2010

Energy Consumption and Latency Analysis for Wireless Multimedia Sensor Networks

Alvaro Pinto
University of Nebraska-Lincoln, alvaro.pinto-albuja@huskers.unl.edu

Zhe Zhang
University of Nebraska-Lincoln, zhe.zhang@huskers.unl.edu

Xin Dong
University of Nebraska-Lincoln, xdong@cse.unl.edu

Senem Velipasalar
University of Nebraska-Lincoln, velipasa@engr.unl.edu

Mehmet C. Vuran
University of Nebraska-Lincoln, mcvuran@cse.unl.edu

See next page for additional authors

Authors
Alvaro Pinto, Zhe Zhang, Xin Dong, Senem Velipasalar, Mehmet C. Vuran, and M. Cenk Gursoy

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

2010 IEEE Global Telecommunications Conference GLOBECOM 2010
Pages 1 - 5. DOI: 10.1109/GLOCOM.2010.5683898

Energy Consumption and Latency Analysis for Wireless Multimedia Sensor Networks

Alvaro Pinto*, Zhe Zhang*, Xin Dong†, Senem Velipasalar *, M. Can Vuran†, M. Cenk Gursoy*
* Electrical Engineering Department, 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
†Dept. of Computer Science and Engineering, University of Nebraska-Lincoln
E-mail: xdong@cse.unl.edu, mcvuran@cse.unl.edu

***Abstract*—Energy and bandwidth are limited resources in wireless sensor networks, and communication consumes significant amount of energy. When wireless vision sensors are used to capture and transfer image and video data, the problems of limited energy and bandwidth become even more pronounced. Thus, message traffic should be decreased to reduce the communication cost. In many applications, the interest is to detect composite and semantically higher-level events based on information from multiple sensors. Rather than sending all the information to the sinks and performing composite event detection at the sinks or control-center, it is much more efficient to push the detection of semantically high-level events within the network, and perform composite event detection in a peer-to-peer and energy-efficient manner across embedded smart cameras. In this paper, three different operation scenarios are analyzed for a wireless vision sensor network. A detailed quantitative comparison of these operation scenarios are presented in terms of energy consumption and latency. This quantitative analysis provides the motivation for, and emphasizes (1) the importance of performing high-level local processing and decision making at the embedded sensor level and (2) need for peer-to-peer communication solutions for wireless multimedia sensor networks.**

I. INTRODUCTION

Energy and bandwidth are limited resources in wireless sensor networks. When wireless vision sensors are used to capture and transfer image and video data, the problems of limited energy and limited bandwidth become even more pronounced, since the amount of data to be handled is much larger compared to scalar sensors [1]. In addition, communication consumes significant energy. Frequent transfer of large-size data requires more power and incurs more communication delay. In many systems, communication is 100 to 1000 times more expensive in energy than computation [5]. Thus, our goal is to reduce the communication cost by decreasing the amount of message traffic.

In many applications, the interest is to detect composite and semantically higher-level events based on information from multiple sensors. In existing multimedia sensor network setups [2], each primitive event detected by multimedia nodes are sent to a sink, most probably in a multi-hop manner. Accordingly, the sink or a control center combines information from multiple sensors to make higher-level decisions. In addition, local aggregation can be performed at aggregation points along the path between multiple sensors and the sink. However, event superposition that includes information from spatially distant sensors can only be performed at the sink. In case these composite events are not required to the end user, this creates highly redundant message traffic, consumes a lot of energy, and may overload sink nodes. In addition to the sensor sensing the primitive event, sensors on the multi-hop route also consume energy. Hence, our goal is to push the detection of semantically high-level events within the network, and perform composite event detection in a peer-to-peer (P2P) manner across the heterogeneous and embedded sensors. Accordingly, message traffic, and thus the overall energy consumption of the network will be significantly decreased.

