Robotic Conveyor Tracking with Dynamic Object Fetching for Industrial Automation

Ren C. Luo, Chun-Hao Liao

Abstract— **For industrial robotic applications, conveyor tracking is one of fundamental function in the robot manipulator. While the target is moving on the production line, the task becomes sophisticated problems. Thus, a distinct grasping method under visual control system is definitely one of the essential solutions. In this paper, we propose a tracking strategy on moving objects for a robot arm object fetching system combined with distinct recognition algorithm. In addition, the grasping pose of robot arm is corrected by visual feedback system. The system is separated into two parts and discussed in detail. Each part owns its core algorithm to complete industrial tracking and fetching tasks. Because of limitation from the environment, the working conditions will also be illustrated. Eye to hand and eye in hand both contribute to the visual feedback system. Grasping pose for each type of workpiece is adjusted by tracking and optimization algorithms. The result of object recognition is enhanced by visual system in determined pose and orientation. All experimental results were completed by a 7-DoFs robot arm developed in our lab at NTU.**

I. Introduction

For industrial robotic applications utilizing a robot arm, the most fundamental functionality required is that the arm should be able to move from the current pose to the target pose so that it picks the machined objects to place on the platform. When the target starts to change its movement on the production line, the task turns out to be more complicated without any feedback control system. Undoubtedly, a visual system helps to track the moving target and transmits new data to the robot arm. As a result, a new grasping technique under visual control system is definitely required.

The technique for grasping objects such as industrial products should be developed and teach robot manipulators on the assembly lines. First, machine tools manufacture various industrial products respectively. When those things are produced, the way to put on the conveyor is to grab them with proper command to robot arms. Actually, object gripping happens among different working processes. In [1] and [2], this is the common method to pick component from one place to the target position. This can easily be done by repeating mechanical commands. To accomplish more complicated tasks, target information is essential for robots. Sensors play an important role in the scheme of targets on the assembly [3]. Sensory data from vision sensors usually contain the position and orientation of objects on the conveyor. [4] and [5] provide a technique to tell robot where and when to grab workpieces under designation. By using a tilting laser range finder in [6] would be more straightforward to describe object situation relative to robot arms. They all provide environmental information to the robot to implement commands.

Ren C. Luo is with the Center for Intelligent Robotics and Autiomation Research (iCeiRA), National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106 (corresponding author to provide phone: +866-2-3366-9822; e-mail: renluo@ntu.edu.tw).

Chun-Hao Liao, is with iCeiRA, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106 (e-mail: jhliao@ntu.edu.tw).

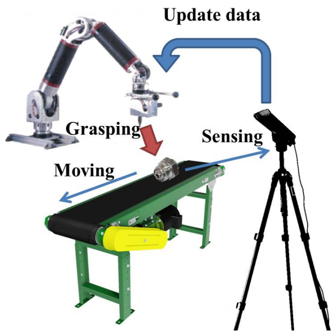


Fig. 1: An industrial task outlining the pickup of moving objects.

Visual camera is a common piece of equipment in the robotic picking tasks. The robot vision system is considered as assistant to finish work. The 2D data is derived from cameras and transformed into poses for robot arms [7]. Some image processing algorithms have high impact on object recognition [8]. Moreover, Kinect sensors are famous for its depth information, which assist robot end-effectors in grasping. [9] and [10] show the advantages of depth data to enhance entire process of pick and place. As a result, cameras are one of the key points under pick and place tasks.

However, there are a lot of unexpected conditions to consider in the assembly lines and these problems need to be solved. In this paper, the unexpected part is moving in Fig. 1. Sensor integration brings about a comprehensive conception on robot grasping. Equipped more sensors can not only deal with complex environment but also decreases the error rate of a task [11]. There are two ways to describe the relationship between arms and cameras. One is camera to hand and the other is eye-in-hand. Camera to hand is common to send background information to control center. Eye-in-hand usually transmits object state to computer to revise the position and orientation of targets immediately if problems happen in the environment. Moreover, the movement of robot manipulators is not always stable. To strengthen the function in perception system, the condition should be recorded so that all motion will remain more fluent next time.

