Absolute Localization for Low Capability Robots in Structured Environments Using Barcode Landmarks

Received 10th October 2012; accepted 22nd November 2012.

Duarte Dias, Rodrigo Ventura

Abstract:
Research on multi-robot systems often demands the use of a large population of small, cheap, and low capability mobile robots. Many non-trivial behaviors demand these robots to be localized in real-time. This paper addresses the problem of absolute localization of such low capability robots using onboard sensors and local computation. The approach is based on the use of a pair of scan lines perceived by an onboard B&W camera to detect and decode artificial visual landmarks deployed along the environment. Each landmark consists on a dual-layer barcode which can encode its pose with respect to a global coordinate frame. Thus, the robot is not required to store a map of the landmark locations onboard. The method is based on an Extended Kalman Filter (EKF) fusing odometry readings with absolute pose estimates obtained from the camera. Experimental results using an e-puck robot with 8 KB of RAM and a 16 MIPs processor are presented, comparing the location estimates with both ground truth and odometry.

Keywords: *vision based barcode detection, homography transformation, absolute robot localization, low capability robots*

1. Introduction

There are several multi-robot systems which make use of numerous relatively simple robots for potential group-level benefits including scalability, flexibility, and robustness to individual failures. Swarm robotics is a good example of such a system [1, 2]. However, due to practical reasons (*e.g.*, cost), each robot contains reduced sensing, energy and computational capabilities. This makes it difficult to include localization systems necessary for most practical, task oriented, applications, since these involve the movement of individuals and/or the transport of objects through precise goal points. This paper addresses this problem by proposing an onboard localization system for low capability robots.

The localization system is based on an EKF, fusing movement predictions directly provided by the noisy measurements of the wheel movement, with absolute pose estimates obtained from scanning barcode landmarks with a camera, onboard the robot. Each barcode is uniquely identified by its embedded code. However, in presence of critical memory space restrictions, the embedded code can also encode the landmarks absolute poses in the environment, thus preventing the usage of lookup tables. A simplified version of the standard barcode encoding methods is used to save computational effort. This work assumes structured environments based on corridors and vertical white walls. The robot used to test the developed system is the e-puck educational robot from EPFL university, due to its extremely low computational (16 MIPs) and memory (8 KB) capabilities, and also the broad development support since it has a large userbase. Its reduced memory only allow for the extraction of 8 B&W lines of camera image. More information about the e-puck features and architecture can be found in [3].

Some localization systems have been developed for the e-puck. In [4] a decentralized localization system is described, using an external mobile component with high computational capability and preforming autonomous localization, to compute the position in the environment of each e-puck. In [5], another decentralized system is proposed, in which an external component receives the sensor data from the e-puck camera and performs landmark detection and localization processing with particle filters. The landmarks are colored which demands more computational and memory capabilities. In [6], a centralized system was proposed, also based on particle filters for localization, using gray images to extract bearing information from black landmarks in the scenario. This system is lighter in terms of landmark detection but it allows only for relative positioning of the robot to the landmarks, requiring extra model matching techniques [7] to obtain the absolute position of the robot.

The first two systems involve external components to compensate for computational limitations, which can compromise navigation decisions through communication overhead, and is sometimes impractical due to communication limitations. The last two systems require a lookup table for the landmark positions, which compromises the scalability of the number of landmark used, bounding the environment size with respect to the landmark density necessary to achieve the desired localization performances. This factor is here of extreme importance given the low memory space of the used robot.

Several algorithms that can perform barcode detection already exist for simple portable 2D cameras. The detection process can be divided into three parts: 1) Barcode localization in the image: most methods extract regions where it is most likely the barcode to exist, called regions of interest (ROI), using gradient methods [8–11] and wavelet transforms [12] to find regions with high unidirectional derivative density caused by the barcode stripes, or binarization methods to find dense black and white regions [13]. 2) Obtaining barcode geometric parameters such as the lengths of projected stripes in the camera, and geometric distortions induced by the perspective transformation inherent of the observation point: in [9, 12, 13] single scan lines are used to extract the stripe length, and in [8, 10] a perspective transformation estimation, intrinsically containing the stripe length and the geometric distortions, is performed. 3) Decoding the embedded code, directly using the scan lines and the computed geometric parameters to obtain the barcode stripes, and using the barcode encoding to decode them.

The described methodologies use the whole image and introduce great computational complexity, in terms of barcode localization, through image processing. This paper introduces a simplified 1D version of the above methods, limited to the possible 8 B&W camera lines, based on additional restrictions for the barcode positioning on the environment.

