

# Vision Based Pose Estimation of Multiple Peg-in-Hole for Robotic Assembly

Pitchandi Nagarajan<sup>1(✉)</sup>, S. Saravana Perumaal<sup>1</sup>, and B. Yogameena<sup>2</sup>

<sup>1</sup> Department of Mechanical Engineering, Thiagarajar College of Engineering,  
Madurai 625015, India

{pnmech, sspmech}@tce.edu

<sup>2</sup> Department of Electronics and Communication Engineering,  
Thiagarajar College of Engineering, Madurai 625015, India

ymece@tce.edu

**Abstract.** Vision sensors are used to estimate the pose (position and orientation) of mating components in a vision assisted robotic peg-in-hole assembly which is a crucial step in aligning the mating hole-component with the corresponding moving peg-component. The accuracy of this estimation decides the performance of peg-in-hole robotic assembly with an appropriate mapping between the image and task environment using a fixed overhead camera or camera on robot arm. The wheel and hub assembly in automobile has multiple holes and pegs in their mating parts which lead to more complex pose estimation procedure. The success rate of the assembly process (without jamming) is affected by an inaccurate pose estimation which leads to lateral and/or axial misalignment between the mating components during its insertion phase. On this consideration, this work proposes a pose estimation algorithm for a multiple peg-in-hole assembly with the use of genetic algorithm based two-stage camera calibration procedure. The proposed algorithm has also been tested for its performance in estimating the pose of the multiple pegs in wheel-hub of a car. The result reveals that the proposed method estimates the pose of the pegs accurately with minimum re-projection error.

**Keywords:** Pose estimation · Multiple peg-in-hole · Camera calibration · Genetic algorithm · Robot assembly

## 1 Introduction

Vision sensors are the predominantly used sensors in a robotic assembly system to align, manipulate and assemble the mating component in the robot end-effector with the other mating part of the assembly. The feedback of the vision sensor is used to sense the dynamic mating component in working environment and to improve the alignment of mating components. An assembly has two components: (a) protruded part on the component often termed as peg and (b) hollow part on the mating component named as hole. Often the assembly task is horizontal or vertical insertion of peg component into the hole component to establish a permanent contact. The wheel and wheel-hub assembly of an automobile has multiple pegs and multiple-holes. In these assemblies, the wheel-hub is fixed with the automobile and the hole component (wheel)

is inserted over the multiple pegs. The success rate of the assembly without jamming or wedging is affected by the presence of lateral or angular misalignment of the components. The precision and speed of alignment is influenced by the estimation of position and orientation (pose) of the multiple-peg component [4].

The process of controlling the robotic manipulators using the feedback of the vision sensors is named as Visual servoing. Main shortcomings of a visual servoing system are poor accuracy as well as stability, due to even small pose estimation error [2], and retaining the considered object always in the field of view [3]. Pose of the object cannot be accurately estimated due to non-trackability of the object, roughly calibrated camera and error in 3D model of the target [9].

An object tracking or recognition system depends on the efficiency of the feature extraction system for faster and accurate position estimation in pixel coordinates. Scale Invariant Feature Transform (SIFT) [6, 13] and Speed Up Robust Feature (SURF) [1] are commonly adopted to extract the target features from the video frames. Based on the geometrical features of the object, few researchers [10, 18] have used morphological image processing operations like boundary extraction and gradient operations like edge detection, circle detection, ellipse detection algorithms for feature tracking. The object of interest in this work is a wheel-hub which has four pegs (cylindrical rods) and screws on its surface. These features make the feature tracking based on gradient operations adaptable in this work.

Camera calibration is the process of determining the relationship between real world (metric) information and the 2D image information [19]. The properties of the camera used (intrinsic parameters) for acquisition and parameter set describing the geometric relation between the Cartesian world space coordinate system and the image coordinate system (extrinsic parameters) are determined using this calibration techniques. The presence of the lens distortion in the camera displaces the coordinates non-linearly. The distortion in the lenses is due to the errors in assembly of lens and the geometric features of the lens [15]. Various distortions like radial, tangential and thin prism distortion are possible in images [16]. Radial distortion is common in machine vision cameras which causes pincushion and barrel effect on the images [15]. Direct linear Transformation (DLT) of camera parameters lack the capability to incorporate non-linear distortion in the camera model. Therefore, direct non-linear transformation or two-stage camera calibration [5, 14, 17] is advantageous to include distortion and estimate the camera parameters. The performance of the direct non-linear transformation requires precise initial guess which is crucial. This makes the two-stage camera calibration technique suitable for the proposed work. Hence, a genetic algorithm based two stage camera calibration is adopted in this work.

