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

Ioannis Rekleitis [a] David Meger [b] Gregory Dudek [b]

[a] *Canadian Space Agency, Space Technologies, Saint-Hubert, Quebec Canada* [1]

[b] *School of Computer Science, McGill University, Montreal, Quebec, Canada*

Abstract

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 divide 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 attached to 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. Finally, we move the robot between the camera positions to explore the network using heuristic exploration strategies. The paper includes experimental data from an indoor office environment as well as tests on simulated data sets.

Key words: Sensor Networks, Cooperative Localization.

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. 1). 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

Email addresses: Ioannis.Rekleitis@space.gc.ca (Ioannis Rekleitis), dmeger@cim.mcgill.ca (David Meger), dudek@cim.mcgill.ca (Gregory Dudek).

[1] Work done at McGill University.

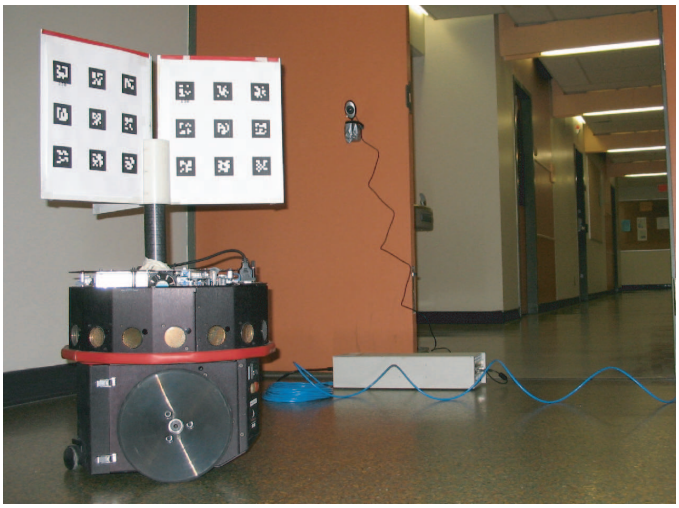


Fig. 1. The robot with the calibration patterns on the target in front of a camera node.

qualitative manner. Furthermore, geometrically accurate initial placement of cameras appears to be inconvenient and costly. Of particular interest is the scenario where one or more service robots operate in an environment already equipped with monitoring cameras. In the event that the cameras are in the same network as the robot they can provide additional sensory input, such as the location of intruders, and/or localize the robot. In order to achieve this goal both the robot and the cameras have to be in the same reference frame and to be able to localize each other. 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 attached to 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:

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.

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.

Data fusion: Combining local measurements from different sources in order to place the cameras and the robot in a common global frame.

Global planning: Determining how the robot should move as it explores the environment.

The first two sub-problems (estimation and local planning behaviour) are part of the local problem of camera-parameter estimation, and the last two (data fusion and global planning) are part of the general SLAM problem.

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 [1,2]. Calibration by standard techniques is a human operator intensive process, which is not suited for use by an autonomous robot. Section 3.1 will detail an automated version where the robot replaces the human operator in moving the calibration pattern and collecting images. A system of bar-code-like markers (see Fig. 3) is used along with a detection library [3] 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 mapping problem can be formulated as a standard 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 relative 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.

In the framework discussed here, measurements can only be made when the calibration target is in the field of view of the cameras. Therefore, robot motion planning is crucial in a number of contexts. During exploration, the order and frequency at which the robot visits the cameras will greatly affect mapping accuracy. Several heuristic strategies for this "global exploration problem" are suggested in Section 5 and are compared experimentally in Section 6.3. In addition, 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 [4] 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

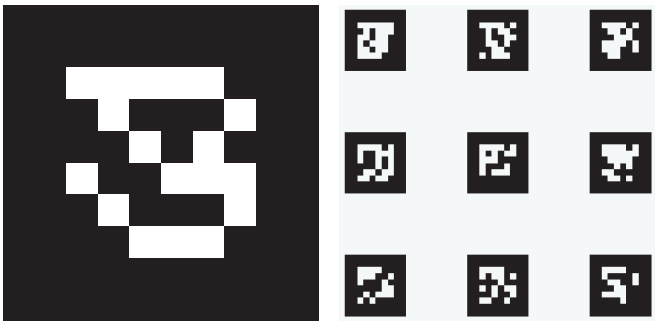
mapping in our context. Section 4 continues the discussion of local calibration paths. The heuristics for exploring the sensor network are discussed in Section 5. Section 6 provides experimental results to examine the effect of different local and global paths and shows the system operating in an office 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) [5–9]. In general, camera-based tracking for either surveillance or activity estimation presupposes that camera positions are either unknown or unnecessary. Inference of camera network topology from moving targets has been considered [8,10]. These methods employ probabilistic inference techniques to find the most likely connections between nodes based on observations. Like our method, they produce a map of the network. 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 [11] used the radio signals from nodes in a sensor network for robot localization. The spirit of this system is quite similar to our own, but the use of cameras instead of radio signal strength presents us with a large number of new challenges and several advantages. Moreover, the previous work considered only localization, while our system also maps the camera poses. Cooperative localization of multiple robots has been considered by many authors, *e.g.*, [12–15], where instead of stationary camera nodes a moving robot is observed by other robots.

Camera calibration is a very well studied problem; a good summary paper by Tsai [16] outlines much of the previous work, and authors such as Zhang [4] and Faugeras [17] present improvements made, more recently. A series of papers by Tsai et al. [18,19] use a 3-D target and a camera mounted on the end of a manipulator to calibrate the manipulator as well as the camera. A fairly complete study of calibration error as a function of properties of a calibration image set is provided, which gives intuition for our local path planning problem. Heuristics are provided to guide the selection of calibration images that minimizes that error [18]. However, these methods only deal with a single camera and use manipulators with accurate joint encoders, *i.e.*, odometry error is not a factor. In the mobile robot context, the presence of large-scale odometry error makes the problem much more challenging.

