Automated Calibration of a Camera Sensor Network*

Ioannis Rekleitis
Canadian Space Agency
6767 Route de l'Aroport Saint-Hubert, Quebec, Canada J3Y 8Y9
yiannis@cim.mcgill.ca

Gregory Dudek
Center for Intelligent Machines
McGill University, Montreal, Quebec, Canada
dudek@cim.mcgill.ca

***Abstract*— In this paper we present a new approach for the on-line calibration of a camera sensor network. This is the first step towards fully exploiting the potential for collaboration between mobile robots and static sensors sharing the same network. In particular we propose an approach for extracting the 3D pose of each camera in a common reference frame, with the help of a mobile robot. The camera poses can then be used to further refine the robot pose or to perform other tracking tasks. The analytical formulation of the problem of pose recovery is presented together with experimental results of a six node sensor network in different configurations.**

***Index Terms*— Sensor Networks, Cooperative Localization, Camera Calibration.**

I. Introduction

As cameras are becoming common in public areas they are becoming a powerful information source that can be exploited by autonomous vehicles. Cameras can be exploited (or are already in use) for environmental observations (*e.g.* near-shore monitoring [1]), surveillance (indoor and/or outdoor), highway traffic monitoring and control, in intelligent spaces [2], *etc*. In most cases though, the cameras are placed in locations that are convenient for human operators with little consideration for their use by an autonomous mobile robot. Furthermore, the exact location and orientation (3D pose) of these cameras usually is not even known with respect to the vehicle's frame of reference. In some cases, even the coarse position of such cameras may not be known (i.e. during the setup process). By establishing a common reference coordinate system for all the cameras in a network, as well as for mobile robots moving in the same environment, we can leverage the respective advantages of both the robots and the emplaced sensors. With the introduction of mobile robots in a variety of roles (cleaning, vacuuming, surveillance, environmental condition monitoring) even preexisting surveillance cameras emplaced for other purposes can be utilized to improve robot pose estimation and contribute additional sensory input. As sensor networks (camera based or not) are being used for other tasks, our methodology can be employed for rapid deployment and automatic calibration of the system and also to provide tune-ups and recalibration in the face of changes and/or failures.

In particular herewith we present our approach for collaboration in a multi-sensor/multi-robot network. In the current experiments reported here, a network of wireless and wired cameras dispersed in the environment (two adjoined laboratories) in unknown positions with no overlapping fields of view is employed (see Fig. 1). A mobile robot equipped with a calibration pattern navigates through free space and stops every time it passes through a camera's field of view. After obtaining a number of images the internal and the external parameters [3] of the camera are estimated. Finally, the 3D pose of the camera is recovered as a function of the pose of the robot for each of the images taken.

Fig. 1. Three wireless camera nodes and the mobile robot with the target.

In our approach we consider the following illustrative scenario: One or more service mobile robots moving in an indoor environment *e. g.* an art gallery. A network of security cameras is already in place and they are connected to the same network as the robots. The robots move through the environment performing their tasks (cleaning, maintenance, monitoring, surveillance, *etc*.). The robots communicate and use the cameras in order to augment their environmental awareness and also to reduce their positioning uncertainty. The first step toward the realization of the above scenario is to place the cameras in the same frame of reference as the mobile robots. To do that the camera's position and orientation are estimated in the robot's world coordinate frame. In addition, the robots can identify and delineate the field of view of each camera and add these regions to the map of the environment. Collaboration among different mobile robots can be improved by sharing these regions and situating every robot in a common frame of reference. Every time a robot enters an area observed by a camera it can update both its pose and the pose of the camera using standard sensor fusion techniques (*e.g.* a KF or a Particle Filter estimator).

In the next Section we discuss related work. Section III presents an overview of our approach and the necessary coordinate transformations required for the camera localization. Section IV presents first an overview of our experimental setup and then experimental results for different robot trajectories in the laboratory. Finally, Section V contains concluding remarks and discussion of future work.

