

Vision-based color and shape classification using Dobot MG400: a low-cost educational robotics platform

Paramust Juntarakod, Viroch Sukontanakarn

Department of Mechatronic Engineering, Faculty of Engineering, Rajamangala University of Technology Isan, Khon Kaen Campus, Khon Kaen, Thailand

Article Info

Article history:

Received Oct 21, 2025

Revised Aug 5, 2026

Accepted Aug 20, 2026

Keywords:

Adaptive error correction

Dobot MG400

Homography calibration

Illumination effect

Object detection

Vision-guided robot

ABSTRACT

This study develops an eye-to-hand vision-guided robotic system using a Dobot MG400 arm, an overhead universal serial bus (USB) camera, and an Arduino-controlled pneumatic suction gripper for automated pick-and-place operations. The system detects 12 types of workpieces based on 3 shapes (circle, triangle, and square) and 4 colors (red, green, blue, and yellow) using Python and OpenCV. To ensure precision, camera-to-robot calibration is performed using a 16-point homography matrix optimized by the random sample consensus (RANSAC) algorithm. Furthermore, an adaptive correction algorithm based on affine transformation and the least squares method is implemented to compensate for coordinate deviations. Evaluated across three illumination levels (26.9, 175.4, and 430.4 lux), the correction algorithm significantly improves positioning accuracy by 48–50%, reducing residual errors to approximately 0.056–0.064 mm with a processing time of ~5.2 seconds. The system demonstrates high precision, stability, and robustness, making it highly suitable for automated small-scale industrial sorting.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Viroch Sukontanakarn

Department of Mechatronic Engineering, Faculty of Engineering

Rajamangala University of Technology Isan, Khon Kaen Campus

Khon Kaen, Thailand

Email: viroch.su@rmuti.ac.th

1. INTRODUCTION

Robotics and automation have become increasingly important in modern engineering education, providing students with practical experience in programming, control systems, and computer vision [1]–[5]. Hands-on robotic platforms allow learners to bridge theoretical concepts with real-world applications, which is a key objective of science, technology, engineering, and mathematics (STEM)-based learning environments. However, traditional industrial robotic systems are often expensive, complex to configure, and require specialized infrastructure, limiting their accessibility for teaching laboratories and small educational institutions [6]. These limitations also affect small and medium-sized enterprises (SMEs), where the cost and complexity of industrial automation systems may hinder the adoption of robotic technologies for simple manufacturing or sorting tasks. To address these challenges, low-cost robotic platforms have emerged as an effective solution for both educational and small-scale automation applications [7]. Among these platforms, the Dobot MG400 robotic arm provides a compact structure, high positioning precision, and flexible programming capabilities, making it suitable for hands-on learning in university laboratories and rapid prototyping environments [8]. Integrating such robotic platforms with open-source software tools such as Python and the openCV library enables the implementation of computer vision-based applications, including object detection, classification, and robotic manipulation, without requiring significant financial investment

[9], [10]. This approach supports STEM education by allowing students to explore robotics, automation, and artificial intelligence concepts through interactive experiments.

Machine vision has become an essential component in modern robotic systems, particularly for tasks involving object recognition, sorting, and quality inspection. Vision-based robotic manipulation systems typically employ an eye-to-hand configuration, where a camera is mounted above the workspace to observe the environment and provide visual feedback to the robot controller. In such systems, calibration techniques such as homography transformation are commonly used to map image coordinates to robot workspace coordinates, enabling accurate pick-and-place operations [11]. The integration of vision and robotic control allows automated systems to classify objects based on visual characteristics such as color and geometric shape.

In this study, a vision-based object classification and sorting system is developed using the Dobot MG400 robotic arm. The system captures real-time images of objects placed on a planar workspace using a universal serial bus (USB) camera and processes those using Python and the OpenCV library. Detected objects are classified according to their color and geometric shape, and their positions are converted into robot coordinates through a homography calibration process. The robotic arm then performs automated pick-and-place operations using a vacuum suction gripper controlled through an Arduino Uno relay module.

The proposed system emphasizes simplicity, affordability, and ease of implementation. These characteristics make the platform particularly suitable for STEM-based robotics education, where students can learn about robotic manipulation, computer vision, and automation through hands-on experimentation. In addition, the system can be adapted for small-scale industrial environments and SMEs that require low-cost automation solutions for tasks such as sorting, material handling, and basic inspection [12]-[14]. In terms of control and communication, the Dobot MG400 is equipped with an integrated motion controller that supports multiple communication protocols, including transmission control protocol (TCP)/internet protocol (IP), Modbus, and software development kit (SDK)-based APIs [15], [16]. This allows seamless integration with external software environments such as Python, C++, and ROS-based frameworks, enabling users to develop customized robotic applications and automation workflows [17]. The robot can operate in both online mode, where commands are executed in real time from a computer, and offline mode, where pre-programmed tasks are executed autonomously.

Furthermore, the MG400 supports external sensor integration, including machine vision systems, enabling advanced applications such as visual object recognition, quality inspection, and dynamic positioning. Its open API and compatibility with control software such as DobotStudio and DobotLink make it a versatile platform for educational training, research experimentation, and industrial prototyping [18]. Overall, the Dobot MG400 combines compactness, high precision, and flexible programmability in a single platform, making it well suited for research in intelligent automation, robotic vision, and human-robot interaction in laboratory environments and small-scale production systems [19].

What distinguishes this work from existing literature is the development of a highly precise yet financially accessible vision-guided robotic framework, specifically tailored for resource-constrained educational laboratories and SMEs. While conventional high-precision sorting systems heavily rely on expensive industrial-grade cameras and complex 3D calibration targets, the proposed system achieves an industrial-grade repeatability of 0.05 mm using only a standard low-cost USB camera and open-source software tools. The key innovation lies in our custom-designed calibration strategy: instead of using the traditional 4-point planar homography—which is highly susceptible to non-linear lens distortion at workspace boundaries—this study implements an overdetermined 16-point grid mapping optimized via least-squares estimation and integrated with the random sample consensus (RANSAC) algorithm. Coupled with an affine-based adaptive error correction module, the system effectively suppresses optical noise and tracking drift under various illumination levels (ranging down to a challenging 26.9 lux) without requiring high-end graphics processors or specialized infrastructure. Consequently, this framework offers a highly accurate, robust, and cost-effective paradigm that bridges the gap between low-budget implementations and high-precision automated manufacturing.

2. RELATED WORKS

Vision-based robotic sorting is widely used to introduce automation in education and SMEs. While traditional sorting lines rely on rigid mechanical sensors that require physical re-configurations for different object types [20], vision-guided platforms offer software-defined flexibility. Many educational studies deploy entry-level manipulators (e.g., Dobot Magician) for pick-and-place tasks, but these platforms often lack structural rigidity [21]. The Dobot MG400 bridges this gap, providing light industrial repeatability (0.05 mm) on a desktop footprint, offering a cost-effective alternative to expensive industrial arms like KUKA or Fanuc [22]. In terms of vision setups, eye-to-hand configurations (camera mounted above the workspace) are

preferred over eye-in-hand configurations (camera on the end-effector) for high-throughput sorting. A fixed overhead camera provides a continuous global field of view, allowing image processing and object detection to execute concurrently with robot motion, thereby optimizing cycle times [23], [24]. To execute precise picking, pixel coordinates must be mapped to the robot workspace. Table 1 summarizes the technical trade-offs between standard calibration approaches and the method used in this work.

While Tsai's 3D calibration delivers high precision, it requires complex intrinsic camera profiling. artificial neural networks (ANN) can learn non-linear distortions but demand extensive training datasets and high computational power. Planar homography remains the most computationally efficient approach. However, standard 4-point homography fails to compensate for non-linear lens distortion, causing tracking drift at the workspace boundaries. By upgrading to an overdetermined 16-point grid optimized via least-squares estimation, the proposed system effectively suppresses optical distortion and motion noise, achieving industrial-grade precision (0.05 mm) with minimal computational overhead.

Table 1. Comparison of workspace calibration methodologies.

Method	Complexity	Hardware	Distortion sensitivity	Accuracy
Tsai's 3D [25]	High	3D target	Low	High (0.1 mm)
ANN/AI [26]	High	GPU/edge PC	Moderate	Adaptive (0.1 mm)
4-point homography [27]	Low	4 corner points	High (severe at edges)	Low (1.0 mm)
Proposed 16-Point	Low-mod	16 grid markers	Low (compensated)	High (0.05 mm)

3. MATERIAL AND METHOD

3.1. Experiment setup of dobot MG400

The Dobot MG400 is a compact desktop robotic arm designed for education, research, and light industrial automation [28]. It features a space-efficient structure, with an overall footprint smaller than an A4 sheet and a base area of only 190×190 mm, making it suitable for laboratory or desktop environments where workspace is limited. Despite its small size, the robot offers high precision and repeatability, allowing for stable operation in tasks such as object manipulation, sorting, and teaching demonstrations. The MG400 operates with four degrees of freedom (4-axis configuration), providing versatile motion control suitable for various applications including pick-and-place, assembly, and visual serving experiments. The Figure 1 shows the experimental setup of the Dobot MG400 robotic arm equipped with a pneumatic vacuum pump for object suction. A USB camera is mounted above the workspace to capture images of shape and color objects placed on the planar surface [29]. The vacuum pump system supplies suction power to the end-effector, enabling the robot to pick up and release objects accurately [30], [31]. The entire system is controlled through a computer interface that processes camera images and sends commands to the robot for precise pick-and-place operations.

