Object Detection from Large-Scale 3D Datasets Using Bottom-Up and Top-Down Descriptors

Alexander Patterson IV, Philippos Mordohai, and Kostas Daniilidis

University of Pennsylvania
{aiv,mordohai,kostas}@seas.upenn.edu

Abstract. We propose an approach for detecting objects in large-scale range datasets that combines bottom-up and top-down processes. In the bottom-up stage, fast-to-compute local descriptors are used to detect potential target objects. The object hypotheses are verified after alignment in a top-down stage using global descriptors that capture larger scale structure information. We have found that the combination of spin images and Extended Gaussian Images, as local and global descriptors respectively, provides a good trade-off between efficiency and accuracy. We present results on real outdoors scenes containing millions of scanned points and hundreds of targets. Our results compare favorably to the state of the art by being applicable to much larger scenes captured under less controlled conditions, by being able to detect object classes and not specific instances, and by being able to align the query with the best matching model accurately, thus obtaining precise segmentation.

1 Introduction

Object detection and recognition in images or videos is typically done based on color and texture properties. This paradigm is very effective for objects with characteristic appearance, such as a stop sign or the wheel of a car. There are, however, classes of objects for which 3D shape and not appearance is the most salient feature. Cars are an object category, whose appearance varies a lot within the class, as well as with viewpoint and illumination changes. Instead of representing these objects with a collection of appearance models, specific to each viewpoint, several researchers have used range scanners and addressed object recognition in 3D. Range as an input modality offers the advantages of using the full dimensionality of an object and avoiding any scale ambiguity due to projection. In addition, figure-ground segmentation is easier in 3D than in 2D images since separation in depth provides powerful additional cues. On the other hand, range sensors have significantly lower resolution compared to modern cameras and alignment between the query and the database models still has to be estimated. The challenges associated with object detection in 3D are due to intra-class shape variations, different sampling patterns due to different sensors or different distance and angle between the sensor and the object, targets that are almost always partial due to self-occlusion and occlusion.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 553–566, 2008.

Fig. 1. Cars detection results from real LIDAR data. Cars have false colors.

In this paper, we present an approach for detecting and recognizing objects characterized by 3D shape from large-scale datasets. The input is a point cloud acquired by range sensors mounted on moving vehicles. A part of the input is used as training data to provide manually labeled exemplars of the objects of interest, as well as negative exemplars where objects of interest are not present. Our algorithm automatically detects potential locations for the target objects in a bottom-up fashion. These locations are then processed by the top-down module that verifies the hypothesized objects by aligning them with models from the training dataset. We show results on a very large-scale dataset which consists of hundreds of millions of points. To the best of our knowledge no results have been published for datasets of this size. State-of-the-art 3D recognition systems on real data [1,2,3,4,5] have shown very high recognition rates, but on high-resolution scenes containing a few small objects captured in controlled environments.

We believe that our research is a first step towards fully automatic annotation of large scenes. Recent advances in sensor technology have made the acquisition and geo-registration of the data possible. Detailed 3D models can be generated very efficiently to provide high-quality visualization [6], but their usefulness for everyday applications is limited due to the absence of semantic annotation. Much like image-based visualizations, such as the Google Street View, these representations cannot answer practical questions, such as "where is the nearest gas station, mailbox or phonebooth". Automatic methods for scene annotation would dramatically increase the benefits users can derive from these large collections of data. While our methods are not currently capable of addressing the problem in its full extend, this paper introduces a framework for object detection from range data that makes a step towards automatic scene annotation. Some results on car detection can be seen in Fig. 1.

