2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Surface-based General 3D Object Detection and Pose Estimation

Zhou Teng
Department of Computer Science
University of North Carolina at Charlotte
Email: tzhou1@uncc.edu

Jing Xiao
Department of Computer Science
University of North Carolina at Charlotte
Email: xiao@uncc.edu

***Abstract*—3D object detection and pose estimation often requires a 3D object model, and even so, it is a difficult problem if the object is heavily occluded in a cluttered scene. In this paper, we introduce a novel approach for recognizing and localizing 3D objects based on their appearances through segmentation of 3D surfaces. The approach can identify multiple occluded objects in a scene, which may include different instances of the same object, and estimate the pose of each entire object even if the object can only be seen partially due to occlusion.**

I. Introduction

Automatic identification and localization (or pose estimation) of a target object through sensing is necessary for autonomous robotic manipulation tasks. However, general object recognition in a cluttered 3D environment is still an unsolved and challenging problem in computer/robot vision, due to occlusions, complicated background, and great variation in object appearance caused by different illumination conditions or viewpoints. To further estimate the pose of an identified object, which consists of the position and orientation of the object in the 3D scene, is even more difficult. The emergence of advanced and cost-effective RGB-D cameras, such as the PrimeSense 3D sensor and Microsoft Kinect, which capture both vision and depth information, provide new opportunities for researchers to seek effective and efficient solutions to those problems.

In this paper, we introduce a new, appearance-based approach to general 3D object detection and pose estimation based on surface features, taking full advantage of RGB-D information. We first establish a database of objects, where each object is captured by RGB-D images from different viewpoints. From such images, each object is represented in terms of automatically segmented 3D smooth surfaces with feature descriptions and an automatically constructed 3D model through registration of these surfaces across the images of different viewpoints. The object frame (i.e., coordinate system) is described via the constructed 3D model. Based on automatic 3D surface segmentation of a scene, our approach identifies known object surfaces and subsequently both identifies the corresponding objects and their homogeneous transformation matrices w.r.t. the camera coordinate system in the 3D scene, even if the objects are partially occluded. Our approach can also effectively solve the problem of detection and pose estimation of multiple instances of the same object in a scene, which can be in occlusion of one another. The key characteristic of our approach is that we build object detection and pose estimation on the basis of smooth 3D surface segments and their visual signatures.

Note that we use both geometric and visual features to characterize an object surface in order to take advantage of the fact that many daily objects, such as cereal boxes, usually have common shapes made of smooth surfaces with a lot of visual information, such as brands, some graphics, and product descriptions. Our approach combines geometrical and visual characteristics of object surfaces to best identify and localize objects partially occluded in a cluttered scene.

The paper is organized as follows. We discuss related work in section II, describe our detailed approach in section III, and provide the results and analysis of our experiments in section IV. We conclude the paper in section V.

II. Related Work

Given an RGB-D image of a real scene, in order to identify known objects and estimate their poses, the key problem is to find the robust correspondences between the current scene and trained 3D models of known objects.

Some recent work [7, 4, 3] proposed to extract and describe key points and key point pairs, and match them directly on the point clouds captured by RGB-D cameras. However, if a point cloud has a very high resolution or contains many geometric and texture details, many unnecessary key points can be extracted, incurring a high computation cost. On the other hand, using de-sampling algorithms here can lead to too few key points or key point pairs for matching if the target object is heavily occluded, and the consequence could be inaccurate or even incorrect pose estimation. Using sliding window methods [6, 9, 11, 12] as preprocessing can help reduce the candidate area for key points extraction and matching. Nevertheless, as a sliding window is usually a general rectangular area for any target object, it often contains some parts of the background or other objects that occlude one another in a cluttered scene.

