International Journal of CAD/CAM Vol. 7, No. 1, pp. 113~120 (2007)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

2D Sketch Based Query Interface with Coarse Filter for 3D Shape Matching

Jaeho Lee and JoonYoung Park*1

Department of Industrial and Systems Engineering, Dongguk University, Seoul, Korea

Abstract – Sketch based query plays an important role in shape matching for 3D shape retrieval system. Some researchers suggest the sketch based query interfaces. It is more effective to capture the needs of users rather than query by example. In this paper, we propose a new 2D sketch based query interface with coarse filters for shape matching. Coarse filtering enables to eliminate unfavorable shapes from shapes in DB. For the purpose of coarse filtering, we use two filters with the topological and geometric patterns. For the validity of our method, we show the experimental results.

Keywords **: Shape Matching, Query Interface, Query by Sketch, Coarse Filtering**

1. Introduction

3D shape searching is widely developed and researched due to the expansion of 3D model acquisition and handling. 3D shape matching algorithm plays an important role in the shape based search system. Especially, the shape search using the shape similarity can be used to improve the process of product design and manufacturing. There are two examples. The first is the cost estimation for machined parts. In these days, many job shops allow designers to submit a 3D model of the part to be machined over the internet and provide a cost estimate based on the 3D part model. Currently, humans perform cost estimation. However, if a sufficiently similar part can be found in the database of previously manufactured parts, then the cost of the new part can be more easily estimated by suitably modifying the actual cost of the previous part. The second is the reduction of the part proliferation by reusing previously designed part. Reusing design and manufacturing information stored in database would result in a fast and more efficient design process [4].

With vast 3D models being added to databases, the need for organizing and indexing databases of 3D models is more important. This will provide a systematic and efficient way of retrieving similar models from the database [7]. For the effective shape retrieving, the effective shape matching methods are required. The shape is defined by many properties and metrics. Thus, the general framework of shape matching is developed. However, many matching algorithms are focused in what and how are they different in the given two models [16].

One of the direct methods of shape matching between 3D models can be done by using the Boolean operations between two models. However, the direct assessment of similarity between 3D models with Boolean operations is computationally very slow due to difficulty in aligning the models before performing the Boolean operations.

Thus, it is often not practical when the number of 3D models is very large. Therefore, a computationally efficient way to solve this problem is to abstract 3D shapes into shape signatures and use them to compare the actual shapes, and they have lesser shape discrimination capability compared to the complete object model. Over the last few decades, many papers have been written which describe algorithms for the assessment of the shape similarity. All these shape matching methods have been developed for shape search system like the sketch based search.

2. Related Works

In general, users feel free to query by string like Google. However, users do not feel free to query by string in the shape searching because of the lack of information in the string code. For example, the string code of a 3D modeling file could be 'A120.dwg' which does not mean anything about its shape. Famous string based approach is used in Group Technology (GT). It takes the methodology of capturing semantic structure like bolt type A with diameter 2. In order to implement GT, one must have a concise coding scheme for describing products and the method for grouping (or classifying) similar products, such as the popular Opitz, and DCLASS schemes. However, as the classification is done manually, it is subject to the individual interpretation. It means that the human perception of similarity is subjective [2]. Therefore, there are possibilities for errors in such classifications.

Table 1 shows the three types of shape search. As shown in table 1, the sketch interface of the shape search is naturally query by example or query by modeling [12]. For the use of the query by sketch methodology for shape matching, some

*Corresponding author:
Tel: +82-(0)2-2260-3714
Fax: +82-(0)2-2292-047
Homepage: http://ise.dongguk.edu/
E-mail: jypark@dgu.edu

Table. 1. The three query types of shape search system.

Query type	Advantage	Disadvantage
Query by string		
Bolt_2pi_HR JPxx042-se String	Keywords are very Familiar to user	Keywords do not reflect the shape well
Query by model		
 Model	User own 3D model is used	Model is fixed users can't change freely
Query by sketch		
 Sketch	User can sketch freely in their mind	The effective method is Unknown

methods are suggested [14]. Query by sketch has been developed by some research groups for developing the shape search system by using the user's sketch based input query.

