Multi-Cue Localization for Soccer Playing Humanoid Robots

Hauke Strasdat, Maren Bennewitz, and Sven Behnke

University of Freiburg, Computer Science Institute, D-79110 Freiburg, Germany
strasdat@gmx.de,{maren,behnke}@informatik.uni-freiburg.de,
WWW home page: http://www.NimbRo.net

Abstract. An essential capability of a soccer playing robot is to robustly and accurately estimate its pose on the field. Tracking the pose of a humanoid robot is, however, a complex problem. The main difficulties are that the robot has only a constrained field of view, which is additionally often affected by occlusions, that the roll angle of the camera changes continously and can only be roughly estimated, and that dead reckoning provides only noisy estimates. In this paper, we present a technique that uses field lines, the center circle, corner poles, and goals extracted out of the images of a low-cost wide-angle camera as well as motion commands and a compass to localize a humanoid robot on the soccer field. We present a new approach to robustly extract lines using detectors for oriented line pints and the Hough transform. Since we first estimate the orientation, the individual line points are localized well in the Hough domain. In addition, while matching observed lines and model lines, we do not only consider their Hough parameters. Our similarity measure also takes into account the positions and lengths of the lines. In this way, we obtain a much more reliable estimate how well two lines fit. We apply Monte-Carlo localization to estimate the pose of the robot. The observation model used to evaluate the individual particles considers the differences of expected and measured distances and angles of the other landmarks. As we demonstrate in real-world experiments, our technique is able to robustly and accurately track the position of a humanoid robot on a soccer field. We also present experiments to evaluate the utility of using the different cues for pose estimation.

1 Introduction

The knowledge about the poses of the robots during a soccer game is one of the prerequisite features for successful team play. Many approaches to localization on the RoboCup field or less structured environments have already been presented. Several approaches exist that use distance information provided by a proximity sensor, like a laser scanner [1], or images of an omnidirectional camera [2–5] for localizing a robot in the RoboCup environment. However, for some types of robots such sensors do not seem to be appropriate since they do not agree with their design principle. Humanoid robots, for example, which are constructed to resemble a human, are typically only equipped with directed cameras.

In this paper, we present an approach to vision-based localization of a humanoid robot that uses a single wide-angle camera. Many problems exist that make localization for humanoid robots a complex problem. The robot can only observe a constrained part of the soccer field due to the directed camera and due to occlusions caused by moving objects. The camera images are distorted and often blurred because of camera motion. Furthermore, the roll angle of the camera changes continously and can usually only be roughly estimated. The camera perspective, however, has a significant impact on the measured distances and angles to landmarks. Finally, compared to a wheeled robot equipped with odometry sensors, we get a very noisy estimate by dead reckoning, i.e. the prediction of the robot's pose based on executed motion commands.

We apply the well-known Monte-Carlo localization (MCL) technique [6] to estimate the robot's pose (x, y, ϕ), where (x, y) denotes the position on the field and ϕ is the orientation of the robot. MCL uses a set of random samples, also called particles, to represent the belief of the robot about its pose. We extract environment-specific landmarks out of the camera images. Features we use for localization are the field lines, the center circle, the corner poles, and the goals. Additionally, we consider compass data as well as the motion commands sent to the robot.

We present a new technique to robustly extract lines out of images. In contrast to previous approaches that find line transitions on scan lines in the image [2, 7], simply detect edges based on certain color changes [3, 8, 4], or using classical kernels that detect edge gradients in greyscale images [9], we extract line segments out of an image by applying four different detectors that find oriented line points. The detectors guarantee that green is detected on both sides of a line. Since the detectors provide a reliable estimate of the orientations of the individual line points, they are localized well in the Hough domain and we can direct the search towards lines that match these orientations. Before applying the Hough transform to find lines in the two main orientations, we locate the center circle.

