AUTOMATIC ASSEMBLY OF INTERLOCKING BUILDING CUBES BY USING VISUAL SERVOING APPROACH

Zhe Zhi Chew[1,*], Kok Seng Eu[3], Tsung Heng Chiew[2], Yoon Ket Lee[2], Xian Zhu Lim[2], Kian Meng Yap[1], Julian Kok Ping Tan[3] and Yee Wai Sim[3]

Received: February 17, 2023; Revised: January 04, 2024; Accepted: January 16, 2024

Abstract

Most of the robotic arms are set up for operation by the teach-and-repeat (T&R) technique, also known as coordinate-based control, using pre-recorded coordination values to generate robotic arm trajectories. This technique is inefficient because it requires teaching a robotic arm every single coordinate point to complete an operation. It is a manual, tedious and time-consuming task. Moreover, this task needs to be repeated if there are slight changes in the operation. This paper studied the visual servoing approach to replace the T&R technique in assembling different interlocking building cubes (IBC) combinations. The proposed method uses an image-based visual servo (IBVS) with a dynamic look-and-move system, and the eye-to-hand system, which requires computing the incremental joint angles of the robotic arm, a novel kinematic formula with a combination of Trigonometry formulas has been proposed in this paper. The experimental results show the feasibility of using a visual servoing approach to complete the assembly tasks without relying on the T&R technique and achieved an average success rate of 90% and an average of 40.2 seconds for task duration. Thus, this eliminates the tedious and time-consuming tasks of teaching the coordinate points to the robotic arms. Further improvement on the servo motor, camera and computer processor can be done to increase the accuracy, computation time and task duration in future.

Keywords: **Robotic Arms; Kinematics Formulas; Automatic Assembly; Visual Servoing**

---

[1] *HUMAC, Department of Computing and Information Systems, School of Engineering and Technology, Sunway University, Bandar Sunway, Malaysia. E-mail: 18033449@imail.sunway.edu.my*
[2] *Faculty of Engineering and Technology, Tunku Abdul Rahman University of Management and Technology, Kuala Lumpur, Malaysia.*
[3] *School of Engineering, Faculty of Innovation & Technology, Taylor's University, Subang Jaya Malaysia.*
[*] *Corresponding author*

DOI: https://doi.org/10.55766/sujst-2024-01-e0462

Suranaree J. Sci. Technol. 31(1):010274(1-10)

Introduction

Robotic arms have been widely used in various industries to perform repetitive tasks such as pick and place, assembling, welding, spray painting, and cutting. These repetitive tasks are programmed through a handheld unit teach pendant using the teach-and-repeat (T&R) technique, also known as coordinate- based control (Gao *et al*., 2014).

Operators use the tech pendant to move the robotic arm to the desired coordinate point and record it by repeating this process from one point to another until all coordinate points are recorded in the memory. The recorded coordinate points are replayed repeatedly to generate the robotic arm's trajectory path for repetitive tasks. Most robotic arm manufacturers, such as Kuka, Honda, ABB, Epson, Omron, Fanuc, etc., still use the T&R technique, albeit the limitations, e.g., inflexibility and poor efficiency (Ogawa and Kanada, 2010; Smith *et al*., 2012). The setup of the T&R technique is a manual, tedious, and time-consuming task because it requires a lot of time to manually control the robotic arm using the teach pendant, with multiple trails and tests to ensure the correct coordinate points are recorded for the repetitive tasks. Furthermore, a well-trained operator must use the teach pendant to control and manipulate a six-degree-of-freedom robotic arm. This incurs a cost to train a skillful operator. Moreover, this technique is inflexible and cannot perform in a dynamic environment. If there are any changes in the operation, the teaching process (re-recording all the coordinate points) of the T&R technique must be redone.

The visual servoing approach is proposed to overcome this problem. There are two main groups of visual servoing approach: position-based visual servo (PBVS) and image-based visual servo (IBVS) (Malis *et al.*, 1998; Chaumette and Hutchinson, 2006; Hafez *et al*., 2008; Kazemi *et al*., 2009; Palmieri *et al*., 2012; Sun *et al*., 2018). PBVS uses vision data to detect the pixel coordinate of a target object and convert it into the Cartesian space coordinate location of the robotic arm. For instance, the pixel-X and -Y values of the target object in the image are transformed into the Cartesian space coordinate location in X- and Y-axis values (coordinate point). This requires solving the kinematics problem to map the Cartesian frame of the camera to the Cartesian frame of the robotic arm using transformation operations, which involves the matrix's combination of translation and rotation operations. PBVS is sensitive to calibration and modelling errors. If mapping the Cartesian frames between the camera and the robotic arm is misconfigured, it could affect the overall performance of PBVS in terms of accuracy.