*This work was partially supported by NSERC

II. Background

In the standard approach to mapping, or mapping and localization (i.e. SLAM) a mobile robot equipped with a sensor moves through the environment registering the position of landmarks in the environment [4]. In our work the landmark is placed on the mobile robot and the sensors are distributed in fixed positions in the environment. Our approach is suited for *smart environments* where the sensors are already in place, or for the rapid deployment of mixed-capability sensor networks which can auto-calibrate even though the cameras are far from one another. An additional advantage of our approach is that since our landmark is placed on the robot there is no alteration or defacement of the environment due to the placement of artificial landmarks, and reduced sensitivity to vandalism, but we still obtain the efficiencies of manually engineering the environment.

The problem of camera calibration is a well established area of computer vision [3], [5]. In general, camera calibration involves both the estimation of intrinsic and extrinsic parameters associated with the camera, and is most readily accomplished via the use of synthetic targets. While various target configurations have been considered, planar patterns are typically the most convenient. In our approach we use the algorithm implemented by Bouguet [6] based on previous work in [7], [8].

Several authors have considered the problem of using cameras with proximal or overlapping fields of view to estimate inter-camera geometries [9]–[11]. Makris *et al*. [10] proposed learning the topology of the sensor network using observations of moving targets. Marinakis *et al*. [12] used a Monte-Carlo Expectation Maximization approach in order to infer the topology of a sensor network with minimal-information observations. The above work placed emphasis on estimating the connectivity relationships between the regions seen by different cameras, as opposed to the specific coordinate transformations between camera frames. The problem of managing large volumes of data in a "Visual Sensor Network" by controlling the information flow and distributing some of the computer vision among the nodes (depending on computing power) is discussed in [13].

More broadly, the problem of tracking or observing a robot moving using a camera network is related to that of tracking and observing people [14], [15]. While in principle one can use people tracking to estimate inter-camera transformation parameters, the problem is complicated by the issues of recognition, tracking, rigid deformation and motion estimation. Since we can rely both on the cooperation of our targets, as well their internal motion estimates, our problem is substantially more tractable.

Several authors have also considered the issues that arise when using a sensor network with known sensor positions to track a robot as it moves and refines its pose estimate [16], [17]. In addition, Batalin *et al*. also consider the iterative update of both the robot and sensor positions based on laser range data in two dimensions (although results in [18] are based on simulation data). Moore *et al*. [19] proposed a linear-time algorithm for localizing the nodes in a sensor network using range data. Other related work is that of Howard *et al*. which uses energy minimization to optimize the poses in a network of robots [20]. In this work we deal explicitly with the 3D problem, begin with unknown camera positions, use only passive vision-based sensing, and only impose weak visibilty requirements.

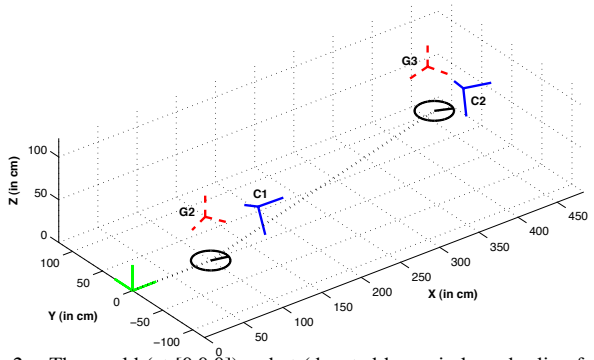


Fig. 2. The world (at [0,0,0]), robot (denoted by a circle and a line for the orientation), target grid (dashed lines G1,G2) and camera (solid lines C1,C2) coordinate frames. The trajectory of the robot is marked by a dotted line.

III. Cooperative Localization in a Multi-Robot/Multi-Sensor Network

In this section we outline our approach to allowing the robot to calibrate the sensor network. Note that we neglect the problem of robot detection by the cameras since, by virtue of the cooperative target, the detectability of the robot can be assured (prior work on the synthesis and detection of specially constructed targets includes but is not limited to [21]–[23]).

