efficiency in creating the product model [Sha95]. While the concept of feature has been mainly investigated in mechanical environment, it was also introduced to freeform area. A freeform feature is a portion of a single or a set of freeform surfaces. The boundary of the feature may consist of curve segments that lie within a surface [Li00]. Unlike for mechanical features, for freeform features, a clear, unique boundary cannot always be specified. In 1999, a freeform feature taxonomy was proposed by Fontana *et al.* [Fon99]. In their work, according to the shape and different contributions to the freeform surface, the detail freeform features were divided into two main categories: shape deformation features and shape elimination features.

With a parameterized freeform feature template, freeform shape information can be directly recognized and transferred into high-level parameters [Chi95][Var97]. Thus, with freeform feature template, Li and Hui [Li00] developed a two-phase approach to recognize freeform feature from B-rep model. For freeform feature parameterization, Surazhsky and Elber [Sur01] developed a metamorphosis process, which is defined as gradual and continuous transformation from one key shape into another. With four boundary curves, Piegl and Tiller [Pie01] presented a parameterization method for a given point set.

In this paper, *matching existing freeform shapes with templates defined by high-level parameters in 3-D space is studied for parameter-driven freeform deformation* [Son03]. Commonly, parameters can be efficiently used for shape manipulation only if the model was built according to a specific framework of parameters. If designers need a particular shape manipulation method controlled by an "intuitive" parameter, but the parameter was never built in before, then the wanted method cannot be provided, conventionally. In order to give designers the ability to introduce new shape parameters at runtime, an algorithm has been developed to modify a shape using templates approximating that shape. Designers can apply one of many shape templates to any surface portion of a shape model. At first, the template conforms maximally to the selected surface portion of the shape. By a mapping between the template and the shape, the parameters belonging to the template become available to modify the original shape model. That is, *the template is used as a bridge between the known parameters and the unknown shape*. By inverse mapping, the deformation of the template is easily transferred to the freeform shape.

To achieve this purpose, the similarity of the template and the freeform shape should be optimized to as close as possible. In the presented fitting algorithms, considering the target application and the existing techniques of shape matching and reconstruction, freeform feature templates are introduced as the basis of the deformable model. Base on mean direct Hausdoff distance, the template is fitted to a given shape with a given optimization function. Analyzing the fitting results, automatic feature recognition can be achieved with a standard deviation-like function. To complex shapes, an extendable concept is proposed base on the simple feature templates. Finally, the shape matching algorithms and shape control methods were tested and verified by numerical experiments.

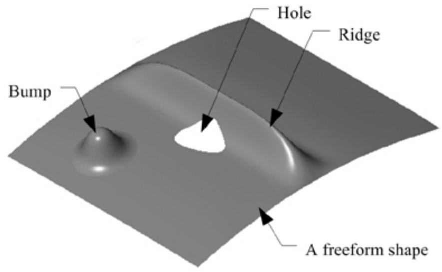


Figure 1: A freeform surface containing several freeform features

2. FREEFORM FEATURE AND FETURE PARAMETERIZATION

In Fontana *et al.*'s work [Fon99], freeform features were divided into two main categories: shape deformation and shape elimination features. Then, according to the features' shape and contribution to the surface, they were classed to step-up, step-down, cavity, bump, n-groove, n-rib, hole, inlet and outlet. In this paper, a bump, a ridge and a hole feature are studied as the basis. Various kinds of freeform shape can be approximated by those features [Son02]. Typical examples of these three freeform features are presented in **Figure 1**. In the figure, a given freeform surface contains three features: a bump, a hole and a ridge, where the bump is an isolated feature and the ridge happens to be interfered by the hole.

Since no clear boundary can be defined for freeform features, the parameterization of a freeform feature is usually based on intrinsic shape information, such as height and width of a bump. Here, the concept of freeform feature *template* is introduced as an idealization of a freeform feature. A freeform feature template is an isolated freeform feature with additional information. It not only contains all the shape characteristics of a specified freeform feature, but also captures the information of shape elimination, *e.g.* the area of the hole(s). Furthermore, an adjustable clear boundary, such as a rectangle or a circle, can also be assigned to a template. For example, in **Figure 2**, a hole feature template with a rectangular outer boundary is defined.

