Sensor Localization and Camera Calibration using Low Power Cameras

Dimitrios Lymberopoulos, Andrew Barton Sweeney, and Andreas Savvides

Embedded Networks and Applications Lab, ENALAB
Yale Univerisity, New Haven, CT 06520, USA

Abstract. Camera sensors constitute an information rich sensing modality with many potential applications in sensor networks. Their effectiveness in a sensor network setting however greatly relies on their ability to calibrate with respect to each other, and other sensors in the field. This paper examines node localization and camera calibration using the shared field of view of camera pairs. We compare two approaches from computer vision and propose an algorithm that combines a sparse set of distance measurements with image information to accurately localize nodes in 3D. We evaluate our algorithms in a real testbed using a COTS camera interfaced to our sensor nodes. Sensors identify themselves to cameras using modulated LED emissions. Our indoor experiments yielded a 2-7cm error in a 6x6m room. Our outdoor experiments in a 30x30m field resulted in errors 20-80cm, depending on the method used.

1 Introduction

The use of low-cost imagers in sensor network applications is imminent. Small low-power imagers offer an information rich sensing modality that can detect features from a scene, perform visual confirmation and complement other sensing modalities. Because of these properties, we believe that camera enabled sensor networks will find numerous applications in everyday life. An intelligent camera-enabled sensor network can be used for understanding and creating behaviors in security, safety and entertainment applications. Imagine a sensor network that can seamlessly identify the formation of groups, the attack on a person or the removal of assets from a building. These are actions that could be seamlessly processed by camera-enabled sensor networks that could also provide a specific service as a response. With these application, it is easy to argue that smart cameras also have their limitations and should therefore function in close cooperation with other sensors. Other, non-imaging sensors offer a source of diversity and have potential for simplifying some of the vision tasks. To make this collaboration possible, one of the first problems that needs to be considered is the calibration of multiple camera views with respect to each other, and the localization of sensor nodes. The former will allow the combination of information from multi-camera views and the latter will allow cameras to collaborate with other non-imager sensors.

In this paper we provide an assessment of this problem by evaluating two complementary algorithms in real sensor network, where a fraction of the nodes are equipped with a COTS camera module, similar to the one used in cellphones. To experiment with camera related sensor networks we have build a camera module on top of a sensor node (Fig. 1a)that is capable of taking snapshots of a scene and perform some basic feature-extraction functionalities such as sobel edge detection (shown in Fig. 1b), motion detection, and direction of motion. We used these modules in an indoor and an outdoor deployment to collect data to compare two complimentary algorithms that use overlapping fields of view between pairs of cameras to localize nodes and calibrate cameras. We also propose an algorithm that uses a set of sparse distance measurements to refine locations. Our results show that a few simple deployment considerations and the use of imagers can result in lightweight, yet very accurate 3D localization that bypasses some of the challenges posed by rigidity constraints in the case of distance only localization [2]. Furthermore, as our experimental evaluation verified, the requirement of having distance measurements can be entirely eliminated by having 3 LEDs form a triangle pattern with known sides on the surface of the node or by simply making nodes to have a specific shape with known dimensions. This distance information is required since cameras cannot perceive depth. Without distance information, locations and translations can only be computed up to scale.

The approaches presented here are suitable both for indoor scenarios as well as for scenarios that need to localize dense networks of hundreds of nodes. Furthermore, although most of our discussion presents our work in the context of our testbed, the concepts discussed here can be applied in a broader set of deployments. Instead of using camera nodes, one could use a digital camera to photograph a sensor field from multiple viewpoints and use the image information to localize all the nodes in 3D. Instead of using LEDs to identify nodes, one could use nodes of a specific shape or color. Our approach would also be interpretable with some other approaches such as the one used with MIT Crickets in [7]. It could also be applied on other platforms such as Cyclops [11].

Our work builds up on concepts from computer vision but it is also bears a few differences. We perform an in-situ evaluation in the context of sensor network applications based on the actual camera technology we intent to use. We examine the feasibility of solving the problem using small sensor nodes, and we propose an algorithm for reconstructing node coordinates that utilizes deployment information that is more lightweight that the stratification approaches used in computer vision.

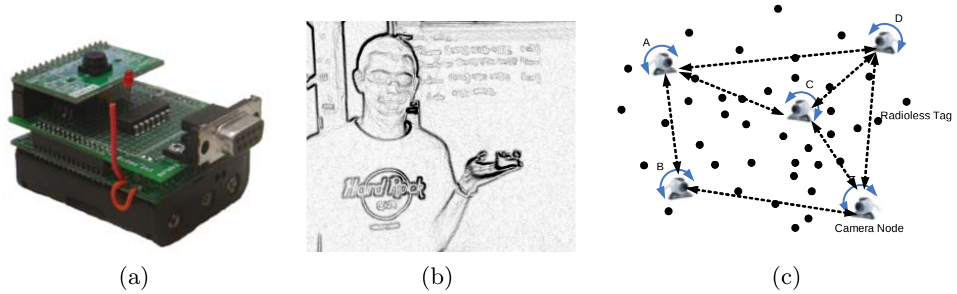
2 Problem Statement

Our presentation in this paper deals with computing the relative translation and rotation matrices between cameras and localizing nodes in 3D, using information between camera views. A practical solution to this problem can easily be extended to multihop scenarios by using the translation and rotation information among adjacent cameras. This process is known as *transfer* in computer vision.

