EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2006)
Thomas Stahovich and Mario Costa Sousa (Editors)

Sketch-based 3D Engineering Part Class Browsing and Retrieval

Suyu Hou and Karthik Ramani

Purdue University, West Lafayette, IN, USA

Abstract

We present a two-tier sketch-based engineering part retrieval system enhanced with classifier combination. Given a free-hand user sketch, we propose to use an ensemble of classifiers to estimate the likelihood of the sketch belonging to each category by exploring the strengths of individual classifiers. This supports high quality part retrieval by motivating user feedback with a ranked list of top choices. Three shape descriptors have been used to generate the probability-based classifiers independently. Experiments are conducted using the Engineering Shape Benchmark database in order to evaluate the selected combination rules before we integrate the best rule for sketch classification. User studies with the system show that users can easily identify the desired groups and then the parts. In addition, the precision attained using the synthesis is better than results from independent classifiers when applied to both user sketches and 3D models.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 Interaction techniques, I.5.4 Application

---

1. Introduction

It is well-recognized that engineering design starts with a sketch. Sketch-based part retrieval is a more natural form for searching during the stage of earlier concept design than example-based part retrieval. When a 3D query example is not available, sketch will be especially useful. Therefore, it is necessary to have a fast and effective system for sketch-based engineering part retrieval.

Most of the sketch-based retrieval systems focus on searching of 2D sketches/images. Recently, several studies have been conducted to retrieve 3D models based on 2D sketches [FMK*03, PR05]. A common method to retrieve 3D models using sketches is to represent the sketch and views of the database model by a set of shape descriptors. The system then computes the similarity metric between the query and the database model based on a predefined cost function. However, the system often retrieves mixed classes of models without fully considering the user intent embedded in the query, thus causing a gap between user expectations and system retrievals. For engineering reuse, it is important not only to retrieve parts with similar shape, but also to match retrievals with similar functions to the query. Therefore, it is important for the user to obtain functional class consistency besides shape matching.

In this paper, we introduce an approach to support sketch-based engineering part class browsing and retrieval driven by classification. We mainly focus on applying sketch-based classification for the goal of high quality retrieval. The key idea is to elicit the user to provide a relevance feedback to a list of part categories obtained by sketch classification. In addition, the strategy of classifier combination is employed to boost the performance of sketch classification.

The main advantage of the proposed approach is its use of a probability-based classification to orienteer the user in a two-tier search framework. The probability-based classification can narrow down the choices for user

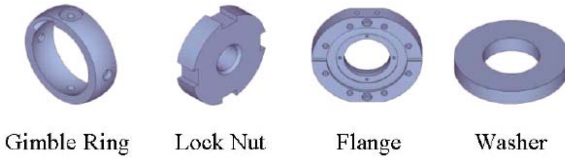

Figure 1: *Similar engineering model from different classes*

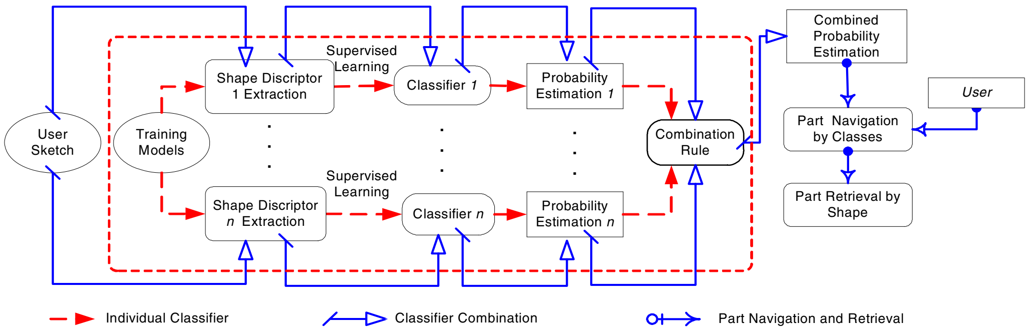

Figure 2: *System architecture*

selection without the risk of binary decisions obtained from regular classifiers. This is especially beneficial for sketch-based engineering part retrieval. First, sketches are always ambiguous. By motivating the user to disambiguate the intermediate result, the system actively gets the consent from the user. Secondly, engineering parts have a more complex scenario than regular multimedia models for categorization. There is no unique criterion for classifying engineering models. Even one engineering model sometimes can be classified into different classes by various standards [IR05, JKI*06]. Figure 1 shows an example of some similar engineering models from different classes. Therefore, another objective of this paper is to provide applicable classification mechanisms for engineering parts. Lastly, the probability estimation can facilitate post processing. The interpretation of the probability output is independent of the types of classifiers; only its quality depends on classifiers. We take advantage of this fact to attain a combined estimation so as to improve the confidence for the decision making.