In this paper, we analyze three different operation scenarios for a heterogeneous sensor network consisting of scalar sensors (for motion detection) and embedded smart cameras (Fig. 1). In the first setup, a scalar sensor wakes up the camera mote when it detects motion in the scene. Then, the camera captures a frame, and then, transmits the whole image frame in a multi-hop manner to a sink node. In the second setup, cameras perform local processing and send the images to the sink only if a primitive event is detected. Finally, in the third setup, cameras perform local processing, one camera communicates with another in a P2P manner to detect a composite event, and only when the composite event is detected, they transmit the interesting portion of a frame to the sink. All three operation scenarios are described in detail in Section III. We present a detailed quantitative comparison of these scenarios in terms of the energy consumption when the goal is detecting a composite and semantically high-level event. In addition to providing motivation for and emphasizing the importance of pushing the high-level decision making to the sensor level, this analysis gives quantitative results in

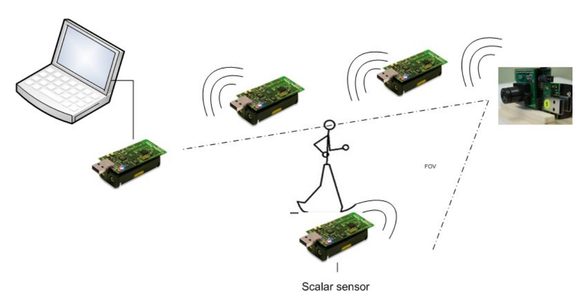

Fig. 1. Heterogeneous Wireless Multimedia Sensor Network

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

terms of savings in energy. We also present a latency analysis for these operation scenarios. The results highlight the need for efficient peer-to-peer communication solutions for wireless multimedia sensor networks (WMSNs).

Using heterogeneous sensors provides energy savings by keeping the low-power scalar sensors active for monitoring, and more power-consuming embedded smart cameras in idle mode until scalar sensors detect an activity. In our testbed, we use CITRIC motes [3] as our embedded smart cameras. A TelosB is attached to the camera boards for wireless communication. The camera board runs with 4 AA batteries, while the TelosB uses 2 AA batteries. We broadcast trigger messages from stand alone TelosBs to emulate the waking up of the cameras by scalar sensors.

II. Wireless Multimedia Sensor Network

A. Embedded Vision Sensors

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. 2. 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. The embedded Linux system includes the JPEG compression library, which provides the advantage of saving the video frames in JPEG format. Each camera board connects to a wireless mote via a serial port.

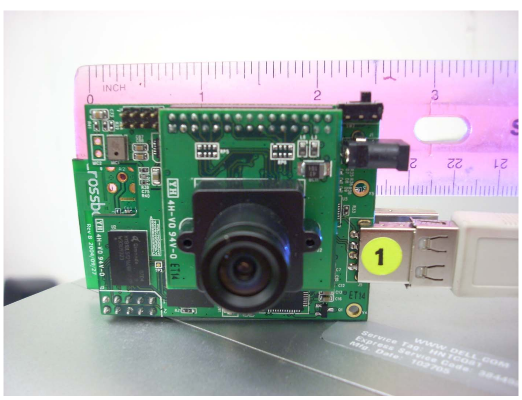

Fig. 2. The wireless embedded smart camera platform.

B. TelosB: The Wireless Mote

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.

The TelosB motes are used both as a scalar sensor using vibration sensors to detect movement in an indoor environment and are also attached to the CITRIC motes to provide wireless connectivity between vision sensors and scalar sensors. As stated above, trigger messages are broadcast from scalar sensor TelosBs to wake up the cameras by scalar sensors.

III. Operation Scenarios for Event Detection

In this section, we describe three different operation scenarios for detecting a composite and high-level event. We consider the event of interest to be a composite event that can be detected by two vision sensors. Accordingly, the composite event is detected if (1) a large vehicle is detected entering a facility through the entrance watched by camera A, *and then* (2) the same vehicle is detected as parking in a region defined in the view of camera B. It is assumed that cameras A and B have partially overlapping fields of view.

In all the operation scenarios, scalar sensors are always active, and camera sensors are idle to save energy. If/when motion is detected, a scalar sensor wakes up nearby camera sensors by broadcasting a trigger message.