The main technical contribution of our work is the combination of a bottom-up and a top-down process to efficiently detect and verify the objects of interest. We use spin images [7] as local descriptors to differentiate between the target

objects and clutter and Extended Gaussian Images (EGIs) [8] to ascertain the presence of a target at the hypothesized locations. This scheme enables us to process very large datasets with high precision and recall. Training requires little effort, since the user has to click one point in each target object, which is then automatically segmented from the scene. The remaining points are used as negative examples. Spin images are computed on both positive and negative examples. EGIs only need to be computed for the positive exemplars of the training set, since they are used to align the bottom-up detection with the model database. Accurate alignment estimates between similar but not identical objects enable us to segment the target objects from the clutter. The contributions of our work can be summarized as follows:

- The combination of bottom-up and top-down processing to detect potential targets efficiently and verify them accurately.
- The capability to perform training on instances that come from the same object category as the queries, but are not necessarily identical to the queries.
- Minimal user efforts during training.
- Object detection for large-scale datasets captured in uncontrolled environments.
- Accurate segmentation of target objects from the background.

2 Related Work

In this section, we briefly overview related work on local and global 3D shape descriptors and 3D object recognition focusing only on shape-based descriptors. Research on appearance-based recognition has arguably been more active recently, but is not directly applicable in our experimental setup.

Global shape descriptors include EGIs [8], superquadrics [9], complex EGIs [10], spherical attribute images [11] and the COSMOS [12]. Global descriptors are more discriminative since they encapsulate all available information. On the other hand, they are applicable to single segmented objects and they are sensitive to clutter and occlusion. A global representation in which occlusion is explicitly handled is the spherical attribute image proposed by Hebert et al. [11].

A method to obtain invariance to rigid transformations was presented by Osada et al. [13] who compute shape signatures for 3D objects in the form of shape statistics, such as the distance between randomly sampled pairs of points. Liu et al. [14] introduced the directional histogram model as a shape descriptor and achieved orientation invariance by computing the spherical harmonic transform. Kazhdan et al. [15] proposed a method to make several types of shape descriptors rotationally invariant, via the use of spherical harmonics. Makadia et al. [16] compute the rotational Fourier transform [17] to efficiently compute the correlation between EGIs. They also propose the constellation EGI, which we use in Section 5 to compute rotation hypotheses.

Descriptors with local support are more effective than global descriptors for partial data and data corrupted by clutter. Stein and Medioni [18] combined surface and contour descriptors, in the form of surface splashes and super-segments,

respectively. Spin images were introduced by Johnson and Hebert [7] and are among the most popular such descriptors (See Section 4). Ashbrook et al. [19] took a similar approach based on the pairwise relationships between triangles of the input mesh. Frome et al. [20] extended the concept of shape contexts to 3D. Their experiments show that 3D shape contexts are more robust to occlusion and surface deformation than spin images but incur significantly higher computational cost. Huber et al. [21] propose a technique to divide range scans of vehicles into parts and perform recognition under large occlusions using spin images as local shape signatures.

Local shape descriptors have been used for larger scale object recognition. Johnson et al.[1] use PCA-compressed spin images and nearest neighbor search to find the most similar spin images to the query. Alignment hypotheses are estimated using these correspondences and a variant of the ICP algorithm [22] is used for verification. Shan et al. [23] proposed the shapeme histogram projection algorithm which can match partial object by projecting the descriptor of the query onto the subspace of the model database. Matei et al. [3] find potential matches for spin images using locality sensitive hashing. Geometric constraints are then used to verify the match. Ruiz-Correa et all. [5] addressed deformable shape recognition via a two-stage approach that computes numeric signatures (spin images) to label components of the data and then computes symbolic signatures on the labels. This scheme is very effective, but requires extensive manual labeling of the training data. Funkhouser and Shilane [24] presented a shape matching system that uses multi-scale, local descriptors and a priority queue that generates the most likely hypotheses first.

In most of the above methods, processing is mostly bottom-up, followed in some cases by a geometric verification step. A top-down approach was proposed by Mian et al. [4] who represent objects by 3D occupancy grids which can be matched using a 4D hash table. The algorithm removes recognized objects from the scene and attempts to recognize the remaining data until no additional library object can be found.

