International Journal on Interactive Design and Manufacturing (IJIDeM) (2021) 15:239–247
https://doi.org/10.1007/s12008-021-00762-4

ORIGINAL PAPER

3D matching by combining CAD model and computer vision for autonomous bin picking

Le Duc Hanh[1] · Khuong Thanh Gia Hieu[1]

Received: 29 June 2021 / Accepted: 27 July 2021 / Published online: 5 August 2021

Abstract
Since today, most of the manufacturing companies have the operated CAD model, so CAD-based object autonomous bin picking which using 6DOF Manipulator may be a good option that can save time and increases productivity for an assembly line. This research aims to present an effectively autonomous method that can increases productivity as well as respond quickly of changing items based on customer demand for an assembly line which using 6DOF Manipulator by combining CAD data and computer vision system. Firstly, The 3D CAD model of grasped object is projected onto six different 2D planes, then combining six views to form the final pointcloud. Secondly, a voting scheme is used to estimated the 3D pose of object which is obtained by a 3D camera. For tuning a precision of an estimation such as surface normal, angle and location of an object, Iterative closest point (ICP) algorithm is applied. Before doing experiments, the recognition algorithm is verified through the simulation program. Through implement experiments, the system proved that it is stable, good precision and applicable in production line where mass product is produced. Moreover, the developed system allows non-expert users with basic knowledge about CAD drawing and image processing can generate a pose of an object from CAD model and transmit data to a manipulator for the bin picking task.

Keywords CAD model · Bin picking · Image processing · Computer vision · Industrial robot

1 Introduction

In the recent years, traditional manufacturing systems are being replaced by intelligent manufacturing systems. Therefore, the participant of intelligent robot is indispensable. The integration between robot and vision systems has rapidly spread to satisfy consumer demands [1, 2]. Therefore, bin picking operations in assembly line using 6DOF manipulator and computer vision have been widely studied. However, quickly adapting to changes in products in the line is always a priority to ensure productivity. For the production line using manipulator and computer vision for picking object, workers need to have deep knowledge of 2D and 3D image processing for reprogramming the system to recognize the new grasped object. Currently there are some approaches for this issue from traditional 2D, 3D recognition to machine learning method.

- The first one is to use 2D image for matching to recognize 3D object. Ulrich et al. [3] presented 2D edge matching and pyramid voting method to calculate the pose of 3D object by searching for maximum similarity in a 2D model. Casado [4] proposed method to apply algebraic relations between the invariants of a 3D model and those of its 2D image under general projective projection for estimating object 3D poses. However, because of the occlusion, pose estimation uses 2D image matching only works when their poses are limited to a few cases especially in automatic random binpicking systems. camera Collet et al. [5] proposed a method to reconstruct 3D models based on a structure from motion algorithm. The object is determined by using the set of input images from many different angles. For each 2D image captured, the local feature points will be matched with the points on the prepared model. However, this method can not applied if the subject lacks of color which are very common in the mechanical industry.

✉ Le Duc Hanh
ldhanh@hcmut.edu.vn

[1] Mechanical Department, Ho Chi Minh City University of Technology, Vietnam National University, Ho Chi Minh City 70000, Vietnam

