2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration

A Combined Texture-Shape Global 3D Feature Descriptor for Object Recognition and Grasping

Zhun Fan, Zhongxing Li, Wenji Li, Yugen You, Wenzhao Chen, Chong Li
Key Lab of Digital Signal and Image Processing of Guangdong Province
Shantou University
Guangdong, Shantou 515063
E-mail: zfan@stu.edu.cn

***Abstract*—This paper presents a global 3D feature descriptor for object recognition and grasping. The proposed descriptor stems from the clustered viewpoint feature histogram (CVFH) feature descriptor. Since the CVFH feature descriptor relies on shape information only, it obtains a poor performance when recognizing the objects with similar shapes. In order to improve the robustness and accuracy of object recognition, we extend CVFH feature descriptor with color information. Then this new global 3D feature descriptor is tested for multiple classes of 3D object classification by using support vector machines (SVM), and it is evaluated with a public dataset and real scenes respectively. The experimental results show that the proposed descriptor outperforms the CVFH feature descriptor in terms of recognition rate. Finally we utilize the proposed descriptor on our grasping system to recognize and grasp the objects, showing that the grasping system can accomplish the tasks well.**

***Keywords*-object recognition; 3D feature descriptor; color information; SVM; grasping;**

I. Introduction and related work

Object recognition plays a key role for robot grasping, because the automation of robot grasping in a warehouse not only needs to identify what the object is, but also requires where the object is. Object recognition can make a robot easier get these kinds of information. Besides, object recognition is widely used in many areas, such as objects sorting, objects manipulation, robot localization and navigation. However, to design a robust and effective object recognition system is still a relatively challenging and difficult task.

A common way to implement object recognition is to extract the features from the objects in the 2D image plane and match these features with corresponding features of previously stored object models, such as SIFT[1] image features, SURF[2] image features and ORB[3] image features. These methods can obtain a good performance on high textured objects, but relatively inefficiency on low textured ones.

With the recent advent of low-cost, real-time stereo sensors, such as the RGB-D sensor Microsoft Kinect[4], 3D feature descriptors are widely used in object recognition. There are two types of 3D feature descriptors, including local 3D feature descriptor and global 3D feature descriptor.

For the former, the local 3D feature descriptor relies on the local repeatable key points which are extracted from the surface of the 3D object, and it is computed for individual key point, each key point only has one local 3D feature descriptor. These local 3D feature descriptors are used to match with corresponding local 3D feature descriptors of previously saved object models for object recognition. There are some types of local 3D feature descriptors, which are used for object recognition, such as the point feature histogram (PFH)[5] descriptor, the fast point feature histograms (FPFH)[6] descriptor, the radius-based surface descriptor (RSD)[7] and the signature of histograms of orientations (SHOT)[8] descriptor. These methods can perform well on the objects with rich geometrical information and obtain the exact position of the object. Besides, they can be directly applied on the cluttered scenes and don't need segment the objects from the scenes. However, because of the complexity of the local 3D feature descriptors in the features matching stage, the local 3D feature descriptors need a lot of computational resource for implementing object recognition.

According the later one, the global 3D feature descriptor is designed for describing the whole object, which means that each object only has one single global 3D feature descriptor. These global 3D feature descriptors can be used for object recognition and classification by the means of features matching. Obviously, the global 3D feature descriptor can't be directly applied on the cluttered scenes. Thus we have to segment the objects from the scenes before using the global 3D feature descriptor for object recognition. There are some types of global 3D feature descriptors, which are used for object recognition, such as the ensemble of shape functions (ESF)[9] descriptor, the global radius-based surface descriptor (GRSD)[10], the viewpoint feature histogram (VFH)[11] descriptor and the clustered viewpoint feature histogram (CVFH)[12] descriptor. These global 3D feature descriptors reduce computational burden, make features matching faster and require less memory resources to store the object models with respect to the local 3D feature descriptor, since its complexity is less than the local ones.

