MedCrave
Step into the World of Research
International Robotics & Automation Journal

Research Article

Open Access
CrossMark

A comparative study of pose estimation algorithms for visual navigation in autonomous robots

Abstract

Autonomous navigation is a research field that gives mobile robots the capacity to perform various tasks without human assistance. Autonomous navigation based on visual sensors can be used in GPS denied environments.

Vision-based navigation performs feature detection, matching, and pose estimation using camera images. This paper presents a new approach to autonomous navigation for mobile robots using a Color-based Image Segmentation and Centroid Detection algorithm instead of traditional feature detection algorithms. The algorithm intentionally matches features across images using known feature points and uses conventionaltechniques such as Epipolar Geometry and Perspective-N-Points algorithms for camera pose estimation.

The study includes camera calibration to estimate intrinsic parameters and their physical unit conversion ensuring accurate measurements. Experimental datasets are used to analyze the performance of the Epipolar geometry, Perspective-3-Point, and Efficient-Perspective-n-Point based pose estimation algorithms. To enhance accuracy, the P3P algorithm is modified to consider combinations of image points for pose estimation. The paperconcludes with a comparative analysis between the original P3P algorithm and the modified version, providing valuable insights into their respective performances. Overall, the paper aims to provide a comparison of differentpose estimation algorithms used for visual navigation.

Keywords: pose estimation, visual navigation, camera calibration, perspective-n-point, efficient pnp, epipolar geometry

Volume 9 Issue 3 - 2023

Sharu Susan Jacob,[1] Sreeja S[2]
[1]Dept. of Electrical Engineering, College of Engineering Trivandrum, India
[2]Dept. of Electrical Engineering, College of Engineering Trivandrum, India

Correspondence: Sreeja S, Dept. of Electrical Engineering, College of Engineering Trivandrum, Kerala, India, Email sreeja@cet.ac.in

Received: December 19, 2023 | **Published:** December 29, 2023

Abbreviations: PnP, perspective-n-point; P3P, perspective-3-point; EPnP, efficient perspective-n-point; SLAM, simultaneous localization and mapping; VO, visual odometry; GPS, global positioning system; SURF, speeded up robust features; SIFT, scale-invariant feature transform; EKF, extended kalman filter; RGB, red-green-blue; HSV, hue- saturation-value; CNN, convolutional neural network

Introduction

Autonomous robots are becoming increasingly popular in various fields, such as manufacturing,[1] agriculture,[2] and healthcare,[3] due to their ability to perform tasks without human intervention. One of the key challenges in developing autonomous robots is enabling them to navigate through complex and dynamic environments. Simultaneous Localization and Mapping (SLAM)[4] is one such technique employed in autonomous robots which can be used in GPS denied environments.[5] describes the SLAM problem and the essential methods for solving the SLAM problem and summarizes key implementations and demonstrations of the method while[6] discusses about the recent works in addressing some of the remaining issues in SLAM, including computation, feature representation, and data association.[4] reviews the basic paradigms of SLAM which are EKF SLAM, Particle-filter based SLAM, and graph optimization along with V-SLAM and RGB SLAM.

Vision-based navigation is a very promising approach to the problem of a robot navigating through complex environments. It involves cameras and other vision sensors to perceive the environment and estimate the robot's position and orientation relative to a map. SLAM using cameras is referred to as visual SLAM and it depends solely on visual information.[7] includes a review on the different Visual SLAM algorithms developed between 2010 and 2016. There are different approaches in visual SLAM which includes feature-based, direct, and RGB-D camera-based approaches. These approaches can be based on monocular or stereo vision.

In[8] and,[9] the authors have presented a study on Visual odometry (VO), including the stages involved, as well as its benefits and uses. It explains the application of VO as a building block for Visual SLAM. The advantages of using visual sensors in SLAM algorithms and a review on different Visual SLAM systems are given in.[10]

The basic step in feature based visual SLAM or Visual odometry is feature detection. A number of algorithms are available for feature detection. The pros and cons of commonly used feature detection algorithms like SURF,SIFT are discussed in.[11] S. A. K. Tareen[12] shows the detailed comparison between the algorithms based on available data sets. The detected features can be used to estimate the position and orientation of the camera. This process can be accomplished by matching identical features in various images and using pose estimation algorithms to estimate the relative rotation and translation.

