Sketch Template Based Parametric Modeling in Reverse Engineering

Yongqing Li [1], Xiaoping Huang [1], Chunhe Gong [2] and Jianming Zheng [1]

[1]GE Global Research Center, China, Catcher.Li@geahk.ge.com
[2]GE Capital – Rail Services, U.S.A., michael.gong1@ge.com

ABSTRACT

In this paper, a sketch template based parametric modeling technology is proposed in the field of reverse engineering. With this technology, a sketch template representing the key features of a family parts' profile is firstly established while capturing design intentions through constraints imposed among the geometric objects. Then this sketch template is matched and best fitted to the cross sectional points of the profile as well as satisfying all the constraints. The final parametric solid model can be created based on the optimized sketches. The main contribution of this paper is the comprehensive study on the sketch optimization with a broad extension on both types of engineering constraints (including geometric and dimensional constraints) and geometric objects (including point, line, arc, conic and B-spline curve). The real examples have demonstrated that this technology is of great significance and success in practical industrial applications.

Keywords: Sketch template, parametric modeling, constraints, sketch optimization

1. INTRODUCTION

Because of data noise from the measurement and complexity of the physical part, the reconstructed composite model by a pure re-engineering method generally far deviates from the original part and is hardly acceptable by practical industrial applications. Furthermore, forward designers always include particular engineering rules in the product [1]. If such rules are ignored in reverse engineering, the reconstructed model will be useless for real CAD/CAM purposes. Hereby, capturing designer's intentions is of paramount importance to obtain an intelligent model in reverse engineering, which is not only best fitted to the point cloud, but also achieves specific engineering requirements.

An efficient way of representing design intentions in CAD model is to transform the engineering rules into various constraints including geometric constraints [2-4] (geometric relationship between objects), dimensional constraints as well as some engineering constraints like functional gradient, material strength and machining parameters [5, 6]. Recent research progress shows that constraints based object reconstruction is a rapidly evolving topic in reverse engineering [7, 8].

The reconstructed surface model is often bounded by a huge number of sub-surfaces to keep features from original design intentions. Because of the continuity requirements and other constraints, it becomes extremely difficult for users to edit or modify the composite surface model. Additionally, for tons of similar parts from the same family, if each part is handled individually, it will become a very time-consuming and tedious job to reconstruct all the models. So in order to improve the modeling efficiency for a part family and conduct iterative modifications on a re-engineering model, the idea of sketch template based parametric modeling is introduced into conventional reverse engineering in this paper. The parameterization idea was originally innovated for forward conceptual design and the scope was only restricted to some simple geometric entities [9]. The integration of reverse engineering and parametric modeling will enable reconstruction of an accurate 3D solid more convenient to edit by changing parameters according to some specified engineering rules.

Sketch template design is one of the paradigms that are explored for parametric modeling. For most current CAD systems, a typical modeling process firstly starts from a series of rough sketches, which do not stand for the exact dimensions and locations of the geometric objects, but represent the basic topology of the geometric shape. Then some constraints derived from design intentions are imposed among the rough sketch. The final solid model is created based on the adjusted sketches through some feature operations such as extruding, revolving and lofting, etc. This model is called parametric model since it can be edited easily by modifying the parameters of the sketches without essential change on the topology.

In this paper, the authors systematically discussed sketch template design on the basis of designers' intentions and sketch optimization based on planar points. Compared to previous work on 2D constrained fitting [7], the conic curve and B-spline curve are taken into account as the basic geometric objects to represent the features. Moreover, the concept of sketch template based parametric modeling is firstly introduced into the area of reverse engineering, which provides good initial estimates of the features for further sketch optimization.
The organization of the rest paper will be as follows: the next section discusses the engineering constraints based sketch template design. Section 3 describes the numerical method for sketch optimization. In section 4, some examples are showed how this concept works in real industrial applications. Lastly, some conclusions are drawn in section 5.

2. SKETCH TEMPLATE REPRESENTATION

During the initial product design stage, the sketch plays an important role for the designers to express their ideas in forming a target CAD model. Currently, most commercial CAD systems provide such a sketch interface for the users to realize parametric modeling interactively. Opposite to forward conceptual design, sketches can also be utilized to represent the profile features and capture the design intentions in reverse engineering.