On the other hand, IBVS uses image features, i.e., the characteristic information observed by the camera that can directly give feedback to the controller to find the movement transformation between the end-effector of the robotic arm and the coordinate of the target object (Kazemi *et al*., 2010). As a result, IBVS is less sensitive to calibration and modelling errors. However, it has a global convergence problem; it is unable to converge the entire working space of the robotic arm. Furthermore, the absence of direct control between the camera and robotic arm motion induced by the IBVS control law causes infeasible manoeuvres simply due to the joint limit of the robotic arm (Kazemi *et al*.,2010; Sun *et al*.,2018).

The control type of visual servo systems can be classified into dynamic look-and-move systems or direct visual servo systems (Hutchinson *et al*., 1996; Palmieri *et al*., 2012). The former uses the vision system as an outer control feedback loop to stabilize the system and the servo system as an inner control feedback loop to control the robot's movement. The latter uses the vision system directly in the only control loop to compute the robotic arm's movement and stabilize it simultaneously. Most application systems belong to the dynamic look-and-move systems because of the variance loop rate between vision and servo systems.

The camera setup can also be classified based on their mounting positions, such as the eye-in-hand and eye-to-hand systems (Flandin *et al*., 2000; Kelly *et al*., 2000; Kulpate *et al*., 2005; Sun *et al*., 2018;). In the former, the camera is mounted on the wrist of the robot to observe the target object only. In the latter, the camera is mounted in a fixed position in the working space of the robotic arm. Thus, the camera can observe the target object and the end-effector of the robotic arm. The former has a higher computational consumption than the latter because the Cartesian frame of the camera varies over time during computation for each loop rate. The former may have inconsistent camera lighting issues because the view changes over time. In contrast, the latter does not have this problem because the camera is in a fixed position with consistent lighting.

This study examines the visual servoing to replace the T&R technique in assembling different combinations of Interlocking Building Cubes (IBC), also known as Lego bricks. The proposed solution uses a combination of IBVS, a dynamic look-and-move system, and the eye-to-hand system. Section 2 discusses the justification for using this combination.

Methodology

The experimental setup for assembling different IBC combinations involves four degrees of freedom (DOF) robotic arm with a cube space coverage of 30cm×30cm×30cm dimension, two webcams with 640x480 pixels, a microcontroller Arduino Uno with servo motor controller shield, and a laptop to run image processing library OpenCV in Python programming language. Webcam 1 is mounted side by side at the robotic arm to observe the front view of the robotic arm's working space. Webcam 2 is mounted on top of the working space, as shown in Figure 1.

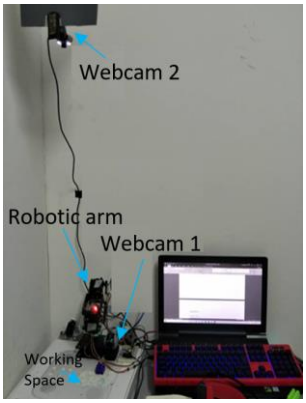

Figure 1. Experimental setup

Figure 2 shows the logic flow of the proposed method, consisting of three major steps. First, webcam 1 captures a random combination of IBCs and sends the image to the part attachment analysis (PAA) to understand the part attachment relationship between each IBC. The part attachment relationship is stored as a mapping matrix for further action. Second, the rebuild sequence analysis and action (RSAA) uses the mapping matrix to plan the movement sequence of the robotic arm to rebuild the combination of IBC. Lastly, the visual servoing function combines IBVS, dynamic look-and-move system, and eye-to-hand system, which performs the pick-and-plan tasks according to the movement sequence planned by RSAA. The details of each step are discussed in the following sub-sections. Figure 3 summarises the step-by-step procedure of how the project is carried out.

Part Attachment Analysis (PAA)

This function analyses the part attachment relationship between each IBC. The information extracted from the IBC combination includes colours, size, and the relative position of the IBC. As such, PAA is designed based on the grid-based analysis approach that extracts the necessary information of each IBC into a tracking code and stores it in a grid-based format. The first step of PAA is to detect the colour and contour of IBC in pixel coordination. The second step is calculating the length of the IBC. Lastly, colour and length information is converted into a tracking code and stored in the grid-based format.

