

Video Sensor Node for Low-Power Ad-hoc Wireless Networks

Yu M. Chi, Ralph Etienne-Cummings
Department of Electrical
and Computer Engineering
The Johns Hopkins University
Baltimore, MD 21218
Email: ychi2, retienne @jhu.edu

Gert Cauwenberghs
Division of Biological Sciences
The University of California, San Diego
La Jolla, CA 92093
Email: gert@ucsd.edu

Paul Carpenter, Kent Colling
Innovative Wireless Technologies
Forest, VA 24551
Email: pcarpenter, kcolling @iwt.com

Abstract—A video sensor platform consisting of a smart image sensor with focal plane motion processing is presented. The camera node is intended for ultra-low bandwidth ad-hoc wireless networks with data rates of less than 2kB/sec. To meet the targets of low power consumption and low complexity, the CMOS image sensor (consuming 4.2mW of power at 30fps) autonomously monitors for scene changes, outputting full image data only when relevant. For this sensor node, several different operating configurations leveraging the analog processing and storage capability of the imager, including a full motion DCT based video compression algorithm are shown. The entire sensor module can operate from 3 AA batteries and consumes 225mW at full operation.

I. INTRODUCTION

Video surveillance technology for low-power, low-bandwidth networks has remained a challenge despite numerous advancements in electronics. Video rate data contains an extremely large amount of data, making transmission across constrained networks a difficulty. Although advanced video compression algorithms, like H.264, have been successful at reducing this large amount of information, they are typically too complex to implement in real-time on portable electronics. Successful methods to low-power video compression require an alternative to the motion estimation algorithms used in a standard video encoder [1].

For such a scenario, a different approach must be taken. First, the device must be able to autonomously process the data, before transmission, in order to prioritize and transmit the most critical data over the low-bandwidth network. Secondly, the computation must be accomplished at a minimal cost in terms of power consumption. Focal plane based processing has begun to address the issues by implementing the data processing elements at the pixel level in highly parallelized and efficient analog circuits.

Some work in this area have involved using address event representation (AER) [2] image sensors that convert a pixel's light intensity into self-regulating time domain spikes. An AER interface has the advantage of enjoying a wide dynamic range, owing to the variable time pixel integration, an important feature in autonomous visual system as well as offering a host of onboard signal processing functions. However, for

an ultra-low bandwidth network this presents a few problems. First, if spike events are sent as-is, then the data congestion in the network is dependent on the light intensity. Secondly, it does not provide any form of spatial or temporal compression (a large static bright patch would have many redundant pixels outputting at full rate).

State-of-the-art video sensors have also demonstrated the use of change detection to transmit video [5] across networks. However, power consumption (5W) is a typical problem in such designs because of commercial image sensors and hardware platforms are used. While these cameras are capable of providing high image quality, they are not suited for battery powered operations.

Our approach takes a fundamentally different philosophy. We combine traditional video compression techniques with the analog processing capability of the CMOS imager. The change detection imager used [3] is capable of storing and detecting image brightness changes at the pixel-level. Not only does this provide a "wake-up" trigger for data transmission, it can be combined with conditional replenishment compression algorithm [6] [7] to provide for full motion video. In this paper we demonstrate compression algorithms that require only general purpose low power microcontroller to perform full video encoding by leveraging the imager's focal plane processing.

II. SYSTEM DESCRIPTION

A. Image Sensor

Each active pixel sensor in the 90×90 imaging array includes the standard photodiode, reset, buffer and access transistors along with an additional three transistor, two capacitor comparator circuit. The operation of this imager, which can perform both in-pixel change detection and ADC has been presented at previous conferences [3] [8]. As output, the imager provides a standard analog image intensity output as well as a digital flag indicating the presence and direction of intensity changes. The imager consumes 4.2mW at 30fps.

A typical output for the change detection is shown in Fig. 1. The thresholds for change detection is user adjustable with a rejection band in order to eliminate spurious readings. By itself, three-level change detection output contains a significant

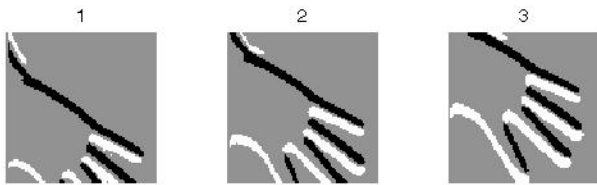


Fig. 1. Imager change detection output showing a moving hand across the frame.

