Simultaneous Planning, Localization, and Mapping in a Camera Sensor Network

David Meger[1], Ioannis Rekleitis[2], and Gregory Dudek[1]

[1] McGill University, Montreal, Canada [dmeger,dudek]@cim.mcgill.ca
[2] Canadian Space Agency, Longueuil, Canada Ioannis.Rekleitis@space.gc.ca

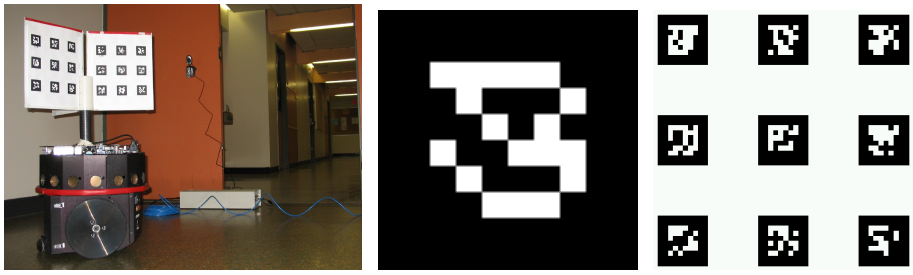
Summary. In this paper we examine issues of localization, exploration, and planning in the context of a hybrid robot/camera-network system. We exploit the ubiquity of camera networks to use them as a source of localization data. Since the Cartesian position of the cameras in most networks is not known accurately, we consider the issue of how to localize such cameras. To solve this hybrid localization problem, we subdivide it into a local problem of camera-parameter estimation combined with a global planning and navigation problem. We solve the local camera-calibration problem by using fiducial markers embedded in the robot and by selecting robot trajectories in front of each camera that provide good calibration and field-of-view accuracy. We propagate information among the cameras and the successive positions of the robot using an Extended Kalman filter. The paper includes experimental data from an indoor office environment as well as tests on simulated data sets.

1 Introduction

In this paper we consider interactions between a mobile robot and an emplaced camera network. In particular, we would like to use the camera network to observe and localize the robot, while simultaneously using the robot to estimate the positions of the cameras (see Fig. 1a). Notably, networks of surveillance cameras have become very commonplace in most urban environments. Unfortunately, the actual positions of the cameras are often known only in the most qualitative manner. Furthermore, geometrically accurate initial placement of cameras appears to be inconvenient and costly. To solve this hybrid localization problem, we will divide it into two interconnected sub-problems. The first is a local problem of camera-parameter estimation which we solve by using fiducial markers embedded in the robot and by selecting robot trajectories before each camera that provide good calibration and field-of-view accuracy. The second problem is to move the robot over large regions of space (between cameras) to visit the locations of many cameras (without *a priori* knowledge of how those locations are connected). That, in turn, entails uncertainty propagation and planning.

In order for the camera network and the robot to effectively collaborate, we must confront several core sub-problems:

1. **Estimation -** detecting the robot within the image, determining the camera parameters, and producing a metric measurement of the robot position in the local reference frame of the camera.



(a) (b) (c)

Fig. 1. (a) The robot with calibration patterns on the target in front of a camera. (b) An example ARTag marker. (c) A calibration target formed from ARTag markers

2. **Local planned behavior -** planning the behavior of the robot within the field of view of a single camera, the robot needs to facilitate its observability and, ideally, maximize the accuracy of the camera calibration.
3. **Data fusion -** combining local measurements from different sources in order to place the cameras and the robot in a common global frame.

The task of computing camera parameters and obtaining metric measurements is referred to as *camera calibration* and is well-studied in both photogrammetry and computer vision [6, 1]. Typically camera calibration is a human intensive task. Section 3.1 will detail an automated version where the robot replaces the human operator in moving the calibration pattern. A system of bar-code-like markers (see Fig. 1) is used along with a detection library [7] so that the calibration points are detected robustly, with high accuracy, and without operator interaction.

Measurements from the calibration process can be used to localize the robot and place each camera within a common reference frame. This process can be formulated as an instance of Simultaneous Localization and Mapping (SLAM). Typically the robot uses its sensors to measure the relative locations of landmarks in the world as it moves. Since the measurements of the robot motion as well as those of the pose of landmarks are imperfect, estimating the true locations becomes a filtering problem, which is often solved by using an Extended Kalman filter (EKF). Our situation differs from standard SLAM in that our sensors are not pre-calibrated to provide metric information. That is, camera calibration must be performed as a sub-step of mapping.