One important step in the automation of camera calibration is the accurate



(a) (b)

Fig. 2. (a) An example ARTag marker. (b) A calibration target formed from ARTag markers.

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 [20] and ARTag [3] are two common examples of fiducial markers. ARTag markers are square black and white patches with a relatively thick solid outer boundary and an internal 6 by 6 grid (see Fig. 2(a)). The outer border is used for quad detection and the internal grid uniquely identifies each marker even under arbitrary rotation and reflection. 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 where a camera viewed robots from above, each of which had one marker attached to its top [21]. Our system extends this concept to allow multiple cameras in general position.

The EKF is used for mapping in the presence of odometry error [22,23], a method that began the now very mature SLAM field. An example of previous use of camera networks for SLAM is Rekleits and Dudek [24]. 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 both local and global planning. This gives our system a higher level of autonomy and allows mapping of much larger environments.

In this work, we calibrate the entire camera network by covering the environment using an exploration-like strategy (since we do not presuppose a geometric map in correspondence with the camera layout). Exploring this camera network is related to work in both coverage and exploration. In the case of a graph-like environment, as pioneered by Kuipers [25], various techniques have been proposed often assuming that minimal information is available [26–28].

The exploration problem has been addressed by a variety of researchers. Some prior works put the emphasis on the completeness of the exploration [29,30]. An alternative approach emphasizes exploration while minimizing uncertainty [31]. Yet another class of approaches seeks to optimize both accuracy and

efficiency [32] keeping in mind the trade-off between the two. The present work relates to all three, although the latter is the closest in spirit to our objectives.

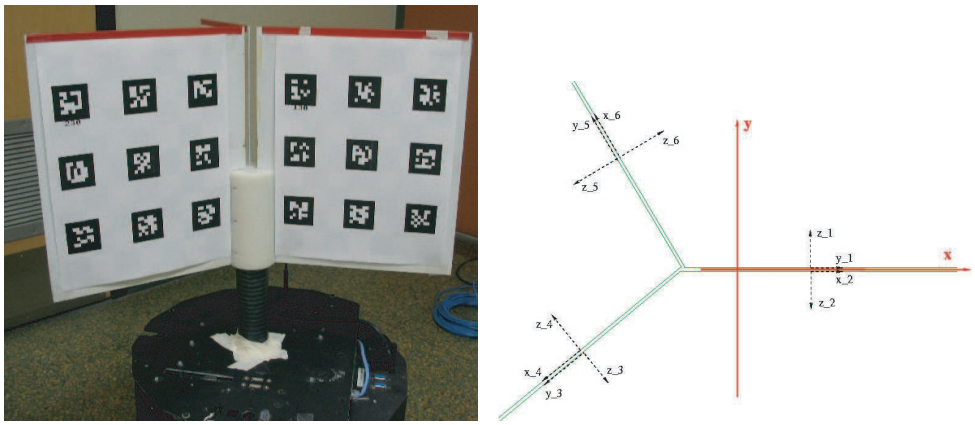
3 Mapping and Calibration Methods

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. The robot will move between camera locations to accomplish its long term objectives. 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 and a heuristic planner.

A target constructed from 6 grids of ARTag 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 nonlinear optimization. The key to this process is the calibration target mounted atop a mobile robot as shown in Fig. 1. The markers on the panel are easily and robustly detected, so that the system can immediately be aware each time the robot passes in front of a camera. The robot can then move slightly, so that different views of the calibration targets are obtained until a sufficient number is available for calibration. Each planar panel comprises nine square ARTag markers (four corners each), thus providing 36 calibration



(a) (b)

Fig. 3. (a) The calibration target is formed by 3 panels and mounted on top of the robot. (b) A top view of the target. The x and y axes of the robot coordinate frame are displayed as thick solid arrow lines and the x (or y) and z axes of each of the six grid coordinate frames are displayed as dashed arrow lines; solid thin lines represent the outline of the target.

points in an evenly spaced planar grid. Six panels are mounted on the front and the back sides of three vertical metal planes. The three planes are separated by 100, 120, and 140 degrees. The 3-D locations of each marker corner in the robot frame can be determined through simple measurements, and the ARTag software library provides robust detection of these corner points in the image.

The ARTag system requires that each marker occupies a sufficient portion of the image for the relatively fine details of the internal six by six grid to be identified. This imposes a limit of approximately 15 pixels as the minimum marker size in the image for robust detection, which translates into a maximum distance from the camera at which the calibration panel can be identified. The specific distance depends on camera resolution and imaging properties as well as the size of the target. With inexpensive cameras and a letter-sized paper target, approximately a 2 m maximum detection distance is achieved. Of course higher-resolution camera hardware and larger calibration patterns will increase this distance.

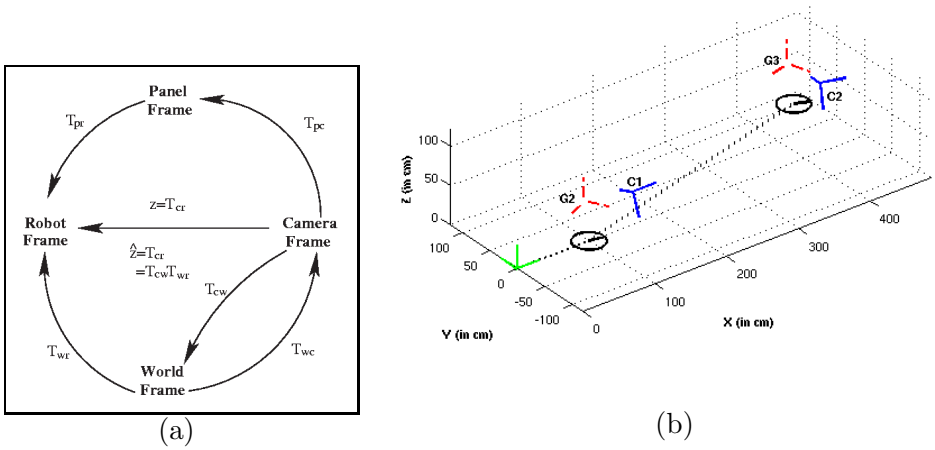
The nonlinear optimization procedure used for camera calibration [4] warrants a brief discussion. A camera is a projective device, mapping information about the 3-D world onto a 2-D image plane. A point in the world $M = [X, Y, Z, 1]^T$ is mapped to pixel $m = [u, v, 1]^T$ in the image, under the following equation:

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

where s is the arbitrary scale factor of the projective equation. 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. The T matrix is a homogeneous transformation made up of a 3×3 rotation matrix R and translation vector t of length 3, 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 relative position of the calibration target 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 nonlinear optimization which includes polynomial distortion parameters. It will be important for our further discussion to mention what Zhang [4] calls "degenerate configurations" where additional views of the calibration target do not provide additional information for calibration. The strongest result given is that any two calibration planes which are parallel to each other do not provide sufficient information for calibration. The intuition here is that the rotation matrix R is used to produce constraints on the intrinsic parameters; as the rotation matrices for parallel planes are linearly dependent, they produce an under-constrained system. To avoid this situation, several different local motion strategies, used to obtain an adequate set of images, are discussed in Section 4.

In conclusion, detecting a set of images of the robot-mounted target and then detecting the grid pattern from the corners of the ARTag markers provide enough information to extract the camera intrinsic parameters and then calculate the extrinsic parameters. The extrinsic parameters of the camera provide an estimate of the camera pose relative to the robot. 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.



(a) (b)

Fig. 4. (a) Measurement Coordinate Frame Transformations. (b) Coordinate frames for the world (origin at [0,0,0]), robot (denoted by a circle and a line for the orientation), target grid (dashed lines G2,G3) and camera (solid lines C1,C2). The trajectory of the robot is marked by a dotted line.

3.2 Six-DOF EKF

The previous section described a method for obtaining an estimate of the camera position relative to the robot through the extrinsic parameter matrix T. These measurements 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 a complete discussion of angle representations, see [33].

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 corrects the robot and the camera poses based on the measurements from the calibration process. Since only the robot moves during the propagation phase, the equations are identical to those used in previous work [22]. The state vector and the covariance matrix are updated as:

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

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

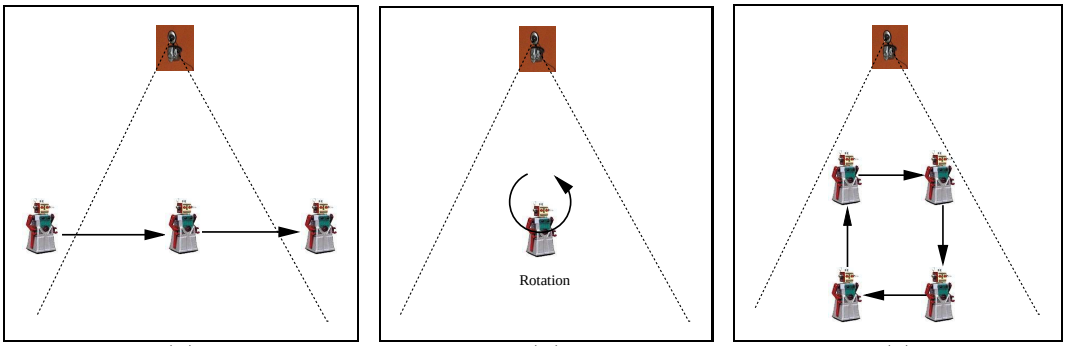
where F obtained by linearizing the nonlinear propagation function $f(X, u)$ where u are control actions, and C_u is a matrix representing odometry error. For the update phase, the measurement equation is a nonlinear expression of the state variables so we must again linearize before using the Kalman 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. In general, any two coordinate frames are related by a transformation matrix as follows [33]:

$$^{b}_{a}T = \begin{bmatrix} ^{b}_{a}R & ^{b}P_{a_{orig}} \\ 0_{1x3} & 1 \end{bmatrix} \tag{4}$$

In this case, the transformation expresses frame a in coordinates of frame b. $^{b}_{a}R$ is the 3×3 rotation matrix which represents the orientation of a as seen in b. $^{b}P_{a_{orig}}$ gives the translation of the origin of frame a in coordinates of frame b. Going back and forth between transformation T and roll, pitch, and yaw angles in state vector X is a simple process [33]. The EKF will only deal with state vectors, but the transformation matrices are used in the following to derive the measurement equations.

The measurement update equation is derived below. The calibration process estimates the extrinsic parameters which represent the calibration panel in the camera frame, that is $^{C}_{P}T$. Since the panels are rigidly attached to the robot, the transformation between the two, namely $^{P}_{R}T$, is easily measured and treated as a constant throughout the procedure. When a new measurement arrives, it can immediately be used to relate the camera and the robot coordinates by $^{C}_{R}T = ^{C}_{P}T\ ^{P}_{R}T$. This is the measurement z. Next, the measurement is expressed in terms of the filter states X_r and X_c. As mentioned, these state vectors are used to get the transformations for the robot and the camera in world coordinates: $^{W}_{R}T$ and $^{W}_{C}T$. Fig. 4 illustrates the relationships between the EKF state variables and the information obtained from camera calibration which jointly form the measurement equation:

$$\begin{aligned} z_{measured} = {}^{C}_{R}T = {}^{C}_{W}T\,{}^{W}_{R}T = {}^{W}_{C}T^{-1}\,{}^{W}_{R}T &= \begin{bmatrix} ^{W}_{C}R^T & -^{W}_{C}R^T\,^{W}_{C}P \\ 0 & 1 \end{bmatrix} \begin{bmatrix} ^{W}_{R}R & ^{W}_{R}P \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} ^{W}_{C}R^T\,^{W}_{R}R & ^{W}_{C}R^T(^{W}_{R}P - ^{W}_{C}P) \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{5}$$



(a) (b) (c)

Fig. 5. Sample trajectories for local calibration: (a) translation; (b) rotation, (c) square.