The rest of the paper is organized as follows: In section II, we first review the overall system including important techniques in the research. This section covers trajectory

978-1-5386-0837-1/17/$31.00 ©2017 IEEE

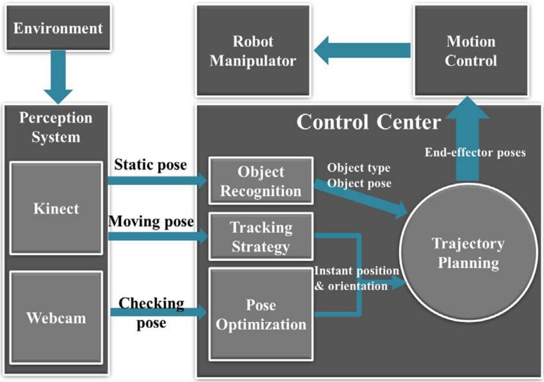


Fig. 2: Flow chart of whole system architecture.

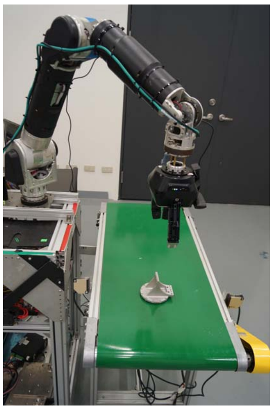


Fig. 3: 7-DoF robot manipulator.

planning, robot arm control and two methods which are the key components of this paper. The first technique is the tracking strategy of grippers described in section III. Section IV. explains the other important algorithm, optimized grasping, which increases the success rate of pick-and-place task. In section V, we carry out the experimental results and analyze these including discussions. Finally, we conclude in section VI.

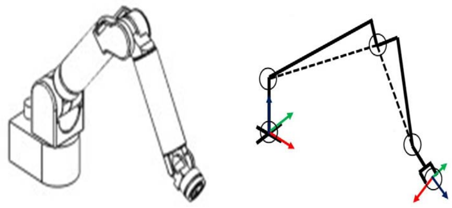
II. Methodology

The main objective in this research is to build up a sensor-integrated system that can grab moving objects with pose optimization in the assembly lines. The system architecture is shown in Fig. 2.

Poses of robot end-effector are decided by 3D sensor data. The poses will be separated into two conditions. First, we have to collect static poses for each workpiece from visual recognized results. This can be determined by Kinect sensor. The task would become tough after objects are moving on the conveyor. Information from perception system will update the state of objects to the control center. Therefore, robot arm is able to grasp moving objects successfully. The detail of our system is described in the following subsections.

A. Hardware Structure

In this paper, a delicate robot manipulator with 7-DoFs is shown in Fig. 3. This is a redundant manipulator. With one more DoF, the arm can achieve an arbitrary end-effector pose



(a) (b)

Fig. 4: (a) Original 7-DoF robot manipulator CAD model, (b) Simplified robot model from base to end-effector.

with infinite arm configurations. Fig. 4a shows the original 7-DoF arms CAD framework, and Fig. 4b displays the simplified model of manipulator. Since the kinematic equation is highly dependent on the manipulator structure, this robot is considered as typical 7-DoF redundant manipulator by applying spherical revolute spherical (S-R-S) model. The advantage of S-R-S structure is that it can act as human arm. Therefore, the S-R-S model is also called anthropomorphic arm. This special construction provides a common view to analyze the dynamic motion of manipulator.

