The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Development and Calibration of a Low Cost Wireless Camera Sensor Network

Vimal Mehta, Weihua Sheng, Tianzhou Chen, Quan shi

***Abstract*— In camera sensor network research, physical camera sensor network platforms with low power and low cost are needed for testing and validating algorithms. It is also necessary that the camera nodes be calibrated precisely. In this work, we first develop and evaluate a low power and low cost wireless camera sensor network platform. The camera sensor nodes in this platform transmit a grayscale image over a wireless channel to a master control station. Then we propose a simple, light-weight algorithm to perform distributed calibration of the camera sensor nodes. The camera sensor nodes use their imaging abilities in collaboration with a cooperative moving target to determine their own positions and orientations. The proposed algorithm requires simple arithmetic calculations and hence it can be realized on low power processors. This calibration algorithm has been implemented and evaluated on the developed physical camera sensor network platform.**

***Index Terms*— camera sensor network, camera calibration, localization**

I. INTRODUCTION

A. Motivation

Wireless camera sensor networks have been employed in various real-time applications [1] which involve data acquisition, sensing and control. Some platforms of camera sensor networks can provide images with great fidelity. However, a high quality image imposes high demand on the processing speed and power consumption on the camera sensor nodes, as well as the communication bandwidth. Therefore there is a need to develop a low power and low cost image sensing platform which conveys the image more efficiently rather than with high fidelity. Our main goal in this paper is to develop a low power, low cost and low bandwidth camera sensor network platform that can be employed to implement and validate various algorithms.

On the other hand, due to the large number of randomly deployed nodes in a typical camera sensor network application, it is necessary to have the knowledge of the positions and orientations of the camera sensor nodes. Hence, it is necessary that the cameras be calibrated within certain precision. Camera calibration [8, 9, 10, 11, 12] is a process of determining some of its internal parameters like focal length, skew factor, lens distortion and external parameters like position and orientation. For a camera sensor network, calibration can be implemented either in a centralized or in a distributed manner. In a centralized implementation, all the nodes in a network communicate with a central node which runs the calibration algorithm to determine the parameters of the remaining sensor nodes of the network. This method has the disadvantage that there is a single point-of-failure in the network and also requires the exchange of large amounts of information which is both time and power consuming. In a distributed implementation [2], each sensor node calibrates itself without relying on a central node. The processors of the sensor nodes run the computational algorithms required to perform the calibration. Most of the existing distributed calibration techniques [13, 15] solve a complicated non-linear optimization problem which is difficult to implement on low power processors. Hence, there is a great need to develop a simple and light-weight camera calibration algorithm that can be implemented in a distributed manner on low power processors.

B. Related work

In this section, we will examine the various physical camera sensor network platforms that have been developed in recent years. In [3], Teixeira *et al.* propose a simple camera sensor network to reduce power consumption and bandwidth. They also aimed to develop a framework wherein they can use low-level sensors to model high-level behavioral events. In [4], Chen *et al*. propose a low bandwidth wireless camera sensor network platform. It uses a 1.3 mega pixel camera integrated with a PDA. The Cyclops platform [5] is an integration of CMOS camera and a wireless sensor node. It consists of programmable logic devices for high speed data transfer and image processing. The WiSN platform [6], though powerful, makes use of the same microprocessor for image processing as well as for networking. Hence this platform has a reduced frame rate. In our work, we develop a physical camera sensor network platform which separates the image capture and processing parts from the networking part, which can avoid the frame rate reduction.

Camera calibration has been widely researched in the computer vision community [7, 8, 9, 10]. In [9], Zhang proposes a flexible calibration technique in which, from the captured images of a planar pattern, feature points are extracted to determine the internal and external parameters of the camera. In [11], a service mobile robot equipped with a planar pattern collaborates with the camera sensor nodes in the environment and calculates their external parameters by

Vimal Mehta and Weihua Sheng are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: weihua.sheng@okstate.edu).
Tianzhou Chen is with the School of Computer Science, Zhejiang University, Hangzhou, 310027, China. (tzchen@zju.edu.cn)
Quan Shi is with the Process Engineering and Process Control Group PPG Industries, Inc. Cheswick, PA. 15024 USA (qshi@ppg.com).

