SVM-based Semantic Clustering and Retrieval of a 3D Model Database

Suyu Hou, Kuiyang Lou and Karthik Ramani

Purdue University, {suhou, kuilou, ramani@purdue.edu}

ABSTRACT

In this paper, we present a semi-supervised semantic clustering method based on Support Vector Machines (SVM) to organize the 3D models semantically. Ground truth data is used to identify the pattern of each semantic category by supervised learning. The unknown data is then automatically classified and clustered based on the resulting pattern. We also propose a unified search strategy which applies semantic constraints to the retrieval by using the resulting clusters. A query is first labeled with its semantic concept therefore shape-based search is only conducted in the corresponding cluster. Experiments are performed to evaluate the effects of the semantic clustering and retrieval respectively by using our prototypical 3D Engineering Shape Search System (3DESS).

Keywords: SVM, similarity gap, semantic clustering, shape similarity, CAD

1. INTRODUCTION

Engineering design and manufacturing has progressed extensively from 2D to 3D in the past decades. At the same time, 3D CAD models have proliferated with the advances in hardware and the benefits of using Computer Aided Design (CAD) and Manufacture (CAM) software. Therefore, the capability to access and reuse the existing CAD data is critical for competitive engineering product development. In the engineering domain, shape has played an important role in various stages throughout the product lifecycle such as concept design, analysis, process planning, cost estimation, and part family formulation [3]. Hence, shape-based 3D model retrieval, as a complement to text-based engineering information systems, makes knowledge reuse feasible based on geometry. Different shape representations have been developed to support effective retrieval of 3D models, including CAD data [7,14]. However, all of these representations are prone to the negative effects caused by the similarity gap between the lower level visual (shape) features and higher level semantic concepts. Therefore the search effectiveness is seriously affected.

Semantic concepts directly affect the human understanding of the visual content. In the engineering domain, geometry is closely related with higher level semantics such as manufacturing process, function and behavior. At the same time, geometry is also associated with lower level semantics such as assemblies and components [21], which are commonly referred to as objects in computer vision. In general, the higher the semantic level, the more domain-dependent expert knowledge is required to interpret the visual content to it. Similarly, the lower the semantic level, the closer it is to common human cognition. However, the lower level visual features generated using the techniques in [7, 14] usually do not by themselves reflect the semantic concepts of the data [20]. Therefore, feature vectors belonging to irrelevant semantic concepts may reside close in the distance space, which may result in low precision during the retrieval. At the same time, visual data belonging to the same semantic category may have feature vectors that are far away in the feature space, which bring low recall to the retrieval. In engineering, 3D CAD models are not just limited to their lower level visual (shape) representations. At the same time, search for information reuse of CAD data is also dependent on the different level of semantic concepts. Hence, methodologies and algorithms to reduce the similarity gap between the lower level visual features and higher level semantic concepts is one of the major challenges in this field.

In this paper, we propose a SVM-based semi-supervised clustering method for a 3D model database. The motivation for choosing SVM for pattern recognition is explained in the later part of the paper. The system first employs such semantic concepts as supervision for learning the patterns, and then clusters the database in an unsupervised way based on the resulting pattern. This leads to the result where the semantic concepts are embedded in each cluster. Therefore, indexing by semantic concepts can be applied later. The proposed method is conducted on a 3D engineering shape search engine [13]. By using the above mentioned approach, the database is clustered and indexed by referring to the mechanical part family catalogue at the semantic level. The part catalogue is commonly used by industry, which requests less domain knowledge while at the same time is acknowledged by most people. The query is first labeled by its part family name before the search based on its shape content is conducted. An automatic feature vector selection is performed in order to obtain the best retrieval. The implementation is built on C++, with the SVM engine provided by SVM Light [11].

This paper is organized as follows: Section 2 introduces the related work in brief. Section 3 presents the methods by which the semantic clustering and retrieval are designed and implemented. Also in this section, the system under which the experiments are performed is introduced. Section 5 describes the experiments and analyzes the results. The paper concludes in Section 6.

