Partial Object Matching with Shapeme Histograms

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

Computer Vision Laboratory
Sarnoff Corporation
201 Washington Road, Princeton, NJ 08540
{yshan, hsawhney, bmatei, rkumar}@sarnoff.com

Abstract. Histogram of shape signature or prototypical shapes, called shapemes, have been used effectively in previous work for 2D/3D shape matching & recognition. We extend the idea of shapeme histogram to recognize partially observed query objects from a database of complete model objects. We propose to represent each model object as a collection of shapeme histograms, and match the query histogram to this representation in two steps: (i) compute a constrained projection of the query histogram onto the subspace spanned by all the shapeme histograms of the model, and (ii) compute a match measure between the query histogram and the projection. The first step is formulated as a constrained optimization problem that is solved by a sampling algorithm. The second step is formulated under a Bayesian framework where an implicit feature selection process is conducted to improve the discrimination capability of shapeme histograms. Results of matching partially viewed range objects with a 243 model database demonstrate better performance than the original shapeme histogram matching algorithm and other approaches.

1 Introduction

The effectiveness of using semi-local shape signature, like spin image and shape context, for 2D and 3D shape matching and recognition has been demonstrated in previous work. Shapemes are defined as clusters of shape signatures that correspond to different parts on the objects. Histograms of shapemes characterizing shape distributions is a compact, albeit rich descriptor ideal for rapid and accurate shape matching. Readers can refer to the next section for a detailed description of shapeme histogram. Normally, matching shapeme histograms of two objects requires both objects to be complete. It would not work properly if the query object is only a part of the model object, e.g., the query is a range image that covers only a limited portion of an object. To address this problem, we propose to divide a model into smaller parts and compute shapeme histograms for each of them. Matching a query histogram with a model involves two steps: (i) find the parts on the model object that correspond to the query object, and (ii)

[1] This work was supported in part by DARPA/AFRL Contract No. F33615-02-C-1265

match the shapeme histogram computed from those parts to the histogram of the query object. The first step is formulated as a constrained optimization problem where the goal is to find the optimal projection of the query histogram onto the subspace spanned by all the shapeme histograms of the model. We then propose a sampling based algorithm to solve this optimization problem efficiently. The second step is expressed into a Bayesian framework where an implicit feature selection process is conducted to improve the discrimination capability of shapeme histograms.

The proposed method provides a general framework that can be used for matching partial objects of any kind with shapeme histograms. In this paper, we are interested in the application of finding 3D models in a database that are similar to an object presented in a range image. The 3D model objects and the query range images are both represented as 3D point clouds. The point clouds for a model object cover the complete object, whereas the query range images provide only partial views of a query object. The goal is to find a short list of model objects that are the closest matches to a query object.

2 Related Work

Our basic representation is based on the idea of shapeme and shapeme histogram, proposed by Mori, Belongie and Malik [1] for 2D object recognition. A shapeme is a prototype shape feature that represents a cluster of similar invariant features. These features are computed at basis points that are densely sampled on the 2D object, and hence a single object may contain many features. Each feature on an object is assigned to its closest shapeme. Counting the frequency of shapemes over the whole object gives the shapeme histogram of the object. Fig. 1 shows an example. Shapeme histogram is used by [2] and [3] for texture recognition. In this paper, we extend shapeme histogram based approach to handle the matching of partially observed objects. Moreover, we also propose to select shapemes that are critical to a given task and embed this process into a Bayesian matching framework.

The shapeme histogram for 3D objects can be constructed likewise in terms of 3D shape features and the associated 3D shapemes. Invariant, semi-local shape features such as splash feature [4], spherical harmonic [5], and spin image [6] have been developed in the past few years for matching and recognizing 3D objects. Among these features, we have chosen spin image represenation because it has been widely used and reported to be robust against clutter and occlusion [7, 8]. It should be noted that though selecting an optimal invariant feature is always important, it is not critical to the points that we want to make in this paper. More details about the spin image representation can be found in [6].