The path that the robot follows in front of a single camera during calibration will allow a variety of images of the target to be taken. During this local exploration problem, the set of captured images must provide enough information to recover camera parameters. The calibration literature [22] details several cases where a set of images of a planar target does not provide sufficient information to perform the calibration. The robot must clearly avoid any such situation, but we can hope for more than just this simple guarantee. Through analysis of the calibration equations, and the use of the robot odometry, the system discussed here has the potential to perform the calibration optimally and verify the results.

The following section discusses related research. Section 3 details camera calibration using marker detection and a 6 degree of freedom (DOF) EKF for mapping in our context. Section 4 continues the discussion of local calibration paths. Section 5 provides experimental results to examine the effect of different local paths and shows the system operating in an office environment of 50 m in diameter. We finish this paper with concluding remarks.

2 Related Work

Previous work on the use of camera networks for the detection of moving objects has often focused on person tracking in which case the detection and tracking problem is much more difficult than that of our scenario (due to lack of cooperative targets and a controllable robot) [9, 4, 5]. Inference of camera network topology from moving targets has been considered [4, 13]. Ellis *et al.* depend on cameras with overlapping fields of view. Marinakis *et al.* deal with non-overlapping cameras, but only topological information is inferred here while we are interested in producing a metric map of the cameras. Batalin and Sukhatme [2] used the radio signals from nodes in a sensor network only for the localization of the robot. Cooperative localization among robots has been considered [11, 15, 17, 10], where instead of camera nodes a robot is observed by other robots.

Camera calibration is a well studied problem; a good summary paper by Tsai [19] outlines much of the previous work, and [22] presents improvements made more recently. A series of papers by Tsai et al. [21, 20] use a 3-D target and a camera mounted on the end of a manipulator to calibrate the manipulator as well as the camera. Heuristics are provided in [21] to guide the selection of calibration images that minimizes that error. However, these methods only deal with a single camera and use manipulators with accurate joint encoders, *i.e.*, odometry error is not a factor.

One important step in the automation of camera calibration is the accurate detection of the calibration pattern in a larger scene. Fiducial markers are engineered targets that can be detected easily by a computer vision algorithm. ARToolkit [14] and ARTag [7] are two examples. ARTag markers are square black and white patches with a relatively thick solid outer boundary and an internal 6 by 6 grid (see Fig. 1b,c). The advantages of this system are reliable marker detection with low rates of false positive detection and marker confusion. ARTag markers have been previously used for robot localization in [8] where a camera from above viewed robots, each of which had a marker attached on top.

The EKF is used for mapping in the presence of odometry error, a method that was detailed by [18],[12] and others to form the now very mature SLAM field. An example of previous use of camera networks for localization and mapping is [16]. Our work extends this previous method by using ARTag markers for much more automated detection of calibration target points, performing SLAM with 3-D position and orientation for cameras and examining the effect of local planning. This gives our system a higher level of autonomy and allows mapping of much larger environments.

3 Mapping and Calibration Methods

Our approach to the general problem of mapping a camera sensor network is divided into two sub-problems: acting locally to enhance the intrinsic parameter estimation; and moving globally to ensure coverage of the network while maintaining good accuracy. As it visits each location for the first time, the robot is detected by a camera. Thus, it can exploit its model of its own pose, and the relative position of the camera to the robot to estimate the camera position. In order to recover the coordinate system transformation between the robot and the camera, it is necessary to recover the intrinsic parameters of the camera through a calibration procedure. This process can be facilitated by appropriate local actions of the robot. Finally, over the camera network as a whole, the robot pose and the camera pose estimates are propagated and maintained using a Kalman filter.

A target constructed from 6 grids of AR-Tag markers is used for automated detection and calibration. When the robot moves in front of a camera, the markers are detected, and the corner positions of the markers are determined. A set of images is collected for each camera, and the corner information is used to calibrate the camera. Once a camera is calibrated, each subsequent detection of the robot results in a relative camera pose measurement. The following sub sections provide details about the steps of this process.

