3D Visual-Guided Robot Arm Control for a Warehouse Automation System

Minh-Tri Le, Soonmyun Jang, and Jenn-Jier James Lien

***Abstract*— Warehouse automation is greatly beneficial in improving a wide variety of industries. However, the prevalent automation methods apply in industrial fields where systems are difficult to initialize and it is hard to recognize the system status. In this work, a 3D visual-guided robot arm system with marker detection and object detection is proposed. There are two main parts in this study, including system initialization and validation using marker detection and storage and retrieval using magazine detection. The system is composed of two cameras for the stereo system, a robot arm, and computer vision algorithms to form the system for detecting, classifying, and picking objects by a robot arm. Besides, magazines that can store items such as nuts and bolts and a frame that can store magazines into its grids are used. Firstly, the system is initialized by marker detection method which detects marker positions on a frame and saves frame and grid positions where the robot arm can approach to store or retrieve magazines. After that, using contour detection of deep learning method and Hough line transform, the correct magazine center position in a grid can be estimated. If an impact occurs such as an earthquake, the warehouse system must check the status to see if the system can be run perfectly. This study introduces solutions which avoid the above problem. The work also shows an error under 1mm between magazine position and grid position.**

Index Terms*— Robot arm, storing and retrieving, marker detection, contour segmentation, deep learning

I. Introduction

It is widely acknowledged that automation technologies have played a crucial role in industrial fields. A traditional warehouse automation system [1] mainly replaces manpower, reduces working space, and improves working accuracy using precise robot arms.

With the rapid development of the automation system, the robot arm and the machine vision have been applied to a variety of systems. And many kinds of machine visions are used according to purposes, such as a single camera, 3D stereo camera, laser sensors, etc. However, 3D stereo cameras are more cost-effective than other 3D solutions, so this work uses a 3D stereo camera. The cameras can be installed in two ways: the cameras are rigidly mounted on the robot arm, called Eye-In-Hand, or they are installed in a fixed position away from the robot arm, called Eye-To-Hand. In this work, the Eye-In-Hand solution is applied because cameras on a robot arm move together along a rail of the system to detect magazines and grids in frames. For the robot arm control with cameras, camera calibration and hand-eye robot calibration are necessary. Accurate calibration algorithms are especially important for target positioning, and they are affected by diverse factors that should be considered, such as hardware specification, working distance, and quality of calibration board.

This work aims to implement a system that combines a robot arm and 3D stereo cameras for a warehouse automation system. There are 3 main approaches in this work: Firstly, frame and grid position detection by the robot arm with the stereo camera. Secondly, object detection using a deep learning method and image processing. Lastly, storing and retrieving objects (magazine) by robot arm.

The rest of this paper is organized as follows. In Section II, related work is discussed. We present our overview hardware in Section III. Section IV shows the warehouse automation system using 3D visual-guided robot arm control. Then, in Section V we present experimental results. Lastly, we conclude and discuss future work in Section VI.

II. Related Work

For the warehouse system, Schwarz et al. [2] proposed a combination between object detection, segmentation, and registration method to design a robotic system for the Amazon Picking Challenge 2016 tasks. They achieved a good result, obtaining second and third place in that competition. They also used a stereo camera system for visual-guided robotics grasping. Prakash et al. [3] designed a dynamic robot manipulator that can be applied to a warehouse automation system. They proposed an optimal controller by using a dual-loop control scheme with outer and inner loops. In which, the former used a kinematic loop to assign a joint velocity reference signal to the latter. The Hamilton-Jacobi-Bellman equation was used in the kinematic as a closed-form analytic solution. While the inner loop used a neural network for the tracking control scheme. As a result, the system obtained effective and productive implementation in real-time for robotics picking. Moreover, warehouse automation systems using visual and intelligent approaches have been considered. Team NAIST-Panasonic [4] used an array of RGB-D cameras combined with a custom-made end effector for their robotic manipulator. They used YOLO-v2 [5-6] for object detection. While He et al. [7] proposed a novel policy, namely, Differentiated Probability Queuing on Automated Guided Vehicles for smart warehouse automation systems. In addition, for object contour detections, diverse methods have evolved from the traditional way to the deep-learning-based way. In computer vision, Sobel and Canny contour detection are the

This paper was first submitted in November 29, 2020.

Minh-Tri Le is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, (email: lmtkdt@gmail.com).

Soonmyun Jang is currently a technical engineer with the Contrel technology Co., Ltd, Tainan, Taiwan, (email: jsm890803.3@gmail.com)

Jenn-Jier James Lien is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, (email: jjlien@csie.ncku.edu.tw)

Fig. 1. The warehouse automation system. (1) Robot arm, (2) stereo camera *c*, (3) frame *f* (store magazines), and (4) magazine.

most popular methods. On the other hand, there are many deep learning solutions for contour or edge detection currently. One famous method is the holistically nested edge detection (HED) [8] which predicted contours in an image by a deep learning model. Yu et al. [9] improved the HED method to detect object contours and classify objects with fused activation maps. Furthermore, Hu et al. [10] introduced dynamic feature fusion for edge detection with a trainable adaptive weight fusion module which can solve multi-scale response problems in deep category-aware semantic edge detection (CASENet) [9].

The first contribution of this work is to initialize and retool the warehouse automation system automatically. To set the system, initialization must be done at the beginning of any process, such as setting the initial position of the robot arm, tools, and frames (tray) for magazines. These processes can be done manually in typical automation systems. Besides, it should be retooled manually when external influences occur such as an earthquake. To reduce the time for this process, the frame position detection method is used in this work, which can detect frame coordinates by robot arm and 3D camera. If there is an external shock to the frame or robot arm, the system checks the status and retools automatically.

