

From Multiple Images to a Consistent View

R. Hanek, T. Schmitt, M. Klupsch, and S. Buck

AGILO RoboCuppers, <http://www9.in.tum.de/agilo>
Technische Universität München, Germany

Abstract

The approach presented in this paper allows a team of mobile robots to estimate cooperatively their poses, i.e. positions and orientations, and the poses of other observed objects from images. The images are obtained by calibrated color cameras mounted on the robots. Model knowledge of the robots' environment, the geometry of observed objects, and the characteristics of the cameras are represented in curve functions which describe the relation between model curves in the image and the sought pose parameters. The pose parameters are estimated by minimizing the distance between model curves and actual image curves. Observations from possibly different view points obtained at different times are fused by a method similar to the extended Kalman filter. In contrast to the extended Kalman filter, which is based on a linear approximation of the measurement equations, we use an iterative optimization technique which takes non-linearities into account. The approach has been successfully used in robot soccer, where it reliably maintained a joint pose estimate for the players and the ball.

1 Introduction

1.1 Motivation and Goal of the Work

To successfully perform their tasks, most autonomous mobile robots must be able to estimate their own poses, consisting of position and orientation. Furthermore, the interaction with other robots and the manipulation of objects require them to localize other, possibly moving, objects. A strong restriction is that the localization problem has to be solved in real-time.

Often the required localization accuracy varies with the distance of the object to be localized. A robot that wants to grasp an object requires an accurate position whereas less accurate estimates are sufficient to approach the object. Due to the lower price and weight, visual sensors are often preferred against laser range finders. In this paper, we propose a novel approach for estimating the poses of cooperating, mobile robots and the poses of other objects observed by the robots.

1.2 Previous Work

The problem of pose-estimation from images is frequently addressed by the robotics, the computer vision, and the photogrammetry communities. Due to the huge number of publications, a comprehensive review would be beyond the scope of this paper.

Recently sample-based versions of Markov localization became very popular [2, 5, 3, 6]. Closely related to sample-based Markov localization is the CONDENSATION algorithm [1] which is often used for object tracking. Both approaches

represent the posterior distribution of the sought parameters (e.g. the pose) by samples, which allows to approximate virtually any distribution. However, in order to achieve high accuracy, usually a high number of samples is necessary. The conditional probability density of the environment observation has to be computed for each sample pose. A good match between the observation and a sample pose leads to new randomly generated sample poses in the vicinity of the good sample. However, Markov localization does not try to directly improve the already good pose. Hence, Markov localization leads to limited accuracy and/or relatively high computational cost.

Usually vision-based pose estimation methods consists of three steps. First, features are extracted from the image. Second, correspondences are established between image features and model features. In the third step, the pose is estimated using relations between image features and model features. Image features are for example points [10, 8, 7], lines [7, 10, 9], or ellipses [12]. Most methods are only applicable to specific feature types (e.g. lines, circles, ellipses), whereas our method can be applied to all types of smooth curves. We use model knowledge of the robots' environment in order to restrict the search for features and correspondences. Thus, only local operations on the image are necessary, which significantly increases processing speed.

2 Self-Localization from a Single View

In this section, we address the problem of estimating the pose of a robot in 3D world coordinates from a single image grabbed by a calibrated camera mounted on the robot. In the following, the robot pose is estimated by minimizing the distances between observed curves in the image and 2D projections of curves from the 3D world model.

