IAES International Journal of Robotics and Automation (IJRA)
Vol. 11, No. 2, june 2022, pp. 111∼121
ISSN: 2722-2586, DOI: 10.11591/ijra.v11i2.pp111-121

An efficient regression method for 3D object localization in machine vision systems

Xiem HoangVan, Nam Do
Department of Robotics Engineering, Faculty of Electronics and Telecommunications
Vietnam National University – University of Engineering and Technology, Hanoi, Vietnam

Article Info

Article history:

Received Nov 10, 2021
Revised Dec 25, 2021
Accepted Jan 10, 2022

Keywords:

Camera calibration
Machine (robot) vision system
Regression

ABSTRACT

Machine vision or robot vision plays is playing an important role in many industrial systems and has a lot of potential applications in the future of automation tasks such as in-house robot managing, swarm robotics controlling, product line observing, and robot grasping. One of the most common yet challenging tasks in machine vision is 3D object localization. Although several works have been introduced and achieved good results for object localization, there is still room to further improve the object location determination. In this paper, we introduce a novel 3D object localization algorithm in which a checkerboard pattern-based method is used to initialize the object location and followed by a regression model to regularize the object location. The proposed object localization is employed in a low-cost robot grasping system where only one simple 2D camera is used. Experimental results showed that the proposed algorithm significantly improves the accuracy of the object localization when compared to the relevant works.

Corresponding Author:

Xiem HoangVan
Department of Robotics Engineering, Faculty of Electronics and Telecommunications
Vietnam National University - University of Engineering and Technology
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
Email: xiemhoang@vnu.edu.vn

1. INTRODUCTION

Nowadays, machine vision technology has been playing an important role in automation and industrial 4.0. A large number of applications have been introduced such as: part identification of complex systems, defect inspecting, optical character recognition (OCR) reading, 2D code reading, and especially object picking [1]. Figure 1 illustrates a general structure of typical industrial vision systems. The system includes three main parts: a computer or embedded processor to be connected with a camera, a manipulator (arm) Robot and a flat table or conveyor [2]. In such systems, the computer is employed to process images captured from the camera. This is achieved by applying special-purpose image processing analysis and classification software. The position of the camera is usually fixed. In many cases, machine vision systems are designed to inspect and pick only known objects at variable positions. The scene is then appropriately illuminated and arranged to facilitate the reception of the image features necessary for processing and classification.

In machine vision systems, camera calibration is a necessary step to extract information from 2D-image to understand the real 3D object and to devote the identification of pixel/mm ratio between a projected object in the image and real 3D object [3]. This parameter is fundamental for the correct valuation of the object under inspection or picking. To achieve object localization, several methods have been introduced such as plumb line method [4], two-stage method [5].

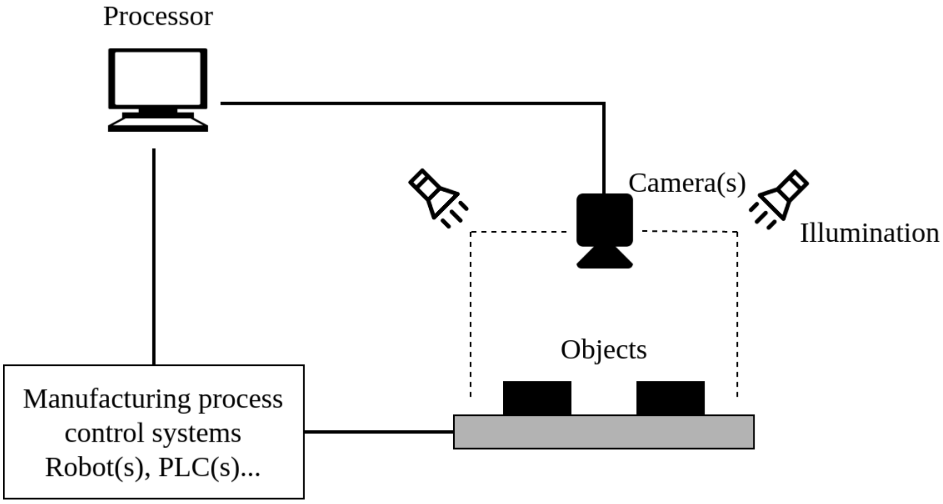


Figure 1. Illustration of a machine vision system

Plumb line method [4] is one of few approaches to achieve camera calibration which implements a practical model to solve the problem that the lenses are not symmetrical due to manufacturing. The disadvantages of this method lie in the manual determination of calibration points and the complexity in determining the real principal points at the offset from the ideal axis of the lens.

