Mobile robot localisation for indoor environments based on ceiling pattern recognition

Francisco Dias, Hanna Schäfer, Leonardo Natal, Carlos Cardeira
IDMEC/LAETA, Instituto Superior Tecnico, Universidade de Lisboa
1049-001 Lisboa, Portugal
francisco.m.c.dias@tecnico.ulisboa.pt, hanna.schaefer@tum.de, leonardo.natal@usp.br, carlos.cardeira@tecnico.ulisboa.pt

***Abstract*—In this paper a multi-modal localisation system, that estimates a robot position in indoor environments using only on-board sensors, namely a webcam and a compass, is presented. Ceiling lights are used as beacons. Their position is previously known or self-learned during normal operation. Markov Localisation (ML) is both simulated and experimentally validated. For the prediction step it combines IMU (Inertial Measurement Unit) data and image parameters to compute the attitude of the robot. The update step is then calculated by measuring the distance to possibly visible ceiling lights. The experimental validation of the proposed solution shows that the robot position estimate converges to its real position and the error is kept within decimetres of magnitude.**

I. Introduction

Mobile robots are a widespread research area. Some industrial mobile robots, namely AGV (Automatic Guided Vehicles) with restricted movements are already being used in many industrial fields. A common problem is the realization of independent and autonomous movement strategies that do not rely on physical changes in the environment. This paper focuses a localization system environmentally independent. One approach that has been taken recently is to use the ceiling, which is usually immune to changes, as a reference and implement landmarks like QR codes, sensors and other reference points [1]. This might be complex and cost intensive. Other solutions from the field of outdoor localization like GPS (Global Positioning System) or AGPS (Assisted GPS) are generally too imprecise for the robot to move to specific points like a production line with the required level of precision. Wi-Fi localization systems [2] still lack of precision. The localization based on the local signature of the earth magnetic field [3] seems limited in an industrial environment due to motors and controllers magnetic noise. In addition, the factory environment is usually noisy in terms of electromagnetic interference.

Previous studies on indoor localization were made by J. Leonard and H. Durrant-Whyte [4] using static geometric beacons of known positions. Panzieri at al. [5] introduce an image-based localization for mobile robots that detects ceiling lamps. Rodrigues at al. made studies in warehouse environments using visual odometry of ceiling images [6], Markov Localization [7] and a 3D camera applied to a depth map of the ceiling. Carreira et al. [8] also use depth maps and introduce a technique to deal with missing data from the 3D camera. Both S. Kim & C. Park[9], C. Huang et al. [10] and [11] present some interesting studies based on localization with ceiling images.

The now classical approaches to deal with localization problems are mainly EKF based or Particle Filter based [12]. EKF based approaches are unimodal and may have issues on convergence and stability due to the linearization involved. Particle Filter based approaches [12] are multi-model but their performance is strongly related to the amount of particles and re-sampling techniques used. The approach shown in [8] is based on an LPV (Linear Parameter Varying) robot model that avoids linearization, proving globally asymptotic dynamics. However it is unimodal and requires a map of the environment.

The main contribution of this work is to present and experimentally validate a Markov Localization based multi-modal localization system. In addition the approach can be used without previous knowledge about the map, if the beacon positions are self-learned by the robot.

The rest of this paper is organized into five main sections. After this introduction (section I) follows the general problem statement (section II), where the details of the problem are stated and the main techniques and tools are presented. Section III presents a brief algorithmic proof of concept by applying the previous techniques to a well-behaved simulation scenario and comparing the results with the expected outcome. In section IV, the real world experiments are introduced. Their results are then discussed in section V. Finally, in section VI, the conclusion and further improvements are outlined.

II. Problem statement

Fig. 1. Ceiling with periodic patterns of lamps acting as landmarks.