2. RELATED WORK

Recently, several research studies on clustering have been conducted to address the issues of the similarity gap and efficiency. Conventionally, an important procedure in a database is to cluster the data before indexing them within each cluster so as to make the retrieval more efficient. However, the results obtained from traditional unsupervised clustering methods do not reflect the semantics, thus causing the similarity gap. In [4], a system based on an unsupervised learning approach, called CLUE, is used to cluster the image database. The system is built upon the hypothesis that semantically similar images tend to cluster. However, this hypothesis may not be valid for other visual databases. This is because first, it may depend on the quality of the feature vectors; second, the similarity gap has been a common problem in various databases by different feature vectors [20]. In addition, as each cluster is only based on the numeric value of the visual features in CLUE, this system tries to alleviate the gap by the user's relevance feedback as well. In [17], a semantics-based clustering and indexing approach is implemented in SemQuery. Each semantic cluster (sub-cluster) is represented by a feature vector template and a scope of the feature space that contains ground truth images belonging to this cluster. The feature vector template is the centroid of the cluster, while the scope is measured by the statistical parameter of the cluster distribution. Again, it is based on the same assumption as CLUE. The general idea is that unknown data are classified into a specific semantic cluster if their feature vectors fall into the range of the cluster. Previously in [9], the K-Nearest Neighbor (KNN) is used for 3D engineering part classification. A supervised learning algorithm is employed to find a weight triplet for each part category, which is later used for a KNN-based classification. However, this work is limited to classification. Furthermore, there is no explicit pattern for each category, and the computations are performed between the query and all existing 3Dmodels for KNN based methods.

SVM, as a statistical learning approach, has been widely used in pattern recognition for the past several years. Besides the success of SVM in the area of text mining [10] and image classification [1], it is used in [16] to recognize 3D objects from images without the requirement of feature extraction and pose estimation. Moreover, it has been incorporated with relevance feedback to improve the shape-based 3D model retrieval [6]. However, the potential advantage of using SVM to semantically cluster the visual database has not been addressed. Particularly in 3D solid model databases, supervised machine learning techniques have seldom been used to improve the system performance.

Mathematically speaking, given a training set $\{x_i\}$ for two-class binary classification, if it is linearly separable, then a separating hyperplane, defined by a normal w and a bias b, will satisfy the inequalities:

$$y_i(w \cdot x_i + b) \geq 1 \qquad \forall i \in \{1, ..., N\} \tag{1}$$

Where $x_i \in \Re^D$ is the set of training data in feature space, D is the dimension of the feature space, $y_i \in \{-1, 1\}$ is the label for the binary classification, and N is the size of the training set. The algorithm tries to find an optimal hyperplane that leaves the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. The optimization is implemented through solving a quadratic problem in Eqn. (2) for $\{x_i\}$ if it is not linearly separable:

$$\begin{aligned} &\text{Minimize} && \frac{\|w\|^2}{2} + C\sum \zeta_i \\ &\text{Subject to} && y_i(w \cdot x_i + b) \geq 1 - \zeta_i \qquad i = 1, 2, ..., N, \quad \zeta_i > 0 \end{aligned} \tag{2}$$

Where ζ_i is the slack variable to relax the constraints, C is the parameter to regulate the trade-off between the training error and the margin. The points closest to the hyperplane are called support vectors. The optimization is built upon the idea of maximizing the margin $2/\|w\|^2$, the shortest distance between two points (support vectors) on two sides of the hyperplane as shown in Fig. 1. According to the generalization theory in [18], the larger the margin, the better the generalization is expected to be.

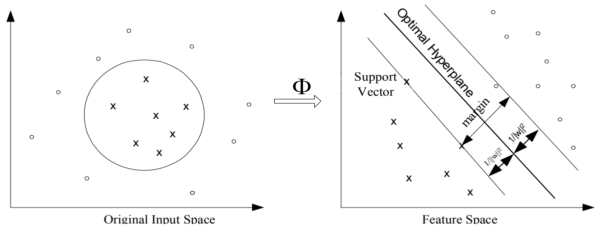


Fig. 1. Support vector machines

The most commonly used multi-class SVM classifier are "one-versus-rest" [18] and "pair wise coupling" [12]. By combining independently produced binary classifiers through the above-mentioned method, a multi-class SVM classifier can be obtained. The former method searches the classifier by training one class versus the other classes. A particular point is assigned to the class for which the distance from the margin, in the positive direction to this particular class, is the maximal. The latter one formulates $N(N-1)/2$ classifiers by comparing each class to each other class. To classify a point, the method combines the discrimination function from these $N(N-1)/2$ classifiers by using some voting scheme. Besides these two intuitive but brute force methods, other solutions have been studied by combining the constraint optimization problem with the quadratic objective function [5, 19]. In the latter one, a piecewise linear separation of k classes problem can be formulated in a single optimization solution. Besides the constraints exerted on Eqn. (2), another constraint is subjected to consideration for the piecewise comparison:

$$(w_{y_i} \cdot x_i) + b_{y_i} \geq (w_m \cdot x_i) + b_m + 2 - \xi_i^m$$
$$\xi_i^m \geq 0, i = 1, ..., N, m \in \{1, ..k\} \setminus y_i \quad (3)$$

Finally the decision function is made by:

$$f(x) = \arg\max_{n} \left(\sum_{i:y_i = n} \overline{\alpha}_i K(x' \cdot x_i') + \overline{b}_n \right) \qquad n = 1, ..., k \quad (4)$$

where x_i is any support vector with Lagrange multiplier $0<\alpha_i<C$.