For a discrete set of possible poses of the robot, we compute the largest lines that are expected to be visible from that pose. During localization, these lines are then compared to the largest currently observed lines. To get a much more reliable estimate how well two lines fit, we do not only compare their Hough parameters. We also take into account a measure that estimates the positions and lengths of the lines. In the observation model of the particle filter, we additionally consider the differences between measured and expected distances and angles of the other landmarks as well as the compass data. As we demonstrate in practical experiments with a humanoid robot in the RoboCup environment, our technique is able to robustly and accurately track the pose of the robot. In order to assess the benefit of using the different types of features, we present experiments illustrating their influence on the localization result.

This paper is organized as follows. After discussing related work in the following section, we introduce our robot and the RoboCup environment. In Section 4 we describe the Monte-Carlo localization technique that is applied to estimate the robot's pose. In Section 5 we explain how to extract lines out of the images provided by our camera. Afterwards, we present the observation model used for MCL in Section 6. Finally, in Section 7, we show experimental results illustrating the robustness of our approach to estimate the robot's pose.

2 Related Work

Many systems that perform vision-based localization in the RoboCup domain have already been presented. None of them can directly be applied for localization using our humanoid robot as we will point out in the following.

Several localization systems make use of the field lines. De Jong et al. [8] identify edges and divide the image in subimages afterwards. They apply the Hough transform in each subimage assuming that the line segments (even the ones corresponding to the center circle) are mostly straight. They propose to estimate the position of the robot by considering the distance from the observed line segments to the closest expected lines for poses in a local area around its last pose. However, no pose tracking experiments are presented. Furthermore, it remains unclear how to choose an appropriate discretization of the image. Iocchi and Nardi [10] apply an extended Kalman filter for localization with a perspective camera. They match observed and model lines in the Hough domain. One assumption for this method to work is that odometry data yields a good estimate of the robot's pose. This cannot be guaranteed in our case. Furthermore, we have to deal with the problem of changing camera perspectives during walking. Marques and Lima [4] apply a similar approach of line matching in the Hough domain. They search color transitions on circles in omnidirectional images. Röfer et al. [7] distinguish between four different types of lines. In each image, they search for line transitions corresponding to these types and randomly draw three transitions for each. During MCL, the measured angles to the drawn points are then compared to the expected angles of the closest points of each type. Furthermore, they take into account other landmarks like goals and poles. Due to the unpredictable roll angle of the camera, it is in our case not enough to consider only a few points on the lines for localization. We have to use as much information about the lines as we can extract.

Lauer et al. [2] use line transitions extracted out of omnidirectional images and apply gradient decent to minimize an error term in order to locally find the best match between the lines in the current image and the field lines. It is not clear yet how this method performs under real conditions, especially in case of sensor noise. Schulenburg et al. [3] apply a Kalman filter and combine omnivision with laser data for localization. They search for color transitions in the omnidirectional image and find lines in the Cartesian space. Their approach associates each point to a unique line. In order to avoid the data association problem and in order to be able to deal with noisy data, we search for lines in the Hough domain. The approach presented by von Hundelshausen and Rojas [5] uses a technique to track the green regions and search for transitions on the boundaries of these tracked regions. Tracking regions is much easier in omnidirectional images since, due to occlusions during a dynamic soccer game, usually the small field of view changes rapidly when using a directed camera only.

Enderle et al. [11] rely only on particular landmarks during MCL and do not take into account the field lines. This approach may have problems in case of frequent occlusions as they typically occur during a soccer game. Note that Hoffmann et al. [12] propose to utilize negative information during localization. In case the robot did not detect a certain landmark, this information could be used to exclude some states. The idea has so far only been tested in simple setups and not during a soccer game where occlusions are likely to cause problems.

3 The Design of Our Robot and Environmental Setup

The left image of Figure 1 depicts one of our robots, which we used to carry out the experiments presented in this paper. The height of the robot is 60cm and it has 19 degrees of freedom, which are driven by servo motors. We use a Pocket PC, which is located in the chest, to control the robot. The Pocket PC is equipped with a 520MHz processor and 128MB RAM. The robot uses a compact flash color camera with a wide-angle lens to get information about the environment. The lens is located approximately at the position of the larynx. The large field of view (150° horizontally and 112° vertically) allows the robot to see at the same time the ball at its feet and objects above the horizon (see right image of Figure 1). The camera delivers images at a rate of up to 5fps with a resolution of 320×240 pixels. Additionally, the robot is equipped with a compass, which is located in its head, to facilitate localization.

