Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram

Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, John Hsu
Willow Garage
68 Willow Rd., Menlo Park, CA 94025, USA
{rusu, bradski, thibaux, hsu}@willowgarage.com

***Abstract*— We present the Viewpoint Feature Histogram (VFH), a descriptor for 3D point cloud data that encodes geometry and viewpoint. We demonstrate experimentally on a set of 60 objects captured with stereo cameras that VFH can be used as a distinctive signature, allowing simultaneous recognition of the object and its pose. The pose is accurate enough for robot manipulation, and the computational cost is low enough for real time operation. VFH was designed to be robust to large surface noise and missing depth information in order to work reliably on stereo data.**

I. Introduction

As part of a long term goal to develop reliable capabilities in the area of perception for mobile manipulation, we address a table top manipulation task involving objects that can be manipulated by one robot hand. Our robot is shown in Fig. 1. In order to manipulate an object, the robot must reliably identify it, as well as its 6 degree-of-freedom (6DOF) pose. This paper proposes a method to identify both at the same time, reliably and at high speed.

We make the following assumptions.

- Objects are rigid and relatively Lambertian. They can be shiny, but not reflective or transparent.
- Objects are in light clutter. They can be easily segmented in 3D and can be grabbed by the robot hand without obstruction.
- The item of interest can be grabbed directly, so it is not occluded.
- Items can be grasped even given an approximate pose. The gripper on our robot can open to 9cm and each grip is 2.5cm wide which allows an object 8.5cm wide object to be grasped when the pose is off by +/- 10 degrees.

Despite these assumptions our problem has several properties that make the task difficult.

- The objects need not contain texture.
- Our dataset includes objects of very similar shapes, for example many slight variations of typical wine glasses.
- To be usable, the recognition accuracy must be very high, typically much higher than, say, for image retrieval tasks, since false positives have very high costs and so must be kept extremely rare.
- To interact usefully with humans, recognition cannot take more than a fraction of a second. This puts constraints on computation, but more importantly this precludes the use of accurate but slow 3D acquisition using lasers. Instead we rely on stereo data, which suffers from higher noise and missing data.

Fig. 1. A PR2 robot from Willow Garage, showing its grippers and stereo cameras

Our focus is perception for *mobile* manipulation. Working on a mobile versus a stationary robot means that we can't depend on instrumenting the external world with active vision systems or special lighting, but we can put such devices on the robot. In our case, we use projected texture[1] to yield dense stereo depth maps at 30Hz. We also cannot ensure environmental conditions. We may move from a sunlit room to a dim hallway into a room with no light at all. The projected texture gives us a fair amount of resilience to local lighting conditions as well.

[1]Not structured light, this is random texture

Although this paper focuses on 3D depth features, 2D imagery is clearly important, for example for shiny and transparent objects, or to distinguish items based on texture such as telling apart a Coke can from a Diet Coke can. In our case, the textured light alternates with no light to allow for 2D imagery aligned with the texture based dense depth, however adding 2D visual features will be studied in future work. Here, we look for an effective purely 3D feature.

Our philosophy is that one should use or design a recognition algorithm that fits one's engineering needs such as scalability, training speed, incremental training needs, and so on, and then find features that make the recognition performance of that architecture meet one's specifications. For reasons of online training, and because of large memory availability, we choose fast approximate K-Nearest Neighbors (K-NN) implemented in the FLANN library [1] as our recognition architecture. The key contribution of this paper is then the design of a new, computationally efficient 3D feature that yields object recognition and 6DOF pose.

The structure of this paper is as follows: Related work is described in Section II. Next, we give a brief description of our system architecture in Section III. We discuss our surface normal and segmentation algorithm in Section IV followed by a discussion of the Viewpoint Feature Histogram in Section V. Experimental setup and resulting computational and recognition performance are described in Section VI. Conclusions and future work are discussed in Section VII.

II. Related Work

The problem that we are trying to solve requires global (3D object level) classification based on estimated features. This has been under investigation for a long time in various research fields, such as computer graphics, robotics, and pattern matching, see [2]–[4] for comprehensive reviews. We address the most relevant work below.

