Tracking the Absolute Position of a Mobile Robot Using Vision-Based Monte Carlo Localization

Jan Benda[1], Zbyněk Winkler[2]

Charles University, Faculty of Mathematics and Physics,
Department of Software Engineering,
Malostranské nám. 25, 118 00 Praha 1, Czech Republic

[1]jan.benda@matfyz.cz

[2]zbynek.winkler@mff.cuni.cz

Abstract

In this paper, we present a new technique for the localization of a mobile robot using panoramatic visual data only. This new method employs the composition of fast colour thresholding, look-up coordinate transformation, vision-based motion prediction and Monte Carlo Localization to gain robust and reliable pose tracking using a colour map of a delimited environment. Since our method uses visual data both to determine the relative motion and to verify the current location, it can cope with an unexpected motion such as wheel slippage or collision. We also propose a fast and robust technique to correct the estimated location using random pixels from the omnidirectional image.

1 Introduction

In this paper we present a reliable algorithm that enables robot's self-localization in a dynamic environment. We developed a variant of the Monte Carlo Localization (MCL) [3] that is able to track the position of a mobile robot using visual data from a panoramatic sensor only.

The most common implementations of probabilistic localization use the information from odometry to predict the current pose of the robot. However, there is a shortcoming in relying on the odometry data as the wheel slippage introduces an unmodeled noise into the system. Often, especially in robotic competitions, the robot moves in an unexpected direction because of collision with another moving object. Such motion, not considered in the prediction phase, results in the so called *kidnapped robot problem* [9]. Therefore we propose to use the image data for both the position and the weight updates. Compared to odometry-based approach our method reliably deals with any kind of motion within a limited displacement and rotation speed.

Our task is to navigate a robot in a robotic competition, where we can benefit of a highly structured environment with well distinguishable visual navigation marks on one side, but face a large amount of unexpected influences including numerous partial view occlusions and unexpected motion due to robot interaction. Reliable position information is a necessary condition for many high level tasks such as path planning or avoidance of known obstacles. Fast and robust self-localization is crucial when a robot has to be able to execute such tasks effectively.

2 Related Work

The weights of samples in a particle-filter-based robotic application are usually updated using data from laser or sonar range finders [3]. The drawback of such approach is that it requires surrounding walls and either a known metric map of the environment or an integrated mapping technique [5].

Vision-based approach is better suited for applications, where the surrounding environment is not known. First experiments combining MCL with visual data can be found in [2], where a light map of the ceiling is compared with information from a camera mounted in the upwards direction. The average brightness value of a small area in the middle of the image is used and compared with the map. It is obvious, that much more information can be harvested from a camera image.

Another application is to search for several geometric markings painted on the floor [10]. Such approach requires a geometrical description of the markings and a complicated evaluation of the match between the estimated pose and the current observations. The position of several selected marks is transformed into the image coordinates and then searched in the omnidirectional image.

Our method proposes the use of colour description of the floor to evaluate the estimated pose and enables to use any image pixel for the evaluation. A balance between reliability and processing speed may be reached altering the number of considered pixels.

3 Vision-Based Pose Tracking

MCL is a particle filter applied to the task of self-localization. Using this approach, the probability of a specific pose (consisting of position and orientation) of the robot on the operation field is modeled using a fixed count of weighted samples that contribute to the overall probability distribution.

The sample set representing the distribution function is updated in each step in two distinct ways:

- the positions of samples are updated using the estimated motion during the last step,
- the weights of samples are updated according to the visual data.

Each image is first *pre-processed* (sec. 3.1) to simplify extraction of traceable features defined by the colour of floor painting. Easily detectable features are located and compared for two subsequent images to determine the relative motion of the camera (sec. 3.2), which is used for the *prediction* step of the MCL. The final phase uses other visual features for *correction* of the probability distribution of the robot's pose (sec. 3.3).

