Adaptive sensing and image processing with a general-purpose pixel-parallel sensor/processor array integrated circuit

Piotr Dudek

School of Electrical and Electronic Engineering, University of Manchester
PO Box 88, Manchester M60 1QD, United Kingdom
p.dudek@manchester.ac.uk

Abstract

In this paper, a pixel-parallel image sensor/processor architecture with a fine-grain massively parallel SIMD analogue processor array is overviewed and the latest VLSI implementation, SCAMP-3 vision chip, comprising 128×128 array, fabricated in a 0.35µm CMOS technology, is presented. Examples of real-time image-processing executed on the chip are shown. Sensor-level data reduction, wide dynamic range and adaptive sensing algorithms, enabled by the sensor-processor integration, are discussed.

1. Introduction

Computer vision, especially low-level image processing, is computationally demanding. Since the early vision algorithms exhibit inherent fine-grain (pixel-level) parallelism, it is beneficial to employ massively parallel processor arrays. Our work [1-4] has focused on the development of such arrays, integrated with image sensors, in a single silicon chip on a pixel-per-processor basis. The latest 'vision chip', SCAMP-3 contains a 128×128 general-purpose SIMD processor array, capable of executing over 20 GOPS at below 250 mW power consumption.

The integration of image sensors within general-purpose processor array enables implementation of many sensor-level processing algorithms that have been implemented before using dedicated circuitry. These include motion-detection, multi-resolution read-out with pixel binning, high dynamic range sensing with multiple exposure, locally adaptive sensing, in-pixel A/D conversion, active contours etc. It should be noted, that this multiple functionality can be obtained without significant penalty in terms of cell area, indeed, in some cases the general-purpose solution can achieve smaller cell area and power consumption than dedicated hard-wired ASIC solutions.

In this paper, the design and implementation of the SCAMP-3 vision chip will be briefly overviewed. This will be followed by presentation of experimental results, illustrating real-time image-processing performed by the chip. In particular, adaptive sensing algorithms will be considered. A spatially adaptive thresholding will be used as an example, and it will be demonstrated how sensor/processor integration and pixel-parallel processing enables effective handling of wide dynamic range images.

2. SIMD pixel-parallel vision chip

The basic architecture of the SIMD pixel-parallel vision chip is shown in Fig. 1. Images are focused onto the chip using a lens. The device acts as a CMOS camera, capturing images via an array of photosensors. However, unlike a conventional camera, the 'pixels' in the vision chip contain processors or 'processing elements' (PEs), one PE per pixel. These processors are very simple – they contain a few memory storage locations, an ALU capable of performing arithmetic and logic operations, and basic control and I/O circuits. Overall, the PEs make up a processor array. The array operates in SIMD (Single Instruction Multiple Data) mode, i.e. the same instruction word is delivered to all PEs in the array, and they operate on their local data (initially, these data are image pixels acquired via individual photodetectors), manipulating, and locally exchanging information, to execute desired image processing operation.