Figure 2 is the picture showing the relationships of the original shapeme histogram method and our proposed method with respect to other recognition approaches. Obviously, the most accurate method is to align the objects being matched [4, 6, 9], because it takes into account the global geometric constraints. The alignment-based approach is expensive, and has difficulty when aligning

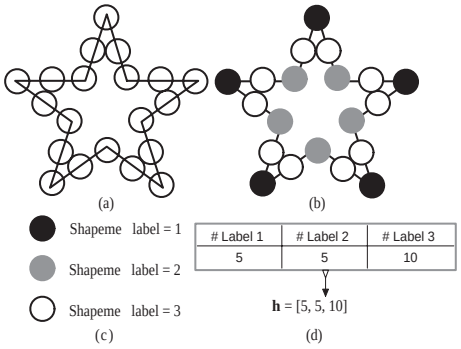

Fig. 1. Shapeme histogram. (a) A star-shaped object and the 20 basis points (the circles) where 2D invariant features are computed. (b) The same object with each feature point labeled with a set of 3 shapemes as in (c). (d) The shapeme histogram of the object in both table and vector formats. Since there are 5 #1, 5 #2, and 10 #3 labels on the object, respectively, the shapeme histogram of this object is $\mathbf{h} = [5, 5, 10]$

non-rigid objects. The nearest neighbor approach without global geometric constraints can be quite accurate when using semi-local features such as spin images. The problem is that it is too slow especially when the feature dimension is high and the database is large. The approximate nearest neighbor approach [10] and the prototype-base nearest neighbor approach [11, 12] are employed to address this problem. However, the performance of the former depends on the type of feature and the distribution of the features in the database, while the latter is in general not accurate. Shapeme histogram is a fast method because both the model and the query are represented with single vectors. However, the original shapeme histogram approach does not handle a partial query. This justifies the use of our proposed approach, which is fast and accurate even when the query object is partially observed. An ideal application of our method is to use it as a model pruning front-end where the goal is to find a short list of model objects that can be verified by more sophisticate and accurate approaches.

3 Histogram Projection

We now introduce the histogram projection idea to address the problem of matching the histogram of a partial query against a complete model. As a illustration example, consider an image matching problem. Figure 3 shows an original image (image A) and another image (image B) consisting of only four segments from the original image. Obviously the intensity histogram computed from image A looks different from image B. Given the intensity histogram $\mathbf{h}_i^s$ for each segment i of A, and the histogram $\mathbf{h}^q$ of B, the problem is to find the set $\mathcal{S}$ of all the segments such that $\mathbf{h}^q = \sum_{k \in \mathcal{S}} \mathbf{h}_k^s$.

To solve the problem, let $\mathbf{A} = [\mathbf{h}_1^s, \mathbf{h}_2^s, ..., \mathbf{h}_9^s]$ be the $l \times 9$ matrix, where l is the number of bins of each histogram, and $\mathbf{x} \in \mathcal{R}^9$ is an unknown vector.

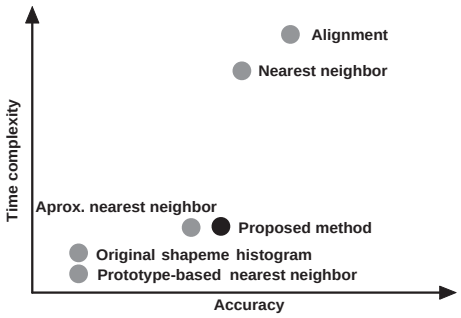

Fig. 2. The proposed method with respect to other approaches for partial object recognition. The alignment-based method is the most accurate but expensive approach. Our method is fast and accurate, and can ideally be used as a model pruning front-end for the alignment-based approach. See the text for more details

Suppose that $rank(\mathbf{A}) = 9$, solving the linear system $\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{h}^q = 0$ gives us the solution $\mathbf{x} = [0, 1, 0, 1, 0, 1, 0, 1, 0]^T$. We now conclude that $\mathcal{S} = \{2, 4, 6, 8\}$, i.e., image B consists of these numbered segments from image A.

In summary, we have projected $\mathbf{h}^q$ onto the subspace spanned by $\mathbf{h}_i^s$, and the projection is $\mathbf{A}\mathbf{x}$, which in this case is exactly the same as $\mathbf{h}^q$. Note that this is just an ideal example where the query tiles are the same as the tiles used to construct the histogram subspace.

4 Shapeme Histogram Projection for Partial 3D Object Recognition

This section elaborates the idea in the previous section, and proposes a shapeme histogram projection approach for matching partially viewed 3D objects. A schematic description of the approach is given in Fig. 4. Blocks 1, 2, and 3 are covered in this section, and block 4 is covered in the next section. In the following discussion, model objects are assumed to be complete.