Our method can detect cars in real scenes in the presence of clutter and sensor noise. Very few of the papers mentioned above ([1,2,3,4,5]) present results on real data. Among the ones that do, Matei et al. [3] classified cars that had been previously segmented. Johnson et al. [1], Carmichael et al. [2] and Mian et al. [4] show object detection from real scenes containing multiple objects. It should be noted, however, that the number of objects in the scene is small and that all objects were presented to the algorithm during training. Ruiz-Correa et. all [5] are able to handle intra-class variation, at the cost of large manual labeling effort. The goal of our work is more ambitious than [1,2,3,4,20,23] in order to make more practical applications possible. Our algorithm is not trained on exemplars identical to the queries, but on other instances from the same class. This enables us to deploy the system on very large-scale datasets with moderate training efforts, since we only have to label a few instances from the object categories we are interested in.

3 Algorithm Overview

Our algorithm operates on 3D point clouds and entails a bottom-up and a top-down module. The steps for annotation and training are the following:

1. The user selects one point on each target object.
2. The selected target objects are automatically extracted from the background.
3. Compute surface normals for all points[1] in both objects and background.
4. Compute spin images on a subset of the points for both objects and background and insert into spin image database DB_{SI} (Section 4).
5. Compute an EGI for each object (not for the background). Compute constellation EGI and density approximation. Insert into EGI database DB_{EGI} (Section 5).

Processing on test data is performed as follows:

1. Compute normals for all points and spin images on a subset of the points.
2. Classify spin images as positive (object) or negative (background) according to their nearest neighbors in DB_{SI}.
3. Extract connected components of neighboring positive spin images. Each connected component is a query (object hypothesis).
4. Compute an EGI and the corresponding constellation EGI for each query.
5. For each query and model in DB_{EGI} (Section 5):
   (a) Compute rotation hypothesis using constellation EGIs.
   (b) For each rotation hypothesis with low distance according to Section (5.3), compute translation in frequency domain.
   (c) Calculate the overlap between query and model.
6. If the overlap is above the threshold, declare positive detection (Section 5).
7. Label all points that overlap with each of the models of DB_{EGI} after alignment as object points to obtain segmentation.

4 Bottom-Up Detection

The goal of the bottom-up module is to detect potential target locations in the point cloud with a bias towards high recall to minimize missed detections. Since detection has to be performed on very large point clouds, we need a representation that can be computed and compared efficiently. To this end we use spin images [7]. A spin image is computed in a cylindrical coordinate system defined by a reference point and its corresponding normal. All points within this region are transformed by computing α, the distance from the reference normal ray and β the height above the reference normal plane. Finally a 2D histogram of α and β is computed and used as the descriptor. Due to integration around the normal of the reference point, spin images are invariant to rotations about the normal. This is not the case with 3D shape contexts [20] or EGIs (Section 5) for which several rotation hypotheses have to be evaluated to determine a match. Since

[1] During normal computation, we also estimate the reliability of the normals, which is used to select reference points for the spin images.

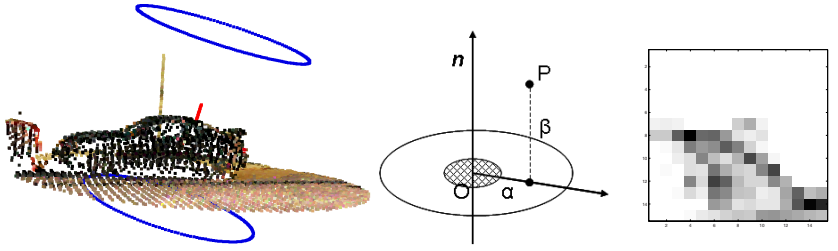

Fig. 2. Left: spin image computation on real data. The blue circles delineate the cylindrical support region and the red vector is the normal at the reference point. Middle: illustration of spin image computation. O is the reference point and $\boldsymbol{n}$ its normal. A spin image is a histogram of points that fall into radial (α) and elevation (β) bins. Right: the spin image computed for the point on the car.

spin image comparison is a simple distance between vectors, their comparisons are computationally cheaper, but less discriminative.