1-4244-0686-2/06/$20.00 C2006 IEEE

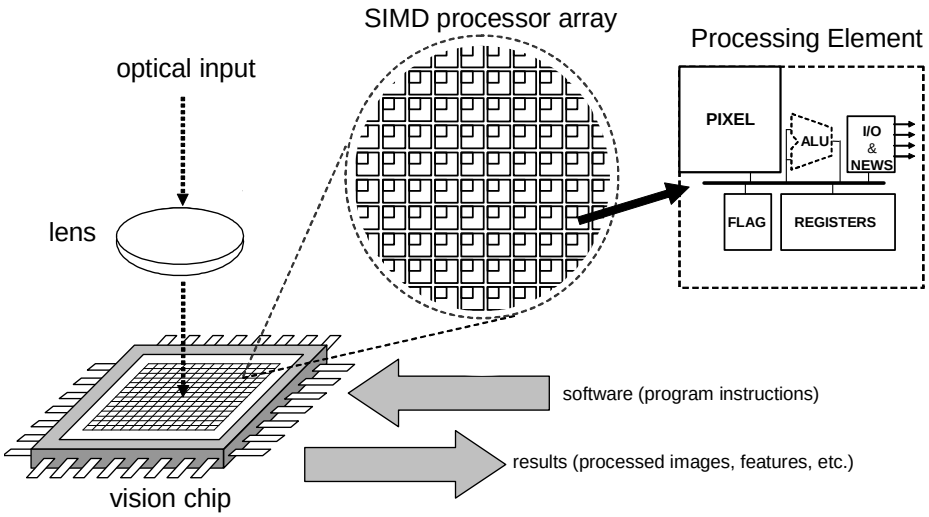


Fig.1. Overview of the pixel-parallel SIMD vision chip concept

3. Integrated circuit implementation

The main challenge when designing a SIMD pixel-parallel sensor/processor array is the design of a compact, low-power, but versatile and fully programmable processing element. The 'analogue microprocessor' concept was introduced in [1] and several generations of vision chips based on this concept have been implemented [2-4]. The latest device is a 128×128 array chip, implemented in a 0.35µm technology. The layout of the processing element is shown in Fig. 2.

More detail about the chip design can be found in the aforementioned papers, here the basic concept behind the PE design will be only briefly overviewed.

3.1 Analogue Processing Element

The processing element should include local memories, perform basic arithmetic and logic operations, and communicate with its nearest neighbours. This functionality is achieved in a very small footprint, using analogue sampled-data approach and switched-current (SI) circuitry. A simplified schematic diagram of the analogue processing element (APE) is shown in Fig. 3. All switches are controlled externally, and direct the current flow to/from the SI memory cells, via the 'analogue bus', following the instructions of the software program. Inversions and additions are obtained via current-mode arithmetics, which results in small silicon area and low power consumption requirements.

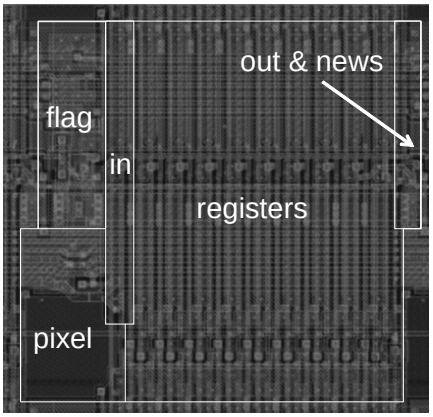


Fig.2. The APE layout. In a 0.35µm CMOS technology the cell size is 49.35µm×49.35µm

The photodetector works in the integration mode, integrating photocurrent on the parallel capacitance of the photodiode and gate-source capacitance of transistor M_{PIX}. The integrating voltage is reset closing switch r_P. Non-destructive readout can be performed at any time – when switch s_{PIX} is closed the current of transistor M_{PIX} is delivered to the analogue bus of the APE. This current, indicating the brightness of the image at the corresponding pixel, can be stored in one of the registers, and manipulated in the APEs, according to the program instructions.

The diagram in Fig. 3 contains simplified views of the basic cells. The actual circuitry is somewhat more complicated: additional devices are used to linearise the relation between pixel current and voltage at the integrating capacitance of the photodiode, to implement 'local activity flag' feature of the SIMD

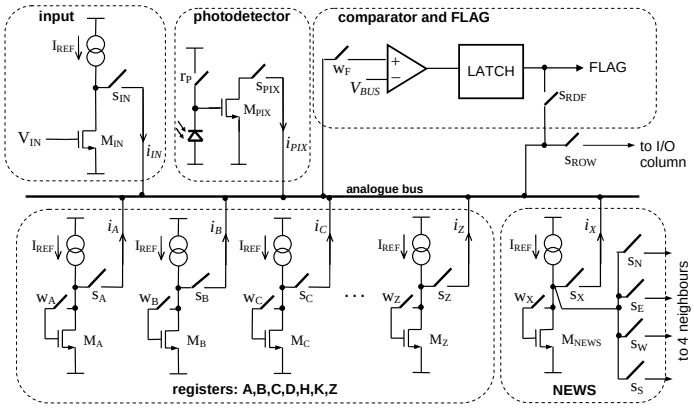


Fig.3. Simplified schematic diagram of the Analogue Processing Element

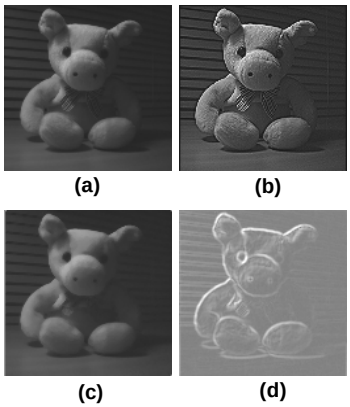


Fig.4. Examples of image processing executed on chip: (a) acquired image, (b) sharpening filter, (c) 3x3 median filter, (d) Sobel edge detection

architecture, to reduce power consumption of idle cells, to compensate for some errors of analogue storage cells, etc. Overall, the APE comprises 111 transistors, and implemented in a 0.35µm CMOS technology measures below 250µm^2 and consumes 12µW when executing instructions at a rate of 1.25MHz.