A. Scenario 1: No Local Processing

As mentioned previously, in most existing sensor network setups, individual sensors transfer the captured data to a sink node and/or control center for further processing. To analyze the cost attached to this type of operation, we implement the first scenario, wherein camera sensors do not perform any local processing. After receiving the broadcast image from a scalar sensor, the processor activates the camera board and the sensor warms up. The camera captures a frame and sends the complete image to a sink node (Fig. 1) by multi-hop communication. The captured image size is 320×240 and it is transmitted in gray scale format after JPEG compression. This scenario serves as a baseline for the following two scenarios.

In this operation scenario, every time an object enters the facility or every time motion is detected, the scalar sensor will wake up the camera, and the camera will transmit the whole frame to the sink node. It should be noted that even though the interest is detecting large vehicles, this way of operation will cause an image transfer every time motion is detected, since no local processing is performed.

B. Scenario 2: Low-level Detection

In this scenario, the embedded camera sensor not only captures images, but also performs local processing. Specifically, it performs foreground detection, and then computes the size of the detected object(s).

As stated above, the event of interest is to detect a large vehicle, which enters a facility through the entrance watched by camera A, *and then* parks in a region defined on the view of camera B. In this operation scenario, after the camera wakes up, it performs background subtraction to detect the moving object, and then computes its size. If the size of the detected foreground object is larger than a threshold, the camera transmits only the portion of the image containing the object. This way of operation provides savings in two different ways. First, event messages are not transmitted every time motion is detected. Instead, cameras transmit images

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

only if the size of the detected object satisfies a certain criteria. Second, the cameras only transmit the portion of the image containing the object, instead of the whole frame. This scenario serves as the state-of-the-art in WMSNs.

C. Scenario 3: P2P Composite Event Detection

Cameras perform local processing in this mode as well. If a composite event is defined as a sequence of primitive events across multiple camera views, the first camera in this sequence transmits a message addressed to the next camera when it detects the first primitive event.

The event of interest described above can be defined as a sequence of two primitive events. The first primitive event is detecting the entrance of a large vehicle on the view of camera A. The second primitive event is detecting that vehicle parked in the region specified in the view of camera B. When camera A detects that a large vehicle entered into the scene, it transmits a message addressed to camera B, instead of transmitting a portion of the image to a sink. Compared to the second scenario, P2P composite event detection avoids redundant communication, since the application is not interested in every large vehicle entering the facility. Instead, a higher-level composite event is of interest. If camera B detects the second primitive event, only then an image portion will be transmitted to a sink.

IV. Experimental Results

In this section, we present the results of a detailed analysis of the energy consumption and latency of the three operation scenarios described above.

A. Energy Consumption

We measure the energy consumption in each scenario during different parts of the operation including warming up of the camera, processing a frame, and transmitting data. We also measure the energy consumption of the forwarders in multi-hop communication to obtain the overall energy consumption caused by each scenario. For all the results presented below, the communication between a camera sensor and the sink is performed in two hops.

Figure 3 shows the overall energy consumption of different operation scenarios. Scenarios 1, 2 and 3 in this figure are the operation scenarios described in Sections III-A, III-B and III-C, respectively. In Scenario 1, the camera does not perform local processing of the frame, but transmits the whole image frame (320×240) to the sink by two-hop communication. The total energy consumption for this scenario including the energy consumption of the forwarding node is 16.67 J. Figure 4 shows the distribution of the energy consumption among different components. Since no local processing is performed to make decisions, and the whole frame is transferred to the sink, the image data transfer causes the largest energy consumption, i.e., 58.9%.

In Scenario 2-A, the camera performs local processing to detect foreground objects and to determine their sizes. Since the size of the detected object does not satisfy the specified criteria, the camera does not transmit anything. The overall energy consumption of the camera, including the energy consumption during warming up, frame capturing, foreground detection and size check, is 6.1 J. In Scenario 2-B, the size of the detected object satisfies the specified criteria, and the camera sends only the portion of the image that contains the object to the sink. The size of this image portion is 50×50. As seen in Fig. 3, the total energy consumption for this scenario including the energy consumption of the forwarding node is 8.19 J, which is significantly less compared to Scenario 1. Figure 5 shows a distribution of the energy consumption among different components. Compared to Fig. 4, this way of operation provides a significant decrease in the energy consumption caused by the image data transfer.