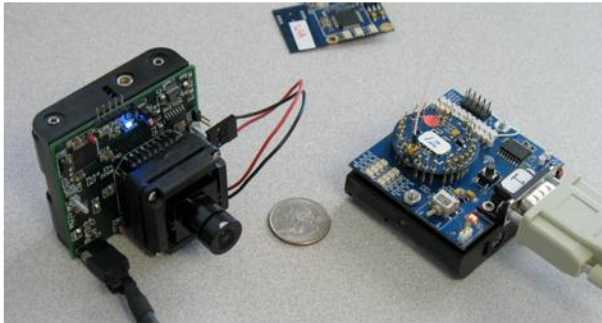


Fig. 2. Image of the video sensor node and wireless module. The two modules are connected via a standard RS232 cable and are powered by 4AA batteries each.

amount of useful data. First, it indicates the presence of motion, and can be used as a wake-up trigger. Secondly, the output can be used as an edge detector since temporal changes occur only along areas with non-zero spatial gradients.

For image processing, a low power microcontroller with DSP instructions (dsPIC33) is used. The controller runs at a clock rate of 18MHz and includes 16 kilobytes of on board memory. At this speed, an 8×8 DCT [4] requires a little more than $150\mu s$ to complete (less than 2ms for the entire scene). It is also used to generate the timing and clocking signals for the microcontroller, perform DCT on the image, and handles communications with the wireless node. A picture of the image sensor board is shown in Fig. 3. The firmware was written in C and operates completely within the processor's internal RAM.

III. NETWORK PROTOCOL

Networking is provided by a separate, proprietary commercial module. In this setup, the sensor and wireless module communicate via a standard RS232 link operating at 38.4kbps. Each wireless node functions as part of a peer-to-peer ad-hoc wireless network and include transceivers that can function at both 900MHz and 2.4GHz. The custom protocol for the wireless network includes provision for triangulating a single node's position. Because of the low-bandwidth nature of the network, no effort had been made to integrate a camera, with previous efforts focusing on lower data rate devices like acoustic trip sensors.

To monitor the network, a PC is used to display the location and status of each wireless node. The program controls the camera's settings including exposure, frame rate and transmit rate and displays the received images (Fig. 4).

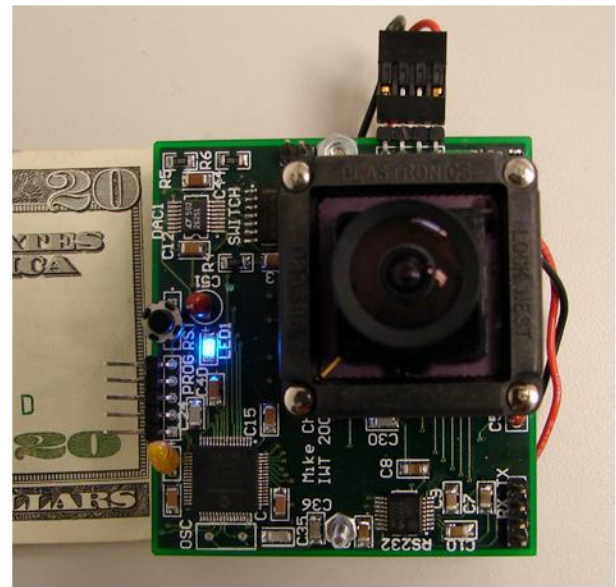


Fig. 3. Camera circuit board showing image sensor, lens and microcontroller.

