Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition

Yu Zhong
AIT, BAE Systems
6 New England Executive Park
Burlington, MA 01803-5012 USA
yu.zhong@baesystems.com

Abstract

This paper presents a new approach for recognition of 3D objects that are represented as 3D point clouds. We introduce a new 3D shape descriptor called Intrinsic Shape Signature (ISS) to characterize a local/semi-local region of a point cloud. An intrinsic shape signature uses a view-independent representation of the 3D shape to match shape patches from different views directly, and a view-dependent transform encoding the viewing geometry to facilitate fast pose estimation. In addition, we present a highly efficient indexing scheme for the high dimensional ISS shape descriptors, allowing for fast and accurate search of large model databases. We evaluate the performance of the proposed algorithm on a very challenging task of recognizing different vehicle types using a database of 72 models in the presence of sensor noise, obscuration and scene clutter.

1. Introduction

3D object representation and matching has been a fundamental part of computer vision research with applications in modeling, visualization, recognition, classification, and scene understanding [3][4][12]. Descriptor based 3D recognition algorithms, which match two objects by matching local, semi-local regions or surfaces, have emerged as promising solutions for real world 3D applications due to their robustness to clutter and obscuration [5][6][7][8][9][14][18][20][21][23].

The pioneering work on shape descriptors by Stein and Medioni [21] used local surface normals called "splashes" together with the curvature and "torsion" information of 3D curves to match 3D objects. Subsequently, the methods of "point-signatures" [5] and "principle curvature", proposed by Chua and Jarvis, were also used to match local surfaces of 3D objects. The popular "spin images (SI)" were introduced by Johnson and Hebert [14] to recognize 3D surface meshes. The "spin image" at a vertex is a 2D histogram of the counts of 3D points in a surrounding supporting volume, established using the local surface normal as a reference for invariance. The spin images can be computed for local or semi-local surface regions, making it flexible in compromising between the locality for robustness to occlusion/clutter, and the requirement of sufficient surface coverage for increased discriminating power. However, the discriminating power of SI may be limited due to the possible loss of valuable differentiating 3D shape information when it is collapsed into a 2D histogram.

Frome et al. [7] first used features preserving the 3D information for improved shape discrimination by extending the idea of 2D shape contexts to 3D. A 3D shape context (3DSC) at an oriented basis point is a 3D occupational histogram of the data points in a surrounding support sphere, with its north pole aligned to the surface normal. However, given only the surface normal as a reference, there is a gauge of freedom in the rotation around the axes that needs to be eliminated in order to define the 3D histogram. This problem is worked around by uniformly sampling the reference rotation angle and computing one feature vector for each sample. This handling of the free rotation multiplies the computational and storage cost, and decreases the recognition performance due to the limited sampling of the rotation parameter. Mian et al. [15] also used feature descriptors maintaining 3D shape information to match surface meshes. They defined a 3D reference frame for a pair of oriented points (a vertex and its surface normal), and then computed a "tensor", which is a Cartesian partition of the cubic volume centered at the origin of the defined frame. The shape feature consists of the intersected object surface area in each bin. The drawback of this approach is the combinatorial exploration of vertex pairs required to define a local frame.

Most recently orthonormal frames using the principle component space of local neighborhood was used by Taati et al. [22] to extract invariant 3D property features. Mian et al. [16] also enhanced the "3D Tensors" using such reference frames to rid of the combinatorial exploration of vertex pairs. We have independently [24] used eigen analysis of local points to establish invariant reference to encode shape features. Different from [16] and [22], we noted that there are a total number of four distinguishing frames that can be derived from the PCA subspaces, with each frame transforming into another by rotating along one of its axis by 180^o. There is a need to disambiguate these symmetries in order to compute highly discriminative shape descriptors.

With the limitations of existing approaches in mind, we introduce a new 3D shape representation method called Intrinsic Shape Signatures (ISS) for 3D object recognition which is designed to be stable, repeatable, informative, and discriminative, ensuring highly accurate 3D shape matching and recognition. In the paper we make the following contributions:

1. We introduce a new descriptor based shape representation for 3D objects:
   - We generalize the popular surface normal reference to provide a full 3D coordinate system for local shape descriptors. The z-axis of the frame is the surface normal widely used in the past to establish view-independent shape descriptors. In addition, we determine the x-, and y-

2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops
978-1-4244-4441-0/09/$25.00 ©2009 IEEE

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.

axes in the same systematic way as we compute the z-axis to establish a stable 3D reference enabling highly discriminative 3D shape descriptors.

– We compute, at a 3D point, a highly discriminative shape feature as a weighted 3D occupational histogram of data points in its spherical neighborhood using a 3D partition that evenly divides the angular space. This choice of feature extraction ensures robustness w.r.t data noise as well as errors in computed reference frames.

2. We present an efficient algorithm to match 3D point clouds using the proposed shape descriptor. It requires neither combinatorial exploration of match results of feature descriptors for the pose estimation [14], nor combinatorial exploitation of data points to compute view independent references [15]. As a result, we eliminate noise introduced by combinational exploitation of noisy input in the match process for better recognition accuracy.

3. We present a highly effective and efficient index scheme to index feature descriptors for fast and accurate model-based recognition of large model databases.

The ISS algorithm operates directly on 3D point clouds and requires neither surface meshing nor triangulation of the data points. These preprocessing steps are error prone for data in real applications which is typical of sensor noise, surface discontinuity due to obscuration and occlusion, and close proximity of objects. It also makes little assumptions about the sensor physics, or viewing geometry.

