Exploiting Model Similarity for Indexing and Matching to a Large Model Database

Yi Tan, Bogdan C. Matei, and Harpreet Sawhney

Sarnoff Corporation,
201 Washington Road, CN5300, Princeton, NJ 08543-5300
{ytan, bmatei, hsawhney}@Sarnoff.com

Abstract. This paper proposes a novel method to exploit model similarity in model-based 3D object recognition. The scenario consists of a large 3D model database of vehicles, and rapid indexing and matching needs to be done without sequential model alignment. In this scenario, the competition amongst shape features from similar models may pose serious challenge to recognition. To solve the problem, we propose to use a metric to quantitatively measure model similarities. For each model, we use similarity measures to define a model-centric class (MCC), which contains a group of similar models and the pose transformations between the model and its class members. Similarity information embedded in a MCC is used to boost matching hypotheses generation so that the correct model gains more opportunities to be hypothesized and identified. The algorithm is implemented and extensively tested on 1100 real LADAR scans of vehicles with a model database containing over 360 models.

1 Introduction

1.1 Background

In a model-based 3D object recognition system, two model-related issues are challenging for the recognition performance: the number of models in the database and the degree of similarity amongst the models. In an indexing based recognition system that employs shape features for indexing, as the number of models increases, so does the number of the model features. As more features need to be searched, the recognition process may become inefficient. When a large number of similar models exist in a database, features from these models will compete with each other, the matching uncertainty may result in missing the correct target model. Numerous methods have been proposed to solve the first problem, such as the locality sensitivity hashing (LSH) techniques [2], which result in sublinear efficiency in feature search. There are, however, far fewer methods proposed to solve the model similarity problem in order to improve the recognition performance.

In this paper, we propose a new approach to tackle the model similarity issue in a model-based 3D object indexing system to improve the object indexing performance. Our indexing system takes the 3D object data obtained from a LADAR sensor, searches through the model database, and outputs a short list of models with high likelihood of matching to the scene target. Our application is vehicle recognition. A large

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part II, LNCS 3952, pp. 536 – 548, 2006.

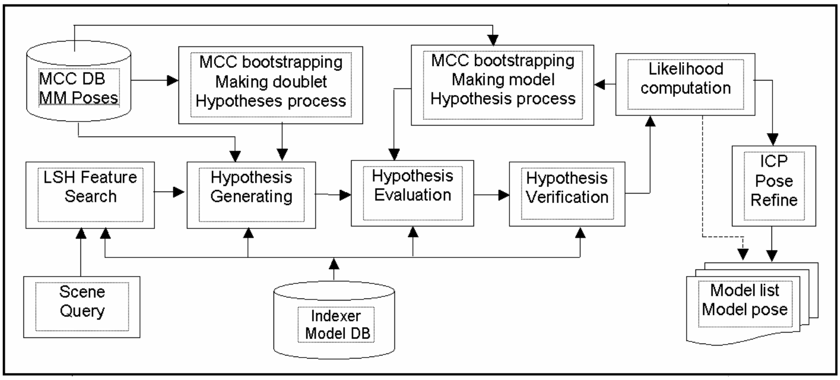


Fig. 1. The model-based 3D object indexing system

model database is built which contains several hundred commercial and military vehicles, of which about 2/3rd are sedans, pickup trucks, minivans and SUVs.

In our previous work [14], as shown in middle-row blocks in Figure 1, we have developed a method of coarse alignment and hypothesis testing by using linear model hashing and batch RANSAC for rapid and accurate model indexing. Facet models are converted into point clouds, and a set of shape signatures, the spin images [3], in the form of high-dimensional shape vectors is computed for each model and stored in the database. During the indexing process, the set of shape signatures is computed from a scene object and matched to the models features in the database. Features matched between the scene and models are used to generate indexing hypotheses. These hypotheses are verified by using the geometrical correspondence between the scene and model signatures, and the final matching likelihood is computed for each matched model. The indexer outputs a list of matched models with high likelihood value, as well as the model pose estimates.