Features used for extraction and description of key points or key point pairs also play a significant role in matching. Compared to features based on color, geometric features are more invariant to viewpoint changes. The simplest geometric

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

features could be just 3D coordinates of points, which can be used by the iterative closest point (ICP) algorithm [2] to register two similar 3D point clouds. However, the ICP algorithm requires a good initial pose estimation, which is difficult or nearly impossible from images of real-world scenes with multiple occluded objects shot in random viewpoints. Surface normals and curvature estimates [14, 13], which can be computed fast and easily, are also commonly used geometric features. [13] proposed pose-invariant key points by combining geometric relations between the nearest k neighbors of these keypoints based on estimated surface normals.

However, individual geometric primitives, such as normals, curvatures, lines, and planes, are not very discriminative; as a result, researchers developed various oriented point pair features [7, 4, 3]. Pair features are more discriminative than individual primitives, which provide more informative characteristics. Based on key point pairs, an approach [4] combined both oriented surface points and boundary primitives, such as boundary points with directions and boundary line segments, to enhance the accuracy and efficiency of the algorithms for detection and pose estimation. The work was focused on parts for assembly with many geometric details.

After results of correspondence matching are obtained based on key points, the locations of known objects can be predicted, and their poses can be estimated. Existing research often predicts an object pose by analyzing the pose distribution through either adopting a Hough voting scheme [1, 7, 3] or clustering in the parameter space. However, in a cluttered scene where objects are occluded and there are multiple instances of the same object, some known object instances may only have small visible regions and not enough votes to distinguish themselves from noises; this is because there are few valid matches of key points or key point pairs, and incorrect matches can result from matching between the whole testing scene and trained known 3D object models.

As for calculating a transformation matrix based on matched key point pairs, a comprehensive survey [8] provides detailed comparision results for four major algorithms: SVD, OM, UQ and DQ. The RANSAC [10, 13] algorithm can also be utilized here to make the calculation more robust to noise. The ICP [2] algorithm can be adopted appropriately later to refine the results of localization and pose estimation after coarse estimation results are obtained from the approaches described above.

III. Approach

In contrast to existing approaches, our approach in this paper is characterized by the following:

- Instead of directly matching the current scene to trained object models, our approach matches 3D surface segments obtained by segmentation of the current scene to the surface segments of known objects in the object database, which provides the relations between an object, its segmented smooth surfaces, and the corresponding point clouds. Key points are extracted, described, and matched only based on the surface segments more robust to viewpoint changes.
- We use *both* geometric and visual features to increase the discriminative power in key point matching. In addition to color, we use a more powerful visual signature ASIFT [16], which is improved upon SIFT and especially designed for affine transformations.
- We integrate the information from surface matching of all surfaces in the current scene to detect the target object, and obtain the pose estimate of the entire object instance by projecting the object model for each detecte, even if the object is heavily occluded and there are multiple instances of the same object in the scene.

In this section, we describe our surface-based approach for general 3D object detection and pose estimation in details.

A. Surface-based Object Representation and Detection

We start from a sufficient number of RGB-D images of different viewpoints of an object (captured by Microsoft Kinect) in order to establish an appearance-based representation of the object. First, images are segmented based on smooth surfaces, where a smooth surface is defined by the continuity of depth and surface normal values, and its boundary is characterized by depth and surface normal discontinuity [15], as shown in Fig. 1. Each object is characterized in terms of surface segments from multiple views and the visual signatures of the surface segments. Since our object models are built upon the original RGB-D images (without calibration), which are subject to noise, object detection based on our models is inherently more robust to noise. We use both HSV color histogram and ASIFT key point histogram [16] as the visual signatures because of their robustness to viewpoint and illumination changes.

As an example, Fig. 2 shows a common cereal box and its smooth 3D geometric surfaces in terms of the corresponding image segments from different views.

Note that in our object representation, only a coarse segmentation of geometric surfaces is required, which makes it flexible to model daily objects with irregular shapes, rather than cuboid or cylinder, and to tolerate small bulges on a surface.

To detect objects in a 3D test image, smooth geometric surfaces in the test image are first segmented, and their corresponding visual signatures are described. By applying the k-nearest neighbor algorithm [5], surface segments in the test image are identified and labeled by the objects they belong to respectively even if they are partially occluded.