With transfer, the coordinate system of any camera i, can be translated to the the coordinate system of any other camera j if there is a path of cameras from i to j, on which the relative rotations and translations, among adjacent camera pairs are known. Assuming that transfer is possible because camera nodes can communicate with each other, we state our problem as follows:

Given a network of N sensor nodes $t_1, t_2, t_3 ... t_N$ where a subset $m < N$ nodes are equipped with cameras, compute all possible node locations and the relative camera translations and rotations with respect to a local coordinate system. As we will explain in section 5, the setup we investigate requires some distance information. This can be provided with a distance measurement system on the nodes or by properly placing a set of LEDs in a special pattern on the surface of each node.

Section 6 provides an overview of the multihop localization algorithm, but the rest of the discussion and our evaluation focuses on how to exploit camera epipoles when two cameras have one or more nodes in their shared field of view. In such a setup each camera node will compute node positions with respect to its own coordinate system. The knowledge of the relative camera rotations and translations will allow the transformation of coordinates, to either coordinate system. Two algorithms are examined, measured epipoles and estimated epipoles. Camera coverage and mobility issues are discussed in section 8.



(a) (b) (c)

Fig. 1. a) XYZ sensor node with OV camera module, b) an image acquired by the camera module after sobel edge detection, c)deployment scenario

3 Related Work

The reconstruction of 3D imagery from images is a problem treated in computer vision for several years. In 1992 the work done by Tomasi and Kanade in [16] has proposed a way for reconstructing a scene and estimating camera parameters and feature point locations using matrix factorization. This initial work was performed under orthographic projection, and more recent work has treated the paraperspective case [8]. A more complete solution that compensates for the camera depth effects has been described in [13]. More recently, the works of Mantzel et. al. [5] and Devarajan et. al. [1] have begun to consider the problem of camera calibration in a networked setup. Both approaches have examined the problem from a computer vision perspective without the consideration of other sensor node capabilities. The former has proposed an iterative approach for localizing cameras (position, relative translation and orientation up to a scale) using linear relationships and the DLT camera calibration developed by Faugeras in [6].The work of Devarajan et.al. builds upon, more computatonally demanding factorization methods proposed by Sturm and Triggs in [13]. Cameras form microclusters with other nodes in the same coordinate system. The camera localization algorithm requires 4 cameras having at least 12 feature points in their common field of view. The algorithm also provides a scheme for aligning image frames.

Camera calibration using mobile entities such as people has also been considered in the computer vision community. Taylor et. al. at MIT [15] considers camera calibration for cases of non-overlapping field of views. In sensor networks, the problem of ad-hoc node localization has been treated in great detail. Our work is more related to fine grained localization schemes such as the ones that have been demonstrated in MIT's Cricket system [9, 10] and UCLA's AHLoS system [12]. In fact, many of the concepts presented here will be interoperable with the Cricket localization schemes recently presented in [7] or any other similar scheme.

4 Sensor Assisted Camera Localization

4.1 Background

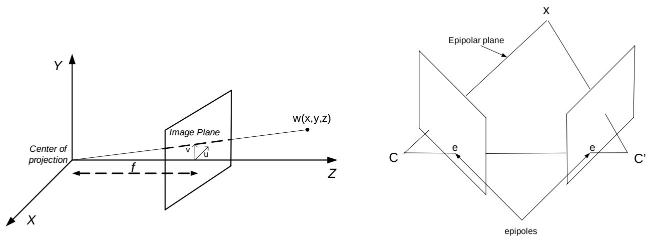
The extrinsic camera calibration parameters of interest are a rotation matrix $R_{3\times 3}$ and a translation vector $T_{3\times 1}$, also referred to as the extrinsic calibration parameters. In the absolute sense, the two matrices R and T represent the camera coordinates with respect to a 3-D world origin O. In general, our discussion about cameras will deal with three types of coordinates. World coordinates w are the coordinates given with respect to a real 3-D world origin. The camera also has its own camera centric coordinates $\bar{w}$ that can also be expressed in world coordinates with (1).

$$\bar{w} = Rw + T \tag{1}$$

The image coordinates u, v observed by the camera, can also be related to the camera centric coordinate system though the following equations.

$$\frac{u}{f} = \frac{\bar{w}_x}{\bar{w}_z}, \frac{v}{f} = \frac{\bar{w}_y}{\bar{w}_z} \tag{2}$$

where f is the focal length of the camera. The above expressions can be obtained by considering the ratio of triangles as shown in figure 2.



