

Event-Based Imaging with Active Illumination in Sensor Networks

Thiago Teixeira, Andreas G. Andreou
Electrical & Computer Engineering
Johns Hopkins University
Baltimore MD 21218

Eugenio Culurciello
Electrical Engineering
Yale University
New Haven CT 06520
eugenio.culurciello@yale.edu

Abstract—We discuss a distributed imaging architecture with active illumination for sensor network applications. An event-based CMOS imager is employed at the sensor level, to convert light intensity at each pixel into pulse density modulated stream of address events. The wireless nodes are commercial off-the-shelf Motes. Energy-aware communication is implemented at the sensor level by employing an event-based readout. Additional computation for data reduction is accomplished at the sensor/mote interface level by modulating the event-rate produced by the sensor array to match the bandwidth and latency constraints in the communication network. Information transmitted in the limited bandwidth links of the network yields effective means for detection and partial recognition of the object even at very low bit rates and frame latency as low as 1s.

I. INTRODUCTION

Sensor network nodes must operate under strict power budgets, as dictated by the need to prolong battery life [1]. This requirement calls for a power-conscious design from high-level algorithms down to the circuit implementation. While today a majority of sensor network applications are aimed at sensing and communication of scalar values of data such as temperature, pressure and humidity [2], [3] future networks need to address the challenging problem of sensing and communicating vector data such as images [4], [5]. It should be pointed out that more often than not, the goal of the information processing in such systems is not the precise restitution of the vector data (such as in the scalar sensing case), but rather the extraction of relevant information in a timely manner. It is about finding answers just-in-time locally and co-operatively to questions such as:

- Is there something interesting in the environment? (detection) In a specific class of objects (identification)
- Where is it? (location)
- What is it? (recognition)

In this paper, we report on an architecture for "eyes" in sensor networks. The system employs event-based image sensor (ALOHA imager) [6] and COTS motes forming a sensor network [7] using standard *TinyOS* software interfaces. By actively illuminating the scene through the sensor network infrastructure, the delivery and capture of photons is done in context and in a situated environment, thus spending resources only when necessary.

II. EXPERIMENTAL TESTBED

A. Sensor Network Hardware

The current experimental setup comprises of three nodes, as shown in Figure 1. The network is formed using commercial-off-the-shelf (COTS) wireless radio nodes (Motes) from Crossbow [7]. The Mica2 based transmitter node includes, an address-event image sensor (ALOHA imager) [6], and a prototype interface board. A second node (illuminator) consisting of a *Mica2Dot* mote with a single low-voltage incandescent bulb provides active illumination of the subject. The receiver is a *Mica2* mote, connected to the base station computer

through the MIB500 programmer board from Crossbow. Additional nodes for active illumination or image sensing can be added to the system; a 20 node network with a Linux based gateway is under deployment. The Motes operate at a carrier frequency of 916MHz.

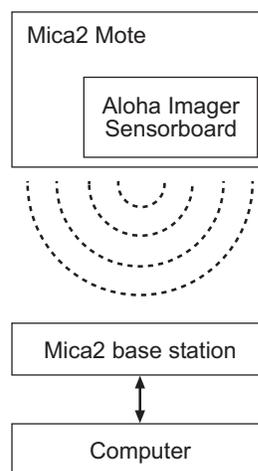


Fig. 1. A schematic of the prototype system.



Fig. 2. Mica2 mote with interface board and event-based CMOS imager.

The Mica2 mote was chosen as sensor network node for being the fastest mote available at the time, as well as the one with the highest number of available input/output interface lines. A system-specific prototype board (sensorboard) was fabricated in order to interface the mote to the image sensor (see Figure 2). The sensorboard consists of direct electrical connections from the image sensor's pins to the appropriate input/output pins on the Mica2-standard DF9B-51S socket [7]. The sensorboard also contains a voltage regulator and passive components used to power the image sensor.