3.1 Automated Camera Calibration

A fully automated system is presented for the three tasks involved in camera calibration: collecting a set of images of a calibration target; detecting points in the images which correspond to known 3-D locations in the target reference frame; and performing calibration, which solves for the camera parameters through non-linear optimization. The key to this process is the calibration target mounted atop a mobile robot as shown in Fig. 1a. The marker locations can be detected and the robot can then move slightly, so that different views of the calibration targets are obtained until a sufficient number is available for calibration. Six panels, each with 9 markers, are mounted on three vertical metal planes. The 3-D locations of each marker corner in the robot frame can be determined through simple measurements.

The ARTag detection algorithm relies on identification of the fine internal details of the markers. This requires the marker to occupy a large portion of the image and limits the maximum detection distance to about 2 m in our setup. Of course higher-resolution camera hardware and larger calibration patterns will increase this distance.

The non-linear optimization procedure used for camera calibration [22] warrants a brief discussion. A camera is a projective device, mapping information about the 3-D world onto a 2-D image plane. A point in the world $M = [X, Y, Z]^T$ is mapped to pixel $m = [u, v, 1]^T$ in the image, under the following equation:

$$s \underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\mathbf{m}} = \underbrace{\begin{bmatrix} f_x & \alpha & u_x \\ 0 & f_y & u_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} R\, t \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\mathbf{M}} \tag{1}$$

In matrix A, f_x and f_y represent the focal lengths in pixel related coordinates, α is a skew parameter and u_x and u_y are the coordinates of the center of the image. Collectively, these are referred to as intrinsic camera parameters. s is a projective scale parameter. The T matrix is a homogeneous transformation made up of rotation R and translation t, and it expresses the position and the orientation of the camera with respect to the calibration-target coordinate frame. The elements of T are referred to as extrinsic parameters and change every time the camera or the calibration target moves to describe the position of the target relative to the camera. We will use the T matrix as a measurement in the global mapping process described in detail in Section 3.2.

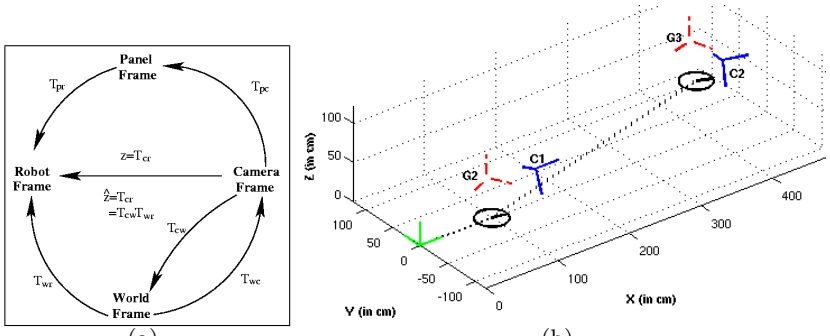
The calibration images give a number of correspondences $(u, v) \rightarrow (X, Y, Z)$, which are related by (1). This relation allows the intrinsic camera parameters and the extrinsic parameters of each image to be jointly estimated using a two-step process. The first step is a linear solution to find the most likely intrinsic parameters. The second step is a non-linear optimization which includes polynomial distortion parameters. Zhang [22] mentions "degenerate configurations" where a set of calibration points do not provide enough information to solve for A and T. This occurs when all of the points lie in a lower dimensional linear subspace of R^3. To avoid this situation, several different local motion strategies are discussed in Section 4.

In conclusion, from a set of images of the robot-mounted target, the camera intrinsic and extrinsic parameters are estimated. The next section will discuss the use of an Extended Kalman filter to combine these estimates with robot odometry in order to build a map of camera positions.

3.2 Six-DOF EKF

The measurements of the extrinsic camera parameters can be used to build a consistent global map by adding the camera position to the map when initial calibration finishes and by improving the estimate each time the robot returns to the camera. To maintain consistent estimates in this global mapping problem, an Extended Kalman filter is used to combine noisy camera measurements and odometry in a principled fashion. The robot pose is modeled as position and orientation on the plane: (x, y, θ). However, the cameras may be positioned arbitrarily; so, their 3-D position and orientation must be estimated. Roll, pitch, and yaw angles are used to describe orientation, thus the state of each camera pose is a vector $X_c = [x, y, z, \alpha, \beta, \gamma]^T$ (for more information, see [3]).

