

Received: 10 October 2020 / Accepted: 11 November 2020 / Published online: 29 November 2020

*small and medium size enterprises,
programming by demonstration,
circle Hough transform,
machine vision*

Aly M. EISSA^{1*}
Mostafa R. ATIA¹
Magdy R. ROMAN²

AN EFFECTIVE PROGRAMMING BY DEMONSTRATION METHOD FOR SMES' INDUSTRIAL ROBOTS

Traditional programming methods often require expertise and significant time investment, which does not conform with Small and Medium size Enterprises (SMEs) nature in which High-Mix, Low-Volume (HMLV) orders are usually encountered. In this research, a Programming by Demonstration (PbD) method which aims to reduce the programming time and complexity while keeping a suitable level of execution accuracy is proposed. For this purpose, a special teaching tool is designed and manufactured. The tool has 5-spherical passive markers to indicate the position and orientation along the desired 3D path. An optical tracking system using stereo camera is used to capture the 3D pose of the teaching tool. The capturing algorithm is based on Circle Hough Transform (CHT) and Singular Value Decomposition (SVD). The developed tool and programming method have been tested experimentally. The results show successful capturing of the desired path points with a competitive level of accuracy compared with other methods.

1. INTRODUCTION

Robot programming has become one of the difficulties which face the users and still in the focus of many researchers' interest. Generally, there are two types of robot programming methods: offline programming and online programming. In offline programming methods the operator does not interfere with the robot environment. The program is written in the conventional text-based way or by using a CAD model of the workpiece. Virtual Reality (VR) is sometimes used to simulate the planned task before program execution. Offline programming methods are, however, inflexible to environment and workpiece changes due to the need to re-texting or re-modeling to compensate for the changes. This makes the offline programming methods suitable only for large batch size and often require technician specialized in software. On the other side, online robot

¹ Mechanical Engineering Department, College of Engineering and Technology-Cairo Campus, Arab Academy for Science, Technology and Maritime Transport (AASTMT), Heliopolis, Cairo, Egypt

² Mechanical Power Engineering Department, Faculty of Engineering, El Materia Helwan University, Cairo, Egypt
E-mail: engalieissa@gmail.com

<https://doi.org/10.36897/jme/130944>

programming is where the program code is generated while the robot devices are ON. The two traditional online methods are the lead-through and the walk-through programming. The lead-through programming approach relies on the use of a teach pendant where the programmer moves the robot to the desired position with specified velocity. It is only suitable for producing large lot size [1]. The walk-through approach allows the user to physically move the robots' end effector through the desired trajectory and the controller records the trajectory. Thus, the robot can be programmed in a very intuitive manner. However, walk-through programming is restricted due to the effect of mechanical loads, which make the movement of the robot difficult. An extension to walk-through programming is provided by the concept of Programming by Demonstration (PbD) where the operator shows the robot how the task should be performed. It enables users to construct programs without writing codes.

2. RELATED WORKS

The Robot PbD has a diverse of teaching tools, each has its own advantages and drawbacks. One of the most popular teaching tools used in PbD is the marker. Markers can be passive or active. Passive markers are made of retro-reflective materials, which reflect light in the direction from where it comes. Consequently, cameras are equipped with a band of infrared light emitting diodes (LEDs) where the reflected light from the marker back in the direction of the camera. This may, results in low Signal to Noise Ratio (SNR) as some reflective surfaces in the scene may be handled as markers. This necessitates adding extra complexity in processing the information sent by the tracking system. Kulić et al. [2–4] proposed an approach for incremental learning of full body motion with fingers from observation of human motion. Antonelli et al. [5] presented a new probe equipped with two reflective markers at the extremes and a stick at one end to physically touch the measurement points. It is used for welding a real-world curve on the free-form object. The standard deviation of the reconstructed 3D positions for reproducibility tests was (3.45, 2.10, 2.05) mm. Antonelli and Astanin [6] made also an edit for the previous marker which equipped with three spherical IR reflective markers and a spherical endpoint. The markers are arranged in a scalene triangle, thus the three markers are sufficient to calculate tool's location and orientation. Experimental validation on a 2D plate showed a maximum error of 2.6 mm in the end-point location. It has standard deviation components of (0.014, 0.015, and 0.15) mm in x, y, and z directions respectively. Mueller et al. [7] presented an off-line method to program a welding robot. The pointing device, whose spatial position is traced, was equipped with five infrared markers. The markers are captured by MoCap system with four cameras. The results showed a deviation distance between captured trajectory and a given spline function within a range of 2.12 mm to 3.08 mm. Chen et al. [8] presented an Augmented Reality (AR)-based interactive robot teaching programming system. The system virtually projects the robot onto the physical industrial environment. The maximum tracking error of the used handheld teaching device through its motion along the edge of a rectangular frame was 3.231 mm. Active markers on the other hand are equipped with LEDs which require power source to emit their own light. Active markers are more expensive to fabricate, and it is usually harder to replace than the passive markers.

