3D Localization of an Object Using a Monocular Camera

Thisali S Rathnayake
Department of Electrical Engineering
University of Moratuwa
Moratuwa, 10400, Sri Lanka
thisalisrathnayake@gmail.com

D Kasun Prasanga
Department of Electrical Engineering
University of Moratuwa
Moratuwa, 10400, Sri Lanka
prasanga@ieee.org

A.M. Harsha S. Abeykoon
Department of Electrical Engineering
University of Moratuwa
Moratuwa, 10400, Sri Lanka
harsha@uom.lk

***Abstract*—Three-dimensional(3D) localization plays a crucial role in numerous computer vision applications. While 3D localization traditionally relied on specialized hardware setups or multiple cameras, recent advancements have explored the potential of monocular cameras for achieving 3D localization. This research paper investigates and develops techniques for three-dimensional (3D) localization using a monocular camera on a 3D space. By leveraging the principles of geometrical method, particularly triangulation, the study aims to achieve accurate 3D localization. A red LED bulb is used as the object to be localized. Hence the proposed approach utilizes color thresholding for establishing correspondences between multiple images. Extensive experiments are conducted using an industrial robot arm to validate the developed algorithms, evaluating their accuracy against known 3D positions. The outcomes of this research provide insights on the change of accuracy in 3D localization using a geometrical method, for various camera positions.**

***Index Terms*—3D localization, Monocular vision, Triangulation, Depth Estimation, Color Thresholding**

I. Introduction

One of the most discussed key problems in the computer vision is the 3D localization of objects using two dimensional images of the scene. Similar to the cameras, the human eyes also only receive two-dimensional perception of the world. However, in everyday lives, humans effortlessly operate in the three-dimensional world. The ability to interpret the three-dimensional world using the two-dimensional projections of the scene comes naturally to the human brain. Replicating this seemingly easy process has been proven to be quite challenging over the years [1].

In order to solve this problem different types of camera systems have been used. Monocular camera systems, stereo camera systems and multiple camera systems are used in several studies for object localization in three-dimensional space. Losada et al. [2] used a multi camera system fixed in the space and synchronized with each other to localize a mobile robot in the given space. The three-dimensional position of the mobile robots within the given environment is determined by the information taken by the multi camera sensor. Kim and Choong Yow proposed using a monocular camera system mounted on a drone for depth estimation purposes [3]. Stereo camera system is the senser system that is most frequently utilized camera vision systems. In this system two cameras separated by a baseline are used.

The reason for the popularity of the stereo systems in the depth estimation applications is the ease of use and comparatively high accuracy of the estimated depth. Depth information can be obtained by attaining the disparity of the corresponding points of left and right images [4]. However, there are several downsides of using two or multiple camera systems compared to using a single camera. Increased pay load, space constraints, camera placement are a few things to be concerned when using multiple camera systems [3].

When using a monocular camera system, the depth information which is inherent to stereo camera system is lost. Hence obtaining depth perception has proven to be a challenging task in monocular camera systems. With the development of Artificial Intelligence several approaches have been proposed as a solution. Muslikhin et al. [5] introduced a method based on both k-nearest neighbors (kNN) and the fuzzy inference system (FIS) to localize junction boxes in three-dimensional space. Another approach discussed by Saxena and Jamie [6] is incorporating monocular cues such as texture variations and gradients, color, etc.- into stereo systems. This is done using Markov Random Field (MRF) learning algorithm. Leitner et al. [7] tackle the problem of localizing objects from vision system on a humanoid robot. They compare the performance of the approaches Artificial Neural Network (ANN) and Genetic Programming. The methods which uses learning based methods can capture complex scenes but the accuracy highly relies on the quality of the training data set.