communicating tracking information. In [12], Kulkarni *et al.* propose a technique to determine the relative positions and orientations of the camera sensor nodes in the environment. The camera nodes capture the images of an object of known size placed at various locations in the environment. Each camera node estimates its external parameters, the degree of overlap and region of overlap with its neighbors and uses this information to track the object. In [13], Lee and Aghajan present a collaborative technique to localize the nodes of a camera sensor network using opportunistic observations of a target. Each node uses in-node image processing to determine the coordinates of the center of the blob of a detected object. Two nodes setup a reference coordinate system. The reference nodes collaborate with more than one uncalibrated node to track the moving object simultaneously. Each node uses Gauss–Newton method [13, 14, 15] to determine its position and orientation in the reference frame. However, this method requires a minimum of three sensor nodes to capture at least five observations simultaneously. The cooperative target based self calibration protocol in [16], uses the coordinate information at four different locations in the field of view of a camera sensor node to perform calibration. The sensor node uses a non-linear estimation method to solve for its external parameters. It uses this information with a subset of three other locations to determine its orientation.

Though our work is similar in spirit to the work in [16], it is different in three aspects. Firstly, we assume that the moving target is equipped with a dead reckoning [17] based position sensor. Secondly, we use the location information at only three locations in the field of view to calibrate a camera sensor node. Thirdly, we use a simple method to solve the distributed calibration problem. The paper is organized as follows. Section II gives the details of the physical camera sensor network platform that we have developed. In Section III, we propose novel algorithms to solve the distributed calibration problem. In Section IV, we evaluate the physical platform and also implement the proposed calibration techniques on this platform. The evaluation and the experimental results are also presented in the same section. Section V concludes the paper and also throws light on the future work.

II. DEVELOPMENT OF THE CAMERA SENSOR NETWORK

We developed a low power and low cost wireless camera sensor network platform using commercial-off-the-shelf (COTS) parts. The sensor nodes of this platform transmit the grayscale image captured by their cameras to a central station such as a PC via low bandwidth wireless channel. Here we will describe the camera sensor node design, the networking and the image capture and transmission aspects in the following sections respectively.

A. Camera sensor node

For the purpose of vision sensing, the CMUCAM2 CMOS vision camera sensors [18] were used while the wireless connectivity was established by using four Jennic JN5121 wireless modules [19]. The CMUCAM2 CMOS camera sensor has a SX52 micro-controller interfaced to OV6620 or OV7620 omni vision CMOS camera [20]. It has an adjustable resolution of up to 166 x 255 pixels and can track user defined blobs at up to 50 frames per second. The CMUCAM2 is factory programmed with a firmware to output the pixel-wise image values of a captured image frame, the coordinates of the centroid of a detected blob, and the mean and variance of the detected blob over an RS-232 or a TTL level serial port. The Jennic module is based on the JN5121 wireless micro-controllers from Jennic Corporation. JN5121 is a 32 bit RISC CPU with an built-in programmable IEEE 802.15.4 [21] protocol stack operating at 2.4 GHz. The camera sensor node is shown in Figure 1.

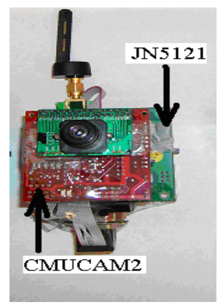

Figure 1: The camera sensor node.

B. Networking

A set of four JN5121 modules is used to setup the wireless communication network. One of the devices, called the coordinator is programmed to start the zigbee [21] network and allow other devices to join it. The device that joins this network is called an end-device. Each end-device is allocated a unique 16 bit MAC address by the coordinator, on joining the network. The end-devices are interfaced to the CMUCAM2 cameras using an RS-232 serial communication link. The coordinator transmits control commands to the camera sensor of a specific end-device. These commands are used to reset the camera, capture an image frame or track an object if it is present in the field of view of the camera.

