Mobile Robot Navigation in a Corridor Using Visual Odometry

Bayramoglu, Enis; Andersen, Nils Axel; Poulsen, Niels Kjølstad; Andersen, Jens Christian; Ravn, Ole

Published in:
Prooceedings of the 14th International Conference on Advanced Robotics

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Mobile Robot Navigation in a Corridor Using Visual Odometry

Enis Bayramoğlu* , Nils Axel Andersen*, Niels Kjølstad Poulsen†,
Jens Christian Andersen* and Ole Ravn*

*Department of Electrical Engineering, Automation and Control Group,
Technical University of Denmark, Lyngby, Denmark
Emails: {eba, naa, jca, or}@elektro.dtu.dk
†Department of Informatics and Mathematical Modelling.
Technical University of Denmark, Lyngby, Denmark
Email: nkp@imm.dtu.dk

***Abstract*— Incorporation of computer vision into mobile robot localization is studied in this work. It includes the generation of localization information from raw images and its fusion with the odometric pose estimation. The technique is then implemented on a small mobile robot operating at a corridor environment. A new segmented Hough transform with an improved way of discretization is used for image line extraction. The vanishing point concept is then incorporated to classify lines as well as to estimate the orientation. A method involving the iterative elimination of the outliers is employed to find both the vanishing point and the camera position. The fusion between the vision based pose estimation and the odometry is achieved with an extended Kalman filter. A distance driven error model is used for the odometry while a simple error model with constant noise is assumed for the vision. An extended Kalman filter as a parameter estimator is also applied to estimate odometry parameters. Experimental results are included. The robustness and the precision of the entire system is illustrated by performing simple navigation tasks.**

I. INTRODUCTION

The field of robot vision is gaining prominence as the possibilities are explored. The importance of vision in humans as compared to all other senses pays credit to that. Many tasks performed by humans, today, require vision and their automation could be made possible as the computer vision field develops. The most significant advantage of vision is its capability to acquire information even in very complex environments, without interfering with the surroundings. This makes it a very flexible sense.

On the other hand, mobile robotics, still in its infancy, is one area where flexibility towards the environment is most desired. The major challenge of mobile robotics, as the name suggests, is to navigate the robot to where it needs to be. In addition to the obvious requirement of primary motion capabilities, such as turning and moving backwards and forwards, the robot has to sense its possibly dynamic environment, determine its own location and generate a motion plan accordingly.

The project described in this article aims to contribute to both fields by applying computer vision to perform one of the most important tasks of mobile robotics, localization. The scope of this project includes the generation of this information from the images as well as its fusion with wheel encoders. A small mobile robot operating in a corridor environment is then equipped with this pose estimator to perform simple navigation tasks as an indicator of overall performance.

The fields of computer vision and mobile robot localization have been studied extensively to date. The work of Kleeman[8] is a good example of mobile robot localization with multiple sensors, namely odometry and advanced sonar. The camera pose estimation, independent from robotics is also a very active research field. Yu et al. [16] used the trifocal tensor with point features to estimate the path of the camera from a sequence of images and Makadia [11] investigated the camera pose estimation restricted on a plane.

In the subject of mobile robot localization with vision, Andersen et al. [2] employed monocular vision to assist the laser scanner. Lin [9] used stereo vision and Munguia and Grau [12] studied monocular vision directly.

Previous work investigating problems similar to this project should also be noted; Tada [15] uses monocular vision in a corridor and incorporates the vanishing point to follow the center while Shi et al. [14] studies navigation in a corridor using lines but they are mainly interested in a safe region to travel instead of the pose and follow a very different strategy. Guerra et al. [6] obtain incremental measurements from lines in environments unknown apriori and they also carry out experiments in a corridor.