Some of the widely used 3D point feature extraction approaches include: spherical harmonic invariants [5], spin images [6], curvature maps [7], or more recently, Point Feature Histograms (PFH) [8], and conformal factors [9]. Spherical harmonic invariants and spin images have been successfully used for the problem of object recognition for densely sampled datasets, though their performance seems to degrade for noisier and sparser datasets [4]. Our stereo data is noisier and sparser than typical line scan data which motivated the use of our new features. Conformal factors are based on conformal geometry, which is invariant to isometric transformations, and thus obtains good results on databases of watertight models. Its main drawback is that it can only be applied to manifold meshes which can be problematic in stereo. Curvature maps and PFH descriptors have been studied in the context of local shape comparisons for data registration. A side study [10] applied the PFH descriptors to the problem of surface classification into 3D geometric primitives, although only for data acquired using precise laser sensors. A different point fingerprint representation using the projections of geodesic circles onto the tangent plane at a point $\boldsymbol{p}_i$ was proposed in [11] for the problem of surface registration. As the authors note, geodesic distances are more sensitive to surface sampling noise, and thus are unsuitable for real sensed data without a priori smoothing and reconstruction. A decomposition of objects into parts learned using spin images is presented in [12] for the problem of vehicle identification.

Methods relying on global features include descriptors such as Extended Gaussian Images (EGI) [13], eigen shapes [14], or shape distributions [15]. The latter samples statistics of the entire object and represents them as distributions of shape properties, however they do not take into account how the features are distributed over the surface of the object. Eigen shapes show promising results but they have limits on their discrimination ability since important higher order variances are discarded. EGIs describe objects based on the unit normal sphere, but have problems handling arbitrarily curved objects.

The work in [16] makes use of spin-image signatures and normal-based signatures to achieve classification rates over 90% with synthetic and CAD model datasets. The datasets used however are very different than the ones acquired using noisy 640×480 stereo cameras such as the ones used in our work. In addition, the authors do not provide timing information on the estimation and matching parts which is critical for applications such as ours. A system for fully automatic 3D model-based object recognition and segmentation is presented in [17] with good recognition rates of over 95% for a database of 55 objects. Unfortunately, the computational performance of the proposed method is not suitable for real-time as the authors report the segmentation of an object model in a cluttered scene to be around 2 minutes. Moreover, the objects in the database are scanned using a high resolution Minolta scanner and their geometric shapes are very different. As shown in Section VI, the objects used in our experiments are much more similar in terms of geometry, so such a registration-based method would fail. In [18], the authors propose a system for recognizing 3D objects in photographs. The techniques presented can only be applied in the presence of texture information, and require a cumbersome generation of models in an offline step, which makes this unsuitable for our work.

As previously presented, our requirements are real-time object recognition and pose identification from noisy real-world datasets acquired using projective texture stereo cameras. Our 3D object classification is based on an extension of the recently proposed Fast Point Feature Histogram (FPFH) descriptors [8], which record the relative angular directions of surface normals with respect to one another. The FPFH performs well in classification applications and is robust to noise but it is invariant to viewpoint.

This paper proposes a novel descriptor that encodes the viewpoint information and has two parts: (1) an extended FPFH descriptor that achieves $O(k*n)$ to $O(n)$ speed up over FPFHs where n is the number of points in the point cloud and k is how many points used in each local neighborhood; (2) a new signature that encodes important statistics between the viewpoint and the surface normals on the object. We call

this new feature the Viewpoint Feature Histogram (VFH) as detailed below.

III. Architecture

Our system architecture employs the following processing steps:

- Synchronized, calibrated and epipolar aligned left and right images of the scene are acquired.
- A dense depth map is computed from the stereo pair.
- Surface normals in the scene are calculated.
- Planes are identified and segmented out and the remaining point clouds from non-planar objects are clustered in Euclidean space.
- The Viewpoint Feature Histogram (VFH) is calculated over large enough objects (here, objects having at least 100 points).
  - If there are multiple objects in a scene, they are processed front to back relative to the camera.
  - Occluded point clouds with less than 75% of the number of points of the frontal objects are noted but not identified.
- Fast approximate K-NN is used to classify the object and its view.

Some steps from the early processing pipeline are shown in Figure 2. Shown left to right, top to bottom in that figure are: a moderately complex scene with many different vertical and horizontal surfaces, the resulting depth map, the estimated surface normals and the objects segmented from the planar surfaces in the scene.