The EKF tracks the states of the robot and the cameras in two steps: the propagation step tracks the robot pose during motion, and the update step



(a) (b)

Fig. 2. (a) Measurement Coordinate Frame Transformations. (b) The world (at [0,0,0]), robot (denoted by a circle and a line for the orientation), target grid (dashed lines G1,G2) and camera (solid lines C1,C2) coordinate frames. The trajectory of the robot is marked by a dotted line.

corrects the robot and the camera poses based on the measurements from the calibration process. For the propagation phase, the state vector and the covariance matrix are updated as in [18].

$$\hat{X}_{k|k-1} = F\hat{X}_{k-1|k-1} \tag{2}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + C_v \tag{3}$$

where F is obtained by linearizing the non-linear propagation function $f(X, u)$ at state X and control actions u, and C_v is a matrix representing odometry error. For the update phase, the measurement equation is a non-linear expression of the state variables so we must again linearize before using the Kalman filter update equations. The measurement equation relates two coordinate frames, so that the language of homogeneous coordinates transformations is used in order to express the relation [3].

The calibration process estimates the calibration panel in the camera frame, that is ${}^C_P T$. Using ${}^P_R T$ which is measured initially this can be transformed into a relation between the camera and robot: ${}^C_R T = {}^C_P T {}^P_R T$. This is the measurement z. Next, the measurement is expressed in terms of the EKF states X_r and $X_c$through which we obtain the transformations for the robot and the camera in world coordinates: ${}^W_R T$ and ${}^W_C T$. Fig. 2 shows the relationships between the EKF state variables and the information obtained from camera calibration which jointly form the measurement equation:

$$z_{measured} = {}^C_R T = {}^C_W T {}^W_R T = {}^W_C T^{-1} {}^W_R T = \begin{bmatrix} {}^W_C R^T & -{}^W_C R^T {}^W_C P \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^W_R R & {}^W_R P \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} {}^W_C R^T {}^W_R R & {}^W_C R^T ({}^W_R P - {}^W_C P) \\ 0 & 1 \end{bmatrix} \tag{4}$$

Equation 4 provides the measurement equation $\hat{z} = h(\hat{X})$. To use this in a Kalman filter, we must differentiate h with respect to each parameter to obtain a first-order linear approximation $z = h(\hat{X}) + H\tilde{X}$ where H is the Jacobian of vector function h. Measurement noise C_ω expresses the uncertainty of transformation parameters from camera calibration. The EKF update equations can be applied as usual:

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K(z - h(\hat{X}_{k|k-1})) \tag{5}$$

$$P_{k|k} = \left[I - KH^T\right] P_{k|k-1} \tag{6}$$

$$K = P_{k|k-1}H(HP_{k|k-1}H^T + C_\omega)^{-1} \tag{7}$$

4 Local Calibration Procedures

Using a robot-mounted target provides a unique opportunity to collect calibration images in an intelligent fashion by controlling the robot motion. However, it is not immediately clear what the best motion strategy will be. There are numerous sources of error including detecting the original pixels, approximating the linear parameters, and convergence of the non-linear optimization all of which should be minimized if possible. As mentioned previously, [22] showed that it is essential to avoid having only parallel planes. [21] discussed heuristics for obtaining images to calibrate a manipulator system. Also, the accumulated odometric error is an important factor for the overall accuracy of the system.

As an initial investigation into this problem, five motion strategies were examined. These were chosen to cover the full spectrum of expected calibration accuracy and odometry buildup:

- **Stationary** - the robot moves in the camera field of view (FOV) and stays in one spot. Due to the target geometry, this allows for two non-parallel panels to be observed by the camera, which provides the minimal amount of information necessary for calibration.
- **One Panel Translation-only** - the robot translates across the camera FOV with only a single calibration panel visible always at the same angle. This is a degenerate case and produces inconsistent results.
- **Multi-Panel Translation-only** - the robot translates across the camera FOV with two panels visible. This provides numerous non-parallel planes for calibration and accumulates minimal odometry error.
- **Rotation-only** - the robot rotates in place in the center of the camera FOV allowing the panels to be detected at different angles.
- **Square Pattern** - the robot follows a square-shaped path in front of the camera. Since there is variation in the detected panel orientation and in depth, this method achieves good calibration accuracy. However, the combination of rotation and translation accumulates large odometry error.