978-1-5386-2434-0/17 $31.00 © 2017 IEEE
DOI 10.1109/ICIICII.2017.52

These aforementioned 3D feature descriptors are only based on the shape information of the objects, which obtain a poor performance when recognizing the objects with similar shapes but different textures. There are also some types of 3D feature descriptors which combine shape information with texture information, such as Color-SHOT (CSHOT)[13] descriptor. It is a local 3D feature descriptor which based on the shape information and the color information. Thus it can utilize both shape and textures information, and improve the accuracy of object recognition by feature matching. But it still needs extra memory resources to store the object models and computational resources for feature matching.

In this paper, the proposed method can improve the robustness and accuracy of object recognition. The contributions of this paper are: (1) We propose a global 3D feature descriptor based on both shape and color information, and it is built by extending the clustered viewpoint feature histogram (CVFH)[12] descriptor with color information. (2) We train the multi-class support vector machine (SVM)[14] classifier off-line, then we use this trained classifier for object recognition instead of feature matching. (3) Our proposed feature descriptor is evaluated with both public dataset and real scenes, and it is utilized on our proposed grasping system to recognize and grasp the objects.

The structure of this paper is organized as follows. Our proposed methods is described in Section II. The results and analysis of experiments are presented in Section III. Finally, conclusions and future work are made in Section IV.

II. Methodology

We begin this section by introducing the method of segmentation first, which is use for segmenting the target objects from scenes. Then we introduce our proposed feature descriptor in detail. At the end of this section, we present our grasping system.

A. Segmentation

Before utilizing the proposed 3D global descriptor to recognize the object, we need to segment the target objects from scenes. In this work, we use the Microsoft Kinect to capture the 3D point cloud image, Figure 1 shows the process of our method for segmentation.

As showing from Figure 1, the first work for segmentation is capturing the 3D point cloud image of scene by Microsoft Kinect, and then we filter the source scene point cloud to obtain the region of the target objects. After that, we use the random sample consensus (RANSAN)[14] algorithm to find large planar objects and subtract these planar objects from scene. Finally we utilize the Euclidean Cluster Algorithm (based on Euclidean Distance)[15] to extract the target objects from the scene.

RANSAN[14]: Firstly, this algorithm chooses a set of points from point cloud image randomly, and estimates the parameters of a mathematical model for the chosen set

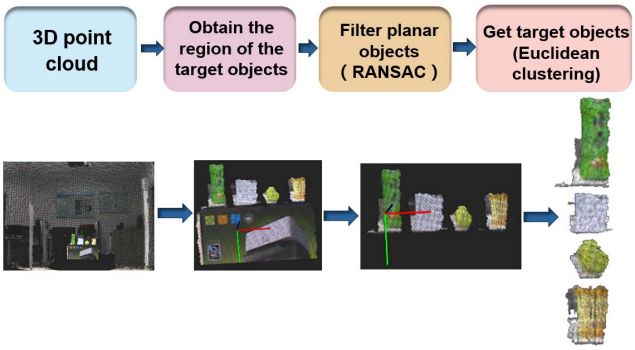


Figure 1. The process of our method for segmentation

of points by the method of Minimum Variance Estimation Algorithm. Then the RANSAN algorithm calculates the error value of each point in point cloud image based on the parameters of the model. If the value isn't greater than a predetermined threshold value, this point belong to the planar object which denoted as inliers, otherwise this point does not belong to the planar object which denoted as outliers. After iteration, the set of points which has the largest number of inliers is the planar object we need to find.

Euclidean Cluster Algorithm[15]: Firstly, the algorithm randomly selects a point p_0 from point cloud image, this point p_0 is belong to the point set Q_1 ($Q_1 = \{p_0\}$), and the algorithm calculates the distance between point p_0 and its neighbor point q_i. If the value of distance is lower than the predetermined threshold value, adding this point q_i to the point set Q_1 ($Q_1 = \{p_0, q_i\}$). Then this algorithm calculates the distance of other point which belong to the point set Q_1 from its neighbor point. Repeating previous steps, until the algorithm can't find a legal point add to the point set Q_1, the point set Q_1 is the object we want to extract from the scene. Similarly, we use the same method to extract other objects from the scene, until all the objects are extracted from the scene, the segmentation is completed.