B. Robot Arm Control

Given a point in the world coordinate, the angle of each joint is calculated based on inverse kinematics (IK) equation. In general case, there may be no analytic IK solution for configurations of each joint from a manipulator. The numerical method is introduced to solve the problem for general manipulators. The velocity of joint can be mapped to Cartesian space with Jacobian linearization method, as shown in Eq. 1. Pseudo-inverse is used to solve the joint velocity for the linear velocity in Cartesian space as shown in Eq. 2.

$$v = J(q)\dot{q} \tag{1}$$

$$\dot{q} = J'\dot{x}, J' = (J^T J)^{-1} J^T \tag{2}$$

The procedure of an IK solver with dynamic gain is shown in Fig. 5. Firstly, the manipulator pose is calculated with FK, and then the IK solver updates the joint angles with Pseudo-inverse calculation. These two steps are repeated till the end tip reaches to the target pose within an acceptable error. The dynamic gain makes the speed achieve its maximum and minimum rapidly if the initial value and changing rate are adjusted properly. During the iteration procedure, the error is influenced by angles of joint. Under the circumstance of inappropiate angles, the gain decrease dynamically, and vice versa.

C. Object Recognition

The information of workpieces, including type and position, will be obtained by using 3D vision sensors. 3D geometry data of the environment will be transformed into 3D PointCloud, including position (x, y, z) and (r, g, b). At certain distance, the dense depth map from Kinect sensor is trusty for object recognition.

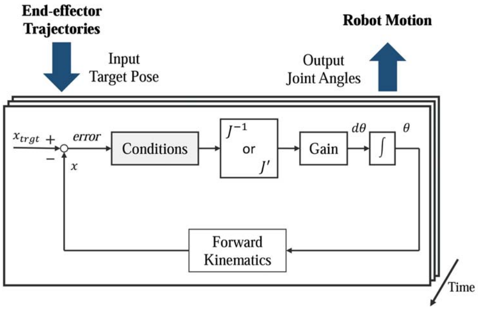


Fig. 5: The procedure of numerical method for solving IK.

The PointCloud is used to build up identification technique in industrial elements. Everything on the conveyor is given an identification to symbolize its static type and pose. Besides, the predefined CAD models of objects are associated with the recognition algorithm. With predefined CAD models, the 3D recognition algorithm [12] is suitable in our scenario. Another advantage is that the way to recognized is based on the geometry surface of industrial components rather than color. Therefore, the lighting of environment, which is an important issue on object recognition, is unnecessary.

Before start to distinguish the type of targets, the relationship between sensors and robots should be calculated to make sure the precision of grasping. Even if Kinect gives 3D sensing data to the computer, the frame data from spherical vision should be revised, especially non-homogeneous space in depth. The position of object in color image will be altered to actual position in the world coordinate. After completing the calibration, the accurate data information can be applied to robot manipulator. Hence, the command will be transmitted to robot trajectory planning to tell gripper how to go.

D. Trajectory Planning

The goal of trajectory planning is to generate smooth and stable motion in world coordinate based on a set of planned points. To show continuous motion of robot arm, there are some parameters will be considered including the maximum of velocity, acceleration and jerk. The movement in time continuous environment is distinct. And the first and second derivative of this function exists respectively. Jerky motion tends to cause vibration in the robot manipulator and cause increased wear on the mechanism.

After dealing with the prerequisite parameters, the basic path interpolation functions are conducted, including point-to-point movement and linear movement. The path segments will be generated by interpolation, which will be computed by prior parameters. In Cartesian space, a self-defined function plays an important role in path segment generation. The point-to-point movement can be followed by joint command and linear command. All current states of joints are marked in the joint space. Because the velocity is zero at target point, all joints will arrive at the same time before approaching to targets.

Contrast to point-to-point motion, linear movement is simple to illustrate the condition from reference coordinate at the end-effector. Current position relative to end point in the reference coordinate can be shown in Fig. 6. If the current velocity point toward target position is defined, the motion will only be associated with one DoF during variation. As a result, the robot arm will change moving state through path segments to perform grasping operation.

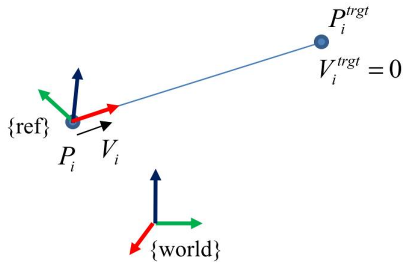


Fig. 6: Linear movement with reference coordinate.

With the help of object recognition algorithm and trajectory planning, it is executablefor manipulators to grasp a static workpiece.