II. SOLUTION OUTLINE

In the assumed setup, the mobile robot has a single camera mounted on it without any ability to turn or move w.r.t the robot itself. The active wheels of the robot also have encoders available for dead reckoning. The robot localization is performed at a corridor, where the vision is used to estimate the robot orientation and its position across the width of the corridor. Dead reckoning, on the other hand, is used to keep track of both the orientation and the position across the width and the depth of the corridor. The corridor dimensions are assumed to be known apriori.

The choice of the corridor as the working environment has two important reasons. First, the corridor is a common part of most domestic environments and being able to navigate in

it has potential on its own. Second, it has a very regular and simple structure, making it easier to start the development of a more general vision based solution.

Dead reckoning is always applied to keep track of the pose estimation with a growing error. Therefore, for each raw image taken, a prior estimate is available to aid the visual estimation. Initially, the robot is either started from a known location or a high uncertainty in the pose is assumed.

Visual estimation is performed using image lines as features. Lines are extracted using a form of segmented Hough transform. These lines are then classified w.r.t direction using invariant environmental information and the prior estimate. The vanishing point corresponding to the direction along the corridor is also calculated robustly. The lines are then matched to each of the four lines lying along the corners of the corridors. Finally, the vanishing point is used to estimate the orientation while the line matches are used to estimate the translation.

When the visual estimate is available it is checked for consistency using the prior estimate and Bayesian hypothesis testing. If it passes the check, it is fused with the dead reckoning using an extended Kalman filter (EKF). The model for the dead reckoning is modified to allow for the estimation of its parameters along with the pose itself resulting in an extended Kalman filter as a parameter estimator (EKFPE).

The processing of each image spans a few sampling periods of the wheel encoders. During this time, dead reckoning is continued and all the encoder output is recorded at the same time. When the visual estimate is available, it is used to refine the estimate at the time of the taking of the image and the estimate for the current time is recalculated using the recorded wheel encoder output.

The next section describes the vision based part of the pose estimation including line extraction, line matching and pose estimation steps. Section IV is concerned with the fusion of the vision based estimation with the odometry. Section V presents the results of the work.

III. VISION

A. Line Extraction

An edge image is first created to be used in the line extraction. Canny edge detection algorithm is used for this purpose. The algorithm is first proposed and explained in [4]. It is only slightly modified by performing the non-maximal edge pixel suppression by comparison with vertical and horizontal neighbors only, that is, excluding the diagonal neighbors. This modification results in thinner lines, faster and more precise for line extraction.

Segmented Hough transform is then applied to the resulting edge image. The reader could refer to [1] for Hough transform used for general feature extraction. The standard Hough transform (SHT) used for line extraction is explained in [5]. [7] gives an overview of the variations of SHT. Segmented Hough transform is preferred in this work due to its higher speed and robustness.

The exact procedure is to first segment the edge image into 10x10 sub images. Hough transform is then applied to those images. This segmentation can be shown to speed up the overall Hough transform proportional to the square root of the number of segments. The performance is further increased by the extensive use of table lookups made possible by the small size of each sub image. The lines obtained from each sub image are then traced across the sub images.

The algorithm with all the performance optimizations resulted in 15-60 times speed increase compared to the OpenCV [1] implementation of SHT. The Hough transform is also modified to keep track of supporting pixels for each line through the use of look-up tables. Two line segments are combined if their combined supporting pixels describe a line precisely enough with a threshold. As well as being a robust criterion for line combination, this provides the endpoints of lines robustly and without any need for further processing.

B. Vanishing Point Detection

If a number of lines are parallel in the 3D scene, their projection on the image all meet at a single point. This point is the so called "vanishing point" specific to the direction of those lines. The vanishing point is a useful tool both for the detection of the 3D directions of image and for the calculation of the camera orientation w.r.t that direction.

The vanishing point is expected to sit on a point where many lines intersect with all others, if there are enough supporting lines. When all the intersection points between the image lines are calculated, a dense cluster is supposed to be formed around the vanishing point. Using the available prior estimate, it is also possible to calculate an estimate for the vanishing point along with an uncertainty.