The two-stage method [5] proposed by Tsai. uses the same camera model in [4], but focuses more on the characterization of real-time operation. This method uses a checkerboard pattern to compute the scale factor in the first stage and computes the effectiveness of focal length, distortion coefficients in the second stage. Due to the assumption of a simple camera model (e.g. pinhole model) and ignorance of the projection of 3D objects in 2D images, this method still has limits in many cases.

The methods proposed in [6]-[11] are based on direct linear transformation (DLT) assumption. The strategy is to find the radial and tangential distortion coefficients based only on linear transformation. Researchers [7], [8] the DLT is utilized to simplify the algorithm of the plumb line method. Sturm and Maybank [9] proposed using a 3D calibration pattern composed of 3 planes with calibrated dots on each. This method is simple to implement and general, but requires an accurate 3D calibration pattern.

Meanwhile, Zhang study a different technique of camera calibration [12], [13]. The method requires only a simple planar pattern but needs several pictures of the pattern taken from different orientations, which is not certain and also hard to implement in the industrial context.

Modern methods [14]-[21] tend to utilize deep learning to efficiently estimate the parameters in the camera model [14], [15], or expensive devices(e.g. lidar or 3D camera) to localize objects on the z-axis [16]-[19]. These methods either only focus on undistorting images and do not consider the actual position of each object on the images (caused by deep learning hypothesis) or require modern firmware, which is costly and not suitable in the industrial context.

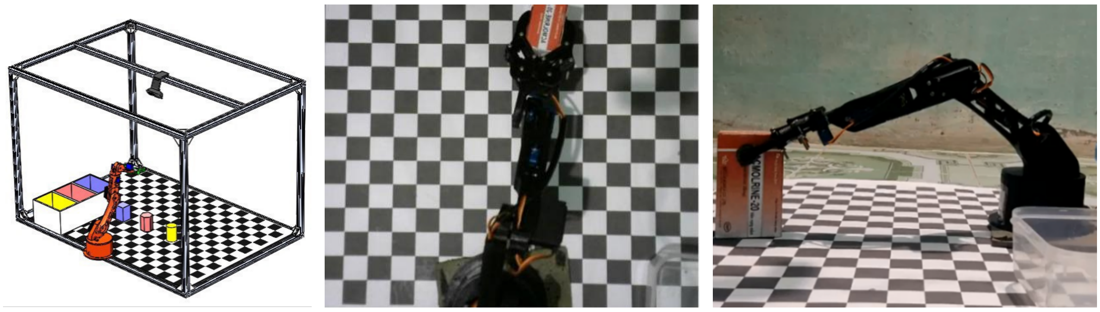
Although all the mentioned methods give good results in localizing flat objects (e.g. books, papers, and smartphones). They tend to fail in the case of localizing 3D-shaped objects. Due to their 3D natural shape, the location determined by calibration methods is in fact the location of their projections on the image instead of their real location. Therefore, the error distance to the real location of the objects still remains. In many cases, which we will examine in Section 2, this error is crucial and can greatly affect the performance of the grasping system. To address this problem, in this paper we propose a machine learning – regression-based method for improving the accuracy of 3D object localization. The proposed method is created based on a mathematical modeling of 3D objects and their projected image in the 2D plane and followed with a regression-based algorithm to achieve model parameters. Experimental results captured from our practical machine vision system demonstrated the advancements of the proposed regression model in both location accuracy determination and low complexity time requirement.

The organization of this paper is as follows. Next, Section 2 presents the system design of our machine vision system and introduces a regression model for correcting the 3D object localization. Section 3 evaluates the performance of the system. Finally, Section 4 gives some conclusions and works for the future.

2. RESEARCH METHOD

2.1. Machine vision system design

To study the machine vision problems, we examined a popular system design for robot picking objects [2]. Figure 2 illustrates a system where a camera is located at the top of a frame and connected with a PC. Here, we use a manipulator robot with 4DoF (degree of freedom) to be controlled with an Arduino processor, and a camera connected with a personal computer (PC). Figure 2(a) shows Illustration of our robot vision system, Figure 2(b) shows object and robot from top view, and Figure 2(c) shows object and robot from side view. The system is designed to adaptively pick and move objects. The image captured with the camera will be used to detect and localize the object and sent to the robot for the picking task. For easy demonstration, we used a checkerboard pattern with 3×3 cm each square (black or white) at the bottom of the frame. This checkerboard pattern will be used in determining the location of the object, see Figure 3.