3.1 Image Enhancement

As the first step, the input RGB image is normalized so that the mean and standard deviation of every image are constant. This reduces the influence of colour and intensity of the ambient light on the tint and brightness of the observed visual marks.

An implementation of [1] performs fast colour classification. The different colour classes are defined in the YCrCb colour space using one interval per class and channel to threshold the input image. Allowing a wide span in the brightness channel reduces the effect of object shape, orientation and position relative to the camera and light sources on the colour classification.

Finally, we process the classified image and search for the locations where different colours occur within a small area to determine the boundaries of visual features, which are later used in motion estimation.

3.2 Pose Change Estimation

It is obvious that the omnidirectional image usually contains all the information necessary to compute the precise location in a known delimited environment. Such approach would require extensive processing of the image data and would be very sensitive to image imperfections, such as partial occlusions or geometrical distortions caused by the tilt of the robot.

Instead, we pick several easily detectable features to estimate the current pose within some level of accuracy. Because of the periodicity of the basic pattern, an ambiguity in the motion estimation has to be solved imposing an upper bound on the robot's speed.

The omnidirectional image is first *rectified* to recover the shape of floor patterns and then the relative motion is estimated from the motion of *boundaries* among subsequent images.

Image Rectification

As the floor markings form a periodical orthogonal pattern, it is straightforward to begin the processing with projection of the camera image to the ground plane. This approach yields to an image where straight lines correspond to straight lines on the floor, which significantly simplifies the following process.

The mirror used in our application can be described as a part of a rotationally symmetrical surface described by a hyperbolic equation $z^2/a^2 - t^2/b^2 = 1$. So, when we compute projection of a light ray from a point on the *ground plane* to a point on the *camera sensor*, we can redefine the problem in polar coordinates, where only the radius may be considered because the optical axes of the mirror and the camera match and the angle remains unchanged. To make the computations even simpler, we assume the camera placement (see fig. 1) so that the first principal point (FPP) is aligned with one of the foci (F) of the hyperbola.

For a given radius ρ in the image coordinates, we have to find the intersection point of the corresponding ray with the hyperbolic curve $[t, z(t)]$. This is done by defining the curve function

$$z(t) = a\sqrt{1 + \frac{t^2}{b^2}}$$

and solving the equation

$$\frac{e + z(t)}{t} = \frac{f}{\rho}$$

for an unknown t. The solution of this equation is

$$t = \frac{2ab^2\rho f}{b^2 f^2 - a^2\rho^2}$$

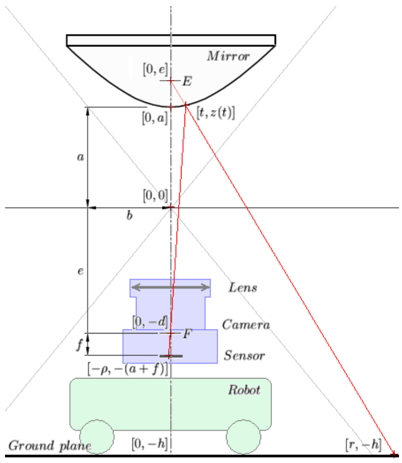
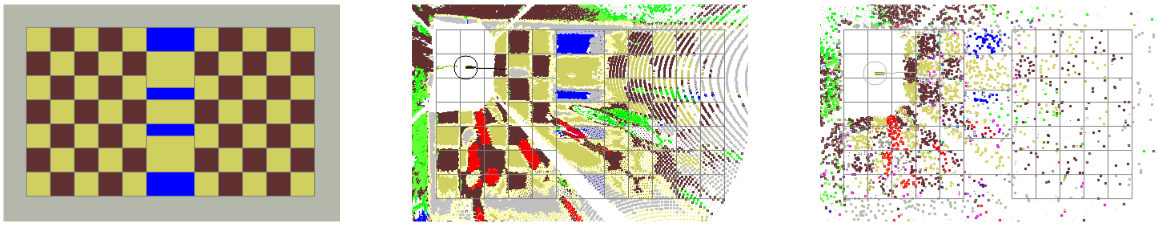


Figure 1: Schematic of the catadioptric image formation

Because the tangent in a given point of the hyperbolic curve halves the angle defined by this point and the two foci, all the rays originating at the camera's FPP (F) reflect as if they originated at the mirror focus (E). The intersection with ground plane ($[r, -h]$) is then computed using triangle similarity (the points $[0, e]$, $[t, z(t)]$ and $[r, -h]$ are collinear).