Pose estimation is a critical aspect of autonomous navigation for mobile robots. It enables robots to locate themselves in their surroundings and move efficiently without human intervention. These methods involve detecting features or unique points in real-time images taken by the visual sensors and using them to find the motion of the camera relative to the environment.

Epipolar Geometry[13] and Perspective-n-Point (PnP)[14] are the conventional methods used for camera pose estimation, while the advanced methods for camera pose estimation includes CNN and deep learning.[15] Discusses about camera pose estimation using deep learning while[16] discusses a convolutional neural network (CNN) based method. Due to their higher reliability in comparison to advanced methods, conventional approaches are more frequently employed.

This paper uniquely contributes to the field by:

a. This paper uniquely contributes to the field by providing a comprehensive analysis of three pose estimation algorithms along with a modified version of the P3P algorithm. To the best of the authors knowledge, these algorithms have not been compared together with experimental data in previous studies.

b. Development of a Color-based Segmentation and Centroid detection method for known feature point detection.

c. Conversion of camera parameters into real world units.

Thus, this work aims to evaluate the performance of pose estimation algorithms, Epipolar Geometry, P3P, and EPnP. The camera calibration process and conversion of camera parameters into real-world units will be performed to improve the accuracy of pose estimation. The development of P3P and EPnP algorithms will be undertaken to enhance the accuracy and speed of pose estimation. Special consideration is given to minimizing the size of the functions, taking into account the real-time implementation requirements of these algorithms. Furthermore, the proposed method will use color-based image segmentation and centroid detection to identify known feature points in the experimental data.

The evaluation of the proposed method will be performed based on experimental data, and the performance of the three algorithms will be compared. The results obtained from this work will help in selecting the appropriate algorithm for pose estimation based on the specific requirements of the navigation task.

Feature detection based on color based segmentation and centroid detection

Features or key points are localized regions such as edges and corners within an image which provides interesting information about its content thus differentiating that part from the others. They are repeatedly detected in images captured from various viewpoints of an object, to provide a relation between these images. The features in an image are detected and matched across different images to estimate the position and orientation of the robot in autonomous robots. To do this, commonly employed algorithms such as SURF and SIFT are utilized where the feature points are randomly selected and matched. These feature points are not known prior to the process.[17] Presents a comparison of these algorithms based on different scenarios, outlining the results obtained in each case. This paper employs a distinctive approach that involves utilizing known feature points for feature detection. The need for using known feature points is to remove the difficulties related with physically determining the positions of unknown feature points. The use of known feature points facilitates a straightforward physical analysis of their relative positions, making them easily comparable. The features to be detected are given in different colors. The algorithm is expected to identify and segment the images based on their respective colors and determine the corresponding centroids. The steps involved in the algorithm are shown in Figure 1.

RGB (Red, Green, Blue) and HSV (Hue, Saturation and Value) are color spaces. As most digital color pickers are based on HSV scale which facilitates more accurate color separation than RGB, the conversion from RGB scale to HSV is employed. Hue refers to basic colors of the rainbow, Saturation refers to the intensity of the color and Value refers to the lightness or darkness of a particular color. RGB values will be within the range (0-255) while HSV will be between (0-1). To convert RGB values to HSV, the steps given in[18] are followed.

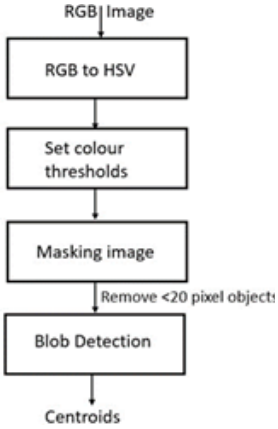

Figure 1 Color based image segmentation and centroid detection algorithm.