2.1 Specification of Model Curves

The geometric properties of the robots' operating environment are specified by a predefined world model which consists of curve features. A 3D curve feature is given by a curve function $C_i : D_i \rightarrow \mathcal{R}^3$ defining the set $\mathcal{G}(C_i)$ of curve points in 3D world coordinates by

$$\mathcal{G}(C_i) = \{C_i(s) | s \in D_i\} \quad (1)$$

where $D_i = [s_{i,min}, \dots, s_{i,max}]$ is the domain for s . In practice, all relevant curves, e.g. circles or B-splines, can be specified or at least approximated by curve functions. Especially for a man-made environment a polyhedral model is often used. For such a model the curves are line segments which are given by

$$C_i(s) = s \cdot B_{i1} + (1 - s) \cdot B_{i2} \quad (2)$$

where the points B_{i1} and B_{i2} are the endpoints of the line segments and D is equivalent to $[0, \dots, 1]$. The observation of a 3D curve $\mathcal{G}(C_i)$ in the image is a 2D image curve

$$\mathcal{G}(c_i) = \{c_i(s, \Phi) | s \in D_i\} \quad (3)$$

where Φ is the robot pose and c_i is the image curve function given by

$$c_i(s, \Phi) = proj(C_i(s), \Phi) \quad (4)$$

The function *proj* projects a 3D point, given in world coordinates, into the image and returns the pixel coordinates of the resulting 2D point. First, the function *proj* converts the 3D world coordinates into 3D robot coordinates. This conversion depends on the pose Φ of the robot. The first two elements of the robot pose Φ are the x - and the y -coordinate of the robot position in world coordinates and the third element specifies the angle of the robot's orientation. Since we use a calibrated camera mounted on the robot, the 3D robot coordinates can be transformed into 3D camera coordinates and finally, the pixel coordinates of the 2D projection can be computed. The resulting image curve function c_i describes the relation between the sought robot pose Φ and the position of the model curve points in the image.

In order to obtain curve functions of occluding edges, e.g. the silhouette of a sphere or a cylinder, we first compute the tangent points in 3D world coordinates and project them into the image.

We distinguish two types of image curves: 1.) edge curves. An edge curve separates two image regions which differ in the distribution of color vectors. An edge is specified by a single curve function. 2.) line curves. For a line curve two curve functions are used which not only describe the position of the line but also its width.

Usually, not only knowledge of the geometric properties of a curve feature is given but also knowledge of possible color distributions of the adjacent regions. This knowledge is used in 2.2 in order to reduce the search space of possible correspondences between observed image curves and model curves. We assign a set of possible color vectors to each side of an edge. Similarly, for a line curve two color sets are used, one for the background color and one for the line itself.

2.2 Iterative Optimization of the Pose

From the context of the application or from previous observations usually uncertain a-priori knowledge of the robot pose Φ is given. We model the a-priori probability density $p(\Phi)$ by a multi-variate Gaussian density with mean vector $\bar{\Phi}$ and covariance matrix $\mathbf{C}_{\bar{\Phi}}$. In the following the mean vector $\bar{\Phi}$ is used in order to establish correspondences between image curves and model curves.

Search for points on the corresponding image curve For each projected model curve c_i , the set $\mathcal{H}(c_i)$ of initial points is computed by

$$\mathcal{H}(c_i) = \{c_i(s_{i,j}, \bar{\Phi}) \mid s_{i,j} \in S_i\} \quad (5)$$

where the set $S_i \subset D_i$ contains the values $s_{i,j}$ for which the curve functions are evaluated. The set S_i has to be chosen such that for all $s_{i,j} \in S_i$, the resulting image points are valid, i.e. the 3D points are in front of the camera and the projections are within the image area.

For each projected model point $P_{i,j} = c_i(s_j, \bar{\Phi}) \in \mathcal{H}(c_i)$, the tangent vector $\mathbf{v}_{i,j}$ is computed by

$$\mathbf{v}_{i,j} = \frac{\partial}{\partial a} c_i(a, \bar{\Phi}) \Big|_{a=s_{i,j}} \quad (6)$$

Image points $\tilde{P}_{i,j}$ of the corresponding image curve are sought along the perpendiculars of $\mathbf{v}_{i,j}$, see Fig. 1. In order to avoid wrong correspondences, an image

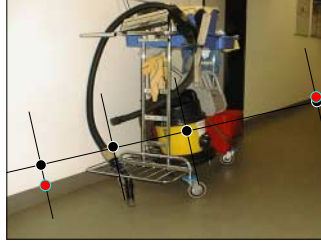


Fig. 1. Search for image points along the perpendiculars: for the two perpendiculars in the center of the image no image points are found which lie between regions of the required colors.

