Pixel Level Processing — Why, What, and How?

Abbas El Gamal, David Yang, and Boyd Fowler

Information Systems Laboratory, Stanford University
Stanford, CA 94305 USA

ABSTRACT

Pixel level processing promises many significant advantages including high SNR, low power, and the ability to adapt image capture and processing to different environments by processing signals during integration. However, the severe limitation on pixel size has precluded its mainstream use. In this paper we argue that CMOS technology scaling will make pixel level processing increasingly popular. Since pixel size is limited primarily by optical and light collection considerations, as CMOS technology scales, an increasing number of transistors can be integrated at the pixel. We first demonstrate that our argument is supported by the evolution of CMOS image sensors from PPS to APS. We then briefly survey existing work on analog pixel level processing and pixel level ADC. We classify analog processing into intrapixel and interpixel. Intrapixel processing is mainly used to improve sensor performance, while interpixel processing is used to perform early vision processing. We briefly describe the operation and architecture of our recently developed pixel level MCBS ADC. Finally we discuss future directions in pixel level processing. We argue that interpixel analog processing is not likely to become mainstream even for computational sensors due to the poor scaling of analog compared to digital circuits. We argue that pixel level A/D conversion will become increasingly popular since it minimizes analog processing, and requires only simple and imprecise circuits to implement. We then discuss the inclusion of digital memory and interpixel digital processing in future technologies to implement programmable digital pixel sensors.

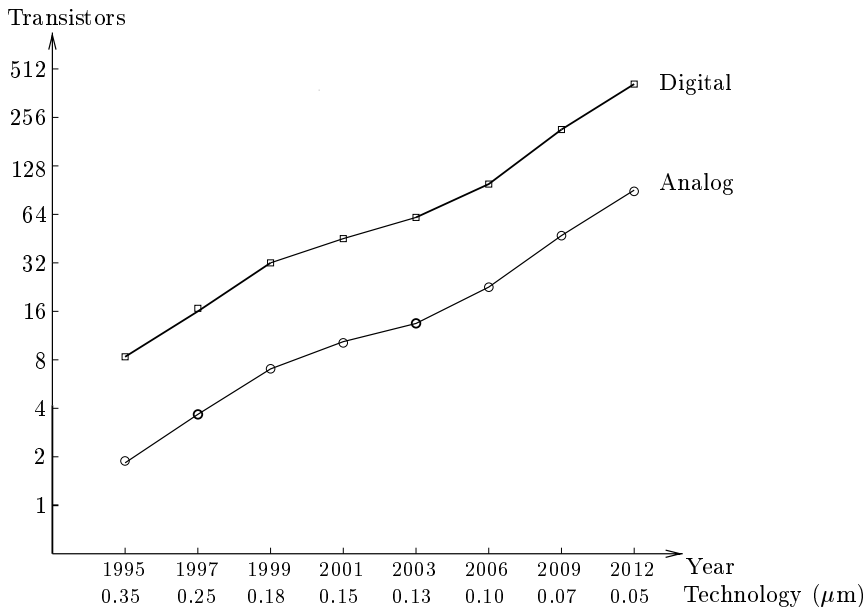**Keywords:** pixel level processing, pixel level ADC

1. INTRODUCTION

The main advantage of CMOS image sensors is the ability to integrate sensing and processing on the same chip. This advantage is especially important for implementing imaging systems requiring significant processing such as digital cameras and computational sensors. Processing can be integrated with a sensor at the chip level using a "system-on-chip" approach, at the column level by integrating an array of processing elements each dedicated to one or more columns, and at the pixel level by integrating a processing element at each pixel or group of neighboring pixels. At present chip and column level processing are the most widely used. With the exception of signal conditioning, pixel level processing is generally dismissed as resulting in pixel sizes that are too large to be of practical use. Most of the reported work on CMOS single chip digital cameras involve the integration of a sensor with chip or column level processing.[1,2] The work on computational sensors involves the integration of analog processing at the pixel level. However, it is not widely accepted.

Pixel level processing promises very significant advantages. Analysis by several authors[3,4] shows that pixel level A/D conversion achieves higher SNR than chip or column level A/D conversion approaches. Moreover, substantial reduction in system power can be achieved by performing processing at the pixel level. By distributing and parallelizing the processing, speed is reduced to the point where analog circuits operating in subthreshold can be used. These circuits can perform complex computations while consuming very little power.[5] The most important advantage of pixel level processing, however, is that signals can be processed during integration. We recently demonstrated an example of this advantage — the ability to programmably enhance dynamic range via multiple sampling using our recently developed pixel level ADC.[6]

Other author information: Email: abbas@isl.stanford.edu, dyang@isl.stanford.edu, fowler@isl.stanford.edu, ; Telephone: 650-725-9696; Fax: 650-723-8473

Transistors

512

256

128

64

32

16

8

4

2

1

Digital

Analog

Year	1995	1997	1999	2001	2003	2006	2009	2012
Technology (μm)	0.35	0.25	0.18	0.15	0.13	0.10	0.07	0.05

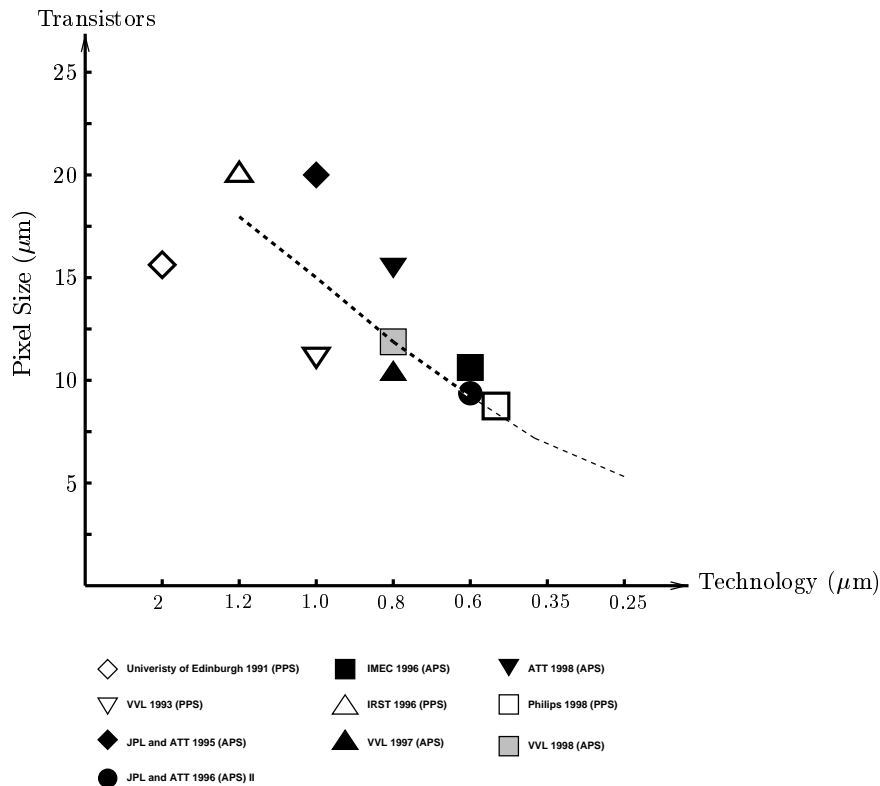**Figure 1.** Transistors per pixel as a function of time and process technology.