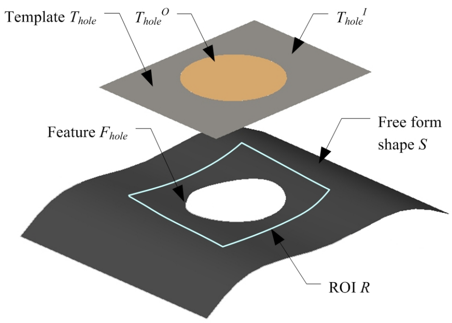
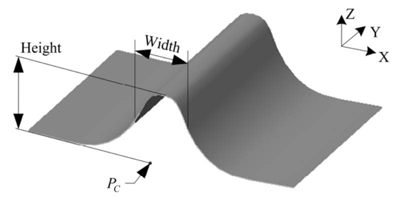
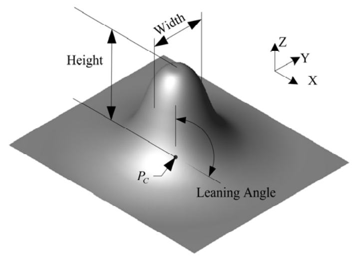


Figure 2: Surface, ROI, feature instance and feature template

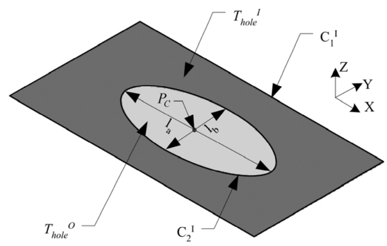
Using freeform feature templates to approximate a given freeform shape, the relations of the shape, Region Of Interest (ROI), freeform feature and freeform feature template are shown in **Figure 2**. Given a shape $S \subset \mathbb{R}^3$ as in **Figure 2**, the ROI $R \subseteq S$ is selected as being a candidate for containing a freeform feature F_{hole}. Here, the ROI can be treated as a provisional boundary of the freeform feature. For matching the feature in the ROI, a feature template of type t (here is the hole) is defined by a mapping: $T_t : Q_t^T \rightarrow 2^{\mathbb{R}^3}$, where $2^{\mathbb{R}^3}$ is the power set (i.e. the set of all subsets) of $\mathbb{R}^3$ and $Q_t^T = Q_1^T \times Q_2^T \times \cdots Q_i^T \cdots \times Q_m^T$, which is the parameter domain of T_t. Here, Q_i^T represents the domain of a continuous scalar variable q_i. For every given $q \in Q_t^T$, $T_t(q)$ specifies a subset in $\mathbb{R}^3$ referred to as a freeform feature template. Generally, a freeform feature template may contain two major parts $T_t^I(q) \subseteq T_t(q)$ and $T_t^O(q) \subset T_t(q)$, such that $T_t(q) = T_t^I(q) \bigcup T_t^O(q)$. Here, $T_t^I(q)$ represents the portion of the template that, in a matching procedure, would be similar to the feature. However, $T_t^O(q)$, would be similar to a shape *not* contained in the feature. Normally, $T_t^O(q)$ can be considered a sub-template characterizing eliminated surface data. For shape deformation feature templates, usually $T_t^O(q) = \emptyset$. When a hole template $T_{hole}(q)$ is applied for matching with a hole in a freeform surface, $T_{hole}^I(q)$ of the feature template ought to surround the hole whereas $T_{hole}^O(q)$ (light grey area) ought to locate itself in the void of the surface S (**Figure 2**).



(a) Ridge



(b) Bump



(c) Hole

Figure 3: Parameterization of feature templates

For template matching, a mathematical surface representation of freeform feature template should be generated based on high-level parameters. The complexity of a template depends on the application at hand. Certainly, the more parameters are selected, the more flexible a feature template is. But problems appear as follows: first, too many parameters might confuse the designer who preferably works with five to eight parameters, second, more parameters will make the optimizing procedure much more difficult or even impossible. In this paper, several freeform feature templates are parameterized according these requirements. First, templates for matching shape deformation features are studied, where a ridge template (**Figure 3a**) and a bump template (**Figure 3b**) are taken as representatives. Then, a hole template (**Figure 3c**) is investigated as a typical example of a shape elimination feature template. Detail mathematical representations of those templates can be found in the authors' former works [Son02].