C. Image capture and transmission

Upon receiving a *capture frame* command from the coordinator, the camera on an end-device outputs the values of each pixel in its field of view over the RS-232 link to the JN5121 controller of the end-device. Each pixel value of the image is transmitted as 3 bytes of the primary colors red, green and blue which make up the pixel.

Color image frame information sent by the CMUCAM2 camera has a typical size of 37K bytes. However, due to the limited memory in the JN5121 module, it is not possible to store the entire color image in the wireless module. Hence, it is necessary that the color image be compressed to its grayscale equivalent and then stored into the memory of the wireless module for local processing and transmission to the central station. The grayscale image will occupy only about 15K bytes of memory which includes the raw image and some control characters to distinguish between each row of image. A color pixel can be converted to its grayscale equivalent (Y) using the equation below

$$Y = 0.59 \times G + 0.31 \times R + 0.11 \times B$$

The conversion from color to grayscale is done on the fly by the Jennic processor when it is receiving the data from the camera over the serial link. Once the grayscale image in the entire field of view has been stored into the memory, the processor uses its radio to transmit the image to the central station. It transmits the grayscale image in packets of 100 bytes to a Jennic coordinator node, which is connected to a PC. The PC runs a JAVA program that constructs the image from its pixel values.

III. DISTRIBUTED CALIBRATION OF A CAMERA SENSOR NETWORK

Camera calibration is an important step before the camera senor network can be used. In this work, we focus on determining the external parameters of a camera sensor node. Consider a set of n camera sensor nodes deployed into an unknown environment for the purpose of tracking as shown in Figure 2. A camera sensor node must know its own location and orientation in order to localize the moving target. So, we firstly need to generate a reference coordinate system in the environment and for every camera sensor node $i\ (i = 1\ to\ n)$ in the environment. We need to determine its pose (x_i, y_i, θ_i), where (x_i, y_i) specify the 2D-coordinates of the camera sensor node and θ_i represents the orientation of the camera in the reference coordinate system.

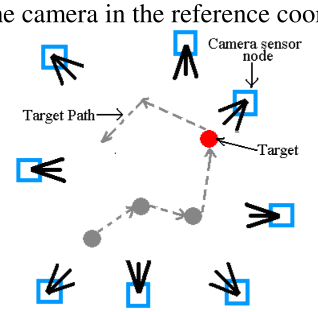

Figure 2: A camera calibration scenario.

We propose a novel method for distributed camera calibration using a cooperative target, which is suitable for low power applications. We assume that a cooperative target equipped with wireless communication capability moves around in the environment to assist the calibration. The target is assumed to be equipped with a dead reckoning based position sensor module that helps it to determine its current location coordinates relative to its previous known position. Such a target can be a solider or a vehicle. The camera is equipped with image processing software to extract the center of the detected blob and determine the coordinates of the center of the blob with respect to its field of view. Whenever a camera sensor node detects the presence of this target, it sends beacons across the network triggering frame captures at the other nodes. One of the remaining nodes that can simultaneously observe this target is designated as a helper node. The triggering node and the helper node together form a reference coordinate system with its origin assumed to be at the triggering node. The reference nodes calibrate themselves, localize the target and communicate its current location information in the reference frame. This localized target, while moving around in the environment, collaborates with the camera sensor nodes that it encounters in its path and assists them in their calibration.

A. *System model*

The system model for the proposed calibration algorithms is shown in Figure 3, which shows the reference coordinate system with the triggering node 0 located at the origin. Assume the helper node 1 is at a distance d units from the triggering node. In Figure 3, θ_i is the unknown orientation of each camera sensor node with respect to the positive X-axis of the reference coordinate system. The angle between the optical axis of a camera sensor node and the line joining its camera center to the moving target is denoted by ϕ_i. This angle information is calculated using the pin-hole camera model shown in Figure 4. If D is the horizontal resolution of the camera in pixels, ψ is the angle of the field of view of the camera and l is the distance between the centre of the camera and the point on the image plane of the camera where the image of the object is formed, then ϕ_i is given as follows.

$$\phi_i = \tan^{-1}\left(\frac{2l}{D} \times \tan\left(\frac{\psi}{2}\right)\right)$$

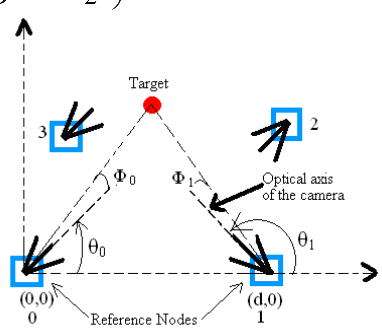

Figure 3: System model.

