IEEE/RSJ International Conference on Intelligent Robots and Systems -- IROS 2013

Detecting Objects of a Category in Range Data by Comparing to a Single Geometric Prototype

Ulrich Hillenbrand

***Abstract*—Object detection is here considered as the problem of retrieving from scene data segments that belong to objects from the sought category. The method proposed and investigated works with dense range data, as can be acquired with low-cost sensors. It does not require any training, but just a single geometric prototype that may be taken from an internet repository. Experiments with various household and office scenes are reported, and the performance is quantified on a public dataset. One of the tested variants achieves an F-score and average precision of 94% at total recall, and a correct nearest-neighbor rate of 97%.**

I. Introduction

The task of scene analysis is, broadly speaking, to assign labels to parts of the scene dataset, usually pixels, feature points, or 3D data points. The labels of interest are related to semantic categories, often generic object class labels (car, telephone, coffee mug etc.). The mainstream technique today trains classifiers, binary (for an object class versus background) or multiclass (for several object classes and background), on a very large set of labeled examples of the relevant objects and background [1, 2, 3, 4]. Considerable progress has been made along this line, on RGB images, range data, and RGB-D data.

However, it is attractive to consider systems that would be able to find instances of generic categories just by knowing a *single* example. Such a system would need a measure of similarity between instances that roughly corresponds to our human category concept. It will be dependent upon the application domain, but should be valid across all the categories of interest. The appeal of such a system lies in the great flexibility and rapid adaptability it would offer when faced with changing demands and knowledge.

This statement of a detection problem reframes scene analysis as a retrieval scenario. In the absence of classifiers that predict definite labels for objects and background, indeed without any background model at all, the best we can get is a ranked sequence of parts of the data that reflects gradual similarity to the queried prototype.

Most work on shape retrieval is about drawing relevant 3D models out of a database [5, 6]. The models are usually synthetic, complete, and lack physical scale. Shape similarity is then scale invariant and can be computed from integral shapes or their descriptors.

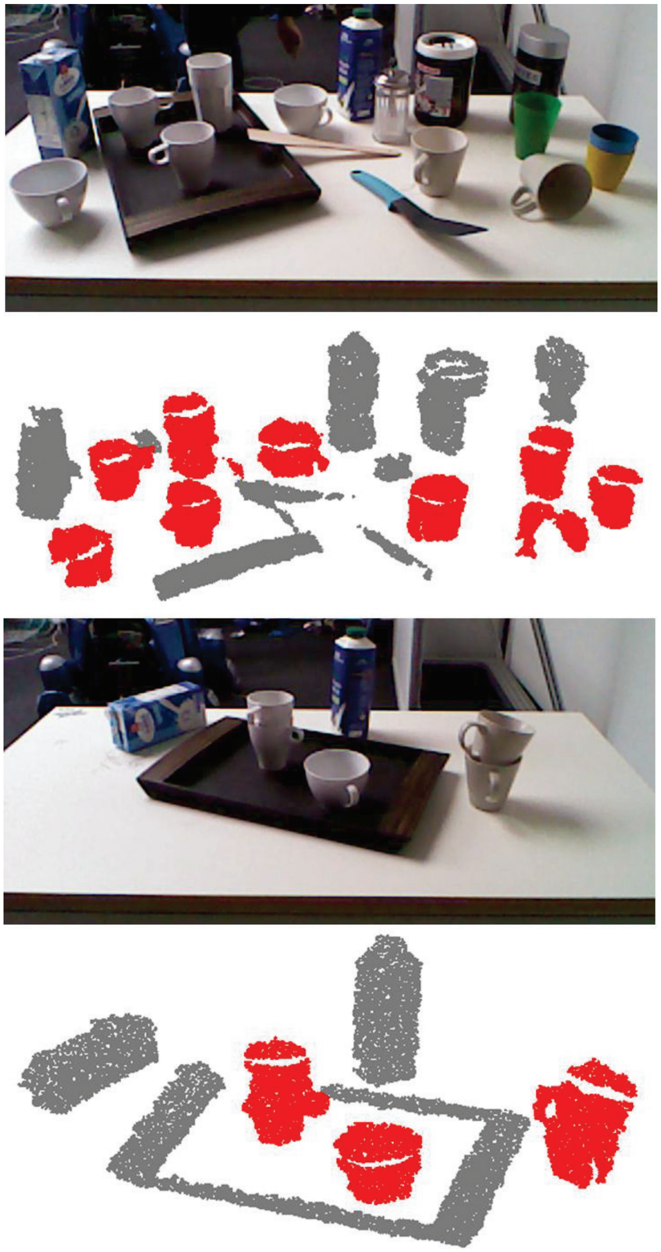

Figure 1. Two tabletop scenes and their range data, in which various kinds of cups are detected and their segments labeled (red).