1.2 Issues

As discussed above, the matched scene and model features are used to generate matching hypotheses. Based on indexing with shape signatures, for efficiency, the method tests for a limited number of pose hypotheses generated through the feature-pair of spin image matches. The method works well on diversified and mixed models. However, shape signatures tend to be alike if they are generated from the same location among the similar models. With the constraint of limited number of hypotheses, when the number of similar models increases in the model database, the best-matched model features may not come from the right model, but from models that are similar to the right one. This may result in the target model not being hypothesized and tested through the indexing process. The problem gets worse when large model database is indexed and many similar models present with quite severe ambiguities.

To alleviate the similar model indexing problem, in this paper we discuss a new approach to (1) measure the model similarity, (2) define the model-centric class to make use of model similarity, (3) use the model similarity to bootstrap model hypothesis to increase indexing and matching performance, and (4) use ICP for pose refining to improve likelihood computation. The new approach is shown in top-row blocks in Figure 1.

1.3 Related Work

Model similarity has been addressed in 2D/3D model retrieval applications, where on a given query of a model, the similar models are searched and retrieved from a domain specific database or from the Internet [4, 5]. The most common approaches for estimating model shape similarity is to establish the correspondence between two models and then to define the measure of similarity in terms of the mean-squared distance between pairs of points in correspondence [6]. 3D pose estimation between two 3D objects is well studied and a variety of methods were proposed [10, 11, 12, 13].

In recent years, instead of direct matching model raw points, a wide range of shape descriptors, in form of high-dimensional feature vectors, either global or semi-local, such as spin image [3] or shape context [7, 17], were introduced to represent 3D models. In shape descriptor or signature representations, model similarity measures become distance measurement between two sets of shape description vectors [2, 3, 7, 8, 9]. Point-based methods measure the local features; it is highly sensitive to precision of model alignment and to noise – if in the case the similarity is measured between a model and a scene object. Shape descriptors are, in general, more global or semi-local, and more immune to noise. The approach is generally invariant to viewing point transformations, which enables the computing the model similarity without using alignment. In [15] Sharp et al proposed the combination of shape and point-based methods for refining the ICP registration in the ICPIF algorithm.

Similarity measures between models were employed in [16] to define part based classes of 3D models, however the similarity was obtained using shape signatures only and the measures obtained were relative to the models used in the database. Thus, if several models were added to the database, the similarity between two models would change.

One of the approaches in dealing with similar models is to categorize models into classes and compose the model prototypes to represent the classes. Recognition process starts with matching on class prototypes, and then matching the individual models in the selected classes [18]. While the method sounds efficient, there are certain unresolved problems with such scheme. One is how to select the class prototypes such that they can well represent a model class? Bad prototyping will sure fail the recognition in the classification stage. The second is what if the right class prototype misses the matching at the first place? The continuing identification process inside classes using wrong prototypes will guarantee to generate wrong results.

Our work does not use class prototypes explicitly. However, the way the LSH process organizing the model features implies that the similar model features are grouped for scene-model match. Since such grouping is performed in model feature space, certain bad feature matches would not have fatal impact to the final recognition – because of large number of features used for each model (>1500).

1.4 Notation

In this paper, models are represented by a group of shape descriptor features. Each feature, $F=(s, x, n, m_{id})$, consists of four elements: $\boldsymbol{s}$ – the shape signature, e.g. a spin image defined w.r.t. a locally planar coordinate system; $\boldsymbol{x}$ – the 3D coordinate of the shape signature origin; $\boldsymbol{n}$ – the local surface normal, and $\boldsymbol{m_{id}}$ – the model ID. Match from scene object to model is represented by transformation $\boldsymbol{\Phi}(\boldsymbol{R}, \boldsymbol{T})$, where $\boldsymbol{R}$ is a 3x3 rotation matrix and $\boldsymbol{T}$ is 3x1 translation vector. We call a model the ***target model*** if the model is the ground truth for the scene object. The similarity between model $\boldsymbol{i}$ and $\boldsymbol{j}$ is represented by $\boldsymbol{S_{ij}}$, with the value in the range of [0, 1]; 0 means completely dissimilar, 1 means identical. $\boldsymbol{S_{ij}}$ also has the property of symmetry, i.e. $\boldsymbol{S_{ij}} = \boldsymbol{S_{ji}}$.

The remainder of this paper is organized as follows: In Section 2, we discuss how to quantitatively measure the model similarity, which provides the basis for using model similarity in 3D object indexing. We then propose a novel concept to define a model class for each model by using similarities, which is different from the conventional model-clustering methods. We then propose a new method to use our definition of model class to improve the matching hypothesizing process so that the target models gain more chances to be indexed. In Section 3, we discuss the models and database used in the experiments, and present the results from extensive testing on real LADAR vehicle data to show the performance improvement.

