Image Filtering by Reduced Kernels Exploiting Kernel Structure and Focal-Plane Averaging

J. Fernandez-Berni, R. Carmona-Galan, A. Rodríguez-Vázquez
Institute of Microelectronics of Seville (IMSE-CNM)
CSIC-Universidad de Sevilla
C/ Américo Vespucio s/n, 41092, Seville, Spain
Email: berni@imse-cnm.csic.es

Abstract— Incorporating multi-resolution capabilities into imagers renders additional power saving mechanisms in the subsequent image processing. In this paper, we show how, by exploiting a certain mask structure, 3×3 kernels can be reduced to 2×2 kernels if charge redistribution is provided at the focal plane of the imaging device. More precisely, by successively averaging two shifted half-resolution pixel grids, we will have a pre-processed image, subsampled by a factor of 2 on each dimension. It can also be extended to even lower resolution images if required. These pre-processed images can be filtered, for kernels matching the prescribed conditions, with a mask of a reduced size. Very useful image filtering kernels, like a 3×3 Gaussian kernel for image smoothing, or the well-known Sobel operators, fall into this category of reducible kernels. Operating onto the pre-processed image with one of these reduced kernels represents a smaller number of operations per pixel than realizing all the multiply-accumulate operations needed to apply a 3×3 kernel. Memory accesses are reduced in the same fraction. Concerning the difficulties of providing this pre-processed image representation, we propose a methodology for obtaining it at a very low power cost. It requires the implementation of user definable image subdivision and subsampling. Experimental results are given, obtained from measurements on a CMOS imager prototype chip incorporating these multi-resolution capabilities.

I. INTRODUCTION

The advances in CMOS integration have permitted the development of smart CMOS image sensors [1]. These chips incorporate concurrent image sensing and processing. One of the main advantages of this integration is the possibility of transferring a large part of the computational load associated with early vision tasks to the focal plane. The sensor array becomes a specialized processor with an adapted architecture. In early vision, processing is characterized by regular, local computations with inherent pixel-level parallelism. These computations are precisely the most time- and power-consuming tasks on DSP-based systems [2]. By incorporating processing capabilities at the focal-plane, in most of the occasions by efficiently using devices operating in analog mode, the computational load of the main digital processor can be greatly alleviated. The result is the realization of early vision tasks at record performances in speed, power and area [3–5]. In this paper we are showing how very simple additional circuits at the focal plane lead to improvements in the system power consumption. In particular, we will demonstrate that, by exploiting the internal structure of the convolution kernels, we can operate onto a focal-plane pre-processed image obtaining

virtually identical results with only 45% of the operations needed when applying the original kernels.

One of the most basic operations that can be implemented in the focal plane, without producing a significant signal degradation, is the averaging of disjoint groups of pixels. This can be achieved by charge redistribution right after, or even in parallel with, photocurrent integration. This will be the starting point of our proposal. We will explain how 3×3 kernels of a particular structure can be transformed into 2×2 kernels to be applied to the pre-processed images. Then we will show how this focal-plane pre-processing can be implemented at negligible power consumption with a working prototype chip.

II. IMAGE SUBDIVISION AND KERNEL REDUCTION

Let us consider a $M \times N$-pixel array. The value of each pixel is represented by a voltage resulting from integrating a photocurrent into a sensing capacitor during the exposure time —which is a very feasible implementation. Consider these capacitors being 4-connected through switches, what permits dividing the focal plane into rectangular blocks. Within each block, charge redistributes itself achieving voltage averaging. By configuring the grid to be regularly subdivided into 2×2-pixel size blocks, we will have an image of $M/2 \times N/2$ blocks. This state is depicted in Fig. 1(a), where every four pixels are labelled with the same value.

Now let us re-define the grid by shifting the edges of the grouping scheme one pixel down and one pixel to the right (Fig. 1(b)). Once the new grouping is enabled, charge redistributes again, and the values of the pixels, originally p_{ij}, $p_{i,j+1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$, are now averaged within each new block, resulting in:

$$
\begin{aligned}
p'_{ij} &= \tfrac{1}{4}\left(p_{i-1,j-1} + p_{i-1,j} + p_{i,j-1} + p_{ij}\right) \\
p'_{i,j+1} &= \tfrac{1}{4}\left(p_{i-1,j} + p_{i-1,j+1} + p_{ij} + p_{i,j+1}\right) \\
p'_{i+1,j} &= \tfrac{1}{4}\left(p_{i,j-1} + p_{ij} + p_{i+1,j-1} + p_{i+1,j}\right) \\
p'_{i+1,j+1} &= \tfrac{1}{4}\left(p_{ij} + p_{i,j+1} + p_{i+1,j} + p_{i+1,j+1}\right)
\end{aligned}
\tag{1}
$$