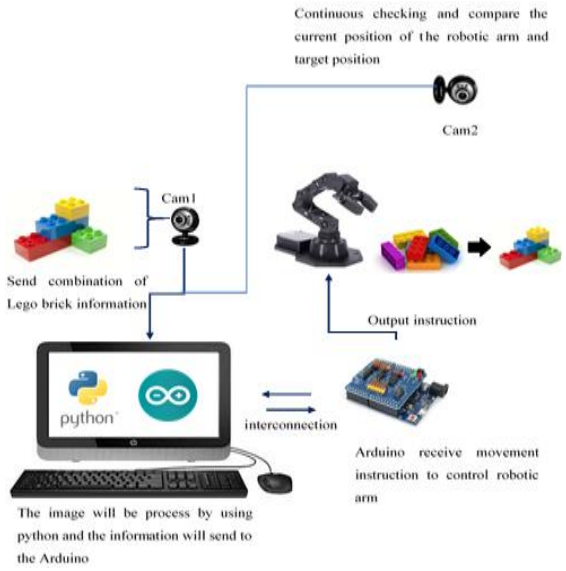

Figure 2. Proposed method

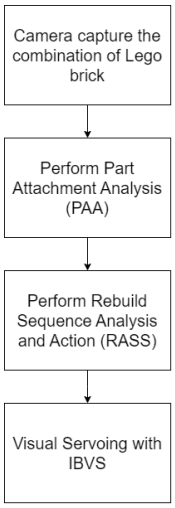

Figure 3. Summary of the steps

Rebuild Sequence Analysis and Action (RSAA)

This function rebuilds the IBC combination based on the tracking codes provided by the PAA function. Figure 4 shows the steps in RSAA. The rebuild action starts from the bottom row and most right-hand side of the IBC that is stored in the grid-based tracking code to perform the pick-and-place task and then proceed to the left for the next IBC until the first row is completed. Subsequently, it incrementally increases the row number, for example, moving to the second row from the bottom row and rebuilding from the most right to the left side until the second row is completed. This process continues until all the rows are covered, rebuilding the IBC combination.

Visual Servoing with IBVS, dynamic look-and-move system, and the eye-to-hand system

Visual servoing is applied in the pick-and-place task in the RSAA. This paper uses IBVS and the dynamic look-and-move system because of the ease of implementation in the continuous control process. Likewise, the eye-to-hand system is adopted to minimize computation consumption because it does not involve complex mathematical solving processes like solving inverse kinematics calculations (Mahanta *et al*., 2019). In addition, this paper aims to develop a low-cost solution for the automatic assembly of IBCs. Therefore, IBVS, the dynamic look-and-move system, and the eye-to-hand system are compatible with this objective.

Webcam 2 observes the top view of the working space to detect the position of the target objects (IBCs) and the end-effector of the robotic arm, which is mounted with a red LED light (see Figure 5), allowing the position detection of the end-effector of the robotic arm. The top view image indicates the relative positions between the target IBC and the end-effector of the robotic arm. The zero point (Px=0, Py=0) is at the top right corner. The relative positions can be derived into four movement directions, as shown in Table 1. For instance, the LED light is located at (Px=50, Py=50) and the target IBC at (Px=250, Py=250), as shown in Figure 6. The outputs of the pixel calculations for pixel-X and pixel-Y are negative signs, meaning that the movement directions are to the left and backward for the robotic arm to get the target IBC. The way the dynamic look-and-move system is implemented in this study, the movement directions are broken down into incremental steps. Each increment makes the robotic arm one step closer to the target IBC until it reaches the destination.

Table 1. Derivation of the movement directions from the relative positions

Relative Positions	Pixel Sign	Movement direction
The target IBC on the right-hand side of the end-effector of the robotic arm	LED's Px – IBC's Px = +ve	Move to right
The target IBC on the left-hand side of the end-effector of the robotic arm	LED's Px – IBC's Px = -ve	Move to left
The target IBC is in front of the end-effector of the robotic arm	LED's Py – IBC's Py = +ve	Move forward
The target IBC is behind of the end-effector of the robotic arm	LED's Py – IBC's Py = -ve	Move backward

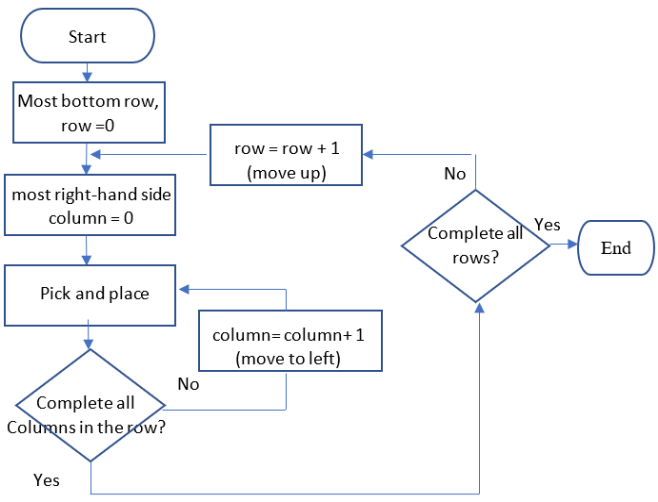

Figure 4. Rebuild Sequence Analysis and Action (RSAA)