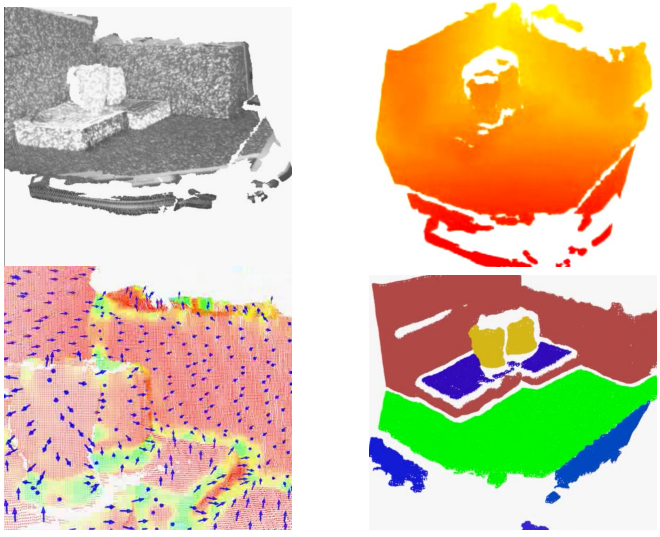


Fig. 2. Early processing steps row wise, top to bottom: A scene, its depth map, surface normals and segmentation into planes and outlier objects.

For computing 3D depth maps, we use 640x480 stereo with textured light. The texture flashes on only very briefly as the cameras take a picture resulting in lights that look dim to the human eye but bright to the camera. Texture flashes only every other frame so that raw imagery without texture can be gathered alternating with densely textured scenes. The stereo has a 38 degree field of view and is designed for close in manipulation tasks, thus the objects that we deal with are from 0.5 to 1.5 meters away. The stereo algorithm that we use was developed in [19] and uses the implementation in the OpenCV library [20] as described in detail in [21], running at 30Hz.

IV. Surface Normals and 3D Segmentation

We employ segmentation prior to the actual feature estimation because in robotic manipulation scenarios we are only interested in certain precise parts of the environment, and thus computational resources can be saved by tackling only those parts. Here, we are looking to manipulate reachable objects that lie on horizontal surfaces. Therefore, our segmentation scheme proceeds at extracting these horizontal surfaces first.

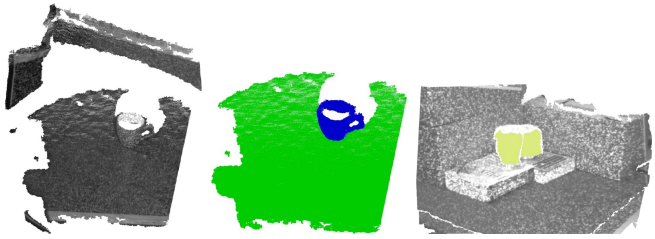


Fig. 3. From left to right: raw point cloud dataset, planar and cluster segmentation, more complex segmentation.

Compared to our previous work [22], we have improved the planar segmentation algorithms by incorporating surface normals into the sample selection and model estimation steps. We also took care to carefully build SSE aligned data structures in memory for any computationally expensive operation. By rejecting candidates which do not support our constraints, our system can segment data at about 7Hz, including normal estimation, on a regular Core2Duo laptop using a single core. To get frame rate performance (realtime), we use a voxelized data structure over the input point cloud and downsample with a leaf size of 0.5cm. The surface normals are therefore estimated only for the downsampled result, but using the information in the original point cloud. The planar components are extracted using a RMSAC (Randomized MSAC) method that takes into account weighted averages of distances to the model together with the angle of the surface normals. We then select candidate table planes using a heuristic combining the number of inliers which support the planar model as well as their proximity to the camera viewpoint. This approach emphasizes the part of the space where the robot manipulators can reach and grasp the objects.

The segmentation of object candidates supported by the table surface is performed by looking at points whose projection falls inside the bounding 2D polygon for the table, and applying single-link clustering. The result of these processing steps is a set of Euclidean point clusters. This works to reliably segment objects that are separated by about half their

minimum radius from each other. An example can be seen in Figure 3.

To resolve further ambiguities with respect to the chosen candidate clusters, such as objects stacked on other planar objects (such as books), we repeat the previously mentioned step by treating each additional horizontal planar structure on top of the table candidates as a table itself and repeating the segmentation step (see results in Figure 3).

We emphasize that this segmentation step is of extreme importance for our application, because it allows our methods to achieve favorable computational performances by extracting only the regions of interest in a scene (i.e., objects that are to be manipulated, located on horizontal surfaces). In cases where our "*light clutter*" assumption does not hold and the geometric Euclidean clustering is prone to failure, a more sophisticated segmentation scheme based on texture properties could be implemented.