4.1 Spatial Object Segmentation

In Block 1, a set of spin image $\mathcal{B}_i$ is computed for object M_i at basis points $\mathcal{P}_i$. Each spin image $\mathbf{b}_{i,j} \in \mathcal{B}_i$ is associated with a 3D basis point $\mathbf{p}_{i,j} \in \mathcal{P}_i$. The spatial segmentation algorithm performs k-means clustering in the basis point set $\mathcal{P}_i$ and returns n clusters of 3D points, where n is a predetermined number. The set of spin images $\mathcal{B}_i$ is split into n groups accordingly. Note that this process should not be confused with clustering in the spin image space, where the goal is to find the prototype features, the shapemes. Figure 5 shows two views of a segmented pickup truck.

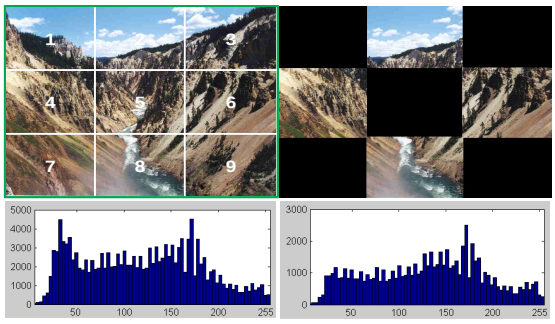

Fig. 3. An example of using histogram projection for 2D image matching. The left column shows the original image and its intensity histogram. The image is segmented into 9 pieces and each of them is labeled with a number. The right column shows an image consist of only 4 pieces of the original image. The corresponding intensity histogram is computed without including the black areas

4.2 Histogram Subspace Representation

In Block 2, once the spin images of object M_i have been separated into n groups, a shapeme histogram $\mathbf{h}_i^s$ can then be computed for each group. The histogram subspace representation is then the $m \times n$ matrix $\mathbf{A}$, as mentioned in the previous section, where $\mathbf{A} = [\mathbf{h}_1^s, \mathbf{h}_2^s, \cdots, \mathbf{h}_n^s]$, and m is the number of shapemes. This matrix is added into the model database. Obviously, as compared with the approaches that put all the raw features inside the database, the saving in storage space with the shapeme histogram representation is huge.

4.3 Shapeme Histogram Projection

In Block 3, given a query shapeme histogram $\mathbf{h}^q$, its projection (the closest approximation) in the subspace spanned by the ith model histograms can be computed as $\hat{\mathbf{h}}^q = \mathbf{A}_i\mathbf{x}$, where $\mathbf{A}_i$ is the $\mathbf{A}$ matrix of the ith model, and $\mathbf{x}$ can be computed as

$$\text{minimize} \|\mathbf{A}_i\mathbf{x} - \mathbf{h}^q\|_2 \quad \text{subject to } \mathbf{x} \in \{0,1\}^n \,. \tag{1}$$

The solution of the unconstrained version of (1) is the solution of the linear system $\mathbf{A}_i^T\mathbf{A}_i\mathbf{x} - \mathbf{A}_i^T\mathbf{h}^q = 0$, if $\mathbf{A}_i$ has full column rank. This is what we used to solve the ideal image matching problem in Fig. 3. In the case when the query histogram is noisy, or the query range image cuts through significant portions of some segments, the constraint $\mathbf{x} \in \{0,1\}^n$ can be used to "regularize" the solution. Solving (1) with exhaustive search requires 2^n operations, which is tractable only when n is small. When n is large, a Gibbs sampler [13] is employed to explore the binary solution space more efficiently. The Gibbs distribution that corresponds to the objective function in (1) is

$$G(x) = \frac{1}{Z}\exp[-(\|\mathbf{A}_i\mathbf{x} - \mathbf{h}^q\|_2)/T] \,, \tag{2}$$

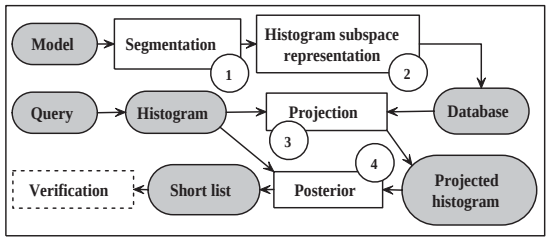

Fig. 4. Shapeme histogram projection for partial 3D object recognition. Model objects are segmented in block 1 and component based histogram are computed to form a subspace representation in block 2. The component histograms are added into a histogram database. Query histogram is computed and projected in block 3 onto the subspace spanned by the component model histograms in the database. Posteriors are computed in block 4 based on the query histogram and its projected histogram, as described later in Sec.5 (14). A short list of matching model objects are generated for optional verification