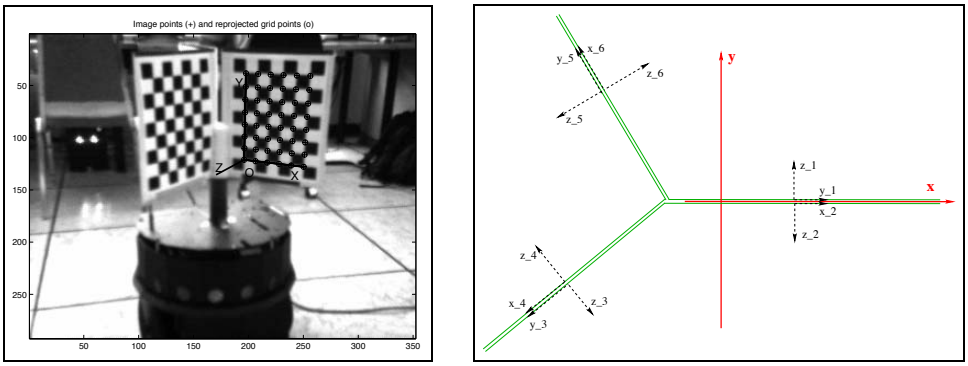
A. Coordinate System Transformation

In order to compute the camera positions from the robot's path, we compute a transformation from the world coordinate frame to the target and then to the frame of the camera, by way of the robot. Fig. 2 presents the trajectory of the robot (dotted line) as it stops in front of two cameras. The world coordinate frame is located at [0,0,0]; the robot is displayed by a circle with a line indicating the forward direction; over the robot one of the calibration grids attached to the three plane target is drawn in dashed lines (see Fig. 3 for the target construction); the camera coordinate frames are also drawn in solid lines.

The camera frame in world coordinates (O_w^c) can be computed analytically using Equation 1.

$$\begin{aligned} O_g^c &= \begin{bmatrix} \mathbf{R}_{gc}^T & -\mathbf{R}_{gc}^T\mathbf{T}_{gc}^T \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix} O_c^c \\ O_r^c &= \begin{bmatrix} \mathbf{R}_{gr} & \mathbf{T}_{gr}^T \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix} O_g^c \\ O_w^c &= \begin{bmatrix} \mathbf{R}_{rw} & \mathbf{T}_{rw}^T \\ 0\ \ 0\ \ 0 & 1 \end{bmatrix} O_r^c \end{aligned} \tag{1}$$

where $O_c^c = [0001]^T$ is the origin of the camera coordinate system in homogeneous coordinates. $\mathbf{R}_{gc}$ and $\mathbf{T}_{gc}^T$ are the extrinsic parameters returned from the camera calibration algorithm and represent the coordinate transformation from the target grid to the camera. The matrices $\mathbf{R}_{gr}$ and $\mathbf{T}_{gr}^T$



(a) (b)

Fig. 3. (a) An image from one of the cameras showing most of the robot and two faces of the target can be seen. The calibration grid coordinate system as detected by the camera is overlaid to the grid on the right side (red). (b) A top view of the target 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 frame of the target.

specify the coordinate transforms between the origin of the grid-based coordinate system (see Fig. 3a) and the robot centered coordinate system; they are manually estimated for each one of the six grids attached to the robots (see Fig. 3b for a top view of the arrangements of the grid and the robot coordinate frames, see also section IV-A). Finally, $\mathbf{R}_{rw}$ and $\mathbf{T}_{rw}^T$ are calculated from the robots pose $P_r = [X_r, Y_r, \theta_r]^T$ (X_r, Y_r, θ_r in world coordinates) as follows:

$$\mathbf{R}_{rw} = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) & 0 \\ \sin(\theta_r) & \cos(\theta_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\mathbf{T}_{rw} = [X_r Y_r 0]^T \quad (3)$$

The (x, y, z) coordinate frame associated with the calibration grid is displayed in Fig. 3a. In this instance the y-axis is vertical the x-axis is horizontal and the z-axis is facing away from the grid surface toward the camera. In general the z-axis always faces toward the camera and the x and y axes switch between horizontal and vertical. A top view of the target arrangement can be seen in Fig. 3b.