In this paper we argue that these advantages coupled with CMOS technology scaling will make pixel level processing increasingly popular. Since pixel size is limited primarily by optical and light collection considerations, as technology scales an increasing number of transistors can be integrated at each pixel without adversely affecting its size or fill factor. It is generally believed that a pixel size below 4μm (on a side) is not desirable, since it would require unacceptably expensive optics. The performance of such small pixels also suffers from the decrease in dynamic range and SNR due to the decrease in well capacity, and the increase in nonuniformity due to the small feature sizes and increase in dark signal relative to the photo signal.[*]. Figure 1 plots the estimated number of transistors per pixel for both digital and analog circuits[†] as technology scales assuming a 5μm pixel with constant fill factor of 30%. As can be seen from the figure the number of (digital) transistors grows according to Moore's law from 8 at 0.35μm, to 32 at 0.18μm, and to 410 at 0.05μm! Wong[7] points out that CMOS technology will eventually migrate to SOI and as a result it will become infeasible to build photodetectors in the standard process. Photodetectors can be built on top of a standard CMOS chip, however, using, for example, amorphous silicon.[8,9] In this case all of the area under the pixel becomes available to use for processing.

Our assertion that more pixel level processing will be performed as technology scales is supported by past developments of CMOS image sensors. Scaling has been the driving force in the evolution of CMOS image sensors from PPS to APS. As technology scaled more transistors were added to the pixel to increase the sensor speed and improve its SNR, while achieving competitive pixel sizes. We expect this trend to continue. As demostrated by our recent pixel level A/D conversion work,[10] an 8-bit Nyquist rate pixel level ADC can be implemented in a 10μm pixel with fill factor of 30% using a standard digital 0.35μm CMOS technology.

The rest of the paper is organized as follows. In section 2 we provide a historical perspective, which supports our assertion that technology scaling has been the driving force behind the evolution from PPS to APS. In section 3 we briefly survey the work on analog pixel level processing. We classify this work into two general categories — intrapixel, where the processing is performed on the individual pixel signals, and interpixel, where the processing is performed locally or globally on signals from several pixels. The

---

[*]Dark current for a small pixel increases relative to the signal since the leakage from the edges of a photodetector is higher than from its area.

[†]These estimates are based on the SIA roadmap and our pixel layouts. The number of digital transistors is about 5 times larger than the analog, which is consistent with our 0.35μm technology designs. We assumed that this ratio does not change with scaling. We believe that this is optimistic, and that the ratio should in fact increase with scaling. However, we do not have enough data to quantify this belief.

Transistors

25

20

Pixel Size (μm)

15

10

5

Technology (μm)

2   1.2 1.0 0.8 0.6 0.35 0.25

◇ Univeristy of Edinburgh 1991 (PPS)   ■ IMEC 1996 (APS)   ▼ ATT 1998 (APS)

▽ VVL 1993 (PPS)   △ IRST 1996 (PPS)   □ Philips 1998 (PPS)

◆ JPL and ATT 1995 (APS)   ▲ VVL 1997 (APS)   ▨ VVL 1998 (APS)

● JPL and ATT 1996 (APS) II

Figure 2. PPS and APS pixel sizes as a function of CMOS process technology. The dotted line represents the 15F estimates of APS pixel size of Fossum.[22]

purpose of intrapixel processing is to improve image quality and lower the sensor's power consumption. The purpose of interpixel processing on the other hand is to perform early vision processing, not merely to capture images. In section 4 we discuss the work on pixel level A/D conversion. We briefly describe the operation and architecture of our recently published Nyquist rate pixel level ADC.[10] Finally in section 5 we look into the future of pixel level processing. We envision the convergence of these different types of processing into programmable digital pixel sensors. These sensors can be programmed to adapt to different imaging environments or programmed to peform different vision processing functions.

2. HISTORICAL PERSPECTIVE

The history of MOS image sensors is detailed in two excellent survey papers by Fossum.[11,12] Although MOS image sensors first appeared in the late 1960s,[13] most of todays CMOS image sensors are based on work done starting around the early 1980's. Until the early 1990s PPS was the CMOS image sensor technology of choice.[14-18] The feature sizes of the available CMOS technologies were too large to accomodate more than a single transistor and three interconnect lines in a pixel. The speed and SNR of PPS were significantly lower than CCD sensors. This limited their applicability to low performance applications such as certain machine vision applications. In the early 1990s work began on modern APS.[11,19,20] It was quickly realized that adding an amplifier to each pixel significantly increases sensor speed and improves its SNR, thus alleviating the shortcomings of PPS. CMOS technology feature sizes, however, were still too large to make APS commercially viable. With the advent of deep submicron CMOS technologies and microlenses, APS has not only become the CMOS image sensor technology of choice,[1,2,21] but has also made it a serious competitor to CCDs. Figure 2 plots several reported PPS and APS pixel sizes indicating the minimum CMOS technology feature size used. Note the continual decrease in APS pixel size down to the 4μm minimum around 0.25μm.