2 Our Approach

2.1 Quantitative Measure for Model Similarities

We use the distance measures between a pair of models to evaluate their similarity $\boldsymbol{S}$. Two sets of data are used for the measurement: the raw model point clouds and the shape signatures extracted from the sampled positions. Prior to the measurement, two models are aligned to obtain their relative pose by using the Iterative Closest Point (ICP) algorithm [6]. The pose alignment is initiated by feeding one of the models into the indexing system, which generates coarse alignment between this model and the one in the database. The similarity is calculated by a weighted sum of point-to-point distance and shape signature differences, as shown in Eq. 1, in which, $\boldsymbol{x}$ is a 3D point, $\boldsymbol{s}$ is the shape signature at point $\boldsymbol{x}$. The α's are used to weight the contribution from each component. The value of $\boldsymbol{S}$ is normalized between [0, 1]. Figure 2 shows the overlap a 1995 Toyota Tercel (white) aligned with a 1995 BMW 318i (black). The alignment and match measure show that the difference between the two is very small.

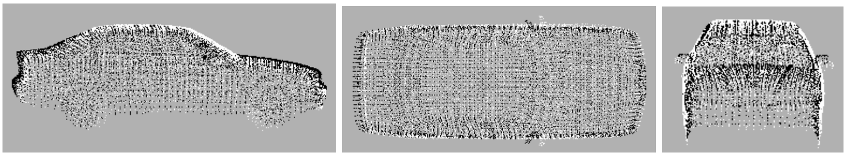


Fig. 2. The overlap of model 1995 Toyota Tercel (white pixels) with 1995 BMW 318i (black pixels) shows the small shape difference between the two vehicles

$$S(M_1, M_2) = \exp\left(-\alpha_1 \frac{1}{N_x} \sum_i \left\| \vec{x}_{1,i} - \vec{x}_2(\vec{x}_{1,i}) \right\|^2 - \alpha_2 \frac{1}{N_s} \sum_{\vec{x} \in S} \left\| s_1(\vec{x}) - s_2(\vec{x}) \right\|^2 \right). \qquad (1)$$

2.2 Model-Centric Class

For each model, we define a class, the model-centric class (MCC), to specify the association of a model with its class members. The MCC$_i$ for model m_i contains a group of models which have the highest similarities to model m_i. For a given database containing N models, the similarities S_{ij} between m_i and rest of $N\text{-}1$ models are calculated. Models satisfying the following criterion are defined as a member in MCC_i:

$$m_j \in MCC_i \ \ \textit{iff} \ \ S_{ij} \geq S_{threshold} \ (\mathrm{i}\neq\mathrm{j}) \ . \qquad (2)$$

MCC_i also includes the pose transformation Φ_{ij} between model m_i and m_j, which is obtained in similarity computation process.

MCC has the following properties: (1) a model can be a class member in multiple MCCs, (2) the number of class members can vary in different MCCs, and (3) if model m_j is a member of MCC$_\mathrm{i}$, then model m_i must be a member in MCC$_\mathrm{j}$.

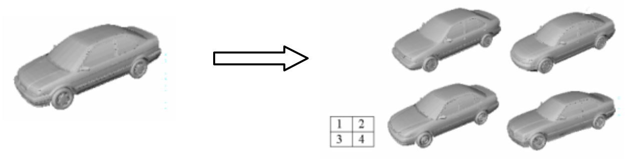


Fig. 3. 1995 Toyota Tercel (right) and a group of models with high similarity values: (1) 1996 Nissan Sentra-GXE s=0.79, (2) 1999 Volkswagon Passat s=0.76, (3) 1999 Toyota Corolla s=0.75, and (4) 1996 BMW 318i s=0.75

For instance, in Figure 3 if $S_{threshold}$ set to be 0.7, all four models as shown will be the members of MCC for the 1999 Toyota Tercel. By property #3, under the same similarity threshold, 1999 Toyota Tercel is also the member of MCCs of these 4 vehicle models.

Model similarity association within MCC provides a convenient way to bootstrap the matching hypothesis from a model to its class members without involving an expensive correspondence search and pose computation process.