B. The design of 3D global feature descriptor

Our proposed feature descriptor is a global 3D feature descriptor, which based on both shape and color information, so that we can make use of both shape and texture information of object. This proposed feature descriptor stems from clustered viewpoint feature histogram (CVFH)[12] descriptor, and here we denote the proposed feature descriptor as Color-CVFH. Next we give a description of the CVFH descriptor.

CVFH[12]: CVFH is an extension to the viewpoint feature histogram (VFH)[11] descriptor. The VFH descriptor is built by a histogram of the four different angular distributions based on the surface normal of the object, and it contains two components, one is viewpoint direction component and another is extended FPFH[6] component. In this work, we

define p_c and n_c ($||n_c|| = 1$) as the centroid of the whole surface points and the normal of the centroid, and use (u_i, v_i, w_i) defines a coordinate frame for each point p_i of the whole surface points[11].

$$\begin{cases} u_i = n_c \\ v_i = \dfrac{p_i - p_c}{||p_i - p_c||} \times u_i \\ w_i = u_i \times v_i \end{cases} \tag{1}$$

The n_i is the norm of each point p_i, then the four different types of angles can be calculated as following[11].

$$\begin{cases} \cos(\alpha_i) = v_i \cdot n_i \\ \cos(\beta_i) = n_i \cdot \dfrac{p_c}{||p_c||} \\ \cos(\Phi_i) = u_i \cdot \dfrac{p_i - p_c}{||p_i - p_c||} \\ \theta_i = atan2(w_i \cdot n_i, u_i \cdot n_i) \end{cases} \tag{2}$$

The $\cos(\alpha_i)$, $\cos(\Phi_i)$ and θ_i are the three types of angles which are used to build the extended FPFH[6] component, and $\cos(\beta_i)$ is used to build the viewpoint direction component. In order to improve the robust to deal with occlusion object, the CVFH is obtained by calculating the VFH histogram for each region of the object surface instead of a single VFH histogram for the whole surface, and this region is stable, smooth region which is extracted from the whole surface by using region-growing segmentation. Thus, the CVFH descriptor has the extended FPFH component and viewpoint direction component. Besides, the CVFH descriptor has an additional component which is shape distribution component (SDC)[12], it describes the distribution of the points around the region's centroid, and SDC can be defined as following[12].

$$SDC = \frac{(p_c - p_i)^2}{max((p_c - p_i)^2)} \quad \text{where } i = 1, 2, ..., N \tag{3}$$

In Equation 3, N presents the number of the points for the whole point cloud. The SDC component has 45 histogram bins, each angular distribution of the extended FPFH[6] component also has 45 histogram bins, and the viewpoint direction component has 128 histogram bins. Thus a CVFH histogram has 308 bins in total.

Color-CVFH: In this work, we design a global feature descriptor named as Color-CVFH, and it contains two parts, one is a global color histogram, and another is a CVFH histogram. The points of the point cloud which are captured by the RGB-D sensor Microsoft Kinect not only contain the position information, but also have the color information, and the color information is described in RGB space. By using the aforementioned method of segmentation, we can get the individual object from the scene, and we can also obtain the color information of the individual object in RGB space. As the color features in HSV space is more effective than the color features in RGB space for object recognition, we convert the color information from RGB space to HSV space. Because the hue value is a very important component in HSV space for object recognition, we set 90 bins for the hue value histograms, and the size of histogram bins for saturation value and value noise are both set as 51. Thus the global color histogram we design has 192 histogram bins in total, and each object only has one single global color histogram. Now we denote the global color histogram of the object O_i as C_i, and the CVFH histogram of object O_i is denoted by V_i, the Color-CVFH descriptor of the object O_i is denoted by F_i. The Color-CVFH descriptor F_i can be defined as following.

$$F_i = C_i \cup V_i \tag{4}$$

From the definition of Color-CVFH descriptor F_i, we can see the first part of Color-CVFH descriptor is the global color histogram C_i, and then is followed by the CVFH histogram V_i. Thus the Color-CVFH descriptor has 500 histogram bins in total. The structure of Color-CVFH descriptor is showed as Figure 2. The Color-CVFH descriptor utilize both shape and color information, and it can be easily used to train the multi-class SVM classifier for object recognition.

