

# A Template-matching Approach to Free-form Feature Recognition

C L Li

*Department of Manufacturing Engineering  
and Engineering Management,  
The City University of Hong Kong,  
Tat Chee Avenue, Kowloon, Hong Kong.  
Email: meccli@cityu.edu.hk*

K C Hui

*Department of Mechanical  
and Automation Engineering,  
Chinese University of Hong Kong,  
Shatin, N.T., Hong Kong.  
Email: kchui@mae.cuhk.edu.hk*

## Abstract

*Feature recognition has been an active research area in the past two decades. While many techniques have been developed, most of them are limited to the detection of features that are of regular shapes. Examples include holes, slots, pockets and steps that are commonly used in engineering parts. This paper reports our investigation on a method that is able to detect free-form features. A template matching approach is used where the template represents the shape of the free-from feature to be detected. The feature recognition method works on a two-dimensional parametric space which represents all possible configurations of the template. A match between the template and the object from which the features are to be detected corresponds to a match-curve in the parametric space. Feature recognition is achieved by first searching the rough location of the match-curve, followed by searching a precise location of a point on the match-curve. A program has been implemented to verify the feasibility of the approach, and examples are given to demonstrate the major steps in the method. The use of the method in computer graphics applications such as shape blending is demonstrated. The application of the technique in industrial applications such as injection mould design is also discussed.*

## 1. Introduction

The purpose of feature recognition is to extract high-level semantic information from a low-level representation of an object. There are three major approaches that have received much attentions: syntactic pattern recognition (e.g [1]-[3]); graph-based method (e.g. [4]-[8]; and volumetric approach (e.g. [9]-[11]). One of the basic assumptions in most existing feature recognition techniques is that a shape feature is bounded by a set of

edges, which are explicitly represented in the CAD model. The concavity or convexity of all the edges, determined from the dihedral angle between the adjacent faces, are used in the determination of the type of the feature. This assumption limits existing feature recognition techniques to feature that are of regular shapes. Examples of such features that are commonly found in engineering parts are various types of hole, slot, pocket and step etc.

The present study focuses on free-from feature recognition. Free-from feature is an important class that has not been well investigated. A free-from feature is a feature that is embedded in a single or a set of free-from surfaces. The boundary of the features consists of curve segments that may lie within a surface. A free-form feature may even smoothly blend with other part of the object and thus an explicit boundary does not exist. Figure 1 shows a telephone handset. While a human can easily identify the regions that correspond to the ear-piece, the mouth-piece and the handle, it should be noticed that the mouth-piece blends smoothly to the handle. It is not possible to identify the boundary of the mouth-piece feature using existing feature recognition technique.



Figure 1. A telephone handset consists of three features: the ear-piece, the mouth-piece and the handle.

## 2. Template matching

We employ the template matching approach to tackle the free-form feature recognition problem. A template is initially specified by the user that represents the shape of the feature to be detected from the CAD model of an object. The feature recognizer then tries to detect the occurrence of the feature in the object by transforming the template to all possible orientations and positions. The feature is recognized if the template is considered to match the local shape of the object. One possible matching criterion is to measure the extent of coincident surface boundaries between the transformed template and the object. This template-matching approach in feature recognition was originally suggested by Woodward [12] and was further developed by Parry-Barwick and Bowyer [13]. The main challenge in template matching is to develop an efficient method that can locate the match in the continuous space of translation, rotation and scaling of the template. Parry-Barwick and Bowyer [13] tackled this problem by representing all the possible configurations in a multidimensional space, and developed a method that can prune the space efficiently. However, the trade-off in their method is that the object and template must be represented by half-spaces using implicit equations so as to facilitate an efficient pruning. Therefore, this method is limited to CSG model and cannot be applied to B-rep model, which is a more popular representation scheme.