Johnson and Hebert [7] computed spin images on meshes. This can compensate for undesired effects due to varying sample density, since triangles contribute to each bin of the histogram with their area. Triangulating the point cloud to obtain a mesh is not trivial in our case, not only because of the computational cost, but also due to noise, sparsity and sampling patterns of the data. Similar to [20], we compute spin images directly from the point clouds and weigh the contribution of each point by its inverse density to account for sampling differences. Local density is computed in balls centered at every point. Isolated points are removed. Accounting for variations in point density is important for point clouds captured by range sensors since the density of samples on a surface is a function of sensor type, as well as distance and angle to the sensor.

Given a point cloud, regardless of whether it contains training or test data, normals for all points are computed using tensor voting [25]. We then need to select reference points for the spin images. Our experiments have shown that spin images vary smoothly as long as the reference point is on the same surface and the normal is accurate. Therefore, reference points need to be dense enough to capture all surfaces of the object, but higher density is redundant. For cars, a distance of $0.4m$ was found to offer a good trade-off between coverage and computational efficiency. We obtain such a sampling by placing a 3D grid of the desired resolution in the dataset and dropping vertices that have no scanned points in their voxel. Since the reference points need to be among the points sampled by the scanner, the retained vertices are moved to the median of the nearest points to account for noisy samples. (The grid can be seen in the two rightmost images of Fig. 3.) A spin image is computed for each of these points unless the eigenvalues of the tensor after tensor voting indicate that the estimated normal is unreliable [25]. Our spin images have 15 radial and 15 elevation bins resulting in a 225-D descriptor.

For the training data, the user has to specify the targets, which are assumed to be compact objects lying on the ground, by clicking one point on each. Then, an

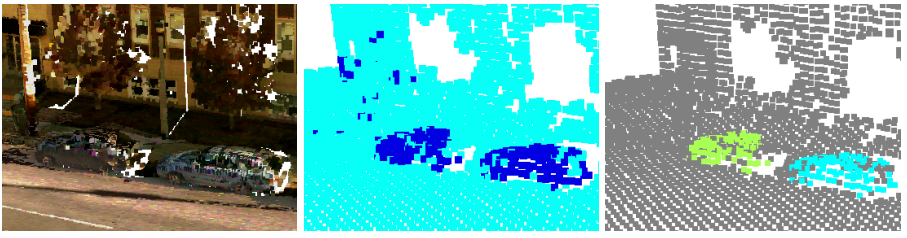

Fig. 3. Left: input point cloud. Middle: Classification of spin images as target (blue) and background (cyan). (Only the reference points are shown.) Right: target spin image centers clustered into object hypotheses. Isolated target spin images are rejected.

automatic algorithm segments the object as a connected component protruding from the ground. The ground can be reliably estimated in a small neighborhood around the selected point as the lowest smooth surface that bounds the data. Spin images computed for points on the targets are inserted into the spin image database DB_{SI} as positive exemplars, while spin images from the background are inserted as negative exemplars. We have implemented the database using the Approximate Nearest Neighbor (ANN) k-d tree [26].

During testing, query spin images are computed on reference points on a grid placed on the test data as above. Each query spin image is classified according to the nearest neighbor retrieved from DB_{SI}. Some results on real data can be seen in Fig. 3. Potential locations of the target objects can be hypothesized in areas of high density of positive detections, while isolated false positives can be easily pruned from the set of detections.

Object hypotheses (queries) are triggered by spin images that have been classified as positive (target). Target spin images are grouped into clusters by a simple region growing algorithm that starts from a spin image reference point and connects it to all neighboring target spin images within a small radius. When the current cluster cannot be extended any further, the algorithm initializes a new cluster. Raw points that are within a small distance from a cluster of spin images are also added to it to form a query. Since neighboring spin images overlap, the bottom-up portion of our algorithm is robust to some miss-classifications.