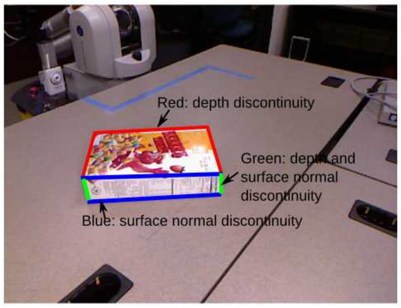

Red: depth discontinuity
Green: depth and surface normal discontinuity
Blue: surface normal discontinuity

Figure 1: Depth and surface normal discontinuity at surface boundaries

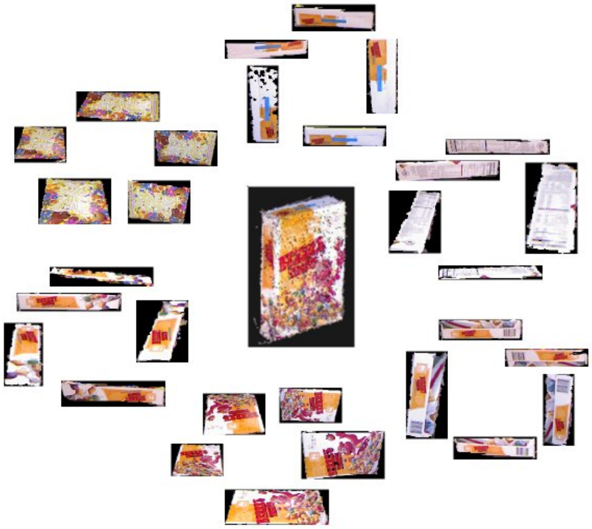


Figure 2: A common cereal box and images of its smooth 3D surfaces from different views

B. Surface-based Object Model building and Pose Estimation

In the representation of an object, after surface segments are obtained from all images of different views, key point matching is then conducted to establish correspondence among surface segments of the same 3D smooth object surface in different images. A coarse 3D model of the entire object is next established based on the matching results, and an object frame (i.e., coordinate system) is specified[1]. This is done by applying SVD [8] and RANSAC algorithms. Fig. 3 shows some reconstruction results.

Once object labels are provided to the surface segments in a test image, as described in Section III.A, the pose of an object can be estimated in terms of a homogeneous transformation matrix with respect to the camera coordinate system. For each surface segment with the detected object label in the test image, our algorithm searches for the most similar image of the surface segment with the same object label in the object database. It then matches the key points between the surface segments in the two images to estimate the pose of the corresponding object in the test scene. We further use the ICP algorithm to fit the predicted object model under the estimated pose to the corresponding actual point cloud in the test image to achieve a more accurate pose estimation.

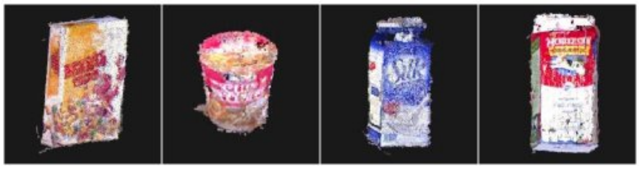


Figure 3: Examples of reconstructed objects, from left to right: cereal box, cup noodle, milk box 1 and milk box 2

To evaluate how well the considered object model fits a surface segment s, we introduce a *fitness* measure as a distance function between the point cloud of the projected object model and the point cloud of the surface segment. For each point q_i in the point cloud of the surface segment ($i = 1, ..., N$), we can find a matched nearest point $p(q_i)$ in the point cloud of the projected object model, and the fitness is computed as follows:

$$fitness(s) = \frac{1}{N}\sum_{i=1}^{N} ||q_i - p(q_i)||. \tag{1}$$

If the fitness value is greater than a threshold value, the surface segment is ruled out as unrelated to the considered object to reduce incorrect object detection.