2.3 Generate Matching Hypotheses

As mentioned in Introduction section, we use shape signatures, e.g. spin images, augmented with 3D positions to represent both models and scene objects. In the feature search process, $\boldsymbol{Q}$ best-matched model features are obtained for each of the scene object's features. The $\boldsymbol{Q}$ features can belong to $\boldsymbol{P}$ different models. A data-driven method, described in [14], is used to generate the scene-model matching hypotheses. In the method, a feature pair (doublet) is randomly generated from the scene data and the corresponding model doublet, sampled from the matched model feature list $\boldsymbol{Q}$, is

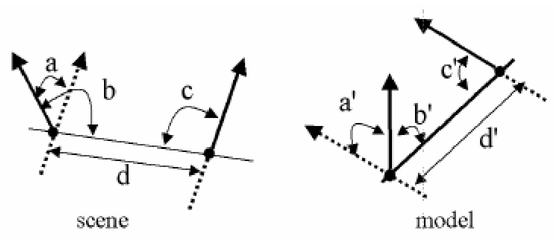


Fig. 4. Scene-model doublets are used to generate the matching hypothesis. Geometrical constraints are applied to ensure the goodness of matching.

selected. This doublet of features, if passes all the geometrical constraints, is used to generate a scene-model matching hypothesis, i.e. the ***R*** and ***T*** transformation, as shown in Figure 4.

For ***Q*** model features matched to each scene feature, the maximum number of doublets to be generated is Q^2. In our case, scene data usually contains 2000 features. If we match each scene feature to Q=100 model features, then, the potential matches to be generated could be $C^2_{2000}\times 100^2 \sim 10^{10}$, which is clearly impractical. To make the matching process efficient, scene features are checked to make sure they are salient before use; matched model features are sorted to make sure the best-matched ones are first used. A sequence of geometrical constraints are applied between the matched doublets, such as that $|d| > d_{min}$, $|d\text{-}d'| \leq \varepsilon$, $|a\text{-}a'| < \eta$, $|b\text{-}b'| < \eta$, and $|c\text{-}c'| < \eta$, to ensure choosing good hypotheses. After all checks have passed, it is checked if the best-matched model doublet belongs to the same model. Otherwise, the matching cannot be established.

At the end of the hypothesis generating process, we find that it may never guarantee to generate good hypotheses, if any, for the target model, largely due to (1) The features of target model never get in the Q list, (2) even though they get in Q list, there is a chance that they never get sampled, (3) even though they get sampled and used to form the matching doublet hypothesis, the constraint checks as discussed above may fail the match.

2.4 Use MCC to Enhance Matching Hypothesis

As discussed, we have the problem that when two best-matched model features in a doublet do not belong to the same model, no matching hypothesis will be generated. In such case, if one of the features belongs to the target model, then the target model may be missed in hypothesis generation process.

To solve this problem, we propose to use information in MCC to boost the hypothesis making process. It is observed that (1) the best set of model features matched to the scene largely comes from models that are similar to the scene object, and (2) most of these models are similar to the target model. The proposed idea is, if two best-matched features belong to two models that are similar to each other, we can use Φ in MCC to transform a feature from the location of one model to the corresponding location of the other, and then use the transformed features to generate hypothesis. This is illustrated in Figure 5. The method is stated as follows:

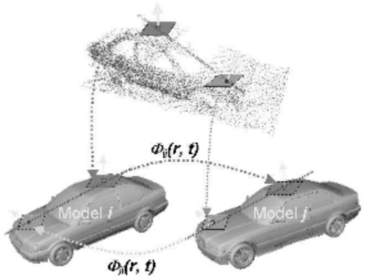


Fig. 5. Two matched features belong to two similar models, as indicated by solid-line patches. Features can be mapped to the same location from one model to the other by using MCC, as indicated by dashed-line patches. Thus for each model, a feature doublet, a solid-dashed pair, can be constructed.