3. SIMILARITY MEASUREMENT AND COMPUTATION

For fitting a freeform shape with a deformable template, a similarity measurement is needed to determine the goodness of fit between the original shape and the freeform feature template. There are many similarity measurements defined in the literatures[Hag99]. In the proposed method, to reduce the sensitivity to noise and inaccuracies in the shape data, Mean Directed Hausdorff Distance (MDHD) [Ver01] is introduced. Given any two shapes A and B, MDHD of the two shapes can be defined as, $M(A,B)=\iint_A \inf_{r\in A,s\in B}|r-s|\,dA/\iint_A dA$, where the integration is over the surface of A, normalized by the surface area of A.

With the dissimilarity measure M, the matching problem can be extended to search among multiple feature template types $T_t\ (q)$ (the learning set) to find the best fit to a freeform feature in R of shape S. Then, the matching procedure aims at obtaining the proper parameters of the feature template $T_t\ (q)$ under variation of the parameters $q_{opt}\in Q_t^T$, where Q_t^T is the fitting parameters domain, and

$$q_{opt}=\operatorname{Arg}\min_{q\in Q_t^T} f_d^{MDHD}(T_t^I(q),T_t^O(q),R,\lambda), \qquad \textbf{(1)}$$

where $f_d^{MDHD}(T_t^I(q),T_t^O(q),R,\lambda)=M(T_t^I(q),R)-\lambda M(T_t^O(q),R)$ and $\lambda>0$. Function f_d^{MDHD} is named optimization function, which is applied as the objective function in the fitting procedure.

Equation (1) delivers the feature instance of type t that matches R optimally. In the Equation, $M(T_t^I(q),R)$ measures the dissimilarity between part $T_t^I(q)$ of the feature template and R. When the feature template fits the surface, according to the definition, $M(T_t^I(q),R)$ will be minimal. $M(T_t^O(q),R)$ measures the dissimilarity between $T_t^O(q)$ and R. However, since $T_t^O(q)$ represents an eliminated part of a shape elimination feature, the term $M(T_t^O(q),R)$ should become maximal. By scalar coefficient λ, the "weights" of $M(T_t^I(q),R)$ and $M(T_t^O(q),R)$ can be adjusted in the overall similarity measurement. The whole match procedure can be accelerated by setting λ different in different stages of the fitting. Thus, a matching procedure is then simplified to search for the optimized parameters $q_{opt}\in Q_t^T$ of the feature template.

4. FEATURE RECOGNITION

Freeform features do not have a clear boundary, and sometimes intrinsic characters of a feature are not clear enough. Such as, a stretched bump is similar to a ridge. Thus, identifying the feature types in an existing freeform shape and selecting a proper template are crucial topics for shape fitting. In this section, instead of the designers' help, an automatic feature recognition algorithm is proposed to find the suitable template.

In template matching, when parameters of a template are optimized to an optimal referring to a R of a shape, the MDHD from the R to $T_t^I(q)$ is supposed to be a minimum. Given a pointset shape $P^R=\{P_j^R\in R\mid j=1,a\}$, which represents the R of the shape, and a template $T_t\ (q)$, by the fitting methods, the template can be put in an optimal position. In this position, the Hausdoff distance [Ver01] from each point in R to $T_t^I(q)$ part of the template can be measured as $HD^{RI}=\{H(P_j^R,T_t^I(q))\mid j=1,a\}$.

Given a standard deviation-like function $\sigma=\sqrt{\sum_{i=1}^{n}x_i^2/(n-1)}$ for a real number set $x=\{x_i\mid i=1,n\}$, σ delivers the measure of variability of set x referring to 0. Thus, $\sigma(HD_i^{RI})$ means the variability of the Hausdoff distance from each point in R to the template. Suppose a series of templates $T_{t_i}(q)$, $t=\{t_i\mid i=1,m\}$, are used to fit a R, in each optimal position, $\sigma(HD_i^{RI})$ can be computed. By the shape similarity analysis, the feature type can be recognized as $t_i=\operatorname{Arg}\min_{t_i\in t}\sigma(HD_i^{RI})$. With automatic feature recognition, ***Algorithm 1***, which represented in a C liked language, is presented as the template fitting methods.