2. Barcode Localization

2.1. Barcode Design Framework

To maximize the number of encoded bits, the used barcode type, illustrated in Fig. 1 is composed of two layers of N_{stripes} geometrically identical white (spaces or '0's) and black (bars or '1's) stripes of X_{dim} width and $\frac{h}{2}$ height. It is composed of three major areas:1) embedded code, which contains the unique barcode sequence, and can also allow the encoding of the binary representation of the barcode's pose in the environment, $(x_w^b, y_w^b, \theta_w^b)$; 2) guard sets, included at each barcode horizontal edge, with N_{guard} bar-space pairs; 3) quiet zones, composed of a set of spaces placed around the barcode to prevent detectors from confusing the barcode's signal from the rest of the signal. Fig. 2 depicts the geometry of the problem. A 3D frame associated to each barcode is inserted in barcode's center, thus defining its pose in the environment. The z axis defines the direction from the lower layer to the upper layer.

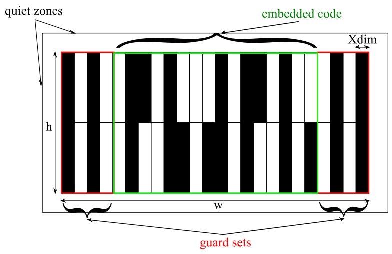


Fig. 1. Barcode major areas and geometric parameters.

The camera, modeled here by the pinhole model, with a focal length of f and a principal point of coordinates (o_x, o_y), uses two horizontal scan lines, equally distant from the optical axis, for barcode detection. The scan lines are taken from the top and bottom of the captured image, which contains Q_x pixels width and Q_y pixels height, producing respectively, horizontal and vertical fields of view, named FOV_x and FOV_y. The vertical center of the barcode must be aligned with the camera's height to ensure each scan line is aligned with its layer.

Fig. 2 and Fig. 3 illustrate a barcode observation situation and its respective projection in the ground plane. The intersections of the optical axis, r_o, and its orthogonal axis, r_p, with the y axis of the barcode are named respectively, B_2 and B_1. From Fig. 3, it can be concluded that a point in the barcode surface, b, relates to the respective projected point in the camera plane, Δ, by the following:

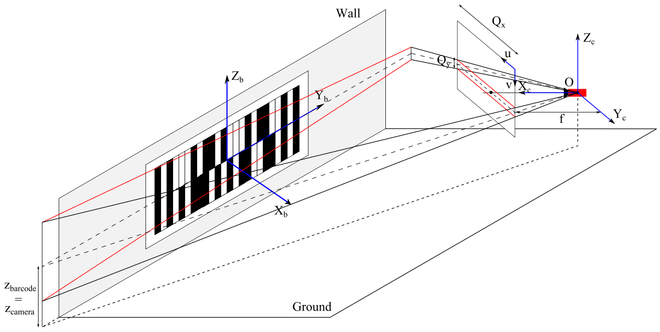


Fig. 2. Geometry of the problem: a double layered barcode scan with the chosen scan lines for each layer and the respective projections in the barcode surface (in red/darker).

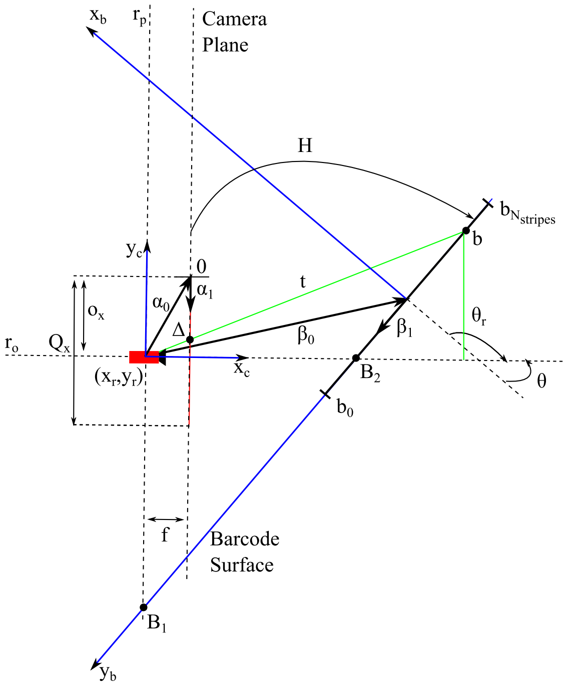


Fig. 3. The Ground projection of a double barcode scan, seen through the camera frame.