The goal is to compute the pose of a mobile robot inside an indoor environment using a camera and an IMU device. The environment consists of a ceiling with a pattern of static landmarks whose positions are known a priori (this assumption will be released later). All landmarks are indistinguishable against each other and might additionally be distributed along the ceiling in a quasi-periodic pattern (Figure 1).
A camera is installed on the robot, which takes snapshots of the ceiling. The robot pose relative to the landmark is then calculated by the distance of the landmark to the center of the image and its angle relatively to the direction of the

978-1-4673-6991-6/15 $31.00 © 2015 IEEE
DOI 10.1109/ICARSC.2015.32

robot motion. Additionally an IMU device is used to give the absolute orientation of the robot inside the environment.
The Markov Localization (ML) algorithm is used to estimate the belief grid of the robot position inside the environment.

$$Bel(x_t) = Predict(x_{t'}, x_t, u_t)\ Correct(x_t, m_t) \quad (1)$$

The ML estimation of the belief $Bel(x_t)$ over x_t (vector of all robot position coordinates (x, y) at time t) iterates over two steps: prediction and correction (see equation 1). The initial believe is a uniform distribution. $x_{t'}$ denotes the previous positions at time t' and u_t the movement vector of the mobile used on prediction step. m_t are sensor measurements from the environment used on correction step.

$$Predict(x_{t'}, x_t, u_t) = \int P(x_t|u_t - x_{t'})\ Bel(x_{t'})\ dx_{t'} \quad (2)$$

The prediction step (equation 2) uses u_t (movement vector derived of control actions done by the robot at time t). With this information it shifts the previous belief $x_{t'}$ (vector of all robot position coordinates (x, y) at time t') along vector u_t. The movement of the robot is derived from the orientation measured by the IMU device and the speed of the robot. The belief estimates are not changed by this operation. They are merely redistributed to each position resulting of the movement.

$$Correct(x_t, m_t) = \eta\ P(m_t, x_t) \quad (3)$$

The correction step (equation 3) uses the measurements m_t (vector of robot sensor feedback at time t) from the sensors and compares them to all values inside the environment. According to the position, if the values are similar to the measurement, the belief of the robot being at that position x_t is increased.

III. Proof of concept

First the concept of ML was proved on an abstract approach. In this simplified program a single picture presents the whole ceiling, while the measurement is a single pixel representing the current position inside the ceiling. For the movement, it was used a simulated trajectory with a given speed and attitude for each step.
The measurement update is calculated with the expression in equation 4, which determines the belief of each pixel being the correct one. Measurements closer to the pixel value increase the belief of the robot being at that position:

$$f(x, y) = \frac{1}{1 + |mesvalue - pixvalue|} \quad (4)$$

The convergence of this approach is shown in figure 2 and shows the summation of all belief estimates over time. In the background of the map (blue tones) the structure of the original image can be recognized, since on the first step all pixels in the measured tones are equally probable. Additionally the green stripes indicate all the movements, which were still probable after a few steps, because of the redundancy of pixel values in some areas. Finally the yellow circular movement shows the converged path. All other trajectories stopped, once the yellow path was found.

After this successful convergence, robustness was tested, adding randomized drifts in the movement data, and noise on the measured gray value of the pixel. Although the convergence was still given, high noises would introduce errors on the converged path, which were then again corrected by

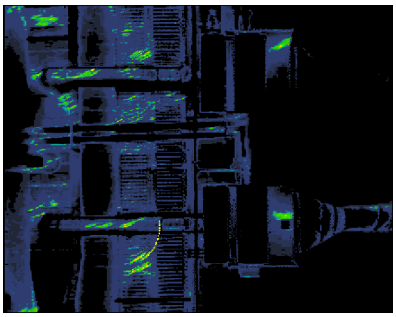

Fig. 2. Sum of probabilities during the convergence shows preliminary estimates and the final covered path. Blue and black areas have lower belief and green and yellow areas have higher belief.

convergence. Finally a kidnapping was added after a certain time to test, whether the robot would localize itself a kidnap to other position. In figure 3 we can see how the robot starts at A with a converged estimation. At B the robot is then kidnapped to position C. Right after kidnapping, the estimation follows the direction of the true path in a wrong position. After some measurements a new peak grows and after sufficient non-matching measurements, the estimate flips to the correct peak and follows the path flawlessly.

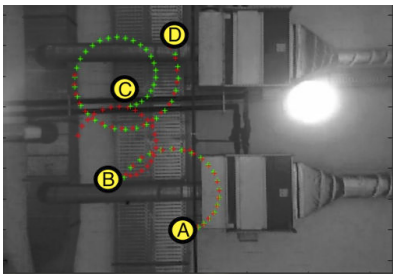

Fig. 3. Simulation of a kidnapping with ground truth positions in green and estimated positions in red. (A) Start of trajectory after convergence. (B) True path switches to a different position and orientation (C) after kidnapping. Estimated positions continue to be computed from the point of kidnapping. (D) End of trajectory with estimated position converging again.