One of the most important features about SVM is its use of kernels to improve the computational efficiency [18]. The most commonly used kernels for nonlinear pattern recognition are Polynomial kernels in Eqn. (5) and Gaussian Radial Basis Function (RBF) kernels in Eqn. (6):

$$K(x_i' \cdot x_j') = (x_i' \cdot x_j' + 1)^p \quad (5)$$

$$K(x_i' \cdot x_j') = e^{-\|x_i' - x_j'\|^2 / 2\sigma^2} \quad (6)$$

where parameter p is a natural number ,σ is a positive real value.

3. CLUSTERING AND SEARCHING

We have developed and implemented a SVM-based semantic clustering and searching system. There are three modules of work included in this process: (1) supervised training and validation, (2) semantic clustering and (3) a unified search based on the combination of the semantic concept and the shape content. Before the details of these modules are presented, the system on which these modules has been conducted and the motivation to adopt SVM for pattern recognition are introduced first.

3.1 3D Engineering Shape Search System

3DESS is a shape-based search system for 3D engineering parts [13]. The database of 3DESS includes 3D CAD models with each CAD model having its shape descriptor stored in the database as well. Through the processes of voxelization and skeletonization, various shape descriptors are generated for each CAD model. These descriptors include feature vectors such as moment invariants, geometric ratios, principal moments, and the skeletal graph [13].

The system searches through the database to find similar CAD models based on these shape signatures. Like other content-based search (CBS) systems, 3DESS has also been affected by the similarity gap occurring in the one-shot search system. Previously, relevance feedback had been employed to alleviate the problem [15]. However, how to address the problem of the similarity gap fundamentally is still an open issue for 3DESS and other 3D model search systems.

In this paper, we use 3DESS as the test bed for the proposed research. The system has already been well developed for the content-based search, which provides an experimental reference to be compared with the proposed method. Therefore through examining the feasibility of applying the proposed method to this system, we explore a new application area for SVM, which is targeted to alleviate the similarity gap in the shape database. In addition, we use a mechanical part family catalog as the reference for the semantic concepts in this paper. Mechanical part family is an important guide for engineering design at the semantic level while at the same time geometry plays an essential role in reflecting the semantic concept implicitly. The concept of part family has been widely used in text-based search systems and online catalogs. Therefore, studying the associations between some semantic concepts and geometry is another motivation for the research in this paper. In [21], an ontology-based product retrieval approach is proposed to support both the geometry and nongeometry in the engineering information system. The automatic association from geometry to nongeometry is based upon the results from this paper. In later sections, a detailed description of how the method is applied to 3DESS is presented.

3.2 Motivations for SVM

Generally speaking, SVM performs the pattern recognition by mapping the original lower dimensional input space into a higher dimensional feature space via a nonlinear function [2]. The motivation is that a linear model can be recognized in a higher dimensional feature space for the input training data, which may only be separated by nonlinear boundaries in the original space, as shown in Fig. 1. Moreover, even though we can think of the algorithm as a linear model in a higher dimensional feature space, it does not really involve any computation in the higher dimensional feature space. By using a kernel, all necessary computations are executed at the lower dimensional original input space thus having lower computational complexity. Finally, SVM extracts the pattern from the training set and represents it through a limited number of support vectors. Most important of all, unlike other pattern recognition methods such as neural networks, KNN, and Expectation Maximization (EM), which are built based on minimization of the empirical risk, SVM minimizes the structural risk, which is the probability of misclassifying unknown data drawn randomly from a fixed but unknown probability distribution [18]. In addition, unlike EM, which is also a widely used statistical pattern recognition approach, it has no assumption about the type of the probability density function of the data.

The data we address here are probably nonlinearly distributed in the original space based on their semantic concepts, in this case, the part families. This can be shown from the results of the experiments in the later part of the paper. Without knowledge of the data distribution, SVM can explicit the shape pattern embedded in each part family with the support from the above-mentioned advantages. In the future, experiments will be performed to compare the performance of different pattern recognition approaches under the same database.

3.3 Supervised Training and Validation

The purpose of this module is to recognize the pattern used for semantic clustering and semantic labeling thereafter. Once the pattern is identified, new data can be classified and inserted into the database without the necessity of updating the pattern again and again. There are three stages of work to obtain the pattern: data pre-processing, forming the data set, and mathematical model selection. The resulting pattern represents the implicit association between the semantic concepts and the visual content. Feature vectors extracted from the 3D models have to be normalized so as to fall within a small specified range. In this paper, each input datum is a hybrid of these three kinds of feature vectors: moment invariants, geometric ratios and principal moments, through which each shape is reflected from different perspectives. Next, the data are normalized by means of z-score normalization:

$$x_i = (x_i - \mu)/\sigma \tag{7}$$

where x_i is the input data of model m_i, μ is the mean value of these data and σ is their standard deviation. Compared with the commonly used min-max normalization method, z-score works well in the cases where the actual minimum and maximum values of the input data are unknown.