Although the AI techniques are quickly gaining popularity the fundamental methodology of 3D localization is the geometric approach. It is based on the principles of triangulation method. Our approach for localizing an object on the three-dimensional plane utilized distance-based triangulation method. This is typically done using a stereo camera system. Building on [8], in this paper we discuss how triangulation can be used to obtain the depth perception for a single camera system. It can effectively handle scenes with depth discontinuities and occlusions and provide accurate depth information for textured and well-calibrated scenes. However, handling varying lighting conditions and dynamic scenes can be difficult using this method. During this experiment localization of a static object using a moving camera is attempted. Furthermore, the accuracy of the estimated depth values is presented using

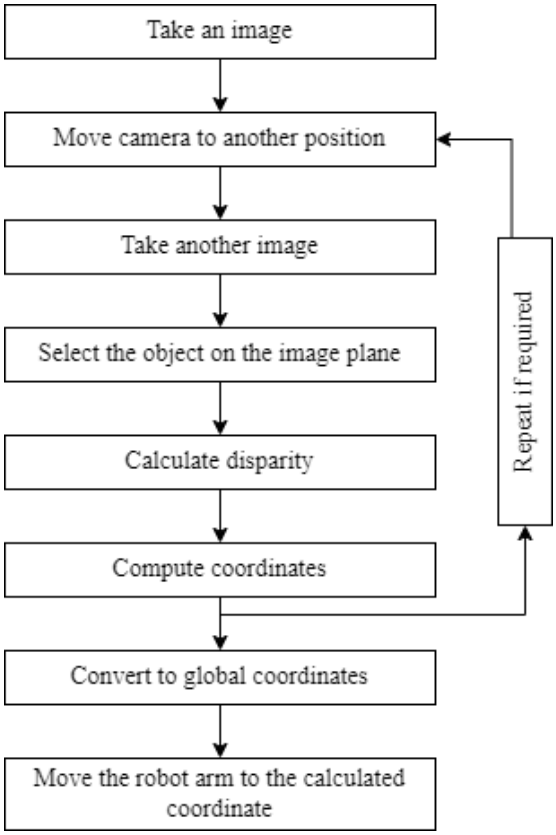

Fig. 1. Process of the proposed model.

the experimental data set.

II. Proposed Methodology

In this proposed method, for the purpose of obtaining the two-dimensional projection of the scene a simple RGB camera is required. For this study, a Raspberry PI camera module V2 is used and in the place of the object a red LED bulb is used. A red LED bulb was chosen as the object considering the ease of isolating the object on the image plane using simple color thresholding techniques.

In this approach the conventional stereo camera system has been replaced by a single moving camera which allows to take the pictures from different positions and angles. In stereo vision the cameras are fixed. Hence the relative position and the orientation of the cameras do not change. Thus, it has a fixed baseline distance between the cameras. The baseline is inversely proportional to disparity between two images [4]. Hence the baseline distance affects the accuracy of the calculated depth. In contrast, the relative position and the orientation can be varied in this proposed model to achieve an accurate depth estimation. The change in relative position and the orientation of the two camera positions also poses a challenge to the calculations of the geometrical method.

A. System Overview

Overview of the proposed approach is illustrated in figure 1. The approach consists of three main steps: taking images from two different positions, stereo correspondence and disparity calculation, depth calculation. in order to validate the proposed method, a 'KUKA KR 6 R900 sixx' robot arm manipulator is used. The robot arm manipulator utilized in this study consists of six degrees of freedom. The camera is attached on the end effector of KUKA KR 6 R900 sixx. The camera is fixed in the same orientation as that of the end effector. Therefore, the orientation and the position of the camera can be obtained. Using the robot arm the images of the scene can be taken from different positions and orientations as required. The position of the object can be calculated with respect to camera coordinate frame using those images. The calculated coordinates can be translated and get the coordinates of the object with respect to the Robot base frame as shown in figure 2. Thus, the robot arm can be driven to the calculated position. This can be used to validate the calculated position.

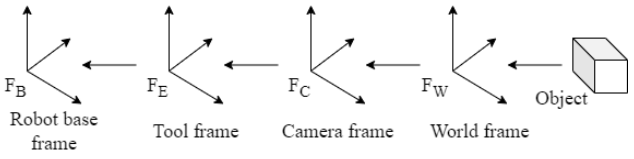

Fig. 2. Relationship between different coordinate frames.

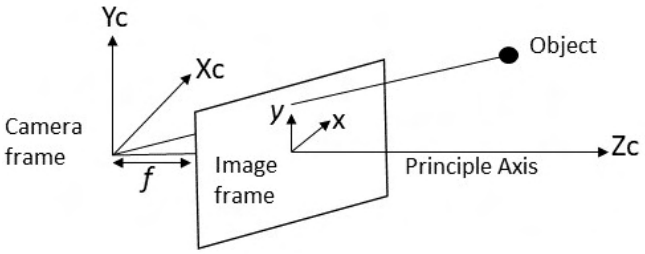

Fig. 3. Pin hole camera model.

