IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Automatic Lighthouse Calibration Using Conics for Indoor Robot Localization

Said Alvarado-Marin[1], Alexandre Abadie, Martina Balbi, Thomas Watteyne, Filip Maksimovic

***Abstract*—In this letter, we propose a technique for calibrating Lighthouse localization systems using a single view of two or more coplanar circles traced by a moving robot. The calibration method leverages conic algebra to compute the homography between the Lighthouse view and the world plane, up to similarity. This approach requires minimal user intervention and is particularly suited for automatically calibrating large-scale deployments involving hundreds of mobile robots.**

We validate our method using a centimeter-scale differential-drive robot, utilizing 5 cm circles to calibrate a $2\times2m^2$ area. The proposed technique achieved a mean positional accuracy of 7.77 mm, compared to the 5.37 mm accuracy of a previous calibration method based on manual measurements and known correspondences. We demonstrate that the conics traced by the robot are accurate enough for reliable homography estimation, even under varying conditions of tire material and surface type. A camera-based motion capture system served as the ground truth for all experiments. This work represents a step toward scalable and decentralized lighthouse calibration, enabling efficient 2D localization in large-scale robotic systems.

***Index Terms*—Localization, Conic Sections, Wheeled Robots, Lighthouse Positioning.**

I. Introduction

THE Lighthouse localization system is a technology capable of achieving millimeter-level precision for indoor robot localization tasks [1]. One of its key advantages over camera-based systems [2] is its decentralized nature: once calibrated, each robot computes its own location without relying on a centralized controller. This is particularly beneficial for large swarms of robots with hundreds of agents [3], especially in scenarios where low-latency communication is unavailable or impractical for delivering position updates.

When a target is equipped with multiple Lighthouse-compatible sensors in known locations [4], its position can be determined by solving the perspective-n-points (PnP) problem [5]. However, in the single-lighthouse single-sensor case, localization requires time-consuming calibration. In this calibration process, the robot must be manually placed in known locations within the Lighthouse's field of view to compute the homography between the Lighthouse image and the world plane. The accuracy of this calibration directly affects the precision of subsequent tracking. Homography estimation [6] is performed using known correspondences between image features, such as points [7], discrete contours [8], or conics [9].

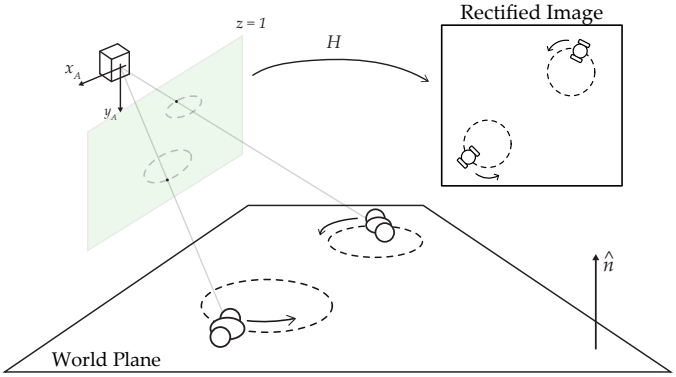


Fig. 1: A Lighthouse base station tracking two DotBots that trace circles. The circles are used to correct the perspective of the scene.

In this paper, we propose a technique that leverages a robot's motion to automatically calibrate a Lighthouse localization system. The method computes the homography up to a similarity transformation, producing a metrically rectified view of the world plane. The method presented is an adaptation of the algorithm previously proposed by Mudigonda et al. [10]. We adapt it to work with single-sensor single-Lighthouse 2D localization: we extract image features from the circular paths traced by a robot, and incorporate more than two circles for improved accuracy. We test the proposed technique using a DotBot[2] [11] tracked by a Lighthouse v2 base station. We show that our approach achieves accuracy comparable to a previous manual calibration algorithm, while requiring significantly less human intervention. We use a Qualysis motion capture system to provide the ground truth for the experiments.

II. Lighthouse Operation

The Lighthouse Positioning System was originally developed by Valve and HTC Vive to localize virtual reality headsets. It uses static base stations that emit rotating sweeps of infrared light encoded with positional information. A robot with a photodiode can detect these light pulses, decode the data, and compute its azimuth (θ) and elevation (ϕ) relative to the base station. A combination of readings from multiple lighthouses can be used to locate the robot in 3D space [12].

Manuscript received: January, 24, 2025; Revised April, 10, 2025; Accepted May, 13, 2025.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the OpenSwarm project.

[1]said-alexander.alvarado-marin@inria.fr

All authors are affiliated with Inria-AIO, Paris, FR, 75013

Dataset and code available at https://github.com/DotBots/alvarado24automatic.