point $\tilde{P}_{i,j}$ on an edge is only accepted if both sides of the edge have consistent colors according to the curve model. In Fig. 1, the region below an image point has to be 'grey' and the region above has to be 'white'. Analogously, for line curves the color distributions of the line and the background are used in order to eliminate wrong image points. Furthermore, the two curve functions defining a line curve are used in order to estimate the line width in the image. A line point is rejected if the line width in the image is not similar to the line width of the projected model line. For each image point $\tilde{P}_{i,j}$, the standard deviation $\sigma_{i,j}$ is estimated which describes the probable precision of the observation $\tilde{P}_{i,j}$ along the perpendicular.

MAP-Estimation In the following, we describe how the robot pose is estimated by minimizing the distances between observed image points $\tilde{P}_{i,j}$ and projected model curves $\mathcal{G}(c_i)$. The approach presented here is a modification and generalization of [7].

In order to express the displacements between image points $\tilde{P}_{i,j}$ and projected model curves $\mathcal{G}(c_i)$ as a function of the robot pose Φ , the projected model curves $\mathcal{G}(c_i)$ are approximated in the vicinities of the projected model points $P_{i,j} = c_i(s_{i,j}, \Phi)$ by straight lines, see Fig. 2. We denote the normal vector perpendicular to $\mathbf{v}_{i,j}(\Phi)$ by $\mathbf{n}_{i,j}(\Phi)$. The displacement $d_{i,j}(\Phi)$ between an observed image point $\tilde{P}_{i,j}$ and the corresponding tangent of the projected model curve c_i is

$$d_{i,j}(\Phi) = (\mathbf{n}_{i,j}(\Phi))^T \cdot [c_i(s_{i,j}, \Phi) - \tilde{P}_{i,j}] \quad (7)$$

where $(\cdot)^T$ indicates vector transposition. If the curvature of the projected model curve is not zero and the observation $\tilde{P}_{i,j}$ is not exactly on the perpendicular then $d_{i,j}(\Phi)$ is just an approximation of the real displacement between $\tilde{P}_{i,j}$ and the projected model curve $\mathcal{G}(c_i)$.

The observations $\tilde{P}_{i,j}$ are noise-affected. We assume that for all observations $\tilde{P}_{i,j}$, the components of the noise in the direction perpendicular to the projected model curves are mutually independent and Gaussian distributed with mean value zero and standard deviation $\sigma_{i,j}$. With this assumption, the maximum

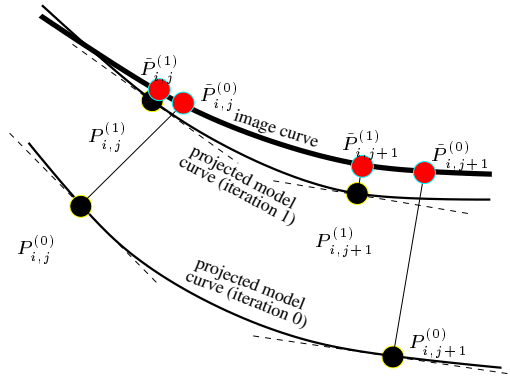


Fig. 2. One iteration step: the new projected model curve is closer to the image curve.

a-posteriori (MAP) estimation $\hat{\Phi}$ of the robot pose is given by

$$\hat{\Phi} = \underset{\Phi}{\operatorname{argmax}} p(\Phi) \cdot \prod_{(i,j)} \frac{1}{\sqrt{2\pi\sigma_{i,j}}} \exp\left(\frac{-d_{i,j}^2(\Phi)}{2\sigma_{i,j}^2}\right) \quad (8)$$