B. Uncertainty Propagation

The robot pose is modeled using a particle filter. As we saw in the previous section, the pose of the camera is expressed as a function of the pose of the robot. After the robot starts moving, dead reckoning error accumulates and the accuracy of the robot pose deteriorates. We take advantage of the kinematic equations relating the robot and camera frames to compute the camera uncertainties as well.

The accuracy of the pose estimate for every new camera becomes progressively worse as the robot accumulates more odometry error. By employing different exploration strategies (see [24]) the robot could alternate between exploring/mapping new cameras and localizing using the cameras already mapped. The deterioration of the landmark accuracy is a general problem that plagues all SLAM approaches that rely on sparse landmarks and is thus unavoidable. It is worth noting that our approach could be used in parallel with standard SLAM techniques that utilize the range sensors on the robot and thus reduce (but not eliminate) the odometric error.

IV. Experimental Results

A. Experimental Setting

In this paper we use a network of six cameras and modified Super Scout robots (from Nomadic Technologies). To simplify the vision-based detection and calibration sub-problem, a custom made three plane target was mounted on the robot; (as in [25]) on each side of every plane a grid pattern was placed such that from any position around the robot a camera could see one or two grids (see Fig. 3a).

1) Sensor Network: The sensor network was deployed in our lab in a variety of configurations for the experiments reported here. One of our objectives was to develop a sensor configuration that was a compromise between limited cost, portability and computing power. Specifically, we are interested in a network with sufficient processing power per node to accomplish near-real-time vision and sensor fusion, and to permit rapid prototyping and program development. As a result, we have opted for a Linux-based platform based on conventional processor architecture but with an embedded-systems, small-form-factor, motherboard. In ongoing work, we are also examining the integration of these rather capable nodes in a heterogeneous network that also encompasses miniaturized less powerful nodes (e.g. Crossbow motes).

Fig. 4 presents two nodes side by side with the cameras and the wireless bridge on top. Each node consists of a diskless Linux computer with a Celeron/PIII 500MHz processor and 128MB of RAM. A web-cam with a USB interface was selected as an inexpensive, readily available solution. A central computer that provides long-term storage space for each diskless node and coordinates computation is used. In the present experiments, the nodes are connected to the central node using a "star" topology. Connectivity was provided in two different modes, wired and wireless. The maximum length of the wired links was 15m. For the wireless connection we used standard 802.11b/g protocols.

2) Camera Calibration Pattern: A planar checkerboard pattern (a grid) was used for the calibration and localization of the cameras. On each face of the three plane target (six faces) a 7 by 9 grid was attached, each cell of the grid having

Fig. 4. Two sensor node subsystems with a camera and a wireless bridge.

a size of 28.5mm. Two grid patterns can be seen in Fig. 3a attached to two target planes. It is clear that the choice of the grid size directly affect the quality of the calibration. Before performing the experiments with the target mounted robot we used a large grid (13 by 17 squares, see Fig. 5) as a base of comparison for the calibration of the internal parameters.

B. Calibration of the Intrinsic Parameters

A lengthy discussion of camera calibration is outside the scope of this paper we present next a brief overview of the internal parameters of a camera [3]:

- *Focal length*: The focal length in pixels is stored in the 2x1 vector fc.
- *Principal point*: The principal point coordinates are stored in the 2x1 vector cc.
- *Skew coefficient*: The skew coefficient defining the angle among the x and y pixel axes is stored in the scalar α_c.
- *Distortions*: The image distortion coefficients (radial and tangential distortions) are stored in the 5x1 vector kc.

A set of 10-20 images of the large grid of 13 by 17 cells (see Fig. 5) was used to calibrate manually the cameras and to obtain their internal parameters. These parameters are used as a basis for comparison to the intrinsic parameters recovered from the images of the smaller grid mounted on the robot.

Table I presents the internal parameters for the two grids (large for manual and small for automated calibration). Even though the error is higher for the smaller grid it was low enough to compare with the odometric error accumulated.