In robotics, by contrast, object size is a major discriminating factor between semantic categories, determining in particular the object affordances. Additionally, objects in a real-world scene are not completely represented in the data, and the represented parts are corrupted by noise and sensing artifacts. Moreover, objects

This work was funded through European Community's Seventh Framework Programme under grant agreement number ICT-248273 (GeRT).

The author is with the Institute of Robotics and Mechatronics,
German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: Ulrich.Hillenbrand@dlr.de).

do not appear in isolation, so object candidates must be found before comparisons can be made.

In this article, a search and compare strategy for objects is investigated on scenes captured with a PrimeSense sensor. Only the range data are used, not the RGB data, so the methods are applicable to dense range sensing in general. For each sought category, a single geometric prototype is taken quite arbitrarily from the internet. The scene data are segmented into connected components, and an exploratory search for alignments of the prototype to segments is carried out. A set of similarity measures for the aligned object geometries is then tested for ranking the scene segments.

The method is tested on table-top scenes taken in our lab with encouraging results. Quantitative evaluation is done on a large public dataset. It turns out that one of the tested similarity measures achieves excellent retrieval of object instances from the diversely populated scenes.

II. Methods

We are given two dense sets of 3D data points, one from the scene to analyze and one as a geometric model, the prototype for the sought object category. The goal is to retrieve segments from the scene, ranked by shape similarity to the prototype. The following procedure is here proposed and investigated.

A. Segment the scene data into object-related components.

B. For each scene segment, compute an alignment of the geometric prototype, allowing for rigid motion and scaling.

C. Evaluate the alignments of the prototype to the scene segments by similarity of the prototype and segment shapes; rank the alignments by shape similarity.

D. For each point in each scene segment, determine the alignment from the similarity-ranked list that brings the prototype closest; the rank of a segment for retrieval is equal to the similarity rank of the alignment chosen by the majority of its points.

Each of these steps is now explained in some detail.

A. Scene Segmentation

Segmentation into object-related components is facilitated by the fact that most objects in the real world are supported by a plane. By finding and removing the supporting plane from the scene data, different objects become disconnected in data space. A simple connectivity analysis can then split the data along object boundaries; see [7] for an implementation.

This strategy brakes down in situations where different objects touch each other. A common extension then is to consider surface normal directions to still find object boundaries. This was not necessary, however, for the datasets used in this study.

The analysis of segments to follow can tolerate some over-segmentation of objects. Under-segmentation, on the other hand, may cause mislabeling of data or prevent objects from being detected.

Supporting planes may be sought globally or locally, the single strongest plane or multiple strong planes. For the datasets used in this study, featuring man-made indoor environments, a single global plane estimate turned out to be sufficient. See [4] for a similar analysis of outdoor scenes.

The present segmentation procedure is as follows.

a. Randomly draw 100,000 point triples from the scene.

b. For each point triple, compute plane parameters (plane normal and normal component of the point triple center).

c. Let the plane parameters vote for a region in the quantized parameter space.

d. Initialize a mean shift procedure [8] in the region voted for, using all the plane parameters in that region. The converged mean location is the estimate of the dominant plane.

e. Remove the data close to the dominant plane from the scene.

f. Collect the remaining scene points in a voxel grid.

g. Label the connected components on the voxel grid (26-neighborhood) as the scene segments.

h. Transfer the segment labels from each voxel onto the data points they contain.

A voxel size of 8 mm was used here for segmentation. Small segments containing less than 0.1% of all points were discarded.