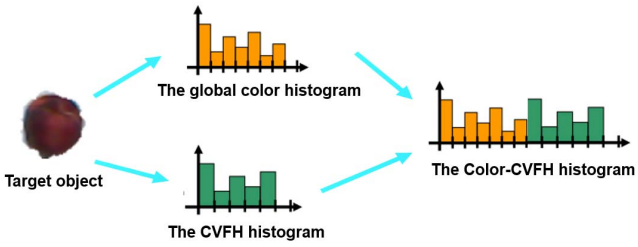


Figure 2. The structure of Color-CVFH descriptor

C. The grasping system

In this work, we need to utilize the proposed descriptor Color-CVFH on our grasping system to sort, pick and place objects. The overview of our grasping system is showed as Figure 3.

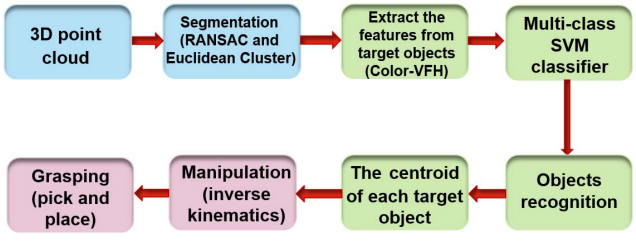


Figure 3. The overview of grasping system

From the Figure 3 we can see the first step of grasping system is that the system captures the 3D point cloud by the RGB-D sensor Microsoft Kinect. Then it extracts the target objects from scene by using the method of RANSAN[14] and Euclidean Cluster[15] algorithms. When the grasping system obtaining the target objects, this grasping system calculates the Color-CVFH descriptor of these objects, and uses the multi-class SVM classifier based on the Color-CVFH descriptor which we have trained before to recognize these target objects. After that, the grasping system can obtain the class and location of each target object. In this work, we use the centroid of each target object as the grasping point for the robotic manipulator to grasp. In order to control the robotic manipulator to grasp the target objects, the grasping system needs to calculate the solutions of inverse kinematics for the robotic manipulator. Here this grasping system leverages the geometric method to calculate the solutions of inverse kinematics, because the geometric method not only can save the computational resources, but also can obtain high accuracy solutions of inverse kinematics. After the grasping system obtaining the solutions of inverse kinematics, the end effector of robotic manipulator can approach to each target object, then grasp the target object according to the grasping point which is calculated by the grasping system before. Finally, the robotic manipulator picks up the target object, and places it to its specific location according the class of the target object. After that, the grasping task has been completed.

III. Experiments and Results

In this section, the Color-CVFH descriptor is evaluated with a public dataset, which is Washington University RGB-D dataset[16]. Then the Color-CVFH descriptor is evaluated with real scenes (150 different scenes in total). In these experiments, we use four metrics which are recall, precision, accuracy and F1-score to evaluate the performance of the Color-CVFH descriptor and the CVFH descriptor. The definition of these four metrics are showed as following.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{8}$$

In these four equations, TP presents the number of true positives, FP presents the number of false positives, TN presents the number of true negatives and FN presents the number of false negatives.

At the end of this section, in order to evaluate the robustness of the Color-CVFH descriptor, we utilize the Color-CVFH descriptor on our grasping system to sort, pick and place objects.

A. Evaluated with a public dataset

The public dataset which is used for evaluating the performance of the Color-CVFH descriptor is Washington University RGB-D dataset[16], which contains 300 objects in 51 categories, and each instance is captured by 3 different camera poses. Here we show some samples of the public dataset in Figure 4.

Figure 4. The samples of the public dataset

In this experiment, we select six categories of the objects from this public dataset, which are apple, tomato, lime, orange, cereal box and food box. Apple and tomato have similar shapes and colors, lime and orange have similar shapes but different colors, cereal box and food box are both rectangular shaped objects. Each category of the objects is selected 840 point clouds from the public dataset. These six categories of objects are showed as Figure 5.