Moreover, ghost markers may be seen as the markers light reflects in shiny surfaces. Aleotti and Caselli [9] presented a virtual demonstration environment which comprises a virtual reality glove and a motion tracking device. Skoglund et al. [10] presented an approach to PbD based on a next-state-planner with fuzzy modeling learning algorithm. For the robot to interpret a demonstration the hand-state space is used which allows the robot to see the demonstrated motion as its own. Experimental validation showed a position accuracy in xyz axes of about (6, 4,12) mm. Ruppel et al. [11] presented a low-cost system for visual motion tracking which supports fusion with IMU (Inertial Measurement Unit) data. The average marker tracking position error is about of 0.26 mm and a median marker position error of 0.17 mm. Ferreira et al. [12] presented new tool based on a luminous marker built with high-intensity LEDs. The markers are captured by a set of industrial cameras. The proposed marker included 20 visible-light (RGB) LEDs. As the results indicated, the hue, saturation, and value (HSV) color space was useful for this classification. The max absolute error in position was (3.6, 8.2, 5.4) mm and in rotation was (4.1, 3.9, 3.6) degree.

Duque et al. [13] presented a methodology for the execution of parts assembly operations, using Learning by Demonstration (LbD) techniques, such as TP-GMM (Task Parametrized Gaussian Mixture Model). The technique was used to generate trajectories based on a set of parameters that characterize the task. Wherefore, the results showed a maximum error of 1.5 cm. in position, and 10 degrees in orientation. Kyrarini et al. [14–16] presented a framework that permits a robot manipulator to learn how to execute structured tasks from human demonstrations which is automatically segmented into basic movements. The proposed system combined physical human robot interaction with attentional supervision in order to support kinesthetic teaching. Incremental learning and DTW (Dynamic Time Warping) methods were implemented to normalize the demonstrated observations in the same time domain. Fang et al. [17, 18] presented handheld cubic tool for the purpose of human-robot interaction in task and path planning processes. Four different planar markers are attached to the faces of the cube. Each marker consists of two-dimensional matrix code, which encode data in a pattern of black and white squares. The system can be calibrated, and the tool is tracked continuously using the AR Toolkit tracking method. The average tracking error of acquiring 3D data points was about 11 mm. Rabbi et al. [19] made comprehensive analysis about fiducial marker attributes of AR Toolkit. This analysis has shown that the tracking accuracy is affected by marker size, marker distance from camera, marker speed, the brightness in the environment, and finally the contrast level of lighting. More researches that emphasize the necessity of finding flexible and more intuitive robot programming method include using robot in measurement [20] and inspection [21].

As discussed before, the HMLV orders character of SMEs requires searching for a flexible, cost-effective method of robots programming. Conventional programming is only cost-effective for large batch sizes. By analyzing the above literature, it is found that PbD methods can partially meet the requirements of SMEs since it does not require experienced operators or CAD designs. However, for better achievement of the requirements the following factors should be taken into consideration while designing the tracking system in PbD methods:

- Reducing programming cost: This includes reducing the number of high resolution cameras, finding the best design of the teaching tool, and using passive ways as possible.
- Accuracy and robustness: Tracking system should be robust and accurate specially against lighting condition changes (intensity and light direction), reflective surfaces, and ghost markers should be kept at acceptable level.

In this research we target the above-mentioned factors by investigating an effective programming by demonstration method for SMEs' industrial robots. The method uses a specially designed teaching tool with 5-spherical passive markers and two single cameras. The markers are robustly detected using CHT feature extraction technique. The method is validated experimentally using well-defined 3D points under different lighting conditions. The results show effectiveness of the proposed method for SMEs' robot programming applications.

The rest of the paper is organized as follows: Section 3 presents the proposed method with details of its hardware setting and implementation. Section 4 presents the system components which used in this research. Section 5 presents and discusses the results and the experimental work. Lastly, Section 6 concludes the research and proposes future research opportunities.

3. PROPOSED SYSTEM

3.1. MARKER DESCRIPTION

The proposed marker is of the passive type, it consists of 5 non-coated different-sized spheres, see Fig. 1. The spheres (markers) are arranged such that its centers are located on an imaginary spherical surface of radius 67 mm. The sphere equation can be applied for all markers' centers to calculate the center of the imaginary sphere.