To the best of our knowledge, we do not know of any existing work that supports sketch-based 3D part class browsing and retrieval using classification. The rest of this paper is organized as follows. In Section 2, we briefly describe the system architecture of the proposed framework. We then present the major modules of the framework in Section 3. Section 4 shows the user studies and includes a discussion. The paper concludes in Section 5.

2. System Architecture

We use a 3D part retrieval system, ShapeLab [PR05], and the Engineering Shape Benchmark database (ESB) [JKI*06] as the test bed for this study. Given a query in the form of sketches from three orthogonal views of a 3D object, we allow the user to browse the most possible classes based on the query sketch. This is obtained by a probability-based classification engine. The classifier differentiates the likelihood of the query belonging to each 3D part category from ESB. A ranked list of top categories is provided to the user based on the degree of agreement between the query sketch and the classifier for each class. The system will then perform the shape matching within the categories that the user prefers. The idea of classifying a query sketch into 3D part category comes from the notions that i) engineers usually express their concept of a 3D shape with three 2D orthogonal views without losing much information [PR05]; ii) consistency exists between the user sketch of orthogonal views of 3D objects and the views automatically generated from the 3D model by the pose estimation method based on Virtual Contact Area (VCA) used by the ShapeLab system [PR05]. In this paper, instead of using a conventional single classifier, we propose to synthesize independent classifiers to improve classification performance and avoid a biased decision.

Figure 2 presents the system work flow. First, training data from ESB is used to finalize the individual classifiers as shown inside the left dotted window of Figure 2. Each shape descriptor corresponds to a specific classifier. Different classifiers which output the probability estimation of data being classified to a particular class are developed separately using supervised learning. Meanwhile, we exploit the classification output from the training data to estimate the optimal weight for the linear combination model used later for the real searching. The main idea is that given a classifier, its contribution to the combined prediction of the testing data is dependent on its performance with the training data. A classifier with better classification accuracy is considered to have better predication capability and will be given more weight for the combination model. Several candidate combination rules are proposed for this work which will be presented in Section 3. Testing data from ESB is employed to assess these combination rules before the rule is applied to the sketch input. The testing data and the sketch pass through the same processes of shape descriptor extraction and the classification estimations before reaching the combination stage, except that the sketch input utilizes the combination rule selected by the testing data. The combination rule designated in Figure 2 is the one finalized by the testing data. At the end, the system enables the user to browse the parts organized in classified groups and then to pinpoint the desired parts while avoiding browsing irrelevant parts. In some form our system performs the function of relevancy feedback using part classes.

3. Approach

3.1. Sketch Acquisition and Representation

The sketch acquisition module records users' search intent using sketch. Users can employ a pen or a mouse to sketch. In our system design, the sketches are drawn

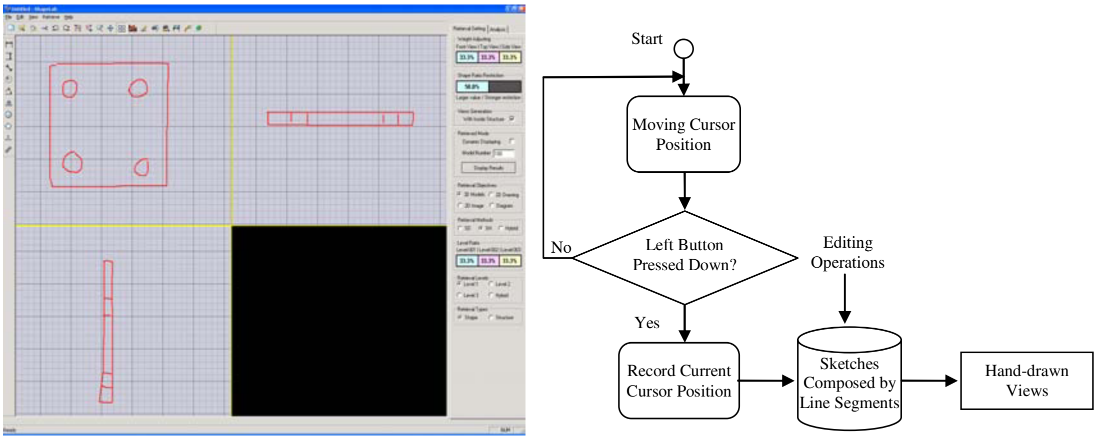

Figure 3: *Sketch acquisition module*: *(left) sketch editor, (right) information work flow*

through a sequence of strokes. Figure 3 shows the visual appearance and the architecture of the sketch editor. During the sketching process, the system monitors the action of the mouse or the pen. Once the mouse cursor is moving and the left button is pressed down, it can be concluded that users have begun to draw sketches. Now the moving path of the cursor is recorded in real time with the end of the stroke indicated by the release of left button.