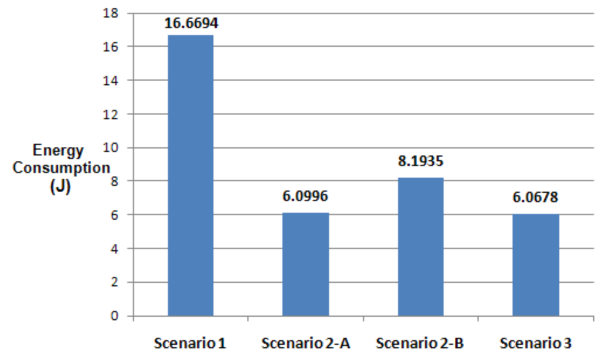

Fig. 3. The overall energy consumption for different scenarios.

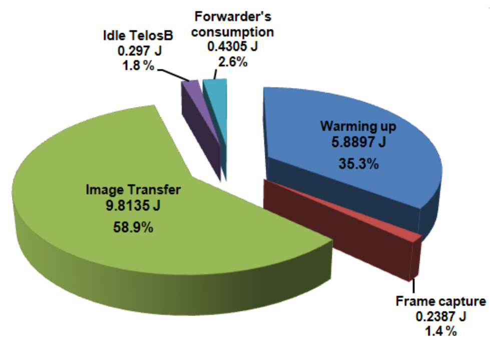

Fig. 4. **Scenario 1:** The distribution of the consumed energy.

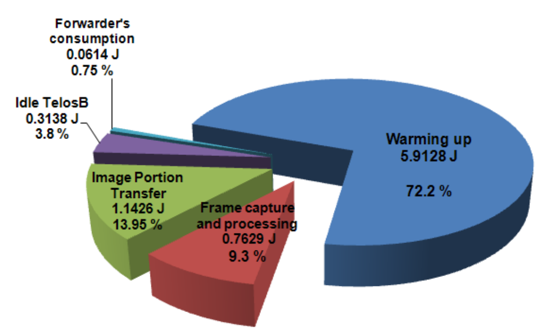

Fig. 5. **Scenario 2-B:** The distribution of the consumed energy in the second operation scenario when only a portion of the image is transmitted.

Fig. 6(a) and 6(b) show the operating currents of the camera

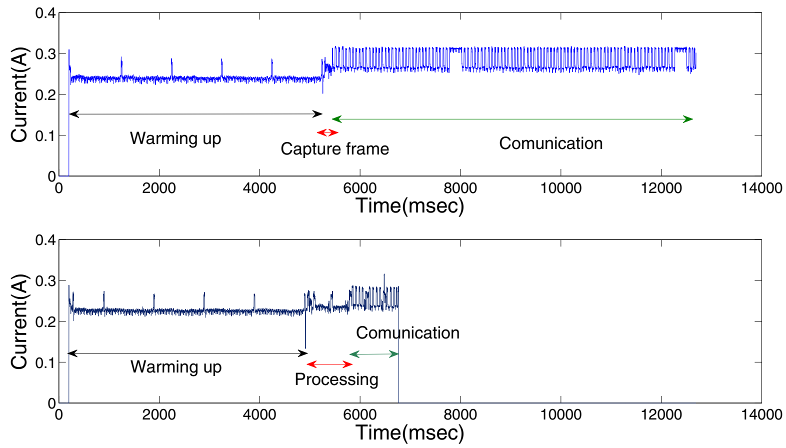

Fig. 6. Operating current of the camera board when transmitting (a) the whole frame, (b) only the portion of the image containing the detected object.

board during Scenario 1 and Scenario 2-B, respectively. As can be seen, when local processing is performed, and the interesting portion of the image is extracted and transmitted, the amount of latency, and the energy consumption due to image data transfer decrease significantly.