The maximization of this product can be transformed into a more convenient minimization of a sum by taking the negative logarithm:

$$\hat{\Phi} = \underset{\Phi}{\operatorname{argmin}} \chi_1^2 + \chi_2^2 \quad \text{with} \quad (9)$$

$$\chi_1^2 = C_1 (\bar{\Phi} - \Phi)^T C_{\Phi}^{-1} (\bar{\Phi} - \Phi) \quad (10)$$

$$\chi_2^2 = C_2 \sum_{(i,j)} \frac{d_{i,j}^2(\Phi)}{2\sigma_{i,j}^2} \quad (11)$$

where $\bar{\Phi}$ is the mean vector and C_{Φ} is the covariance matrix of the a-priori density $p(\Phi)$. The variables C_1 and C_2 do not depend on the pose Φ . In order to minimize $\chi_1^2 + \chi_2^2$, we use Newton iteration [11]. Fig. 2 illustrates the result of one iteration step. After the iteration step, the projected model curves are closer to the observed image points. However, the projected points $P_{i,j}$ on the model curves are shifted along the model curves. Due to this shift, the correct displacements between the image points $\tilde{P}_{i,j}$ and the projected model curves may differ from the estimations $d_{i,j}(\Phi)$ which are obtained by linear approximation. In order to yield precise measurements between the model curves and the image curves, new image points are sought along the perpendiculars passing through the new projected model points. Since the deviation between image curves and projected model curves is already reduced, the new search can be performed at low computational cost.

In experiments where the initialization was obtained from previous images after two or three iterations the changes of the projected model curves were less than one pixel. Since the pose consists of three variables, three independent displacements $d_{i,j}$ corresponding to three image point $\tilde{P}_{i,j}$ are sufficient to estimate the pose. However, in order to increase robustness and accuracy, for each visible feature three image points $\tilde{P}_{i,j}$ are sought.

For most applications, a rough quantification of the accuracy of $\hat{\Phi}$ is needed which enables the robot to adapt its behavior to the uncertainty of its presumed pose. In order to estimate the uncertainty of the pose, we apply a method based on Taylor approximation of the minimized term $\chi_1^2 + \chi_2^2$, see e.g. [11]. This method can be applied when the measurement errors are normally distributed and the number of image points is large enough, so that the uncertainties in the estimated parameters Φ do not extend outside a region in which the functions c_i could be replaced by suitable linearization.

3 Combined Self-Localization and Object-Localization

3.1 Localization from a Single View

On the one hand, the estimated world coordinates of the observed objects depend on the world coordinates of the observing robot. On the other hand, a-priori knowledge of the poses of the observed objects can be used for the self-localization of the observing robot. In order to exploit these interdependencies, the self-localization and the localization of other objects are performed simultaneously in a single optimization.

Analogously to the self-localization we use curve functions c_i which describe the position of curve points in the image. In order to describe the appearance of an observed object, the object's features are first transformed into world coordinates and thereafter, the observations of these features are computed.

The optimization is done (as for the self-localization) simultaneously over all curves, no matter whether a curve feature belongs to the static world model or to a mobile object. For the combined minimization, the sought parameter vector Φ contains the pose of the observing robot and the poses of the observed objects.

3.2 Data Fusion Over Time

In this section, we discuss the fusion of observations made by several autonomous mobile robots into a single dynamic view of the world. This dynamic world model contains the poses of the robots and the poses of all other observed objects. Since the variety of the robots should not be restricted, we assume that the cameras of the robots are not synchronized and the frame rate may vary from camera to camera. Furthermore, the presented approach allows to use an arbitrary number of cameras mounted on a single robot. The communication between the robots takes place via a wireless LAN. Due to the wireless communication some data packages may be lost.

In order to fuse observations over time, we use a modification of the extended Kalman filter, see e.g. [4] for a description of the extended Kalman filter. For an image obtained at time t , from previous observations an estimation of the a-priori distribution of the sought parameter vector Φ is computed. As described in section 3.1, the new image is used in order to update the estimate of the parameter vector.