The second contribution is to detect a center line of an object after a contour segmentation using a deep learning method. This work refers to [10], in which a dynamic feature fusion (DFF) network is used for semantic edge detection. However, the deep learning model cannot detect object position, it only does the segmentation and classification, so that post-processing of the output segmentation image is needed. Hough Transform method [11] is applied to the object lines and its center line.

III. The Overview of Hardware and Functions in The Warehouse Automation System

The system consists of a 6-DOF Epson ProSix C4-A601S robot arm, a stereo camera with two FLIR BFS-PGE-50S5C cameras, frames for storing magazines, and magazines (see in Fig.1). Frame is defined as ***f*** and includes Grid ***g*** and Marker ***mk***. One frame means the shelf structure within four markers, with a size of *30mm x 30mm*, of the entire storage. Grid is the partitions of the frame and it can be called cell. Marker is the ArUco

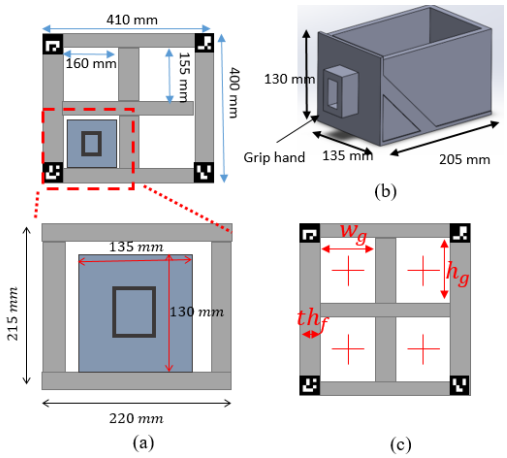

Fig. 2. Shapes of storage frame and magazine. (a) shows the demotration of shape of a frame with 2x2 grids; (b) shows a magazine with its size; and (c) shows definition of frame: w_g and h_g are width and height of a grid, respectively, and th_f is thickness of a frame.

marker of the OpenCV library for pose estimation. Markers have marker coordinates based on the top-left of the frame. The shapes of the storage frame and magazine are shown in Fig.2.

The principle of the system is positioning the robot arm and allowable tolerance at a storage grid in the frame structure to retrieve a magazine. The position of magazines stored in the system must be set to move the robot arm to a target storage grid. To control the system, the robot, frames, and conveyor belt positions must be saved in advance. Then, the system can be run in two stages: initialization process, and storing and retrieving magazines processes. Fig.3 shows the flowchart of these two stages. Fundamentally, the computer commands and controls the robot and the cameras for every step. As in Fig.3a, the robot moves to markers on the frame and the cameras detects marker 3D positions. Then, the computer computes each grid's coordinate using saved markers' positions. This scenario is to initialize the frame and grid coordinates to control the automation system. Fig.3b shows the procedure of storing and retrieving magazines using the robot arm and cameras.

IV. The Warehouse Automation System Using 3D Visual-Guided Robot Arm Control

The overview of the system is shown in Fig.4. There are 2 main parts: 3D Transformation Estimation between the Robot Base and the Grid Centers in the Frame; and 2D Center Position Alignment between a Magazine and a Grid using DFF-Net [10] which is a deep learning solution for contour detection. Firstly, we find 3D frame coordinates and each grid position based on the robot arm. Four ArUco Markers *mk (n=0, 1, m=0, 1)* [12] are captured at four corners on a frame by the cameras on the robot arm (see in Fig.4a). With collected marker images $I_{mk_{nm}}$ *(2448 x 2048, RGB)*, marker IDs and 3D position ${}^{c}T_{mk_{nm}}(R,t)$ based on the cameras can be obtained. However, due to inaccurate depth value of marker position detection, it is replaced by accurate depth value from a stereo camera. After that, according to the frame specification, each 3D grid position based on the frame coordinate $p_{g_{nm-ij}}$ is calculated manually. The output ${}^{B}T_{g_{ij}}$ of this procedure is saved to the system and used for the next step. Secondly, 3D magazine positions in a frame can be found using a deep learning method. And the status

