2014 IEEE International Conference on
Automation Science and Engineering (CASE)
Taipei, Taiwan, August 18-22, 2014

3D Object Detection and Pose Estimation from Depth Image for Robotic Bin Picking

Hao-Yuan Kuo, Hong-Ren Su, Shang-Hong Lai, and Chin-Chia Wu

Abstract—**In this paper, we present a system for automatic object detection and pose estimation from a single depth map containing multiple objects for bin-picking applications. The proposed object detection algorithm is based on matching the keypoints extracted from the depth image by using the RANSAC algorithm with the spin image descriptor. In the proposed system, we combine the keypoint detection and the RANSAC algorithm to detect the objects, followed by the ICP algorithm to refine the 3D pose estimation. In addition, we implement the proposed algorithm on the GPGPU platform to speed-up the computation. Experimental results on simulated depth data are shown to demonstrate the proposed system.**

I. Introduction

3D object alignment is essential to bin-picking applications. It usually consists of object detection and pose estimation. Most of the previous methods are based on the 2D image data. As depth sensor technology is getting mature in recent years, we can easily acquire real depth data for industrial applications. However, it is quite challenging to efficiently align 3D objects only from depth data. In this paper, we focus on aligning 3D industrial objects, such as those depicted in Fig. 1, from a single depth image.

Depth image contains some advantages over the 2D color image counterpart. The main advantage is the depth image contains rich and direct geometric information. However, there are some challenges for object detection from depth image. First, the texture information on the object is missing in the depth image. Second, depth data is usually quite noisy. Finally, converting depth image to 3D representation, such as the point cloud, for 3D alignment usually requires high computational cost.

In this paper, we present an efficient 3D object alignment system for bin picking from a single depth image. The proposed system consists of 3D object detection and pose estimation. The object detection from depth image is accomplished by detecting keypoints from depth image and finding correspondences between keypoints extracted from the input and template depth images by using the RANSAC algorithm. Then, the 3D pose of the object is refined by using the ICP algorithm. In addition, we accelerate the 3D object alignment system via implementation on the GPGPU platform.

II. Related work

Previous works in 3D alignment from point-cloud data could be divided into the following categories: segmentation, classification, matching, modeling, registration and detection.

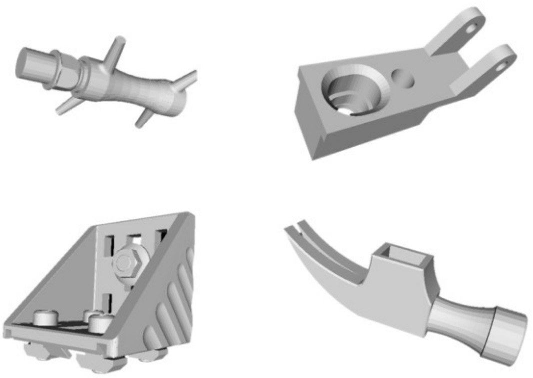


Figure 1. CAD models of some industrial parts.

There are several previous works proposed to process different types of point cloud data for different purposes. One category is focused on urban scenes. For example, some focused on developing algorithms for detecting vehicles [1][2], and some focused on detecting poles [3][4]. Another category deals with indoor scenes. For example, [5] presented a model-based method to detect chairs and tables in the office. [6] proposed a graphical model to capture various features for indoor scenes. [7] proposed a system to obtain 3D object maps from scanned point-cloud datasets of indoor household objects. However, only a few previous works developed algorithms for processing point cloud data for industrial applications. In [8], Vosselman et al. developed techniques for recognition in industry as well as urban scenes. Schnabel et al. [9] presented a RANSAC algorithm to detect some basic shapes, such as plane, spheres, cylinders, cone and tori, from the point-cloud dataset. Liu et al. [10] developed a novel 3D object alignment system based on using an image acquisition system that captures the contour of the object. However, it may not work for objects with complex shapes.

The 3D local shape descriptor is very critical to the keypoint matching from depth images. For example, spin image [11][12] is one of the most widely used 3D descriptor. Some 3D descriptors are extended from the original 2D descriptors, such as the 3D Shape Context[13], 3D SURF [14] and 3D SSIM [15]. Heat Kernel Signature [16] and its variation [17] can be applied to deal with non-rigid shapes. In this paper, we use spin image and 3D detector [18] in our system.