Each track of a stroke S is composed of a sequence of small line segments rather than image bitmaps: $S = \{((x_i, y_i), (x_{i+1}, y_{i+1}), t_i) \mid 0 \le i \le n\}$ where n is the total number of line segments included in a single stroke S, (x_i, y_i) and (x_{i+1}, y_{i+1}) are the two ending points of a small line segment at time t_i. Consequently, sketching activity A is usually formed by a sequence of stroke $A = \{S_i \mid 0 \le i < m\}$ where m is the number of strokes. In the end, the desired shape descriptor will be extracted from these strokes [PR06].

In the sketch process, it is inevitable that the user will make some mistakes. Therefore, besides the sketch operations, some editing operations are also provided to users. Some basic operations, such as erase, trim, move, rotate, zoom, and view copy are included. More operations can be added into this system, although only a few basic operations are provided in this system. In the future, sketch beautification from [PHR06] will be integrated with the current system to regulate the freehand sketch, which is expected to boost the performance of sketch-based shape analysis.

3.2. View-based 3D Shape Description and Its Benefits for Sketch Recognition

In our system, three 2D orthogonal views by pose determination and projection are automatically generated from each 3D triangulated model [PR05]. Therefore, given a shape description from views of a query, the system can find similar 3D models. Compared to most other existing 3D shape descriptors which capture the form from 3D models directly, shape signatures generated from views perform well and can be applied to view-based 3D model retrieval directly [COT*03]. Similarly, it is intuitive to accept the idea of sketch-based 3D model classification given the fact that sketches are the most natural form for shape expression. Sketch-based 2D symbol classification/recognition has progressed extensively in the past decades. Most classifiers/recognizers either use a coded template for matching [CDP*04, FPJ02, FJ00, VCC01, AD04] or require sets of training data to reliably learn new symbols [LQX01, SD05, HN04, KS04, KS05, RUB91]. Among them, methods using statistical learning for symbol classification share a similar background with this paper even though we mainly focus on sketch-based 3D part classification. In [KS05], classifier combination is applied to sketch symbol recognition using user-defined training examples. This method can reach higher classification accuracy because the sketch query is consistent with the training data. However, the idea is not applicable for 3D engineering parts because the engineering part classification scheme and training data are hard to define on the fly. Besides, engineering parts are difficult to sketch formally for training purposes. Therefore, we motivate the user to help the system obtain the best retrieval with the classification engine defined by real engineering models.

For our work, we rely on shape descriptors as feature vectors for the classification problem. Our experience with sketching has shown that users prefer to draw a model at a higher level, thus closer to the contour level of the view generated from the 3D model , which captures an outer boundary and internal boundaries from a specific view of a 3D model [PJH*06]. In this paper, two criteria are needed to meet the shape descriptor selection. First, it has to be applicable to both 2D views and the sketch. Second, it is rotation invariant so that optimal alignment identification can be saved. Three shape descriptors are chosen to represent the shape content from the sketches/views in this context: 2.5D Spherical Harmonics (SH) from the contours [PR06], Fourier Transform (FT) from the outer boundary [ZL01], and the Zernike moments (ZM) from the region inside the outer boundary [KH90]. These three shape descriptors have been shown empirically to perform well in the task of shape matching. Although our framework is independent of the shape descriptor selected, we choose

Table 1: *Classification accuracy for testing data from ESB*

	Individual Classifiers			Combination Rules				
	SH Contour	Fourier	Zernike	Majority Vote	Product Rule	SA	WA MSE	WA MCE
Case I	*68.69%*	*66.38%*	*63.35%*	*71.80%*	*74.50%*	*75.00%*	*73.31%*	*75.00%*

2.5D SH, FT and ZM because they complement the shape description from different perspectives using dissimilar techniques. For example, 2.5D SH includes the internal boundaries in addition to the outer boundary considered by FT, while ZM reflects more of the internal details by describing the distribution inside the region. Therefore, it is expected that classifier combination can achieve a better performance. For the current work, we concatenate shape signatures generated from three views to form a single feature vector $x \in \Re^n$. Data produced at this stage will then be employed for classifier recognition.