$$\frac{o_x - \Delta}{f} = \frac{\cos(\theta)}{\sin(\theta)} \frac{b - B_2}{B_1 - b} = m\frac{B_2 - b}{B_1 - b} \qquad (1)$$

where θ is the inclination of the camera to the barcode and m is the slope of r_p defined as $-(\tan(\theta))^{-1}$. Relevant θ's are contained in the $[-\frac{\pi}{2}, \frac{\pi}{2}]$ interval, but due to the symmetry of the problem one only needs to consider the $[0, \frac{\pi}{2}]$ part to analyze the problem for the whole interval. Considering the points associated to the stripe transitions of the barcode, respectively b_k and Δ_k, the width of each barcode stripe projected in the camera, L_k, can be defined as $|\Delta_{k-1} - \Delta_k|$. Using (1), this width can be expressed as follows:

$$L_k = fX_{\text{dim}}m\frac{B_1 - B_2}{(B_1 - b_{k-1})(B_1 - b_k)}, \qquad (2)$$

where $b_k = (0.5N_{\text{stripes}} - k)X_{\text{dim}}$. The minimum observation boundary which allows barcode detection, for each θ, is described by the geometrical *locus* from where at least one barcode edge coincides with the limits of the camera's FOV_x, defined by two lines, with the following expressions:

$$y = b_0 + \tan\left(\theta - \frac{FOV_x}{2}\right) x \qquad (3)$$

$$y = b_{N_{\text{stripes}}} + \tan\left(\theta + \frac{FOV_x}{2}\right) x \qquad (4)$$

The maximum observation boundary, for each θ, is described by the geometrical *locus* from where the camera is observing the barcode with at least one projected stripe with the minimum length that still allows barcode detection, L_{min}. Replacing, in (2), L_k with L_{min}, and B_1 and B_2 with their respective expressions in function of x and y, respectively defined by the line expressions of r_p and r_o, one can describe this geometrical place as an hyperbola:

$$x^2 A_{xx} + y^2 A_{yy} + 2xyA_{xy} + 2xB_x + 2yB_y + C = 0 \quad (5)$$

$$A_{xx} = 1 \qquad A_{yy} = \tan^2(\theta) \qquad A_{xy} = \tan(\theta)$$

$$B_x = \frac{X_{\text{dim}}}{2}\left((N_{\text{stripes}} - 1)\tan(\theta) - \frac{f}{C_{\text{min}}}\frac{1}{\cos^2(\theta)}\right)$$

$$B_y = \frac{X_{\text{dim}}}{2}(N_{\text{stripes}} - 1)\tan(\theta)$$

$$C = -\frac{X_{\text{dim}}^2 (N_{\text{stripes}} - 2) N_{\text{stripes}} \tan^2(\theta)}{4}$$

The Nyquist theorem for digital signal recovery states that the receiver cannot recover signals with frequencies higher than one half the sampling period. Since the camera samples the signal with a period of 1 pixel, the lowest period allowed for the signal is of 2 pixels, which means L_{min} is 1 pixel. The detection area, for each θ, is found between the minimum and maximum observation boundaries, defined respectively, by (3) and (4) and by (5). Also, the scan line inclination cause, from a certain distance, d'_{max}, illustrated in Fig. 4, parts of the barcode not to be captured, thus precluding the detection. The respective distance to the barcode center, can be expressed as:

$$d = d'_{\text{max}} - 0.5\, w \sin(\theta) = h\frac{f}{Q_y} - 0.5\, w \sin(\theta) \quad (6)$$

Expressions (3) to (6) constrain the following design variables: the number of stripes (N_{inf}), their size (X_{dim} and h), the camera's focal length (f) and its image size (Q_x and Q_y). They provide a framework which allows the choice of the barcode and camera parameters according to the detection area requirements.

2.2. Barcode Detection Algorithm

Each barcode layer is independently detected by each respective scan line. Fig. 5 represents a typical grayscale image retrieved by the camera. The detection process for each scan line can be described as follows:

1) Binarize the signal into black and white regions;
2) Generate barcode hypotheses;
3) Obtain the embedded codes from each hypothesis;

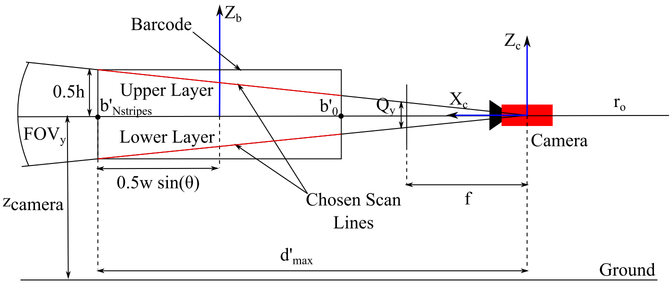


Fig. 4. Side view of the detection problem, from a plane orthogonal to the ground, parallel to the camera's optical axis and passing through the barcode's center.