On the consideration of above said issues, this work is aimed at developing an accurate pose estimation algorithm for determining the pose of the multiple pegs with minimum re-projection errors. This paper is organized as follows: Sect. 2 presents the vision assisted multiple peg-in-hole robot assembly environment, Sect. 3 explains the proposed pose estimation algorithm, Sect. 4 depicts the experimentation of the proposed calibration and pose estimation algorithm, Sect. 5 presents the results and discussion of the performance of the proposed work and Sect. 6 provides the conclusion of this work.

## 2 Vision Assisted Multiple Peg-in-Hole Robot Assembly Environment

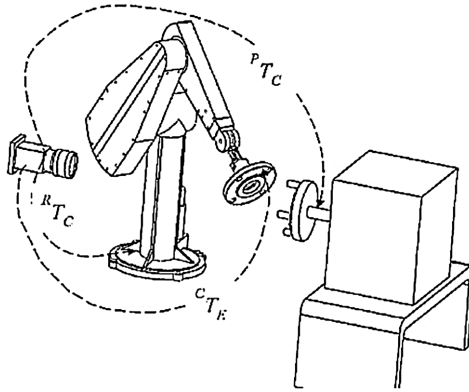
Vision sensors are used to perceive the changes in the assembly environment and to control the manipulator in performing the assembly operation in accordance to the changes. Figure 1, shows a vision assisted robotic assembly environment considered in this work to perform a multiple peg-in-hole assembly. In the considered environment, the multiple-peg component is mounted on a shaft which rotates about its center axis. The camera (vision sensor) is mounted at a fixed position in the robotic assembly environment such that the multiple-peg components lie in the field of view of the camera at all instances. The camera and the peg-component are mounted such that the optical axis of the camera and the z-axis of the component lies in the same plane. Hence, the distance between the peg component and the camera remains same because only rotation about the z-axis is allowed. Since the distance remains same, monocular camera is sufficient and adopted in this work to estimate the coordinates of the peg centers. The robotic manipulator has the mating multiple-hole component in its end-effector. In order to assemble the mating components, the co-ordinate frames of the hole and peg components have to be aligned and then the insertion task has to be executed. Pose of the multiple-peg component  ${}^C T_P$  is required to align the manipulator with axes of the mating components.  ${}^C T_E$  represents the transformation matrix of end effector with respect to camera. The current pose of the end-effector with respect to the robot base is given as:

$${}^R T_E = {}^R T_C {}^C T_P {}^P T_E \quad (1)$$

Where,

${}^R T_C$  is the pose of the camera with respect to robot base (known).

${}^C T_P$  denotes the pose of the multiple-peg part in the table with respect to the camera in the current position.



**Fig. 1.** Vision assisted multiple peg-in-hole robotic assembly environment

${}^P T_E$  represents the pose of the end-effector with respect to the hole part in the current frame and is given by (2)

$${}^P T_E = [{}^C T_P]^{-1} C T_E \quad (2)$$

In visual servoing,  ${}^C T_P$  is computed through pose estimation algorithm and compared with the target  ${}^C T_P^*$  at every instant to minimize the error and to generate the  ${}^R T_E$  for robot manipulation. Precise camera calibration procedure is required to compute the  ${}^C T_P$  accurately in the Cartesian space which enables the accurate alignment of mating components.

### 3 Proposed Pose Estimation Algorithm

The proposed pose estimation algorithm is divided into two modules: position estimation module and orientation estimation module. In the position estimation module, the multiple-pegs are tracked and the pixel coordinates of the wheel-hub and the peg centers are estimated using the feedback from the monocular camera. The pixel coordinates are fed to genetic algorithm based camera calibration to estimate the camera parameters and the position of centers in metric coordinates. Using the position of the centers in metric, the orientation estimation module estimates the pose of the wheel-hub with respect to the camera. Figure 2, shows the overview of the proposed pose estimation algorithm.