In order to obtain the pattern from supervised learning, the data are grouped and labeled based on their semantic concepts, i.e. part families. 3D models belonging to the same semantic concept are not necessarily visually similar. It is

$$\bigcup_{i=1}^{n} C_i = \Omega \qquad C_i \bigcap C_j = \varnothing \qquad i \neq j,\ i, j = 1, \ldots, n \tag{8}$$

assumed here that the data in each category C_i share only one semantic concept and:

where Ω is the universal set and n is the number of the semantic categories specified. Half of the data are used for training purposes and the remaining half are left for the semantic clustering.

Since the relations between the data and the semantic classes are probably nonlinear, the SVM mathematical model in Eqn. (2) is selected for recognizing the pattern from the training set. Polynomial kernels and Gaussian kernels are commonly used to deal with nonlinear cases. However, the polynomial kernel has more hyperparameters than the Gaussian (RBF) kernel to influence the complexity of the pattern. In addition, the Gaussian (RBF) kernel has fewer numerical difficulties [8]. In case the order is large, the kernel value of the polynomial kernel may go to infinity. We choose Gaussian kernel as the start for this stage. Next we need to identify the values for the parameters C in Eqn. (2) and σ in Eqn. (6) of the mathematical model so that the classifier can predict the class for the unknown data accurately. The most popular and practical method for estimating the generalization error of a classification system is k-fold cross-validation [8]. Among its different versions, 10-fold cross validation has been proved to work well in various studies. In this paper, we divide the training set into ten groups of approximately equal size, use nine sets for training, and use the remaining one to test for errors. The process repeats ten times and finally the best pair of (C,σ) is identified as the one under which the system has the minimum training errors.

3.4 Semantic Clustering

The initial clusters are formed from the ground truth training data, with each cluster representing the specified semantic category. The testing data formed during the previous module are clustered here. Based on the pattern developed in the module of training and validation, a multi-class classification based on [5] is used to classify the data. Unknown data are classified to a specific category if the distance from the optimal hyperplane in a positive direction to this category is the maxim. As a result, the database is semantically clustered through this approach. Fig. 2 illustrates the process of semantic clustering and search. In this paper, we assume the universal set Ω includes all semantic categories and each unknown datum must come from one of them. In future work, the unknown data with unidentifiable semantic categories will be treated with special care. In Fig.2, the process of semantic clustering and the process of unified search are illustrated. The solid lines represent the flow of semantic clustering while the dotted lines mean the flow of the unified search. Both of them use the resulting pattern from the training module to perform semantic clustering and semantic labeling respectively. The unified search uses the resulting semantic clusters to improve the search effectiveness.

3.5 Searching and Retrieval

The semantic clustering method developed in the first module not only allows for indexing at the semantic level, but also reduces the similarity gap by labeling the query before the search happens. The query is labeled with a semantic concept by being classified using the pattern developed before. The system then performs the shape-based search through the corresponding cluster to the query label (Fig. 2). The details of shape-based searching and retrieval may be referred to in [15].Through the proposed approach, the system can not only improve the search effectiveness, but also can save computational time by adding semantic constraints on the database. In 3DESS, various feature vectors corresponding to different shape descriptors are extracted from the 3D CAD model. An automatic feature vector selection is gone through to obtain the best shape signature for the specific query. The selection starts as follows. We treat each feature vector as a player p_i and the process of selection as a tournament made up of n players. The system obtains the top m retrievals of each feature vector p_i separately and stores them under an array r_i correspondingly. Each match takes out the lost side. Each player p_i has its vote v_i initialized to be zero before the match. The system makes the decision by following the rule that p_i wins the match if

$$\begin{gathered} v_i < v_j \qquad v_i = \sum_{k=1}^{m} w_k \times r_{ik} \\ with \quad \sum_{k=1}^{m} w_k = 1, \quad 0 < w_1 < w_2 < \ldots < w_m < 1 \qquad i \neq j, \quad i, j = 1, \ldots n \end{gathered} \tag{9}$$

The tournament ends when the system is left with only one player which is supposed to be the best shape descriptor for the query. Hence, the system can obtain the most shapes similar to CAD models with a semantic concept consistent with the query.