This paper reports our initial investigation on template matching in 2D curved shape. The underlying geometry of the object and template is defined by composite NURBS curve segments. An efficient algorithm is developed that can recognize free-form features from the boundary of the object. An experimental program has been implemented to verify the feasibility of the method.

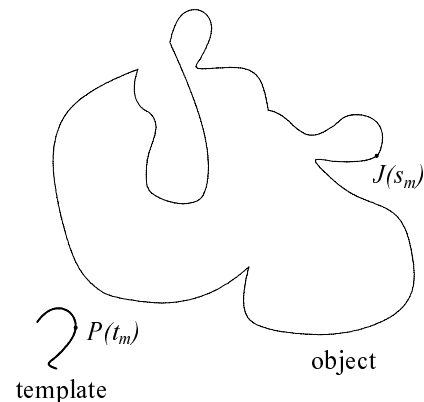
### 2.1 The basic approach

Consider an object and a template shown in Figure 2(a). Let  $J(s)$  and  $P(t)$  denote respectively the boundary of the object and template. A transformation is applied on the template as shown in Figure 2(b). The transformation consists of a translational component such that the point  $J(s_m)$  on the object coincides with the point  $P(t_m)$  of the template, and a rotational component such that the tangent at  $J(s_m)$  becomes parallel to the tangent at  $P(t_m)$ . We say that the template is at configuration  $Cfg(s_m, t_m)$ .

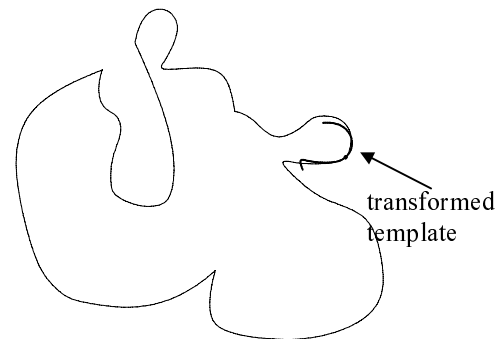
At a certain configuration  $Cfg(s, t)$ , portions of the template and object boundaries may overlap. If the overlapped portion corresponds to the entire template boundary, the template fully matches the object at that configuration. If the overlapped portion only corresponds to part of the template boundary, the template partly matches the object. The extent of match is measured by a **template-matching ratio**, which is defined as the ratio of

the arc length of the overlapped portion to the arc length of the template. If the template-matching ratio exceeds certain threshold value specified by the user, a match is detected and the feature is recognized at  $Cfg(s, t)$ .

Now, consider a space spanned by the parametric axes of the object and template. Let's denote the axes of this two-dimensional parametric space by the  $s$ -axis and  $t$ -axis, which correspond respectively to the parametric variables of the object and template. When a match is detected at a configuration  $Cfg(s, t)$ , a point is marked at  $(s, t)$  in the parametric space. A match occurs when portion of the transformed template overlaps with the object. Thus, if a match occurs at configuration  $Cfg(s, t)$ , matches also occur at configurations  $Cfg(s_i, t_i)$ , where the points  $J(s_i)$  are any neighbouring points of the object that are in the overlapped portion, and  $P(t_i)$  are corresponding points of the template that are in the overlapped portion. Therefore, a set of points can also be marked at  $(s_i, t_i)$  in the parametric space. It can be proved



(a) Template and object



(b) The template is translated and rotated to the configuration  $Cfg(s_m, t_m)$ .

Figure 2 Transforming template to configuration  $Cfg(s_m, t_m)$ .

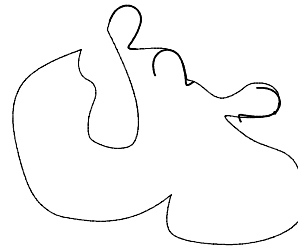
(see Li and Hui [23] for details) that this set of points form a smooth curve known as the **match-curve** on the parametric space. It has also been proved that each unique match corresponds to a unique match-curve on the parametric space. Figure 3 shows three matches with the corresponding match-curves. Thus, feature recognition can be achieved by first searching for a match-curve, and then locating any one precise point on the match-curve. The precise point on the match-curve gives the configuration at which the feature is recognized.