The rest of the paper is organized as follows. We introduce the intrinsic shape signatures in Section 2. Section 3 describes how to match and recognize 3D objects using intrinsic shape signatures. An indexing scheme for the high dimensional feature descriptors is proposed in Section 4. Section 5 presents the test data set and performance evaluation of the new algorithm, where we also compare its performance to the spin image and 3D shape context algorithms. We draw conclusions in Section 6.

2. Intrinsic shape signatures

An intrinsic shape signature consists of an intrinsic reference frame enabling both view-invariant feature extraction and fast pose registration, and a highly discriminative feature vector encoding the 3D shape characteristics.

2.1. Intrinsic reference frame

It is desirable for shape descriptors to compute view independent representation of the local object shape. The invariance is accomplished by extracting features w.r.t a view independent reference. The surface normal vector of local surface patches has been the unanimous choice of reference to build view independent region/surface descriptors. Although it is popular, the surface normal vector alone is not sufficient to define a 3D coordinate system.

We define an intrinsic reference frame F_i at a basis point p_i with a supporting radius r_{frame} using the eigen analysis of the point scatter matrix as follows:

1. Compute a weight for each point p_i inversely related to the number of points in its spherical neighborhood of radius $r_{density}$: $w_i = 1/\left\|\{p_j : | p_j - p_i | < r_{density}\}\right\|$. (1)

This weight is used to compensate for uneven sampling of the 3D points, so that points at sparsely sampled regions contribute more than points at densely sampled regions.

2. Compute a weighted scatter matrix $\mathrm{cov}(p_i)$ for p_i using all points p_j within a distance r_{frame}:

$$COV(p_i) = \sum_{|p_j - p_i| < r_{frame}} w_j (p_j - p_i)(p_j - p_i)^T \Big/ \sum_{|p_j - p_i| < r_{frame}} w_j \quad (2)$$

3. Compute its eigen values $\{\lambda_i^1, \lambda_i^2, \lambda_i^3\}$ in the order of decreasing magnitude and their eigen vectors $\{e_i^1, e_i^2, e_i^3\}$;

4. Use p_i as the origin, use e_i^1, e_i^2, and their cross product $e_i^1 \otimes e_i^2$ as the *x*-, *y*- and *z*- axes respectively to define a 3D coordinate system F_i at p_i, which we call intrinsic reference frame (See Figure 1(a)).

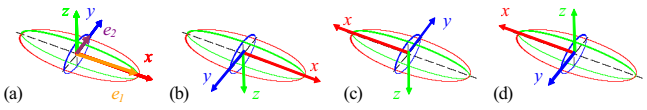


Figure 1 For a scatter matrix with eigen vectors $\{e_i^1, e_i^2, e_i^3\}$ of decreasing magnitude of eigen values, four intrinsic reference frames are computed.

Since an eigen vector of the scatter matrix computes a direction in the 3D space based on the amount of point position variations, its orientation has a 180^o ambiguity. Therefore, we have two choices for the orientation of each axis. As a result, we can only uniquely define the intrinsic reference frame at a basis point up to four variants as shown in Figure 1, i.e., F_i as described above, and three others, each obtained by rotating F_i along the *x*-, *y*-, or *z*- axes by 180^o respectively (see Figure 1(b)-(d)). Note that this ambiguity exists for the surface normal reference as well. It is possible to apply additional constraints to reduce the number of variants at a basis point. For example, if the position of the sensor is known, we can enforce the dot product of the z- axis of the intrinsic reference frame and the sensor direction to be positive to impose an "inside"/"outside" constraint and reduce the number of variants from four to two, as people do to remove the ambiguity in the surface normal. We do not impose the constraints here to make our algorithm sensor independent. However, when such information becomes available, it can be easily incorporated into our approach.

The intrinsic reference frame is a generalization of the classical surface normal reference for view independent shape feature extraction. The *z*- axis of the intrinsic reference frame, which is the eigen vector of the point scatter matrix with the smallest eigen value, is in fact the total least square estimate of the surface normal using the point data [17]. While the reference of the surface normal vector comes short of defining a 3D reference (which is necessary in encoding discriminative 3D shape information), the intrinsic reference frame determines the *x*-,

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.

y-, and z- axes systematically to yield a 3D reference frame pertinent to the local object shape. With the intrinsic reference frames we can define unique (with four symmetries) and highly discriminative representations for local/semi-local 3D shapes patches.

2.2. 3D shape feature extraction

For an intrinsic reference frame F_i at a basis point p_i, we extract an invariant feature vector w.r.t F_i to encode the 3D spatial shape characteristics in a local/semi-local support volume. On one hand, it is desirable to maintain as much spatial shape information as possible to make the features discriminative to subtle shape differences. On the other hand, the features have to be robust to data noise from multiple sources: unrepeatable sampling of the object surface, sensor noise, errors in the estimated reference frames, scene clutter and obscurations, and variations in viewing geometries, etc. We construct the feature vector using a 3D occupational histogram of the supporting spherical neighborhood of radius $r_{feature}$ centered at p_i, based on a partition in the polar coordinate system (ρ, θ, φ) aligned with F_i (see Figure 2(a)). A polar partition is preferred to a Cartesian partition because the latter is more vulnerable to rotation errors in the estimated reference frames.

A straightforward way to partition the spherical surface for 3D occupational histograms is to evenly divide the azimuth and elevation [7]. However, this creates not only bins of drastically varying sizes which need to be compensated for, but also degenerating bins near the pole: as many as L (number of azimuth divisions) bins intersect at the pole which makes these bins extremely vulnerable to noise since a noisy data point near the pole can easily fall in any of these bins.