III. Tracking Strategy

Picking operation has been implemented in the previous section. Now the scenario tends to be more difficult with some dynamic changes. When an object starts to move on the conveyor, the information of it should be updated to the robot to implement tracing skill. There are some challenges of tracking technique under visual control system. First, we need to track the objects by using the image processing in real time frame by frame. We then take the previous ten frames as its average reference. Secondly, we need to define the position of the gripper contrast to the moving target. We thought of one strategy about how the position of the objects can be obtained. The Kinect that also perform recognition can provide us information about not only the targets color, but also their depth data. Then, we generate a method to get 3D coordinate with the collected information and calibrate the scale to y axis and z axis with the consideration of depth frame that being x axis under Kinect view. Lastly, robot reaction will follow the renew data to finish grabbing movement. There are three subsections in our tracking strategy. After we obtain previous ten frames as the reference to compare with the current frame, we generate a promising perspective to improve the position of objects by calibrating the scale of y axis and z axis. After we get the position of the tracker and the target, then new motion is derived, and the tracker in the end would follow the target according to the calculated trajectory.

A. Image Segmentation

Because we want to display the tracking in real time, the algorithm of image segmentation must be simple and effective. After trying a lot of algorithms about tracking moving objects, we want to find one that can process the fastest in the time manner. Some algorithms take too long or lagging too much in updating frames, and that may go against at initial goal in this paper. If we think of a suitable algorithm, it is natural to continue on our next step in testing them, checking whether they match for us. In detecting objects: First, different intensity between previous and current frame is used to generate the contour of the object. However, the contour of objects is fragmented because the small size relative to the big conveyor is too small for detection, which may lead to some errors about finding the center of contour. In addition, the working area is limited by the vision of Kinect, we cannot speed up the conveyor to get the intensive difference data such that the threshold of finding the moving objects cant be high enough. Sometimes, the noise of Gaussian white noise would affect the contour. Hence, we could not use this way to find the objects. We choose the other way to generate the contour by recording the first ten frames as the average reference. The reason why we pick ten frames is that the Gaussian noise should be diminished as much as possible by get the average of the reference frames. This work can be shown by minimize the standard deviation with sampling. Then, we can get the difference between the reference and current frame, finding the contour. This segmentation result comes out to be very well. Both the end-effector and the shifting workpiece are tracked.

B. Position Localization

After the contours of objects are derived from the camera, we want to get the instant position of the tracker (gripper) and the target (workpiece) separately. First, the contour is the difference between reference and current frame from the color frame. Afterward, we get the coordinate of the objects in depth frame. The desired data include situation along y axis and z axis from Kinect color frame and x axis in the depth frame. Because of the different x-y planes that the tracker and target belong to, we can separate the objects by a threshold about the data along z axis in color frame in Kinect view. If the coordinate along z axis object is higher than threshold, then it is defined as the robot palm; otherwise, its the industrial component. The concept of multi-objects tracking is indicated.

C. Position Calibration

From Kinect view, we can get the information about depth along z axis. However, true position is altered due to quantization level among pixels. The scale from color frame to depth map should be measured before tracking. The length between each pixel is proportional to the vertical distance between the targets and Kinect. This relationship is also the same as the content mentioned in Object Recognition.

Once the tracking strategy is actuated, the static pose recognition will stop. The data include the position of the end-effector and object on the conveyor. The workpiece will move straight forward through the conveyor. Thus, only the pose of workpieces will be updated and orientation maintain present. Then, the control center will convey new command to the manipulator for grasping tasks.

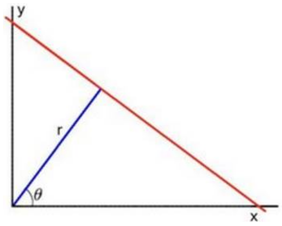


Fig. 7: Line description in polar coordinate.

IV. Optimized Grasping

It is worth notice that the end-effector may obey the next message derived from tracking strategy. However, this has little help in grabbing articles. To make sure the completeness of whole grasping movement, the gripper should take aim-off into consideration while it carry out object tracking. Therefore, the main point turns out to be accurate grasping pose through tracking process. When some problems occur on the conveyor, the arm may not know without extra indication.