A two-phase approach is proposed for the process of template matching: a hunting phase followed by a refinement phase. In the hunting phase, potential regions in the parametric space where the match-curves lie are identified. These potential regions are rough indication of where the match-curves reside. In the refinement phase, each potential region is systematically searched to locate a point in the match-curve to within a specified tolerance. The following sections explain the details of each phase.

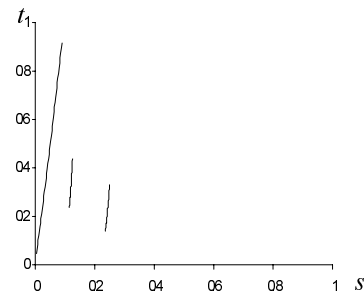
### 3. Searching for match-curve

To identify regions in the parametric space where the match-curves lie, the template-matching ratio at each point in the parametric space is first determined. A set of locations where the template-matching ratio exceeds the threshold value can then be found. From the set, clusters of locations can be formed by grouping locations that are adjacent to each other. From the resulting clusters, rectangular regions in the parametric space can be formed from the intervals in the  $s$  and  $t$  axes in which all the points within a cluster lie.

However, the computation of the template-matching ratio for all the points in the continuous parametric space is impossible because there are infinite number of points. Thus, discretization of the parametric space must be applied. This discretization also corresponds to a linear approximation of the curve boundaries that defined the object and template. With the discretization and linear approximation, the template-matching ratio can only be estimated. A method for estimating the template-matching ratio has been proposed (please refer to Li and Hui [23] for details) which will not under-estimate the actual value. This ensures that a region in which a match-curve lies will always be identified in the searching process. Our analysis also indicates that the extent of over-estimation depends on the resolution ( $m \times n$ ) of the discretization, and the algorithm is  $O(m^2 n^2)$ , where  $m$  and  $n$  are the number of intervals used in the discretization of the  $s$  and  $t$  axes. Thus, there is a trade-off in choosing the resolution. If a high resolution is used, the estimate is accurate but the computation time increases. If a low resolution is used, the computation is efficient but the effect of over-estimation is that a large area in the parametric space



(a) The matches.



(b) The match-curves in the parametric space.

Figure 3 The matches between the object and template and the corresponding match-curves in the parametric space.

would be considered as potential regions in which match-curves will be found.

An iterative scheme is thus proposed to find an 'optimal' resolution. Initially, a coarse resolution is used which results in relatively fewer points in the discretized parametric space. Template-matching ratios are determined for all the points. In addition to the estimation of template-matching ratios, the total arc length of the object that are considered as being matched by the template is determined. An **object-matching** ratio, which is defined as the ratio of the matched arc length to the perimeter of the object, is determined. When the resolution is coarse, template-matching ratio is over-estimated and thus object-matching ratio can be as high as 90% indicating that the entire object boundary is matched by the template. The resolution is then reduced in an iterative manner until the object-matching ratio falls below a threshold value specified by the user. Our experience indicates that this iterative process will terminate in few steps. Figure 4 illustrates three consecutive steps in the iteration process using the template and object shown in Figure 2. The figure shows the template at all configurations where the template-matching ratio exceeds a threshold value of 0.9. The object-matching ratio falls from 0.82 to 0.08 in three

iterations. It can be seen that potential matches are reported almost along the entire object boundary in the first iteration. Many of them do not actually correspond to a real match. This is due to over-estimation on the template-matching ratio when the resolution is low. In the next two steps, the majority of potential matches reported are located in the neighborhood of the real matches. Potential matches at other locations reported in the first iteration are removed when a higher resolution is used.