C. Cooperative Localization of one robot and a Visual Sensor Network

While both the camera configuration and the robot trajectory can be independently modified and both influence the behavior of the overall system, we confine our experiment here to an evaluation of the effect of different robot paths and data acquisition strategies. A variety of experiments was performed in our laboratory for different placements of the sensor nodes and for different trajectories. In the next sections we discuss the localization accuracy of two different methods for exploiting the robot's motion. In particular, the two strategies represent two qualitatively different behavioral extremes.

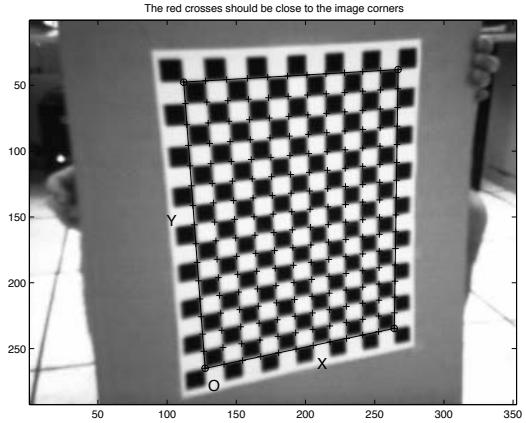


Fig. 5. A large grid (13 by 17 squares) used for calibrating the intrinsic parameters of a sensor node.

TABLE I

TYPICAL CAMERA INTRINSIC PARAMETERS

20 images using the large grid (see Fig. 5). The units are in mm.	
Focal Length:	fc=[476.19427 475.86057]±[1.33583 1.39354]
Principal Point:	cc=[191.13565 138.20564] ±[1.08906 1.31981]
Skew:	$\alpha_c = [0.00000] \pm [0.00000]$
angle of pixel axes	=90.00000 ±0.00000degrees
Distortion:	kc=[0.64126 -2.05749 -0.01727 0.01525 0.0] ±[0.01751 0.17530 0.00229 0.00193 0.0]
40 images using the target from 3 different positions (see Fig. 3).	
Focal Length:	fc=[487.64314 485.92405]±[4.68233 4.15016]
Principal Point:	cc=[185.34096 143.34743] ±[2.01044 3.08005]
Skew:	$\alpha_c = [0.00000] \pm [0.00000]$
angle of pixel axes	=90.00000 ±0.00000degrees
Distortion:	kc=[0.68579 -2.42735 -0.01194 0.00069 0.0] ±[0.02894 0.20148 0.00452 0.00293 0.0]

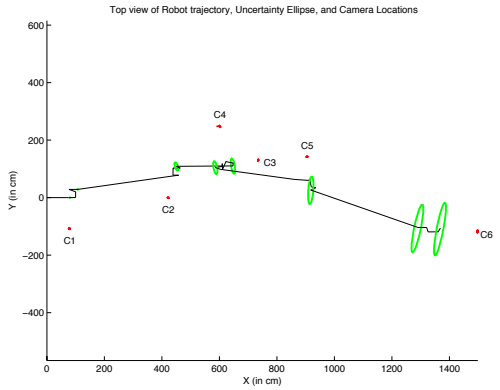


Fig. 6. The trajectory of the robot and the estimated locations of the cameras using the "dance" scenario.

1) Multiple poses per camera - "dance" scenario: The first approach is to attempt to compensate for errors that may arise during image acquisition as a result of imperfections in the lighting, partial occlusion, specularities, geometric distortion, lens distortion and other factors. While these sources of error are difficult to predict or quantify, it is often the case that a new image from a new vantage point may overcome them. This suggests a data acquisition strategy in which the robot executes a short "dance" in front of each camera (in practice, this might also occur while a task was being executed). As this locally observable path is being executed, multiple images are acquired. This allows a more refined estimate of the robots position relative to the camera, but also implies a larger

accumulation of odometry error. Fig. 6 illustrates the effects of this strategy in a sensor network of six nodes.

Table II presents the standard deviation for the six cameras along the x,y,z axis and for the three orientations.

TABLE II

VARIABILITIES OF THE CAMERA POSITION ESTIMATES. THE UNITS FOR $\sigma_x, \sigma_y, \sigma_z$ ARE IN CM; AND FOR $\sigma_{\theta_X}, \sigma_{\theta_Y}, \sigma_{\theta_Z}$ ARE IN DEGREES.