(a) (b)

Fig. 2. a) Camera diagram. Camera coordinate system z-axis is piercing the normalized $(u,\, v)$ image plain in the image plain origin, b)epipoles between a pair of cameras .

Our work uses the epipoles between pairs of cameras. The epipoles between a pair of cameras are defined as the points where a straight line connecting the two camera centers intersect the image plane of each camera [3] as shown in Figure 2.

Throughout the paper we use the following notation:
l_{ab} - distance between nodes A and B
v_{ab} - unit vector from A to B
u_a, v_a - x, y coordinates of an object on the image plane of camera A.

4.2 Localization Based on Two Camera Views

We now examine two methods for extracting camera epipoles, direct observation, and fundamental matrix estimation.

Direct Epipole Observation - *Measured Epipoles*(**ME**) : If a tag C is observed by two cameras A and B that can also observe each other, the distances between A, B, C can be determined up to a scale. This was demonstrated in [14].

If cameras A and B can observe each other and a tag C, then we can derive the unit vectors v_{ab}, v_{ac}, v_{ba} (see Fig. 3), and v_{bc}. From these we can derive the normalized versions $n_a = v_{ab} \times v_{ac}$ and $n_b = v_{ba} \times v_{bc}$. Note that v_{ab}, v_{ba}, n_a and n_b are related by a rotation matrix R_{ab}, the relative orientation matrix between cameras A and B.

$$v_{ab} = -R_{ab} v_{ba}$$

$$n_a = -R_{ab} n_b$$

From the two perpendicular unit vectors v_{ab} and n_a, we can construct the orthonormal matrices R_a and R_b. $R_a = \left[v_{ab}\ n_a\ (v_{ab} \times n_a\ ) \right]$ and $R_b = \left[v_{ba}\ {-n_b}\ (v_{ba} \times n_b) \right]$.

Substituting we get the relative orientation as

$$R_{ab} = R_a (R_b)^T \tag{3}$$

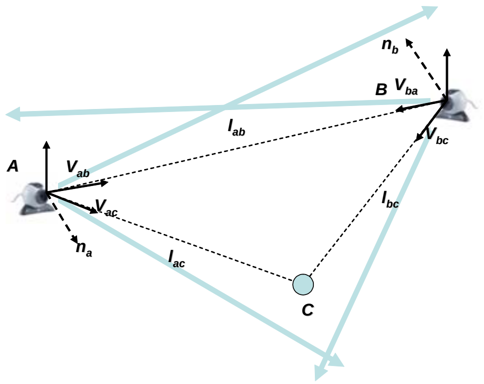


Fig. 3. Measured epipole: A tag can be observed by 2 cameras that can also observe each other

from this we can extract the following linear system

$$l_{ab}v_{ab} + l_{bc}(R_{ab}v_{bc}) - l_{ca}v_{ac} = 0 \tag{4}$$

Solving these equations we can get l_{ab}, l_{ac} and l_{bc} up to a scale. Since we also know the distance l_{ab} we can solve to Euclidian scale.

Extracting the Epipoles from the Fundamental Matrix - *Estimated Epipoles* **(EE)** : If two cameras are not facing each other, then the epipoles cannot be directly observed. Nonetheless, the camera epipoles can still be determined if the camera observations can provide enough information to compute the fundamental matrix between two cameras [4]. Although this can be done with a minimum of 5 nodes in the common field of view of two cameras, the resulting problem is highly non-linear and very difficult to solve. In our implementation we chose to use the widely used normalized eight-point algorithm proposed by Hartley in 1997 [3].

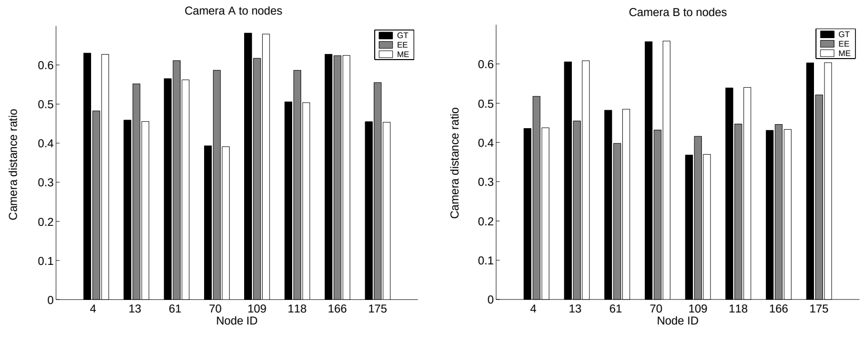
The eight-point algorithm uses eight or more points in the common field of view of a pair of cameras to estimate the fundamental matrix F between them. F is defined by the equation

$$u'^T F u = 0$$

u' and u are corresponding feature points in the images of the two cameras. The epipoles of the two cameras can be then extracted from F. For any point x, the epipolar line $l' = Fx$ contains the epipole e'. Thus $e'^T(Fx) = (e'^T F)x = 0$ for all x. From this it follows that $e'F = 0$ and $Fe = 0$, thus the epipoles of the two cameras, e' and e are the left and right null vectors of F respectively [4]. The estimation of both camera's epipoles allows us to compute the rotation between the two cameras and the distances from both cameras to all the nodes up to a scale using the formulation described in the previous subsection. The only difference in this case is that instead of using the measured epipoles, as in ME, we use the estimated, from the fundamental matrix, epipoles.