For two best matched model features, $F_i(s_i, x_i, n_i, m_i)$ and $F_j(s_j, x_j, n_j, m_j)$ where $i{\neq}j$, if the following conditions are met:

$$\begin{aligned} & m_j \in MCC_i \text{ and } S_{ij} \geq S_{threshold}\,. \\ or \quad & m_i \in MCC_j \text{ and } S_{ji} \geq S_{threshold}\,. \end{aligned} \tag{3}$$

Then, the two new features, one for each model, can be generated:

$$\begin{aligned} & F'_i(s_i(\Phi_{ji}\,(x_j)),\ \Phi_{ji}\,(x_j),\ n(\Phi_{ji}\,(x_j)),\ m_i)\,. \\ & F'_j(s_j(\Phi_{ij}\,(x_i)),\ \Phi_{ij}\,(x_i),\ n(\Phi_{ij}\,(x_i)),\ m_j)\,. \end{aligned} \tag{4}$$

A hypothesis can be generated for each model by using two original best-matched features and two newly generated features:

For model i: $[F_i(s_i, x_i, n_i, m_i),\ F'_i(s_i(\Phi_{ji}\,(x_j)),\ \Phi_{ji}\,(x_j),\ \Phi_{ji}\,(n_j),\ m_i)]$.
For model j: $[F_j(s_j, x_j, n_j, m_j),\ F'_j(s_j(\Phi_{ij}\,(x_i)),\ \Phi_{ij}\,(x_i),\ \Phi_{ij}\,(n_i),\ m_j)]$.

These new hypotheses are added to the hypothesis list of the corresponding model and evaluated by feature alignment and matching.

2.5 Use MCC to Bootstrap Poses for Similar Models

The goal of the indexer is to produce a pruned short list of potential matching models. Indexer fails if the target model is not in the output list. To increase the chance of target model match, we use MCC to bootstrap new pose hypotheses for un-hypothesized models, evaluate the newly generated hypotheses, and increase the probability of target model detection. The idea is again based on two observations: (1) models in the indexer's output list are typically the good matches to the scene object; (2) the target model should ideally be in the list or at least be similar to some of the models in the list. The method is described as follows.

We begin by examining the top K (m_1, m_2, …, m_k) highest ranked models from the indexer output. For each of these models m_i, we look at its MCC_i and perform the following operation: for each model m_j in MCC_i, if $S_{ij} \geq S_{threshold}$, we generate a new matching hypothesis for model m_j:

$$\Phi_{sj}\,(m_j)= \Phi_{si} \bullet \Phi_{ij} \,. \tag{5}$$

Where Φ_{si} is the pose transformation from scene to model m_i, Φ_{ij} is the relative pose transformation from model m_i to its class model m_j, and Φ_{sj} is the new matching hypothesis generated for m_j. This idea is depicted in Figure 6.

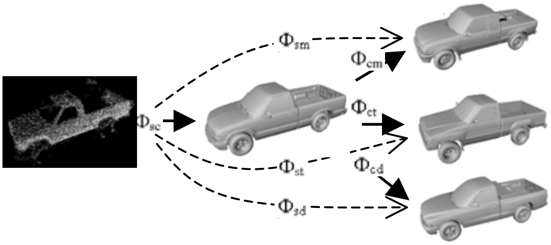


Fig. 6. Scene object (leftmost column) matches to model 1998 Chevrolet_S10 (mc) (middle column), which has a list of similar models in MCC: right top - 1999 Mazda B2500 (mm) , right middle - 1988 Toyota SR5 (mt), and right bottom - 1997 Dodge Dakota (md). Using scene-to-model and model-to-model transformations in MCC$_c$, new matching hypotheses are generated for the three similar models, indicated by the dashed arrows. The final best-matched model is 1988 Toyota SR5 (right middle), which is the true model.

The pose bootstrapping process generates a new set of matching hypotheses, which are evaluated through the verification process, and added to the previously generated pruned model list. The final indexer output is generated by ranking the likelihoods of the models in the expanded list.