Multiple surface segments of the same object can provide different object pose estimates. If these estimates are quite similar, i.e., consistent, we can simply choose one of them as the object pose. However, with small segments, sometimes the pose estimates are quite different and inconsistent. In such a case, we use the fitness measure to choose the correct object pose estimate as the one with better fitness values from all segments. Fig. 4 shows an example, where two surface-based pose estimates for the blue milk box were inconsistent, and the best-fit predicted object model was chosen. Note that the viewpoint used to display the reconstructed scene is different from the viewpoint of the original test image, as evident from the red milk box – two sides of the red milk box not visible in the original test image were shown in the reconstructed images.

C. Discussion

It is important to emphasize the major difference between our method for pose estimation and existing work: we match key points between each pair of corresponding surface segments of two images respectively rather than between the two entire images directly. By providing corresponding

[1] Note that every surface segment has its own frame w.r.t. the object frame.

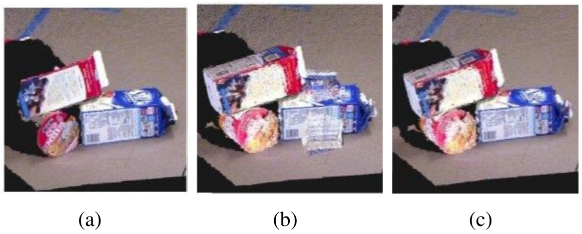


Figure 4: An example of handling inconsistent pose estimates: (a) original scene captured by a RGB-D camera, (b) intersecting projections of the model for the blue milk box at two inconsistent, initial object pose estimates, (c) object models at the best-fit pose estimates

surface segments as inputs to the matching algorithm, our method provides more accurate results of matching and subsequent object pose estimation.

First, for a partially occluded object, by focusing on its visible surfaces, we can provide a substantially greater number of valid matches of key points as input to the SVD algorithm to estimate the object. It is thus less likely to result in the singular problem for SVD (caused by key points aligned along a line). Fig. 5 shows an example: (a) shows the two images used to model the cereal box, (b) shows the key points matched based on the two whole images and the estimated object model, which suffers the singular problem, and (c) shows the key points matched based on the side surface detected in both original images and the estimated object model. Note that since only three surfaces of the cereal box can be seen in the two images, the result of estimation consists of just the three surfaces. Clearly, through focusing on surfaces, our algorithm obtained more correctly matched key points pairs and subsequently a much better estimation result.

Second, by extracting key points on the target object surfaces instead of an entire image, our method avoids the extraction of invalid key points, such as those on the boundaries between the target object and other objects or on the background in the image of a cluttered scene. For objects of regular and symmetric shapes with fewer discrimitive visual features on the boundaries, matching errors caused by invalid key points are very likely if an entire image is used for matching. Even if some sliding window is used to localize the target object, with objects occlude one another in a complex scene, the matching errors can still occur. Our approach effectively avoids such errors by focusing on each surface segment itself, so that both the extraction and description of key points are not affected by anything outside the surface segments.

Fig. 6 shows an example: (a) shows two images used to model the yogurt cup, (b) shows the key points matched based on the two whole images and the estimated object model displayed in both front and top views, and (c) shows the key points matched based on the top surface detected in both original images and the estimated object model displayed in both front and top views. From Fig. 6(b), we can see that the reconstructed model of the yogurt cup is quite distorted (from the front view image) and the blue spindle shape on the top surface is also clearly distorted (from the top view image). Whereas, the results in Fig. 6(c) are more accurate. By focusing on surfaces, our approach avoided a great number of incorrectly matched key point pairs and subsequently achieved a much better estimation result.