The color thresholds are set based on the converted values and subsequently the image is masked. Masking is a filtering process wherein the filter mask traverses an image and at each point the filter response is calculated using a predefined relationship. Based on the set thresholds, the regions in the image falling outside a specified range is assigned a value of zero. A binary image is thus obtained, and it undergoes further refinement using area opening morphology. Area opening morphology is employed as a filter to eliminate the components with an area smaller than a designated threshold. This step is crucial for preventing inaccuracies in centroid detection by excluding objects of negligible size. Subsequently, blob detection is carried out through connected components labelling, wherein pixels belonging to the same connected component or object are grouped together. This process allows for the identification of blobs and their associated properties. Based on the properties of the identified blobs, the centroid is determined. The centroid thus detected in both the reference image and the i^{th} image is matched. Centroids are detected individually for circles of each color and are then matched together. This methodology ensures precise determination of centroids thus increasing the accuracy of further procedures.

Camera calibration

Camera calibration should be performed prior to the camera pose estimation. Camera calibration is performed to determine the relationship between the image's 3D real points and corresponding 2D projections.[19] Checker board pattern images are commonly used as the input for camera calibration. A minimum of two images will be needed. The dimensions of the checkerboard patterns and the image are known.

The estimation of the intrinsic parameters and the distortion parameters is called camera calibration and these are estimated using the closed form solution described in.[20] Consider a 2D point $\mathrm{m}=[\mathrm{u},\mathrm{v}]^{\mathrm{T}}$ and a 3D point $\mathrm{M}=[\mathrm{X},\mathrm{Y},\mathrm{Z}]^{\mathrm{T}}$. Let the augmented vectors be represented as $\tilde{\mathrm{m}}=[\mathrm{u},\mathrm{v},1]^{\mathrm{T}}$ and $\tilde{\mathrm{M}}=[X,Y,Z,1]^{T}$. For a pinhole camera, the relationship between a 3D point M and its image projection m can be given as,

$$s\tilde{m}=K[\mathrm{R}\ \mathrm{T}]\tilde{M}=\begin{bmatrix} f_x & \gamma & u_o \\ 0 & f_y & v_o \\ 0 & 0 & 1 \end{bmatrix}[\mathrm{R}\ \mathrm{T}]\tilde{\mathrm{M}}=H\tilde{M} \quad (1)$$

The homography H can be denoted as $\mathrm{H}=[\mathrm{h}_1\ \mathrm{h}_2\ \mathrm{h}_3]$. Therefore,

$$[h_1\ \ h_2\ \ h_3\ ]\ =\ \lambda K[r_1\ \ r_2\ \ \mathrm{T}] \quad (2)$$

Citation: Sharu SJ, Sreeja S. A comparative study of pose estimation algorithms for visual navigation in autonomous robots. *Int Rob Auto J.* 2023;9(3):124–130. DOI: 10.15406/iratj.2023.09.00274

- s - Arbitrary scale factor.
- R and T - Extrinsic parameters representing rotation and translation.
- K - Camera intrinsic matrix
- λ - Arbitrary scalar

Based on the knowledge that r_1 r_2 are orthonormal, the two constraints on the intrinsic parameters are,

$$h_1^T K^{-T} K^{-1} h_2 = 0$$
$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h \quad (3)$$

As H has eight degrees of freedom and there are six extrinsic parameters (three for rotation and three for translation), there will be two constraints on the intrinsic parameters. Here $K^{-T}K^{-1}$ represents the image of the absolute conic. Let B be a symmetric matrix defined by a 6D vector b = [B11, B12, B22, B13, B23, B33].

$$B = K^{-T}K^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{f_x^2} & -\frac{\gamma}{f_x^2 f_y} & \frac{v_o\gamma - u_o f_y}{f_x^2 f_y} \\ -\frac{\gamma}{f_x^2 f_y} & \frac{\gamma}{f_x^2 f_y^2} & -\frac{\gamma(20\gamma - u_o f_y)}{f_x^2 f_y^2} - \frac{v_o}{f_y^2} \\ \frac{v_o\gamma - u_o f_y}{f_x^2 f_y} & -\frac{\gamma(v_o\gamma - u_o f_y)}{f_x^2 f_y} - \frac{v_o}{f_y^2} & \frac{(v_o\gamma - u_o f_y)}{f_x^2 f_y} + \frac{v_o^2}{f_y^2} + 1 \end{bmatrix} \quad (4)$$