5 Top-Down Alignment and Verification

The second stage of processing operates on the queries (clustered points with normals) proposed by the bottom-up stage and verifies whether targets exist at those locations. Spin images without geometric constraints are not discriminative enough to determine the presence of a target with high confidence. Spin image classification is very efficient, but only provides local evidence for the presence of a potential part of a target and not for a configuration of parts consistent with a target. For instance a row of newspaper boxes can give rise to a number of spin images that are also found in cars, but cannot support a configuration

of those spin images that is consistent with a car. The top-down stage enforces these global configuration constraints by computing an alignment between the query and the database models using EGI descriptors.

Early research has shown that there is a unique EGI representation for any convex object [27], which can be obtained by computing the density function of all surface normals on the unit sphere. If the object is not convex, its shape cannot be completely recovered from the EGI, but the latter is still a powerful shape descriptor. The EGI does not require a reference point since the relative positions of the points are not captured in the representation. This property makes EGIs effective descriptors for our data in which a reference point cannot be selected with guaranteed repeatability due to occlusion, but the distribution of normals is fairly stable for a class of objects.

5.1 Computing EGIs

EGIs are computed for the positive object examples in the training set. Objects are segmented with assistance from the user, as described in Section 4. For the test data, an EGI is computed for each object hypothesis extracted according to the last paragraph of Section 4. Each EGI contains the normals of all input points of the cluster, oriented so that they point outwards, towards the scanner. These orientations can be computed since the trajectory of the sensor is available to us. The majority of objects are scanned only from one side and, as a result, the normals typically occupy at most a hemisphere of the EGI. This viewpoint dependence occurs for both the queries and database objects and thus requires no special treatment. If necessary, database models can be mirrored to increase the size of the database without additional manual labeling since model symmetry is modeled by the EGI.

5.2 Constellation EGIs

Unlike spin images, comparing two EGIs requires estimating a rotation that aligns them before a distance can be computed. One can compute the rotational Fourier transform [17] to efficiently compute all correlations between EGIs [16]. This technique is efficient if all rotations need to be computed, but it is sensitive to clutter, missing parts and quantization. Our experiments have shown that quantization can have adverse effects on rotation and distance computations

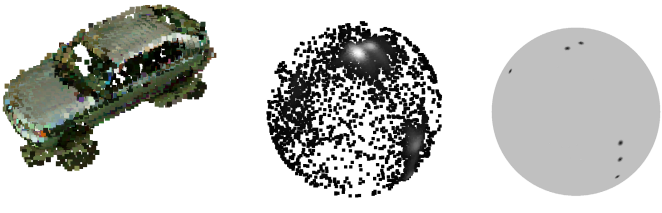

Fig. 4. Left: a database model of a car. Middle: illustration of an EGI in which points are color-coded according to their density. Right: the corresponding constellation EGI.

using EGIs. We can use the constellation EGI to cue a more efficient distance computation. Therefore, we avoid quantizing the orientations of the normals in an EGI and do not treat it as an orientation histogram.

Instead of an exhaustive search using either spatial or Fourier methods, we use a technique that generates discrete alignment hypotheses, which was originally proposed in [16]. A *constellation EGI* records the locations of local maxima in the distribution of normals in the EGI. We call these maxima *stars*, since they resemble stars in the sky. An EGI and the corresponding constellation EGI for an object can be seen in Fig. 4. Two constellation EGIs can be matched by sampling pairs of stars that subtend the same angle on the sphere. Each sample generates matching hypotheses with two stars of the other EGI. If the angles between each pair are large enough and similar, a rotation hypothesis for the entire descriptor is generated. Note that a correspondence between two pairs of stars produces two possible rotations. Similar rotations can be clustered to reduce the number of hypotheses that need to be tested. The resulting set of rotations are evaluated based on the distance between the entire EGIs and not just the stars.

