

# Linear Model Hashing and Batch RANSAC for Rapid and Accurate Object Recognition

Y. Shan B. Matei H. S. Sawhney R. Kumar

D. Huber M. Hebert

Sarnoff Corporation  
201 Washington Road  
Princeton, NJ 08540

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

{yshan, bmatei, hsawhney, rkumar}@sarnoff.com

dhuber@cs.cmu.edu, hebert@ri.cmu.edu

## Abstract

*This paper proposes a joint feature-based model indexing and geometric constraint based alignment pipeline for efficient and accurate recognition of 3D objects from a large model database. Traditional approaches either first prune the model database using indexing without geometric alignment or directly perform recognition based alignment. The indexing based pruning methods without geometric constraints can miss the correct models under imperfections such as noise, clutter and obscurations. Alignment based verification methods have to linearly verify each model in the database and hence do not scale up.*

*The proposed techniques use spin images as semi-local shape descriptors and Locality-Sensitive Hashing (LSH) to index into a joint spin image database for all the models. The indexed models represented in the pruned set are further pruned using progressively complex geometric constraints. A simple geometric configuration of multiple spin images, for instance a doublet, is first used to check for geometric consistency. Subsequently, full Euclidean geometric constraints are applied using RANSAC-based techniques on the pruned spin images and the models to verify specific object identity. As a result, the combined indexing and geometric alignment based pipeline is able to focus on matching the most promising models, and generate far less pose hypotheses while maintaining the same level of performance as the sequential alignment based recognition. Furthermore, compared to geometric indexing techniques like Geometric Hashing, the construction time and storage complexity for the proposed technique remains linear in the number of features rather than higher order polynomial. Experiments on a 56 3D model database show promising results.*

## 1 Introduction

The task of object recognition involves the solution of several complicated problems, i.e., (i) the unknown pose between the query and the model, (ii) the query-model discrepancy due to occlusion and clutter in the scene, and (iii) the computational cost of comparing each individual model from the database to match against the query [1]. Alignment based verification techniques (often based on RANSAC) have been used to address the first two problems. However, existing alignment based techniques [2, 3] apply RANSAC sequentially to each individual model from the database, and hence do not address the computational issues related to the third problem when matching to a large model database.

This has led to a two pronged approach of first indexing into the model set and estimate a viable and small collection of model hypotheses, and subsequent verification of the correct model using geometric alignment and matching. The model pruning step ensures that the computational complexity does not grow linearly with the size of the model database and alignment based verification ensures accuracy at the cost of linear complexity hopefully for a small fraction of the models. When scenes involve noise, clutter and occlusions, the challenge for model pruning is to ensure that a substantial number of models can be rejected while ensuring that the correct model remains in the short list. The challenge for alignment based verification is to rapidly prune the high dimensional space of pose hypotheses for efficient but accurate verification of the correct model.

This paper proposes a joint feature-based model indexing and geometric constraint based alignment pipeline for efficient and accurate recognition. The domain used for application of the proposed recognition techniques is recognition of 3D objects from a large model database from 3D range images. The proposed technique employs high-dimensional semi-local distinctive features, like spin images [2], to index into a joint model database of all the

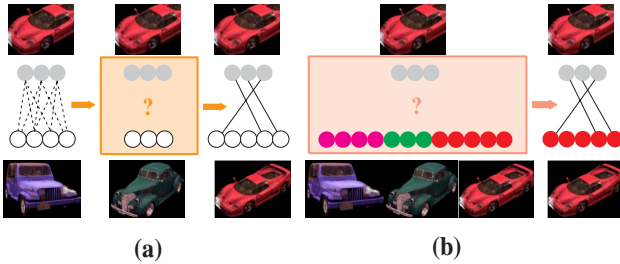


Figure 1: Sequential RANSAC (a) vs. Batch RANSAC (b) for object recognition. Each picks the model with the most consensus features as the match of the query (shown in the top row). Pictures represent 3D objects, circles represent object features, the left rectangle represents the sequential RANSAC, the right represents the batch RANSAC, dashed lines represent possible matches, and solid lines represent solved matches. Sequential RANSAC (acting like a sliding window in (a)) matches query object with one model at a time, spending almost the same amount of computational effort on each model even it is very different from the query. The proposed batch RANSAC is able to focus on the matching of the most promising models, and generate far less pose hypotheses while maintaining the same level of performance as the sequential RANSAC. The batch RANSAC uses an aggregated model database where a model id is hashed with each feature (color coded in (b)). This is similar to Geometrical Hashing. The major difference is that the model construction time and storage space of our approach is linear with the number of features in the database.