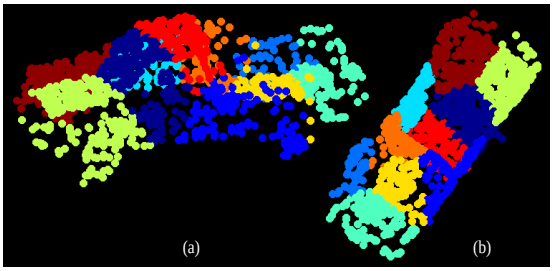

Fig. 5. Spatial object segmentation of a pickup truck (best viewed with color). The circular points represent the 3D basis points where the spin images are computed. Basis points from different spatial segments are labeled with different color. (a) A side view of the segmented object. (b) The top view

where Z is an unknown normalization factor, and T is the temperature constant. Because x is binary, the local conditional probability (or the local characteristic function) can be derived easily from (2) as

$$\begin{aligned} & G(x_j = 0 \,|\, \{x_k | k \neq j\}) \\ &= \frac{G(x_j = 0, \{x_k\})}{G(\{x_k\})} \\ &= \frac{G(x_j = 0, \{x_k\})}{\sum_{x_j \in \{0,1\}} G(x_j, \{x_k\})} \\ &= \frac{1}{1 + G(x_j = 0, \{x_k\}) / G(x_j = 1, \{x_k\})} \,, \end{aligned} \tag{3}$$

where x_k is the kth coordinate of $\mathbf{x}$. Note that the unknown factor Z is canceled out in (3). Given a random initial guess, the sampler sweeps through each

coordinate x_k sequentially, and flips its value according to the local conditional probability in (3). The computational cost in each step is negligible since only one coordinate is touched, and the actual cost $\mathbf{A}_i\mathbf{x} - \mathbf{h}^q$ can be computed incrementally. The same process is repeated for several iterations, and the $\mathbf{x}$ that has the smallest $G(\mathbf{x})$ is selected as the solution. Because our objective function is simple, and the dimension is relatively small ($n < 100$), Gibbs sampler can quickly converge to a solution very close to the global optimum. In fact, we find that when $n = 10$, Gibbs sampler gives the identical result as the exhaustive search.

5 A Bayesian Framework for Shapeme Histogram Matching

This section provides a Bayesian framework for the shapeme histogram matching. It explains why under some mild assumptions, matching shapeme histogram is equivalent to computing the posterior of a model given the query. More importantly, it reveals that an implicit feature selection process is naturally embedded when matching under the proposed framework. To simplify the discussion, we will lay out the framework based on the assumption that both the query object, and the model object are complete, and then apply it to a partial query using the histogram projection approach proposed in the previous section.

5.1 Notations

Let M_i be the ith model object, and Q the query object. The problem is to find a set $\mathcal{C}$ of candidate model objects with the largest posteriors $P(M_i \mid Q)$, such that $\sum_{i \in \mathcal{C}} P(M_i \mid Q) \geq \epsilon$, where $0 \leq \epsilon \leq 1$ is a predefined value close to 1. For each model object M_i, a set of spin images $\mathcal{B}_i = \{\mathbf{b}_{i,1}, \mathbf{b}_{i,2}, \cdots, \mathbf{b}_{i,t_i}\}$ is computed at a number of basis points $\mathcal{P}_i = \{\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \mathbf{p}_{i,t_i}\}$ densely sampled along the surface of the object, where $\mathbf{b}_{i,\cdot} \in \mathcal{R}^d$, $\mathbf{p}_{i,\cdot} \in \mathcal{R}^3$, d is the spin image dimension, and t_i is the number of spin images in the object. For all the model objects under consideration, a set of shapemes $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m\}$ is computed from the full set of spin images $\Omega = \bigcup_{j=1}^{o} \mathcal{B}_j$ by clustering in the spin image space, where m and o are the number of shapemes and model objects, respectively. For each shapeme $\mathbf{a}_k$, a median radius r_k is computed during the clustering process. Median radius is the median distance from the center of the shapeme (cluster) to the spin images that belong to this shapeme (cluster).

Each model object M_i is then represented by a shapeme histogram $\mathbf{h}_i = [u_{i,1}, u_{i,2}, \cdots, u_{i,m}]^T$, where the number of bins m is the same as the number of shapemes in $\mathcal{A}$, and $u_{i,k}$ is the number of spin images that have been labeled with the kth shapeme. The probability of each shapeme $\mathbf{a}_k$ given a model object M_i is then given by

$$P(\mathbf{a}_k \mid M_i) = u_{i,k}/u_i \,, \tag{4}$$

where $u_i = \sum_k u_{i,k}$ is the total number of labeled spin images in the object. The query is represented by the set of all spin images computed from the object.

