Rapid Object Indexing Using Locality Sensitive Hashing and Joint 3D-Signature Space Estimation

Bogdan Matei, *Member*, *IEEE*, Ying Shan, *Senior Member*, *IEEE*,
Harpreet S. Sawhney, *Member*, *IEEE*, Yi Tan, *Senior Member*, *IEEE*,
Rakesh Kumar, *Member*, *IEEE*, Daniel Huber, *Member*, *IEEE*, and Martial Hebert, *Member*, *IEEE*

Abstract—We propose a new method for rapid 3D object indexing that combines feature-based methods with coarse alignment-based matching techniques. Our approach achieves a sublinear complexity on the number of models, maintaining at the same time a high degree of performance for real 3D sensed data that is acquired in largely uncontrolled settings. The key component of our method is to first index surface descriptors computed at *salient locations* from the scene into the whole model database using the Locality Sensitive Hashing (LSH), a probabilistic approximate nearest neighbor method. Progressively complex geometric constraints are subsequently enforced to further prune the initial candidates and eliminate false correspondences due to inaccuracies in the surface descriptors and the errors of the LSH algorithm. The indexed models are selected based on the MAP rule using posterior probability of the models estimated in the joint 3D-signature space. Experiments with real 3D data employing a large database of vehicles, most of them very similar in shape, containing 1,000,000 features from more than 365 models demonstrate a high degree of performance in the presence of occlusion and obscuration, unmodeled vehicle interiors and part articulations, with an average processing time between 50 and 100 seconds per query.

Index Terms—Three-dimensional object recognition, hashing, indexing, pose estimation, approximate nearest neighbor.

1 Introduction

Three-dimensional object recognition is a well-studied problem in computer vision. A comprehensive survey can be found in [4]. Most research published until recently has used either synthetic data, or real data obtained in controlled environments (e.g., turntables), due to the relatively costly 3D sensors and the high price of acquiring data in real settings. Most often, data consisted of complete views of objects, with little noise affecting the measurements, and no occlusion present. Typically, surface meshes are assumed to be available for both the objects and the models in the database. In a real practical situation, however, we are interested in utilizing the raw 3D point clouds, rather than the surface meshes, since most surface reconstruction methods are slow and can create artifacts for noisy and partial measurements. The ability to rapidly insert objects acquired in the field into the model databases was another reason that prompted us not to use surface meshes in the recognition process.

Our research on 3D object recognition has been driven by the need to achieve high degree of recognition rates with a scalable method, using data acquired by 3D sensors in *uncontrolled* environments and with databases of tens, or hundreds of models. We have observed that under these operating conditions, many state-of-the-art recognition techniques proposed in the literature may not yield an adequate performance. For example, for larger databases, alignment-based recognition systems do not scale up well, because the computation time is linear in the number of models. Feature-based methods, employing surface descriptors and nearest neighbor classifiers, will have an increased recognition error when the similarity between the descriptors is reduced due to the smaller radius of descriptors required to cope with articulations in the data. The large number of features existing in the database mandates efficient retrieval techniques. Approaches based on feature dimensionality reduction with PCA, or random projections, database compression with feature editing, or approximate nearest neighbor algorithms were proposed to alleviate the increased time in searching large databases; however, some of the assumptions made do not hold well. For example, PCA requires normality assumptions about the feature distribution, many approximate nearest neighbor algorithms work well only for features with small dimensionality (less than 20).

Recognizing objects in clutter was acknowledged early on as an important problem and several effective algorithms were proposed [19], [32]. On the other hand, effects produced by articulation of parts were not addressed thoroughly, though they have been presenting difficult challenges to numerous approaches due to the drastic change in object appearance induced by the articulations. While clutter can be removed to some extent by scene segmentation techniques

- *B. Matei, Y. Shan, H.S. Sawhney, Y. Tan, and R. Kumar are with Vision Technologies Laboratory, Sarnoff Corporation, 201 Washington Road, Princeton, NJ 08540.*
  E-mail {bmatei, yshan, hsawhney, ytan, rkumar}@sarnoff.com.
- *D. Huber and M. Hebert are with Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213.*
  E-mail: dhuber@cs.cmu.edu, hebert@ri.cmu.edu.

Manuscript received 3 Mar. 2005; revised 4 Aug. 2005; accepted 5 Oct. 2005; published online 11 May 2006.
Recommended for acceptance by C. Schmid.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0113-0305.

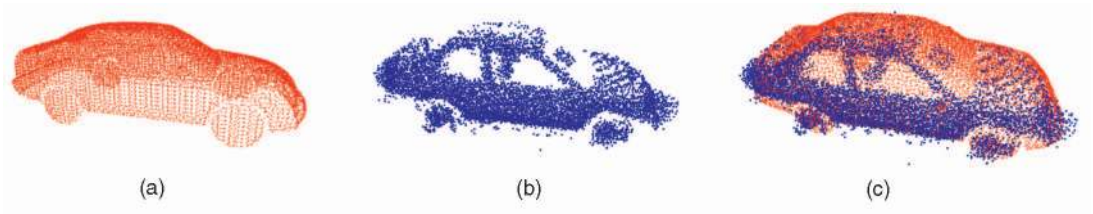


Fig. 1. (a) Dense point cloud rendered from a faceted model. (b) Real data obtained by a LADAR sensor. (c) Overlap between the scene and the model. Note the significant difference between the scene and the corresponding model in terms of blooming effects (see the right front side of the vehicle), interiors, missing data in transparent regions corresponding to windows, clutter due to unmodeled interiors.

prior to recognition, the recognition algorithms must be robust to articulations and (self-)occlusions in the target.

In real operating conditions, a 3D sensor views a scene from a limited number of viewpoints, thus only a partial region of an object is available for recognition. For instance, typically, vehicles may be close to each other and to buildings, or placed below foliage. Aerial or ground sensors may be able to scan the vehicle only from a limited set of viewpoints: Ground sensors are more stable than aerial sensors; however, the top of vehicles may not be scanned well; aerial sensors can view objects at a better elevation level; however, they may be harder to control in the case of blimps, or may induce noise in the data due to the vibrations of the craft in the case of helicopters. Measurements can have nonlinear density variation throughout the object, depending on the distance and elevation of the 3D sensors with respect to the scene. Some LADAR sensors have the ability of detecting objects concealed under foliage, thus small pieces of the target surface similar to "mouse-bites" may be missing throughout the object. Missing data due to solid obscuration produced by clutter objects located between the scene and the target is another factor adversely impacting the quality of the available data.

Sensed data is generally affected by noise mainly along the viewing direction due to the manner in which the depth information is measured. *Blooming* effects produced by "scattering" the measurements at sharp discontinuities may be inherent in the data. Discrepancies between the query objects and the corresponding models from the database have to be tolerated. For example, interiors of vehicles can be imaged by seeing through windows, but may not be present in the opaque corresponding model. Other rigid changes in an object such as articulations, affect a large class of 3D objects of interest: construction vehicles (cranes, bulldozers), military vehicles (tanks), nature (people). Articulated object recognition is challenging because the changes in a single object due to articulations can be even more significant than the differences between objects having different identity. In Fig. 1, we illustrate a typical instance of an object that has to be handled by our system together with a point cloud model consisting of concatenated views rendered from a corresponding faceted model. Note the dissimilarity between the target and the model.

1.1 Our Method

The paper focuses on *indexing* models from a database, i.e., on pruning the database into a small set of candidates containing the correct model type with a high probability. We assume that the scene consists of a target object and additional structured clutter, such as ground plane, vegetation, or other objects. The output of model indexing can be fed into a fine alignment-based verification module to determine a unique model. We illustrate the approach on recognizing vehicles, a very hard problem due to the significant similarity existing between the models from the database.

Our proposed method belongs to the class of object recognition using surface descriptors matching and builds on the earlier work of Johnson and Hebert [18], [19]. The paper extends an earlier work of Shan et al. [35]. The main contributions of our method include:

1. computational framework in the joint 3D and feature space which ensures a higher degree of discriminability between similar models and enhanced robustness,
2. scalability of the approach with a large model database maintaining at the same time a high accuracy and speed,
3. the ability to handle obscuration of the data, clutter, irregular sampling, and partial 3D views that occur when measurements are acquired in uncontrolled settings, and
4. extensive testing of the proposed approach with real data and very large model databases of similar objects.

To our knowledge, no other research published in the literature reported recognition results with real LADAR 3D data and large model database of similar objects.

In the first stage of our method, an approximate nearest neighbor (ANN) algorithm using the Locality Sensitive Hashing (LSH) algorithm is employed to find the most similar descriptors from the database to each scene descriptor. LSH is a stochastic method based on the theory of random projections that was shown to be significantly faster compared with other fast nearest neighbor search methods, such as the dynamic space partitioning, and more robust to the choice of the internal parameter settings in higher dimensional spaces. The main idea in LSH is to hash features into bins based on a probability of collision. Thus, features that are far in the parameter space will have a high probability of landing into different bins, while close feature will go into the same bucket. See [17], [13] for more theoretical and practical details. Existing recognition algorithms were shown to be theoretically sublinear with the number of models in the database; however, the databases used were relatively small and the objects in the database very dissimilar. Due to the small support of the features, errors of the LSH method and the distortions in the descriptors produced by articulations and other imperfections in the data, we employ a relatively large number of candidate matches, $Q = 50 - 100$, for each descriptor in the scene.

In the second stage, simple configurations of matches, for example, doublets of matches are sampled from the

candidate correspondences using importance sampling. Doublets that are mutually consistent are further used to generate pose hypotheses between the scene and models. Each doublet of scene-model matches is used to compute a pose candidate. Since, in our case, the number of possible pose candidate can be $O(Q^2)$ for each pair of scene features, we employ importance sampling of matches based on a similarity measure in feature space and we progressively enforce geometric constraints between the descriptors base points in order to retain only the best hypotheses.

In the third stage of the algorithm, the candidate poses are evaluated using MLESAC [38] using the likelihood of matches computed in the joint 3D-signature space. Preemptive schemes [29] that dynamically prune incorrect hypotheses are used to further speed up the evaluation of each pose hypothesis. Combining 3D and local shape information at matching level can significantly improve the recognition performance for similar models in the database. After the alignment parameters are selected for the candidate models, the final indexing decision is based on the Maximum A Posteriori Probability (MAP) rule that associates a probabilistic confidence measure with each returned model.

In Section 2, we present a brief review of the related techniques proposed in the literature. The notations are established in Section 3. Surface descriptor computation using 3D measurements acquired by range sensors is discussed in Section 4. The indexing of features using the LSH method is described in Section 5. Generation and evaluation of alignment hypotheses between the scene and the models in the joint 3D-signature space is addressed in Section 6. In Section 7, the object indexing based on MAP criterion is described. Finally, experimental results using real 3D data are presented in Section 8.

2 Related Work

The literature on free-form 3D object recognition is vast [4]. A list of proposed techniques include, without being nearly exhaustive, geometric hashing, feature-based methods, alignment of 3D data using Iterated Closest Point (ICP), surface descriptor matching.

Geometric hashing [22], [31] and its variants employ low-dimensional object descriptions that combine (quasi-)invariant coordinate representations with geometric coordinate hashing to prune a model database while employing simple geometric constraints. The time and space complexity of creating geometric hash tables is, however, polynomial in the number of feature points associated with each model. Furthermore, since the (quasi-)invariant coordinate representations are very low-dimensional (typically, two or three), the hash tables can become crowded even with small model databases and the runtime complexity can deteriorate to a linear complexity that, again, does not scale with the size of the database. Geometric hashing was shown to have poor performance with noise and clutter [14]. Geometric hashing algorithms that employ higher dimensional features (eight-dimensional) have been proposed [23].