3.3. Probability-based Classification

Many algorithms have been presented to classify 3D objects using machine learning techniques [BD06, HLR05, IR05, and ZC02]. Given a classification scheme of C classes $\Omega = \{\omega_1, \omega_2, ...\omega_C\}$, and a set of labeled training examples $X = \{(x_i, y_i), i = 1,...N\}$ with $x_i \in \Re^n$ and $y_i \in \Omega$ from database, a common goal is to classify a unique example x into a particular class ω_i. A simple method is to recognize the classifiers using the training data and to assign the query to the class that has the largest confidence from the prediction. Usually, the system outputs binary decision $P = \{p_1 = 0,...p_k = 1,...p_C = 0\}$ when $x \in \omega_k$, indicating that only the class that has the largest confidence wins the verdict. This approach, however, may lead to an inappropriate consequence for sketch-based engineering part classification. Unlike the binary classifier which hardens confidence measurement into a binary decision, the proposed probability-based classifier normalizes the confidence measurement into a probability output $P = \{p_1,...p_k,...p_C\}$ with $\sum_{k=1}^{C} p_k = 1$. Besides, there is no need to normalize the classification output for synthesis because the probability can be universally interpreted. Several algorithms have been presented to produce the probability output from pattern classifiers [WLW04]. In this paper, we chose Support Vector Machines (SVM) [CL01] as the classifier because of its quality although there are other applicable classifiers such as KNN [KUN04], Gaussian linear classifier [TBD*00]. Steps following the conventional procedures are taken to produce a classifier for each shape descriptor. A set of probability estimations $\{P_{2.5DSH}, P_{FF}, P_{ZM}\}$ for the query will then be generated in a parallel way using the resulting independent classifiers.

3.4. Classifier Combination Rules and Evaluations

Recent applications in combining multiple classifiers for the classification problem have shown strong evidence that strategies of taking advantage of various resources outperform traditional monolithic classifiers [RKW04]. The combined estimation theoretically always avoids the worst case and it even outperformed individual classifiers in our experiment as we demonstrate later. Inspired by this observation, we employ the strategy of classifier combination for sketch-based classification. The competency of classifier combination also implies that the system does not require as much training data as a monolithic classifier in order to reach the same performance. Therefore, the tradeoff between classification accuracy and amount of training data can be coordinated with the tactic of classifier combination in case the database does not have enough training data, as is often seen in reality. Several existing popular classifier combination rules are presented here for selection: Majority Vote, Product Rule [KHD*98], Simple Average (SA) [KHD*98, FR05], Weighted Average (WA) using Minimum Square Error (MSE) for weight estimation [BSE*97] and WA using Minimum misclassification Error (MCE) for weight estimation [UED00].

We examined the combination rule proposed above using real data from ESB. There are a total of 856 models in ESB with 55 out of 856 models which are miscellaneous and do not belong to each of the 42 classes. Therefore, there are a total of 801 models grouped in 42 classes in ESB. The size of each group varies. The maximum size of a group in ESB is 58, while the minimum size of a group is only 4. Half of the data from each group is randomly selected as the training data. The average training size from the 42 groups of training data with different sizes is 19.6 with a standard deviation of 14.6, which indicates the complexity of our classification problem. Training data from half of the ESB is first used to recognize the classifier and then to estimate the weight for a linear combination using MSE or MCE. Testing data from the remainder of the database will then be employed to evaluate the quality of the combination rules using the classification estimation from independent classifiers.

The results from Table 1 show that each combination rule outperformed the classification performance from individual classifiers. Even the worst combination (Majority Vote) had over 3% accuracy increase over the best individual classifier. SA and WA using MCE had the most competitive performance than other combination rules over our testing data. However, WA by MCE needs training data for weight estimation but without guarantee of better synthesizing results. The product rule also shows good performance in this experiment. However, the risk associated with this method when one classifier has a large