This successful demonstration allows a transfer of the approach to further experiments on real scenarios.

IV. Experiments

The methods validated in the proof of concept (Section III) were applied to a real indoor localisation system using feature detection on the ceiling images. This section is split into 4 parts. The first part describes the equipment used in the experiments, including sensors, actuators and interfaces to the software tools. The second part defines the environment of the experiments with a special focus on the features that will be detected on this ceiling. The third part of this section describes the data preprocessing and the adaptions made in order to have a clean testing set. The last part describes the program steps and how the data is used in the software.

A. Equipment

Figure 4 presents the mobile robot that was used for the experiments. The robot is controlled by a Laptop software

interface (B) to the motion controller of the mobile platform (A). The movement of the mobile robot can be controlled manually using the computer keyboard or automatically following a black line on the floor detected by a webcam placed in the front of the platform.

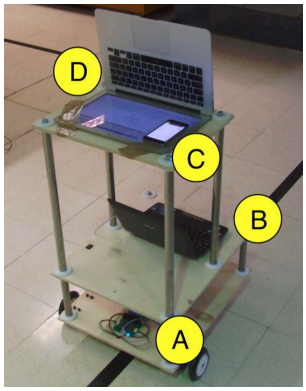

Fig. 4. Mobile robot used on experiments. (A) Mobile platform. (B) Laptop with software interface to controller. (C) Device with IMU sensors sending data to application (D) Laptop with webcam to capture ceiling images

In addition to the control components, the robot carries two types of sensors. The most important sensor is the camera installed on top of the robot. This camera records images of the ceiling as seen from the mobile robot during the movement. The camera's view was placed orthogonally to the floor and was configured for its widest angle in order to get the maximum field of view of the ceiling. Different cameras where used in previous experiments but were disregarded in contrast to better quality and wider field of view. The second sensor is an IMU device, aligned with the direction of movement of the mobile robot. During the experiment, it records the values of the magnetometer in order to determine the orientation angle of the robot. It then sends live data to the program, where it is used to do the localization.

B. Environment

The experimental environment used for the tests is a laboratory. Figure 5 presents a part of the ceiling of the environment, showing that this sort of ceiling offers several structures that can be used as landmarks. Some peculiar structures like the blue air conditioning, lamps, pipelines and electricity wires are visible in this image. Round light bulbs have been chosen as landmarks on these experiments because they are common in an indoor environment and they are also the easiest feature to detect from an image. Since the light bulbs can be turned off or fail, these features are not always reliable. In our tests it is assumed, that lights are working and always turned on. Future work would have to show, whether the robustness against kidnapping also extends to the failure of landmarks. Another part of the experiments that is environment dependent is the path that the robot will follow. Regarding the constraints and obstacles inside the laboratory, two distinct trajectories were chosen for the experiments (figure 6) in order to capture different situations of feature detection. In both images the trajectory is shown as a row of black crosses. The light bulbs are marked with white crosses and surrounded by a color code related to the distance to them. The top trajectory in figure 6 is the shorter path, meeting only four visible light

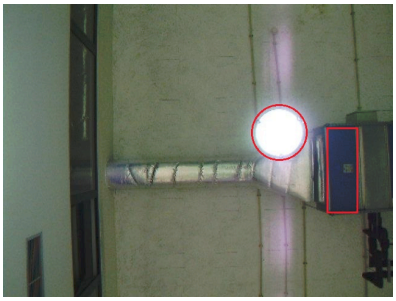

Fig. 5. Ceiling Patterns. Different patterns can be used depending on the ceiling.

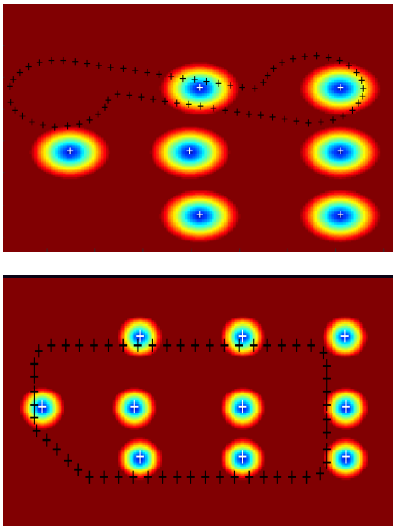

Fig. 6. Two different paths used in experiment. Trajectory as black crosses and landmarks as white crosses.

bulbs. To follow this trajectory, the robot tracks automatically a predefined line on the floor that is used for Ground Truth validation. The bottom trajectory in figure 6 has a much wider range and meets seven visible landmarks. The manual keyboard control was used to guide the robot on this trajectory.

