

Analysis of the Accuracy-Latency-Energy Tradeoff for Wireless Embedded Camera Networks

Alvaro Pinto*, Zhe Zhang*, Xin Dong[†], Senem Velipasalar*, Mehmet C. Vuran[†] and M. Cenk Gursoy*

* Dept. of Electrical Engineering, University of Nebraska-Lincoln

E-mail: alvaro.pinto-albuja@huskers.unl.edu, zhe.zhang@huskers.unl.edu, velipasa@engr.unl.edu, gursoy@engr.unl.edu
http://vision.unl.edu/

[†] Cyber-Physical Networking Laboratory, Dept. of Computer Science and Engineering, University of Nebraska-Lincoln
Email: xdong@cse.unl.edu, mcvuran@cse.unl.edu

Abstract—Wireless embedded smart cameras provide flexibility in camera deployment in terms of the locations and number of the cameras. However, these battery-powered embedded vision sensors have very limited energy, memory, and processing power. Energy consumption and latency are two major concerns in wireless embedded camera networks. In multi-camera tracking applications, the amount of data exchanged between cameras has an effect on the tracking accuracy, the energy consumption of the camera nodes and the latency. In this paper, we provide a detailed quantitative analysis of the accuracy-latency-energy tradeoff for overlapping and non-overlapping camera setups when different-sized data packets are transferred in a wireless manner. The experiments have been performed with an actual wireless embedded smart camera network employing CITRIC motes, and performing tracking of objects.

I. INTRODUCTION

Wireless embedded smart cameras are stand-alone units that combine sensing, processing and communication on a single embedded platform [1]. Wireless embedded camera networks have promising applications in surveillance, traffic analysis and wildlife monitoring. Unlike wired camera systems, these cameras have very limited energy, processing power and memory.

Energy consumption and latency are two major concerns in wireless embedded camera networks. Methods and systems have been presented in literature to reduce energy consumption of the cameras, either for image transmission [6] [9] [14], or object detection and tracking [8] [5] [11]. For object detection and tracking, multi-tier camera sensor networks have been introduced to reduce energy consumption. Kulkarni et al. [8] introduced *SensEye*, which is a two-tier camera sensor network, compared the sensing reliability as well as the energy usage with one-tier systems. Another two-tier system is presented in [5], which employs cameras with different resolutions. Low resolution cameras continuously determine position, range, and size of moving objects and trigger high resolution cameras. High resolution cameras perform the subsequent image processing. This two-level structure is also used in [11], wherein a probabilistic algorithm is employed to reduce the sensing work of the lower level cameras. The aforementioned work focuses on two-tier structures and cameras with different resolutions. Overlapping and non-overlapping camera setups are not addressed, and energy-accuracy-latency

tradeoffs are not analyzed.

Margi et al. [10] analyzed the energy consumption of each basic task in the camera motes, such as processing, flash memory access, image acquisition and communication. Ko et al. [7] empirically study a camera sensor node which uses Scale Invariant Feature Transform (SIFT) to identify objects in the environment. They analyze the performance (classification accuracy, latency and energy consumption) of SIFT for visual classification on a Blackfin DSP processor. The simulation results are provided.

In this paper, we implement a multi-camera tracking algorithm on actual embedded smart cameras. We provide a detailed quantitative analysis of the accuracy-latency-energy tradeoff for overlapping and non-overlapping camera setups when different-sized data packets are transferred in a wireless manner. The details of the embedded camera platform are described in Section II. All the processing, including foreground detection and tracking, is performed on the microprocessor of the camera boards.

The rest of the paper is organized as follows: Section II introduces the embedded smart camera platform employed in our experiments. Section III gives a detailed description of the camera configurations and scenarios considered to analyze the energy consumption, latency and accuracy of the system. Section IV briefly discusses the communication protocols in different configurations. The experimental results and discussions are provided in Section V, and the paper is concluded in Section VI.

