Intel Serv Robotics (2011) 4:159–166
DOI 10.1007/s11370-011-0088-5

ORIGINAL RESEARCH PAPER

Towards cognitive grasping: modeling of unknown objects and its corresponding grasp types

Hyoungnyoun Kim · Inkyu Han · Bum-Jae You · Ji-Hyung Park

Received: 26 April 2011 / Accepted: 14 May 2011 / Published online: 17 June 2011

Abstract This paper describes an intuitive approach for a *cognitive grasp* of a robot. The cognitive grasp means the chain of processes that make a robot to learn and execute a grasping method for unknown objects like a human. In the learning step, a robot looks around a target object to estimate the 3D shape and understands the grasp type for the object through a human demonstration. In the execution step, the robot correlates an unknown object to one of known grasp types by comparing the shape similarity of the target object based on previously learned models. For this cognitive grasp, we mainly deal with two functionalities such as reconstructing an unknown 3D object and classifying the object by grasp types. In the experiment, we evaluate the performance of object classification according to the grasp types for 20 objects via human demonstration.

Keywords Robotic grasp · 3D object modeling · Grasp types · Object classification · Human demonstration

1 Introduction

Grasping objects is a fundamental skill to interact a robot with human or environment [1,2]. Many researches have concerned with robotic grasp and tried to figure out the best grasp type according to the various environmental conditions. Especially, humanoid robots including mobile robots with dexterous hands need to learn and execute grasping task like a human for a natural interaction with objects. We define the human-like grasping approach as a robotic *cognitive grasp*.

A human determines a grasp type depending on the situation of a target object. Generally, when a human tries to grasp an object, he/she approximates the 3D shape of the object, and then attempts to grasp it as a similar method that he/she had experienced before. The grasp type, however, can be changed by the state of the object or the purpose of grasping even for a same object. For example, a cup has different grasp types by its states including an upright position, a prone position, and filled up with water. In addition, a grasp type of the cup can be determined by the intention such as throwing it out, giving it to someone, or washing it.

For a robot, it is hard to know identities of all the objects faced in real world. Most of the objects that a robot meets are supposed to unknown objects [3–10]. In addition, the grasp type is affected by the degree of freedom in a robot hand such as the number of fingers on the robot hand [7]. Therefore, in order to enable a robot to grasp an unknown object, we have to consider the relation between the external states of objects and the mobility of the robot. Since the ability of current service robots is much lower than that of a human in grasping task, it is challenging for a robot to grasp an object like human. Instead, if a robot observes a human's demonstration for grasping task, we may teach the robot the correlation between grasp types and the states of objects more naturally and intuitively [11–14].

In this paper, we propose an intuitive approach for a robotic cognitive grasp. Our proposed robotic cognitive grasp is composed of three steps as follows. First, a robot looks around the unknown target object, like a human, to recognize a 3D shape and its state. Second, the grasp type of the target object is learned from human demonstration. Third, when the robot is faced with an unknown object to grasp, it

H. Kim · I. Han · B.-J. You · J.-H. Park (✉)
Interaction and Robotics Research Center,
Korea Institute of Science and Technology,
39-1 Hawolgok-Dong, Seongbuk-Gu, Seoul, Korea
e-mail: jhpark@kist.re.kr

H. Kim · I. Han
Department of HCI and Robotics,
University of Science and Technology, Seoul, Korea

searches for the most similar object model that the robot has already learned and tries to grasp the unknown object.

We implement two major functions for the robotic cognitive grasp. One is the reconstruction of 3D shapes of various objects by the stereo vision. The other deals with classifying the target object into a known grasp type and determining an appropriate grasp type for the robot. Then we evaluate the recognition performance by matching shape information of the objects with its grasping types via Euclidean distance, multi-layer perceptron, and support vector machine.

2 Related work

The researches for a robotic grasp have considered several conditions that include the existence of predefined object models, the degree of freedom in a robot hand, the type of sensors for object recognition, and the learning method of the grasping skills. Especially, the requirement of predefined object models hinders a robot to be applicable. Many recent researches, therefore, have focused on the grasp planning without predefined 2D and 3D shape information of target objects.