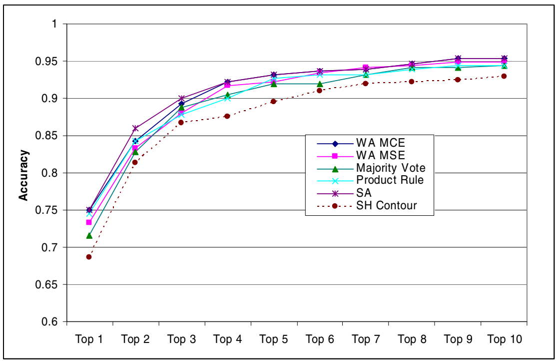

Figure 4: *Relaxed classification accuracy*

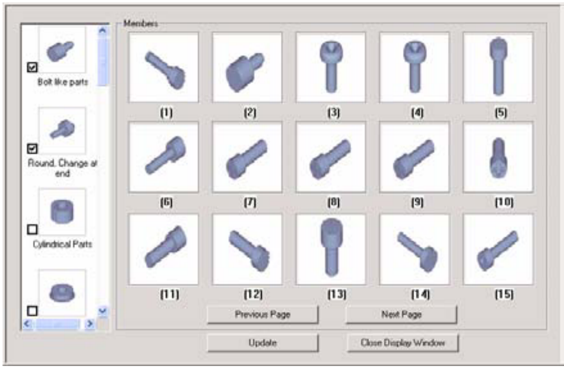

Figure 5: *GUI of part class browsing and retrieval*

estimation discrepancy from others [KHD*98] rules it out from our selection. The results are consistent with the experiment studies in [FR05] which conclude that SA is as good as WA sometimes in reality although the author also claims that WA is better than SA theoretically. We therefore choose SA as our sketch-based classifier combination rule. Besides, high quality and dissimilar classifiers can be further inserted into the combination without making modifications to the current system.

Figure 4 shows the relaxed classification accuracy for selected classifiers up to the top 10 using the testing data from ESB. The relaxed classification accuracy for top K is defined as $RCA(K) = \sum_{i=1}^{K} n(i) \Big/ N$ where $n(i)$ is the number of correct classifications at i^{th} rank, and N is the total number of the testing data. Solid lines represent the relaxed classification accuracy under different combination rules, while the dotted line comes from the best individual classifier, the one developed from 2.5D SH. The combination rule can reach about 92% within the top five for the testing data. After the top 8 results, the relaxed classification accuracies become stable and approximate to 96% at top 10. At this stage there is not much difference among different combination rules. The output shows some promising results given the classification complexity in this problem: 42 classes with non-uniform training size. Difficulties may arise when the classification engine is applied to the sketch input. However, the overall performance boosts our confidence in using the proposed framework for sketch-based 3D part retrieval.

3.5. GUI Design

We have implemented the proposed sketch-based part retrieval for ShapeLab. After the user submits the sketches using the sketch editor described in Section 3.1, the system will provide a list of 20 classes sorted by the probabilities from the classification output. We let the user browse the top 20 out of the total of 42 classes to show the advantages of the proposed work while at the same time avoiding missing identifications. Figure 5 shows the GUI of the implementation. On the left-hand side of the GUI are the class images of the ranked list that prompt the user to choose. Models of the selected class will be shown in the order of shape similarity to the query on the right-hand side. The default images of the models will be the ones belonging to the class that has the highest possibility. The user is then able to browse groups of models based on the selection that he/she thinks as the right classes. The proposed framework can not only improve the search effectiveness and efficiency, but also enhance user interaction by involving only a limited number of highly possible choices. This design will be especially useful for a large database with a large number of classes.

4. User Studies

The overall purpose of the experiments is to appraise the proposed idea with respect to the system performance for query by sketch. Besides, we formally quantify how well the combination rule can improve the performance over single classifiers. The results assist us to understand the difference between sketches and views from a 3D model. To obtain an objective evaluation, we conduct a user study consisting of two independent experiments. In the first experiment, users are given examples of models from our ESB to sketch. In the second experiment, engineering CAD models outside ESB are provided for the user to sketch.

In this experiment, people with no background of ShapeLab system are chosen to participate in the study. Users are allowed to take some time to acquaint themselves with the hardware and the system. During the practice, users did not encounter any problem. Typically they spend several minutes before the real tests begin. There is no instruction on how to sketch the 3D object in particular, for example, the view definition for the orthogonal views (e.g., front, side, or the top view), or the amount of detail to sketch (e.g. whether to sketch the external contour alone or the complete drawing with hidden lines).

Five different users are asked to generate a freehand sketch for each example. For each sketch input, two kinds of classification engines are used for the tests. The first one is the best individual classifier which employs 2.5D SH feature vector. The other is the combined classifier using Simple Average rule selected through the ESB testing data.