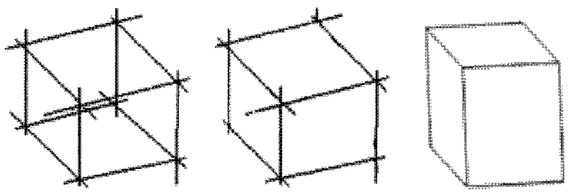
Sketch based interface is classified into the two types. First type of sketch is the 3D entities based structure with modeling sequence for the shape matching like Princeton shape search system [6]. Second type of sketch is the 2D pixels or 2D geometric entities based structure for the shape matching [8].

Sketch is generally made of freehand lines using ink and pen in art [3]. In computer graphics and geometric modeling, the meaning of sketch is the collection of geometric components like line segments, polygons, and curves. Thus, effective sketch structure for geometric modeling is developed by many researchers [1]. To take advantage of sketch based interaction, many methods have been developed to parse and recognize the sketches. Unfortunately, these methods are still not capable of handling the sketch parsing problem robustly [5].

For freehand sketch based parsing purpose, the work outlined in [18] tried to explore the interactive properties of sketching such as the stroke direction and speed. However, the activities of sketching have to be recorded in real time. Thus, the perturbation of a user's hand will lead to clear changes in speed and curvature [9]. Therefore this results in incorrect segmentation for compact sketch pattern saving [13]. To overcome such limitations, many other approaches were proposed such as the template based method [5] and Bayesian based statistical model [17].

Fig 1 (a) shows 3D sketch based approach and fig 1(b) show the 2D sketch based approach and fig 1(c) show the 3D model.

The goal of this paper is to provide a polygonal sketch based intuitive shape matching algorithm rather than freehand sketch based method. In section 2, we describe the polygonal based sketch with loop counts. In section 3, we describe the shape matching of polygonal sketch with loop counts and



(a) (b) (c)

Fig. 1. Sketch models. (a) 3D sketch, (b) 2D Sketch, and (c) 3D Model

show the experimental results. Finally, we conclude in section 4.

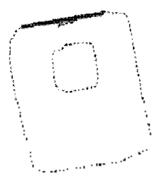
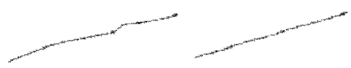
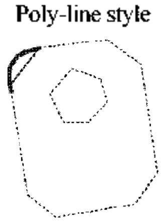
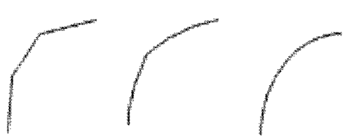
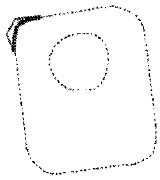
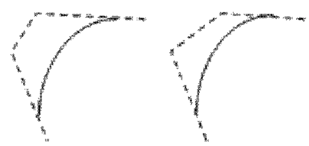
3. Sketch Interfaces

Our design strategy is based on some simple observations. First, most users are familiar with the query by string method like Google search engine. However, it is not sufficient to prepare the similarity between the models by using the string code of the shape model, because the shape code is very difficult to define. For example, we do not know the good abstraction of the polygon soup model. Second, most users are not familiar with the sketch interface, because they sketch the model what they imagine in their mind.

Table 2 shows three styles of the line metaphor in sketch query systems. Their freehand, poly-line and curve based method have own properties respectively. However, we prefer the area metaphor rather than the line metaphor because of its easiness in the sketch configuration like drag and drop method and click and click method. Fig. 2 shows our area metaphor.

Our system is designed by adapting the polygon based

Table. 2. Line metaphor in sketch method.

Query type	Properties
Freehand style	
	Rectification and beautification is needed 
Poly-line style	
	Many line segments are required for more exact sketching 
Curve based style	
	Control of curve segment is difficult 

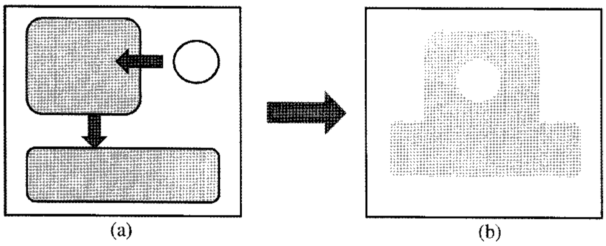


Fig. 2. Area metaphor. (a) Entity configuration with drag and drop, (b) Resulting sketch.

sketch interface, the polygonal structure which enables to use the rectangle type, partly ellipsoid type geometry and circle rather than freehand sketch as shown.