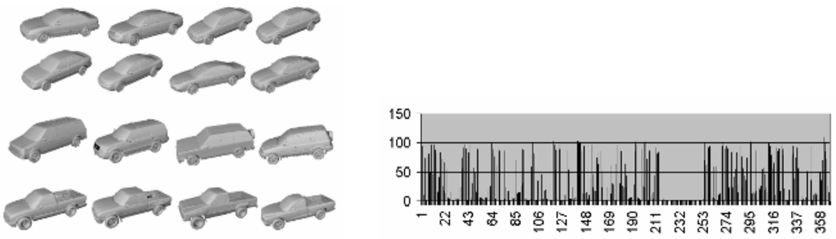
2.6 ICP for Scene-Model Pose Refinement

Point distance between aligned scene and model is used for likelihood computation. It is observed that the accuracy of the likelihood computation heavily depends on the fine pose alignment between the model and the scene. This is especially critical when many similar models exist in the hypothesis list; in such case, a slight misalignment may cause the target model to be ranked low in the candidate list. To ensure that the target model will prevail in competing with its similar rivals, the pose of scene-model alignment is refined by using ICP algorithm on both model and scene point cloud data. The final model candidates is sorted on the likelihood values and constrained by top K threshold.

3 Experimental Results

3.1 Model Database and Similarity Distribution

We generated a 3D model database containing 366 vehicle models, of which most are civilian vehicles, such as sedans, SUVs, pickups trucks etc. Samples of vehicle models in the model database are shown in Figure 7(a). Similar models are commonly seen in database.



(a) **(b)**

Fig. 7. (a) Sample models in model database; (b) Model similarity distribution on 366 model database (Sthreshold=0.5), and most models in the database are sedans

Figure 7(b) shows the similar model distribution for 366 models in the database with the similarity threshold, $S_{threshold}$ set at 0.5. Each model, on average, contains 35 similar models. There are over 127 models containing more than 50 similar models in their MCCs. These models are all sedans. On the other hand, most of construction or military vehicles, located between models 211 and 253 in Figure 7(b), have very few similar models since their geometrical shapes are unique.

Facet model data are processed to generate point clouds. Spin images are generated on evenly sampled locations around the model point clouds. Model features are built by combining the spin image, its 3D location, the local surface normal, and the model ID. A 366x366 model similarity matrix is computed. MCC for each model is extracted from the similarity matrix. Model database consists of nearly 1,000,000 shape features and 366 MCCs.

3.2 Real LADAR Scene Data and Pre-processing

We used three sets of LADAR data collected from high-lift on ground, airship, and helicopter platforms. About 250 real vehicles, both civilian and military, situated in the natural settings (urban, suburban) were scanned by Laser Terrain Mapper. Vehicles range from cars, SUVs, minivans, to trucks and construction utilities. More than 1000 volume of interest (VOIs) containing scene targets were extracted and used for testing. Note that typically each VOI contains only partial "views" of an object since only 2 or at most 3 sides of vehicles are scanned by the LADAR and that the point density is quite non-uniform. Scene data input to indexer are noisy with 5-10 cm standard deviation, contain ground plane and tree-like clutter, are sparse at times, may have articulations (doors, hood, trunk may open), and may be partially occluded.

Data from multiple views covering approximately 90° of viewing angle (2-3 sides) are registered to form the input scene object. Prior to feature computation, ground plane is removed through automated pre-processing. The input VOI in Figure 11 shows that with high degree of noise, fine features from vehicle data are lost. This increases the challenge for indexing process.

3.3 Experimental Results

We used data from all three collections and the 366-models database to test indexing system with and without using the new algorithm. For each input VOI query, the indexer outputs a variable length of ranked models (from 1 up to 25) as the best-matched model candidates. The probability of correct identification (P_{id}) is computed by comparison with the ground truth. We use the precision vs. the recall (ROC) curve to present the performance.

The first set testing is on 344 vehicles' VOI from ground high-lift data collection. The results are shown in Figure 8, where the triangle curve indicates the indexing performance with using the new algorithm; the diamond curve shows the performance without using the new algorithm. It shows that the new method increases the indexing performance on P_{id} by about 15%.

The second set testing is on 210 vehicles' VOI from airship data collection. The results are shown in Figure 9. Again, the triangle curve indicates the indexing performance with using the new algorithm; the diamond curve shows the performance without. It shows that the new algorithm increases the indexing performance by up to 20% on airship data.