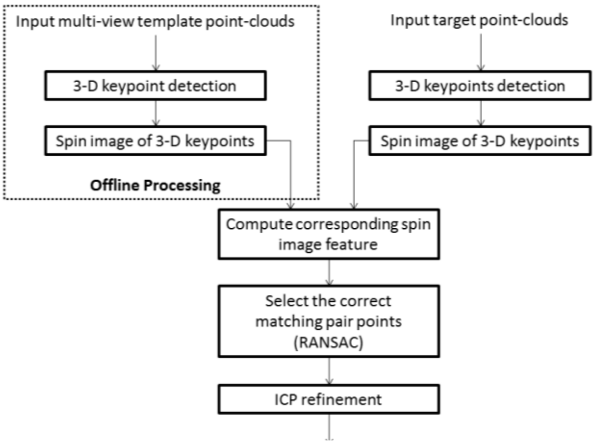
978-1-4799-5283-0/14/$31.00 ©2014 IEEE

III. System overview

The pipeline of the proposed 3D object alignment system is shown in Fig. 2.

The point-cloud data in our system is generated from the depth information. Unlike general depth image, the point-clouds data are more precise. We use the structure of the object as the detection information.

Our goal is to detect the object and estimate its 3D pose based on different pose hypotheses for the object. We define the template as a specific view of the object and the target as the detected scenes. Different views of the object contain different partial information of the object, so we simulate a sufficient number of depth maps of the object at different poses from its 3D model as the templates. In our problem, there are many the identical objects in the container, as depicted in Fig. 3. In the target scene, there are many identical objects, so it might contain numerous possible detection results. In this work, we are interested in finding the object closest to the camera in target scenes as our detection target.


Input multi-view template point-clouds
3-D keypoint detection
Spin image of 3-D keypoints
Offline Processing
Input target point-clouds
3-D keypoints detection
Spin image of 3-D keypoints
Compute corresponding spin image feature
Select the correct matching pair points (RANSAC)
ICP refinement

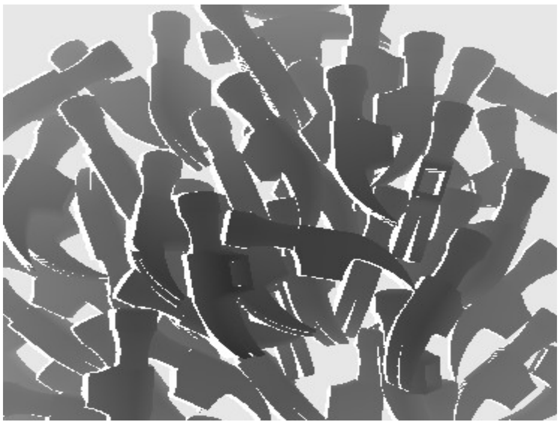


Figure 3. Objects stacked in container

We use the 3D keypoint detection to extract interest points, such like the corner points, as candidate points for matching. Then, we compute the spin image feature at the keypoints and use RANSAC to find the best point matching between the candidate point sets in the target and template. However, the target and template might not align well due to partial occlusion and noise in the keypoint data, so we apply the Iterative Closest Points (ICP) algorithm to refine the 3D pose estimation result.

A. 3-D keypoint detection

Given a point of a 3-D object, we are interested in finding the corner point. However, the structure of the 3D object is different from the 2D object. We need mesh from the point-cloud to obtain adjacency information for the neighboring points.

Let v be the interest point and $V_k(v)$ be the neighborhood considering k rings around v. Figure 4 shows point v (black circle), the first ring around v (path formed by green circles), and the second ring (path formed by yellow circles). All these points correspond to the neighborhood $V_k(v)$. Then, we compute the best fitting plane to the translated points via Principal Component Analysis to the set of points and compute the 3-D Harris corner response to extract the keypoint [18].