B. Camera Model

For this approach a pin hole camera model must be used to capture the images. A pin hole camera model consists of a light-tight box with a tiny aperture on one side. Pin hole camera sets apart from other cameras due to its absence of lens. In this study, a Raspberry camera has been modelled as a pin hole camera for convenience. Cameras typically have different distortions due to lens imperfections. The projection of the 3D scene on the pin hole camera can be model can be mathematically modelled as shown in figure 3.

In figure 3 the object in 3D world is projected onto the image plane. The object coordinates with respect to the camera coordinates (X_C, Y_C, Z_C) can be derived from the object coordinates with respect to the world coordinates (X_W, Y_W, Z_W). This transformation is done using the equation (1) [8].

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} {R_w}^c & {T_w}^c \\ {1_{1x3}}^T & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (1)$$

Equation 1 can be expressed as below.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (2)$$

Here in equation 1 the rotational matrix and the translation matrix of the camera coordinate frame with respect to the world coordinate frame is expressed as R and T respectively. Here the camera coordinate frame is centered on the camera lens. As shown in the equation 2 world coordinates are transformed into camera coordinates using the extrinsic matrix. The camera coordinates can be transformed into the 2D coordinates on the image plane by using the intrinsic matrix [9].

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (3)$$

Intrinsic matrix expresses the internal parameters of the camera. Here the f_x, f_y are the focal lengths of the camera in the x and y directions respectively. C_x , C_y are the principle points of the image plane. It gives the point of intersection between the image plane and the optical axis. The equation 3 gives a mathematical model of 2D projection of the 3D world.

C. Camera Calibration

The aim of camera calibration is to accurately estimate the intrinsic and extrinsic parameters of the camera to correct for distortions and accurately project 3D points onto the image plane. Since the camera used has lenses unlike in the pin hole model camera the images may have distortions. The most significant distortions in images are radial distortion and tangential distortion [9]. With proper calibration, the captured images can be redeemed from such distortions. If not, it can affect the accuracy of measurements and the quality of localizing algorithm.

The key objective of camera calibration is to estimate the camera's intrinsic matrix. The focal length determines the scale of the image, the principal point represents the optical center of the camera, and the distortion factors account for any non-linear deformations introduced by the lens.

In this study a checkerboard pattern is used as the calibration target. It is the most commonly used target. A detection algorithm is used to identify the corners of the squares. The corners of the checkerboard pattern are infinitely small and consistent against lens distortions. Therefore, the corners detection is done up to a sufficient accuracy [10].

In this experiment the camera calibration is done with the use of OpenCV library is used. Since it offers a variety of computer vision algorithms, it can be used in many computer vision tasks such as camera calibration effectively [11]. To calibrate the camera a number of photos of the checkerboard is taken covering different angles and orientations. Then it is processed to detect the corners in the checkerboard. Here

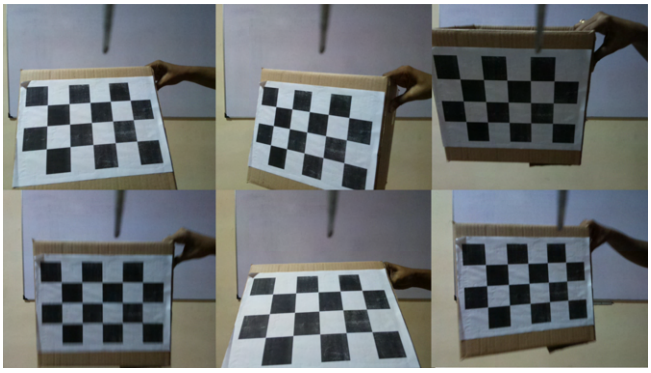

Fig. 4. Captures of the checkerboard used for Raspberry pi camera calibration.

Fig. 5. Corner points detection of the checkerboard.

the correspondence between the image and the object is established. The cv2.calibrateCamera() function is then used to estimate the camera parameters, adapting Zhang's algorithm. This function uses objects and corresponding image points detected in the earlier step. As a result, this gives the intrinsic matrix and distortion factors.