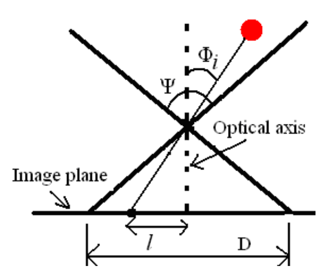

Figure 4: The pin-hole camera model redrawn from [13].

B. Calibration of the reference nodes and localization of the moving target

The first step towards the calibration of the nodes of the network is the localization of the moving target in the reference coordinate system. The moving target is simultaneously observed by the reference nodes 0 and 1. Assume the target move from location 1 to location 2 while maintaining itself in the field of view of both the reference nodes as shown in Figure 5.

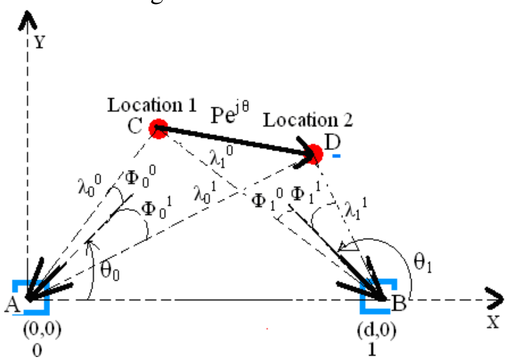

Figure 5: Target moves from location 1 to location 2.

The readings of the step size and the heading angle obtained by the dead reckoning position sensor module between locations 1 and 2 be denoted by a vector $Pe^{j\theta}$. For every m^{th} observation of the target made by each sensor node k, ϕ_k^m is the angle calculated using the pin-hole camera model and λ_k^m is the unknown distance between the target and each camera sensor node involved in the localizing process. Applying the sine rule to the triangles ABC and ABD shown in Figure 5, the following relations can be obtained.

$$\frac{\lambda_1^0}{\sin(\theta_0+\phi_0^0)}=\frac{\lambda_0^0}{\sin(\theta_1+\phi_1^0)}=\frac{d}{\sin[(\theta_1+\phi_1^0)-(\theta_0+\phi_0^0)]} \quad (1)$$

$$\frac{\lambda_1^1}{\sin(\theta_0-\phi_0^1)}=\frac{\lambda_0^1}{\sin(\theta_1-\phi_1^1)}=\frac{d}{\sin[(\theta_1-\phi_1^1)-(\theta_0-\phi_0^1)]} \quad (2)$$

$$\lambda_0^1 e^{j(\theta_0-\phi_0^1)} = Pe^{j\theta}+\lambda_0^0 e^{j(\theta_0+\phi_0^0)} \quad (3)$$

$$\lambda_1^0 e^{j(\theta_1+\phi_1^0)} = -Pe^{j\theta}+\lambda_1^1 e^{j(\theta_1-\phi_1^1)} \quad (4)$$

We define $\bar{x}=[\theta_0\ \theta_1\ \lambda_0^0\ \lambda_0^1\ \lambda_1^0\ \lambda_1^1]^T$ as a state variable matrix of the unknown parameters. Gauss-Newton method is used to estimate the state variables. Once the unknown parameters are estimated, the triggering node, determines the external parameters for the reference nodes and also the coordinates of the target at locations 1 and 2. The triggering node, communicates the coordinates of locations 1 and 2 to the cooperative target. The target then uses dead reckoning to updates its coordinates from time to time. The next step is the distributed calibration of the remaining nodes in the environment.