models. Subsequently, progressively complex geometric alignment constraints are employed using RANSAC like methods to rapidly prune and verify the most likely model hypotheses. Instead of matching models sequentially or pruning the models without alignment constraints, the key idea of this paper is to generate pose hypotheses for only those models with high probability of being the match, and vote for these models simultaneously with robust match measures. By employing approximate high dimensional nearest-neighbor search techniques like Locality-Sensitive Hashing (LSH) and RANSAC based geometric model verification, we avoid the problem of pre-mature model pruning while maintaining the accuracy of alignment based verification methods. Figure 1 demonstrates the difference between the traditional sequential model verification and the proposed simultaneous multi-model RANSAC-based verification and recognition.

## 2 Related work

The proposed method belongs to the class of appearance feature and alignment based approaches to object recogni-

tion. Since our domain is 3D models and range images and 3D point clouds, “appearance” features are computed not based on intensities or color but on semi-local 3D configuration of points in the scenes and models. Under the feature-based framework, both the model and the query objects are represented as a collection of shape features or signatures, for instance spin images [2]. The shape features represent appearance as a high-dimensional feature and have an associated local coordinate system, a 3D location on the object and a local normal. Therefore, each feature encapsulates the local distribution of 3D points within its scope and also is in a coordinate geometric relation with other features on the object. Recognition algorithms typically vary according to their treatments of the feature attributes and the feature configurations.

Geometric hashing [4, 5] and its variants proposed object recognition using high dimensional representations that combined (quasi-)invariant coordinate representations with geometric coordinate hashing to prune a model database while employing geometric constraints. However, the time and space complexity of creating geometric hash tables is polynomial in the number of feature points associated with each model. Furthermore, since the (quasi-)invariant coordinate representations are relatively low-dimensional (typically 2 or 3), the hash tables can become crowded even with small model databases and the run time complexity can deteriorate to a linear complexity that again does not scale with the size of the database. Variants of geometric hashing that employ higher dimensional features have been proposed. This paper can be considered as an improvement over those variants.

RANSAC[6] (Random Sample Consensus) is an effective data-driven alignment and verification technique. Instead of generating pose hypotheses exhaustively, RANSAC generates only a limited number of hypotheses with a sampling process guided by the matching of feature attributes. For one model, the storage complexity of RANSAC is  $O(n)$ , where  $n$  is the number of features in the model. This is a significant gain compared with Geometric Hashing, of which the complexity is  $O(n^{c+1})$ . The constant  $c$  is related to the degrees of freedom of the pose, and is usually larger than 1. DARCES [7] and preemptive RANSAC [8] are recent advances over the original RANSAC technique. The former makes it more robust, and the latter makes it faster. RANSAC has been used for alignment based verification for object recognition [2, 3]. The RANSAC class of techniques iterate through all the models in the database, and therefore are linear in the number of models in the database.

Generation of potentially good quality pose hypotheses can be aided by the use of high-dimensional appearance features by employing approximate nearest neighbor search algorithms like the recently developed algorithm

called Locality-Sensitive Hashing (LSH) [9], LSH is a probabilistic method for approximate nearest neighbor search, and achieves sub-linear complexity in the number of features in the database. See [10] for more theoretical and practical details.

The proposed method uses both high-dimensional feature attributes and global geometric configurations for recognition. In [11], 2D shape signatures called shape contexts are used to find a short list of candidate models. A matching algorithm using global geometric constraints is then applied to pick the best match. While efficient, this approach has the risk of committing to a short list prematurely and miss the correct match. Other related methods include [12],[13], and [14]. The paper [12] uses Hough transformation to find the pose, which requires searching peaks in a high dimensional array and the knowledge of the ranges of the pose parameters. The paper [13] uses conventional hash table to store the model database, but uses a cluster-based approach to search for the consistent poses in the pose parameter space. The paper [14] uses low-distortion graph embedding to map vertex-labeled graphs to a set of vectors in a low-dimensional space, and solves the matching problem with the Earth Movers' Distance (EMD). The vertex-labeled graph encodes both the feature attribute in the nodes and the global configuration in the edges. Though promising, it is not clear whether the mapping is stable in the presence of high percentage of outliers in the query.