5 Optimized Estimated Epipoles (OEE)

The resulting epipole estimates are noisy and thus will provide noisy distance estimates. An illustrative example of the error based on our testbed measurements is shown in Fig. 4 . The figure shows the distance error ratio for ground truth (GT), ME and EE when the distances between the cameras A and B and the nodes are computed (by equation 4). The figure shows that ME produces accurate measurements comparable to ground truth while EE has noisy measurements. Furthermore note that the error from EE at the two cameras is complimentary. If a distance between camera A and a node is overestimated then the distance between the same node and camera B will be underestimated and vice-versa. This is of course an artifact of the lightweight algorithm we use. It does however suggest that refinement is possible by attempting to match the two camera views that have complimentary error as shown in Fig. 5. The alternative is to use stratification, a much more complex algorithm from computer vision [4], that is not suitable for small sensor nodes.

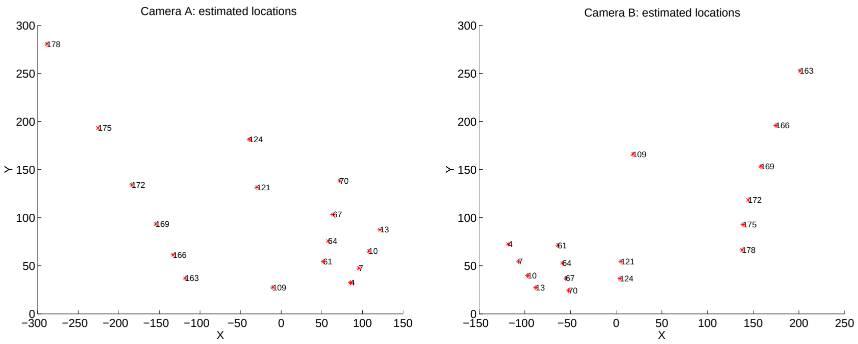


(a)

(b)

Fig. 4. Estimated distances between two camera nodes and other observed nodes. GT - Ground truth, EE - estimated epipoles, ME - measured epipoles

To reduce this error, we formulate a more constrained optimization problem, where the distances between the observed nodes and the camera nodes are estimated simultaneously. To enforce matching views between the cameras, we also consider the additional constraints that the pairwise distances between all the nodes observed by the two cameras should be equal from the view point of both cameras. Consider the scenario in Figure 5 with cameras A and B and observed nodes i and j. The distance l_{ij} can be estimated by both cameras as



(a) (b)

Fig. 5. Estimated locations using an optimized version of the distances in Fig. 4

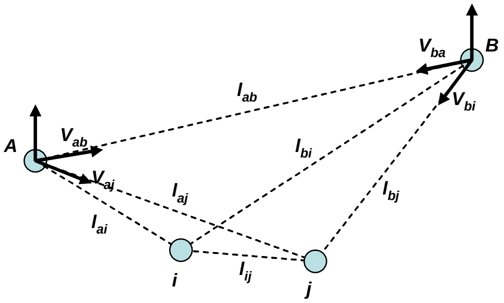


Fig. 6. Imposing the constraint that two cameras should agree on the l_{ij} estimate

$$
\begin{aligned}
{}^{A}l_{ij} &= ||l_{ai}v_{ai} - l_{aj}v_{aj}|| \\
{}^{B}l_{ij} &= ||l_{bi}v_{bi} - l_{bj}v_{bj}||
\end{aligned}
\qquad (5)
$$

If we assume that some of the l_{ij} distances are known, we can impose these as constraints. The rest of the analysis is identical for both cameras, therefore we will focus on camera A. The required camera-to-node distances can be estimated up to a scale with respect to camera A's view point as:

$$
\begin{aligned}
l_{ab}v_{ab} + l_{bi}(R_{ab}v_{bi}) - l_{ai}v_{ai} = 0 \\
l_{ab}v_{ab} + l_{bi}(R_{ab}v_{bj}) - l_{aj}v_{aj} = 0
\end{aligned}
\qquad (6)
$$