2.1 Topology Representation

As mentioned in the introduction, a sketch template just represents the basic geometric topology of the features, but does not indicate the exact dimensions and locations. In this paper, the sketch template T is defined as a set of n serialized 2D geometric objects g_i with m particular constraints s_j, noted as:

$$T(S \to G) := \{(s_j, g_i) \mid f(s_j) = s_j[r_j] \subseteq G, \forall g_i \in E \,\&\, s_j \in C\} \ (i = 0...n-1; j = 0...m-1),$$

where G is the set of serialized 2D geometric objects g_i in a specific sketch template T;

E is the set of basic 2D geometric objects including point, line, arc (or circle), conic curve (ellipse, parabola, hyperbola) and B-spline curve (see section 2.2);

S is the set of constraints in a specific sketch template T;

C is the set of basic constraints including dimensional constraints and geometric constraints (see section 2.3);

$f(s_j)$ is the set of geometric objects associated to the constraint s_j;

r_j is the number of geometric objects associated to the constraint s_j.

With the above definition, the sketch template can be concisely represented by the sets of G, S and $f(s_j)$. For example (Figure 1), a sketch template composed of one line segment and one circular arc with one coincident constraint and one tangent constraint can be represented as:

$$n = 2, m = 2; G = \{g_0, g_1\} = \{line, arc\}; S = \{s_0, s_1\} = \{coincident, tangent\}; r_0 = 2, r_1 = 2;$$

$$s_0[0] = g_0, s_0[1] = g_1; s_1[0] = g_0, s_1[1] = g_1.$$

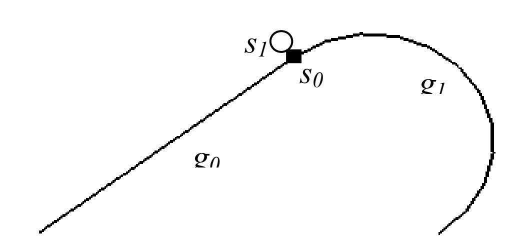


Fig.1. An example of sketch template

2.2 Geometry Object Representation

Normally, the profile features of a part can be expressed by a sketch template made up of a combination of point, line, arc, conic and B-spline curve.

2.2.1 Point

A point $\mathbf{P}$ is represented by its coordinates (p_x, p_y) relative to the coordinate system of the sketch template. The point could be a primary object or an auxiliary object connecting the primary ones.

2.2.2 Line

A line is described by a vector of $\mathbf{x}_l = [l_0 \quad l_1 \quad l_2]^T$ using the usual implicit equation [7]:

$$l(x, y) = l_0 x + l_1 y + l_2 = 0, \tag{1}$$

which is subject to a normalization constraint $l_0^{\ 2} + l_1^{\ 2} - 1 = 0$. Its normal vector is $\{l_0, l_1\}$ and its distance to the origin is l_2. For a line segment, two extra end points are needed to specify the length.

2.2.3 Circular Arc (Circle)

A circular arc is represented by the same equation of a circle [10]:

$$c(x, y) = c_0(x^2 + y^2) + c_1 x + c_2 y + c_3 = 0, \tag{2}$$

where the vector $\mathbf{x}_c = [c_0 \quad c_1 \quad c_2 \quad c_3]^T$ defines the circle. Under the normalization condition of $c_1^{\ 2} + c_2^{\ 2} - 4c_0 c_3 - 1 = 0$, the circle's radius is $1/|2c_0|$ and its center is $(-c_1/2c_0, -c_2/2c_0)$.

2.2.4 Conic

A general conic is represented by an implicit second order polynomial [11]:

$$e(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{3}$$

where $\mathbf{x}_e = [a \quad b \quad c \quad d \quad e \quad f]^T$ is the parameter vector. In order to fit the general conic into a specific type, the parameter vector should be constrained in some way. The well-known *discriminant* $4ac - b^2$ is an appropriate one and here the equality constraint is imposed on the *discriminant* to force the conic to be a specific type:

if $4ac - b^2 - 1 = 0$, the conic is an ellipse;

if $4ac - b^2 + 1 = 0$, the conic is a hyperbola;

if $4ac - b^2 = 0$, the conic is a parabola.

2.2.5 B-spline curve

A B-spline curve of order *k* (*degree* + 1) can be defined as [12]:

$$\mathbf{s}(u) = \sum_{i=0}^{n} N_{i,k}(u)\ \mathbf{V}_i, \qquad u_{k-1} \le u \le u_{n+1},\ n \ge k-1, \tag{4}$$

For the sketch, only planar B-spline curve is considered and the vector $\mathbf{x}_s = [V_{0x} \quad V_{0y} \quad \ldots \quad V_{nx} \quad V_{ny}]^T$ is used to describe the B-spline curve.

2.3 Constraints Classification and Representation

The constraints in a sketch template can effectively represent the designer's unique intentions or specific engineering requirements. Basically, the set of constraints associated with a sketch template can be divided into two categories: geometric constraints and dimensional constraints. By modifying the constraints, the designers can precisely control the dimensions, locations and geometric relationships of the objects in a sketch template.