Algorithm 1

```
fitting_a_freeform_shape
{
    R = input_shape();
    // n is the number of template types
    for( t = 0 ; t < n ; t ++)
    {
        // q_i is the parameter of each template
        T_t (q_i) = init_parameters();
        //using quasi-newton method to optimize
        q_i = fitting_low_density_digitized_template
        ( f_d^MDHD (T_t (q_i), R ) );
        σ_t = standard_deviation(HD( T_t (q_i), R ));
    };
    //find template type t
    t = arg_min_item( σ );
    fitting_high_density_digitized_template
    ( f_d^MDHD (T_t (q_t), R));
    out_put( q_t , f_d^MDHD (T_t (q_t), R));
}
```

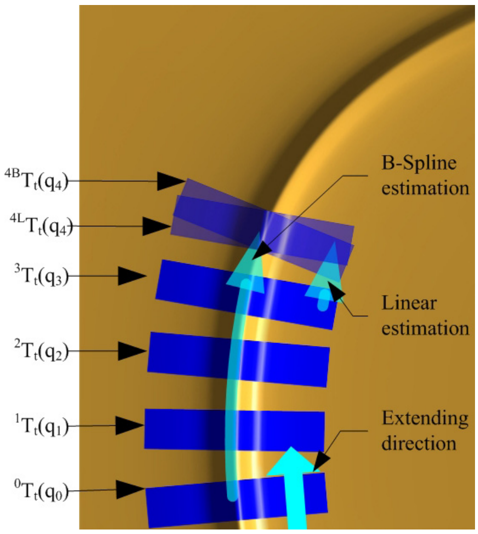


Figure 4: Definitions of the extendable template

5. EXTENDABLE TEMPLATES

Sweep and loft are common modeling actions in freeform shape design. Conventional methods require many user actions and a huge number of linear systems to fit such complex shapes. In the presented research, an extendable freeform template concept is proposed to solve the fitting problems.

Given ROI R of a freeform S, suppose R is much more larger than a template $T_t(q)$ and R contains a sweep or a loft feature, a template $T_t(q)$ is used to fit the shape first at a given position. This position can be taken as the first profile of the feature, recorded as ${}^0T_t(q_0)$ in **Figure 6**. By a given extending direction, a same type template, ${}^1T_t(q_1)$, is defined and fitted to the shape. Repeating this progress, an extendable template is created as

$$ {}^ET_t = \{{}^iT_t(q_i) \mid i = 1, n\} . \quad (2)$$

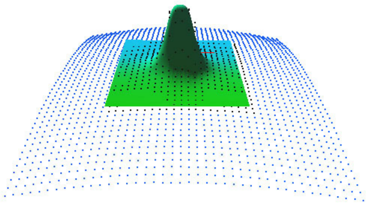
Interpolating templates ${}^iT_t(q_i)$ in **Equation 2**, a uniform surface of the extendable template is created.

For estimating the next template position, two algorithms, the linear estimation and the B-Spline estimation, are proposed. Suppose ${}^0T_t(q_0)$, ${}^1T_t(q_1)$, ${}^2T_t(q_2)$ and ${}^3T_t(q_3)$ are template positions which already been found for a complex shape as **Figure 4**, in linear estimation, the next template initial position is determined by final found template, such as ${}^3T_t(q_3)$ in **Figure 4**. For finding next template initial position and orientation, template ${}^3T_t(q_3)$ is offset along its extending direction by a given distance to ${}^{4L}T_t(q_4)$, which is taken as the initial estimation. In B-Spline estimation, all the position and orientation information of ${}^0T_t(q_0)$, ${}^1T_t(q_1)$, ${}^2T_t(q_2)$ and ${}^3T_t(q_3)$ is used. By interpolating those positions, a 3-D curve is got as the figure. Extending this curve along the extending direction with the given length, the next template estimation position, ${}^{4B}T_t(q_4)$, is found, where the derivate of the curve in this position is defined as the orientation of the template. Normally, B-Spline estimation is more effective than linear estimation, which always be used in the estimations of initial stages. ***Algorithm 2*** summarized the extendable template fitting method.