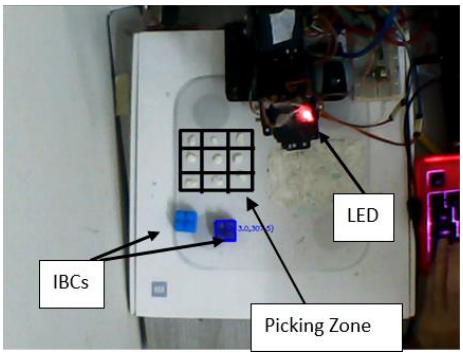

Figure 5. The top view of the working space

Once the incremental movement steps are obtained from the top-view images, these movement directions are converted into the servo motors' rotation angle - the robotic arm's joint angles - for controlling the robotic arm. The system uses IBVS and a dynamic look-and-move system that requires computing the absolute angles for left and right movement and incremental joint angles of the robotic arm for forward and backward movement. Therefore, this study proposes a novel kinematic formula with a combination of Trigonometry formulas. The well-known inverse kinematics (i.e., Denavit - Hartenberg parameters) are not used because they are meant to compute the absolute joint angles of the robotic arm, which are more suitable for PBVS (Ghaniwala *et al*., 2021).

Figure 7 shows an IBC located on the left-hand side and in front of the end effector of the robotic arm. The top view indicates that the end effector of the robotic arm (M4) needs to move incrementally to the left and forward to reach the target IBC. The absolute left movement can be achieved by adding the absolute rotation angle of servo motor M1 from θ_{m1} to θ_t (Tee *et al*., 2013). Likewise, the incremental forward movement can be achieved by adding an incremental extension from L_b to L_t.

Extending L_b involves three variables: rotation angles of servo motors M2, M3, and M4. These variables are determined by the height position (Z-axis) of the target IBC, which can be solved using the trigonometry approach under two sets of conditions. The first condition specifies that the lower range trigonometry calculation (LRTC), i.e., the target Z-axis of the end effector of the robotic arm, must be equal to or lower than the height position of servo motor M2. The second condition specifies that the higher range trigonometry calculation (HRTC), i.e., the target Z-axis of the end effector of the robotic arm, must be higher than the height position of the servo motor M2. The reason for these conditions is that when the end effector of the robotic arm moves into another quadrant range of the designed formula, another trigonometry formula has to be redesigned based on the condition. The target Z-axis of IBC is determined from webcam 1, which can determine which condition to use for calculating L_b.

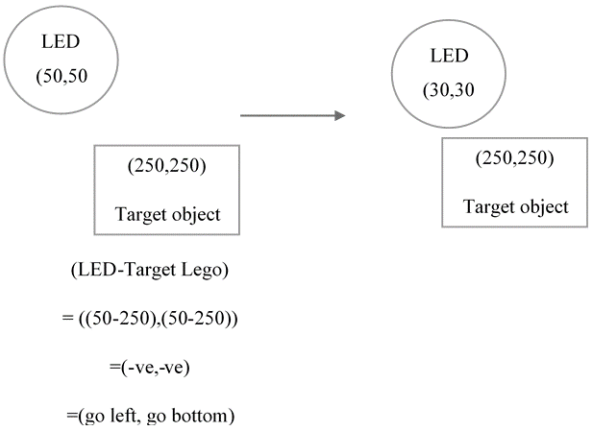

Figure 6. The movement of the end effector based on the relative positions

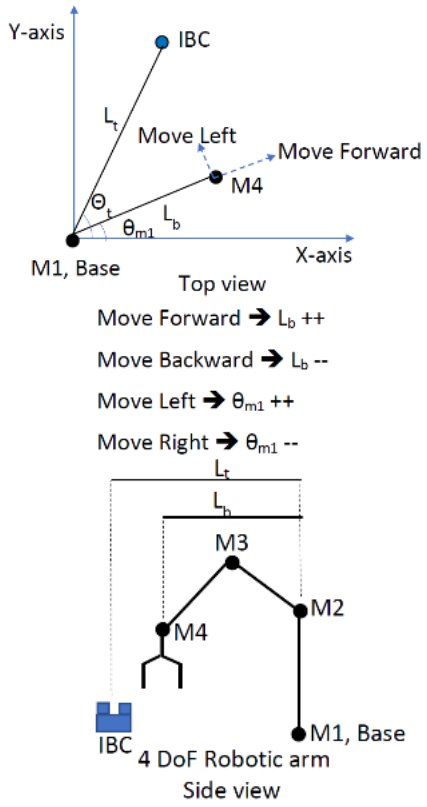

Figure 7. The illustration of the incremental movement step

Lower Range Trigonometry Calculation (LRTC)

Figure 8(a) shows the side view of the robotic arm that complies with the first condition. Figure 8 (b) shows that by changing the side view of the robotic arm into a geometric view to ease the calculation, L_a can be obtained using Equation 1, according to Figure 8 (a). On the other hand, θ_4 can be obtained using Equation 2 based on Figure 8(b). Triangle 1 of Figure 8 (b) shows that it is an obtuse scalene triangle that connects with three sides; two sides of triangle 1 are the linkage of the robotic arm L_2 and L_3, which are constant. The remaining side is L_a, which determines the angle θ_3 using the Pythagorean Theorem as shown in Equation 3. θ_3 can be used to calculate θ_1 by substituting it in Equation 4.