Figure 5. One view of one object from each of the six categories used in the experiment. Left to right: apple, tomato, lime, orange, cereal box and food box.

This experiment is used for evaluating the performance of the Color-CVFH descriptor and the CVFH descriptor by using multi-class SVM classifier. We select 630 point clouds of each category for training the multi-class SVM classifier, and the rest of the selected point clouds are used for testing. This experiment is used for evaluating the performance of the Color-CVFH descriptor and the CVFH descriptor for object recognition, the results of the experiment are showed as Table I.

From the Table I, we can see the classifier which is trained by the CVFH descriptor has a poor performance compared to the Color-CVFH descriptor. This classifier are much more likely to consider the apple and orange as the

Table I
THE RESULTS OF THE EVALUATION FOR CVFH AND COLOR-CVFH BY USING THE PUBLIC DATASET

	CVFH				Color-CVFH			
	Recall	Precision	Accuracy	F1-score	Recall	Precision	Accuracy	F1-score
Apple vs. all	71.43%	85.71%	93.25%	0.78	**100%**	**100%**	**100%**	**1**
Tomato vs. all	85.24%	50%	83.33%	0.63	**100%**	**100%**	**100%**	**1**
Lime vs. all	14.76%	50%	83.33%	0.23	**100%**	**100%**	**100%**	**1**
Orange vs. all	88.10%	75.51%	93.25%	0.81	**100%**	**100%**	**100%**	**1**
Cereal box vs. all	100%	100%	100%	1	100%	100%	100%	1
Food box vs. all	100%	100%	100%	1	100%	100%	100%	1

same object, and it also likely to recognize the lime and tomato as the same thing, because apple and orange have similar shapes, lime and tomato also have similar shapes. However cereal box and food box get good results in the metrics of recall, precision, accuracy and F1-score, because the CVFH descriptor is based on the shape information, and cereal box and food box have different shapes though they are both rectangular shaped objects.

In contrast, the classifier which is trained by the Color-CVFH descriptor has a remarkable performance in this experiment, because the Color-CVFH descriptor takes advantages of shape and color information, unlike the CVFH descriptor based on the shape information only. Thus this classifier can recognize every object in the testing set correctly, though apple and tomato have similar shapes and colors, this classifier also can distinguish them correctly.

B. Evaluated with real scenes

In this experiment, the Color-CVFH descriptor and CVFH descriptor is evaluated with real scenes. Here we choose objects including bottle0, cup, ball, bottle1, bottle2 and box for this experiment, which are showed in Figure 6, and their point clouds are showed in Figure 7. In this work, we sort these objects in four categories which are bottle0, cup, ball and others. The bottle1, bottle2 and box are belong to the category of others. Each category of the objects except the category of others has 100 different point clouds which are captured by RGB-D sensor Microsoft Kinect on different views. The category of others has 45 different point clouds in total. From the Figure 6 and Figure 7, we can see bottle0 and bottle1 have the same shape but different colors, the bottle0 and bottle2 have similar shapes and colors in some views, the cup and box also have similar shapes and colors in some views, while the ball is the only spherical shaped object here. Then we extract the Color-CVFH features and CVFH features from these 345 different point clouds, and we use these two types of feature descriptors to train the four classes SVM classifier respectively.

Finally we use the aforementioned method of segmentation to extract the target objects from the real scenes. In this experiment, we use 150 different real scenes to evaluate the performance of the Color-CVFH descriptor and CVFH descriptor, and here we show some samples of real scenes in Figure 8. Then we utilize these two trained SVM classifiers to recognize the target objects, and the results of experiment are showed in Table II. From the Table II, we can see the classifier based on the CVFH descriptor doesn't perform well in this experiment compared to the Color-CVFH descriptor. As the bottle0, bottle1 and bottle2 have similar shapes, this classifier recognizes some objects belonged to the bottle0 as bottle1 or bottle2 (the bottle1

Figure 6. One view of the objects used in the experiment. For the first row, left to right: bottle0, cup, ball. For the second row, left to right: bottle1, box, bottle2.