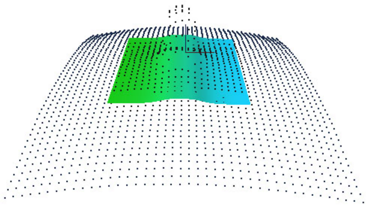
Algorithm 2:

```
fitting_a_shape_with_extendable_template
{
    R = input_shape();
    i = 0;
    pick_start_point();
    decide_template_width( ^iT_t (q_i));
    //automatic feature recognition
    ^iT_t (q_i) = find_a_proper_template(t);
    do{
    //find local ROI to save fitting time
    R_Local = finding_region_close_to_template( R )
    fitting_tempate( f_d^MDHD ( ^iT_t (q_i), R_Local ))
    add_current_template( ^iT_t (q_i), ^ET_t )
    i ++;
```

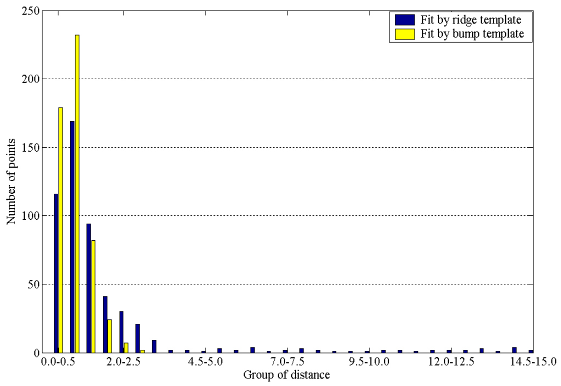
estimate_next_template_position(${}^{E}T_t$, ${}^{i}T_t\ (q_i)$)
// G is a gate value, when template goes out of R
// it will stop the loop
}while($f_d^{MDHD}\ ({}^{i}T_t\ (q_i), R) < G$)
interpolate_templates_and_out_put (${}^{E}T_t$);
}



(a) Fitting a bump template to a bump like shape



(b) Fitting a ridge template to a bump like shape



(c) The distribution of Hausdoff distance from each point in the shape to the template at the optimal position

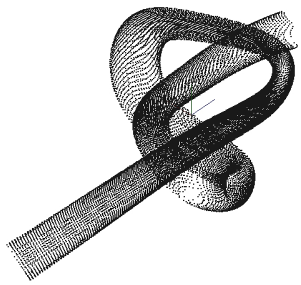
Figure 5: Automatic feature recognition

6. NUMERICAL EXPERIMENTS

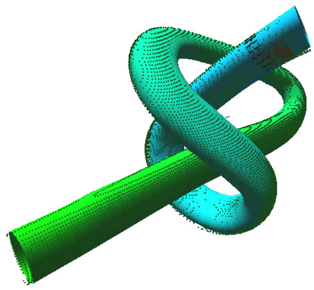
In this section, numerical experiments are presented in order to verify the proposed fitting theory. Using ACIS®, the whole system is modeled by Visual C++® and search procedures were conducted by help of the IMSL® C numerical libraries.

To verify the automatic feature detection algorithm, a simple experiment was taken as shown in **Figure 5**. In **Figure 5a** and **Figure 5b**, a bump-like shape was fitted by a bump and a ridge template, respectively. The distribution of the Hausdoff distance from each point in the shape to the template at the optimal position is shown in **Figure 5c**. In the experiments, to achieve a faster solution, the template was digitized to a relative low density (5x5) in the automatic feature recognition process. With a Pentium III 1.2GHZ processor, the recognition process was conducted in less than 60seconds. From the results, it is clear that the fitting result of the bump template is much better than the ridge template. σ in the two cases were calculated as 0.857 and 4.311 corresponding to the bump and the ridge template. Thus, it is concluded that the shape is a bump feature.

Figure 6 shows a complicate case of the proposed template matching. With a given pointset (**Figure 6a**), a extendable hole template was selected to fit it. By 57 steps, the shape was fitted in 2 hours with the same processor as **Figure 6b**.