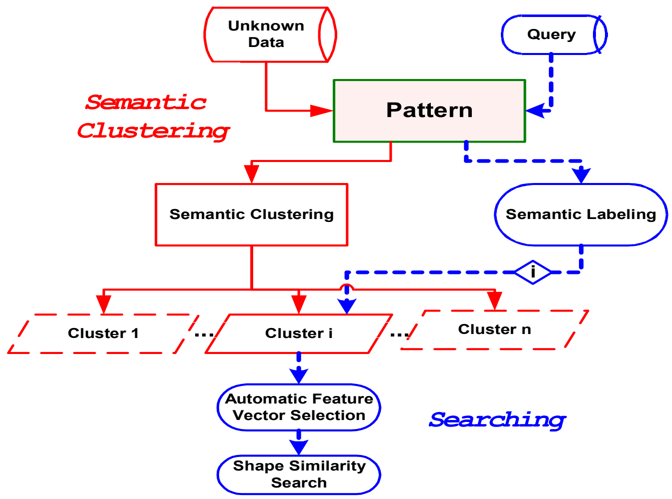


Fig. 2. Semantic clustering and retrieval based on the **Patten** obtained from the supervised training

4. EXPERIMENTS AND DISCUSSIONS

There are 218 3D CAD models selected from the 3DESS database. These models belong to six part families in this case: bracket, gear, handle, screw, shaft and door handle. Feature vectors including moment invariants, geometric ratios and principal moments are generated for each 3D model. The hybrid of these feature vectors is used as the input data for training and clustering. There are 116 data from these six categories used for training. The pattern used for semantic clustering and searching is developed by following the method described in the previous section. Finally, Gaussian kernel with $C=0.01$ and $\sigma=0.707$ is selected as the best mathematical model through 10-fold cross validation. The overall training error is 8.6%. In the next two sections, the experiments on clustering and searching are presented and discussed.

4.1 Clustering Results

There are 102 models from these six categories used for clustering. In Tab. 1, the initial size means the size of the original cluster before the clustering. The ideal size is the number of models in the database that belong to the cluster. The actual size is the size of the cluster after the clustering. The difference between the ideal size and the initial size is the number of the testing data. Among the 102 models, 90 models are accurately classified. Tab. 1 shows the results summarized from the clustering. The overall error rate is only 11.76%. Also, the average clustering error rate is 0.19 with a standard deviation of 0.18. From this result, it can be observed that the more training data a category has, the more accuracy it can obtain. The difference between the average error rate and the overall error rate is because of the different criteria used for measurement. We use the overall error rate to evaluate the system performance, while using the average error rate to demonstrate the importance of the size of the training data. Fig. 3 presents the results of the clustering under a selected mathematical model with $C=0.01$ and $\sigma=0.707$.

Moreover, Tab. 2 shows the results of the overall clustering error from some kernels under $C=0.01$ and different parameter values. Among them, Gaussian kernel with $\sigma=0.707$ has the minimum error rate. This result is consistent with the result from the 10-fold cross validation. From the two tables, we find that the testing error rate is higher than the value of the training error rate, which is the case in most of the real applications by machine learning methods. Also, our mathematical model selection is justified in that the relations between the data and the semantic classes are probably nonlinear.

Cluster name	Initial size	Ideal size	Actual size	Error	Average error	Standard deviation
Bracket	8	14	12	2/6	0.19	0.18
Gear	24	45	47	1/21		
Handle	24	46	49	0/22		
Screw	32	60	63	2/28		
Shaft	17	34	32	3/17		
Door handle	11	19	15	4/8		
Total	116	218	218	12/102	Overall error: 0.1176	

Tab. 1. Experimental clustering error

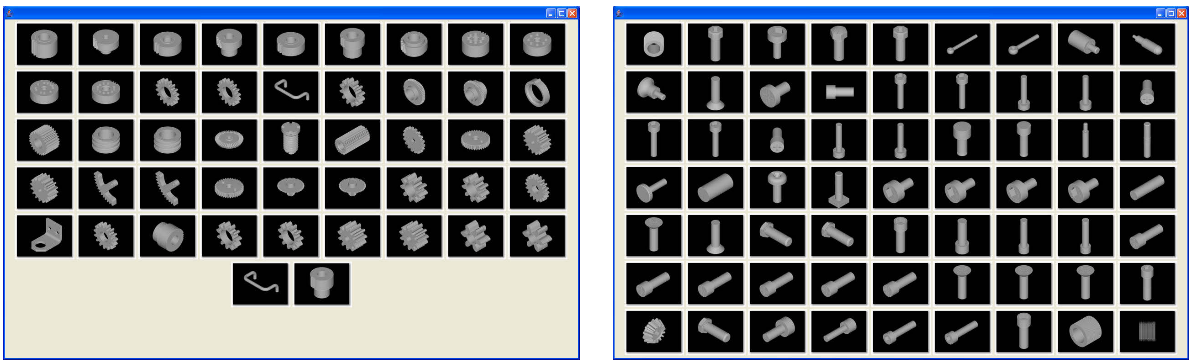


Fig. 3. Gear cluster (left) and screw cluster (right)