$$L_a = \sqrt{((L_1 - L_4 - z)^2 + (L_b)^2)} \quad (1)$$

$$\theta_4 = \sin^{-1}\left(\frac{L_b}{L_a}\right) \quad (2)$$

$$\theta_3 = \cos^{-1}\frac{(L_3)^2+(L_2)^2-(L_a)^2}{2(L_3)(L_2)} \quad (3)$$

$$\theta_1 = \sin^{-1}\left(\frac{L_3}{L_a}\sin\theta_3\right) \quad (4)$$

A simplified geometry derived from Figure 8(b) is drawn to find θ_6, as shown in Figure 9. The trigonometry rule states that if line 1 and line 2 are parallel and another line intersects them, the angles θ_6 and θ_a are the same. Therefore, since θ_1 has been obtained from Equation 4, Equation 5 can be used to find θ_6.

$$\theta_6 = \theta_a = \theta_1 + 90 \quad (5)$$

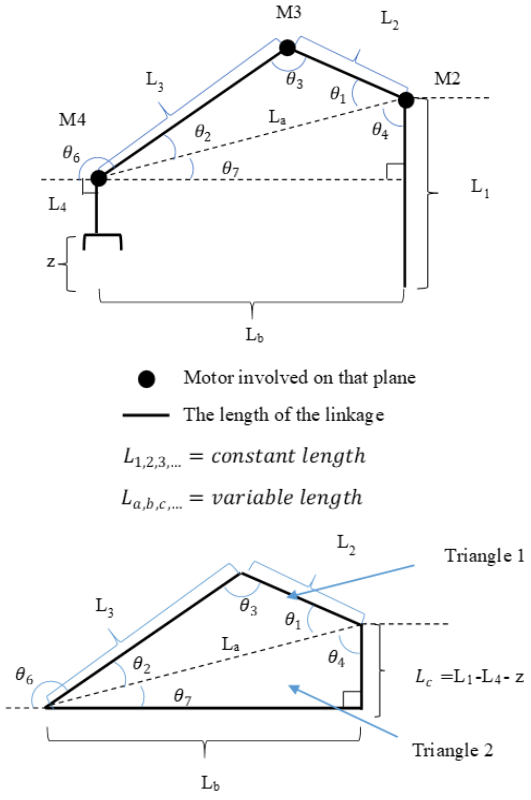

Figure 8. Lower Range Trigonometry Calculation (LRTC)

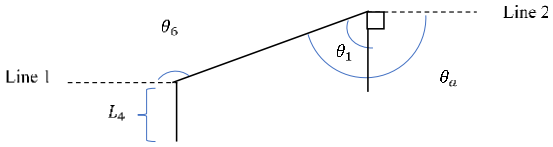

Figure 9. A simplified geometry dimension to find θ_6

With the derivations of these equations, the angles of servo motors M2, M3, and M4 can be determined using Equation 6. The angle of servo motor M2 is the sum of θ_4 and θ_1. θ_3 is equal to θ_{m3}. Notably, when performing the pick-and-place process, the end effector of the robotic arm must face downward. Therefore, a 90° must be added to θ_6 to find θ_{m4}.

Higher Range Trigonometry Calculation (HRTC)

The Z-axis of the end effector of the robotic arm is higher than the height position of servo motor M2 (linkage length. L_1), as shown in Figure 10(a). Figure 10(b) shows the side view conversion of the robotic arm into a geometric view to ease calculation. L_a can be obtained using Equation 7, according to Figure 10(a), whereas θ_4 can be obtained using Equations 8 and 9, based on Figure 10(b). Using the same method as LRTC, both θ_1 and θ_3 can be computed using Equations 3 and 4, respectively. Lastly, the angles of servo motors M2, M3, and M4 can also be found using the same equation of LRTC, i.e., Equation 6.

$$\begin{aligned} \theta_{m2} &= \theta_4 + \theta_1 \\ \theta_{m3} &= \theta_3 \\ \theta_{m4} &= \theta_6 + 90 \end{aligned} \tag{6}$$