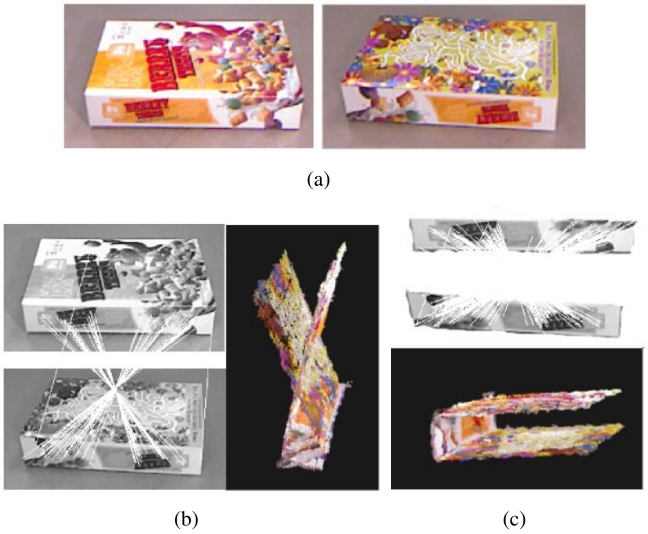
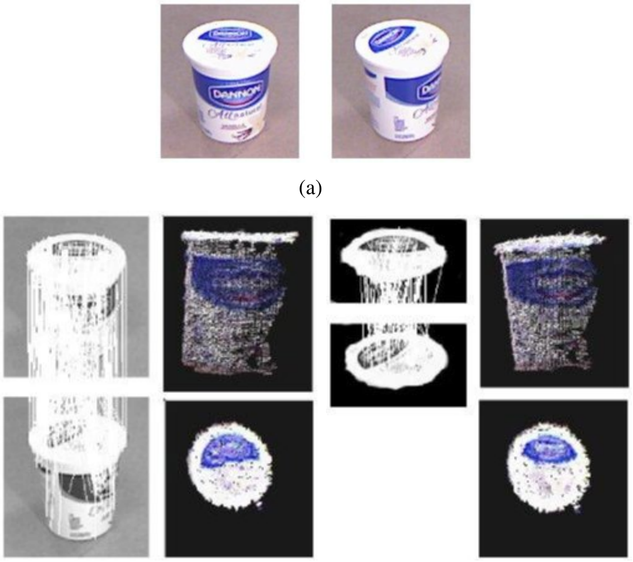


Figure 5: Comparison of key points matching and pose estimation results: (a) the original images for matching, (b) the matched key points and the result of estimated object model based on matching the whole images, and (c) the matched key points and the result of the estimated object model based on matching the common side surface visible in both images

D. Detection of Multiple Instances of the Same Object

One important advantage of our approach is that it can directly detect multiple instances of the same object conveniently even if the object instances are occluded in different ways. Specifically, for two surface segments S_1 and S_2 that detect the same object with consistent pose estimates, then they belong to the same object instance; otherwise, two instances of the same object are detected. See Fig. 10 for some examples of detection and pose estimation results of multiple, occluded instances of a cereal box and a blue milk box.

In contrast, most existing approaches use the Hough voting scheme to determine object instances based on votes, which requires setting some threshold value and can be subject to noise, especially for heavily occluded objects.



(a)

(b)

(c)

Figure 6: Comparison of key points matching and pose estimation results: (a) the original images for matching, (b) the matched key points based on the whole images and the result of estimated object model displayed in front and top views, (c) the matched key points based on matching the common top side surface visible in both images and the result of the estimated object model displayed in front and top views

IV. Experiments

We have built an object database of four objects, including a cereal box, a cup noodle, and two different milk boxes, each from 24 images of different views. The four objects are shown in Fig. 7.