Although the main purpose of the extra transistors in the APS pixel is to improve the sensor speed and SNR, they can be also used to perform other useful functions such as electronic shuttering,[23] antiblooming, correlated double sampling (CDS),[24] and frame differencing.[25] By appropriately setting the gate voltage of the reset transistor in an APS pixel blooming can be avoided. In a photogate APS, the signal is transfered to a sense node that is decoupled from the photodetector.[19,26] This not only provides useful signal amplification and enables the implementation of CDS, but can also be used to perform motion detection and frame differencing.[25] The reset transistor can also be used to enhance dynamic range using the well capacity adjusting scheme.[27] Higher dynamic range can also be achieved via individual reset,[28] i.e., where each pixel can have its own exposure time. Note that implementing these additional functions requires almost no modifications to the pixel, and only minor modifications to the column level circuitry.

3. ANALOG PIXEL PROCESSING

In this section we survey the work on analog pixel processing beyond APS. We classify the work into two categories — intrapixel and interpixel processing — and briefly survey some of the work in each category. We focus our survey on image sensors in the visible range, even though there is a wealth of literature on analog pixel level processing for IR sensors. We do not claim comprehensiveness, or that the work we mention is the only important work in the area. The purpose of the survey is to provide a flavor for the types of analog pixel processing that has been proposed and implemented.

Several authors have reported on analog pixels that peform intrapixel processing beyond APS. Kymasu[29] describes a CMOS imager that empolys a transfer gate between the photodiode and a source follower gate. The transfer gate functions as a common gate amplifier, which helps improve sensitivity. Fixed pattern noise is also reduced in this design using a clever feedback technique. Aizawa et al.[30] describes a pixel circuit which can be used to perform video compression using conditional replenishment. A pixel is updated or replenished only if its current value differs substantially from its previously stored value. Hence only the moving areas of an image are detected and coded. Mead[31] and Dierickx *et al.*[32] describe pixels using instantaneous readout mode with logarithmic response to achieve very wide dynamic range.

Most of the work on interpixel processing is focused on computational sensors (neuromorphic vision sensors), and silicon artificial retinas. Many authors have reported on sensors that perform optical motion flow,[33–37] which typically involve both local and global pixel calculations. Both temporal and spatial derivatives are locally computed. The derivatives are then used globally to calculate the coefficients of a line using least squares approximation. The coefficients of the line represent the final optical motion vector. The work on artificial silicon retinas[38–40] has focused on illumination independent imaging, and temporal low pass filtering, both of which involve only local pixel computations. Astrom[41] describes an image sensor for segmentation and global feature extraction. Brajovic et al.[42] describe a computational sensor using both local and global interpixel processing. The sensor can perform histogram equalization, scene change detection, image segmentation, in addition to normal image captue. Before an image is readout, the sensor computes the image indices as well as its histogram. The image of indices never saturates and has a uniform histogram. Rodriguez-Vazquez et al.[43] report on programmable computational sensors based on cellular nonlinear networks (CNN), which are well suited for the implementation of image processing algorithms. A salient feature of their work is making the CNNs programmable via local interactions, as most ealier CNNs were function specific and not programmable. Another approach, which is potentially more progammable, is the Programmable Artificial Retina (PAR) described by Paillet *et al.*.[44] A PAR vision chip is a SIMD array processor in which each pixel contains a photodetector, (possible) analog preprocessing circuitry, a thresholder, and a digital processing element. The thresholder is the same as the one described by Astrom *et al.*.[41] Its purpose is to provide gray scale vision, while processing only binary images. Although very inefficient for image capture, the PAR can perform a plethora of retinotopic operations including early vision functions, image segmentation, and pattern recognition.

4. PIXEL LEVEL A/D CONVERSION

Although most of the work on pixel level processing has focused on analog processing, there has been a recent trend towards using the increasing number of available transistors at the pixel to perform A/D conversion,

instead. This trend is motivated by the many very significant advantages of pixel level A/D conversion. Analysis by several authors[4,3] shows that pixel level A/D conversion should achieve higher SNR and the lower power consumption than column or chip level approaches, since it is performed in parallel, close to where the signals are generated, and is operated at very low speeds. Another advantage of pixel level A/D conversion is scalability. The same pixel and ADC design and layout can be readily used for a very wide range of sensor sizes. Pixel level A/D conversion is also well suited for standard digital CMOS process implementation. Since the ADCs can be operated at very low speeds, very simple and robust circuits can be used.

Unfortunately, none of the well established A/D conversion techniques meets the stringent area and power constraints of pixel level implementation. Several authors[45-47] use a voltage-to-frequency converter at each pixel so that no analog signals need to be transported. However, since the A/D conversion is performed one row at a time, this method is essentially a column level A/D conversion method. Fowler et al.[48] and Yang et al.[49] describe the first true pixel level A/D conversion technique. Each ADC employs a one bit $\Sigma\Delta$ modulator at each pixel. The ADCs are implemented using very simple and robust circuits, and operate in parallel. The implementation had several shortcomings, however, including: large pixel size, high output data rate, poor low light performance, high fixed pattern noise, and lag.

The large pixel size quickly disappears with technology scaling. Yang et al.[10] describe the first viable Nyquist rate pixel level ADC, which is called multi-channel bit-serial (MCBS) ADC. The ADC overcomes the other shortcomings of the aforementioned $\Sigma\Delta$ ADC technique. Output data rate is reduced by using Nyquist rate conversion instead of oversampling. Low light performance is improved to the level of analog CMOS sensors by using direct integration instead of continuous sampling. Nonuniformity is significantly reduced by globally distributing the signals needed to operate the ADCs and by performing local autozeroing. Lag is eliminated by resetting the photodetectors after A/D conversion is performed. The ADC has several other advantages. It can readily implement variable step size quantization, e.g., for gamma correction or logarithmic compression. The pixel level circuits can be fully tested by applying electrical signals without any optics or light sources. Yang *et al.*[6] describe, arguably, the most important advantage of this ADC technique — the ability to programmably enhance dynamic range via multiple sampling. Since the signals are available to the ADCs during integration, they can be sampled at any time and to any desired resolution. The samples can then be combined to achieve floating point resolution.