$$L_a = \sqrt{((z - L_1)^2 + (L_b)^2)} \tag{7}$$

$$\theta_4 = tan^{-1}\left(\frac{L_c}{L_b}\right) \tag{8}$$

$$L_c = z - L_1 \tag{9}$$

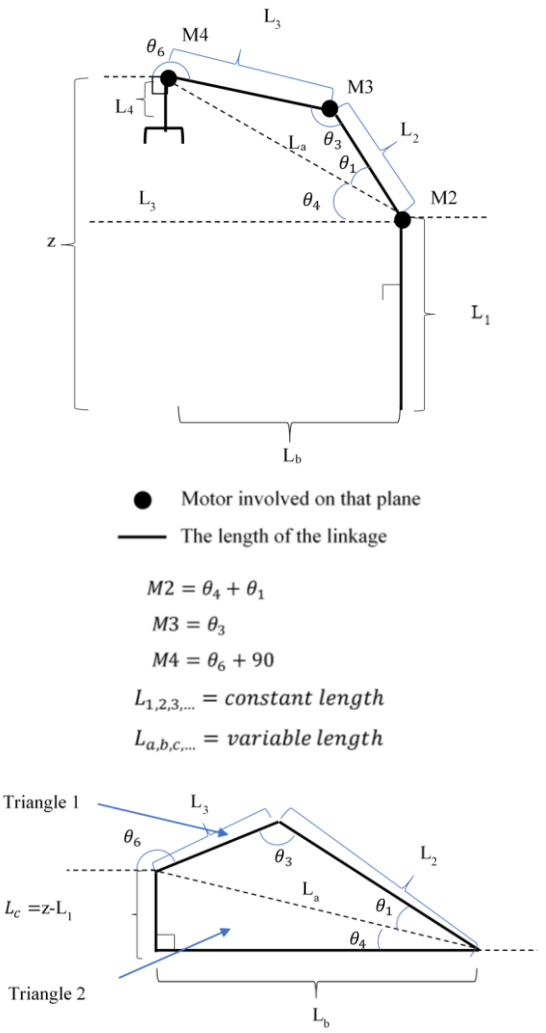

Figure 10. Higher Range Trigonometry Calculation (HRTC)

Results and Analysis

Figure 11 shows the workflow of assembling IBSc using IBVS, dynamic look-and-move system, and eye-to-hand system. Figures 11(a)-(c) show the outcome of PAA. Figures 11(d)-(f) show the RSAA process. Lastly, Figure 11(e) shows the visual seroving process with IBVS, dynamic look-and-move system, and the eye-to-hand system.

In the experiment step, the definition of the IBC accuracy placed from one to another is shown in Figure 12. If one IBC is successfully placed on top of another IBC, it is considered 100%. On the other hand, if the IBC is partially placed on top of another, it is defined as 50% and 25%, respectively, according to Figure 12.