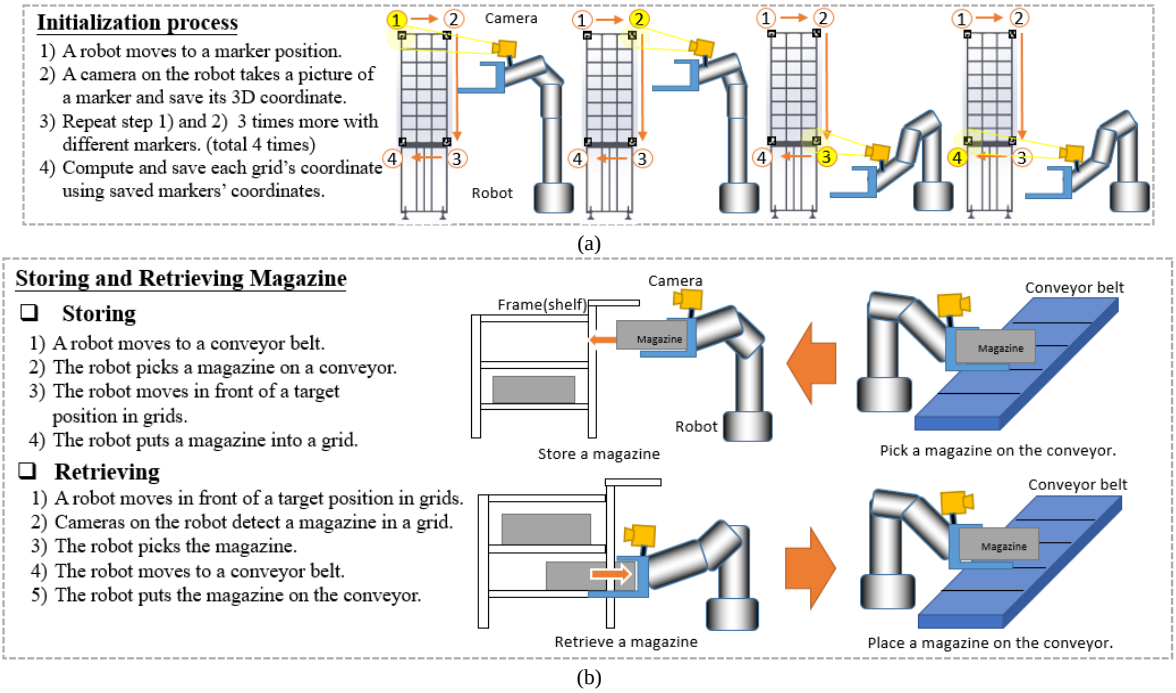

Fig. 3. The system scenario. (a) shows the initialization process, (b) shows the storing and retrieving magazine process

of the system can be checked whether magazines are at the center of each grid or not. First, the robot moves in front of a selected grid position and the camera captures an image. Second, the captured image I_g is put into DFF Net [10] which is a deep learning model for contour segmentation and classification (see in Fig.4e). There are two classes, magazine, and grid, in the system. After post-processing of two contour images, each center line will be detected and compared. If the magazine error E_{mg} between a magazine center point $P_{mg_{cnt}}$ and a grid center point $P_{g_{cnt}}$ is bigger than the requirement (less than 1mm), the system shows a warning alarm. If there is no error, the system completes the next procedure, which is magazine storing or retrieving.

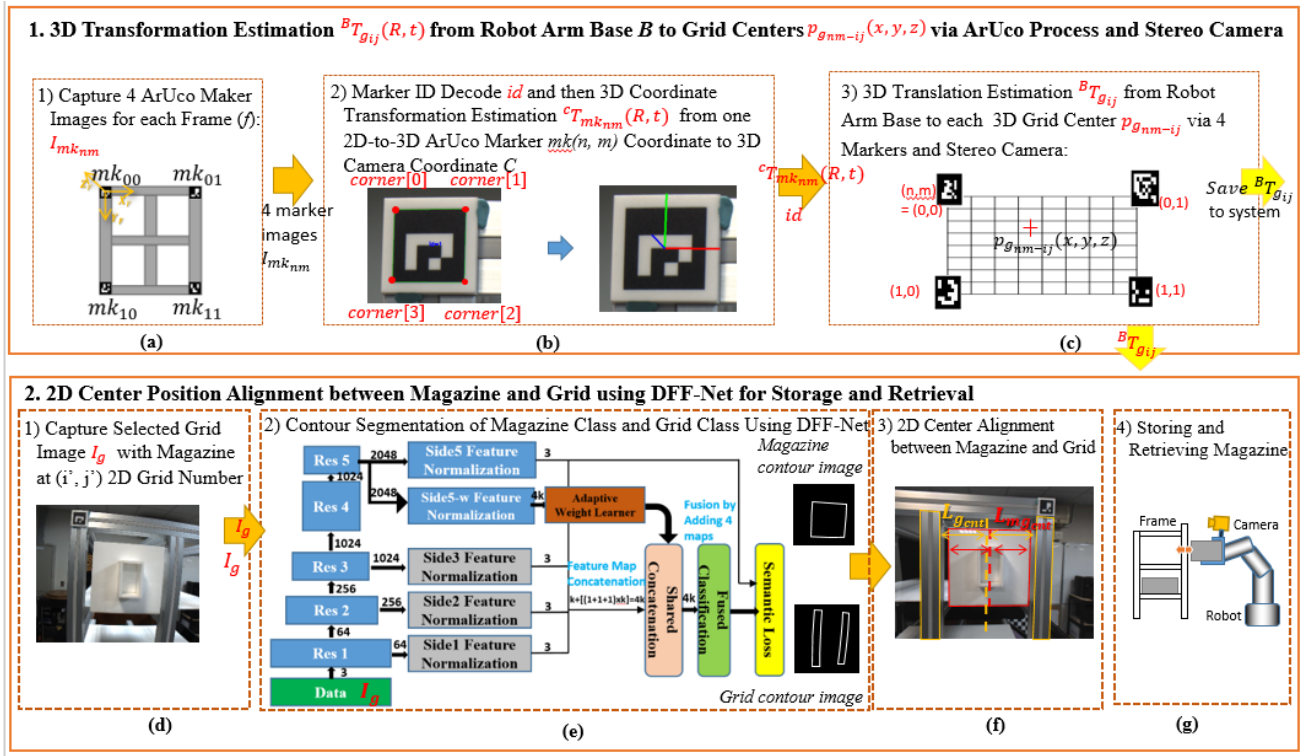

Fig. 4. The global framework of 3D visual-guided robot arm control for warehouse automation system. (a) shows markers capture, (b) 3D coordinate transformation estimation, (c) shows 3D translation estimation, (d) shows grid capture, (e) shows contour segmentation, (f) 2D center alignment, and (g) shows storing and retrieving magazines process.

A. 3D Transformation Estimation ${}^{B}T_{g_{ij}}(R,t)$ from Robot Arm Base B to Grid Centers $p_{g_{nm-ij}}(x, y, z)$ via ArUco Process and Stereo Camera

The main purpose of this section is to find 3D positions of a frame and grid centers based on the robot arm. There are three parts in this process as shown in Fig.4 (a, b, c). Firstly, the robot moves in front of markers on the frame, and the cameras capture marker images, and then markers' ids and three-dimensional positions are detected by ArUco functions. Finally, the transformations between the robot arm base and the grid centers are computed and saved to the system.