B. Prototype Alignment

The geometric prototype of the sought object category is aligned with each of the scene segments. The goal is to bring corresponding parts of the scene and prototype shapes close to each other [9, 10]. The aligning transformation consists of a 6DoF rigid motion and isotropic scaling, i.e., a similarity transform. There are two stages of estimating the transform: a global pose clustering phase [11] and a local optimization phase, in the spirit of an ICP algorithm [12]. For each segment, the procedure is as follows.

a. Randomly draw 1,000,000 pairs of surflets (points with local surface normal vector) from the scene.

b. For each scene surflet pair, retrieve from a hash table a surflet pair from the prototype with roughly congruent geometry (point distance and normal orientations); compute a rigid transformation that aligns the scene surflets with the prototype surflets.

c. Compute the 6 consistent parameters (similar to the exponential rotation parameters and the translation components) [13] for all the rigid motions. Let them vote for a region in the quantized parameter space.

d. Initialize a mean shift procedure [8] in the region voted for, using all the motion parameters in that region. The converged mean location is the estimate of the initial rigid alignment.

e. Refine the initial rigid alignment through an iterative procedure: each scene point is corresponded to its closest aligned prototype point; these correspondences are used to update pose and scale by minimizing the sum of squared point distances; at most 10 updates are allowed. The final similarity transformation is the estimate of the alignment.

Because the shapes to be aligned differ, unlike in most work on alignment of 3D data, the parameters have to be chosen such that a suitable trade-off between different parts of the surfaces can be found. Hence, we need a rather coarse quantization of the pose parameter space and a large window for the mean shift procedure. In this study, the bin size was around 20 degrees along each of the rotational dimensions[1] and 10 mm along each of the translational dimensions. The initial size of the mean shift window was the same, and was shrunk as the local parameter density increased during convergence to the density maximum.

C. *Evaluation by Shape Similarity*

Having aligned the prototype to each scene segment, the shape distance can be measured by statistics of the point-wise distances between the two surfaces. However, the alignment allows for arbitrary scaling of the prototype, which does not seem right for comparing real objects, as physical scale is an important discriminating feature of object categories. Rather, the effect of scaling should be only on finding a suitable relative pose between prototype and segment. After aligning the scaled prototypes to the segments, we hence set the scaling factor of the aligning transform to unity. This corresponds to shrinking or expanding the aligned prototype about its center to its original size. The prototype is then compared to the scene segment; see Fig. 2 for an illustration.

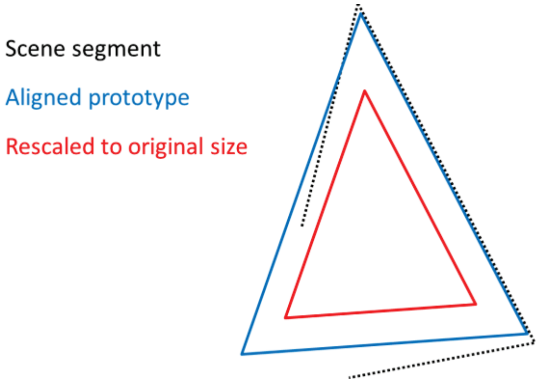

Figure 2. Illustration of a prototype being up-scaled (blue) for scene alignment and rescaled to original size (red) for shape comparison.

In this study, some variants of shape distance measures based on point-wise distances have been evaluated. Perhaps the most straightforward one takes the sum of point distances, symmetrically from scene segment to prototype and reversely,

$$d_{\mathrm{sum}}(S,P) = \sum_{s\in S} \|s - p^*(s)\| + \sum_{p\in P} \|p - s^*(p)\| \,,$$

where S is a scene segment and P is the set of points of the aligned prototype, re-scaled to original size, $p^*(s) \in P$ is the prototype point closest to $s \in S$ and $s^*(p) \in S$ is the segment point closest to $p \in P$. This measure has been used in [4]. It may be preferable, however, to normalize the sums by the number of points, making the shape distance independent from point numbers, thus obtaining

$$d_{\mathrm{mean}}(S,P) = \mathrm{mean}\{\|s - p^*(s)\| \mid s \in S\} + \mathrm{mean}\{\|p - s^*(p)\| \mid p \in P\} \,.$$

A more robust version, tolerating more local deviation between shapes, is obtained by using the median of the point distances, instead of their mean,

$$d_{\mathrm{median}}(S,P) = \mathrm{median}\{\|s - p^*(s)\| \mid s \in S\} \cup \{\|p - s^*(p)\| \mid p \in P\} \,.$$