Thanks to the technique of eye-in-hand, the additional information will be added under visual feedback system. A webcam camera is allocated for optimized grasping. When the robot receives instruction from control center to start tracking, the difference among frames will be recorded through the process. First of all, check the direction of gripper. Next, webcam records the frame while the gripper has been grabbing. Last, revision including grasping pose and orientation will be saved and will update the next pose and orientation if the system recognizes the same object again.

A. Moving Calibration

Every time the palm of robot arm start to execute moving task, the trajectory follow the commands derived from trajectory planning and eye-in-hand vision. Under the view of webcam, the edge of conveyor will emerge on the upper side of image. Because the moving path is linear on the conveyor, following the slope of the edge of conveyor is the way to correct the motion of robot.

To search a graph in an image, there is lots of image processing to select. One simple method is Hough transform which is well known as object detection. Under the background, the object is just a set of colorful point in camera view. Those point set will be mapped to a certain point or a high dimension plane. This can gain a parametric equation to represent all possible condition to describe the characteristic of pattern. Furthermore, search the extreme value to find out the position of the pattern. Last, collect all similar points together to illustrate the feature which is part of workpiece.

To find out the line edge of conveyor, line equation can be shown in polar coordinate in Fig. 7. The advantage of polar parametric form is no singular condition under sinusoidal function domain.

$$r = xcos\theta + ysin\theta \qquad (3)$$

Actually this condition simply contains a set of all possible solution of r, where the maximum count happens to.

However, it takes lots of time in calculation. To decrease time consumption, the generalized of Hough transform helps to develop a structure to compute the candidate reference points. The implementation for general case can be written as:

$$x_t = (x_0 cos\theta - y_0 sin\theta)s \quad (4)$$

$$y_t = (x_0 sin\theta - y_0 cos\theta)s \quad (5)$$

where x_0, y_0 is initial point, is rotation angle and s is uniform scaling value. To imply the reference point, the equation turns out to be:

$$x_c = x - (x_0 cos\theta - y_0 sin\theta)s \quad (6)$$

$$y_c = y - (x_0 sin\theta - y_0 cos\theta)s \quad (7)$$

The generalized method didnt mean to create a common way to find any kind of shape but to record all edges into a table corresponding to a reference point. After the slope of edge is gained, the orientation of gripper will gradually change through moving process.

B. Pose Optimization

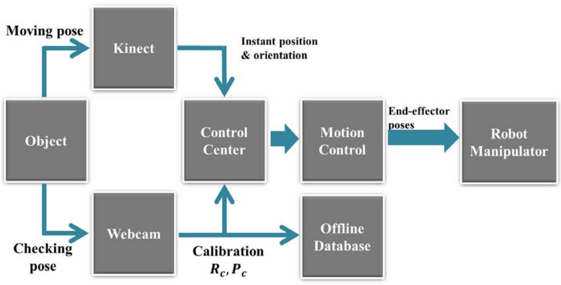


Fig. 8: System structure for optimized grasping.

To check out whether the task is successful or not, the webcam records image while the gripper is holding the object. The conceptual structure is shown in Fig. 8. To find out any variation of grasping pose, webcam will first store lots of correct pose picture info offline database. After gathering the poses as a standard level, the mission starts to grasp moving target. In this step, the grasping pose may update its position and orientation based on calibration in previous subsection and optimized pose.