1) ArUco Marker Images Acquisition

At the beginning of the process, ArUco marker images $I_{mk_{nm}}$ must be acquired by the cameras. In order to do so, the robot arm position ${}^{B}t_{mk_{nm}}$ in front of the markers need to be saved by users in advance. Along with the saved positions, the robot arm moves to a marker and the cameras on the robot take a picture of the marker. This process is repeated 4 times until 4 maker images are acquired. The sequence of the robot arm trajectories is top-left, top-right, bottom-right, and bottom-left of marker positions on a frame.

2) 3D Transformation Estimation from Markers to 3D Camera and Marker ID Decoding

With collected marker images $I_{mk_{nm}}$, three-dimensional marker positions and marker ids can be detected by the ArUco library. We used Opencv library [13] for the process of marker detection and marker position estimation. The first function *detectMarkers* of the library is marker detection that can decode the marker pattern and get its id and detect 4 corners of it. Using marker corners data corners, the marker's 3D position from the camera can be obtained through the function *estimatePoseSingleMarkers* of the library. Outputs from the function are a 3x1 rotation vector and a 3x1 translation vector ${}^{c}T_{mk_{nm}}$ including a 3x1 translation, and then they are converted a to 4x4 transformation form for matrix operation as shown in (1)

$$ {}^{c}T_{mk_{nm}} = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{1} $$

where, $R_{3\times3} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$, $t_{3\times1} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}$: are rotation matrix and translation matrix

Furthermore, due to the inaccuracy of t_Z value in the translation vector in ${}^{c}T_{mk_{nm}}$ from a single camera, it should be refined by depth information of the stereo camera.

3) 3D Translation Estimation from Robot Arm Base to each Grid Center via Four Markers and Camera

The computed markers' three-dimensional transformations from the cameras are transformed to the 3D transformation of the markers based on the robot arm base, which is presented in (2). Note that transformation between the robot arm and the camera ${}^{B}T_{c}$ is calculated from approaches [14-17].

$$ {}^{B}T_{mk_{nm}} = {}^{B}T_{c}\,{}^{c}T_{mk_{nm}} \tag{2} $$

In the next step, each 3D grid position based on both the frame and the robot base is found by computed 3D transformation values of markers. First, according to the obtained 4 markers positions and the frame size, each grid position can be calculated. With a frame size and a grid size, a center of a grid position from the frame coordinate can be calculated by (3).

$$ \mathrm{p}_{\mathrm{g_{nm-ij}}}(\mathrm{x},\,\mathrm{y},\mathrm{z}) \begin{cases} x = \dfrac{th_f}{2} + \dfrac{w_g}{2} + (th_f + w_g) \times j \\ y = \dfrac{th_f}{2} + \dfrac{h_g}{2} + \left(th_f + h_g\right) \times i \\ z = 0 \end{cases} \tag{3} $$

In (3), a value of *nm* which means row and column of marker position on the frame is 00 because the top-left marker position is the frame coordinate. And *i* and *j* indicate row and column of grid center position. While w_g is width of a grid, h_g is height of a grid and th_f is thickness of a frame (see Fig.2).

B. 2D Center Position Alignment between Magazine and Grid using DFF-Net for Storage and Retrieval

1) DFF-Net [10]: Training and Inference Frameworks

Manually labeled dataset with magazine and grid classes based on the Cityscapes dataset is used. The input image is RGB and its size is 2448 x 2048 pixels. The labels of the dataset are contours segmentation of objects and their image format is grayscale. Basically, this architecture is composed of two main parts: feature extractor with the pre-trained ResNet and adaptive weight fusion module (see Fig.4e). The feature extractor blocks are connected to the residual blocks in the ResNet. In the feature extractor, single maps including object contours of all classes are extracted from Side1, Side2, and Side3 feature normalizations while K channel maps are extracted from Side5 feature normalization. Each map from the Side5 layer has contours of one class for classification so that the number of maps is the number of classes K. On the other hand, the number of maps from Side5-w feature normalization is 4K and they proceed to generate adaptive weights for concatenation of the maps from Side1-3 and Side5. And then the concatenated maps are fused to classified output results by adding 4 maps as shown in Fig.3e. For the training process, the estimation loss of outputs of both Side5 and Fusion are minimized in (4). For the inference process, the loss function is not used, and output images are used for center line estimation of a magazine and a grid in the next steps.

$$ \mathcal{L}_{total} = \mathcal{L}_{side5} + \mathcal{L}_{fuse} \tag{4} $$

2) 2D Center Alignment between Magazine and Grid

This section introduces how to find center lines of detected object contours from the deep learning model.

2.1. Hough Line Transform

Hough line transform is a popular solution to detect shapes. It uses two terms *(ρ, θ)* to represent a line equation in (5), where ρ is the perpendicular distance between origin to the line, and θ is the angle between the horizontal axis of the image plane and the perpendicular line. The parameters *(ρ, θ)* are changed by minimum unit, and a line generated according to the parameters temporary, and the line and the pixels of the image are compared. After comparing the pixels and the line, if the comparison result is higher than the threshold of the function, then the function returns the detected lines. This process is repeated with changing parameters until most of the lines are found.