4) Extract the most likely camera observation pose in the barcode frame, for each hypothesis.

The detections on both scan lines, whose horizontal positions differ by less than a few pixels, are merged together to form a complete barcode. Next, the embedded code is extracted and the binary sequence obtained, inversely applying the method used for encoding. In this case, the NRZ method is used. A Cyclic Redundancy Check (CRC) sequence in the end of the code is used to prevent bit detection errors.

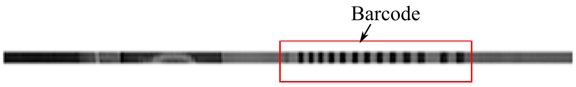


Fig. 5. Typical barcode structure on a captured image.

The binarization process is performed using a threshold computed from the mean of the maximum and minimum pixel intensities of the scan line. To filter regions which are too close to the computed threshold, a distance between that threshold and the intensity of each region is used.

Barcode hypotheses are formed by extracting ROIs from the binarized signal. Black regions around a seed point, consisting in the smallest black region, are connected until the predicted barcode size, computed from the width of the seed and the barcode specifications (N_{stripes} and quiet zones), is reached. Several ROIs can be extracted by repeating the process to the remaining regions of the scan line. The method has the problem of not considering the geometric distortions for the ROI length computation, which requires the barcodes to be a certain distance apart from each other, otherwise more than one barcode can be caught inside the same ROI. Since the walls of the scenario are assumed white, an extracted ROI must be related to a barcode.

Also, for each hypothesis, a perspective transformation, H, translating points in the camera plane, Δ, to points in the barcode surface, b, is computed, using the transitions of the hypothesis's guard sets, present in each ROI edge. The homography concept is used to define the transformation, since it provides a linear system, resulting in fast estimations. The former points in homogeneous coordinates are respectively, $\Delta^h = \begin{bmatrix} \Delta & 1 \end{bmatrix}^T$ and $b^h = \begin{bmatrix} b & 1 \end{bmatrix}^T$.

From Fig. 3, one can conclude the following:

$$A = \begin{bmatrix} \alpha_0^T \\ \alpha_1^T \end{bmatrix}^T, \quad B = \begin{bmatrix} \beta_0^T \\ \beta_1^T \end{bmatrix}^T, \quad \Delta^h = \lambda A^{-1} B b^h \tag{7}$$

where α_0 and α_1, β_0 and β_1, are the vectors describing the position and orientation of respectively, the camera plane and the barcode frame, in respect to the camera's focal point, which defines the camera frame. α_1 and β_1 are defined as unit vectors. Thus, A and B are 2x2 matrices. $A^{-1}B$ is encapsulated in a 2x2 homography matrix H.

The embedded code for each hypothesis is obtained by sampling the respective ROI according to a grid of stripe middle points, which is built using the estimated perspective transformation, as shown in Fig. 6. The sampling is done sequentially from the borders to the center of the ROI while enhancing the perspective transformation, H, with each new transition found.

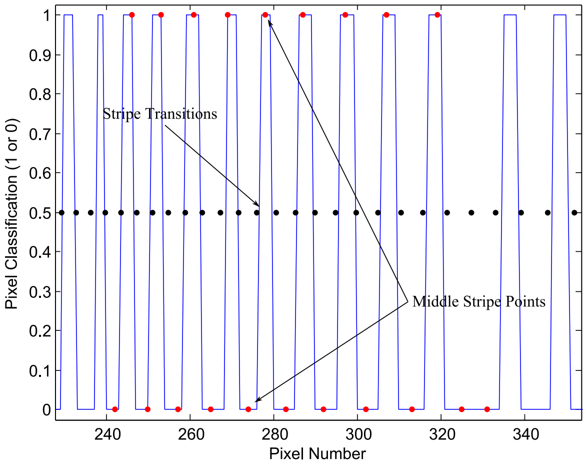


Fig. 6. Projected middle stripe points and barcode transitions, predicted from the estimated transformation during the detection process.