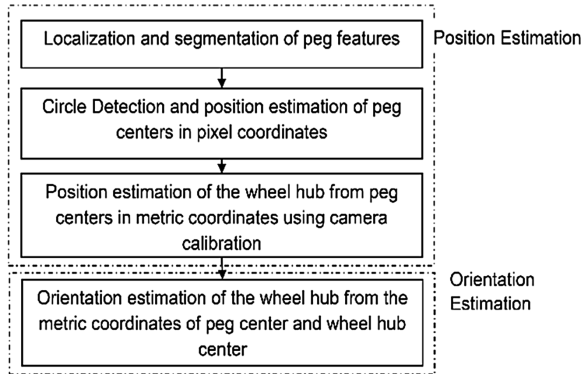
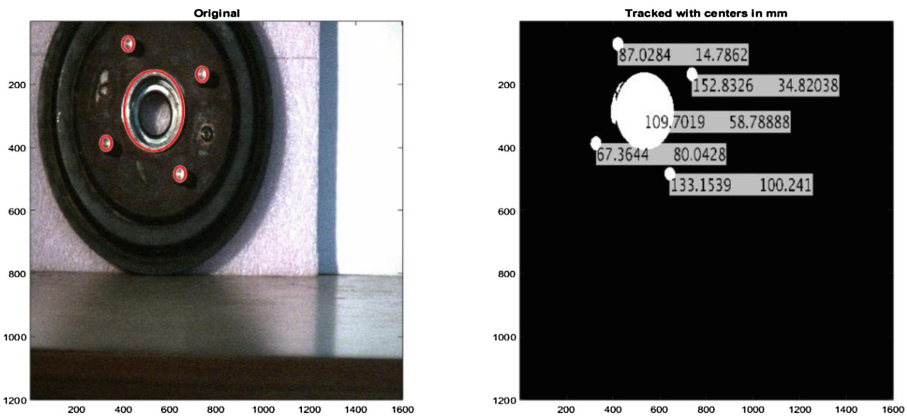


Fig. 2. Overview of the proposed pose estimation algorithm

#### 3.1 Position Estimation Module

The wheel-hub used in this work has four pegs. Determining the centers of each peg gives the position of the wheel-hub in pixels at tracking stage followed by metric position in calibration stage.

**Multiple-pegs Tracking.** In this module, each peg on the wheel-hub is tracked using the monocular camera frames. Since the top surface of the pegs and the component base area of the wheel-hub are at different planes, there is a significant change in their intensity values. This intensity difference aids segmenting the pegs from the background by estimating a global threshold value using the Otsu's thresholding method. The segmentation process replaces all pixels in the input image with intensity greater than threshold value by 1 (white) and replaces all other pixels with 0 (black). Pegs are having higher intensity than the component base of the wheel-hub and hence they are shown as 'white' in Fig. 3. Further, the noise present in the binary image after the segmentation process is removed by area based filtering. An 8-connectivity neighborhood is used to create the connected components in the binary image. The area properties of the connected components are calculated. The areas of interest are the pegs and the wheel-hub center hole and the screw portion area. The area of the noise components present in the image are comparatively lesser than the area of screw. Hence the connected components with the area property less than that of the screw area are removed on this filtering. In case of noise components with higher area, they are removed in the circle detection stage as they are non-circular objects.



**Fig. 3.** Tracked peg and wheel-hub center

The circles in the segmented image are identified by a two stage Circular Hough Transform (CHT) method [11]. In this CHT, a circle is drawn for each edge point with a radius 'r'. This method adopts an (3D) accumulator array computation technique in which the first two dimensions represent the coordinates of the circle and the last specify the radii. The values in the accumulator (array) are increased every time whenever a circle is drawn with the desired radii over every edge point. The accumulator, which keeps count of the circles passing through coordinates of each edge point, and makes a vote to find the highest count. The coordinates of the center of the circles in the images are the coordinates with the highest count. The details pertaining on this method related to matlab function is cited in reference [11]. There are four pegs in the wheel-hub and two screws present between the pegs labelled 1&4 and pegs 2&3. Taking the first screw as reference, the first peg could be labeled as 1 and other pegs are

labelled sequentially in anticlockwise direction. Figure 3, shows the pegs detected in a frame using the multiple-peg tracking module.