### 3 Notations

For the query object, a set of spin images  $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_s\}$  and a set of normals to the surface  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_s\}$  are computed at basis points  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_s\}$  uniformly sampled along the surface of the object, where  $\mathbf{b}_k \in \mathcal{R}^d$ ,  $\mathbf{p}_k, \mathbf{q}_k \in \mathcal{R}^3$ ,  $d$  is the number of bins in each spin image, and  $s$  is the number of spin images in the query. Let  $\alpha^k = (\mathbf{b}_k, \mathbf{p}_k, \mathbf{q}_k)$ ,  $k = 1, \dots, s$ , represent an augmented spin image that includes its local coordinate system, and  $Q$  be the set of all augmented spin images for the query. Similarly, let  $\beta_i^k = (\mathbf{b}_i^k, \mathbf{p}_i^k, \mathbf{q}_i^k, i)$ ,  $k = 1, \dots, t_i$  be an entry in the model database, where  $t_i$  is the number of spin images in the  $i$ th model. Obviously, each entry is an augmented model spin image hashed with the corresponding model ID. We will generally refer to  $\alpha$  and  $\beta$  as *features* in the following discussion. Let  $m$  denote the number of models in the database. Let  $\Phi_i$  be the set of hypothesized poses from the  $i$ th model to the query, and  $n_i$  the number of pose hypotheses in the set. Each pose is a 3D rigid motion, and hence  $\Phi_i^j \in \mathcal{R}^6$ ,  $j = 1, \dots, n_i$ . Suppose that the likelihood of a pose given the query is define as  $p(Q | \Phi_i^j)$ , the problem is then to find the model label  $i^* = 1, \dots, m$  that maximizes

the likelihood among the poses of all models in the database

$$i^* = \arg \max_{i=1, \dots, m} \max_{j=1, \dots, n_i} p(Q | \Phi_i^j) \quad (1)$$

For 3D rigid motion, pose hypothesis  $\Phi_i^j$  can be generate from a corresponding pair of query features and model features whose model IDs are the same.

## 4 Algorithm overview

Solving (1) with pose hypotheses generated from all possible pairs of query features and model features is prohibitively expensive. On the other hand, it is also important to ensure the accuracy if only a small portion of these hypotheses are used. Our approach uses three steps, i.e., spin image-based pruning, doublet-based pruning, and RANSAC-base verification, to alleviate these problems. Spin image-based feature pruning uses one query feature at a time to prune the model features whose spin images are not similar in the parameter space. With the help of LSH, this step can be proceeded very efficiently without touching all the model features in the database. Doublet-based pruning uses a pair of query features at a time to prune the model-pose hypotheses that are inconsistent with the double constraint. The doublet constraint is a powerful constraint that uses both distance and surface normal to check the consistency between the query feature pair and the model feature pair. After the first two steps, the remaining pose hypotheses are verified by warping all the features of the hypothesized model to the query features with the hypothesized pose. After warping, the likelihood of the hypothesized model being matched with the query is computed based on the consensus features or the inliers. The maximum likelihood is recorded for each hypothesized model, and the model with the highest maximum likelihood is selected as the match out of all the hypothesized models.

## 5 Spin image-based feature pruning with LSH

Given a single query feature  $\alpha$ , we want to find the model features that are the nearest in spin image space. Searching a large model database for the nearest neighbors in a high dimensional space is extremely time consuming. Locality-sensitive hashing (LSH) is the state-of-art technique to solve this problem. LSH is a probabilistic solution for the approximate nearest neighbor problem. The unique property of LSH is that it relates the probability of collision to the  $L_1$  distance between two vectors. In other words, if two vectors are close in distance, they will have high probability of landing in the same bucket of the hash table. The problem