Camera id	σ_x	σ_y	σ_z	σ_{θ_X}	σ_{θ_Y}	σ_{θ_Z}
Camera 1	1.2078	0.7996	0.8099	0.3745	0.3097	0.3133
Camera 2	1.2467	0.8046	0.8587	0.5211	0.5381	0.5495
Camera 3	0.4971	1.4010	1.6303	0.6404	0.2414	0.8013
Camera 4	2.8144	1.5290	1.0550	0.6110	0.6392	0.6452
Camera 5	1.2958	0.6462	0.7289	0.5095	0.3309	0.3362
Camera 6	0.6118	2.3258	1.9296	0.6899	0.2879	0.7997

2) Simple Closed-loop Trajectory: In the second approach the robot traversed a simple path and had its pose estimated by every camera exactly once; an image was taken and the pose of the camera was estimated based on that single image. In the current implementation the robot stopped to allow manual intervention while the image was acquired although, in principle, its trajectory could have continued uninterrupted. Fig. 7 presents the robot path and the camera locations for 6 cameras. Because only a single image was taken the intrinsic parameters of each camera where considered known *i.e.* calculated in a previous experiment using the dance strategy.

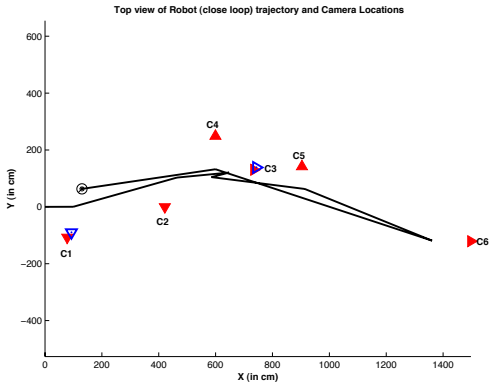


Fig. 7. A close loop trajectory of the robot and the estimated locations of the cameras. Camera 1 and 3 are visited twice.

A Particle Filter [26] was used to model and reduce the odometric uncertainty. Fig. 8 presents the trajectory of the robot together with 3σ uncertainty ellipses derived from the particle distribution. When the robot revisits cameras C3 and C1 (Fig. 8c,e) the observed position of the camera is used to update the particle weights thus reducing the pose uncertainty (see Fig. 8e,f respectively).

The inter-camera distances were measured by hand in the context of the experiments described above and compared with the automated results. The mean error for twelve measurements was 2.11cm with a standard deviation of 1.08cm.

Our results suggest that the calibration of an embedded camera system with a cooperating mobile robot is both practical and efficient. The approach we have examined is simple enough that diagnostic recalibration can be performed regularly. The odometric error for the simple trajectory was smaller as compared to the dance trajectory (less rotations). That can be easily seen by comparing the uncertainty ellipses in Fig. 6 and 8b. Although the simple trajectory makes inter-camera motion estimation easier, the dance trajectory reduces errors from other sources, as noted above. This suggests that in some cases simpler trajectories may be preferable. An in-depth analysis of this trade-off is ongoing, but it appears that each strategy may have situations in which it proves superior.

V. CONCLUSION

In this paper we have examined the use of cooperative moving targets in the calibration of a vision-based sensor network. We show that by using a moving robot we are able to calibrate a network of cameras whose initial deployment was completely unknown. The robot carries a conspicuous calibration target. The camera nodes of the network are based in single-board computers that are capable of local image processing and which communicate via TCP/IP.

By taking advantage of the cooperative behavior of the moving robot, we can obtain inter-camera transformations far more accurate than those that could be obtained using only people tracking or other passive means. Further, we observed that the specific motion control strategy used by the robot has a substantial bearing on the accuracy of the pose estimates. By minimizing the accumulation of odometry error between cameras, the accuracy of the entire network can be optimized. In addition, as with SLAM systems, an iterative update strategy may provide even better final pose estimates.