2.3.1 Geometric Constraints

Similar to Werghi's classification [8], there are two types of geometric constraints: intrinsic constraints and extrinsic constraints. An intrinsic constraint establishes the geometric characteristics or properties of a single object in a sketch template (for example, requiring that a line have a constant length). Below are some basic intrinsic constraints in a sketch template:

- Normalization constraint for a line or an circular arc;
- Modified discriminant constraint for a conic;
- Horizontal – Defines a line parallel to the X-axis of the sketch template's coordinate system;
- Vertical – Defines a line parallel to the Y-axis of the sketch template's coordinate system;

The extrinsic constraint defines the geometric and topological relationships between two or more geometric objects. In this paper, we studied various constraints that are of great importance for multiple curves simultaneous fitting in a sketch template.

- Coincident – Defines two points as having the same location in a sketch template;
- Tangent – Defines two objects as being tangent to each other;
- Parallel – Defines two lines as being parallel to each other;
- Perpendicular – Defines two lines as being perpendicular to each other;
- Equal Length – Defines two line segments as having the same length;
- Equal Radius – Defines two arcs as having the same radius;
- Collinear – Defines two lines as lying on or passing through the same straight line;
- Concentric – Defines two circular and elliptical arcs as having the same center;
- Constant Angle – Defines two lines as having a constant intersection angle.

A comprehensive summary of extrinsic constraints between two geometric objects can be found in Table 1. Further explanation is needed for some certain cases: A coincident constraint means the endpoints of the curves are at the same location. A tangent constraint between two neighboring objects in a sequential sketch template includes an inferred coincident constraint on the common endpoints.

For the constraint involving more than two objects, it can generally be decomposed into several sub-constraints between two ones. For example, a constraint of "three lines are parallel" can be divided into two parallelism constraints between two lines. Some of the constraints involving line segments and circular arcs are thoroughly studied by Benko et al [7]. And as shown in Table 1, the coincident and tangent constraints are the most frequently happened cases between two geometric objects. So in this paper, we are mainly focusing on these two types of constraints associated with a conic or a B-spline curve. Apart from the method in [7], the tangent constraint is decomposed into two sub-constraints: coincident and equal slope, so that both the coincident and tangent constraints between any two types of curves can be handled in the same way. Again, for the tangent constraint involving a conic, the direct deduction of constraint relationship will result in a high order equation, which will increase the difficulty of derivatives calculation and influence the robustness of the numerical method. But if a auxiliary point $\mathbf{P}\ (p_x, p_y)$ is introduced as the tangent point, then tangent constraint can be expressed as two simple conditions: 1) The two curves both pass through the auxiliary point; 2) The slopes of the two curves at the auxiliary point equal to each other.

	Point	Line	Arc	Conic (Ellipse)	B-spline
Point	Coincident	Coincident	Coincident	Coincident	Coincident
Line	Coincident	Coincident Parallel Perpendicular Equal length Collinear Constant angle	Coincident Tangent	Coincident Tangent	Coincident Tangent
Arc	Coincident	Coincident Tangent	Coincident Tangent Equal radius Concentric	Coincident Tangent Concentric	Coincident Tangent
Conic (Ellipse)	Coincident	Coincident Tangent	Coincident Tangent Concentric	Coincident Tangent Concentric	Coincident Tangent
B-spline	Coincident	Coincident Tangent	Coincident Tangent	Coincident Tangent	Coincident Tangent

Tab. 1. Extrinsic Geometric Constraints between Two Objects

In order to process G1 continuity for a B-spline curve, the first derivatives on both endpoints are calculated as:

$$\mathbf{s}'(0) = \frac{k-1}{u_k}(\mathbf{V}_1 - \mathbf{V}_0); \tag{5}$$

$$\mathbf{s}'(1) = \frac{k}{1-u_{n-k}}(\mathbf{V}_n - \mathbf{V}_{n-1}). \tag{6}$$

The tangent constraint involving a conic and B-spline can be represented as shown in Table 2.