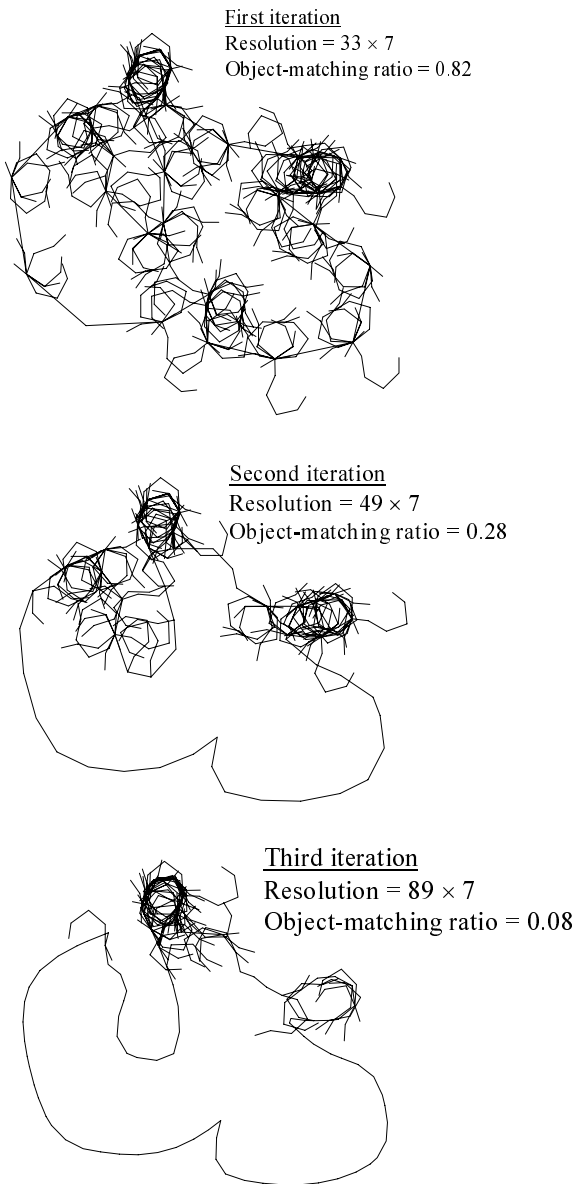


Figure 4 Iteration process to find 'optimal' resolution.

#### 4. Locating a match

Once a region within which the match-curve lies is identified, a precise curve point can be found to within a specified tolerance. This precise location gives the configuration at which the template matches the object. It should be understood that as the entire match-curve corresponds to a unique match, just one a single point in the match-curve needs to be found.

Each region corresponds to an interval in the object boundary and an interval in the template boundary. These intervals are usually small compared with the entire parametric range. Therefore, a high resolution can be afforded on these restricted intervals of the boundaries. The refined resolution thus enables an accurate estimation of the template-matching ratio.

An exact match in each match-curve is found by sampling a sufficient number of sample points in each region of the parametric space. Each region found in the previous phase is only a rough indication of the location of a match. Moreover, it may be possible that two distinct match-curves are merged into the same region during the process of cluster formation. Therefore, a sufficient number of sample points must be generated to ensure that matches corresponding to all distinct match-curves within each region can be detected. When a match occurs, the template boundary overlaps with the object boundary. An estimation  $l_e$  of the arc length of the overlapped boundary can be obtained by multiplying the perimeter of the template boundary by the threshold value of the template-matching ratio. When the template matches the object at one location, the next match location on the object boundary must be at least  $l_e$  units away from a previous location along the object boundary. That is, two distinct matches will not occur between two neighbouring points on the object boundary that is less than  $l_e$  units apart. Let  $\{s_i\}$  be the sequence of parametric values sampled on the object parametric axis  $s$ . Then,  $J(s_i)$  and  $J(s_{i+1})$  are two neighbouring points on the object boundary. Let the arc length between the two points be denoted by  $l(s_i, s_{i+1})$ . The first sample value  $s_1$  is set such that  $l(s_u, s_1) < l_e/2$ , where  $s_u$  is the lower limit of the parametric range of the region. The subsequent sample values are set such that  $l(s_i, s_{i+1}) < l_e$ . This sequence of sample values  $\{s_i\}$  thus guarantees the detection of matches for all distinct match-curves within a region.