II. THE EMBEDDED SMART CAMERA PLATFORM

The wireless embedded smart camera platform employed in our experiments is a CITRIC mote [3]. It consists of a camera board and a wireless mote, and is shown in Fig. 1. The camera board is composed of a CMOS image sensor, a fixed-point microprocessor, external memories and other supporting circuits. The camera is capable of operating at 15 frames per second (fps) in VGA and lower resolutions. The microprocessor PXA270 is a fixed-point processor with a maximum speed of 624MHz and 256KB of internal SRAM. Besides the internal memory of the microprocessor, the PXA270 is connected to a 64MB of SDRAM and 16MB of NOR FLASH. An embedded Linux system runs on the camera board. Each camera board

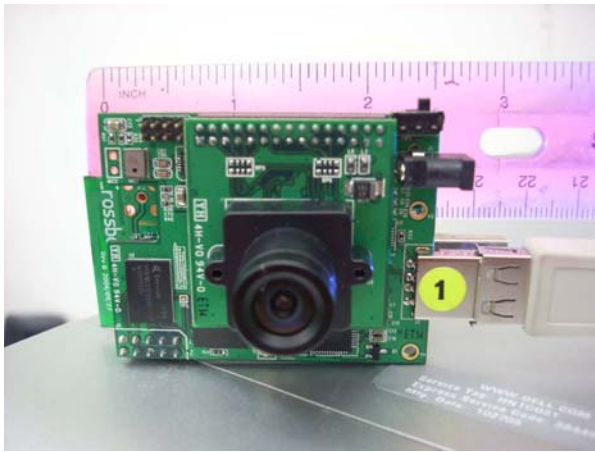


Fig. 1. The wireless embedded smart camera platform.

connects to a wireless mote via a serial port. The wireless mote employed is a TelosB mote from Crossbow Technology. The TelosB uses a Texas Instruments MSP430 microcontroller and Chipcon CC2420 IEEE 802.15.4-compliant radio [3]. The maximum data rate of the TelosB is 250kbps.

To measure the energy consumption of the platform, a National Instruments DAQ device is used. The voltage as well as the current of each mote is measured. Since totally 8 channels are used in the DAQ device (4 voltages and 4 currents), and the maximum total sampling rate of the device is 250K samples/s, the sampling rate of each parameter is 31250 samples/s. For the purpose of measuring communication delay, all the motes are connected to a PC using USB cables to record the communications among the motes and the time stamp of each packet.

III. ANALYSIS OF ENERGY CONSUMPTION, LATENCY AND ACCURACY

We have used two different camera configurations (partially overlapping and non-overlapping) and performed object tracking for different scenarios. Within these scenarios, different amount of data is exchanged to perform a detailed quantitative analysis of the accuracy-latency-energy consumption tradeoff. To this end, the energy consumption, accuracy, and latency are measured when tracking one or two objects with very similar or different colors.

In the following, the camera configurations and the deployed scenarios are described.

A. Camera Configurations

Two different camera configurations, i.e., partially overlapping and non-overlapping, are used for the experiments as described below.

1) *Partially Overlapping Cameras*: For this setup, we installed four embedded smart cameras (CITRIC motes) with partially overlapping fields of view as seen in Fig. 2. Since camera views are overlapping, they only exchange the location information of objects, specifically the x and y coordinates of the midpoint of the bottom line of the bounding boxes around the objects (the black cross shown in Fig. 2).

Consistent labeling of objects is performed as follows:

- (i) The homography matrices are estimated between camera pairs. When a new object enters its view, a camera uses the field of view (FOV) lines to determine which camera(s) can also see this object. We recover the FOV lines off-line as described in [12].
- (ii) The camera transmits a message addressed to the camera(s) that has already been tracking the object. This message contains the object's x and y coordinates and its temporary label.
- (iii) The receiving camera uses the homography matrix to convert the received point to its own coordinate system, and finds the object in its view that is closest to this point. It sends a reply packet to the requesting camera, and this packet includes the answer label and the received temporary label.