Digital Object Identifier (DOI): 10.1109/LRA.2025.3575319

[2] The DotBot is an open-source robotic platform, its firmware and hardware design files can be found at https://github.com/DotBots

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Studies have demonstrated localization with as few as four sensors [13] and even with a single sensor [14], achieving centimeter- and millimeter-level accuracy, respectively. There are two hardware versions of the Lighthouse system: v1 and v2. While the algorithm presented in this study is compatible with both, this work focuses on version 2 due to its improved accuracy and wider adoption.

As demonstrated in our previous work [14], the Lighthouse base station can be modeled as a pinhole camera with an intrinsic matrix equal to the 3×3 identity matrix.

To calculate the camera points in the Lighthouse frame, we assume an image plane positioned at a distance of one unit in front of the base station. A ray is projected from the base station in the direction of the tracked photodiode, determined by the decoded angles θ and ϕ. The intersection of this ray with the image plane defines the corresponding camera point in pixel units. Equation 1 shows the conversion between spherical angles and image points in the Lighthouse frame of reference:

$$\begin{aligned} x &= -\tan(\theta) \\ y &= -\tan(\phi)/\cos(\theta) \\ z &= 1 \end{aligned} \tag{1}$$

We adopt this pinhole camera model for the remainder of the paper.

III. Alternative Calibration Methods

To motivate the use of conic algebra for Lighthouse calibration, this section reviews alternative approaches and evaluates their suitability for low-cost multi-robot systems.

One of the simplest methods for calibrating a single-camera tracking system is based on computing a homography from four known point correspondences [15]. In practice, this requires the user to manually mark the corners of a precisely measured square in the robot's plane, as done in [14]. Although this method is accurate, it is labor-intensive and does not scale well to large or distributed deployments. In a later section, we will compare our approach against this manual calibration technique.

Another strategy is to frame the calibration process as a Simultaneous Localization and Mapping (SLAM) problem. In this approach, the robot's onboard sensors (e.g., wheel encoders or IMUs) are used to track its motion relative to an initial pose. Then, multiple relative positions are used as correspondences to compute the homography between the robot's plane and the base station.

This method is unobtrusive, allowing calibration to occur continuously as the robot moves during normal operation, without requiring a dedicated calibration procedure–further simplifying the user experience. The main disadvantage is that most odometry sensors are prone to drift. Small sensor errors accumulate over time, and severely degrade the quality of the localization. Correcting these errors requires the addition of an external localization system (e.g. GPS or external beacons) [16]. This phenomenon is exacerbated when working with low-cost sensors. Thus, odometry alone is not a suitable reference for calibrating a Lighthouse system.

This issue highlights a key advantage of our proposed method. Although low-cost odometry sensors are not reliable, we demonstrate in later sections that even a low-cost differential-drive robot can trace a circle with high precision. This makes circular trajectories a practical and scalable solution for Lighthouse calibration, particularly in large multi-robot systems.

IV. Review of Conic Theory

This section summarizes the key concepts required to understand the algorithm presented in this paper. Detailed explanations can be found in work by Richter [17] and Hart [15].

Conic sections are 2D representations of the intersection between a dual cone and a 3D plane. They are well-suited for describing geometric shapes such as circles, ellipses, parabolas, and hyperbolas. The following equation defines any arbitrary conic section:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{2}$$

These can also be expressed in matrix form as:

$$C = \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix} \tag{3}$$

Points lying on the conic satisfy the condition:

$$\mathbf{x}^T C \mathbf{x} = (x, y, w) \cdot \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ w \end{pmatrix} = 0 \tag{4}$$

where $\mathbf{x} = (x, y, w)^T$ is a 2D point in homogeneous coordinates.

The conic properties most relevant to this work are as follows. When the conic is a circle, there are two imaginary points at infinity that always satisfy condition (4). These points are known as the "circular points", denoted as $\mathbf{I} = (1, i, 0)^T$ and $\mathbf{J} = (1, -i, 0)^T$. Consequently, all circles in a plane intersect each other at the circular points. Additionally, we can define a conic dual to the circular points as:

$$C_\infty^* = \mathbf{I}\mathbf{J}^T + \mathbf{J}\mathbf{I}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

The conic C_∞^* is invariant under similarity transformations. In other words, applying a homography will not affect C_∞^* if the homography represents a similarity transformation, such as a translation, rotation, or scaling. This property is shown in equation (6)

$$H_S C_\infty^* H_S^T = C_\infty^* \tag{6}$$

Where H_S is a homography that represents a similarity transformation. Thus, C_∞^* remains unchanged.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