Given a direction in the world coordinates described by the unit vector $\vec{v}$, this vector is first transformed to the image coordinates where it results in $\vec{v}^i$. Note that $\vec{v}^i$ will be a function of the camera orientation but not the camera translation. The coordinates of the vanishing point on the image will then be given by:

$$x_v = \frac{v_x^i}{v_z^i}, y_v = \frac{v_y^i}{v_z^i} \tag{1}$$

In this implementation, first the intersection points are calculated. They are then filtered using the vanishing point estimate obtained from (1). The actual vanishing point is calculated by iteratively removing the furthest point from the center of gravity of the remaining points. When the number of points is reduced to a certain threshold, the center of the remaining points is used as the vanishing point. Equation (1) is then used to calculate the camera orientation. This scheme has proven to be very robust. For an alternative method of vanishing point detection, the reader should refer to [3].

It is important to note that (1) provides two constraints whereas the complete camera orientation is described by three parameters. In the case of the mobile robot, the camera is constrained to turn around a single axis, therefore those two constraints suffice.

[1] An open computer vision library for C/C++, originally developed by Intel.

C. Pose Estimation

As the vanishing point contains information about only the orientation of the camera, image lines are also matched to the known 3D lines to obtain constraints about the camera translation. Each line contributes one constraint for the position along the width of the corridor.

Assume that an image line is described by the Cartesian line equation given in 2.

$$ax + by = c \tag{2}$$

Where a, b and c are calculated during line extraction. Then if the point $\vec{p}$ is a point on the line in 3D and the vector $\vec{v}$ is the direction of the line, the transformation of those vectors to the image coordinates ($\vec{p}^i$,$\vec{v}^i$) satisfies the following constraints:

$$\begin{aligned} av_x^i + bv_y^i &= cv_z^i \\ ap_x^i + bp_y^i &= cp_z^i \end{aligned} \tag{3}$$

Ideally, the first of those is simply satisfied by calculating the orientation using the estimated vanishing point. The second constraint is then enough to solve for a candidate value for the position.

In this work, the lines along the corridor corners are used. The lines passing through the vanishing point are classified to be along the corridor. In order to match those lines to a particular corner, image lines are investigated for their position with respect to the vanishing point. The robot is known to be inside the corridor, therefore those lines lying below the vanishing point are known to be on the floor and those lines lying to left of the vanishing point are known to be on the left wall and vice versa, assuming that the lines actually correspond to an actual corner.

The position constraint for each line after matching is solved to obtain a position candidate. The same iterative elimination of the furthest element is again used to arrive at a consistent set of position candidates, and the result is used as the visual estimate along with the previously calculated orientation.

D. Consistency Check

Under rare circumstances the vanishing point is detected over a spurious candidate cluster, or by coincidence an incorrect set of lines along the corridor constitutes a consistent set of corners. In these cases, the error of the visual estimate is much higher than what is expected of a regular estimate. Such a high error causes a bias at the EKF output for a long time due to the low assumed uncertainty. Instead of simply increasing the uncertainty in the vision error model, a two hypotheses error checking is used to reject such faulty estimates.

The first of those two hypotheses is that the measurement is a regular, accurate one. This hypothesis has the same error distribution as the error model used in fusion. The second hypothesis is that the measurement is a faulty measurement. In this case the error is assumed to be much larger. The second hypothesis is given a lower prior probability compared to the first, since such measurements occur rarely.

The probability of the first hypothesis being correct, given the actual measurement and the prior estimate, along with its uncertainty is calculated. If this probability is above 95% the estimate is accepted and used for fusion as described next.

IV. SENSOR FUSION

Extended Kalman filter is chosen for the fusion of the two sources of information available for localization. This choice requires an error model to be defined for both sources.