Feature-based recognition methods encode object regions into points in a very high-dimensional space and define the similarity between objects by measuring how close the scene features are to stored model features. They have the advantage of being able to handle large model databases and changes due to pose variation without the need of explicit alignment between the scene and model features. However, their performance degrades in the presence of clutter and articulations. Various surface descriptors were proposed in the literature: global features (Spherical Attribute Image (SAI) [8], curvedness orientation shape map on a sphere (COSMOS) [9], shape distributions [30]), semilocal features (spin-images [18], harmonic shape images [40], 3D shape context [11], surface signatures [39], point signatures [6]). In [30], the *shape distributions* were proposed to measure global geometric properties of an object and compare 3D models without pose registration and feature correspondence. The method requires full scans of objects to be available. In general, global features are more discriminative, but are more affected by occlusion and clutter.

Features proposed employed Euclidean or geodesic distance measures computed on the surface. Features using geodesic distances, though potentially having a higher discriminant power, are more affected by noise and surface errors and usually require surface meshes to be available.

Alignment-based recognition methods recognize an object by aligning it sequentially to the database models using variants of the ICP algorithm [3], [36] and selecting the models that yield the smallest alignment error. Though generally very accurate, alignment techniques are not scalable to a very large model database and require good initialization of the pose between scene and the models in order for the ICP algorithm to converge. Robust initialization can be done by generating pose hypotheses using either RANSAC guided by features [18], or triplets of points sampled from the measurements [5]. Other initialization methods for finding initial poses between scene and models, include clustering [37], or Hough transformation [24]. Keselman et al. [21] formulate the recognition problem in terms of graph-matching of configuration of features. They use a low-distortion graph embedding to map vertex-labeled graphs to a set of vectors in a low-dimensional space and solve 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.

Alternatively, feature-based methods can be used to prune the database into a short list, that is used by the alignment methods to make the final recognition system. Mori et al. use 2D shape signatures called *shape contexts* to find a short list of candidate models [27]. Though the idea was used in the context of 2D recognition, it can be easily extended to 3D object recognition. 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 of models prematurely and miss the correct match. Other related methods include [24], [37], [21].

An effective 3D recognition approach based on using rotationally invariant surface descriptors to guide the generation of pose candidates between the scene and the model features in order to explore more efficiently the space of alignment parameters can be traced to Stein and Medioni [37]. They introduced the *splash* feature to describe the local surface of objects and obtained correspondences between scene and model splashes using hashing. To compensate for the exponential time required by hashing in higher (3 to 14) dimensional spaces, the authors used very coarse bins that lead to a very large number of potential correspondences.

These correspondences were used subsequently to generate hypotheses that were checked for mutual consistency using distance and orientation constraints. The model corresponding to the hypothesis yielding the smallest reprojection error was finally chosen as the recognized instance of the target. The high number of possible correspondences, due to diminished feature discriminability, requires the generation and verification of a large number of hypotheses and may result in low recognition rates.

Johnson and Hebert in [18] proposed a richer, rotationally invariant semilocal surface descriptor, the *spin image*, which is a 2D histogram of the surface locations around a 3D point. Spin-images are defined in a high-dimensional space (100-300) and have a higher discriminant power compared with splashes. In the first stage of the spin-image-based recognition system from [19], spin-images are computed at sampled locations in the scene and compared with model spin-images stored in a database. To reduce the time required for searching in large databases, the authors resorted to dimensionality reduction of the features using PCA and employed the fast nearest neighbor method of Nene and Nayar [28]. In the second stage, candidate poses between scene and models are hypothesized using the spin image correspondences and verified using RANSAC [10]. The recognized model is determined as the model having the best overlap with the scene.

Hypothesize and test methods compare favorably with respect to geometric hashing approaches in terms of storage requirements, having a constant, rather than polynomial storage requirements. When the number of models increases, a high degree of compression with PCA may not be possible, due to the multimodality of the data, or other departures from the normality assumption. Many fast nearest neighbor algorithms can become very sensitive to the choice of parameters, such as the radius used to locate the nearest neighbor feature(s), with the sensitivity increasing with the dimensionality of the space in which the search is performed. For example, the fast nearest neighbor algorithm of Nene and Nayar uses dynamic space partitioning to return the closest features within a ball with radius ϵ. A small value for ϵ may result in no correspondences returned, while a higher value yields too many neighbors and a huge increase in the computational cost. These effects become more prevalent in higher dimensional spaces, $d > 20$, thus mandating the use of dimensionality reduction for the features when dynamic space partitioning methods are used.

In [32], a variant of the spin-image, the *spherical spin-image*, which is robust to clutter and occlusion, is used for feature-based object recognition in a similar framework with [18]. To improve the speed of retrieving similar features from the database, the authors used random projections to reduce the dimensionality of features; however, the largest model database tested contained only five models.

Three-dimensional object recognition and classification using a part-based model representation offers increased robustness to articulations of parts and better generalization to objects which are not present in the database. Ruiz-Correa et al. [33] employ *surface shape signatures* to encode the relationship between parts and a support vector machine to perform the final classification. Huber et al. [16] use K-means clustering of spin-images computed for semantic parts defined for each model and a nearest neighbor indexer to perform the final classification of models.

3 Notation

Let M denote the number of models available in the database, and denote by N_k the number of features stored for model $k = 1, \ldots, M$. The total number of features in the database is

$$F = \sum_{k=1}^{M} N_k.$$

Let $\boldsymbol{\xi}_u^k = (\boldsymbol{o}_u^k, \boldsymbol{n}_u^k, \boldsymbol{x}_u^k)$, $u = 1, \ldots, F$ denote a surface descriptor for model k. The model assignment of feature $\boldsymbol{\xi}_u^k$ will be represented by the indicator variable ν_u, with $\nu_u = 1, \ldots, M$. The origin of the coordinate system for feature $\boldsymbol{\xi}_u^k$ is $\boldsymbol{o}_u^k \in \mathbb{R}^3$, the normal to the local surface is $\boldsymbol{n}_u^k \in \mathbb{R}^3$, $\|\boldsymbol{n}_u^k\| = 1$, while $\boldsymbol{x}_u^k \in \mathbb{R}^s$ is the signature corresponding to the surface descriptor, characterizing the semilocal 3D shape information around $\boldsymbol{o}_u^k$. Similarly, a surface descriptor for the scene is $\boldsymbol{\zeta}_j = (\boldsymbol{o}_j, \boldsymbol{n}_j, \boldsymbol{x}_j)$, $j = 1, \ldots, N$ with N being the number of features computed for the scene.

Let $\Upsilon_k = \{\boldsymbol{\xi}_{u_1}^k, \ldots, \boldsymbol{\xi}_{u_{N_k}}^k\}$ be the surface descriptors corresponding to model k and denote by $\mathrm{Z} = \{\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_N\}$ the surface descriptors computed for the scene. Let J_k represent the indices of the model k features in the flat database. Features from the whole model database are expressed by $\Upsilon = \{\Upsilon_1, \ldots, \Upsilon_M\}$. The number of elements of a set A is $|\mathrm{A}|$.

4 Surface Descriptor Computation from Range Data

4.1 Data Preprocessing

The scene consists of multiple views (looks) $v = 1, \ldots, N_v$ of a scene, coregistered to the same coordinate system using multiview matching algorithms. Automatic registration of views is facilitated by the fact that many range sensors provide GPS information for the data acquired which can be used as initialization for the registration methods. We assume that, for every look acquired by the range sensor, we have available information about its location with respect to the scene. In general, the size of the scene is relatively small compared to the distance from the sensor to the scene, thus we can make the simplifying premise that all the measurements of that look were viewed under the same direction $\boldsymbol{l}_v$ defined as $\boldsymbol{l}_v = (\boldsymbol{P}_v - \tilde{\boldsymbol{z}}_v)/\|\boldsymbol{P}_v - \tilde{\boldsymbol{z}}_v\|,$, where $\tilde{\boldsymbol{z}}_v$ is the centroid of the data within view v and $\boldsymbol{P}_v$ is the location of the sensor in the scene coordinate system.

In order to access efficiently the measurements we have employed a uniform voxelization of the data. Voxels are allocated only at those locations in which there are measurements present. The voxels can be accessed directly by traversing a list of allocated voxels (lists have, in general, faster insertion than vectors), or by traversing a 3D array which stores pointers to the allocated voxels and NULL for unallocated voxels. Other more memory efficient data structures such as oct-trees or k-d trees [7] could have been employed instead, however, for the typical scenes handled which are around $10 \times 10 \times 3$ m, we have found that a 3D array structure offers very fast access (no need to traverse trees) with a relatively small memory penalty required.

Prior to feature estimation, scene has to be preprocessed to remove nontarget clutter produced by vegetation, or ground. For instance, ground can be removed by making use of existing physical constraints: For example, laser

scanners cannot penetrate below the ground; therefore, the ground points have typically the lowest vertical coordinate. Vegetation clutter can be removed because it violates coherent surface constraints. Clutter removal can be carried out using the eigenanalysis of the 3D scatter of measurements and grouping regions into larger, coherent patches that have consistent normals, or alternatively using tensor voting [26]. The aspects related to target acquisition are not discussed in the paper. Instead, we will assume that the scene consists of the target plus residual clutter.

The density of points acquired by range sensors varies nonlinearly within the scene due to:

- elevation of the sensor with respect to the scene. Surfaces having normals close to being perpendicular to the corresponding viewing direction are sampled with less points,
- combining the measurements from multiple views results in data with higher density in the overlapping regions, and
- specularity of the surface depending on the material reflectivity properties.

In order to mitigate the disparities between the density of the point clouds in the scene and the models, which may affect the corresponding features calculated, we employ a simple, nonparametric density estimation based on counting the number of measurements that lie within a 3D region. The calculation is performed at every allocated voxel and the density at a voxel $V(\boldsymbol{z})$ having the 3D coordinate $\boldsymbol{z} \in \mathbb{R}^3$ is calculated as

$$\rho_{V(\boldsymbol{z})} = 1/N_{V(\boldsymbol{z})}, \tag{1}$$

where $N_{V(\boldsymbol{z})}$ is the number of 3D points $\boldsymbol{z}_k$ which lie within a sphere with radius R_d centered at $\boldsymbol{z}$, the origin of the voxel. All the measurements belonging to a voxel will have assigned the same value of the density (1) computed at that voxel. In our system, we have used $R_d = 0.2$ m. More accurate techniques using kernel smoothing can be employed to reduce the existing artifacts; however, at a higher computational cost.

Besides the density compensation, another factor that can improve the discriminability of the surface descriptors is how well the scene local coordinate systems $(\boldsymbol{o}_j, \boldsymbol{n}_j)$, of descriptor $\boldsymbol{\zeta}_j$, $j = 1, \ldots, N$ are estimated under noise. Typically, for many 3D sensors, the noise affecting the measurements is much higher along the viewing direction compared to the noise along the other two directions, i.e., is *heteroscedastic*. Finding normals in heteroscedastic noise requires solving an iterative generalized eigenproblem [25].

At a given location $\boldsymbol{P}_j$, the normals are estimated by employing the data within a radius w. A larger region can improve the normal estimates under noise; however, it can be affected by discontinuities in the surface. Iterative eigenvalue methods, such as HEIV [25], though more accurate, are relatively computationally intensive. Since the scenes consist of tens of thousands of points, the use of HEIV would be too slow. Therefore, we employ an approximation to the HEIV algorithm by using the Generalized Total Least Squares (GTLS) estimator and determining a dominant viewing direction for each 3D region for which the estimation is performed. The normal estimate $\hat{\boldsymbol{n}}$ is selected as the smallest generalized eigenvalue of the eigenproblem

$$\begin{aligned} &\mathrm{S}\,\boldsymbol{y} = \lambda \Sigma_v \,\boldsymbol{y}, \quad \mathrm{S} = \sum_{i=1}^{N_w} (\boldsymbol{z}_i - \tilde{\boldsymbol{z}})(\boldsymbol{z}_i - \tilde{\boldsymbol{z}})^\top, \\ &\tilde{\boldsymbol{z}} = \frac{1}{N_w} \sum_{i=1}^{N_w} \boldsymbol{z}_i, \quad \|\boldsymbol{z}_i - \boldsymbol{P}_j\| \le w, \end{aligned} \tag{2}$$

where Σ_v is the covariance associated with view v (computed by rotating the covariance $\mathrm{diag}(\sigma_x, \sigma_y, \sigma_z)$ of the noise in sensor coordinate system using $\boldsymbol{l}_v$) and S is the scatter of the measurements within the radius w of $\boldsymbol{P}_j$. Normals can be defined only at those locations $\boldsymbol{P}_j$ satisfying $\lambda_{3j} \le \tau \lambda_{2j}$, where $\lambda_{1j} \ge \lambda_{2j} \ge \lambda_{3j}$ are the generalized eigenvalues of (2) and $\tau = 0.1 - 0.2$ is a threshold on the aspect ratio of the eigenvalues. The normals estimated using (2) have a gauge freedom about the sign that can be resolved uniquely by enforcing the additional constraint $\boldsymbol{n}^\top \boldsymbol{l}_v > 0$.