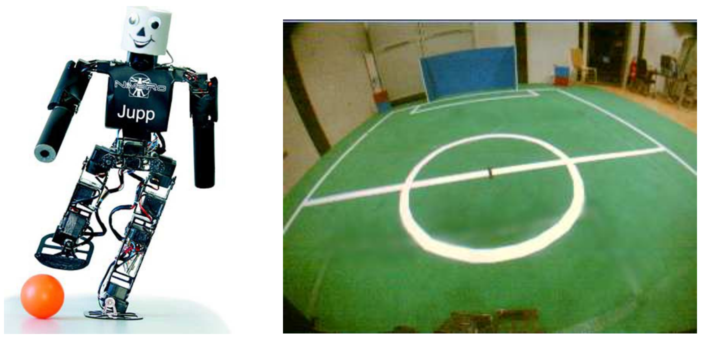

Fig. 1. The left image shows one of our humanoid robots. The robot is controlled by a Pocket PC and uses the images of a low-cost wide-angle camera for perceiving the relevant information about the environment. An image captured from the robot's perspective while it was walking can be seen on the right.

The KidSize soccer field in the Humanoid League has a size of 4.5m $\times$ 3m. Objects that can be used for localization are the two goals (colored blue and yellow), the field lines, the center circle, and four corner poles.

4 Monte Carlo Localization

To estimate the pose x_t of the robot at time t, we apply the well-known Monte-Carlo localization (MCL) technique [6], which is a variant of Markov localization. MCL recursively estimates the posterior about the robot's pose:

$$\begin{aligned} & p(x_t \mid z_{1:t}, u_{0:t-1}) \\ & \quad = \eta \cdot p(z_t \mid x_t) \cdot \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_{t-1}) \cdot p(x_{t-1} \mid z_{1:t-1}, u_{0:t-2})\, dx_{t-1} \end{aligned} \tag{1}$$

Here, η is a normalization constant resulting from Bayes' rule, $u_{0:t-1}$ denotes the sequence of all motion commands executed by the robot up to time $t-1$, and $z_{1:t}$ is the sequence of all observations. The term $p(x_t \mid x_{t-1}, u_{t-1})$ is called motion model and denotes the probability that the robot ends up in state x_t given it executes the motion command u_{t-1} in state x_{t-1}. The observation model $p(z_t \mid x_t)$ denotes the likelihood of making the observation z_t given the robot's current pose is x_t.

MCL uses a set of random samples to represent the belief of the robot about its state at time t. Each sample consists of the state vector $x_t^{(i)}$ and a weighting factor $\omega_t^{(i)}$ that is proportional to the likelihood that the robot is in the corresponding state. The update of the belief is carried out according to the sampling importance resampling particle filter. First, the particle states are predicted according to the motion model. For each particle, a new pose is drawn given the executed motion command since the previous update. In the second step, new individual importance weights are assigned to the particles. Particle i is weighted according to the likelihood $p(z_t \mid x_t^{(i)})$. Finally, a new particle set is created by sampling from the old set according to the particle weights. Each particle survives with a probability proportional to its importance weight. This step is also called resampling.

In order to allow for global localization, e.g. in case of the "kidnapped robot problem", a small amount of the particles is replaced by particles with randomly drawn poses.

5 Finding Lines Using Detectors for Oriented Line Segments and the Hough Transform

In this section, we introduce our approach to robustly locate lines in the images of the wide-angle camera. Figure 2 illustrates the whole process of finding individual line points, extracting line segments, determining their position in egocentric coordinates and their transformation into the Hough space. The details are described below.

5.1 Extracting Oriented Line Segments

To reduce the influence of different lighting conditions, we color-classify pixels in the YUV color space using the pie-slice method [13]. In a multistage process, insignificant colored pixels are discarded so that only the colors of the relevant features remain. To find lines, we are only interested in the pixels that are classified as white or green.