4. Sensor-level Image Processing

The SCAMP-3 chip contains a general-purpose processor array, its functionality is thus determined by the algorithm (software). Several typical early vision algorithms executed on the chip are shown in Figure 4. The chip acquires and processes the images, outputting only the processing result. The execution times for these algorithms are as follows: edge detection (using Sobel algorithm) 30 µs, sharpening filter (using 3×3 convolution kernel) – 17 µs, median filter (using 3×3 neighbourhood) – 157 µs. It is apparent that processing at very high frame rates is possible. At the same time, power consumption is extremely low, for example to execute the Sobel Edge Detection at 20 frames per second the chip requires only 7µW of power.

In the current system, the read-out speed is the bottleneck. The chip supports read out speeds of up to 200 gray-scale 128×128 images per second. However, higher frame rates can be achieved, employing sensor-level data compression and feature extraction.

The chip supports multi-resolution readout and region-of-interest readout, as well as global operations (summation and OR). For example, Fig. 5 illustrates multi resolution readout with pixel binning - only one gray-scale value per oversized pixel is read-out from the chip.

Another useful feature is the ability to output processing results only (rather than gray-scale images) in the form of binary images (via high-speed, digital, 8-column parallel read-out port) or just pixel coordinates. In this way, processing speed of thousands fps are feasible. An example is shown in Fig.6 where the chip executes a segmentation algorithm based on active contours [5]. For display purposes, the result in Fig.6 is superimposed on the gray-level image, however in an application only a binary image of the contour could be read-out.

Furthermore, rather than outputting binary images, some image features could be extracted and only scalar values read-out. Examples include object counting, or returning coordinates of object centres in a target tracking application.

5. Adaptive Sensing

An important aspect of sensor/processor integration is the ability to access sensory data not only *after*, but also *during* image acquisition. This makes the system able not only to perform image processing faster and at lower power than a conventional "sensor followed by a processor" system, but also providing additional levels of functionality. One of these is the ability to handle very wide dynamic range images, characteristic of natural scenes, which are difficult (or impossible) to handle using conventional systems. Consider a scene shown in Fig. 7. The dynamic range of light intensities (from the object in the shadow indoors, through outdoor scene, to a light bulb) is in excess of the dynamic range of the image sensor working in the integration mode, using simple linear image acquisition. It is thus impossible to reliably process this scene after image acquisition using a single exposure/iris setting. Two different exposure times (i.e. photocurrent integration times) have been used in Fig. 7a and Fig.7b. It can be seen that in each case some

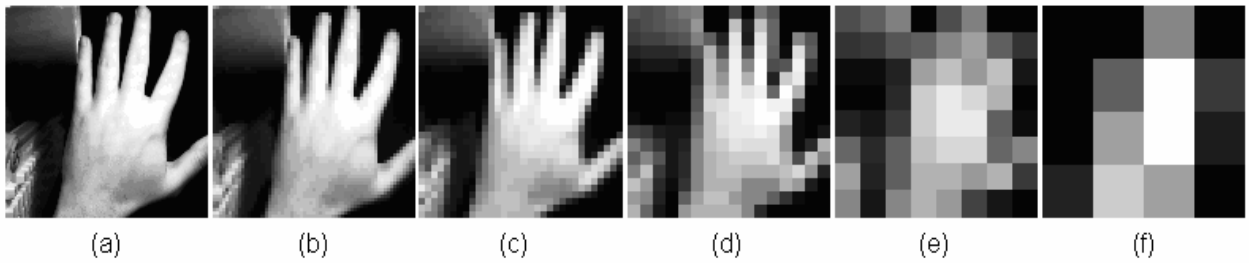


Fig.5. Multiple resolution readout with pixel binning: (a) 128x128 - full resolution, (b) 64x64, (c) 32x32, (d) 16x16, (e) 8x8, (f) 4x4