As a precise ceiling configuration in not known, the approximate distance between landmarks was calculated using the shadow each light casts over the vertical plates on the floor and measuring the ground distance between them.

C. Data

The collected data consists of a sequence of image frames from the video stream and orientation angles from the magnetometer of the IMU device. This data was processed, synchronized and selected to adequately use it in the localization. The synchronization and selection of the data is only necessary during the test, because the localization process is run offline in an infinite loop and both streams are not synchronised. As the sample rate of the IMU data is higher than that from the video stream, a magnetometer sample is assigned to each image frame. To be able to seamlessly combine the data of two consecutive loops, the data needs to be restrained to a common start and end location. All these steps are irrelevant once the

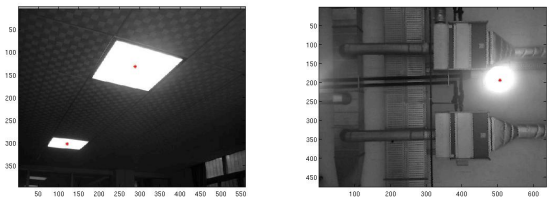

Fig. 7. Centroid of different lights. Coordinates of red marker give the position of the centroid in pixels.

system is working online. Each frame of the video stream is processed in order to retrieve the correct measurements of x, y distances to the nearest landmark and to correct the orientation of the robot. Three steps are needed to connect the image with a point in the ceiling: create a digital map of the ceiling with the distances to the nearest landmark for every possible position in the map, then measure the distance in pixels from the center of the image (supposedly the robot position) to the nearest landmark for each frame, and finally to convert that distance from pixels to meters.
The digital map is a matrix representing the environment in a discrete manner combining each position inside that environment to distance to the nearest landmark. An example of the map values is presented as the color range in the background of figure 6. The positions of the landmarks can either be given or they can be learned by the robot. Since the positions of the landmarks are given in decimeters, the map is generated as a grid of values with one-decimeter resolution.
In order to calculate the distance to landmarks from an image, a filter is needed to detect landmarks and their centers. Since the light bulbs are landmarks with bright shapes, a binary filter with high threshold value is enough for detection. Of this binary shape the centroid is calculated. Figure 7 presents some results of this approach over two images. As long as the form of the landmark is symmetrical, the centroid should have the same precision, independent on the size of version of the feature.

The resulting data from these steps is used as the input to the ML algorithm.

D. Software

The main steps of the process of localization can be seen in Algorithm 1. Without using any kidnapping or learning options, this program reflects the steps of a basic ML.

First, it reads data, then predicts the belief map for this step by shifting the previous belief map according to the movement. After that, a belief is calculated with the current measurement using the x, y distance values to the nearest landmark in the image. If there is no landmark in sight for the current position, the distance value is set to the maximum possible distance value. This penalty is used to reduce the belief of locations where the landmarks are not visible.

Finally, the belief according to this measurement is added to the overall belief and the loop begins again.

In addition to the basic approach, three features were added in order to improve the convergence. The first feature is the correction of the attitude given by the IMU sensor with the orientation of features available in the current image frame. Applying Hough transform over the current image and

while $frame \in stream$ **do**
  $frame \leftarrow stream$
  $imu \leftarrow process(IMUdata)$
  $\theta \leftarrow getangle(imu)$
  if feature was found in frame **then**
    $xy \leftarrow getcentroid(feature)$
    $xy' \leftarrow rotate(xy, \theta)$
    $learnedfeatures \leftarrow (feature)$
    $belief \leftarrow correct(map, xy')$
    $belief \leftarrow gaussfilter(belief)$
  else
    $belief \leftarrow correct(map, maxdist)$
  end if
  $prediction \leftarrow shift(belief, speed, \theta)$
  $prediction \leftarrow gaussfilter(prediction)$
  $belief \leftarrow prediction * belief$
  if $simulatekidnappinge$ **then**
    skip frames
  end if
  if $learningfeatures$**e then**
    recalculate map with new features
  end if
end while