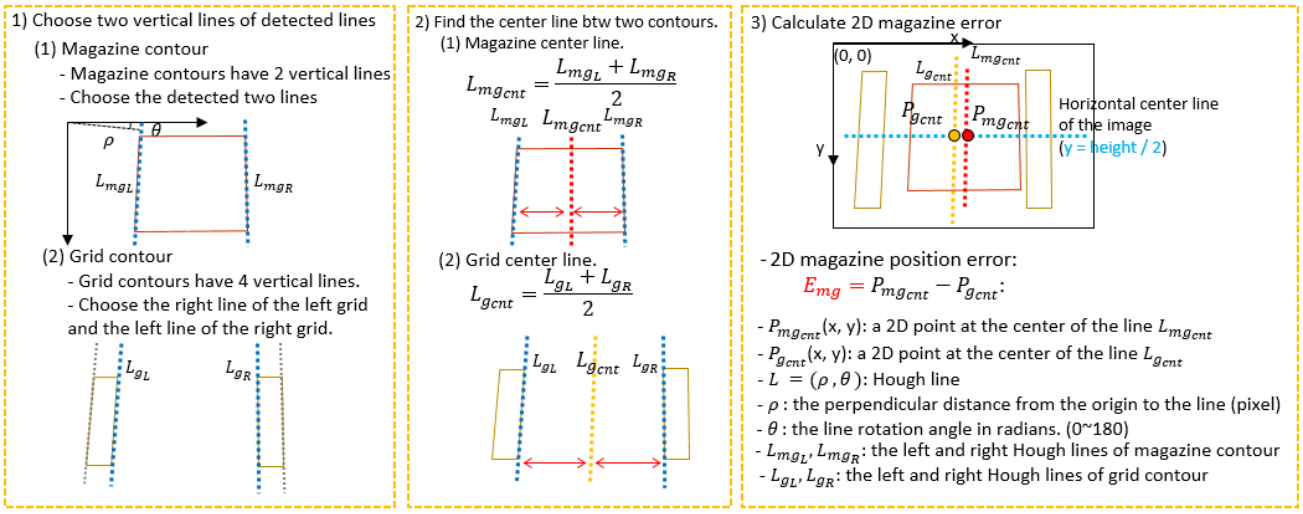

Fig. 5. The description of the 2D magazine error calculation.

$$\rho = x * cos(\theta) + y * cos(\theta) \quad (5)$$

2.2. Vertical Contour Lines Detection.

Using the Hough line transform solution, contour lines in an input image are found. However, in order to find a center line of an object, only two vertical lines are needed. Firstly, set a threshold of angle of a line for Hough line detection to get only vertical lines. In this project, the threshold was $\theta \leq 10°$ and $\theta \geq 350°$. Secondly, two vertical lines by the threshold are obtained in the magazine contour image $S_{c=0}$, while four vertical lines are obtained in the grid contour image $S_{c=1}$. After that, the rightmost line of the left grid and the leftmost line of the right grid are chosen to proceed to the next step. Fig.5 illustrates vertical lines, where L_{mgL}, L_{mgR} are the left and right Hough lines of magazine contour. L_{gL}, L_{gR} are the left and right Hough lines of grid contour.

With 2 vertical lines, a center line can be found by an average method. There are two parameters in Hough line transform, which are angle and distance. From both the left line and the right line, angles and distances averaged angle and distance of a center line (see Fig.5).

2.3 2D Magazine Position Error Estimation.

In this part, finally, 2D magazine position and grid position can be compared for the error estimation. One center point in each center line is needed where is in the horizontal center line of the image. The center points are the intersection points between the horizontal center line of the image and two vertical center lines of a magazine and a grid. Each center point $P_{mg_{cnt}}$ and $P_{g_{cnt}}$ in each center line from the previous step can be compared as:

$$E_{mg} = P_{mg_{cnt}} - P_{g_{cnt}} \quad (6)$$

V. Experimental Results

In this section, there are two experimental results. The first experiment shows an accuracy test of the marker detection with frame movement. Lastly, storing and retrieving test using the marker detection and the deep learning solution is implemented. In the experiments, Intel i7 CPU and NVidia RTX 2070 are used and two operation systems, Windows 10 and Ubuntu 16.04 are used.

A. *Experimental Result of Marker Detection Accuracy*

This experiment shows the accuracy of the marker detection according to the different positions of a frame.

1) Test Procedure.

First, the system set an initial position of the frame f_0: (0, 0, 0) and generate 12 different frame position cases as shown in Table I. In which: *x, y, z*: are horizontal axis, up-down axis, and forward-backward axis, respectively. Secondly, four marker positions based on the robot arm are saved to the system using the marker detection method. Third, according to the generated frame positions, the robot arm moves to one frame position along x, y, z-axis with a distance of 1mm. Fourth, the marker detection is executed as the second step. After that, the third and fourth steps are repeated until the test cases are finished. Finally, with the collected marker positions, accuracy is computed using metrics.

2) Frame Moving Positions for the Test.

The frame position control system has two directions that can move the frame, which are *x* and *z* directions. In this test, 12 different position cases were generated (see Table I).

3) Metrics.

Average error and standard deviation (ST_DEV) metrics are used as shown in (7) and (8). In the metrics, P_i is the i[th]

TABLE I
THE DIFFERENT FRAME POSITION CASES

f_i	(x, y, z) *(mm, mm, mm)*	Position	f_i	(x, y, z) *(mm, mm, mm)*	Position
f_0	(0, 0, 0)	Initial			
f_1	(−1, 0, 0)		f_7	(0, 0, -1)	
f_2	(−2, 0, 0)	Left	f_8	(0, 0, -2)	Closer
f_3	(−3, 0, 0)		f_9	(0, 0, -3)	
f_4	(1, 0, 0)		f_{10}	(0, 0, 1)	
f_5	(2, 0, 0)	Right	f_{11}	(0, 0, 2)	Away
f_6	(3, 0, 0)		f_{12}	(0, 0, 3)	