Perrin et al. [3] proposed a statistical deformable contour model to determine plausible grasp axes of an unknown object. After modeling an object with the contour model, a robot found possible grasp axes and tried to grasp the object. Morales et al. [4] also tried to grasp an object by analyzing the 2D contour. They made a grasp planning for objects with a hole, synthetic images, and overlapped objects by expanding or squeezing a gripper of a robot. Taylor et al. [5] reconstructed unknown target objects by analyzing range data acquired from a 3D scanner. Miller et al. [6] classified various objects into four shape primitives such as spheres, cylinders, cones, and boxes. Saxena et al. [7] proposed an approach to detect grasping points of the object using the previously trained images instead of 3D models. They made a robot grasp unknown objects that were even overlapped in different environment such as dishwashers, office, and kitchen. Yamazaki et al. [8–10] recognized an unknown object and found a stable grasp pose from the automatically generated 3D model. In their research, a robot with a camera rotated around a target object and reconstructed the 3D shape. After that, the robot determined an appropriate type for its gripper depending on the 3D shape of the object.

Among the methods of grasp planning, the learning by human demonstration has several advantages. It is intuitive and natural because it enables a robot to learn through interaction with a human. Maeda et al. [11] let a robot learn the moving path of objects by demonstrating a manipulation of an object, which had a marker for tracking. Ekvall et al. [12,13] classified ten grasp types by observing the hand trajectory, hand rotations, and postures of a human hand via a data glove. Hueser et al. [14] taught a robot grasping skills by an instructor who demonstrates a grasping action with bare hands. They applied machine-learning methods to track poses of human hands.

In the above-mentioned researches about unknown objects, the most human-like cognitive approach is that of Yamazaki et al. The process of looking around an object and recognizing the object by vision system is similar to the operation of human's eyes. However, in order to connect with the process of grasping an unknown object, additional researches are needed. In this paper, we apply Yamazaki's approach, then a 3D shape of an unknown object with partial viewpoints is reconstructed after the robot locomotion. Moreover, we train a grasp type of the modeled object via the human demonstration so that we combine both grasping unknown objects and learning by a human demonstration for the cognitive grasp planning.

3 Cognitive grasp planning

The cognitive grasp, in this paper, means the chain of processes that make a robot to learn and execute a grasping method for unknown objects like a human. Figure 1 illustrates the learning and executing steps by the cognitive grasp planning. The cognitive grasp has three steps such as looking around to figure out the shape of an object, learning a grasp type by a human demonstration, and determining an appropriate grasp type for the target object by comparing with previous grasp types. The two former steps are related to the learning process, and the last step is the execution step.

In the learning step, a robot learns a grasp type of an object through a human demonstration. When the robot faces with a novel object, it looks around the target object and reconstructs a rough 3D model of the object. Although a robot cannot observe the target object from all viewpoints, the robot reconstructs an incomplete 3D model of the object by merging images obtained from several viewpoints. After that, the robot learns the grasp type of the object through a human demonstration. By correlating the grasp type with the 3D model, the robot finishes learning at last. On the other hand, in the execution step, the grasp process involves acquiring the shape information of the target object and searching for the most similar 3D model among previously learned models. Then the grasp type is chosen by the corresponding grasp type with the selected 3D model.

In this paper, we propose an intuitive and natural task planning for a cognitive grasp. We focus on the reconstruction of an object and the classification for the reconstructed object to determine the best grasp type. We do not perform the process on real robot architecture and remain it for the future works. In addition, we do not consider the decision of grasp points depending on the size, texture, and weight of the target