	Linear	Polynomial(p)		Gaussian (σ)		
Parameter p or σ	N/A	2	3	0.5	0.707	1.0
Overall training error%	32.76	10.34	10.72	10.72	8.6	12.59
Overall clustering error%	38.24	21.57	14.71	12.75	11.76	15.69

Tab. 2. Overall clustering error of different kernels

From the overall clustering errors obtained through the experiments, we have demonstrated that with the proposed method, the system can accomplish the objective of organizing the data at the semantic level within an overall error rate of 11.76%. In addition, from the observation of the clustering results, we find that most of the models in the same cluster belong to the same part family although they are not necessarily visually similar. This means that SVM can incorporate the designated supervision well into the process of pattern recognition, which is the key to associating semantic concepts with the 3D shape contents. This is especially important to the engineering domain because it is possible for various parts in the same part family to have different shapes.

In this paper, we have only focused on the steps needed to cluster the concepts semantically prior to the search although the results are based only on the use of a limited set of feature vectors. In the future, we will try to use optimized feature vectors for the purpose of improving the search performance under the proposed approach. As to the problem of misclassification, we will try to use a fuzzy classification scheme combined with other elaborate concepts and contents such as skeletal graph and size to improve accuracy. Also in this paper, we mainly focus on general purpose classification and clustering. We leave automatic hierarchical classification and clustering for our future work.

4.2 Search Results

In order to characterize the effectiveness of searching combined with semantic labeling and visual content, we plot the precision-recall curves (PRC) for selected queries under different methods. The selected queries are from different categories. Fig. 4 shows two of the query examples. These two selected queries belong to two different part families although they are visually similar. The fact that they have different functionalities and behaviors is important to engineering knowledge reuse. Therefore, by using the proposed approach, the system can avoid retrieving irrelevant models from the database. In this paper, the criteria to measure the precision and recall are dependent on the semantic consistency and the results from the semantic clustering. We determine whether two models are similar by first identifying if they are in the same part family, which is closer to human cognition compared with shape similarity. Also, we specify the maximal size of the retrieval equal to the ideal size of the specific semantic cluster. Therefore, the recall will not retrieve 100% if there is any clustering error for that cluster. In addition, the comparison between the proposed method and the shape-based methods are on the same scale. To simulate the real conditions of the search, these queries do not have semantic labels associated with them and do not exist in any of the clusters. The automatic feature vector selection chooses the best one among the hybrid feature vector, moment invariants, geometric ratios and principal moments. Fig. 5 presents some of the search results. Because the results from the moment invariants and the principal moments are similar and the space is limited, we do not show the results obtained from the moment invariants. In Fig. 5, the query is shown at the first column of each row. The first row shows the top eight retrievals from the combination of the semantic labeling and the shape content, which further resultes from the automatic feature vector selection; the second, third and fourth rows show the retrievals from the hybrid feature vector, geometric ratios and principal moment, respectively.

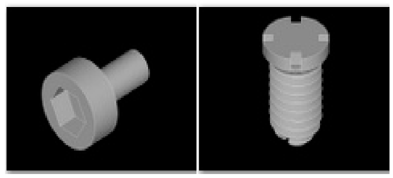


Fig. 4. Query examples No.1 screw (left) and No.2 worm gear (right)