Line points can be detected by processing the color-classified image as follows. We apply elongated Gaussian kernels to determine the likelihood of pixels being part of a line. In short, we search for elongated white regions that have green neighbors on two opposite sides. By taking into account the relative difference of the sum of the likelihoods of green and white pixels within a neighborhood, we estimate the likelihood that a pixel corresponds to a line point. We use Gaussian kernels of two different sizes to search for closer line points, which contain more white pixels and which are in the lower part of the image, and for line points that are farer away. In order to estimate the orientation of a line point, we consider four different orientations and apply individual kernels (see Figure 3) to compute the individual likelihoods. For each line point with a

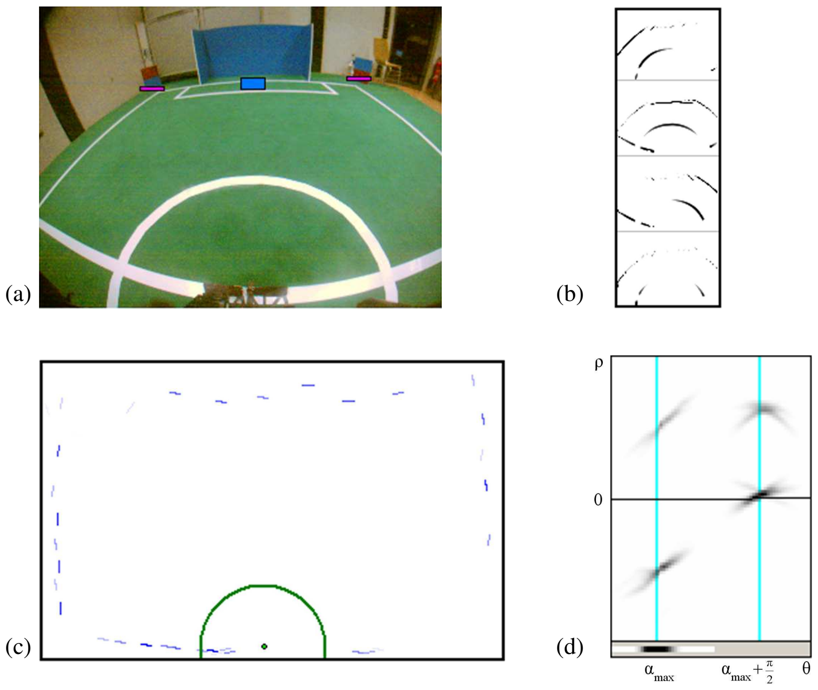

Fig. 2. Illustration of the extraction of lines: The robot's current field of view can be seen in image (a). The four small images (b) show responses of four oriented line detectors. Image (c) depicts the extracted line segments projected into the egocentric space. The center circle is detected and removed. Image (d) shows the representation of the line segments in the Hough space, where the major orientation α_{max} is estimated and the main lines are detected.

likelihood above a threshold, we store the orientation that yielded the maximum likelihood as well as the best neighbor orientation. Figure 2 (b) illustrates the responses of the line detectors for the example image shown in Figure 2 (a).

After identifying the line points in the image, we project them onto the soccer field (see Figure 2 (c)). To get the metric egocentric coordinate on the soccer field corresponding to an image coordinate, we first eliminate the radial distortion in the image and apply an affine transformation afterwards. The parameters of the projective transformation can be calculated from four pairs of points on the field and the corresponding points in the camera image. Here, we assume a fixed viewing angle of the camera onto the field. Note that after the undistortion process and the projection, the line segments have a rather high distance to each other.

In addition to the coordinates of the line points, the corresponding two orientations are also transformed from the image space into the metric egocentric space. We estimate the actual orientation of a line point by computing the weighted mean of these transformed orientations using their likelihoods. Thus, we have a set of weighted line points that are described by their positions and their orientations relative to the robot.