In the remainder of this section we briefly describe the operation and architecture of our MCBS ADC. A more detailed description, which also includes circuit design details and description of a 320×256 pixel sensor implemented in a standard 0.35μm CMOS technology, is provided in the paper by Yang et al..[6]

The operation of the MCBS ADC is based on the observation that an ADC maps an analog signal S into a digital representation (codeword) according to a quantization table, and thus each bit can be separately generated. For example consider the generation of the LSB in the 3-bit Gray coded example given in Table 1, where S is assumed to take on values in the unit interval (0,1]. From the table, the LSB is a 1 iff $S \in (\frac{1}{8}, \frac{3}{8}] \cup (\frac{5}{8}, \frac{7}{8}]$. To generate the LSB, any bit-serial Nyquist rate ADC must be able to answer the question: is $S \in (\frac{1}{8}, \frac{3}{8}] \cup (\frac{5}{8}, \frac{7}{8}]$?. Thus, the ADC is essentially a one-detector that indicates the input ranges resulting in a 1. Interestingly, by judiciously selecting the sequence of comparisons to be performed, the one-detector can be implemented using only a one-bit comparator/latch pair.

A block diagram of a one bit comparator/latch pair is shown in Figure 3. The waveforms in the figure illustrate how it performs bit-serial ADC. The signal **RAMP** is an increasing staircase waveform. The output of the comparator feeds into the latch's gate, while the digital signal **BITX** feeds into its data terminal. The MSB is simply generated by comparing S to a **RAMP** value of $\frac{1}{2}$. To generate the LSB, **RAMP** starts at zero and monotonically steps through the boundary points $(\frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8})$. At the same time **BITX** starts at zero and changes *whenever* **RAMP** changes. As soon as **RAMP** exceeds S, the comparator flips, causing the latch to store the **BITX** value just after the **RAMP** changes. The stored value is the desired LSB. After the comparator flips, **RAMP** continues on, but since **RAMP** is monotonic, the comparator flips exactly once so that the latch keeps the desired value. For example, for input1, which is between $\frac{3}{8}$ and $\frac{5}{8}$, the comparator flips when **RAMP** steps to $\frac{5}{8}$ which is just above the input1 value, and **BITX** also changes to zero. When the comparator output goes low, a zero, which is the desired LSB, is latched. After that, **RAMP** continues to increase and **BITX** continues to change. Since the latch is closed, however, **BITX**

ADC Input Range	Codeword
$0 - \frac{1}{8}$	0 0 0
$\frac{1}{8} - \frac{2}{8}$	0 0 1
$\frac{2}{8} - \frac{3}{8}$	0 1 1
$\frac{3}{8} - \frac{4}{8}$	0 1 0
$\frac{4}{8} - \frac{5}{8}$	1 1 0
$\frac{5}{8} - \frac{6}{8}$	1 1 1
$\frac{6}{8} - \frac{7}{8}$	1 0 1
$\frac{7}{8} - 1$	1 0 0

Table 1. Gray code quantization table for the $m=3$ example
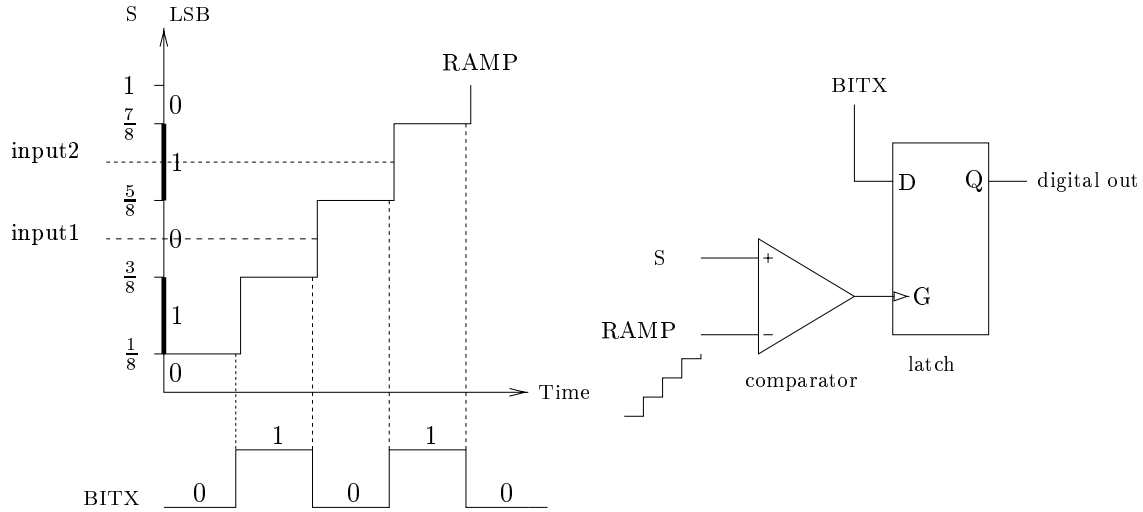
Figure 3. Comparator/latch pair operation

can no longer influence the output. After **RAMP** completes stepping through the boundary points, the latched output is read out. Then **RAMP** and **BITX** are reset to zero in preparation for another sequence of comparisons. In this fashion, all bits from MSB to LSB are generated. The NMSB is similarly generated by comparing input1 to $\frac{2}{8}$ and to $\frac{6}{8}$, which yields a 1.

This 3-bit example can be easily generalized to perform any m-bit ADC. To quantize S to m bits of precision, the unit interval is divided into 2^m input ranges ($\frac{i}{2^m}$, $\frac{(i+1)}{2^m}$], $0 \leq i \leq 2^m - 1$, and each range is represented by an m-bit codeword. To determine the m-bit codeword for S, the ADC generates each bit serially (in any desired order). Each bit is generated by answering the question: is signal $S \in A$?, where A is the set of input ranges that result in 1. The ADC implements the question by successive comparisons at the boundary points of the ranges in A using the comparator/latch pair described. The **RAMP** signal steps through the boundary values monotonically, while **BITX** indicates the value (0 or 1) of the particular range. At the end of each sequence of comparisons, the latched value is read out.