Instead, we use a discrete spherical grid [10] recursively computed from a base octahedron (see Figure 2 (b)) to divide the spherical angular space (θ, φ) into relatively uniformly and homogeneously distributed cells (Figure 2 (c)). Discrete spherical grids of increasing resolutions are generated as follows using the base octahedron:

1. Divide each triangle in the polyhedron into four smaller triangles by adding the midpoint of each edge into the vertex set and connecting them;
2. Extend the vector from the origin to each new vertex until it intersects the unit sphere, and replace the vertex with the intersected point on the sphere surface;
3. Assign each point on the sphere surface to its nearest vertex to form a partition of the sphere surface, which is the discrete spherical grid.
4. Repeat steps 1 to 3 to compute the discrete spherical grid at the next resolution level.

Figure 2 (c) shows the discrete spherical grid at resolution level *3*. This grid partitions the angular space (θ, φ) into $N_{spherical\_grid} = 66$ bins. We compute a lookup table $LUT_{spherical\_grid}$ to map each angle pair (θ_i, φ_i) to a bin label: $LUT_{spherical\_grid}(\theta_i, \varphi_i) = l_i \in \{0,1,\cdots, N_{spherical\_grid} - 1\}$. During feature extraction, we use the discretized radial distance $\rho : \rho_0, \rho_1, \cdots, \rho_{L-1}$, and the spherical tessellation to partition the supporting spherical volume around a basis point into $K = 1 + (L-1) * N_{spherical\_grid}$ bins with no angular discrimination for $\rho < \rho_0$:

$$\{P_0 = \{p(\rho,\theta,\varphi) : \rho < \rho_0\},$$
$$P_{1+(i-1)*N_{spherical\_grid}+j} = \{p(\rho,\theta,\varphi) : \rho_i < \rho \le \rho_{i+1}, LUT_{spherical\_grid}(\theta,\varphi) = j\}\}.$$

The sums of weights (Eq. 1) of all points that fall in each bin form the shape feature vector $f_i = (f_{i0}, f_{i1}, \cdots, f_{iK-1})$ at a basis point p_i.

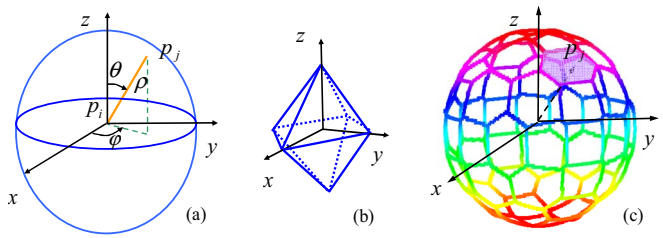


Figure 2 Partition in a 3D polar coordinate system: (a) the polar coordinate system (ρ, θ, φ) for intrinsic reference frame F_i, (b) the base octahedron used to construct the spherical grids, and (c) the spherical grid used to divide the angular space (θ, φ). The bins of the 3D histogram are formed using discretized levels of ρ and the spherical grid in (c).

2.3. Intrinsic shape signatures

The intrinsic shape signature (ISS) S_i={F_i, f_i} at a basis point p_i consists of two components:

1. The intrinsic reference frame $F_i = \{p_i, \{e_i^x, e_i^y, e_i^z\}\}$ where p_i is the origin, and $\{e_i^x, e_i^y, e_i^z\}$ is the set of basis vectors described in subsection 2. We should point out that the intrinsic frame is a characteristic of the local object shape and independent of viewpoint. Therefore, the view independent shape features can be computed using the frame as a reference. However, its basis $\{e_i^x, e_i^y, e_i^z\}$, which specifies the vector of its axes in the sensor coordinate system, are view dependent and directly encode the pose transform between the sensor coordinate system and the local object-oriented intrinsic frame, thus enabling fast pose calculation and view registration.
2. The 3D shape feature vector $f_i = (f_{i0}, f_{i1}, \cdots, f_{iK-1})$, which is a view independent representation of the local/semi-local 3D shape. These features can be compared directly to facilitate the matching of surface patches or local shapes from different objects.

Due to the fact that there are four symmetries for the intrinsic reference frame at a basis point (Figure 1), we compute four intrinsic shape signatures at a basis point for one of the two point clouds to be matched. When recognizing 3D objects, we store four intrinsic shape signatures for each basis point from objects in the model database, and compute only one intrinsic shape signature for each basis point from the query point clouds. For a query point and its corresponding database counterpart, the query descriptor will most likely match, among the four database descriptors, the one computed using the same reference

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.

frame the best. This best match is then retained to compute the pose transform and assess the confidence of the match. Although we encode four descriptors for each database point, the matching process is very expedited thanks to a novel similarity based indexing scheme to prune most of the "less similar" database features, which may arise due to different shape characteristics or inconsistent reference frames, from ever being compared to a query descriptor.

2.4. Matching intrinsic shape signatures

Two intrinsic shape signatures are matched by comparing the view independent shape feature vectors. Since the feature vector consists of weighted counts of points in the volume partitioning bins, we use χ^2 statistics to compute the distance between two shape feature vectors $f_i = (f_{i0}, f_{i1,} \cdots, f_{iK-1})$ and $f_j = (f_{j0}, f_{j1,} \cdots, f_{jK-1})$:

$$dist(f_i, f_j) = \sum_{k=0,K-1} (f_{ik} - f_{jk})^2 / (f_{ik} + f_{jk}) \cdot \quad (3)$$

2.5. Pose estimation

The introduction of the intrinsic reference frame for each shape descriptor greatly simplifies the pose estimation process for 3D object matching and recognition. Unlike most other methods, which require two or more matching descriptor pairs to generate a pose hypothesis, the rotation and translation between two matching intrinsic shape signatures can be directly computed using their intrinsic reference frames. For a matching pair of intrinsic shape signatures $S_i = \{\{p_i, \{e_i^x \quad e_i^y \quad e_i^z\}\}, f_i\}$ and $S_j = \{\{p_j, \{e_j^x \quad e_j^y \quad e_j^z\}\}, f_j\}$, the pose transform (R,T) is readily available via simple matrix and vector operations:

$$R = \Re_j R_i^t, \quad T = p_j - R p_i \quad (4)$$

where $\Re_i = \left[e_i^x \quad e_i^y \quad e_i^z\right]$ and $R_j = \left[e_j^x \quad e_j^y \quad e_j^z\right]$.