As it was described in the previous section, in the case of the estimated epipoles method(EE), the computed distances from the camera to the nodes will be erroneous. These erroneous distances can be refined if a subset of distances between the nodes seen by the camera is known. Assuming that n distances $d_{i_x j_x}$, $x = 1, 2, \ldots, n$ are known, the following equations should hold:

$$^{A}l_{i_x j_x} = ||l_{ai_x} v_{ai_x} - l_{aj_x} v_{aj_x}|| = d_{i_x j_x} \tag{7}$$

where x is an index running over all n known edges and i_x, j_x are the nodes connected by the x^{th} edge. In order to refine the distance estimates $l_a i_x$ and $l_a j_x$ based on the known n distances we would need to minimize the following function:

$$L = min \sum_x \left(d_{i_x j_x} - ||l_{ai_x} v_{ai_x} - l_{aj_x} v_{aj_x}||\right)^2 \tag{8}$$

Since x is running over all n known distances, function L is basically a set of n equations where the number of unknowns depends on the known edges. If all the n known edges are independent[1] the number of variables is $2n$ because in that case each edge (i, j) would involve two unique unknowns: l_{ai} and l_{aj}. Since the number of equations is less than the number of unknowns the minimization of function L is impossible. However, if different known edges share common nodes then the minimization of L is possible. The simplest case, where the minimum number of edges is needed and function L can be minimized, is when all the distances among three nodes are known. In other words, when all the edges of a triangle that is formed by three nodes are known the set of equations represented by L can be solved. Each edge of the triangle provides an equation giving a total of 3 equations. For each node in the triangle there is only one unknown. Therefore the number of unknowns is equal to the number of equations and the distances from camera A to the nodes forming the triangle can be refined. This set of equations is a non-linear set of equations that can be solved with an Extended Kalman Filter(EKF) or another gradient descend method.

Note that any other closed geometric shape (square, pentagon, hexagon etc) could be used in the optimization process. We use triangles because they require the minimum number of known distances (only 3) and therefore they require the minimum number of local information. Therefore based on local triangles we can optimize the distances between the cameras and the non-camera nodes with respect to the camera's coordinate system. After the refinement of the distances between the camera nodes and the non-camera nodes, the rotation matrix between two camera nodes with overlapping fields of view can be refined using equation 6 and the already refined distances.

The preceding ME, EE and OEE algorithms provide us with a means of bootstrapping a coordinate system and also computes the relative rotation and translation between a pair of cameras. Next section shows how these two components can be applied on a distributed localization algorithm.

[1] Two edges are independent if they do not share any common nodes

6 Multihop Localization Algorithm Overview

Although our focus in this paper is on how to perform calibration based on two camera views, we briefly describe an example of a distributed localization algorithm. This algorithm assumes the existence of a coordinate transformation service that runs in the background and distributes the rotations and transformations between pairs of cameras so that nodes can be localized on demand in multiple coordinate systems, using *transfer*.

Immediately after deployment(or when triggered), each node broadcasts a list of IDs for all the nodes it can observe to its radio neighbors. The camera nodes processes these broadcasts and creates a vision graph that connects camera nodes that have one or more nodes in their shared field of view. The same information also allows each camera node to identify which other nodes can use its observations to run the ME or EE algorithm. Once a camera node identifies these nodes, it forwards them the observations (node id and image coordinates) for the observed nodes in their shared field of view.

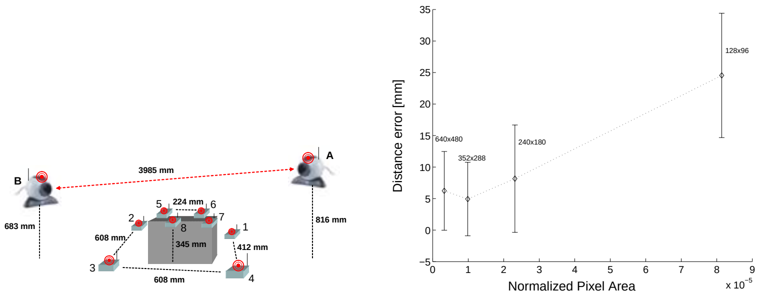
After this phase, each camera node has the required observations to bootstrap its local coordinate system by running ME or OEE. ME and OEE provide a distance(magnitude) and a vector(direction) for each node they consider. This allows the camera node to estimate the coordinates of each of the considered nodes in its own coordinate system. The relative rotation between the two nodes is also computed, and its passed to the coordinate transformation service. At this point the coordinate transformation service is also notified to check if the new information generated will allow computing camera transformations from other nodes.

If a new node appears in a camera's field of view, the camera repeats a similar process to discover if any of its neighbors can observe the same node. Depending on the number of nodes available in the shared field of view each camera node may execute ME, EE or simply compute the node's location based on previously computed rotation information. Do to space limitations, the details and evaluation of the multihop localization and the coordinate transformation service are omitted in this paper. These will be treated in a subsequent paper.