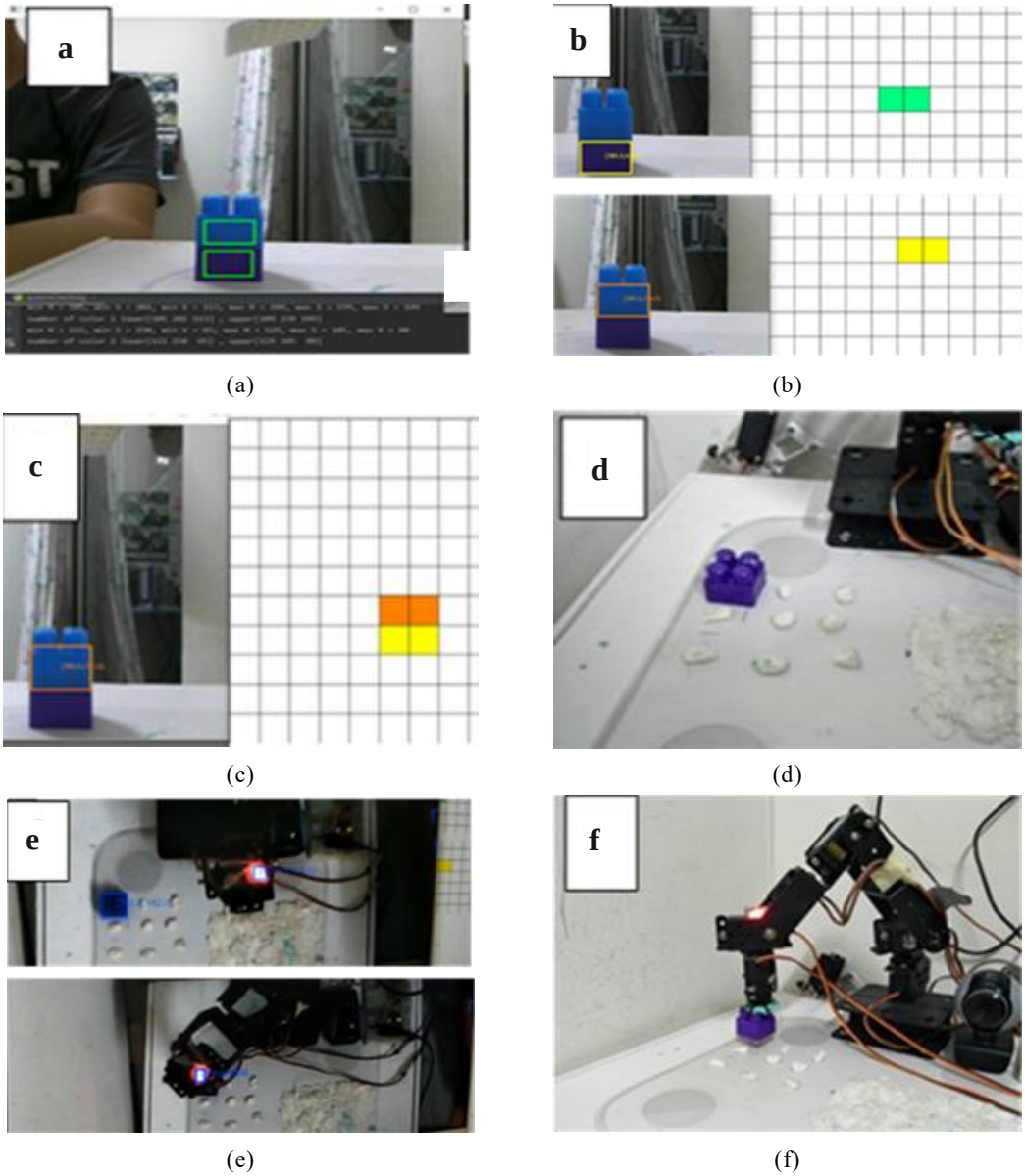

Figure 11. The workflow of assembling IBSc using IBVS, dynamic look-and-move system, and the eye-to-hand system

Figure 13 shows that the experiment built the combination of IBCs. There were four steps to build the IBC combination. The accuracy and duration were recorded at each step of the IBC placing. The IBCs were randomly placed for each trial to prove their flexibility and robustness (Table 2).

The experimental results show that the system is 90% accurate in placing an IBC on top of another IBC. It also has an average of 40.2 seconds to pick and place an IBC on top of another. Given that the average accuracy of IBC placement experienced less than 85% fail and at least 85% success, the experiment shows a total success rate of 80%, with four successes out of five trials.

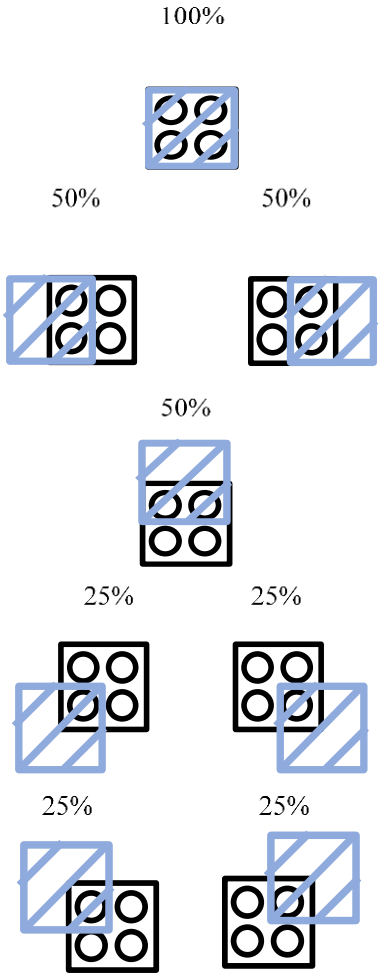

Figure 12. Define the accuracy of IBC placing from one to another

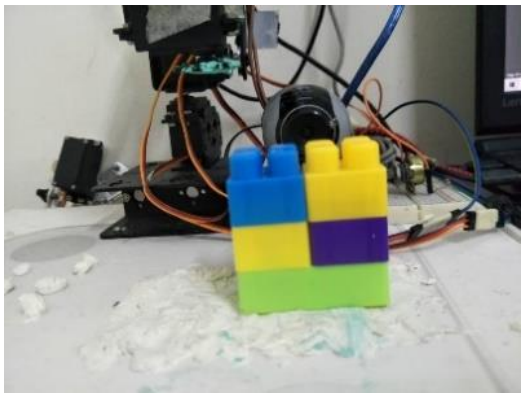

Figure 13. A sample of the IBC combination

Table 2 The time duration and accuracy for the placing of each IBC

Trial	Seq.	Duration (s)	Accuracy (%)	Avg. Accuracy (%)
1	1	29	100	93.75
	2	27	75	
	3	27	100	
	4	80	100	
2	1	32	100	87.50
	2	40	50	
	3	36	100	
	4	37	100	
3	1	32	100	93.75
	2	80	100	
	3	38	100	
	4	36	75	
4	1	29	100	93.75
	2	28	100	
	3	100	100	
	4	26	75	
5	1	28	100	81.25
	2	34	100	
	3	36	75	
	4	29	50	
	Total Average	40.2	90	

Conclusion

In conclusion, this study examined the visual servoing approach to replace the T&R technique in assembling different IBC combinations. The proposed method uses IBVS with dynamic look-and-move and eye-to-hand systems. Thus, a novel kinematic formula with a combination of Trigonometry formulas has been proposed to calculate the incremental joint angles of the robotic arm. The experimental results show the feasibility of using the visual servoing approach to complete the assembly tasks without relying on the T&R technique.