The proposed system has two processing units. First unit is the offline processing unit which enables to convert the inputted model into the 2D contour polygon which means 2D slice of original model by slicing. This is shown as Fig. 3 (b). In this process, we take the slicing operation between 3D model and plane. Then, we can easily capture a loop count from slice contour. This information is very useful to make our topological filter for shape matching. We called this filter as a coarse filter for using the topological filtering. We discuss this process in section 4.2. Second unit is the online processing unit which enables to translate the user input sketches with mouse clicks and/or drag and drops into the polygonal sketch data structure. After translating the input sketches, the similarity calculation between the polygonal sketch and 2D slice of original model is done. And the best match and the near best match results are displayed on the screen. Fig. 3 shows the proposed system with polygonal sketch and freehand based system taken in previous researches.

4. Shape Signature

In our sketch interface system, we use the shape descriptor using the spherical sectioning [10]. Our shape descriptor is called SSRD (spherical sectioning railroad diagram) which is composed of the vertex counts in spherical sectors, 2D slice planes for the capturing topological properties and 2D planar contours of original model.

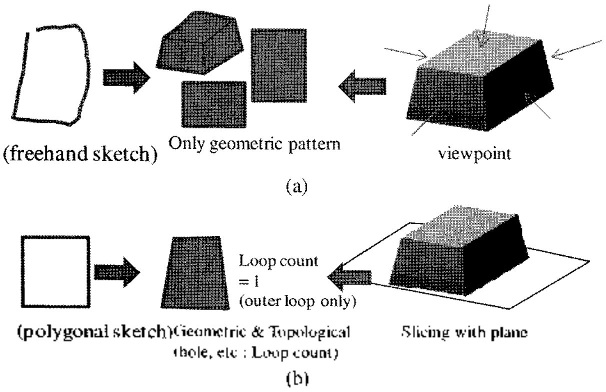


Fig. 3. The comparison between the proposed system and the freehand system. (a) freehand sketch system, (b) proposed system.

SSRD model is designed by means of data structure for the effective shape matching with query by example. In the query by example interface, the user only picks the example model as a query model. Then the picked model is converted into the shape descriptor and the system finds the best matched model in DB using the similarity measure.

4.1 Revised SSRD Model

In this paper, we design the sketch based in-put interface, thus we developed a revised SSRD model which is more suitable for the sketch interface. SSRD model is composed of two parts. First is the 2D slice pattern. Second is the vertex count in spherical sectors. The revised SSRD model has only the first part, which is the 2D slice pattern. In our system, the user sketch is regarded as the 2D slice pattern of a model. Despite of absence of vertex counts in spherical sectors, the 2D slice pattern gives us strong information that is the topological relationship of the model.

The sectioning property of the revised SSRD model is very useful for the 2D sketch based interface because the similarity calculation between 2D input sketch query and the section data from the revised SSRD is very easy. Fig. 4 shows the revised SSRD model building process.

The revised SSRD data structure is as follows:

```
Class Revised_SSRD {
    Loop count: integer
    // this hint us the topological structure
    Outer loop count: integer
    Inner loop count: integer
    // it means that the model has a hole or cavity etc.
    Intersection points: array [ (xyz) , ... , (xyz) ]
    // where x, y, z is float type respectively.
    // we can easily acquire the polygon from this intersection.
}
```

4.2 Polygonal Sketch User Input

We propose the polygonal sketch data structure for sketch composition using two metaphors, click & click operation and drag & drop operation. The data structure of the polygonal sketch is composed of two parts. First part is the composition of 2D geometric entities and second part is the topological relationship. Our system takes the proposition that the user has a mouse interface. Thus our sketch system does not use the freehand sketch system. In the mouse system, users feel uncomfortable to use the click based line and/or curve entity control in the line drawings. Fig. 5 shows this situation.

Users prefer the drag and drop the entities rather than click and click entity drawings. Thus, in our data structure for

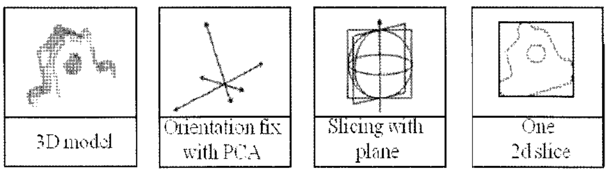


Fig. 4. Revised SSRD model generation.