A block diagram of the MCBS ADC is shown in Figure 4. It consists of multiple channels, each having its own comparator/latch pair. The m-bit DAC and the logic needed to generate **RAMP** and **BITX** are shared among all channels. The **RAMP** and **BITX** signals are broadcast to all channels and the resulting bits are read out of each channel. The readout architecture of an image sensor with MCBS ADC is shown in Figure 5. It consists of a 2-dimensional array of pixel blocks, a row decoder, and column sense amplifiers. Each pixel block comprises one or more photodetectors sharing an MCBS ADC channel. The **RAMP** and **BITX** generation circuitry, which lie outside the image sensor array, are not shown in the figure. The
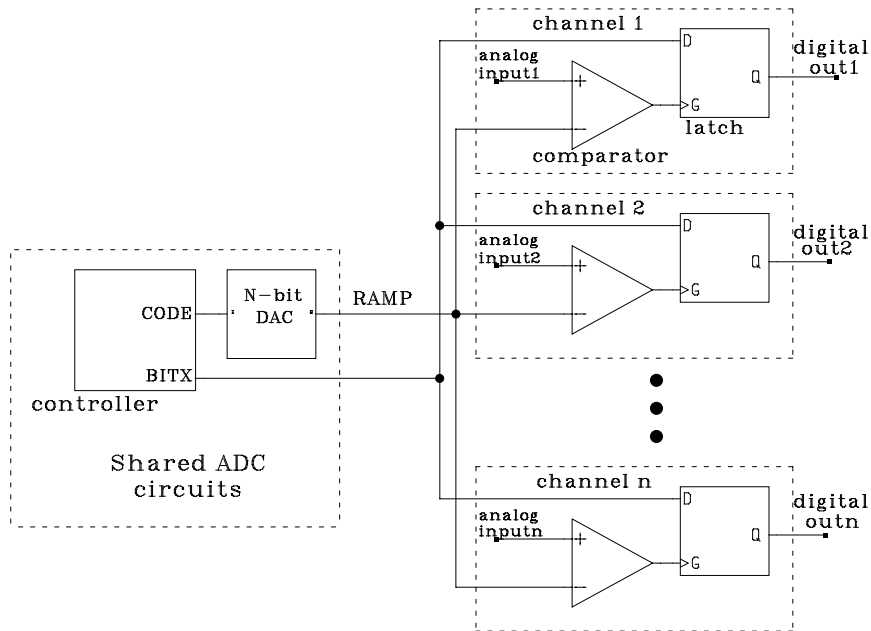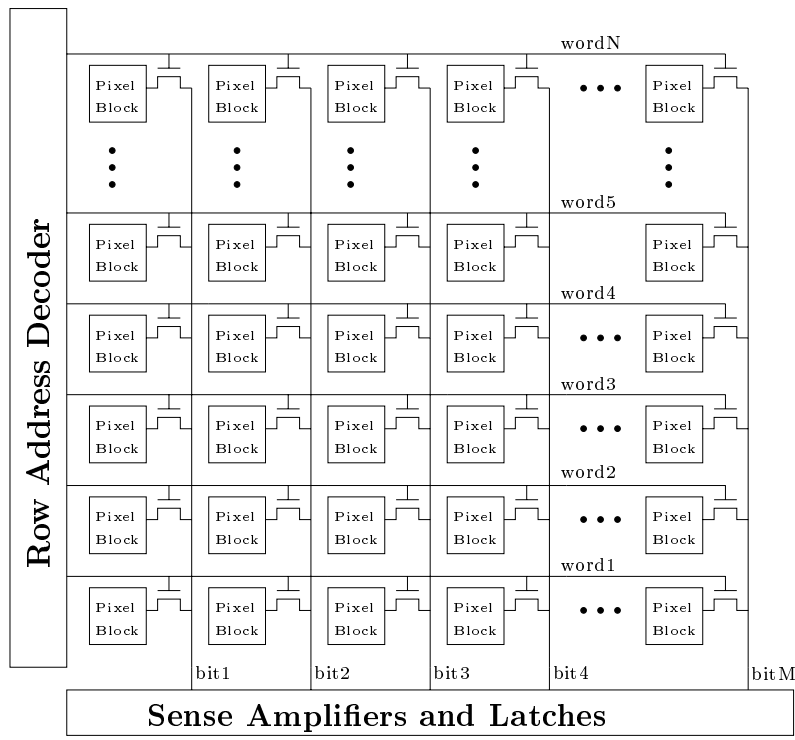
Figure 4. MCBS ADC Block Diagram

captured analog pixel values are digitized in parallel one bit at a time. Each latched set of bits form a bit-plane, which is read out in a manner similar to a standard digital memory, using the row decoder and the column sense amplifiers. Note that this readout format is quite different from the raster scan format commonly used in CCD and APS. However, it has advantages such as programmble pixel resolution and region-of-interest windowing.

5. FUTURE DIRECTIONS

As we argued ealier, pixel size does not scale with CMOS technology below around 4μm. As a result more transistors can be integrated at the pixel level as CMOS technology scales. To date these transistors have been used for intrapixel or interpixel analog processing, or for pixel level A/D conversion. In this section we look into the future of pixel level processing. We argue that interpixel analog processing is not likely to become mainstream even for computational sensors. We argue that the advantages of pixel level A/D conversion are likely to make it very attractive for implementaion at 0.18μm technology and below. We then look beyond pixel level A/D coversion, and discuss the merits and shortcomings of integrating digital storage at the pixel and eventually interpixel programmable digital processing.

The argument for interpixel analog processing is that more processing can be performed in less area and using less power than using digital processing, albeit at low speed and low precision. This argument breaks down as technology scales. On the one hand, analog circuits scale poorly with technology due to several factors including the decrease in power supply voltage and the increase in device leakage currents. On the other hand, digital circuits scale well with technology in area, performance, as well as power. As a result, the argument that analog processing saves power and area over digital processing becomes less valid as technology scales.