The camera intrinsic parameters can be calculated as,

$$v_o = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2)$$
$$\lambda = B_{33} - [B_{13}^2 + v_o(B_{12}B_{13} - B_{11}B_{23})] / B_{11}$$
$$f_x = \sqrt{\lambda B_{11}}$$
$$f_y = \sqrt{\lambda B_{11}} / (B_{11}B_{22} - B_{12}^2)$$
$$\gamma = -B_{12} f_x^2 f_y / \lambda$$
$$u_0 = \gamma v_o / f_x - B_{13} f_x^2 / \lambda$$

K matrix can be written from the above equation. It can be further refined using maximum likelihood estimation. The camera intrinsic parameters derived using the process of camera calibration are derived in pixel units, which means they represent measurements in terms of the camera's imaging sensor. In order to use these intrinsic parameters for pose estimation or other applications, they need to be converted into world units. This conversion is necessary to relate the measurements to real-world values that can be used for further analysis or comparison. If the intrinsic parameters are expressed in pixel units, the physical size of an object in the image cannot be determined unless the pixel size is known. By converting the intrinsic parameters to world units, such as millimeters or inches, it becomes possible to relate measurements in the image to physical measurements in thereal world. Therefore, to use camera calibration parameters for pose estimation, it is necessary to convert them from pixel units to world units. The focal length expressed in pixels are converted to world units and then are compared to the true value in this paper to ensure that the camera is correctly calibrated.

Pose estimation

Pose estimation is a process of tracking the location of features for the given objects in a series of images. Conventional methods for pose estimation include Epipolar Geometry and Perspective-NPoint method. Before carrying out an extensive comparison of various pose estimation algorithms, a camera pose estimation approach based on epipolar geometry, utilizing images captured by a mobile camera was conducted and are presented in[17] The results of this method were then contrasted with the true values.

Epipolar Geometry

Epipolar geometry is employed for 2D image point-based camera motion estimation. Using epipolar geometry, itis possible to estimate the camera motion from monocular images with available 2D point sets. Basic ideas ofthis method were already outlined in[17] with reference to.[20] A 3x3 matrix (R) representing relative orientation and a 1x3 matrix (T) representing the camera location together gives the camera pose. Epipolar geometry enables the estimation of camera motion between two frames by utilizing matched points in two images.

PnP

Perspective-N-Point (PnP) method can also be used for Pose estimation. It requires the world points and its corresponding image points along with the camera parameters to be known to find out the relative rotation and translation.

Triangulation is the process of reconstructing a 3D point using its projections onto two or more images. This process is required when using monocular camera images as inputs for pose estimation. The 3D points in case of monocular vision will not be available, thus triangulation can be performed to determine them. The methods used for triangulation are mid-point method, and using essential matrix.

Essential matrix can be used along with image projections on two or more images to find the 3D points. Let (x1, x2, x3) be the 3D point coordinates and (y1, y2), (y'1, y'2) be the corresponding image coordinates. The coordinate x3 can be written as,[21]

$$x_3 = \frac{(r_1 - y'_1 r_3).T}{(r_1 - y'_1 r_3).y} \quad (6)$$

and the other two coordinates can thus be determined using,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_3 \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (7)$$

where, r_i corresponds to the i^{th} row of the rotation matrix, while T represents the translation between the images.

P3P

Perspective-3-Point algorithm is the basic problem of PnP where three world points and three image points are only required to find the camera pose. It yields up to four real, geometrically feasible solutions. The parameters are calculated with respect to the three match pairs selected using the following equations. The idea behind the P3P algorithm is taken from.[21]

In,[21] the first three points are randomly chosen as the image points required for finding the pose using the P3Palgorithm. But this paper uses a different approach. Different combinations of image points are chosen, and each of their pose is estimated. The combination of image points resulting in less error is then used in P3P algorithm to estimate pose.[22] discusses an approach for estimating the pose of

an object using the Perspective-Three- Point (P3P) algorithm. As mentioned, the P3P algorithm requires the selection of three image points from the object to determine its pose. In this paper, the authors propose a novel method that differs from the method used in,[22] of using only the first three image points for the P3P algorithm. This traditional approach can lead to inaccuracies in pose estimation when the first set of image points does not adequately represent the object's features required for feature matching.