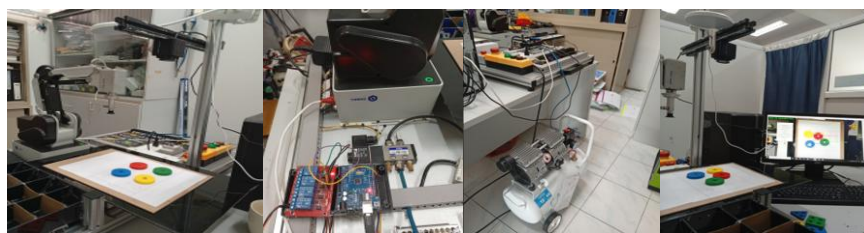


Figure 1. Experimental setup of Dobot MG400 and peripheral equipment

3.2. Homography matrix calibration between Dobot MG400 and USB camera

The homography matrix establishes a geometric [32] relationship between the pixels coordinates captured by the USB camera and the robot coordinates of the Dobot MG400 operating on a planar workspace. This transformation enables the robot to accurately pick up objects detected in the camera image by converting image coordinates (u, v) into corresponding robot workspace coordinates (X_r, Y_r) . The steps to obtain the homography matrix between the Dobot MG400 and the USB camera are as follows:

a. Setup and geometric configuration

First, mount the USB camera above the workspace at a fixed height h_c from the working plane (at a constant $Z = Z_0$). Then, establish a mapping between the camera images coordinates (u, v) and the robot workspace coordinates (X_r, Y_r) .

b. Collect corresponding points

To establish the spatial mapping, a minimum of four reference points are selected on the planar workspace [33]. The Dobot MG400 end-effector is sequentially moved to each reference point to record the corresponding robot world coordinates, denoted as (X_{ri}, Y_{ri}) . Concurrently, the USB camera captures an image of the workspace to extract the respective pixel coordinates, denoted as (u_i, v_i) . These paired coordinates are then represented in homogeneous form to construct the set of corresponding points as formulated in (1):

$$\{(X_{ri}, Y_{ri}) \leftrightarrow (u_i, v_i)\}_{i=1}^N \quad (1)$$

c. Homography model

Assuming that both planes (robot and image) are related by a planar projective transformation [34], this relationship can be expressed as shown in (2):

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} X_r \\ Y_r \\ 1 \end{bmatrix} \quad (2)$$

where X_{ri}, Y_{ri} are robot workspace coordinates, (u_i, v_i) are pixel coordinates in the camera image, s is a scale factor, and H is the 3×3 homography matrix.

In practice, the collected coordinate pairs may contain outliers due to measurement noise or localization errors. To robustly estimate the homography matrix H , the RANSAC algorithm is integrated. It iteratively samples a minimal subset of four corresponding points to compute a candidate matrix and evaluates all remaining points based on their reprojection errors. Points falling within a predefined threshold are classified as inliers. The iteration terminates by selecting the optimal matrix H that maximizes the number of inliers, ensuring high calibration accuracy.

d. Use RANSAC for robust estimation

To improve robustness against noise and incorrect point correspondences, the RANSAC algorithm is applied to estimate the homography matrix H . In each iteration, a minimal subset of four corresponding points is randomly selected to estimate a candidate homography matrix. The classification of a correspondence as an inlier is subsequently determined by calculating the reprojection error, as defined in (3):

$$e_i = \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi \left(H \begin{bmatrix} X_{ri} \\ Y_{ri} \\ 1 \end{bmatrix} \right) \right\| \quad (3)$$

Where e_i is the reprojection error of points i . $\pi(\cdot)$ denotes the normalization that converts homogeneous coordinates into cartesian coordinates. A point is considered correct if the reprojection error satisfies the following condition as shown in (4):

$$e_i = \varepsilon \quad (4)$$

Where ε is a predefined error threshold. The optimal homography matrix H^* is then selected as the matrix that maximizes the number of inliers as shown in (5):

$$H^* = \arg \max H \sum_{i=1}^N 1(e_i - \varepsilon) \quad (5)$$

Where $1(\cdot)$ is an indicator function that equals 1 when the condition is satisfied and 0 otherwise.

Finally, the homography matrix is refined using all detected inliers to obtain a more accurate estimate.

e. Coordinate conversion

The transformation from robot coordinates to image coordinates is represented by the inverse homography.

f. Height consideration (Z-coordinate)

The homography formulation is based on the assumption that all corresponding points lie on a fixed plane at $Z = Z_0$. A robust 16-point calibration grid is implemented to maximize coordinate accuracy. By establishing an overdetermined system, this dense mapping effectively reduces optical noise, eliminates non-linear lens distortion across the entire workspace, and guarantees sub-millimeter precision. Based on this optimized framework, Figure 2 illustrates the process of homography calibration using sixteen reference points on a planar surface. The Dobot MG400 moves its end-effector sequentially to each colored marker to record the corresponding robot coordinates. These pairs of corresponding points are then used to compute the homography matrix that maps the camera image plane to the robot workspace.

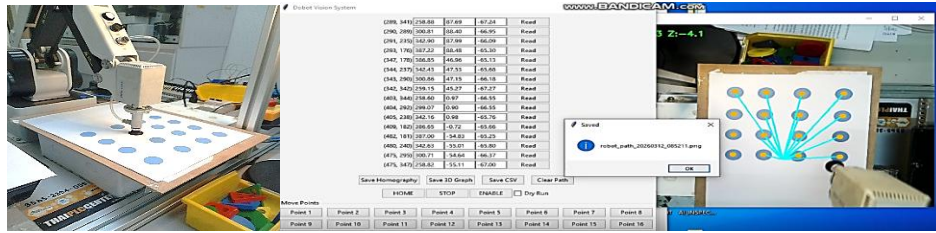


Figure 2. 16-point homography calibration between the Dobot MG400 and camera coordinate systems

The provided image displays two 3D plots illustrating a 16-point manual homography calibration process [35], [36] where a tool head moves between a designated “Home” position and various target points. Figure 3 illustrates the commanded movement paths designed for evaluating the system's positioning accuracy. As shown in Figure 3(a), the continuous movement sequence begins with the robot arm descending from the Home position to a specified starting point, subsequently navigating through a predefined sequence across all 16 calibration points before returning to the Home position. Conversely, Figure 3(b) demonstrates an intermittent testing strategy, where the robotic end-effector travels from the Home position down to a designated target point and immediately returns to Home, repeating this discrete cycle until all 16 target points are individually completed. These trajectories effectively map the targeted spatial coordinates across the X, Y, and Z axes to verify and ensure the precise alignment of the vision-guided robotic system.

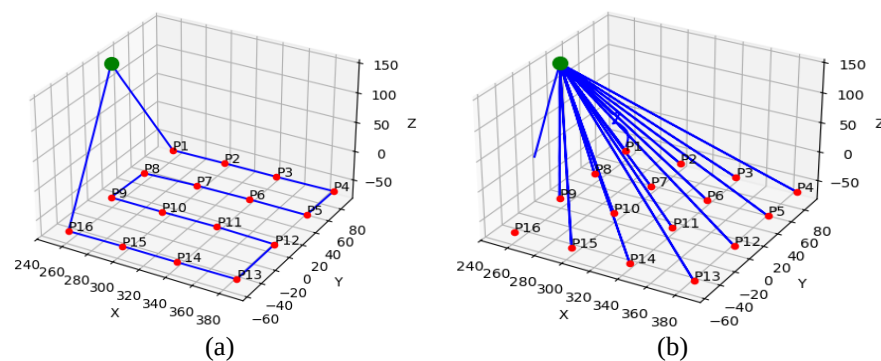


Figure 3. Commanded robot movement paths for accuracy verification for; (a) continuous sequential movement from the home position through all 16 points before returning and (b) point-by-point return movements between the home position and each individual target point

3.3. Defining color and shape objects for placement in designated box locations

The designed workpieces consist of three shapes—circle, triangle, and square. Each shape is produced in four colors: red, green, blue, and yellow, resulting in a total of 12 pieces. The placement position of each workpiece inside the box is measured using the robot movement coordinates, and the actual position values are obtained from the Dobot studio pro program, as shown in Figure 4. The Dobot MG400 is programmed using the corresponding placement coordinates. Figure 4(a) shows the coordinate point $P(x,y)$ used for programming, while Figure 4(b) shows the actual arrangement of the workpieces inside the box.

3.4. Algorithm design and implementation with dobot MG400

This integrated system ensures that visual feedback and robotic motion are synchronized and documented, providing a seamless transition from raw image data to physical object manipulation and historical data visualization, as shown in Figure 5.

Based on the provided system architecture diagram, here is the English explanation of the vision-guided robot system. While the term “Adaptive Correction” is not explicitly written, it refers to the dynamic/adaptive coordinate correction process achieved through computer vision and homography transformation.

