# Single Vision-Based Self-Localization for Autonomous Robotic Agents

Marios Avgeris*, Dimitrios Spatharakis*, Nikolaos Athanasopoulos†, Dimitrios Dechouniotis*,
Symeon Papavassiliou*
*School of Electrical and Computer Engineering, National Technical University of Athens
{mavgeris@netmode, dspatharakis@netmode, ddechou@netmode, papavass@mail}.ntua.gr
†School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast
{n.athanasopoulos}@qub.ac.uk

*Abstract*—**We present a single vision-based, self-localization method for autonomous mobile robots in a known, indoor environment. This absolute localization method is landmark assisted, therefore, we propose an algorithm that requires the extraction of a single landmark feature i.e., the length of a known edge. Our technique is based on measuring the distance from two distinct, arbitrarily positioned landmarks in the robot's environment, the locations of which are known a priori. A single camera vision system is used to perform distance estimation. The developed framework is applied to tracking a robot's pose, i.e., its position and orientation, in a Cartesian coordinate system. The position of the robot is estimated using a bi-lateration method, while its orientation calculation utilizes tools from projective geometry. The validity and feasibility of the approach are demonstrated through experiments.**

## I. Introduction

Estimating a robot's pose (position and orientation) is one of the fundamental problems in mobile robotics, as achieving reliable navigation in any environment becomes more and more important. When dealing with indoor mobile robots, self-localization is important since it allows the agent to perform autonomously. Commonly, depth information regarding distinct objects in the environment is required for position estimation. This can be done with a wide range of sensors, i.e, laser range finder and sonar. However, these solutions require integration over time and high-level reasoning in order to accomplish localization [1].

On the other hand, vision has the potential to provide enough information to uniquely identify the robot's position. From the available self-localization techniques, it seems that vision-based ones are potentially the most flexible and powerful source of information for such a task [2]. Vision-based techniques are also closely intertwined with the robotic agent's environment; elements like ceiling lights and door frames can be utilized for some of their features which are either color transitional or line-based, e.g. the perceived length of a known horizontal edge [3]. Such techniques are based on passive beacon ("natural landmark") detection; in general, landmarks have a known, fixed position, relative to which a robot can localize itself. Thus, the main task in self-localization is to recognize such landmarks reliably and to subsequently compute the robot's position. When dealing with natural landmarks, ambient conditions, such as lighting, can be proven problematic, hence more computational power might be necessary. On the other hand, active beacon navigation, i.e. when the environment needs to be modified with stationary beacon systems which require electric outlets or battery maintenance, provides very accurate localization information with less processing effort. This approach naturally yields higher reliability, but the costs of the installation and maintenance are also higher [4].

In this study, the testbed of our algorithm is a vehicular mobile robot development platform, called AlphaBot [5] (Fig. 3), equipped with a single frontal pivoted camera; omni-directional cameras with viewing angle of 360° could potentially provide more landmarks but suffer from higher cost due to their mirror system, low resolution of the camera, and the requirement of an additional space to fit the mirror and the camera. Differently, with frontal cameras one can have high resolution but the viewing field is limited. That is why our algorithm is extended to utilize the pivoted stereo head to successfully detect the landmarks.

The application setting envisioned resembles that of a warehouse, a factory floor or a ship's cargo hold. In this case, the robot's environment is considered known, relatively static and slightly modified with colored cylindrical navigation aids to make the algorithm's evaluation smoother, as distinct landmarks are naturally scarce. Therefore, our position estimation algorithm has to be based on the minimum number of distinct landmarks. The fact that our approach works with the simplest camera system and the minimum amount of identified natural landmarks, is what differentiates it from similar works in the field.

The rest of the paper is structured as follows. In Section II, the current state of the art is presented, while in Section III the single vision-based self-localization algorithms is broken down into three distinct steps. The experimental setup and the evaluation findings are

978-1-7281-4411-5/19/$31.00 ©2019 IEEE
DOI 10.1109/FiCloudW.2019.00035

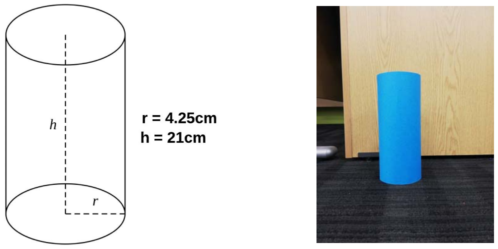
presented in Section IV, while conclusions and future plans are exposed in Section V.