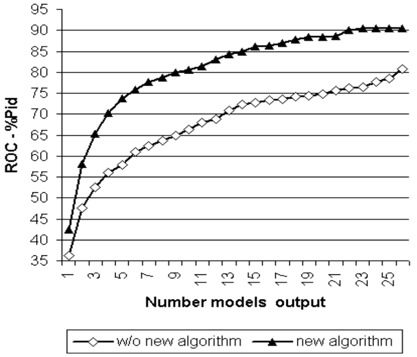


Fig. 8. Testing results on 344 highlift vehicles

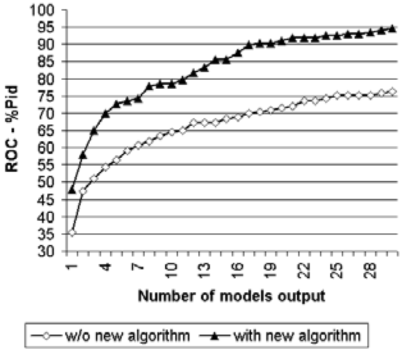


Fig. 9. Testing results on 210 airship vehicles

The last set of testing is on 548 vehicles' VOI from helicopter data collection. The results are shown in Figure 10 (a). Again, the triangle curve indicates the indexing performance by using the new algorithm; the diamond curve shows the performance without using the new algorithm. It shows that the new algorithm increases the indexing performance by up to 30% on helicopter data collection.

Overall, the indexing performance improvement with using new algorithm is significant on the large model database and large data collection sets. Indexer performs better on helicopter data collection is due to better data density in VOI, though the noise level (up to 10 cm) in this data set is larger than in other two data collections.

Together with ranked model output, indexer also outputs the pose alignment for each candidate model to the scene. Figure 10 (b) shows indexing results with scene-model poses; dark pixels are for scene data, light pixels are for matched models.

In general, for scene targets that have distinctive 3D shape, the target model will appear on top of the output list. In the case the 3D shape of input data can match to many models, the top K list will automatically expand to have the target model included. If the target model does not rank on top of the output list, the models ranked

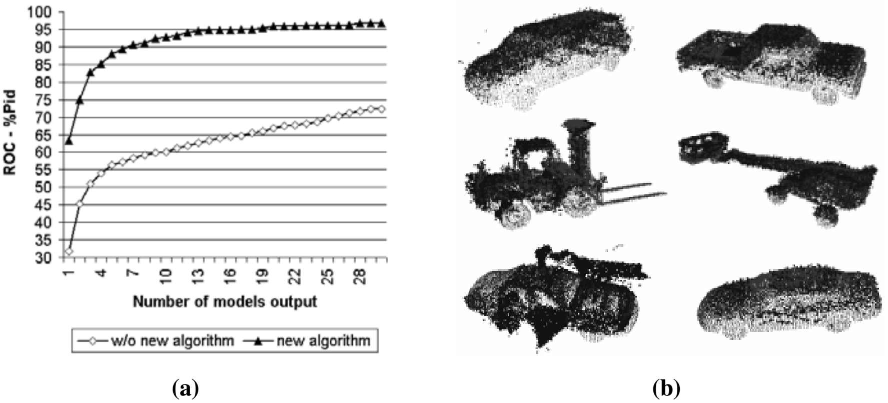


Fig. 10. (a) Results on 548 helicopter collection data; (b) Examples of scene-model pose alignment output from indexing process. In the image, dark pixels are for scene, light pixels are for matched models.

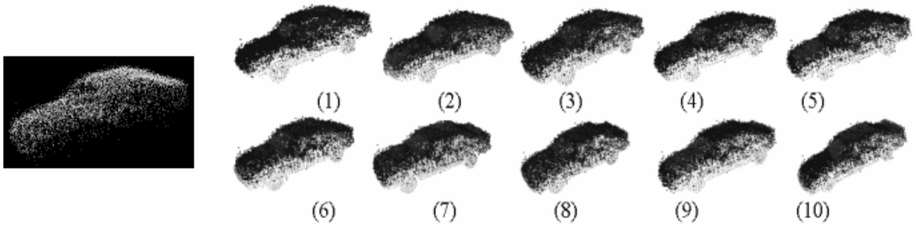


Fig. 11. Input query "1987 Honda Accord" and its top 10 best matched models: 1) 1994 Nissan_Sentra, 2) 1995 Oldsmobile_Cutlass-Ciera, 3) 1994 Ford_Tempo, 4) 1995 Geo_Prizm, 5) 1987 Ford_Escort, 6) 1992 Mazda_626, 7) 1984 Ford_Tempo, 8) 1987 Honda_Accord, 9) 1999 Dodge_Neon, and 10) 1991 Honda_Prelude-SI. High noise data and model similarity pushed the target model to rank 8. In the image, dark pixels are for scene, light pixels are for models.

above it are mostly very similar to the target model. Figure 11 shows an example of 1989 Honda Accord scene data and the top 10 best-matched models. The target model is ranked at 8, but the 7 models above it are very similar in shape and all match well with the scene.