- The second one is to user laser scanner system. Ban et al. [6] and Xu et al. [7] proposed a laser vision system consists of using a projector and a fixed camera to project vertical and horizontal lines, thereby using the camera to determine the X, Y, Z coordinate and direction of objects to pick up. However, the system [6, 7] has limited speed because it requires waiting times for projector operating and also of process complex images of laser vision system. It needs a lot of time for re-programming and re-calibrating camera if the grapsed object is changed.
- The third one is to use simulation to learn the grasping pose then transmit to robot arm. Bousmalis [8] and Mahler [9] use the learning method to identify the object to be picked up, but the algorithm is limited because the object is randomly stacked so the objects can overlap each other. When the algorithm performs learning, it will also learn the black overlap region, so the amount of data learned is very large and when performing the pick-up, robot will have a lot of difficulties due to these black shadows data.
- Over past decade, deep machine learning has also been applied to object recognition such as Tuan et al. [10], Iglesias et al. [11], Benjamin et al.[12] Chun-Tse et al. [13], use large data collection of 2D image and 3D pointcloud data then applying deep neural network for training to identify and take objects in containers. Chen re-model the 3D object to be picked up, simulating and training the position and grip-ping position of the object to be picked up [14]. However, this technique of [10–14] need to retrain if using other objects and training data is very large.
- Nowadays CAD software is very powerful for generating the 3D pointcloud data and 3D model is often provided for the operation in industry, so there are also many methods combining the advantages of an image based system and the use of data from CAD for the recognition of 3D ob-jects and the estimation of coordinates. Jay et al. [15] has also developed the system using point clouds with color information for detecting the object. However, the method has the same problem of relying on color details to create characteristics to define objects, so it is difficult to apply the method to mechanical objects. To overcome the disadvantage of lack of color features. Aldoma et al. [16] proposed the method to calculate the pose of the object by using smoothing area growth algorithm. However, this approach depends much on the partition of each object, because each partition have its own characteristics.
- To obtain higher precision for industrial parts, Song et al. [17] built a real Bin Picking system using the method of using point pairs to initialize the feature. From pairs of similarities between the CAD model and actual pixels, a voting scheme is generated and the algorithm's result is the most voted posture. However, this method is heavily influenced by the noise caused by the depth sensor, because it directly affects the algorithm to find the normal vector, thereby deviating the characteristics of the pair of points. Recently, Li and Hashimoto [18] proposed the method for bin picking by using Voting and Hypothesis Verification by discrete spaces into a grid instead of searching for neighbors by Kd tree, the improvement has improved computation time to an average of 156 ms. However, instead of using 3D CAD data, the research uses directly the pointcloud obtained from the 3D camera.
- Recently, method of combining 2D images and 3D images for the binpicking application is also introduced [19]. The 3D pointcloud of the object is obtained then transformed to 2D image for object recognition. After that the image is transform back to 3D one for the rough pose estimation. The object then put in the sub position for direction and centroid estimation. The approach have high accuracy and stable. However, the method requires the user to be an expert in 2D and 3D image processing when dealing with object recognition, especially different objects in the industrial chain.
- There are also some existing commercial systems for such applications such as the 3D Area Sensor [20], the Pick-itTM Robotic Picking [21]. Although these systems being commercial and performing well in such applications, those kind of systems are still very expensive and very complicated. The vision module is not popular since it is sold with the robot system.
- Given these discussed shortcomings and challenge about time constraints, system complexity, and price constraints. This study has come up with a solution that is based on an image based method for 3D objects recognition using CAD files data. The system extracts the 3D pointcloud of the CAD model, then estimate the 3D pose of grasped object which obtained by a low cost 3D camera attached at the end-effector of Manipulator. The matching is developed based on algorithm of Li and Hashimoto [18]. However there is an important difference that this research uses CAD data instead of direct data obtained from a complex 3D camera system for the computation of 3D pose coordinates of object. The use of CAD data, instead of a 3D complex vision system, reduces the complexity for the non-expert operator thereby improving system productivity as well as quick response to changing grasped object in the industry line.

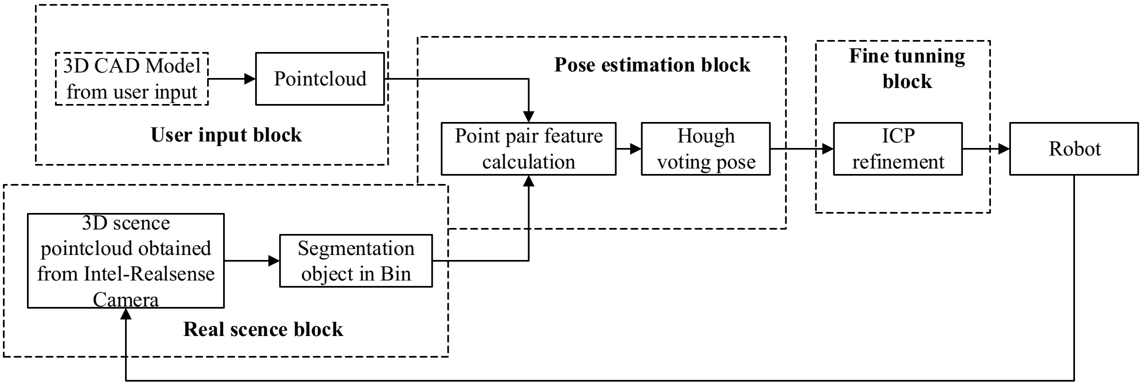
3D CAD Model from user input
Pointcloud
User input block
Pose estimation block
Point pair feature calculation
Hough voting pose
Fine tunning block
ICP refinement
Robot
3D scence pointcloud obtained from Intel-Realsense Camera
Segmentation object in Bin
Real scence block

Fig. 1 The overall pose estimation system

More over by using low cost 3D camera system, the cost of the system is also significantly reduced.

- The overall pose estimation system is illustrated as following Fig. 1. The system consists of four main blocks, user input block to upload 3D CAD model of grased object and convert it to 3D pointcloud, real scence block to obtain the pointcloud of the scence from RGB-D camera, pose estimation block to predict possible poses, fine tuning block to choose the best pose finally, the predicted pose will be transmited to the manipulator to perform the picking task.