A. Error Models

A simple experimentally tuned and validated error model is defined for the visual information. According to this model, the orientation is measured with an independent additive Gaussian error having zero mean and $0.01rad$ standard deviation. The translation measurement is also assumed to have an independent additive Gaussian error with zero mean and $3cm$ standard deviation.

The error model chosen for the dead reckoning is described in [8]. This model could be summarized as follows; the dead reckoning equation is given in 4. In this equation θ is the orientation and x, y are the position estimates. l_r and l_l are the left and right wheel traveled distances while B is the effective separation between the wheels.

$$\begin{bmatrix} \theta(k+1) \\ x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} \theta(k) + \frac{l_r(k)-l_l(k)}{B} \\ x(k) + \frac{l_r(k)+l_l(k)}{2} cos(\theta(k) + \frac{l_r(k)-l_l(k)}{2B}) \\ y(k) + \frac{l_r(k)+l_l(k)}{2} sin(\theta(k) + \frac{l_r(k)-l_l(k)}{2B}) \end{bmatrix} \tag{4}$$

The sources of error in this estimate update equation are assumed to be due to uncertainty on l_l, l_r and B. The uncertainties are further assumed to be independent Gaussian with zero mean and variances proportional to the traveled distance. This model ensures that the resulting uncertainty for a path is independent of the number of samples taken during it.

B. Fusion Using EKF

The Kalman filter is an optimal state estimator for a finite linear system with the initial state estimates and error sources jointly Gaussian in nature. Extended Kalman filter is an extension of this filter to non-linear systems by linearizing the state space description locally. An in depth analysis of both could be found in [13].

The dead reckoning is used in the Kalman filter in place of the system state transition equation, although it is also a measurement. This approach is followed by many authors and it can be validated mathematically. The visual estimation enters the filter as a direct measurement of the orientation and the y coordinate.

It is important to note that the resulting filter is slightly different from a conventional EKF as the measurement is not applied at every state transition sample. Instead, it is applied whenever a visual estimate is available.

C. Modifying the System for EKFPE

An extended Kalman filter as a parameter estimator (EKFPE), as the name suggests, is a modification of the EKF so that it estimates some of the system parameters along with the states. This is achieved by augmenting the state vector with the parameters to be estimated. The state transition equation is also augmented with these new states that do not change over samples. An initial uncertainty is also assumed over these states to allow for estimation. EKFPE is analyzed in detail in [10].

The odometry equation in (4) used as the state transition equation is modified as follows:

$$\begin{bmatrix} \theta(k+1) \\ x(k+1) \\ y(k+1) \\ \hline k_r(k+1) \\ k_l(k+1) \\ B(k+1) \end{bmatrix} = \begin{bmatrix} \theta(k) + \frac{l_r(k)-l_l(k)}{B(k)} \\ x(k) + \frac{l_r(k)+l_l(k)}{2} cos(\theta(k) + \frac{l_r(k)-l_l(k)}{2B(k)}) \\ y(k) + \frac{l_r(k)+l_l(k)}{2} sin(\theta(k) + \frac{l_r(k)-l_l(k)}{2B(k)}) \\ \hline k_r(k) \\ k_l(k) \\ B(k) \end{bmatrix} \tag{5}$$

Here, the new variables k_r and k_l are the traveled distance per encoder tick for the left and right wheels. A very small source of error for these states in the state transition could also be added to allow for slowly varying parameters in the long run.