B. Event-based CMOS Imager

The 32×32 -pixel ALOHA image sensor operates using the address-event protocol, with each pixel producing an event as soon as it has collected a predefined amount of light [6]. When this occurs, the image sensor outputs the 10 bit address of the pixel (5 bit for the X coordinate and 5 bit for the Y coordinate) using a standard acknowledge/request protocol. The image sensor latches the address onto the data line and signals a request, waiting for an acknowledge signal before sending the next pixel. This sequence of events is represented graphically in Figures 3 and 4.

The spatial resolution, dynamic range and latency of the sensor can be programmed at operation time thus making it a very flexible device with digital interface. This sensorboard is aimed at replacing the simple CdS photoconductive cells that are available on the Mica sensor boards from commercial vendors such as Crossbow, which consume many orders of magnitude more power and require an analog to digital converter [7].

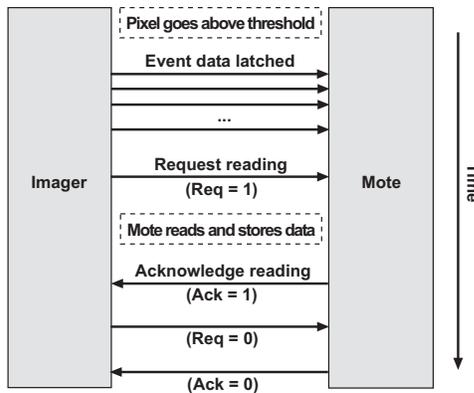


Fig. 3. Overview of ALOHA control signals.

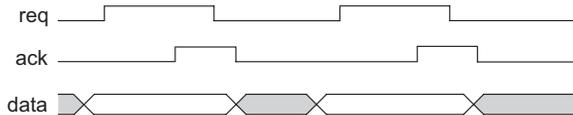


Fig. 4. Time diagram of ALOHA image sensor control signals.

C. Software

We developed a TinyOS 1.1.0-compatible program to collect data from the image sensor and supervise the transmission of information through the mote. This is loaded onto the Mica2's Atmega microcontroller through the MIB500 programmer/acquisition board. The program, as shown in Figure 5, operates in two phases: first, a *data gathering* phase takes place, in which the mote reads and stores the address of the first 500 pixels to signal an event. Currently, a pixel is stored as many times as it fires an event. This is a fairly crude collection system, and can be improved to further optimize the transmission process.

Successively a *data transmission* phase follows, when the mote broadcasts, 28 bytes at a time (plus a standard TinyOS 5-byte standard header), the data collected in the previous phase. The number of transmitted bytes (28) is the maximum even number from the default maximum of 29 bytes for TinyOS radio transmission, since each event takes 2 bytes for X and Y address. This maximum can be increased by tweaking some TinyOS system files, but this is outside the purpose

of this paper for this fairly non-standard procedure would need to be applied with each new version of TinyOS.

At the receiving end a *base* mote is loaded with the default TOSBase software from the TinyOS installation, which sends all incoming radio data packets to the serial port of the computer. A custom Java program (*CamRead*) running on the base computer listens to the serial port for TinyOS packets and extracts from them the addresses of the event-sender pixels and updates a real-time histogram array representing the number of events signaled by each pixel. Once a user-specified number of pixel events are collected, the program normalizes the histogram data into a 0 to 255 bit array which is then displayed as a monochrome image on the screen.

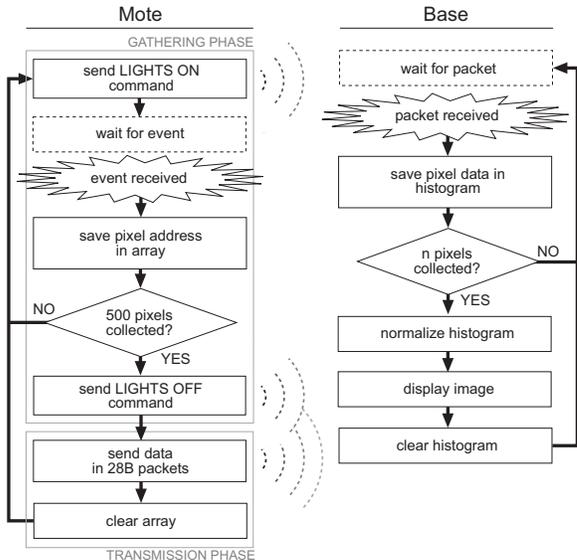


Fig. 5. Flow diagram of the imaging mote's TinyOS program (left) as well as *CamRead*, Java program running at the base computer (right).