5.2 Model Posterior Given a Query

The posterior probability of a model given a query can be computed as

$$
\begin{aligned}
& P(M_i \,|\, Q) \\
&= \sum_{\mathbf{a}_k} P(\mathbf{a}_k \,|\, Q)\; P(M_i \,|\, \mathbf{a}_k,\, Q) && \text{(5a)} \\
&\approx \sum_{\mathbf{a}_k} P(\mathbf{a}_k \,|\, Q)\; P(M_i \,|\, \mathbf{a}_k)\;, && \text{(5b)}
\end{aligned}
$$

where (5b) is an approximation of (5a) assuming that $P(M_i \,|\, \mathbf{a}_k,\, Q) \approx P(M_i \,|\, \mathbf{a}_k)$, that is when both a shapeme and the query are given, the probability of M_i is determined only by the shapeme. The same assumption is used in [11], and is coined as the "homogeneity assumption". Applying Bayes' rule to (5b) leads to

$$
P(M_i \,|\, Q) \approx \sum_{\mathbf{a}_k} P(\mathbf{a}_k \,|\, Q)\; \frac{P(\mathbf{a}_k \,|\, M_i)\; P(M_i)}{\sum_{M_l} P(\mathbf{a}_k \,|\, M_l)\; P(M_l)}\;. \tag{6}
$$

Assuming that all models are equally likely, we have

$$
P(M_i \,|\, Q) \approx \sum_{\mathbf{a}_k} P(\mathbf{a}_k \,|\, Q)\; \frac{P(\mathbf{a}_k \,|\, M_i)}{\sum_{M_l} P(\mathbf{a}_k \,|\, M_l)}\;. \tag{7}
$$

The first term in (7) is the likelihood of a shapeme conditioned on the query, and can be computed as

$$
\begin{aligned}
& P(\mathbf{a}_k \,|\, Q) \\
&= \sum_{\mathbf{b}_l} P(\mathbf{a}_k \,|\, \mathbf{b}_l,\, Q)\, P(\mathbf{b}_l \,|\, Q) && \text{(8a)} \\
&\approx \sum_{\mathbf{b}_l} P(\mathbf{a}_k \,|\, \mathbf{b}_l)\, P(\mathbf{b}_l \,|\, Q) && \text{(8b)} \\
&\propto \sum_{\mathbf{b}_l} P(\mathbf{a}_k \,|\, \mathbf{b}_l) && \text{(8c)} \\
&= \sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l \,|\, \mathbf{a}_k)\, P(\mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l \,|\, \mathbf{a}_g)\, P(\mathbf{a}_g)} && \text{(8d)} \\
&= \sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l \,|\, \mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l \,|\, \mathbf{a}_g)}\;, && \text{(8e)}
\end{aligned}
$$

where (8b) is obtained as in (5b) with the homogeneity assumption, $\mathbf{b}_l$ is the lth spin image of Q. Note that the raw feature likelihood conditioned on the query $P(\mathbf{b}_l \,|\, Q)$ is assumed to be uniformly distributed in Eqs. (8b)–(8c), and $P(\mathbf{a}_g)$ is assumed to be uniformly distributed in Eqs. (8d)–(8e). Substituting (8e) into (7), $P(M_i \,|\, Q)$ is proportional to

$$
\sum_{\mathbf{a}_k} \sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l \,|\, \mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l \,|\, \mathbf{a}_g)}\; \frac{P(\mathbf{a}_k \,|\, M_i)}{\sum_{M_l} P(\mathbf{a}_k \,|\, M_l)}\;, \tag{9}
$$

where $P(\mathbf{b}_l \mid \mathbf{a}_k)$ is the likelihood of a spin image in the query conditioned on a shapeme. This likelihood is defined as

$$P(\mathbf{b}_l \mid \mathbf{a}_k) = \begin{cases} 1 & \mathbf{a}_k = \arg\min_{\mathbf{a}_g} D(\mathbf{a}_g,\, \mathbf{b}_l) \\ 0 & \text{otherwise} \end{cases} , \tag{10}$$

where $D(\cdot, \cdot)$ is the distance between two features. Eq. (10) says that if $\mathbf{a}_k$ is the shapeme that is the closest to the spin image $\mathbf{b}_l$, the likelihood of $\mathbf{b}_l$ is one. This is spin image labeling based on a nearest neighbor vector quantization approach. Since the query object may be noisy, and obscured by scene clutter, a restricted labeling process defined as follows is preferred.