5.3 Hypothesis Verification

Conceptually, the rotation hypothesis that achieves the best alignment of the descriptors is the one that maximizes the cross-correlation between all normal vectors of the first and the second EGI. This computation is exact, but computationally expensive since models and queries consist of thousands of points each. To reduce the computational complexity, we select a smaller set of normals and compute the weights of kernels which are centered on this set, thus closely approximating the original EGI via interpolation. This computation is performed once per EGI and significantly reduces the cost of distance computation. Specifically, to create an approximation of the EGI for a set of input normals, we compute the density at all input normals on the sphere. We then select a subset of samples by greedily choosing a predetermined number of points. Each choice is made by computing the current interpolation via nearest neighbor, and then adding the normal with the largest deviation between approximated and actual values. Our method is similar to [28], but operates on the sphere. Once we have a set of kernel centers N_s which is a subset of all normals N, the weights of the kernels are computed as follows:

$$v_{ij} = \frac{max(d_{max} - \arccos(\hat{n}_i^T \hat{n}_j), 0)}{\sum_j (max(d_{max} - \arccos(\hat{n}_i^T \hat{n}_j), 0))}, \quad i \in N, j \in N_s$$

$$D_j = V^\dagger D_i, \tag{1}$$

Where D_j are the coefficients at the sparse set of normals N_s, and d_{max} is the range of the kernel function. Using this new representation, we can compute the distance between two EGIs, using a sparse set of samples, after applying a rotation hypothesis. If the two shapes are identical, the density values should be equal over the entire sphere. We measure the deviation from an ideal match by predicting the density on the samples of one EGI using the interpolation

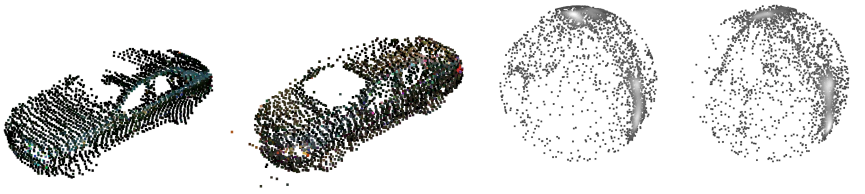

Fig. 5. Alignment of a database model (left car and left EGI) and a query (right car and right EGI) that have been aligned. The car models are shown separately for clarity of the visualization. Notice the accuracy of the rotation estimation. The query has been segmented by the positive spin image clustering algorithm and the model by removing the ground after the user specified one point.

function of the other EGI and comparing them with the original density values. Specifically we use the l_1 distance computed at the query points which we can now interpolate once the normals N_s are rotated according to each hypothesized rotation. The minimum distance provides an estimate of the best rotation to align the two objects, but no estimate of translation and most importantly no indication of whether the objects actually match. Typically, 1-5 rotations are close enough to the minimum distance. For these, we estimate the translation and compute the final distance in the following section.

5.4 Alignment and Distance Computation

Given the few best rotation hypotheses based in section 5.3, we compute the translation that best aligns the two models in the frequency domain. We adopt the translation estimation method of [16] in which translation is estimated using a Fourier transform in $\mathbb{R}^3$. This is less sensitive to noise in the form of missing parts or clutter than global alignment methods that estimate complete rigid transformations in the Fourier domain. We begin by voxelizing the model and the query to obtain binary occupancy functions in 3D. We then compute their convolution efficiently using the fft and take the maximum as our translation.

Finally, we need a measure of distance to characterize the quality of the alignment that is flexible enough to allow for deformation between the query and the model. We experimented with the ICP distance [22], without performing ICP iterations, but found the overlap between the query and model to be more effective because the quantization in the translation estimation caused large ICP distance errors even though the models were similar. The overlap is computed as the inlier ratio over all points of the model and query, where an inlier is a point with a neighboring point from the other model that is closer than a threshold distance and whose normal is similar to that of the point under consideration. Figure 5 shows an alignment between a query and a database object and their corresponding EGIs. Selecting the overlap points after alignment results in precise segmentation of the object from the background.

Fig. 6. Left: The precision-recall curve for car detection on 200 million points containing 1221 cars. (Precision is the x-axis and recall the y-axis.) Right: Screenshot of detected cars. Cars are in random colors and the background in original colors.