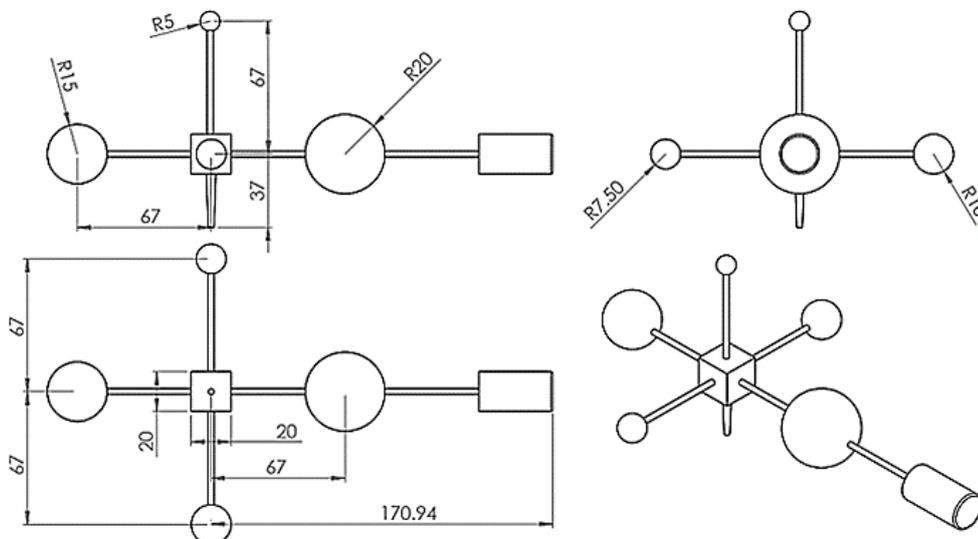


Fig. 1. 3D-Assembly model of the marker

The teaching tool pointer is 37 mm shifted from the center of the imaginary sphere. However, the reason of using five spherical markers is to substitute for the occluded markers. Moreover, increasing the number of markers enhances the accuracy of the estimated pose of the teaching tool. The different size of the markers is used to distinguish each marker center on the imaginary sphere.

3.2. MARKER DETECTION

The 3D retrieving of the markers position in the world coordinates is based on a two-cameras stereo vision. The calibrated stereo setup holds the geometric description of the cameras' pose and their intrinsic parameters. Given two corresponding pixels in two images the reconstructed 3D position is determined using the Triangulation and Sampson correction algorithm [22, 23]. The first stage is to identify the markers in two images for the same scene. This is done by using Circular Hough Transform (CHT) technique since the projection of the markers are always circular. CHT is a feature extraction technique to detect circles in images even if the circle is incomplete [24, 25]. First, all edges in the image are detected which can be done using Canny, Sobel or Morphological operations [26–28]. A voting procedure is then carried out in a parameter space and the characteristic points of circles are determined. After that the pattern of the circles is extracted and with the help of circles radius, the center pixels of the circles are estimated [29].

After using CHT method to detect circles in both images a designed matching algorithm is used to match the detected circles. Figure 2 describes the block diagram of the algorithm of the developed methodology. The algorithm compares the circles diameter in the two images within the specific range of diameters for each marker throughout the workspace.

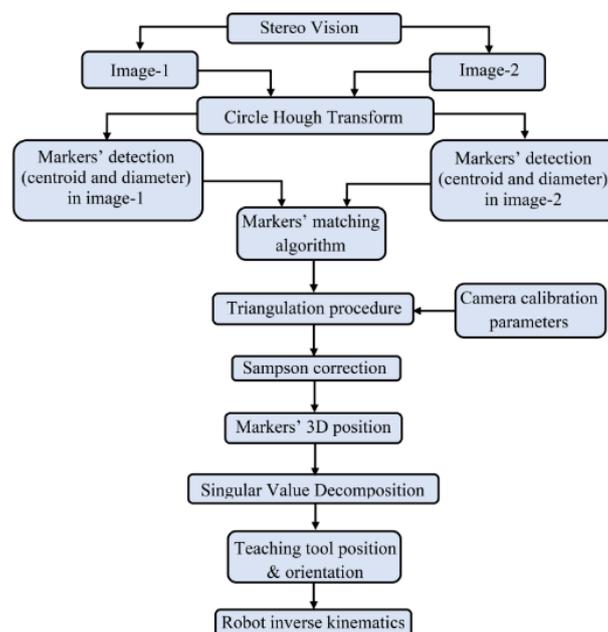


Fig. 2. Algorithm block diagram of the developed methodology

Figure 3 shows the detected markers in both images. After the matching process is done the 3D reconstructed position of each marker is determined as discussed above. The next sections describe in details the methods used in each stage.

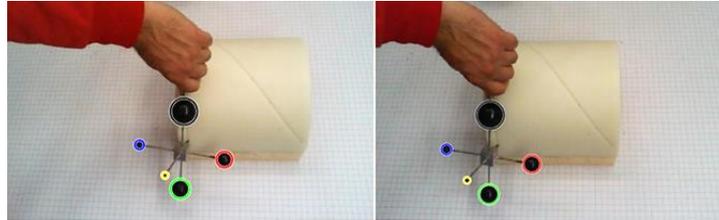


Fig. 3. Markers detection in stereo camera's images

3.3. TEACHING TOOL POSE ESTIMATION

3.3.1. POSITION ESTIMATION

Since the markers' centers are located on a surface of an imaginary sphere the problem of finding the teaching tool origin is reduced to finding this sphere's center. Given at least four points, which are non-coplanar and any three of them are not collinear, the sphere's center in 3D is given as follows:

$$(x_i - x_t)^2 + (y_i - y_t)^2 + (z_i - z_t)^2 = r^2, \quad (1)$$

where, r is the sphere radius, x_t, y_t, z_t are the coordinates of the sphere's center, x_i, y_i, z_i are the marker center coordinates, $i = 1, \dots, N$, and N is the number of identified markers. Obviously, these equations contain quadratic terms. By subtracting one equation from others we get a set of linear equations of the form:

$$a_{1i}x_t + a_{2i}y_t + a_{3i}z_t = b_i, \quad (2)$$

which can be expressed as,

$$\mathbf{A} \mathbf{x}_t = \mathbf{b}, \quad (3)$$

where \mathbf{A} is a matrix of coefficients ($N \times 3$), \mathbf{x}_t is the required coordinates vector and \mathbf{b} , ($N \times 1$), is a vector of the equations' RHS. Since the system described has intrinsic errors due to the accuracy of both the tracking system and the teaching tool manufacturing it can be solved in sense of least-squares problem, $|\mathbf{A} \mathbf{x}_t - \mathbf{b}|^2$. This can be done using Singular Value Decomposition (SVD) method.

3.3.2. ORIENTATION ESTIMATION

Arun et al. [30] presented a method to find the rotation between two data sets of corresponding 3D points. The method estimates the rotation matrix that best describes the

relative orientation between two data sets (in the sense of root-mean-square error). In Fig. 4, data set B represents the teaching tool markers' 3D positions, while A represents the corresponding reference-frame-aligned data set. A and B contain a number N of 3D points elements A_i and B_i , where ($i = 1, \dots, N$). The sequence of calculation is as follows. In order to get the relative rotation a minimum of three corresponding points are required. First, both data sets must be origin centered; this is done by subtracting them from their respective origins [30]. Second, the origin-centered data set A and B are used to calculate the covariance matrix H in the form:

$$H = \sum_{i=1}^N A_i' B_i^t \quad (4)$$

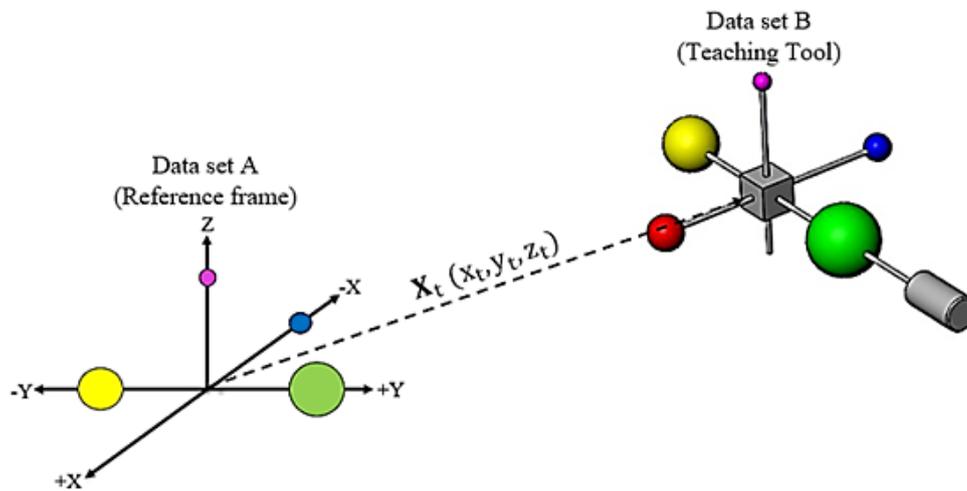


Fig. 4. Teaching tool orientation estimation

The third step is to calculate the singular value decomposition of H ; $H = U S V^T$. Then, the optimal orientation matrix is:

$$R^* = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} U^T, \quad (5)$$

$$q = \text{sign}(\det(V U^T)), \quad (6)$$

where, q is a parameter used to specify the right-handed coordinates system [12].

4. EXPERIMENTAL SETUP

In order to validate the proposed methodology an experimental setup is built which contains two fixed cameras, a host computer, the teaching tool, and 3D validating bodies, see Fig. 5. The cameras (life cam HD Microsoft cameras-1080p) capture the teaching tool at the different waypoints and send the images to the host pc (Intel Core-i5 4200) for image processing and running of the PbD algorithm. MATLAB of MathWorks company is used

for this purpose. Fig. 6 shows a half-cylindrical body fabricated by a CNC milling machine of accuracy ($4\ \mu\text{m}$). The body has two paths which contain a number of 4 mm depth holes. The holes are used to fix the teaching pointer. The body is used to validate the proposed method for position estimation. On the other hand, Fig. 7 show another 3D body used for validating the orientation estimation. The body contains two setting bases for the teaching tool. One base is parallel to xy plane; the second one inclinds with 45° to the xy plane. The teaching tool on each base can be rotated by $0^\circ, 90^\circ, 180^\circ$ and 270° . The two bodies, Figs. 5 and 6 are set on a graph paper to measure their location relative to the reference coordinates.