V. RESULTS

A. Visual Estimation

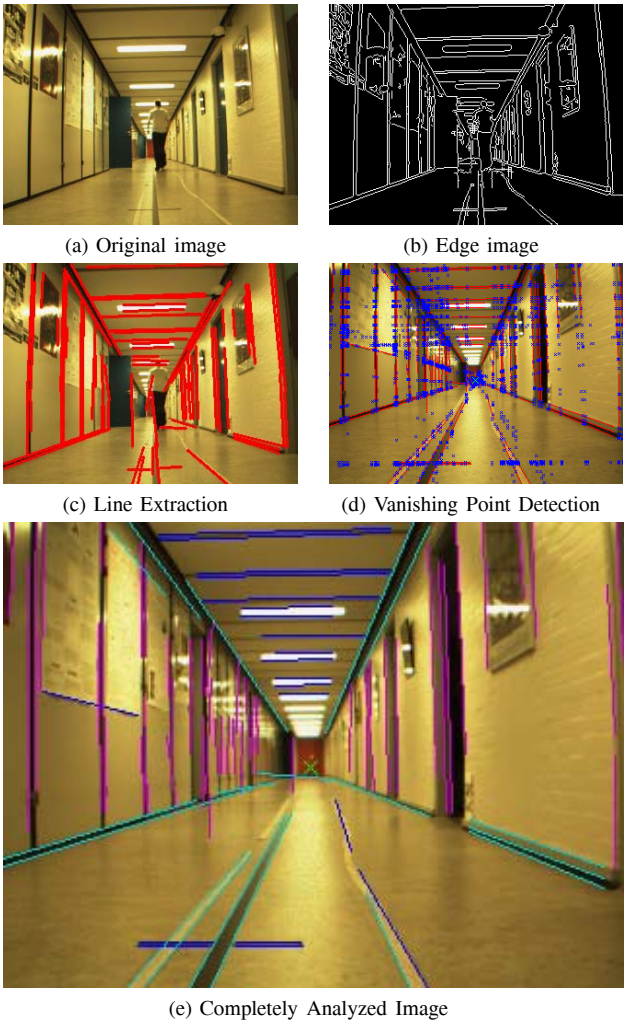
The steps of visual pose estimation are shown in Fig. 1. Fig. 1a shows the raw image taken by the mounted camera. The edge map obtained using Canny edge detector is displayed in Fig. 1b. The result of the line extraction is shown in Fig. 1c, where the extracted lines are drawn in red. Fig. 1d illustrates the calculation of the vanishing point. The tiny blue 'x' marks are where the extracted lines intersect. At the center of that image a dense cluster of those marks are visible. The larger green 'x' mark at the center of that cluster is the detected vanishing point.

Fig. 1e displays the end result of the processing. The cyan lines are the ones classified to be along the corridor.

The visual estimate for this case is found to be $20.8cm$ to the right of the corridor center. The actual position is measured with a ruler and found to be $22cm$ in the same direction.

The error in the orientation is much harder to measure since it is known to be very small as long as a wrong cluster is not found. Fortunately, this rarely happens due to the preprocessing applied before vanishing point detection. When it happens, the estimate fails the consistency check with the prior estimate and it is not used. In a regular image where the vanishing point is detected inside the right cluster, the deviation could be assumed to be below 2 pixels, which translates to roughly 0.3 degrees.

The robot platform consists of a CPU running with 500MHz to 1200MHz clock speed depending on the age of the robot. The entire visual estimation takes from $50ms$ to $100ms$ for completion, depending on the particular robot and the image.