4.2 Surface Descriptors Used

A large class of recognition algorithms proposed in the literature characterize the surface of an object using surface descriptors computed at sampled locations in the data. Since the pose between the scene and the models is not known, the surface descriptors are chosen to be rotationally invariant. For example, spin-images require the definition of an object-centric local coordinate system specified by an origin and the normal direction to the local surface. To remove the degree of freedom left, the 3D information within the support region of the feature is integrated into the descriptor by "spinning" the coordinate system around the normal. 3D shape context descriptors [11] define a similar coordinate system; however, the descriptors are not rotationally invariant, therefore, the descriptors are calculated at sampled azimuth angles between 0 and 360 degrees.

The surface descriptors for the models are computed at locations sampled uniformly from the data. In general, between 2,000 and 3,000 surface descriptors are estimated for each model to ensure a dense enough representation of the model required in the later stages of the recognition algorithm.

Surface descriptors characterizing planar regions have little discriminant power. In general, descriptors with a smaller support radius are even more affected by the lack of 3D texture information. Uniform sampling of features in the scene may be undesirable in this case, because it decreases the percentage of scene descriptors that can uniquely identify the correct model from the database. Instead, the scene features can be computed at locations biased toward regions that have multiple surface orientations, thus are rich in 3D texture. Selection of salient locations is based on computing, at all 3D locations $\boldsymbol{P}_j \in \mathbb{R}^3$ for which normals can be estimated, the eigenvalues $\lambda_1^s(P_j) \ge \lambda_2^s(P_j) \ge \lambda_3^s(P_j)$ of the scatter matrix

$$\begin{aligned} &\mathrm{S}(P_j) = \\ &\frac{1}{N_{P_j}} \sum_i (\boldsymbol{z}_i - \boldsymbol{P}_j)(\boldsymbol{z}_i - \boldsymbol{P}_j)^\top, \quad \|\boldsymbol{z}_i - \boldsymbol{P}_j\| \le w_s, \quad ; w_s >> w, \end{aligned} \tag{3}$$

where N_{P_j} is the number of 3D measurements $\boldsymbol{z}_i$ having a distance to $\boldsymbol{P}_j$ less than w_s. The smallest eigenvalues $\lambda_3^s(P_j)$ of the scatter (3) provide a good 3D saliency measure, similar to the one used in the Harris corner detection [15]. Candidate locations $\boldsymbol{P}_j$ for which the normal can be estimated are sorted in decreasing order of their 3D saliency measure. Subsequently, locations are selected in a greedy manner from the

sorted list. In order to ensure a good coverage of the object surface locations, unassigned locations $\boldsymbol{P}_k$ within a distance D from already selected locations $\boldsymbol{P}_j$ are removed from the candidate list.

We have chosen spin-images as local surface descriptors, although other similar surface descriptors (e.g., shape contexts) could have been used as well. With shape contexts, the size of the database would be however be much larger, since local coordinate systems have to be sampled at various azimuth angles. We have found that spin-images offer a good balance between the discriminant power of the features, the storage space required and the computational complexity of enforcing geometric constraints between features. Spin-images were also shown to be tolerant to occlusions and clutter.

Spin-images are defined as 2D histograms of *in-plane* and *out-plane* distances of all the 3D points within a radius r of a local coordinate system $(\boldsymbol{o}, \boldsymbol{n})$, where $\boldsymbol{o} \in \mathbb{R}^3$ is the origin and $\boldsymbol{n}, \|\boldsymbol{n}\| = 1$ is one of the directions [19]. Since only one direction is specified, there is a gauge of freedom left about the angle around $\boldsymbol{n}$. For a point $\boldsymbol{z} \in \mathbb{R}^3$, $\|\boldsymbol{z} - \boldsymbol{o}\| \leq r$ we find the projection $\hat{\boldsymbol{z}}$ of $\boldsymbol{z}$ onto the plane specified by $(\boldsymbol{o}, \boldsymbol{n})$. The *in-plane* distance $\|\hat{\boldsymbol{z}} - \boldsymbol{o}\|$ is unsigned, while *out-plane* distance $\boldsymbol{z} - \hat{\boldsymbol{z}}$ is signed. Note that by construction the descriptor is rotationally invariant. To ensure that the descriptors computed for scene and models are invariant to density, the in-plane and out-plane distances of each point $\boldsymbol{z}$ are weighted by the density (1), computed for the voxel to whom $\boldsymbol{z}$ belongs, when added to the 2D histogram. Previous research accounted for different mesh resolutions by weighting the 3D points in the histogram by the area of the triangles forming the mesh, or employed mesh simplification [20]. Alternatively, sampling points from the mesh surface can achieve a similar effect for balancing the mesh resolution. However, recovering the mesh structure for the queries is affected by noise in the data, is more computationally intensive than fitting planar patches and can lead to artifacts in the surface.

The feature discriminability depends on several factors: 1) Larger support regions yield features that are more discriminant; however, less resilient to obscuration, clutter, and articulation of parts. 2) Smaller bins in the histogram yield features that are again more discriminant, but can tolerate less noise. Depending on the 3D data and the operating conditions, the spin image internal parameters (radius r, number of rows and columns of the 2D histogram) have to be adjusted.

5 Feature Indexing Using LSH

Searching a large model database for the nearest neighbors in a high-dimensional space may be extremely time consuming. Locality-sensitive hashing (LSH) is a state-of-the-art technique introduced by Indyk and Motwani [17] to alleviate this problem. LSH is a probabilistic solution for the approximate nearest neighbor problem. The method is based on sampling K times the feature space with hyperplanes aligned with the coordinate axes. Assuming that feature vectors $\boldsymbol{x}_k \in \mathbb{R}^s$ are translated such that they belong to a bounding box centered at the origin (hence, coordinates $x_{k,i} \geq 0$), we generate K random pairs (u, g_u), where $u = 1, \ldots, s$ is a random integer coordinate index and g_u is a float value between 0 and G_u, the largest value of features along axis u, i.e. $0 \leq x_{k,u} \leq G_u, \forall k$.

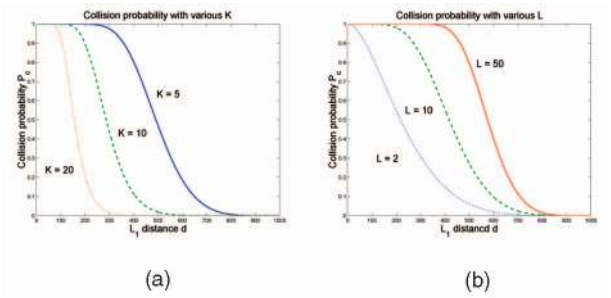


Fig. 2. LSH probability of collision with various K and L. (a) Parameter K is varied and $L = 20$. (b) Parameter L is varied and $K = 5$. d_c is set to $1,000$.

For a feature point $\boldsymbol{x}_k$ in the database, the Boolean test whether $x_{k,u} \lessgtr g_u$ is used to compute a hash code into a second level of hashes which stores the corresponding index k for fast online retrieval. The previous procedure is repeated L times, therefore each point will belong to L tables. At run time, we determine for a query point $\boldsymbol{X}$ the L first-level, K-dimensional hash codes and retrieve all the points within the corresponding second-level hash tables.

The unique property of LSH is that it relates the probability of collision to the L_1 distance between two vectors [13]. 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 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.

The probability of collision as the function of the L_1 distance has the following form

$$P_c = 1 - \left(1 - (1 - d/d_c)^K\right)^L, \tag{4}$$

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. Fig. 2 plots the curves of the function in (4) with different K and L.

Intuitively, increasing K reduces the probability of collision, and increasing L increases the probability. A large probability of collision will result in numerous possible candidates returned for a query point and a sharp increase in the execution time. A detailed discussion of the influence of the two parameters on the execution time and the accuracy of the approximate neighbors returned is done in Section 8.2. It can also be seen that the probability of collision 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.

Methods for tuning the LSH parameters K and L in order to maximize the performance of LSH were also proposed [12]. A similar approach is presented in Section 8.2. To reduce the dependency of the LSH on the two parameters, Bawa et al. proposed recently the *LSH forest* [2] which was shown by the authors to improve the retrieval performance on skewed data distributions without the need to retune the algorithm.

The downside of LSH is that it may introduce some errors in the closest features returned comparatively with the nearest neighbor method. In our approach, this can be tolerated because of the extra geometric constraints that are used.

Feature indexing using LSH returns for each scene descriptor $\boldsymbol{\zeta}_j$, $j = 1, \ldots, N$, a set of features $\Xi_j = \{\boldsymbol{\xi}_{u_1}^{\nu_{u_1}}, \ldots, \boldsymbol{\xi}_{u_q}^{\nu_{u_q}}\}$, with $q \leq Q$ the number of similar features

extracted from the database and Q the maximum number of features allowed. Let $I_j = \{u_1, \ldots, u_q\}$ be the set of indices of the model features and $\Psi = \{\Psi_1, \ldots, \Psi_N\}$, be the available putative correspondences between the scene and the models according to the similarity in the signature space, where $\Psi_j = (\boldsymbol{\zeta}_j, \Xi_j)$.

6 Scene to Model Alignment Using 3D-Signature Constraints

We have shown how LSH can be used to find the approximate nearest neighbors Ξ_j for each scene surface descriptor $\boldsymbol{\zeta}_j$ using only feature information. In this section, we will show how the space of possible correspondences Ψ can be pruned using geometric constraints. We will address two important factors that influence the speed of finding correct alignment hypotheses between the scenes and the database models. The first factor is related to the generation of candidate alignment candidates between the scenes and the models and is discussed in Section 6.1. The second factor regards the evaluation of hypotheses using the likelihood computation in the joint 3D-signature space and is discussed in Section 6.2. Efficiency is achieved by using preemptive evaluation schemes that employ only sections of the data at a time.

6.1 Doublet-Based Hypothesis Generation and Pruning

A candidate alignment between the scene and a model can be determined by sampling two surface descriptors $\boldsymbol{\zeta}_{j_1}$ and $\boldsymbol{\zeta}_{j_2}$ from the scene and sampling candidate matches $\boldsymbol{\xi}_{u_1}^{\nu_{u_1}}, \boldsymbol{\xi}_{u_2}^{\nu_{u_2}}$ from Ξ_{j_1}, respectively, Ξ_{j_2} such that they belong to the same model, $\nu_{u_1} = \nu_{u_2}$. The size of the total space of alignment parameters Θ between the scene and the models, assuming that the models are analyzed independently is $6M$. Moreover, the alignment parameters cannot be found by analyzing each model sequentially since the approach will not be scalable. Data-driven alignment hypothesis generation can speed up finding good solutions by more efficiently exploring the pose space. We propose the use of *feature saliency* for scene descriptors and the binning of descriptors to ensure a good coverage of the scene. The generation of doublets of matches can be further improved by using the signature similarity of the candidate matches and use of simple mutual consistency checks to prune the incorrect matches prior to evaluation.