In this approach, the authors consider all possible combinations of three image points from the object and estimate the pose of the object using each combination. This process results in several estimates of the object's pose, each corresponding to a different combination of image points.

The authors then analyze the estimation error associated with each pose estimate and choose the combination of image points that results in the lowest error. This combination is then used in the P3P algorithm to obtain the final estimate of the object's pose.

By considering multiple combinations of image points and choosing the one with the lowest error, the proposed approach is able to improve the accuracy of the pose estimation process. This can be particularly useful in situations where the object is not easily visible, or where the image points are noisy or uncertain. The approach also provides a degree of robustness, as it is less sensitive to errors or outliers in the image points.

The combination equation, or binomial coefficient formula, is a mathematical tool used to calculate the number of possible combinations of k elements that can be selected from a set of n elements as given in equation (8).[23]

$$^{n}C_{k} = \frac{n!}{k!(n-k)!} \tag{8}$$

where n represents the total number of elements in the set and k represents the number of elements to beselected.

In the context of the paper's proposed approach, the authors utilize the combination equation to generate all possible combinations of image points for estimating the pose of an object. This allows the algorithm to consider various sets of image points and select the one that results in the most accurate pose estimate. This approach can improve the accuracy of pose estimation, particularly when the first set of image points does not sufficiently represent the object's features required for feature matching. Additionally, using multiple combinations of imagepoints can make the algorithm more resilient to noise or errors in the image points.

EPnP

EPnP method is said to be more efficient than P3P, as P3P is the basic case and considers only three match pairs. EPnP solves the problem of PnP using n
four match pairs. In EPnP, the world points are expressed as a weighted sum of the control points. The algorithm employed in this paper for EPnP problem is solely based on the.[24]

Results

Calibration of mobile camera

Camera calibration is the first step in determining camera pose. As previously mentioned, the checkerboard images captured with the mobile camera served as inputs for the camera calibration process as shown in Figure 2.

Figure 2 Checkerboard images.

For the analysis of Epipolar geometry-based pose estimation algorithm, Nokia 6.1 plus mobile was used. Camera calibration is performed to compute the camera intrinsic parameters which includes focal length, principal points of the camera. The specifications of the camera used are,

- Camera sensor model- Samsung S5K3P9SX
- Physical sensor size- 4.7 x 3.5 mm
- Focal length- 4mmThe image details are,
- Image dimension- 864 x 1152 pixels
- Square size- 2.4cm (Squares in the checkerboard pattern image)
- Mean Re projection error- 0.13 pixels
- Intrinsic parameters

If the pixel pattern of camera is perfectly square, then $f = f_x = f_y$. But when the pixels are a bit rectangular (mostly in practical cases), f_x differs from f_y . Here, $f = (f_x, f_y)$,

$$K = \begin{bmatrix} 1006.0855 & 0 & 570.9394 \\ 0 & 1077.4764 & 433.5705 \\ 0 & 0 & 1 \end{bmatrix}$$

K is the intrinsic matrix formed using the intrinsic parameters computed using camera calibration. It will be constant for a camera. It can be used to determine the camera pose using previously discussed algorithms.

- Distortion parameters are determined by camera calibration using

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1 x_d y_d + p_2(r^2 + 2x_d^2) \\ 2p_2 x_d y_d + p_1(r^2 + 2y_d^2) \end{bmatrix} \tag{9}$$

where, k1, k2, k3 represents the radial distortion parameters. p1, p2 represents the tangential distortionparameters.