**Genetic Algorithm Based Camera Calibration.** The estimated coordinates in pixel values are to be converted to metric coordinates to calculate the pose of the multiple-peg object in Cartesian space. The precise calculation of the 3D coordinates and pose of the object depends on the accuracy of the calibration procedure. In this regard, a genetic algorithm based two-stage camera calibration procedure is adapted in this work. In the first stage, a linear solution is computed by considering distortion-free camera model and in the second stage, genetic algorithm is used to compute the optimal camera parameters including distortion by using the closed form solution as an initial guess [12].

A pin-hole model with lens distortion as shown in Fig. 4, is adopted in this work. Considering  $(X_w, Y_w, Z_w)$  as the 3D world coordinate system, and  $(X_c, Y_c, Z_c)$  as camera coordinate system,  $(X_i, Y_i)$  represents the image coordinate system. Further,  $O_i$  and  $O_c$  represent the center of the image coordinate system and the optical center of the camera coordinate system, respectively.  $O_i$  and  $O_c$  are collinear and aligned with the  $Z_c$  axis of the camera coordinate system.  $(x_i, y_i)$  be the image coordinates measured through any point extraction algorithm. Let  $P$  be the test point  $(x_w, y_w, z_w)$  as the world coordinate and  $(x_u, y_u)$  be the estimated undistorted image coordinate  $(x_c, y_c, z_c)$  point. Focal length i.e. the distance between the image plane and the camera plane is denoted as  $f$ .

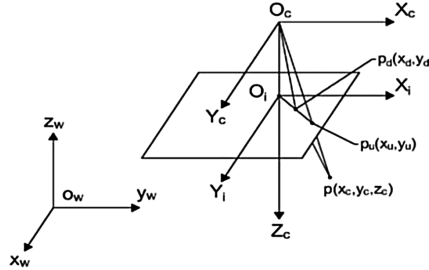


Fig. 4. Complete camera model with radial distortion

On considering the radial distortion, the image coordinate  $(x_u, y_u)$  of the distortion-free model is subtracted with the distortion factor  $D_x$  to estimate the distorted image coordinate  $(x_d, y_d)$ . The distortion is modelled as a second order polynomial [15].

$$\begin{aligned} x_d + D_x &= x_u \\ y_d + D_y &= y_u \end{aligned} \tag{3}$$

$$D_x = x_d(k_1 r^2)$$

$$D_y = y_d(k_1 r^2)$$

$$r = \sqrt{x_d^2 + y_d^2}$$

$k_1$  is the distortion coefficient.

The distorted image coordinates  $(x_d, y_d)$  are transformed to computer image coordinates  $(x_f, y_f)$  by multiplying with the uncertainty scaling factor and by adding the center of the frame. In the first stage, a linear approximation method as described by Tsai [7, 15] is used to determine the camera parameters by considering a distortion-free camera model which forms the bounds for the second nonlinear genetic algorithm stage.

Genetic Algorithm is adopted to identify the optimal intrinsic and extrinsic parameters of the camera incorporating the radial distortion in the camera model. Linear approximation stage estimates the rotation matrix, translation vector in  $x$  and  $y$  direction. Therefore, the proposed GA has focal length and translation about  $z$ -axis as the genes in the chromosome which enables the proposed method to have faster convergence to optimal results.  $P_1$  represents the initial population of the 1st generation,  $f$  and  $T_z$  represent the genes (parameters) present in a chromosome,  $j$  represents the chromosomes in the population of a generation,  $k$  represents the generations and  $n$  &  $m$  are the number of chromosomes in the population and number of generations, respectively. The initial population for the genetic algorithm is

$$\begin{aligned} (f^j)_k &= f_{linear}; \forall 1 \leq j \leq n \text{ and } k = 1 \\ (T_z^j)_k &= T_{zlinear}; \forall 1 \leq j \leq n \text{ and } k = 1 \\ (q^j)_k &= \left\{ (f^j)_k, (T_z^j)_k \right\} \forall 1 \leq j \leq n \text{ and } k = 1 \\ P_1 &= (q_i^j)_1 \end{aligned} \quad (4)$$

where  $f_{linear}$ ,  $T_{linear}$  are the focal length and translation about  $z$ -axis estimated in first stage. The bounds on the parameters  $f$  and  $T_z$  are taken as  $\pm 25\%$  of the linearly estimated values (stage 1 results).