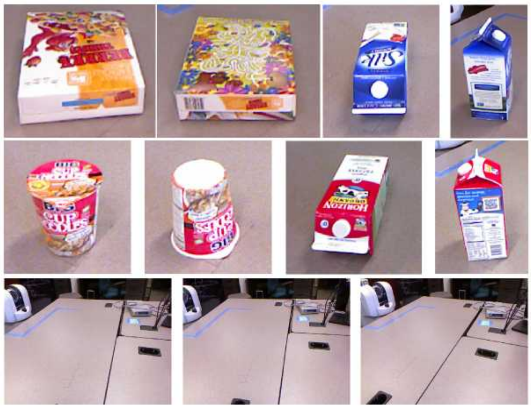


Figure 7: Examples of images in the dataset: top two rows show some images of the four objects used for training; bottom row shows some background images for training

To verify the performance of our approach efficiently, we have also created 15 different test images of cluttered environments, which include different objects or multiple instances of the same object. Each image includes 3–6 mutually occluded objects or object instances, as shown in the first column of Fig. 10.

Fig. 10 shows the results of our approach for object detection and pose estimation for the example test images. The 1st column shows four test images captured by a RGB-D camera. The 2nd–4th columns display the corresponding object detection and pose estimation results in four different views: left, back, right, and top views. By localizing all the known objects in the current scene and estimating their poses correctly, our algorithm reconstructed the whole scene based on a test image from a single viewpoint, as shown in the images of the reconstruction results from multiple views.

As described above, we use the k-nearest neighbor algorithm to recognize all the detected surface segments in the testing scene and then estimate their poses. A surface segment is labeled by the most voted label among its most similar k surface segments in the training dataset. Euclidean distance is used here as our distance metric in the HSV color and the ASIFT feature spaces.

Selecting a proper value for k is essential for the performance of the recognition of the object surface segments. Fig. 8 and Fig. 9 show the precision-k curves and the recall-k curves for all the objects in the dataset. We can see that for the value of k in the range of 5–10, both precision and recall rates are quite high for all known objects in the dataset. In our experiments, we set $k = 10$.

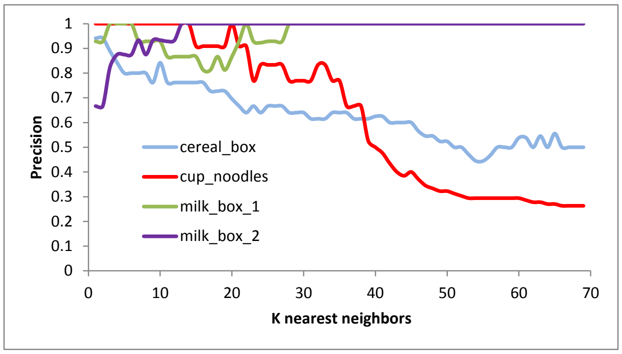


Figure 8: The precision-k curves for all the objects in the dataset

On the other hand, for very large k values, labels with most samples in the training dataset tend to dominate the prediction of the labels for test samples. Since we had more background surface segments than object surface segments for each object used for training, with very large k values, the recall rates for all the objects in the dataset eventually approach zero, because segments of background tend to donimate the results of labeling. For k values set in the middle range of the charts, objects with more surface segments tend to affect the labeling of surfaces with similar features. For

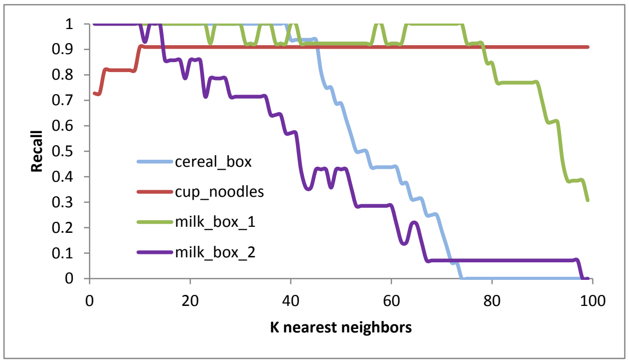


Figure 9: The recall-k curves for all the objects in the dataset

	cereal box	cup noodles	milk box 1	milk box 2
#Instances	16	11	12	14
Correct PE	14	9	9	13
Wrong PE	1	1	0	0
Missing	1	1	3	1
Fitness(mm)	5	2.9	6.1	6.1

Table I. Detection and pose estimation (PE) results with k=10

example, our experiments show that surfaces of milk box 2 tend to be misclassified as the cup noodle due to similar color features for $k > 10$.