Tangent between conic and line:	
Conic passes through auxiliary point:	$ap_x^2 + bp_x p_y + cp_y^2 + dp_x + ep_y + f = 0$
Line passes through auxiliary point:	$l_0 p_x + l_1 p_y + l_2 = 0$
Equal slope at auxiliary point:	$l_0(2cp_y + bp_x + e) - l_1(2ap_x + bp_y + d) = 0$
Tangent between conic and circular arc (circle):	
Conic passes through auxiliary point:	$ap_x^2 + bp_x p_y + cp_y^2 + dp_x + ep_y + f = 0$
Arc passes through auxiliary point:	$c_0(p_x^2 + p_y^2) + c_1 p_x + c_2 p_y + c_3 = 0$
Equal slope at auxiliary point:	$(2c_0 p_x + c_1)(2cp_y + bp_x + e) - (2c_0 p_y + c_2)(2ap_x + bp_y + d) = 0$
Tangent between B-spline and line	
Line passes through B-spline endpoint:	$l_0 V_{0x} + l_1 V_{0y} + l_2 = 0$
	Or $l_0 V_{nx} + l_1 V_{ny} + l_2 = 0$
Equal slope at endpoint:	$l_0(V_{1x} - V_{0x}) + l_1(V_{1y} - V_{0y}) = 0$
	Or $l_0(V_{nx} - V_{(n-1)x}) + l_1(V_{ny} - V_{(n-1)y}) = 0$
Tangent between B-spline and circular arc (circle):	
Arc passes through B-spline endpoint:	$c_0(V_{0x}^2 + V_{0y}^2) + c_1 V_{0x} + c_2 V_{0y} + c_3 = 0$
	Or $c_0(V_{nx}^2 + V_{ny}^2) + c_1 V_{nx} + c_2 V_{ny} + c_3 = 0$
Equal slope at endpoint:	$(2c_0 V_{0x} + c_1)(V_{1x} - V_{0x}) + (2c_0 V_{0y} + c_2)(V_{1y} - V_{0y}) = 0$
	Or $(2c_0 V_{nx} + c_1)(V_{nx} - V_{(n-1)x}) + (2c_0 V_{ny} + c_2)(V_{ny} - V_{(n-1)y}) = 0$

Tab.2. Tangent Constraint Representations

2.3.2 Dimensional Constraints

In real industrial applications, some parts may have specific dimensional requirements on some key features. For example, the diameter of a hole or the width of a slot must be within a dimension tolerance because of the assembly requirement. Dimensional constraints define the size of a single geometric object or the relationship between two geometric objects in a sketch template. A dimensional constraint looks much like a labeled dimension on a 2D drawing. According to different attachment objects and dimension types (length and angle), we classify the dimensional constraints as follows:

- Point-to-Point: Defines the oriented distance between any types of two points such as individual point, curve endpoint or midpoint, circle or ellipse center point and B-spline control points, etc. The distance orientation could be along X-axis direction or Y-axis direction of the sketch template, a line normal direction or a given direction;

- Point-to-Line: Defines the distance between a point and a line. The line could be an existing line or an auxiliary line;
- Line-to-Line: Defines the distance between two parallel lines;
- Radius/Diameter: Defines the radius or diameter for an arc or circle;
- Angle: Defines the intersection angle between two nonparallel lines.

Compared to the geometric constraints, the dimensional constraints for sketch optimization are more difficult and complex to represent because of various expressions attached to different geometric objects. This problem is beyond the topic of this paper and it is one of the feature research areas we are going to involve.

3. SKETCH OPTIMIZATION

Given a set of cross sectional points measured from a part's contour or sliced from a point cloud, the goal of sketch optimization is to best fit an initial estimate of the features to the points, while taking into account the constraints among the geometry objects.

3.1 Objective Function

Assume that the profile features of a part be expressed by a sketch template composed of basic geometry objects such as line, arc and B-spline curves. The parameters describing all those geometry objects in a sketch template form a parameter vector $\mathbf{x}$, whose values are to be estimated in the optimization process. Note $F(\mathbf{x})$ to be the objective function identifying the deviation from the sketch to the 2D point set $\mathbf{P}_i(i=1...n)$. So the vector $\mathbf{x}$ has to be optimized to minimize the objective function $F(\mathbf{x})$ while satisfying the constraints. Here the 2D point set has been segmented into different curve segments before optimization. Take the equations of $C_k(\mathbf{x})=0 \quad (k=1...m)$ to describe constraints where $C_k(\mathbf{x})$ is the constraint function associated with constraint k. In practice, the tolerance of constraint function is not zero but assigned with a certain value according to specific requirements. So the above problem to be solved can be stated as:

Minimize $F(\mathbf{x})$ while subject to $C_k(\mathbf{x}) < \xi_k$, ($k=1...m$).

Here ξ_k is the tolerance attached to the constraint C_k.