## II. Related Work

During the last few years, significant effort has been put into estimating a robot's pose, i.e. position and orientation, through the use of a sensing system, e.g., GPS, infrared or ultrasonic distance sensor, lidar, accelerometer / compass / magnetometer or a camera. Among these, the image-based camera localization is the most flexible and low cost approach [6], but at the same time it is the most complex one. As a result, many research works have been involved with developing novel techniques for dealing with the peculiarities on this domain.

In [7] the authors discuss the localization of small, autonomous mobile robots using a single landmark feature, found in the environment, specifically a color transition, a junction or a line intersection. Ultimately, the position and orientation of the robot is calculated by estimating the distance from a single landmark, from two arbitrary points, whose displacement can be measured using dead-reckoning sensors. Distance measurements are performed through a stereo vision system. A different approach is followed in [8], where a database of landmark images from representative viewpoints is compiled off-line to be later used in an on-line position estimation process; during this, observed landmarks are matched to stored landmarks and the transformation subspace that relates the observed ones with a set of tracked ones is exploited to perform the localization. A novel algorithm for landmark matching, triangulation reconstruction and comparison (LTRC) is demonstrated in [9]. In a similar fashion to the previously mentioned work, landmark matching is also performed here, with a panoramic camera this time. In this case, at least three landmarks are identified and triangulated to estimate the current robot position. Ambiguities during this process are resolved in a reconstruction and comparison stage where the most realistic estimation is selected.

Strategic placement of the landmarks is, finally, demonstrated in [10]; a bilateration method is utilized instead of trilateration, as the distance from exactly two landmarks is sufficient for providing a unique pose of the robot. The ambiguity between the two possible poses, resulted from the bilateration, is resolved by considering a fixed order of the landmark points. The camera system used here is omni-directional.



(a) Beacon Schematic

(b) Actual Beacon

Fig. 1: Landmarks

## III. Single Vision-Based Self-Localization

In our case, in order to estimate the AlphaBot's 3-degrees-of-freedom pose in the grid, in terms of $(x, y)$ coordinates and $\theta$ heading orientation, a single vision -based self-localization algorithm was developed. This algorithm is classified as single vision -based, because the robot uses a single frontal camera. Moreover, it is landmark-assisted, since it requires a single landmark feature (width of beacon) to work. The approach taken here is feature-based and relies on the principles of projective geometry. In particular, the cylinder was selected as the shape of the Beacons (Fig. 1), because of its interesting property; its 2D projection is a rectangle, independently of any viewing angle that has a rotation axis parallel to the cylinders axis. The Beacons are colored differently from the environment colors, e.g., blue, red, orange and green, in order to facilitate the detection from the AlphaBot's camera. The camera mounted on the AlphaBot has a 30-degree horizontal angle of view, so a rotation of six 30-degree steps is needed in order to cover the 180-degree-area in front of it, on aggregate, and detect two, at least, Beacons. This requirement is discussed further on Subsection III-A. The following three steps are performed:

### A. Beacon Recognition

In order to detect the presence of a Beacon within an image, the Python OpenCV Library [1]was utilized; first, the image is transferred to the HSV (Hue, Saturation and Value) color space, because this conversion is robust towards external lighting changes. In particular, in cases of minor changes in external lighting, such as pale shadows, Hue values vary relatively less than RGB values. After this, the algorithm applies an offline calculated HSV mask to the image, acting as a color filter for each of the Beacon colors. Then, it groups the adjacent filtered pixels and draws the minimum-area rectangles that surround each of these groups. This mask consists of a set of lower and upper values regarding the Hue, Saturation and Value of each color, acting as boundaries. In this study, the following ranges where used: $H \in [0^o, 180^o]$, $S \in [0, 255]$ and $V \in [0, 255]$. For example, the $[H, S, V]$ mask corresponding to blue colored pixels is: $lower[30, 75, 100]$ and $upper[110, 255, 255]$.