We extract keypoints on the template and target data. Let U be the points in template and W be the points in target. H denotes the 3-D keypoint detection. Let υ be the keypoint from template and ω be the keypoint from target.

$$\upsilon = \mathrm{H(U)} \qquad \omega = \mathrm{H(W)} \tag{1}$$

B. Spin image of 3D keypoint

Johnson and Hebert [11] proposed the spin-image for object recognition. A spin image is 2D representation of the surface surrounding a 3-D point as Fig. 5. The formulation is:

$$S_o : R^3 \to \mathrm{R}^2 \qquad S_o \to (\alpha,\beta) = (\sqrt{\|\mathrm{x-p}\|^2 - (n\cdot(x-p)^2)}, \mathrm{n\cdot(x-p)}) \tag{2}$$

An oriented point O at a surface mesh vertex is defined by

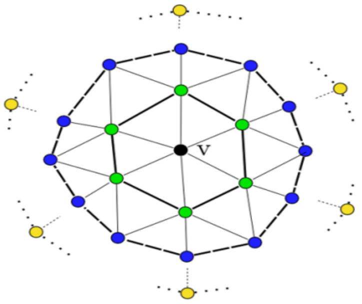


Figure 4. Point v and its neighbor rings

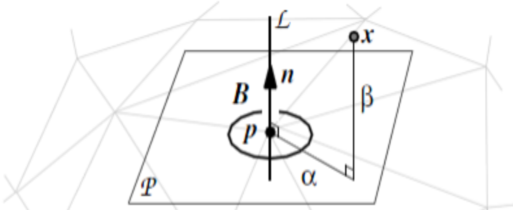


Figure 5. The cylindrical system and its (p,n) 2-D basis

the 3-D position of the surface vertex (denoted as p) and a surface normal (denoted as n). With p and n defined, we can formulate a 2D basis (p, n), which corresponds to n and the line L through p parallel to n. This results in a (α,β) cylindrical coordinate system, where α is the perpendicular distance to L while β is the signed perpendicular distance to P.

We use the spin image as the 3D descriptor and we compute it for all keypoints for finding point correspondences. For example, there is a triangular body converted into a point-cloud representation. The structure of the triangular body with four corner points can be detected via 3-D keypoint detection and the matching problem focus on the spin image of the keypoints. It is easy to find the correct correspondences between the two sets of keypoints by comparing their spin images.

We add the equation (1) to equation (2) to obtain the spin image from template (denoted as P) and target (denoted as Q) as follows:

$$P = S_{oP} \rightarrow (\alpha,\beta) = (\sqrt{\left\|\mathrm{x}_{\mathrm{p}} - \upsilon\right\|^2 - (n_p \cdot (x_p - \upsilon)^2)}, \mathrm{n}_{\mathrm{p}} \cdot (\mathrm{x}_{\mathrm{p}} - \upsilon))$$

$$Q = S_{oQ} \rightarrow (\alpha,\beta) = (\sqrt{\left\|\mathrm{x}_{\mathrm{q}} - \omega\right\|^2 - (n_q \cdot (x_q - \omega)^2)}, \mathrm{n}_{\mathrm{q}} \cdot (\mathrm{x}_{\mathrm{q}} - \omega)) \tag{3}$$

The subscript p and q denote that it is from template or target, respectively. Then, we have to compute the correlation coefficient between the spin images of the points in the template and target, respectively, and it is given by

$$R(P,Q) = \frac{N\sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N\sum p_i^2 - (\sum p_i)^2)(N\sum q_i^2 - (\sum q_i)^2)}} \tag{4}$$

Note that p_i denotes the size of the spin image from P and q_i denotes the size of the spin image from Q and N denotes the bin size from the spin image. R is between -1 (anti-correlated) and 1 (completely correlated), and it measures the normalized error using the distance between the data and the best line fitted to the data. The correlation coefficient R provides a measure for the comparison of two spin images. The higher the measure R is, the more similar between the spin images are. Thus, we can find similar matching points based on the correlation coefficients as the candidates for point correspondence pairs. Thus, we can obtain the corresponding points from template, denoted as U_c, and from target, denoted as W_c

C. *Select the point correspondence pairs*

There are many similar pairs of matching points but they may contain wrong correspondences, which are considered as outliers. For rigid transformation, we need to solve for rotation matrix R and translation matrix T from the set of point correspondences. Our target is the object that is similar to the template and closest to the camera. We use the concept of RANSAC to pick up the correct matching pair points to align the target to the template. To solve for the rigid transformation, we need at least 3 correct pairs of point correspondences. The rigid transformation is given by:

$$U' = RU + T \tag{5}$$

The following two steps are performed to confirm that the rigid transformation (R, T) estimated from the randomly selected three pairs of candidate point correspondences is acceptable.