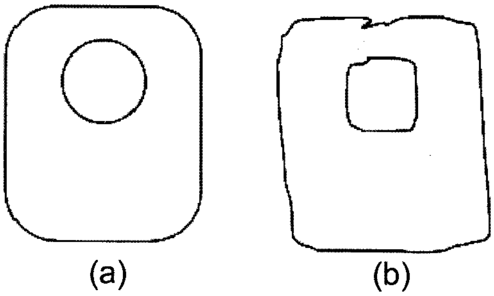


Fig. 5. Two sketch. (a) the geometric modeling, (b) the frechand sketch

sketch composition, we use the two metaphors, click & click operation and drag & drop operation as shown in Fig. 6. The data structure of polygonal sketch is composed of two parts. First part is the composition of the 2D geometric entities and second part is the topological relationship.
Polygonal sketch data structure is as follows:

```
Class sketch_ polygon {
      Loop count: integer // this is given by user in sketch pad
      Outer loop count: integer
      Inner loop count: integer // it means that the model has
                         a hole or cavity etc.
      Edge list : { (Vertex1, vertex2), ... , (Vertex n, Vertex n+1) }
      Vertex list : { (x1, y1), ... , (xn, yn) }
}
```

Polygonal sketch with rectangle and hole is as shown in Fig. 7. The proposed polygonal sketch interface contains five components for constructing the sketch. Table 3 shows the components.

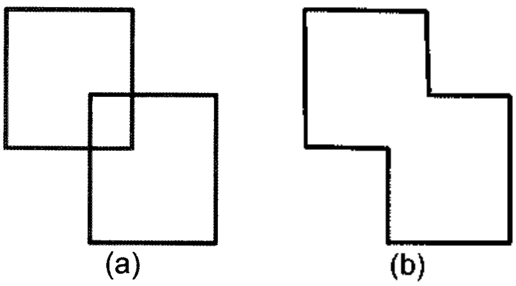


Fig. 6. The sketch configuration with the Boolean operation. (a) two primitives, (b) the result of Boolean operation.

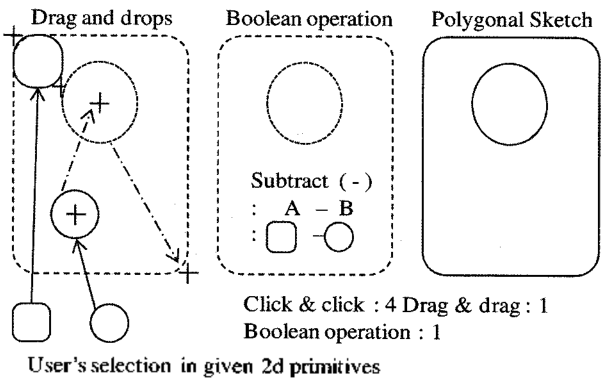


Fig. 7. The example of the polygonal sketch.

Table. 3. The five components and the data structure of 2D basic entities.

Query type	Entity types	Data structure
	Rectangle	Poly-line
	Rounded Rectangle	Poly-line
	Circle	Poly-line
	Quadrant of circle	Poly-line
	Ellipse	Poly-line

4.3 Line Segments in Sketch

The polygonal sketch created by user has been converted into the line segments which enables to calculate the similarity between sketch and revised SSRD described in section 4.1~4.2. The proposed polygonal sketch consists of 2D basic entities and their Boolean operator. The resulting sketch is converted into the closed poly-line structure. Then, we can easily handle the line segments of the sketch. This is very useful for the shape matching. It can easily calculate the D2 shape distance and shape histogram.

4.4 Closed Loop of Polygonal Sketch

After the Boolean operation of between the user-selected 2D entities, the closed loop of polygonal sketch is counted. This count means the measure of the topological structure in the sketched part and also the partial topological structure of the 3D model in DB. This value is used for the similarity calculation described in section 5.1.