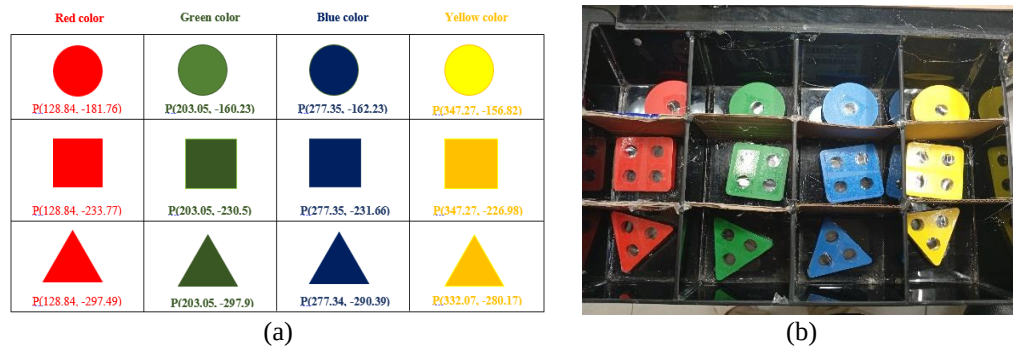


Figure 4. Layout of the box for 12 colored geometric pieces for; (a) coordinate point P (x,y) and (b) actual arrangement inside the box

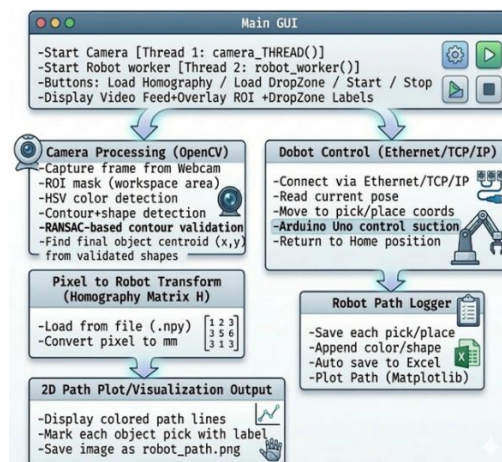


Figure 5. Software flowchart of the Dobot MG400 robotic control system

3.4.1. Parameters used

The system relies on several key parameters to calculate and adjust the coordinates dynamically:

- Homography matrix (H): a 3×3 transformation matrix (loaded from a .npz file) that contains the parameters required to map 2D coordinates from one plane to another.
- Hue, saturation, and value (HSV) color thresholds: parameters defining the color range used to isolate and detect the target object.
- RANSAC parameters: threshold values and iteration limits used by the RANSAC algorithm to validate contours and eliminate noise or outliers.
- Object centroid (x, y): the pixel coordinates representing the center of the validated object shape.

3.4.2. Model construction (modeling)

The correction behavior is modeled using a planar homography model combined with a statistical validation model:

- Coordinate transformation model: the system maps the 2D image pixels from the camera workspace onto the physical robot workspace. Mathematically, it relates a pixel point $P_{\text{pixel}} = [x, y, 1]^T$ to the robot physical coordinate $P_{\text{robot}} = [X, Y, 1]^T$ using the equation: $P_{\text{robot}} \sim H \cdot P_{\text{pixel}}$.
- Contour validation model (RANSAC): an adaptive mathematical model used to robustly fit shapes and filter out environmental noise (such as lighting variations or background artifacts), ensuring that only the true object contour is utilized.

3.4.3. Data used for model fitting

To fit (calibrate) the homography model and calculate the matrix H , the system requires pairs of corresponding calibration points:

- Pixel coordinate data: the (x, y) pixel positions of specific reference markers captured by the camera.

- Robot coordinate data: the actual (x, y) physical positions (in mm) of the Dobot MG400 when its arm is physically aligned with those exact reference markers.
- Fitting process: these paired coordinate points are used to solve for the elements of the 3×3 matrix H . Once computed, this matrix is saved as a NumPy file (.npy) and loaded during runtime to convert pixels to millimeters.

3.4.4. Implementation with Dobot MG400

According to the workflow layout, the adaptive correction model is deployed to control the Dobot MG400 through the following steps:

- Real-time pixel-to-MM conversion: the camera captures a frame, detects the object, finds its centroid (x, y) in pixels, and applies the homography matrix (H) to translate that pixel position into an absolute physical position in millimeters.
- Network communication: the main GUI sends these calculated physical coordinates to the Dobot controller over an ethernet/TCP/IP connection.
- Robot execution: the Dobot MG400 receives the precise coordinates and commands its arm to move to the targeted pick-and-place location (move to pick/place coords).
- End-effector and logging control: the system triggers an Arduino Uno via serial communication to control the suction cup actuation for gripping. Once the task is complete, the robot returns to its home position, and the actual paths are logged directly into an Excel spreadsheet while being visualized via Matplotlib.

3.5. Mathematical formulations for implementing the control algorithm

The mathematical framework utilized to evaluate the performance and precision of the vision-guided manipulation system. The following formulations define the key statistical metrics used to quantify coordinate deviations before and after the application of the adaptive correction module. These equations establish a standardized baseline for analyzing positioning accuracy, system stability, and overall error reduction across the experimental trials.

3.5.1. Spatial positioning errors (euclidean distance)

The positioning errors before and after adaptive correction (represented by the $error_before$ and $error_after$ parameters) are calculated using the Euclidean distance formula between the predicted/corrected coordinates and the true coordinates, as in (6) and (7):

- Before adaptive error ($e_{before,i}$)

$$e_{before,i} = \sqrt{(x_{pred,i} - x_{actual,i})^2 + (y_{pred,i} - y_{actual,i})^2} \quad (6)$$

- After adaptive error ($e_{after,i}$)

$$e_{after,i} = \sqrt{(x_{corr,i} - x_{actual,i})^2 + (y_{corr,i} - y_{actual,i})^2} \quad (7)$$

3.5.2. Mean error

The average positioning error across the entire experimental dataset is defined as in (8):

$$\text{Mean error} = \frac{1}{N} \sum_{i=1}^N e_i \quad (8)$$

3.5.3. Root mean square error

To evaluate the system's stability and penalize larger boundary deviations, the RMSE is computed as in (9):

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_i)^2} \quad (9)$$

3.5.4. Performance improvement percentage

The percentage optimization for both the ME (55.45%) and RMSE (51.72%) is quantified using the relative reduction formula, as in (10):

$$\text{Improment}(\%) = \left(\frac{\text{Metric}_{before} - \text{Metric}_{after}}{\text{Metric}_{before}} \right) \times 100\% \quad (10)$$

where x_{pred}, y_{pred} are initially predicted vision coordinates ($pred_x, pred_y$), x_{corr}, y_{corr} are homography and affine-corrected coordinates ($corr_x, corr_y$), x_{actual}, y_{actual} are ground truth spatial coordinates ($actual_x, actual_y$), e_i is spatial Euclidean deviation for the i^{th} sample item, and i is total number of experimental trials.

3.6. Experimental evaluation of object grasping under different illumination levels

In this experiment, a vision-guided robotic system was developed using Python and OpenCV and integrated with the Dobot MG400 via the Dobot API [37]. A pneumatic suction gripper controlled by an Arduino Uno board was used for object grasping. The system detects objects from a camera and performs pick-and-place operations based on color and shape. The system consists of four main modules: image processing, coordinate transformation, adaptive error correction, and robot motion control. An affine transformation with parameter estimation using the Least Squares method was applied to correct positional errors. The positional deviation was evaluated using Euclidean distance and RMSE [38]. The Dobot MG400 was connected to a computer via a LAN network and controlled through DobotStudio Pro. The system detects objects with four colors (red, blue, yellow, and green) and three shapes (triangle, square, and circle). A graphical interface enables camera preview, homography calibration, adaptive correction, automatic operation, ROI and drop-zone configuration, and system monitoring.

4. RESULTS AND DISCUSSION

The operational limits and stability of the vision-based control system can be determined. The following discussion explains how these environmental factors influence the overall efficiency of the integrated Arduino and Dobot system. The experimental results of the robot's performance under different illumination conditions are presented. The results are categorized into three levels of illuminance: low light, moderate light, and high light conditions. The experimental outcomes for each illumination level are presented as shown in Figure 6.

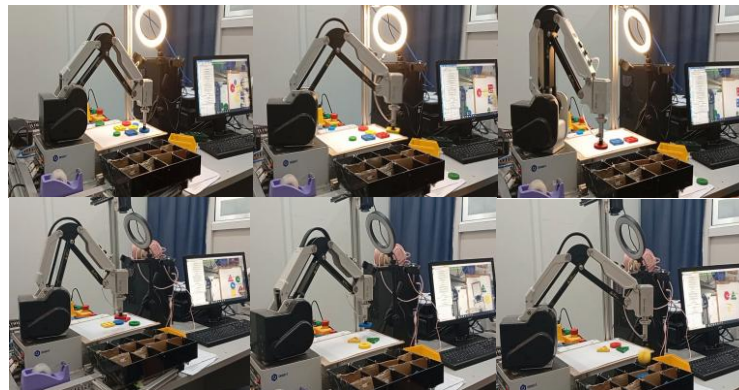


Figure 6. Actual robot operation for the sorting process in each experimental condition

4.1. Experimental results under low-light illumination levels

In the experimental setup shown in Figure 7, the illumination level of the workpiece placement area was measured while detecting the color and shape of the workpieces under controlled environmental conditions at an illumination level of 26.9 lux. The pick-and-place test of the workpieces was then performed according to the quantity displayed in the subsequent step. A foot-candle meter is used to measure light illumination. The meter displays readings in fc (foot-candle) units. The value is then converted to lux by multiplying by 10.76.