Pixel level A/D conversion minimizes the amount of analog processing, and requires very simple and imprecise analog circuits. As a result, it is quite amenable to scaling. In Figure 6 we plot the estimated pixel sizes needed to implement our MCBS ADC, assuming 30% fill factor, for both multiplexed, where each block of 2×2 pixels share a 1-bit comparator/latch pair, and nonmultiplexed implementations. The size of the multiplexed implementation at 0.35μm is the actual size of our implementations.[10] As can be seen the

**Figure 5.** Block diagram of an image sensor with pixel level ADC
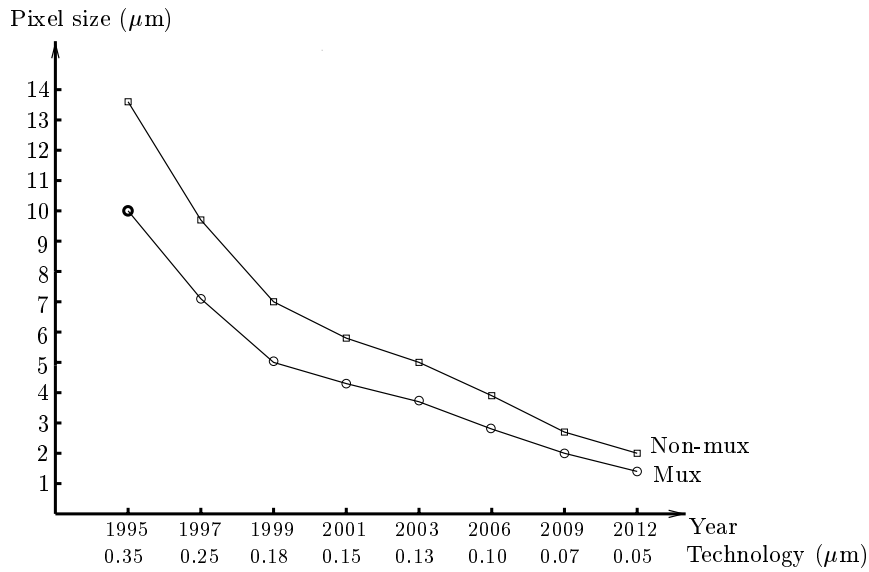
Figure 6. Estimated pixel size of MCBS ADC for multiplexed and unmultiplexed implementations.

multiplexed pixel size approaches the 4μm limit at around 0.15μm, whereas the nonmultiplexed pixel size approaches the limit at 0.1μm.

The next logical step beyong pixel level A/D conversion is to include digital memory at the pixel. This memory can serve a number of important functions, including local state information, pipelining for fast operation, frame storage, and system power reduction. In Figure 7 we plot the projected number of embedded DRAM and SRAM bits that can be accomodated in a 5μm pixel as technology scales assuming
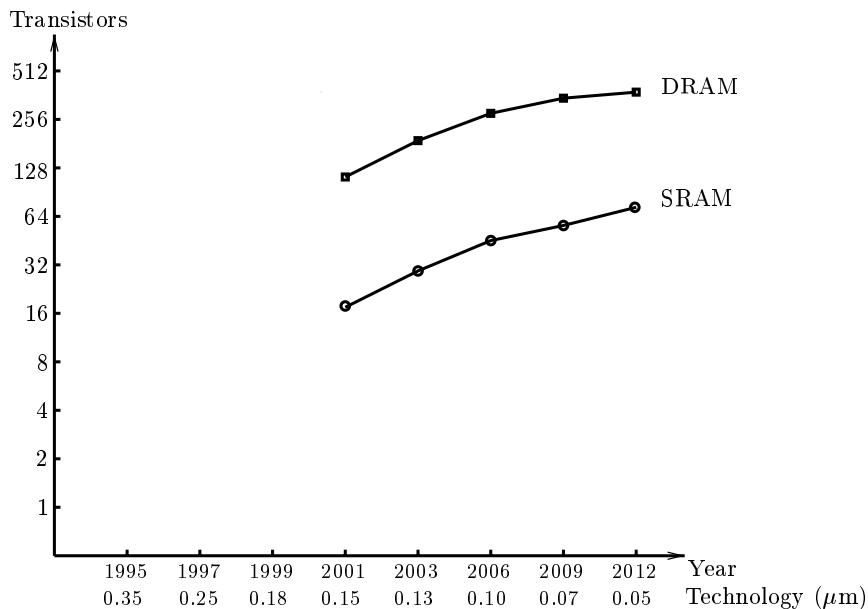
Figure 7. Projected number of DRAM and SRAM bits in a pixel.

30% fill factor and nonmultiplexed MCBS ADC implementation. Note that above 0.15μm, no extra pixel area is vailable for memory. At 0.15μm and below, the number of bits that can be accomodated in a pixel grows quickly to 64 for SRAM and over 350 for embedded DRAM. Even more bits can be included if a DRAM process is used.

Instead of using the available pixel area only for memory, part of the area may be used for interpixel digital processing. Such processing capability can provide significant advantages for image capture, and for implementing truely programmable computational sensors. This is especially the case for applications requiring that processing be performed on the signals during integration, or that the operation of individual photodetectors be altered during integration, e.g., to adapt the image capture or processing to different environments. For other applications, the only advantage we see for interpixel, versus column or chip level processing, is to reduce power consumption.

Figure 8 depicts the generic architecture of a programmable digital pixel. It comprises a photodetector, analog signal conditioning, an ADC, memory, and a programmable digital processor. The processor can communicate locally, or may be globally, with other pixel processors. An example of such architecture is the PAR vision chip.[44] Excluding the photodetectors, analog conditioning circuits, and the ADCs, a programmable digital pixel sensor is basically a fine grain parallel processing system. The most likely candidates would be bit-serial architectures[50,51] due to the limited pixel area. Once the target applications are defined, the pixel level processor architecture and programming paradigm can borrow from the vast literature and experience in this area.