To improve robustness, we do not transform individual line points into the Hough space (see next section). Instead, we merge line points that are close in the image space

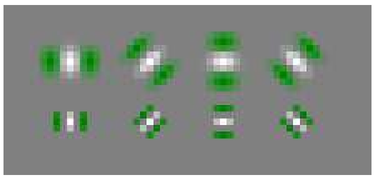

Fig. 3. The four kernels used for the detection of oriented line segments in two different sizes.

and have similar orientations to one longer line segment. This way, we can deal with false detections caused by noisy observations.

5.2 The Hough Transform for Line Extraction

The Hough transform is a robust method to find lines fitting a set of 2D points [14]. It relies on a transformation from the Cartesian plane in the Hough domain. The following curve in the Hough domain is associated with a point (x, y) in the Cartesian plane:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \tag{2}$$

Here, ρ is the perpendicular distance from the origin and θ is the angle with the normal. This curve describes all the lines that go through (x, y), since each point in the Hough space corresponds to a line in the 2D Cartesian space.

By discretizing the Hough space into cells $h(\theta_i, \rho_j)$, one can search for local maxima. These local maxima in the Hough domain correspond to the best fitting lines for the input points. The Hough transform has the advantage that it is quite robust to sensor noise, since outliers do not affect the local maxima. Furthermore, since distances of points on the lines are not considered, it can deal with occlusions.

Before we apply the Hough transform to find the best fitting lines for the extracted oriented segments, we locate the center circle. Whenever large parts of the circle are visible in the image, this can impair the line detection. Parts of the circle can be misclassified as short lines and the circle can potentially affect the estimation of the main orientations. To avoid these problems, we first identify the center circle following the approach presented by de Jong et al. [8]. We consider the individual line segments and vote for the positions at a distance of the radius of the center circle, orthogonal to the orientation of the segment. By determining the largest cluster of points and identifying the segments that voted for it, we find the segments corresponding to the center circle. To avoid false positive detections, we only interpret a cluster as the center circle if the line segments that voted for it have a large range of orientations. The corresponding segments are eliminated, i.e. they are not transformed into the Hough space. Additionally, the knowledge about the position of the center circle is used in the observation model of the particle filter (see next section).

In the standard Hough transform, each line segment (x, y) votes for all bins $h(\theta, \rho)$ which fulfill Eq. (2). Since we have already estimated the orientation of the line segments, we only have to vote for a small subset of bins, which reduces the computational costs. In particular, we accumulate its likelihood in the bins $h(\theta \pm \epsilon, \rho)$ that correspond to the estimated orientation θ of a segment. Here, ϵ indicates that we consider a local neighborhood of θ whose bins are also incremented. In this way, we direct the search towards lines that fit the preestimated orientations. In our current implementation, we use a discretization of $2.5°$ and 6cm for the Hough space.

In general, to locate lines in the Hough space one has to search the entire space for local maxima. In the RoboCup domain, we only have two possible orthogonal orientations for the field lines. This allows us to use a robust method for finding lines that additionally reduces the computational costs: We can determine the two main orientations by adding the bins corresponding to α and $\alpha + \frac{\pi}{2}$, with $\alpha \in [0; \frac{\pi}{2}[$ and finding the maximum:

$$\alpha_{max} = \operatorname*{argmax}_{\alpha=\theta_i \bmod \frac{\pi}{2}} \sum_{\rho_j} h(\theta_i, \rho_j) \tag{3}$$

Finally, we search for maxima in the bins of α_{max} and $\alpha_{max} + \frac{\pi}{2}$, respectively. In this manner, we extract from the Hough space four field lines, two for each main orientation, that are used in the observation model of the particle filter (see next section). Figure 2 (d) illustrates the representation of the segments in the Hough space (the darker the bins the higher the values) as well as the main orientation α_{max}.

6 Observation Model

In this section, we describe how to integrate the observations made by the robot in order to update the weights of the particles.