This suggests several avenues for future work. The incorporation of an active cooperative target in a SLAM-based update scheme is a natural incremental extension of this work. It may also be feasible to select path planning strategies that optimize either the overall network accuracy, or some components of it. The closed-loop selection of robot behavior to tune the network calibration is a problem we are currently exploring.

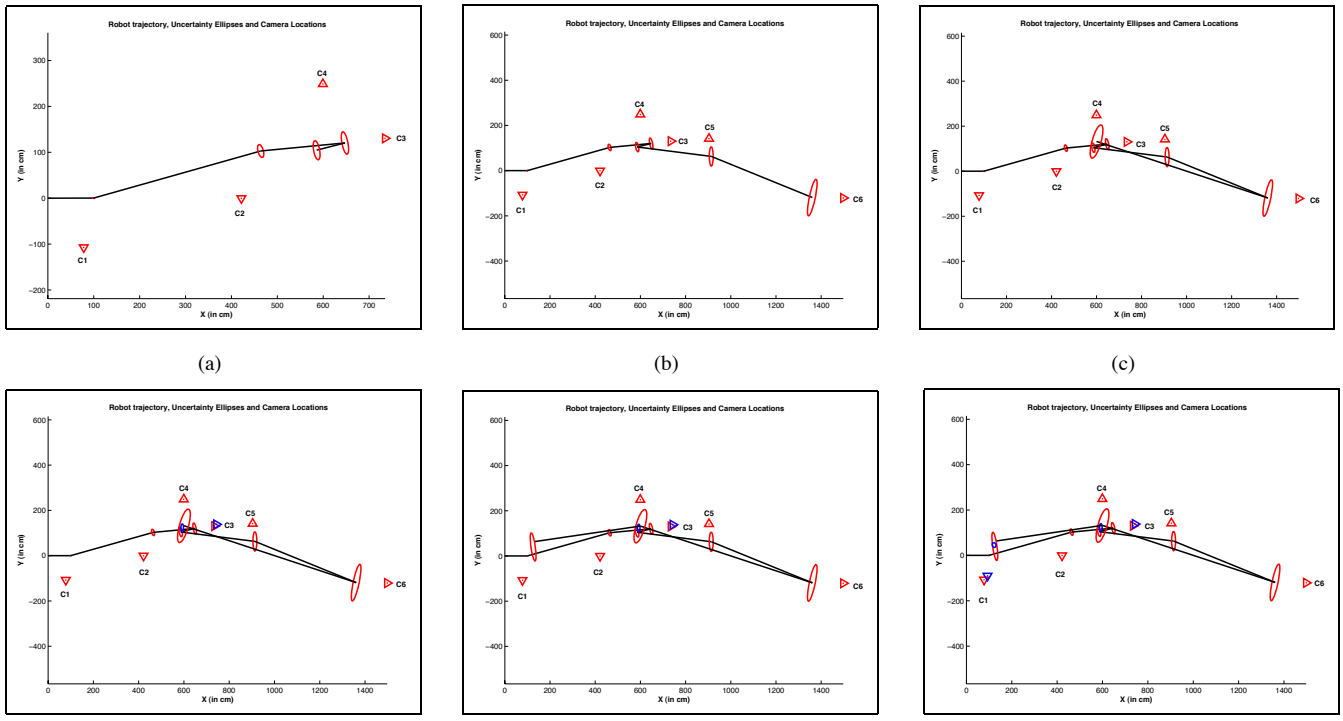
In order to address the noise problems from the camera distortions, spatial quantization and the odometric uncertainty we are currently implementing a 3D probabilistic framework based on an Extended Kalman Filter (EKF) estimator. When the mobile robot revisits an area observed by the camera we could use the prior knowledge to reduce the positioning uncertainty for both the robot and the camera.

ACKNOWLEDGMENT

We would like to thank David Aristizabal Leonard D'Cunha and Robert Mill for the initial construction of the distributed wireless sensor network. Francis Peron and Chris Prahacs provided valuable software and hardware support. Finally, Dimitri Marinakis and Junaed Sattar contributed with valuable insights during the whole project.