(a) Original image (b) Edge image

(c) Line Extraction (d) Vanishing Point Detection

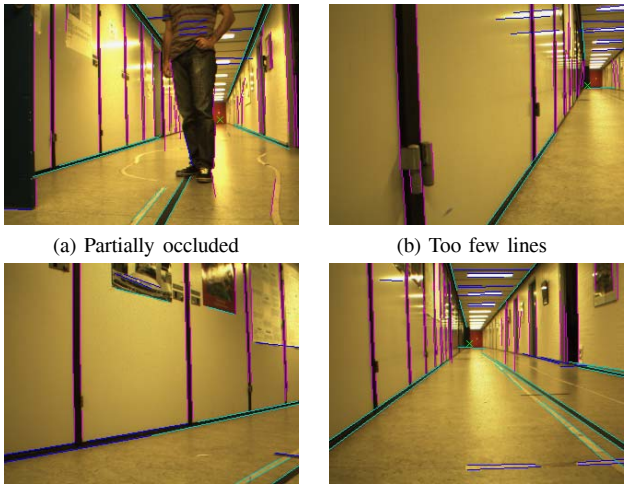
(e) Completely Analyzed Image

Fig. 1: Visualization of the steps taken for visual pose estimation

A few other example cases are shown in Fig. 2. They illustrate different cases possibly encountered during operation. The visual estimate works well in Figs. 2a, 2b and 2c with position errors of $2.5cm$, $0.2cm$ and $0.9cm$ respectively, despite different challenges. Fig. 2d is a rare example, where although the image is well suited for estimation and the vanishing point is correctly found, the error is as high as 11.8 cm. These rare cases are actually the reason why the error model for the visual estimate assumes $3cm$ standard deviation for the position error although the error is lower usually.

B. Simple Navigation Tasks

The performance of the entire system is evaluated by performing simple navigation tasks using the localization described in this article. The first task is to drive back and forth $10m$ for two complete turns. Fig. 3 contains relevant information for this task. Fig. 3a plots the visual estimates (green dots), the overall estimate (red line) and the actual



(a) Partially occluded (b) Too few lines

(c) Looking away (d) A case with high position error

Fig. 2: Various cases for visual pose estimation; these are intended to be interesting cases encountered during operation. The estimation errors for images a, b and c are all below $3cm$ despite various difficulties. On the other hand, the error for image d is $11.8cm$ although there are no disrupting factors, a rare occurrence.

ruler measurements (blue line) together. Fig. 3b displays the dead reckoning alone for the same task. Note how the actual error stays within $\pm 3cm$ while dead reckoning alone quickly loses track of the pose.

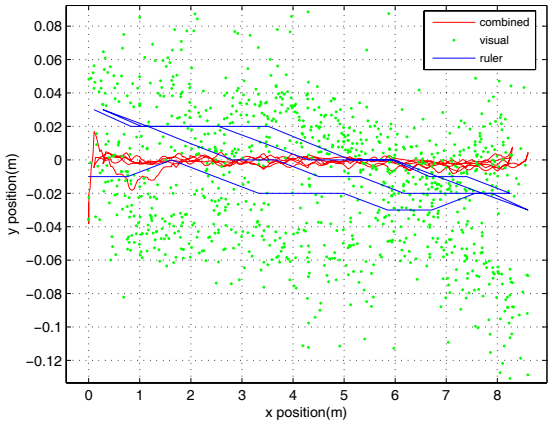
Fig. 4 displays the results of two more tasks. The task given in Fig. 4a is to move on a zigzag shaped path. The next task given in Fig. 4b is to drive straight, but during the execution of this task the camera is blinded along the path segment between $6m$ and $13m$. The black dots in the figure illustrate the samples where visual estimates are discarded automatically. Note how the robot diverges from its path while it is blinded and also how it quickly incorporates the images once the vision is back (watch the red line).

Finally, it is important to observe the evolution of the dead reckoning parameters. Fig. 5 is recorded during a long zig-zag type navigation task. The task spans 2500 visual estimates and the dead reckoning parameters are estimated at each sample. The figure shows the plot of estimated B, the effective wheel separation and the estimated R, which is the ratio of the traveled distances per encoder tick for the left and right wheels. In this experiment the initial values are deliberately set to wrong values (B to $0.22cm$ instead of $0.27cm$, R to 1.1 instead of 0.99). Observe how R converges very quickly while B takes longer.

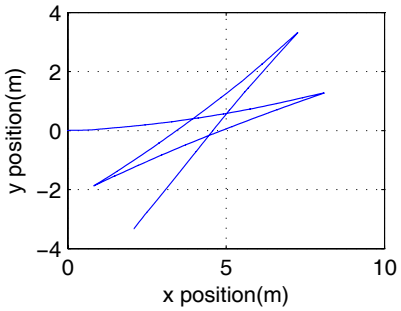
VI. CONCLUSION

A visual odometry method for mobile robot localization is presented here. The method relies on straight lines along the corridor, whose width and height are assumed to be known apriori. Image lines are then robustly matched to these to obtain pose constraints. The pose information is then fused with dead reckoning using an extended Kalman filter.