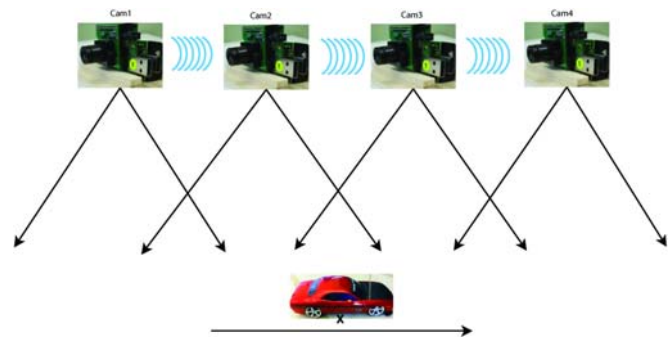


Fig. 2. Camera configuration with four partially overlapping cameras.

Figure 3 shows representative frames from an experiment with four overlapping cameras. The remote-controlled car enters the scene in the view of Cam1, and gets a new label 10. At time instant $t = t_1$, it enters the view of Cam2, and Cam2 gives it a temporary label 0. Then Cam2 gets the correct label 10 from Cam1 via location exchange. At time instant $t = t_2$, it enters the view of Cam3, and gets a temporary label 0. Then, Cam3 receives the correct label 10 from Cam2 via location exchange. Same steps happen after time instant $t = t_3$.

2) *Non-overlapping Cameras*: In the second configuration, the camera views do not overlap, i.e. there is a spatial discontinuity between cameras as shown in Fig. 4. In this case, cameras cannot perform consistent labeling by location exchange. Instead, larger amount of data needs to be transmitted to consistently track objects across different camera views.

Even if the cameras are initially installed with overlapping views, potential camera failures can cause non-overlapping camera setups and blind regions. Thus, analyzing this setup is important.

B. Tracking Scenarios

In the following, we describe several tracking scenarios that differ in terms of the type and amount of data transferred between cameras.

1) *Scenario I: Gray-level Histogram Exchange*: In this scenario, gray-level histograms are transferred to match objects

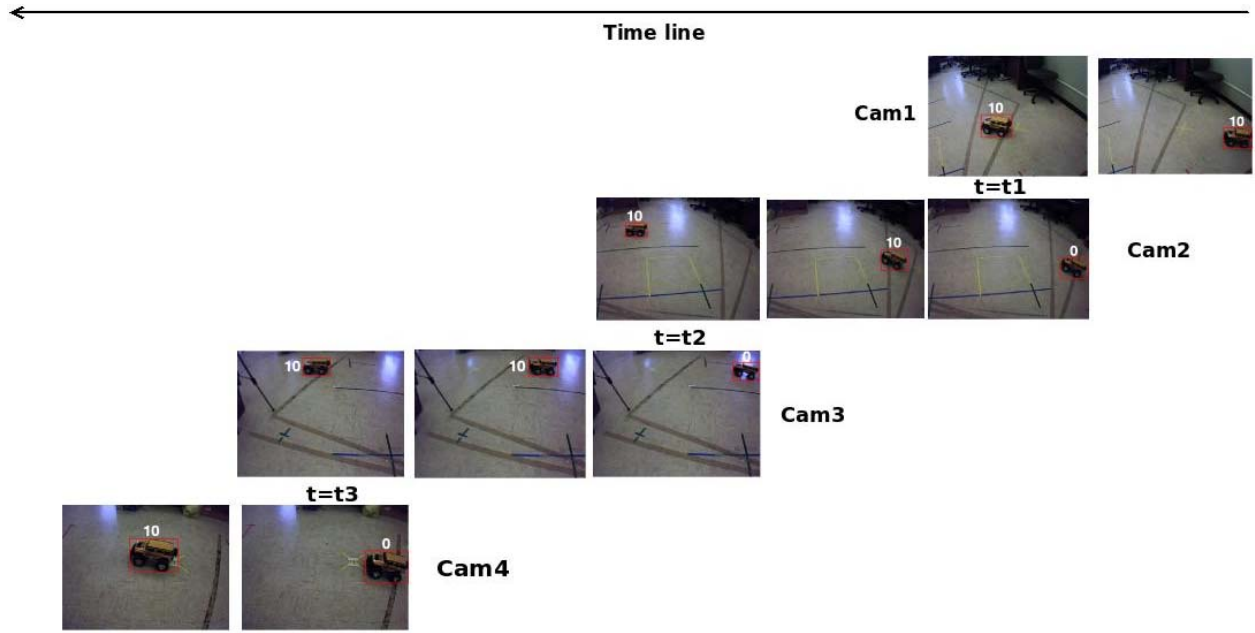


Fig. 3. Representative frames from the setup with four overlapping camera views.