For each sample value in the  $s$  axis, a search in the  $t$  axis is conducted. The search looks for the largest template-matching ratio that can be achieved within the parametric range in the  $t$  axis of the region. The search consists of the following steps:

1. From the parametric range of the  $t$  axis, samples a sequence of parametric values  $\{t_{ij}\}$ .
2. From the sequence of parametric values  $\{t_{ij}\}$ , compute the template-matching ratios and identify  $t_m$  such that its template-matching ratio is maximal.

3. Construct a new parametric range by setting the lower limit as  $t_{m-1}$  and the upper limit as  $t_{m+1}$ .
4. Repeat steps 1 to 3 until one of the following two termination conditions is satisfied:
  - (i) The distance between the two points  $P(t_{m-1})$  and  $P(t_{m+1})$  on the template boundary is less than the positional tolerance, and the angle between the two tangent vectors  $P'(t_{m-1})$  and  $P'(t_{m+1})$  is less than the angular tolerance.
  - (ii)  $t_{m-1}$  equals to  $t_{m+1}$ .

When condition (i) is reached, the template matching process finds a match location to within the positional and angular accuracy specified by the user. In case (ii), the template matching process terminates because the maximum accuracy that can be achieved on the computing platform has been reached.

## 5. Implementation and Potential Applications

An experimental program has been implemented to verify the feasibility of the proposed method. The program is written in C and runs on a Pentium PC. The program is interfaced with a simple shape blending system to study the advantage of using free-form feature recognition in shape blending.

Applications of shape blending or shape averaging in computer animation and industrial design have been studied by many researchers (e.g. [14]-[18]). In general, the first step in the shape blending process is to establish the correspondence between the source and target objects. Vertex correspondences have been used by many researchers (e.g. [14], [15], [19], [20]) and their methods applied only to polygonal shapes. The importance of feature correspondence has been noticed by other researchers (e.g. [21], [22]) and recently, Hui and Li [18,19] has developed a fully automatic algorithm to establish feature correspondence between two curved parts. However, the correspondence generated by his method depends on the relative orientations of the two parts. A change in the orientation may result in a different set of correspondences. This makes his method inappropriate in computer animation. By employing the proposed template matching approach to free-form feature recognition, features in the two parts can be recognized and the correspondence so established is independent of the relative orientation.

Figure 5 shows a shape blending example, which is adapted from one of the examples illustrated in the work of Sederberg and Greenwood [15]. The 'dots' shown on the boundary of the source object in Figure 5(a) are 'vertices' on the boundary. They are detected by employing heuristics similar to those developed by Hui

and Li [18]. Any two neighbouring vertices define a potential feature, which is used as a template. The template matching process then detects matches from the target object for all the templates. Figure 5(b) shows the five matches found in the target. These matches enable the establishment of correspondence between the source and target at the feature level. The source and target boundaries can thus be partitioned into sequences of *feature segments* and *connector segments* as shown in Figure 5(c). Figure 5(d) shows the blending result obtained by linearly interpolating corresponding feature segments and connector segments. The result is comparable to the work of Sederberg and Greenwood [15]. Notice that the left arm is shortened in the in-betweens. This problem also occurs in the work of Sederberg and Greenwood [15], which is due to the linear interpolation between the two arm positions that have excessive difference in orientation. With the establishment of feature correspondence by template matching, a better