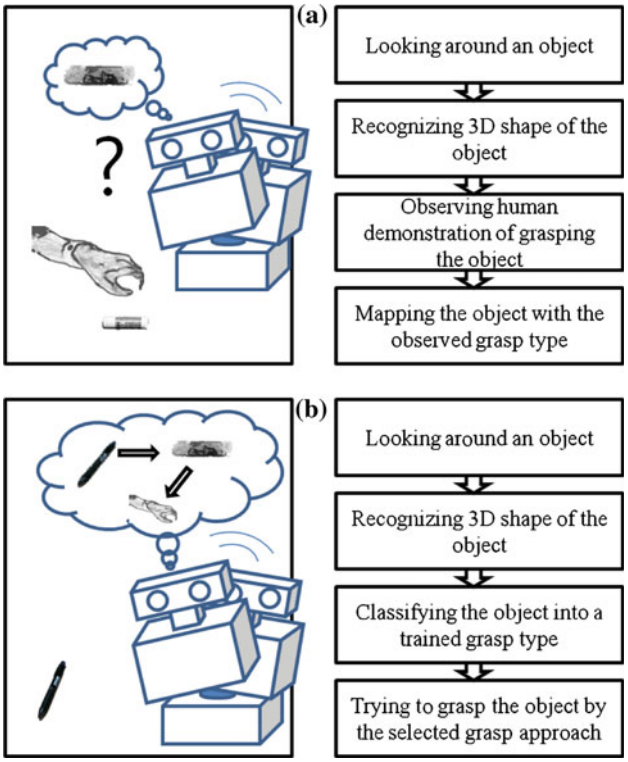

Fig. 1 Cognitive grasp planning. **a** Learning step. A robot learns the relation between the shape of an object and the corresponding grasp type by a human demonstration. **b** Execution step. When the robot meets an unknown object, it finds the most similar object in the knowledge that it has learned before and tries to grasp the object

object. Instead, the grasp planning can be applied to the various situations. In particular, when the robot is faced with an unknown object, it can figure out the appropriate grasp type by searching for a similar model learned previously.

4 3D object modeling

In this section, we describe a process of reconstructing a 3D model of a target object. A robot facing an object usually perceives only the frontal view of the object. Although the robot uses a depth sensor or a stereo camera, it cannot acquire enough 3D information of the object from one viewpoint. Therefore, we use sequential depth images, which are captured from different viewpoints, to reconstruct the 3D model of the target object.

Choi et al. [15] proposed a real-time 3D registration system that acquires and registers multi-view range images simultaneously. They used a stereo camera and merged depth images captured from different viewpoints for the 3D registration. Similarly, we also use 3D depth images to merge the images. However, we additionally apply a segmentation

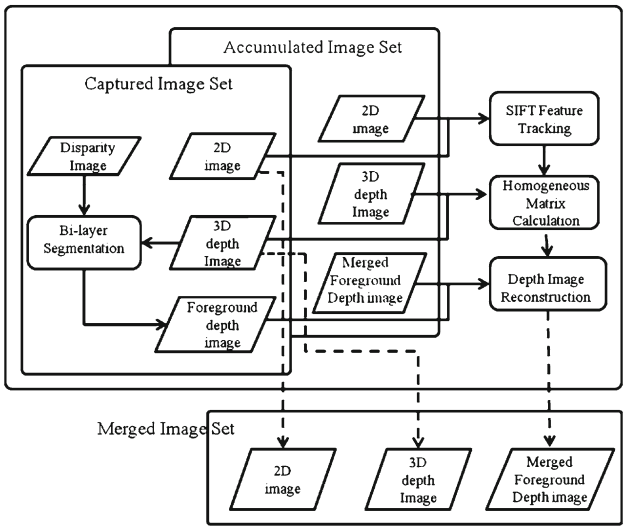

Fig. 2 A process of merging sequential image sets. Both an accumulated image set and a captured image set are combined into a merged image set. Then the merged image set replaces the accumulated image set

algorithm in order to reconstruct a foreground object in the complex environment.

4.1 Merging sequential images

The object modeling system accumulates all sequential image sets captured from different viewpoints. An image set contains 2D images (left view and right view), a disparity image, and a 3D point cloud. After saving the sequential image sets, the system starts to reconstruct a 3D shape model of the target object. Figure 2 shows the merging process, which combines an accumulated and captured image sets. At the beginning, the system merges the first captured image set with the second captured image set. As a result, the system produces the first merged image set. This merged image set is used to combine with the next captured image set, which is the third image set. This operation runs iteratively until the last captured image set is applied.