In the first step, we measure how well the estimated rigid transformation matches to the remaining pairs of candidate point correspondences, whose points in template are denoted by U_c and those in target are denoted by W_c. We use eq. (5) to compute the transformed U_c as U_c' . Then, the mean distance between the matching points in U_c' and W_c is defined by:

$$P_e = \frac{\left\|U_c' - W_c\right\|}{\sqrt{n}} \tag{6}$$

where n is the total number of point correspondence pairs. If P_e is less than a threshold ε_1, then we confirm that the estimated transformation provides correct fitting from the target to the template.

In the second step, we compute the mean position from transformed template as U_{m}' and the same for the target as W_{m}, We want to ensure that the alignment from the template to target is on the object and close to the camera, so we define another error C_e as follows:

$$C_e = \left\|\mathrm{U}_{\mathrm{m}}' - \mathrm{W}_m\right\| \tag{7}$$

If C_e is less than a threshold ε_2, we can confirm that the alignment from template to target is close to the camera. Once the object that is close to the camera is detected, then we apply the ICP algorithm to refine the 3D pose of the detected object.

D. *GPU acceleration*

Extensive parallel processing by using graphics processing unit (GPU) can be employed in the implementation of our 3D alignment system. We have many hypothesis poses of the object obtained from the RANSAC process. We detect the 3D keypoints from the depth image and compute the associated spin images. Our acceleration is focused on finding the correct pairs of matching points from different views. We use the concept of parallelization on the matching problem. For a hypothesized pose of the object, we need to run specific round to confirm if the number of correct pairs of matching points is sufficient enough or not. We parallelize the computation for all hypothesized poses of the object in the RANSAC process on the GPGPU platform to accelerate the proposed 3D object alignment system.

IV. Experimental Results

A. *Hypothesized pose of object*

The template contains different poses of the object. The point-cloud data are generated from the depth information, so the depth map of the object from one view only contains partial information of the object. The 3D object is more complex. We try to use a small number of depth maps of the object from different views to represent the 3D object. In our experiments, we observe that the tolerance of deviation in the rotation angle is about 20 degree. We use different objects to evaluate the pose estimation accuracy by using the proposed algorithm. We use the specific pose of the object and randomly select the object poses to synthesize the depth images (see Fig.6 for example). Then, we compute the pose estimation and the error of the pose estimation by using the proposed pose estimation algorithm, and the errors are summarized in Table I.

Algorithm 1 Alignment from template to target

```
INPUT : $U_c$ , $W_c$ , U, W
OUTPUT : estimated pose
Set $C_e = C_{max}$, $P_e = P_{max}$
While $C_e > \varepsilon_2$
    While $P_e > \varepsilon_1$
        iter = 1
        Select 3 similar matching pair points randomly.
        Compute the rigid transformation.
        $U_c^{'} = RU_c + T$
        $P_e = \left\| U_c^{'} - W_c \right\| / \sqrt{n}$
        iter = iter + 1
        If iter > Iter_max
            Break;
        END If
    END While
    If $P_e > \varepsilon_1$
        Return 1, Break
    END If
    U’ = RU + T
    Compute the means for U’ and W as $\mathbf{U}_m^{'}$ and $\mathbf{W}_m$
    $C_e = \left\| U_m^{'} - W_m \right\|$
END While
ICP Refinement on U’ and W
Compute the pose estimation
```

TABLE I. ERROR OF POSE ESTIMATION FOR THE CASE WITH A SINGLE OBJECT

6DOF	$\|\alpha\|$	$\|\beta\|$	$\|\gamma\|$	ΔR	$\|T_x\|$	$\|T_y\|$	$\|T_z\|$	ΔT
Obj. A	2.92	1.17	1.22	3.59	0.48	0.90	1.21	1.71
Obj. B	1.23	2.43	1.48	3.10	0.38	1.57	1.12	1.96
Obj. C	1.87	3.21	2.43	4.43	1.83	2.13	1.56	3.21
Obj. D	1.55	0.24	0.76	1.74	0.83	1.13	0.24	1.42