Next, from the rectangles drawn on the image, the ones that possess the following features are considered to be classifiable as a Beacon:

1) The identified rectangle is in upright position.

[1] https://github.com/opencv/opencv

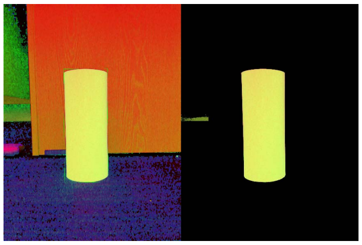


Fig. 2: Beacon contour detection in HSV color space.

2) Its shorter side is parallel to the $x$ axis and its longer side parallel to the $y$ axis of the image plane.
3) Its aspect ratio stays between Beacon-specific, predefined, boundaries.

These simple criteria are defined to filter out objects on the field that are similarly shaped and colored as the Beacons. If no rectangle fits these criteria, then it is assumed that the image does not depict a Beacon in whole. On the other hand, if more than one Beacons are detected, a selection is made to consider only one of them. Fig. 2 depicts the result of the above process. The resulting information retrieved is the perceived width $p$ of the contour rectangle surrounding the Beacon, in pixels.

### B. Distance and Angle Relative to Beacon Estimation

After detecting a Beacon within an image, the process of estimating the AlphaBot's distance and angle from this Beacon, takes place. Performing this step twice, for two distinct Beacons, allows the deduction of the exact position and pose of the AlphaBot on the grid. It is worth noting that in the scope of this work, we assume the positions of the Beacons to be known. However, we believe this method to be extensible to the case where the positions of the Beacons is unknown, e.g., where a Simultaneous Localization and Mapping (SLAM) technique can be leveraged to initially identify these positions. The camera mounted on the AlphaBot follows the pinhole camera model [11]; that means that the relative size of the projected objects depends on their distance to the focal point. To find the distance, we utilize the *triangle similarity theorem*, i.e., the distance of the object to the camera, $d_c$, is given from the following equation:

$$d_c = \frac{wf}{p}$$

where:

$w$ = Beacon width in $cm$.
$f$ = camera's focal length im $mm$ (known from camera's datasheet or computed through camera calibration).
$p$ = perceived Beacon width in pixels ($px$).

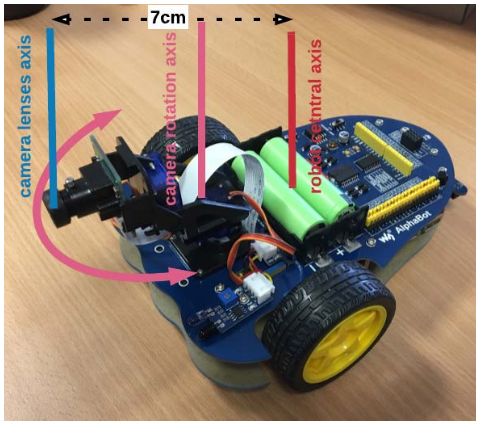


Fig. 3: AlphaBot's main axes.

We should note here that across the localization process we consider the AlphaBot to be a dimensionless point on the center of its wheel axis. However, the AlphaBot's camera lenses axis is placed $7cm$ from the robot's center, as shown in Fig. 3. Thus, the actual estimated distance, $d$, between the Beacon and the AlphaBot is:

$$d = d_c + 7$$

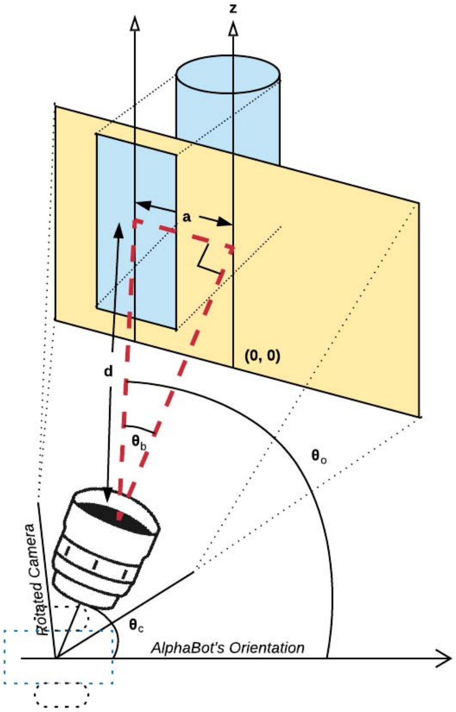


Fig. 4: Method for calculating the angle between the AlphaBot and the Beacon.