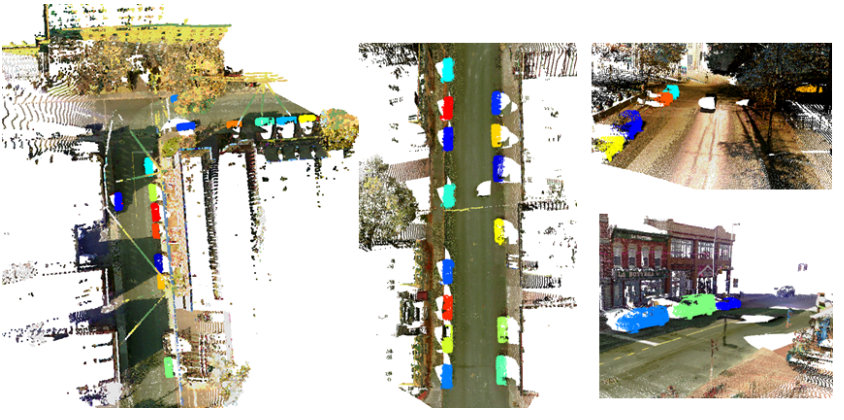

Fig. 7. Screenshots of detected cars, including views from above. (There is false negative at the bottom of the left image.) **Best viewed in color.**

6 Experimental Results

We processed very large-scale point clouds captured by a moving vehicle equipped with four range scanners and precise Geo-location sensors. The dataset consists of about 200 million points, 2.2 million of which were used for training. The training set included 17 cars which were selected as target objects. We compute 81,172 spin images for the training set (of which 2657 are parts of cars) and 6.1 million for the test set. Each spin image has a 15×15 resolution computed in a cylindrical support region with height and radius both set to $2m$. Reference points for the spin images are selected as in Section 4 with an average distance between vertices of $0.4m$. The spin images of the training set are inserted into DB_{SI}.

EGIs are computed for each target object in the training set, approximated by picking a smaller set of 200 normals, that minimize the interpolation error on all samples. The approximated EGIs are inserted into DB_{EGI}, which is a simple list with 17 entries. Since our method only requires very few representatives from each class, we were able to perform the experiments using a few sedans, SUVs and vans as models.

The query trouping threshold is set to $1m$ (Section 4). This groups points roughly up to two grid positions away. The EGI matching thresholds (Section 5.2) are set as follows: Each pair must of stars must subtend an angle of at least 30° and the two angles must not differ by more than 5°. Rotations that meet these requirements are evaluated according to Section 5.3. For the best rotation hypotheses, the metric used to make the final decision is computed: the percentage of inliers on both models after alignment. For a point to be an inlier there has to be at least one other point from the other model that is within $30cm$ and whose normal deviates by at most 35° from the normal of the current point. We have found the inlier fraction to be more useful than other distance metrics.

Results on an test area comprising 220 million points and 1221 cars are shown in Figs. 6 and 7. After bottom-up classification there were approximately 2200 detections of which about 1100 were correct. The top-down step removes about 1200 false positives and 200 true positives. The precision-recall curve as the inlier threshold varies for the full system is shown in Fig. 6. For the point marked with a star, there are 905 true positives, 74 false positives and 316 false negatives (missed detections) for a precision of 92.4% and a recall of 74.1%.

7 Conclusion

We have presented an approach for object detection from 3D point clouds that is applicable to very large datasets and requires limited training efforts. Its effectiveness is due to the combination of bottom-up and top-down mechanisms to hypothesize and test locations of potential target objects. An application of our method on car detection has achieved very satisfactory precision and recall on an area far larger than the test area of any previously published method. Moreover, besides a high detection rate, we are able to accurately segment the objects of interest from the background. We are not aware of any other methodology that obtains comparable segmentation accuracy without being trained on the same instances that are being segmented.

A limitation of our approach we intend to address is that search is linear in the number of objects in the EGI database. We are able to achieve satisfactory results with a small database, but sublinear search is a necessary enhancement to our algorithm.