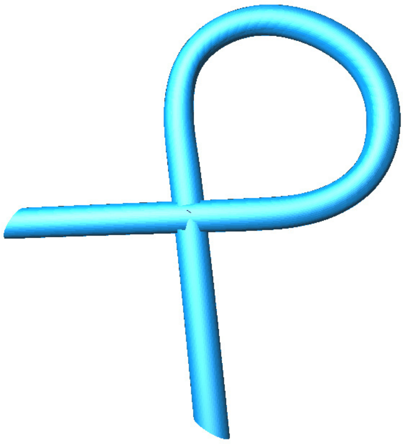
(a) A pointset freeform shape



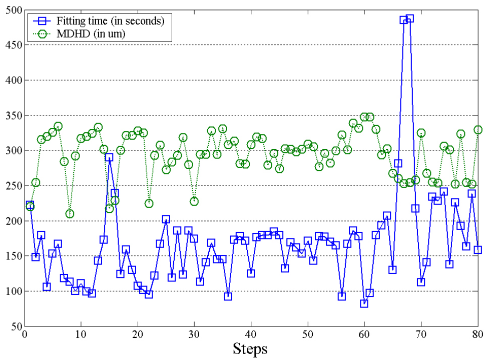
(b) Fitting result

Figure 6: Fitting a pointset with an extendable hole template

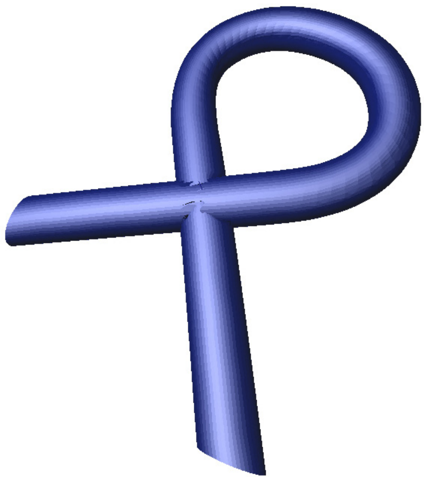
To test the robustness of the proposed template fitting, another experiment was carried out in a more complicate case. In **Figure 7a**, a self-intersection tube model is shown. With the proposed extendable template, the tube was fitted in 80 steps. In **Figure 7b**, fitting times and optimization function values of each step are presented. From the figure, it is shown that in step 15 and step 67-68, the fitting time were much longer than the others since the jumbled data in the intersection region. On the country, the optimization function values in those positions are always lower than the neighbors because the jumbled data also offers the advantage to find a better solution of the template position. With the fitting result, the intrinsic parameters of the tube was found, thus further shape modification can be easily performed. In **Figure 7c**, the diameter of the tube was increase 80%, where in **Figure 7d**, a wave-like function was performed on the shape of the tube, In **Figure 7e**, the modified shape was stretched to straight.



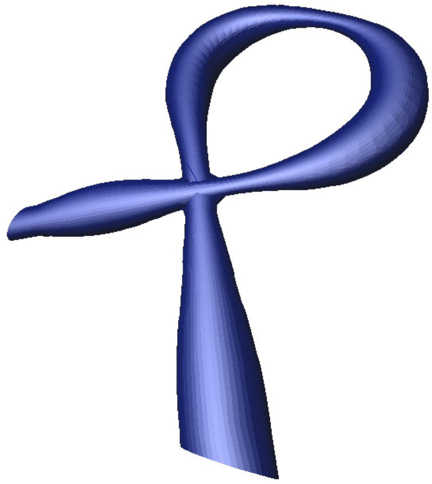
(a) Original self-intersection tube model



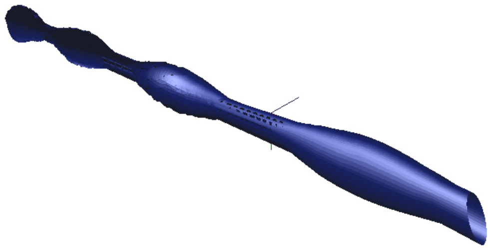
(b) Fitting time and MDHD



(c) Increasing the diameter of the tube



(d) Adding wave-like effect on the tube model



(e) Stretching the tube model to straight

Figure 7: Fitting a self-intersection shape

7. CONCLUSION

A method of direct fitting freeform surface by parameterized freeform feature templates has been presented in this paper. Definitions and implementation forms of two shape dissimilarity measures in full 3-D space were compared. An optimization function has been presented based on the mean directed Hausdorff distance to cope with noisy and incomplete data samples. Based on a standard deviation-like function σ, freeform features can be automatically recognized. For complex freeform shapes, an extendable template concept is developed. Numerical experiments were conducted on different kinds of freeform shapes. The fitting and deforming results indicate that directly matching freeform surface with 3-D digitized freeform feature templates can be applied as a tool in reusing an existing design.