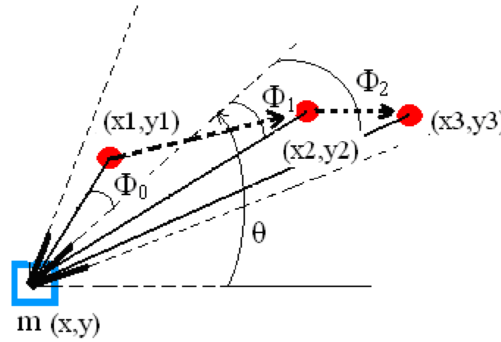

Figure 6: Target at three distinct locations in the field of view of an uncalibrated camera sensor node.

C Calibration of the remaining nodes

Assume that the cooperative moving target has been detected by an uncalibrated camera sensor node n situated at an unknown location (x, y) with its camera oriented at an unknown angle θ. Assume that the target has passed through three distinct locations $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3) in the field of view of the camera sensor node m as shown in Figure 6. The target communicates its coordinates at each of the location to the camera sensor node.

At each location, the camera extracts the coordinates of the detected target and uses the pin-hole mode to determine the angle. Let ϕ_0, ϕ_1, ϕ_2 be the angle between the target and the camera at $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3), respectively. From Figure 6, we have.

$$\tan(\theta+\phi_0)=\frac{(y_1-y)}{(x_1-x)} \quad (5)$$

$$\tan(\theta-\phi_1)=\frac{(y_2-y)}{(x_2-x)} \quad (6)$$

$$\tan(\theta-\phi_2)=\frac{(y_3-y)}{(x_3-x)} \quad (7)$$

Solving equations (5), (6) and (7) gives a unique solution, which involves simple arithmetic operations. Therefore, they can be programmed directly onto the processors. Thus, by using the above method, a camera sensor node is able to calibrate itself in a distributed fashion. Unlike the other calibration techniques that involve non-linear estimation methods, this part of the calibration technique does not require any guesses to solve for the external parameters of the remaining camera sensor nodes . Hence, this approach has an added advantage that the problem of local minimum due to bad initial guesses is avoided. Also, as the algorithm involves only simple arithmetic computations, the calibration can be performed at a relatively faster rate and with a less power consumption.

IV. EXPERIMENT EVALUATION

The wireless camera sensor network test bed we developed is shown in Figure 7. We evaluated the performance of image capturing and transmission. One of the images is shown in Figures 8.

The proposed camera calibration algorithms were implemented on the test bed shown in Figure 7. One node was placed at the origin of the reference frame and the other reference node was placed at a distance of 14 inches away from the origin. The reference nodes are always fixed. The camera sensor nodes are programmed to run the calibration algorithms. The third camera sensor node is placed at various positions and orientations on the test bed during the experimental evaluation. All the end devices send their calculated external parameters to the coordinator which directs them to the PC. On the PC, a MATLAB program is run to plot the experimentally obtained values against the original locations and orientations of the three camera sensor nodes. The results for two experiments are shown in the Figures 9.1 and 9.2. The statistical parameters are shown in Table I.

Table I: Statistical parameters for the pose of camera sensor node 3 in five experiments.

Parameter	X (inches)	Y (Inches)	θ (Degrees)
Mean Error	-0.220	0.9360	-1.8400
Standard deviation	0.796	0.8736	7.1618

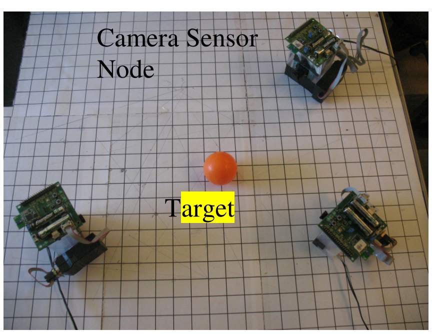

Figure 7: The camera sensor network.

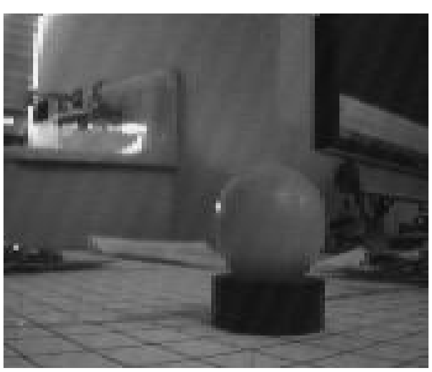

Figure 8: The image from camera 1.