Equation 5 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:

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K(z - h(\hat{X}_{k|k-1})) \tag{6}$$

$$P_{k|k} = \left[I - KH^T\right] P_{k|k-1} \tag{7}$$

$$K = P_{k|k-1}H(HP_{k|k-1}H^T + C_\omega)^{-1} \tag{8}$$

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 nonlinear optimization. Ideally, the robot should move in such a way that the resulting image set reduces the combined effect of all these error sources and gives the most accurate calibration possible. There are several sources of information on how to approach this task. As mentioned previously, Zhang [4] showed that it is essential to avoid having only parallel planes and Tsai [18] discussed heuristics for obtaining images to calibrate a manipulator system. These heuristics included having the camera lens center near the calibration block, and maximizing the rotation angle between subsequent images. In addition to the calibration error reduction, the accumulated odometric error is another important factor for the overall accuracy of the system as it increases the uncertainty of our EKF estimate. As

such, schemes that require excessive robot motion to achieve good calibration should be avoided.

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 error buildup:

Stationary: The robot moves in front of the camera 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 field of view (FOV) with only a single calibration panel visible always at the same angle. This is a degenerate case and did not produce good calibration.

Multi-Panel Translation-only: The robot translates across the camera FOV with two calibration panels visible. This provides two non-parallel planes for calibration and accumulates a minimal amount of odometry error (see Fig. 5a).

Rotation-only: The robot rotates in place in the center of the camera FOV allowing the panels to be detected at a variety of angles (see Fig. 5b).

Square Pattern: The robot follows a square-shaped path in front of the camera, alternating translation and rotation by 90 degrees. This forms a square with 4 corners. At each corner, the robot has two poses with perpendicular orientation. Since a large portion of the image is covered and there is variation in the detected panel orientation and depth, this method achieved good calibration accuracy. However, the combination of rotation and translation caused large odometry error (see Fig. 5c).

5 Global Exploratory Trajectories

While performing the mapping process described in Section 3, the robot has a partially constructed map, and must travel into previously unvisited territory in order to add new cameras to this map. If the robot were to continually move into unexplored regions, it would be able to cover its environment quickly at the cost of accumulating a large amount of uncorrected odometry error. The robot can slow this error buildup by periodically returning to regions of the map that have already been visited, so its position can be corrected by camera measurements. This behavior will be referred to as "relocalizing". This relocalization behavior will allow for mapping with lower uncertainty, but will require the robot to travel farther in order to cover the space. This describes a trade-off which will be present in any exploration system; covering the space with small distance traveled and mapping with low uncertainty are conflicting goals.

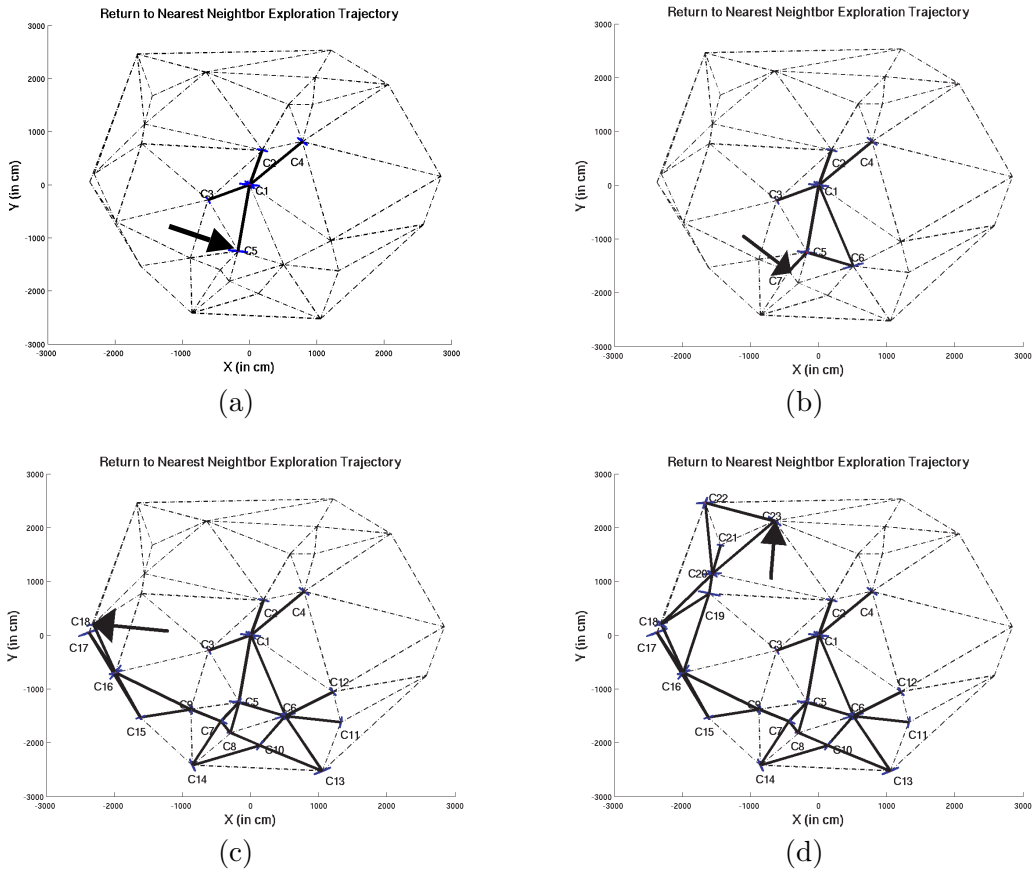


Fig. 6. The progress of the Return-to-nearest exploration strategy while exploring a random graph is shown at 4 intermediate steps progressing left to right and top to bottom. The arrow points to the last node visited by the robot.

In view of the distance and uncertainty trade-off, we consider several specific static policies that typify the behavioral extremes with respect to which most other mechanisms can be described.

Depth-first: The robot always moves into unexplored territory, never relocalizing. This strategy provides coverage of the environment with minimal distance traveled, but the uncertainty of the robot position grows quickly.