The camera pose in the barcode frame, $(x_b^c, y_b^c, \theta_b^c)$, is obtained from a simple homography decomposition. Observing Fig. 3, one can easily build matrix A, in (7), using the camera parameters. From A and the estimated H, B can be computed apart from a scale factor, necessary to make β_1 a unit vector. From Fig. 3, one can see that β_0 is directly the barcode position in the camera frame, and β_1 defines its orientation in the same frame, which is of opposite value of the camera orientation in the barcode frame, θ_b^c. The expression for that angle is as follows:

$$-\theta_b^c = \pi + \operatorname{sgn}\left(\frac{\beta_{1_x}}{\beta_{1_y}}\right) \arccos\left(\frac{1}{\sqrt{1+\left(\frac{\beta_{1_x}}{\beta_{1_y}}\right)^2}}\right) \tag{8}$$

The camera position in the barcode frame is obtained by applying the inverse of the barcode axis transformation matrix to the origin of the camera frame, as follows:

$$\begin{bmatrix} x_b^c \\ y_b^c \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_b^c) & \sin(\theta_b^c) & \beta_{0_x} \\ -\sin(\theta_b^c) & \cos(\theta_b^c) & \beta_{0_y} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{9}$$

3. Robot Localization System

3.1. Observation Model

The robot pose in the environment, (x_k, y_k, θ_k), is estimated onboard the robot. Periodic movement information, obtained directly from the stepper motor pulses, is used in the predict of the EKF algorithm, applying a linear motion model. Barcode measurements, acquired by a barcode sensor, from applying the above described method to the captured images, are used in the update of the EKF algorithm.

3.2. Observation Model

Observing Fig. 7, (x_k, y_k, θ_k) can be obtained from the robot pose in the barcode frame, $(x_b^r, y_b^r, \theta_b^r)$ and the barcode absolute pose in the environment, $(x_w^b, y_w^b, \theta_w^b)$, using the following transformation:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \cos(\theta_w^b) & -\sin(\theta_w^b) & 0 \\ \sin(\theta_w^b) & \cos(\theta_w^b) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_b^r \\ y_b^r \\ \theta_b^r \end{bmatrix} + \begin{bmatrix} x_w^b \\ y_w^b \\ \theta_w^b \end{bmatrix} \tag{10}$$

The $(x_w^b, y_w^b, \theta_w^b)$ can be directly decoded from the embedded code, or obtained from a lookup table using the embedded code has the landmark identifier. The $(x_b^r, y_b^r, \theta_b^r)$ is obtained relating $(x_b^c, y_b^c, \theta_b^c)$ computed for the respective barcode detection, with the camera's pose in the robot frame. Expression (10) is used as the observation model. An empirical model for the noise associated to the extraction of $(x_b^c, y_b^c, \theta_b^c)$, is also added. This model is discussed in the result section.

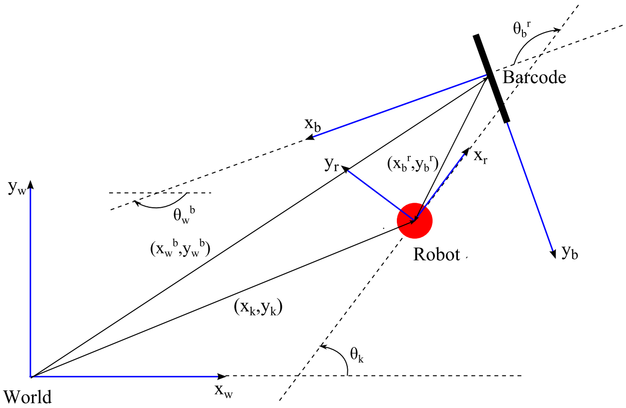


Fig. 7. Relations between world frame, barcode frame and robot frame.

3.3. Onboard implementation

The robot pose in the environment, (x_k, y_k, θ_k), is estimated onboard the robot. Movement information, obtained directly from the stepper motor pulses, is used in the predict of the EKF algorithm, applying a linear motion model. Barcode measurements are acquired from applying the above described method to the captured images, and are used in the update of the EKF algorithm. The localization system is divided into three main phases: 1) barcode detection for every captured image, which is always running, 2) periodic extraction of movement information relative the last extraction, assigning a time stamp referring to the extraction time, and 3) pose estimation, applying EKF predict and update steps with the odometry information and the barcode observations obtained during the last image processing, ordered according their time stamps. The barcode observations have their time stamps set to the middle of their image processing period. Fig. 8 shows the time diagram for the three phases.