2 3D model database generation

Most of the mechanical objects in industrial production lines have 3D CAD models in the form of sldprt, STEP, dwg. They are capable of opening by Softwares such as Solidworks, Inventor, … Therefore, the research proposes a method to use specialized software to convert it into Standard Triangle Language (STL) file format, the very common format in CAD software [22], then perform mesh sampling on the mesh surface of that file by using uniform sampling methodology which applied in [23–26].

As known, regular CAD models contain internal details, for example: Bearing will contain a model of the marbles inside the bearing, while the camera cannot see those spots from the outside. Our task is to get the model that have the highest possible similarity in order to increase percentage of successful of recognition algorithm. It is therefore necessary to identify points visible from the outside. Therefore, this research applied the method of Sagi Katz et al. [27], a method of detecting points visible from the camera perspective outside the object. The object is projected onto six different 2D planes, then combining six views to form the final pointcloud as show in Figs. 2 and 3

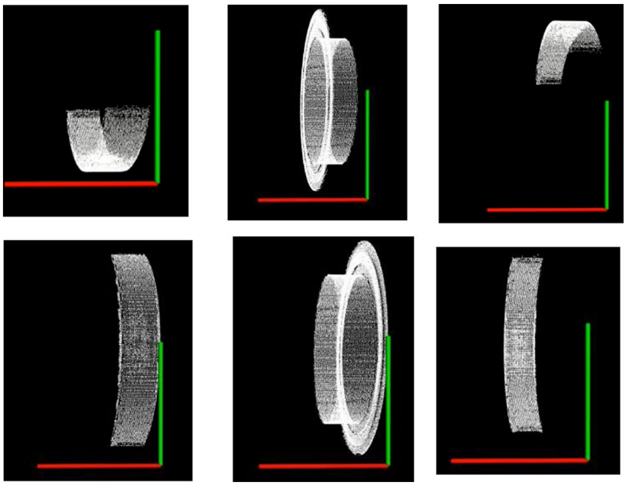

Fig. 2 Six projected 2D planes

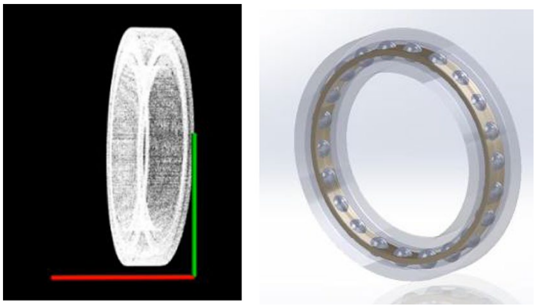

Fig. 3 The point clouds generated from 3D CAD data

3 Pose estimation

3.1 Normal vector calculation

The data read from Real-sense camera is in the form of a point cloud consisting coordinates (x, y, z), represented three dimensions in 3D space. The origin O (0.0f, 0.0f, 0.0f) is attached to the midpoint of the surface camera. The raw data read from the camera has a very high density, which takes a lot of computation time. To increase processing speed, the location of point cloud inside the registered range are kept by using pass-through filter algorithm. Then voxel grid filter is applied. Voxel mesh filter is a method of reducing the sample by bringing the point cloud into a space of 3D meshes, with the size of the mesh larger than the distance between a point and the closest point. In other words, the resolution of the point cloud should be greater than the resolution of the voxel grid. Then for each voxel, the algorithm calculates the mean center of the inner points and replaces these points with only one point at the mean center. Therefore, the larger the edge size of the voxel cell, the more points will be filtered and vice versa. Based on installation and experiment, a mesh with a size equal to 0.04 diameter of the model is selected.

In order to be able to identify objects, one of the important features of the point cloud is the normal vector at each point. Neighboring points are used to describe the space around a point, in other words to represent surfaces passing through that point. Normal vector is also one of the basic features in point clouds. Estimating the normal vector for each point is also equivalent to defining the normal vector of the plane.

To determine a plane equation in space, it is necessary to determine an x point in that plane and its normal vector $\vec{n}$. The point x is determined by focusing on nearby points of the point

$$x = \overline{p} = \frac{1}{k}\sum_{i=1}^{k} p_i \tag{1}$$

As discussed in [28, 29], the solution for $\vec{n}$ is given by finding the eigenvalues and eigenvectors of covariance matrix $C \in R^3$, with:

$$C = \frac{1}{k}\sum_{i=1}^{k} \xi_i (p_i - \overline{p})(p_i - \overline{p})^T \tag{2}$$