Table 2: *Results for sketches of examples from ESB*

	Average Best Rank by Sketch	Overall Average Rank by Example	Average Best Classification Accuracy		
			Top 1	Top 5	Top 10
Classifier by 2.5 SH	3.33	1.67	33.33%	75.00%	100.00%
Classifier by SA Combination	3.00	1.50	33.33%	75.00%	100.00%

Table 3: *Results of sketches of examples from outside ESB*

	Average Best Rank by Sketch	Average Best Classification Accuracy			
		Top 1	Top 5	Top 10	Top 15
Classifier by 2.5 SH	7.29	28.57%	28.57%	71.43%	100.00%
Classifier by SA Combination	6.29	28.57%	42.86%	85.71%	100.00%

Each sketch goes through the two classification engines separately. The user is then asked to give his/her evaluation of the rank of the right class as shown by the class images on the left hand side of the window. We then record the ranks for this example from the results of the five sketches. The overall performance of the selected classification engine can then be evaluated based on the results from sketches of all the examples.

4.1. Sketch Examples from ESB

There were a total of 12 examples, 60 user sketches tested in this experiment. The examples containing a wide variety of engineering shapes are randomly picked from the testing dataset of ESB. These examples are not involved in the training process. Besides, the sketch inputs are different from the views generated from the training examples. Therefore, it is fair to say that this experiment can objectively reflect the performance of the system. However, it is expected that the result will be different from the result of the second experiment since these examples come from the same database and belong to one of the 42 classes in the training data. Table 2 shows the average best performance of each classification engine for the sketched input. We pick the best rank for each example. This is because the sketches created by the users are sometimes different from the view permutation generated from the 3D models as we find out during the experiments. Figure 6 gives an example shown to the user and the five sketches involved in this experiment. It is obvious that the user sketches have some dissimilar characteristics from each other and are not guaranteed to have the consistent view correspondence compared to the views generated by the system as shown in Figure 6 (a). The views generated by the ShapeLab system have certain patterns driven by the VCA algorithm. Therefore, it decides the classifier produced by the training models. If the sketch does not follow the convention of view generation, it is impossible to get the best matching. The overall performance expressed by relaxed classification accuracy is obtained by putting the best rank of each example into a histogram. We also provide the average rank of these examples in Table 2, with the views automatically generated by the system instead of user sketch. The purpose is to compare how much influence the user sketch can have on the classification performance. From the results, it can be seen that there is certain difference between classification for views generated from examples and classification for sketches of the examples. The difference is mainly because of the sketch ambiguities between the training views and the sketches. This is because we have already excluded the reason for different view correspondence in calculating the overall rank. The combination rule marginally improves the classification performance from the best individual classifier. A smaller value of the average rank indicates a better performance of the classifier. Although it is not the determining factor for the classification output, the strategy of classifier combination can certainly help the system to obtain the best performance when no prior knowledge of the individual classifier is available.

4.2. Sketch Examples from outside ESB

The aim of this experiment is to find out how flexible and robust our system is when the data is outside the range of our database. Half of these examples do not conceptually belong to any of our ESB classes. Some of them are even hard to sketch based on user experiences. There are a total of 7 examples and therefore 35 user sketches to evaluate the performance of the classifiers. Since there is no information as to which classes these examples belong to, we let the user decide the rank based on the similarities between the query and the class images shown on the left-hand side of the window. It is possible that the same query may belong to multiple classes. Therefore, the rank evaluated by the user is the highest rank from possible classes given by the system. We calculate the overall performance for each example following the same procedure as the first experiment. Table 3 lists the results obtained from this experiment. The results are not as good as those of the first experiment as expected. We further investigated the results and found out that those examples not belonging to any of the classes have lower rank evaluations. However, the user can still find promising categories as he/she browses the classes. Similarity, the combination rule improves the classification performance.

Both experiments demonstrate that the sketch has more uncertainties compared to real examples. In fact, query by sketch commonly does not have as good retrievals as query-by-example. The goal for our work is to improve the end retrieval by orienteering user feedback at the first tier. Therefore if we successfully obtain the user feedback for the second tier search, we can still achieve the goal. The experimental results support our proposition of providing the user with the desired choices within a certain range.

5. Conclusions and Future Work

This paper presented and explored a framework to support fast and effective sketch-based 3D engineering part retrieval driven by classification. We described the idea of