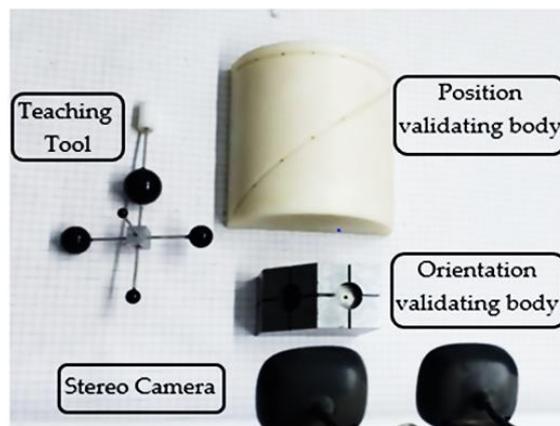


Fig. 5. Experimental setup

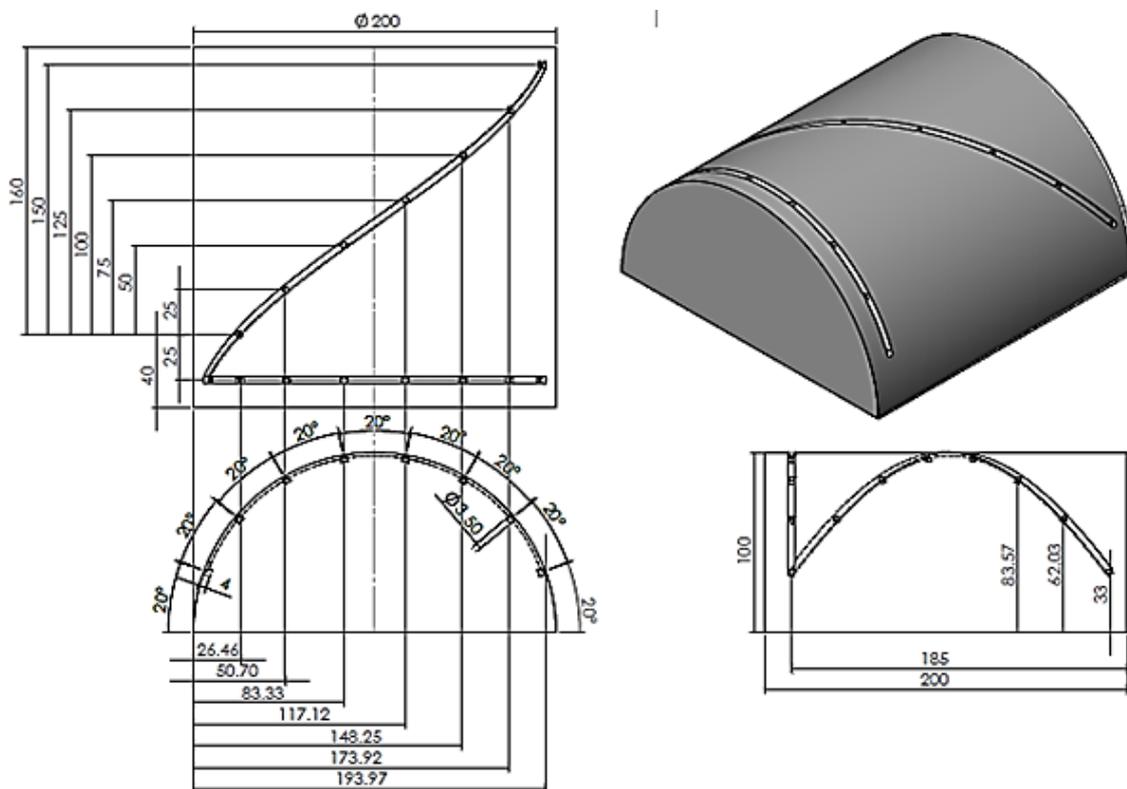


Fig. 6. Half-cylindrical body used for position validation

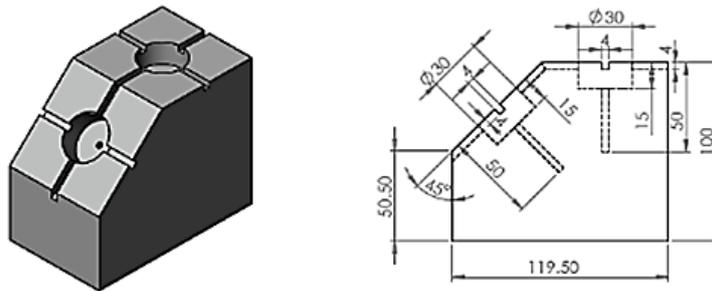


Fig. 7. 3D-Metallic object used for orientation validation

5. RESULTS

5.1 POSITION TEST

Fig. 8 shows the validation process for position estimation where the teaching tool is set on the well-defined holes of the half-cylindrical body. Fig. 9 shows the absolute error of the teaching tool position (x, y, z). The maximum recorded value is about 5 mm which is more accurate compared with [11–14]. The root mean square error (RMSE) for all the tested points is (0.75, 1.61, 2.42) mm. Fig. 10 illustrates an implementation for the tracking teaching tool position along 3D path. The results show that the proposed system is competitive, and less cost compared with the infrared tracking system which costs around 3300 Euro for the configuration with two cameras. The developed system in the current research costs around 1200 Euro.