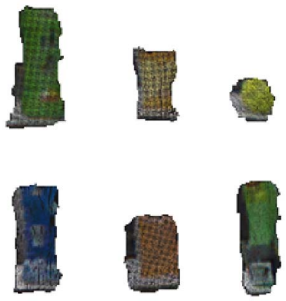


Figure 7. The point cloud of the objects used in the experiment (One view). For the first row, left to right: bottle0, cup, ball. For the second row, left to right: bottle1, box, bottle2.

(a) (b) (c)

Figure 8. Three samples of real scenes. (a): There are bottle0, box, cup, ball on the planar object. (b): There are ball, bottle0, cup, bottle1 on the planar object. (c): There are bottle2, ball, bottle0, cup on the planar object.

and bottle2 are belong to the category of others), and it also recognizes some objects belonged to the bottle1 or bottle2 as bottle0, thus the bottle0 obtains a poor precision (61.39%). Besides, this classifier recognizes some objects belonged to the cup as box (the box is belong to the category of others), and therefore the cup doesn't obtain a good precision (90%). While the ball has good performance in this experiment, because the ball has distinct shape in this dataset.

However, the classifier which is trained by the Color-CVFH descriptor has a remarkable performance in this experiment. This classifier can recognize every object in these real scenes, though the bottle0 and bottle2 have similar shapes and colors, this classifier can also distinguish them correctly.

All in all, the CVFH descriptor can perform well for recognizing the objects with different shapes, such as cereal box and food box in the first experiment. But it obtains poor performance for the objects with similar shapes, such as bottle0 and bottle1 in the second experiment. While our proposed descriptor Color-CVFH obtains remarkable recognition performance both in public dataset and real scenes, even the objects with similar shapes and colors, such as bottl0 and bottle2 in the second experiment. All these results of these two experiments, which show that our proposed descriptor Color-CVFH have high robustness and accuracy for object recognition.

C. Tested with grasping system

In this experiment, we use the four classes SVM classifier based Color-CVFH descriptor which trained in the second experiment, and utilize it to the aforementioned grasping system. Then we test it to recognize and grasp three different objects, which are showed in Figure 9.

From the figure 10, we can see our proposed grasping system not only can figure out the category of the target objects, but also can grasp the objects and place them to their specify location respectively. From these three experiments, we can see the Color-CVFH descriptor has a remarkable performance both in public dataset and real scenes, and it

Figure 9. The real scene of grasping task.

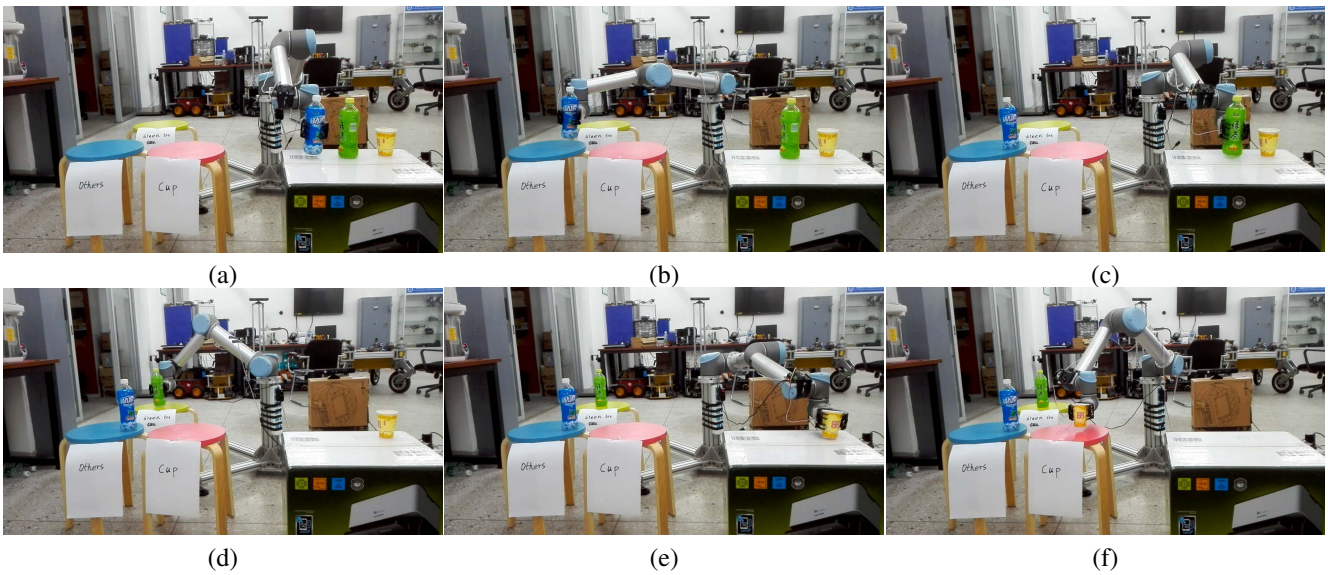
can also be utilized for object recognition and grasping in real scenes, and accomplish these tasks well, which means that the proposed descriptor can perform well for grasping applications.