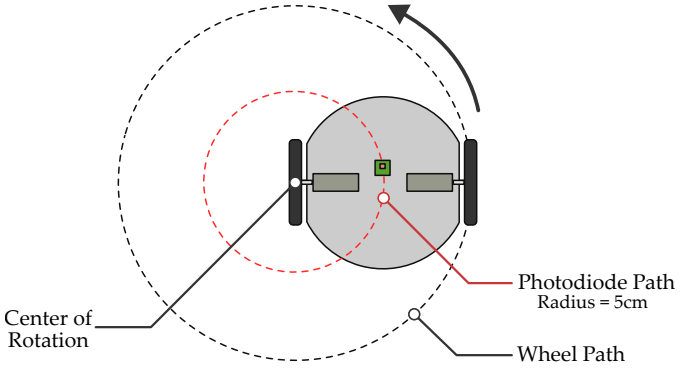


Fig. 2: Diagram of the DotBot driving in a circle by powering one of its wheels. The path traced by the photo diode is highlighted in red.

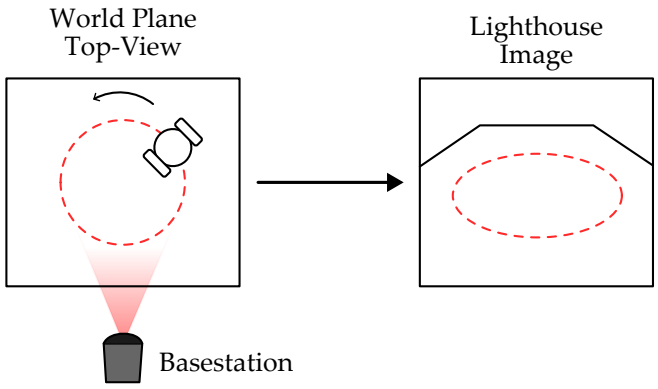


Fig. 3: Illustrating the perspective distortion introduced when a Lighthouse base station images a circle.

V. Homography Estimation Using Conics

Conic sections offer several advantages as image features: their identification is robust to noise, they compactly encode numerous data points into only six parameters, and their projective properties are well studied [18]. As a result, many techniques have been proposed for estimating homographies using conics. The homography between a plane and a camera can be computed from the images of three [19] or seven [9] conics at known locations. If two views are available, the correspondence between two unknown conics can also be used [18]. Additionally, conics can also be reconstructed from the path of a moving object, such as a turn table [20], or a robot [21].

The method proposed by Zdevsar et al. [21] achieves metric rectification using a single view of a circle traced by the motion of a robot. This approach relies on the robot maintaining a constant angular speed, allowing its position on the world plane to be estimated based on the elapsed time. A point correspondence problem is then setup to compute the homography between the estimated world-plane points and their matching image points. The advantage of this technique is that it requires only a single imaged circle. However, it imposes a strict requirement for the robot to maintain a constant speed.

In contrast, our proposed method relaxes this constant-speed requirement by leveraging information from two imaged circles. By doing so, it accommodates variations in the robot's motion while still achieving accurate metric rectification. The remainder of this section details our proposed method for homography estimation.

A. Generating the Circles

The conics in the Lighthouse image are generated by tracking a robot as it drives in a circular path. This is straightforward for a differential drive robot: by powering only one motor, the robot spins in a circle with a radius equal to the distance between its wheels, centered on the stationary wheel, as shown in Fig. 2. In the case of the DotBot, this results in the photodiode tracing a circle with a 5 cm diameter.

This approach is not entirely immune to distortions caused by wheel slippage. In practice, the robots tend to drift slightly, creating a spiral rather than a perfect circle. However, a perfectly circular path is not a strict requirement, as our experiments demonstrate that the algorithm is robust to minor imperfections in the traced circle. Detailed results of these tests are provided in section VII.

A larger and easier-to-track circle could be generated by powering both motors in the same direction at slightly different speeds. However, a longer trajectory increases the likelihood of distortions caused by occasional wheel slippage. For this reason, we rely on the more stable single-motor approach.

B. Ellipse Capture and Fitting

To capture a Lighthouse image, we place a robot equipped with a photodiode in front of a base station and record the detected Lighthouse camera points as the robot moves. These points are aggregated into a single frame, forming what we call a Lighthouse image. An example is shown in Fig. 4. Since we control the robot's motion, we can synchronize the start and stop of the recording so that only the points corresponding to a circle are collected.

Because in the general case the robot is not right underneath the base station, the circles appear as ellipses in the image, as illustrated in Fig. 3. A least-squares method is then used to fit the observed points to the general conic form in (2). We utilize the algorithm proposed by Halíř and Flusser [22], which is robust to small measurement errors.