In the active illumination mode of operation, the event-based CMOS imager, through the transmitting mote, communicates with the illumination node and turns on a light source when the image sensor is about to collect data. The light source is turned off after the image has been collected, during radio communication with the base mote.

III. RESULTS AND ANALYSIS

The number of events collected from the image sensor can be specified by the user from the *CamRead* Java program user interface. This allows one to tailor the number of samples to the desired image quality and the specific scene to be imaged. The ideal value of collected event depends heavily on the subject imaged, as does the frame-rate. This is a consequence of the image sensor's address event architecture. The sensor in fact produces few events in low-lighting conditions, and a high number of events in presence of bright illumination.

The collected data (when imaging the object in Figure 6) as a function of events is reported in Figure 7. With as little as 100 events it is possible to recognize simple shapes in the generated picture, which is usually black and white due to the small probability of more than one event coming from the same pixel. As the events-per-frame ratio increases, this probability also increases, giving rise to different shades of grey. This is already noticeable in the image generated from 300 events. There is very little difference between



Fig. 6. Subject to be captured by the image sensor.

1000, 1500 and 2000 events per frame, and although at 5000 events many more shades are generated, the frame-rate falls dramatically. The encoding of the events here is performed by means of a pulse density modulation scheme.



Fig. 7. Received image as a function of events. From left to right, top to bottom: 100, 200, 300, 500, 1000, 1500, 2000, 5000 events. Captured with ambient lighting.

Figures 8 and 9 show the linear relation between the frame latency (the time required to display one frame) and the number of events, averaging at $7.3ms$ per event. Our tests with the Mica2 data-rate show that the Mica2 motes can send 29-byte packets (the maximum size allowed by TinyOS) plus a 5-byte header at $28.6packets/s$ ($6.629kbps$ or $0.829kBps$). This enables us to derive an empirical relationship for the frame latency as specified by Equation 1.

$$\frac{1s}{829B} \times 2 \left(1 + \frac{5B}{29B} \right) \frac{B}{event} = 5.1ms/event. \quad (1)$$

where the multiplier $2 \left(1 + \frac{5B}{29B} \right)$ distributes the header overhead to each 2-byte event.

The experimental frame latency and expected lower-bound, as calculated in equation 1, are shown in Figure 8. The dashed line shows the expected limits while the solid line shows the trend calculated from the data-points, and follows Equation 2.

$$y = 0.0066s/event^2 \times x \text{ events} + 0.42s/event. \quad (2)$$

The difference between experimental and expected values can be explained by the presence of the gathering phase, during which the mote transmits no data at all. Equation 2 evidences this assumption in the presence of the y -intersect of $0.42s/event$. The duration of the data gathering phase (and magnitude of the y -intersect) depends on the amount of light available, and faster speeds can be achieved with brighter subjects. The subject for this experiment (Figure 6) was mostly black, yielding higher frame latency.

Figure 9 shows the same data as Figure 8, with the y -axis representing the frames transmitted per second.