The potential behind robot vision is generally acknowledged due to both the low cost of commercially available



(a) Position data from various sources for the first experiment



(b) The path recorded by odometry only

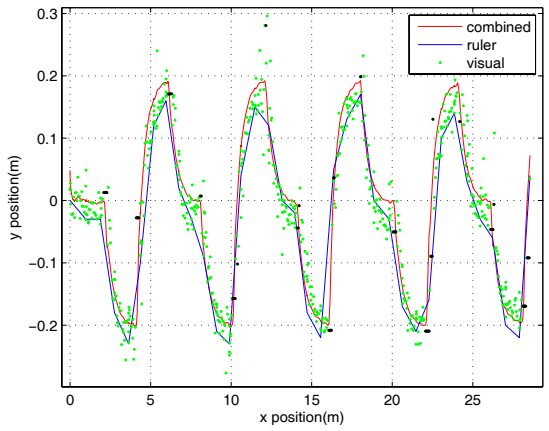
Fig. 3: Localization data for driving straight. Note how the odometry accumulates more than $1m$ error for a path segment shorter than $10m$. The fused estimate maintains less than $3cm$ error independent of distance travelled.

cameras, the flexibility of using vision and the high information content in images. This work involves the successful application of vision for robust and real time estimation of pose on a moderate system. Furthermore, the measured processing time allows the use of slower systems running with clock speeds down to roughly 200MHz.

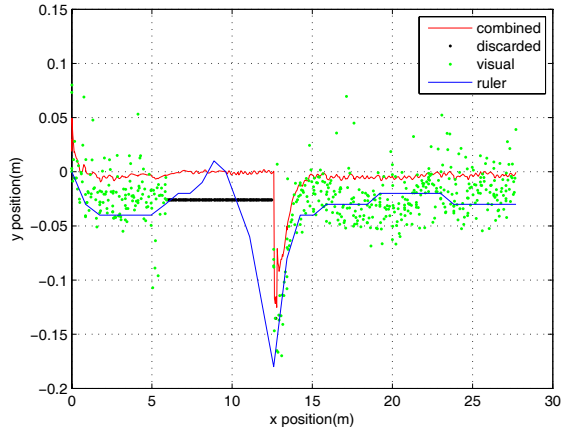
The line extraction step is the backbone of the method followed in this work, because the lines are the primary sources of information. This step is also the most time consuming one. Therefore a new fast line extraction algorithm is developed which is demonstrated to be robust in detecting even the shorter lines.

The vanishing point detection and line matching parts both employ the iterative furthest point removal. This ensures that they are highly tolerant to outliers. Furthermore, the iterative furthest point removal is implemented using a quad tree data structure so that the processing time scales well with the number of vanishing point candidates.

Visual estimation and dead reckoning both rely on the results of each other. However, for robustness, both of them are designed to be tolerant to more error than the other one generates. An overall demonstration of this is provided in the results section where the robot is blinded for as long as a $6m$ path segment. During this segment it manages to stay on its track within an acceptable error bound. When the blinding is removed it takes less than half a meter to relocate itself. On



(a) Position data for a zig-zag shaped drive



(b) Position data for the partially blinded drive

Fig. 4: Additional navigation tasks

the other hand, vision is also demonstrated to successfully compensate for the errors in the dead reckoning parameters as well as estimating them.

The accuracy of the system is also demonstrated through experiments. The measured position error is maintained below $3cm$ and the orientation error is estimated to be below 1 degree.