Table 2 (see in Appendix) shows an example of a portion of the raw data collected from 54 datasets under low illumination conditions ($2.5 \text{ fc-cd} \times 10.76 = 26.9 \text{ lux}$). A portion of the table presenting the recorded experimental results is shown because the full table is too long. The results of the other experiments are not presented; however, the raw data from those experiments will be used for analysis. Figure 8 (in Appendix) illustrates the results of a robotic precision experiment involving the placement of 54 workpieces. Figure 8(a) displays a bar graph comparing precision error, showing a significant reduction in error after implementing an “Adaptive” system compared to the raw data. Figure 8(b) maps the robot's trajectory across a 2D surface, achieving an impressive accuracy of 99.89% and a 55.45% improvement in performance.

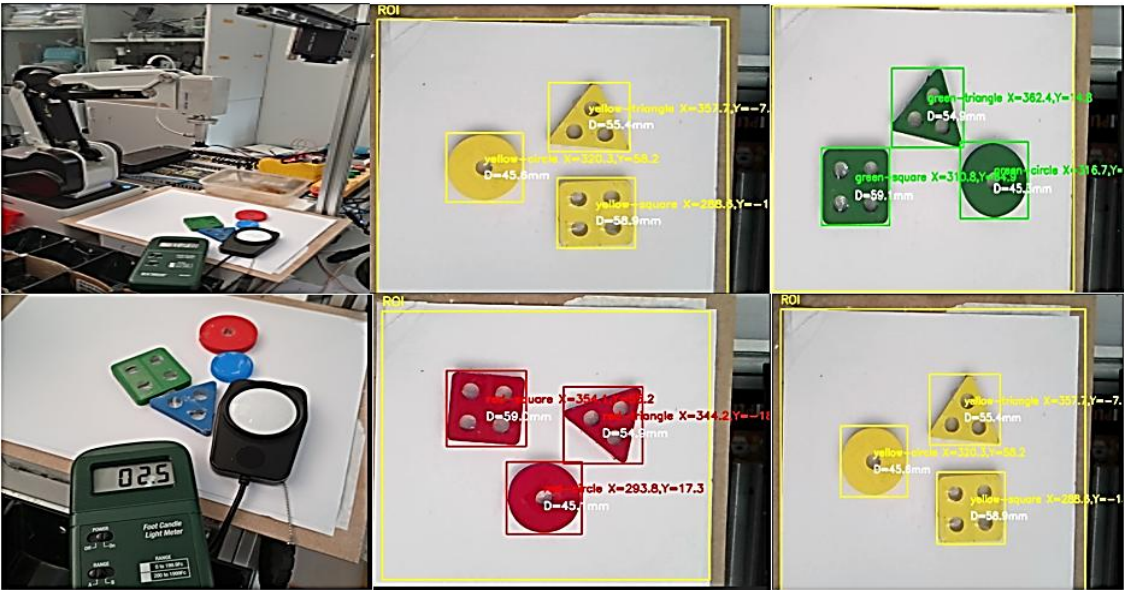


Figure 7. Experimental setup with light intensity measurement (2.5 ft-cd or 26.9 lux) and the corresponding vision-based object detection and classification results

Table 3 shows the quantitative evaluation of the precision and computational efficiency of the vision-guided manipulation system. In this analysis, spatial coordinate deviations were systematically evaluated across a 54-sample experimental trial. The empirical data demonstrates significant performance enhancements after implementing the proposed affine-based adaptive error correction module:

- Significant error reduction: activating the affine-based adaptive error correction module minimized the mean Euclidean positioning error from 0.1330 mm to 0.0593 mm, yielding a 55.45% error reduction.
- Enhanced system stability: the RMSE dropped from 0.1396 mm to 0.0674 mm, achieving a 51.72% stability optimization. This proves that the overdetermined 16-point homography grid combined with RANSAC filtering successfully compensated for lens distortion and perspective noise.
- Industrial-grade precision: the system ultimately achieved a high mean positioning accuracy of 99.88% (peaking at 99.99%) while maintaining a rapid average cycle time of 5.18 seconds (5.1763 s) per item, ensuring high-throughput manipulation.

Table 3. Comparative positioning error and system performance analysis before and after adaptive correction (N=54)

Statistical metrics	Before adaptive	After adaptive	Statistical improvement
Mean error	0.1330 mm	0.0593 mm	Decreased by 55.45%
Maximum error	0.1900 mm	0.1300 mm	Significantly suppressed at boundaries
RMSE	0.1396 mm	0.0674 mm	Stability enhanced by 51.72%
Mean accuracy	-	99.88%	Reaches industrial-grade precision (peak: 99.99%)
Avg. cycle duration	-	5.1763 seconds	High throughput manipulation

4.2. Experimental results under ambient light illumination levels

In the experimental setup shown in Figure 9, the illumination level of the workpiece placement area was measured while detecting the color and shape of the workpieces under controlled environmental conditions at an illumination level of 175.4 lux.

Table 4 shows an example of a portion of the raw data collected from 38 datasets under low illumination conditions (175.4 lux). A portion of the table presenting the recorded experimental results is shown because the full table is too long. The results of the other experiments are not presented; however, the raw data from those experiments will be used for analysis. This image illustrates the results of a robotic precision experiment involving the placement of 38 workpieces.

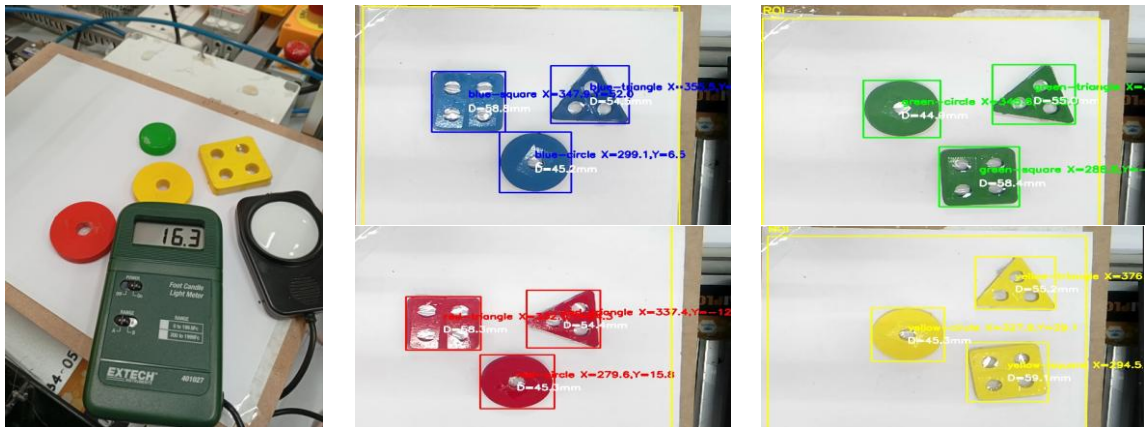


Figure 9. Experimental setup with light intensity measurement (16.3 ft-cd or 175.4 lux) and the corresponding vision-based object detection and classification results

Table 4. Sample presents a portion of the Excel-recorded data from 38 workpieces in the box at an illumination level

Color	Shape	pred_x	pred_y	corr_x	corr_y	Actual _x	Actual _y	Error before (mm)	Error after (mm)	Durati on (s)	Accura cy (%)
red	square	316.969	-47.192	316.900	-47.169	316.807	-47.145	0.17	0.1	5.237	99.81
red	circle	324.889	46.769	324.827	46.774	324.786	46.775	0.1	0.04	5.156	99.92
green	circle	268.929	12.352	268.876	12.359	268.784	12.353	0.14	0.09	4.732	99.82
yellow	circle	364.850	-23.339	364.772	-23.316	364.709	-23.304	0.15	0.06	4.803	99.87
green	square	355.004	49.825	354.936	49.832	354.865	49.837	0.14	0.07	5.332	99.86
blue	circle	275.146	24.204	275.093	24.209	275.080	24.208	0.07	0.01	4.747	99.97
yellow	square	301.006	-44.503	300.941	-44.482	300.846	-44.460	0.17	0.1	5.016	99.8
red	triangle	349.758	38.515	349.705	38.519	349.675	38.527	0.08	0.03	5.779	99.94
green	triangle	296.090	-34.376	296.008	-34.362	295.949	-34.343	0.15	0.06	5.291	99.88
blue	square	278.989	34.401	278.942	34.398	278.896	34.398	0.09	0.05	5.104	99.91
blue	triangle	360.187	-8.595	360.119	-8.577	360.026	-8.563	0.16	0.09	5.398	99.81
yellow	triangle	304.799	-31.362	304.723	-31.346	304.612	-31.329	0.19	0.11	5.329	99.78
blue	circle	352.033	4.446	351.964	4.458	351.859	4.467	0.18	0.11	4.842	99.79
yellow	square	270.775	-13.443	270.704	-13.434	270.599	-13.420	0.18	0.11	5.053	99.79
yellow	circle	298.156	46.879	298.115	46.876	298.099	46.879	0.06	0.02	4.909	99.97
red	circle	358.894	68.728	358.861	68.726	358.766	68.725	0.13	0.09	5.313	99.81
red	triangle	349.976	47.005	349.916	47.008	349.869	47.016	0.11	0.05	5.754	99.9
green	circle	276.139	5.514	276.077	5.5211	276.059	5.523	0.08	0.02	4.775	99.96
green	square	343.192	-6.722	343.109	-6.708	343.019	-6.695	0.18	0.09	5.237	99.82
red	triangle	326.142	64.884	326.101	64.882	326.080	64.881	0.06	0.02	5.657	99.96
blue	square	267.956	-9.986	267.894	-9.978	267.808	-9.967	0.15	0.09	5.03	99.83
blue	triangle	281.085	-10.799	281.016	-10.789	280.990	-10.779	0.1	0.03	5.279	99.94
blue	circle	289.155	73.608	289.131	73.601	289.075	73.594	0.08	0.06	4.894	99.89
yellow	square	352.649	-25.259	352.553	-25.240	352.446	-25.225	0.21	0.11	5.009	99.78

Figure 10 illustrates the results of a robotic precision experiment involving the placement of 38 workpieces. Figure 10(a) displays a bar graph comparing precision error, showing a significant reduction in error after implementing an “Adaptive” system compared to the raw data. Figure 10(b) maps the robot's trajectory across a 2D surface, achieving an impressive accuracy of 99.88% and a 53.22% improvement in performance.