In the step of merging two sequential point clouds, each point has its own 3D coordinate. To align the coordinates, 3D position of point cloud should be transformed. The transforming equation is obtained by correlated points between the two depth images. The correlated points can be acquired by extracting features from 2D images and finding the corresponding features. There are several algorithms for extracting features and tracking them. We apply SIFT [16] because it can find precise features better than other algorithms [17]. Then a homogeneous matrix [18] for the transformation is calculated using the 3D data points connected to a corresponding feature set of 2D images. Now, all the points

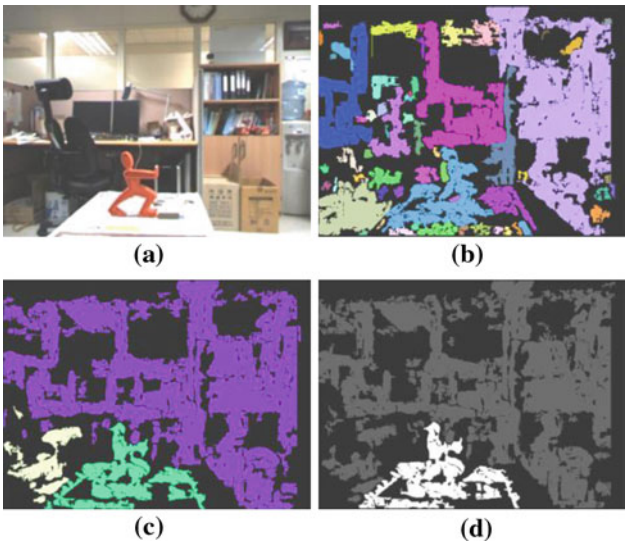

Fig. 3 A process of bi-layer segmentation. **a** a 2D image that can be divided by foreground and background, **b** labeling by disparity values, **c** assembling the fragmented labels according to the distance, size and location, **d** deciding the foreground area

of one 3D depth image are transformed and merged with the other 3D depth image using the homogeneous matrix.

4.2 Object segmentation

Although every point cloud is merged well, the accumulatively merged image contains useless information such as background and table area. Therefore, the segmentation of the target object from an environment model is necessary. We process two steps which are consisted of bi-layer (background and foreground) segmentation and the plane reduction. Kolmogorov et al. [19] developed a segmentation algorithm that extracts foreground area from background layers in stereo video sequences. It demonstrated good performance, but we need to consider more dynamic conditions. The target object may not be a foreground. Sometimes, the occlusion would appear in front of the robot view. Hence, we approach the problem of bi-layer segmentation with a more simplified but robust algorithm.

In this paper, we start from disparity images for bi-layer segmentation as shown in Fig. 3. In a disparity image, every valid pixel has a disparity value. If disparity values of two pixels are similar, which means the distances of two 3D points from a stereo camera are also similar. Each pixel on the image is assembled to the fragmented label based on the continuity of disparity values. When the arrangement of labels is done, information of distances between labels is used to determine whether each label is a foreground or not. More detailed procedure for segmenting and tracking the target area is described in [20].

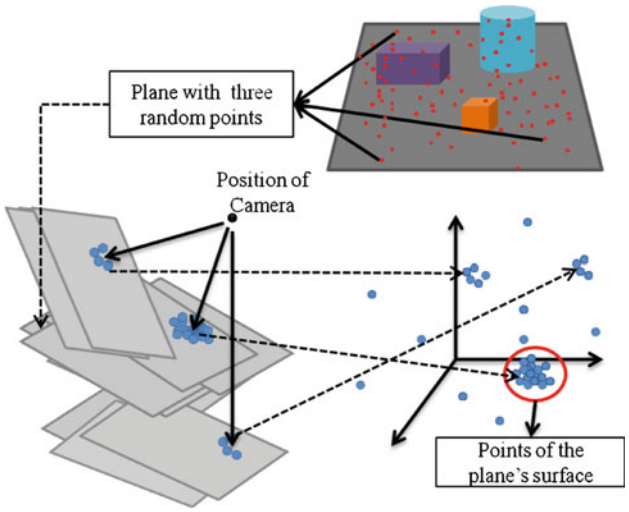

Fig. 4 An approach of plane reduction. Through this process, the table area is eliminated from a foreground model. Even in the case that many objects are placed on the table, they can be segmented at a time

In our targeting domain, a foreground image usually contains the target object and adjacent areas like a floor of a table. By eliminating the table or floor area from the foreground, the final object model can be obtained. Figure 4 shows the process of plane reduction. From the points that belong to the surface of the plane, we can calculate an equation of the 3D plane. Three points are randomly selected on a foreground image, and then a plane equation is formed by the three points. This step is repeated until the system obtains enough points to find the main plane. The area where the points are clustered indicates the surface plane because surface area occupies the largest area on the foreground. Finally, to segment the object, the plane area is eliminated using it. Consequently, the object image is extracted by eliminating all points under the surface of the plane.