V. Viewpoint Feature Histogram

In order to accurately and robustly classify points with respect to their underlying surface, we borrow ideas from the recently proposed Point Feature Histogram (PFH) [10]. The PFH is a histogram that collects the pairwise pan, tilt and yaw angles between every pair of normals on a surface patch (see Figure 4). In detail, for a pair of 3D points $\langle \boldsymbol{p}_i, \boldsymbol{p}_j \rangle$, and their estimated surface normals $\langle \boldsymbol{n}_i, \boldsymbol{n}_j \rangle$, the set of normal angular deviations can be estimated as:

$$\begin{aligned} \alpha &= \mathsf{v} \cdot \boldsymbol{n}_j \\ \phi &= \mathsf{u} \cdot \frac{(\boldsymbol{p}_j - \boldsymbol{p}_i)}{d} \\ \theta &= \arctan(\mathsf{w} \cdot \boldsymbol{n}_j, \mathsf{u} \cdot \boldsymbol{n}_j) \end{aligned} \tag{1}$$

where $\mathsf{u}, \mathsf{v}, \mathsf{w}$ represent a Darboux frame coordinate system chosen at $\boldsymbol{p}_i$. Then, the Point Feature Histogram at a patch of points $\mathcal{P} = \{p_i\}$ with $i = \{1 \cdots n\}$ captures all the sets of $\langle \alpha, \phi, \theta \rangle$ between all pairs of $\langle \boldsymbol{p}_i, \boldsymbol{p}_j \rangle$ from $\mathcal{P}$, and bins the results in a histogram. The bottom left part of Figure 4 presents the selection of the Darboux frame and a graphical representation of the three angular features.

Because all possible pairs of points are considered, the computation complexity of a PFH is $O(n^2)$ in the number of surface normals n. In order to make a more efficient algorithm, the Fast Point Feature Histogram [8] was developed. The FPFH measures the same angular features as PFH, but estimates the sets of values only between every point and its k nearest neighbors, followed by a reweighting of the resultant histogram of a point with the neighboring histograms, thus reducing the computational complexity to $O(k * n)$.

Our past work [22] has shown that a global descriptor (GFPFH) can be constructed from the classification results of many local FPFH features, and used on a wide range of confusable objects (20 different types of glasses, bowls, mugs) in 500 scenes achieving 96.69% on object class recognition. However, the categorized objects were only split into 4 distinct classes, which leaves the scaling problem open. Moreover, the GFPFH is susceptible to the errors of the local classification results, and is more cumbersome to estimate.

In any case, for manipulation, we require that the robot not only identifies objects, but also recognizes their 6DOF poses for grasping. FPFH is invariant both to object scale (distance) and object pose and so cannot achieve the latter task.

In this work, we decided to leverage the strong recognition results of FPFH, but to add in viewpoint variance while retaining invariance to scale, since the dense stereo depth map gives us scale/distance directly. Our contribution to the problem of object recognition and pose identification is to extend the FPFH to be estimated for the entire object cluster (as seen in Figure 4), and to compute additional statistics between the viewpoint direction and the normals estimated at each point. To do this, we used the key idea of mixing the viewpoint direction directly into the relative normal angle calculation in the FPFH. Figure 6 presents this idea with the new feature consisting of two parts: (1) a viewpoint direction component (see Figure 5) and (2) a surface shape component comprised of an extended FPFH (see Figure 4).

The viewpoint component is computed by collecting a histogram of the angles that the viewpoint direction makes with each normal. Note, we do not mean the view angle **to** each normal as this would not be scale invariant, but instead we mean the angle between the central viewpoint direction translated to each normal. The second component measures the relative pan, tilt and yaw angles as described in [8], [10] but now measured between the viewpoint direction at the central point and each of the normals on the surface. We call the new assembled feature the Viewpoint Feature Histogram (VFH). Figure 6 presents the resultant assembled VFH for a random object.

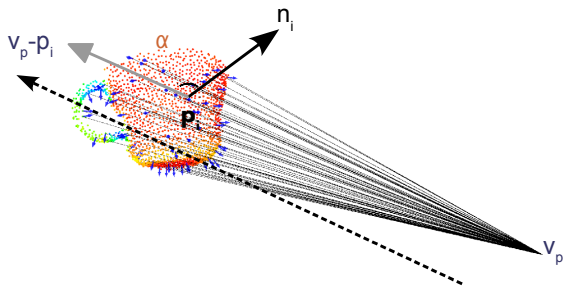


Fig. 5. The Viewpoint Feature Histogram is created from the extended Fast Point Feature Histogram as seen in Figure 4 together with the statistics of the relative angles between each surface normal to the central viewpoint direction.

The computational complexity of VFH is $O(n)$. In our experiments, we divided the viewpoint angles into 128 bins and the α, ϕ and θ angles into 45 bins each or a total of 263 dimensions. The estimation of a VFH takes about 0.3ms on average on a 2.23GHz single core of a Core2Duo machine using optimized SSE instructions.