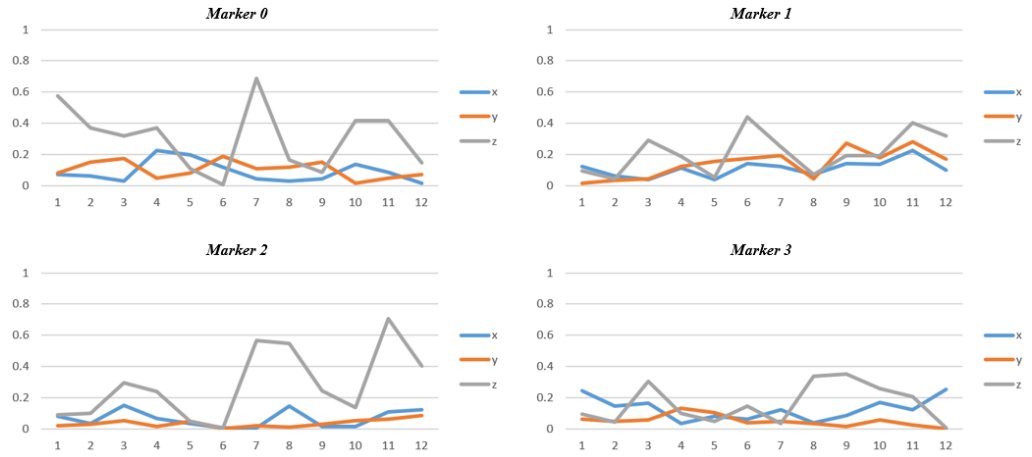

Fig. 6. Graph of marker detection accuracy test with 4 markers

TABLE II
MARKER POSITIONS ERROR DATA OF ACCURACY TEST

	Marker 0	Marker 1	Marker 2	Marker 3	Average
	(x, y, z) *(mm, mm, mm)*	(x, y, z) *(mm, mm, mm)*	(x, y, z) *(mm, mm, mm)*	(x, y, z) *(mm, mm, mm)*	(x, y, z) *(mm, mm, mm)*
E_{avg}	(0.05, 0.03, 0.32)	(0.05, 0.01, 0.15)	(0.03, 0.12, 0.09)	(0.04, 0.09, 0.08)	(0.04, 0.06, 0.16)
ST_DEV	(0.01, 0.01, 0.16)	(0.01, 0.02, 0.10)	(0.02, 0.07, 0.05)	(0.02, 0.03, 0.06)	(0.01, 0.03, 0.09)

translation position between the frame and the markers. Average error metric literally indicates average values of marker detection errors, while standard deviation shows stability of errors.

$$Average\ Error\ (E_{avg}) = \sum_{i=1}^{k=12} (|(P_i - P_0) - f_i|)/k \quad (7)$$

$$Standard\ Deviation\ = \sqrt{\frac{\sum_{i=1}^{k=12}\left(|(P_i - P_0) - f_i| - E_{avg}\right)^2}{k-1}} \quad (8)$$

where: f_i is the i^{th} frame position

4) Result.

The requirement of allowance error in this work is less than 1mm for *x*, *y*, and *z* direction of marker position. As shown in Table II, the results show all errors are under 1mm. However, there are still big errors of *z*-axis when compared to x and y errors. Fig.6 shows stable *x* and *y* errors but unstable *z* errors. The reason for the big error on z-axis is because the camera's hardware specification for z-axis has low precision which is about 0.36mm. This precision can be improved by changing the baseline or using other cameras.

B. Experimental Result of Contour Detection using DFF-Net

1) Data Collection.

In order to collect the dataset, frame and magazine images are captured using the stereo camera and the robot arm. After that, the dataset is labeled using a labelling tool which is *Labelme* [18] (see in Fig.7). There are 2 classes, which are frame and magazine, and the image size is 2448 x 2048. Although the dataset of this case has only one kind of magazine, the model can be trained with other shapes of magazines. For augmentation of data, collected images are rotated and copied with [-20, 0, 20] degree. 350 images were collected for the work and they are increased to 1,050 images. For training, validating, and testing sets, the dataset separated to 900, 100, and 50, respectively.

2) Metric

To evaluate the segmentation performance of DFF-Net, F1 score metric is used. F1 score is obtained by Precision and Recall metrics which are the popular methods as in (). F1 method is to find balance between Precision and Recall. A good model should have a high F1 score with balanced Precision and Recall. To get Precision and Recall, different types of results from comparison outputs and ground truths.

$$F1 = 2x\frac{precision\ x\ recall}{precision + recall} \quad (9)$$

3) Training Results.

In the test case, 50 testing images are used for Precision, Recall, and F1 score as shown in Table III. Training time was approximately 20 hours with 50 epochs and 1 batch size with a NVidia GPU RTX 2070.

There are examples of contour detection and center line detection of magazine and grid as shown in Fig.8. After the center line detection procedure, two center lines can be compared by pixel unit. For instance, in Fig.9, the distance between two lines is 14 pixels, and the camera specification is 0.14mm/pixel, so that the real distance can be calculated: 14-