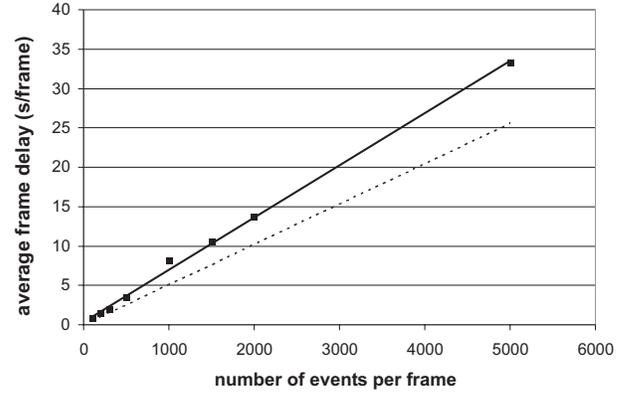


Fig. 8. Frame latency (the time required to display one frame) as a function of events/frame. The solid trendline is shown with a solid line and follows the equation $y = 0.0066x + 0.42$. The dashed line represents the expected lower-bound for Mica2 communications ($5.1ms/event$ slope, according to equation 1) which is due to the mote's hardware limitations.

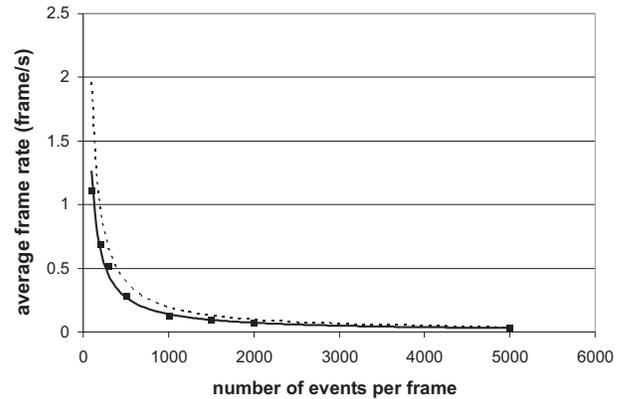


Fig. 9. Frame rate as a function of events/frame. This is the same set of data as in Figure 8, but in the more common frames/second representation. Similarly, the solid trendline follows the equation $y = (0.0066x + 0.42)^{-1}$, while the dashed upper-bound line is $y = (5.1ms \times x)^{-1}$.

Figure 10 depicts a frame captured in a pitch dark environment with an active light source operated by the sensor network. When the image sensor mote is ready to capture a frame, it sends a control sequence to an illuminating mote, which, then, lights up the scene. The first mote starts the data collection and, when done, communicates again with the second mote, in order to turn the active illumination off. This allows to have illumination on demand by using the network operation. Multiple illuminators are possible as well as

multiple cameras. 2000 events were collected to display Figure 10. The image quality is lower than the quality of the image at 2000 events in Figure 7 because of the low illumination intensity provided by a single low-voltage incandescent light bulb. The addition of more active light sources can improve the image quality.

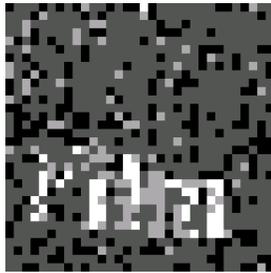


Fig. 10. Frame captured in a pitch dark environment with a sensor-network operated light bulb. When the image sensor mote is to capture a frame, it sends a control sequence to a second mote. The second mote will turn on a light bulb to illuminate the scene, and turn it off as soon as the data collection is terminated.

IV. POWER DISSIPATION

During the transmission phase the Mica2 dissipates about $111mW$ of power, compared to $60.4mW$ during the gathering phase. The Mica2's microprocessor alone draws $24mW$ from the power supply and an extra $81mW$ are used for transmitting at maximum power [7] (we used the TinyOS default value of 50% power), thus the ALOHA's consumption is around $36mW$ in either phase. These data indicate that Mica2 consumes between $0.68\times$ and $1.95\times$ as much power as the image sensor. Running on two $1800mAh$ AA batteries (1.5V each), and assuming equal gathering and transmission times, each image sensor node can run for about $126h$, or 5.25 days. This number could rise up to 6.7 days if a $2300mAh$ battery is used.