Acknowledgments

This work is partially supported by DARPA under the Urban Reasoning and Geospatial ExploitatioN Technology program and is performed under National Geospatial-Intelligence Agency (NGA) Contract Number HM1582-07-C-0018. The ideas expressed herein are those of the authors, and are not necessarily endorsed by either DARPA or NGA. This material is approved for public release; distribution is unlimited.

The authors are also grateful to Ioannis Pavlidis for his help in labeling the ground truth data.

References

1. Johnson, A., Carmichael, O., Huber, D., Hebert, M.: Toward a general 3-d matching engine: Multiple models, complex scenes, and efficient data filtering. In: Image Understanding Workshop, pp. 1097–1108 (1998)
2. Carmichael, O., Huber, D., Hebert, M.: Large data sets and confusing scenes in 3-d surface matching and recognition. In: 3DIM, pp. 358–367 (1999)
3. Matei, B., Shan, Y., Sawhney, H.S., Tan, Y., Kumar, R., Huber, D., Hebert, M.: Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(7), 1111–1126 (2006)
4. Mian, A., Bennamoun, M., Owens, R.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. Pattern Analysis and Machine Intelligence 28(10), 1584–1601 (2006)
5. Correa, S.R., Shapiro, L.G., Meila, M., Berson, G., Cunningham, M.L., Sze, R.W.: Symbolic signatures for deformable shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(1), 75–90 (2006)
6. Frueh, C., Jain, S., Zakhor, A.: Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. IJCV 61(2), 159–184 (2005)
7. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 433–449 (1999)
8. Horn, B.: Extended gaussian images. Proceedings of the IEEE 72(12), 1656–1678 (1984)
9. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(2), 131–147 (1990)
10. Kang, S., Ikeuchi, K.: The complex egi: A new representation for 3-d pose determination. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(7), 707–721 (1993)
11. Hebert, M., Ikeuchi, K., Delingette, H.: A spherical representation for recognition of free-form surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(7), 681–690 (1995)
12. Dorai, C., Jain, A.K.: Cosmos: A representation scheme for 3d free-form objects. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(10), 1115–1130 (1997)
13. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Transactions on Graphics 21(4) (2002)
14. Liu, X., Sun, R., Kang, S.B., Shum, H.Y.: Directional histogram model for three-dimensional shape similarity. In: Int. Conf. on Computer Vision and Pattern Recognition (2003)
15. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Symposium on Geometry Processing (2003)
16. Makadia, A., Patterson, A.I., Daniilidis, K.: Fully automatic registration of 3d point clouds. In: Int. Conf. on Computer Vision and Pattern Recognition, vol. I, pp. 1297–1304 (2006)
17. Driscoll, J., Healy, D.: Computing fourier transforms and convolutions on the 2-sphere. Advances in Applied Mathematics 15, 202–250 (1994)

18. Stein, F., Medioni, G.: Structural hashing: Efficient three dimensional object recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 125–145 (1992)
19. Ashbrook, A., Fisher, R., Robertson, C., Werghi, N.: Finding surface correspondence for object recognition and registration using pairwise geometric histograms. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 674–686. Springer, Heidelberg (1998)
20. Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
21. Huber, D., Kapuria, A., Donamukkala, R., Hebert, M.: Parts-based 3d object classification. In: Int. Conf on Computer Vision and Pattern Recognition, vol. II, pp. 82–89 (2004)
22. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
23. Shan, Y., Sawhney, H.S., Matei, B., Kumar, R.: Shapeme histogram projection and matching for partial object recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(4), 568–577 (2006)
24. Funkhouser, T., Shilane, P.: Partial matching of 3d shapes with priority-driven search. In: Symposium on Geometry Processing (2006)
25. Medioni, G., Lee, M., Tang, C.: A Computational Framework for Segmentation and Grouping. Elsevier, New York (2000)
26. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching. Journ. of the ACM 45, 891–923 (1998)
27. Smith, D.A.: Using enhanced spherical images. Technical Report AIM-530. MIT (1979)
28. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3d objects with radial basis functions. In: SIGGRAPH, pp. 67–76. ACM, New York (2001)