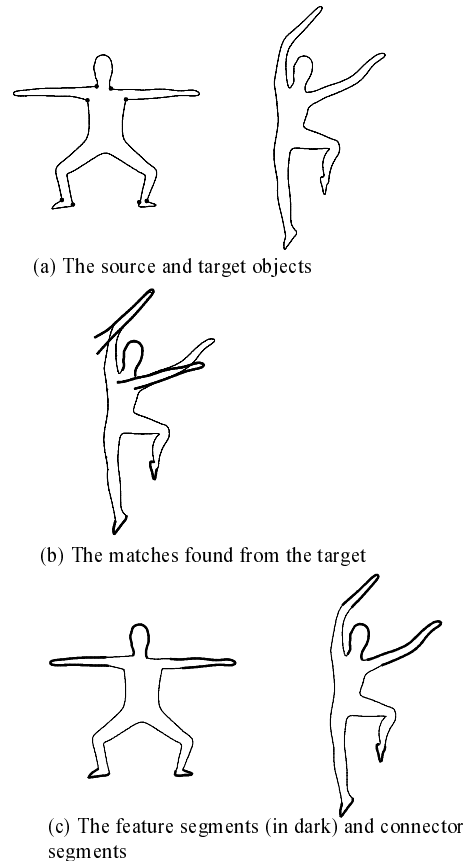


Figure 5 An example of shape blending using the template matching method.

interpolation scheme can be employed to overcome the problem. A rigid transformation, which involves rotation

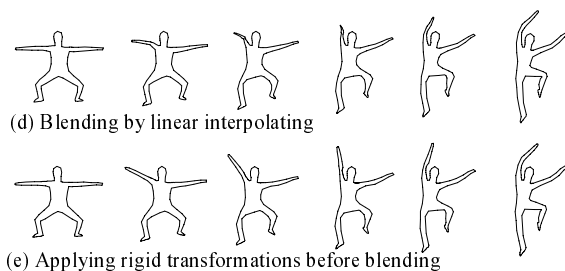


Figure 5 An example of shape blending using the template matching method(continued).

and translation, is first applied to align a corresponding feature pair before linear interpolation is applied. The idea of applying rigid transformation is also used in the method of Carmel and Cohen-Or [22]. However, their method is limited to the use of a single rigid transformation for each of the entire source and target objects. With feature correspondence established by template matching, a different rigid transformation can be devised to ensure alignment of each individual feature pairs. A better blending result can thus be generated as shown in Figure 5(e).

Besides the shape averaging problem, another important application that is being investigated by the authors is the design of plastic injection mould. The cooling system of a plastic injection mould plays a crucial role in the optimization of the injection moulding process. A survey on a set of existing mould designs has been conducted which concludes that the cooling system depends very much on the moulded part shape. A set of typical part shapes is thus identified and their corresponding cooling system designs has been captured as a set of parametric design rules. Cooling system design for a new plastic part is then obtained by retrieving and instantiating the existing parametric design of a stored part with similar part shape. The major challenge in automating this retrieval process is the ability to detect parts with similar shapes. The proposed template matching algorithm, being able to detect free-form features, can be used to solve this problem. Further investigation on applying the template matching algorithm to detect shape similarity in 3D is being conducted by the authors.

## 6. Conclusion

Free-form feature recognition is an important research area that has not been well investigated. We have developed a method that achieves free-form feature

recognition in two-dimensional curved object. The efficiency of the method relies on the two-phase approach. In the first phase, a low resolution is used to discretized the parametric space where potential regions are identified. The resolution is iteratively refine until an 'optimal' resolution is obtained. In the second phase, high resolution is used to locate the exact match to a specified accuracy. Application of the free-form feature recognition method in shape averaging has been demonstrated and the application in mould design is discussed.

## Acknowledgements

The authors would like to express their gratitude to the Research Grants Council of Hong Kong for the financial support to the project.