$$\begin{aligned} (q_{1\ bound}^j)_k &= f_{linear} \pm (0.25 * f_{linear}) \\ (q_{2\ bound}^j)_k &= T_{zlinear} \pm (0.25 * T_{zlinear}) \\ (q_{\max}^j)_k &= \{f_{\max}, T_{z\max}\} \forall 1 \leq j \leq n \text{ and } 1 \leq k \leq m \\ (q_{\min}^j)_k &= \{f_{\min}, T_{z\min}\} \forall 1 \leq j \leq n \text{ and } 1 \leq k \leq m \end{aligned} \quad (5)$$

As lens distortion is considered in this camera model, the distortion coefficient  $k_1$  for each chromosome is calculated as:

$$\begin{aligned} (k_1^j)_k &= \frac{C_3(f^j)_k}{C_1(C_2 + (T_z^j)_k)} - \frac{C_4}{C_1} \forall 1 \leq j \leq n \text{ and } 1 \leq k \leq m \\ c_1 &= d_y y_i r^2 \end{aligned} \quad (6)$$

$$c_2 = r_7x_w + r_8y_w$$

$$c_3 = r_4x_w + r_5y_w + T_y$$

$$c_4 = d'_y y_i \text{ and } r = \sqrt{x_d^2 + y_d^2}$$

The variation between the actual image coordinates and the calculated coordinates using the estimated camera parameters is termed as re-projection error [8] which is the common performance measure of any camera calibration technique.

$$E_{rms} = \frac{1}{n} \sum_{l=1}^n \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2} \quad (7)$$

Hence the objective of this proposed genetic algorithm based calibration technique is to determine the optimal values of  $f$  and  $T_z$  for minimum re-projection error ( $E_{rms}$ ).

$$(q^j)_k^* = \min(E_{rms}) \quad (8)$$

$$\text{Subject to } f_{\min} \leq f^* \leq f_{\max} \text{ and } T_{z\min} \leq T_z^* \leq T_{z\max}$$

**GA Operators.** The convergence of a genetic algorithm depends on the selection of the mutation and cross over operators employed in the algorithm. The genes are encoded as real numbers to have understanding in the computation. The best chromosomes among the population are selected for reproduction using the proportionate reproduction method. Crossover operation creates new children in the population. The off-springs of the crossover operation have better fitness than the parent chromosomes. Even though if bad off-springs are created in the crossover phase, they will be eliminated by the subsequent reproduction phase in the next generation and thus off springs with better fitness than their parents are retained in the subsequent generation. A blend over crossover operation is adopted in this work, since they prevent the algorithm from getting trapped in the local optimal solution.

Two parents from the initial population  $(q_i^j)_k$  and  $(q_i^{j+1})_k$  are selected and the off-springs  $(o1_i^j)_k$  and  $(o2_i^j)_k$  are generated by

$$\begin{aligned} (o1_i^j)_k &= \left( \min \left( (q_i^j)_k, (q_i^{j+1})_k \right) \right) - \alpha \left( (q_i^j)_k - (q_i^{j+1})_k \right) \\ (o2_i^j)_k &= \left( \max \left( (q_i^j)_k, (q_i^{j+1})_k \right) \right) + \alpha \left( (q_i^j)_k - (q_i^{j+1})_k \right) \\ (q^j)_{k+1} &= \left( (o2_i^j)_k - (o1_i^j)_k \right) * rand + (o1_i^j)_k \end{aligned} \quad (9)$$

where  $i = 1, 2, j = 1, 2, \dots, n$  and  $k = 1, 2, \dots, m$  and the value of  $\alpha$  is taken as 0.75.