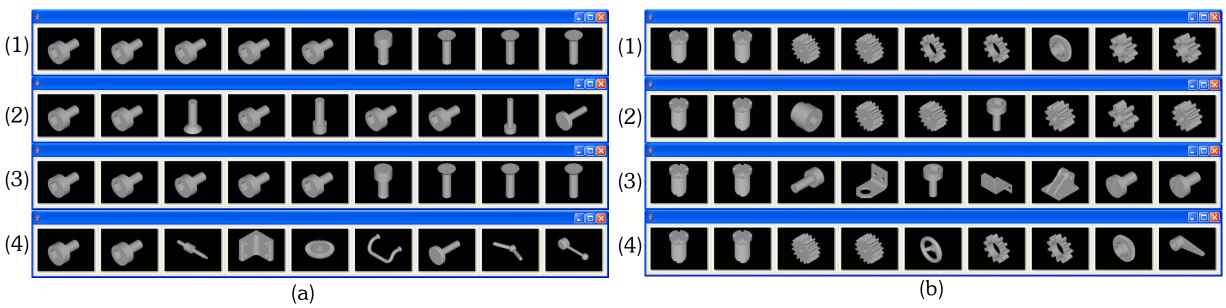
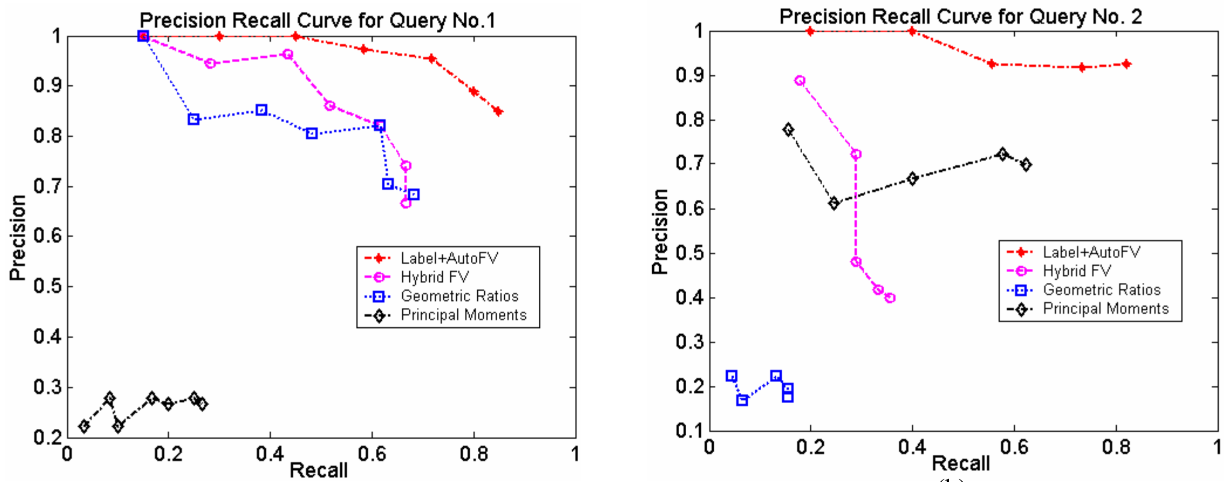


Fig. 5. Retrieval examples: (a.1 & b.1) semantic label + automatic feature vector selection, (a.2 & b.2) hybrid feature vector, (a.3 & b.3) geometric ratios, (a.4 & b.4) principal moments

From the observation of the search results, the proposed method preserves the semantic consistency if it correctly classifies the query. The automatic feature vector selection optimizes the search results, while the other methods, which are only based on shape, do not consistently have good results. This instability and uncertainty are also reflected by the contrast of the PRCs in (a) and (b) of Fig. 6. The PRCs of the proposed method consistently have higher precision and can reach higher recall than those obtained only from shape-based retrieval. The stability of the proposed method reflected from the PRCs further demonstrates its advantages over the other methods. Ideally, the precision should be 100% for the proposed method. However, the clustering errors may cause it to be less than anticipated. Nevertheless, the stability and high precision-recall of the proposed method still mark it as the best among the others. The precision of the proposed method drops as recall increases in Fig. 6. This fact implies that those that are misclassified at the semantic level rank lower when compared by shape. This fact supports the idea mentioned before that it is possible to further identify those false positives by other content or concepts. Meanwhile the PRCs for other methods which are based merely on shape sometimes increase as recall increases. This contradicts most of the common cases. It can be explained by the fact that the retrievals in these cases are based on shape only, while the PRC curves are obtained based on the measurement of semantic consistency. In these cases, the shape-based searches have a large percentage of higher rank retrievals semantically inconsistent with the query, while they have some lower-rank retrievals consistent with the query semantically. This fact reveals that content-based retrieval is generally unpredictable as to the semantic consistency since the similarity measurements by systems and by the humans are built on two different foundations. Also from the PRC, we find that different shape descriptors are good at evaluating shape similarity for different examples. However, it is hard to predict which one is better than the others for a specific query. This implication supports our case-by-case feature vector selection strategy.



(a) (b)

Fig. 6. PRC's for (a) query No. 1, (b) query No. 2

It is implied in PRC that, with the complexity of the shape increasing and the size of the database increasing, the differences among the results will be more obvious and the advantage of the proposed method will become more evident. However, the proposed method depends on the condition of correct classification. In the future, a fuzzy classification scheme combined with more knowledge will be designed to improve the accuracy of the classification.

5. CONCLUSION

In this paper, we have presented a clustering mechanism based on SVM to organize the data on a semantic level and a unified search strategy to conduct content-based search from the resulting semantic clusters. The results show that the semantic clustering can group the data on a semantic level with an overall error rate of 11.76% in our case. The search results demonstrate that the unified search strategy is promising. Compared with other search methods based only on shape signatures, the proposed search method improves the search effectiveness, as illustrated by the PRC results. In a word, the combination of supervised classification and content-based similarity retrieval can be an approach to alleviate the similarity gap for the current content-based search system, especially for 3D model retrieval. In the future, we will develop a fuzzy classification scheme based on conditional probability to deal with the risk of misclassification by the present method.