Current research is directed towards more complicated conditions of matching freeform features. Research of the extension to different types of shape templates is on-going. Different estimation functions of extendable freeform template are also being studied in order to match more complicated freeform shapes.

8. ACKNOWLEDGMENTS

The presented research is a part of the Dynash project conducted in Faculty of Industrial Design Engineering, Delft University of Technology. (www.dynash.tudelft.nl) This research project DIT.6071 is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs, The Netherlands..

9. REFERENCES

[Bar98] Bardinet, E. and Cohen L. D., A parametric deformable model to fit unstructured 3D data, Computer vision and image understanding, 71(1) 39-54, 1998.

[Chi93] Chivate, P.N. and Jablokow, A.G., Solid-model generation from measured point data, Computer-aided Design, 25(9), 587-600, 1993.

[Chi95] Chivate P. N. and Jablokow A. G., Review of surface representations and fitting for reverse engineering, Computer Integrated Manufacturing Systems, 8 (3), 193-204, 1995

[Fon99] Fontana M., Giannini F. and Meirana M., A freeform feature taxonomy, Proceeding of Eurographics '99, 18 (3), 1999

[Hag99] Hagedoorn M. and Veltkamp R. C., Reliable and efficient pattern matching using an affine invariant metric. International Journal of Computer Vision, 31(2/3), 203-225, 1999.

[Li00] Li C. L. and Hui K. C., Feature recognition by template matching, Computer & Graphics, 24, 569-583, 2000.

[Lin97] Lin, C. Y., Liou, C. S. and Lai, J.Y. A surface-lofting approach for smooth-surface reconstruction from 3-D measuring data, Computer in Industry, 34, 73-85, 1997.

[Mot94] Motavalli, S. and Bidanda, B., Modular software development for digitizing system data analysis in reverse engineering applications:case of concentric rotational parts, Computer industrial Engineering, 26 (2), 395-410, 1994.

[Pie01] Piegl L. A. and Tiller W., Parametrization for surface fitting in reverse engineering, Computer-Aided Design, 33 (8), 593-603, 2001

[Sak91] Sakar, B. and Menq, C.H., Smooth-surface approximation and reverse engineering, Computer-aided Design,23 (9), 623-628, 1991.

[Sha95] Shah J. J. and Mantyla M., Parametric and featured based CAD/CAM, Wiley-Interscience Publication, John Wiley Sons In., 1995

[Sol90] Solina, F. and Bajcsy, Recover of parametric models from range images: The case for superquadrics with global deformations, IEEE transactions on pattern analysis and machine intelligence, 12(2), 131-147, 1990.

[Son02] Song Y., Vergeest J. S. M. and Horvath I., Feature interference in free form template matching, Proceeding of EuroGraphics, short presentations, 2002.

[Son03] Song Y., Vergeest J. S. M. and Saakes D. P., Parameter-driven freeform deformation, Proceeding of EuroGraphics, short presentations, 2003.

[Sur01] Surazhsky T. and Elber G., Matching free-form surfaces, Computers & Graphics, 25 (1), 3-12, 2001.

[Uen98] Ueng, W.D., Lai J. Y. and Doong J. L., Sweep-surface reconstruction from three-dimensional measured data, Computer-aided design, 30(10), 791-805, 1998.

[Var97] Varady T., Martin R. R. and Cox J., Reverse engineering of geometric models--an introduction, Computer-Aided Design, 29 (4), 255-268, 1997

[Ver01] Vergeest J. S. M., Spanjaard S., Horvath I. and Jelier J. J. O., Fitting Freeform Shape Patterns to Scanned 3D Objects. Journal of Computing and Information Science in Engineering, Transactions of the ASME, 1 (3), 218-224, 2001

[Ver02] Vergeest J.S.M, Wang C., Song Y. and Horvath I., "Dynamic Shape Typing". Proceedings of Computers and Information in Engineering, DETC'02, CIE, ASME, New York, 2002