Following the above method, the intrinsic matrix of the camera was found as,

$$\begin{bmatrix} 499.8846516 & 0 & 318.9251849 \\ 0 & 502.5206681 & 243.8386911 \\ 0 & 0 & 1 \end{bmatrix}$$

It also gives the radial distortion coefficients (k_1, k_2, k_3) and tangential distortion coefficients (p_1, p_2). All the experimental results for the camera calibration are mentioned in the table I

D. Object Selection

In this study a red LED bulb is used. Since the object of interest has a distinctive color the object on the image plane can be selected using a color thresholding method. Color thresholding is a simple yet effective method that involves segmenting objects based on their color information. The output of the camera is an RGB image. RGB image produce color using a combination of Red, Green, Blue color. This indicates that each pixel in the captured image possesses values for Red, Green, Blue ranging from 0 to 255 depending on the intensity of the color [12].

Due to the ease of color separation the RGB color space is converted into the HSV color space. Here color information previously mentioned as RGB values in separated into three components: Hue, Saturation, and Value. Hue represents the dominant color information. It describes the type of color, such as red, green, blue, yellow, etc. Saturation represents the

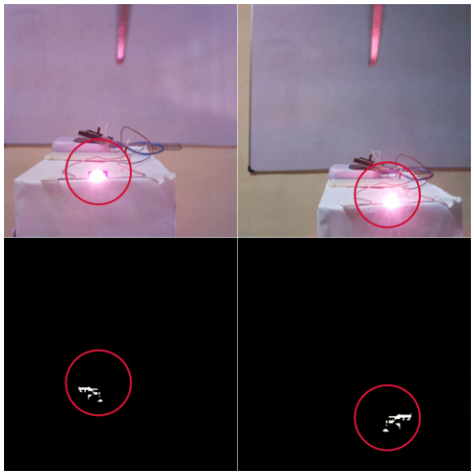

Fig. 6. Captured images of the object and the color thresholding without removing contours.

TABLE I
RESULTS OF CAMERA CALIBRATION

Parameter	Result
f_x	499.885
f_y	502.521
c_x	318.925
c_y	243.838
k_1	0.174
k_2	-0.476
k_3	-0.003
p_1	0.011
p_2	0.191

intensity of a color. This can also be described as the vividness or vibrancy of the color. Value represents the brightness of the color where the lowest value applicable for V corresponds to black color. Each of the three can take any integer ranging from 0 to 255. This method is less sensitive to the variations of the light making it more robust compared to the RGB color space [13].

For the image processing and selecting the object, OpenCV library was used. In order to convert the RGB color space into HSV color space "cv2.cvtColor(image, cv2.COLOR_BGR2HSV)" function was used. Then a upper and lower threshold for the selected color is set. Using these threshold values, the image is filtered as shown in 6 These filtered images may have noises. To avoid the noises in the filtered image the area of a contour is calculated. Areas below a set value are considered as noise and they are filtered out to separate the object of interest and figure 7. Then the pixel coordinates of the center of the object of interest can be easily obtained using the filtered image.

E. Depth Estimation

Using the above discussed methods, we were able to obtain the corresponding of the object on the image plane. These coordinates are used to estimate the depth of the object of interest. Our approach to depth estimation in 3D plane was a

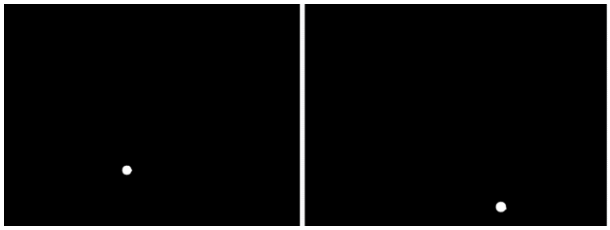

Fig. 7. Isolated of object of interest after filtering contours based on area.