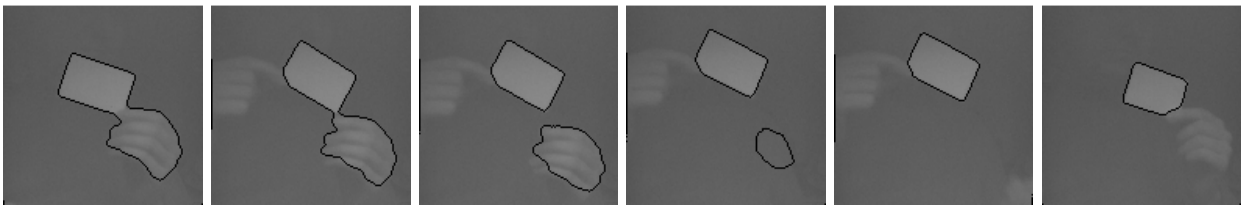


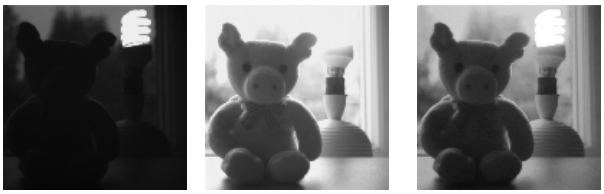
Fig.6. Image segmentation with cellular active contours algorithm [5] implemented on chip. In an application, only the contours locations (binary images) could be read out.

detail is lost - either dark elements are below noise level, or bright elements saturate.

Several methods can be used to deal with such images, either based on multiple exposure times [6], or using sensors with logarithmically compressed sensitivity [7]. In this paper it will be illustrated how these various strategies can be employed in our sensor/processor system to acquire and process wide dynamic range images, taking a simple example of image segmentation via thresholding (i.e. generating a binary image which, ideally, preserves image features in all image regions). A simple way to threshold an image is to assign '1' to all pixels that have their gray-level value above some fixed threshold level and '0' to all pixels below the threshold level. Obviously, using a simple threshold on images in Fig.7(a-b) is not going to provide adequate results, resulting in loss of information either in the light or dark parts of the image.

A simple method to extend dynamic range of the image sensor is to acquire image with various exposure settings. In Fig. 7(c) a composite image, resulting in adding two images with different exposure times is shown. (In this, and all further examples, all images are experimental results and all processing operations are performed on the vision chip.) The dynamic range has been thus compressed and detail in both the light bulb and the dark foreground is visible. It has to be emphasized, that unlike in a conventional system implementing this strategy, the complete gray-level image does not have to be read out from the chip multiple times. Indeed, in our example, the gray-level images don't need to be output at all - the only read-out information will be the optimally thresholded binary image.

A summation of images, taken with multiple exposure times corresponds to a piecewise-linear compression curve, as shown in Fig.8(a). A corresponding characteristic, shown on the logarithmic scale of light intensities is shown in Fig. 8(b). For a more regular compression curve many exposures should be ideally taken. One difficulty in increasing the number of exposures on an analogue processor array is the necessity to handle many multiplications and additions, which introduce noise. Another is related to the leakage of analogue memories, especially prominent at high light intensity locations (the chip surface is exposed to light, which makes long-term analogue storage unreliable). This makes it difficult to use both very short and very long exposures for the same frame. A solution is to perform in-pixel A/D during integration, in a way similar to described in [8]. Any shape of compressed characteristic can be achieved, simply by changing the relation between code and integration time. The digital values can be refreshed, so there is no



(a) **(b)** **(c)**

Fig.7. High dynamic range images acquired by the chip using two different exposure settings: (a) short integration time, (b) long integration time; (c) The composite image (calculated on-chip)