The eigenvectors $\vec{v}_j$ form an orthogonal frame, corresponding to the principal components of p_i^k

$$C\vec{v}_j = \lambda_j \vec{v}_j, \quad j \in \{0, 1, 2\} \tag{3}$$

If $0 < \lambda_0 < \lambda_1 < \lambda_2$ the eigenvector $\vec{v}_0$ corresponding to the smallest eigenvalue λ_0. Therefore the approximation of $\vec{n}$ is $\vec{n} = \{n_x, n_y, n_z\}$ or $-\vec{n}$. The calculation of the normal vector at the center will have errors, so to increase the reliability for the estimated vector, the maximum allowed error between the normal vector at the center and the normal vector at the adjacent point satisfy the following conditions:

$$\theta_{\overline{x}i} = ar\cos\left(\vec{n}_x \vec{n}_i\right) \tag{4}$$

4 Point pair feature calculation

In order to be able to identify the correlation between the 3D model object and the 3D pointcloud objects, pointcloud features are necessary, which is like finding similarities between two 2D shapes by using SIFT or SURF. Therefore, in 3D space, to recognize the similarities between two pointclouds, Drost et al. [30] proposed the PPF characteristic method which reduced computation time and increased recognition accuracy. This method instead of using only each point and its neighbors for the feature, it uses any pair of any 2 points m_1, m_2 in pointcloud with 2 normal vectors n_1, n_2 found, and distance $d = m_2$ -m_1 to calculate the characteristic vector. Then, discrete the values of the vector into multiples of 5 mm for the first element of the vector, and multiples of 120 for the remaining three elements according to [27]. Thanks to that, the storage and calculation are faster. The model's global feature is defined as the projection from the PPF feature vector to pairs of points of that feature in the model.

4.1 Hough voting

After preparing the model's global features, we proceed to compute the pointcloud features of the object capture. Suppose with any pair of points (s_i, s_j) ϵ S^2, we compute their charac-teristic vector PPF, and find a pair of similarities (m_i, m_j) ϵ M^2 having the same feature vector. This means that the point pair (m_i, m_j) of the object can be located at the corresponding (s_i, s_j) position on the captured pointcloud, and thus the object's center of gravity and direction can be determined. Thus, each pair of points (s_i, s_j) represents a probable pose of the object and will act as one vote for that pose. The simplest way is to identify all possible poses and choose the ones with the highest number of votes. However, this approach consumes a lot of resources because the storage and computation of many 4×4 displacement matrices is too large, so Bertram Drost et al. [29] proposed calculating the posture of the object with only pairs of numbers $(m_i$, α) with α is the variable rotate around x axis. For each pair of points (s_i, s_j) ϵ S^2 in the captured pointcloud, find pairs of numbers $(m_i$, α). Since this pair of numbers is enough to define a pose, we will perform a Voting Scheme similar to the General Hough Voting algorithm in 2D space, with 2 axes m_i and α.

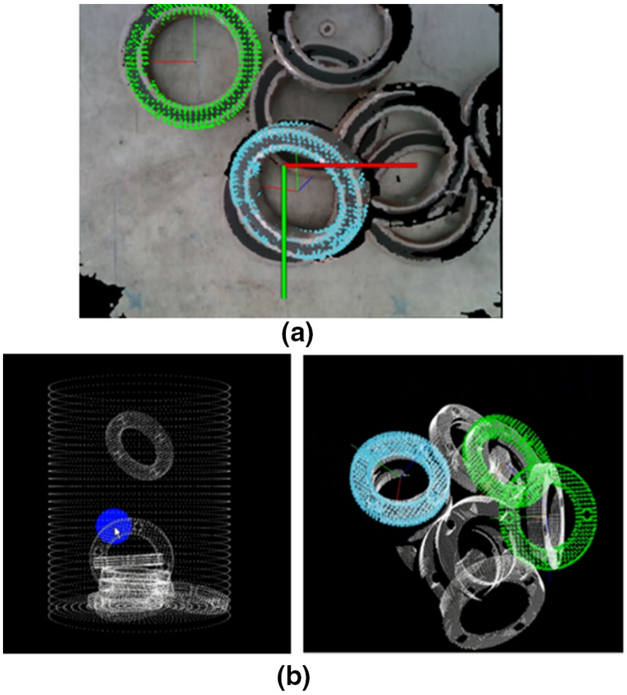

Fig. 4 The result of pose estimation using proposed algorithm: **a** real environment, **b** simulation environment using bullet physic SDK

4.2 Hypothesis verification & ICP refinement

After performing the Hough Voting Scheme in conjunction with the PPF feature, the number of poses found is numerous, and only a few of them contain the desired results. Therefore, to eliminate false poses, we perform a re-test of posture hy-potheses found by Hypothesis Verification.

First, do a model transition into the captured pointcloud. In-stead of using the Kd tree method to find the proximity of each point in the pointcloud model, it consumes a lot of re-sources. Here we use the discrete method. Specifically, before performing Hypothesis Verification, captured pointcloud will be divided into voxel grids with L voxel length.