Fig. 8. Structure of program

calculating the relative angle of the longest line, or the median of all the lines, it gives a good indicator of the orientation of the robot relative to the environment. This feature can be used wherever the ceiling presents perpendicular structures. Since this orientation is relative to each quadrant, it can be combined with the quadrant from the current IMU angle. The combination works according to the equation 5:

$$angle = \lfloor angle/(\pi/2) \rfloor * \pi/2 + relativeAngle(img) \quad (5)$$

The second feature is a smoothing filter applied over the belief map whenever no landmark is detected for more than 50 samples. This filter increases the divergence in uncertain areas and thus improves the convergence in case of kidnapping. When kidnapping occurs, the robot is carried to a different position without any record of its movement. It continues to follow the correct trajectory, but in a different position. In order to converge again it needs to change its belief according to the unfitting measures it is getting. Since at that point it has a high belief of its position in a wrong point, it takes a longer row of landmarks to shift that certainty to the correct position. If the belief is additionally decreased in every step without detection of a landmark, it will recover faster to a new estimate.

The last feature is the independent learning of landmarks. Since the estimation of the given landmarks position is not precise, it was decided to let the robot learn the position of the landmarks on his own. In order to do this, the robot stores the position of all detected landmarks during a given number of loops. Then, it clusters them in order to cancel out duplicates caused by noise in the position and builds a map of landmarks. The self-learning algorithm converges much slower the first time, since one full loop it needed to gather the landmarks. It then converges faster after a kidnap. The self-learned map only contains the landmarks detected and none of the invisible landmarks. This reduces symmetry and improves the convergence. In practice, this method could be used to

detect the landmarks in a first guided tour and later use them as given landmarks for localization.

V. Discussion of results

Different visualizations were used during the experiments to observe the convergence of the localization. The most important were the accumulated belief map for a state of the cycle and the summation for all belief maps, useful to show the history of convergence. In figure 9 the convergence of the belief map is shown. At the beginning of the localization (A) the robot detected only one landmark. Since it cannot decide which landmark it is, the algorithm increases the belief of all. After half a loop (B) some landmarks start to be excluded because they did not fit the movement. For example, the landmark on the left was shifted out of the map. After one loop (C) the lower beliefs are decreasing and the estimate appears more focused. Finally after 1.5 loops (D) the localisation is focused on a very small area of belief around the point of convergence.

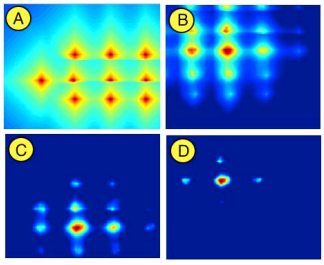

Fig. 9. Belief map of localisation; at (A) beginning; (B) after 0.5 a loops; (C) after 1 loop; (D) after 1.5 loops. Using offline localisation with processed IMU data.

The same behaviour can be observed in the sum of beliefs (figure 10), which shows us the history of convergence. At first (A) all landmarks moved along as the robot moves. In (B) only two estimates remain but one is preferred. Since it did not fit the measured landmarks, the assumed position converged to a unique path. Afterward (C) and (D) this path was further confirmed by all measurements and remains converged.

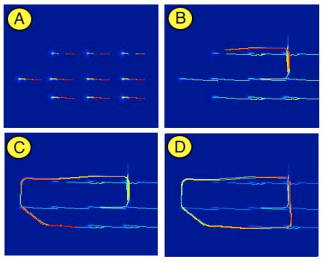

Fig. 10. Sum of belief maps of localisation; at (A) beginning; (B) after 0.5 a loops; (C) after 1 loop; (D) after 1.5 loops. Using offline localisation with processed IMU data.

For the evaluation of the quality of this convergence both paths shown in section IV were used and compared. The circular one is called short loop for distinguishing, and the squared, long loop. First we compare the map resulting for the self learned landmarks with the given map. Here we will see, that main differences are caused by the uncertainty of the true motion and the visibility of landmarks during the path. For visualizing the convergence, the transformation of the ellipsoidal uncertainty is shown in the following section. Both the short and the long loop are visualized after 2,3 and 4 loops. In figures 12 and 13 the estimated path is shown in red, while the true path is shown in black. The green ellipsoids represent the area of all probabilities higher than 99.9% of the maximum probability. Large ellipsoids show a higher uncertainty of the chosen estimate.