(a) (b) (c)

Figure 2. Machine vision system (a) Illustration of our robot vision system, (b) object and robot from top view and (c) object and robot from side view

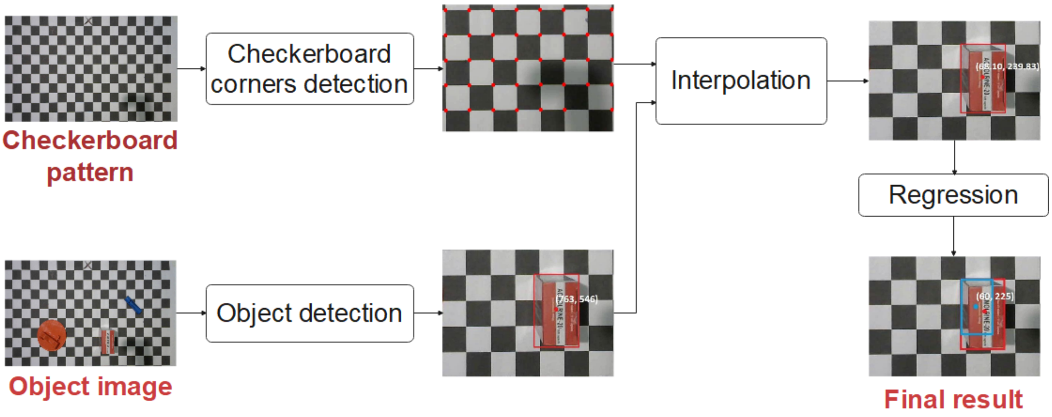
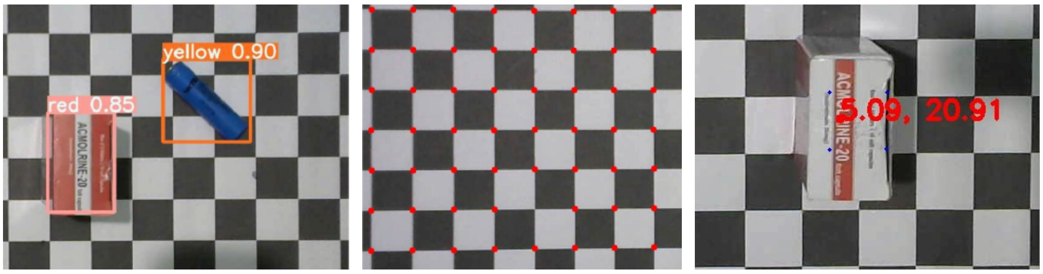


Figure 3. Proposed object localization flow

To achieve the object location, we take a picture of the checkerboard floor to get checkerboard corners. Then, with each object image, we employ a convolutional neural network (CNN) based object detection method [22] to find their image coordinates. Combining those two pieces of information, we interpolate real coordinates of the objects as illustrated in Figure 4(a), and finally regulate the location with a regression method.

In checkerboard corners detection, we find the correlation between the image coordinates and real-world coordinates. Practically, the distances on images and their respective real-world distances are not always linearly dependent due to lens distortion. Camera calibration methods [23] tried to solve this problem by finding radial and tangential distortion coefficients to undistort the image by detecting patterns (usually a checkerboard) with a fixed size. In most industrial cases, a camera calibration method is employed and real-world coordinates are then acquired by scaling up distances on the image. This method is simple to deploy thus suitable for the industrial environment, yet needs a good camera which is costly.

In a small region, the error caused by the camera lens is low, we can instead employ the checkerboard pattern as a grid net and acquire real-world coordinates of the object by interpolating from coordinates of neighbor checkerboard corners. Figure 4 shows initial object location can be determined as the following steps: i) Step1. Determine the checkerboard edges and the coordinates of their intersections (checkerboard corners) using the Hough transformation [24] (as illustrated in Figure 4(b)); ii) Step 2. Assume that the Robot is located at (x_0, y_0) in the center of a checkerboard cell, and the coordinate system created by the chessboard is nearly parallel to the image coordinate system. Then the corner at (x_C, y_C) has the real coordinates of $\left(\frac{[2dx_C/l]}{2}, \frac{[2dy_C/l]}{2}\right)$ with $dx_C = x_C - x_0$; $dy_C = y_C - y_0$; l and r is average length of a cell on the image and in real world respectively; the "[]" notation denotes floor function; iii) Step 3. With the image coordinates (x, y) of the object. We find 4 checkerboard corners that are closest to the object in 4 ways: left top, right top, left bottom, right bottom. And simply utilize a bi-linear interpolation [25] to compute the object's real location as shown in Figure 4(c).



(a) (b) (c)

Figure 4. Object detection result (a) object determination with CNN, (b) corner detection results, and (c) object localization result

2.2. Proposed regression method

Although estimating the object location with the checkerboard calibration method is simple and easy to deploy in the industrial field, it tends to be inefficient when the object is not flat but a 3D-shaped object. In such cases, the bounding box of the object detected by deep learning techniques (described in Section 2.1) may not accurately match the location of objects. Specifically, the coordinates of the center of the bounding box will hardly be the coordinates of the center of the object on the resulting image. Note that the center of the bounding box can only be the center of the object if and only if the object is placed in the center of the projection of the camera onto the floor.