$$P(\mathbf{b}_l \mid \mathbf{a}_k) = \begin{cases} 1 & D(\mathbf{a}^*, \mathbf{b}_l) \leq 2.5\, r^* \\ 0 & \text{otherwise} \end{cases} , \tag{11}$$

where $\mathbf{a}^*$ is the shapeme closest to $\mathbf{b}_l$, and r^* is the median radius of the shapeme as defined in Sec. 5.1. Spin images severely corrupted by noise or scene clutter may not belong to any shapeme. By simply deleting them from the query spin image set, probabilities computed based on the remaining spin images become more reliable. It is then obvious from both (10) and (11) that always $\sum_{\mathbf{a}_g} P(\mathbf{b}_l \mid \mathbf{a}_g) = 1$. More generally,

$$\sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l \mid \mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l \mid \mathbf{a}_g)} = v_k \ , \tag{12}$$

where v_k is the number of spin images in the query that are labeled by $\mathbf{a}_k$. Finally, substituting (4) and (12) into (9) leads to

$$P(M_i \mid Q) \propto \sum_{k=1}^{m} \left(\frac{v_k\ u_{i,k}}{u_i}\ \gamma_k^{-1} \right) , \tag{13}$$

where $\gamma_k = (\sum_{M_l} P(\mathbf{a}_k \mid M_l))/o$ is a normalization factor that counts for the discrimination capability of the kth shapeme, and o is the number of model objects. It is obvious that (13) represents the correlation between the model and the query, with each bin scaled by γ_k^{-1}. The normalization factor γ_k can be regarded as the average probability of the shapeme $\mathbf{a}_k$'s occurrence within the whole model database. Eq. (13) gives high weights to shapemes that are rare in the database since they have small γ_k. This represents a feature selection process where shapemes that belong to many model objects are down weighted. γ is similar to the $tf \times idf$ weight [14] used widely in the document analysis community.

5.3 Posterior for Projected Histograms of a Parial Query

When the query is partially observed, we can use the method from Sec. 4 to find its projection $\hat{\mathbf{h}}_i = [\hat{u}_{i,1}, \hat{u}_{i,2}, \cdots, \hat{u}_{i,a}]^T$ in the subspace spanned by all the component histograms of model M_i. This histogram corresponds to a virtual model

$\hat{M}_i$ that matches the observed part of the query object. The model posterior can then be computed as

$$P(\hat{M}_i \,|\, Q) \propto \sum_{k=1}^{m} \left(\frac{v_k \ \hat{u}_{i,k}}{\hat{u}_i} \ \gamma_k^{-1} \right) \ . \tag{14}$$

Strictly speaking, γ_k should also be computed according to the virtual models corresponding to the query object. In practice, we find it sufficient to compute γ_k from all the complete model objects in the database.

6 Experimental Results

We have tested our method with a model set of 243 vehicles on a 2GHz PC with 2GB main memory. We will present comparative results of our proposed method against other methods. We used 500 prototype features for all the prototype-based methods and the shapeme histogram methods. We observed in our experiments that increasing the number of prototype features beyond 500 did not lead to a significant improvement in the recognition accuracy. The following is a list of the analyzed approaches.

1. The proposed histogram projection method, denoted as HP, is described in Sec. 4.3. For the Gibbs sampler, we use 30 iterations (n steps per iteration, where n is the number of segments of each model), and set the temperature const $T = 2$.
2. Nearest neighbor method, denoted as NN, matches each spin image in the query object with all the spin images in the database. A vote for that model is incremented whose spin image is the closest to the query spin image. The total matching score for each model object is computed by w_i/w, where w_i is the number of votes the ith model collects, and w is the total number of scene points.
3. Locality-sensitive hashing method, denoted as LSH, uses the same voting mechanism as in NN, but uses an approximated nearest neighbor search strategy while speeding up the search. We use 8 hash tables. See [10] for details of the LSH method.
4. Prototype-based method, denoted as PR, is detailed in [11].
5. Prototype-based subspace method, denoted as SPR, is detailed in [12]. We use a 3 dimensional subspace.