Mutation operation increases the diversity in the population which improves the convergence to global optimal solution. A power mutation operator is adopted in this work using the following expressions.

$$(q_i^j)_{k+1} = \begin{cases} (q_i^j)_k - s \left( (q_i^j)_k - (q_{i \min}^j)_k \right) & \text{if } u < r \\ (q_i^j)_k - s \left( (q_{i \max}^j)_k - (q_i^j)_k \right) & \text{if } u \geq r \end{cases} \quad (10)$$

Where  $u = \left( (q_i^j)_k - (q_{i \min}^j)_k \right) / \left( (q_{i \max}^j)_k - (q_{i \min}^j)_k \right)$  and  $r$  is a uniformly distributed random value between 0 and 1.

$$s = p^* s_r^{p-1}, \quad 0 \leq s_r \leq 1$$

Where  $p$  is the index of distribution and  $s_r$  is a uniform distributed random number between 0 and 1.

### 3.2 Orientation Estimation

After obtaining the image coordinates of the centers of pegs from tracking stage and the camera parameters from calibration stage, the metric coordinates of each pegs are determined. The wheel-hub is allowed to rotate about z-axis and/or to translate along x-axis only by constraining the translation along z-axis and rotation about X&Y axes. The linear displacement along x-axis (position of the peg) is estimated using the metric information of the peg centers. The orientation of the wheel-hub about z-axis is estimated by calculating the angle between the line1 connecting the peg1 center and wheel-hub center, and the horizontal line (line2) passing through wheel hub center as shown in Fig. 5 and (11).

$$\tan \theta = \left[ \frac{m_1 - m_2}{1 + (m_1 \circ m_2)} \right] \quad (11)$$

$m_1, m_2$  are the slopes of the line1 and line2 respectively.

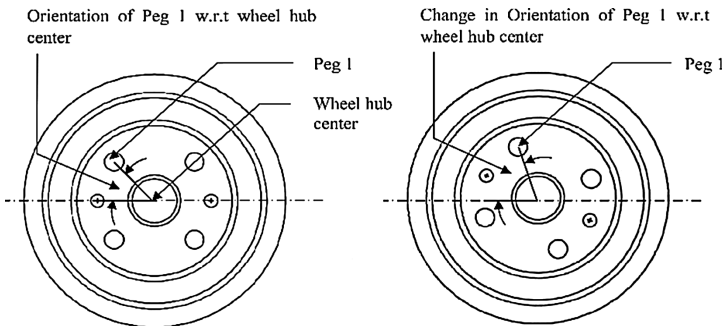


Fig. 5. Orientation estimation from the peg1 center and wheel-hub center

## 4 Experimental Arrangement

This section explains the experiments performed with the wheel-hub to evaluate the performance of the proposed pose estimation algorithm in terms of its accuracy. The experimental environment has the target object wheel-hub placed at a distance of 0.27 m from a fixed camera. DNV 3001 CCD camera with f50 lens is used to capture the wheel-hub images in this experiment at 10 fps. As discussed, the accuracy of the pose estimation algorithm is influenced by the accuracy of the camera calibration procedure and the feature tracking procedure. To ensure the performance of the proposed camera calibration procedure and to estimate the camera parameters, the algorithm is tested with the checker board images placed at the same location of the wheel-hub. The chess board pattern of  $6 \times 8$  corner points with an interval of 29 mm is captured from five various orientations at an image resolution of  $1600 \times 1200$  with the DNV camera. The camera parameters estimated by the proposed work is listed in Table 1. The image coordinates of the peg centers and the metric coordinates after the camera calibration are listed in Table 2. The accuracy of the pose estimated could be ensured only when the pose is fed to the manipulator to align the end-effector with respect to the object. In order to ensure the performance of the proposed pose estimation algorithm, the accuracy of tracking module is tested by comparing the metric values of radii of pegs at each instant with their corresponding true values.