Our approach differs from the extended Kalman filter [4] as follows: The Kalman filter theory assumes a linear relation between the measurements in the image and the sought parameter vector, in our case the poses Φ . However, due to rotation(s), perspective projection, and radial distortions of the lens, the curve functions c_i are non-linear in Φ . The extended Kalman filter uses a first-order Taylor approximation in the vicinity of the predicted parameter vector.

The quality of this approximation depends on the distance between the predicted parameter vector and the correct vector, and on the curvatures of the displacements $d_{i,j}(\Phi)$. In contrast to the extended Kalman filter, we use Newton iteration which approximates the displacement functions in the solution of the previous iteration step. This yields a more accurate solution if the relation between the observations and the parameter vector is not linear. The computational cost for a single image is about n times larger, where n is the number of iterations. However, due to the resulting higher accuracy of the predictions, only few iterations (usually two or three) are necessary.

After a robot R_1 has updated its estimate $\hat{\Phi}$, it could broadcast the values of $\hat{\Phi}$ to other robots which could adapt it. However, this communication strategy might be error-prone. If robot R_1 has not received some data packages via the wireless LAN then it would spread out inaccurate data.

Therefore, a robot broadcasts the time t of its latest observation and a second order Taylor approximation $\tilde{\chi}_2^2(\Phi)$ of $\chi_2^2(\Phi)$ obtained for the vicinity of $\hat{\Phi}$. The function $\chi_2^2(\Phi)$, see Eq. (11), summarizes all observations obtained from the latest image. The receiving robots estimate their own predictions for time t , substitute in Eq. (9) the term $\chi_2^2(\Phi)$ by the received Taylor approximation $\tilde{\chi}_2^2(\Phi)$ and minimize $\chi_1 + \tilde{\chi}_2$ according to Eq. (9). It is noteworthy that for the minimization of $\chi_1 + \tilde{\chi}_2$ an efficiently computable closed-form solution exists.

In situations where not enough features are given, pure vision based navigation is impossible. In order to cope with these temporal cases, we combine the visual information with odometric data. The odometric data is used in order to improve the prediction of the robot's pose.

4 Experimental Results

The presented method is applied in our middle-size RoboCup team, the AGILO RoboCuppers. At present, the RoboCup scenario defines a fixed world model with field-boundaries, lines and circles (see Fig. 4). Our approach was successfully applied in 1999 during the RoboCup World Soccer Championship and the German Vision Cup. During a RoboCup match, every robot is able to process 15 to 18 frames per second with its on-board Pentium 200 MHz computer. The localization runs with a mean processing time of 18 msec for a 16-Bit RGB (384 * 288) image. Only for 4% of the images the processing time exceeds 25 msec.

In the first experiment, the accuracy of the self-localization is investigated. One robot is set up at six different positions (see Fig. 3) and the self-localization algorithm is run for about 30 seconds. From about 750 pose estimates, the mean error of the x -/ y -coordinates and the rotation angle are computed and displayed in Tab. 1. In general, the accuracy of the estimated pose depends on the visible features and their distance to the camera. The poses 1), 4) and 5) allow for quite accurate pose estimation. In pose 2) the visible features are far away and therefore, the resulting estimates are less accurate. The problem of pose 3) is that an error in the y -coordinate can be compensated by the rotation angle, and vice versa. From pose 6) only one edge is visible which is not sufficient in order to determine all three pose parameters. In this case data-fusion with e.g. odometric data is a necessity to overcome this problem. In areas where precise robot movements are required, e.g. near the goals, enough features for robust vision-based pose estimation are present (see also Experiment 3). An analysis of the errors shows that the variance of the estimates can be neglected. The