The transformation is pre-calculated and stored in a look-up table to reach high speed of execution.



(a) Reference map (b) All observations (c) Used observations

Figure 2: (a) the map of the playing field denoting the expected observations, compared to the (b) transformed and classified pixels from the omnidirectional image, (c) some of which are actually used in one iteration of MCL to evaluate individual samples.

Boundary Localization

After the image is rectified, the boundaries extracted in the enhancement phase now form several parallel lines in two perpendicular directions. To determine the horizontal shift of vertical lines in the rectified image, local maxima in the sum of the individual image scan-lines may be used. In the general case, there is always one of the boundary line directions projected to a line that differs from the vertical direction by an angle within $(-\frac{\pi}{2}, \frac{\pi}{2})$. Thus, if we skew and sum the image in a fixed count of steps, we may estimate the orientation of the operation field within $\frac{\pi}{2}$-periodicity: the local maxima reach highest values when the skew angle matches the orientation.

Analogical approach may be used for the perpendicular boundaries, whose orientation is close to the horizontal direction, but as we already know the skew angle, only one sum (in the appropriate direction) is performed.

The drawback of the boundary-based approach is that it only provides location within one checkerboard square. If all the boundaries were always detectable, counting the peaks on both sides of the robot would provide an absolute location. This unfortunately is not true, because several boundaries may be occluded by the robot itself, as well as other objects randomly placed on the playing field. An average of the peak offsets is used instead to estimate the robot's offset within the square-size period.

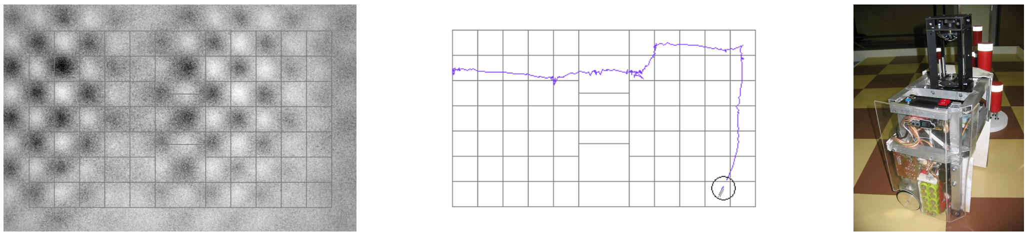
Motion Estimation

Returning to the Monte Carlo Localization, now we know how to update the samples to reach best possible match with the true current pose. Each sample is offset and turned using the difference of boundary location among two successive images. Due to the local nature of boundary extraction, a difference within half of the square size for position and $\pm\pi/4$ for orientation is identified. A random noise term is also added to deal with prediction errors.

3.3 Weight Update

In the final phase, the motion prediction is verified and the samples that best describe the true robot pose are highlighted. The weight of each sample is multiplied by an evaluation function, which reaches global maximum at the true robot pose.

To evaluate one sample, a fixed count of observations is performed, each of which takes a random point in the classified image and compares its value with the expected observation (fig. 2). As the selected point projects on the floor in a given direction and distance from the robot, the expected colour is the map entry at this distance from the pose of the sample. The weight of the sample increases if and only if the two values match. The divergence of the distribution decreases, as the samples far from the true pose get poor weights (fig. 3(a)).