**Table 1.** Camera parameters estimated by proposed work

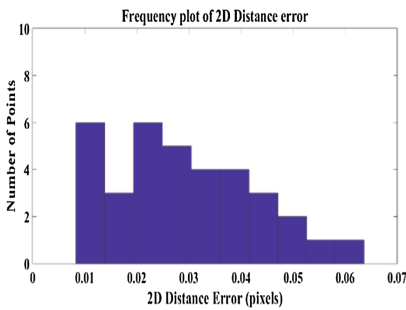
Stages	Focal length $f(\text{mm})$	Translation $T_z$	Rotation matrix $[R]$	Distortion coefficient $k_1$ ( $\text{pixel}^{-2}$ )
Linear (stage1)	54.74	$\begin{bmatrix} -945.2 \\ 663.51 \\ 308.52 \end{bmatrix}$	$\begin{bmatrix} 0.999 & 0.004 & 0.034 \\ -0.011 & 0.979 & 0.203 \\ -0.032 & -0.203 & 0.978 \end{bmatrix}$	0
Genetic algorithm (stage2)	47.95	$\begin{bmatrix} -945.2 \\ 663.51 \\ 267.38 \end{bmatrix}$	$\begin{bmatrix} 0.999 & 0.004 & 0.034 \\ -0.011 & 0.979 & 0.203 \\ -0.032 & -0.203 & 0.978 \end{bmatrix}$	1.097 E-5

**Table 2.** Image and metric coordinates of pegs at frame 1

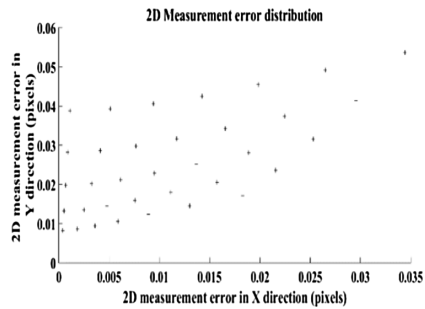
Peg no.	Image coordinates (Pixels)	Metric coordinates (mm)
1	(140.064, 127.673)	(28.993, 26.428)
2	(470.513, 93.238)	(97.396, 19.300)
3	(506.212, 421.383)	(104.785, 87.226)
4	(175.285, 456.696)	(36.284, 94.536)

## 5 Results and Discussion

Each peg’s center in pixels is estimated using the multiple-peg tracking and the camera parameters from the genetic algorithm based camera calibration technique. The metric coordinates of the centers are obtained from the estimated camera parameters. The results of our proposed calibration algorithm are compared with Zhang’s (2001) camera calibration algorithm based matlab camera calibration toolbox. The obtained root mean square re-projection error of toolbox is 0.37 pixels whereas that of the proposed algorithm is 0.0310 pixels. The results also show that our proposed algorithm has better accuracy in estimating the camera parameters than the matlab calibration toolbox which in turn ensures minimum pose estimation error. It is evident from Fig. 6, that 57% of re-projected points has less than 0.03 pixel distance error which proves the measurement accuracy of the proposed calibration technique. It is observed from Fig. 7, that the proposed method is able to estimate a point and re-project a point within an error range of 0.5 pixels. Table 3 represents the mean and standard deviation of the error between the true value and the estimated metric value of the radius of peg1 for specific interval of frames. The tracking and calibration stages of the proposed pose estimation technique are capable to estimate the pose within an error range of less than 1 mm.



**Fig. 6.** Frequency plot of 2D measurement error for checkerboard calibration.



**Fig. 7.** 2D measurement error distribution for checkerboard calibration.

**Table 3.** Mean and standard deviation of the errors between true and estimated radius of peg1

Frame range	True metric value of peg-1 (mm)	Mean error	Standard deviation
1–10	10.1	0.677	0.689
11–20	10.1	0.206	0.333
21–30	10.1	-0.171	0.259
31–45	10.1	-0.431	0.463

## 6 Conclusion

Vision sensors offer flexibility in perceiving the multiple hole component in a dynamic environment. Pose of the mating components in an assembly are determined using the feedback of vision sensor. The successful attainment of assembly action is influenced by the accuracy of estimated pose of the hole component and alignment between each peg and the corresponding hole in the wheel-hub. In this regard, a pose estimation algorithm has been proposed to determine the position and orientation of the multiple-pegs in the wheel-hub. Each peg in the wheel-hub is tracked for its center using the circle detection algorithm. With the use of pixel coordinates of tracked centers the position of the pegs with respect to the camera in metric values are determined by a two-stage genetic algorithm based camera calibration technique. The change in orientation is also obtained by calculating the angle of line connecting the centers of peg and wheel-hub, with the horizontal axis. The proposed pose estimation algorithm is experimented to validate its performance in terms of accuracy. Experimental results show that the calibration technique used in the proposed pose estimation algorithm has capability to re-project the peg with an accuracy of 0.0310 pixels. Besides, the proposed algorithm is suitable for a vision assisted robot assembly system with the positioning accuracy of 1 mm.