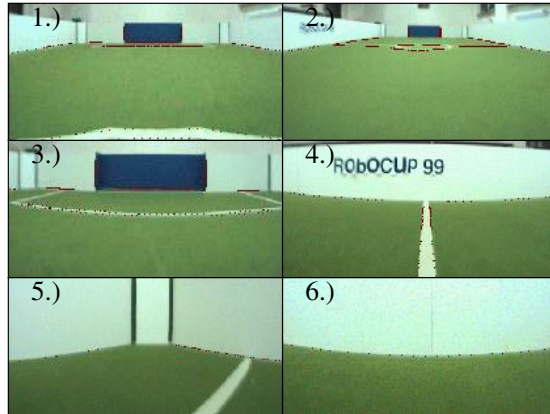


Fig. 3. Views from 6 different poses on the field

Mean pose errors	1)	2)	3)	4)	5)	6)
Φ_x (cm)	0.4	26.0	1.9	0.3	4.3	>99
Φ_y (cm)	1.0	20.3	15.5	8.0	3.2	2.3
Φ_φ (degree)	1.1	4.1	8.1	2.4	7.5	2.8

Table 1. Mean error of the self-localization for the x -, y -coordinates, and the rotation angle φ

bias of the estimation is caused by inaccuracies in the camera calibration and unevenness of the floor.

In the second experiment, the robots sensor data fusion is tested. Three robots are put at predefined positions on the field. After the robots have estimated their pose, a ball is put on the field, such that all robots can see the ball. Fig. 4 outlines the experimental setup. Tab. 2 gives a brief summary of the results. Three rows are given for every robot. The first row exhibits the robot’s exact pose and the exact ball position. In the second row, the robot’s pose estimate and the corresponding uncertainties are displayed. The third row shows the estimated robot and ball poses after the ball was put on the field. All

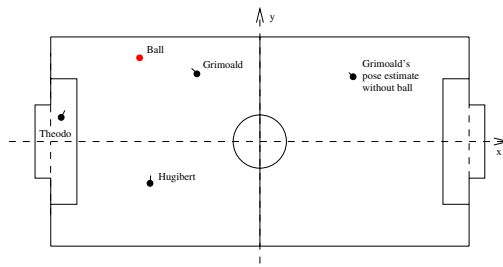


Fig. 4. Cooperative localization: Theodo and Hugibert tell Grimoald where they see the ball. Grimoald used the received ball coordinates to estimate his shift along the wall.

Robot		x	y	φ	ball x	ball y	σ_x	σ_y	σ_φ	σ_{ballx}	σ_{bally}
Theodo	ground truth	-3.80	0.45	40	-2.30	1.60	—	—	—	—	—
	estim. without ball	-3.80	0.43	39	—	—	.0433	.0563	1.439	∞	∞
	estim. with ball	-3.77	0.47	40	-2.26	1.63	.0396	.0472	1.106	.0767	.0731
Hugibert	ground truth	-2.10	-0.80	90	-2.30	1.60	—	—	—	—	—
	estim. without ball	-2.13	-0.94	92	—	—	.1130	.0924	8.738	∞	∞
	estim. with ball	-2.12	-0.78	90	-2.31	1.63	.0941	.0872	5.072	.1057	.0937
Grimoald	ground truth	-1.15	1.30	140	-2.30	1.60	—	—	—	—	—
	estim. without ball	1.77	1.25	138	—	—	∞	.0149	2.753	∞	∞
	estim. with ball	-1.14	1.27	140	-2.31	1.58	.1053	.0143	2.544	.0863	.0146

Table 2. Results of the cooperative localization experiment (See Figure 4 for the experimental setup.)

positions are given in cartesian world coordinates and are measured in meters. Orientations are measured in degrees. The robots Theodo and Hugibert estimate their poses with high accuracies. Grimoald can only detect the edge between the field and the top boundary. Thus, he estimates y and φ correctly but sees itself too far to the right. After the ball was put on the field, Grimoald receives from the other two robots the ball’s position estimates and applies them to correct its own pose. This experiment was also performed several times with moving robots and equally good results.