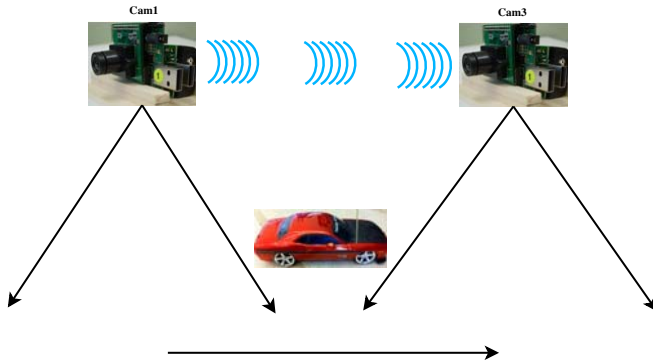


Fig. 4. Camera configuration with two non-overlapping cameras.

across non-overlapping cameras. For the histograms, we use 32 bins and the Y channel of the YUV color space. For a pixel, the value of the Y channel ranges between 0 and 255. The background subtraction and tracking algorithms running on the camera boards detect moving objects, build their histograms and track them on a camera's view. In order to detect the salient moving objects, we use the algorithm we presented in [2]. The details of the tracking algorithm can be found in [13].

In Fig. 4, after an object leaves the view of Camera 1, this camera saves the object's histogram in memory. When a new object enters into the view of Camera 3, this camera gives a temporary label 0 to this object, and transmits the object's gray-level histogram. The receiving camera (Camera 1) calculates the Bhattacharyya coefficient [4] between the received histogram and its saved histograms to find the best match. When the match score is larger than a predefined threshold, Camera 1 sends a reply packet that includes the label of the matched object. Camera 3 then assigns the received label to the object with the temporary label. The Bhattacharyya

coefficient is derived from the sample data by using:

$$\hat{\rho}(y) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u}, \quad (1)$$

where $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\dots m}$, and $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\dots m}$ are the probabilities estimated from the m-bin histogram of the model in the tracker and the candidate blobs, respectively.

In our system, cameras can save 10 object histograms in their memory. Under *Scenario I*, we analyzed the energy consumption and latency when tracking one and two objects. We also analyzed the accuracy when tracking two objects with similar brightness levels and two objects with different brightness levels. Results are discussed in Section V.

Figure 5 shows representative frames from an experiment with two non-overlapping cameras. Two remote-controlled cars are tracked in the view of Cam 1 first. Then, the car with label 11 leaves the view of Cam1, and after going through the blind region in between, it enters the view of Cam3. It first gets a temporary label 0. Cam3 sends the car's gray-level histogram, and Cam1 sends the correct label 11 back.

2) *Scenario II: Color Histogram Exchange*: In this scenario, each camera builds color histograms of the moving objects. Information from additional channels helps especially when tracking objects with similar brightness levels. For each channel (Y, U and V) the range of values is divided into 32 bins. This creates a 32^3 -dimensional array, and this generated histogram is usually sparse, which means that there are not too many nonzero values in the array. To decrease the memory requirement, the amount of the transmitted data and the energy consumption of the embedded smart cameras, we compress the color histogram before saving it and/or transmitting it over wireless channel. Only the nonzero values and the indices of the nonzero values in this array are transmitted.