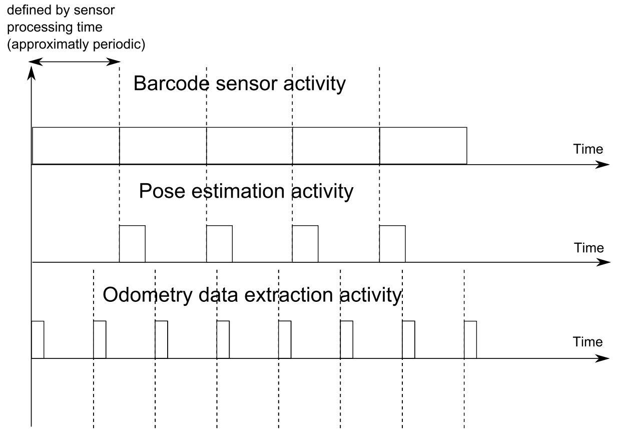


Fig. 8. Time diagram for the major processing phases of the localization system.

4. Experimental Results

The accuracy of the barcode design framework and the performance of the barcode detection algorithm were evaluated based on a dataset with 30 barcode visualizations for each distance, d, and inclination, θ, pair, with d from 10 cm to 60 cm in 5 cm steps, and θ from 0^0 to 60^0 in 15^0 steps. The barcodes used have 12 stripes in each layer for the embedded code, 2 bar-space pairs for each guard set, and each stripe have 1.5 mm of width and 1.25 cm of height. The embedded code reserves 24 bits to store $(x_w^b, y_w^b, \theta_w^b)$ and the last 8 bits to store the CRC error detection bits. The camera used has a focal length of 80 vertical pixels and 640 horizontal pixels, and the image frame has 480 pixels width and 8 pixels height.

Two cases were here considered. The first, shown in Fig. 9 a), consists in frontal barcode detections ($\theta = 0^0$), to test the robustness of the detection range to barcode variations. Four random barcodes were used and for all of them, the respective detection rates drop sharply at 50 cm, which is the maximum distance derived from the scan line inclination, computable from expression (6) for $\theta = 0^0$. Only 3 false detections in the 1320 detections were found. The second case, shown in Fig. 9 b), consists in barcode detections for the several θ values, using just one random barcode. The predicted detection ranges, computable from the barcode design framework (equation (5)) and represented in Fig. 9 b), were matched with the experimental ranges. One can observe that, for the first two θ values, the detection ranges are below their predicted values. This happens since the respective ranges are higher than the maximum distance derived by the scan line inclinations, which is about 50 cm. For the other θ values, predicted ranges are consistent with the results. An $L_{\min}$ of 1.5 pixels, instead of the early discussed 1 pixel, was used since it better explains the results.

For the latter case, the accuracy for the estimation of the relative pose of the camera in the barcode frame was analysed by performing a statistical analysis on the computed relative pose estimates from all the experiments, shown in Fig. 10. From this data, an empirical model, based on the 3x3 covariance matrix representing the noise when extracting $(x_b^c, y_b^c, \theta_b^c)$, can be built. The coordinates (x_b^c, y_b^c) are assumed independent from θ_b^c. The data shows that the uncertainty in (x_b^c, y_b^c) has an exponential behavior with the distance to the barcode, d, and follows the inclination to the barcode, θ. The empirical regressions, using mean squares, for both eigenvalues and the rotation angle, for the 2x2 matrix representing that uncertainty, are as follows:

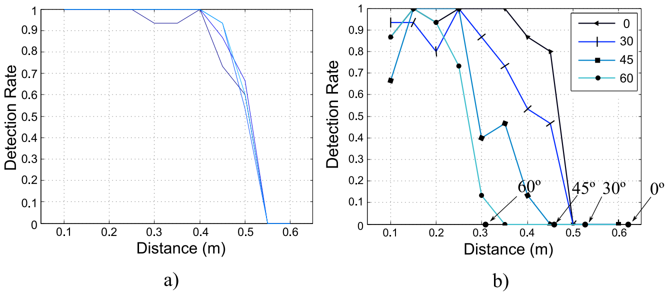


Fig. 9. Barcode detection rate with distance: a) Frontal perspective situation using 4 random barcodes. b) Several inclination situations for a random barcode, with the theoretical maximum observation distances (black dots) for each considered inclination.

$$\lambda_1(d) = e^{25.70d-15.46} \quad \lambda_2(d) = e^{16.15d-17.27} \tag{11}$$

$$\text{rotation}(\theta) = 1.910\theta - 1.43\,\text{sgn}(\theta) \tag{12}$$

Where θ can be replaced with $\theta_b^c + \pi$, and d with $\sqrt{(x_b^c)^2 + (x_b^c)^2}$.

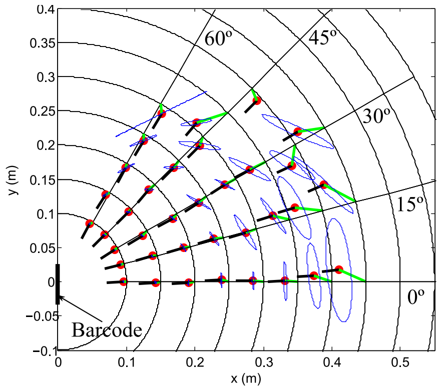


Fig. 10. Statistics for the extraction of the camera's pose in barcode frame, for a random barcode.