The core novelty of our localization method lies in the calculation of the angle between the AlphaBot and the Beacon. As shown in Fig. 4, we assume the 2D projection of the Beacon on the plane of the captured image. We also assume the origin $(0, 0)$ at the middle of the bottom border of the image plane and the axis $z$, coming through it, as shown in Fig. 4. To calculate the angle $\theta_b$ between the camera's line of sight and the line which starts from the camera lens and is perpendicular to the Beacon's axis, $d$ and $a$ are required. An insight of this angle's real-world nature would be this: "the angle that the camera has to rotate to horizontally centre the Beacon's 2D projection on the image plane". The distance $a_p$, in pixels, is the perpendicular distance between the axis $z$ and the Beacon's axis, which are parallel to each other, and can be readily calculated as the contour's vertices coordinates are known from the last step. The distance between the Beacon's axis and the AlphaBot, $d$, in cm, was calculated in the previous step as well. Hence, it is only needed to translate the distance $a_p$ to the distance $a$ in cm. To enable this conversion, we first ensure that the cm-per-pixel ratio, which applies to the Beacon's 2D projection on the image plane, is preserved throughout the rest of the plane. This holds true, as the real-world $z$ axis and its projection on the image plane coincide, which subsequently means that the real-world distance $a$ and its projection coincide as well. Moreover, the pixel size has the same cm length, independently of the pixel's position through the camera's conformity with the pinhole model, which makes the projection free of any linear distortion. Consequently, we have

$$a = a_p(\frac{w}{p}),$$

where $\frac{w}{p}$ is equivalent to the cm-per-pixel ratio. With $a$ and $d$ known, we can calculate $\theta_b$,

$$\theta_b = arcsin(\frac{a}{d}).$$

The approach still works when other objects, such as obstacles in the environment, are depicted in the projection. The only constraint is that the Beacon has to be captured in whole. The last thing to note is that, as mentioned earlier, the camera rotates on its pivoted system in order to scan the area in front of the robot for Beacons. However, the angle $\theta_c$ to which the camera is rotated is known. As a result, the overall $\theta_o$ to which the AlphaBot is rotated, with the given Beacon as reference, is

$$\theta_o = \theta_c + \theta_b.$$

### C. Grid Position and Orientation (Pose) Estimation

After having the distance, $(d_0, d_1)$, and angle $(\theta_{o_0}, \theta_{o_1})$ from two Beacons available, the AlphaBot's pose in terms of position and orientation is estimated. The locus formed by the set of possible $(x, y)$ locations whose distance from Beacon $P_i$ equals the estimated distance $d_i, i \in [0, 1]$, is a circle. This observation allows us to utilize the bilateration method in order to estimate the position of the AlphaBot, as shown in Fig. 5 with $P_3$; this method has been used extensively in previous works regarding localization in wireless sensor networks [12], as it requires much lower computational complexity, yet still retains the same localization accuracy, if the environmental setup allows it.

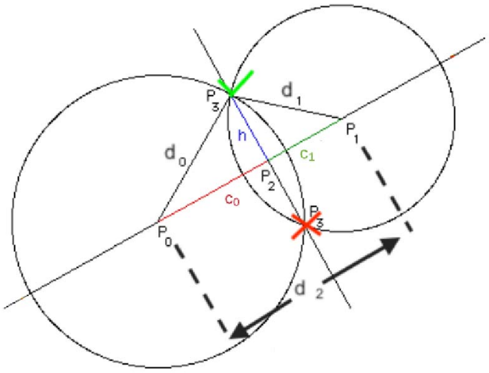


Fig. 5: Bilateration method for calculating AlphaBot's position in the grid.