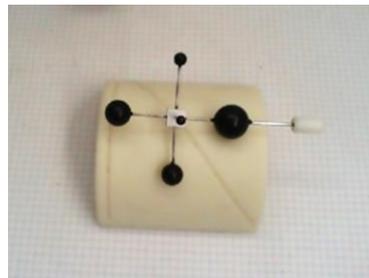


Fig. 8. Teaching tool position validating process

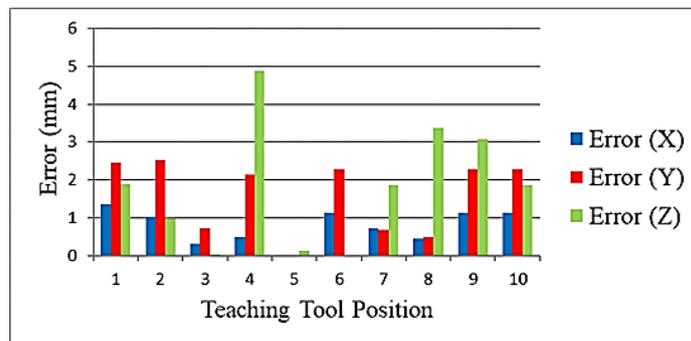


Fig. 9. Teaching tool position error estimation

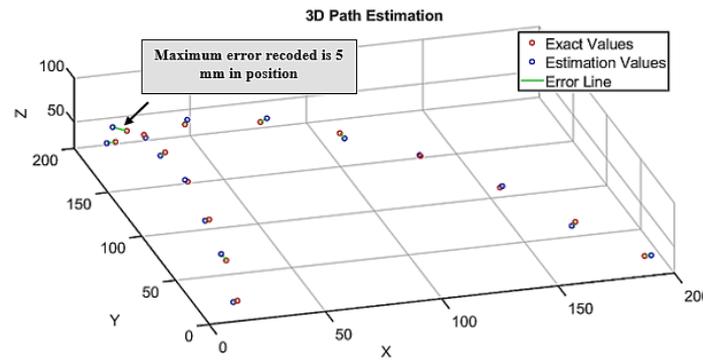


Fig. 10. Tracking for teaching tool position along 3D path

5.2. ORIENTATION TEST

The same process is repeated to validate the estimation of the teaching tool orientation. As shown in Fig. 11, the teaching tool is set on a well-defined orientation using the 3D specially designed metal object. Ten different orientations are selected randomly throughout the workspace. The results are recorded in terms of error angles of rotation ($e_\alpha, e_\beta, e_\gamma$) between the actual and estimated X, Y, Z axes respectively. Fig. 12 shows the absolute error in teaching tool orientation ($e_\alpha, e_\beta, e_\gamma$). The maximum recorded error is 5.4° and the RMSE for all orientations is ($1.11^\circ, 2.18^\circ, 2.99^\circ$).

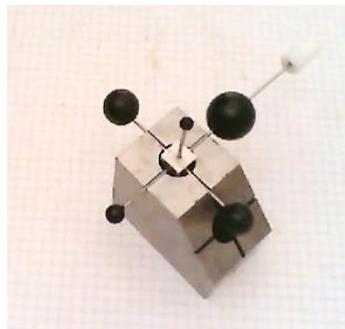


Fig. 11. Teaching tool orientation validating process

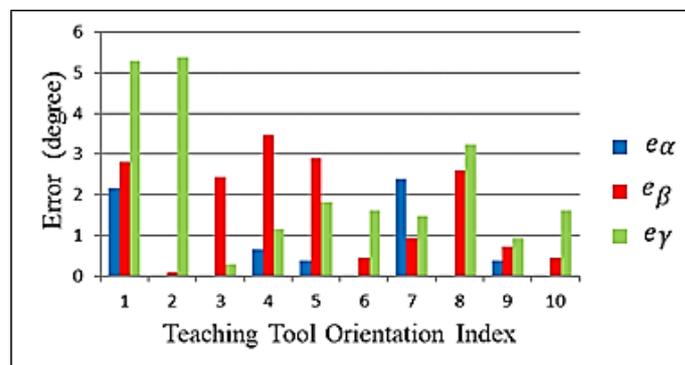


Fig. 12. Teaching tool orientation error estimation

5.3. LIGHTING CONDITION ROBUSTNESS TEST

Finally, the proposed method robustness against different lighting conditions is tested. The teaching tool is placed in a well-defined pose and the algorithms are run on different lighting conditions, see Fig. 13. The lighting conditions are changed both in intensity and direction of illumination; distracted light (Fig. 13 a–e), overhead light (Fig. 13 f–h), and inclined light (Fig. 13 i–j).