Table I shows the detection and pose estimation results for all the objects in our dataset with $k = 10$. Most of the object instances are detected with pose estimation correctly. Note that the fitness shown under each object is the average of fitness values associated with all tested instances of the same object, where for each object instance, the fitness value is the average of fitness values associated with all the surface segments that correctly detect the object. The accuracies of each average fitness is quite consistent with the accuracy of the raw data captured by Microsoft Kinect, which is about ±3mm for objects within 1m from the camera. For the few missing and wrong detections, possible reasons include: wrong surface segments were detected based on visual features, or the detected surface segments were too small to provide sufficient information for generating reasonable poses.

We have developed a real-time online program in a single thread. It takes about 1 – 2 minutes to detect and estimate the poses of all the known object instances in a test image. The program spends most of the time in computing and matching ASIFT key points. However, many procedures in our program, such as normal estimation and key points matching, can be readily implemented in parallel. As a result, our approach could run efficiently and effectively on a suitable GPU in real time.

V. Conclusions

In this paper, we have introduced a new, appearance-based approach for general 3D object detection and pose estimation based on segmented 3D surfaces and their features, taking full advantage of RGB-D information. Our approach can detect and estimate the poses of occluded known objects, including occluded multiple instances of the same object, effectively in cluttered environments. With the detected objects and their poses w.r.t the current RGB-D camera frame, and with the transformation from the camera frame to the robot frame known, the results can be used directly for robotic manipulation of target objects, and the reconstructed 3D scene can also be used directly for motion planning to avoid obstacles.

As the next step, we plan to introduce more objects with more complicated shapes and surfaces to test and further improve and extend our approach. We will also integrate our work with robotic manipulation and motion planning approaches to enable autonomous object manipulation and obstacle avoidance based on sensing in real-time.

Acknowledgment

This work is supported by the US National Science Foundation grant IIS-0904093 and grant IIS-1258335.

References

[1] D. H. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, pp. 111–122, 1981.

[2] P. Besl and H. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, pp. 239–256, 1992.

[3] C. Choi and H. I. Christensen, “3d pose estimation of daily objects using an rgb-d camera,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, 2012, pp. 3342–3349.

[4] C. Choi, Y. Taguchi, O. Tuzel, M. Liu, and S. Ramalingam, “Voting-based pose estimation for robotic assembly using a 3d sensor,” in *IEEE International Conference on Robotics and Automation, (ICRA)*, 2012, pp. 1724–1731.

[5] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, 1967.

[6] N. Dalal and B. Triggs, “Histograms of orientaed gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[7] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3d object recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 998–1005.

[8] D. W. Eggert, A. Lorusso, and R. B. Fisher, “Estimating 3-d rigid body transformations: a comparison of four major algorithms,” *Machine Vision and Application*, vol. 9, pp. 272–290, 1997.