Return-to-Nearest: The robot alternates between exploring a new camera position and relocalizing at the nearest previously explored camera. For example, Fig. 6 shows several stages of the exploration process. The ability to relocalize accurately depends on the uncertainty of the nearest camera only, which might not be mapped as accurately as cameras which are farther from the robot. However, only regressing by one camera at a time means the extra distance traveled is minimal.

Return-to-Origin: The robot alternates between exploring a new camera position and returning to the first camera it mapped, which has the lowest uncertainty. This strategy allows extremely good relocalization, but means

the robot must travel a large distance between each newly explored camera.

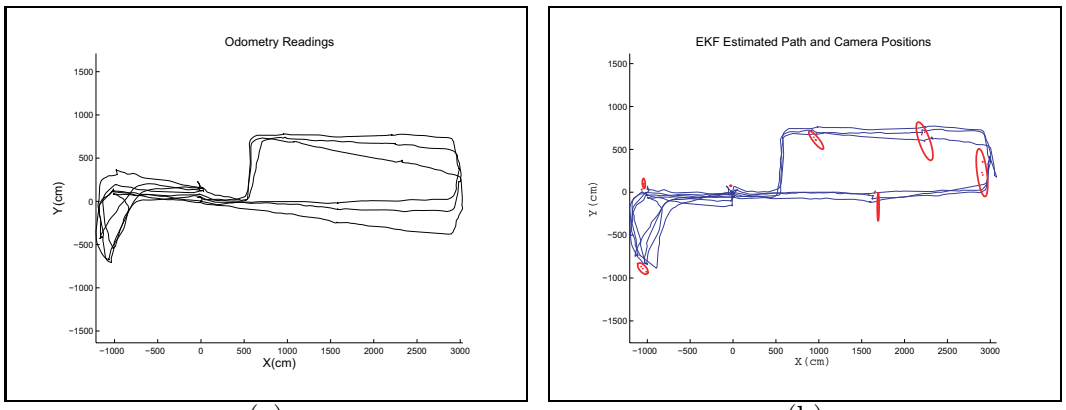
These three strategies do not capture the full range of possibilities, and they provide neither flexibility nor adaptation to the environment. However, they are presented as an initial study of the effects of different strategies on the resulting maps and the robot's ability to navigate in the environment once mapping is completed. Developing adaptive strategies which provide a parameter to weigh the effects of distance traveled and uncertainty in mapping is a topic worthy of future work. Section 6.3 presents comparison of the three simple algorithms presented here.

6 Experimental Results

Three separate sets of experiments were conducted using the camera sensor network (see [24] for a detailed description of the experimental setup) which dealt with the mapping, calibration and planning aspects of our system. First, 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. Second, five different local motion strategies were examined with respect to the resulting intrinsic parameters and the position accuracy. Finally, a series of simulated environments was used to examine the effect of global planning strategies. For all simulated results, we used odometric noise values determined experimentally in our laboratory [34].

6.1 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 consisted of a rectangular loop and a triangular loop connected by a hallway with length approximately 50 m. A Nomadics Scout robot mounted with a target with six calibration patterns was used to perform the calibration and mapping procedure described in Section 3. The robot traversed the two loops of 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. In these tests 30 images were collected for calibration, which took less than 1 minute per camera node to collect. The OpenCV computer vision library was used for camera calibration [35]. Fig. 7a,b show the odometry path estimate and the path corrected by observations from the cameras.



(a) (b)

Fig. 7. (a) Odometry Readings for Hallway Path. (b) EKF Estimate of the Hallway Path. Estimated camera positions with uncertainty ellipses (in red where colour is available).

It is visually clear (Fig. 7a,b) that the use of camera measurements was able to correct for the buildup of odometry error relatively well. However, there are some regions where the filtered path is still a very rough approximation due to large distances between cameras. These distances are traveled without correction of the odometry error. This is most obvious on the far right of the image during the 3rd loop where there is a very noticeable discontinuity in the filtered path as the large odometry error is corrected by a camera measurement. Since the current system does not provide a means for odometry correction between the camera fields of view, this type of result is unavoidable where large odometry errors occur. Solutions include placing the cameras much closer together to limit the size of the unobservable regions, performing SLAM with another sensor such as sonar or laser to provide a complementary approach, or adopting a smoothing technique. We will leave further discussion of these possibilities for the conclusions.

No ground truth data was collected for this experiment due to lack of available accurate measuring devices over this size of environment. In previous work, in a similar experiment over a 15 m environment, the camera positions were found to deviate from the true positions by 2.11 cm on average with standard deviation 1.08 cm [34].

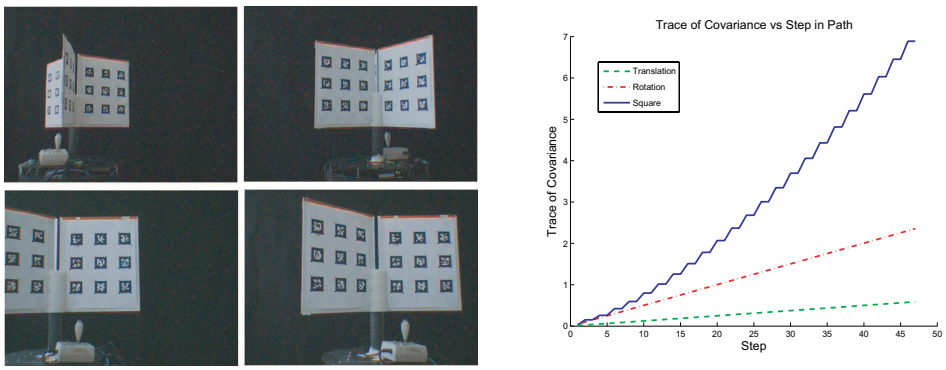
6.2 Local Calibration Paths

A second 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 the same robot and calibration panel as the previous experiment but using only a single camera.