ACKNOWLEDGEMENTS

REFERENCES

1. M. Loinaz, K. Singh, A. Blanksby, D. Inglis, K. Azadet, and B. Ackland, "A 200mW 3.3V CMOS color camera IC producing 352x288 24b Video at 30frames/s," in *ISSCC Digest of Technical Papers*, pp. 168–169, (San Fransisco, CA), February 1998.
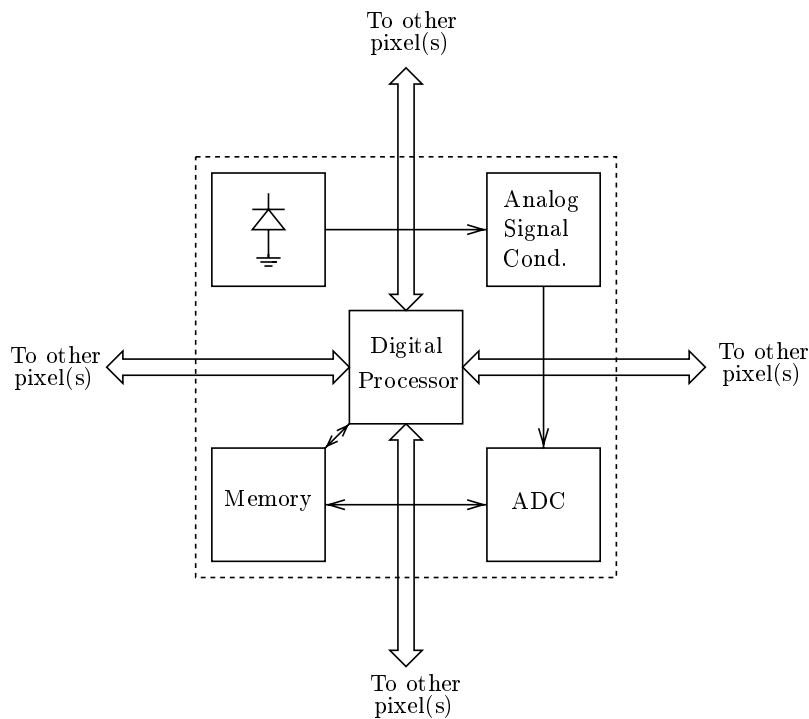
Figure 8. Programmable digital pixel.

2. S. Smith, J. Hurwitz, M. Torrie, D. Baxter, A. Holmes, M. Panaghiston, R. Henderson, A. Murray, S. Anderson, and P. Denyer, "A single-chip 306x244-pixel CMOS NTSC video camera," in *ISSCC Digest of Technical Papers*, pp. 170–171, (San Fransisco, CA), February 1998.

3. U. Ringh, C. Jansson, and K. Liddiard, "Readout concept employing a novel on Chip 16 bit ADC for smart IR focal plane arrays," in *Proc. SPIE*, vol. 2745, pp. 99–110, (Orlando, FL), April 1996.

4. B. Pain and E. Fossum, "Approaches and analysis for on-focal-plane analog-to-digital conversion," in *Proc. SPIE*, vol. 2226, pp. 208–218, (Orlando, FL), April 1994.

5. C. Mead, *Analog VLSI and Neural Systems*, Addison Wesley, 1989.

6. D. Yang, A. El Gamal, B. Fowler, and H. Tian, "A 640×512 CMOS Image Sensor with Ultra Wide Dynamic Range Floating Point Pixel Level ADC," in *ISSCC Digest of Technical Papers*, (San Fransisco, CA), February 1999. Submitted to ISSCC99.

7. H. Wong, "Technology and Device Scaling Considerations for CMOS Imagers," *IEEE Transactions on Electron Devices* **43**(12), pp. 2131–2141, 1996.

8. Manabe *et al.*, "A 2 million pixel CCD image sensor overlaid with amorphous silicon photoconversion layer of HDTV systems," *Optoelectronis - Devices and Technology* **6**(2), pp. 301–310, 1991.

9. R. B. Apte *et al.*, "Large Area Low Noise Amorphous Silicon Imaging System," in *Proc. SPIE*, vol. 3301, pp. 2–8, (San Jose, USA), Jan 1998.

10. D. Yang, B. Fowler, and A. El Gamal, "A Nyquist Rate Pixel Level ADC for CMOS Image Sensors," in *Proc. IEEE 1998 Custom Integrated Circuits Conference*, pp. 237–240, (Santa Clara, CA), May 1998.

11. E. Fossum, "Active Pixel Sensors: are CCD's dinosaurs," in *Proc. SPIE*, vol. 1900, pp. 2–14, (San Jose, CA), February 1993.

12. E. R. Fossum, "CMOS Image Sensors: Electronic Camera On A Chip," in *Proceedings of International Electron Devices Meeting*, pp. 17–25, (Washington, DC), December 1995.

13. G. Weckler, "Operation of p-n junction photodetectors in a photon flux integration mode," *IEEE Journal of Solid State Circuits* **SC-2**(3), pp. 65–73, 1967.

14. R. Forchheimer and A. Odmark, "A Single Chip Linear Array Processor, in Applications of digital image processing," in *Proc. SPIE*, vol. 397, pp. 425–30, (Geneva, Switzerland), April 1983.

15. R. Forchheimer, P. Ingelhag, and C. Jansson, "MAPP2200, a second generation smart optical sensor," in *Proc. SPIE*, vol. 1659, pp. 2–11, (San Jose, CA), February 1992.

16. P. Denyer, D. Renshaw, G. Wang, M. Lu, and S. Anderson, "On-Chip CMOS Sensors for VLSI Imaging Systems," in *VLSI-91*, 1991.

17. P. Denyer, D. Renshaw, G. Wang, and M. Lu, "A Single-Chip Video Camera with On-Chip Automatic Exposure Control," in *ISIC-91*, 1991.

18. P. Denyer, D. Renshaw, G. Wang, and M. Lu, "CMOS Image Sensors For Multimedia Applications," in *Proc. IEEE 1998 Custom Integrated Circuits Conference*, pp. 11.5.1–11.5.4, (San Diego, CA), May 1993.

19. S. Mendis, S. Kemeny, and E. Fossum, "A 128×128 CMOS Active Pixel for Highly Integrated Imaging Systems," in *IEEE IEDM Technical Digest*, pp. 583–6, (San Jose, CA), December 1993.

20. S. Mendis *et al.*, "Progress in CMOS Active Pixel Image Sensors," in *Proc. SPIE*, vol. 1994, pp. 19–29, (San Jose, CA), February 1994.

21. R. Panicacci *et al.*, "128 Mb/s Multiport CMOS Binary Active-Pixel Image Sensor," in *ISSCC96 Techincal Digest*, February 1996.

22. *Technology Roadmap for Image Sensors*, OIDA Publications, 1998.

23. O. Yadid-Pecht, R. Ginosar, and Y. Diamand, "A random Access Photodiode Array for Intelligent Image Capture," *IEEE Transactions on Electron Devices* **38**, pp. 1772–1780, August 1991.