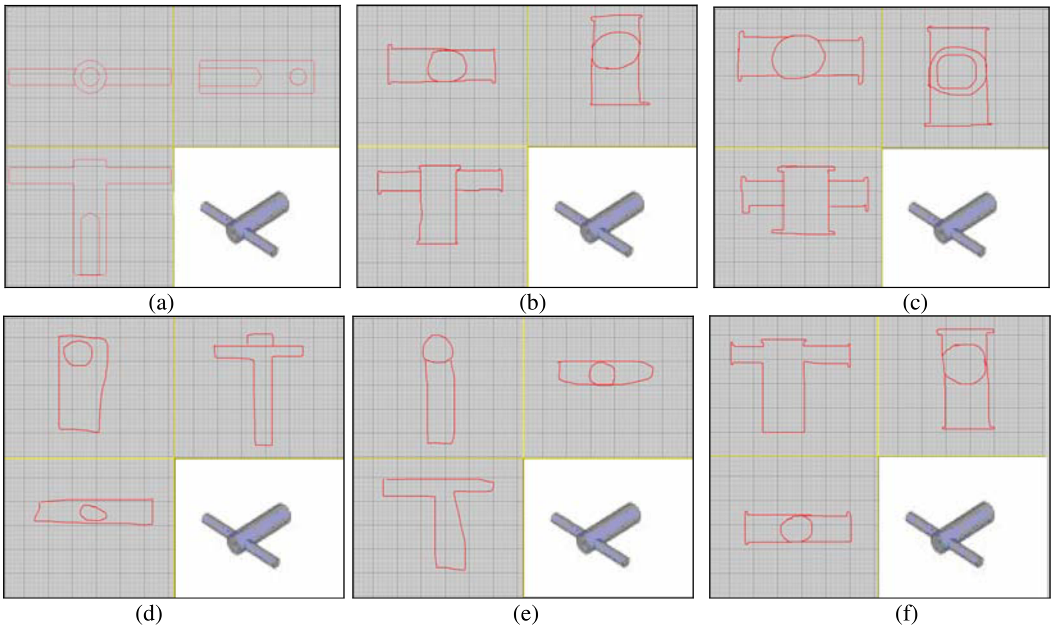

Figure 6: *(a) views generated by system for the example, (b-f) sketches created by different users for the same example*

using classifier combination to improve the sketch classification performance for the good of downstream part class browsing and retrieval. The use of a probability-based classifier and its merits in classifier combination can be applied to any type of two-tier content-based search system. We then conducted two user studies to evaluate the robustness of the proposed work. Different datasets were used to examine the classification accuracy of two classification engines: the best individual classifier and the Simple Average combination rule. The experimental results showed that the system can output the right class within a tolerance range, thus guiding the user to choose the preferred class for shape matching. The use of an ensemble of classifiers improved the classification accuracy both for sketches and 3D models. In addition, our system showed distinction in classification output to examples that did not come from any of the classes of ESB. Important factor to decide the classification performance for sketches included the ambiguities and inconsistencies of the sketches with regards to the training views. In the future, the sketch beautification will be integrated into the current system in order to regulate the sketch input. It is possible that sketching three views of a complex object will be hard for most people. We will use sketch beautification to partially help the user to draw complex views, and we may add more sketch utilities to address this issue in the future. Also the user may be able to use a photograph of an object and use a model with detected edges as input. In addition, for better classification performance, we will permute the sketches in order to find the best correspondence with the training views.

Acknowledgements

We would like to acknowledge partial support of the National Science Foundation Grant (IIS 0535156).

Reference

[AD04] ALVARADO, C., DAVID, R.: SketchREAD: a multi-domain sketch recognition engine. *In Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, (2004), 23-32.

[BD06] BARUTCUOGLU,Z., DECORO, C.; Hierarchical shape classification using Bayesian aggregation. *Shape Modeling International*. (June 2006)

[BSE*97] BENEDIKTSSON, J.A., SVEINSSON, J.R., ERSOY, O.K., SWAIN, P.H.: Parallel consensual neural networks, *IEEE Trans. Neural Networks*, 8, 1, (1997),54-64.

[CDP*04] COSTAGLIOLA, G., DEUFEMIA, V., POLESE, G., RISI, M.: A parsing technique for sketch recognition systems. *In Proceedings of 2004 IEEE Symposium on Visual Languages-Human Centric Computing* (VLHCC'04), (2004), 19-26.

[CL01] CHANG, C-C., LIN, C-J.: LIBSVM : a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. (2001).

[COT*03] CHEN, D-Y., OUHYOUNG, M., TIAN, X-P., SHEN, Y-T.: On visual similarity based 3D model retrieval. *Computer Graphics Forum*, (2003) 223–232.

[FJ00] FONSECA, M.J., JORGE, J.A.: Using fuzzy logic to recognize geometric shapes interactively. *In Proceedings of the 9th IEEE Conf. on Fuzzy Systems*, 1, (2000), 291-296.

[FMK*03] FUNKHOUSER,T., KAZHDAN, M., CHEN,J.,HALDERMAN,A.,DOBKIN,D., JOCOBS,D.: A search engine for 3D models. *Transactions on Graphics*, 22, 1 (2003), 83-105.