The localization system was evaluated using the scenario defined by Fig. 11. Using a joystick, the robot performs several laps around the scenario. The pose was estimated in real time and transmitted to an external computer. Odometry measurements were saved between periods of 100 ms and the barcode sensor took about 400 ms to process each image for barcodes (300 ms for image capture and 100 ms for image processing). A marker is placed on top of the robot, solidary with the robot frame, and is tracked by an external camera, using the ARToolKit Toolbox[1], to provide a groundtruth for the robot pose in the environment. A calibration to transform the 3D poses, provided by the ARToolKit, to 2D poses, for comparisons with the groundtruth, was also performed. 11 shows the results of the algorithm in comparison with odometry alone, for the whole path of an experiment, in which the robot is supposed to follow a triangular route. It is observed that while the algorithm estimations maintain the robot localized in a triangular route, the odometry based estimator progressively diverges from the route.

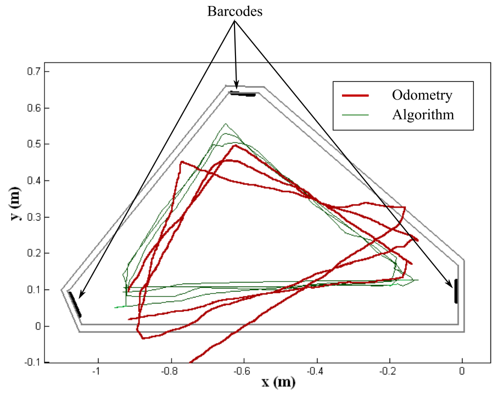


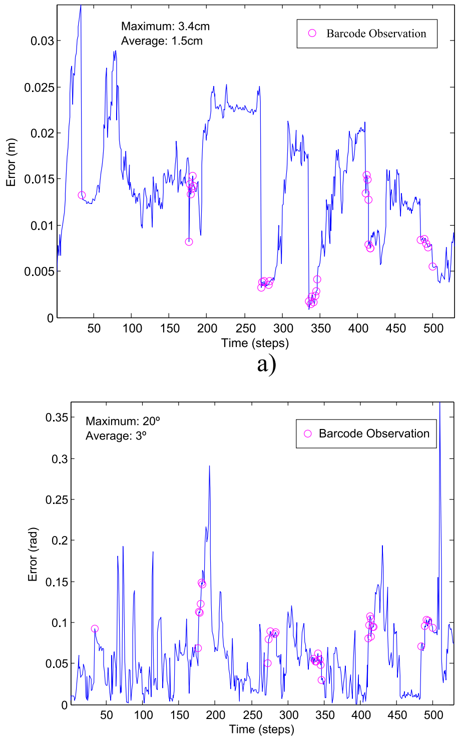
Fig. 11. Path estimation with the developed localization system (green/lighter) against the odometry alone (red/darker), for an experiment in which a triangular route is performed.

Figs. 12 and 13 show the Euclidean distance and the orientation errors between the algorithm's and the groundtruth pose estimations, throughout the whole path of an experiment. The last figure corresponds to a kidnapping situation. The marked points correspond to instants where the algorithm used barcode information for pose update. In normal conditions (Fig. 12) the localization error is shown to be smaller than the robot radius. In the kidnapping situation, the results show that the algorithm can recover from wrong estimations provided very few barcode observations (in this case two to four). One can also observe that there is normally a high error decrease when pose correction events occur due to barcode observations, which shows the high accuracy of the barcode related measurements.

5. Conclusions

In this paper we described a localization system onboard low capability robots (8 KB memory and 16 MIPs processor). The proposed system was implemented in the e-puck, having shown to converge with an error of about just 1.5 cm in average, for the barcode, robot and scenario specifications used in the experiments. Further experiments are foreseen using an autonomous navigation controller on the robot, relying on the localization results provided by the developed system.