To reveal this fact, we tested the checkerboard calibration method for some cases as shown in Figure 5. The bounding box obtained by 2D object detector [22] with the red center indicates the coordinates of the projection of the object while the blue center is the actual coordinates of the object. The obtained results demonstrate the difficulty of the checkerboard calibration method with 3D objects.

Here, two main factors are affecting the error of 3D object coordinate estimation with the checkerboard pattern method are the height of the object: the higher the object, the larger the error and the relative position of the object to the camera: the farther the object is placed from where the camera projects down on the floor, the greater the error. In this study, we will analyze the second feature, which is the influence of the position of the object, and apply this feature to correct the error of estimating the coordinates of the object.

Figure 6 illustrates the relative position of the object to the camera. We can easily recognize that when the object is placed in the center of where the camera projects on the floor (object 1), the bounding box can accurately reflect the projection of the object and thus the position of the center of the bounding box coincides with the center of the object. Whereas with object 2 placed deviated from the position of the camera, the center of the object and the center of the bounding box and the center of the object will not coincide.

Roughly speaking, if we know the distance from the camera to the floor, the height of the object, and the position of the camera, we can use geometric methods to determine the difference between the actual coordinates object and its projection coordinates if the camera is considered as a light source, see Figure 7.

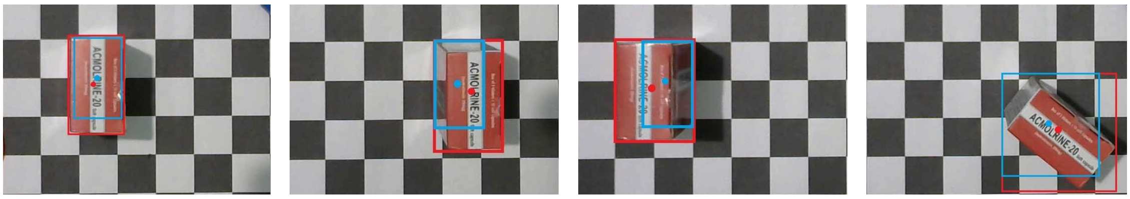


Figure 5. Error in object localization caused by object shape and tilt angle it made with the camera. The red box and dot denotes the object bounding box and center detected by you only look once (YOLO) respectively. While the blue box and blue dot is the base and center of the object on the floor

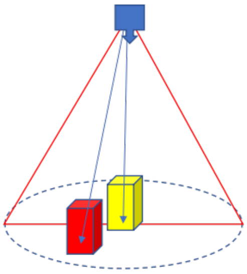


Figure 6. Illustration of relative position of objects to the camera

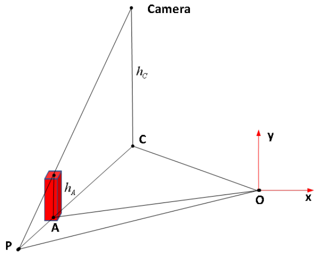


Figure 7. Illustration of the projection of object on the floor

According to Thales theorem, we have (1).

$$\frac{PA}{PC} = \frac{h_A}{h_C} \tag{1}$$

Here, PA is the distance from the object to its projection on the floor, PC is the distance from the camera's projection of the object and is determined by (2).

$$\vec{PC} = \vec{OP} - \vec{OC} \tag{2}$$

Then PA can be determined by (3).

$$PA = PC.\frac{h_A}{h_C} \tag{3}$$

Where O is the origin coordinates of the robot. Then, the actual position of the object can be calculated as (4).

$$\vec{OA} = \vec{AP} + \vec{PO} \tag{4}$$

However, practically with the robot vision system, we usually do not know the height of the object as well as the distance from the camera to the floor. Therefore, we propose a machine learning method, using a regression technique to estimate the actual position of the object from the position of the center of the bounding box of the projection obtained from the checkerboard calibration method.

To determine the distance from the actual position of the object to the position of the projection, (AP) we state the following two propositions. i) Proposition 1: There always exists a point of convergence (called C - convergence) at which the coordinates of the projection P coincides with the coordinates of object

A. Proof: This can be easily seen as in the previous section. Here, the convergence point C is the coordinates of the camera projection to the floor; ii) Proposition 2: The distance from the object to the convergence point C is proportional to the distance from the object to its projection. Proof: This proposition can be proved by geometric methods as illustrated in Figure 8, two objects A and B have the same shape and size but are placed in two different distance to C. We can easily prove that (5).

$$\frac{P_A}{P_C} = \frac{h_A}{h_C} = \frac{h_B}{h_C} = \frac{P'_B}{P'_C} \tag{5}$$