In Scenario 3, the camera again performs local processing to detect foreground objects and to determine their sizes. If the size of an object satisfies the specified criteria, the camera sends (in a single hop) a small-size packet to the second camera, which is responsible for detecting the second part of a composite event. This packet contains the label information of the tracked object. Since cameras have partially overlapping fields of view, they can track objects with consistent labels, and can determine if the same object is performing the primitive events in a composite event scenario. The energy consumption of the camera caused by warming up, frame capturing, frame processing and data transfer is 6.07 J. Figure 7 shows the distribution of the energy consumption among different components.

This analysis provides the motivation for pushing the semantically high-level event detection and decision making to the sensor level by providing quantitative results. It should be noted that the amount of the saved energy becomes much more significant and apparent when we consider the actual composite events that we are interested in. Consider the event of interest described above, where we want to detect large vehicles entering a facility through the entrance watched by camera A, *and then* parking in a region defined on the view of camera B. Assume that during a day, 10% of the objects (people, cars, trucks, bikes) entering the facility are large vehicles. Also assume that only 10% of the large vehicles entering the facility actually park in the restricted region defined on the view of camera B.

Let N be the number of objects entering the facility. In Scenario 1, the camera A will wake up, and transmit the complete image to the sink N times. Thus, its energy consumption will be approximately $N \times 16.24$ J. The forwarder's energy consumption will be $N \times 0.43$ J. In Scenario 2, camera A will wake up N times, but $\frac{N\times 9}{10}$ many times it will not transmit anything, since the size of the object will not be large enough (assuming the object detection and size check does not fail). It will transmit only the portion of the image containing the object $\frac{N}{10}$ times. The energy consumption of camera A in this case will be approximately $\frac{6.1\times N\times 9}{10} + \frac{8.1\times N}{10} \simeq 6.3 \times N$. In Scenario 3, camera A will wake up N times, and will send a message packet to camera B, $\frac{N}{10}$ many times. Thus, the energy consumption of camera A will be $\frac{6.1\times N\times 9}{10} + \frac{6.06\times N}{10} \simeq 6.09 \times N$. Camera B will transmit an image only $N/100$ times.

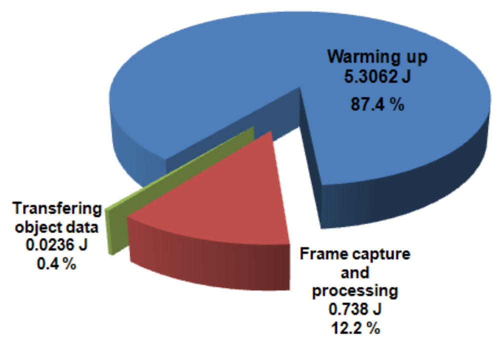

Fig. 7. **Scenario 3:** The distribution of the consumed energy when the first camera sends information about the object to the second camera.

Thus compared to Scenario 1, Scenario 2 and Scenario 3 provide 61.21% and 62.5% savings, respectively, in the energy consumption of camera A. In addition, Scenario 1 involves an image transfer N times. In Scenario 2, an image portion is transferred $N/10$ times, and in scenario 3 an image portion is transferred only $N/100$ times.

B. Latency

We also measured latency introduced during these different operation scenarios. It should be noted that in all latency measurements, the measured time intervals include the warming-up time of the camera sensor, which is around 5 sec.

For the first scenario described in Section III-A, we measured the time interval from camera waking up to a sink node receiving the complete image. The distance between the camera sensor and the sink node is 12 meters, and communication is performed in two hops. The file size for the whole image is 9.5 kB. After camera wakes up, it captures a frame, compresses it, and sends the whole frame to the sink node. We performed this experiment five times, and took the average of all measurements. The average time obtained is 11.85 sec.

For the second scenario described in Section III-B, we measured two different latencies. In the first case, camera wakes up, performs foreground object detection, and determines the size of the detected object. If the size of the object is not large enough, the camera does not transmit anything. We measured the time interval between camera waking up and determining if the size of the object satisfies the criteria. This was repeated five times. The average measured time interval is 5.36 sec. If the camera determines that the size of the detected object satisfies the specified criteria, then it transmits only the portion of the frame containing the object. For this case, we measured the time interval starting from camera waking up, including processing of the frame, and ending when the sink receives the transmitted portion of the image completely. Again, the distance between the camera and the sink is 12 meters, and communication is performed in two hops. The size of the image portion that is transmitted is 50×50, and the file size is 1.8 kB. The average measured latency for this case is 6.24 sec.