- k1=0.2478, k2=-0.3610, k3=0
- p1= p2=0

- As the axes x and y are exactly perpendicular to each other, the skew value of the intrinsic matrix is zero. Equation (10) converts the focal length from pixels to mm and compare it with the value in the manual provided for the camera sensor (ie, f = 4mm),

$$f_{mm} = \frac{(f\,\text{pixel})(\text{sensor}_{\text{width}}\,(\text{mm}))}{\text{image}_{\text{width}}\,(\text{pixels})} \tag{10}$$

$f_x = 4.0755\text{mm}, f_y = 4.3959\text{mm}$

Citation: Sharu SJ, Sreeja S. A comparative study of pose estimation algorithms for visual navigation in autonomous robots. *Int Rob Auto J.* 2023;9(3):124–130. DOI: 10.15406/iratj.2023.09.00274

Calibration of the turtlebot:

The images of the checker board pattern taken with the Raspberry pi camera module mounted on a turtlebot were used as inputs for camera calibration. The images are shown in Figure 3. The pose estimation algorithms Epipolar Geometry, P3P and EPnP are analysed using the images collected by the turtlebot. The details of the camera used are,

- Camera sensor - Raspberry pi cam module V2.1
- Physical sensor size - 3.68 x 2.76 mm
- Focal length - 3.04 mm

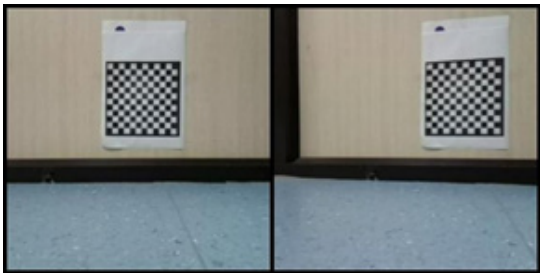

Figure 3 Checkerboard images for camera calibration of the turtlebot.

The image details are:

- Image dimension - 410 x 308 pixels
- Square size - 1.5cm (Squares in the checkerboard pattern image)
- Mean re-projection error - 0.08 pixels

$$K = \begin{bmatrix} 369.8489 & 0 & 241.5459 \\ 0 & 365.8828 & 117.7330 \\ 0 & 0 & 1 \end{bmatrix}$$

Radial distortion parameters:

k1 = 6.8343, k2 = -241.9467, k3 = 0

- Skew, $\gamma = 0$
- Tangential distortion=0, p1 = p2=0

The equation (10) converts the focal length from pixels to mm and compare it with the value in the manual provided for the camera sensor (ie, f = 4mm), Substituting values gives,

- $f_x = 3.3142$ mm, $f_y = 3.2840$ mm

Thus, the calculated focal length of the mobile camera and the turtlebot camera aligns closely with the provided manual value, underscoring the significance of conducting camera calibration in real-world units. This observation enhances the robustness and accuracy of the calibration process.

Camera pose estimation using experimental data

To generate experimental data, the Turtlebot3 Waffle Pi robot, depicted in Figure 4, was programmed to execute forward, backward, and lateral movements for a duration of 1 second each. During these maneuvers, the robot captured images of a predefined pattern. The Turtlebot3 Waffle Pi robot operates on Robot Operating System (ROS). ROS is an open-source middleware framework designed to facilitate the development of software for robots to help in building and controlling robots. It is widely used in research and industry for developing roboticapplications.

Figure 4 Turtlebot used for the experiment.

The sensors mounted on the robot are Raspberry Pi camera, LIDAR 360°LDS-01, 3 axis gyroscope, accelerometer, and magnetometer. The Raspberry pi camera module mounted on top of the robot is used to collect the images. The robot underwent controlled movements, including forward and backward motions at a velocity of 0.1m/s for 1-second intervals, as well as left and right spins at a velocity of 0.2m/s for 1-second durations. The experimental dataset, comprising 370 images, is illustrated in Figure 5, with the algorithm focusing on the analysis of the initial 305 images where the pattern remains fully visible.

Figure 5 Images from the dataset.

The pattern in the images consists of eight circles of varying colors. Employing the algorithm described in Section 2, centroids for each circle were detected. Figure 6 visualizes the 3D spatial relationship between the camera's position and the identified feature points in the images. These 3D point cloud was obtained by applying triangulation on the available 2D points from the images as described above.