First, an aim-off happens because there exists relative motion between moving target and end-effector. Thus, an aim-off is defined as:

$$\Delta P = V_c \Delta t \quad (8)$$

where ΔP is aim-off of gripper to the moving target, V_c is the velocity of conveyor and $Deltat$ is sampling time, nearly 0.5 sec. Then, the next state of position can be calculated as:

$$P_{t+1} = R_c R_t P_t + \Delta P \quad (9)$$

where R_c is the rotation matrix of calibration, R_t and P_t is previous state of rotation matrix and pose. The actual command to robot will be a 44 transformation matrix. The general expression is

$$T_{t+1} = \begin{bmatrix} R_{t+1} & P_{t+1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_c R_t & P_t \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} I_{3x3} & \Delta P \\ 0 & 1 \end{bmatrix} \quad (10)$$

In addition, the revision will be recorded into offline database for the same type and pose next time. To clarify the state of position, the offline part collects the average value from time zero to time t. The equation shows as following:

$$P_n = \frac{1}{t}(\sum_{x=1}^{t} R_{c\_avg} R_x P_x) \quad (11)$$

where R_{c_avg} means the average value of calibration rotation matrix, which first is regarded as identity matrix. After combing new grasping pose with the updated grasping modification, robot manipulator will work better in the next task.

V. Experimental Results and Discussion

There are two steps to carry out in the complicated scenario. The environment is built in Fig. 10. Two vision sensors have huge effect on the performance of conveyor tracking. The first part is object recognition in Kinect sensing. When the type and pose of object is obtained, Kinect change the view to tracking strategy. Rather than static object detection by recognition algorithm, the moving workpiece puts much more unexpected factors during grasping process. The first skill is tracking moving target on the conveyor shown in Fig. 9. When a static workpiece finishes recognition and the result is correct, the conveyor starts to transport the object. By means of Kinect, the instant pose and orientation of component will be captured to let the gripper know where the target is right now. Although it may takes time to update those desired information, the end-effector follow commands from the control center and catch the target before it falls down.

The other technique is grasping optimization shown in Fig. 11. First, the offline database is obligated to gather the correct pose of grabbing. Every correct grasping in webcam view will be stored in the offline database. Before capturing the target, a picture will be taken to check out the edge of conveyor is altered or not during tracking process. If the orientation changes, it adjusts the gripper to rotate exact direction.

Later, end-effector is seizing the object. Correct poses will be recorded to the database. When the task begins, those incorrect photo will also got to expand the feasibility of eye-in-hand system.

After two experiments are implemented, it is obvious that the complex part in this research is useful. Moreover, time consumption in the recognition and tracking is not bad. Based on the visual feedback system, the image data have been defined as necessary modification, which assists robot manipulator to analyze the current state of object.

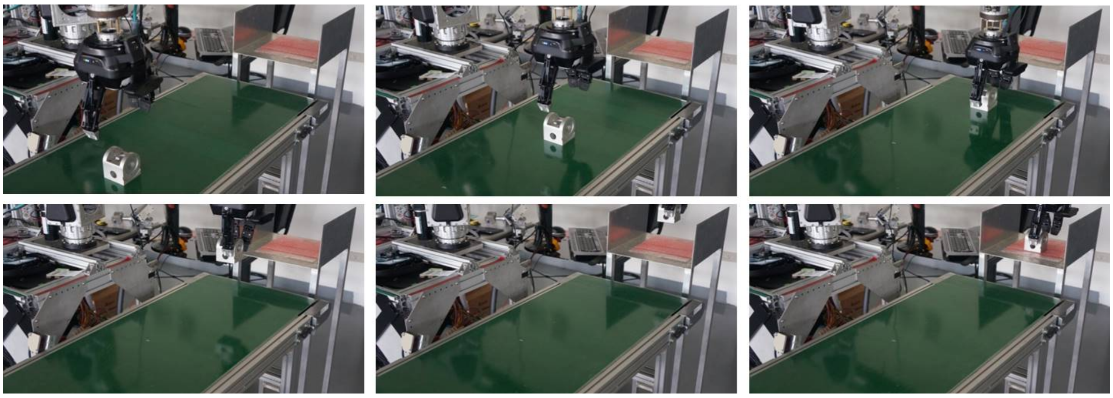


Fig. 9: The process of moving object tracking and fetching.