The results selected by Hypothesis Verification will include poses that approximate the subject's true posture, but the accuracy is unacceptable. To solve that problem, ICP algorithm is implemented to improve object detection accuracy. However, doing ICP on the entire pointcloud model will yield misleading results, by aligning the invisible points at the opposite angle to the visible points of the captured pointcloud.

To solve this problem, the research will perform the calculation of the visible points of the pointcloud model first by the Direct Visibility of Point Sets method [30], then, implementing ICP algorithm until the allowable average deviation is 1 mm. To verify the ability of the recognition algorithm, the study used both the open source simulation bullet physic SDK [31] and the real images pointcloud taken from the realsense camera. In the simulation, the object is released freely in the middle of the bin and falls from above into the bin. By using physics simulation the objects in the bin have random posture and stack on top of each other. The results of simulation and real enviroment are obtained as shown in Fig. 4a and b.

5 Manipulator and camera calibration

5.1 Calibration

After identifying the pose of the object, the next step is to calculate the transformation matrix from the camera coordinate to the end effector coordinate of the robot. Because it is hard to determine the position of the camera coordinate that corresponds to the coordinate of the ma-nipulator, if using the method of measuring the distance from the manipulator to the camera will not give accurate results. Therefore, this research uses additional fixed 2D camera installed above the workding space to find the transformation matrix. The using of additional fixed 2D camera provides a higher accuracy than the commonly used method of moving the teach-pendant to each known point in camera coordinates because it eliminates the measurement error of the human eye.

The calibration process is summaried as following.

1. Preparing a plane, on which four markers are mounted at known distances to the center of the plane as shown in the Fig. 5a. This plane is attached at the end-effector of the manipulator. From this setup the transformation matrix $T^{6}_{mar\ker}$ between the marker and the end-effector is calculated.
2. Using 2D camera installed above workspace to read the position of the marker as shown in Fig. 5b then the transformation matrix $T^{mar\ker}_{camera2D}$ between the marker to the camera 2D is calculated.
3. Reading current angle of the joint from teach-pendant and apply the calculation as shown in Eq. (5) and Eq. (6)

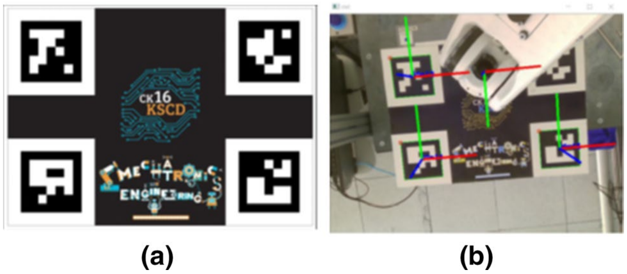

Fig. 5 **a** Aruco Marker plane; **b** Marker plane attached at end-effector

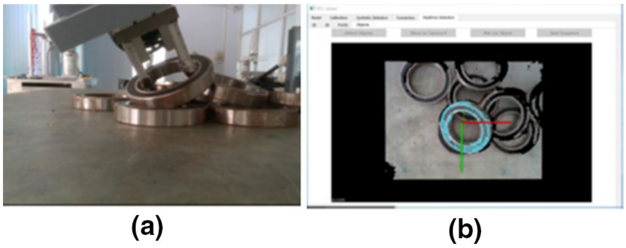

Fig. 6 **a** Picking pose position; **b** Calculating pose by image processing

to find the transformation matrix from robot coordinate to camera 2D:$T^{0}_{camera2D}$

$$T^0_6 = T^0_1 T^1_2 T^2_3 T^3_4 T^4_5 T^5_6 \tag{5}$$

$$T^0_{camera2D} = T^0_6 . T^6_{mar\,\ker} . T^{mar\,\ker}_{camera2D} \tag{6}$$

Then:

4. Placing markers inside the view angle of both the 2D and 3D camera and find the transformation matrix from the Real-sense camera to the 2D camera:

$$T^{camera2D}_{realsense} = \left(T^{mar\,\ker}_{camera2D}\right)^{-1} . T^{mar\,\ker}_{realsense} \tag{7}$$

5. Reading current angle of the joint from teach-pendant again and apply the calculation as shown in Eq. (8) to find the transformation matrix from realsense to end-effector:$T^6_{realsense}$

$$T^6_{realsense} = T^6_0 . T^0_{camera2D} . T^{camera2D}_{realsense} \tag{8}$$

5.2 Pose estimation verification

The procedure to verify the accuracy of the estimation system is perform as following step.