VII. FUTURE WORK

The developed method presents many opportunities in conjunction with other systems. Other sensors could be smoothly integrated through the Extended Kalman Filter. The method itself could also be inserted in other vision systems where vanishing points are occasionally available. This would provide the host system with an independent source of measurement. Although it is developed for a straight corridor, it could be used in more complex corridor environments provided that the corners and junctions are handled with auxiliary algorithms.

Ongoing research aims to extend the proposed solution to incorporate a generic set of lines that are not restricted to be parallel. The use of multiple cameras in conjunction is also investigated. The motivation behind the extension is to enable localization based on this method in less regular

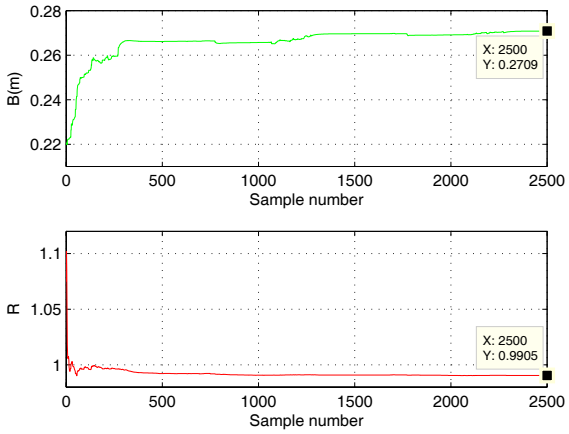


Fig. 5: The evolution of the odometry parameters

environments with available lines.

REFERENCES

[1] A.S. Aguado, M.E. Montiel, and M.S. Nixon. Arbitrary shape hough transform by invariant geometric features. *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 3:2661–2665, 1997.

[2] Jens Christian Andersen, Nils Axel Andersen, and Ole Ravn. Vision assisted laser scanner navigation for autonomous robots. In *Proceedings of the 10th International Symposium on Experimental Robotics 2006 (ISER '06) ; 39*, pages 111–120. Springer-Verlag Berlin, 2008. Presented at: 10th International Symposium on Experimental Robotics 2006 (ISER '06) : ISER '06, 2006.

[3] S.T. Barnard. Interpreting perspective images. *Artificial Intelligence*, 21:435–62, 1983.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–98, 1986.

[5] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[6] JJ Guerrero and C Sagues. Uncalibrated vision based on lines for robot navigation. *MECHATRONICS*, 11(6):759–777, SEP 2001.

[7] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[8] L. Kleeman. Advanced sonar and odometry error modeling for simultaneous localisation and map building. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:699–704, 2003.

[9] Huei-Yung Lin and Jen-Hung Lin. A visual positioning system for vehicle or mobile robot navigation. *IEICE Transactions on Information and Systems*, E89-D:2109–16, 2006.

[10] L. Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24:35–50, 1979.

[11] Ameesh Makadia, Dinkar Gupta, and Kostas Daniilidis. Planar egomotion without correspondences. *Proceedings - IEEE Workshop on Motion and Video Computing, MOTION 2005*, pages 160–165, 2007.

[12] R. Munguia and A. Grau. Monocular slam for visual odometry. *IEEE International Symposium on Intelligent Signal Processing*, pages 1–6, 2007.

[13] Maria Isabel Ribeiro. Kalman and extended kalman filters, 2004. `omni.isr.ist.utl.pt/~mir/pub/kalman.pdf` .

[14] Wenxia Shi and J. Samarabandu. Corridor line detection for vision based indoor robot navigation. *Canadian Conference on Electrical and Computer Engineering*, pages 1988–1991, 2006.

[15] N. Tada, T. Saitoh, and R. Konishi. Mobile robot navigation by center following using monocular vision. *SICE Annual Conference*, pages 331–335, 2007.

[16] Ying Kin Yu, Kin Hong Wong, M.M.Y. Chang, and Siu Hang Or. Recursive camera-motion estimation with the trifocal tensor. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetic*, 36, 2006.