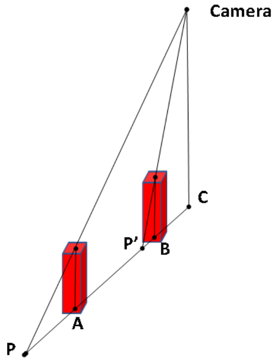


Figure 8. Illustration of distance from projection to object in different locations

Based on propositions 1 and 2, we can model the relationship between the distance of the convergence point C - the object's position and the distance between the object's position and the projection's position as (6).

$$\|C - A(i)\|_2 = \alpha\|P(i) - A(i)\|_2 + \beta \tag{6}$$

Here, the location of C is fixed, $A(i)$ and $P(i)$ are the position of the object and the bounding box corresponding with i^{th} example respectively. α, β are the model parameters to be determined. Bias parameter β can be added to handle error caused by object shape, camera tilt angle etc. In our experiment we simply set $\beta = 0$.

We can estimate the location of convergence point $C(x_C, y_C)$ and α, β using linear regression with least square optimization, i.e. minimize the following objective function as (7).

$$\alpha, \beta, C = \operatorname{argmin} f(\alpha, \beta, C) \tag{7}$$

With :

$$f(\alpha, \beta, C) = \frac{1}{n}\sum_{i=1}^{n}(\|C - A(i)\|_2 - (\alpha\|P(i) - A(i)\|_2 + \beta)^2 \tag{8}$$

Or:

$$f(\alpha, \beta, x_C, y_C) = \frac{1}{n}\sum_{i=1}^{n}\left(\alpha - \frac{\sqrt{(x_C - x_A(i))^2} - \beta}{\sqrt{(x_P(i) - x_A(i))^2 + (y_P(i) - y_A(i))^2}}\right)^2 \tag{9}$$

To be simple, we solve the problem with $\beta = 0$, the remaining case can be solved similarly.
Set $\sqrt{(x_P(i) - x_A(i)^2 + (y_P(i) - y_A(i))^2} = \gamma(i)$, we have (10).

$$\alpha, x_C, y_C = \operatorname{argmin} \frac{1}{n}\sum_{i=1}^{n}\left(\alpha - \frac{\sqrt{(x_C - x_A(i))^2}}{\gamma(i)}\right)^2 \tag{10}$$

Then α, x_C, y_C can be determined by solving as (11a) to (11c) and (12a) to (12c).

$$\frac{df}{d\alpha} = 0 \tag{11a}$$

$$\frac{df}{dx_C} = 0 \tag{11b}$$

$$\frac{df}{dy_C} = 0 \tag{11c}$$

Or:

$$\frac{1}{n}\sum_{i=1}^{n} 2\left(\alpha - \frac{\sqrt{(x_C - x_A(i))^2}}{\gamma(i)}\right) = 0 \tag{12a}$$

$$\frac{1}{n}\sum_{i=1}^{n} -2\left(\alpha - \frac{\sqrt{(x_C - x_A(i))^2 + (y_C - y_A(i))^2}}{\gamma(i)}\right) \frac{2(x_C - x_A(i))}{\gamma(i)\sqrt{(x_C - x_A(i))^2 + (y_C - y_A(i))^2}} = 0 \tag{12b}$$

$$\frac{1}{n}\sum_{i=1}^{n} -2\left(\alpha - \frac{\sqrt{(x_C - x_A(i))^2 + (y_C - y_A(i))^2}}{\gamma(i)}\right) \frac{2(y_C - y_A(i))}{\gamma(i)\sqrt{(x_C - x_A(i))^2 + (y_C - y_A(i))^2}} = 0 \tag{12c}$$

From (12a), we have:

$$\alpha = \sum_{i=1}^{n} \frac{\sqrt{(x_C - x_A(i))^2 + ((y_C - y_A(i))^2}}{n\gamma(i)} \tag{13}$$

Therefore, can be calculated based on (x_C, y_C). The problem then only has 2 variables x_C and y_C. Equation (12b) and (12c) can't be solved directly, instead we can use the gradient descent algorithm to estimate root as algorithm 1:

Algorithm 1

Step 1. Randomly set $(x_C(0), y_C(0))$. Calculate α_0 based on (13).

Step 2. Update the value of $(x_C(k+1), y_C(k+1))$ by $(x_C(k), y_C(k))$ using gradient descent:

$$x_C(k+1) = x_C(k) - \sigma\frac{df}{dx_C(k)} \tag{14}$$

$$y_C(k+1) = y_C(k) - \sigma\frac{df}{dy_C(k)} \tag{15}$$

In which σ is learning rate. In our work we simply set $\sigma = 1$.

Step 3. Calculate $\alpha(k+1)$ based on (13).

Step 4. If $\frac{df}{dx_C(k+1)}$ and $\frac{df}{dy_C(k+1)}$ are small enough, stop. Else back to step 2.