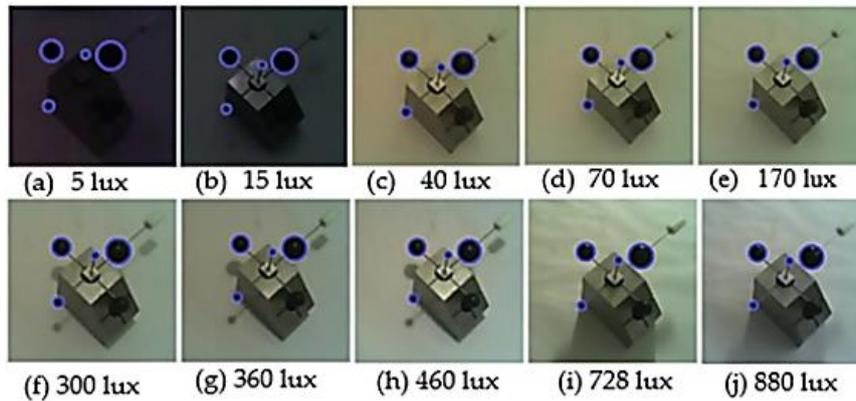


Fig. 13. Lighting condition robustness test

In each lighting condition both position and orientation errors are estimated, see Fig. 14. The maximum position error is found to be 4.8 mm and the maximum orientation error is 4.97 degree. The standard deviation for the teaching tool position is (2.38, 2.73, 1.76) mm and for the orientation is (2.52, 1.42, 1.55) degree.

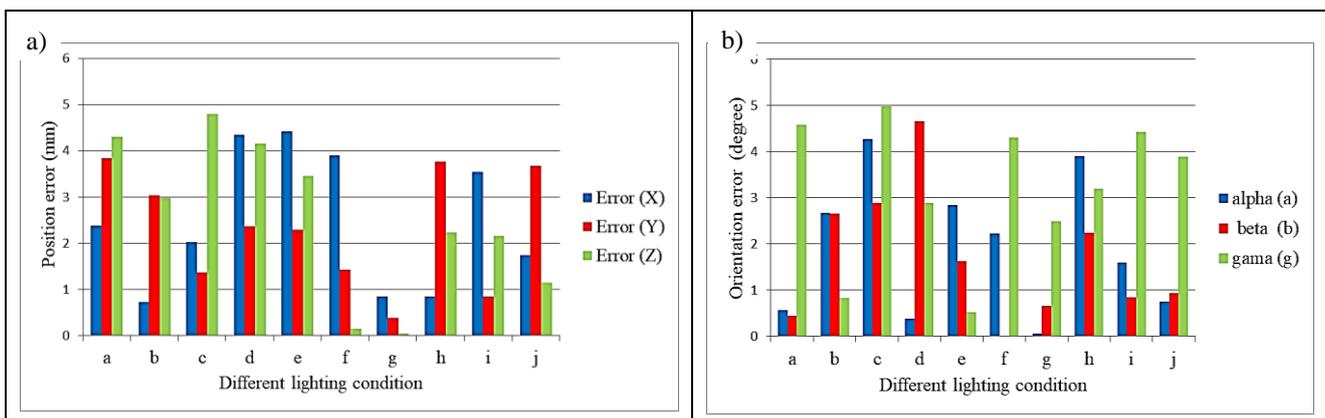


Fig. 14. Pose measurement uncertainty at different lighting conditions (a) position errors (b) orientation errors

6. CONCLUSION

In this research a PbD method for programming industrial robots, particularly used in SMEs, is presented. A simple passive teaching tool is used together with the robust method of feature extraction, CHT. A matching algorithm is developed to match the detected

markers in two images of the teaching tool. Markers' 3D positions are reconstructed using Triangulation and Sampson correction algorithms. SVD method is used to determine teaching tool position and orientation in the space. The developed tool and programming method have been tested experimentally using well-defined 3D paths and orientations. The results showed competitive level of accuracy compared with more complex methods of using more cameras and active systems. The results also showed robustness against different lighting conditions ranging from 5 lux – 880 lux and under different illumination directions. The research is considered a step forward to meet the requirements of programming industrial robots in SMEs. Future work on the research contains two consecutive steps: (1) robot tool-center-point (TCP) position and tool orientation will be considered. (2) the system will be customized for a specific industrial application.