5 Object classification

Finding similarity between reconstructed 3D models is to classify the object into an appropriate grasp types. We assume that the objects, which have similar shape information each other, can be separated to the same grasp type. In some cases, the grasp type can be determined by the pose of the target object; hence, the grasp type is differentiated based on the position that the object lays rather than the original shape of the object.

Previous researches [21,22] for recognizing similarity of 3D models are not suitable for the partial, unordered point cloud data set, which is created by our proposed modeling approach. Therefore, we apply machine-learning techniques to our data set. Principal component analysis (PCA) has been used for dimension reduction and pattern recognition.

Fig. 5 Projected depth image of 3D object models. The projected views are representative image of the 3D model of objects

In particular, it was applied to the recognition of 2D facial images as eigenfaces [23]. Some researchers have also tried to apply 3D range data to PCA, and they used the converted 2.5D data that contain depth information depending on viewpoints [24,25]. When 3D data are converted to 2.5D data, depth images also need to be modified. Therefore, according to the viewpoint, the system has to model many 2.5D data. On the other hand, there is another approach to calculate *eigenshape* using a full 3D shape model. In this approach, however, classification of models into similar shapes is more difficult because the eigenshapes are sensitive and changeable according to the size or the shape of the target object. In this paper, we convert a 3D object model into a 2D shape image, but we create only one depth image using a representative viewpoint. After that, we use the created depth image to obtain similarity between objects using PCA.

In order to apply 3D object models to PCA, we process two steps: alignment and normalization. In the alignment step, a best front view is selected from an object model. We consider only the rotation angle related to a plane of the table (in the segmentation process) rather than the arbitrary axis to align the object model. The point cloud data of the 3D model are projected to the most dominant plan that is orthogonal to the rotation plane. We calculate eigenvectors to find the principal axis of the 3D point cloud. Then the first eigenvector can be obtained, which is corresponding to the frontal face of the object. 3D data of the model are converted to a gray-scale depth map that has a pixel value range from 0 to 255. Now we obtain a 2D projected depth image (see Fig. 5). In the normalization step, the projected images are modified to regularize the size and the position. Through these steps, we extract coefficients of eigenvectors by PCA decomposition from depth images. The coefficients are used to recognize similarity of objects by comparing distance between them.

The performance of the system is determined by the alignment of objects, classification of grasp types, and classification methods. In order to compare coefficients, we apply classification method such as Euclidean distance, multi-layer perceptron, and support vector machine.

The advantage of our proposed eigenshape is that it maintains benefit of PCA. The objects can be classified very quickly because the system compares only coefficients set, which has reduced dimension, instead of using 3D model data that contain huge dimension. In addition, even if a model is incomplete and a partial point cloud set, the 3D model can be presented to a typical 2D image so that the classification does not strongly depend on the modeling performance.

6 Experimental result

6.1 Setup

We performed experiments of object modeling and classification for various objects that a robot can grasp in the real world. Instead of a robot system, we used a stereo camera (*Bumble bee 2).* The image resolution of the camera is fixed to 640 × 480. The target objects were placed on a table. The camera rotated around the object to capture sequential images of the object. We tested totally 20 objects. Tests were repeated three times at two different rotating angles for each object. The two test angles were 90° and 180°.

6.2 Modeling process

The process of modeling was accomplished by the order as shown in Fig. 6. The modeling system extracted features and segmented foreground area from each frame in sequential images that were obtained by a stereo camera rotating around the target object. After the system merged all the foreground images based on the corresponding features, it reconstructed a 3D environment that includes both a target object and table area like Fig. 6b. Finally, the system acquired a 3D object model (see Fig. 6c) using a plane reduction process that eliminates the table area.