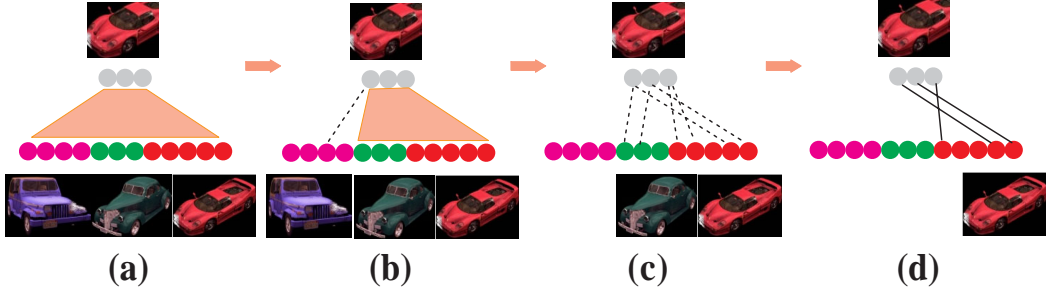


Figure 2: An illustration of the algorithm. Pink cones and dashed lines represent possible matches. Solid lines represent solved matches. (a) Initially any query feature can be matched with any model features. (b) After spin image-based pruning, there are very few possible matches remaining for the left most vehicle because other vehicles’ spin images are more similar to the query spin images. As a result, only few pose hypotheses will be generated for this model. This stage involves only one query feature at a time. (c) After doublet-based pruning, possible matches between the query features and the possible models are dramatically reduced. The left most vehicle does not have any possible matches, and is no longer considered for matching beyond this point. This stage involves a pair of query features at a time. (d) After RANSAC-based verification, the right model is picked as the match because it has the most consensus points/inliers. This stage involves all or a percentage of all query features at a time.

of finding the nearest neighbors then boils down to searching only the vectors in the bucket that have the same hash code as the query vector. The probability of collision as the function of the  $L_1$  distance has the following form

$$P_c = 1 - (1 - (1 - d/d_c)^K)^L, \quad (2)$$

where  $d_c$  is a constant related to the maximum distance between any two vectors in the set under the consideration,  $d$  is the actual distance between two vectors,  $K$  is the number of bits used to sample the vectors in the Hamming space, and  $L$  is the number of hash tables. Figure 3 plots the curves of the function in (2) with different  $K$  and  $L$ . Intuitively, increasing  $K$  reduces the probability of collision, and increasing  $L$  increases the probability. It can also be seen that the probability drops down quickly as the distance increases. In our case, this prevents the matching of features from model objects that are not similar, and is one of the key factors contributing to the efficient pruning demonstrated in Fig.2(b). See[10] for a practical way of tuning  $K$  and  $L$  to maximize the performance of LSH.

The downside of LSH is that some good matches may be missed. In our approach, this can be tolerated because of the robustness of the RANSAC-based verification stage.

## 6 Feature saliency

Feature saliency is another useful information to guide the generation of pose hypotheses. In Fig.4, the  $x$ -axis of each histogram is the pose hypothesis, while the  $y$ -axis is the number of features of a particular model that is in the top  $W$  list  $C^l$  of a query feature  $\alpha^l$ . If a query feature is salient, the

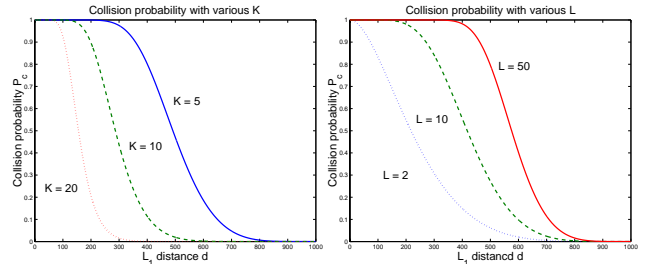


Figure 3: LSH probability of collision with various  $K$  and  $L$ . Left: various  $K$  with fixed  $L = 20$ , Right: various  $L$  with fixed  $K = 5$ .  $d_c$  is set to 1000.

histogram should be peaky. One example is the red point in the front of the vehicle, whose corresponding saliency histogram is the displayed in the right most image. On the contrary, the feature point on the top of the vehicle is not salient, and its histogram looks flat. The saliency measure is defined as the entropy of the saliency histogram. Similar measures were proposed in [15, 16] in the context of shape-based histogram matching. In the recognition algorithm described in Sec.9, sampling of the scene features can then be concentrated on those salient features to reduce the number of hypotheses.