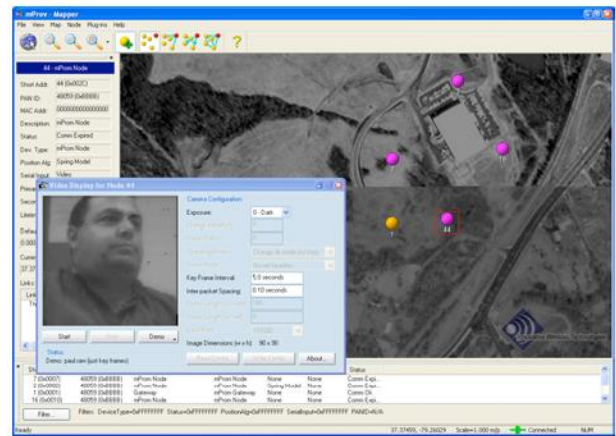


Fig. 4. GUI for camera display. Program shows map of each node and a separate window for the video. The circles superimposed on a map represent the physical location of a sensor. Clicking on each node brings up the camera control window which includes the video output.

In the current version of the complete sensor plus network, the packet size of the network is predefined at 144 bytes with a typical delay of 100ms or more between each packet. From the standpoint of the video sensor's processor, the network appears as an RS232 channel (appropriately throttled by software as to not overload the connection). This approach enables rapid integration of various sensors and devices by using a common well established standard.

Finally, this architecture also allows the provision of higher data rate networks in the future allowing for the possibility of higher quality video rate data.

A. Compression Algorithm

With an effective data rate of less than 2 kilobytes per second, a compression algorithm is required to minimize the data rate, even for relatively low resolution images. As current

state-of-the-art video compression algorithms demonstrate, a temporally compensated spatial transform coding of the image sequence can provide high compression ratios while maintaining acceptable image quality.

Two modes are presented in this paper, one suited for the current low bandwidth network and a more advanced one that provides for full motion video, but with no significant increase in complexity or physical requirements and can be implemented on the exact same hardware. The low bandwidth mode utilizes the change detection circuitry as a "wake-up" trigger to begin image transmission.

1) *Image Encoder*: Image encoding is facilitated by a JPEG-like algorithm, simplified to operate on the microcontroller. Since the data rate available is not sufficient for any meaningful full motion video, the image sensor only transmits when scene changes are detected. The microcontroller monitors the output of the image sensor, and when the changed pixel count crosses a threshold, a full image scan out is initiated.

A JPEG-like algorithm, simplified to run on the microcontroller is used to compress the image. Since the microcontroller has limited capabilities, a fast DCT approximation [4] is applied to the 8×8 blocks followed by the JPEG perceptual quantization matrix. Entropy encoding is facilitated by a combined RLE and fixed table Huffman code.

For most images a 3.5:1 compression ratio, at a minimal loss in visual quality, can be achieved corresponding to one image every 15 packets or 1.5 seconds.

2) *Proposed Video Encoder*: While the aforementioned algorithm is suitable for the nature of the low bandwidth network, it does not provide for full motion video (useful if there are constant scene changes), and does not fully leverage the analog processing capability of the imager. In addition to the standard image output, we also propose a full video encoder that takes advantage of the change detection output to temporally de-correlate the video data.

The image is again broken into 121 8×8 DCT transformed blocks. Associated with each block is a counter indicating the number of pixels that have changed and the time since the last refresh. The coefficients are scanned in a zigzag order, like other DCT based algorithms. Transmissions begins by sending the first few low frequency coefficients. In the next frame, blocks that do not undergo change have the next set of coefficients transmitted. For static areas, this results in a progressive transmission that stops when a set number of coefficients have been sent. On the other hand, blocks that experience change are completely flushed and refreshed.

Unlike more complex schemes, this is an entirely open loop system, and an age counter is used to prevent long term steady state error accumulation. This also has the added advantage of avoiding costly single keyframes that contain redundant information - blocks are updated on a need-to basis depending on temporal change and time since last update.

High quality image transmission is accomplished by allocating all the bits only to dynamic blocks after all the static regions are complete. In a typical surveillance scenario, the

image is usually composed of a largely static background with a few moving portions. As a result, the video stream responds quickly to changes since the relatively few blocks that do undergo change have a large majority of the bandwidth allocated to transmitting it's DCT coefficients.