6.3 Object classification by grasp types

Figure 7 demonstrates that a grasp type of an object can be changed by the original shape of objects or the pose of the object. According to the grasp demonstration of a human wearing a data glove, our system learned the relation between objects and grasp types. Figure 7a and b shows that different objects had a similar grasp type. Likewise, Fig. 7e and f represents that the grasp types could be different for a same object.

We divided the objects into nine grasp types by considering the shape of object (see Fig. 9), and evaluated the recognition rate after learning the classified types (see Fig. 8). We tested the coefficients that were acquired from PCA with three classification methods such as Euclidean distance,

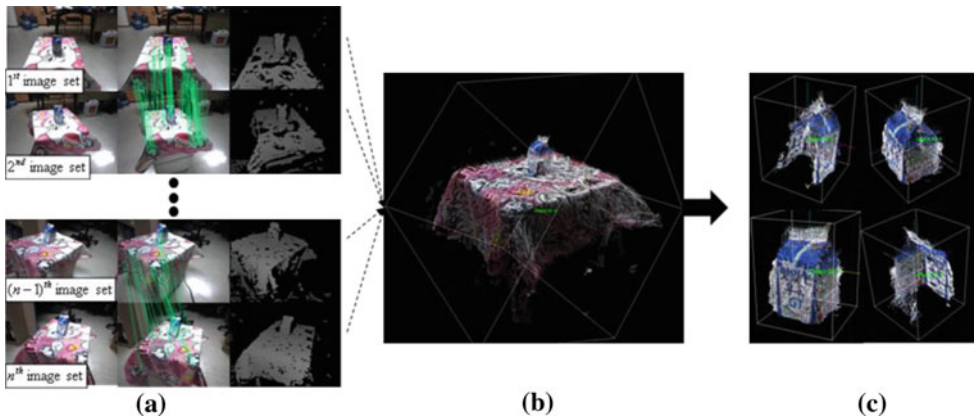

Fig. 6 The process of modeling a 3D object. **a** After acquiring the sequential images, each image is processed to segment foreground area and extract feature points (*rightmost images*). Then, corresponding features between adjacent images are selected (*center images*). **b** All the sequential images are merged into a foreground model using homogeneous transform. **c** The 3D object is segmented by the process of plane reduction. The model is composed of a partial, point cloud data

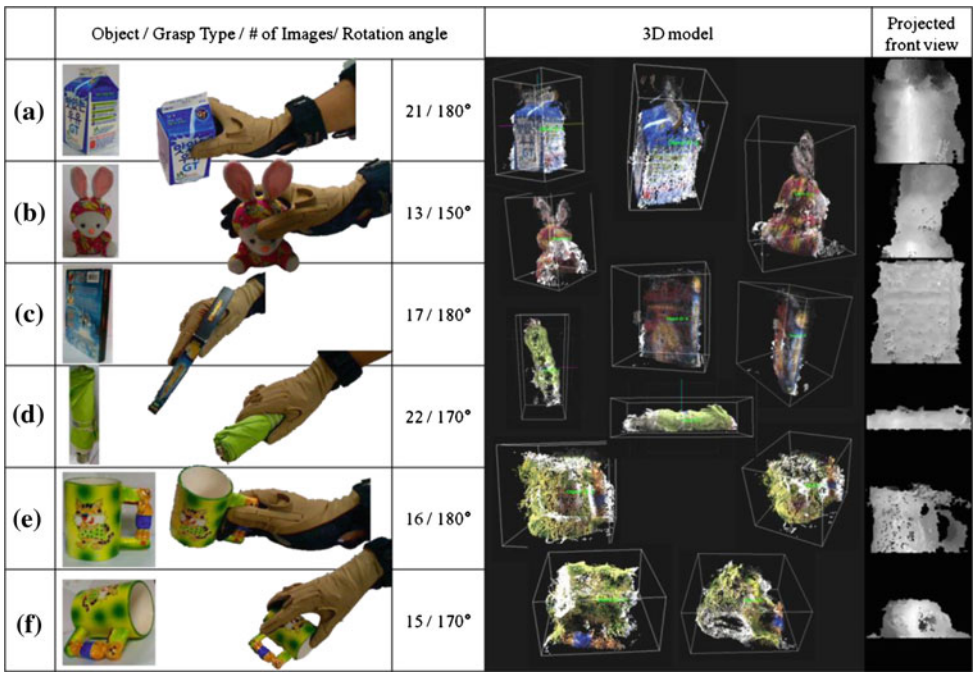

Fig. 7 *Left side* The correlation of object and its grasp type. The grasp type is learned by a human demonstration. A human wears a data glove that captures movement of a hand. *Right side* 3D models of objects and its projected front views