3. 3D matching/recognition using ISS

In this section we describe how to match two 3D point clouds using intrinsic shape signatures.

3.1. Representing 3D point clouds using intrinsic shape signatures

First, we select a set of salient basis points for which we compute the intrinsic shape signatures to represent the object / point cloud. These basis points, rich in 3D structure in its surrounding volume, are computed as the data points which possess large three dimensional point variations in their neighborhood. These variations are measured using the smallest eigen value of the point scatter matrix of its spherical neighborhood of radius $r_{salient}$. As the 3D reference frames can become ambiguous when two of the eigen values for the scatter matrix are equal, we also impose an additional constraint on the ratios of the eigen values $\lambda_i^2 / \lambda_i^1 < \gamma_{21}, \lambda_i^3 / \lambda_i^2 < \gamma_{32}$ to exclude frames of ambiguous axes at points of local symmetries. Resolution control is enforced to extract at most one basis point from a cubic volume of size d_{voxel}. Such extracted key points are shown in Figure 3 (b) for a query point cloud and its reference (model) point cloud (Figure 3 (a)). Once the basis points are determined, we then compute the intrinsic shape signatures as described in Sec. 2. The set of computed intrinsic shape signatures forms the representation for the point cloud / 3D object of interest, which is later used for matching and recognition.

3.2. Matching two point clouds

Having represented the 3D point clouds using the intrinsic shape signatures, we next establish one to one correspondences between likely matching intrinsic shape signatures from two 3D point clouds:

1. For the ith intrinsic shape signature S_i from query cloud Q, compute its distance to each intrinsic shape signature S_j^k for each model basis point p_j in the model/reference point cloud P, where k indexes one of the four variants at the basis point:

$$D(S_i, S_j) = \min_{k=1,4} dist(f_i, f_j^k), \quad (5)$$

where f_j^k is the shape feature vector of the k-th intrinsic shape signature at p_j. If this distance is less than a threshold value t_f, add the matching score and the point pair $(D(S_i,S_j),i,j,k)$ in a result list;

2. Sort the results in order of increasing distance value.

3. Starting from the front of the sorted match result list, save a result if the two basis points for the result have not been used yet, and mark the two points as used. Otherwise, remove this result from the list so that each basis only counts at most once toward valid matches. Advance to the next match result in the list and repeat the operation until the end of the result list is reached.

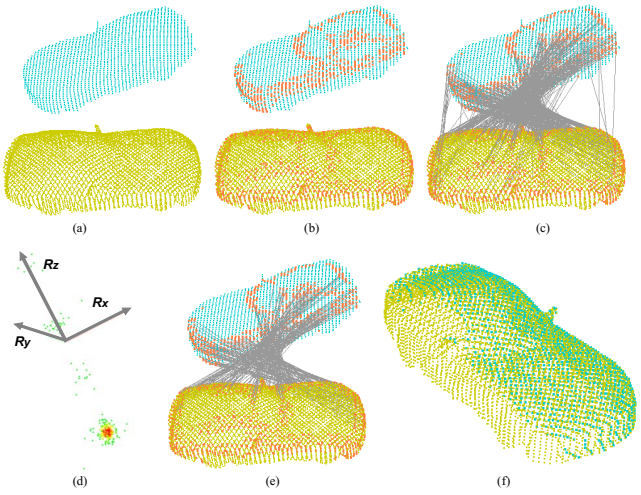


Figure 3 Matching two point clouds using ISS. (a) query point cloud of the partial view of a car in blue and the model point cloud in yellow; (b) extracted salient basis points marked in red, (c) matching descriptor pairs between query and model, (d) rotation estimates from pairs of matching descriptors: each point corresponds to one rotation "observation" from a pair of matching intrinsic shape descriptors, with the color indicating the density at the estimate (green for low density and red for high density), (e) matching descriptor pairs corresponding to the largest pose cluster, and (f) aligning model overlaid on query using estimated pose.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.

The outcome from the above process is a set of correspondences between intrinsic shape signatures with similar shape properties, which we denote as $\Omega_{match\_pair\_all}$. The ISS at each basis point is used at most once in $\Omega_{match\_pair\_all}$ to avoid one-to-many matching which may degrade the matching accuracy. This set of corresponding descriptor pairs will later be used to measure the shape similarity and the pose transform between the two point clouds. We show in Figure 3(c) the computed matching correspondences for the two point clouds in Figure 3(a), with the grey lines connecting the basis points of the matching ISS descriptors.

3.3. Pose transform estimation

One major advantage of the proposed approach is that a pose hypothesis can be generated from one pair of matching intrinsic shape signatures as opposed to two or more matching descriptor pairs required by most other methods. This eliminates the combinational exploitation of the matching pairs, not only reducing the computational cost but also suppressing noise in the generated pose hypotheses for robust pose estimation and verification. This is because the probability in repeatedly selecting the same two or three descriptors is much less than the probability in selecting the same one descriptor for accurate matching.