## 7 Doublet-based hypothesis generation and pruning

From a pair of query features  $(\alpha^{l_1}, \alpha^{l_2})$ , and a corresponding pair of model features  $(\beta_i^{k_1}, \beta_i^{k_2})$ , we can generate a

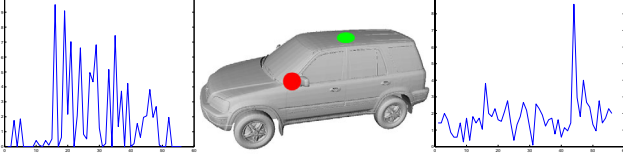


Figure 4: Feature saliency is another kind of information that can be used to improve the performance of the proposed method.

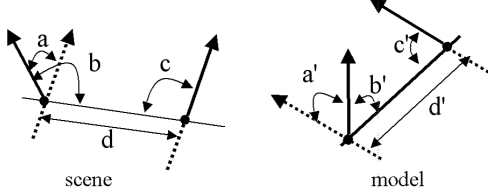


Figure 5: Two sets of correspondences in addition to have similar surface signatures, are required to be geometrically consistent:  $d \approx d'$ ,  $a \approx a'$ ,  $b \approx b'$ ,  $c \approx c'$ . These constraints are applied sequentially to prune pose hypotheses.

pose hypothesis  $\Phi_i$  for the  $i$ th model. A good hypothesis should be geometrically consistent, that is, if we warp  $(\beta_i^{k_1}, \beta_i^{k_2})$  to  $(\alpha^{l_1}, \alpha^{l_2})$  according to the hypothesized pose, both the locations and the normal directions should be similar between the corresponding features. Since the actual pose computation involves SVD decomposition of a  $3 \times 3$  matrix and some other expensive operations[17], we designed a cascaded filter using the four doublet geometric constrains defined in Fig.5. Given a pair of correspondences, we apply these four constrains sequentially, and compute the real pose only if it passes all of them. In this way, we can throw away bad hypotheses early on without spending time computing an actual pose.

In order to compute stable pose from corresponding query-model pairs that pass the consistency check, we also require that

$$\|\mathbf{p}^{l_1} - \mathbf{p}^{l_2}\| \geq d_{min}, \text{ and } \arccos(\mathbf{q}^{l_1} \cdot \mathbf{q}^{l_2}) \geq \theta_{min}. \quad (3)$$

We call these stability constraints.

## 8 Likelihood computation

Once a pose hypothesis  $\Phi_i$  is generated for a hypothesized model  $i$ , we can warp all the features of the model into the query coordinate system. For each query feature  $\alpha^l$ , we then search for a warped model feature  $\beta_i^{k_l}$  that maximizes the likelihood

$$p(\alpha^l, \beta_i^k | \Phi_i) = p_o(\alpha^l, \beta_i^k | \Phi_i) p_n(\alpha^l, \beta_i^k | \Phi_i) p_s(\alpha^l, \beta_i^k), \quad (4)$$

where  $p_s(\alpha^l, \beta_i^k)$  measures the similarity between the spin images  $\mathbf{b}^l$  and  $\mathbf{b}_i^k$ ,  $p_o(\alpha^l, \beta_i^k | \Phi_i)$  is the probability of matching the origins given the pose

$$p_o \propto \begin{cases} \exp\left[-\frac{(\mathbf{p}^l - \mathbf{p}_i^k)^2}{2\sigma^2}\right], & \text{if } \|\mathbf{p}^l - \mathbf{p}_i^k\| \leq \Delta \\ \exp\left(-\frac{\Delta^2}{2\sigma^2}\right), & \text{if } \Delta < \|\mathbf{p}^l - \mathbf{p}_i^k\| \leq \Delta_{\max} \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

and  $p_n(\alpha^l, \beta_i^k | \Phi_i)$  is the probability of matching the normals given the pose

$$p_n = \begin{cases} 1, & \text{if } \arccos(\mathbf{q}_i^{k_l} \cdot \mathbf{q}^l) \leq \eta_{\max} \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

In (5)  $\Delta$  is a threshold suitably chosen to separate inliers from outliers,  $\Delta_{\max}$  is the maximum spread of the outliers. We have assumed that the distribution of the inliers is Gaussian, and having noise standard deviation  $\sigma$ . In (6)  $\eta_{\max}$  is the maximum angle error.