5. Shape Match with Coarse Filter

5.1 Shape Distribution

Some methods for computing the shape distribution of solid models using shape functions and assessing the similarity by using this shape distribution is known in [11]. Once a set of random points is obtained on the surface of the solid model, the different shape functions are used to compute the shape

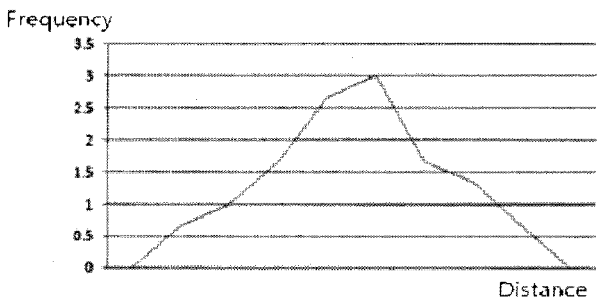


Fig. 8. The D2 shape function.

distributions for the solid model. The typical shape functions are as follows :

D1 : Computes the distance between a fixed point and a random point. This function is not suitable as the chosen fixed point is usually not invariant to the rotation or translation.

D2 : Computes the distance between two random points. This function is invariant to the rotation and translation and is robust.

D3 : Computes the square root of the area of triangle generated by three random points. This function is also invariant to the translation and the rotation but not as efficient as D2.

A1 : Computes the angle between three random points. This function is invariant to translation, rotation and scaling but it is not very robust.

In general, the D2 shape function is suitable for computing shape distribution due to its robustness and efficiency along with invariance to rotation and translation. After calculating the distance between two random points, they are normalized using the mean distance. The shape distribution is the histogram which is composed of the frequency of occurrence of distance within a specified range of distance values. Fig. 8 shows the D2 shape histogram.

5.2 Loop Counts as Topological Filter

Shape descriptor is robust to geometric transformation especially rotation. In this paper, we use the D2 shape function which is rotational invariant. However, the D2 shape function has some limitations. As the number of the points increases the comparison is more robust, but the computational time increases. Furthermore as objects become more and more complex, the shape distributions tend to assume similar shapes.

Fig. 9 shows this situation. In fig. 9, two parts are very different in view of human visual system. However, the results of in view of D2 shape histogram of each model are very similar. It arises to increase the semantic gap between the result of the algorithm and human visual system.

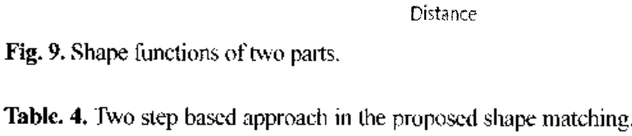


Fig. 9. Shape functions of two parts.

This situation is caused by lack of the topological information of the D2 shape function which can handle only the distance between the geometric entities.

Our shape matching metric is the distance based metric. Most 3D shape representation schemes convert a shape into a feature vector or a relational data structure (e.g. graphs or trees). Feature vectors are represented as points in the feature space in a database. The similarity between two feature vectors reflects the distance between corresponding points in the feature space. For the purpose of the similarity measure calculation between two feature vectors, the Minkowski distance metric is generally used. We use the L_2 distance which is the Minkowski distance with $p = 2$. Simply, it is called the Euclidean distance between two points [15]. The L_2 distance metric is defined as

$$L_2(x,y) = \left[\sum_{i=0}^{N} |x_i - y_i|^2 \right]^{1/2} \tag{1}$$

We use the L_2 distance. Sketch by query model is converted into the polygonal sketch structure as a shape descriptor in described section 3.3. Then the problem of similarity measure between the query model and revised SSRD model in DB is transformed into the 2D version of similarity calculation between the two shapes. We already know the topological information of two models. This information helps us to acquire the strong differentiation power.

Our similarity step has two phases. At first, we take the filtering using the topological difference table. Table 4 shows the topological information based filtering process. Secondly, we calculate the D2 shape function between the polygonal sketch and the revised SSRD. This problem is the calculation of 2D shape distance function called '2D shape histogram' on the basis of [11].

To measure the similarity between 2D shapes, we summarize it as the following steps:

Step 1. Discretize all entities contained in a drawing into a set of line segments.

Step 2. Sample enough points on the line segments uniformly.

Step 3. Select two points randomly from the point set and compute their Euclidean distance.

Step 4. Calculate the distance distribution of all sampled

Table. 4. Two step based approach in the proposed shape matching.

Proposed shape matching process	Process A : Coarse filtering with using loop counts	Process B : D2 shape matching using the geometric information with (Eq.2)
Input sketch (polygonal sketch and revised SSRD model)	Boolean function test If the result of the Process A is exact match, go to process B	If the result of (Eq. 2) is near 1, two shapes are very similar. If the result of (Eq. 2) is near 0, two shapes are very different. Otherwise, Two shapes are not similar.

point pairs, and build a histogram.