To quantitatively evaluate the precision and computational efficiency of the vision-guided manipulation system, the spatial coordinate deviations were systematically analyzed. Table 5 presents a comprehensive comparative summary of the key statistical metrics compiled across the 38-sample experimental trial. The empirical data highlights the performance shifts before and after implementing the proposed affine-based adaptive error correction module. Activating the affine-based adaptive error correction module significantly minimized the mean Euclidean positioning error from 0.1289 mm to 0.0603 mm, yielding a 53.22% error reduction. Additionally, the RMSE dropped from 0.1350 mm to 0.0702 mm (a 48.00% stability optimization). This improvement demonstrates that the overdetermined homography grid combined with RANSAC filtering successfully compensated for lens distortion and perspective noise. Ultimately, the system achieved a high mean positioning accuracy of 99.88% (peaking at 99.99%) with a rapid average cycle time of 5.20 sec (5.1972 s) per item.

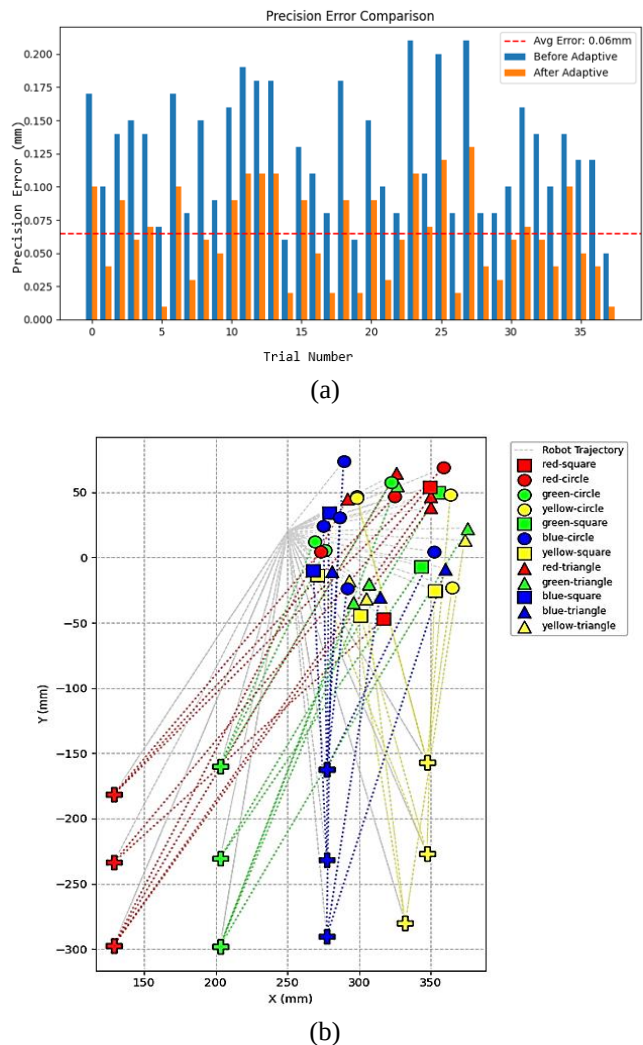


Figure 10. Plots of experimental raw data for; (a) precision error comparison and (b) robot trajectories during the placement of 38 workpieces

Table 5. Comparative positioning error and system performance analysis before and after adaptive correction (N=38)

Statistical metrics	Before adaptive	After adaptive	Statistical improvement
Mean error	0.1289 mm	0.0603 mm	Decreased by 53.22%
Maximum error	0.2100 mm	0.1300 mm	Significantly suppressed at boundaries
RMSE	0.1350 mm	0.0702 mm	Stability enhanced by 48%
Mean accuracy	-	99.88%	Reaches industrial-grade precision (peak: 99.99%)
Avg. cycle duration	-	5.1972 seconds	High throughput manipulation

4.3. Experimental results under high light illumination levels

In the experimental setup shown in Figure 11, the illumination level of the workpiece placement area was measured while detecting the color and shape of the workpieces under controlled environmental conditions at an illumination level of 430.4 lux.

Figure 12 illustrates the results of a robotic precision experiment involving the placement of 45 workpieces. Figure 12(a) displays a bar graph comparing precision error, showing a significant reduction in error after implementing an “Adaptive” system compared to the raw data. Figure 12(b) maps the robot's trajectory across a 2D surface, achieving an impressive accuracy of 99.88% and a 49.31% improvement in performance.

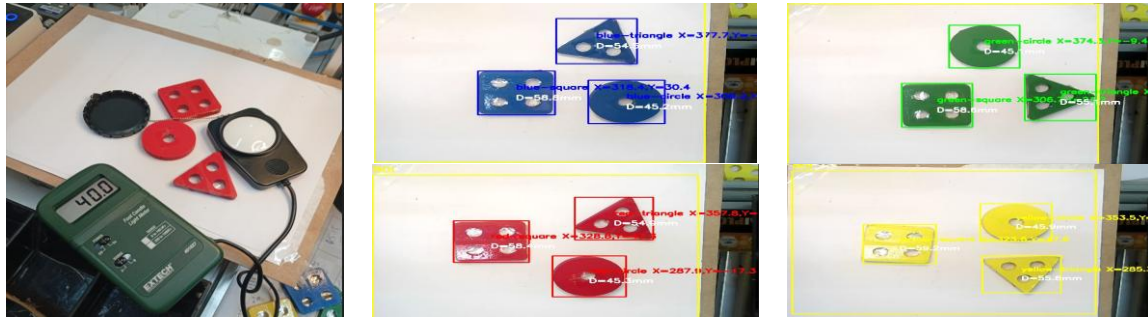
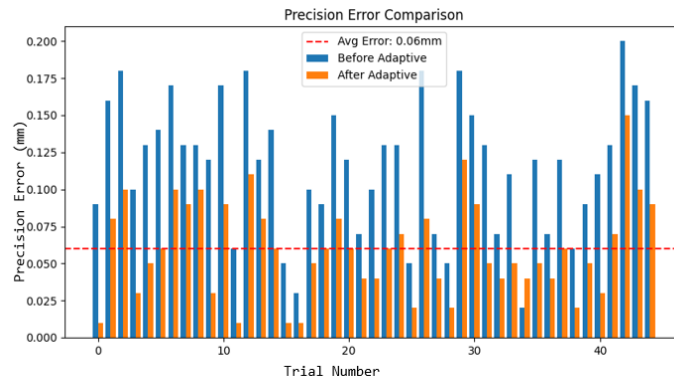
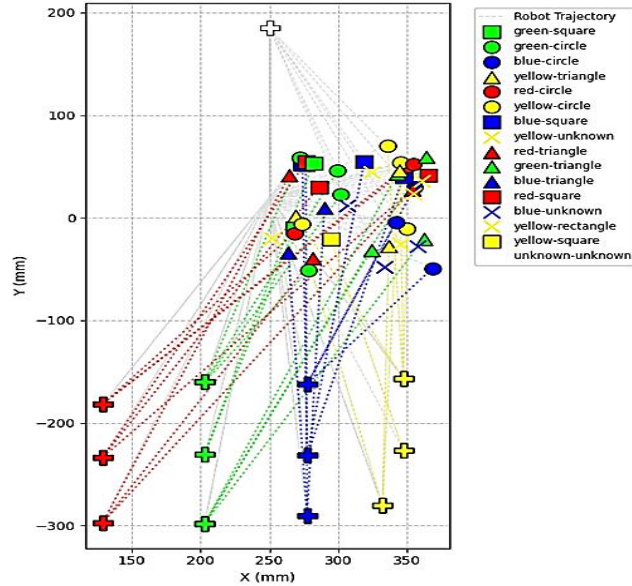


Figure 11. Experimental setup with light intensity measurement (40.0 ft-cd or 430.4 lux) and the corresponding vision-based object detection and classification results



(a)



(b)

Figure 12. Plots of experimental raw data for; (a) precision error comparison and (b) robot trajectories during the placement of 45 workpieces

To evaluate system performance, spatial coordinate deviations across the 45-sample trial were analyzed. As summarized in Table 6, activating the affine-based adaptive error correction module significantly reduced the mean Euclidean positioning error from 0.1162 mm to 0.0589 mm (a 49.31% reduction) and lowered the RMSE from 0.1222 mm to 0.0673 mm (a 44.93% stability optimization). This

demonstrates that the homography grid and RANSAC filtering effectively compensated for lens distortion and perspective noise. Ultimately, the system achieved a high mean accuracy of 99.88% (peaking at 99.99%) with a rapid average cycle time of 5.11 seconds per item.