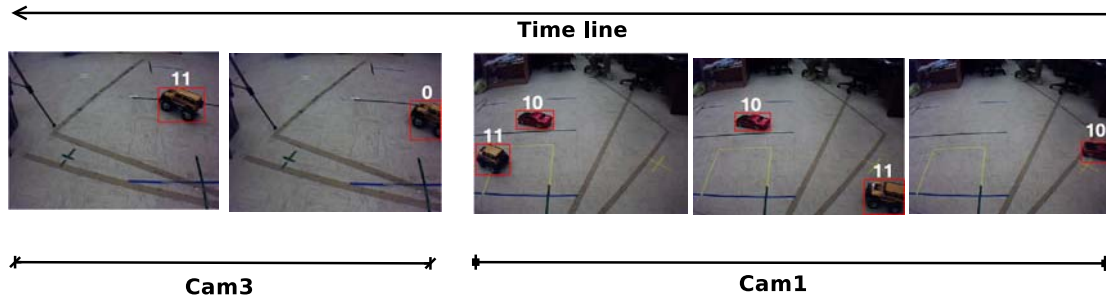


Fig. 5. Representative frames from a two non-overlapping camera setup.

We analyzed the energy consumption, accuracy and latency when tracking one or two objects, and when tracking objects with very similar or with different colors.

As will be discussed in Section V, there is a tradeoff between the amount of data transmitted and the energy consumption as well as the latency introduced. Also, when the amount of transmitted data increases, which means that richer image descriptors are transmitted across the cameras, the accuracy of consistent tracking also increases. However, the energy consumption also increases significantly.

IV. COMMUNICATION PROTOCOL

In the overlapping camera setup, when a new object enters into a camera's view, this camera transmits a message addressed to the camera(s) that has already been tracking the object. This message contains the object's x and y coordinates and its temporary label. The receiving camera sends a reply packet to the requesting camera, and this packet includes the answer label and the received temporary label. The payload of the request and reply packets are 4 and 2 bytes, respectively.

In Scenario I of the non-overlapping configuration, gray-level histograms are sent. The message packet contains 32-byte histogram information and the temporary label of the object. Thus, the total payload is 33 bytes.

In Scenario II of the non-overlapping configuration, compressed color histograms are transmitted and the size of the packet varies according to the number of nonzero entries in the histogram. Thus, it depends on the detected object. The TelosB allows sending 114 bytes of payload. Thus, the compressed histogram is divided into multiple packets. Moreover, only char type data can be sent through this packet structure. However, the indices of the nonzero entries in the 32^3 -dimensional array can have large values. Thus, the index information is divided into three different bytes. As a result, for each nonzero value in the array, 4 bytes of information is sent resulting in a total packet size of $4 \times N$ bytes, where N is the number of nonzero entries in the histogram.

V. EXPERIMENTAL RESULTS

In our experiments, we started with a 4-camera setup shown in Fig. 2, where cameras have partially overlapping fields of view. Objects enter the scene through the first camera's view, and are tracked across Cameras 1 through 4. We then studied the non-overlapping setup shown in Fig. 4 assuming

that Cameras 2 and 4 have failed. Objects are tracked across Camera 1 and 3. We analyzed the energy consumption, latency and accuracy of the system when tracking objects across different camera views. We repeated every experiment for every scenario 10 times and present the average value of the obtained results. This study provides a quantitative analysis of the accuracy-latency-energy tradeoff.

A. Energy Consumption

In Figure 6, the resulting energy consumption is shown for each camera during a time window that starts when a target object enters the camera's view and ends when the object leaves the camera's view. This amount includes the energy consumed during message exchanges. We also looked at cases when there were one or two objects in the scene. In Figure 6, the total energy consumption for all the cameras in the system can also be seen for overlapping and non-overlapping setups.

The results show that deploying four overlapping cameras and communicating less data provides 23.56% savings in energy consumption compared to using two non-overlapping cameras and exchanging compressed color histograms. The total energy consumption of Scenario I when tracking one object is 20.75% less than the 4-camera setup. However, as will be discussed in Section V-D, it results in 15.8% lower accuracy/reliability.