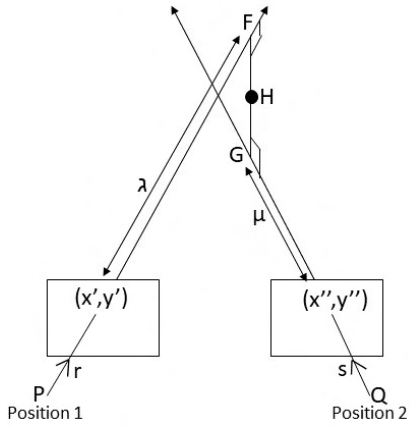

Fig. 8. Geometrical method for depth estimation.

geometrical method to precisely calculate the depth. As shown in figure 8 the camera will capture an image from the position 1 and the move to the position 2. Both images from position 1 and position 2 will be processed.

To get the depth of the object the lines of sight of the camera to object from two positions are drawn as shown in figure 8. The crossing point of two lines is the position of the object in the 3D world. Hence the position of object on the 2 lines is taken as F and G respectively. The midpoint of the shortest distance between the 2 lines is considered as the actual position of the object. The equations for the 2 lines can be written as given in the equation 4 and 5.

$$F = P + \lambda.r \tag{4}$$

$$G = Q + \mu.s \tag{5}$$

Here P and Q denote the camera position in the world coordinates(coord.). r and s are the unit vectors along the two lines. Here λ and μ are unknowns. Hence to calculate the F and G, λ and μ must be found. The unit vectors along the two lines, r and s can be found using the equation.

$$r = (R')^{\mathrm{T}}. \begin{bmatrix} x' \\ y' \\ c \end{bmatrix} \tag{6}$$

$$s = (R'')^{\mathrm{T}}. \begin{bmatrix} x'' \\ y'' \\ c \end{bmatrix} \tag{7}$$

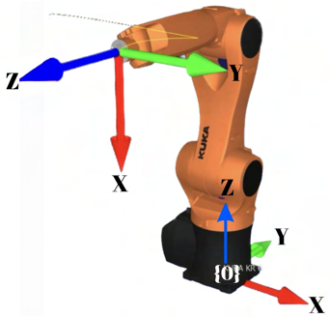

Fig. 9. Reference coordinate frame for calculations.

TABLE II
ERROR PERCENTAGE ON THE CALCULATED COORDINATES WHEN THE OBJECT IS PLACED IN DIFFERENT POSITIONS

	Image	Calculated Coord.	Actual Coord.	Error Percentage(%)
x	1	7.70	-22.20	-134.60
	2	223.40	194.80	14.65
	3	51.40	28.10	82.71
y	1	-812.90	-838.80	-3.08
	2	-803.20	-780.00	2.97
	3	-794.50	-792.30	0.27
z	1	813.50	833.80	-2.43
	2	818.10	818.50	-0.04
	3	777.10	757.10	2.64

Here the R' and R" are the rotational matrices of the camera in position 1 and position 2 respectively. Pixel coordinates of the corresponding points is obtained with respect to principle point as (x', y') and (x", y") respectively. Focal length calculated during the camera calibration is denoted as c.

Since FG $\perp PF$,

$$(P + \lambda.r - (Q + \mu.s)).r = 0 \tag{8}$$

Likewise, FG $\perp QG$,

$$(P + \lambda.r - (Q + \mu.s)).s = 0 \tag{9}$$

P, Q are the relative positions of the camera. The equation (8), (9) can be rephrased as below.

$$\begin{bmatrix} r^{\mathrm{T}}.r - s^{\mathrm{T}}.r \\ r^{\mathrm{T}}.s - s^{\mathrm{T}}.s \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} r_{\mathrm{T}}.r - s^{\mathrm{T}}.r \\ r_{\mathrm{T}}.s - s^{\mathrm{T}}.s \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} \tag{10}$$

By solving the equation (10) the values for λ and μ can be calculated accordingly. By getting the middle point between the F and G the coordinates of the object on the 3D plane is obtained.

$$H = \frac{F + G}{2} \tag{11}$$

III. EXPERIMENTAL RESULTS

In order to validate our purposed algorithm, experimental results are taken. Here the camera is mounted on the end effector of the industrial robot arm 'KUKA KR 6 R900'. Using the robot arm the camera is moved. The position and orientation of the camera can be taken through the robot arm. After calculating the object coordinates with respect to the camera coordinates it is converted into the robot base coordinate frame. All the results for analysis are taken with respect to the robot base coordinate frame.