- *Step 1* Placing objects at different positions, at any angle.
- *Step 2* Manual moving the manipulator using teach-pendant to the picking pose position and record the three coordinates X, Y, Z and the three rotation angle Rx, Ry, Rz reading from Tech-pendant as shown in Fig. 6a
- *Step 3* Move the manipulator to any position where the recognized object is still visible in the view of camera and using the recognition algorithm discussed above. After calculating the location of needed object, moving the manipulator to the calculated position and record the co-ordinates and 3 rotation angles from Tech-pendant as shown in Fig. 6b. Then calculating the absolute error recorded data between step 2 and step 3. The meas-

Table 1 Absolute position error (mm)

Times	Manual (X,Y,Z)	Image processing (X,Y,Z)	Absolute error (X,Y,Z)
1	117.23; −422.94; 2.5	118.30; −423.14; 2.76	1.07; 0.20; 0.26
2	292.25; −403.05; 25.21	295.74; −403.13; 29.54	3.49; 0.08; 4.33
3	183.44; −462.63; 24.87	182.59; −462.35; 24.47	0.85; 0.28; 0.4
4	224.64; −399.13; 26.15	227.99; −401.77; 24	3.35; 2.04; 2.15
5	223.72; −413.30; 32.68	222.38; −413.77; 29.42	1.34; 0.47; 3.26
Average error			2.02; 0.61; 2.08

Table 2 Absolute angle error (degree)

Times	Manual (Roll, Pitch, Yaw)	Image processing (Roll, Pitch, Yaw)	Absolute error (Roll, Pitch, Yaw)
1	−89.99; 0.02; 180	−90.11; 0.22; −179.19	0.12; 0.2; 0.81
2	−70.62; 0.02; −149.41	−69.92; −0.4; −148.23	0.7; 0.42; 1.18
3	−36.5; 9.44; −174.12	−35.31; 10.68; −172.94	1.19; 1.24; 1.18
4	−11.74; 15.73; −177.57	−11.6; 15.72; −176.99	0.14; 0.01; 0.58
5	−68.6; 27; 160.54	−68.4; 26.85; −160.99	0.2; 0.15; 0.45
Average error			0.87; 0.4; 0.84

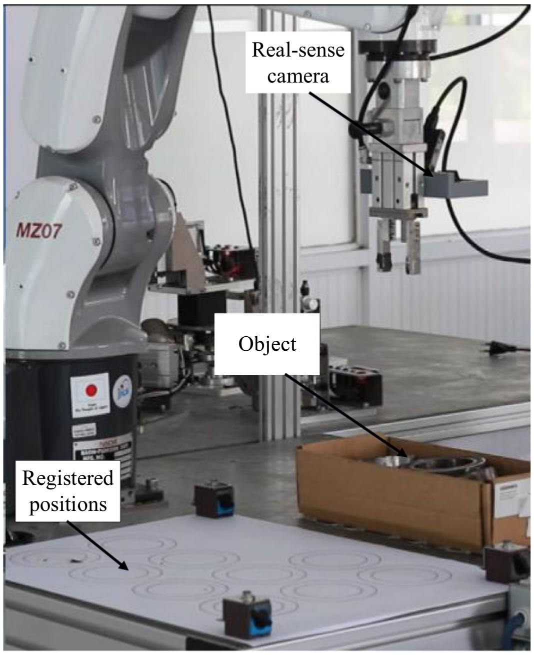

Fig. 7 Experimental apparatus

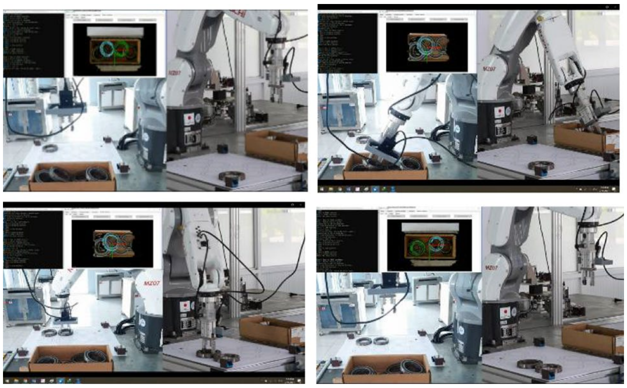

Fig. 8 Snapshots of the experiment

Table 3 The experimental result of successful rate

Times	1	2	3	4	5
Successness	10/10	9/10	10/10	10/10	9/10

ured results after five times are shown in the following Table 1.

It is seen from Table 1 and Table 2 that the error of pose estimation occurs due to the viewing angle and light refelection on the material, leading to get inaccurate 3D pointcloud convex. The bearing have the internal and external diameter are 70 mm and 100 mm respectively so from the international tolerance grades Table 6–30 [32] the absolute error

Table 4 Time distribution (mms)

Times	Pre-processing	PPF Voting	ICP	Moving robot	Total
1	387	2913	300	8522	12,122
2	319	3254	292	8476	12,341
3	427	3706	299	8658	13,090
4	358	3157	317	8457	12,289
Average	395	3342	300	8536	12,573

of position and angle sastify the IT12 standard when using for direct assembly.