Notice that the output image, since we have started by averaging the 2×2-pixel blocks, will be a quarter of the size of the full-resolution sensor, i. e. half of the height and half of the width of the original image. The output images will be the result of applying the reduced kernel to the pre-processed image and the original kernel to the image before grid shifting
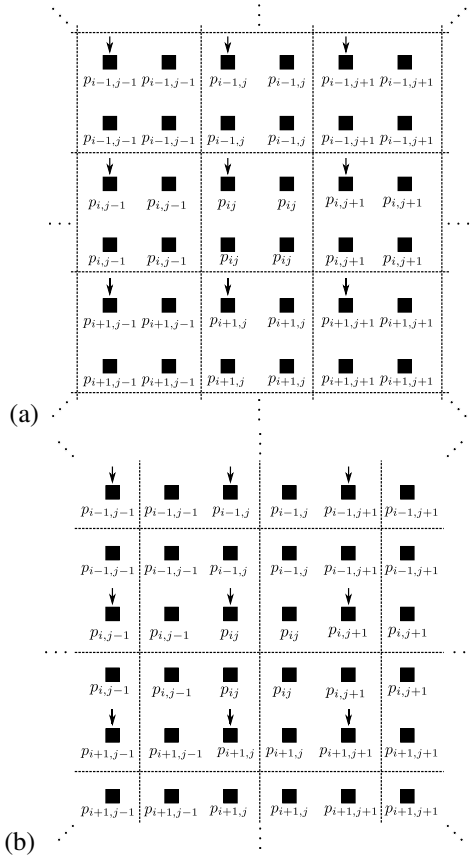
Fig. 1. (a) Focal plane subdivision into 2×2 blocks (a) and (b) shifted grid.

and averaging. Small arrows in Fig. 1(a) signal the quarter-size input image, while those in Fig. 1(b) point to the pixels that are going to be sampled to obtain the $M/2 \times N/2$ pre-processed image. This scheme can be extended to lower image resolutions provided that the size of the blocks is $B \times B$, being B an even number. In such a case, the grid must be shifted $B/2$ pixels to obtain a representation that is equivalent to that already described at the corresponding lower resolution. By applying some algebra, it can be seen that the result of applying the reduced kernel:

$$\mathbf{K}' = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad (2)$$

being pixel p'_{ij} the one weighted by the upper-left element, a, is the same as applying a 3×3 kernel of the form:

$$\mathbf{K} = \frac{1}{4} \begin{bmatrix} a & a+b & b \\ a+c & a+b+c+d & b+d \\ c & c+d & d \end{bmatrix} \qquad (3)$$

but centered in p_{ij}.

From the point of view of the digital implementation of the required signal processing, this simplification —if the pre-processed image can be efficiently generated at the focal plane, as we will see later— represents an important reduction in the computing needs. Instead of 9 MACs (multiply-accumulate operations), the pixel output can be obtained by using 4

MACs. This means only 45% of the resources required for the convolution of the original 3×3 kernel. Memory accesses has been reduced as only 4 pixels, instead of 9, need to be considered to evaluate the outputof each pixel. It must be said also that the relations required between kernel elements greatly restrict the number of kernels that can be reduced. Fortunately, some very useful templates in early vision processing fall into this category. For instance, the usual 3×3 binomial mask for image smoothing, which is a good approximation of a Gaussian filter with $\sigma \approx 0.7$ [6], is transformed into a 2×2 kernel in this way:

$$\mathbf{G}_s = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \rightarrow \quad \mathbf{G}'_s = \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad (4)$$

Other interesting templates are the Sobel operators [6]. They compute an approximation to the components of the image intensity gradient:

$$\mathbf{G}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \rightarrow \quad \mathbf{G}'_x = 4 \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \rightarrow \quad \mathbf{G}'_y = 4 \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$(5)$$

where $\mathbf{G}_x$ approximates the derivative in the horizontal direction while $\mathbf{G}_y$ approximates the derivative in the vertical direction. They are employed for edge detection as they highlight the fine details in a scene. Both of the kernels hold the prescribed structure expressed by Eq. (3).