Acknowledgements

Initial funding for this project came from the 21st Century Research and Technology Fund award from the state of Indiana. We also acknowledge the support from Purdue Research and Education Center of Information Systems in Engineering (PRECISE) and the University Faculty Scholar Award from Purdue University to Professor Karthik Ramani, which seeded this project.

6. REFERENCES

[1] Ardizzone, E., Chella, A. and Pirrone, R. 2000, Feature-based Shape Recognition by Support Vector Machines, *ECAI-2000 Workshop, Machine Learning in Computer Vision*, Berlin, Germany.

[2] Burges, C. ,A tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, Vol. 2, No. 2, 1998,pp. 121-167.

[3] Cardone, A., Gupta, S.K., and Karnik, M., A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications, *ASME Journal of Computing and Information Science in Engineering*, Vol. 3, No. 2, 2003, pp.109-118.

[4] Chen, Y., Wang, J. Z. and Krovetz, R., CLUE: Cluster-based Retrieval of Images by Unsupervised Learning, *IEEE Transactions on Image Processing*, Vol. 13, No.15, 2004

[5] Crammer, K. and Singer, Y. 2001, On the Algorithmic Implementation of Multi-class Kernel-based Vector Machines. *Journal of Machine Learning Research*, Vol.2, pp. 265–292.

[6] Elad, M., Tal, A., and Ar, S., Directed Search In a 3D Objects Database Using SVM, Available: http://www.hpl.hp.com/techreports/2000/HPL-2000-20R1.html (November, 25, 2004)

[7] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D. and Jacobs, D., A Search Engine for 3D models, *ACM Transactions on Graphics*, Vol.22, No. 1, 2003, pp.83-105.

[8] Hsu, C. W., Change C.C., and Lin, C. J., A Practical Guide to Support Vector Classification, 2004 http://www.csie.ntu.edu.tw/~cjlin/libsvm (November, 25, 2004)

[9] Ip,C.Y., Sieger,L., Regli,W.C., and Shokoufandeh,A., Automated Learning of Model Classifications, *8th ACM/SIGGRAPH Symp. on Solid Modeling and Applications,* Seattle, Washington, 2003, pp. 322-27

[10] Joachims, T., *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1st edition. 2002,

[11] Joachims, T., Available: http://www.cs.cornell.edu/People/tj/svm_light/ *SVM Light Software,* 2004

[12] Kressel, U. H. G.,Pairwise Classification and Support Vector Machines, *In Advances in Kernel Methods – Support Vector Learning,* eds, Sch¨olkopf, B., Burges, C.J.C. and Smola, A.J. The MIT Press, 1999.

[13] Iyer N., Lou K., Jayanti S., Kalyanaraman Y. and Ramani K. ,A Multi-Scale Hierarchical 3d Shape Representation for Similar Shape Retrieval, *TMCE 2004*, Lausanne, Switzerland. 2004, pp. 1117-1118 .

[14] Iyer N., Jayanti S., Lou K., Kalyanaraman Y. and Ramani K. , Three-dimensional Shape Searching: State-of-the-art Review and Future Trends ,*Computer-Aided Design,* Available online.

[15] Lou, K., Jayanti, S., Iyer, N., Kalyanaraman, Y., Ramani, K. and Prabhakar,S.,A Reconfigurable, Intelligent 3D Engineering Shape Search System Part II: Database Indexing, Retrieval and Clustering, *Proceedings of ASME DETC' 03, 23rd Computers and Information in engineering (CIE) Conference,* Chicago, Illinois,2003.

[16] Pontil, M., and Verri, A., Support Vector Machines for 3D Object Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No.6, 1998, pp. 637–646.

[17] Sheikholeslami, G., Chang, W. and Zhang, A., SemQuery: Semantic Clustering and Querying on Heterogeneous Features for Visual Data, *IEEE Transactions on Knowledge and Data Engineering (TKDE),* Vol. 14,No.5 , 2002, pp. 988-1002.

[18] Vapnik, V., *The Nature of Statistical Learning Theory*. Springer-Verlag, 1st edition. 1995

[19] Weston, J., and Watkins, C., Support Vector Machines for Multi-class Pattern Recognition. *In Proceedings European Symposium on Artificial Neural Networks,* 1999.

[20] Zhao, R., Grosky,W.I., Bridging the Semantic Gap in Image Retrieval, *Distributed Multimedia Databases: Techniques & Applications*, 2002, pp.14 – 36.

[21] Li,Z., Anderson,D.C., and Ramani,K., Ontology-based Design Knowledge Modeling for Product Retrieval, *ICED 05*, MELBOURNE, AUGUST 15-18, 2005 (paper accepted)