7 Evaluation

The evaluation of the the ME, EE and OEE algorithms is performed on real measurements obtained from an indoor and outdoor dataset collected using our camera enabled sensor nodes. Each node is equipped with an extension board carrying a COTS OV7649 camera module from OmniVision. The node processor can acquire frames from the camera and perform some basic feature extraction, such as sobel edge detection, differencing to detect motion and LED identification. All nodes also carry a Lumex CCl-CRS10SR omnidirectional LED with an axial intensity of 40 millicandellas. In our lab setup, the cameras nodes can clearly observe these LEDs for distances up to 4 meters. In our tests, camera nodes can uniquely identify the nodes using an asynchronous protocol which can read the node id of each node by toggling the LED to send bits across.

Using this platform we acquired 2 datasets, one indoor and one outdoor. The indoor scheario was comprised of 2 camera nodes and 16 non-camera nodes identified with blinking LEDs. The two camera nodes were placed at different heights and angles with respect to the non-camera nodes yielding four different datasets. For the outdoor scenario, we placed 80 bright orange postit nodes in an outdoor plaza next to our building. In this outdoor test we were interested in the accuracy and range of our system, and we assumed that the correspondences for each post it node were known [2]. The layout of the outdoor scenario is shown in Fig 7.



(a) (b)

Fig. 7. a) Characterization scenario using 8-tags and 2 camera nodes, b) Error across different resolutions

Before evaluating our algorithms, we also verified that the error varies linearly with camera resolution. We verified this with a characterization of the measurement error across four different camera resolutions: 640 x 480 (VGA), 352 x 288 (CIF), 240 x 180, and 128 x 96 (SQCIF). Our evaluation was done using the ME method using a topology of two camera nodes and 8 tags as shown in Figure 7a. The maximum error was 3.32 cm in the lowest resolution. Figure 7b describes the measurement error as a function of the normalized pixel area for each resolution, as computed using the pairwise distance measurements. With the exception of the VGA resolution, the error follows a linear trend $y = 2.8 \cdot 10^5 x + 2$. This result verifies our intuition that error scales with resolution and also indicates that the lens distortion effects are almost negligible.

7.1 ME, EE, OEE Evaluation

The perfomance of the ME, EE and OEE algorithms in the case of one of the indoor experiments can be seen in Figure 9. The accuracy of the ME method shows that the camera is a very reliable measuring modality. On the other hand, the poor performance of the EE algorithm shows that the estimates of the epipoles

[2] The image correspondences can be found by methods from computer vision not described here

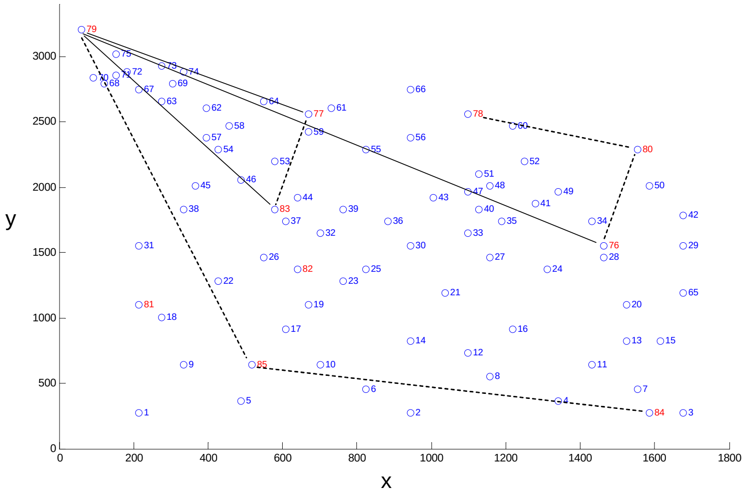
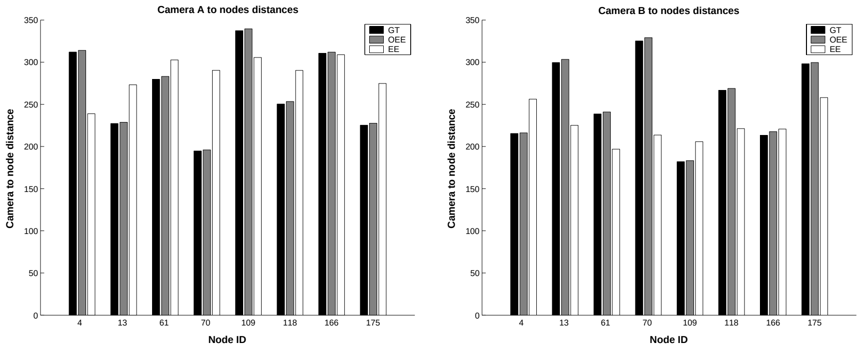


Fig. 8. Layout of the outdoor setup. Camera node pairs that can run ME are connected with solid lines and camera pairs that can run EE and REE are connected with dotted lines. The units are in cm.