[FPJ02] FONSECA, M.J., PIMENTEL, J., JORGE, J.A.: Cali-an online scribble recognizer for calligraphic interfaces. *In Proceedings of the 2002, AAAI Spring Symposium on Sketch Understanding*, (2002), 51–58.

[FR05] FUMERA, G., ROLI, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27, 6, (2005) 942-956.

[HLR05] HOU S., LOU K., RAMANI K. SVM-based semantic clustering and retrieval of a 3D model database, *Computer Aided Design and Application*, 2, 2, 2005, 155-164.

[HN04] HSE, H., NEWTON, A.R.: Sketched symbol recognition using Zernike moments. *In Proceedings of 17th International Conference on Pattern Recognition* (ICPR'04,, (2004),367-370.

[IR05] IP, Y., REGLI, W. C.: Manufacturing processes recognition of machined mechanical parts using SVMs, *AAAI2005*, (2005), 1608-1609.

[JKI*06] JAYANTI, S., KALYANARAMAN, Y., IYER, N., RAMANI, K.: Developing an engineering shape benchmark for CAD models, the Special Issue on Shape Similarity Detection and Search for CAD/CAE Applications, *Journal of Computer Aided Design*, in print, (2006).

[KUN04] KUNCHEVA, L.I., *Combining pattern classifiers: methods and algorithms*. Wiley, 2004.

[KH90] KHOTANZAD, A., HONG, Y.: Invariant image recognition by Zernike moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, 5, (1990), 489-497

[KHD*98] KITTLER, J., HATEF, M., DUIN, R. MATAS, J.: On combining classifiers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20, 3,(1998),226–239.

[KS04] KARA, L.B.., STAHOVICH, T.F.: An image-based trainable symbol recognizer for sketch-based interfaces. *In Proceedings of AAAI Fall Symposium Series 2004*: Making Pen-Based Interaction Intelligent and Natural, (2004), 99-105.

[KS05] KARA, L.B.., STAHOVICH, T.F.: An image-based, trainable symbol recognizer for hand-drawn sketches. *Computers & Graphics* 29, 4, (2005) 501-517.

[LQX01] LIU, W.Y., QIAN, W.J., XIAO, R. Smart sketchpad - an on-line graphics recognition system, *In Proceedings of Sixth International Conference on Document Analysis and Recognition*, (2001), 1050-1054.

[PHR06] PU, J.T., HOU,S., RAMANI K.: Toward freehand sketch beautification driven by geometric constraint. Submitted to *ACM Transactions on Graphics*.

[PJH*06] PU J.T., JAYANTI S., HOU S., RAMANI K.: Similar 3D model retrieval based on multiple level of detail. Accepted by *the 14th Pacific Conference on Computer Graphics and Applications*, (2006).

[PR05] PU, J.T., RAMANI K.: A 3D model retrieval method using 2D freehand sketches. *Lecture Notes in Computer Science,* vol. 3515 (2005) 343-347.

[PR06] PU, J.T., RAMANI K.: On visual similarity based 2D drawing retrieval. *Journal of Computer Aided Design*, 38, 3 (2006) 249-259.

[RUB91] RUBINE, B.: Specifying gestures by example. *ACM Transaction on Computer Graphics*, 25, 4, (1991), 329-337.

[RKW04] ROLI, F., KITTLER, J., WINDEATT, T., eds.: Multiple classifier systems, *Lecture Notes in Computer Science*, 3077, (2004).

[SD05] SEZGIN, T.M., DAVIS, R.: HMM-based efficient sketch recognition. *In Proceedings of the 10th International Conference on Intelligent User Interfaces,* (2005) 281-283.

[TBD*00] TAX, D., BREUKELEN, M. VAN., DUIN, R. KITTLER, J.: Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33, 1, (2000), 475-1485.

[UED00] UEDA, N.: Optimal linear combination of neural networks for improving classification performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (2000), 207-215.

[VCC01] VALOIS, J.P., CÔTÉ, M., CHERIET, M.: Online recognition of sketched electrical diagrams. *In Proceedings of Sixth International Conference on Document Analysis and Recognition*, (2001), 460-464.

[WLW04] WU, T.-F., LIN, C.-J., WENG, R. C.: Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, (2004)

[ZC02] ZHANG, C., CHEN, T.: A new active leaning approach for content-based information retrieval, *IEEE Trans. On Multimedia Special Issue on Multimedia Database*, 4, 2, (2002), 260-268.

[ZL01] ZHANG D. S., LU. G. J.,: Shape retrieval using Fourier descriptors. *IEEE International Conference on Multimedia and Expo (ICME),* (2001)1139- 114.