24. S. Mendis, S. Kemeny, R. Gee, B. Pain, C. Staller, Q. im, and E. Fossum, "CMOS Active Pixel Image Sensors for Highly Integrated Imaging Systems," *IEEE Journal of Solid State Circuits* **32**, pp. 187–197, February 1997.

25. A. Dickinson, B. Ackland, E. Eid, D. Inglis, and E. Fossum, "A 256×256 Active Pixel Image Sensor with Motion Detection," in *ISSCC95 Techincal Digest*, February 1995.

26. A. Dickinson, S. Mendis, D. Inglis, K. Azadet, and E. Fossum, "CMOS Digital Camera With Parallel Analog-to-Digital Conversion Architecture," in *1995 IEEE Workshop on Charge Coupled Devices and Advanced Image Sensors*, April 1995.

27. S. Decker, R. McGrath, K. Brehmer, and C. Sodini, "A 256x256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output," in *ISSCC Digest of Technical Papers*, pp. 176–177, (San Fransisco, CA), February 1998.

28. O. Yadid-Pecht, B. Pain, C. Staller, C. Clark, and E. Fossum, "CMOS Active Pixel Sensor Star Tracker with Regional Electronic Shutter," *IEEE Journal of Solid State Circuits* **32**, pp. 285–288, February 1997.

29. M. Kyomasu, "A New MOS Imager Using Photodiode as Current Source," *IEEE Journal of Solid State Circuits* **26**, pp. 1116–1122, August 1999.

30. K. Aizawa *et al.*, "On Sensor Image Compression," *IEEE Transactions on Circuits and Systems for Video Technology* **7**, pp. 543–48, June 1997.

31. C. Mead, "A Sensitive Electronic Photoreceptor," in *1985 Chapel Hill Conference on VLSI*, (Chapel Hill, NC), 1985.

32. B. Dierickx, D. Scheffer, G. Meynants, W. Ogiers, and J. Vlummens, "Random Addressable Active Pixel Image Sensors," in *Proc. SPIE*, vol. 2950, pp. 2–7, (Berlin, FRG), October 1996.

33. R. Lyon, "The optical mouse, and an architectural method for smart digital sensors," in *Conference on VLSI Systems*, p. 1, (Pittsburg PA, USA), 1981.

34. J. Tanner *et al.*, "A correlating optical motion detector," in *Conference on Advanced Research in VLSI*, p. 57, (Cambridge MA, USA), January 1984.

35. X. Arreguit *et al.*, "A CMOS motion detector system for pointing devices," in *ISSCC Digest of Technical Papers*, (San Fransico, CA), February 1996.

36. J. Kramer, G. Indiveri, and C. Koch, "Motion Adaptive Image Sensor for Enhancement and Wide Dynamic Range," in *Proc. SPIE*, vol. 2950, pp. 50–63, (Berlin, FRG), October 1996.

37. N. Ancona, G. Creanza, D. Fiore, and R. Tangorra, "A Real-time, Miniaturized Optical Sensor for Motion Estimation and Time-to-Crash Detection," in *Proc. SPIE*, vol. 2950, pp. 75–85, (Berlin, FRG), October 1996.

38. M. Sivilotti *et al.*, "Real-time visual computations using analog CMOS processing arrays," in *Conference on Very Large Scale Integration*, p. 295, (Cambridge MA, USA), 1987.

39. T. Delbruck, *Investigations of Analog VLSI Visual Transduction and Motion Processing*. PhD thesis, California Institute of Technology, 1993.

40. I. Koren *et al.*, "Design of a Focal Plane Array with Analog Neural Preprocessing," in *Proc. SPIE*, vol. 2950, pp. 64–74, (Berlin, FRG), October 1996.

41. A. Astrom, R. Forchheimer, and J. Eklund, "Global Feature Extraction Operations for Near-Sensor Image Processing," *IEEE Transactions on Image Processing* **5**, pp. 102–110, January 1996.

42. V. Brajovic and T. Kanade, "A Sorting Image Sensor: An Example of Massively Parallel Intensity-to-Time Processing for Low-Latency Computational Sensors," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 1638–43, (Minneapolis, Minnesota), April 1996.

43. A. Rodriguez-Vazquez, S. Espejo, R. Dominguez-Castro, R. Carmona, and E. Roca, "Mixed-Signal CNN Array Chips for Image Processing," in *Proc. SPIE*, vol. 2950, pp. 218–229, (Berlin, FRG), October 1996.

44. F. Paillet, D. Mercier, and T. Bernard, "Making the most of $15k\lambda^2$ silicon area for a digital retina PE," in *Proc. SPIE*, vol. 3410, pp. 158–167, (Zurich, Switzerland), May 1998.

45. B. Pain, S. Mendis, R. Schober, R. Nixon, and E. Fossum, "Low-power low-noise analog circuits for on-focal-plane signal processing of infrared sensors," in *Proc. SPIE*, vol. 1946, pp. 365–374, (Orlando, FL), April 1993.

46. W. Yang, "A Wide-Dynamic-Range Low-Power photosensor Array," in *ISSCC Digest of Technical Papers*, pp. 230–231, (San Francisco, CA, USA), February 1994.

47. U. Ringh, C. Jansson, C. Svensson, and K. Liddiard, "CMOS analog to digital conversion for uncooled bolometer infrared detector arrays," in *Proc. SPIE*, vol. 2474, pp. 88–97, (Orlando, FL), April 1995.

48. B. Fowler, A. El Gamal, and D. X. D. Yang, "A CMOS Area Image Sensor with Pixel-Level A/D Conversion," in *ISSCC Digest of Technical Papers*, pp. 226–227, (San Fransico, CA), February 1994.

49. D. Yang, B. Fowler, and A. El Gamal, "A 128×128 Pixel CMOS Area Image Sensor with Multiplexed Pixel Level A/D Conversion ," in *Proc. IEEE 1996 Custom Integrated Circuits Conference*, pp. p303–306, (San Diego, CA), May 1996.

50. S. Y. Kung, *VLSI Array Processors.*, Prentice Hall Series, 1987.

51. R. Lyon, "A Bit-Serial VLSI Architectural Methodology for Signal Processing," in *Proceedings of the First International Conference on Very Large Scale Integration*, pp. 131–40, (Edinburgh, UK), August 1981.