Equation 4 computes the rotation and translation between two intrinsic shape signatures, which is the pose transform between the local object oriented reference frames of represented surface patches/regions. If two point clouds do match, the pose transform between two corresponding local surface patches/regions is the same as the global pose transform between the two point clouds. As a result, the pose transforms computed from matching descriptor pairs for two matching point clouds are expected to cluster in the 6D pose space near the true pose transform $(R_x^{gt}, R_y^{gt}, R_z^{gt}, T_x^{gt}, T_y^{gt}, T_z^{gt})$ between the two point clouds. By matching intrinsic shape signatures and computing the pose transforms between the matched pairs, we are able to obtain direct "observations" of the pose transform between two 3D objects, and estimate it by finding the mode or the largest cluster of computed pose observations from the plausible matching descriptor pairs in $\Omega_{match\_pair\_all}$. We further simplify the search in 6D pose space by using two 3D searches: to determine first the 3D rotation and then the 3D translation. The rotation between two point clouds is estimated as follows:

1. Compute a 3D histogram for rotation angles for all matched descriptor pairs in $\Omega_{match\_pair\_all}$ using (4).
2. Gaussian smooth the histogram;
3. Find the bin with the maximum vote;
4. The cluster center/size is the Gaussian weighted average of all poses/counts in its neighborhood;

The cluster center gives the maximum likelihood estimation $(\hat{R}_x, \hat{R}_y, \hat{R}_z)$ of the rotation between the two objects. The size of the cluster, which is the number of matching intrinsic shape signature/surface patch pairs voting for this concerting rotation, provides a confidence measure for the match. We use $\Omega_{match\_pair\_rotation}$ to denote the set of matching intrinsic shape descriptor pairs whose rotation belong to the largest cluster. $\Omega_{match\_pair\_rotation}$ is the set of local shape regions satisfying the same global rotation constraint. Figure 3(d) shows the rotation space for all rotations computed from the matching intrinsic shape descriptors representing the two point clouds in Figure 3(a). Each point corresponds to one rotation "observation" from a pair of intrinsic shape descriptors as shown in Figure 3(c), with the color indicating the density at the point (green for low density and red for high density). The rotation between the two point clouds is apparent in the rotation space as the dense cluster of points in red indicating high density.

The translation estimate is then computed from the matching intrinsic shape signature pairs in $\Omega_{match\_pair\_rotation}$ using the rotation estimate $\hat{R} = (\hat{R}_x, \hat{R}_y, \hat{R}_z)$. The translation for each pair of the descriptors with the concerting rotation in $\Omega_{match\_pair\_rotation}$ is determined by substituting the estimated rotation $\hat{R}$ in Eq. 4. These translation "observations" are then clustered and the center of the largest cluster is the estimate for the translation $\hat{T} = (\hat{T}_x, \hat{T}_y, \hat{T}_z)$ between the two point clouds. The size of the cluster, N_C, is the number of matching surface patch pairs with concerting rotation $\hat{R}$ and translation $\hat{T}$. We denote the set of matching intrinsic shape descriptor pairs in this cluster as $\Omega_{match\_pair\_pose}$, as shown in Figure 3(e). This is the set of all matching descriptors with concerting poses, including both rotation and translation. $\hat{R}$ and $\hat{T}$ form the estimated pose transform between two point clouds. Figure 3(f) shows the reference point cloud (in yellow) in Figure 3(a) after applying the pose transform computed from the matched intrinsic shape signature pairs, which is overlaid with the query point cloud (in aqua-blue).

3.4. Criteria to match two 3D objects

A similarity measurement between two point clouds P and Q is the percentage of intrinsic shape signatures with matching descriptor pairs that comply with the estimated global pose transform between the two point clouds. It measures the similarity as the proportion of the object surface that matches globally:

$$Similarity(P,Q) = N_c / \sqrt{N_P \cdot N_Q}, \quad (6)$$

where N_P and N_Q are the number of basis points for P and Q respectively and $N_c = |\Omega_{match\_pair\_pose}|$ is the number of matching descriptors with consistent pose transforms.

An alternative error measurement is the position residual between the aligned pairs of matching basis points by applying the estimated pose to transform the model basis points to the query coordinate system:

$$E_{residual}(P,Q) = \sum_{i,j,\{p_i,q_j\} \in \Omega_{match\_pair\_pose}} (\hat{R}p_i + \hat{T} - q_j)^2 / N_c \cdot \quad (7)$$

This provides a stricter and more detailed error measurement between two point clouds.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.

We use a match error $E_{comprehensive}(P,Q)$, combining the above distance / similarity measurements to measure the matching error between two point clouds P and Q. It is defined as follows:

$$E_{comprehensive}(P,Q) = \frac{E_{residual}(P,Q)}{Similarity(P,Q)}. \quad (8)$$

A small value in the comprehensive error, implying a large number of matching descriptor pairs voting for the same global pose transform, and/or a small position misalignment using the estimated pose, indicates a good match between the two point clouds.

4. Indexing

We propose a new indexing scheme called "Locality Sensitive Trees (LST)", inspired by the popular "Locality Sensitive Hashing (LSH)" [11] approach, to rapidly retrieve the approximate nearest neighbors of a query feature in order to search large databases efficiently. An LST tree is a randomized binary tree [1] with which all its leaf nodes form a partition of the feature space and each the internal nodes contains its own data-dependant random test to assign incoming feature vectors to one of its two children. Although variants [2][13] of LSH have been proposed to address the rigidity of LSH by using data-adaptive hash functions, LST provides even greater data-adaptivity as the former can be considered as a subset of random binary trees with the constraint that nodes of the same depth use the same random test. LST has the following properties:

1. Each internal node selects its own random test based on the data distribution at this node to reflect the conditional distribution of feature vectors.