IV. Conclusion and Future Work

This paper presents a global 3D feature descriptor which combine shape and color information for object recognition and grasping, and we propose a method of segmentation as the preprocessing before utilizing our proposed 3D feature descriptor. We also present a method that using multi-class SVM classifier to recognize target objects instead of features matching, which can save computational time and memory resources. Besides, we evaluate the performance of our proposed descriptor Color-CVFH by using both public dataset and real scenes, and the results of the experiments show that this proposed descriptor not only perform well for recognizing the objects with different shapes, but also make a remarkable performance for recognizing the objects with similiar shapes. Finally, we present our own grasping system, and the utilize the Color-CVFH descriptor in this grasping system for objects recognition and grasping, and the results show that our grasping system can accomplish these tasks well.

Table II
THE RESULTS OF THE EVALUATION FOR CVFH AND COLOR-CVFH BY TESTING IN REAL SCENES

	CVFH				Color-CVFH			
	Recall	Precision	Accuracy	F1-score	Recall	Precision	Accuracy	F1-score
Bottle0 vs. all	82.67%	61.39%	82.67%	0.70	**100%**	**100%**	**100%**	**1**
Cup vs. all	96.00%	90.00%	96.33%	0.93	**100%**	**100%**	**100%**	**1**
Ball vs. all	96.00%	100%	99.00%	0.98	**100%**	**100%**	**100%**	**1**



(a) (b) (c)

(d) (e) (f)

Figure 10. The result of experiment for grasping task. (a) to (f) show the process of grasping.

The datasets which are used for our experiments aim at recognizing and grasping target objects, thus our proposed descriptor can obtain remarkable recognition performance in these experiments. However, our proposed descriptor may not distinguish the objects with the same shape and color information correctly, and our current work only consider the object recognition without the pose estimation of object. Thus we will consider the pose estimation of object and combine our descriptor with more texture information in our future work, which make our descriptor more robustness and accuracy for object recognition and grasping.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant 61300159, 61473241 and 61332002, by the Project of Internation as well as HongkongMacao&Taiwan Science and Technology Cooperation Innovation Platform in Universities in Guangdong Province under grant 2015KGJH2014, by the Science and Technology Planning Project of Guangdong Province of China under grant 2013B011304002, by Educational Commission of Guangdong Province of China under grant 2015KGJHZ014.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571.

[4] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

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

[6] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3212–3217.

[7] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3d modelling of novel objects from a single view," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3700–3705.

[8] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European conference on computer vision*. Springer, 2010, pp. 356–369.

[9] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2987–2992.

[10] Z.-C. Marton, D. Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 365–370.

[11] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2155–2162.

[12] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 585–592.

[13] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3d feature matching," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 809–812.

[14] V. Vapnik, "The support vector method of function estimation," *Nonlinear modeling: Advanced black-box techniques*, vol. 55, p. 86, 1998.

[15] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

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