(a) Evaluation function (b) Estimated trajectory (c) Robot Lojza

Figure 3: (a) the plot of the average evaluation for different positions using the items shown in 2(b) and (b) the estimated trajectory rendered using the visually tracked position; (c) our robot with the omnidirectional sensor mounted on top.

This new approach enables us to use virtually any pixel of the input image and decreases the influence of local misclassification as different observation items are selected for different samples. The evaluation is very fast, because all the items (i.e. pairs of source index and projected coordinates) are pre-calculated.

The important advantage of our omnidirectional sensor is that the image of whole operation field is available regardless to the robot's pose. Additionally, the evaluation function does not require a specific background beyond the border of the operation field – items out of the map are ignored.

4 Experiments

For the following experiments, we used our test robotic platform on the playing field for the Czech National Cup of Eurobot^{open} [4] — a rectangle of 2.1×3.64 meters consisting of two fields with a brown-beige 30×30cm checkerboard pattern separated by a blue ditch and surrounded by a white border. The robot has a catadioptric sensor with the mirror placed 45cm above ground and a field-of-view covering the whole playing field. The image is captured using an off-the-shelf web-camera connected by USB and providing thirty 640×480 uncompressed images per second.

We tested the algorithm on approximately 5 minutes of captured video from the robot's camera (about 3GB of uncompressed video streams), where we were applying all kinds of possible robot motions up to the speeds of $1ms^{-1}$ under several different lighting conditions. The data were later processed with our method using 400 samples to represent the probability distribution and 20 observations on every sample in the correction phase, reaching approximately 18fps on a 1.3GHz mobile computer. This imposes an upper bound of approximately $2.5ms^{-1}$ of forward and $4\pi s^{-1}$ of angular speeds to keep the ability to resolve the motion prediction ambiguity.

The method showed to be very reliable with respect to the lighting, occlusions, and image blurs and distortions caused by the robot's motion and tilt. The weakest point shows to be the pose change estimation that is sensible to the surroundings of the playing field. Large objects visually similar to the markings of the field tend to confuse the prediction — especially in corners, where the playing field influence is not dominant. The correction phase usually deals with such errors (see fig. 3(b)).

5 Conclusion

In this paper, we described an application of the Monte Carlo Localization algorithm on the data acquired using a web-camera, performing real-time pose tracking on a general hardware.

Since the large field of view is important to avoid lack of traceable features in the camera image, a solution using catadioptric omnidirectional visual sensor is presented. The method is best suited for localization in a highly dynamic environment of robotic competitions taking place on delimited operation field painted with an easily detectable colour pattern.

Compared to the most common approach, which uses odometry for motion estimation, our method reliably deals with any kind of motion within a limited displacement and rotation speed. This is especially important in situations, where many moving objects may influence the robot's intended trajectory.

Further work will focus on merging the odometry information with the vision-based motion prediction to remove the limitation on the speed of intended motion. Future enhancements may consider a redefinition of the motion estimation to be less sensitive to the surrounding of the operation field, either using the current pose estimation to clip the outside of the field or applying a more general method of camera motion estimation [6, 7].

However, most ideas on the future improvements will be revealed during the intensive testing of the reliability of our algorithm with the robot (fig. 3(c)) of the team MART [8] during the Czech National Cup of Eurobot^{open}2005 robotic contest [4] and hopefully in the international finals at Yverdon-les-Bains, Switzerland.

References

[1] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2000.

[2] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation*, 1998.

[4] Eurobot^{open}. Eurobot assosiation. http://eurobot.org.

[5] D. Haehnel, D. Burgard, W. Fox, and Thrun S. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[6] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[8] MART. Matfyz robotic team. http://mart.matfyz.cz.

[9] S. Thrun, D. Fox, W. Burgard, and F. Dellaer. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[10] J. Wolf and A. Pinz. Particle filter for self localization using panoramic vision. *ÖGAI Journal*, 22(4):8–15, March 2003.