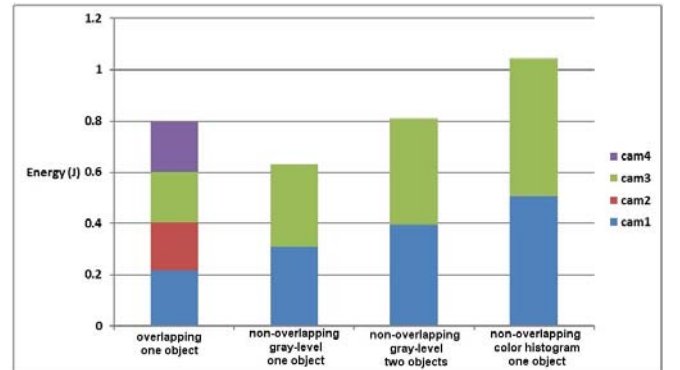


Fig. 6. Energy consumption for the 4-camera overlapping and 2-camera non-overlapping setups for different types of data transfer.

B. Average Power

In addition to energy consumption, we measured the average power consumption of the cameras for the camera configura-

tions and scenarios described in Section V-A. Figure 7 shows the obtained values.

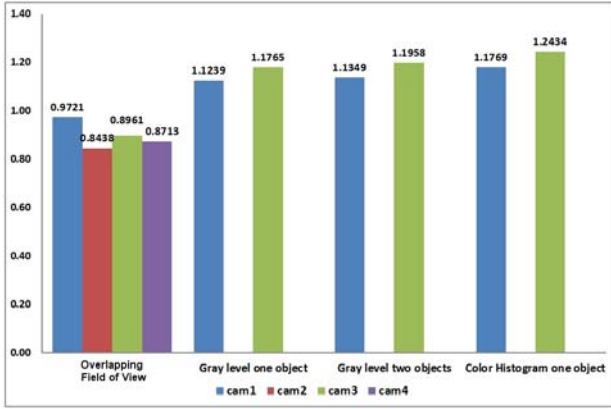


Fig. 7. Average power per camera.

C. Latency

We measured the latency from the time a camera detects a new object until the time it receives the answer label from another camera and assigns the received label to the newly detected object. Figure 8 shows the measured latency values for different scenarios. As can be seen, sending larger-sized packets introduces longer delays.

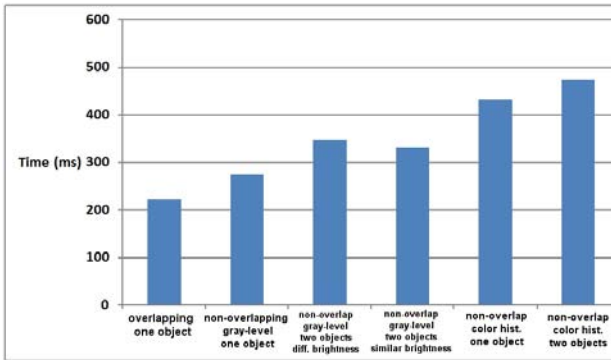


Fig. 8. Latency

D. Reliability

We measured the accuracy of tracking objects consistently across different cameras for overlapping and non-overlapping configurations. The results are displayed in Fig. 9. As seen in this figure, when using cameras with overlapping fields of view, the reliability is very high for uncrowded scenes. Since only location data is exchanged, the reliability will decrease for densely crowded scenes. Also, the reliability of using only gray-level histogram for non-overlapping cameras decreases especially when there are multiple objects in the scene with similar brightness values. Using all three channels increases the accuracy with the cost of higher delays and higher energy consumption.

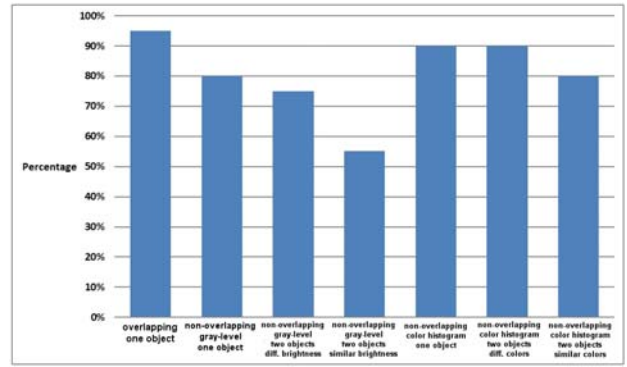


Fig. 9. Accuracy of correctly labeling an object across different camera views for the 4-camera overlapping and 2-camera non-overlapping setups.