In our setting, we are able to retrieve a unique solution of the location of the AlphaBot by combining the knowledge of the relative angle observations $\theta_{o_0}$ and $\theta_{o_1}$. Indeed, for the two (at most) candidate locations that the observations were taken as shown in Fig. 5, there is always exactly one feasible configuration that allows both angle values to be attained, or equivalently, that result in the same absolute angle $\theta$ estimation. This is demonstrated in Fig. 6; for example, let $\theta_{o_1} = 20°$ and $\theta_{o_2} = 150°$. As shown there is only one feasible point where both angle measurements are verified.

Our developed method requires only two beacons for localization, under the assumption of course that the

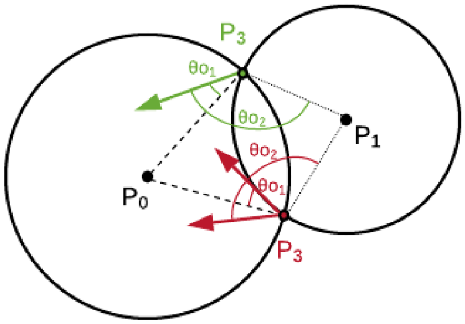


Fig. 6: Method for invalidating one of the two possible solutions.

measurements are accurate. Nevertheless, the problem of placing the minimal number of landmarks in the map still remains. This number depends on the viewing angle of the camera and the density of the obstacles, or equivalently, visibility of the beacons from all directions. For our setting where the viewing angle is $180^o$ and for a rectangular map with obstacles not obstructing the visibility of the beacons, the minimum number of beacons need to be placed is $4$. The problem becomes significantly harder for non-convex and/or non-static maps and tall obstacles, and is the object of our future research.

The bilateration process is briefly depicted in Fig. 5; the mathematical justification behind calculating the circles' intersection points is the following; considering the two triangles $P_0P_2P_3$ and $P_1P_2P_3$ we can write

$$c_0^2 + h^2 = {d_0}^2$$

and

$$c_1^2 + h^2 = {d_1}^2$$

where $c_0$ and $c_1$ are the distances of $P_0$ and $P_1$, respectively, from the bisector coming through the two intersection points of the circles and $c_0 + c_1$ equals the distance $d_2$ between the two Beacons. Using $d_2 = c_0 + c_1$ we can solve for $c_0$,

$$c_0 = \frac{{d_0}^2 - {d_1}^2 + {d_2}^2}{2d_2}$$

Then we solve for $h$ by substituting $c_0$ into the first equation, $h^2 = {d_0}^2 - c_0^2$, so we get

$$P_2 = \frac{P_0 + c_0(P_1 - P_0)}{d_2}$$

And finally, $P_3 = (x_3, y_3)$ in terms of $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, is either

$$x_3 = \frac{x_2 + h(y_1 - y_0)}{d_2}, y_3 = \frac{y_2 + h(x_1 - x_0)}{d_2}$$

or

$$x_3 = \frac{x_2 - h(y_1 - y_0)}{d_2}, y_3 = \frac{y_2 - h(x_1 - x_0)}{d_2}$$

As mentioned above, one of the two solutions is always rejected as invalid.

The final part of the localization process is to calculate the AlphaBot's orientation in the grid with respect to a given reference point. As a first step, we assume a Reference Point in a known location on the AlphaBot's South, in Cartesian coordinates; as shown in Fig. 7, the exact locations of both the Beacon ($P_0$) and the AlphaBot are known by now, thus calculating the distances $b$ (Beacon - Reference Point) and $r$ (AlphaBot - Reference Point) is straightforward. Also, distance $d_0$ (Beacon - AlphaBot) and angle $\theta_o$ (AlphaBot's angle with Beacon as reference) have been calculated in the previous steps; hence, by utilizing the cosine rule, the $\theta_t$ angle can be obtained:

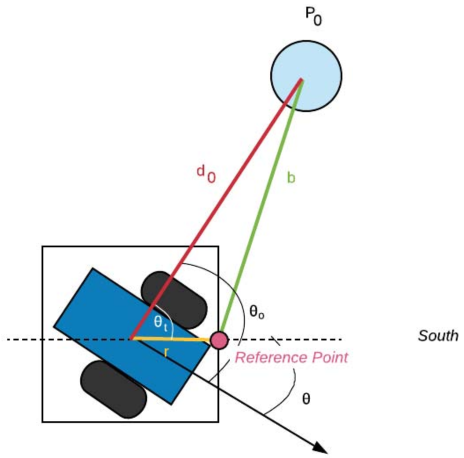


Fig. 7: Method for calculating the angle between the AlphaBot and the Beacon.