Rate control is accomplished by simply dividing the total number of bits per frame by the number of blocks that need to be updated. In this manner, the overall bitrate stays constant, an important feature for a low bandwidth network. For transmission groups of blocks are packetized along with the relevant flags. This eliminates any dependency on frame-level synchronization. Furthermore, this guarantees a constant bitrate, useful in a low bandwidth network.

In order to better quantify and characterize the performance of the algorithm, a PC based simulation was used so that an established video test sequence (*hall.qcif*) can be compressed. Images from the sequence are shown in Fig. 5. Each frame was allocated 384 bytes (a 20:1 compression ratio) at a change sensitivity of 15%. DCT coefficients are uniformly quantized to 9 bits. Macro-blocks were updated if more than five pixels report an intensity change, stopping when 32 coefficients have been transmitted.

The distortion performance of the algorithm is plotted in Fig. 6. Static areas quickly converge to a high quality image as more DCT coefficients are updated. For scenes with significant motion, the PSNR gradually falls to around 26dB PSNR, but nevertheless remains highly usable.

In effect, the algorithm is performing a temporally compensated video coding, minus the motion prediction aspect. Using the change detection output as an indicator of areas that have undergone motion maximizes the quality of the image by properly allocating the available bits into the appropriate image regions. At the same time, it remains no more complex than a pure intra-frame encoder as the imager incorporates all the memory and processing for the motion detection at the focal plane. The only significant off-imager computation is the DCT. In contrast, if the change detection imager was replaced with a standard video sensor, then the best that can be done at the equivalent system complexity is a pure DCT based intrafame encoding, which results in a significant image degradation at the same bitrate (Fig. 6).

IV. CONCLUSION

We demonstrate a low power video sensor platform for bandwidth constrained networks that do not traditionally support image data. By using a computational image sensor with focal plane motion processing, the overall complexity of the system is reduced, making it suitable for ad-hoc deployment and battery powered operations. In addition, we show that despite the limited processing facilities of the board's microcontroller, a DCT based temporally compensated video encoder can be implemented. This architecture can be easily scaled to higher resolutions since the majority of the memory and motion processing is handled within the imager's pixel array, leaving the processor responsible only for maintaining status data at the block level and performing the DCT coding. This