In Table I, we calculate the average errors of the pose estimation on four objects. Note that $|\alpha|$, $|\beta|$, and $|\gamma|$ denote absolute value of rotation errors along x-axis, y-axis, and z-axis, respectively, and $|T_x|$, $|T_y|$, and $|T_z|$ denote absolute value of translation error along x-axis, y-axis, and z-axis. The rotation is in degree and the translation is in millimeter. The total rotation error ΔR and total translation error ΔT are defined by:

$$\begin{aligned} \Delta R &= \sqrt{|\alpha|^2 + |\beta|^2 + |\gamma|^2} \\ \Delta T &= \sqrt{|T_x|^2 + |T_y|^2 + |T_z|^2} \end{aligned} \tag{8}$$

To represent the 3D model for an object, we use a simulator to generate the depth images of an object along x-axis from 0^o to 180^o and y-axis from 0^o to 360^o. Thus, We have totally 172 poses of an object. We use some industrial objects, as depicted in Fig. 7, in our experiments to evaluate the accuracy of the proposed system.

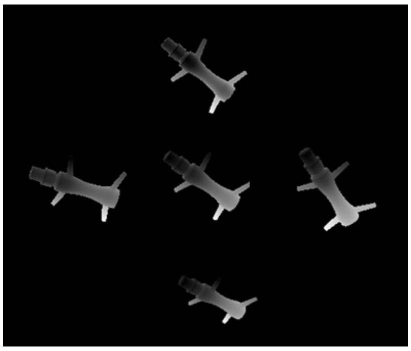


Figure 6. The center is a specific pose of the object and others are similar poses of the object.

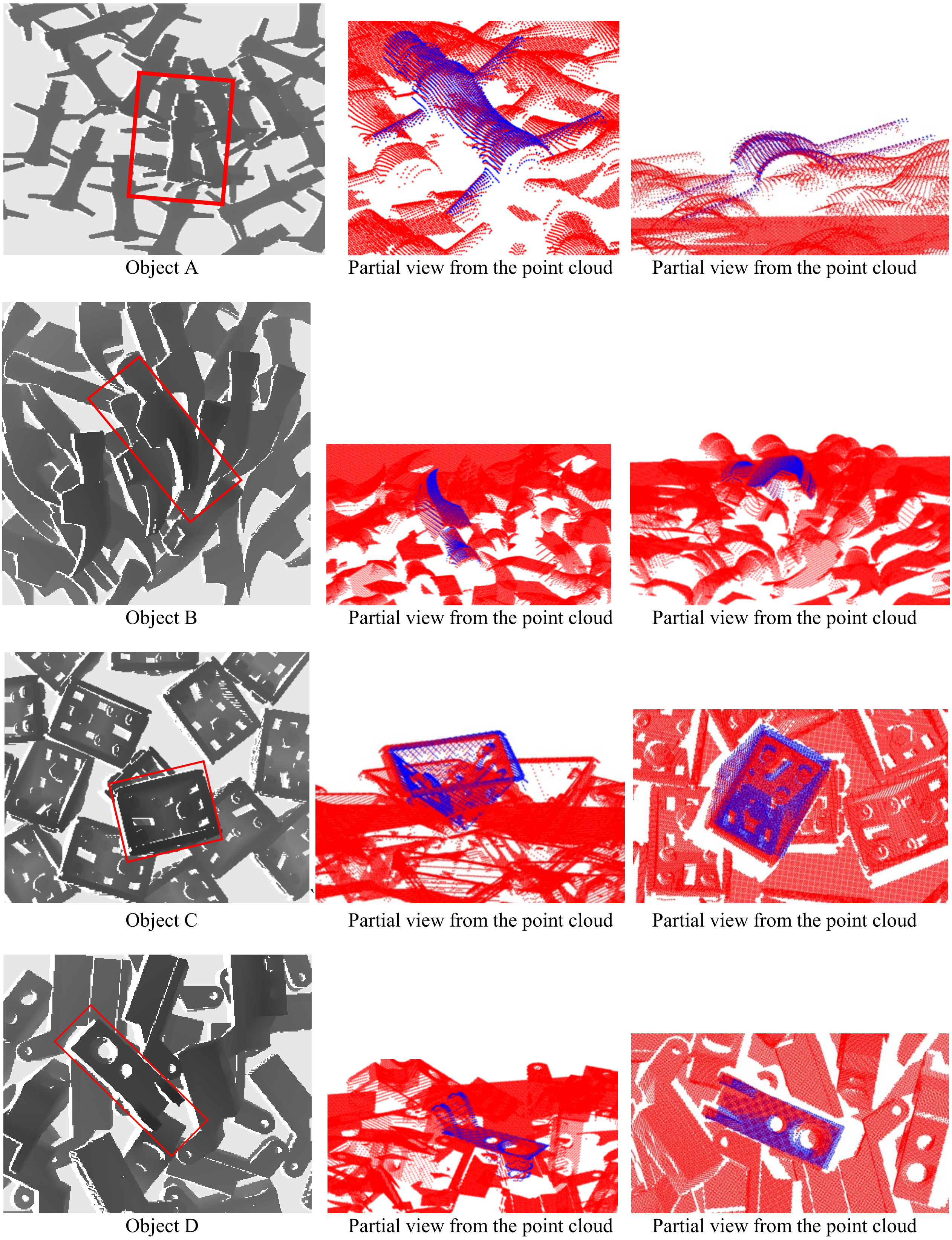
B. *Object pose estimation in container*