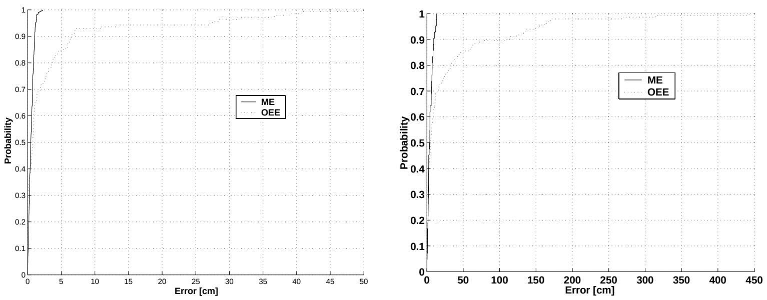


(a) (b)

Fig. 9. Estimated distances between two camera nodes and other observed nodes for the indoor experiment. GT - Ground truth, OEE - optimized estimated epipoles, EE - estimated epipoles. All distances are in cm.

are erroneous and therefore unreliable. However, as it can be seen in Figure 9 the OEE method proposed in this paper reduces drastically the error and performs almost equally well to the ME method.

The drastic reduction in the error in the case of the OEE method can be seen better in Figures 10 and 11 where the empirical CDF of the node-to-node distance error is shown. In the indoor setup, EE reports an error of 60*cm* with probability 90% while OEE reports an error of 7*cm* with the same probability. Given that the average node-to-node distance in the indoor setup is approximately 85*cm* the OEE algorithm performs fairly well. The ME algorithm has the best performance reporting an error of 2*cm* with 90% probability. Figures 10 and 11 show very similar results for the outdoor experiment. Again, ME has the best performance(20cm error with 90% probability) but OEE performs comparably well reporting an error of 60*cm* with 90% probability given that the average node-to node distance is approximately 297*cm*. In the case of the outdoor setup a more detailed view of the performance of all the methods can be seen in Figure 12.



(a) (b)

Fig. 10. Empirical CDF for node to node distance estimates a) all indoor scenarios with optimized EE, b) all outdoor scenarios with optimized EE

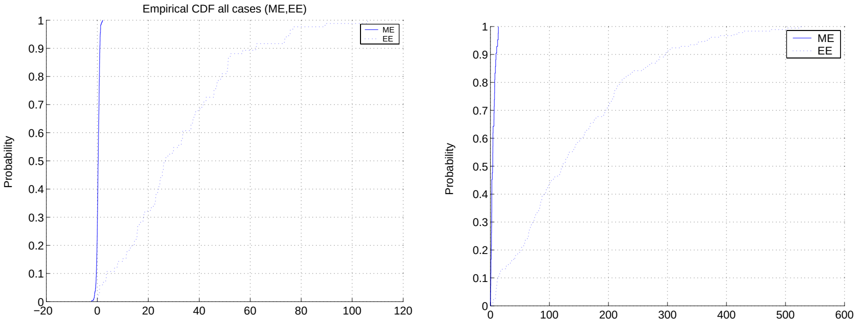
To compare our results to other approaches we also evaluate the estimated internode distance error with respect to the follow two metrics:

$$p = \sum_{i=1}^{N} \frac{|l_i - \hat{l}_i|}{N} \tag{9}$$

and

$$q = \sqrt{\sum_{i=1}^{N} \frac{(l_i - \hat{l}_i)^2}{N}}. \tag{10}$$

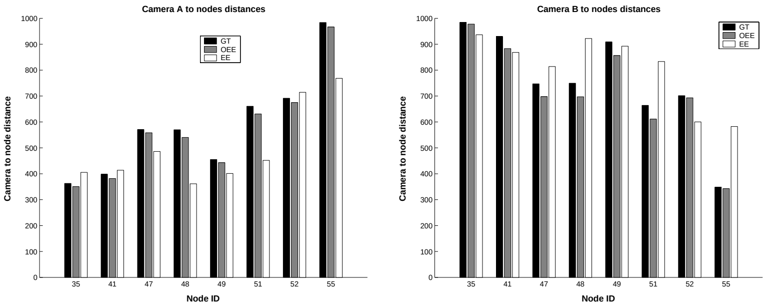
N is the number of pairwise distances, l_i are the measured distances and $\hat{l}_i$ are the estimated pairwise distances computed from the localization results



(a) (b)

Fig. 11. Empirical CDF for node to node distance estimates a) all indoor scenarios with unoptimized EE, b) all outdoor scenarios with unoptimized EE

of the algorithm. Metric (9) is the *mean error*. Metric (10) is the *square root of the mean square error* and it is also used in [7]. The standard deviation of the error, noted by σ, is also computed. Table 1 summarizes our results and compares to the ultrasonic approach from [7]. At a room level scale, the ME and OEE approaches are more accurate than ultrasound. In the outdoor scenarios the error is much higher. This is expected since the average internode distances are much larger. In this case camera accuracy degrades since the small angular error of the camera measurements increases tangentially with distance.