In the feature-based pruning with LSH, we have employed only feature information. As mentioned previously, though the spin image is a rich descriptor, the maximum number of model candidates Q employed depends on the feature discriminability, imperfections in the data, or errors due to the LSH algorithm. Enforcing simple geometric constraints between configurations of candidate matches was proposed in the literature before to prune inconsistent matches, without recovering alignment information. For example, Stein and Medioni use *distance constraints*, *orientation constraints*, and *direction constraints* between pairs of splashes in the scene and in the model [37]. A slightly different approach was followed by Johnson and Hebert in [18].

The approach we use is similar to [37] and is illustrated in Fig. 3. Four constraints are sequentially tested:

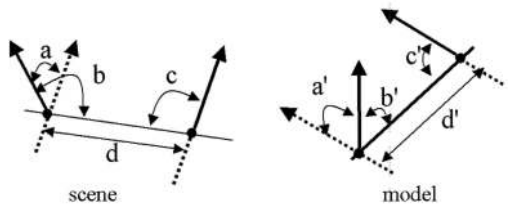


Fig. 3. Two sets of correspondences in addition to having similar surface signatures, are required to be geometrically consistent: $d \approx d'$, $a \approx a'$, $b \approx b'$, $c \approx c'$. These constraints are applied sequentially to prune pose hypotheses without recovering the alignment parameters.

1. Distance between the signatures origins has to be small, $|d - d'| \leq \epsilon \max(d, d')$.
2. The angles must obey the following constraints: $|a - a'| < \eta$,
3. $|b - b'| < \eta$, and
4. $|c - c'| < \eta$.

We have used $\epsilon = 0.1 - 0.2$ and for computational efficiency reasons we employ cosine values for the angle constraints. The threshold of the cosine angles depends on the noise affecting the scene and how reliable the normals are estimated. In our recognition system, we employ $\eta = 0.1 - 0.2$ for the angle-based doublet pruning.

The upper bound on the number of hypotheses that can be generated exhaustively for two randomly chosen scene descriptors is Q^2. Let $1 - P_\epsilon$ be the probability of the inliers in the scene, defined as the probability that a scene feature has a correspondent within the indexed model features I_j which is validated also by the 3D geometry. The minimum number of doublets of scene features N_d that have to be sampled can be found from the known condition

$$\left(1 - (1 - P_\epsilon)^2\right)^{N_d} < P_{fail}.$$

For $P_{fail} = 0.01$ and $P_\epsilon = 0.9$, we obtain that $N_d \approx 200$. The maximum number of hypotheses $N_h^{max} = N_d Q^2$ and under the previous conditions with $Q = 50$ would yield $N_h^{max} = 500,000$ which is clearly impractical. We show next how the number of hypotheses to be generated can be reduced using feature saliency and signature similarity between features.

Ideally, we want to select scene descriptors $\boldsymbol{\zeta}_{j_1}$ and $\boldsymbol{\zeta}_{j_2}$ such they are not close and do not belong to the same surface or, equivalently, their base points satisfy

$$\|\boldsymbol{o}_{j_1} - \boldsymbol{o}_{j_2}\| \geq d_{min} \qquad |\boldsymbol{n}_{j_1}^\top \boldsymbol{n}_{j_2}| \leq \alpha_{max}. \tag{5}$$

Bucketing of features into 3D bins was used previously as a simple and effective way of maximizing the probability of sampling features from different locations in the scene. We have employed bins with a side equal to 0.5 m.

Feature saliency is another useful information that can be used to guide the selection of scene descriptors. Feature saliency for a descriptor $\boldsymbol{\zeta}_j$ can be defined from the histogram of models IDs k to whom the features from the set Ξ_j belong. When a scene descriptor has putative descriptors belonging to different models in the database (i.e., the histogram is flat) it means that it is likely to have a small discriminant power. On the other hand, when there are relatively few models present, the scene feature has a higher discriminant power, i.e., is *salient*. One possible feature saliency measure is the entropy of the histogram of model IDs k, which is similar to the

measures used in [34] in the context of shape-based histogram matching. Scene features are selected according to this entropy measure using importance sampling.

Let $\boldsymbol{\zeta}_{j_1}$ and $\boldsymbol{\zeta}_{j_2}$ be two scene descriptors obeying constraints (5). One approach to limit the number of hypotheses to be generated is to sample model candidate features guided by the distance between the scene and model signatures. The candidate correspondences Ξ_j are assumed to be sorted in the decreasing order of their similarity with the descriptor $\boldsymbol{\zeta}_j$. Though LSH employs the L_1 norm, the signature similarity can be measured using other distances, such as L_2, χ^2, etc. For example, the probability of a match in the signature space $p_s(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^{\nu_u})$ can be defined by employing a normal kernel

$$K(\boldsymbol{y}) = \frac{1}{(2\pi)^{p/2}} \exp(-0.5\|\boldsymbol{y}\|^2), \tag{6}$$

where p is the number of degrees of freedom of vector $\boldsymbol{y}$. Using (6), we have

$$p_s(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^{\nu_u}) = K[\mathrm{H}_j^{\dagger}(\boldsymbol{x}_j - \boldsymbol{x}_u^{\nu_u})], \tag{7}$$

where $\mathrm{H}_j^{\dagger}$ is the pseudoinverse of H_j, the square of the covariance matrix C_j, defined as $\mathrm{C}_j = \mathrm{H}_j^{\top}\mathrm{H}_j$. Estimating C_j is very hard, due to the high-dimensionality of the signature space, thus instead of (7), we can use the approximation

$$p_s(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^{\nu_u}) = K[\sigma_s^{-1}(\boldsymbol{x}_j - \boldsymbol{x}_u^{\nu_u})], \tag{8}$$

where σ_s is the standard deviation of features in the signature space. We discuss the estimation of σ_s in Section 7.

Assuming independence between features, the probability of a doublet of matches $p[(\boldsymbol{\zeta}_{j_1}, \boldsymbol{\xi}_{u_1}^{\nu_{u_1}}), (\boldsymbol{\zeta}_{j_2}, \boldsymbol{\xi}_{u_2}^{\nu_{u_2}})]$ is

$$\delta_{\nu_{u_1},\nu_{u_2}} p_s(\boldsymbol{\zeta}_{j_1}, \boldsymbol{\xi}_{u_1}^{\nu_{u_1}}) p_s(\boldsymbol{\zeta}_{j_2}, \boldsymbol{\xi}_{u_2}^{\nu_{u_2}}), \tag{9}$$

with $\delta_{\nu_{u_1},\nu_{u_2}}$ being the Dirac function $\delta_{\nu_{u_1},\nu_{u_2}} = 1$ if $\nu_{u_1} = \nu_{u_2}$, and 0 otherwise. The importance sampling of candidate hypotheses, given two scene descriptors $\boldsymbol{\zeta}_{j_1}, \boldsymbol{\zeta}_{j_2}$ can be carried out using (9). The importance sampling depends on the scale σ_s: A value that is too small leads to the elimination of large section of the possible matches and can hinder finding good alignment hypotheses. Likewise, a value that is too large will lack any selectivity yielding a performance similar to uniform sampling. Note that the distribution of residuals in the signature space can depart from the independent and identically distributed (i.i.d.) assumptions made in (8).

A compromise between the exhaustive evaluation of all the possible combinations and the importance sampling when a scale in signature space is not available at this stage of the algorithm is illustrated in Fig. 4. This method does not rely on any knowledge about the scale and is based on deterministically generating and testing hypotheses using matches sorted according to the similarity between the scene and model signatures, while ensuring that the pair of model features belongs to the same model. The number of hypotheses that are generated for a pair $\boldsymbol{\zeta}_{j_1}, \boldsymbol{\zeta}_{j_2}$ is limited to a maximum value n_t, while the number of consistent hypotheses for a model is limited to $n_m \leq n_t$. In Fig. 4, for example, assuming that n_m hypotheses can be generated using features *solely* from block $(a, 1)$ for the model denoted by dark gray, only hypotheses for the model represented by light gray are subsequently searched in $(a, 1), (a, 2), (b, 1)$ until the terminating conditions are met. In this manner, the number of hypotheses generated is drastically limited and the performance is superior compared to uniform sampling of hypotheses.

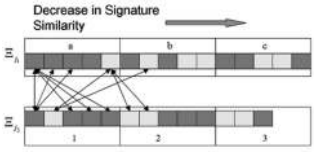


Fig. 4. Hypothesis generation using sorting of matches according to their signature similarity. For illustrative purpose, it is assumed that the features candidates belong to only two models. Each list of feature candidates Ξ_j is divided into blocks of features a, b, c, respectively, 1, 2, 3. Candidate matches are selected by starting the hypothesis generation and evaluation within blocks $(a, 1)$, then $(b, 1)$, $(a, 2)$, $(b, 2)$, etc. Candidate doublets are generated for a model k until the number of consistent hypotheses is equal to n_m, or until the number of all the hypotheses is equal to n_t. Arrows denote pair of features which belong to the same model.

The doublet of scene-model matches is sufficient to estimate the alignment between the scene and a given model. Denote an alignment hypothesis by $\boldsymbol{\omega}_k^h = \{\boldsymbol{\zeta}_{j_1}, \boldsymbol{\zeta}_{j_2}, \boldsymbol{\xi}_{u_1}^k, \boldsymbol{\xi}_{u_2}^k, \boldsymbol{\theta}_k^h\}$, where $\boldsymbol{\theta}_k^h \in \mathbb{R}^6$ is the minimum pose representation from the scene to model k. Also, let $(\mathrm{R}_k^h, \boldsymbol{t}_k^h)$ be the rotation and translation corresponding to $\boldsymbol{\theta}_k^h$. We have employed the algorithm of Arun et al. [1] based on SVD to recover the rotation and translation between the scene and the model. Other estimators based on Horn's method using quaternions were shown to give identical results.

Application dependent criteria can be used to further prune the hypotheses $\boldsymbol{\omega}_h^k$. For example, in the case of vehicle recognition, one can assume that vehicles cannot be upside down, hence $|\mathrm{R}_k^h(2,2)| > \eta > 0$, with $\mathrm{R}_k^h(i,j)$, $0 \leq i, j \leq 2$ being the (i,j)th entry of the rotation matrix R_k^h.

6.2 Hypothesis Evaluation Using MLESAC

In Section 6.1, we have shown how alignment hypotheses between the scene and the models can be generated from doublets of matches. We discuss next how these alignment hypotheses can be further pruned using the evidence gathered by using more support in the data. We associate with each alignment hypothesis $\boldsymbol{\omega}_k^h$, a likelihood measure $\mathcal{L}(\Psi \mid \boldsymbol{\omega}_k^h)$, $h = 1, \ldots, H_k$, where H_k is the number of hypotheses generated for model k. Note that not all models will have hypotheses generated, due to the constraints imposed by feature competition in the signature space. The likelihood of hypothesis $\boldsymbol{\omega}_h^k = \{\boldsymbol{\xi}_{j_1}, \boldsymbol{\xi}_{j_2}, \boldsymbol{\xi}_{u_1}^k, \boldsymbol{\xi}_{u_2}^k, \boldsymbol{\theta}_k^h\}$, belonging to model k can be estimated using *joint* 3D-signature space constraints.