The 5 strategies were performed for 10 trials, each with $n = 30$ 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. One Panel Translation-only is omitted due to divergence. Deviations are with respect to the mean, ground truth error is not provided.

Path	Mean Values				Standard Deviation (%)			
	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
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

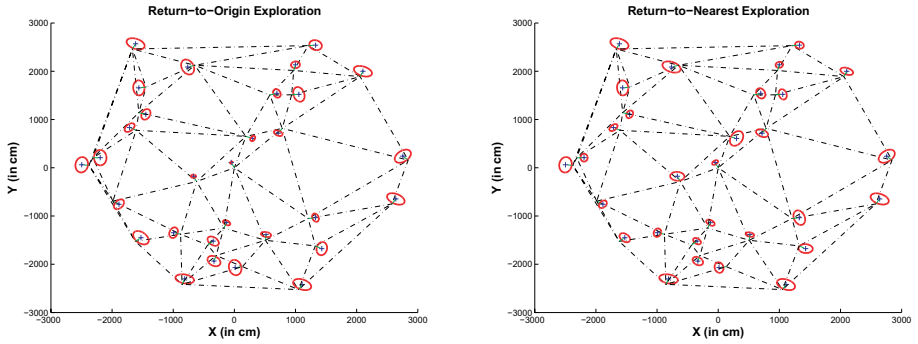


(a) (b)

Fig. 8. (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, for all the other strategies, the mean parameter estimates are not statistically significantly different.

To examine the difference between odometry buildup among the different paths, each of the three paths which involved motion was simulated using an EKF (the stationary approach clearly does not build any odometry error). To ensure a fair comparison, the step size in the Translation-only method was set equal to the side length of the square pattern (8 cm each) and the angle step in the Rotation-only method was set to 90 degrees. This meant that the square pattern translated half of the distance of Translation-only and rotated



(a) (b)

Fig. 9. Camera uncertainty ellipses after mapping completed using strategy: (a) Return-to-Origin and (b) Return-to-Nearest

half the angle of Rotation-only. Fig. 8(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. We must note that the relative slopes in this figure are influenced by the choice of odometry error covariances in the EKF, but that realistic values established through previous experimentation were used [34].

6.3 Exploration Trajectories

The heuristic exploration trajectories discussed in Section 5 were examined in simulation to compare their effect on uncertainty buildup and distance traveled. Two different classes of simulated environments were chosen. The first was the class of uniform planar graphs with a dense set of edges, produced by triangulation. This type of environment allowed for ease in creation of various sizes of environment and densities of cameras. The second environment was based on a sample floor-plan image of a hospital environment obtained as part of the Player/Stage system [36]. In this environment the walls and obstacles prevent the formation of a dense set of edges, and paths between cameras must be much less direct.

On the first environment, the uniform graphs, each exploration trajectory was performed on the same thirty node instance to provide direct comparison. Figure 9 shows final 3σ camera uncertainty ellipses after exploration has completed for both of the relocalization strategies presented, but does not include the Depth-first strategy since the final uncertainty was over twelve times greater than the other strategies. Table 2 summarizes the numerical results for all 3 strategies in terms of area of uncertainty and distance traveled. The Return-to-Nearest strategy produced approximately twenty five percent less area of uncertainty at the end of exploration than the Return-to-Origin strat-

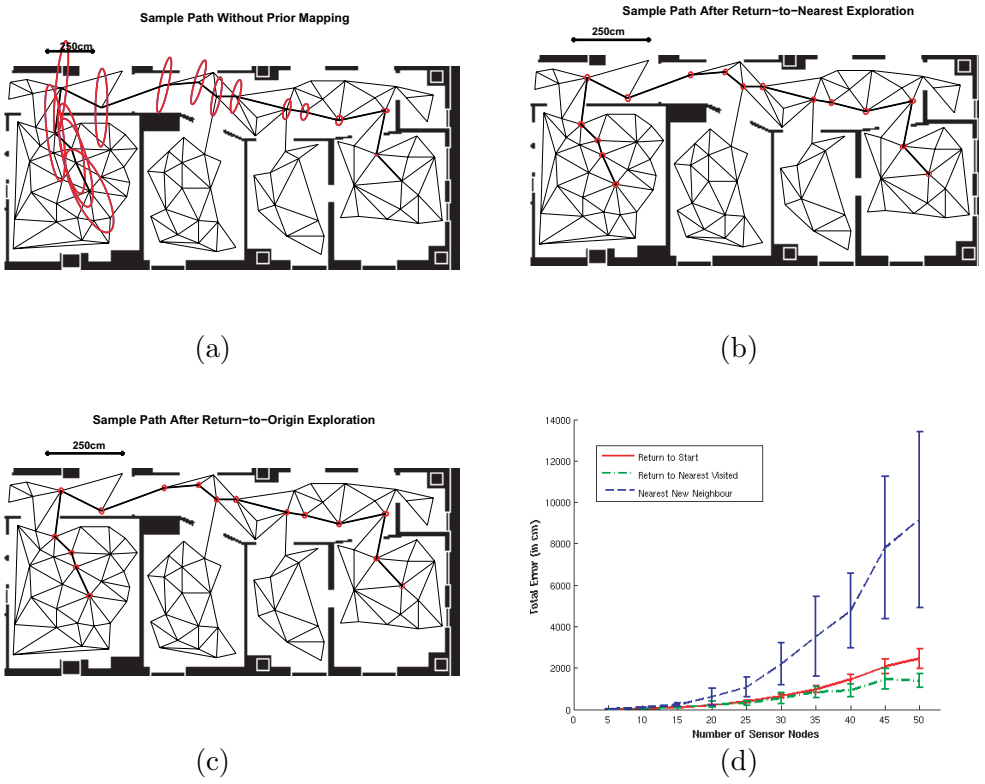


Fig. 10. Uncertainty ellipses along a sample path after (a) no exploration, (b) Return-to-Nearest, and (c) Return-to-origin. (d) The error buildup as graph size increases for the three heuristic exploration strategies.

egy. However, in terms of distance traveled, the Return-to-Nearest strategy required twice the distance of the Depth-First and half the distance of the Return-to-Origin strategies to cover the environment. These results demonstrate that the Depth-First strategy allows uncertainty to increase unchecked, Return-to-Origin provides relocalization, but at the cost of excessive distance traveled, and Return-to-Nearest provides relatively good results for both criteria.