## References

1. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). doi:[10.1007/11744023\\_32](https://doi.org/10.1007/11744023_32)
2. Chaumette, F., Hutchinson, S.: Visual servo control part I: basic approaches. *IEEE Robot. Autom. Mag.* **13**, 82–90 (2006)
3. Chaumette, F.: Potential problems of stability and convergence in image based and position based visual servoing. In: Kriegman, D.J., Hager, G.D., Morse, A.S. (eds.) *The Confluence of Vision and Control*. LNCIS, vol. 237, pp. 66–78. Springer, London (1998). doi:[10.1007/BFb0109663](https://doi.org/10.1007/BFb0109663)
4. Dong, G., Zhu, Z.: Position based visual servo control of autonomous robotic manipulators. *Acta Astronaut.* **115**, 291–302 (2015)
5. Ji, Q., Zhang, Y.: Camera calibration with genetic algorithms. *IEEE Trans. Syst. Man. Cybern. Part A: Syst. Hum.* **31**, 120–130 (2001)
6. Lee, S., Kim, E., Park, Y.: 3D object recognition using multiple features for robotic manipulation. In: *Proceedings of IEEE International Conference on Robotics and Automation, USA*, pp. 3768–3774 (2006)
7. Lenz, R., Tsai, R.: Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 68–75 (1987)
8. Li, W., Gee, T., Friedrich, H., Delmas, P.: A practical comparison between Zhang's and Tsai's calibration approaches. In: *Proceedings of International Conference on Image and Vision Computing (IVCNZ), New Zealand*, pp. 166–171, November 2014
9. Mallis, E.: Survey of vision based robot control. In: *Proceedings of European Naval Ship Design, Captain Computer IV Forum, Brest, France*, pp. 1–16 (2002)

10. Marchand, E., Bouthery, P., Chaumette, F., Moreau, V.: Robust real time visual tracking using a 2D-3D model based approach. In: Proceedings of the 1999 IEEE International Conference on Computer Vision, pp. 262–268 (1999)
11. Mathworks Circle Detection Algorithm, June 2016. <http://in.mathworks.com/help/images/examples/detect-and-measure-circular-objects-in-an-image.html> accessed
12. Nagarajan, P., Saravana Perumaal, S.: GA based camera calibration for vision assisted robotic assembly system. *IET Comput. Vision* (2016). doi:[10.1049/iet-cvi.2016.0004](https://doi.org/10.1049/iet-cvi.2016.0004)
13. Song, K.T., Chang, C.H.: Object pose estimation for grasping based on robust center point detection. In: proceedings of 8th Asian Control Conference, Taiwan , pp. 305–310 (2011)
14. Song, X., Yang, B., Feng, Z., et al.: Camera calibration based on particle swarm optimization. In: Proceedings of IEEE International Conference on Image and Signal Processing, Tianjin, pp. 1–5 (2009)
15. Tsai, R.: A versatile camera calibration technique for high accuracy 3D machine vision metrology using off the shelf TV cameras and Lenses. *IEEE J. Robot. Autom.* **3**(4), 323–344 (1987)
16. Weng, J., Cohen, P., Herniou, M.: Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 965–980 (1992)
17. Yang, Z., Chen, F., Zhao, J., et al.: A novel camera calibration method based on genetic algorithm. *IEEE Conference on Industrial Electronics and Applications*, Singapore, pp. 2222–2227, June 2008
18. Youngrock, Y., DeSouza, G.N., Avinash Kak, C.: Real-time tracking and pose estimation for Industrial objects using geometric features. In: Proceedings of IEEE International Conference on Robotics and Automation, Taiwan, pp. 3473–3478 (2003)
19. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1330–1334 (2000)