2. The structure of the tree adapts to the feature distribution. The leaf nodes may have varying depths.

3. The LST distributes database feature points relatively evenly among the leaf nodes to maximize its capacity.

We use a sequential method to construct an LST tree for the database features. Starting with an empty root/leaf node, we proceed as follows:

1. Randomly select without replacement a database point,

   i. Drop it down the tree until it reaches a leaf node based on the tests of the non-leaf nodes it encounters;

   ii. Split the leaf node if all of the following hold: a) the number of features at the node exceeds a pre-specified capacity $\mathcal{K}$, b) the depth of the leaf is less than a pre-specified depth threshold T_{depth}, and c) the leaf is split-able using the following method:

   – Select a random dimension i and perform a zero check of the i-th component f_{ji} of each feature f_j at the leaf node. This degenerated test checks for the occupancy at the i-th bin of the 3D histogram for the shape feature. The test divides the features at this node into two groups $\{f_j : f_{ji} = 0\}$ and $\{f_j : f_{ji} > 0\}$. If each group contains more than $\mathcal{K}/3$ feature points, split the leaf node into two child leaf nodes, one for each group, and save the test for this now non-leaf node. Otherwise, repeat the random dimension selection and test until a successful split or all dimensions are tested.

2. Repeat 1 until all feature points are inserted in the tree.

$\mathcal{L}$ randomized trees are constructed similar to LSH.

During the search, a query feature is dropped down each LST tree until it reaches a leaf and all stored database feature vectors at these leaf nodes are retrieved. The query feature is only compared to features in the union of retrieved feature vectors from the $\mathcal{L}$ trees, which often counts for a tiny fraction of all the database features to drastically reduce that matching time.

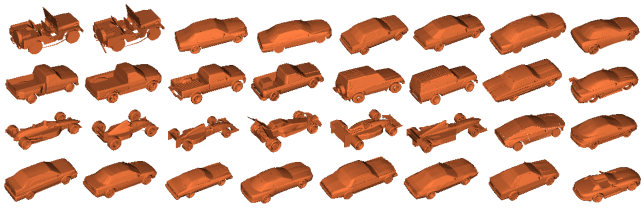


Figure 4 Sample models from the database of 72 vehicles.

5. Experiments

In order to evaluate the recognition performance of the proposed algorithm, experiments were carried out using a database of 72 vehicle models from the Princeton Shape Benchmark (PSB) [19] to recognize different vehicle types. It is a very challenging problem to identify the vehicle types accurately because that, as shown in Figure 4, many of the vehicles are highly similar in shape. For example, the first two cars shown in Figure 4 are almost identical except for a small part in the rear of the vehicles. Since the original models from PSB are of different scales, we rescaled each model to 4.2m in the direction of the principle axis of the largest eigen value to make each 3D model approximately the size of an actual car.

We adopted the experiment settings similar to [7] to evaluate the performance of the ISS algorithm in the presence of sensor noise, obscuration and clutter. A fan beam 3D LADAR simulator was used to generate model and query point clouds from the CAD models. The point clouds for the database models were generated by scanning each model noise-free with the sensor at 400m away, with a 45^o declination angle. On the other hand, the query point clouds were rendered at a 30^o declination angle and azimuth angles that were at least 15^o apart from the model scans. Table 1 lists the parameter values used in the experiments.

In all experiments, the performances of ISS were compared to that of well established 3D recognition algorithms including Spin Image and 3D Shape Context. We used the publically available SI software (http://www2.cs.cmu.edu/~vmr/software/meshtoolbox/downloads.html) and implemented 3DSC algorithm according to [7]. We used conforming parameters including the supporting radius, its discretization level, the size of the volume used to compute the reference frames, etc. Shape correlation of uncompressed spin images was used to obtain the best recognition performance [14]. The same angular partition parameters as in [7] were used for 3DSC, which created feature vectors of 1320 dimensions. While basis points were only extracted at locations with salient shape information for the ISS approach, one basis point was extracted from every non-empty voxel of size d_{voxel} for the 3DSC approach. No indexing was applied to SI and 3DSC.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.

5.1. Sensor noise

The first set of experiments assesses the effects of sensor noise on the performance of the algorithm. Two noise free model scans were acquired for each vehicle, one from the front left and the other from the front right, resulting a total number of 144 model scans in the database. We rendered two query clouds of each vehicle at azimuth angles that were 15° apart from the model scans. Gaussian sensor noise with a zero mean and standard deviations of 0.05m, 0.10m, and 0.15m, were added in the direction of the LADAR beam (Figure 5). Each query cloud was then matched against the model database, and the best model was retrieved. Each model scan consisted of 600 to 4,000 points, where about 350 to 1000 basis points were selected. Four ISS descriptors were encoded for each model basis point to yield a database of ~480,000 595-dimensional feature vectors for the 144 models. The query clouds contained between 1,200 to 2,600 points, with between 980 to 1,700 key points extracted.

LST was used to index the feature database. A total number of $\mathcal{L}$ =40 randomized trees were constructed where the capacity for the leaf nodes was $\mathcal{K}$ =50. With the LST indexing, only about 0.4% of the matches between feature vectors were performed comparing to the exhaustive match, pruning away 99.6% of the pair wise feature comparisons. Even with the drastic reduction in computations, the LST indexing still yielded recognition accuracy almost the same as the one using the exhaustive matching method. In a separate study, we investigated a more generalized indexing scheme which used a random test to evenly distribute points to the child nodes instead of the degenerated occupancy test as in LST and found it achieving more than 95% nearest neighbor accuracy by examining less than 1% database points for several high dimensional datasets.