5 Experimental Results

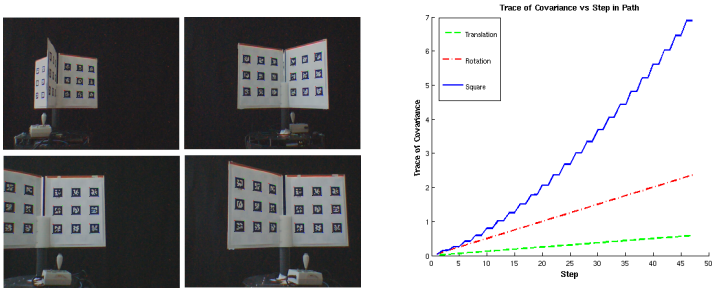
Two separate sets of experiments were conducted using the camera sensor network (see [16] for a detailed description of the experimental setup) which dealt with the mapping and the calibration aspects of our system. First, the five different local motion strategies were examined with respect to the resulting intrinsic parameters and position accuracy. Second, to show that mapping is feasible in a real-world environment, a robot equipped with the calibration target moved through one floor of an office building which was over 50 m in diameter. We show that the robot path estimate is improved through the use of position measurements from a set of cameras present in the environment.

5.1 Local Calibration Paths

A set of experiments was performed to test the effects of the local calibration paths suggested in Section 4. The goal was to study the motion strategies in terms of reliable camera calibration as well as magnitude of odometry error. This test was done inside our laboratory with a Nomadics Scout robot mounted with a target with six calibration patterns. The 5 strategies were performed for 10 trials, with 30 calibration panels detected per trial. The automated detection and calibration system allowed for these 50 trials and 1500 pattern detections to occur in under 3 hours (using a Pentium IV 3.2 GHz CPU running linux for both image and data processing).

Table 1. Mean Value and percentage of Standard Deviation of the Intrinsic Parameters for each strategy over 10 trials. Deviations are with respect to the mean.

Path	Mean Values				Std. Deviation (% of mean value)			
	f_x	f_y	u_x	u_y	f_x	f_y	u_x	u_y
Stationary	903.2	856.0	233.5	190.6	6.3	5.6	30.9	17.1
2 Panel Translation	785.8	784.3	358.0	206.4	2.7	2.3	3.6	5.0
Rotation	787.7	792.0	324.1	236.6	1.6	1.6	3.9	10.3
Square	781.2	793.1	321.4	274.2	1.2	2.0	2.4	13.9



(a) (b)

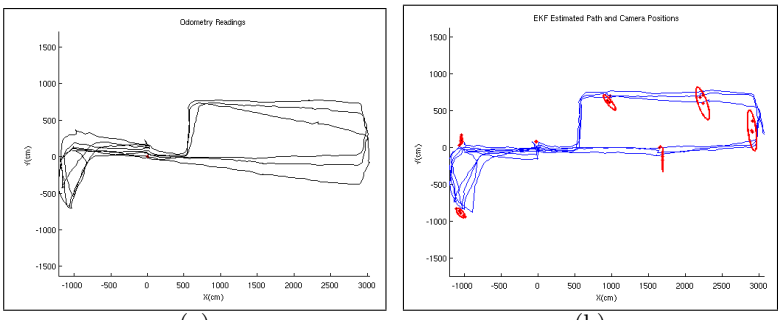
Fig. 3. (a) Sample Images from Square Pattern. (b) Odometry Error Accumulation for 3 Local Calibration Paths

Table 1 summarizes the intrinsic parameters obtained for each method. The lack of data for the One Panel Translation-only path is due to that, as expected, calibration diverged quite badly in all trials with this method. Other than the stationary method, statistically, the mean parameter estimates are not significantly different between methods.

To examine the difference between odometry buildup among the different paths, each of the three paths which involved motion was simulated. To ensure a fair comparison, path parameters (distance and rotation angles) were scaled accordingly for each trajectory. Fig. 3(b) shows the trace of the covariance matrix as each method progresses. The square pattern accumulates much more odometry error than the other two methods, as expected.