The likelihood of the pose given the query can then be computed assuming that the query features are independent

$$p(Q | \Phi_i) = \prod_{l=1}^s p(\alpha^l, \beta_i^{k_l} | \Phi_i). \quad (7)$$

Note in (7) that  $\beta_i^{k_l}$  is the warped model feature that maximizes the likelihood in (4). The query feature  $\alpha^l$  is inlier if  $p(\alpha^l, \beta_i^{k_l} | \Phi_i)$  is large.

## 9 Recognition algorithm

Given a set of scene features, a model database, and an LSH table generated from the model database, the batch RANSAC recognition algorithm is given next.

**Initialization** Let  $\gamma_i = 0, i = 1, \dots, m$ , where  $\gamma_i$  is the maximum likelihood for the  $i$ th model.

**Spin image-based pruning** For each query feature  $\alpha^l$  that is salient, find the set  $\mathcal{C}^l$  of at most  $W$  nearest neighbors in the database with LSH. Let the size of each candidate set to be  $W_l \leq W$ . Saliency is define in Sec. 6.

**Doublet-based pruning** Pick up randomly two indexes  $1 \leq k, l \leq s, k \neq l$  such that  $\alpha^k$  and  $\alpha^l$  satisfy the stability constrains in (3). [a] Pick up  $u, v$  such that  $1 \leq u \leq W_k, 1 \leq v \leq W_l$  using either importance sampling, or by exhaustively enumerating all  $W_k W_l$  possibilities. Pick up the candidate model feature  $\beta_{i_u}^{w_u}$  from  $\mathcal{C}^k$ , and  $\beta_{i_v}^{w_v}$  from  $\mathcal{C}^l$ . If  $i_u \neq i_v$ , go to [a]. Otherwise denote  $i = i_u = i_v$ , and check the geometric consistency of  $(\alpha^k, \alpha^l)$  and  $(\beta_i^{w_u}, \beta_i^{w_v})$  as in Fig.5. If

it is not consistent, go to [a]. Otherwise, compute and generate a hypothesized pose  $\Phi_i$  and move on to the next step.

**RANSAC-based verification** Compute the likelihood  $\gamma = p(Q | \Phi_i)$  as described in Sec.8 (7). If  $\gamma > \gamma_i$ ,  $\gamma_i = \gamma$ . If need to pick up next  $u$  and  $v$ , go to [a]. Otherwise, if the maximum trial number is reached, go to [Output]. Otherwise, go to [Doublet-based pruning]

**Output**  $i^* = \arg \max_{i=1, \dots, m} \gamma_i$ .  $i^*$  is the solution of (1).

Given the above algorithm description, we can now explain why the batch RANSAC can generate far less hypotheses while still maintaining the accuracy. The key is the feature competition from the features of all the models in the database as illustrated in Fig.6.

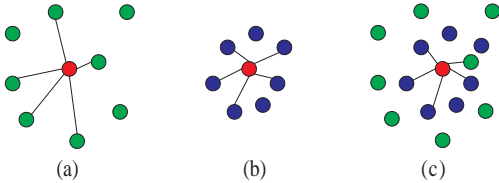


Figure 6: The key to the success of the batch RANSAC algorithm is the feature competition from all models. Suppose we select the maximum number of candidates  $W = 5$ . For a scene feature (the red circle) to be matched with features of each individual model separately, each will have 5 nearest candidates as shown in (a) and (b). However, if the matching is applied to the features of all models simultaneously, the scene feature will be matched with five candidates on the model whose features are more similar than others. These features usually come from the models whose shapes are similar to the scene. In the case shown in (c), four candidates will be selected from the blue model (the correct model), and only one will be from the green model. In this way, the batch RANSAC can generate far less hypothesis while still maintaining the same level of performance as the sequential RANSAC.

## 10 Experiments

We have tested our method with a model set of 56 vehicles on a 2GHz PC with 2GB main memory. We will present comparative results with the sequential RANSAC-based method. We will also show results of a spin image-based indexing approach which is similar to the one proposed in [11]. This method performs model indexing without using the geometric constraint, and is usually used to provide a short list for more sophisticated matching algorithm. The

comparison is based on the best performance of each individual method. In the following discussion, we will denote our approach as BRAN, the sequential RANSAC-based approach SRAN, and the spin image-based indexing approach as SIND.

### 10.1 Synthesized data



Figure 7: Examples of vehicle models used for the test. To simulate real sensor data, these facet models are converted into point clouds. Model database is constructed based on the features computed from the converted model point clouds.