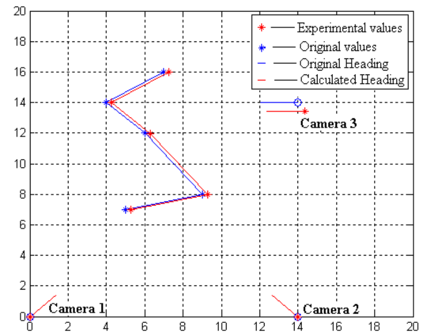

Figure 9.1: Result plot for Experiment 1.

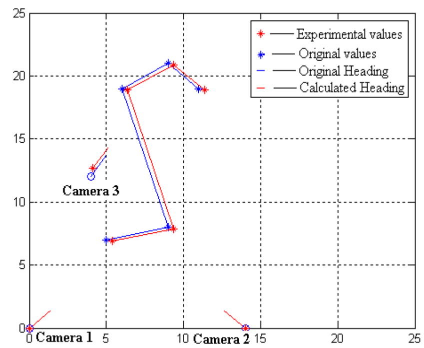

Figure 9.2: Result plot for Experiment 2.

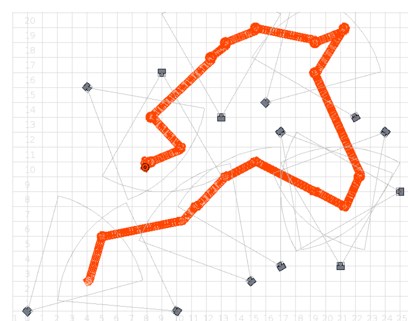

Figure 10.1: The path trajectory in simulation.

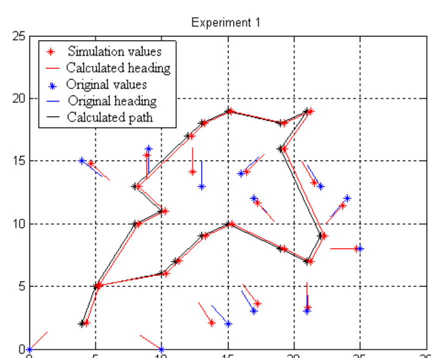

Figure 10.2: The result in simulation.

The correctness, reliability and scalability of the proposed calibration algorithms were validated by simulating large scale camera sensor networks using the Player/ Stage robotic simulation software and the simulation results are shown in Figure 10. The statistical parameters like mean and standard deviation were calculated and they are shown in the tables II, III and IV.

Table II: Statistical parameters for the coordinates of the cameras obtained on path 1

Statistical parameter	X value	Y value	θ value
Mean Error	-0.1131	0.0800	-0.2319
Standard Deviation	0.4867	0.4585	3.1069

Table III: Statistical parameters for the coordinates of the cameras obtained on path 2

Statistical parameter	X value	Y value	θ value
Mean Error	-0.0715	0.1438	1.2461
Standard Deviation	0.2849	0.5724	4.4407

Table IV: Statistical parameters for the coordinates of the cameras obtained on path 3

Statistical parameter	X value	Y value	θ value
Mean Error	-0.0246	0.1892	0.8935
Standard Deviation	0.3838	0.4286	4.7952

V. CONCLUSIONS AND FUTURE WORK

In this work, we have developed and evaluated a low power and low cost camera sensor network platform for monitoring and surveillance applications. We also propose a simple, light-weight algorithm to perform distributed camera calibration. Most of the existing calibration techniques have the disadvantage of complex optimization computations. Our calibration algorithm is computationally less intensive, faster and can be implemented in a distributed fashion. Currently, the serial communication between the camera and the sensor node is a bottleneck in performing image compression simultaneously with the color image reception. The next step would be to use some existing communication protocols like I2C and SPI to speed up the compression and the wireless image transmission processes. In large scale environments, the dead reckoning error will also accumulate with the increased target movement and hence may lead to large errors in the target localization. This may in turn lead to calibration errors by a collaborating camera sensor node. Hence, the target locations have to be refined simultaneously by a calibrated camera. This will lead to the Simultaneous Localization and Tracking problem (SLAT). A Kalman filter approach which provides a recursive solution to estimate the state of a dynamic process can be used to solve this problem.