C. Ellipse Intersection

If two circles in a plane always intersect at the circular points I and J, then the image of those circles intersects at the image of the circular points $I^{'}$ and $J^{'}$ [10]. Identifying $I^{'}$ and $J^{'}$ is a crucial step in estimating the homography required for metric rectification. Given two conics $C_1^{'}$ and $C_2^{'}$ imaged as ellipses by the Lighthouse, but known to represent circles, these conics are described by the equations:

$$\begin{aligned} C_1^{'} &= a_1^{'}x^2 + b_1^{'}xy + c_1^{'}y^2 + d_1^{'}x + e_1^{'}y + f_1^{'} = 0 \\ C_2^{'} &= a_2^{'}x^2 + b_2^{'}xy + c_2^{'}y^2 + d_2^{'}x + e_2^{'}y + f_2^{'} = 0 \end{aligned} \tag{7}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

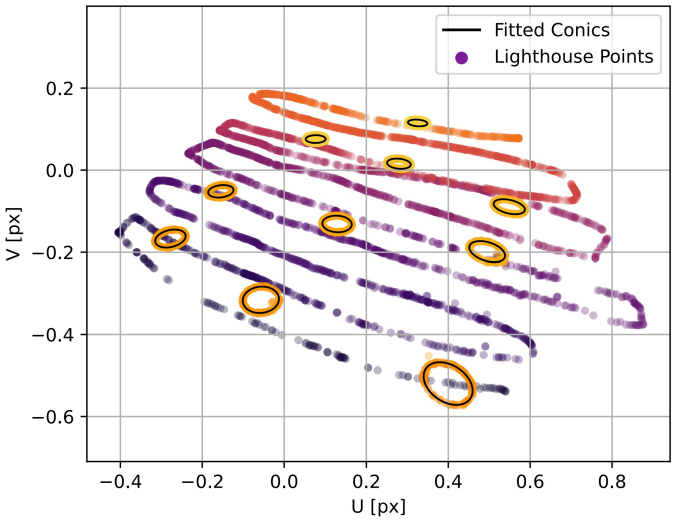


Fig. 4: Example of a Lighthouse image captured from a DotBot moving through the test area. The trajectory includes 10 calibration circles. Color indicates the passage of time, with darker points representing older measurements.

Their intersection must contain the points $I^{'}$ and $J^{'}$. Since a pair of ellipses can intersect in up to four points, care must be taken to determine which solutions correspond to $I^{'}$ and $J^{'}$. The following considerations are useful for this purpose [17]:

- By definition, circular points are imaginary points that are complex conjugates of each other. This property also applies to their images [10]. Thus, any real solutions can be safely discarded.
- The image ellipses are real, so all imaginary solutions must occur in complex conjugate pairs.
- Due to the projective distortions introduced by the Lighthouse, $I^{'}$ and $J^{'}$ should not lie at infinity. Their homogeneous component (w) should not be zero.
- If the intersection yields only one pair of complex conjugate points, these are the imaged circular points.

The case where the intersection results in two pairs of complex conjugate points is addressed in Section V-E.

On a practical note, solving for the intersection of (7) involves solving a system of equations with two variables and two second-degree polynomials. For simplicity, we use a numerical solver. Algebraic techniques, such as those proposed by Richter [17] and Chomicki [23], are also available.

D. Corrective Homography Estimation

To rectify the Lighthouse image, we compute the image of the dual conic C^*_∞. This is achieved by substituting $I^{'}$ and $J^{'}$ into (5) to obtain $C^{*}_{\infty}{}^{'}$, As shown in (8).

$$C^{*}_{\infty}{}^{'} = I^{'}J^{'T} + J^{'}I^{'T} \tag{8}$$

Next, we define H as the homography that relates the dual conic in the world plane C^*_∞ to its image captured by the Lighthouse $C^{*}_{\infty}{}^{'}$, such that:

$$C^{*}_{\infty}{}^{'} = HC^*_\infty H^T = H \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} H^T \tag{9}$$

Eq. (9) can be interpreted as a matrix decomposition of $C^{*}_{\infty}{}^{'}$. Therefore, H can be computed by performing the singular value decomposition (SVD) of $C^{*}_{\infty}{}^{'}$ [15].

$$SVD(C^{*}_{\infty}{}^{'}) = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \tag{10}$$

By comparing the right-hand sides of equations (9) and (10), we observe that their forms match. From this, we deduce that $H = U$. Consequently, the metric rectification homography is given by H^{-1}.

An important limitation of this method is that it computes the corrective homography only up to a similarity transformation. The rectified image will be fronto-parallel, but it may be arbitrarily translated, rotated and scaled. Because, as discussed earlier, C^*_∞ is invariant under similarity transformations.