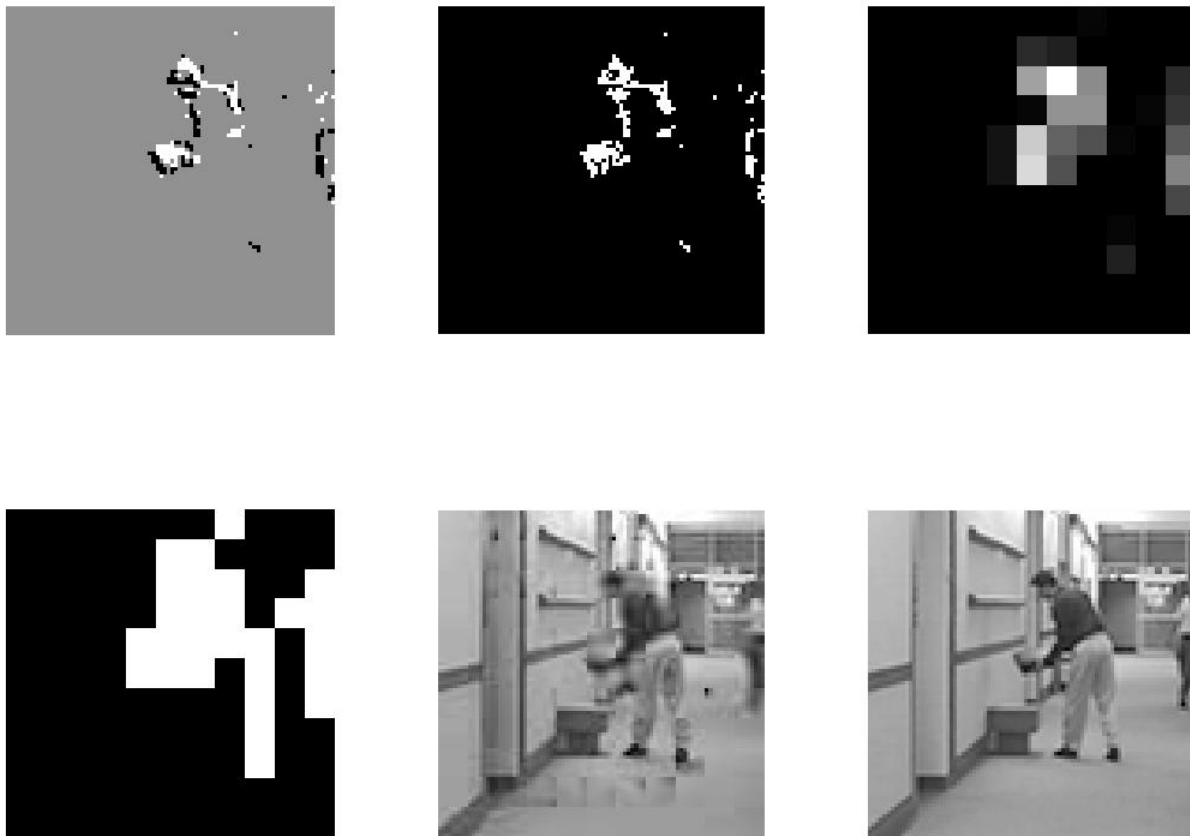


Fig. 5. Test video sequence. Top row, left to right: Bright/Darker change output, Pixels that report change, Number of changes in a block. Bottom row, left to right: Macroblocks that are updated for this frame, Reconstructed image, Original image.

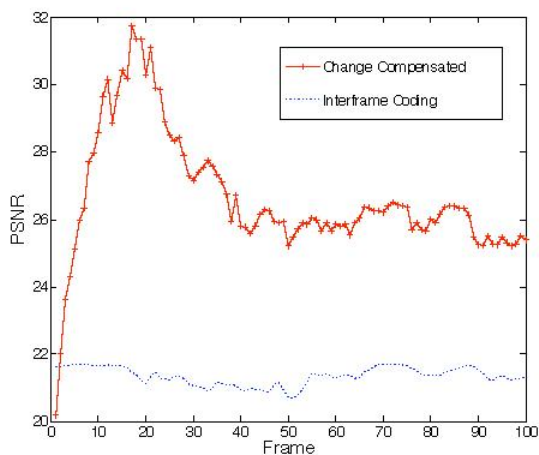


Fig. 6. PSNR for test sequence for change compensated and pure intraframe coding.

design provides the possibility of surveillance type images in applications heavily limited by power and bandwidth.

REFERENCES

[1] R. Puri, A. Majumbar, P. Ishwar, K. Ramchandran, "Distributed video coding in wireless sensor networks," *IEEE Signal Processing Magazine*, Jul. 2006.

[2] E. Culurciello, A. Savvides, "Address-Event Image Sensor Network," *IEEE International Symposium on Circuits and Systems*, May. 2006.
 [3] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs and R. Etienne-Cummings, "Temporal change threshold detection imager," *IEEE Solid State Circuits Conference* Feb. 2006.
 [4] J. Liang and T. D. Tran, "Fast Multiplierless approximation of the DCT with the lifting scheme," *IEEE Transactions on Signal Processing*, Vol. 49, No. 12, pp.3032-3044, Dec. 2001.
 [5] W.C. Feng, B. Code, E. Kaiser, M. Shea, L. Bavoil, "Panoptes: Scalable low-power video sensor networking technologies," *Proc. 11th ACM Int. Conf. Multimedia*, Nov. 2003.
 [6] Y. Chiu, T. Berger, "A Software-Only Videocodex Using Pixelwise Conditional Differential Replenishment and Perceptual Enhancements," *IEEE Trans. Circuits and Systems for Video Technology*, April 1999.
 [7] E. Amir, S. McCanne, M. Vetterli, "A layered DCT coder for Internet Video," *Proc. International Conference on Image Processing* Sept. 2006
 [8] Y. M. Chi, U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs and R. Etienne-Cummings, "CMOS Pixel-Level ADC with Change Detection," *IEEE International Symposium on Circuits and Systems*, May. 2006.