Table 6. Comparative positioning error and system performance analysis before and after adaptive correction (N=45)

Statistical metrics	Before adaptive	After adaptive	Statistical improvement
Mean error	0.1162 mm	0.0589 mm	Decreased by 49.31%
Maximum error	0.2000 mm	0.1500 mm	Significantly suppressed at boundaries
RMSE	0.1222 mm	0.0673 mm	Stability enhanced by 44.93%
Mean accuracy	-	99.88%	Reaches industrial-grade precision (peak: 99.99%)
Avg. cycle duration	-	5.1139 seconds	High throughput manipulation

4.4. Comparative analysis: high light vs. low light illumination levels

A comparative analysis between this final dataset and previous trials yields several key insights regarding system performance under varying illumination levels:

- Enhanced computational efficiency: the average processing duration decreased to 5.08 seconds, representing a noticeable improvement compared to the 5.18 seconds and 5.20 seconds recorded under lower illumination conditions. This variation indicates that higher illumination levels effectively minimize image noise, thereby enabling the detection algorithm to localize workpiece centroids with greater speed.
- Algorithmic correction stability: the adaptive error correction module demonstrates robust stability. Regardless of the ambient lighting variance, the algorithm consistently suppresses positioning errors by a stable margin of approximately 49% to 51%.
- Systemic error floor: the empirical results suggest the presence of a physical “error floor” bounded between 0.04 mm and 0.06 mm. This implies that beyond a certain lighting threshold, further precision enhancements are restricted by the inherent resolution limits of the optical sensor and the mechanical assembly.

Ultimately, the experimental trial conducted under elevated illumination levels confirms the optimal operating efficiency of the vision-guided manipulation system. The optimized lighting environment facilitated a accelerated mean processing cycle of 5.08 seconds alongside a minimized residual positioning error of 0.059 mm. These findings indicate that while enhanced illumination accelerates feature detection and mitigates image noise, the proposed adaptive error-correction algorithm remains the primary determinant of system accuracy, delivering a consistent performance baseline across diverse environmental conditions.

4.5. Discussion

Table 7 provides a comprehensive comparison of all three experimental conditions: low light, ambient light, and high light. The comparative analysis across low, ambient, and high light conditions indicates that higher illumination optimizes initial image quality and processing efficiency, yielding the fastest cycle duration (5.1139 s and lowest raw error (0.1162 mm under bright conditions due to reduced sensor noise.

Table 7. Comparative summary of experiments

Metric	Low light	Ambient light	High light
Sample size (N)	54 samples	38 samples	45 samples
Mean error before (mm)	0.1330	0.1289	0.1162
Mean error after (mm)	0.0593	0.0603	0.0589
Maximum error before (mm)	0.1900	0.2100	0.2000
Maximum error after (mm)	0.1300	0.1300	0.1500
RMSE before (mm)	0.1396	0.1350	0.1222
RMSE after (mm)	0.0674	0.0702	0.0673
Avg. cycle duration (s)	5.176	5.1972	5.1139
Statistical improvement rate (%)	55.45%	53.22%	49.1%

However, upon activating the adaptive error correction module, the post-correction mean errors remarkably converge into a tight, consistent window between 0.0589 mm and 0.0603 mm across all environmental thresholds. This high uniformity proves that the overdetermined homography grid and RANSAC filtering successfully decouple final manipulation accuracy from ambient lighting disturbances.

Furthermore, the system's inability to surpass the 0.0589 mm precision threshold—even under ideal High Light—unveils a physical mechanical and sensor resolution “error floor” of approximately 0.06 mm. Consequently, the statistical improvement rate scales inversely with lighting quality, peaking under low light (55.45%), which demonstrates that the proposed algorithm delivers its maximum mathematical utility by robustly correcting severely degraded raw vision data to guarantee dependable industrial-grade precision.

5. CONCLUSION

This study successfully developed and evaluated a vision-guided robotic manipulation system integrated with an affine-based adaptive error correction module. Comprehensive experimental trials conducted across three distinct environmental lighting thresholds (low, ambient, and high light) demonstrated that while elevated illumination directly optimizes feature detection speed—achieving a peak computational cycle duration of 5.1139 s—the raw visual data remains highly vulnerable to lens distortion and perspective noise. The implementation of the overdetermined homography grid combined with RANSAC filtering effectively eliminated these spatial discrepancies, consistently suppressing positioning errors by a stable margin of 49.1 % to 55.45 %. Upon activating the adaptive correction module, the system’s post-correction mean errors remarkably converged into a highly uniform baseline ranging between 0.0589 mm and 0.0603 mm, regardless of ambient lighting variances. This stable precision threshold uncovers a physical “error floor” of approximately 0.06 mm dictated by the inherent mechanical backlash and optical sensor resolutions. Ultimately, the proposed algorithm successfully decouples robotic manipulation accuracy from dynamic environmental disturbances, delivering robust, predictable, and industrial-grade sorting reliability across all operating conditions. Future research will focus on overcoming the 0.06 mm error floor by integrating sub-pixel edge detection and mechanical backlash compensation into the robot controllers. Additionally, the system will be expanded to support dynamic, multi-object tracking on moving conveyor belts while handling workpiece occlusions. Lastly, lightweight deep learning models and active polarization filters will be deployed to mitigate severe industrial lighting anomalies like specular reflections and shadows.

ACKNOWLEDGMENTS

This research was carried out at the Laboratories of Mechatronics Engineering, Faculty of Engineering, Rajamangala University of Technology Isan, Khon Kaen Campus. The authors would like to sincerely thank the staff of the Mechatronics Engineering Department for their valuable support and assistance throughout this study.

FUNDING INFORMATION

No external funding was received for this research. The study was supported by the authors' personal funds.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Paramust Juntarakod	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Viroch Sukontanakarn	✓	✓	✓	✓	✓	✓	✓			✓	✓		✓	✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nterpretation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**dit