multi-layer perceptron (neural network), and support vector machine. The Euclidean distance metric was modified to include the weight of distance according to dimension of the coefficients. The recognition rate was, respectively, 89.1, 91.67, and 87.5% for the previous trained objects. The precision and recall were high when the shape of an object group was unique as shown in Fig. 8. For the unknown object that a robot had never learned before, we tested only three groups (group 3, 4, and 5) which contain more than three different objects. Our approach showed that the robot could choose an appropriate grasp type according to the shape and the state of the object (see Table 1).

7 Discussion

Our proposed 3D modeling algorithm has high computational overhead. In this approach, the system deals with complex 3D data and applies SIFT algorithm which requires high computation cost. The image resolution even is high because small objects that a robot grasps need to be recognized. In the real situation, however, further speed improvement is necessary for a natural interaction between a human and a robot. By decreasing the number of merged images, the computation time can be reduced without change of performance of modeling and classification.

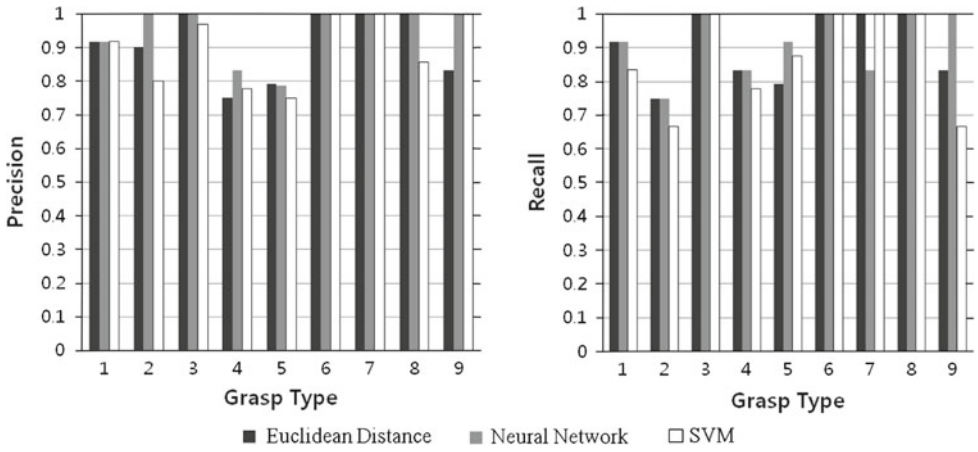

Fig. 8 The recognition results of object grasp types

Fig. 9 Nine grasp types among 20 objects

On the other hand, two objects, which have a similar shape but different sizes, can be determined to a same grasp type since our approach normalizes the size of objects for learning object models. If two people demonstrate grasping of an object to a robot with differentiated grasp types, the robot confuses to select an appropriate grasp type of the object. These problems are rather connected to the cognitive grasp planning. By adding information (e.g. the size and identity of the object, the status of a person who interacts with a robot, environment, and context) to the data set, the performance of the cognitive grasp can be improved.

Table 1 Classification of unknown objects

Measurement	Grasp types		
	3	4	5
Precision	1.0	0.556	0.6
Recall	1.0	0.526	0.652

8 Conclusion

In this paper, we proposed a robotic cognitive grasp planning that was imitated through a human's grasping approach. A robot learns the grasping skill from a human demonstration, and then it tries to grasp an unknown object based on previous knowledge. In order to implement the basic functions for the cognitive grasp, we made a robot automatically recognize the shape of an object and find the appropriate grasp type.