The Hausdorff shape distance is obtained by taking the maximum of the point distances,

$$d_{\max}(S,P) = \max\{\|s - p^*(s)\| \mid s \in S\} \cup \{\|p - s^*(p)\| \mid p \in P\} \,.$$

Again, a more robust variant of the Hausdorff distance involves computing the mean of point distances separately for the two directions (segment to prototype and reversely) before taking the maximum,

$$d_{\mathrm{maxmean}}(S,P) = \max\{\mathrm{mean}\{\|s - p^*(s)\| \mid s \in S\}, \mathrm{mean}\{\|p - s^*(p)\| \mid p \in P\}\} \,.$$

Yet more robust is taking the median of point distances in each direction,

$$d_{\mathrm{maxmedian}}(S,P) = \max\{\mathrm{median}\{\|s - p^*(s)\| \mid s \in S\}, \mathrm{median}\{\|p - s^*(p)\| \mid p \in P\}\} \,.$$

Now, most of the time a segment of the scene data will represent only part of an object, while the prototype is a complete model. Therefore, a non-symmetric similarity measure that tolerates a high proportion of prototype points being distant from the segment while being strict with deviations of segment points from the prototype may be desirable. Such a measure is

$$d_{\mathrm{max2median}}(S,P) = \max\{\max\{\|s - p^*(s)\| \mid s \in S\}, \mathrm{median}\{\|p - s^*(p)\| \mid p \in P\}\} \,.$$

Each alignment of the prototype to a segment gives rise to a shape distance value by one of these measures. The alignments are then ranked according to shape distance, lower values giving higher ranks.

D. *Segment Ranking*

The final goal of the algorithm is the ranking of scene segments for retrieval. The retrieval rank of each segment is

[1] Note that a homogeneous quantization in the consistent rotation space gives rise to an inhomogeneous quantization of rotation angles [13].

inherited from the similarity rank of the alignments as follows.

a. For each point in the segment, find the alignment of the prototype that makes the minimum Euclidean distance to that point; cast a vote for that alignment.

b. The segment adopts for retrieval the similarity rank of the alignment that has received the most votes.

The rationale of this procedure is that different segments belonging to the same over-segmented object can all be retrieved together at a high rank, even if some of these segments give rise to rather low-similarity ranked alignments of the prototype. Thus, parts of an over-segmented object will be reunited at the ranking stage, if one of its segments, say, the geometrically most informative one, yields an alignment superior to those found for the others and matching all object parts. Using instead the similarity ranks from the alignments directly for segment retrieval would make it impossible to recover from over-segmentation.

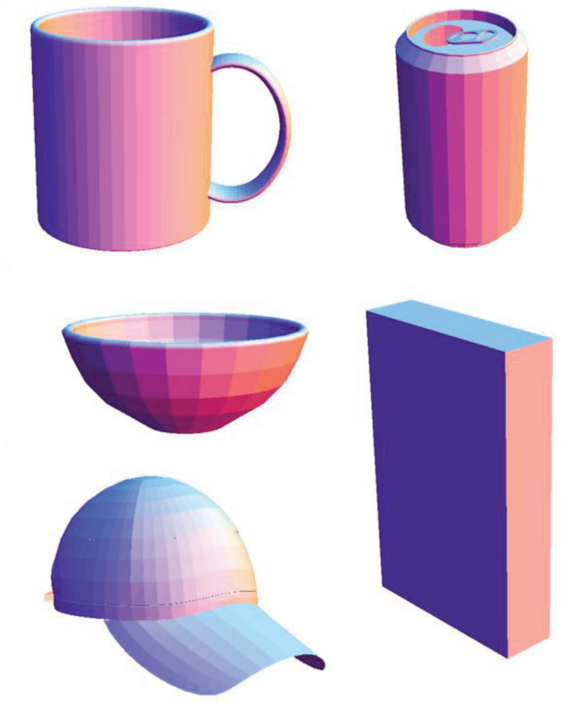

Figure 3. The five prototype shapes taken from the internet and used in the present experiments: “coffee mug”, “soda can”, “bowl”, “cereal box”, and“cap”.