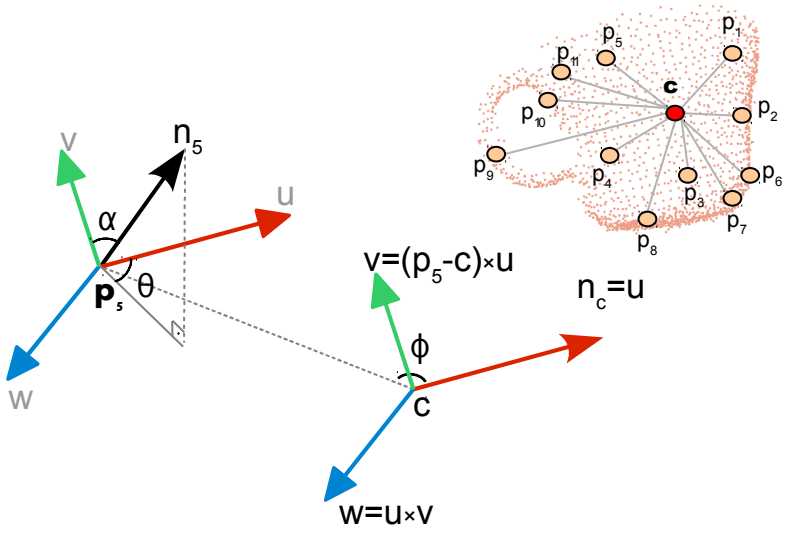


Fig. 4. The extended Fast Point Feature Histogram collects the statistics of the relative angles between the surface normals at each point to the surface normal at the centroid of the object. The bottom left part of the figure describes the three angular feature for an example pair of points.

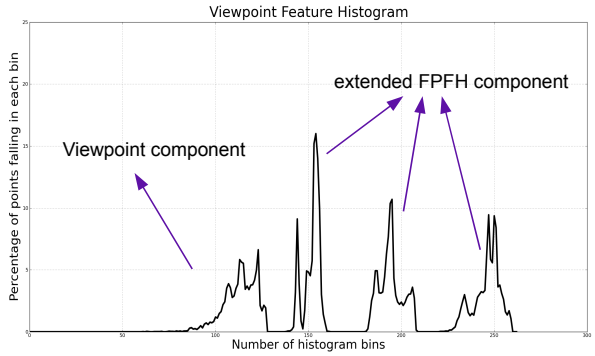


Fig. 6. An example of the resultant Viewpoint Feature Histogram for one of the objects used. Note the two concatenated components.

Fig. 7. The turn table used to collect views of objects with known orientation.

VI. Validation and Experimental Results

To evaluate our proposed descriptor and system architecture, we collected a large dataset consisting of over 60 IKEA kitchenware objects as show in Figure 8. These objects consisted of many kinds each of: wine glasses, tumblers, drinking glasses, mugs, bowls, and a couple of boxes. In each of these categories, many of the objects were distinguished only by subtle variations in shape as can be seen for example in the confusions in Figure 10. We captured over 54000 scenes of these objects by spinning them on a turn table 180°[2] at each of 2 offsets on a platform that tilted 0, 8, 16, 22 and 30 degrees. Each 180° rotation was captured with about 90 images. The turn table is shown in Fig. 7. We additionally worked with a subset of 20 objects in 500 lightly cluttered scenes with varying arrangements of horizontal and vertical surfaces, using the same data set provided by in [22]. No pose information was available for this second dataset so we only ran experiments separately for object recognition results.

The complete source code used to generate our experimental results together with both object databases are available under a BSD open source license in our ROS repository at Willow Garage [3]. We are currently taking steps towards creating a web page with complete tutorials on how to fully replicate the experiments presented herein.

Both the objects in the [22] dataset as well as the ones we acquired, constitute valid examples of objects of daily use that our robot needs to be able to reliably identify and manipulate. While 60 objects is far from the number of objects the robot eventually needs to be able to recognize, it may be enough if we assume that the robot knows what

[2]We didn't go 360 degrees so that we could keep the calibration box in view

[3]http://ros.org

Fig. 8. The complete set of IKEA objects used for the purpose of our experiments. All transparent glasses have been painted white to obtain 3D information during the acquisition process.

TABLE I

RESULTS FOR OBJECT RECOGNITION AND POSE DETECTION OVER 54000 SCENES PLUS 500 LIGHTLY CLUTTERED SCENES.