E. Choosing the Correct Solution

The method presented so far is based on the assumption that a picture of a circle taken from an angle results in an ellipse. The goal is to determine the transformation that converts the ellipse back into a circle. This framework provides a way to evaluate the viability of a potential solution. Consider two ellipses $C_1^{'}$ and $C_2^{'}$ found in a Lighthouse image. Suppose their intersection yields two pairs of complex conjugate points.

We can evaluate the correctness of each solution by applying it to $C_1^{'}$ and $C_2^{'}$, and computing their eccentricity using (11).

$$e = \sqrt{\frac{2\sqrt{(A-C)^2 + B^2}}{\eta(A+C) + \sqrt{(A-C)^2 + B^2}}} \tag{11}$$

Where A, B, C, D, E, and F are the coefficients of the conic, as given by (2). $\eta = 1$ if the determinant of the conic matrix (3) is negative; $\eta = -1$ if it is positive.

For ellipses, the eccentricity serves as an indicator of "circularity". Rounder ellipses yield values closer to 0; more elongated ellipses result in values closer to 1. After rectification, the conics C_1 and C_2 should be circles and their eccentricity should be zero [24].

In practice, due to measurement noise, achieving $e = 0$ is unlikely. However, we can still identify the correct solution by selecting the homography that minimizes the eccentricity of the rectified conics C_1 and C_2.

F. Accuracy Improvement with Redundant Conics

If additional conics are available, they can be utilized to improve rectification accuracy. This is done by identifying the conic pair that yields the best rectification, following a method similar to that proposed by Lourakis et al. [25].

Our proposed method is as follows. Given N conics in a Lighthouse image, denoted by $C_N^{'}$, we exhaustively compute the corrective homography for each possible pair. We then

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

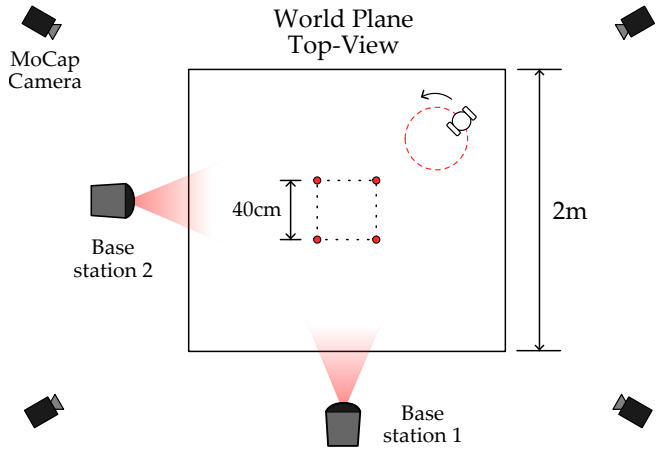


Fig. 5: Diagram of the experimental setup. A Qualisys motion capture (MoCap) system and two Lighthouse base stations track a DotBot drawing circles. Four reference points are marked in red at the center of the experimental area.

select the pair that minimizes the average eccentricity of all the post-rectification conics, as given in (12).

$$e_{ave} = \frac{1}{N}\sum_{n=0}^{N} e\left(H_{ij}^{-T} C_n^{'} H_{ij}^{-1}\right) \tag{12}$$

Where H_{ij} is the corrective homography estimated with conics $C_i^{'}$ and $C_j^{'}$.

VI. Experimental Setup

We test the accuracy of our algorithm by simultaneously tracking a moving DotBot with a Qualysis motion capture system (MoCap) and two Lighthouse v2 base stations, as shown in Fig.5. The base stations are positioned approximately 2 m apart, at a height of 1 m, overlooking the $2{\times}2$ m^2 area where the robots are. We draw a $40{\times}40$ cm^2 square on the ground and record the DotBot's position when it was placed at each corner. These four reference points are used to scale and align the data between the Lighthouse and the MoCap system. The effective range of the Lighthouse system extends well beyond the 2 $\times 2$m^2 test area. We verified that a DotBot can detect usable Lighthouse signals at distances of up to 5 m from a base station. However, our experiments are constrained by the coverage area of the MoCap system.

We compare our calibration method against two techniques presented in a previous study [14]: "One Lighthouse 2D", which relies on four manually placed calibration points, and "Two Lighthouse 2D", which uses the correspondence between the views of two base stations. The data from the MoCap system (which has an accuracy of less than 1 mm) serves as the ground truth for all the experiments. We record two separate experiments with the base stations at different angles and positions. As part of each experiment, the DotBot is placed at 10 regularly spaced positions throughout the testing area. The DotBot is programmed to autonomously trace a circle. This provides a variety of conics imaged from different angles, allowing us to thoroughly test our algorithm.