ACKNOWLEDGEMENT

This project is partially supported by the US Department of Transportation, Research and Innovative Technology Administration grant DTRT06-G-0016.

REFERENCES

[1] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks”, *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] M. Paskin S. Funiak, C. Guestrin and R. Sukthankar, “Distributed localization of networked cameras”, In *ISPN*, 2006.

[3] T. Teixeria, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, A. Savvides, “A Lightweight Camera Sensor Network Operating on Symbolic Information”, *ACM SenSys Workshop on Distributed Smart Cameras*, 2006.

[4] W. Chen, P. Ahammad, C. Boyer, S. Huang, L. Lin, E. J. Lobaton, M. L. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L. C. Chang, D. Tygar and S. S. Sastry, “CITRIC: A Low-Bandwidth Wireless Camera Network Platform”, USC Berkeley, *Technical Report No*. UCB/EECS-2008-50 May 13, 2008

[5] M. Rahimi, R. Baer, J. Warrior, D. Estrin, and M. B. Srivastava. Cyclops: In-situ Image Sensing and Interpretation in Wireless Sensor Networks. In *SenSys 05*.

[6] I.Downes, L. Rad, and H. Aghajan. “Development of a mote for wireless image sensor networks”, In COGIS, 2006.

[7] T.A. Clarke and J.G. Fryer, “The development of camera calibration methods and models”, *Photogrammetric Record*, 16:51–66. 1998.

[8] F. Remondino, C. Fraser, “Digital camera calibration methods: considerations and comparisons”, *InternationalArchives of Photogrammetry, Remote Sensing and the Spatial Sciences*, 36(5), pp. 266-272, 2006.

[9] Z. Zhang, “A Flexible New Technique for Camera Calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22(11):1330–1334. NOVEMBER 2000.

[10] R. Y. Tsai, “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*, vol. 3, pp. 323-343. August 1987.

[11] 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,* pp. 401–406. 2005,

[12] P. Kulkarni, P. Shenoy, and D. Ganesan, “Approximate Initialization of Camera Sensor Networks”, *In Proceedings of the Fourth European Conference on Wireless Sensor Networks (EWSN).* 2007.

[13] H. Lee, H. Aghajan, “Collaborative Node Localization using opportunistic Target Observations”, *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks,* pages 9-18. October *2006*.

[14] J. M. Ortega and W. C. Rheinboldt, “Iterative Solution of Nonlinear Equations in Several Variables*”, New York: Academic Press*, 1970

[15] D.G. Taylor and Song Li, “Damped Gauss-Newton Method for Direct Stable Inversion of Continuous-Time Nonlinear Systems”, *Industrial Electronics Society (IECON '03). The 29th Annual Conference of the IEEE.* November 2003.

[16] X. Liu, P. Kulkarni, P. Shenoy, and D. Ganesan, “Snapshot: A Self-Calibration Protocol for Camera Sensor Networks”, *In IEEE/CreateNet BASENETS.* October 2006.

[17] C. Randell, C. Djiallis, and H. Muller, “Personal position measurement using dead reckoning”, *In Proceedings of The Seventh International Symposium on Wearable Computers.* October 2003.

[18] CMUCAM2. http://www.cs.cmu.edu/~cmucam2/. January 2007.

[19] JN5121 wireless micro-controller datasheet at http://www.jennic.com/. January 2007.

[20] Omnivision Technologies Incorporated, OV6620 Single-Chip CMOS CIF Color Digital Camera *Technical Documentation at http://www.ovt.com/.*

[21] G. Pekhteryev, Z. Sahinoglu, P. Orlik, and G. Bhatti, “Image transmission over IEEE 802.15.4 and zigbee networks*”, In IEEE International Sym- posium on Circuits and Systems (ISCAS 2005), volume 4, pages 3539–3542.* 23-26 May 2005.