Method	Object Recognition	Pose Estimation
VFH	98.52%	98.52%
Spin	75.3%	61.2%

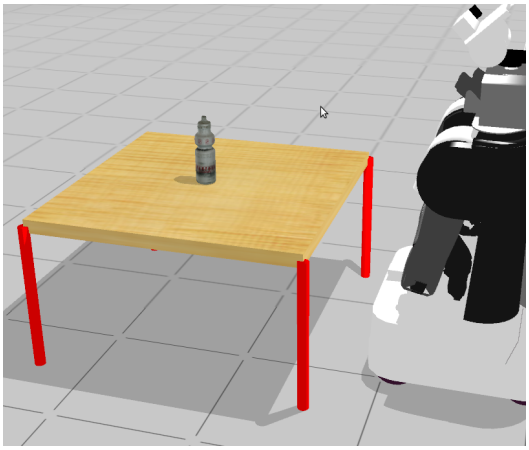
context (kitchen table, workbench, coffee table) it is in, so that it needs only discriminate among a small context dependent set of objects.

The geometric variations between objects are subtle, and the data acquired is noisy due to the stereo sensor characteristics, yet the perception system has to work well enough to differentiate between, say, glasses that look similar but serve different purposes (e.g., a wine glass versus a brandy glass).

As presented in Section II, the performance of the 3D descriptors proposed in the literature degrade on noisier datasets. One of the most popular 3D descriptor to date used on datasets acquired using sensing devices similar to ours (e.g., similar noise characteristics) is the spin image [6]. To validate the VFH feature we thus compare it to the spin image, by running the same experiments multiple times.

For the reasons given in Section I, we base our recognition architecture on fast approximate K-Nearest Neighbors (KNN) searches using kd-trees [1]. The construction of the tree and the search of the nearest neighbors places an equal weight on each histogram bin in the VFH and spin images features.

Figure 11 shows time stop sequentially aggregated examples of the training set. Figure 12 shows example recognition results for VFH. And finally, Figure 10 gives some idea of the performance differences between VFH and spin images. The object recognition rates over the lightly cluttered dataset were 98.1% for VFH and 73.2% for spin images. The overall recognition rates for VFH and Spin images are shown in Table I where VFH handily outperforms spin images for both object recognition and pose.



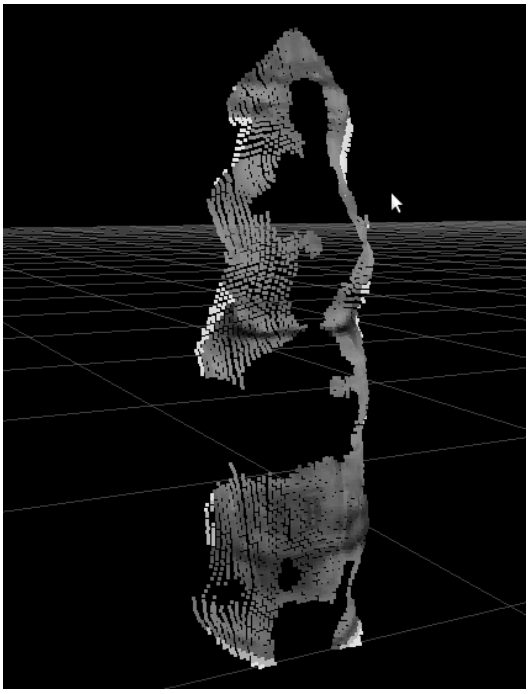


Fig. 9. Data training performed in simulation. The figure presents a snapshot of the simulation with a water bottle from the object model database and the corresponding stereo point cloud output.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented a novel 3D feature descriptor, the Viewpoint Feature Histogram (VFH), useful for object recognition and 6DOF pose identification for application where a priori segmentation is possible. The high recognition performance and fast computational properties, demonstrated the superiority of VFH over spin images on a large scale dataset consisting of over 54000 scenes with over 60 objects. Compared to other similar initiatives, our architecture works well with noisy data acquired using standard stereo cameras in real-time, and can detect subtle variations in the geometry of objects. Moreover, we presented an integrated approach for both recognition and 6DOF pose identification for untextured objects, the latter being of extreme importance for mobile manipulation and grasping applications.