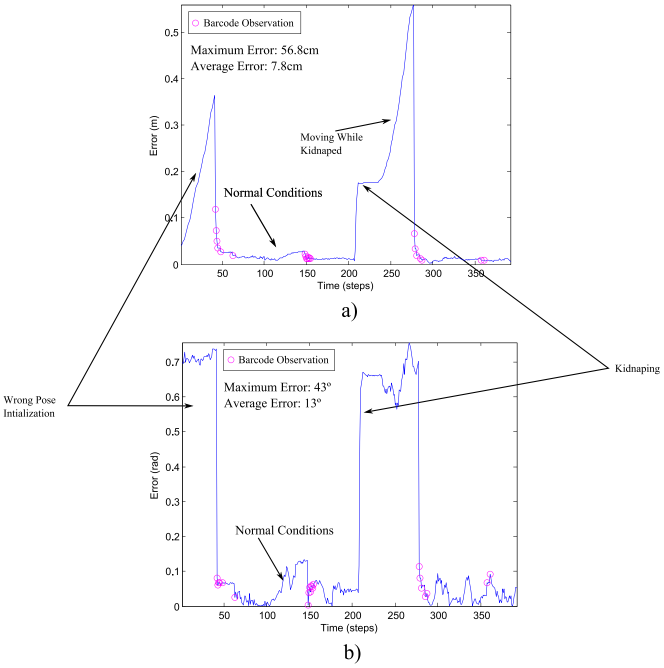
Future developments will target common situations where only a portion of the barcode is inside the image, which prevents barcode observations and can possibly lead the robot off the navigation track. Also, a study of the optimal positioning of the barcodes in the environment will be considered, using the developed mathematical framework. A new scenario will be used to translate an example of a real case scenario where this localization system should be deployed.



a)

b)

Fig. 12. Estimation error quantification of the algorithm along the robot's path. a) Position error. b) Orientation error.



a)

b)

Fig. 13. Estimation error quantification for a kidnapping situation. The robot starts with a wrong pose estimate, and in the middle of the experiment a kidnap is also applied. a) Position error. b) Orientation error.

Notes

[1] URL: http://www.hitl.washington.edu/artoolkit/ (Retrieved on 1 January 2013).

AUTHORS

Duarte Dias* – Institute for Systems and Robotics, Instituto Superior Técnico, TULisbon, Portugal, e-mail: duarte.dias@ist.utl.pt

Rodrigo Ventura – Institute for Systems and Robotics, Instituto Superior Técnico, TULisbon, Portugal, e-mail: rodrigo.ventura@isr.ist.utl.pt

*Corresponding author

Acknowledgements

This work was supported by the FCT project [PEst-OE/EEI/LA0009/2011].

References

[1] J. N. Pereira, A. Christensen, P. Silva, and P. Lima, "Coordination through institutional roles in robot collectives". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.)*, May 2010, pp. 1507–1508.

[2] L. Bayindir and E. Sahin, "A review of studies in swarm robotics", *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 15, no. 2, 2007, pp. 115–147.

[3] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering". In: *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, May 2009, pp. 59–65.

[4] M. Nasser, "On E-Puck Mobile Robots for Distributed Robotics", ser. National University of Singapore.

[5] S. Slusny, R. Neruda, and P. Vidnerová, "Localization with a low-cost robot". In: *Proceedings of the Conference on Theory and Practice on Information Technologies. CEUR Workshop Proceedings*, September 2009, pp. 77–80.

[6] S. Grzonka, W. Burgard, C. Stachniss, and G. Grisetti, "Introduction to mobile robotics: Localization with a low cost robot", 2008, University of Freiburg.

[7] R. Negenborn, "Robot Localization and Kalman Filters: On finding your position in a noisy world", ser. Utrecht University, Utrecht, Netherlands, September 2003.

[8] A. Burian, M. Vehvilainen, J. Kangas, "Camera barcode reader with automatic localization, detection of orientation and type classification," *Computers and Simulation in Modern Science*, vol. 1, 2008, pp. 214–219.

[9] D. Chai, F. Hock, "Locating and decoding EAN-13 barcodes from images captured by digital cameras". In: *5th Inernational Conference on Information, Communications and Signal Processing*, December 2005, pp. 1595–1599.

[10] J. C. Rocholl, S. Klenk, and G. Heidemann, "Robust 1D barcode recognition on mobile devices". In: *20th Inernational Conference on Pattern Recognition*, August 2010, pp. 2712–2715.

[11] L. Fang and C. Xie, "1-D barcode localization in complex background". In: *2nd International Conference on Computational Intelligence and Software Engineering*, December 2010.

[12] K. Wang, Y. Zou, and H. Wang, "1D bar code reading on camera phones", *International Journal of Image and Graphics*, vol. 7, no. 3, 2007, pp. 529–550.

[13] S. Wachenfeld, S. Terlunen, X. Jiang, "Robust recognition of 1-D barcodes using camera phones". In: *19th Inernational Conference on Pattern Recognition*, December 2008.