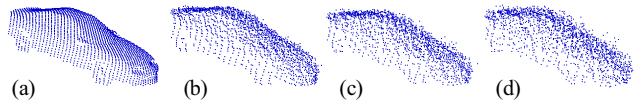


Figure 5 Experiments with sensor noise. (a) the model scan from front left, (b-d) are query scans with Gaussian noises of sigma equal to 0.05m, 0.10m, and 0.15m respectively.

The recognition performances of the three algorithms in the presence of sensor noise are presented in Figure 7(a), where we plot the percentage of correctly recognized query clouds against the number of top models retrieved, at each of the noise level. ISS performs the best at every noise level. As we can see that at moderate sensor noise level when σ=0.05m, all three algorithms work very well. However, as the noise level increases to σ=0.10m, the performance of SI deteriorates sharply, while 3DSC correctly recognizes near 94% of the query with the top match. ISS achieves over 97% accuracy with the best match, and 100% accuracy with the top three matches. Even at a very severe noise level of σ=0.15m, the ISS approach still performs very decently with an accuracy of 85% using only the top match, and 93% and 96% using the top two and three matches respectively.

5.2. Obscuration and clutter

The second set of experiments evaluates the performance of the algorithm in the presence of obscuration and clutter. We integrated four noise free individual model scans acquired at a 90° azimuth interval apart to form one model cloud for each vehicle type. One such model cloud is shown in Figure 6(a). Each vehicle were placed in a cluttered scene consisting of a house and trees as shown in Figure 6 (b), and two query scans from the front left and the front right were acquired. We used as query clouds, the data points within a 6mx4mx3m box enclosing the vehicle from these scans (see Figure 6 (c-d)), and matched them to the 72 models.

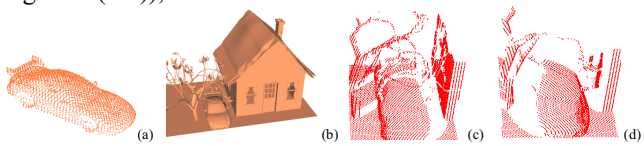


Figure 6 Experiments with obscuration and clutter. (a) A model point cloud; (b) the CAD model of a scene; (c) a query point cloud from the front left view, and (d) a query from the front right view.

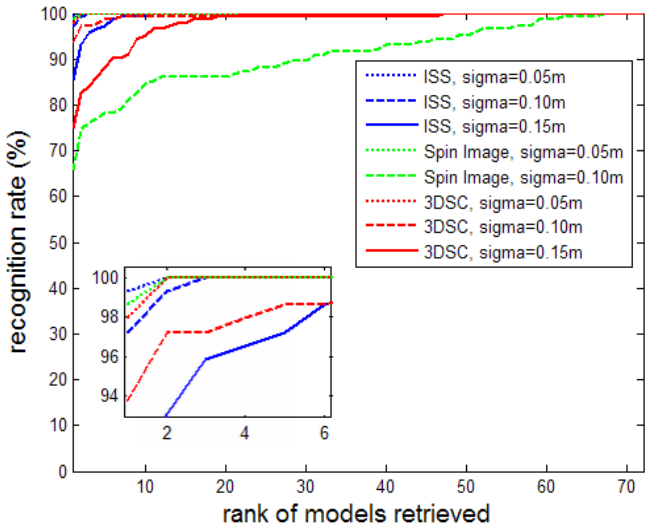
About 700 to 1000 basis points were selected for each model cloud to yield a database of ~307,000 595-D feature vectors for the 72 models. The LST index of this database took less than 100 seconds on a 2.0GHz PC. The recognition of one query point cloud, including the feature extraction, and the matching to the database of 72 vehicles, took an average of 25/13 seconds for the front left/right view, which consisted of between 5200~5700/2200~3000 3D data points and about 1200~1900/440~900 extracted basis points. The LST indexing speeded the recognition process by a factor of 50x comparing to the ISS recognition using exhaustive matching. On the other hand, it took the Spin Image software about 6~8/3~4 hours to process a query cloud from the left/right view.

The performance curves for the three algorithms are shown in Figure 7(b). The ISS approach again demonstrated superior performance in the presence of both occlusion and clutter. It correctly recognized all but one and two of the 72 query clouds for the left front view and right front view respectively, using only the top one match. 100% accuracy was achieved when the top three models were retrieved. The spin image approach appeared to be much more vulnerable to obscuration and clutter, performing with a recognition accuracy of 65% and 45% for the two views using the top matched model. We found the application of 3DSC to cluttered scenes ad hoc due to its inability to both distinguish the object from its background and integrate this knowledge into the objective function. We used the sum of the smallest 50% of the distances from the query points to the reference set as score for a cluttered query [7]. Ideally this percentage should reflect the amount of background in the scene which is rarely available in real applications.