V. DISCUSSION AND CONCLUSIONS

We have reported on an architecture for distributed imaging in sensor network nodes with active illumination. With the current setup (first generation system) each is node capable of transmitting a 32×32 -pixel array in an address-event fashion, robust against packet-shuffling and loss. In our tests we achieved a rate of $7.3ms$ per event (equivalent a frame rate of $0.12fps$) when using 1000 events to generate a frame. This is close to the expected rate of $0.19fps$, which is a measured limitation of the Mica2 mote and may be easily improved by supporting a different mote: given that the whole system runs under the TinyOS platform, hardware upgrades are virtually plug-and-play.

Increasing the default TinyOS packet length may also contribute to increased frame rates, but this is beyond our original purpose of implementing this image sensor node in an unmodified TinyOS environment.

Power consumption varies between $60.4mW$ and $111mW$, depending on whether the node is gathering or sending data, respectively. This allows the image sensor node to run on 2 AA batteries for as long as 5 days, although, with the implementation of a sleep/wake-up system (either node level, or radio controlled), the average power consumption could fall dramatically, to μ Watt scale [7], yielding months of battery life. By using the new generation of Zigbee motes *micaZ*, an order of magnitude higher frame rates will be achieved.

Coordination of the active illumination at each node with the global reset signal, enables a boxcar mode of integration for the sensor array

permitting operation in environments with high background photon noise (such as for example in fog or dust).

A second generation event-based ALOHA imager has been re-designed and fabricated to incorporate four independent imagers on a die (Figure 11) and to include a global reset line for all pixels. The whole system operates at much lower power dissipation levels as compared to the original design by optimizing the digital interface circuits. The quad sensor array is not designed to be a high fidelity imager but rather an "eye" for sensor networks and hence the dead space between the arrays is not important. The second generation can be readily configured and operated as a single photodetector capable of over 120 dB dynamic range imaging, as a quad-detector, capable of rapid determination of important visual information at similar levels of dynamic range or can provide images with a resolution of up to 64×64 pixels but with reduced dynamic range.

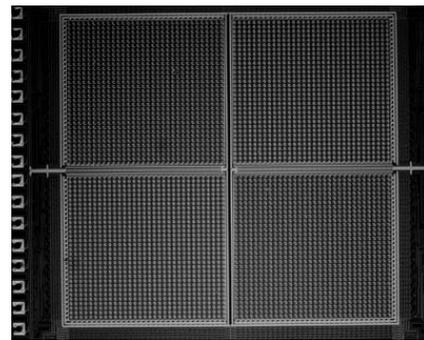


Fig. 11. Die micrograph of the 64×64 -pixels four quadrant event-based ALOHA CMOS imager.

VI. ACKNOWLEDGEMENTS

Fabrication was provided by MOSIS. This work was supported in part by the National Science Foundation grants *Integrated Sensing: Cell clinics on a chip*, ECS-0225489 and *A Comparative Study of Information Processing in Biological and Bio-inspired Systems: Performance Criteria, Resources Tradeoffs and Fundamental Limits*, EIA-0130812.

REFERENCES

- [1] B. Warneke, M. Last, B. Liebowitz, and K. Pister, "Smart dust: communicating with a cubic-millimeter computer," *Computer*, vol. 34, pp. 44–51, January 2001.
- [2] V. Authors, *Proceedings of IEEE Special Issue on Sensor Networks*. IEEE, August 2003.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 42, no. 5, pp. 102–114, August 2002.
- [4] C. Fermuller, Y. Aloimonos, P. Baker, R. Pless, J. Neumann, and B. Stuart, "Multi-camera networks: eyes from eyes," *Proceedings of IEEE Workshop on Omnidirectional Vision, 2000*, vol. 12, pp. 11–18, June 2000.
- [5] P. Doubek, I. Geys, T. Svoboda, and L. V. Gool, *Cinematographic Rules Applied to a Camera Network*. Omnivis, 2004.
- [6] E. Culurciello and A. G. Andreou, "ALOHA CMOS imager," in *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems ISCAS '04*, May 2004.
- [7] Crossbow, *MPR/MIB Mote Hardware Users Manual*, Crossbow Technology Inc., San Jose, CA 95134-2109, 2004, <http://www.xbow.com/>.