For the constrained fitting problem stated above, a modified Newton iteration method is devised to meet the case of multiple constraints in [7]. This method can get satisfying result particularly when the constraints involve redundant or inconsistent equations and the initial value is not so good. But for our case, the sketch template offers a good initial estimated parameter vector $\mathbf{x}_0$, in which all the constraints are satisfied before optimization. To solve this constraint optimization problem, we have reduced it into a sequence of unconstrained minimizations by incorporating a set of penalty functions into the objective function. That means, the above problem can be solved by minimizing a convex function [8]:

$$E(\mathbf{x}) = F(\mathbf{x}) + \sum_k \lambda_k (C_k(\mathbf{x}))^2, \tag{7}$$

where the parameter vector $\mathbf{x}$ need to be found with a set of appropriate penalty factors λ_k.

With the equation (7), the sketch optimization is actually an iterative process of numerical calculation, so the optimization result and computation time are in great extent influenced by the representation of objective function, the goodness of initial iteration value, and the robustness of numerical method. All these factors require further study and must be chosen appropriately. In the following section, we will discuss how to choose the criteria for the objective function

3.2 Distance Representation

When evaluating the accuracy of model reconstruction in reverse engineering, the true Euclidean distance, know as geometric distance, is one of the most reasonable criterion to measure the deviation between the object and its associated points. But in general, the geometric distance has very complex non-linear representation, which often increases the difficulty of numerical calculation and leads to expensive computational cost. In order to get a concise distance representation, various methods have been proposed to approximate the true Euclidean distance in which the method with faithful algebraic distance [13] is the most popular one. With the formulation of faithful algebraic distance, the objective function can be represented by a compact quadratic vector expression:

$$F(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} + B\mathbf{x} + C \ , \tag{8}$$

where $\mathbf{x}$ is the parameter vector describing geometric objects. A , B and C are data matrixes determined only by the data points, but nothing with the objects. Under the above form, the objective function includes separate terms of geometric parameters and data points. The data matrixes A , B and C can thus be computed off-line before optimization. Furthermore, the Jacobian and Hessian matrixes of the objective function can be evaluated quickly as follows:

$$F'(\mathbf{x}) = A\mathbf{x} + B^T, \ E''(\mathbf{x}) = A \ . \tag{9}$$

3.3 Initial Sketch Estimate

The initial estimate of the features is critical to guarantee the convergence and efficiency of the numerical method toward a desired optimal solution. When faced with such a problem, many researchers [7, 8] all assume that the data points be segmented into a series of sub-sets attached to different geometric elements. Then each sub-set of the points is best fitted with a corresponding geometric object separately in the absence of constraints. The combination of the individual fitted geometric objects will be used as an initial estimate for later shape optimization. Because the initial estimate created with this method does not include the constraints, the topological relationship between the geometric objects probably deviate significantly from the optimization target so that it will cost more time to converge or may fail to enter the convergence domain.

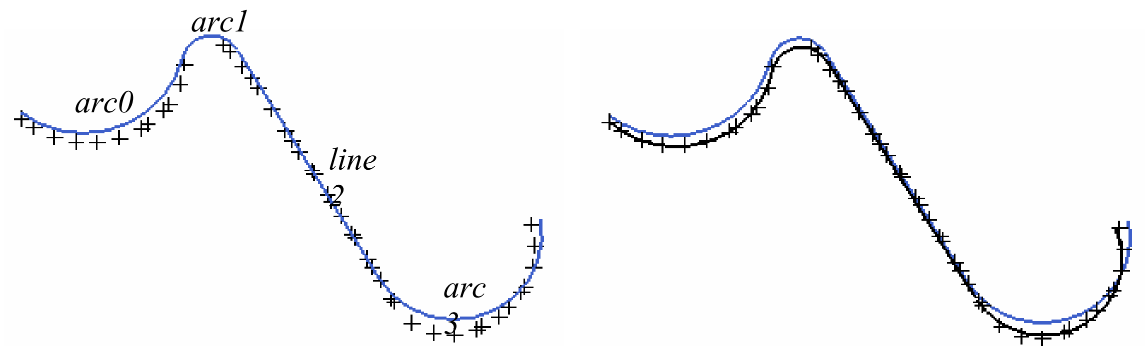
In this paper, a sketch template library is established in advance to provide good initial estimates for shape optimization. Such a sketch template represents the basic geometry topology, but it does not stand for the exact size and location with the real profile features. Meanwhile, all the engineering rules have been translated into dimensional or geometric constraints imposed among the geometry objects in the sketch template. When conducting the iterative optimization process, a corresponding sketch template is firstly scaled and rotated to match the cross sectional points. Then the data points are segmented according to the sketch template topology and the curvature variation law. At the first iteration, the value of constraint functions is almost equal to zero and only the objective function is minimized, so the initial searching direction of the solution is closer to the accurate convergent direction, which can accelerate convergence speed. Later on, the distance error is reduced gradually under the condition of satisfying all constraints until the result satisfies the convergence criteria. With aid of sketch template, the optimization process becomes more robust and efficient to reach the convergence.