6 Experiment

Working space for experiment includes one Real-sense camera attached at the end-effector of the Manipulator as shown in Fig. 7, the 6DOF Nachi Manipulator and pneu-matic suction cup with maximum 1 kg payload. Both of them are parallel fixed with table surface having rectangular box objects. Objects are aluminum type and randomly stacked inside the box. They will be picked up and put at registered position. The pose recognition algorithms were programmed using a PC with an Intel Core i5-2400 CPU (3.1 GHz) and 4 GB RAM. The open source Point Cloud Library was used for the system. Snapshots of the experiment are shown in Fig. 8.

As shown in Fig. 8, at the home position the re-alsense camera accquired the 3D pointcloud enviroment and calcualte the pose of the grasped object in Bin by using the algorithm proposed above. Then the manipula-tor approach the object and grasp it. Finally the manipulator put the object into the registered position then move back to the ready position. After conducting experiments with different arrangements of ten bearings and repeated it many times. The recognition rate and the speed of the system was determined, and their results are shown in Table 3 and Table 4.

From Table 3, It is seen that although, the pose estimation calculation have small error due to the intensity of light reflected from the viewing angle on the bearing discussed in Sect. 3, the percentage of succesfullness is still high. it is solved by using allowable average deviation threshold method of the detected object. If the threshold value is large then using the result matrix to move the manipulator to the position above the object that is considered to be easiest to pick up 30 cm, and execute the identification again.

From Table 4, it is seen that the time to move the robot takes up most of the time by 8.5 s. Time to process images and data transmission is small about 4 s. The total time of implementation is 12.5 s. To improve speed of system, it can increase speed of robot to 100%. But for safety reasons it should only be done at 50% speed. The average time for the experiments is different because the object is randomly placed in the bin, so the complexity of the image obtained from the 3D camera is different, so the execution time of the processes is different. However, the difference is not significant.

7 Conclusion

This paper presented the robotic random bin-picking system based on combining image based method and CAD files data. The 3D model database is generated from CAD model to the point cloud to allow the object pose to be estimated. Before doing the real experiment the simulation program is performed to verify the recognition system. A series of experiments shows that:

- The proposed system can pick up all randomly posed objects in the bin using a low cost 3D camera, Realsense camera. More over The recognition system can be replaced by another 3D camera. This proved that the system is not complicated and easy to replace or maintance.
- As known that, the accuracy of recognition is affected by the camera quality as well as the viewing angle but with the eye in hand structure, the camera can be controlled to get the best viewing angle for the recognition process, even if the recognition result is not achieved as desired, the system can still move to another location for recognition as discussed in the article. This proved that the system is more flexible than other designs which install fixed recogntion system above the bin.
- From the operation of the system it can be realized that, if the object to be picked up is changed as customer demand, the only factor to change is the 3D CAD model, all system components and procedure remain the same, which reduces the complexity for the operator, allows non-expert users with basic knowledge about CAD drawing and image processing can complete the task.
- From the Table 4, The total time of implementation is 12.5 s This speed is also applicable for assembly task in real industry

Acknowledgements We acknowledgment the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study

References

1. Hardin, W.: Vision enables freestyle bin-picking, Vision System Design (2007).
2. Golnabia, H., Asadpour, A.: Design and application of industrial machine vision systems. Robot. Comput. Integr. Manuf. **23**(6), 630–637 (2007)