The proposed method achieved an average success rate of 90% and an average task completion of 40.2 seconds. Thus, this eliminates the tedious and time-consuming tasks of teaching the coordinate points to the robotic arms. However, the current system has performance constraints with the camera resolution and high error resolution of the servo motors, contributing to the failure rates. For future work, better specifications of servo motors, high-resolution cameras, and computer processors are proposed to improve accuracy, computation time, and time duration for task completion performance.

References

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. I. Basic approaches. IEEE Robotics & amp;amp; Automation Magazine, 13(4):82-90. https://doi.org/10.1109/MRA.2006.250573

Flandin, G., Chaumette, F., and Marchand, E. (2000). Eye-in-hand/eye-to-hand cooperation for visual servoing. IEEE International Conference on Robotics and Automation (ICRA), 2,741-2,746 https://doi.org/10.1109/ROBOT.2000.846442

Gao, Y., Su, Y., Dong, W., Wang, W., Du, Z., Gao, X., and Mu, Y. (2014). U-Pendant: A universal teach pendant for serial robots based on ROS. IEEE International Conference on Robotics and Biomimetics. 2,529-2,534. https://doi.org/10.1109/ROBIO.2014.7090721

Ghaniwala, A.N., Khan, M.A., Akhtar, M.M., Shaikh, A., Affan, M., and Uddin, R. (2021). Analytical Kinematic analysis of Multi-DOF serial link robot arm. nternational Electrical Engineering Conference (IEEC 2021), 1-6.

Hafez, A.H.A., Cervera, E., and Jawahar, C.V. (2008). Hybrid Visual Servoing by Boosting IBVS and PBVS. 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications. https://doi.org/10.1109/ICTTA.2008.4530116

Hutchinson, S., Hager, G.D., and Corke, P.I. (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5):651-670. https://doi.org/10.1109/70.538972

Kazemi, M., Gupta, K. and Mehrandezh, M. (2009). Global path planning for robust Visual Servoing in complex environments. 2009 IEEE International Conference on Robotics and Automation, 326-332. https://doi.org/10.1109/ROBOT.2009.5152453

Kazemi, M., Gupta, K., and Mehrandezh, M. (2010). Path-Planning for Visual Servoing: A Review and Issues. Visual Servoing via Advanced Numerical Methods, 189-207. https://doi.org/10.1007/978-1-84996-089-2_11

Kelly, R., Carelli, R., Nasisi, O., Kuchen, B., and Reyes, F. (2000). Stable visual servoing of camera-in-hand robotic systems. IEEE/ASME Transactions on Mechatronics, 5(1):39-48. https://doi.org/10.1109/3516.828588

Kulpate, C., Mehrandezh, M. and Paranjape, R. (2005). An eye-to-hand visual servoing structure for 3D positioning of a robotic arm using one camera and a flat mirror. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. https://doi.org/10.1109/IROS.2005.1545348

Mahanta, G.B., Deepak, B.B.V.L., Dileep, M., Biswal, B.B. and Pattanayak, S.K. (2019). Prediction of Inverse Kinematics for a 6-DOF Industrial Robot Arm Using Soft Computing Techniques. Soft Computing for Problem Solving, 519-530. https://doi.org/10.1007/978-981-13-1595-4_42

Malis, E., Chaumette, F., and Boudet, S. (1998). Positioning a coarse-calibrated camera with respect to an unknown object by 2D 1/2 visual servoing. Proceedings. 1998 IEEE International Conference on Robotics and Automation, 1,352-1,359. https://doi.org/10.1109/ROBOT.1998.677293

Ogawa, T. and Kanada, H. (2010). Solution for Ill-Posed Inverse Kinematics of Robot Arm by Network Inversion. Journal of Robotics, 2010:1-10. https://doi.org/10.1155/2010/870923

Palmieri, G., Palpacelli, M., Battistelli, M., and Callegari, M. (2012). A Comparison between Position-Based and Image-Based Dynamic Visual Servoings in the Control of a Translating Parallel Manipulator. Journal of Robotics, 2012:1-11. https://doi.org/10.1155/2012/103954

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V., and Kragic, D. (2012). Dual arm manipulation - A survey. Robotics and Autonomous Systems, 60(10):1,340-1,353. https://doi.org/10.1016/j.robot.2012.07.005

Sun, X., Zhu, X., Wang, P., and Chen, H. (2018). A Review of Robot Control with Visual Servoing. Proceedings of 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, 116-121. https://doi.org/10.1109/CYBER.2018.8688060

Tee, T.H., Eu, K.S., Yap, K.M., Marshall, A., and Lee, T. (2013). 3D Smart User Interactive System with Real-Time Responding Tele-Robotic Proprioceptive Information. 2013 4th International Conference on Intelligent Systems, Modelling and Simulation, 404-408. https://doi.org/10.1109/ISMS.2013.90