For two histograms, i.e., H_1 and H_2, their similarity S is given by:

$$S(H_1,H_2)=L_2(H_1,H_2)=\sqrt{\sum_{i=0}^{h}(H_1(i)-H_2(i))^{1/n}} \qquad (2)$$

Our similarity measure calculation process is as shown in table 4.

5.3 Shape Similarity with Coarse Filter

Our similarity measure calculation process is two step based approach as shown in Table 5. And, Table 5 shows the result of this two step shape matching in simple parts. User sketches the rectangle with four holes as a query model. We use the model Type A, and Type B which is different in view of the human visual system. Our system allocates 'fail' value in model type A (a) and filter out it in resulting set. In view of a human visual system, it is very natural. However, most sketch based systems does not allocate the any value in sense of the topological filter. Thus, they do not filter out them. Our system allocates also 'exact' value in model type A (b), type B (a) and type B (b). Then, they use D2 Shape histogram using the (Eq. 2). Then, the best match is model type A (b), and next is model type B (b). This result of our system is very similar to human visual system.

We show the experimental results as shown in Table 6. We conclude the polygonal sketch based interface will help us to lessen the semantic gap between existing shape search system and human visual system.

6. Conclusions

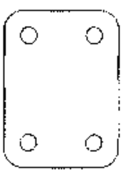
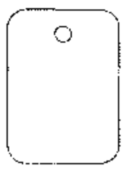
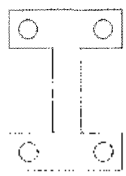
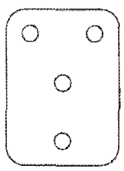
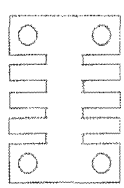
In this paper, we proposed a new sketch based query interface for 3D shape matching using the grid pattern like poly-line, ellipsoid and rectangle. Our method is especially well suited for the shape descriptor, revised SSRD, which is composed of the 2D slice pattern. Users using our interface feel comfortable because it is easy to sketch. Advantage of our method is the easily acceptable metaphor among users. Drag and drop of 2D entities can be learned easily. Users drag and drop the entities and use the Boolean operation between the entities. Our Boolean operations use the pixel-based calculation. Thus the implementation is fast and robust. After some Boolean operations with user controls are added, the sketch is completed.

Since the proposed method adapts the topological filtering, it easily enables to filter out dissimilar parts in DB.

Table. 6. The proposed method with the coarse filter.

Query model	Match result	Original model
 Polygonal sketch	Process A : ok Process B : 42.5%	
	Process A : ok Process B : 64.3%	
	Process A : no Process B : 34.6%	
	Process A : no Process B : 23.5%	
	Process A : ok Process B : 14.5%	
	Process A : ok Process B : 51.2%	
	Process A : no Process B : 25.5%	

Table. 5. The coarse filtering process in the simple mechanical parts.

Query model	A type model	B type model
 Rounded rectangle with 4 holes	 (a) rounded rectangle with 2 holes	 (a) 'I' shape with 2 holes
	 (b) rounded rectangle with 4 holes	 (b) Modified 'I' shape with 4 holes
Results of the proposed similarity calculation	A(a) : Filter out (fail) in Process A A(b) : Pass in Process A and 94% (0.942) in Process B	B(a) : Pass in Process A and 47% (0.467) in Process B B(b) : Pass in Process A and 82% (0.816) in Process B

In sense of human visual system, it easily can do. However, most shape descriptors and shape matching process handle models as one converted value in view of geometric structure. Thus, some models of shape matched parts in resulting screen are non-sense parts in view of human visual system. Although our coarse filtering only use the number of closed loop, many alternatives are possible like shape and size of convex hull of model, bounding sphere or hierarchical bounding boxes etc.

The proposed method has also some limitations. First, some users feel somewhat uncomfortable when they try to make a very accurate sketch. Although most 2D simple shapes can be made easily, it does not handle the sketch easily for more complex shape. Second, our system does not support the complex curves like Bezier, and B-spline. We plan to enhance these limitations for more intuitive and more similar to the freehand composition.

Acknowledgements

This work was supported by the Basic Research Program of the Korea Science & Engineering Foundation (No. R01-2006-000-10327-0), and the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (No. KRF-2005-041-D00903).