TABLE I: Circle Eccentricity of DotBot Across Different Wheel Materials and Surfaces

	Rubber Wheels	40D Wheels
Concrete Floor	0.110	0.115
Vinyl Sheet	0.191	0.126

We also evaluate the DotBot's ability to trace accurate circles under various conditions. The robot is programmed to continuously trace a circle for 60 s while being recorded by the MoCap system. This data is used to calculate the eccentricity of the traced circles. Experiments are conducted using two wheel materials: commercially available rubber tires and 3D-printed tires made from a flexible filament with a shore hardness of 40D. Additionally, the tests are performed on two surface types: a concrete floor and a wooden sheet covered with smooth vinyl.

VII. Results

Table I summarizes the results of the circle accuracy test. The conics traced by the DotBot closely approximate perfect circles, regardless of the tire material or surface type. To contextualize these results, consider the relationship between the eccentricity of an ellipse and the lengths of its semi-major and semi-minor axes, as given by (13) [24]:

$$e = \sqrt{1 - \left(\frac{b}{a}\right)^2} \tag{13}$$

Here, e represents the eccentricity, and a and b denote the radii of the semi-major and semi-minor axis, respectively. The worst-case scenario in our experiments—rubber wheels on a vinyl sheet—amounted to an eccentricity of 0.191. According to (13), this corresponds to a ratio $b/a = 0.981$. For a 5 cm circle, this means that the radii of the semi-major and semi-minor axes differ by less than 1 mm. This suggests that, while the DotBots exhibit a slight drift over time, the resulting error is insufficient to significantly distort the traced circles.

Fig. 6 shows an example trajectory segment rectified using 5 conics. Fig. 7 compares the positional error distributions for rectifications performed with five and ten regularly spaced conics. The histograms are generated by analyzing the entire dataset.

Table II provides a detailed comparison of the accuracy of all the calibration algorithms tested. The entire dataset is used for this comparison. The table shows the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Standard Deviation (SD) of the DotBot position as measured by the Lighthouse. The results show that increasing the number of conics used in calibration improves localization accuracy. Our proposed algorithm achieves accuracy comparable to the manual "One LH - 2D" method, while the "Two LH - 2D" method performs the worst among the group. This discrepancy may be attributed to lens distortion in the Lighthouse base stations. These base stations exhibit a slight fisheye effect, which reduces measurement accuracy near the corners of their field of view (FoV). Unlike the other methods, the "Two LH - 2D" approach does not use any calibration features

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

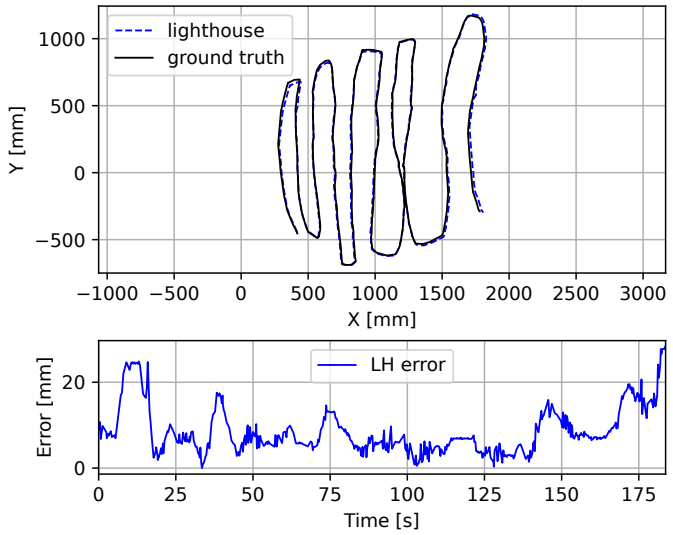


Fig. 6: The X- and Y-coordinates of the DotBot's trajectory, both measured and real, are shown below, along with the euclidean error of the trajectory.