$$\theta_t = arccos(\frac{b^2 + r^2 - {d_0}^2}{2br})$$

The actual orientation angle, $\theta$ is given from the following subtraction:

$$\theta = \theta_t - \theta_o$$

## IV. Experimental Evaluation

For the evaluation of the proposed self-localization technique, experiments were conducted in a floor space of $2.5$ square meters, hereinafter mentioned as the "operating space". The evaluation of the proposed technique is broken down into two parts; *i)* the association of the perceived distance's error with the real distance from the detected Beacon and *ii)* the overall accuracy of the final estimation of the AlphaBot's pose.

As depicted in Fig. 8, the AlphaBot is located between $50cm$ and $250cm$ from the Beacon of interest. The distance of $50cm$ corresponds to the minimum distance from which a Beacon can be portrayed in whole with the current camera setup. One can notice that the absolute error of the distance-to-Beacon estimation increases gradually as the distance increases, but the accuracy never drops bellow $93\%$. Moreover, the different relative orientations of the AlphaBot seem to have a negligible effect in the accuracy of the distance estimation; $-30°$, $0°$ and $30°$ were randomly selected to illustrate this behaviour. We must highlight that the most accurate estimations, though, were observed when the real distance between the AlphaBot and the identified Beacon

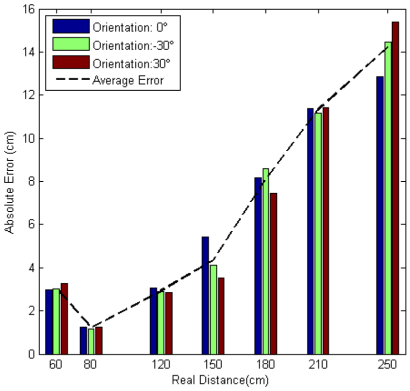


Fig. 8: Absolute error of the estimated distance relative to the real distance from a Beacon, for different orientations.

was in the range $[80cm, 100cm]$, as the average of the estimation's absolute error was in the area of $1.3cm$, or approximately $1\%$.

To illustrate the overall accuracy of our self-localization method, we composed a random walk for the AlphaBot to perform on the aforementioned operating space; the robot followed a predefined trajectory of random poses and estimated its position and orientation at each point. In Fig. 9 the trajectory of the real positions is depicted with the blue dashed line, having at each point a specific orientation depicted with blue arrows, while the estimated positions and orientations are depicted with red dashed lines and green arrows respectively. The lines connecting the different points do not represent the actual movement of the AlphaBot but are drawn for clarity. The deviation between the real poses and the estimated ones produced by the proposed algorithm for this random walk, is considered acceptable for the selected application. We note that in a typical setting where the robotic agent moves autonomously, the measurements generated by our method can be fed to a state observer of the robot's position and orientation, improving significantly the accuracy.

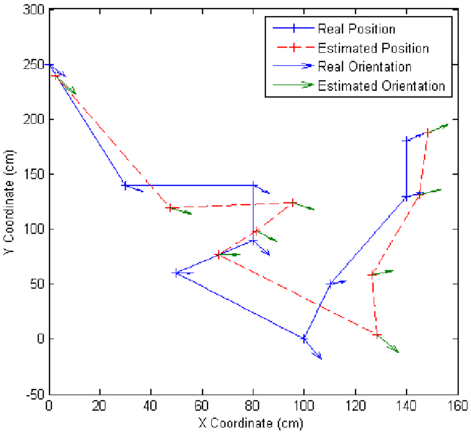


Fig. 9: Real versus estimated AlphaBot trajectory.

When comparing the estimated poses with the real ones, it can be noticed that the combined coordinates error, after the bilateration of the two relative distances takes place, never exceeds $20\%$ in either $x$ or $y$ axis. Regarding the estimation of the orientation, at each point, the absolute error lies in the $[2°, 12.5°]$ range. All in all, when a Beacon is correctly detected within the captured image, we notice that the proposed method is not only precise but also independent of environmental variables, e.g., light conditions, when it comes to pose estimation.