References

[1] Alvarado, C. and David, R. (2004), SketchPad: a multi-domain sketch recognition engine, *In Proc. of the 17th Annual ACM Symposium on User Interface Software and Technology* 23-32.

[2] Burbidge, J. L. (1975), *The Introduction of Group Technology*, Heinemann, London.

[3] Calhoun, C., Stahovich, T. F., Kurtoglu, T. and Kara, L. B. (2002), Recognizing Multi-Stroke Symbols. *In Proc. of the 2002 AAAI Spring Symposium Sketch Understanding* 15-23.

[4] Cardone, A., Gupta, S. K. and Karnik, M. (2003), A survey of shape similarity assessment algo-rithms for product design and manufacturing applications. *ASME Journal of Computing and Information Science in Engineering* **3**, 109-118.

[5] Casagliola, G., Deufemia, V., Polese, G. and Risi, M. (2004), A Parsing Technique for Sketch Recognition Systems. *In Proc. of 2004 IEEE Sympo-sium on Visual Language-Human Centric Computing (VLHCC'04)* 19-26.

[6] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D. and Jacobs, D. (2003), A Search Engine for 3D Models, *ACM Transaction on Graphics* **22**, 83-105.

[7] Hou, S. and Ramani, K. (2006), Sketch-based 3D Engineering Part Class Browsing and Retrieval, *in Proc. of EuroGraphics Symposium Proceedings on Sketch-Based Interfaces & Modeling*, 131-138.

[8] Hse, H., Shilman, M. and Newton, A. R. (2004), Robust Sketched Symbol Fragmentation Using Template. *in Proc. of the 9th International Confe-rence on Intelligent User Interface* 156–160.

[9] Kara, L. B. and Stahovich, T. F. (2004), An Image-based Trainable Symbol Recognizer for Hand-drawn Sketches, *Computers & Graphics* **29**, 501-517.

[10] Lee, J. H., Park, J. Y. and Lee, H. C. (2006), An Algorithm for 3D Shape Matching using Spherical Sectioning, *Journal of Zhejiang University Science A* **1**, 1508-1515.

[11] Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D. (2002), Shape Distributions, *ACM Transactions on Graphics* **21**, 807-832.

[12] Pu, J. T. and Ramani, K. (2005), A 2D Sketch User Interface for 3D CAD Model Retrieval, *Journal of Computer Aided Design and Application* **2**(6), 717-727.

[13] Pu, J. T. and Ramani, K. (2005), A 3D Model Retrieval Method using 2D Freehand Sketches, *Lecture Notes in Computer Science* **3515**, 343-347.

[14] Pu, J. T. and Ramani, K. (2006), On Visual Similarity based 2D Drawing Retrieval, *Computer Aided Design* **38**(3), 249-259.

[15] Schneider, P. J. and Eberly, D. H. (2003), *Geometric Tools for Computer Graphics*, The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling.

[16] Schyns, P. G. (1998), Diagnostic Recognition: Task Constraints, Object information and Their In-teractions, *Cognition* **67**, 147-179.

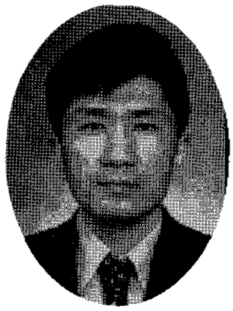
[17] Shilman, M., Pasula, H., Russell, S. and Newton, R. (2002), Statistical Visual Language Models for Ink Parsing. *in Proc. of the 2002 AAAI Spring Symposium – Sketch Understanding* 126-132.

[18] Segzin, T. M., Stahovich, T. F. and Davis, R. (2001), Sketch based Interfaces: Early Processing for Sketch Understanding, *in Proc. of the 2001 Workshop on Perceptive User Interface*, 1-8.

Jaeho Lee is a post-doctoral researcher in the Department of Industrial and Systems Engineering at Dongguk University, Korea. He received his Ph.D. degree in Department of Industrial and Systems Engineering at Dongguk University in 2007. His current research is in the Digital Product Design and Development.

JoonYoung Park is currently a professor of Industrial and Systems Engineering Department in Dongguk University which is located at Seoul, Korea. His research interests include rapid prototyping, mass-customization, haptic rendering, and developing geometric algorithms. He also serves as the vice president of SCCE (Society of CAD/CAM Engineers).

Jaeho Lee



JoonYoung Park