TABLE III.
PRECISION, RECALL AND F1 RESULTS ON FRAME AND MAGAZINE DATASET

No. Images	Precision (%)	Recall (%)	F1(%)
50	62.7	75.9	**68.7**

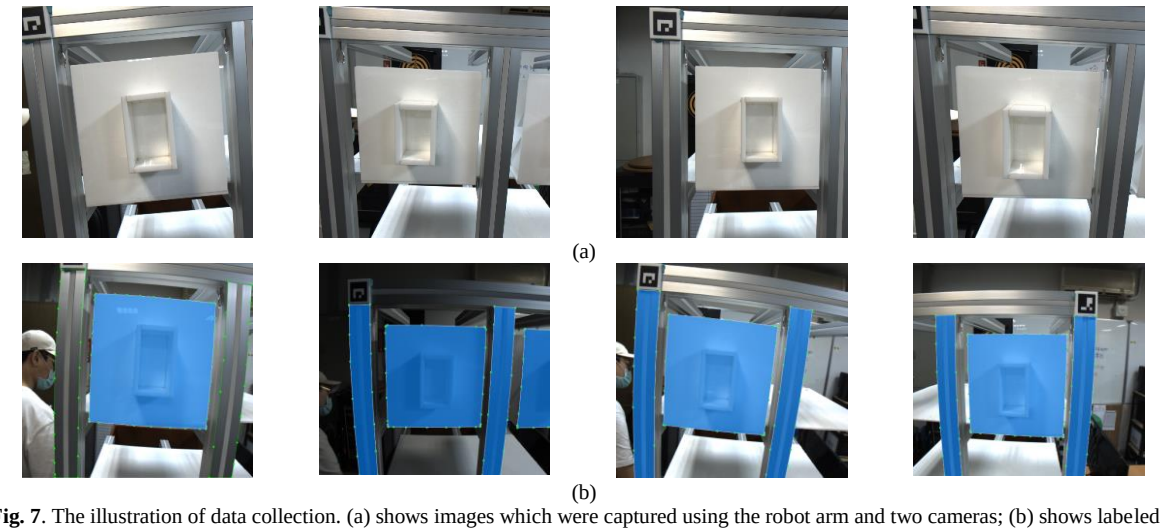

Fig. 7. The illustration of data collection. (a) shows images which were captured using the robot arm and two cameras; (b) shows labeled images.

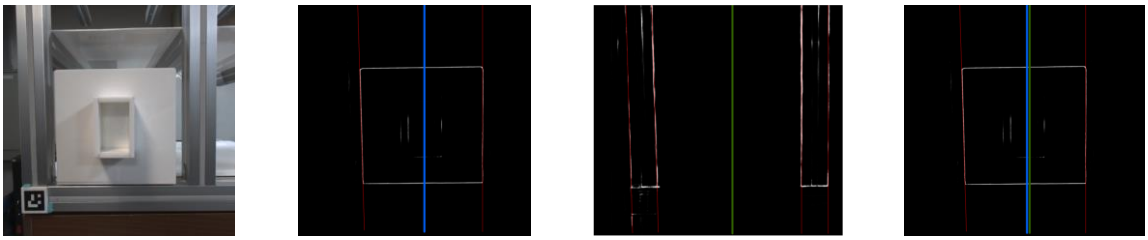

Fig. 8. The illustration of the results of contour detection and center line detection. (a) shows an input image; (b) shows magazine class; (c) shows frame class; and (d) shows center line error.

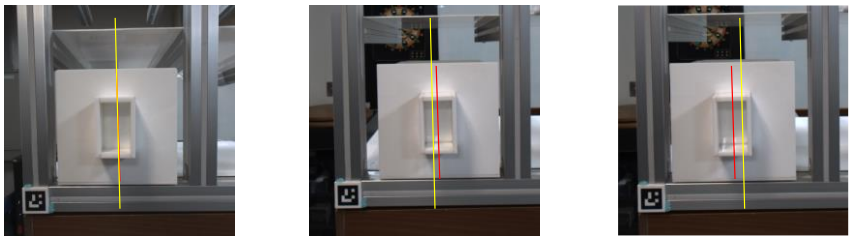

Fig. 9. The illustration of the error estimation; Red: magazine center line; Yellow: grid center line. (a) shows an error of -0.7 mm; (b) shows an error of +6.1 mm; and (c) shows an error of -7.4 mm.

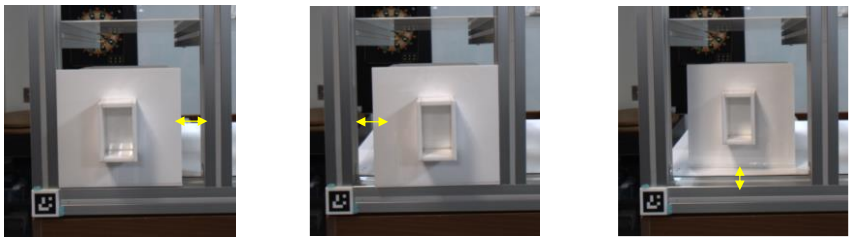

Fig. 10. The illustration of the wrong magazine detection. (a) shows magazine be too close to the grid's left side; (b) shows magazine be too close to the grid's right side; and (c) shows magazine be too far from the grid's front.

pixel x 0.14 mm/pixel = 1.96 mm.

4) Problems

In some situations, magazines cannot be detected correctly as below (see in Fig.10). And sometimes the system detects the interior frames so that it cannot distinguish correct frame surfaces. To solve the problems mentioned above, the model must be trained with datasets with various magazine and frame positions.

VI. Conclusion and Future Work

In the work, a solution of storing and retrieving objects using a stereo camera and a robot arm is introduced. The warehouse automation system with 3D vision, which can check the status periodically, was developed. Furthermore, when there are external effects such as earthquakes, the system can be retooled automatically by frame coordinate detection with ArUco markers. Especially, this is a suitable system where the ground

can be unstable, like in Taiwan, because earthquakes are detected frequently so that many factories suffer from it and must reset the system manually.

Moreover, using a deep learning method, the system can check whether magazines are in the right place or not in the frame. Deep learning methods are applied in many industrial fields these days. In the project, different shapes of magazines are also detected precisely after training magazines. It is very flexible to install various systems.