(a) (b)

Fig. 12. Estimated distances between two camera nodes and other observed nodes for the outdoor experiment. GT - Ground truth, OEE - optimized estimated epipoles, EE - estimated epipoles. All distances are in cm.

.

Algorithm	Indoor		Outdoor	
	$p(cm)$	$q(cm)$	$p(cm)$	$q(cm)$
ME	4.0165	0.3779	17.9794	1.2493
OEE	0.9714	1.4159	29.6057	51.16818
Ultrasound	7.02	5.18	N/A	N/A

Table 1. The p and q distance metrics for the ME and OEE algorithms as well as for the ultrasound technology in the case of both indoor and outdoor experiments.

8 Discussion

Localization and camera calibration when sensors can generate stimuli that allows them to communicate with cameras is appealing in some applications since it implies one could easily build very simple nodes that cost a few cents each with today's technologies. Imagine for instance, thousands of floatable chemical sensors dispersed to measure chemical concentrations in the sea, in water reservoirs or large tanks. Such nodes would consist of a small battery, a tiny 4-bit microcontroller and an LED serving both as a communication and localization device. A smaller set of camera nodes can observe them.

During our evaluation we also realized that camera placement is a main problem. To have adequate field of view, the cameras need to be mounted on the walls or the ceiling or need to be placed in positions higher than other nodes facing down. This together with the limited field of view of cameras suggest that articulation, and autonomous motion are necessary for many camera enabled sensor networks. It also makes a case for using cameras together with ther sensors that can help overcome the limited field of view of the cameras. Another issue in the deployment of camera networks is privacy. We plan to handle this issue with the custom designed cameras under development in our project[reference withheld]. These cameras will be blind to images and will only extract features and sensor stimuli, preserving the privacy of their users.

9 Conclusions and Future Work

In this paper we have compared using a real sensor network two basic computer vision algorithms from computer vision. Using the results of our evaluation we have developed a refinement algorithm that is more lightweight than traditional computer vision algorithms and therefore more suitable for resource constrained sensor nodes. As part of our future work, we plan to use these results to localize and track events, other than sensor nodes. The current infrastructure will allow us to leverage the collaboration of imager and non-imager sensors to identify other events, targets and behaviors in sensor networks. As part of the same project we are also perusing the design of custom camera sensors that have very low power consumption and can detect events with feature extraction mechanisms directly implemented in the camera hardware.

Acknowledgement

This work was funded in part by the National Science Foundation CISE/CNS award #0448082 and by OKI Semiconductor .....

References

1. D. Devarajan and R. Radke. Distributed metric calibration for large-scale camera networks. In First Workshop on Broadband Advanced Sensor Networks (BASENETS) San Jose conjunction with BroadNets 2004, October 25 2004.
2. D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Young, A.S. Morse, A. Savvides, and B. Anderson. Network localization in partially localizable networks. In To appear in the proceedings of INFOCOM 2005, April 2005.
3. R. Hartley. In defense of the eight-point algorithm. In IEEE Transactions on Pattern Analysis and Machine Intelligence, June 1997.
4. R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge Press, 2003.
5. W.E. Mantzel, H. Choi, and R.G. Baraniuk. Distributed camera network localization. In Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers, November 2004.
6. S.J Maybeck and O.D Faugeras. A theory of self-calibration of a moving camera. In International Journal of Computer Vision 8(2):123-151, 2004.
7. D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In Proceedings of ACM SenSys, Baltimore, Maryland, November 2004.
8. C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 19, NO 3, March 1997.
9. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In Proceedings of 6th ACM Mobicom, Boston, MA, August 2000.
10. N. B. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In Proceedings of the 7th ACM Mobicom, Rome, Italy, July 2001.
11. M. Rahimi, D. Estrin, R. Baer, H. Uyeno, and J. Warrior. Cyclops: image sensing and interpretation in wireless networks. In Second ACM Conference on Embedded Networked Sensor Systems, SenSys, Baltimore,MD, November 2004.
12. A. Savvides, C.C. Han, and M. B. Srivastava. Dynamic fine grained localization in ad-hoc sensor networks. In Proceedings of the Fifth International Conference on Mobile Computing and Networking, Mobicom 2001, Rome, Italy, pages pp. 166–179, July 2001.
13. P. Strum and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In In Proceedings of European Conference on Computer Vision, pages 709–720, 1996.
14. C. Taylor. A scheme for calibrating smart camera networks using active lights. In SenSys04 Demostration Session, Baltimore, MD, Novemeber 2004.
15. C. Taylor, A. Rahimi, J. Bachrach, and H. Shrobe. Simultaneous localization and tracking in an ad hoc sensor network. In AI Lab Technical Report AIM-2005-016, 2005.
16. C. Tomassi and T. Kanade. Shape and motion from image streams: a factorization method. In Carnegie Mellon Technical Report CMU-CS-92-104, 1992.