## References

- [1] Jakubowski R, "Syntactic characterization of machine parts shapes", *Cybernetics and Systems*, 3(1), 1982, pp 1-24.
- [2] Staley S M, Henderson M R and Anderson D C, "Using syntactic pattern recognition to extract feature information from a solid geometric data base", *Comp. in Mech. Eng.*, 2(20), 1983, pp 61-66.
- [3] Choi B K, Barash M M and Anderson D C, "Automatic recognition of machined surfaces from a 3D solid model", *Computer-Aided Design*, Vol 16 No 2, 1984, pp 81-86.
- [4] Joshi S and Chang T C, "Graph-based heuristics for recognition of machined features from 3-D solid model", *Computer-Aided Design*, Vol 20 No 2, 1988, pp 58-66.
- [5] De Floriani L and Bruzzone E, "Building a feature-based object description from a boundary model", *Computer-Aided Design*, Vol 21 No 10, 1989, pp 602-610.
- [6] Gavankar P and Henderson M R, "Graph-based extraction of protrusions and depressions from boundary representations", *Computer-Aided Design*, Vol 22 No 7, 1990, pp 442-450.
- [7] Corney J and Clark D E R, "Method for finding holes and pockets that connect multiple faces in 2 1/2D objects", *Computer-Aided Design*, Vol 23 No 10, 1991, pp 658-668.

- [8] Lentz D H and Sowerby R, "Feature extraction of concave and convex regions", *Computer-Aided Design*, Vol 25 No 7, 1993, pp 421-437.
- [9] Woo T C, "Feature extraction by volume decomposition", *Proceedings of the conference on CAD/CAM in Mechanical Engineering*, 1982, MIT Press, Cambridge, MA.
- [10] Vandenbrande J and Requicha A, "Geometric recognition of interacting machining features", *Advanced in Feature Based Manufacturing*, Elsevier, Amsterdam, 1994, edited by Shah J J, Mantyla M and Nau D S.
- [11] Sakurai H, "Volume decomposition and feature recognition: Part I- polyhedral objects", *Computer Aided Design*, Vol 27 No 11, 1995, pp 833-843
- [12] Woodward J R, "Some speculations on feature recognition", *Computer-Aided Design*, Vol 20 No 4, 1988, pp 189-196.
- [13] Parry-Barwick S and Bowyer A, "Multidimensional set-theoretic feature recognition", *Computer-Aided Design*, Vol 27 No 10, 1995, pp 731-740.
- [14] Chen S E and Parent R, "Shape Averaging and Its Applications to Industrial Design", *IEEE Computer Graphics & Applications*, Vol 9 No 1, 1989, pp 47-54.
- [15] Sederberg T W and Greenwood E, "A Physically Based Approach to 2-D Shape Blending", *ACM Computer Graphics*, Vol 26 No 2, 1992, pp 25-34.
- [16] Wang H, "An approach to Computer-aided styling", *Design studies*, Vol 16 No 1, 1995, pp 50-61.
- [17] Hui K C and Li Y D, "Element method for 2-D shape transformation", *Proceedings of the International Conference on Manufacturing Automation ICMA'97*, Apr 28-30, 1997, the University of Hong Kong, Hong Kong.
- [18] Hui KC and Li Y D, "A feature-based shape blending technique for industrial design", *Computer-Aided Design*, Vol 30 No 10, 1998, pp 823-834.
- [19] Burr, D J, "Elastic matching of line drawings", *Proc. 5<sup>th</sup> International Conference on Pattern Recognition*, 1980, pp 223-228.
- [20] Zhang Y, "A Fuzzy Approach to Digital Image Warping", *IEEE Comp. Graph. App.*, July 1996, pp 34-41.
- [21] Beier T, "Feature-Based Image Metamorphosis", *Computer Graphics' 92*, 1992, pp 35-42
- [22] Carmel E and Cohen-Or D, "Warp-guided object-space morphing", *the Visual Computer*, Vol 13, 1997, pp 465-478
- [23] Li C L and Hui K C, "Feature recognition by template matching", to appear in *Computers & Graphics*, Vol 24 No 4, 2000.