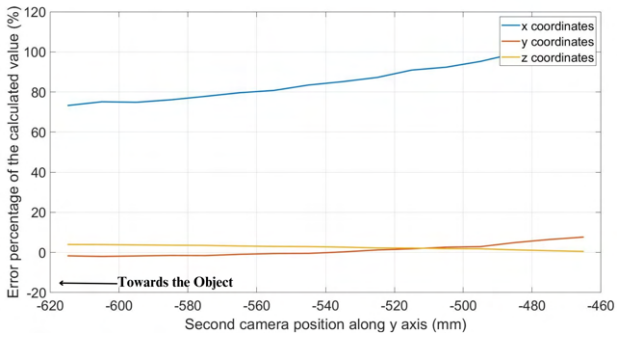

Fig. 10. The error percentage when the camera position changes along Y axis

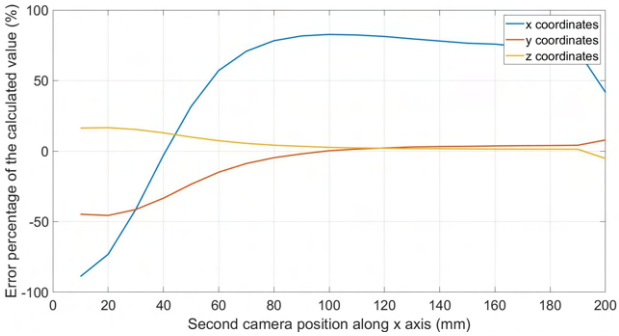

Fig. 11. The error percentage when the camera position changes along X axis

To validate the calculated coordinates, the LED bulb was placed upon different places and the 3D coordinates were calculated as shown in table II.

As shown in the table II the error of the calculated coordinates is very small. Likewise, the error percentage when the camera position changes with respect to the first camera position can be analyzed as well.

As observed in the figure 10 the error percentage of the depth calculation (y coordinate) decreases when camera is moved with respect to the first camera position, along the y axis towards the object. After a certain point the error starts to increase.

As observed in the figure 10 the error percentages of the depth calculation (y coordinate) as well as the x and z coordinates increase when the displacement between two cameras are decreased below a certain value(100mm). When the displacement of camera position decreases the disparity between the two images also decreases as shown in figure 12. When the disparity decreases the error percentage is increased.

The robot can be moved in the path towards the object along the path shown in figure 13. When the camera approaches

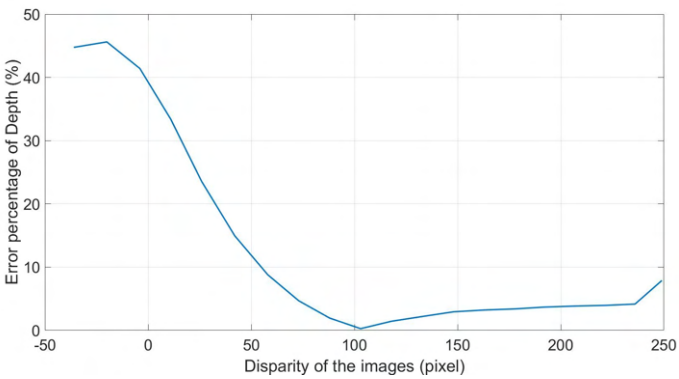

Fig. 12. The error percentage of depth with disparity of the images

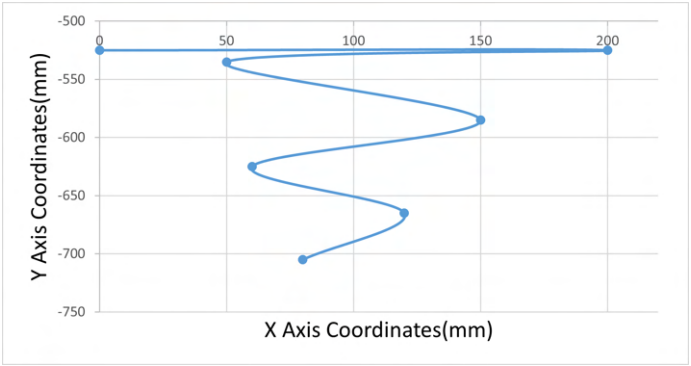

Fig. 13. Robot moving path towards the object.

towards the object the error percentage increases as shown in table III.