III. Experiments

A. Qualitative

A series of experiments was carried out on tabletop scenes shot in our lab with a PrimeSense sensor. These scenes contained six different kinds of cups in diverse arrangements together with other objects. A synthetic 3D mug model was taken from the internet [14] as the prototype, shown in Fig. 3. Using the derived mug points as the prototype, it was possible to successfully retrieve the various cups from the scenes in sequences, without false positives or with just little contamination by small false segments, until total recall was reached. Figure 1 shows two typical example scenes and their segments retrieved as cup.

An arrangement encountered in these scenes was stacked cups. Multiple cups in a stack could mostly not be separated into different segments by the method used here. Rather, a cup stack got labeled jointly as a single cup segment. However, stacking cups with objects of a *different* type may well have produced false labeling of a potentially under-segmented scene.

Over-segmentation of the scenes often occurred, e.g. separating front and back sides of a cup. These cup parts, however, were usually retrieved together as a single entity, being properly joined at the ranking stage (section II.D.).

B. Quantitative

For collecting quantitative results, a public dataset was used. The dataset provided by Lai *et al.* [15, 3] contains 8 large scenes showing office and kitchen environments, reconstructed with data from a moving PrimeSense sensor. Apparently, the alignment of individual frames has introduced some gross errors in two of the scenes; however, this did not have a strong negative effect on the results.

In this scene labeling benchmark, the five object categories “bowl”, “cap”, “cereal box”, “coffee mug”, and “soda can” have to be detected. From these sought categories, each scene contains between 2 and 11 objects, for which ground truth labeling is provided. Altogether there are 19 different objects appearing in the 8 scenes, together with walls, doors, and furniture. In our present retrieval set up, the particular combination of objects and scenes gives rise to 29 queries with between 1 and 3 relevant objects to be returned.

The prototype for “coffee mug” was the same as used for the qualitative experiments. For “bowl”, “cap”, and “soda can”, synthetic models were likewise taken from the internet [16, 17, 18], shown in Fig. 3. The dimensions of the soda can model were adapted to fit a 330 ml Coca Cola can [19]. For “cereal box”, a box model was synthesized with dimensions equal to Kellogg's Frosted Flakes [20].

Segmentation produced between 16 and 41 data segments containing more than 0.1% of all scene points. Most segments produced were smaller than that and were hence discarded. Figure 4 shows the segments retained from three example scenes.

Segments were retrieved in ranked order from each scene until all relevant segments were found (total recall). For the three example scenes, Fig. 4 shows the resulting labels at total recall for three different queries: the correct segments got labeled in these examples.

Some performance measures were computed for the retrieval experiment: mean precision at total recall, F-score at total recall, correct nearest-neighbor rate, and average precision. They are shown in Table 1 for the seven shape distance measures defined in section II.C. Additionally, the table shows the chance level performance, i.e., what is achieved when segments are retrieved in random order from the scenes. The chance level performance gives an idea of the task complexity and of the proposed solutions’ quality.