VI. CONCLUSION

In this paper, we have presented a wireless embedded smart camera network for multi-camera tracking. In multi-camera tracking applications, the amount of data exchanged between cameras has an effect on the tracking accuracy, the energy consumption of the camera nodes and the latency. We have provided a detailed quantitative analysis of the accuracy-latency-energy tradeoff for overlapping and non-overlapping camera setups when different-sized data packets are transferred in a wireless manner.

The experiments have been performed with an actual wireless embedded smart camera network employing CITRIC motes and performing tracking of objects. Our results show that, for the studied scenarios, deploying four overlapping embedded cameras and communicating less data consumes 23.56% less energy than using two non-overlapping cameras and exchanging compressed color histograms. In addition, when using cameras with overlapping fields of view, the reliability is very high for uncrowded scenes. The reliability of using only gray-level histogram for non-overlapping cameras decreases especially when there are multiple objects in the scene with similar brightness values. Using all three channels increases the accuracy with the cost of higher delays and higher energy consumption.

ACKNOWLEDGMENT

This work was supported partly by the National Science Foundation grants CNS 0834753, CCF 0917265, and CA-REER award CNS-0953900.

REFERENCES

- [1] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds," Proc. IEEE October, vol. 96, pp. 1-18, October 2008.
- [2] M. Casares, S. Velipasalar and A. Pinto, "Light-weight salient foreground detection for embedded smart cameras," Computer Vision and Image Understanding, vol. 114, issue 11, pp. 1223-1237, November 2010.
- [3] P. Chen and et al., "Citric: A low-bandwidth wireless camera network platform", Proc. of the ACM/IEEE International Conference on Distributed Smart Cameras, 2008.
- [4] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142-149, 2000.

- [5] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "*Mesheye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance*," IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, pp. 360–369, 2007.
- [6] K. Hong, P. Yan, and P. Chen, "Poster Abstract : Multihop Routing in Camera Sensor Networks - An Experimental Study," Electrical Engineering, 2009.
- [7] T. Ko, Z. Charbiwala, S. Ahmadian, M. Rahimi, M. Srivastava, S. Soatto, and D. Estrin, "*Exploring tradeoffs in accuracy, energy and latency of scale invariant feature transform in wireless camera networks*," Proc. of the ACM/IEEE International Conference on Distributed Smart Cameras, pp. 313–320, 2007.
- [8] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu, "*Senseye: a multi-tier camera sensor network*," MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia, pp. 229–238, 2005.
- [9] D. Lee, H. Kim, S. Tu, M. Rahimi, D. Estrin, J.D. Villasenor, "*Energy-optimized image communication on resource-constrained sensor platforms*," Proc. of the International Symposium on Information Processing in Sensor Networks, pp. 216–225, April 2007.
- [10] C. Margi, V. Petkov, K. Obraczka, and R. Manduchi, "*Characterizing energy consumption in a visual sensor network testbed*," Proc. of the International Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities, 8 pages, 2006.
- [11] P. Skraba and L. Guibas, "*Energy efficient intrusion detection in camera sensor networks*," Distributed Computing in Sensor Systems, Lecture Notes in Computer Science, vol. 4549, pp. 309–323, 2007.
- [12] S. Velipasalar, J. Schlessman, C. Chen, W. Wolf, and J. Singh, "*A scalable clustered camera system for multiple object tracking*," EURASIP Journal on Image and Video Processing, 542808, 2008.
- [13] Y. Wang, S. Velipasalar and M. Casares, "*Cooperative object tracking and composite event detection with wireless wmbdedd wmart cameras*," IEEE Transactions on Image Processing, vol. 19, issue 10, pp. 2614–2633, October 2010.
- [14] H. Wu and A.A. Abouzeid, "*Energy efficient distributed image compression in resource-constrained multihop wireless networks*," Elsevier, vol. 28, pp. 1658–1668, 2005.