Fig. 7 shows the facet models of the original objects used for the test. A laser range sensor simulator is used to convert facet models into range images. View points are sampled from a hemisphere centered around the object, and the viewing direction is always targeting the center. The spherical coordinate for each view point is denoted as  $(r, \phi, \theta)$ , where  $r$  is the radius,  $\phi$  is the azimuthal angle, and  $\theta$  is the polar angle. The radius is set to be a constant such that the object occupies the full view of the simulated laser sensor for most of the times. It is therefore ignored in the view point notation hereafter. Each model object is generated by combining 8 pieces of point clouds sampled from the view point set of  $\{(0, 45^\circ), (45^\circ, 45^\circ), \dots, (360^\circ, 45^\circ)\}$  that covers the whole object. By constructing an octree-like spatial structure from the combined point cloud of an object  $M_i$ , a set of basis points is uniformly selected from the (implicit) object surface, and the corresponding set of spin images is computed. Query objects are generated by combining only three views in order to simulate the real sensor data.

Figure 9 compares the accuracy of the three approaches. The  $x$ -axis of the curve is the number of models picked as the candidates, and the  $y$ -axis is the probability that the right model is in the short list. See figure caption for the setting of this experiment. It can be seen that BRAN and SRAN are on the same level of performance, while SIND is far off. This demonstrates the power of RANSAC-based approaches, and also suggests that it may not be a good idea to generate short list without using geometric constraints.

Figure 10 compares the speed of BRAN and SRAN with various database sizes. It can be seen that BRAN is always



Figure 8: Query object from different viewing angles. Each query combines three views of point clouds. Gaussian noise is added to each view along the viewing direction before these views are combined. The noise level in this example is 10 cm.

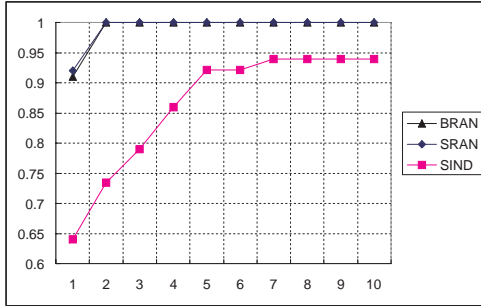


Figure 9: Experiments on the accuracy of three approaches based on 56 queries. The noise level in this experiment is 10 cm. Spin images are computed with 1 meter radius, and  $10 \times 10$  bins. For BRAN, we set the maximum number of model feature candidates  $W$  to be 50. Our experiment shows that increasing  $W$  to 100 does not improve the accuracy.  $W$  for SRAN and SIND are 5, and 1, respectively. Again, increasing these numbers does not lead to significant improvement in the accuracy.

faster than SRAN. More importantly, SRAN is clearly linear to the number of models, while BRAN is sub-linear and almost constant in the example.

## 10.2 Real data

The real data was collected using an airborne scanning Lidar sensor with fairly accurate GPS and INS systems on-board. A variety of vehicles, including many vehicle model types that look similar, were placed in open and partially obscured situations and the vehicles were scanned from the air. All the data consisted of vehicles that were scanned from at most 3 sides. Typically vehicle scans consist of 2 sides with a few hits from the top. Note that obscuration is counted not with respect to the complete vehicle but with respect to 2-3 sides of the vehicle considered as 0% obscuration. Figure 11 shows some examples of the 88 real queries used in the experiment. Figure 12 shows the performance of the BRAN and SIND algorithms when matching these 88 real queries against a 90 model database. It can be seen that SIND failed

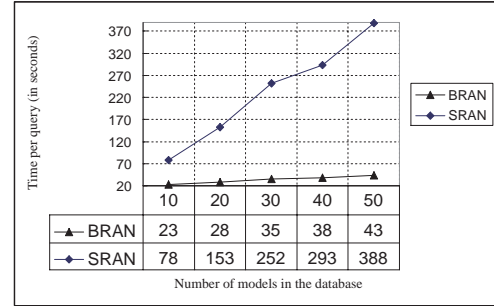


Figure 10: Recognition time per query with various number of models in the database. The sub-linear behavior of BRAN is obvious.

while BRAN reached 97% with top 20 candidates. SRAN was also applied on the same set of data and produced similar accuracy as BRAN, but with much higher computational cost.