Given the position of the camera, we synthesize depth maps of many identical objects in the container scenes as the target depth images. We have the ground truth of poses of the object closest to camera. An example of the target depth image is depicted in Fig.3. Our goal is to estimate the pose of the object closest to the camera. For visualization, we show the detected result in point-cloud display in Fig. 7. The red bounding box contains the detected object that is closest to the camera.

We experiment on the target depth images, and the average errors in pose estimation are summarized in Table II. The error of Object C is slightly higher than the others. It is because object C is with similar spin images around the bounding region, so it might cause the matching result deviates from the ground truth.

TABLE II. ERROR OF POSE ESTIMATION ON MULTIPLE OBJECTS

6DOF	$\|\alpha\|$	$\|\beta\|$	$\|\gamma\|$	ΔR	$\|T_x\|$	$\|T_y\|$	$\|T_z\|$	ΔT
Obj. A	0.34	0.64	0.43	0.84	0.32	0.22	0.65	0.75
Obj. B	0.83	0.64	1.22	1.61	0.34	0.54	0.48	0.80
Obj. C	1.42	2.59	3.12	4.30	1.92	2.24	2.84	4.10
Obj. D	1.52	2.13	1.34	2.94	0.88	1.49	1.76	2.47



Object A

Partial view from the point cloud

Partial view from the point cloud

Object B

Partial view from the point cloud

Partial view from the point cloud

Object C

Partial view from the point cloud

Partial view from the point cloud

Object D

Partial view from the point cloud

Partial view from the point cloud

Figure 7. The 3D alignment results for different objects. The first column contains the target depth images. The second and third columns are different views of the target point clouds overlaid with the detected object template. The blue points are from template and the red points are from target. If the region of the overlapping is high, it means the result is better.

C. Detection Rate

We experiment on different objects, depicted in Fig.1, to evaluate the detection rate of the proposed algorithm. We use 100 different target depth images, with each containing many identical objects. The object in the target depth image is successfully detected if the translation error and the rotation error in the estimated 3D pose are smaller than predefined thresholds (5 degree and 5 mm).

In Table III, we can see that the objects with simple structures can be successfully detected. For the objects with complex structure, the detection rates are decreased because the complex structure of the object captured from different views is more difficult to match well from a single depth image.

TABLE III. DETECTION RATE ON DIFFERENT OBJECTS

Object	Detection Rate	False Detections
Obj. A	96%	4
Obj. B	93%	7
Obj. C	84%	16
Obj. D	92%	8

D. GPU acceleration

We implement our system in C++ with GPGPU by using CUDA programming. Our computing platform is equipped with Intel(R) Core(TM) i7-27670QM CPU and NVIDIA GeForce GT 630M GPU.