Finally, the obtained regression model parameters, α, β and the convergence location, (x_C, y_C), are fed to (6) to determine the actual object location.

3. RESULTS AND DISCUSSION

3.1. Experiment setup

To examine the proposed object location estimation method, we set up a machine vision system with a camera on top, parallel and 60 cm away from the floor. For the interpolation phase we use a checkerboard pattern with size of each square is 3x3 cm^2. Images taken from the camera is shown in Figure 9.

A Robot system with 4DoF and Arduino processor was employed for object picking in this machine vision system. To examine the accuracy of object location, we apply a k-fold validation strategy on 20 obtained images. Specifically, we divide the 20-image dataset into 4 folds, with 5 images in each fold. In each step, we use 1 in 4 folds to be the test set and the remaining folds be the training set.

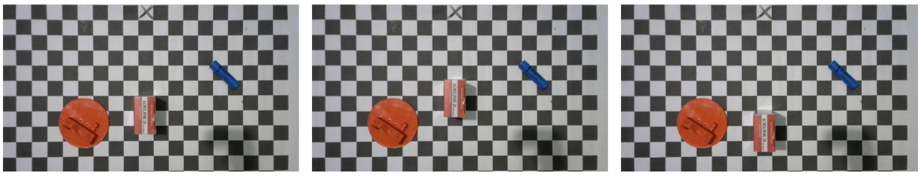


Figure 9. Several images in our dataset

3.2. Object localization evaluation

Table 1 describes error in real world coordinates of predicted location with ground truth location based on x, y axis and Euclidean distance when using traditional method (described in section 2) and our proposal ($Err = \sqrt{\Delta x^2 + \Delta y^2}$).

Table 1. Results of performance evaluation of our regression method in error reduction (mm)

Fold	Sample	Traditional method			Proposed method		
		$\triangle x$	$\triangle y$	Err	$\triangle x$	$\triangle y$	Err
1	1	8.10	14.80	16.87	0.99	2.43	2.63
	2	10.40	15.20	18.42	0.13	0.38	0.40
	3	15.00	8.20	17.10	0.63	0.29	0.69
	4	3.60	10.40	11.01	3.13	1.67	3.55
	5	19.20	6.10	20.15	1.96	2.17	2.93
2	1	20.20	2.30	20.33	4.22	5.82	7.19
	2	2.20	11.90	12.10	2.80	0.32	2.81
	3	5.80	13.30	14.51	1.99	0.93	2.19
	4	12.20	14.00	18.57	1.60	1.56	2.24
	5	14.30	15.40	21.02	0.28	0.29	0.40
3	1	13.50	3.50	13.95	4.37	3.25	5.45
	2	0.10	15.60	15.60	0.76	2.03	2.17
	3	20.20	6.90	21.35	0.59	5.85	5.88
	4	4.70	4.60	6.58	0.67	2.25	2.35
	5	0.10	5.00	5.00	0.58	1.89	1.98
4	1	8.30	2.40	8.64	0.87	2.54	2.68
	2	15.20	7.30	16.86	0.48	1.43	1.51
	3	8.80	12.80	15.50	3.57	0.20	3.57
	4	15.30	13.30	20.27	2.68	2.61	3.74
	5	12.69	9.00	15.56	1.72	0.07	1.72
Average		10.49	9.60	15.47	1.70	1.90	2.80

From the obtained results, some conclusions can be derived as i) The proposed regression method significantly improves the accuracy of object localization in machine vision system for all experiments; ii) The distance error with the proposed method can achieve up to 0.4 mm; iii) The proposed method can reduce 82% of distance error (i.e. from 15.5 to 2.8 mm) when compared to the conventional object localization without regression.

We also examine the training and validation time of the regression module in 4 folds described above. The result is shown in Table 2.

Table 2. Processing time of proposed method in training and validating phases (unit: ms)

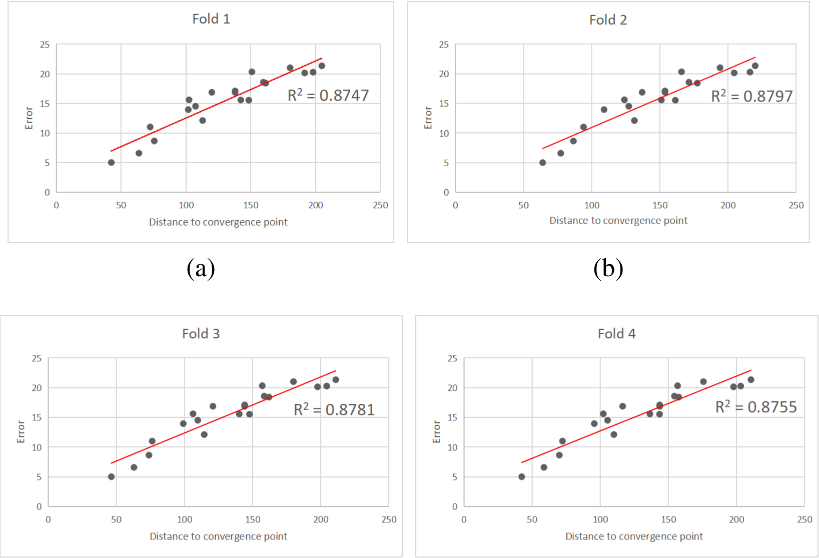
Fold	Train time	Validation time (Whole)	Validation time (Each)
1	138.81	0.13	0.03
2	137.94	0.12	0.02
3	165.41	0.07	0.01
4	346.35	0.13	0.03