## V. Conclusions

In this paper we proposed a vision-based self localization approach for indoor autonomous mobile robots. Based on a bilateration method and some core principles of the projective geometry, our algorithm requires the detection of distinct landmarks in the environment and the calculation of the robot's relative distance from them in order to obtain the robot's pose $(x, y, \theta)$. Distance calculation is based on feature extraction from the landmarks. The localization algorithm has to rely on the minimum number of landmarks as they are scarce in our application's setting, thus a bilateration approach that requires the identification of two landmarks was used.

The experimental evaluation showed that when the robot's real distance from the detected landmark resides in the interval $[80, 200]cm$ the accuracy of estimation lies above 95%, independently of the robot's relative orientation. Regarding the orientation calculation, a quite stable, average error of $6°$ was observed during the experimentation. Finally, to illustrate the overall efficiency of the proposed localization algorithm, a trajectory of poses was given to the robot to follow, the results of which were more than adequate, for a real world application.

The main drawback of this method is that it is heavily affected by ambient conditions, as any other passive beacon -based technique. As a result, our future plans include the utilization of better camera equipment (ultra wide angle / high resolution lens), and the improvement of the feature extraction software component, for more efficient interpretation of landmarks. Also, it is in our future intentions to incorporate this technique to path planning problems for autonomous systems.

## Acknowledgment

This work was supported in part by H2020 Fed4FIREplus Project under Grant Agreement No. 732638.

# REFERENCES

[1] A. Bais, R. Sablatnig, J. Gu, Y. M. Khawaja, M. Usman, G. M. Hasan, and M. T. Iqbal, “Stereo vision based self-localization of autonomous mobile robots,” International Workshop on Robot Vision, Springer, Berlin, Heidelberg, pp. 367-380, February 2008.

[2] R. Vivacqua, M. Bertozzi, P. Cerri, F. Martins, and R. Vassallo, “Self-Localization Based on Visual Lane Marking Maps: An Accurate Low-Cost Approach for Autonomous Driving,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, pp. 582-597, October 2017.

[3] F. Qiang, and C. Xie. “The vision-based metric self-localization of indoor mobile robot using projective geometry,” Proceedings 7th International Conference on Signal Processing, IEEE, ICSP’04, vol. 2, pp. 914-917, 2004.

[4] G. Adorni, C. Stefano Cagnoni, and M. Monica, “Landmark-based robot self-localization: a case study for the RoboCup goal-keeper,” Proceedings 1999 International Conference on Information Intelligence and Systems, IEEE, pp. 164-171, 1999.

[5] Waveshare.com. (2019). AlphaBot - Waveshare Wiki. [online] Available at: https://www.waveshare.com/wiki/AlphaBot [Accessed 22 May 2019].

[6] Y. Wu, F. Tang, and H. Li, “Image-based camera localization: an overview,” Visual Computing for Industry, Biomedicine, and Art, vol. 1, no. 1, September 2018.

[7] A. Bais, R. Sablatnig, and J. Gu, “Single landmark based self-localization of mobile robots,” The 3rd Canadian Conference on Computer and Robot Vision (CRV06), pp. 67-67, October 2006.

[8] R. Sim, and G. Dudek, “Mobile robot localization from learned landmarks,” In Proceedings, 1998 IEEE/RSJ Inter national Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications, vol. 2, pp. 1060-1065, October 1998.

[9] D. C. Yuen, and B. A. MacDonald, “Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison,” IEEE Transactions on robotics, vol. 21(2), pp. 217-226, April 2005.

[10] A. Bais, and R. Sablatnig, “Landmark based global self-localization of mobile soccer robots,” In Asian Conference on Computer Vision, Springer, Berlin, Heidelberg. pp. 842-851, January 2006.

[11] M. Young, “The pinhole camera: Imaging without lenses or mirrors,” The Physics Teacher, vol. 27(9), pp 648-655, 1989.

[12] X. Li, B. Hua, Y. Shang, and Y. Xiong, “A robust localization algorithm in wireless sensor networks,” in Frontiers of computer science in China, vol 2(4), pp. 438-450, 2008.