The proposed approach integrated several functions for execution of human-like learning. There is, however, still a lot of contextual information that a robot should consider. In addition, this cognitive grasp needs to be applied to a robot hardware platform to be performed in real world situation. We anticipate that this research can be a basic approach for the cognitive grasp.

References

1. Tegin J, Ekvall S, Kragic D, Wikander J, Iliev B (2009) Demonstration-based learning and control for automatic grasping. Intel Serv Robotics 2:23–30
2. Mae Y, Takahashi H, Ohara K, Takubo T, Arai T (2011) Component-based robot system design for grasping tasks. Intel Serv Robotics 4:91–98
3. Perrin DP, Smith CE, Masoud O, Papanikolopoulos N (2000) Unknown object grasping using statistical pressure models. In: Proceedings of the IEEE international conference on robotics and automation
4. Morales A, Recatalá G, Sanz PJ, Pobil APD (2001) Heuristic vision-based computation of planar antipodal grasps on unknown

objects. In: Proceedings of the IEEE international conference on robotics and automation
5. Taylor G, Kleeman L (2002) Grasping unknown objects with a humanoid robot. In: Australasian conference on robotics and automation
6. Miller AT, Knoop S, Christensen HI, Allen PK (2003) Automatic grasp planning using shape primitives. In: Proceedings of the IEEE international conference on robotics and automation
7. Saxena A, Driemeyer J, Ng AY (2008) Robotic grasping of novel objects using vision. Int J Robot Res 27(2):157–173
8. Yamazaki K, Tomono M, Tsubouchi T, Yuta S (2004) 3-D object modeling by a camera equipped on a mobile robot. In: Proceedings of the IEEE international conference on robotics and automation
9. Yamazaki K, Tomono M, Tsubouchi T, Yuta S (2006) A grasp planning for picking up an unknown object for a mobile manipulator. In: Proceedings of the IEEE international conference on robotics and automation
10. Yamazaki K, Tomono M, Tsubouchi T (2008) Picking up an unknown object through autonomous modeling and grasp planning by a mobile manipulator. Springer, Berlin 42
11. Maeda Y, Ishido N, Kikuchi H, Arai T (2002) Teaching of grasp/graspless manipulation for industrial robots by human demonstration. In: Proceedings of the IEEE international conference on intelligent robots and systems
12. Ekvall S, Kragic D (2005) Grasp recognition for programming by demonstration. In: Proceedings of the IEEE international conference on robotics and automation
13. Ekvall S, Kragic D (2004) Interactive grasp learning based on human demonstration. In: Proceedings of the IEEE international conference on robotics and automation
14. Hueser M, Baier T, Zhang J (2006) Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. In: Proceedings of the IEEE international conference on robotics and automation
15. Choi SI, Park SY, Kim J, Park YW (2008) Multi-view range image registration using CUDA. In: Proceedings of the international technical conference on circuits/systems, computers and communications
16. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
17. Yilmaz A (2006) Object tracking: a survey. ACM Comput Surv 38(4):article 13
18. Criminisi A, Reid I, Zisserman A (1997) A plane measuring device. Image Vis Comput 17(8):625–634
19. Kolmogorov V, Criminisi A, Blake A, Gross G, Rother C (2005) Bi-layer segmentation of binocular stereo video. In: Proceeding conference on computer vision and pattern recognition
20. Han I, Kim H, Kim K, Park JH (2010) Object segmentation and tracking from sequential stereo images for 3D object modeling. In: Proceedings of Korean conference on HCI Korea
21. Tangelder JWH, Veltkamp RC (2008) A survey of content based 3D shape retrieval methods. Multimed Tools Appl 39:441–447
22. Biasotti S, Falcidieno B, Frosini P, Giorgi D, Landi C, Marini S, Patane G, Spagnuolo M (2007) 3D shape description and matching based on properties of real functions. In: Eurographics
23. Turk M, Pentland A (1991) Face recognition using eigenface. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 586–591
24. Campbell R, Flynn P (1999) Eigenshapes for 3D object recognition in range data. In: Proceedings of the IEEE conference on computer vision and pattern recognition
25. Zhao L-W, Luo S-W, Liao L-Z (2004) 3D object recognition and pose estimation using kernel PCA. In: Proceedings of the international conference on machine learning and cybernetics