For each descriptor match $(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^k)$ we define the indicator variable $\kappa_{j,u}^s$

$$\kappa_{j,u}^s = \begin{cases} 1, & \text{if } (\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^k) \text{ is an inlier in signature space} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

We define $\kappa_{j,u}^o$ an indicator variable specifying whether the origins of the scene descriptor $\boldsymbol{\zeta}_j$ and $\boldsymbol{\xi}_u^k$ are consistent according to the alignment parameters $\boldsymbol{\theta}_k^h$. Let $\hat{\mathrm{o}}_j$ be the warped origin $\boldsymbol{o}_j$ using the rigid transformation $\boldsymbol{\theta}_k^h$, i.e., $\hat{\mathrm{o}}_j = \mathrm{R}_k^h \boldsymbol{o}_j + \boldsymbol{t}_k^h$. The indicator $\kappa_{j,u}^o$ is defined as

$$\kappa_{j,u}^o = \begin{cases} 1, & \text{if} \|\hat{\mathbf{o}}_j - \boldsymbol{o}_u^k\| \leq \Delta_o \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where Δ_o is a user selected threshold that separates the outliers from the inliers. Experimentally, we have determined

that $\Delta_o = 0.5$ m offers a good separation of inliers from the outliers. Similarly to (11), we define

$$\kappa^n_{j,u} = \begin{cases} 1, & \text{if } |\hat{\boldsymbol{n}}_j^\top \boldsymbol{n}_u^k| \geq \Delta_n \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

with $\hat{\boldsymbol{n}}_j = \mathrm{R}_k^h \boldsymbol{n}_j$ and $\Delta_n = 0.9$ a threshold chosen depending on the expected normal dissimilarity. Let

$$\kappa_{j,u} = \kappa^s_{j,u} \kappa^o_{j,u} \kappa^n_{j,u} \tag{13}$$

be an indicator variable that is one if a match is an inlier in the joint 3D-signature space, and zero otherwise. Specifying the thresholds Δ_o and Δ_n which separate the inliers from the outliers in the 3D space is much easier than estimating the threshold Δ_s in the signature space. The threshold Δ_s and the standard deviation σ_s associated with the probability of a scene-model match (7) can be estimated using residuals analysis for the inliers in the joint 3D-signature space. Since we do not know the alignment parameters such a residual analysis cannot be performed. Instead, we define a local scale Δ_{s_j} for each $\boldsymbol{\zeta}_j$ from the approximate nearest-neighbors Ξ_j. In this case, (10) can be expressed quantitatively as $\kappa^s_{j,u} = 1$, iff $u \in I_j$. Gaussian kernels (6) are not robust, hence we employ a truncated kernel to limit the influence of the outliers. The probability of a match conditioned on the alignment $\boldsymbol{\theta}_k^h$ can be written as

$$p(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^k \,|\, \boldsymbol{\theta}_k^h) = \kappa_{j,u} K[\sigma_o^{-1}(\hat{\mathbf{o}}_j - \boldsymbol{o}_u^k)] + (1 - \kappa_{j,u})\upsilon \tag{14}$$

with $\kappa_{j,u}$ defined in (13) and $\upsilon = K\left(\frac{\Delta_o}{\sigma_o}\right)$. In (14), σ_o is the standard deviation of the errors associated with the mismatch of the origins after warping. Under normality assumption, $\|\hat{\mathbf{o}}_j - \boldsymbol{o}_u^k\|^2 \propto \sigma_o^2 \chi_3^2$, where χ_p^2 is a χ^2 distribution with p degrees of freedom. From Δ_o, we choose σ_o such that:

$$\sigma_o^2 = \Delta_o^2 / \chi^2_{3,\beta}, \tag{15}$$

where $\chi^2_{p,\beta}$ is the βth quantile of a χ_p^2 distribution. We have employed $\sigma_o = \Delta_o/3$ which corresponds to using $\chi^2_{3,0.975}$ in (15). The smaller β is, the flatter the distribution of the inliers and the less penalty is given to the mismatches of the origins of the signatures.

The likelihood of correspondences conditioned on the hypothesis $\boldsymbol{\theta}_k^h$ is

$$\mathcal{L}(\boldsymbol{\Psi} \,|\, \boldsymbol{\theta}_k^h) = \prod_{j=1}^{N} p(\boldsymbol{\zeta}_j, \Xi_j \,|\, \boldsymbol{\theta}_k^h). \tag{16}$$

Maximizing $\mathcal{L}(\boldsymbol{\Psi} \,|\, \boldsymbol{\theta}_k^h)$ is equivalent to

$$\max_{h=1,\ldots,H_k} \prod_{j=1}^{N} p(\boldsymbol{\zeta}_j, \Xi_j | \boldsymbol{\theta}_k^h) = \max_{h=1,\ldots,H_k} \prod_{j=1}^{N} \max_{u \in I_j, \nu_u = k} p(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^{\nu_u} | \boldsymbol{\theta}_{\nu_u}^h). \tag{17}$$

An efficient computational scheme for evaluating (17) using *preemption* was proposed in [29]. In a preemption scheme, the hypotheses are evaluated on partial sets of measurements and the worst hypotheses are gradually eliminated. See the referenced paper for more details.

The hypotheses $\boldsymbol{\theta}_k^h$ can be sorted in decreasing order of their likelihood $\mathcal{L}(\Psi \,|\, \boldsymbol{\theta}_k^h)$. In order for a hypothesis to be considered reliable, we enforce two additional consistency constraints: 1) The minimum number of inliers in the scene must be larger than a minimum threshold given as a percentage of the number of scene descriptors N. 2) The overlap area between the warped scene and the hypothesized model must be larger than a minimum coverage area. Since warping can be computationally intensive, we first select the best hypotheses for a model that satisfy 1) and yield distinct alignment hypotheses. The surviving alignment hypotheses are subsequently refined by employing all the inliers in the joint space for pose computation similar to performing one iteration of the ICP algorithm. The condition 2) is verified efficiently by voxelizing the models and counting the percentage of voxels containing warped scene features (3D points or basis points of the surface descriptors).

Let $\boldsymbol{\theta}_k^0$ be the best alignment hypothesis between the scene and model k. Though it is possible to retain multiple alignment candidates for a model, we will assume in the following that only the best hypothesis is retained. Note that only a fraction of the database models will have alignment hypotheses generated that obey all the constraints. Let G denote the indices k of the models that contain valid alignment hypotheses $\boldsymbol{\theta}_k^0$. Ideally, when feature constraints are strong, the cardinality of the set G should be small. The larger the cardinality of G, the more uncertainty there is about the identity of the target.

7 Object Indexing Using MAP Criterion

In Section 6.2, we have discussed how to generate alignment hypotheses between the scene and the models using 3D-signature constraints. We discuss next how the posterior probabilities of a model can be estimated and used in a Maximum A Posteriori (MAP) criterion to index the models from the database.

7.1 Global Signature Scale Estimation

Recall that, in Section 6.2, we have employed the signature constraints by using only the LSH indexed features Ξ_j for each scene descriptor $\boldsymbol{\zeta}_j$ during the likelihood computation. However, the signature similarity scores were not used inside the likelihoods because they require the specification of a threshold Δ_s separating the inliers from the outliers and of a scale σ_s that parameterizes the local distribution of the inliers. Estimating Δ_s and σ_s can be done by analyzing the matches $(\boldsymbol{\zeta}_j, \boldsymbol{\xi}_u^{\nu_u})$ that are inliers in the joint 3D-signature space.

Let Ξ_j^k denote the subset of model features Ξ_j that belong to model k and satisfy the 3D constraints

$$\|\hat{\mathbf{o}}_j - \boldsymbol{o}_u^k\| \leq \Delta_o, \quad |\hat{\boldsymbol{n}}_j^\top \boldsymbol{n}_u^k| > \Delta_n,$$
$$\hat{\mathbf{o}}_j = \mathrm{R}_k^0 \boldsymbol{o}_j + \boldsymbol{t}_k^0, \quad \hat{\boldsymbol{n}}_j = \mathrm{R}_k^0 \boldsymbol{n}_j, \quad u \in J_k,$$

i.e., are inliers in the joint 3D-signature space. Let $q_{j_k} = |\Xi_j^k|$ be the number of inliers belonging to model k corresponding to $\boldsymbol{\zeta}_j$.

The global scale of features σ_s can be obtained by analyzing the signatures between scene descriptors $\boldsymbol{\zeta}_j$ and the inliers Ξ_j^k, $j = 1, \ldots, N$, $k \in \mathrm{G}$. We select σ_s to be the maximum mode of the distribution of the residuals $\|\boldsymbol{x}_j - \boldsymbol{x}_{j_i}^k\|$. The maximum mode can be obtained using mode-seeking techniques such as mean-shift. Experimentally, we have established that for the vehicle database used and the scenes available the distribution of residuals approximates a χ_p^2 distribution, where p, the number of degrees of freedom can be estimated from the rank of the covariance matrix of the residuals.

7.2 Posterior Computation in the Joint Space

The a posteriori probability of a model, given the scene descriptors Z, model descriptors Υ and the alignment parameters Θ^0, of the models $k \in \mathrm{G}$ can be obtained using Bayes rule

$$p(k|\mathrm{Z},\Upsilon,\Theta^0) = \frac{p(\mathrm{Z},\Upsilon|k,\Theta^0)p(k|\Theta^0)}{p(\mathrm{Z},\Upsilon|\Theta^0)} = \frac{p(\mathrm{Z},\Upsilon|k,\Theta^0)p(k)}{p(\mathrm{Z},\Upsilon|\Theta^0)}. \tag{18}$$

We make the simplifying assumption that a priori probability of a model $p(k)$ is uniform, thus

$$p(k\,|\,\mathrm{Z},\Upsilon,\Theta^0) \propto p(\mathrm{Z},\Upsilon\,|\,k,\Theta^0) = p(\mathrm{Z},\Upsilon_k\,|\,\boldsymbol{\theta}_k^0). \tag{19}$$

Assuming independence among the scene feature descriptors, (19) can be written as

$$p(\mathrm{Z},\Upsilon_k\,|\,\boldsymbol{\theta}_k^0) = \prod_{j=1}^{N} p(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0). \tag{20}$$

Using a mixture model, we have

$$p(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0) = \eta_{j,k}p^{(1)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0) + (1-\eta_{j,k})p^{(2)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0), \tag{21}$$

where $\eta_{j,k}$ is the mixing probability of the *inlier* component $p^{(1)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0)$ and $p^{(2)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0))$ is the *outlier* component. Note the relationship between (21) with (14). We have

$$p^{(1)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0) = \sum_{u\in J_k} \rho_{j,u}p^{(1)}(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_u^k\,|\,\boldsymbol{\theta}_k^0), \tag{22}$$

where $\rho_{j,u}$ is an indicator variable that is one if feature $\boldsymbol{\zeta}_j$ is assigned to $\boldsymbol{\xi}_u^k$ and zero otherwise. We enforce the additional constraint that a feature can be assigned to at most one feature from a model, thus $\sum_{u\in J_k}\eta_{j,u} \le 1$. Maximizing (20) implies maximizing $p(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0)$, $j = 1,\ldots,N$ over the space of possible matches. Assuming that the outlier component $p^{(2)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0)$ is uniform and for a given mixing probability $\eta_{j,k}$, we have

$$\begin{aligned}&\max_{u_i\in J_k} p(\boldsymbol{\zeta}_j,\{\boldsymbol{\xi}_{u_1}^k,\ldots,\boldsymbol{\xi}_{u_{N_k}}^k\}\,|\,\boldsymbol{\theta}_k^0) = \\ &\max_{\rho_{j,u}} \sum_{u\in J_k} \rho_{j,u}p^{(1)}(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_u^k\,|\,\boldsymbol{\theta}_k^0) = p^{(1)}(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_{u_0}^k\,|\,\boldsymbol{\theta}_k^0),\end{aligned}$$

where

$$u_0 = \arg\max_{u\in J_k} p(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_u^k\,|\,\boldsymbol{\theta}_k^0) = \arg\max_{u\in J_k} p(\hat{\boldsymbol{\zeta}}_j,\boldsymbol{\xi}_u^k\,|\,\boldsymbol{\theta}_k^0) \tag{23}$$

and $\hat{\boldsymbol{\zeta}}_j = (\hat{\mathbf{o}}_j,\hat{\boldsymbol{n}}_j,\hat{\boldsymbol{x}}_j)$ is the feature $\boldsymbol{\zeta}_j$ warped through the rigid transformation $(\mathrm{R}_k^0,\boldsymbol{t}_k^0)$ corresponding to $\boldsymbol{\theta}_k^0$. Assuming independence between 3D and signature information and Gaussian distribution of residuals

$$p^{(1)}(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_u^k\,|\,\boldsymbol{\theta}_k^0) = K[\sigma_o^{-1}(\hat{\mathbf{o}}_j - \boldsymbol{o}_u^k)]K[\sigma_s^{-1}(\boldsymbol{x}_j - \boldsymbol{x}_u^k)]. \tag{24}$$

Evaluating (23) is linear in the number of features for a model, thus to be computationally efficient we restrict the computation on the subset $\Upsilon_{k,j} = \{\boldsymbol{\xi}_{u_1}^k,\ldots,\boldsymbol{\xi}_{u_{N_{k,j}}}^k\}$, $N_{k,j} = |\Upsilon_{k,j}|$ satisfying

$$\begin{aligned}&\|\hat{\mathbf{o}}_j - \boldsymbol{o}_{u_i}^k\| \le \Delta_o, \quad |\hat{\boldsymbol{n}}_j^\top \boldsymbol{n}_{u_i}^k| \ge \Delta_n,\\ &\|\boldsymbol{x}_j - \boldsymbol{x}_{u_i}^k\| \le \Delta_s, \quad u = 1,\ldots,N_{k,j}.\end{aligned} \tag{25}$$

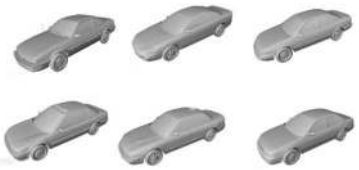


Fig. 5. Example of several similar vehicles employed in our database. Top row: Chevrolet Cavalier 1993, Dodge Avenger 1997, Ford Taurus 1994; Bottom row: Honda Accord 1990, Mitsubishi Galant 1992, Nissan Altima 1993.