In addition to the single thirty node sample graph, the three exploration strategies were each executed on a large set of random graphs ranging from five to fifty camera nodes to confirm that the previous results generalize over many graphs. For this experiment, fully connected random graphs were used instead of triangulations. Fig. 10d shows the average maximum, minimum and mean error over ten trials for each size. The Depth-First strategy clearly allows error to accumulate at an ever increasing rate as the size of the environment grows. The two relocalization strategies both limit error buildup effectively, with Return-to-Nearest producing slightly lower error in the larger graphs. This result shows that the relative ordering of exploration strategies in terms of uncertainty buildup does indeed generalize over graphs of various sizes and

Table 2
The accumulated uncertainty and the total distance traveled for the three exploration strategies applied to two environments.

	Hospital Origin	Hospital Nearest	Hospital Depth 1^{st}	Random Origin	Random Nearest	Random Depth 1^{st}
$3\ \sigma$ (cm^2)	0.524	0.482	6.065	0.209	0.114	3.05
Dist.(m)	2125	322	115	1210	578	243

densities. The three exploration strategies were performed again in the hospital environment. The ability of the maps to aid navigation was evaluated based on an example path of length of approximately 18 m. Table 2 summarizes the numerical results and shows the same relative ordering in terms of uncertainty and distance as for the random graph. Fig. 10 shows the same path executed three times: once for each of the two strategies involving relocalization and once in a previously unexplored network. The uncertainty at the end of the path for the Return-to-Origin strategy was slightly larger than the Return-to-Nearest despite the fact that the robot traveled over six times as far during exploration. This suggests the extreme buildup in distance traveled for the Return-to-Origin strategy as the size of the environment grows does not produce significant corresponding gain in accuracy. One reason for this effect is likely that nearby landmarks in an EKF have much more effect on each other than those which are far apart. This means that relocalizing very accurately at the origin is not able to improve the estimation accuracy at distant cameras.

7 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 environment, there are several indications that a more sophisticated mapping method would be preferable. Because the environment has large stretches without cameras to provide ob-

servations, filtering alone will not be able to correct entirely for the odometric error accumulated in these areas. It will likely be preferable to adopt a filtering and smoothing method which will allow for better correction of paths in regions with few observations. Also, since we expect to build large odometry errors before seeing a camera, it is expected that the linearization procedure, which is only a good approximation when errors are small, will be highly inaccurate. This effect will be seen increasingly as cameras are spaced farther apart in the environment, and will eventually cause the EKF to diverge. A nonparametric method such as Particle Filtering might give improved results in this context, since linearization is not necessary for such a technique.

The combination of the camera measurements we study with a SLAM solution based on dense sensor readings has the potential to produce interesting results. Dense SLAM approaches can often be seen to lose global alignment over very large regions. Solutions to this have been forcing the robot to close loops during mapping which helps correct orientation, or to perform expensive post-processing of the data. Our method provides an economical approach to allow the correction of error in the robot orientation, because a measurement from the camera provides a second level of sensing and can be processed much more cheaply than post-processing all of the dense range data. Moreover, because of the information inherent in each camera, there is no data-association problem.

During exploration the robot has to constantly decide between exploring new nodes, returning to well known (low uncertainty) locations, and improving the positional accuracy of explored nodes. In the proposed strategy of Return-to-nearest the robot always chose to return to the closest node. We are currently examining different strategies in which the decision where to go next is calculated based on the robot's uncertainty and the state of the map. From preliminary experiments we noted that approaching a mapped camera from a different direction reduces the overall uncertainty due to the rules of covariance composition. Incorporating this information to the motion planning strategies would improve the accuracy of the produced map.

Acknowledgment

The authors would like to thank Dr. Mark Fiala for his very helpful suggestions and for making his ARTag libraries free for research use. We would also like to thank David Aristizabal Leonard D'Cunha and Robert Mill for the initial construction of the distributed wireless sensor network. In addition, Dimitri Marinakis provided much of the implementation of the communication architecture for the sensor network. The authors would like to acknowledge the generous funding of the Natural Sciences and Engineering Research Council of Canada.

References

[1] O. D. Faugeras, Three-Dimensional Computer Vision, MIT Press, 1993.

[2] J. C. McGlone (Ed.), Manual of Photogrammetry, 5th Edition, American Society of Photogrammetry, 2004.

[3] M. Fiala, Artag revision 1, a fiducial marker system using digital techniques, in: National Research Council Publication 47419/ERB-1117, 2004.

[4] Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11) (2000) 1330–1334.

[5] W. E. L. Grimson, C. Stauer, R. Romano, L. Lee, Using adaptive tracking to classify and monitor activities in a site, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998, pp. 22–29.

[6] D. Woods, S. McNee, J. Davis, A. Morison, P. Maughan, K. Christoffersen, Event template hierarchies as means for human-automation collaboration in security surveillance, in: Human Factors and Ergonomics Society Annual Meeting, Orlando, FL, 2005.

[7] O. Javed, Z. Rasheed, O. Alatas, M. Shah, Knight: a real time surveillance system for multiple and non-overlapping cameras, The fourth International Conference on Multimedia and Expo (ICME 2003).

[8] T. Ellis, D. Makris, J. Black, Learning a multicamera topology, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Nice, France, 2003, pp. 165–171.

[9] D. Estrin, D. Culler, K. Pister, G. Sukatme, Connecting the physical world with pervasive networks, IEEE Pervasive Computing 1 (1) (2002) 59–69.

[10] D. Marinakis, G. Dudek, D. Fleet, Learning sensor network topology through monte carlo expectation maximization, in: Proc. of the IEEE International Conference on Robotics & Automation, Barcelona, Spain, 2005, pp. 4581–4587.