As we can see in Table 2, the processing time of the regression model is small in both training and testing phases. Therefore in practice, adding this module after the existing object localization and calibration model won't affect the performance of the whole.

3.3. Model assessment

To assess the model accuracy, we verify the relationship between the distance from the center of the projection of the object to the convergence point and the received error by visualizing results over the dataset. The visualization results are shown in Figure 10. The Figure 10(a) is showing R^2 measure 0.8747, Figure 10(b) is showing R^2 measure 0.8797, Figure 10(c) is showing R^2 measure 0.8781, and Figure 10(d) is showing R^2 measure 0.8755.

As we can see through the visualization, the accuracy of the proposed linear model is nearly 0.88 with R^2 measure. In addition, the proposed method is not relying much on the data, notably the R^2 obtained with four different folds is similar, it can be seen that these two quantities have an almost linear correlation with each other.



(a) (b)

(c) (d)

Figure 10. Correlation between the distance from the center of the object's projection to convergence point and the error in calibration phase with measure (a) R^2 = 0.8747, (b) R^2 = 0.8797, (c) R^2 = 0.8781, and (d) R^2 = 0.8755 (Unit: mm)

3.4. Proposed model with bias

We expanding our method with $\beta \neq 0$. The result of model parameters in case of and Euclidean error in result and is shown in Table 3.

Table 3. Comparison of the parameters found in case of $\beta = 0$ and $\beta \neq 0$
(Coordinates of point C is the coordinates relative to the origin of the robot arm, unit: mm)

Fold	Proposed method without bias				Proposed method with bias				
	x_c	y_c	α	Err	x_c	y_c	α	β	Err
1	0.37	123.23	9.19	2.77	-3.31	46.43	7.76	-8.29	5.60
2	-0.78	120.96	9.49	3.74	-8.21	36.27	7.75	-9.30	5.43
3	2.18	118.56	9.71	1.70	14.70	-146.21	7.05	-27.61	9.48
4	3.84	108.28	10.02	2.47	-0.24	71.84	8.95	-4.52	3.32
Avg	1.40	117.76	9.60	2.67	0.74	2.08	7.88	-12.43	5.96

From the obtained results in Table 3, it can be concluded that adding a bias term, β, may not improve the accuracy of the estimated object location. The results show that our method is fast and effective in reducing error caused by the projection of 3D-shaped objects, and thus easy to deploy in industry. On the other hand, the proposed method is still simple as we focused only on the distance to the convergence point and did not consider all the information of the shape and orientation of objects. Therefore the error is still large in some specific cases. We will take all this information into account to improve the accuracy in some future works.

4. CONCLUSION

In this paper, we introduced a novel regression model to reduce errors in 3D object localization, a very important and common step in machine vision problems. The proposed method is created based on the geometry relation between the real object location and its projection information obtained with a CNN model. The proposed method significantly reduces the average error of object location from 15.47 to 2.80 mm, which is small enough to be deployed in a grabbing robot system. For future work, we can further improve the model accuracy with online learning process.

REFERENCES

[1] A. O. Fernandes, L. F. E. Moreira and J. M. Mata, "Machine vision applications and development aspects," *2011 9th IEEE International Conference on Control and Automation (ICCA)*, 2011, pp. 1274-1278, doi: 10.1109/ICCA.2011.6138014.

[2] E. N. Malamas, E. G. . Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, "A survey on industrial vision systems, applications and tools," *Image and Vision Computing*, vol. 21, no. 2, pp. 171–188, Feb. 2003, doi: 10.1016/S0262-8856(02)00152-X.

[3] S. W. H. Chester C. Slama, Charles Theurer, Ed., *Manual of Photogrammetry*, 4th ed. American Society of Photogrammetry, 1980.

[4] C. B. Duane, "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[5] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, Aug. 1987, doi: 10.1109/JRA.1987.1087109.