We focus on parallelizing the procedure of selecting the correct pairs of matching points from different poses of the object. We compute the template keypoints of different poses offline and obtain the correct pairs of matching points. Our execution time of using the proposed algorithm on CPU and GPGPU computing platforms is summarized in Table IV. It takes most of the time in RANSAC on CPU because every candidate object pose needs to be computed sequentially to find the candidate set of matching points. In our experiment, the GPGPU implementation is about 150 times faster than that implemented on CPU for the RANSAC procedure.

TABLE IV. AVERAGE EXECUTION TIME OF THE PROPOSED ALGORITHM ON CPU AND GPGPU COMPUTING PLATFORMS (UNIT: SECOND)

Exec. Time	Keypoint detection	Spin Image	RANSAC	ICP	Total time
CPU	2.21	1.01	312.34	1.72	318.34
GPGPU	2.13	1.56	2.48	1.84	9.06

V. CONCLUSION

We presented a novel 3D object alignment system to detect the object from the depth image for robotic bin-picking applications. The proposed system consists of 3D object detection and pose estimation from a single depth image. For object detection, our algorithm combines 3D keypoint detection, spin-image extraction, and RANSAC to achieve robust and efficient object detection from depth data. In addition, we implement the proposed object detection algorithm on the GPGPU platform to accelerate the computation. We evaluate the proposed 3D object alignment system on simulated depth images. In the future, we will perform extensive testing of our system on real data.

VI. REFERENCES

[1] A. Patterson, P. Mordohai and K. Daniilidis, “Object detection from large-scale 3D datasets using bottom-up and top-down descriptors,” ECCV 2008.

[2] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, “Recognizing objects in range data using regional point descriptors,” ECCV 2004.

[3] H. Yokoyama, H. Date, S. Kanai and H. Takeda, ”Detection and classification of pole-like objects from mobile laser scanning data of urban environments,” ACDDE 2012.

[4] M. Lehtomaki, A. Jaakkola, J. Hyyppa, A. Kukko, H. Kaartinen, “Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data,” Remote Sensing 2010.

[5] B. Steder, G. Grisetti, M. V. Loock and W. Burgard, ”Robust on-line model-based object detection from range images,” IROS 2009.

[6] H. Koppula, A. Anand, T. Joachims and A. Saxena, “ Semantic labeling of 3D point clouds for indoor scenes,” NIPS 2011.

[7] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D point cloud based object maps for household environments,” Robotics and Autonomous Systems Journal, 2008.

[8] G. Vosselman, B. Gorte, G. Sithole, T. Rabbani,” Recognising structure in laser scanner point clouds,” IAPRS 2004.

[9] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for point-cloud shape detection,” Computer Graphics Forum, Vol. 26, no. 2, pp. 214-226, June 2007..

[10] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. Marks, and R. Chellappa,”Fast object localization and pose estimation in heavy clutter for robotic bin picking,” IJRR 2012.

[11] A. Johnson and M. Hebert,” Object recognition by matching oriented points,” CVPR 1997.

[12] S. Ruiz-Correa, L. G. Shapiro, and M. Melia, ”A new signature-based method for efficient 3-D object recognition” CVPR, 2001.

[13] M. Kortgen, G.-J. Park, M. Novotni, and R. Klein, ”3D shape¨matching with 3D shape contexts,” Central European Seminar on Computer Graphics, April 2003.

[14] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, ”Hough transform and 3D SURF for robust three dimensional classification.” ECCV 2010.

[15] J. Huang, and S. You, “Point cloud matching based on 3D self-similarity,” Intern. Workshop on Point Cloud Processing, 2012.

[16] J. Sun, M. Ovsjanikov, and L. Guibas,” A concise and provably informative multi-scale signature based on heat diffusion,” Symposium on Geometry Processing, Berlin, July 2009

[17] M. M. Bronstein and I. Kokkinos,” Scale-invariant heat kernel signatures for non-rigid shape recognition.,” CVPR 2010.

[18] I. Sipiran, and B. Bustos ,”Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshed,” Visual Computer, Vol. 27, No. 11, pp. 963-976, 2011.

[19] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, ”Aligning point cloud views using persistent feature histograms,” IROS 2008.