6.1 Noise Free Data Results

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, ϕ, θ), where r is the radius, ϕ is the azimuthal angle, and θ 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 $\{(0, 45°), (45°, 45°), \cdots, (360°, 45°)\}$ 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 $\mathcal{P}_i$ is uniformly selected from the (implicit) object surface, and the corresponding set of spin images $\mathcal{B}_i$ is computed (see Sec. 5.1 for notations). A set of 500 shapemes $\mathcal{A}$ is computed from the set Ω of all spin images from the 243 models. Each object M_i is then segmented into n pieces, and a histogram h_i^s is computed for each piece. The model histogram database is then constructed as described in Sec. 4. The query set contains also 243 views, one for each object. Each view is randomly sampled from $\phi \in \{10°, 20°, 30°\}$, and $\theta \in \{40°, 45°, 50°\}$, respectively, and covers $20-60\%$ of the complete object.

Table 1. Recognition accuracy of our approach as compared with other methods. HP-1, HP-10, and HP-30 are the histogram projection methods with $n = 1$, 10, and 30, respectively. HP-1 is the original shapeme histogram method

Methods	NN	LSH	HP-1	HP-10	HP-30	PR	SPR
Accuracy with top 1 model (%)	97.0	85.2	78.5	87.2	91.1	78.0	84.0
Accuracy with top 3 models (%)	1.0	93.0	88.9	94.8	97.2	87.0	91.0

Table 1 compares the recognition accuracy of our method against the others, where the accuracy is given with respect to the number of models in the short candidate list. As compared with other HP methods, the one with $n = 30$ components produces the best result. If we represent each model as a single histogram, i.e., $n = 1$, the HP method degenerates into the original histogram matching method. The bad result is expected since histogram matching in its original form does not handle partially observed queries. It can be also observed that increasing n from 10 to 30 does not improve the accuracy as dramatic as from 1 to 10. In fact, the HP method with $n = 50$ produces almost the identical result as $n = 30$.

From Table 1, it can also be seen that the LSH method does not work well for this set of database. On the other hand, the accuracy of the HP method with $n = 30$ and three candidates is close to the nearest neighbor method, and is much better than other approaches.

Table 2 compares the time and storage space used by each method. The time reported is the average time for a single query, while the space is for the whole model database. It can be seen that the speed of the HP method is in general slower than the PR method, but is still more than 60 times faster than the nearest neighbor approach. It is also faster than the LSH method. It can also be observed from the table that the HP method requires only a small fraction of the storage space needed by both the NN method and the LSH method.

One interesting aspect from the table is that the sampling based histogram projection process does not bring in much overhead in terms of the computational time. This is because our Gibbs sampler usually converges within 30 iterations, and each iteration involves a trivial update of the objective function. Fig. 6 shows some snapshots of this iterative convergence process. Fig. 7 compares query histograms with the corresponding complete model histograms, and the projected histograms. Obviously, the projected version is more similar to the query.

Table 2. Time and storage space used by each method

Methods	NN	LSH	HP-1	HP-10	HP-30	PR	SPR
Time (sec)	240	6	1.7	2.7	3.8	1.5	4.1
Space (MB)	480	500	0.6	6	18	1	720

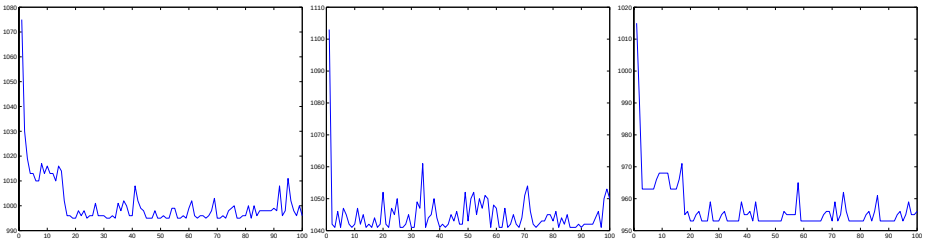

Fig. 6. Snapshots of the iterative sampling process for histogram projection of three queries to their corresponding models. The x-axis is the number of iterations ($0 - 100$), and the y-axis is the value of the objective function in (3). Thanks to the rapid convergence of the process, the best solution can usually be found within 30 iterations

6.2 Data with Synthesized Clutter and Noise

To test the performance of our method under scene clutter and noise, we added to each query histogram h_i^q a percentage of another query histogram h_j^q, $i \neq j$, and form a new query $\tilde{h}_i^q = h_i^q + \lambda h_j^q$, which now contains a certain degree of "structural" noise from another query object. This is a much more difficult test than just adding random noise or unstructured clutter into the query. We varied λ from 0.1 to 0.5 in the test and the result for the HP method ($n = 30$) is shown in Table 3. It can be seen that the accuracy of the method decreases gracefully as the level of noise increases. Note here the decrease in accuracy is mainly caused by the confusion between the projected histogram and the query histogram. In most cases, we observed that the projected histograms were still very accurate.