[6] Y. I. Abdel-Aziz and H. M. Karara, "Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 2, pp. 103–107, Feb. 2015, doi: 10.14358/PERS.81.2.103.

[7] T. A. Clarke and J. G. Fryer, "The Development of Camera Calibration Methods and Models," *The Photogrammetric Record*, vol. 16, no. 91, pp. 51–66, Apr. 1998, doi: 10.1111/0031-868X.00113.

[8] J. G. Fryer, T. A. Clarke, and J. Chen, "Lens distortion for simple C-mount lenses," in *International Archives of Photogrammetry and remote sensing*, 1994, pp. 97–101.

[9] P. F. Sturm and S. J. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, pp. 432–437, doi: 10.1109/CVPR.1999.786974.

[10] J. Heikkila and O. Silven, "Calibration procedure for short focal length off-the-shelf CCD cameras," in *Proceedings of 13th International Conference on Pattern Recognition*, 1996, pp. 166–170 vol.1, doi: 10.1109/ICPR.1996.546012.

[11] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, doi: 10.1109/CVPR.1997.609468.

[12] Zhengyou Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 666–673 vol.1, doi: 10.1109/ICCV.1999.791289.

[13] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000, doi: 10.1109/34.888718.

[14] O. Bogdan, V. Eckstein, F. Rameau, and J.-C. Bazin, "DeepCalib," in *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production - CVMP '18*, 2018, pp. 1–10, doi: 10.1145/3278471.3278479.

[15] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically Supervised Extrinsic Calibration using 3D Spatial Transformer Networks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1110–1117, doi: 10.1109/IROS.2018.8593693.

[16] P. An *et al.*, "Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondences," *Optics Express*, vol. 28, no. 2, p. 2122, Jan. 2020, doi: 10.1364/OE.381176.

[17] S. A. Rodriguez F., V. Fremont, and P. Bonnifait, "Extrinsic calibration between a multi-layer lidar and a camera," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent*

Systems, Aug. 2008, pp. 214–219, doi: 10.1109/MFI.2008.4648067.

[18] P. Wei, L. Cagle, T. Reza, J. Ball, and J. Gafford, "LiDAR and Camera Detection Fusion in a Real-Time Industrial Multi-Sensor Collision Avoidance System," *Electronics*, vol. 7, no. 6, p. 84, May 2018, doi: 10.3390/electronics7060084.

[19] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D Object Detection with Pointformer," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.11409.

[20] M. Zhu *et al.*, "Single image 3D object detection and pose estimation for grasping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3936–3943, doi: 10.1109/ICRA.2014.6907430.

[21] Y. Wang and J. Ye, "An Overview Of 3D Object Detection," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.15614.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[23] G. D'Emilia and D. Di Gasbarro, "Review of techniques for 2D camera calibration suitable for industrial vision systems," *Journal of Physics: Conference Series*, vol. 841, p. 012030, May 2017, doi: 10.1088/1742-6596/841/1/012030.

[24] F. V. C. HOUGH, "Method and means for recognizing complex patterns," US3069654A, 1962.

[25] E. J. Kirkland, "Bilinear Interpolation," in *Advanced Computing in Electron Microscopy*, Boston, MA: Springer US, 2010, pp. 261–263.

BIOGRAPHIES OF AUTHORS

Xiem HoangVan is the Head of the Department of Robotics Engineering, Faculty of Electronics and Telecommunications, Vietnam National University – University of Engineering and Technology (VNU-UET). He received his Ph.D. degree from Lisbon University, Portugal, in 2015, M.Sc. degree from Sungkyunkwan University, South Korea, in 2011, and B.E degree from Hanoi University of Science and Technology, Vietnam, in 2009, all in Electrical and Computer Engineering. His research interests are machine learning, image, and video communications. Dr. Xiem has published about 50 papers on image and video processing and regularly reviews for many renowned IEEE, IET, and EURASIP journals and serves as a technical committee member for international conferences and funding agencies worldwide. He has received several technical awards for his contributions on image and video coding, including a Best Paper Award from the Picture Coding Symposium 2015 (Australia), a Best Paper Award from the International Workshop on Advanced Image Technology 2018 (Thailand), and a Ph.D. award from the Fraunhofer Portugal Challenge 2015, and an Outstanding reviewer award of the Elsevier Journal of Signal Processing: Image Communication. Dr. Xiem is a recipient of the prestigious Golden Globe Award for Young Scientists (under 35 years old) in Science and Technology 2019 and the VNU Top young scientist award 2019. He can be contacted at email: xiemhoang@vnu.edu.vn.

Nam Do is a undergrad research assistant at AI Robotics Laboratory from University of Engineering and Technology, Vietnam National University since 2020. His researches are in fields of artificial intelligence, machine learning, image and language processing. He can be contacted at email: donam.2801@gmail.com.