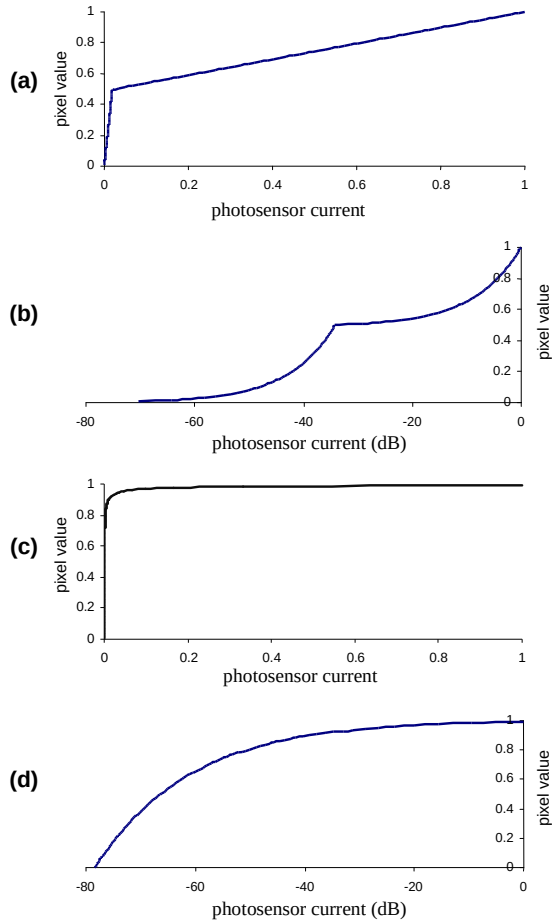


Fig.8. Wide dynamic range imaging characteristics – normalized photosensor current (which is approximately proportional to light intensity) versus normalized pixel gray-scale value result: (a) and example piecewise linear characteristic; (b) as above, but on log scale; (c) compressed characteristic, according to equation (1); (d) as above, but on log scale

problem in long-term storage in this case. After A/D conversion, the result can be converted back to analogue, using in-pixel D/A conversion. An example compression characteristic is shown in Fig. 8(c) and Fig.8(d), using linear and logarithmic scales, respectively. In this case, a code n is assigned to pixels whose integration time t (i.e. time between voltage across photodiode capacitance being reset to this voltage reaching a threshold value, as is discharges due to photocurrent) is:

$$n(n-1)\,\tau < t < n(n+1)\tau \qquad (1)$$

where τ is an arbitrary time constant.

An image obtained using this compression method is shown in Fig.9(b). Figure 9(a) contains result of linear A/D followed by D/A conversion, for direct comparison. Other conversion characteristics, enabling even wider dynamic range, are possible. Similarly,

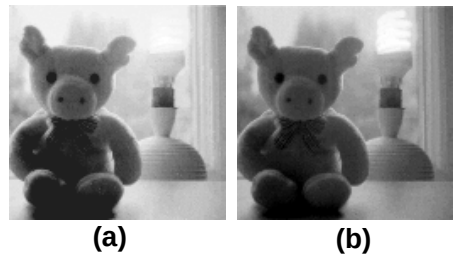


Fig.9. Dynamic range compression achieved through non-linear integration time and A/D followed by D/A conversion: (a) linear conversion result, detail in lights and shadows is lost (b) compression with characteristic obtained using equation (1)

other, more complex adaptive sensing strategies [9] could be implemented with the presented architecture.

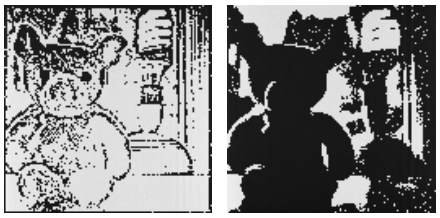
Once the wide-dynamic range images are obtained, thresholding can be performed. Using fixed threshold value will still not give adequate results (Fig.10 a-d), but using adaptive threshold leads to a much better result, shown in Fig.10(e). The threshold level is set individually for each pixel, based on average image brightness in a local area surrounding this pixel. In this example, the average brightness has been determined through a diffusion of the gray-scale image, i.e. spatial convolution of the image with an approximately Gaussian kernel of arbitrary size. This size can be adjusted, to achieve required granularity of the adaptive thresholding. As can be seen in Fig. 10(e) some detail in both dark and light areas has been preserved.

The method described above still relies on image acquisition, followed by image processing. One problem of this approach is that even though the dynamic range is extended, the resulting gray-level image is compressed into the available dynamic range of the processor (determined by the word-length in a conventional processor, and by signal/noise ratio in the analogue processor). This therefore improves the image acquisition, but still makes it impossible to handle fine detail in gray-level intensities, over the entire range.