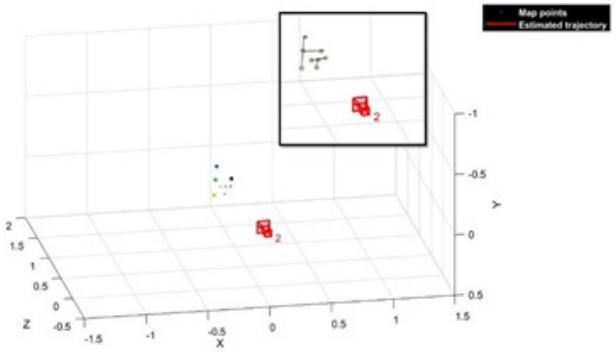

Figure 6 Point clouds and camera position in 3D space.

Citation: Sharu SJ, Sreeja S.A comparative study of pose estimation algorithms for visual navigation in autonomous robots. *Int Rob Auto J.* 2023;9(3):124–130. DOI: 10.15406/iratj.2023.09.00274

In each image, eight feature points were identified and subsequently compared with the reference image (Image 1) to get the relative rotation and translation between the reference image and the i^{th} image. Each of the aforementioned algorithms was employed to get the camera position and orientation. The detailed results are presented in the subsequent tables.

The experimental dataset utilized in this study was intentionally crafted with a focus on addressing docking applications where LED markers are used for pose estimation,m[25] although it is important to note that the algorithms employed are not restricted solely to this specific application.

Due to the movement of the turtlebot, there is no observed rotation about the x and y axes. Therefore, the roll and pitch angles across the sequence of images are ideally zero. However, the turtlebot exhibits rotation about the z-axis during leftward and rightward turns from the center, as well as when returning to the center. The corresponding variations in the yaw angle are detailed in Table 1. These values are compared with the actual orientation of the robot, measured using the sensors mounted on the robot.

Table 1 Estimated values for each algorithm

Algorithm	Rightmost Yaw angle (deg)	Leftmost Yaw angle (deg)	Deviation in Roll and Pitch angle (deg)
True Value	-19.65	21.77	0
Epipolar Geometry	-16.4282	27.7138	5
Original P3P	-23.9705	41.6381	80
Modified P3P	-23.5031	31.4639	40
EPnP	-25.1431	39.6134	20

It can be seen that the modified P3P algorithm is definitely more accurate than the original P3P algorithm, while EPnP yields reduced errors in terms of roll and pitch angle values. Table 2 provides the computational time for each algorithm. As shown in the table, the algorithm for colour segmentation and centroid detection requires relatively more time since it has to individually segment each colour and detect its centroid. EPnP emerges as the fastest among the considered algorithms.

Table 2 Computational time for each algorithm in seconds

Total Execution	41.4563
Color segmentation and centroid detection	29.4395
Epipolar Geometry	0.003602
Original P3P	0.05168
Modified P3P	0.087321
EPnP	0.000491

Thus, in terms of error and time complexity, EPnP algorithm is showing better performance than P3P. In comparison to P3P, EPnP exhibits a lesser deviation in error as shown in Figure 7.

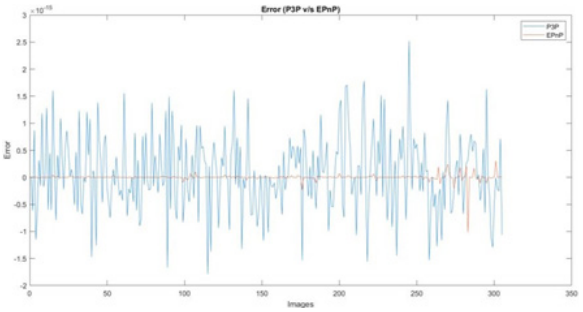

Figure 7 P3P versus EPnP.

Error calculations for both EPnP and P3P, in relation to Epipolar Geometry, were conducted. Deviations in the roll, pitch, and yaw angle values from the true values can be attributed to incorrect feature matching or errors in centroid detection. This can be improved by using a different pattern or a colour that is less likely to cause disturbances.

Conclusion

This paper undertakes a thorough exploration of three prominent pose estimation algorithms- P3P, EPnP, and Epipolar geometry with a specific focus on their performance assessment. The developed color-based segmentation and centroid detection algorithm for feature detection, along with the camera calibration process, contributes to accurate pose estimation. The comparison of the modified P3P algorithm and the EPnP algorithm reveals that the latter demonstrates superior accuracy and robustness in real-world experiments, particularly when utilizing color-based feature detection.