6.1 Corner Poles and Goals

The corner poles and goals can be detected by processing the color classified image. Using constraints about the number of pixels of certain colors in neighborhoods, these landmarks can be found quite robustly. The egocentric coordinates of the landmark's lower borders are obtained using the projective transformation. For each such landmark, we estimate the distance and angle to the individual particle poses. To compute the likelihood of a landmark observation, we evaluate the differences between expected and measured distances and angles of landmarks

$$p_{landmark} = \exp\Big(-\frac{\|d_e - d_o\|}{2 \cdot \sigma_1^2}\Big) \cdot \exp\Big(-\frac{\|\beta_e - \beta_o\|}{2 \cdot \sigma_2^2}\Big), \tag{4}$$

where σ_1 and σ_2 are the variances of the Gaussians, d_e and d_o are the expected and measured (observed) distance to the landmark, and β_e and β_o are the expected and measured angle of the landmark.

6.2 Center Circle and Lines

If the center circle can be detected, the position of its center is estimated as explained in Section 5.2. To compute the likelihood of this observation, we also evaluate the difference between the expected and the observed position

$$p_{circle} = \exp\Big(-\frac{\|c_e - c_o\|}{2 \cdot \sigma_3^2}\Big). \tag{5}$$

Here, c_e denotes the expected and c_o the measured position of the center circle, and σ_3 is the variance of the Gaussian evaluating the distance.

Line matching is a bit more complicated. First, it should be noted that transforming line segments into the Hough space has a serious shortcome. Hough parameters describe straight lines without starting and endpoints. Thus, significant information gets lost when line segments are transformed into the Hough space. To overcome this drawback, we additionally compute the value u_s that corresponds to the mean position of the line segments s on the straight line l as well as its standard deviation $\sigma(u)$.

In particular, we compute for each single line segment s belonging to l a value

$$u_s = x \cdot \sin(\theta) - y \cdot \cos(\theta), \tag{6}$$

where (x, y) is the the position of s in the Cartesian space and θ is the angular Hough parameter of l. The value u_s indicates, in the egocentric space, the displacement of s along l from the perpendicular of l that goes through the origin. Then, we determine for l the mean $\bar{u}$ and the standard derivation $\sigma(u)$ of the u_s values of its segments. A low value of $\sigma(u)$ corresponds to a short line. On the other hand, a long line results in a high value of $\sigma(u)$. Furthermore, $\bar{u}$ is an indication for the position of the line. These correlations in the Cartesian space are lost when applying the plain Hough transform. In order to more reliable estimate how well two lines fit, we do not only compare their Hough parameters but also their $\bar{u}$ and $\sigma(u)$ values. This is especially important in case the actual role angle of the camera is different from the predicted angle.

In the observation model, we do not take into account all observed lines. Instead, we only consider the largest ones. For a discrete set of possible poses on the soccer field, we precompute the six largest field lines (according to the number of line points) that are expected to be visible from that pose as well as their parameters $\theta, \rho, \bar{u}$, and $\sigma(u)$. From the lines extracted out of the current image, we determine for each of the two main orientations two lines by finding local maxima in the Hough space. We assign each of these observed lines to one of the four largest lines that are expected to be visible from the particle pose. Here, we follow a nearest neighbor approach and use the Hough parameters for comparison.

To compute the likelihood of line observations, we evaluate the differences between expected and measured Hough coordinates $h = (\theta, \rho)$ of matched lines, as well as the differences between the expected and measured $\bar{u}$ and $\sigma(u)$. Note that we only consider the lines corresponding to the nearest neighbor matchings m

$$p_{lines} = \prod_m \exp\Big(- \frac{\|h_e^m - h_o^m\|}{2 \cdot \sigma_4^2} - \frac{\|\bar{u}_e^m - \bar{u}_o^m\|}{2 \cdot \sigma_5^2} - \frac{\|\sigma(u)_e^m - \sigma(u)_o^m\|}{2 \cdot \sigma_6^2} \Big). \tag{7}$$

Here again, the σ's are the variances of the Gaussians, e and o denote the expected and measured values. Figure 4 shows the likelihood of positions given the line observation from the indicated pose.