5.2 Mapping an Office Building

To demonstrate the effectiveness of the system when mapping a large space, we instrumented an office environment with 7 camera nodes. The environment



(a) (b)

Fig. 4. (a) Odometry Readings for Hallway Path. (b) EKF Estimate of the Hallway Path. Estimated camera positions with uncertainty ellipses (in red)

consisted of a rectangular loop and a triangular loop connected by a long hallway with length approximately 50 m. The same robot as the previous experiment was used to perform the full calibration and mapping procedure described in Section 3. The robot traversed the environment 3 times and traveled in excess of 360 m in total. The Rotation-only local calibration strategy described in Section 4 was used for simplicity.

From Figs. 4a and b, it is visually clear that the use of camera measurements was able to correct for the buildup of odometry error. However, there are some regions where the filtered path is still a rough approximation since the regions between cameras are traveled without correction of the odometry error. This is most obvious on the far right of the image where there is a very noticeable discontinuity in the filtered path. Since the system does not provide a means for odometry correction between the camera fields of view, this type of behavior is unavoidable without additional sensing.

6 Conclusion

We have outlined an automated method for calibrating and mapping a sensor network of cameras such that the system can be used for accurate robot navigation. The experimental methods show that a system with a very simple level of autonomy can succeed in mapping the environment relatively accurately. A preliminary study was done on local calibration trajectories, which can have a profound effect on the accuracy of the mapping system. Further work in planning and autonomy will likely be the key enhancement in further iterations of this system. The reliance on detection of the calibration target means the robot must move intelligently in order to produce a map of the environment and localize itself within that map.

In this work, we propose the use of a 6-DOF EKF for global mapping. While this approach worked quite well even in a large experiment, it assumes that the system is linear and Gaussian which is a poor approximation in some cases. In particular, the robot builds large odometry errors between cameras and the linearization procedure is only a good approximation when errors are small. A probabilistic method such as Particle Filtering might give improved results in this context, since linearization is not necessary for such a technique.

References

1. *Manual of Photogrammetry*. Am. Soc. of Photogrammetry, 2004.
2. M. Batalin, G. Sukhatme, and M. Mattig. Mobile robot navigation using a sensor network. *International Confernce on Robotics and Automation*, 2003.
3. J. J. Craig. *Introduction to Robotics, Mechanics and Control*. 1986.
4. T.J. Ellis, D. Makris, and J. Black. Learning a multicamera topology. In *IEEE Int. Workshop on Visual Surveillance & Performance Evaluation of Tracking & Surveillance*, pages 165–171, 2003.
5. D. Estrin, D. Culler, K. Pister, and G. Sukatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1):59–69, 2002.
6. O. D. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.
7. M. Fiala. Artag revision 1, a fiducial marker system using digital techniques. In *National Research Council Publication 47419/ERB-1117*, November 2004.
8. M. Fiala. Vision guided robots. In *Proc. of CRV'04 (Canadian Confernence on Computer and Robot Vision*, pages 241–246, May 2004.
9. W. E. L. Grimson, C. Stauer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 22–29, 1998.
10. Andrew Howard, Maja J Mataric, and Gaurav S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 434–459, EPFL Switzerland, 2002.
11. R. Kurazume and S. Hirose. An experimental study of a cooperative positioning system. *Autonomous Robots*, 8(1):43–52, Jan. 2000.
12. J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics & Automation*, 7(3):376–382, 1991.
13. Dimitris Marinakis, Gregory Dudek, and David Fleet. Learning sensor network topology through monte carlo expectation maximization. In *Proc. of the IEEE International Conference on Robotics & Automation*, Spain, 2005.
14. I. Poupyrev, H. Kato, and M. Billinghurst. Artoolkit user manual, version 2.33. *Human Interface Technology Lab, University of Washington*, 2000.
15. Ioannis M. Rekleitis, Gregory Dudek, and Evangelos E. Milios. On the positional uncertainty of multi-robot cooperative localization. Multi-Robot Systems Workshop, Naval Research Laboratory, Washington, DC, USA, March 18-20 2002.
16. I. Rekletis and G. Dudek. Automated calibration of a camera sensor network. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 401–406, 2005.
17. Stergios I. Roumeliotis and George A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.
18. R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167 – 193, 1990.
19. R. Y. Tsai. Synopsis of recent progress on camera calibration for 3-d machine vision. *The Robotics Review*, pages 147–159, 1989.
20. R. Y. Tsai and R. K. Lenz. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, pages 323–344, 1987.
21. R. Y. Tsai and R. K. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. *IEEE Int. Conf. on Robotics & Automation*, 1988.
22. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.