[11] M. Batalin, G. Sukhatme, M. Hattig, Mobile robot navigation using a sensor network, International Confernce on Robotics and Automation.

[12] R. Kurazume, S. Hirose, Study on cooperative positioning system - optimum moving strategies for cps-iii, in: IEEE (Ed.), Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 4, 1998, pp. 2896–2903.

[13] I. M. Rekleitis, G. Dudek, E. Milios, Multi-robot collaboration for robust exploration, in: Proceedings of International Conference in Robotics and Automation, San Francisco, USA, 2000, pp. 3164–3169.

[14] S. I. Roumeliotis, G. A. Bekey, Distributed multirobot localization, IEEE Transactions on Robotics and Automation 18 (5) (2002) 781–795.

[15] A. Howard, M. J. Mataric, G. S. Sukhatme, Localization for mobile robot teams using maximum likelihood estimation, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, EPFL Switzerland, 2002, pp. 434–459.

[16] R. Y. Tsai, Synopsis of recent progress on camera calibration for 3-d machine vision, The Robotics Review (1989) 147–159.

[17] O. Faugeras, Q. Luong, The Geometry of Multiple Images, The MIT Press, 2001.

[18] R. Y. Tsai, R. K. Lenz, Real time versatile robotics hand/eye calibration using 3d machine vision, IEEE International Conference on Robotics and Automation.

[19] R. Y. Tsai, 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 (1987) 323–344.

[20] I. Poupyrev, H. Kato, M. Billinghurst, Artoolkit user manual, version 2.33., Human Interface Technology Lab, University of Washington.

[21] M. Fiala, Vision guided robots., in: Proc. of CRV'04 (Canadian Confernence on Computer and Robot Vision, 2004, pp. 241–246.

[22] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics, Autonomous Robot Vehicles (1990) 167 – 193.

[23] J. J. Leonard, H. F. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation 7 (3) (1991) 376–382.

[24] I. M. Rekletis, G. Dudek, Automated calibration of a camera sensor network, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton Alberta, Canada, 2005, pp. 401–406.

[25] B. Kuipers, Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierachy of spatial representations, Robotics and Autonomous Systems 8 (1991) 46–63.

[26] G. Dudek, M. Jenkin, E. Milios, D. Wilkes, Robotic exploration as graph construction, Transactions on Robotics and Automation 7 (6) (1991) 859–865.

[27] S. Koenig, Y. Smirnov, Graph learning with a nearest neighbor approach, in: Proceedings of the Ninth Annual ACM Conference on Computational Learning Theory (COLT), 1996, pp. 19–28.

[28] I. M. Rekleitis, V. Dujmović, G. Dudek, Efficient topological exploration, in: Proceedings of International Conference in Robotics and Automation, Detroit, USA, 1999, pp. 676–681.

[29] B. Yamauchi, Frontier-based exploration using multiple robots, in: Proceedings of the Second International Conference on Autonomous Agents (Agents '98), Minneapolis, MN, 1998, pp. 47–53.

[30] H. Choset, J. Burdick, Sensor based planning, part ii: Incremental construction of the generalized voronoi graph, in: Proc. of IEEE Conference on Robotics and Automation, IEEE Press, Nagoya, Japan, 1995, pp. 1643 – 1648.

[31] I. M. Rekleitis, G. Dudek, E. Milios, Multi-robot collaboration for robust exploration, Annals of Mathematics and Artificial Intelligence 31 (1-4) (2001) 7–40.

[32] R. Sim, N. Roy, Global a-optimal robot exploration in slam, in: International Conference on Robotics and Automation, 2005, pp. 661 – 666.

[33] J. J. Craig, Introduction to Robotics, Mechanics and Control, Addison-Wesley, 1986.

[34] I. M. Rekleitis, A particle filter tutorial for mobile robot localization, Tech. Rep. TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 3480 University St., Montreal, Québec, CANADA H3A 2A7 (2004).

[35] The opencv computer vision library, http://www.intel.com/research/mrl/research/opencv.

[36] R. T. Vaughan, Stage: A multiple robot simulator, Tech. Rep. IRIS-00-394, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California (2000).

Ioannis Rekleitis is a Visiting Fellow at the Canadian Space Agency and adjunct Professor at the School of Computer Science, McGill University. During 2004 he worked at McGill University as a Research Associate. In 2002 and 2003 he was a Postdoctoral Fellow at the Carnegie Mellon University. He received his Ph.D. in 2002 from the School of Computer Science, McGill University, working in multi-robot collaboration; and his M.Sc. in McGill University in the field of Computer Vision in 1995. He was granted his B.Sc. in 1991 from the Department of Informatics, University of Athens.

His research has focused on space robotics, mobile robotics and sensor networks and in particular in the area of cooperating intelligent agents with application to multi-robot cooperative localization, mapping, exploration and coverage. His interests extend to computer vision and machine learning. His current work also focuses on developing software architectures for autonomous intelligent systems. He has authored or coauthored more than 30 journal and conference papers, in the above areas. He has helped in the organization of several conferences as program chair (twice), program committee member, reviewer, and as special editor for a journal.

David Meger is a Master's student in Computer Science at McGill University's Mobile Robotics Lab in the Centre for Intelligent Machines. His research interests are in intelligent behaviours for mobile robots, vision based robotics and sensor networks.

Gregory Dudek is Director of the McGill University Research Center for Intelligence Machines, and a faculty member with the School of Computer Science. The McGill's Research Center for Intelligent Machines is a 20 year old inter-faculty research facility. In 2002 he was named a William Dawson Scholar. He also directs the McGill Mobile Robotics Laboratory.

He is active on the organizing committees for many major scientific conferences on robotics and has published over 150 research papers on subjects including robot position estimation, visual object description and recognition, robotic navigation and map construction, distributed system design and biological perception. This includes a book entitled "Computational Principles of Mobile Robotics" co-authored with Michael Jenkin and published by Cambridge University Press.