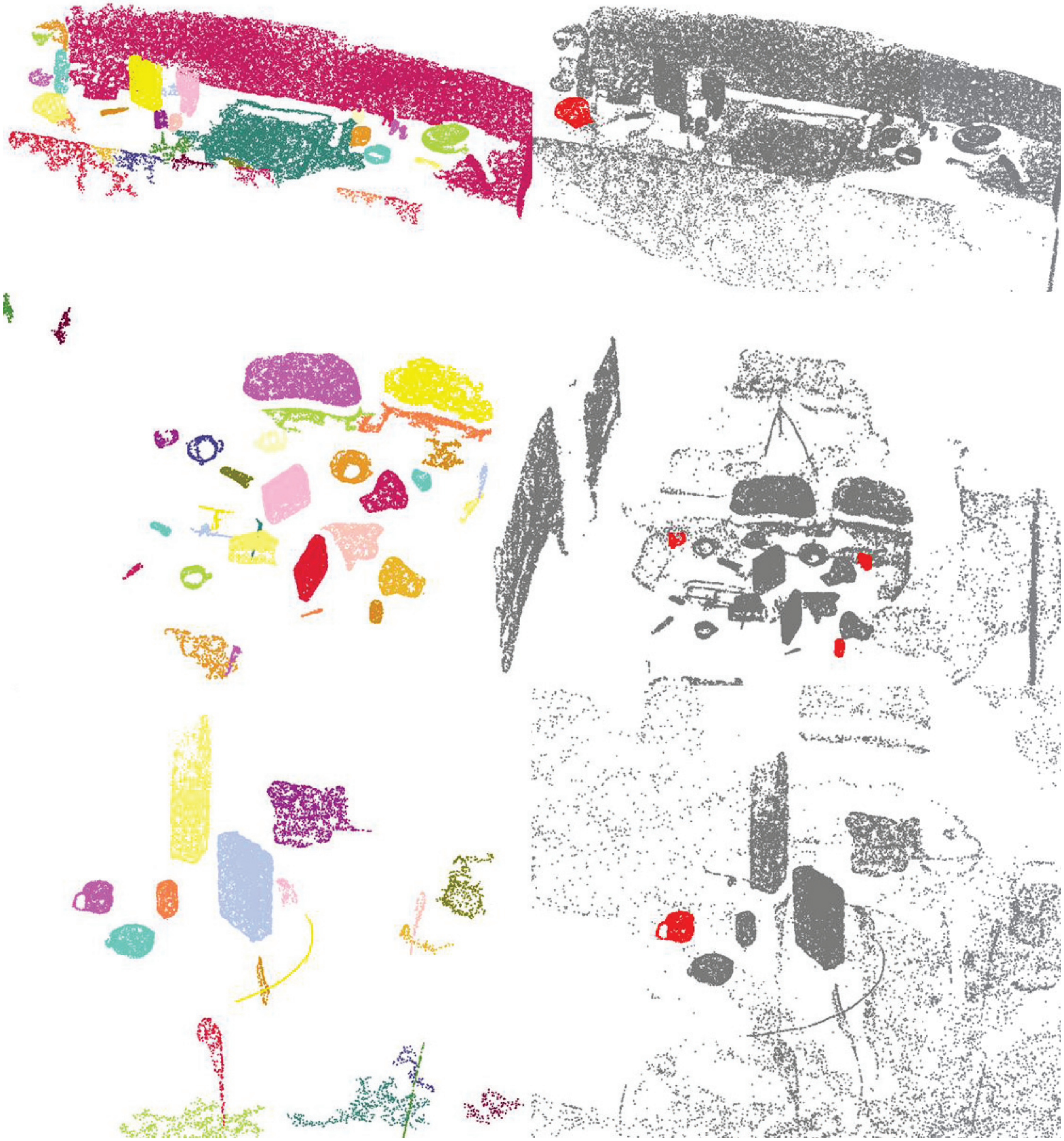

Figure 4. Three scenes from the dataset of Lai *et al.* [15, 3] (top, middle, and bottom rows). The left column shows the segments with more than 0.1% of the scene points. The right column shows segments in red color retrieved after querying for their category: a cap (top row), three soda cans (middle row), a coffee mug (bottom row). Note that the dominant supporting plane is removed from the scenes.

TABLE 1. PERFORMANCE MEASURES OF SEGMENT RETRIEVAL.

	mean precision	**F-score**	**nearest-neighbor**	**average precision**
chance level	0.15	0.26	0.05	0.16
$\mathbf{d_{sum}}$	0.73	0.84	0.62	0.74
$\mathbf{d_{mean}}$	0.78	0.88	0.79	0.83
$\mathbf{d_{median}}$	0.78	0.87	0.76	0.81
$\mathbf{d_{max}}$	0.65	0.79	0.55	0.67
$\mathbf{d_{maxmean}}$	0.72	0.84	0.59	0.72
$\mathbf{d_{maxmedian}}$	0.76	0.86	0.69	0.78
$\mathbf{d_{max2median}}$	0.89	0.94	0.97	0.94

Evidently, all the tested variants of shape distance perform well above chance level. The Hausdorff distance d_{max} is the least adequate, owing probably to the extreme influence that single outlier points have. The second weakest distance measure is d_{sum}, the one used in [4], while its normalized variant d_{mean} achieves a significant improvement. It seems, therefore, that a distance measure that grows with the number of points in the scene segment and prototype unduly penalizes object size or point density. The top performing measure from this set is, by a large margin, $d_{\text{max2median}}$, which treats distances from scene segment points to the closest prototype point and in the reverse direction differently. This measure seems to respect most adequately the fact that scene data is generally incomplete.