TABLE II: Lighthouse Calibration Algorithm Comparison

Algorithm	MAE [mm]	RMSE [mm]	Std. Dev. [mm]
One LH - 2D [14]	5.37	6.93	4.37
Two LH - 2D [14]	14.90	17.60	9.32
Conic rectification 5 conics	9.14	10.90	5.95
Conic rectification 10 conics	7.77	9.27	5.05

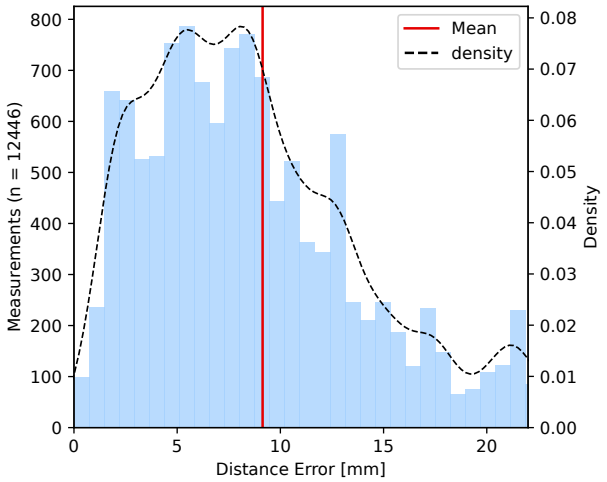
with known proportions that could help compensate for this distortion, making it more susceptible to its effects. This limitation is further exacerbated by our dataset, which covers a large portion of the base stations FoV.

Next, we study the ability of the post-rectification eccentricity to predict the accuracy of the reconstructed trajectory. We test our algorithm with every possible pair of conics captured in our dataset, and record the MAE of the reconstructed trajectory as well as the average post-rectification eccentricity. The results are shown in Fig. 8.

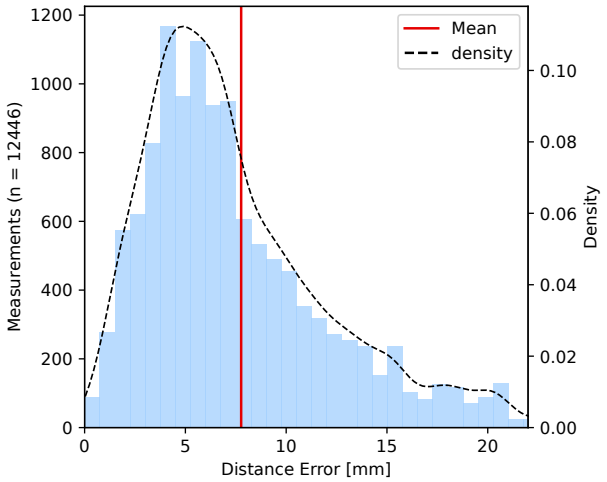
Based on the results from Fig. 8, we recommend a post-rectification eccentricity below 0.22 to ensure an accurate reconstruction. This can be achieved by starting with two conics and iteratively imaging additional circles until the algorithm identifies a pair that achieves the desired eccentricity.

VIII. Conclusion

Manual calibration remains the most common method for calibrating a Lighthouse localization system. This paper presents a calibration routine which can be done entirely automatically without human intervention, while achieving comparable accuracy to previous methods [14]. This makes it particularly useful for enabling easy, automatic calibration in large-scale deployments, where hundreds of robots and dozens of Lighthouses are involved. We verify that the accuracy of the circles traced by a differential-drive robot is sufficient to reliably estimate the homography of the world plane.



(a) Metric rectification with 5 conics.



(b) Metric rectification with 10 conics.

Fig. 7: Histogram of the Euclidean error of the metric rectified trajectory, with 5 and 10 conics.

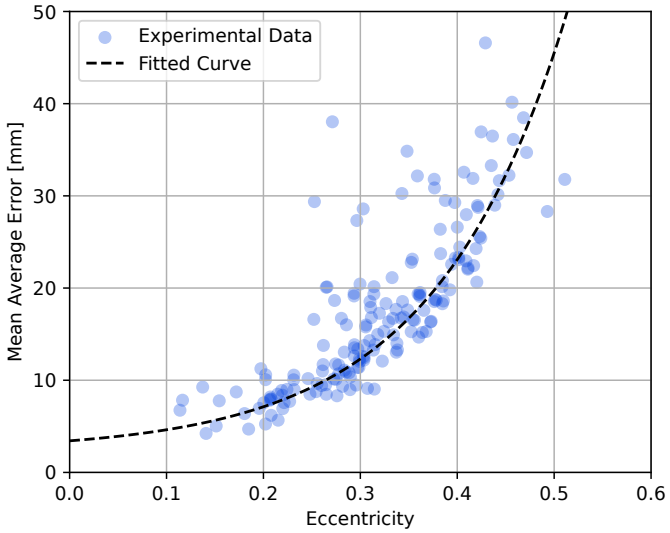


Fig. 8: Average rectification error as a function of post-rectification eccentricity.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The main limitation of the method is that the robot's motion can only be recovered up to an arbitrary scale factor. Future work could address this by leveraging the known radius of the robot's circular trajectory to resolve the real-world scale of the calibration.