Normally, the sketch template can be built in two different ways. For the parts with 2D drawings or 3D model available, the sketch template can be easily designed according to geometric topology descriptions with dimensions and constraints noted in the specification. For those parts without any information, the designers can construct the sketch template based on the measurement data points according to their background knowledge. On the basis of understanding of the parts and feature recognition from the data points, each segmented sub point set can be fitted with a corresponding geometric object separately. Afterward, all the individual objects are trimmed and combined into a single curve chain. At last, the constraints are enforced among the geometric objects. Such a curve chain with constraints can be used as the sketch template for a type of part family.

3.4 Numerical Algorithm

By introducing a set of penalty factors λ_k, the constrained fitting problem has been reduced to sequential unconstrained non-linear minimizations. In order to get an appropriate trade-off between the distance error and constraint error, the initial penalty factors λ_k should be carefully chosen. For our cases, since the constraints are satisfied strictly in the sketch template, the initial values λ_k should not be too big so that the sketch can better approximate the profile points.

When using *Levenberg-Marquardt method* to solve the unconstrained non-linear optimization problem (7), the Hessian matrix will become ill conditioned while the penalty factors λ_k become too large. Faced with this problem, N. Weigh [8] improved the method proposed by Broyden et al. [14] to overcome the numerical instability by applying different weights on the constraints. This method can work and get good results in most cases. But based on our observing and tracking on the iteration process, we have found that the deviation between the geometric objects and their associated coordinate system has a big impact on the convergence of the algorithm. For some cases that the

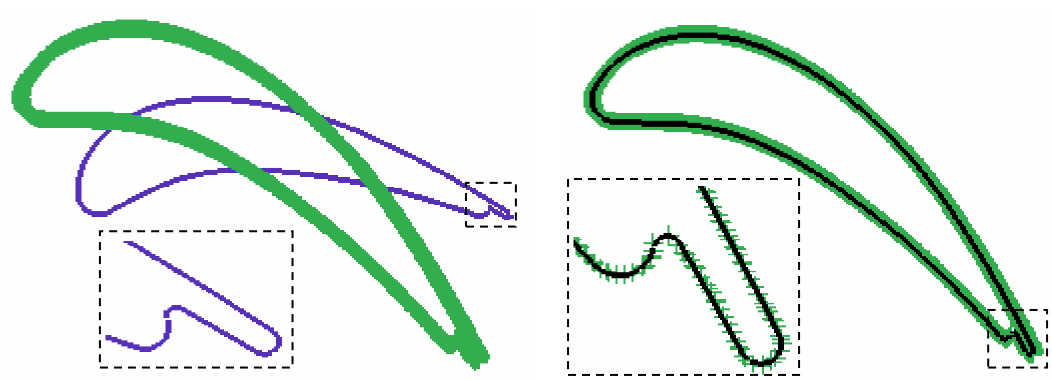


(a) Initial sketch of a trailing edge (b) Optimized sketch w/ and w/o transformation

Fig. 2. Optimization of the trailing edge of a turbine blade's airfoil section

Fitting Error	Initial sketch	Optimal sketch	Iterations
Without CSYS Transformation	0.070028mm	0.063743mm	39
With CSYS Transformation	0.070028mm	0.015342mm	19

Tab. 3. Error analysis of sketch optimization



(a) Topology of an airfoil's profile section (b) Optimized sketch (Black)

Fig. 3. Sketch optimization of a turbine blade's airfoil profile section

sketch template is far away from its associated coordinate system, sometimes it is very difficult to get a satisfied optimized solution. However, if we move the working coordinate system to the bounding box center of the sketch, we can obtain a high-quality optimization result with less iterations and high-speed convergence. Figure 2(a) shows the trailing edge feature of an airfoil section, which is represented by 3 circular arcs and one line segment with tangency between any two adjacent curve segments. Figure 2(b) and Table 3 shows the comparison between the optimization results with (Black color) and without (Blue color) coordinate system transformation. Obviously, the optimized sketch after transformation is closer to the point set.

4. EXAMPLES AND APPLICATIONS

Some simple and complex experiments are performed here to show how the optimization process is conducted with the sketch template based parametric modeling technology when handling the 2D real inspection data points. Moreover, some analysis results regarding to the behavior and convergence of the numerical algorithm are presented.