6.3 Compass

To eliminate ambiguities, which can be caused by the symmetric field lines, the robot can additionally use its compass. Compass data is evaluated with an analogous function

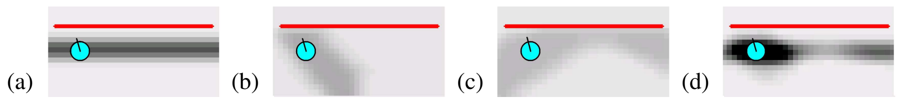

Fig. 4. Likelihood of positions given the observed line using (a) the Hough parameters only, (b) the $\bar{u}$ value, (c) the $\sigma(u)$ value and (d) all four parameters (the darker the more likely).

that is based on the difference between measured and expected values to yield the likelihood p_{com}. Since the compass data is not that precise, we use a Gaussian with a rather high variance to evaluate the difference.

6.4 Integration of All Observations

Finally, we can define for a particle with the pose $x_t^{(i)}$ the likelihood of making the observation $z_t = \{landmarks, circle, lines, compass\}$ as the product of the individual likelihoods

$$p(z_t \mid x_t^{(i)}) = \prod_{landmarks} p_{landmark} \cdot p_{circle} \cdot p_{lines} \cdot p_{com}. \tag{8}$$

Note that we use a confidence value c_j for all individual observations o_j. c_j indicates how sure we are that o_j was correctly observed. We ensure that no observation has a likelihood p_j smaller than $(1 - c_j)$:

$$p_j = (1 - c_j) + c_j \cdot p_j. \tag{9}$$

7 Experimental Results

In order to evaluate our approach to estimate the pose of a humanoid robot on the RoboCup soccer field, we carried out an experiment in which the robot was controlled via joystick to six different poses that were marked with tape on the field. We take these marked positions as “ground truth”. Therefore, part of the errors in the localization results is due to the problem of joysticking the robot exactly onto the marked positions.

Since the robot does not possess any odometry sensors, we perform dead reckoning to estimate the pose of the robot based on motion commands sent to the robot’s base. The control input consists of the robot’s current gait vector that controls the lateral, sagittal, and the rotational speed of omnidirectional walking. The estimated velocities are integrated over time to determine the relative movement. However, due to slippage on the ground the dead reckoning estimate is highly unreliable. We use therefore a rather high noise in the motion model of the particle filter.

Figure 5 shows the true pose of the robot as well as the estimates at the six marked poses using the different cues. It should be noted that in the beginning, the robot globally localized itself on the field. We determined the position of the robot by clustering the particles and computing the weighted mean of the particles of the maximum cluster. In the figure, we only illustrate the pose estimate based on all available cues and the pose estimate resulting when ignoring the lines. As can bee seen, we obtain a much more robust and accurate estimate by considering the lines in the observation model. For example, shortly before the robot reaches position 6 it has a false positive detection of a landmark in the laboratory environment. This results in a false pose estimate when the lines are not used for localization. The average error in the x/y-position was $20cm$ and the average error in the orientation of the robot was $5°$ when the lines were used for localization. The analogous values were $66cm$ and $11°$ in the case that the lines were not used. We also performed experiments in which we ignored the corner poles or the compass. We observed that the lines are the most relevant feature.

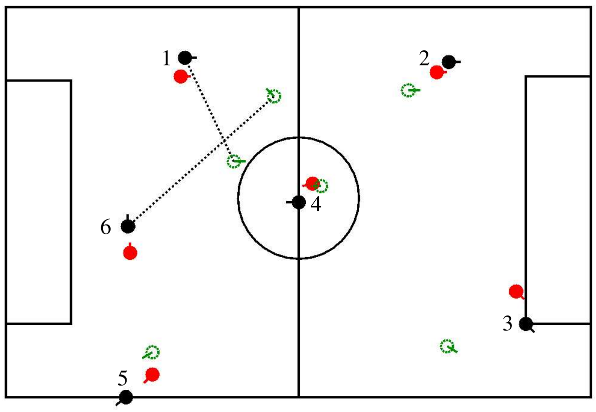

Fig. 5. Result of a localization experiment with our humanoid robot. The black (filled) circles correspond to the true pose, the red (gray) filled circles to the localization result using all cues, and the green (unfilled) circles to the estimate based on all cues except the lines. As can be seen, by considering all available cues the localization works robustly and accurately.