In the third scenario described in Section III-C, camera A performs foreground object detection, and determines the size of the detected object. If detected object is large enough, camera A sends a message addressed to camera B containing the label of the detected object. The average measured time from camera A waking up to camera B receiving the message packet is 5.51 sec. Camera A and camera B communicate in single-hop and the average measured latency for communication between them is 0.33 sec.

Figure 8 shows the amount of time it takes to complete warming up, processing and communication for all three scenarios.

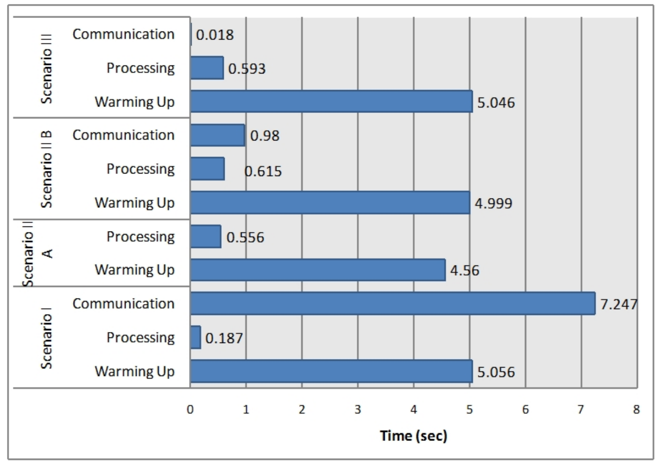

Fig. 8. The latencies of different components of operation for different scenarios.

V. Conclusions

In this paper, we have presented and analyzed three different operation scenarios for a wireless multimedia sensor network. In the first scenario, a scalar sensor wakes up the camera mote when it detects motion in the scene. The camera captures a frame, but does not perform any local processing of the image. It transmits the whole image frame in a multi-hop manner to a sink node. In the second scenario, cameras perform local processing to detect foreground objects. The camera transmits only when the size of the detected object satisfies a specified criteria (for instance to detect large vehicles). In addition, the camera does not transmit the whole frame, but only transmits the portion containing the object. In the third scenario, after performing local processing of the image, one camera communicates with another one in a P2P manner to detect a composite event, and only when the composite event is detected, they will transmit the interesting portion of a frame to the sink. We have presented a detailed quantitative comparison of these three scenarios in terms of the energy consumption and latency when the goal is detecting a composite and semantically high-level event. In addition to providing motivation for and emphasizing the importance of pushing the high-level decision making to the sensor level, this analysis gives quantitative results in terms of savings in energy.

Acknowledgment

This work was supported partly by the National Science Foundation grants CNS 0834753 and CCF 0917265.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A Survey on Wireless Multimedia Sensor Networks,” Computer Networks (Elsevier) Journal, vol. 51, no. 4, pp. 921–960, March 2007.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “Wireless Multimedia Sensor Networks: Applications and Testbeds,” Proceedings of the IEEE, vol. 96, no. 10, pp. 1588–1605, October 2008.

[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, pp. 1–10, 2008.

[4] D. Lee, H. Kim, S. Tu, M. Rahimi, D. Estrin, J.D. Villasenor, D.S. Engineering, and D. Tools, “Energy-Optimized Image Communication on Resource-Constrained Sensor Platforms,” IPSN07, pp. 216–225, April 2007.

[5] B. Rinner, W. Wolf, “An introduction to distributed smart cameras,” Proceedings of the IEEE, vol. 96, no. 10, pp. 1565–1575, 2008.

[6] H. Wu and A. A. Abouzeid, “Energy efficient distributed image compression in resource-constrained multihop wireless networks,” Elsevier, vol. 28, pp. 1658–1668, 2005.