3. Ulrich, M., Wiedemann, C., Steger, C.: CAD-based recognition of 3D objects in monocular images. In: Proc. IEEE International Conference on Robotics and Automation, pp. 1191–1198 (2009)
4. Casado, F., Lapido, Y.L., Losada, D. P., Alonso, A.S.: Pose estimation and object tracking using 2D image. In: 27th International Conference on Flexible Automation and Intelligent Manufacturing, 27–30 June, Modena, Italia, pp. 63–71 (2017)
5. Collet, A., Berenson, D., Srinivasa, S.S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: IEEE International Conference on Robotics and Automation, pp. 48–55 (2009)
6. Ban, K., Warashina, F., Kanno, I., Kumiya, H.: Industrial intelligent robot. FANUC Tech. Rev. **16**(2), 29–34 (2003)
7. Xu, J.: Rapid 3D surface profile measurement of industrial parts using two-level structured light pat-terns. Opt. Lasers. Eng **49**, 907–914 (2011)
8. Bousmalis, K., Irpan, A., Wohlhart Y. P., Bai, M., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K.S., Levine, V. Vanhoucke, Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Australia, pp. 4243–4250 (2018)
9. Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., Goldberg, K. Dex-net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In: Robotics: Science and Systems (2017)
10. Tuan-Tang, L., Chyi-Yeu, L.: Bin-picking for planar objects based on a deep learning network: a case study of USB packs. Sens. Actuators **9**, 3602–3618 (2019)
11. Iglesias, A.T., Pastor-López, I., Urquijo, B.S., García-Bringas, P.: Effective bin picking approach by combining deep learning and point cloud processing techniques. In: de la Cal E.A., Villar Flecha J.R., Quintián H., Corchado E. (eds) Hybrid Artificial Intelligent Systems. HAIS 2020. Lecture Notes in Computer Science, vol 12344. Springer, Cham
12. Benjamin, J., Tevon, W., Remi, G., Konrad, A.: Pose estimation and bin picking for deformable products. IFAC-PapersOnLine **52**(30), 361–366 (2019)
13. Chun-Tse, L., Cheng-Han, T., Jen-Yuan, C.: A CAD-free random bin picking system for fast changeover on multiple objects. In: ASME 2020 29th Conference on Information Storage and Processing Systems (2020)
14. Chen, J., Fujinami, T., Li, E.: Deep bin picking with reinforcement learning. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018, pp. 1–8
15. Wu, G., Bin, L., Qibing, Z., Huang, M., Guo, Y.: Using color and 3D geometry features to segment fruit point cloud and improve fruit recognition accuracy. Comput. Electron. Agric. **174**, 1–8 (2020)
16. Aldoma, A. et al. CAD-model recognition and 6DOF pose estimation using 3D cues. In: Proc. IEEE Int. Conf. Comput. Vis. pp. 585–592 (2011)
17. Song, K.T., Wu, C.H., Jiang, S.Y.: CAD-based pose estimation design for random bin picking using a RGB-D camera. J. Intell. Robot. Syst. **87**, 455–470 (2017)
18. Li, M., Hashimoto, K.: Fast and robust pose estimation algorithm for bin picking using point pair feature. In: 24th International Conference on Pattern Recognition (ICPR), pp. 1604–1609 (2018)
19. Hanh, L.D., Tu, H.B.: Computer vision for industrial robot in planar bin picking application. Adv. Sci. Technol. Eng. Syst. J. **5**(6), 1244–1249 (2020)
20. Robotic Bin Picking Made Possible with the New FANUC iRVision 3D Area Sensor, (2015), http://robot.fanucamerica.com/robot-applications/Bin-Picking-Robot-with-3D-Area-Sensor-Vision-System.aspx
21. Pick-itTM Robotic Picking, (2015), http://www.intermodalics.eu/pick-it-3d
22. Szilvśi-Nagy, M., Mátyási, G.: Analysis of STL files. Math. Comput. Model. **38**(7–9), 945–960 (2003)
23. Hallad, S.A., Banapurmath, N.R., Patil, A.Y., Hunashyal, A.M., Shettar, A.S.: Studies on the effect of multi-walled carbon nanotube–reinforced polymer-based nano-composites using finite element analysis software tool. Proc. Inst. Mech. Eng. Part N J. Nanomater. Nanoeng. Nanosyst. **230**(4), 200–212 (2016)
24. Vishalagoud, S.P., Farheen, B., Kurahatti, R.V., Arun, Y.P., Raju, G.U., Afzal, A., Manzoore, E., Soudagar, M., Ravinder Kumar, C., Ahamed, S.: A study of sound pressure level (SPL) inside the truck cabin for new acoustic materials: an experimental and FEA approach. Alex Eng J **60**(6), 5949–5976 (2021)
25. Dhaduti, S.C., Sarganachari, S.G., Patil, A.Y., Yunus, K.T.M.: Prediction of injection molding parameters for symmetric spur gear. J. Mol. Model. **26**(11), 302 (2020). https://doi.org/10.1007/s00894-020-04560-9
26. Hallad, S.A., Patil, A.Y., BanapurmathNR, H.A.M., Shettar, A.S., Ayachit, N.H.: Experimental and numerical validation on the utilization of polymer based nano-composites for structural applications using FEA software tool. Mater. Focus **6**(6), 685–690 (2017)
27. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. In: Proceeding ACM SIGGRAPH Conference on Computer Graphic, pp. 1–8 (2007)
28. Hanh, L.D., Duc, L.M.: Planar object recognition for bin picking application. In: Proceedings of 5th NAFOSTED Conference on Information and Computer Science (NICS), Ho Chi Minh, pp. 213–218 (2018)
29. Rusu, R.B.: Semantic 3D object maps for everyday manipulation in human living environments. Künstl Intell. **24**, 345–348 (2010)
30. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition, pp. 998–1005 (2010)
31. Bullet Physics SDK, Real-time collision detection and multi-physics simulation for VR, games, visual effects, robotics, machine learning etc, https://opensource.google/projects/bullet3
32. Kverneland, K. O.: The ISO System of Limits and Fits—Tolerances and Deviations. (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.