III. FOCAL-PLANE IMPLEMENTATION OF PRE-PROCESSING

The type of pre-processing described by Eq. (1) can be implemented by the architecture described in [7] and depicted in Fig. 2. Each pixel contains a sensing capacitor and a set of switches that connect it to its nearest neighbors. One interesting property, included to implement multi-resolutional representations of the captured image, is that connections between neighboring rows and columns are user-selectable. Column and row selection signals are stored at serial-in/parallel-out shift registers located at the upper and leftmost sides of the pixel array, respectively. Any possible combination of 1's and 0's can be loaded into these registers in order to reproduce any particular connectivity pattern for, respectively, columns and rows. The connection pattern, however, does not become effective until a global signal is enabled after configuration. In this way, this enhanced imager is able to perform the operations described in Sect. II. First of all, pixels are grouped into 2×2 blocks by loading a bit pattern with alternate 1's and 0's in both the column and the row registers. After enabling the connectivity scheme already loaded, charge redistributes within each block, reaching the configuration depicted in Fig. 1(a). Right after that, connections are disabled, the bit pattern is shifted one position in both directions, horizontal and vertical. Then, the new connection scheme is enabled. The result is depicted in Fig. 1(b). Subsampling at the required
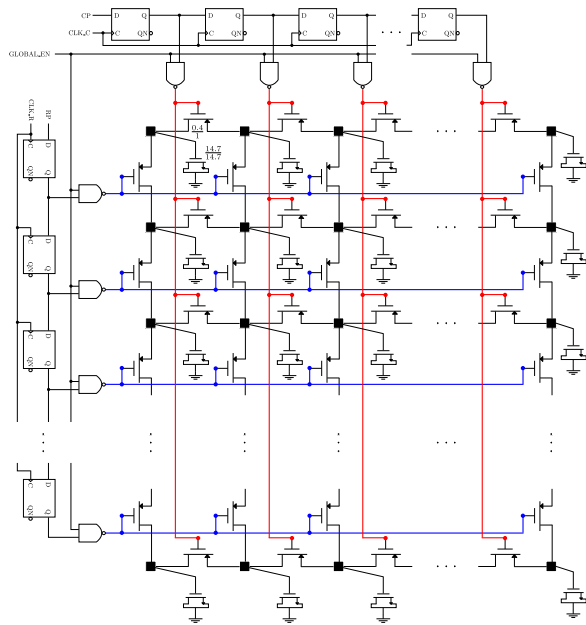
Fig. 2. Implementation of the capacitive lattice in the prototype chip.

Fig. 3. Chip photograph and data

Technology	0.35μm CMOS 2P4M 3.3V
Die size (with pads)	7280.8μm $\times$ 5780.8μm
Resolution	QCIF: 176$\times$144 px
Sensitivity	0.15V/(lux·s) (n-well/p-subs)
FPN/PRNU	0.72%/2.42%
Power consumption (worst case)	5.6mW@30fps

TABLE I
PROTOTYPE CHIP DATA.

positions is easily done because column and row selection is also realized with shift registers at the edge of the array.

Notice that if we consider the $M/2 \times N/2$ image to be our starting point —as we can have the connection pattern already loaded and enabled to realize charge redistribution simultaneously with photocurrent integration—, the only additional energy required to do the work is disabling the connection switches, shifting the registers one position, and re-enabling the connection switches with the new configuration. A rough estimation, obtained by evaluating the energy required to turn on and off the switches and to shift the register content, will render a power consumption below 0.33μW at 30fps for a QCIF-size array.

IV. CHIP RESULTS

We have implemented the focal-plane pre-processing in a prototype chip (Fig. 3) intended for low-power image processing, while the reduced kernels has been applied off-chip. The prototype chip contains all the elements to implement the required pre-processing at a low power cost. The main characteristics of the chip are summarised in Table I. As depicted in Figs. 4 and 5, we have operated on images captured at the laboratory (available at http://www.imse-cnm.csic.es/vmote/redkern). These images correspond to pictures of 'Lena' and the 'Baboon' displayed at a computer screen. Artifacts due to the screen grain can be observed. We proceeded in this way: first we took a snapshot of the computer screen, either showing 'Lena' or the 'Baboon', and then read out the image from the chip and performed the corresponding image filtering to the full-resolution image off-line. This is shown in the first row of Figs. 4 and 5. Then we grouped the pixels to form the half-resolution image, read it out and applied the filters by
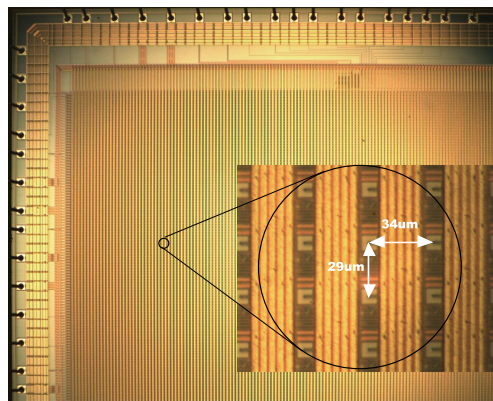
convolution with the 3×3 kernels, off-chip, for a reference. Then we shifted the pixel grouping on-chip and read out the pre-processed half-resolution image to apply the 2×2 kernels, also off-chip. The accuracy of the results is evidenced by the comparison of the two filtered versions of the image. The resulting images are perceptually equivalent. RMSE values are always below 1% in the Gaussian blur experiment, and below 4% in the application of the two Sobel filters. This is consistent with the fact that edge detection is the result of the off-chip application of two masks plus the computation of the absolute gradient value, what contributes to error spreading.