To further distinguish among fine differences in indexed similar models needs to use model saliency features for fine verification. We discuss the issue in a separate paper [19].

The indexing runs on a PC with a 2.0 GHz CPU and 2 GB memory, on Windows and Linux OS. The entire process takes about 100 seconds on a 366 models database.

4 Conclusion

In this paper, we present a new method to solve the model similarity problem encountered in model-based 3D object recognition. We use a new metric to quantitatively measure model similarities on both shape signatures and 3D points. We use the similarity measures to define a model-centric class, the MCC, for each model. MCC contains a group of similar models and the pose transformation between the model and its class members. Similarity information embedded in MCC is used to boost matching hypotheses generation so that the target model gains more opportunity to be hypothesized and identified through the indexing process. The algorithm is implemented and extensively tested in a 3D object indexing system with a large model database containing 366 vehicle models, among which many similar models exist. Over 1000 real LADAR data from vehicle scans with noise up to 10cm standard deviation are used to test the new method. Our test results show that the target recognition performance improved by 15% to 30% in correct target identification with the new approach.

References

[1] Y. Lamdan and H. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, *Proc. 2nd Intl. Conf. on Comp. Vision*, pp. 238-249, 1988.

[2] A. Gionis, P. Indyk, and R. Motwani, Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB99)*, pages 518–529, 1999.

[3] Andrew E. Johnson and Martial Hebert, Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI),* 21(5):433–449, 1999.

[4] Johan W.H. Tangelder and Remco C. Veltkamp, A Survey of Content Based 3D Shape Retrieval Methods", *IEEE International Conf. on Shape Modeling and Applications*, 2004.

[5] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. *A search engine for 3d models*, ACM Transacions on Graphics, pp83-105, 2003.

[6] P. Besl, N. McKay, A method for registration of 3D shapes. *IEEE PAMI*, 14:239-256, 1992.

[7] S. Belongie and J. Malik, Matching with Shape Contexts. *IEEE Workshop on Content-based access of Image and Video-Libraries*, 2000.

[8] G. Mori, S. Belongie, and H. Malik. Shape contexts enable efficient retrieval of similar shapes. *Computer Vision and Pattern Recognition*, 1:723-730, 2001.

[9] R.C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Shape Modeling International*, pp188-197, May 2001.

[10] R. M. Haralick, H. Joo, C.N. Lee, X. Zhuang, V.G. Vaidya, & M.B. Kim, "Pose estimation from corresponding point data", *IEEE Trans. on Systems, Man, and Cybernetics.* Aug. 1989.

[11] R. Campbell and P. Flynn, A survey of free-form object represent and recognition techniques, *Computer Vision & Image Understanding*, 81(2):166-210, 2001.

[12] D.G. Lowe, "Object recognition from local scale-invariant features", in *International Conf. on Computer Vision*, (ICCV99) 1999, pp. 525-531.

[13] F. Stein and G. Medioni, "Structural indexing: Efficient three dimensional object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI),* vol. 14, no. 2, pp 125-145, 1992.

[14] Y. Shan, B. Matei, H.S. Sawhney, R. Kumar, D. Huber, M. Hebert, Linear model hashing and batch RANSAC for rapid and accurate object recognition, *IEEE Conf. on CVPR 2004*.

[15] Gregory C. Sharp, Sang W. Lee, & David K. Wehe, ICP Registration Using Invariant Features, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI),* 24(1):90-102, 2002.

[16] D. Huber, A. Kapuria, R. R. Donamukkala and M. Hebert, Part-based 3D object classification, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 04)*, June, 2004.

[17] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, Jitendra Malik: Recognizing Objects in Range Data Using Regional Point Descriptors. ECCV (3) 2004: 224-237.

[18] Ronen Basri, Recognition by Prototypes, International Journal of Computer Vision, 19(2): 147–168, 1996.

[19] B. Matei, H. Sawhney, and C. Spence, Identification of highly similar 3D objects using model saliency, *ECCV*, May 2006.