An alternative and better method is to perform image processing operations during image acquisition. Images shown in Fig. 11 have been obtained by performing multiple locally adaptive image thresholding, during image acquisition. The dynamic range is thus extended, but it is not being compressed. The processing is performed continually, from the brightest to the darkest image regions, always operating on the full-scale images, such as the ones shown in Fig.7(a-b), and ignoring regions that are either already over-saturated or yet under-exposed. Adaptive thresholding is thus performed reliably over the entire range. Figure 11

(a) (b) (c) (d) (e)

Fig.10. Thresholding of image from Fig.9b. (a)-(d) using various levels of fixed threshold, (e) using adaptive threshold, based on local image brightness information.



(a) (b)

Fig.11. Results of adaptive thresholding algorithm which operates during image integration. Images (a) and (b) are obtained using various sizes of local neighbourhood.

contains results of two complete runs of the algorithm, with various sizes of local neighbourhood used for local threshold value adaptation. This size can be set according to application requirements.

A simple spatially adaptive thresholding has been used as an example here, but the above presented methods are general, and similar strategies can be employed when extracting edges, or other image features, from high dynamic range scenes.

6. Conclusions

SCAMP-3 vision chip is a pixel-parallel sensor/processor SIMD array. General-purpose nature of the system allows the implementation of a variety of low- and mid-level image processing algorithms directly on the chip. Integration of processors right next to the sensors enables data reduction at the sensor level, reducing bandwidth requirements between the front-end device and the rest of the computer vision system. This integration also enables various adaptive sensing strategies, that can be employed to efficiently handle wide dynamic range images.

Massively parallel processing leads to real-time performance, enabling processing at high frame rates. Small cell size, achieved due to performing processing in the analogue domain, makes relatively large pixel-parallel processor arrays feasible to be implemented on a single silicon die. Low power consumption makes the approach attractive, in particular for embedded systems for machine perception in applications such as autonomous robots, vision systems for toys, vehicles, security and surveillance.

Acknowledgements

The author acknowledges David Barr and Steve Carey for their assistance in developing the system and performing experiments. This work has been supported by the EPSRC grants no. EP/D029759/1 and EP/D503213/1.

References

[1] P.Dudek and P.J.Hicks, “A CMOS general-purpose sampled-data analog processing element”, *IEEE Trans. on Circuits and Syst. - II*, vol. 47, no. 5, pp. 467-473, May 2000

[2] P.Dudek and P.J.Hicks, "A General-Purpose Processor-per-Pixel Analog SIMD Vision Chip", *IEEE Trans. on Circ. and Syst. - II*, vol. 52, no. 1, pp. 13-20, January 2005

[3] P.Dudek, "A 39x48 General-Purpose Focal-Plane Processor Array Integrated Circuit", *Proc. ISCAS’04*, vol.V, pp.449-452, May 2004

[4] P.Dudek, "Implementation of SIMD Vision Chip with 128x128 Array of Analogue Processing Elements", *Proc. ISCAS’05*, pp.5806-5809, May 2005

[5] P.Dudek and D.Vilariño, “A Cellular Active Contours Algorithm Based on Region Evolution”, IEEE Workshop on CNNs and their Application, CNNA 2006, August 2006.

[6] O.Yadid-Pecht and E.R.Fossum, “Wide Intrascene Dynamic Range CMOS APS Using Dual Sampling”, *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1721-1723, October 1997

[7] S.Kavadias, B.Dierickx, D.Scheffer, A.Alaerts, D.Uwaerts and J. Bogaerts, “A Logarithmic Response CMOS Image Sensor with On-Chip Calibration”, *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp.1146-1152, August 2000

[8] A. Kitchen, A.Bermak and A.Bouzerdoum, “A Digital Pixel Sensor Array with Programmable Dynamic Range”, *IEEE Transactions on Electron Devices*, vol. 52, no. 12, pp. 2591-2601, December 2005

[9] R. Wagner, A. Zarándy and T. Roska, “Adaptive Perception with Locally Adaptable Sensor Array”, IEEE Trans. On Circuits And Systems-I: Regular Papers, vol 51, no.5, May 2004, pp. 1014-1023