IV. Conclusion

This study aimed to achieve three-dimensional (3D) localization using a monocular camera, in contrast to traditional methods that rely on stereo camera systems, specifically targeting the localization of a red LED bulb. The proposed approach was adapts a geometrical method based on mathematical principles and algorithms to accurately calculate the position of the object on 3D plane. Main steps involved were selecting the object on the image plane, performing 3D localization and manipulating a robot arm to reach the localized object. The study utilizes the 'KUKA KR 6 R900' industrial robot arm for camera movement and validation of the calculated 3D coordinates. Analysis of the results reveals that reducing the camera displacement along the x-axis increases the error in 3D coordinate calculation, while increasing displacement along the y-axis(towards the object) leads to increased depth error. Additionally, when moving the camera in a zigzag path towards the object the error in calculated coordinates were increased. In this method the accuracy of the results highly relies on the calibration data and the lighting conditions of the scene.

TABLE III
Error percentage on the calculated coordinates when the camera approaches towards the object

Camera position(mm)			Error percentage(%)		
X	***Y***	***Z***	***X***	***Y***	***Z***
200	-525	890	81.085	3.686	4.199
50	-535	890	187.438	0.535	6.171
150	-585	890	190.920	0.937	5.201
60	-625	890	837.518	95.293	20.625
120	-665	890	292.986	9.070	0.921
80	-705	890	49.414	12.297	6.918

Verifying the robustness of the algorithm using more challenging scenarios and integrating learning methods to increase the accuracy is to be considered for future work.

V. Acknowledgement

The authors express their sincere gratitude for the assistance received from the Senate Research Committee, University of Moratuwa,Sri Lanka.

References

[1] T. Jebara, A. Azarbayejani, and A. Pentland, "3d structure from 2d motion," *IEEE Signal processing magazine*, vol. 16, no. 3, pp. 66–84, 1999.

[2] C. Losada, M. Mazo, S. Palazuelos, D. Pizarro, and M. Marrón, "Multi-camera sensor system for 3d segmentation and localization of multiple mobile robots," *Sensors*, vol. 10, no. 4, pp. 3261–3279, 2010.

[3] I. Kim and K. C. Yow, "Object location estimation from a single flying camera," *UBICOMM 2015*, p. 95, 2015.

[4] N. Sombekke and A. Visser, "Triangulation for depth estimation," 2022.

[5] J.-R. Horng, S.-Y. Yang, M.-S. Wang *et al.*, "Object localization and depth estimation for eye-in-hand manipulator using mono camera," *IEEE Access*, vol. 8, pp. 121 765–121 779, 2020.

[6] A. Saxena, J. Schulte, A. Y. Ng *et al.*, "Depth estimation using monocular and stereo cues." in *IJCAI*, vol. 7, 2007, pp. 2197–2203.

[7] J. Leitner, S. Harding, M. Frank, A. Förster, and J. Schmidhuber, "Learning spatial object localization from vision on a humanoid robot," *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 243, 2012.

[8] C. Doignon, G. Abba, and E. Ostertag, "Recognition and localization of solid objects by a monocular vision system for robotic tasks," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 3. IEEE, 1994, pp. 2007–2014.

[9] E. Ekanayake, T. Thelasingha, U. Udugama, G. Godaliyadda, M. Ekanayake, B. Samaranayake, and J. Wijayakulasooriya, "Object dimension extraction for environment mapping with low cost cameras fused with laser ranging," *arXiv preprint arXiv:2302.01387*, 2023.

[10] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access*, vol. 9, pp. 113 143–113 155, 2021.

[11] Y. Wang, Y. Li, and J. Zheng, "A camera calibration technique based on opencv," in *The 3rd International Conference on Information Sciences and Interaction Sciences*. IEEE, 2010, pp. 403–406.

[12] S. P. Chapala and M. Dharmapriya, "Color detection of rgb images using python and opencv."

[13] C.-H. Chen, H.-P. Huang, and S.-Y. Lo, "Stereo-based 3d localization for grasping known objects with a robotic arm system," in *2011 9th World Congress on Intelligent Control and Automation*. IEEE, 2011, pp. 309–314.