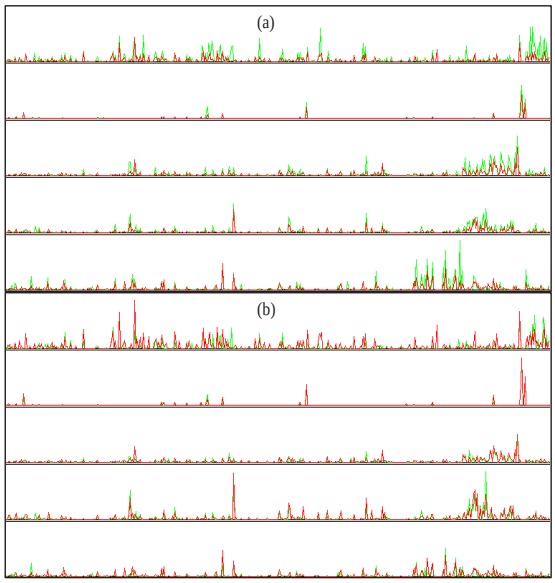

Fig. 7. The accuracy of the projected histograms (best viewed with color). (a) Model histograms (green) overlaid with query histograms (red) (b) Projected histograms (green) overlaid with query histograms (red). The projected histograms are much more similar to the query histograms

To see this, Figure 8 shows an example where (b) is the sum of (c) and (d), i.e., $\lambda = 1$. If we project (b) onto the histogram subspace of the model corresponding to (c), we get the projection (a), which indeed looks similar to (c).

Table 3. Accuracy of the HP method with different level of synthesized structural noise. The larger the λ, the more noise is added into the query

λ	0.0	0.1	0.2	0.3	0.4	0.5
Accuracy with top 1 (%)	91.1	89.0	87.3	84.0	76.5	70.1

7 Conclusion

We have proposed a two-step approach to address the problem of shapeme histogram matching with partially observed objects. We have applied the proposed method to the problem of 3D object recognition with range image. We then compared our method with other approaches and demonstrated its advantages on a commercially available database of 243 models.

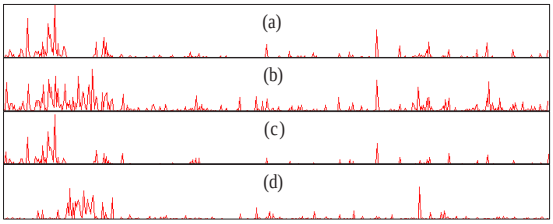

Fig. 8. Projection of a query histogram with large structural noise. See text for details

References

1. Mori, G., Belongie, S., Malik, J.: Shape contexts enable efficient retrieval of similar shapes. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. (2001) 723–730
2. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision (IJCV01) **43** (2001) 29–44
3. Lazebnik, S., Schmid, C., Ponce, J.: Affine-invariant local descriptors and neighborhood statistics for texture recognition. In: International Conference on Computer Vision(ICCV03). (2003)
4. Stein, F., Medioni, G.: Structural indexing: Efficient 3-d object recognition. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **14** (1992) 125–145
5. Kazhdan, M., Funkhouser, T.: Harmonic 3D shape matching. In: Technical Sketch, SIGGRAPH (2002). (2002)
6. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **21** (1999) 433–449
7. Ruiz-Correa, S., Shapiro, L.G., , Meila, M.: A new signature-based method for efficient 3-d object recognition. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. (2001) 769–776
8. Ruiz-Correa, S., Shapiro, L.G., Meila, M.: A new paradigm for recognizing 3-d object shapes from range data. In: International Conference on Computer Vision(ICCV03). (2003)
9. Chen, C.S., Hung, Y.P., Cheng, J.B.: Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **21** (1999) 1229–1234
10. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: The VLDB Journal. (1999) 518–529
11. L.I., K., Bezdek, J.: Presupervised and postsupervised prototype classifier design. IEEE Transactions on Neural Networks **10** (1999) 1142–1152
12. Chien, J.T.: Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **24** (2002) 1644–1649
13. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **6** (1984) 721–741
14. Salton, G.: Developments in automatic text retrieval. Science **253** (1991) 974–980