Although the F-score shown in Table 1 for the best variant is superior to the ones published in [3] on the same dataset, the results are in fact difficult to compare. Lai *et al.* presented experiments with a number of classifiers, trained on the five sought object classes and the background class. In particular, their training set contained 14 of the 19 specific scene objects. For classification, they optimized over a set of class labels, rather than ranking for class membership as done here. Moreover, their F-score was computed on a per-pixel basis, while the present one is per segment.

Nonetheless, it is quite interesting that such high retrieval performance for object categories can be achieved by a method that does not need any training and does not make use of the RGB channels of the dataset.

C. Computation Time

The algorithm is currently implemented in Mathemtica with some linked C functions. It was run on 5 CPU cores in parallel. For ranking the about 30 segments of an average scene, the present implementation took between 10 and 15 seconds. It is to be expected that a pure C++ implementation can reduce this time significantly, as can utilization of more CPU cores.

IV. CONCLUSION

A method for retrieval of object-related segments from scene data by category has been proposed and some variants investigated. The method is entirely based upon geometric data, i.e., shape information. It requires no training but only a single geometric prototype. It has shown excellent performance on our tabletop scenes and a public dataset of office and kitchen environments.

The limitations of the method are mainly in its critical dependence upon useful scene segmentation. If segment boundaries are unlikely to coincide with object boundaries, the presented technique may easily fail. Generally, under-segmentation seems to pose harder problems than over-segmentation, as the latter can be fixed at the segment ranking stage.

One point to be clarified in the future is the range of shape variation that can be tolerated, before the simple measures of shape similarity proposed here will no longer be informative. It is clear, however, that this range may be increased by employing more than one prototype for representing a shape category. Another interesting parameter to explore will hence be the number of geometric prototypes that define a shape category.

ACKNOWLEDGMENT

The author wishes to thank Kevin Lai and his colleagues for providing their dataset to the community.

REFERENCES

[1] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," *CVPR* 2000.

[2] M. Everingham, L. Van Gool, Ch. K. I. Williams, J. Winn, A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Computer Vision* 88, pp. 303–338, 2010.

[3] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3D scenes," *ICRA* 2012.

[4] B. Douillard, J. Underwood, V. Vlaskine, A. Quadros, and S. Singh, "A pipeline for the segmentation and classification of 3D point clouds," *ISER* 2010.

[5] J. W. H. Tangelder, R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications* 39, pp 441-471, 2008.

[6] M. Ovsjanikov, A. M. Bronstein, M. M. Bronstein, L. J. Guibas, "Shape Google: a computer vision approach to isometry invariant shape retrieval," *ICCV Workshops* 2009.

[7] www.pointclouds.org/documentation/tutorials/cluster_extraction.php

[8] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Patt. Anal. Mach. Intell.* 24, pp. 603–619, 2002.

[9] U. Hillenbrand, "Non-parametric 3D shape warping," *ICPR* 2010.

[10] U. Hillenbrand and M. A. Roa, "Transferring functional grasps through contact warping and local replanning," *IROS* 2012.

[11] U. Hillenbrand and A. Fuchs, "An experimental study of four variants of pose clustering from dense range data," *Computer Vision and Image Understanding* 115, pp. 1427–1448, 2011.

[12] P. J. Besl and N.D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Patt. Anal. Mach. Intell.* 14, pp. 239–256, 1992.

[13] U. Hillenbrand, "Consistent parameter clustering: definition and analysis," *Pattern Recognition Letters* 28, pp. 1112–1122, 2007.

[14] sketchup.google.com/3dwarehouse/details?mid=e94e46bc5833f2f5e57b873e4f3ef3a4&prevstart=12

[15] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," *ICRA* 2011.

[16] sketchup.google.com/3dwarehouse/details?mid=68582543c4c6d0bccfdfe3f21f42a111&prevstart=12

[17] sketchup.google.com/3dwarehouse/details?mid=d18c3ce1f186c16976bb31db0358e9c6&prevstart=0

[18] sketchup.google.com/3dwarehouse/details?mid=3a7d8f866de1890bab97e834e9ba876c&prevstart=0

[19] wiki.answers.com/Q/What_is_the_dimensions_of_a_soda_can

[20] www.kelloggstore.com/store/p/795-Frosted-Flakes-Photo-On-A-Box.html