8 Conclusions

In this paper, we presented a robust approach to accurately localize a humanoid robot on the soccer field. This is a challenging task since data from an omnivision camera or a distance sensor is not available. The image of the perspective camera covers only a constrained part of the environment and, additionally, the roll angle of the camera can only be roughly estimated. However, it has a high influence on the measured positions of objects. A further problem is that the robot lacks odometry sensors and that dead reckoning provides extremely noisy pose estimates. For reliable localization, we use the the field lines, the center circle, the corner poles, the goals as well as compass data. We developed a new method to robustly extract lines out of noisy images. The main idea is that we apply robust detectors for oriented line points in the image and aggregate them locally to oriented line segments. In the Hough domain, we then can direct the search towards lines that fit these orientations. While comparing an observed line to a model line during localization, we do not only consider their Hough parameters but also an estimate of their positions and lengths. As we demonstrated using a humanoid robot on the soccer field, our system provides a robust and accurate pose estimate.

Acknowledgment

This project is supported by the German Research Foundation (DFG), grant BE2556/2-1.

References

1. Gutmann, J.S., Weigel, T., Nebel, B.: Fast, accurate, and robust self-localization in polygonal environments. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). (1999)
2. M. Lauer, S.L., Riedmiller, M.: Calculating the perfect match: An efficient and accurate approach for robot self-localisation. In Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y., eds.: RoboCup-2005: Robot Soccer World Cup IX. Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin (2006) to appear.
3. Schulenburg, E., Weigel, T., Kleiner, A.: Self-localization in dynamic environments based on laser and vision data. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). (2003)
4. Marques, C.F., Lima, P.U.: A localization method for a soccer robot using a vision-based omni-directional sensor. In Stone, P., Balch, T., Kraetzschmar, G.K., eds.: RoboCup-2000: Robot Soccer World Cup IV. Volume 2019 of Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin (2001) 96–107
5. von Hundelshausen, F., Rojas, P.: Localizing a robot by the marking lines on a soccer field. In: Proc. of the Computer Vision Winter Workshop (CVWW). (2003)
6. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte Carlo localization for mobile robots. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (1998)
7. Röfer, T., Laue, T., Thomas, D.: Particle-filter-based self-localization using landmarks and directed lines. In Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y., eds.: RoboCup-2005: Robot Soccer World Cup IX. Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin (2006) to appear.
8. de Jong, F., Caarls, J., Bartelds, R., Jonker, P.: A two-tiered approach to self-localization. In Birk, A., Coradeschi, S., Tadokoro, S., eds.: RoboCup-2001: Robot Soccer World Cup V. Volume 2377 of Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin (2002) 405–410
9. Roberts, L.: Machine perception of three-dimensional solids. In J. Tippett, e.a., ed.: Optical and Electro-optical Information Processing. MIT Press, Cambridge (1965) 157–197
10. Iocchi, L., Nardi, D.: Hough localization for mobile robots in polygonal environments. Robotics & Autonomous Systems **40** (2002) 43–58
11. Enderle, S., Ritter, M., Fox, D., Sablatnög, S., Kraetzschmar, G.K., Palm, G.: Vision-based localization in robocup environments. In Stone, P., Balch, T., Kraetzschmar, G.K., eds.: RoboCup-2000: Robot Soccer World Cup IV. Volume 2019 of Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin (2001) 291–296
12. Hoffmann, J., Spranger, M., Göhring, D., Jüngel, M.: Exploiting the unexpected: Negative evidence modeling and proprioceptive motion modeling for improved markov localization. In Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y., eds.: RoboCup-2005: Robot Soccer World Cup IX. Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin (2006) to appear.
13. Thomas, P.J., Stonier, R.J., Wolfs, P.J.: Robustness of colour detection for robot soccer. In: Proc. of the 7th International Conference on Control, Automation, Robotics and Vision (ICARCV). (2002)
14. Hough, P.V.C.: Machine analysis of bubble chamber pictures. In: Proc. of the Int. Conf. on High Energy Accelerators and Instrumentation (CERN). (1959)