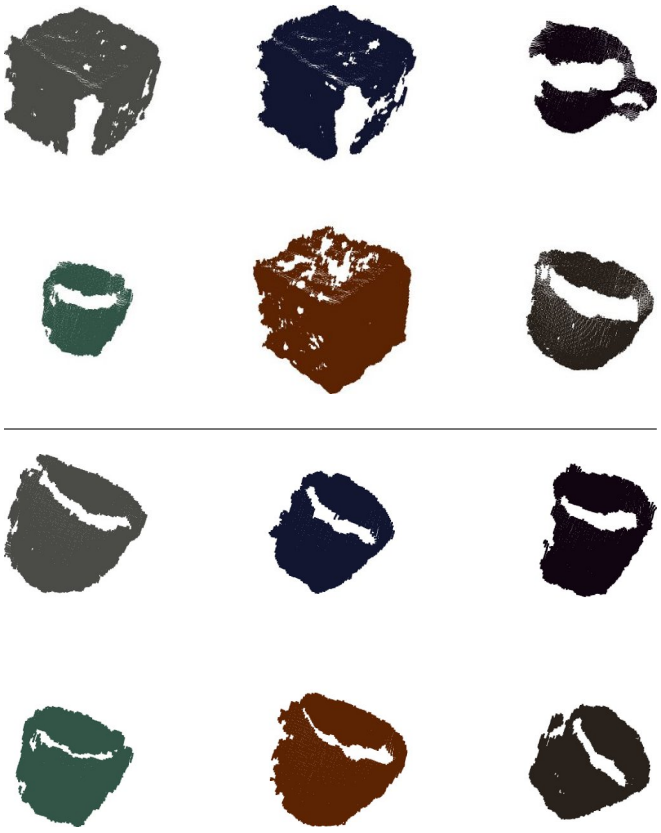


Fig. 10. VFH consistently outperforms spin images for both recognition and for pose. The bottom of the figure presents an example result of VFH run on a mug. The bottom left corner is the learned models and the matches go from best to worse from left to right across the bottom followed by left to right across the top. The top part of the figure presents the results obtained using a spin image. For VFH, 3 of 5 object recognition and 3 of 5 pose results are correct. For spin images, 2 of 5 object recognition results are correct and 0 of 5 pose results are correct.

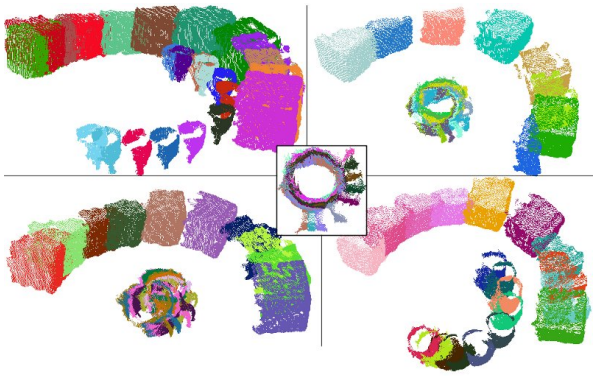


Fig. 11. Sequence examples of object training with calibration box on the outside.

An automatic training pipeline can be integrated with our 3D simulator based on Gazebo [23] as depicted in figure 9, where the stereo point cloud is generated from perfectly rectified camera images.

We are currently working on making both the fully annotated database of objects together with the source code of VFH available to the research community as open source. The preliminary results of our efforts can already be checked from the trunk of our Willow Garage ROS repository, but we are taking steps towards generating a set of tutorials on how to replicate and extend the experiments presented in this paper.

REFERENCES

[1] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP*, 2009.

[2] J. W. Tangelder and R. C. Veltkamp, "A Survey of Content Based 3D Shape Retrieval Methods," in *SMI '04: Proceedings of the Shape Modeling International*, 2004, pp. 145–156.

[3] A. K. Jain and C. Dorai, "3D object recognition: Representation and matching," *Statistics and Computing*, vol. 10, no. 2, pp. 167–182, 2000.

[4] A. D. Bimbo and P. Pala, "Content-based retrieval of 3D models," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 20–43, 2006.

[5] G. Burel and H. Hénocq, "Three-dimensional invariants and their application to object recognition," *Signal Process.*, vol. 45, no. 1, pp. 1–22, 1995.

[6] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 1999.

[7] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature Maps for Local Shape Comparison," in *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005 (SMI' 05)*, 2005, pp. 246–255.

[8] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *ICRA*, 2009.

[9] B.-C. M. and G. C., "Characterizing shape using conformal factors," in *Eurographics Workshop on 3D Object Retrieval*, 2008.

[10] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning Informative Point Classes for the Acquisition of Object Model Maps," in *In Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2008.