A. Self learning

A configuration without self-learning and another with self-learning were tested for each path. The self-learning is especially relevant in environments that are constantly changing. Using only one loop for learning, the map is already fully established. Multiple learning loops may help correcting the position of landmark estimates, but may also increase the uncertainty due to motion and measurement noise. In figure 11 both maps are shown for the same path. The missing landmarks are those in the middle who are not visible for the robot. Also the landmarks on the lowest row are shifted to the top because of measurement errors. Those errors however don't influence the overall convergence, since they will be the same in the next loop and thus match the landmark.

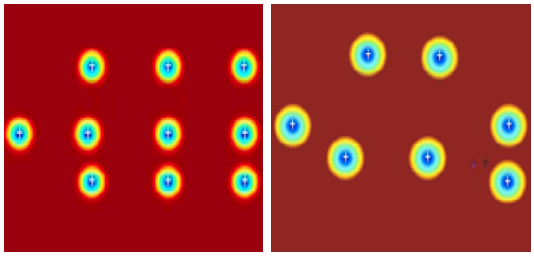

Fig. 11. Map of distances to landmarks with given true landmark position (left) and learned position of landmarks (right)

B. Short loop

Figure 12 shows how the estimate improves after multiple loops. The first loop was excluded, since it is still very cluttered from first belief guesses. In the pictures we can see how the uncertainty region marked as green ellipsoids decreases from top to bottom. Additionally we can see, that some points are still more uncertain, like the circle around the upper right landmark. Here the size of the ellipsoid stays almost the same and also the estimate still has some distance to the true path. Another interesting point is the shape of the ellipsoids, which indicates a small deviation orthogonal to the path, but a larger deviation along the path. This could signify problems with correcting the movement noise and good results with correcting the measurement noise.

C. Long loop

A similar behavior is shown in figure 13 on the long loop. The main difference is that the convergence is less visible in the ellipsoids. Although the estimate still improves, the uncertainty is already almost at its minimum in the first image. Another difference is that the deviation is equally distributed around its center. This has the reason, that the estimate is already further converged, but it also shows, that the dimensions are more equally covered with measurements than in the short loop.

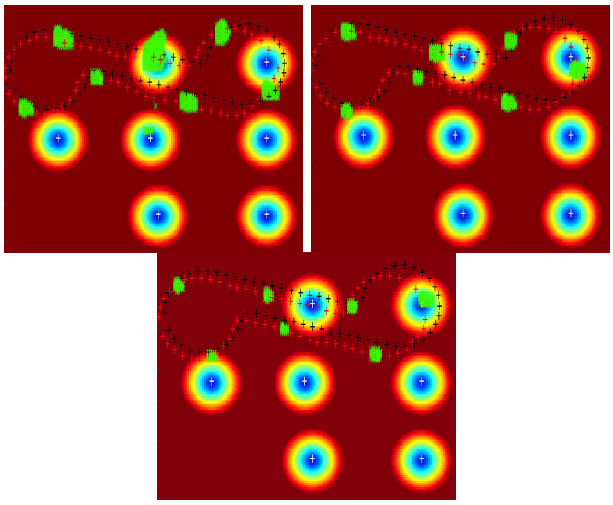

Fig. 12. Estimated path (red), true path (black) and estimate uncertainty (green) after 2, 3 and 4 loops on the short path.

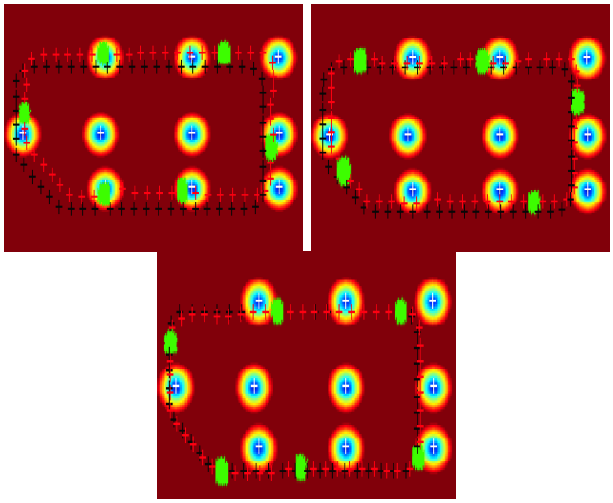

Fig. 13. Estimated path (red), true path (black) and estimate uncertainty (green) after 2, 3 and 4 loops on the long path.