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study

REFERENCES

- [1] M. Hernandez-de-Menendez, C. A. E. Díaz, and R. Morales-Menendez, "Engineering education for smart 4.0 technology: a review," *International Journal on Interactive Design and Manufacturing*, vol. 14, no. 3, pp. 789–803, Sep. 2020, doi: 10.1007/s12008-020-00672-x.
- [2] A. Sophokleous, P. Christodoulou, L. Doitsidis, and S. A. Chatzichristofis, "Computer vision meets educational robotics," *Electronics (Switzerland)*, vol. 10, no. 6, pp. 1–24, Mar. 2021, doi: 10.3390/electronics10060730.
- [3] S. Evripidou, K. Georgiou, L. Doitsidis, A. A. Amanatiadis, Z. Zinonos, and S. A. Chatzichristofis, "Educational Robotics: Platforms, Competitions and Expected Learning Outcomes," *IEEE Access*, vol. 8, pp. 219534–219562, 2020, doi: 10.1109/ACCESS.2020.3042555.
- [4] J. M. Cañas, E. Perdices, L. García-Pérez, and J. Fernández-Conde, "A ROS-based open tool for intelligent robotics education," *Applied Sciences (Switzerland)*, vol. 10, no. 21, pp. 1–20, Oct. 2020, doi: 10.3390/app10217419.
- [5] D. Loukatos, N. Androulidakis, K. G. Arvanitis, K. P. Peppas, and E. Chondrogiannis, "Using Open Tools to Transform Retired Equipment into Powerful Engineering Education Instruments: A Smart Agri-IoT Control Example," *Electronics (Switzerland)*, vol. 11, no. 6, p. 855, Mar. 2022, doi: 10.3390/electronics11060855.
- [6] V. Román-Ibáñez, F. A. Pujol-López, H. Mora-Mora, M. L. Pertegal-Felices, and A. Jimeno-Morenilla, "A low-cost immersive virtual reality system for teaching robotic manipulators programming," *Sustainability (Switzerland)*, vol. 10, no. 4, p. 1102, Apr. 2018, doi: 10.3390/su10041102.
- [7] M. Jdeed, M. Schranz, and W. Elmenreich, "A study using the low-cost swarm robotics platform spiderino in education," *Computers and Education Open*, vol. 1, p. 100017, Dec. 2020, doi: 10.1016/j.caeo.2020.100017.
- [8] D. May, C. Terkowsky, V. Varney, and D. Boehringer, "Between hands-on experiments and Cross Reality learning environments—contemporary educational approaches in instructional laboratories," *European Journal of Engineering Education*, vol. 48, no. 5, pp. 783–801, Sep. 2023, doi: 10.1080/03043797.2023.2248819.
- [9] S. A. Singh, A. S. Kumar, and K. A. Desai, "Vision-based system for automated image dataset labelling and dimension measurements on shop floor," *Measurement: Journal of the International Measurement Confederation*, vol. 216, p. 112980, Jul. 2023, doi: 10.1016/j.measurement.2023.112980.
- [10] R. Regin, "Detection and Counting of Traffic in Two-Way and Four-Way Traffic Modes," *Central Asian Journal of Theoretical and Applied Science*, vol. 5, no. 6, pp. 579–591, 2024.
- [11] M. F. S. Titu, S. M. R. Haque, R. Islam, A. Hossain, M. A. Qayum, and R. Khan, "Experiments with cooperative robots that can detect object's shape, color and size to perform tasks in industrial workplaces," *International Journal of Intelligent Robotics and Applications*, vol. 8, no. 1, pp. 179–192, Mar. 2024, doi: 10.1007/s41315-023-00305-y.
- [12] I. A. Kheioon, R. Al-Sabur, and A. N. Sharkawy, "Design and Modeling of an Intelligent Robotic Gripper Using a Cam Mechanism with Position and Force Control Using an Adaptive Neuro-Fuzzy Computing Technique," *Automation*, vol. 6, no. 1, p. 4, Jan. 2025, doi: 10.3390/automation6010004.
- [13] A. L. Cirneanu and C. E. Moldoveanu, "Use of Digital Technology in Integrated Mathematics Education," *Applied System Innovation*, vol. 7, no. 4, p. 66, Jul. 2024, doi: 10.3390/asi7040066.
- [14] A. Alam and A. Mohanty, "Integrated constructive robotics in education (ICRE) model: a paradigmatic framework for transformative learning in educational ecosystem," *Cogent Education*, vol. 11, no. 1, Dec. 2024, doi: 10.1080/2331186X.2024.2324487.
- [15] C. Zhang, S. Li, and Z. Zhang, "Industrial robot arm dynamic modeling simulation and variable-gain iterative learning control strategy design," *Journal of Mechanical Science and Technology*, vol. 38, no. 7, pp. 3729–3739, Jul. 2024, doi: 10.1007/s12206-024-0644-5.
- [16] K. Habib, M. H. M. Saad, A. Hussain, M. R. Sarker, and K. A. Alaghbari, "An Aggregated Data Integration Approach to the Web and Cloud Platforms through a Modular REST-Based OPC UA Middleware," *Sensors*, vol. 22, no. 5, p. 1952, Mar. 2022, doi: 10.3390/s22051952.
- [17] J. Elsner, "Taming the Panda with Python: A powerful duo for seamless robotics programming and integration," *SoftwareX*, vol. 24, p. 101532, Dec. 2023, doi: 10.1016/j.softx.2023.101532.
- [18] R. Abe, F. Nagata, D. Terasaki, H. Kato, T. Ikeda, and K. Watanabe, "Proposal of hyper-CLS data for industrial robots-HCLS statements for sequence control of multiple robots," *Artificial Life and Robotics*, vol. 28, no. 4, pp. 750–756, Nov. 2023, doi: 10.1007/s10015-023-00887-7.
- [19] J. Y. Choi *et al.*, "Exploring Challenges and Opportunities in Manufacturing and Intelligence for Future Robotics," *International Journal of Precision Engineering and Manufacturing*, vol. 26, no. 9, pp. 2203–2222, Sep. 2025, doi: 10.1007/s12541-025-01318-2.
- [20] H. T. N. Le and H. Q. T. Ngo, "Application of the vision-based deep learning technique for waste classification using the robotic manipulation system," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 391–400, Dec. 2025, doi: 10.1016/j.ijcce.2025.02.005.
- [21] M. Ryalat, N. Almtireen, G. Al-refai, H. Elmoaqet, and N. Rawashdeh, "Research and Education in Robotics: A Comprehensive Review, Trends, Challenges, and Future Directions," *Journal of Sensor and Actuator Networks*, vol. 14, no. 4, p. 76, Jul. 2025, doi: 10.3390/jsan14040076.
- [22] M. Makulavičius, J. J. Petronienė, E. Šutinys, V. Bučinskas, and A. Dzedzickis, "Industrial Robotic Setups: Tools and Technologies for Tracking and Analysis in Industrial Processes," *Applied Sciences (Switzerland)*, vol. 15, no. 18, p. 10249, Sep. 2025, doi: 10.3390/app151810249.
- [23] R. Malhan, R. J. Joseph, P. M. Bhatt, B. Shah, and S. K. Gupta, "Algorithms for Improving Speed and Accuracy of Automated Three-Dimensional Reconstruction With a Depth Camera Mounted on An Industrial Robot," *Journal of Computing and Information Science in Engineering*, vol. 22, no. 3, Jun. 2022, doi: 10.1115/1.4053272.
- [24] A. A. Santos *et al.*, "Integration of Artificial Vision and Image Processing into a Pick and Place Collaborative Robotic System," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 110, no. 4, p. 159, Nov. 2024, doi: 10.1007/s10846-024-02195-z.

- [25] R. Y. 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.
- [26] J. C. Hsiao, K. Shivam, I. F. Lu, and T. Y. Kam, "Positioning Accuracy Improvement of Industrial Robots Considering Configuration and Payload Effects Via a Hybrid Calibration Approach," *IEEE Access*, vol. 8, pp. 228992–229005, 2020, doi: 10.1109/ACCESS.2020.3045598.
- [27] S. Cai *et al.*, "Fast and Interpretable 2D Homography Decomposition: Similarity-Kernel-Similarity and Affine-Core-Affine Transformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 9, pp. 7558–7576, Sep. 2025, doi: 10.1109/TPAMI.2025.3568582.
- [28] K. Kahar, S. Golam, S. Mathurkar, and P. Kale, "Industrial Robotic Arm System," *SSGM Journal of Science and Engineering*, vol. 2, no. 1, pp. 62–67, 2024.
- [29] M. Abdullah-Al-Noman, A. N. Eva, T. B. Yeahyea, and R. Khan, "Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection," *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 180–186, 2022, doi: 10.18196/jrc.v3i2.13906.
- [30] C. Han *et al.*, "Classification, Advanced Technologies, and Typical Applications of End-Effector for Fruit and Vegetable Picking Robots," *Agriculture (Switzerland)*, vol. 14, no. 8, p. 1310, Aug. 2024, doi: 10.3390/agriculture14081310.
- [31] C. H. V. Kumar and P. Sethuramalingam, "Development and control of a suction gripper system for non-contact handling of fragile thin sheets using axial flow fans," *International Journal of Intelligent Robotics and Applications*, vol. 9, no. 1, pp. 278–298, Mar. 2025, doi: 10.1007/s41315-024-00384-5.
- [32] W. Chojnacki, Z. L. Szpak, and M. Wadenbäck, "The equivalence of two definitions of compatible homography matrices," *Pattern Recognition Letters*, vol. 135, pp. 38–43, Jul. 2020, doi: 10.1016/j.patrec.2020.03.033.
- [33] S. D. Marco, M. D. Hua, R. Mahony, and T. Hamel, "Homography Estimation of a Moving Planar Scene from Direct Point Correspondence," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1284–1295, May 2021, doi: 10.1109/TCST.2019.2960483.
- [34] S. Xiang, R. Wang, and C. Feng, "Mobile projective augmented reality for collaborative robots in construction," *Automation in Construction*, vol. 127, p. 103704, Jul. 2021, doi: 10.1016/j.autcon.2021.103704.
- [35] P. Zhang, J. Luo, and X. Liang, "Symmetric Model for Predicting Homography Matrix Between Courts in Co-Directional Multi-Frame Sequence," *Symmetry*, vol. 17, no. 6, p. 832, May 2025, doi: 10.3390/sym17060832.
- [36] Y. Wang, M. Yu, G. Jiang, Z. Pan, and J. Lin, "Image registration algorithm based on convolutional neural network and local homography transformation," *Applied Sciences (Switzerland)*, vol. 10, no. 3, p. 732, Jan. 2020, doi: 10.3390/app10030732.
- [37] D. P. Thinh *et al.*, "Development of an Educational Robotic Platform with Dual Control Interfaces and Intelligent Computer Vision-Based Interaction," *FME Transactions*, vol. 54, no. 1, pp. 93–106, 2026, doi: 10.5937/fme2601093P.
- [38] M. N. Kinyua, A. W. Sichangi, and M. K. Gachari, "Development of ANN optimized affine-6 2D coordinate transformation model," *Applied Geomatics*, vol. 17, no. 1, pp. 63–81, Mar. 2025, doi: 10.1007/s12518-024-00600-8.