Presently, the inspection data can be obtained by different modalities. With the touch probe or optical sensor, we can get accurate data points of the parts' external profiles. CT can penetrate the physical part to capture the internal features, but the accuracy is not very high. Sometimes, the users just want to measure some key feature points to

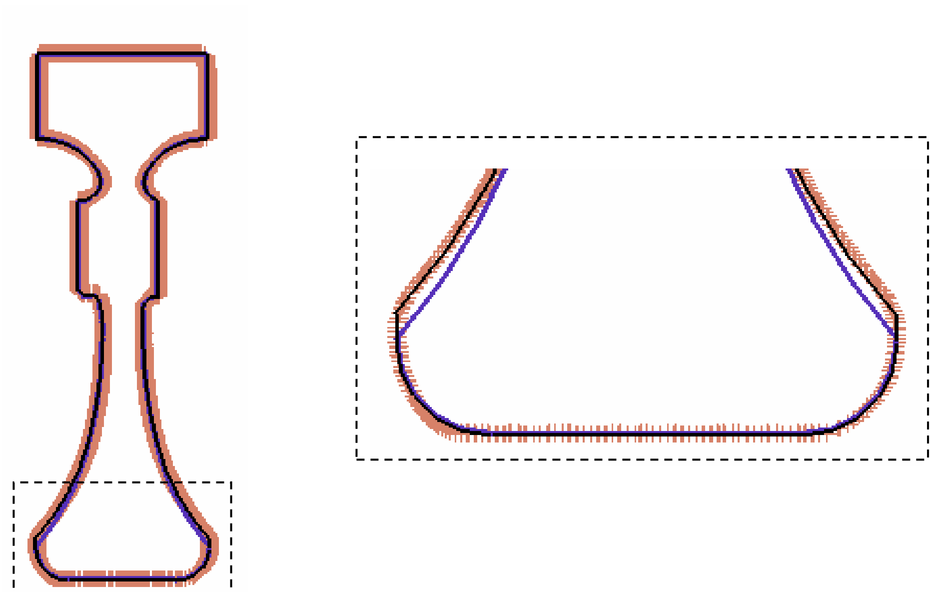


Fig. 4. Initial and optimized sketch (Black)

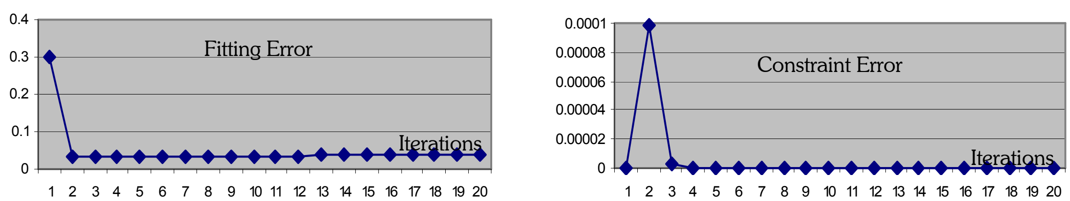


Fig. 5. Variation of fitting error and constraint error with respect to iterations

quickly outline the profiles for later measurement guiding or planning. Under this situation, some areas may have no points and the features can only be created by the constraints associated with others.

Figure 3 (a) shows the sketch template of a typical airfoil section, whose pressure and suction surfaces are represented by B-spline curves and the trailing edge is expressed by a combination of 3 circular arcs and one line segment and the leading edge is normally a circular arc. According to aerodynamic requirements, the G^1 continuity must be at least satisfied between any two adjacent curve segments. In Figure 3 (b), it is obvious that the optimized sketch (black color) is closer to the point set. The average distance error is significantly reduced from *0. 05743mm* to *0. 01172mm* and all the constraint function values are satisfied within the tolerance of *10e-12*.

The goal of the next example is to analyze and measure the efficiency and robustness of the algorithm. As shown in Figure 4, it is a sketch template of a disk's section profile with further complexity on the number of curve elements, geometric constraints and inspection data points. This sketch includes totally 28 curve segments of line and circular arc with coincident constraints between any two adjacent objects. Additionally, some other constraints such as parallelism, perpendicularity, horizontality and verticality are imposed among the sketch objects. The total number of geometric constraints is 58. The inspection data was obtained by laser point sensor with a high density of 5500 points.

Before conducting the sketch optimization process, the sketch template is firstly scaled, rotated and moved to match the inspection data with a rough registration. Then the roughly registered sketch will be taken as the initial value for later constrained fitting. The zoom view of Figure 4 shows the comparison result between the roughly registered sketch and optimized sketch. As shown in the figure, the optimized sketch (black color) has moved closer to the inspection data points while meeting all the constraints. The average distance error is significantly reduced from *0. 03236mm* to *0. 01087mm* and all the constraint function values are satisfied within the tolerance of *10e-12*.