VI. Conclusion

In this paper a visual localization system using ceiling lights as landmarks is presented and experimentally validated on an set of real conditions. Experiments were done with two different trajectories and with different configurations regarding the attitude read from IMU corrected with ceiling angles and the learning of landmarks. The experiments showed different configurations converging to the correct position. The error of the converged state can be estimated by the size of the uncertainty ellipsoids, which is 5-40 cm. Also a correction of noisy IMU data with angles given by the ceiling images proved to be very close to the true attitude of the robot. Depending on the configuration, the convergence may take longer or be more sensitive to kidnapping. The size of errors depends on the granularity of the ceiling map, the precision of the centroid detection and the noise on the ground-truth. Since this is a specific set of data, other real datasets might prove a different error range. In future work, this approach could to be tested against other environments. Also the influence of missing or broken landmarks could to be further investigated, since it might show a different behavior than a kidnapping. All these problems can be reduced with more precise measurements of the camera and the ceiling map as well as by higher computer performance, allowing finer granularity. They do not reflect on the quality of the approaches used. Additionally the approach was successfully used in an online application. Although it had larger errors after the convergence, the problems are due to irregular sampling steps. These final errors could be improved by better sensors for the orientation and additional sensors for the true speed and angular speed of the robot. At the same time, the speed measurement would give more opportunities to test the variance of the accuracy depending on the speed of the robot.

Acknowledgments

This work was partially supported by FCT, through IDMEC, under LAETA UID/EMS/50022/2013 and partially supported by the project PRODUTECH-PTI (Proj. 3904) under the program COMPETE/QREN/FEDER.

References

[1] G. Lin and X. Chen, "A robot indoor position and orientation method based on 2d barcode landmark," *Journal of Computers*, vol. 6, no. 6, pp. 1191–1197, 2011.

[2] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 4379–4384.

[3] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1028–1035, 2009.

[4] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," in *IEEE Transactions on Robotics and Automation*, IEEE, Ed., vol. 7, no. 3, June 1991, pp. 376–382.

[5] S. Panzieri, F. Pascucci, R. Setola, and G. Ulivi, "A low cost vision based localization system for mobile robots," *target*, vol. 4, pp. 1–6, June 2001.

[6] J. Rodrigues, C. Cardeira, F. Carreira, J. M. Calado, and P. Oliveira, "Experimental validation of a visual odometry system for indoor unstructured environments," in *16th International Conference on Advanced Robotics (ICAR)*, IEEE, Ed., 2013, pp. 1–6.

[7] J. Rodrigues, C. Cardeira, F. Carreira, J. Calado, and P. Oliveira, "A bayesian grid method pca-based for mobile robots localization in unstructured environments," in *Proceedings of ICAR 2013, the 16th International Conference on Advanced Robotics*, IEEE, Ed., Montevideo, November 2013, pp. 1 – 6.

[8] F. Carreira, J. Calado, C. Cardeira, and P. Oliveira, "Enhanced pca-based localization using depth maps with missing data," *Journal of Intelligent and Robotic Systems*, vol. 77, no. 2, pp. 341–360, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10846-013-0013-6

[9] W.-T. Huang, C.-L. Tsai, and H.-Y. Lin, "Mobile robot localization using ceiling landmarks and images captured from an rgb-d camera," in *International Conference on Advanced Intelligent Mechatronics*, IEEE, Ed., July 2012, pp. 11–14.

[10] S.-H. Kim and C. Park, "Localization of robot with ceiling-view cameras in indoor environment," in *International Conference on Mechanical, Automobile and Robotics Engineering (ICMAR'2012)*, December 2012, pp. 14–15.

[11] H. Wang, H. Yu, and L. Kong, "Ceiling light landmarks based localization and motion control for a mobile robot," *IEEE International Conference on Networking, Sensing and Control*, 2007.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.