Although an efficient and flexible system has developed, some problems should be solved in the future. Firstly, the deep learning method has no object detection function, only contour detection. Therefore, it takes more time to find the object position by Hough line transform. To solve this, object detection layers can be added to the deep learning model, so that it will reduce time and processes because deep learning is executed on GPU. Secondly, the marker detection method has restrictions to be adapted to other systems because markers should be attached on frames, and marker size must be big enough to be detected by cameras. Besides, the stereo system cannot detect the depth of magazines, but only the depth of markers. Therefore, other deep learning methods that can detect 3D positions of frames and magazines without markers should replace marker detection in future works. Lastly, under a dark environment, the exposure time of cameras must be higher, and therefore affects working time. Adding additional light on the robot arm can be a solution for diverse environments.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science and Technology (MOST), Taiwan, R.O.C., grant MOST 109-2221-E-006-190 - Tongtai machine & tool Co., Ltd. and Contrel technology Co., Ltd.

REFERENCES

[1] F. MK; M. Zulkhairi; M. Aswadi; and A. Ismail, "Development of automated storage and retrieval system (ASRS) for flexible manufacturing system (FMS)," *Journal of Engineering Technology*, vol. 4, pp. 43-50, 2016.

[2] M. Schwarz, A. Milan, C. Lenz, A. Munoz, A. Periyasamy, M. Schreiber, S. Schuller, and S. Behnke. "Nimbro picking: Versatile part handling for warehouse automation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3032-3039, 2017.

[3] R. Prakash, L. Behera, S. Mohan, and S. Jagannathan. "Dual-loop optimal control of a robot manipulator and its application in warehouse automation," *IEEE Trans. on Automation Science Engineering*, pp. 1-18, 2020.

[4] G. Ricardez, L. Hafi, and F. Drigalski, "Standing on giant's shoulders: Newcomer's experience from the Amazon Robotics Challenge 2017," in *Advances on Robotic Item Picking: Springer*, pp. 87-100, 2020.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", in *IEEE conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[6] J. Redmon, and A. Farhadi, "YOLO9000: better, faster, stronger", in *IEEE conference on Computer Vision and Pattern Recognition*, pp. 7263-7271, 2017.

[7] Z. He, V. Aggarwal, and S. Nof, "Differentiated service policy in smart warehouse automation," *International Journal of Production Research*, vol. 56, no. 22, pp. 6956-6970, 2018.

[8] S. Xie, and Z. Tu, "Holistically nested edge detection," in *IEEE International Conference on Computer Vision*, pp. 1395-1403, 2015.

[9] Z. Yu, C. Feng, M.Y Liu, and S. Ramalingam, "CASENet deep category-aware semantic edge detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1761-1770, 2017.

[10] Y. Hu, Y. Chen, X. Li, and J. Feng, "Dynamic feature fusion for semantic edge detection," in *International Joint Conferences on Artificial Intelligence*, pp. 782-788, 2019.

[11] R. Duda, and P. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Artificial Intelligence Center*, vol. 15, no. 1, pp. 11-15, Jan. 1972.

[12] R. Xavier, B. Silva, and L. Goncalves, "Accuracy analysis of augmented reality markers for visual mapping and localization," in *IEEE Workshop of Computer Vision*, pp. 73-77, 2017.

[13] A. Kaehler, and G. Bradski, "Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library", O'Reilly Media, Inc., 2016.

[14] R. Tsai, and R. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand-eye calibration," in *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345-358, June 1989.

[15] Z. Zhang, "A flexible new technique for camera calibration," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.

[16] K. Daniilidis, "Hand-eye calibration using dual quaternions," in *International Journal of Robotics Research*, vol. 18, no. 3, pp. 286-298, 1999.

[17] F. Park, and B. J. Martin, "Robot sensor calibration: Solving AX = XB on the Euclidean group," in *IEEE Transactions on Robotics and Automation*, vol.10, no. 5, pp. 717-721, 1994.

[18] K. Wada, "Labelme: Image polygonal annotation with Python," Accessed on: Jan.7, 2020 [Online]. Available: https://github.com/wkentaro/labelme.

Minh-Tri Le received the M.S degree in the Departments of Electronics and Electrical Engineering from HCM University of Technology and Education, Ho Chi Minh City, Vietnam in 2009. He is currently a Ph.D. candidate with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. His current research interests and publications are in the areas of Embedded System, Deep Learning, and Visual-Guided Robot Arm Control and Automation.

Soonmyun Jang received the B.S degree from Beijing Jiaotong University, Beijing, China in 2014, and the M.S. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2020. He is currently a technical engineer with the Contrel technology Co., Ltd, Tainan, Taiwan. His current research interests and publications are in the areas of Deep Learning, and Visual-Guided Robot Arm Control and Automation.

Jenn-Jier James Lien. (M'00) received his B.S. degree in biomedical engineering from Chung Yuan University, Taiwan, in 1989, and his M.S. and Ph.D. degrees in electrical engineering from Washington University, St. Louis, MO, and University of Pittsburgh, Pittsburgh, PA, in 1993 and 1998, respectively. From 1995 to 1998, he was a research assistant at the Vision Autonomous Systems Center, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. From 1999 to 2002, he was a senior research scientist at Identix, where he was also a project lead for the DARPA surveillance project on Human Identification at a Distance. In 2002, he joined department of computer science and information engineering at National Cheng Kung University (NCKU), Taiwan, as an assistant professor and director of the robotics laboratory. He was a director of Institute of Manufacturing Information and Systems at NCKU from 2015 to 2018. Currently, he is the director of AI Robotics at Miin Wu School of Computing at NCKU. His industry-academia collaboration research fields consist of deep learning for computer vision, 2D/3D automatic optical inspection, visual-guided robot arm control and automatic guided vehicle.