REFERENCES

[1] R. Holman, J. Stanley, and T. Ozkan-Haller, "Applying video sensor networks to nearshore environment monitoring," *IEEE Pervasive Computing*, v. 2, no. 4, Oct.-Dec. 2003.



(a) (b) (c)

(d) (e) (f)

Fig. 8. A simple close loop trajectory with uncertainty ellipses. A particle filter was used to model the odometry error and reduce positioning uncertainty when the robot was seen again by a camera. (a) The robot maps the 4th camera (propagation). (b) 6th camera is mapped (propagation). (c) The robot is in front of camera 3 again (propagation). (d) Update of the robot position based on the second estimate from camera 3 (update). (e) The robot returns in front of camera 1 (propagation). (f) Final update using the information from second view from camera 1 (small red ellipse inside the larger ellipse).

[2] J.-H. Lee, G. Appenzeller, and H. Hashimoto, "An agent for intelligent spaces: functions and roles of mobile robots in sensored, networked and thinking spaces," in *IEEE Conf. on Intelligent Transportation System (ITSC 97)*, 1997, pp. 983 – 988.

[3] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1996.

[4] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. on Robotics and Automation*, v. 7, no. 3, pp. 376–382, June 1991.

[5] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v. 22, n. 11, pp. 1330–1334, 2000.

[6] J.-Y. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/, 2004.

[7] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1997, pp. 1106–1112.

[8] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Int. Conf. on Computer Vision*, 1999.

[9] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, v. 2, 1999, p. 252.

[10] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, v. 2. 2004, pp. 205 – 210.

[11] T. Ellis, D. Makris, and J. Black, "Learning a multicamera topology," in *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003, pp. 165–171.

[12] D. Marinakis, G. Dudek, and D. Fleet, "Learning sensor network topology through monte carlo expectation maximization," in *Proc. of the IEEE Int. Conf. on Robotics & Automation*, 2005.

[13] K. Obraczka, R. Manduchi, and J. Garcia-Luna-Aveces, "Managing the information flow in visual sensor networks," in *5th Int. Symposium on Wireless Personal Multimedia Communications*, 2002.

[14] L. S. D. Ismail Haritaoglu, David Harwood, "W4: real-time surveillance of people and their activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 22, no. 8, pp. 809 – 830, Aug. 2000.

[15] R. Morris and D. Hogg, "Statistical models of object interaction," in *IEEE Workshop on Visual Surveillance*, 1998, p. 81.

[16] M. Batalin, G. Sukhatme, and M. Hattig, "Mobile robot navigation using a sensor network," in *IEEE Int. Conf. on Robotics and Automation*, v. 1, 2004, pp. 636 – 641.

[17] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE Int. Conf. on Robotics & Automation*, pp. 165–172.

[18] M. Batalin and G. S. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, v. 26, no. 2, pp. 181–196, 2004.

[19] D. Moore, J. Leonard, D. Rus, and S. Teller., "Robust distributed network localization with noisy range measurements." in *Proc. of the Second ACM Conf. on Embedded Networked Sensor Systems*, 2004.

[20] A. Howard, M. Mataric, and G. Sukhatme, "Relaxation on a mesh: a formalism for generalized localization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, v. 2, 2001.

[21] D. Scharstein and A. Briggs, "Fast recognition of selfsimilar landmarks," 1999. [Online]. Available: citeseer.ist.psu.edu/scharstein99fast.html

[22] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Experiments in sensing and communication for robot convoy navigation," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, v. 2, 1995, pp. 268–273.

[23] I. M. Rekleitis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration," *Annals of Mathematics and Artificial Intelligence*, v. 31, no. 1-4, 2001.

[24] R. Sim, G. Dudek, and N. Roy, "Online control policy optimization for minimizing map uncertainty during exploration," in *Proc of the IEEE Int. Conf. on Robotics and Automation*, v. 2, 2004, pp. 1758– 1763.

[25] I. Rekleitis, G. Dudek, and E. Milios, "Experiments in free-space triangulation using cooperative localization," in *IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 1777–1782.

[26] ——, "Probabilistic cooperative localization and mapping in practice," in *IEEE Int. Conf. on Robotics and Automation*, 2003.