[11] Y. Sun and M. A. Abidi, "Surface matching by 3D point's fingerprint," in *Proc. IEEE Int'l Conf. on Computer Vision*, vol. II, 2001, pp. 263–269.

[12] D. Huber, A. Kapuria, R. R. Donamukkala, and M. Hebert, "Parts-based 3D object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 04)*, June 2004.

[13] B. K. P. Horn, "Extended Gaussian Images," *Proceedings of the IEEE*, vol. 72, pp. 1671–1686, 1984.

[14] R. J. Campbell and P. J. Flynn, "Eigenshapes for 3D object recognition in range data," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, pp. 505–510.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, pp. 807–832, 2002.

[16] X. Li and I. Guskov, "3D object recognition from range images using pyramid matching," in *ICCV07*, 2007, pp. 1–6.

[17] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 28, pp. 1584–1601, 2006.

[18] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, p. 2006, 2006.

[19] K. Konolige, "Small vision systems: hardware and implementation," in *In Eighth International Symposium on Robotics Research*, 1997, pp. 111–116.

[20] "OpenCV, Open source Computer Vision library," in *http://opencv.willowgarage.com/wiki/*, 2009.

[21] G. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library," in *O'Reilly Media, Inc.*, 2008, pp. 415–453.

[22] R. B. Rusu, A. Holzbach, M. Beetz, and G. Bradski, "Detecting and segmenting objects for mobile manipulation," in *ICCV S3DV workshop*, 2009.

[23] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

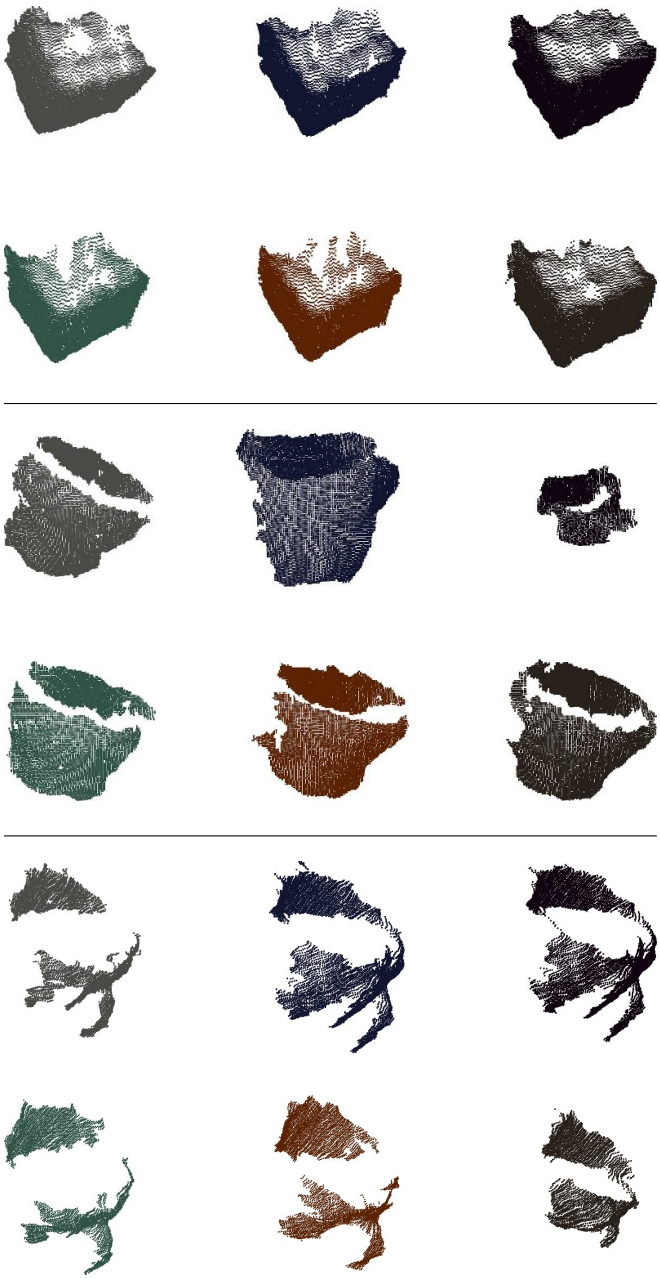


Fig. 12. VFH Retrieval results: The model pose and object is at lower left in each frame. The matches in order of best to worst go from left to right starting at bottom left followed by top right in each frame. We see that the box and the mug (top and bottom) match perfectly while a glass (middle) has 3 correct matches followed by the 4th match having the wrong view and 5th match selecting the wrong object.