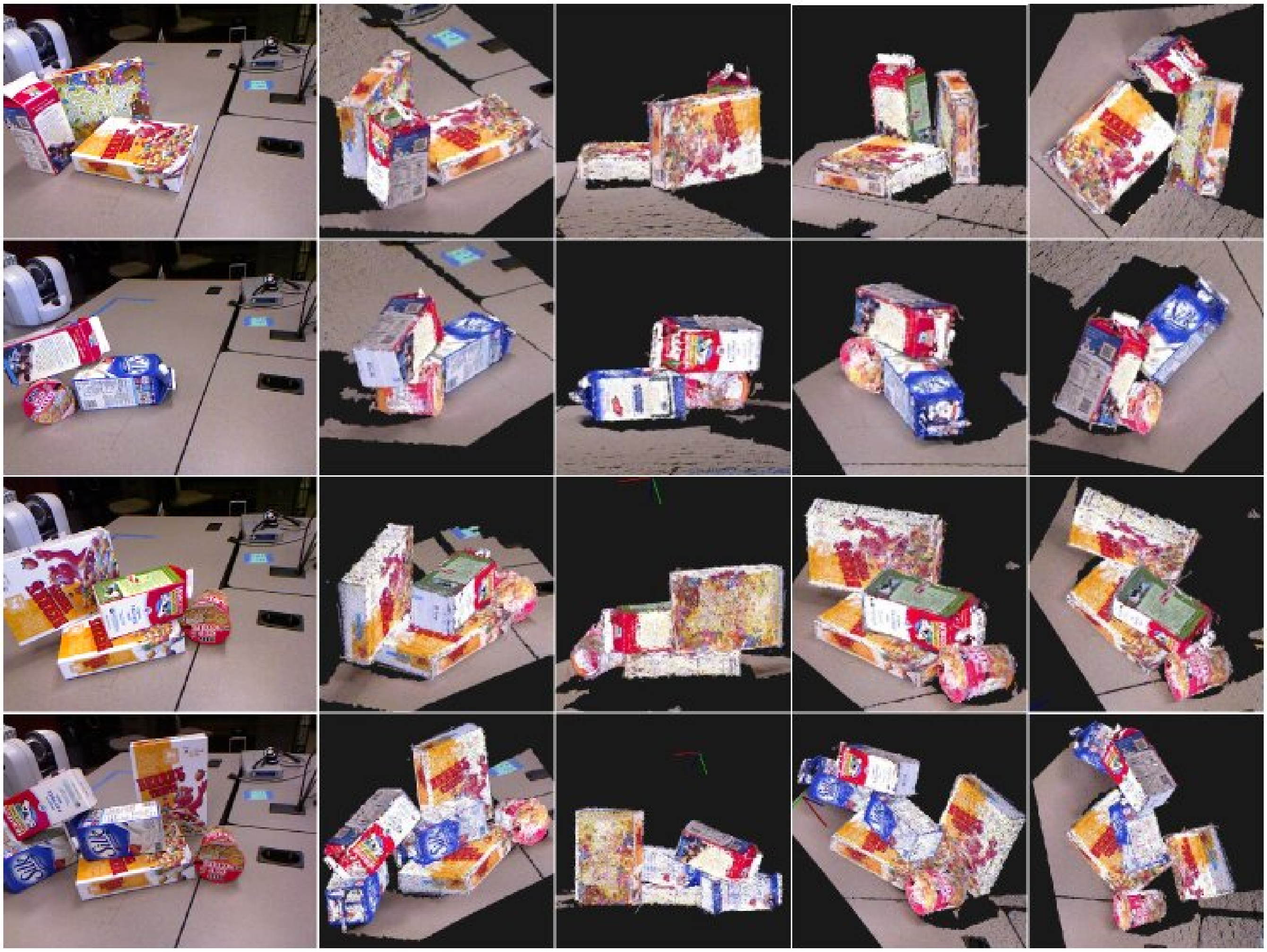


Figure 10: Examples of object detection and pose estimation results: each row starting with an original test image followed by four images from left, back, right, and top views that display the corresponding reconstructed scene

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, pp. 1627–1645, 2010.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[11] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on Robotics and Automation, (ICRA)*, 2011, pp. 1817–1824.

[12] K. Lai, L. Bo, X. Rend, and D. Fox, "Detection-based object labeling in 3d scenes," in *IEEE International Conference on Robotics and Automation, (ICRA)*, 2012.

[13] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE International Conference on Robotics and Automation, (ICRA)*, 2009, pp. 3212–3217.

[14] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, 2008, pp. 3384–3391.

[15] Z. Teng and J. Xiao, "3d object detection based on geometrical segmentation," in *2013 International Conference on Computer and Robot Vision, (CRV)*, 2013, pp. 67–74.

[16] G. Yu and J. M. Morel, "Asift: an algorithm for fully affine invariant comparison," *Image Processing On Line*, 2011.