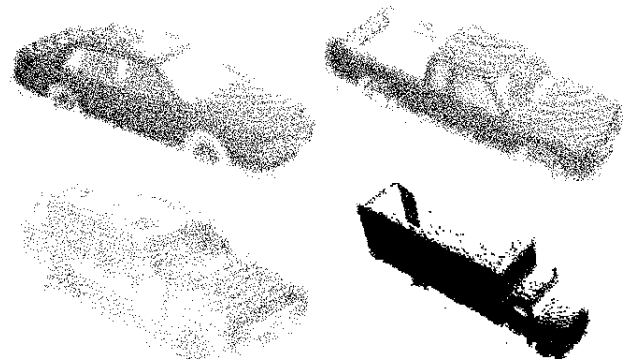


Figure 11: Four examples of the 88 real queries with obscuration in the range of 0 – 25%. obscuration is defined not with respect to the complete vehicle but 2 – 3 visible sides of the vehicle.

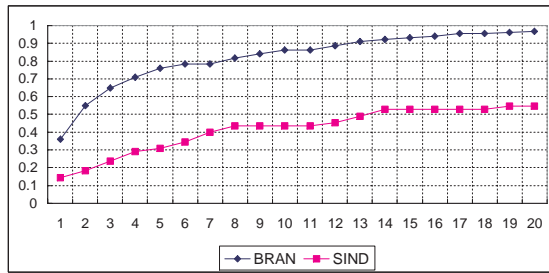


Figure 12: Accuracy of the Batch RANSAC and the SIND algorithms when matching the 88 real queries against a 90 model database.

## 11 Conclusion and future work

We propose a batch RANSAC algorithm for rapid and robust object recognition. As compared with sequential RANSAC-based recognition approaches, our method is capable of focusing pose hypothesis generation and verification on the models that have high probabilities of being the match. It can therefore use far less hypotheses to achieve the same level of accuracy of the sequential RANSAC-based approaches. Moreover, as compared with some previous work such as Geometric Hashing, the construction time for model hashing and the storage space for the resulting model database required by our approach is linear to the number of features in the database. Experiments on a 56 model database confirms the efficacy of the proposed method.

This work does not consider the recognition of articulated objects, while the proposed framework can indeed tolerate some degrees of articulations. An interesting future work is to design a batch RANSAC recognition algorithm that deals with fully articulated objects.

## Acknowledgements

This work was supported in part by DARPA/AFRL Contract No. F33615-02-C-1265. The 3D models and scenes were provided by AFRL as a part of the contract.

## References

- [1] Haim J. Wolfson and Isidore Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science & Engineering*, 4(4):10–21, 1997.
- [2] 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.
- [3] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR03)*, 2003.
- [4] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *International Conference on Computer Vision (ICCV88)*, pages 238–249, 1988.
- [5] I. Rigoutsos and R. Hummel. A bayesian approach to model matching with geometric hashing. *Computer Vision and Image Understanding*, 61(7):11–26, 1995.
- [6] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981.
- [7] C. Chen, Y. Hung, and J. Cheng. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 21:1229–1235, 1999.
- [8] David Nistr. Preemptive ransac for live structure and motion estimation. In *International Conference on Computer Vision (ICCV03)*, pages 199–206, 2003.
- [9] 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.
- [10] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. Mean shift based clustering in high dimensions: A texture classification example. In *International Conference on Computer Vision (ICCV03)*, pages 456–463, 2003.
- [11] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR01)*, pages 723–730, 2001.
- [12] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV99)*, pages 525–531, 1999.
- [13] F. Stein and G.G. Medioni. Structural hashing: Efficient three dimensional object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):125–145, 1992.
- [14] Yakov Keselman, Ali Shokoufandeh, M. Fatih Demirci, and Sven Dickinson. Many-to-many graph matching via metric embedding. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR03)*, 2003.
- [15] Ying Shan, Harpreet S. Sawhney, and Art Pope. Measuring the similarity of two image sequences. In *Asia Conference on Computer Vision, (ACCV04)*, 2004.
- [16] Ying Shan, Bogdan Matei, Harpreet S. Sawhney, Rakesh Kumar, Daniel Huber, and Martial Hebert. Partial object matching with shapeme histograms. In *European Conference on Computer Vision (ECCV04)*, 2004.
- [17] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3D point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 9:698–700, 1987.