The significance of this work extends to its potential for Vision-based Simultaneous Localization and Mapping (SLAM) and machine learning application. The in-depth investigation presented in this paper not only provides avaluable reference for researchers and practitioners in the field of computer vision but also highlights crucial opportunities for refining pose estimation techniques. Future research directions could involve exploring additional feature detection enhancements to reduce drifts, replicating the methodology in OpenCV, expanding the algorithm to optimize performance under specific environmental conditions, and exploring real-time implementations to broaden its practical applicability in dynamic scenarios.

Acknowledgments

None.

Conflicts of interest

Authors declare no conflict of interest.

References

1. Arkin RC, Murphy RR. Autonomous navigation in a manufacturing environment. *IEEE Transactions on Robotics and Automation*. 1990;6(4).445–454.
2. Shalal N, Low T, McCarthy C, et.al. A review of autonomous navigation systems in agricultural environments. *SEAg 2013: Innovative agricultural technologies for a sustainable future*; 2013.
3. Koceski S, Koceska N. Evaluation of an assistive telepresence robot for elderly healthcare. *J Med Syst*. 2016;40(5):1–7.
4. Stachniss C, Leonard JJ, Thrun S. Simultaneous localization and mapping. Springer handbook of robotics; 2016. 1153–1176 p.
5. Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part i. *IEEE Robotics & Automation Magazine*. 2006;13(2):99–110.
6. Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*. 2006;13(3):108–117.
7. Taketomi T, Uchiyama H, Ikeda S. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*. 2017;9(1):1–11.
8. Scaramuzza D, Fraundorfer F. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*. 2011;18(4):80–92.
9. Fraundorfer F, Scaramuzza D. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics &Automation Magazine*. 2012;19(2):78– 90.

10. Fuentes-Pacheco J, Ruiz-Ascencio J, Rendon-Mancha JM. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*. 2015;43:55–81.

11. Bansal M, Kumar M, Kumar M. 2D object recognition: a comparative analysis of sift, surf and orb feature descriptors. *Multimedia Tools and Applications*. 2021;80:18839–18857.

12. Tareen SAK, Saleem Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. *International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*; 2018. 1–10 p.

13. Zhang Z. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*. 1998;27:161– 195.

14. Lu XX. A review of solutions for perspective-n-point problem in camera pose estimation. *Journal of Physics: Conference Series* 2018;1087:052009.

15. Shavit Y, Ferens R. Introduction to camera pose estimation with deep learning. *arXiv preprint arXiv:1907.05272*; 2019.

16. Melekhov I, Ylioinas J, Kannala J, et.al. Relative camera pose estimation using convolutional neural networks. In Advanced Concepts for Intelligent Vision Systems: 18th International Conference. 2017 Sep 18−21, Antwerp, Belgium: Springer; 2017. 675–687 p.

17. Jacob SS, Sreeja S, Dathan NS. Evaluation of feature detection algorithms and epipolar geometry based camera pose estimation. In IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society; 2022. 1–6 p.

18. Kaur, Dilpreet, Yadwinder Kaur. Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing.* 2014:809−814.

19. Chuang YY. Camera calibration. *In tech rep Citeseer*; 2005.

20. Zhang Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000;22(11):1330–1334.

21. Hartley, Richard, Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press; 2003.

22. Ke T, Roumeliotis SI. An efficient algebraic solution to the perspective-three-point problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. 7225–7233 p.

23. Wallis, Walter D, John C. George. Introduction to combinatorics. CRC press; 2016.

24. Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the p n p problem. *International Journal of Computer Vision*. 2009;81:155–166.

25. Waller RJ, Visagie L. Pose estimation for cubesat docking. *IFAC-PapersOnLine*. 2021;54(21) :216−221.

Citation: Sharu SJ, Sreeja S. A comparative study of pose estimation algorithms for visual navigation in autonomous robots. *Int Rob Auto J.* 2023;9(3):124–130. DOI: 10.15406/iratj.2023.09.00274