Figure 5 shows the variation and the behavior of least squares fitting error and the sum of constraint functions error during the optimization with respect to the iterations. Because all the constraints are satisfied in the sketch template initially, it is noticed that the least squares error is decreasing quickly and the constraint functions error increases to a relatively large value during the first iteration. From the second iteration, the least squares error is slightly increasing and stabilizes to a low value. On the other hand, the error of constraint functions is decreasing gradually until it reaches the given tolerance (*10e-12*).

5. CONCLUSIONS

This paper presented a sketch template based parametric modeling technology in reverse engineering. In order to represent forward design intentions, the engineering rules are reduced into geometric or dimensional constraints imposed on the sketch template for a part family. The model reconstructed from the point cloud based on the sketch template is thoroughly parameterized and can be edited and modified conveniently according to the user's specific requirements.

The authors also comprehensively described the representations of the constraints and geometric objects in a sketch template. Superior to the previous work, the conic curve and B-spline curve are considered as primitive geometric objects in sketch optimization besides line and arc. We also extended the constraints into a wide variety both for geometric constraints (e.g. horizontal and vertical, etc.) and dimensional constraints.

In addition, the authors deeply studied the key factors such as initial estimate of the sketch, representation of the objective function, initial value of the penalty factors, which have a big impact on the efficiency and robustness of the algorithm and proposed some critical suggestions.

All the algorithm procedures have been implemented based on the CAD system Unigraphics and the software packages are transferred to real industrial applications. The results of the examples have demonstrated great success of the sketch template based parametric modeling technology applied in reverse engineering with a significant improvement on the efficiency and accuracy.

ACKNOWLEDGEMENTS

This work is supported by Global Research Center, General Electrical.

REFERENCE

[1] Yongqing Li, Xiaoping Huang, Chunhe Gong, Kelvin Wang, An engineering rules based parameterization approach for turbine blade reverse engineering, *Geometric Modeling and Processing 2004*, *IEEE* Computer Society 2004, pp 313-318.

[2] B. Bettig, J. Shah, Derivation of a standard set of geometric constraints for parametric modeling and data exchange, Computer-Aided Design, Vol. 33, No. 1, 2001, pp 17-33.

[3] Ioannis Fudos, Christoph M. Hoffmann, Constraint-based parametric conics for CAD, Computer-Aided Design, Vol. 28, No. 2, 1996, pp 91-100.

[4] William Bouma, Loannis Fudos and Christoph Hoffmann, Geometric constraint solver, *Computer Aided Design*, Vol.27, No.6, 1995, pp.487-501.

[5] Xiaoping Qian, Deba Dutta, Design of heterogeneous turbine blade, *Computer-Aided Design*, Vol. 35, No. 3, 2003, pp 319-329.

[6] Xiaoping Qian, Deba Dutta, Feature-based design for heterogeneous objects, *Computer-Aided Design*, Vol. 36, No. 12, 2004, pp 1263-1278.

[7] Pál Benk, Géza Kós, Tamás Várady, László Andor, Ralph Martin, Constrained fitting in reverse engineering, *Computer-Aided Geometric Design*, Vol. 19, No. 3, 2002, pp 173-205.

[8] N. Werghi, R. Fisher, C. Robertson and A. Ashbrook, Object reconstruction by incorporating geometric constraints in reverse engineer, *Computer-Aided Design*, Vol. 31, No. 6, 1999, pp 363-399.

[9] Javier Monedero, Parametric design: a review and some experiences, Automation in Construction, Vol.9, No. 4, 2000, pp 369-377.

[10] Vaughan Pratt, Direct least-squares fitting of algebraic surfaces, *Computer Graphics (SIGGRAPH 87)*, Vol. 21, No 4, 1987, pp 145–152.

[11] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher, Direct least squares fitting of ellipses, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, 1999, pp 476-480.

[12] Les Piegl, Wayne Tiller, The NURBS Book, second edition, Springer, Berlin, 1997.

[13] Gabor Lukács, Ralph Martin, Dave Marshall, Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation, *Proceedings of the 5th European Conference on Computer Vision* (ECCV'98), Vol. 1, 1998, pp 671–686.

[14] Broyden CG, Attia NF, Penalty functions, Newton's method and quadratic programming, *Journal of Optimization Theory and Applications*, Vol. 58, No.3, 1988, pp 377–385.