The outlier component is, assuming whitening of the 3D and signature components,

$$p^{(2)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0) = \frac{\Gamma(\frac{5}{2})\Gamma(\frac{p+2}{2})}{\pi^{\frac{p+3}{2}}(\chi_{3,\beta}^2)^{3/2}(\chi_{p,\beta}^2)^{p/2}}, \tag{26}$$

where p is the number of degrees of freedom of the signature residuals between the scene and the models and $\Gamma(\cdot)$ is the gamma function.

The mixing probabilities $\eta_{j,k}$ can be found using the EM algorithm as suggested in [38]. Let $y_{j,k}$ denote an indicator variable that is one when scene feature $\boldsymbol{\zeta}_j$ is assigned to model k. We have:

$$\begin{aligned}&p(y_{j,k}=1\,|\,\eta_{j,k})\\ &\quad= \frac{\eta_{j,k}p^{(1)}(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_{u_0}^k\,|\,\boldsymbol{\theta}_k^0)}{\eta_{j,k}p^{(1)}(\boldsymbol{\zeta}_j,\boldsymbol{\xi}_{u_0}^k\,|\,\boldsymbol{\theta}_k^0) + (1-\eta_{j,k})p^{(2)}(\boldsymbol{\zeta}_j,\Upsilon_k\,|\,\boldsymbol{\theta}_k^0)}.\end{aligned} \tag{27}$$

The refined $\eta_{j,k}$ can be found by taking the expected values of $p(y_{j',k}=1\,|\,\eta_{j',k})$ for the features $\boldsymbol{\zeta}_{j'}$ within a radius ϵ of $\boldsymbol{\zeta}_j$, $\|\boldsymbol{o}_{j'} - \boldsymbol{o}_j\| \le \epsilon$. In our algorithm, we start with $\eta_{j,k} = 1/2$ and perform one EM iteration to find the mixing probabilities.

To remove the scale, we ensure that

$$\sum_{k=1}^{M} p(k|\mathrm{Z},\Upsilon,\Theta) = 1, \tag{28}$$

and the indexed models $\mathcal{I} = \{\mu_{k_1},\ldots,\mu_{k_T}\}$ are returned such that

$$\begin{aligned}&\sum_{i=1}^{T} p(\mu_{k_i}|\mathrm{Z},\Upsilon,\Theta^0) > \gamma,\\ &p(\mu_{k_1}|\mathrm{Z},\Upsilon,\Theta^0) \ge \ldots \ge p(\mu_{k_M}|\mathrm{Z},\Upsilon,\Theta^0),\end{aligned} \tag{29}$$

where γ is the confidence level sought in the indexing decision. In practice, we limit the number of indexed models T to the minimum between the value obtained in (29) and T_0 selected as a fraction of the number of models M from the database.

8 Experimental Results

8.1 Experimental Settings

We have applied the 3D object indexing method proposed on vehicle indexing, a very challenging problem due to the high degree of similarity existing between different types of vehicles. In Fig. 5, several models used in our database are presented (the models are displayed in faceted form for clarity of presentation; recall that we do not use the faceted

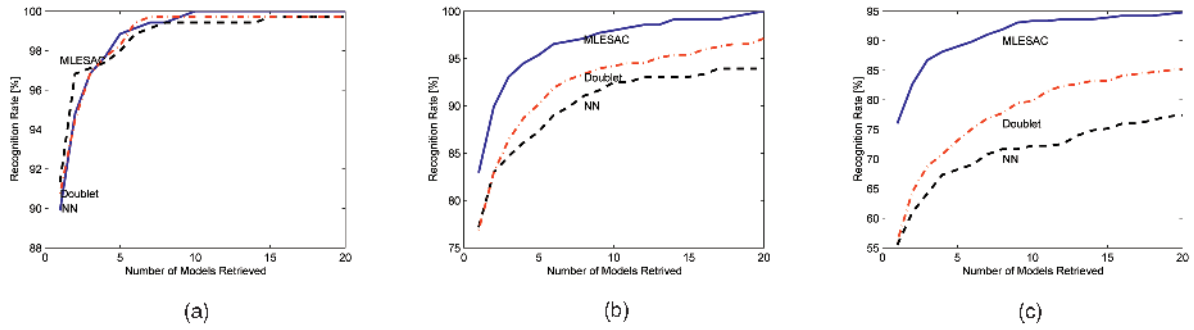


Fig. 6. Average recognition rate versus the number of models retrieved for the synthetic example. Normal noise with different standard deviation σ was added along the viewing viewing direction. (a) $\sigma = 5$ cm, (b) $\sigma = 10$ cm, and (c) $\sigma = 15$ cm. Dashed line denotes the performance of the 1-NN indexer in which the neighbors are computed using the LSH algorithm. Dash-dotted line curve denotes the performance of a doublet-based indexer, while the continuous line denotes the performance of the proposed algorithm.

models directly into our system). Note the similarity between their 3D shapes.

Most 3D object recognition results published in the literature employed complete scans of small objects to produce models that were subsequently used to recognize the same instances from scenes. Therefore, there was little, or no difference between the objects and the corresponding models from the database apart from changes induced by limited views of the objects available, obscuration produced by clutter, etc. Obtaining complete scans of real vehicles can be very difficult due to the large size of the objects requiring expensive specialized hardware. Moreover, one cannot make the assumption that the same sensor will be used to acquire the scene data, thus differences in resolution and quality of the data can be expected between the objects and the models from the database.

Three-dimensional models of large objects are produced from high quality 2D images using specialized software and expert users to reconstruct the 3D structure of the objects. Though faithful replicas of the original objects can be produced, not all the characteristics of the real objects can be modeled. For example, the 3D models can be opaque, thus the interiors of the vehicles can be missing, whereas in real operating conditions, laser scanners can acquire interiors by penetrating through the windows. Unmodeled articulations of parts can reveal new structures in the real objects not present in the corresponding model. For example, an opened hood for a car may reveal the engine block, which is not necessarily present in the model, even if the model has articulating parts.

The experimental results presented next employed vehicle models that were provided through a research contract. Some of the models provided have articulations, however for the indexing we used models placed in a default configuration. We have employed other sources for 3D models such as *De Espona* or *Viewpoint*, however, the results are not reported here. The experiments were performed on a PC with 2 GHz Intel processor and with 2 GB of memory.

The 3D models employed consist of surface meshes that are given in standard formats (VRML, 3D Studio Max). For each model, we have rendered a total of eight views sampled at 45 degrees azimuth angles around the object and at an elevation close to the expected elevation of the sensor in a particular experimental setting. For instance, airborne 3D sensors can acquire the data at elevations close to 45 degrees, while ground 3D sensors will acquire the data at very small elevations, below 20 degrees.

Between 2,000 and 3,000 spin-images per model were generated, depending on the size of the object, by uniformly sampling locations from the combined point clouds. Spin-images were computed using 10×10 signature histogram and with a radius $r = 1.5$ m. We have experimentally found that larger support regions for the features yield a worse performance due to the increased effect of obscuration and clutter. The bin size selected for the features offers a good balance between the tolerance to noise and the corresponding discriminant power.

The performance of the proposed indexer was compared with two other methods: 1) 1-NN indexer using the LSH for finding the nearest neighbor model feature to each scene feature. The posterior of each model k, $p_{1-NN}(k|\mathrm{Z}, \Upsilon)$, is computed as the ratio between the number of scene features having the nearest neighbor model feature coming from model k and the total number of scene features. 2) A doublet-based indexer. Random doublets are generated as discussed in Section 6.1. The posterior $p_{doublet}(k|\mathrm{Z}, \Upsilon)$ can be computed by the ratio between the number of doublets that come from model k obeying the geometric constraints from Fig. 3 and the total number of geometrically consistent doublets that were generated.

The effect of noise affecting the measurements was analyzed using synthetic data generated using a scene generator which places models into a virtual environment. We have employed a model database containing 89 vehicle models: 54 civilian vehicles and 35 military vehicles. The database consists of more than 180,000 features. Scene point clouds are rendered using a realistic sensor simulator and variable noise levels. We have generated hundreds of queries and the plots of the average recognition rate versus the number of models retrieved are presented in Fig. 6. Note that at small noise all three indexers have a comparable performance, however, at higher levels of noise the 1-NN and the doublet-based indexers break down.

8.2 *HighLift* Data Collection

In the *Highlift* data collection, a laser scanner is placed on a cherry picker at a height of approximately 50 m above the ground. Due to the relatively low height of the sensor with respect to the ground, the 3D measurements are acquired at a low elevation angle $e = 10^\circ - 20^\circ$. Each scene consists of multiple vehicles with different articulations present, such as

Fig. 7. Examples from the 88 queries from *Highlift* collection used for testing.

doors and hoods opened. The effective elevation for a target depends on the distance between the target and the sensor.

A total of 88 queries, covering at least three sides of the target, were selected from the *Highlift* data collection and used for testing the 3D object indexing algorithm proposed. Each query contains a single target object in addition to clutter. Examples from the 88 queries are given in Fig. 7. Although three sides of the targets should be visible, due to calibration errors of the sensor, low elevation angle, partial obscuration, some data from the object surfaces may be missing. We define the obscuration level of the data by the percentage of the missing data with respect to the ideal viewing conditions. Thus, 0 percent obscuration indicates that all three vehicle sides are visible, while 20 percent obscuration means that 20 percent of the three sides are missing. The obscuration level is evaluated manually. Highly obscured query data which are not used for evaluation because they lack distinctive features.

We have employed a model database containing 89 vehicle models consisting of 180,000 features (same as used for the synthetic example above).

The accuracy and speed of the LSH method depends on two parameters: K (size of the first-level hash code) and L (number of hash tables). To find the best values for the K and L, we have employed a method similar to [12]. For a scene feature i we compute the nearest neighbor using a sequential search and the LSH method. Let d_i^{NN} denote the L_2 distance between the signatures of the scene and the nearest neighbor obtained with a sequential method and d_i^{LSH} the distance between the signatures of the scene feature and of the LSH indexed model feature. For a given K and L the following criterion is sought to be minimized:

$$\mathcal{J}(K,L) = \frac{1}{m}\sum_{i}^{m}\frac{d_i^{LSH}}{d_i^{NN}}, \tag{30}$$

where m is the number of features used for optimization. To speed up finding the minimum of (30), we evaluate the cost function at coarse values for K and L. For K, we use a range of $[20;200]$ with an increment $\Delta_K = 5$, while for $L \in [2;30]$ with step $\Delta_L = 2$ resulting in 585 LSH tables being generated. Twenty-seven queries were selected from the *Highlift* data collection, covering a large spectrum of the vehicles encountered. For each target, around 500 features were generated resulting in $m = 13,500$ scene features generated and used for LSH optimization.