REFERENCES

- [1] DIETZ T., 2012, *Programming System for Efficient Use of Industrial Robots for Deburring in SME Environments*, ROBOTIK 2012; 7th German Conference on Robotics, VDE.
- [2] KULIĆ D., OTT C., LEE D., 2012, *Incremental Learning of Full Body Motion Primitives and Their Sequencing Through Human Motion Observation*, The International Journal of Robotics Research, 31/3, 330–345.
- [3] TAKANO W., MURAKAMI Y., NAKAMURA Y., 2020, *Representation and Classification of Whole-Body Motion Integrated with Finger Motion*, Robotics and Autonomous Systems, 124, 103378.
- [4] ALEXANDERSON S., O'SULLIVAN C., 2017, *Real-Time Labeling of Non-Rigid Motion Capture Marker sets*, Computers & Graphics, 69, 59–67.
- [5] ANTONELLI D., 2013, *Training by Demonstration for Welding Robots by Optical Trajectory Tracking*, Procedia CIRP, 12, 145–150.
- [6] ANTONELLI D., ASTANIN S., 2016, *Qualification of a Collaborative Human-Robot Welding Cell*, Procedia CIRP, 41, 352–357.
- [7] MUELLER F., DEUERLEIN C., KOCH M.J.I-P., 2019, *Intuitive Welding Robot Programming via Motion Capture and Augmented Reality*, IFAC-PapersOnLine, 52/10, 294–299.
- [8] CHEN C., 2020, *A Virtual-Physical Collision Detection Interface for AR-Based Interactive Teaching of Robot*, Robotics and Computer-Integrated Manufacturing, 64, 101948.
- [9] ALEOTTI J., CASELLI S., 2012, *Grasp Programming by Demonstration in Virtual Reality with Automatic Environment Reconstruction*, Virtual Reality, 16/2, 87–104.
- [10] SKOGLUND A., ILIEV B., PALM R., 2010, *Programming-by-Demonstration of reaching motions—A next-state-planner approach*, Robotics and Autonomous Systems, 58/5, 607–621.
- [11] RUPPEL P., HENDRICH N., ZHANG J., 2019, *Low-Cost Multi-View Pose Tracking Using Active Markers*, IEEE International Conference on Industrial Cyber Physical Systems (ICPS), 18882306.
- [12] FERREIRA M., 2016, *Stereo-Based Real-Time 6-DoF Work Tool Tracking for Robot Programming by Demonstration*, The International Journal of Advanced Manufacturing Technology, 85/1–4, 57–69.
- [13] DUQUE D.A., 2019, *Trajectory generation for robotic assembly operations using learning by demonstration*, Robotics and Computer-Integrated Manufacturing, 57, 292–302.
- [14] KYRARINI M., 2019, *Robot Learning of Industrial Assembly Task via Human Demonstrations*, Autonomous Robots, 43, 239–257.
- [15] CACCAVALE R., 2019, *Kinesthetic Teaching and Attentional Supervision of Structured Tasks in Human–Robot Interaction*, Autonomous Robots, 43, 1291–1307.
- [16] LUO J., 2020, *A Task Learning Mechanism for the Telerobots*, International Journal of Humanoid Robotics, 16/2, 1950009.
- [17] FANG H., ONG S., NEE A., 2012, *Interactive Robot Trajectory Planning and Simulation Using Augmented Reality*, Robotics and Computer-Integrated Manufacturing, 28/2, 227–237.
- [18] FANG H., ONG S.K., NEE A.Y-C., 2009, *Robot Programming Using Augmented Reality*, International Conference on CyberWorlds, IEEE, 13–20.
- [19] RABBI I., 2017, *Analysing the Attributes of Fiducial Markers for Robust Tracking in Augmented Reality Applications*, International Journal of Computational Vision and Robotics, 7/1-2, 68–82.

- [20] KUTS V., 2016, *Robot Manipulator Usage for Measurement in Production Areas*, Journal of Machine Engineering, 16/1, 57–67.
- [21] SANDAK J., ORLOWSKI K.A., 2018, *Machine Vision Detection of the Circular Saw Vibrations*, Journal of Machine Engineering, 18/3, 67–77.
- [22] HARTLEY R.I., STURM P., 1997, *Triangulation*, Computer Vision and Image Understanding, 68/2, 146–157.
- [23] HARTLEY R., ZISSERMAN A., 2003, *Multiple view Geometry in Computer Vision*, Cambridge University Press.
- [24] YUEN H., 1990, *Comparative Study of Hough Transform Methods for Circle Finding*, Image and Vision Computing, 8/1, 71–77.
- [25] DAVIES E.R., 2004, *Machine Vision: Theory, Algorithms, Practicalities*, Elsevier, Morgan Kaufmann.
- [26] PEDERSEN S.J.K., 2007, *Circular Hough Transform*, Aalborg University, Vision, Graphics, and Interactive Systems, 123/6.
- [27] KERBYSON D., ATHERTON T., 1995, *Circle Detection Using Hough Transform Filters*, Fifth International Conference on Image Processing and its Applications, 370–374.
- [28] CUEVAS E., 2012, *Fast Algorithm for Multiple-Circle Detection on Images Using Learning Automata*, IET Image Processing, 6/8, 1124–1135.
- [29] ATHERTON T.J., KERBYSON D.J., 1999, *Size invariant circle detection*, Image and Vision Computing, 17/11, 795–803.
- [30] ARUN K.S., 1987, *Least-Squares Fitting of Two 3-D Point Sets*, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, 9/5, 698–700.