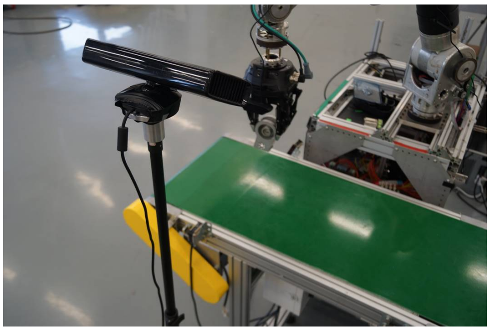


Fig. 10: Experimental setup.

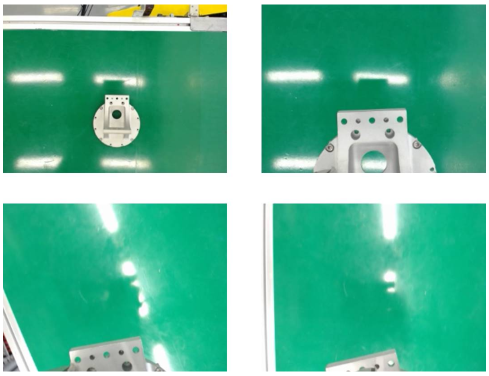


Fig. 11: The edge detection of conveyor. Correct grasping in top row and incorrect grasping in bottom row.

VI. Conclusion

This paper presents a strategy to provide dynamic information to robot arm under sophisticated environment. It seems that the results are well demonstrated. However, in grasping optimization, it may face some restriction under the size and type of workpieces. To revise the pose and orientation, some component of object should go inside the webcam view so that the new pose will be sent. Besides, the integration of vision sensors faces the consumption on programming, which spends much time updating image data from Kinect tracking and webcam fetching. In view of this, the structure can be developed by more precise resolution but low time consuming vision sensors. Moreover, further improvement in grasping issue is also integrated with various sensors in the environment. This may be a functional application in packaging in the production line.

References

[1] K. Berntorp, K. Arzen, and A. Robertsson, "Mobile Manipulation with a Kinematically Redundant Manipulator for a Pick-and-Place Scenario," in *IEEE International Conference on Control Applications (CCA)*, pp. 1596-1602, 2012.

[2] K. Yamazaki, M. Tomono, T. Tsubouchi and S. Yuta, "A Grasp Planning for Picking up an Unknown Object for a Mobile Manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2143-2149, 2006.

[3] X. Cheng, "On-line Collision-free Path Planning for Service and Assembly Tasks by a Two-Arm Robot ," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1523-1528, 1995.

[4] R. B. Rusu, I. A. Sucan, B. Gerkey, S. Chitta, M. Beetz and L. E. Kavraki, "Real-time Perception-Guided Motion Planning for a Personal Robot," in *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4245-4252, 2009.

[5] H. K. Chu, J. K. Mills, and W. L. Cleghorn , "Fabrication of a Microcoil Through Parallel Microassembly," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 50505055, 2012.

[6] Y. Ohshima, Y. Kobayashi, T. Kaneko, A. Yamashita and H. Asama, "Meal Support System with Spoon Using Laser Range Finder and Manipulator," in *IEEE Workshop on Robot Vision (WORV)*, pp. 82-87, 2013.

[7] P. K. Allen, A. Timcenko, B. Yoshimi and P. Michelman, "Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System," in *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 152164, Apr. 1993.

[8] R. B. Rusu, G. Bradski, R. Thibaux and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," in *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2155-2162, 2010.

[9] W. Wohlkinger and M. Vincze, "Ensemble of Shape Functions for 3D Object Classication," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2987-2992, 2011.

[10] W. Xing and Y. Ou, "On the Study of Workpiece Localization with a Cost-Effective Vision Based Method ," in *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER)*, pp. 457-461, 2015.

[11] Y. Suzuki, K. Koyama, A. Ming and M. Shimojo, "Grasping Strategy for Moving Object using Net-Structure Proximity Sensor and Vision Sensor," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1403-1409, 2015.

[12] R. C. Luo and C. W. Kuo, "A Scalable Modular Architecture of 3D Object Acquisition for Manufacturing Automation," in *IEEE 13th International Conference of Industrial Informatics (INDIN)*, pp. 269-274, 2015.