From Fig. 8, it can be seen that a larger number of hash tables L and using smaller K the error (30) produced by LSH is reduced, however, the computational time required is increased. This effect can be understood with respect to the probability of collision (4). For small K, we will have a lot of collisions in creating the LSH tables, therefore, the number of possible neighbors returned for a query point will increase, thus also the chance that one of them will be very close to the correct nearest neighbor. Increasing K, on the other hand, will result in less collisions, thus fewer possible neighbors returned and faster execution time. However, the error measure (30) may also increase, unless the number of tables L is also increased due to the fact that many of the hash tables will be empty (for very low collision probability, even close neighbors will end up in separate hash tables). The effect of L on the error and execution time is evident. The larger the number of tables L is, the slower the algorithm will typically be, because more data points are retrieved.

We seek solutions for the parameters such that $\mathcal{J} \leq 1+\epsilon$, were ϵ denotes the level of errors that is tolerated. From all the solution obeying the error constraints, we select the ones resulting in the fastest processing time. In Table 1, we have displayed the best projected processing time depending on the level of errors in the neighbors returned by LSH. We have selected $K = 65$ and $L = 18$ resulting in a speedup of almost 20 times compared with using the sequential search within the database.

The average timing for the *Highlift* collection is 50 seconds per query. Note that a significant amount of time is spent in generating doublets and computing the approximate nearest neighbors using LSH. A large number of doublets are simply

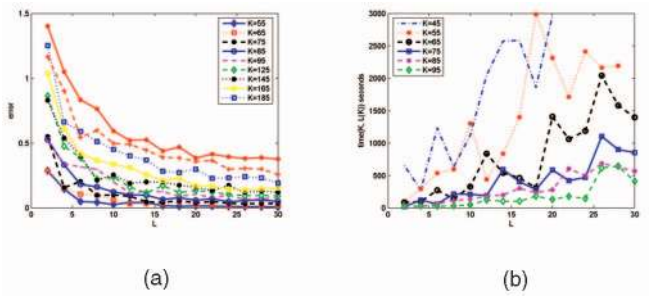


Fig. 8. Optimization of the LSH parameters K and L. (a) The error function $\mathcal{J}(L)$ computed for fixed K. (b) Computational time required in seconds.

TABLE 1
LSH Error ϵ versus Computation Time for the *Highlift* Data Collection

ϵ [%]	Projected Run Time (sec.)	K	L	LSH vs. NN Speed Up
20	5.82	95	10	120
15	8.52	105	12	82
10	16.76	65	8	42
5	37.31	65	18	19
3	79.60	60	16	9
2	149.92	50	12	4.7

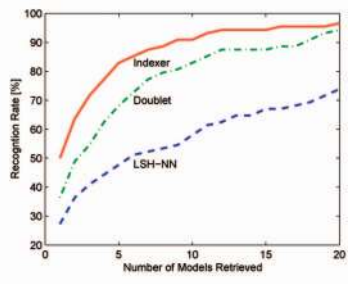


Fig. 9. Plots of the average recognition rates function of the number of models retrieved from the database for the 88 queries selected from the *Highlift* data collection. The dashed line denotes the performance using 1-NN indexer computed with LSH, the dashed-dotted line describes the doublet-based indexer, while the continuous line denotes the proposed indexer.

discarded because they do not obey the geometric constraints due to the significant target and corresponding model differences.

The average recognition rates versus the number of models retrieved for three indexing algorithms using the 88 *Highlift* queries are shown in Fig. 9. With $T_0 = 20$ model picks, the indexing performance is 96.59 percent on the qualified queries and 87.06 percent over the entire collection (including discarded queries). Note the significant improvement of the proposed method.

8.2.1 Influence of Obscuration

To investigate thoroughly the influence of various operating conditions such as obscuration, clutter, and articulations a very large number of queries would be required. Obscuration is a very important factor that can affect drastically the performance of any recognition system. A systematic evaluation of the indexing algorithm proposed under different types and levels of obscuration is presented in the following. Since data acquisition can be very expensive and time consuming, we have employed an obscuration simulator using real queries generated in the *Highlift* data collection.

In practical situations, two types of obscuration present interest: 1) solid obscuration, that is produced by solid objects occluding the target, such as walls, or other objects; 2) "see-through" obscuration which is the produced by vegetation (bushes, foliage of trees). The former type affects only few large regions from the target, while the latter is more localized and affects more locations in the target. As before, we define the level of the obscuration with respect with the ideal viewing conditions when no occlusion is present. The only assumption we make is that under these ideal viewing conditions, three sides of the vehicles are available on average. To account for the sensor resolution, the scene is voxelized and the obscuration level measured by the percentage of the voxels containing no measurements with respect to the ideal viewing conditions.

To simulate see-through obscuration, we remove small regions from the queries throughout the surface. Given the level of obscuration desired and a radius of the regions to be removed the simulator selects randomly voxelized locations from the query and removes the measurements within a radius $R_o < 0.5$ m. The process is repeated until the obscuration level sought is achieved. Under a given radius R_o, a large level of obscuration results in more holes in the data.

TABLE 2
Indexing Performance for Different Levels of See-Through Obscuration and $T_0 = 20$ Models Returned

Obsc. Lev.	$R_0 = 0.10$m	$R_0 = 0.25$m	$R_0 = 0.50$m
10 – 25% (+10%)	95.45	93.18	90.91
20 – 35% (+20%)	94.32	89.77	86.31
30 – 45% (+30%)	93.18	87.50	76.14
40 – 55% (+40%)	88.64	80.68	69.32
50 – 65% (+50%)	83.18	72.73	45.45

TABLE 3
Indexing Performance for Different Levels of Solid Obscuration and $T_0 = 20$ Models Returned

Obsc. Lev.	$R_0 = 1.0$m	Plane Sweeping
10 – 25% (+10%)	96.59	93.18
20 – 35% (+20%)	94.32	87.50
30 – 45% (+30%)	86.36	82.95
40 – 55% (+40%)	75.00	79.55
50 – 65% (+50%)	61.36	65.91

Solid obscuration can be generated using the previous see-through simulator, but with larger radius $R_o \geq 1$ m, or by sweeping the targets along the X, Y, Z directions until the level of the obscuration desired is reached.

The performance on see-through obscuration is relatively high for up to 35 percent of obscuration. The performance deteriorates more gracefully as obscuration level increases, compared to the performance of solid type obscurations, as shown in Tables 2 and 3. This indicates that the indexer can perform well in actual sensing environment where targets hiding behind foliages and "mouse-bite" type of data may be collected by modern sensing technologies.

8.3 *Montana* Data Collection

In the *Montana* data collection, we have available a number of 536 queries which were selected to ensure at least three sides of the vehicles were present, as discussed in Section 8.2. The data was acquired by placing a LADAR sensor into a helicopter which scans the targets in multiple passes. All the scans are registered using GPS information associated with each point. The noise affecting the measurements is higher compared with the *Highlift* data collection due to the vibrations of the aircraft. We do not have precise information about the noise level of the LADAR sensors used, however, we have estimated that the equivalent standard deviation of the noise is 10 cm for the *Montana* data, compared with 7.5 cm for the *Highlift* collection. The viewing elevation angle is close to 45 degrees which ensures that the top of the objects are better scanned compared with the *Highlift* collection. As before, the targets present can have articulations, the interiors can be visible, clutter and ground are present. An illustration of few queries from the *Montana* data collection is given in Fig. 10.

We have employed a database of 366 models consisting of 201 sedans, 53 SUV, jeep and wagons, 17 minivans,

Fig. 10. Examples of queries from the *Montana* collection. Queries correspond to: Jeep M151, GMC Suburban 1989, and Mazda 626. Note the existence of significant articulations, nearby clutter affecting the target.

12 buses and vans, nine construction vehicles, 47 trucks and pickups, and 27 military vehicles. The database consisted of more that 1,000,000 features. The models used for the *Highlift* data collection are a subset of the models used for the *Montana* experiment. The database was extended to show both the scalability of the algorithm proposed and also due to the larger variety of models encountered in this data collection. The curve of recognition rates function of the number of models retrieved for the proposed method is presented in Fig. 11. The average time per query is about 100 seconds on a 2 GHz machine. It can be seen that at a four fold increase in the database, the processing time is doubled compared with the time used for the previous example. The error is 5 percent with 60 models returned, three times as many as for the *Highlift* experiment. The relatively large number of models returned (up to one sixth of the database size, depending on the discriminability of a particular model), is due to the high number of similar models existing in this database (more than 200 sedans) and the noise affecting the measurements.

To our best knowledge, no result published in the literature reported results on such a large number of similar models in the database and with so many queries used for testing.

9 Conclusion

We have described a new 3D object indexing method that employs approximate nearest neighbor search using the Locality Sensitive Hashing and the joint 3D-signature estimation for generation and evaluation of alignment hypotheses between scene and database models. We have applied the approach for a challenging problem of recognizing vehicles, a very difficult problem due to the high degree of similarity between the objects. We have shown that our algorithm can cope with challenging 3D data covering only portions of targets under challenging conditions due to: obscuration of the data, discrepancies between scenes, and available models.

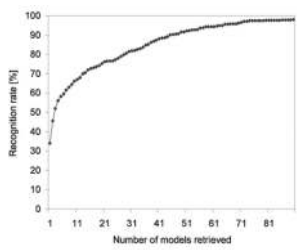


Fig. 11. Recognition rates versus the number of models retrieved curve of the proposed indexer using the *Montana* collection.

Acknowledgments

All vehicle data used in this research project were collected in government-sponsored, planned data collections on military land. All commercial vehicles were unowned, rented vehicles. The research purpose is to develop techniques to discriminate between very similar military targets. Commercial vehicles provide a level of similarity more challenging that military vehicles (tanks, etc.). Thus, use of commercial vehicles in this project provide a more challenging technical problem and are less costly to obtain or logistically move than military vehicles.

References

[1] K. Arun, T. Huang, and S. Blostein, "Least-Squares Fitting of Two 3D Point Sets," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, pp. 698-700, 1987.

[2] M. Bawa, T. Condie, and P. Ganesan, "LSH Forest: Self-Tuning Indexes for Similarity Search," *Proc. 14th Int'l Conf. World Wide Web,* pp. 651-660, 2005.

[3] P. Besl and N. McKay, "A Method for Registration of 3D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 540-547, 1992.

[4] R. Campbell and P. Flynn, "A Survey of Free-Form Object Representation and Recognition Techniques," *Computer Vision and Image Understanding,* vol. 81, pp. 166-210, 2001.

[5] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng, "Ransac-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 10, pp. 1229-1234, Oct. 1999.

[6] C.-S. Chua and R. Jarvis, "Point Signatures: A New Representation for 3D Object Recognition," *Int'l J. Computer Vision,* vol. 25, no. 1, pp. 63-85, 1997.

[7] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry.* Springer, 2000.

[8] H. Delingette, M. Hebert, and K. Ikeuchi, "A Sperical Representation for the Recognition of Curved Objects," *Computer Vision and Pattern Recognition,* pp. 103-112, May 1993.

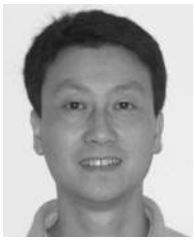
[9] C. Dorai and A. Jain, "Cosmos—A Representation Scheme for 3D Free-Form Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 1115-1130, 1997.

[10] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, 1981.

[11] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing Objects in Range Data Using Regional Point Descriptors," *Proc. European Conf. Computer Vision,* May 2004.