In the third experiment, we examine the capability to fuse visual and odometric data over time. A robot starts at the top left corner of the playing field and moves along an 8-shaped trajectory across the field, see Fig. 5. Starting- and ending-point of the trajectory are the same. The four major tuning-points are at the corners of the penalty-area ($x = +/- 3$ m, $y = +/- 1.25$ m). The first trajectory was derived exclusively from the vision based pose estimates. The second trajectory was obtained by dead-reckoning. The third trajectory is the result of the fusion process combining data from both sensors. The weaknesses of the pure vision-based localization and dead-reckoning approach are clearly visible. For the vision-based approach, comparatively fast changes of the estimated pose may happen if the observed features change. In general this happens only when too few or too distant features are visible. This can be observed when the robot crosses the field diagonally and does not see the middle-line and center-circle anymore (see Fig. 4). The dead-reckoning trajectory exhibits error accumulation over time: starting- and ending-point of the trajectory are about 80 centimeters apart. The approach based on both data types is able to cope with both weaknesses. Vision based pose inaccuracies are compensated by the relative movements derived from the odometric data. The accumulation of the dead-reckoning error is corrected by the vision based pose estimates.

5 Summary and Conclusion

The approach presented in this paper allows mobile robots to estimate cooperatively their own poses and the poses of other observed objects from images. The approach is able to exploit the interdependencies between self-localization and the localization of other observed objects.

The method runs faster than frame-rate. This high speed is achieved by an exclusively local search for image points in the vicinity of predicted model

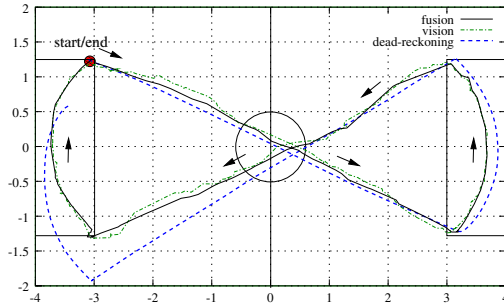


Fig. 5. Data-fusion of vision and dead-reckoning position estimates

points. The fusion over time is also performed efficiently with a technique similar to the extended Kalman filter. In contrast to the extended Kalman filter, we explicitly take the nonlinearity of the measurement equations into account. This leads to high accuracies and good predictions. We have shown that cooperative localization leads to a higher localization accuracy. Furthermore, our method allows for solving certain localization problems that are unsolvable for a single robot.

References

1. BLAKE, A., AND ISARD, M. *Active Contours*. Springer-Verlag, Berlin Heidelberg New York, 1998.
2. DELLAERT, F., BURGARD, W., FOX, D., AND THRUN, S. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (1999), pp. II:588–594.
3. ENDERLE, S., RITTER, M., FOX, D., S., S., AND G., K. Soccer-robot localization using sporadic visual features. In *6th Int. Conf. on Intelligent Autonomous Systems* (2000).
4. FAUGERAS, O. Three-dimensional computer vision: A geometric viewpoint. *MIT Press* (1993), 302.
5. FOX, D., BURGARD, W., DELLAERT, F., AND THRUN, S. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence* (1999).
6. FOX, D., BURGARD, W., KRUPPA, H., AND THRUN, S. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots* (2000). To appear.
7. HANEK, R., NAVAB, N., AND APPEL, M. Yet another method for pose estimation: A probabilistic approach using points, lines and cylinders. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (1999), pp. II:544–550.
8. HARALICK, R., JOO, H., LEE, C., ZHUANG, X., VAIDYA, V., AND KIM, M. Pose estimation from corresponding point data. *IEEE Trans. SMC* 19, 6 (November/December 1989), 1426–1446.
9. IOCCHI, L., AND D., N. Self-localization in the robocup environment. In *3rd RoboCup Workshop, Springer-Verlag, in press.* (1999).
10. PHONG, T., HORAUD, R., YASSINE, A., AND TAO, P. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision* 15 (1995), 225–243.
11. PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1996.
12. SHIU, Y., AND ZHUANG, H. Pose determination of cylinders, cones, and spheres. In *Applications of Artificial Intelligence X: Machine Vision and Robotics* (Orlando, Florida, Apr. 1992).