We are currently working on enhancing the accuracy by leveraging the robot's internal sensors to estimate rotational speed and incorporating this information into the calibration process in a way similar to the approach of Zdevsar et al. [21]. The solution could be further enhanced by compensating for lens distortion in Lighthouse measurements caused by manufacturing imperfections. Additionally, implementing the algorithm directly on the DotBot would enable a fully decentralized localization system. Given that the algorithm is linear and computationally lightweight, such an implementation appears perfectly feasible.

ACKNOWLEDGMENT

This document is issued within the frame and for the purpose of the OpenSwarm project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101093046. Views and opinions expressed are however those of the authors only and the European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] S. Cho, D. Kim, and S. Lee, "A comparative evaluation of a single and stereo lighthouse systems for 3-D estimation," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24 791–24 800, 2021.

[2] J. A. Preiss, W. Hönig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A Large Nano-quadcopter Swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[3] G. Valentini, E. Ferrante, H. Hamann, and M. Dorigo, "Collective decision with 100 Kilobots: Speed versus accuracy in binary discrimination problems," *Autonomous agents and multi-agent systems*, vol. 30, pp. 553–580, 2016.

[4] L. Furtner, M. Schuss, M. Schuh, and C. A. Boano, "An Affordable and Easy-to-Setup Ground Truth System to Facilitate Localization Research," in *Proceedings of the 21st International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2024.

[5] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, February 2009.

[6] A. Agarwal, C. Jawahar, and P. Narayanan, "A survey of planar homography estimation techniques," *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.

[7] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[8] C. Jawahar and P. Jain, "Homography estimation from planar contours," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 2006, pp. 877–884.

[9] A. Sugimoto, "A linear algorithm for computing the homography from conics in correspondence," *Journal of Mathematical Imaging and Vision*, vol. 13, pp. 115–130, 2000.

[10] P. K. Mudigonda, C. Jawahar, and P. Narayanan, "Geometric Structure Computation from Conics." in *ICVGIP*, 2004, pp. 9–14.

[11] S. Alvarado-Marin, A. Abadie, F. Maksimovic, M. Balbi, T. Savić, and T. Watteyne, "Demo: DotBot, a cm-Scale, Easy-to-Use Micro-Robot for Swarm Research," in *"Breaking Swarm Stereotypes Workshop" at IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, May 2024. [Online]. Available: https://hal.science/hal-04673787

[12] L. Kuhlmann de Canaviri *et al.*, "Static and dynamic accuracy and occlusion robustness of steamvr tracking 2.0 in multi-base station setups," *Sensors*, vol. 23, no. 2, p. 725, 2023.

[13] A. Taffanel *et al.*, "Lighthouse Positioning System: Dataset, Accuracy, and Precision for UAV Research," in *ICRA Workshop: Robot Swarms in the Real World*, Xian, China, June 2021.

[14] S. Alvarado-Marin, C. Huidobro-Marin, M. Balbi, T. Savić, T. Watteyne, and F. Maksimovic, "Lighthouse Localization of Miniature Wireless Robots," *IEEE Robotics and Automation Letters*, 2024.

[15] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[16] S. A. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE access*, vol. 7, pp. 97 466–97 486, 2019.

[17] J. Richter-Gebert, *Perspectives on projective geometry: a guided tour through real and complex geometry*. Springer, 2011.

[18] C. Conomis, "Conics-based homography estimation from invariant points and pole-polar relationships," in *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 2006, pp. 908–915.

[19] J. Kannala, M. Salo, and J. Heikkilä, "Algorithms for Computing a Planar Homography from Conics in Correspondence." in *BMVC*, 2006, pp. 77–86.

[20] G. Jiang, H.-t. Tsui, L. Quan, and A. Zisserman, "Single axis geometry by fitting conics," in *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I 7*, 2002, pp. 537–550.

[21] A. Zdešar, I. Škrjanc, and G. Klančar, "Homography estimation from circular motion for use in visual control," *Robotics and autonomous systems*, vol. 62, no. 10, pp. 1486–1496, 2014.

[22] R. Halíř and J. Flusser, "Numerically stable direct least squares fitting of ellipses," in *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, vol. 98, 1998, pp. 125–132.

[23] C. Chomicki, S. Breuils, V. Biri, and V. Nozick, "Intersection of conic sections using geometric algebra," in *Computer Graphics International Conference*, 2023, pp. 175–187.

[24] A. B. Ayoub, "The eccentricity of a conic section," *The College Mathematics Journal*, vol. 34, no. 2, pp. 116–121, 2003.

[25] M. I. Lourakis, "Plane metric rectification from a single view of multiple coplanar circles," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 509–512.