[12] B. Georgescu, I. Shimshoni, and P. Meer, "Mean Shift Based Clustering in High Dimensions: A Texture Classification Example," *Proc. Int'l Conf. Computer Vision,* pp. 456-463, 2003.
[13] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," *Proc. 25th Int'l Conf. Very Large Data Bases,* pp. 518-529, 1999.
[14] W. Grimson and D. Huttenlocher, "On the Sensitivity of Geometric Hashing," *Proc. Int'l Conf. Computer Vision,* pp. 334-338, 1990.
[15] C.J. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Fourth Alvey Vision Conf.,* pp. 147-151, 1988.
[16] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert, "Part-Based 3D Object Classification," *Computer Vision and Pattern Recognition,* vol. 2, pp. 82-89, June 2004.
[17] P. Indyk and R. Motwani, "Approximate Nearest Neighbor—Towards Removing the Curse of Dimensionality," *Proc. 30th Symp. Theory of Computing,* 1998.
[18] A. Johnson and M. Hebert, "Surface Matching for Object Recognition in Complex Three-Dimensional Scenes," *Image and Vision Computing,* vol. 16, pp. 635-651, 1998.
[19] A. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 433-449, 1999.
[20] A. Johnson, R. Hoffman, J. Osborn, and M. Hebert, "A System for Semi-Automatic Modeling of Complex Environments," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling,* pp. 213-220 May 1997.
[21] Y. Keselman, A. Shokoufandeh, M.F. Demirci, and S. Dickinson, "Many-to-Many Graph Matching via Metric Embedding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.
[22] Y. Lamdan and H. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proc. Int'l Conf. Computer Vision,* pp. 238-249, 1988.
[23] Y. Lamdan and H. Wolfson, "Affine Invariant Model-Based Object Recognition," *IEEE Trans. Robotics and Automation,* vol. 6, no. 5, pp. 578-589, 1990.
[24] D.G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. Int'l Conf. Computer Vision,* pp. 525-531, 1999.
[25] B. Matei and P. Meer, "A General Method for Errors-in-Variables Problems in Computer Vision," *Proc. Computer Vision and Pattern Recognition Conf.,* vol. 2, pp. 18-25, June 2000.
[26] G. Medioni, M. Lee, and C. Tang, *A Computational Framework for Feature Extraction and Segmentation.* Elsevier Science, 2000.
[27] G. Mori, S. Belongie, and J. Malik, "Shape Contexts Enable Efficient Retrieval of Similar Shapes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 723-730, 2001.
[28] S. Nene and S. Nayar, "A Simple Algorithm for Nearest-Neighbor Search in High Dimensions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 989-1003, 1997.
[29] D. Nister, "Preemptive RANSAC for Live Structure and Motion Estimation," *Proc. Int'l Conf. Computer Vision,* pp. 199-206, 2003.
[30] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions," *ACM Trans. Graphics,* vol. 21, no. 4, pp. 807-832, 2002.
[31] I. Rigoutsos and R. Hummel, "A Bayesian Approach to Model Matching with Geometric Hashing," *Computer Vision and Image Understanding,* vol. 61, no. 7, pp. 11-26 1995
[32] S. Ruiz-Correa, L. Shapiro, and M. Melia, "A New Signature-Based Method for Efficient 3-D Object Recognition," *Computer Vision and Pattern Recognition,* vol. 1, pp. 769-776, 2001.
[33] S. Ruiz-Correa, L. Shapiro, and M. Miela, "A New Paradigm for Recognizing 3D Object Shapes from Range Data," *Proc. Int'l Conf. Computer Vision,* pp. 1126-1133, 2003.
[34] Y. Shan, H. Sawhney, S.S.B. Matei, and R. Kumar, "Partial Object Matching with Shapeme Histograms," *Proc. European Conf. Computer Vision,* vol. 3, pp. 442-455, 2004.
[35] Y. Shan, B. Matei, H.S. Sawhney, R. Kumar, D. Huber, and M. Hebert, "Linear Model Hashing and Batch RANSAC for Rapid and Accurate Object Recognition," *Computer Vision and Pattern Recognition,* 2004.
[36] G.C. Sharp, S.W. Lee, and D.K. Wehe, "ICP Registration Using Invariant Features," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 90-102, Jan. 2002.
[37] F. Stein and G. Medioni, "Structural Indexing: Efficient Three-Dimensional Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 125-145, Feb. 1992.
[38] P. Torr and A. Zisermann, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding,* vol. 78, pp. 138-156, 2000.
[39] S. Yamany and A. Farag, "Surface Signatures: An Orientation Independent Free-Form Surface Representation Scheme for the Purpose of Object Registration and Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1105-1120, Aug. 2002.
[40] D. Zhang and M. Hebert, "Harmonic Maps and Their Applications in Surface Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1999.

Bogdan Matei received the Dipl Engn and MSc degrees in electrical and computer engineering from the Polytechnic University of Bucharest, Romania in 1994 and 1995, respectively, and the PhD degree in electrical engineering from Rutgers University in 2001. From 1994 to 1997, he was with the Department of Electronics and Telecommunications at Polytechnic University of Bucharest as a teaching fellow and research associate. In 1994, he was awarded a TEMPUS scholarship funded by the European Union at Polytechnics of Turin, Italy to research optical character recognition using neural networks. He had received research fellowships at the Technical University of Darmstadt, Germany in 1995 and 1996. From 1997 until 2001, he was with the Center for Advanced Information Processing affiliated with Rutgers University, researching the application of modern statistical methods for optimal parameter estimation under heteroscedastic noise and performance evaluation of computer vision algorithms. From 2001 until the present, he has been with the Vision Technologies Group at the Sarnoff Corporation, first as a member of the technical staff and lately as a senior member of the technical staff. His research interests include object recognition, statistical learning, video-based aerial, and ground autonomous navigation, real-time aerial video surveillance, and georegistration. He has published more than 15 papers in peer-reviewed journals and conference proceedings and holds multiple US and international patents published or pending. He was in the program committee at numerous international conferences and workshops such as CVPR, ICCV, and ECCV. He received the Best Student Paper award at the IEEE Computer Vision and Pattern Recognition Conference in 1999. He is a member of the IEEE.



Ying Shan received the BE degree in chemical engineering, focusing on automatic process control, from Zhejiang University, P.R. China in 1990, and the MS and PhD degrees in computer science from Shanghai Jiaotong University, P.R. China in 1993 and 1997, respectively. Dr. Shan is currently a senior member of the technical staff in Sarnoff Corporation's Vision and Learning Laboratory. His research interests include computer vision, object recognition, machine learning, and computer graphics, with applications in video understanding, video data mining, video surveillance, 3D object and face modeling, 3D data mining, and 2D/3D image registration. From 1996 to 1997, he was a research assistant at Hong Kong Polytechnic University, Hong Kong. From 1997 to 1999, he was a postdoctoral fellow at Nanyang Technological University, Singapore and from 1999 to 2001, he was a postdoctoral researcher at Microsoft Research. Since he joined Sarnoff Corporation as a member of the technical staff in 2001, he has initiated, led, and contributed to a number of government and commercial projects that have been successfully delivered. He has published more than 25 peer-reviewed papers, holds eight US patents, and has 15 others pending. He is an active reviewer of top journals and conferences such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, the *International Journal on Computer Vision*, and SIGGRAPH. He was on the program committee of numerous international conferences such as ACCV, ICPR, ECCV, CVPR, and ICCV. He was the recipient of Sarnoff's Recognition Award in 2003, Innovation Award in 2003, 2004, and 2005. Dr. Shan is a senior member of IEEE.

Harpreet S. Sawhney received the PhD degree in computer science in 1992 from the University of Massachusetts, Amherst, focusing on computer vision. He is the technical director of the Vision & Learning Technologies Lab. at the Sarnoff Corporation. His areas of interest are object recognition, motion video analysis, 3D modeling, vision and graphics synthesis, video enhancement, video indexing, data mining, and compact video representations. Since 1995, he has led government and commercial programs in immersive telepresence, image-based 3D modeling, video object fingerprinting, video mosaicing, georegistration, 2D and 3D video manipulation, and object recognition. Dr. Sawhney was one of the key technical contributors toward the founding of two Sarnoff spinoffs, VideoBrush Inc., and Lifeclips Inc. From 1997 to 2004, he was awarded the Sarnoff Technical Achievement Awards seven times for his contributions in video mosaicing, video enhancement, 3D vision and immersive telepresence. From 1992 to 1995, he led video annotation and indexing research at the IBM Almaden Research Center in San Jose, California. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*). He has also served on the program committees of numerous computer vision and pattern recognition conferences. He has published more than 60 papers and holds 15 patents. He is member of the IEEE.

Yi Tan received the MS degree in electrical engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1985, and the PhD degree in electrical and computer engineering from Rutgers University, New Brunswick, New Jersey, in 1994. From 1985 to 1988, he was a faculty member at the VLSI CAD center in the Hangzhou Institute of Electronics Engineering, Hangzhou, China. From 1990 to 1994, he was a graduate research assistant at the Center of Advanced Information Processing (CAIP) at Rutgers University. During the summers of 1991 and 1992, he was a R/D intern with Phillips North American Research Laboratories, Briarcliff, New York. From 1994 to 2002, he was a senior engineer at Princeton Video Imaging, Inc., Princeton, New Jersey, where he played the key role in developing the Emmy-winning virtual imaging systems which are widely used in TV virtual advertising and TV sports program enhancement such as the virtual first-down line. Since 2002, he has been a member of technical staff in vision laboratories at Sarnoff Corporation, Princeton, New Jersey. His current research interests include computer vision, multimedia, video and image processing, motion tracking, and pattern recognition on 2D & 3D data. He is a senior member of the IEEE and the IEEE Computer Society.

Rakesh "Teddy" Kumar received the BTech degree in electrical engineering from IIT-Kanpur, the MS degree in Electrical and Computer Engineering from the State University of New York, Buffalo, and he received the PhD degree in computer science from the University of Massachusetts at Amherst in 1992. He is currently the technical director of the Vision and Robotics Laboratory at Sarnoff Corporation, Princeton, New Jersey. Prior to joining Sarnoff, he was employed at IBM. His technical interests are in the areas of computer vision, computer graphics, image processing and multimedia. At Sarnoff, he has been directing and performing commercial and government research and development projects in the areas of video surveillance and monitoring, video and 3D exploitation and analysis, object recognition, immersive telepresence, 3D modeling, medical image analysis, and multisensor registration. He was one of the principal founders from Sarnoff for multiple spin-off and spin-in companies: VideoBrush, LifeClips, and Pyramid Vision Technologies. From 1999 to 2003, he was an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He has served in different capacities on a number of computer vision conferences and US National Science Foundation review panels. He has coauthored more than 40 research publications and has received over 18 patents, with numerous others pending. He is a member of the IEEE.

Daniel Huber received the BS degree in electrical engineering from the University of Texas, the MS degree in computer science from Stanford, and the PhD degree in robotics from Carnegie Mellon University in 2002. Dr. Huber is currently a systems scientist in the Vision and Mobile Robotics Laboratory at Carnegie Mellon's Robotics Institute. His research interests focus on three dimensional computer vision, including modeling from reality, object recognition, and real-time 3D systems. He is currently developing 3D vision-based medical applications and a system to aid in detecting defects on construction sites. He has served as a program committee member for several IEEE conferences, including computer vision and pattern recognition (CVPR) and three-dimensional imaging and modeling (3DIM). He is a member of the IEEE.

Martial Hebert is a professor at the Robotics Institute, Carnegie Mellon University. His current research interests include object recognition in images, video, and range data, scene understanding using context representations, and model construction from images and 3D data. His group has explored applications in the areas of autonomous mobile robots, both in indoor and in unstructured, outdoor environments, automatic model building for 3D content generation, and video monitoring. He has served on the program committees of the major conferences in the computer vision area. He is a member of the IEEE.