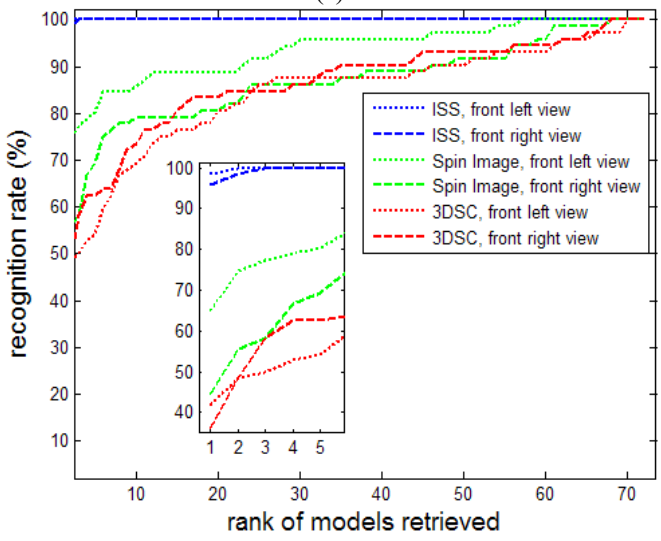
Parameter	value	description
$r_{density}$	*0.3m*	support radius for point density
r_{frame}	*0.3m*	support radius for intrinsic frame
$r_{feature}$	*1.5m*	support radius for shape feature vector
L	*10*	radial distance discretization level
$N_{sperical\_grid}$	*66*	# of bins for discrete spherical grid
d_{voxel}	*0.1m*	grid size for resolution control
r_{21} , r_{32}	*0.975*	Saliency threshold

Table 1 Parameter values for the ISS algorithm.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.



(a)



(b)

Figure 7 Recognition accuracy versus the rank of retrieved models: (a) Recognition accuracy at different sensor noise levels, and (b) Recognition accuracy in the presence of obscuration and clutter.

6. Conclusions

This paper presents a descriptor based approach called Intrinsic Shape Signatures for general 3D object representation and recognition. The ISS approach enables both highly discriminative shape matching and efficient pose estimation and registration for 3D point clouds. In addition, we introduce an efficient and effective indexing scheme for fast and accurate search/retrieval of large databases using ISS. We believe that the sound underlying concepts, and the sensible implementations of these ideas, both carefully crafted to ensure the discriminativeness, descriptiveness, and robustness to noise, make the algorithm one of the best 3D recognition algorithms up to date.

References

[1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees, Neural Computation, 9:1,545-1,587.

[2] M. Bawa, T. Condie, and P. Ganesan. LSH Forest: Self-tuning tndexes for timilarity tearch, Proc. 14th int'l conf. on World Wide Web, 2005.

[3] P. J. Besl and R. Jain. Range image understanding, Proc IEEE CVPR, pp. 430-449, 1985.

[4] R. Campbell and P. Flynn. A survey of freeform object representation and recognition techniques, CVIU, 81(2):166-210.

[5] C. Chua and R. Jarvis. Point signatures: a new representation for 3D object recognition, IJCV, 25(1):63-85, 1997.

[6] O. Faugeras and M. Hebert, The representation, recognition, and localization of 3-D objects, International Journal of Robotics Research, 5(3):27-52, 1986.

[7] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors, Proc. ECCV, 2004.

[8] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann: Robust Global Registration. Symposium on Geometry Processing, pp. 197-206, 2005.

[9] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3D object recognition from range images using local feature histograms, Proc IEEE CVPR, Vol. 2, pp. 394-399, 2001.

[10] B. Horn, Robot Vision, 1986.

[11] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala, Locality-preserving hashing in multidimensional spaces, Proc. 29th ACM Symp on Theory of Computing, pp. 618-625, 1997.

[12] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. Computer-Aided Design, 37:509-530, 2005.

[13] H. Jégou, L. Amsaleg, C. Schmid, and P. Gros, Query adaptative locality sensitive hashing, Proc. IEEE ASSP, 2008.

[14] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans PAMI, 21(5):433-449, 1999.

[15] A.S. Mian, M. Bennamoun, and R. Owens. 3D model-based object recognition and segmentation in cluttered scenes, IEEE Trans. PAMI, (28)10:1584-1601, 2006.

[16] A. S. Mian, M. Bennamoun and R. Owens, "Keypoint Detection and Local Feature Matching for Textured 3D Face Recognition", Int. Journal of Computer Vision, 2008.

[17] N. Mitra, A. Nguyen and L. Guibas. Estimating surface normals in noisy point cloud data. Int. J. Computational Geometry and its Applications, 14(4-5):261-276, 2004.

[18] S. Ruiz-Correa, L.G. Shapiro and M. Meila. A new paradigm for recognizing 3-D object shapes from range data, Proc. IEEE ICCV, Vol 2:1126-1131, 2003.

[19] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser. The Princeton Shape Benchmark, Shape Modeling Int'l 2004.

[20] P. Shilane and T. Funkhouser. Selecting distinctive 3D shape descriptors for similarity retrieval. Proc. IEEE Int. Conf. Shape Modeling and Applications, pp. 18-23, 2006.

[21] F. Stein and G. Medioni. Efficient 3-D object recognition, 14(2):125-145, IEEE Trans. PAMI, 1992.

[22] B. Taati, M. Bondy, P. Jasiobedzki and M. Greenspan. Variable Dimensional Local Shape Descriptors for Object Recognition in Range Data, Proc. IEEE ICCV, 2007.

[23] S. Yamani, A. Fraag, and A. El-Bialy, Free-form surface registration and object recognition using surface signatures, Proc IEEE ICCV 1999.

[24] Y. Zhong, "Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition", technical report, 2007.

Acknowledgements

We would like to thank Dr. Daniel Huber at Carnegie Melon University for the use of their spin image software. This work was supported through subcontract 11-001 by SET Corp. under contract FA8651-06-C-0135 by AFRL. Approved for public release; distribution is unlimited.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 04,2010 at 00:34:09 UTC from IEEE Xplore. Restrictions apply.