APPENDIX

Table 2. Sample presents a portion of the Excel-recorded data from 54 workpieces in the box at an illumination level

Color	Shape	pred_x	pred_y	corr_x	corr_y	Actual _x	Actual _y	Error before (mm)	Error after (mm)	Durati on (s)	Accura cy (%)
red	circle	2.7588	-448.54	2.7583	-448.47	2.758.12	-448.39	0.07	0.03	4.909	99.94
green	square	342.41	3.330.5	3.4234	3.331.7	3.423.05	3.332.7	0.11	0.04	5.329	99.92
blue	triangle	3.3946	-390.20	3.3940	-390.05	3.393.48	-38.986	0.12	0.06	5.319	99.88
yellow	circle	2.7469	292.45	2.7465	2.925.0	2.746.34	2.924.8	0.06	0.02	4.806	99.97
red	square	3.2287	-19.829	3.2281	-198.16	3.227.18	-198.03	0.15	0.1	5.274	99.81
green	circle	3.4336	4.4529	3.4330	4.4541	3.43293	4.4553	0.08	0.02	4.975	99.97
red	triangle	3.5488	3.7100	3.5484	3.7113	3.5476	3.7118	0.12	0.08	5.727	99.84
blue	square	2.8225	-919.70	2.822	-919.19	282.184	-917.80	0.08	0.05	5.051	99.9
blue	circle	3.7056	5.2899	3.7051	5.2910	370.432	529.06	0.14	0.08	4.965	99.84
yellow	square	3.5794	-32.572	3.5786	-325.51	357.774	-325.34	0.17	0.09	5.029	99.82
green	triangle	3.1227	243.91	3.1223	2.4396	3.1221	2.4399	0.06	0.03	5.493	99.95
green	square	3.3139	-534.85	3.31316	-534.66	331.209	-534.42	0.19	0.11	5.092	99.78
blue	triangle	2.97993	3.2557	2.97345	3.2559	2.97861	3.2560	0.1	0.07	5.473	99.85
green	circle	3.5767	529.68	3.57617	5.2971	3.5756	5.2972	0.11	0.05	5.08	99.89
blue	circle	3.4570	-337.16	3.45617	-336.98	3.45582	-336.77	0.12	0.04	4.812	99.92
red	triangle	2.8986	1.7194	2.89832	1.7198	2.8980	1.7200	0.06	0.03	5.521	99.95
red	circle	3.6794	4.8221	3.67880	4.8225	367.804	4.8227	0.14	0.08	5.283	99.85
blue	triangle	3.0432	-262.55	3.04275	-262.42	304.175	-262.24	0.16	0.1	5.37	99.8
yellow	circle	3.03787	3.8515	3.03750	3.8515	3.03707	3.8512	0.08	0.04	4.851	99.91
yellow	square	3.5720	-349.54	3.57117	-349.33	3.57044	-349.18	0.16	0.07	5.043	99.85
red	triangle	3.69286	-275.27	3.69201	-275.08	3.69171	-274.98	0.12	0.03	5.611	99.94
green	circle	2.90116	-341.56	2.90063	-34.142	2.9004	-341.39	0.07	0.02	4.683	99.96
green	square	3.72827	5.6411	3.7245	5.6413	3.7272	5.6418	0.1	0.04	5.41	99.92
blue	square	3.06394	4.8241	3.0636	4.8240	3.06345	4.8252	0.05	0.02	5.151	99.96
blue	circle	2.96530	5.0699	2.9650	5.0699	2.9648	5.0700	0.04	0.01	4.916	99.98
yellow	triangle	3.03080	-223.57	3.0302	-223.45	3.03011	-223.31	0.07	0.02	5.278	99.95
yellow	circle	3.757.65	-179.34	3.75692	-17.918	3.7566	-179.03	0.1	0.03	4.815	99.95
yellow	triangle	3.71481	4.9067	3.70994	4.9031	3.70962	4.9084	0.09	0.03	5.494	99.93
red	triangle	2.92286	-171.60	2.92243	-171.50	2.92173	-171.36	0.12	0.07	5.465	99.86
red	circle	3.7669	-237.83	3.766.25	-237.66	3.76597	-237.45	0.1	0.03	5.27	99.93

Table 2. Sample presents a portion of the Excel-recorded data from 54 workpieces in the box at an illumination level (*continued*)

Color	Shape	pred_x	pred_y	corr_x	corr_y	Actual _x	Actual _y	Error before (mm)	Error after (mm)	Durati on (s)	Accura cy (%)
green	circle	3.13560	-326.84	3.13507	-326.70	3.13453	-326.57	0.11	0.06	4.778	99.89
unkno wn	unkno wn	2.90381	389.51	2.90347	389.53	2.90251	3.8949	0.13	0.1	4.638	99.81
yellow	circle	3.64357	5.1299	3.64126	5.1215	3.64224	5.1209	0.13	0.09	5.002	99.82
green	square	2.78613	-388.38	2.78647	-388.25	2.78520	-388.14	0.1	0.05	4.954	99.91
green	triangle	3.80325	-111.57	3.80662	-111.42	380.164	-11.133	0.16	0.1	5.615	99.79
blue	circle	2.87933	1.3637	2.87889	1.3641	2.8788	1.3643	0.05	0	4.769	99.99
blue	square	3.67407	5.7429	3.67290	574.33	3.67229	5.7439	0.11	0.06	5.318	99.88
red	square	3.19583	-41.196	3.19033	-411.80	3.19102	-411.57	0.16	0.1	5.276	99.79
red	triangle	388.888	5.3388	388.83	5.3354	3.8876	5.3348	0.13	0.07	5.924	99.86
yellow	square	307.985	2.1171	3.07941	2.1176	3.07894	2.1173	0.09	0.05	5.071	99.91
red	circle	3.43266	9.6676	3.43211	96.697	3.43140	9.6719	0.13	0.07	5.09	99.86
green	triangle	3.2505	-374.58	3.25491	-374.42	3.25414	-374.27	0.14	0.08	5.352	99.84
blue	square	3.78209	59.611	3.78154	5.9613	3.78135	596.17	0.07	0.02	5.306	99.96
yellow	triangle	387.057	-285.82	3.86984	-285.64	3.86915	-285.57	0.14	0.07	5.349	99.86
red	triangle	3.16269	-282.73	3.16131	-282.60	3.16129	-282.48	0.14	0.08	5.46	99.83
green	square	3.17843	-274.40	3.17785	-274.27	3.17692	-274.12	0.15	0.09	5.054	99.81
yellow	square	3.20165	2.3819	3.20118	238.24	3.19993	2.3822	0.17	0.13	5.136	99.75
green	circle	3.62117	-426.86	3.62045	-426.68	3.61572	-426.57	0.16	0.09	4.903	99.82
red	triangle	3.16269	-282.73	3.16131	-282.60	3.16129	-282.48	0.14	0.08	5.46	99.83

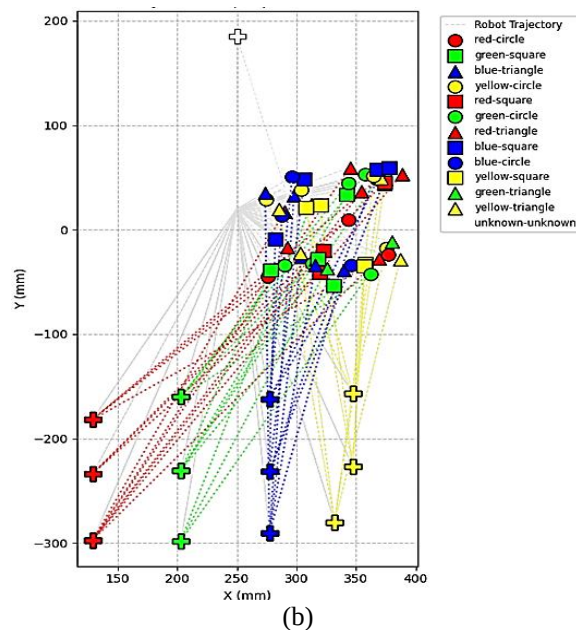
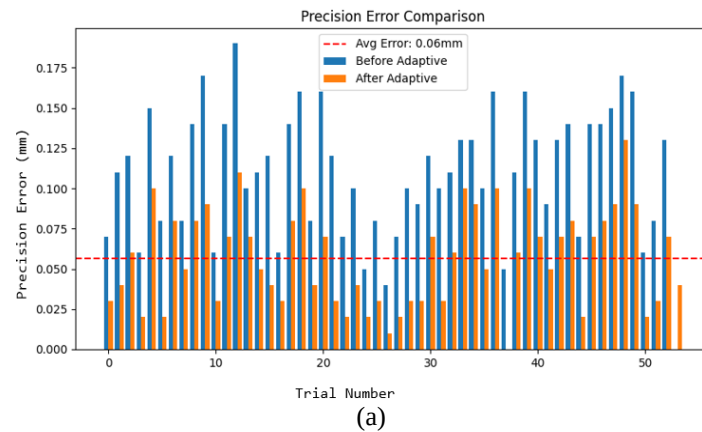


Figure 8. Plot from raw experimental data showing; (a) a bar graph of precision error comparison and (b) the robot trajectory of the 54 workpiece placement surfaces

BIOGRAPHIES OF AUTHORS

Paramust Juntarakod    is a Lecturer at Field of Mechatronics Engineering at the Rajamangala University of Technology Isan, Khon Kaen Campus, Thailand. He received the B.Eng. in Mechanical engineering from the King Mongkut's University of Technology North Bangkok in 2005, and the Master's degree in Mechanical Engineering and the Ph.D. in Mechanical Engineering from the King Mongkut's University of Technology North, in 2008 and 2013, respectively. His current research interests include robotic and automation control. He can be contacted at email: paramust.ju@rmuti.ac.th.



Viroch Sukontanakarn    is a lecturer in Mechatronics Engineering at the Rajamangala University of Technology Isan, Khon Kaen Campus, Khon Kaen, Thailand. He received his M.Eng. in Electric Power System Management and D.Eng. in Mechatronics from Asian Institute of Technology, in 1998 and 2011, respectively. He has been an Associate Professor in the Field of Mechatronics Engineering at the Rajamangala University of Technology Isan Khon Kaen Campus, Thailand since 2002. His research interests are power electronics, electrical power systems, microcontrollers, robotics, programmable logic controller, and electric motor drive. He can be contacted at email: viroch.su@rmuti.ac.th.