V. CONCLUSIONS

In this paper we have reported an example of how multi-resolution capabilities can lead to additional power savings. In particular, using the most elementary operations at the focal-plane, namely charge redistribution and user-definable image subdivision, the pre-processed image can be subsequently filtered with only 45% of the computational resources. In order to illustrate the validity of the approach, we have implemented the required image pre-processing in a prototype imager with focal-plane processing capabilities for multi-resolution representation of the scene. The errors committed, due to the analog nature of the processing in the focal plane, are kept below a reasonable bound for early vision applications.

REFERENCES

[1] J. Otha, *Smart CMOS Image Sensors and Appl.* CRC Press, 2008.
[2] S. Qureshi, *Embedded Image Processing on the TMS320C6000(TM) DSP.* Springer, 2005.
[3] A. Rodriguez-Vazquez et al., "ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 5, pp. 851–863, 2004.
[4] P. Dudek and P. Hicks, "A general-purpose processor-per-pixel analog SIMD vision chip," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 1, pp. 13–20, 2005.
[5] J. Poikonen, M. Laiho, and A. Paasio, "MIPA4k: A 64x64 cell mixed-mode image processor array," in *ISCAS 2009*, 2009, pp. 1927–1930.
[6] R. Gonzalez and R. Woods, *Digital Image Proc.* Prentice Hall, 2002.
[7] J. Fernández-Berni and et al., "FLIP-Q: A QCIF resolution focal-plane array for low-power image processing," *IEEE J. of Solid-State Circuits*, vol. 46, no. 3, pp. 669–680, March 2011.
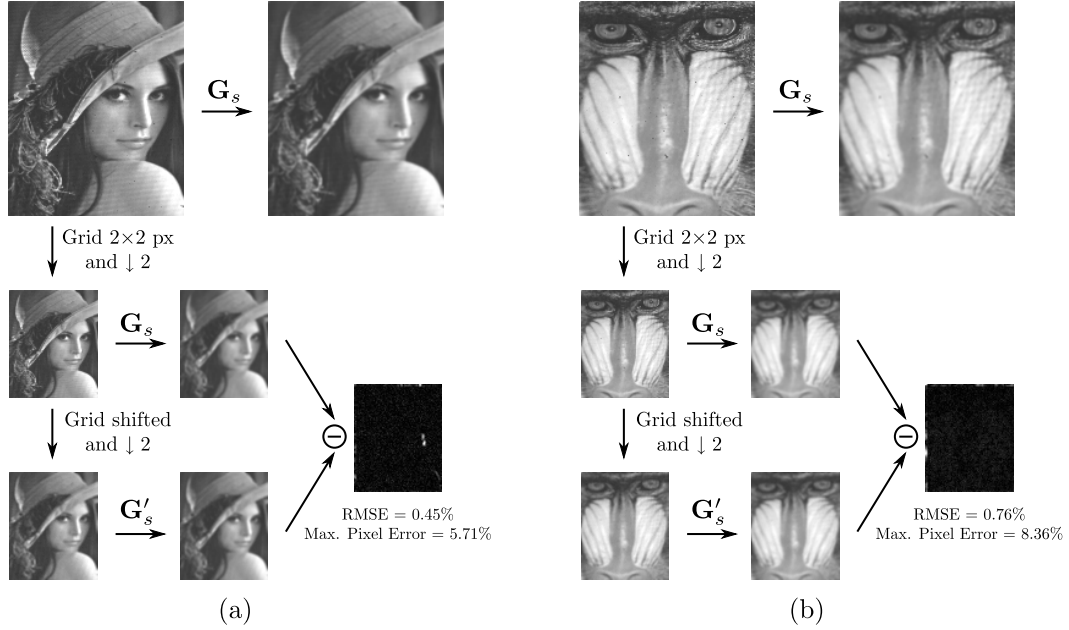
Fig. 4. Results obtained by applying $\mathbf{G}_s$ to (a) 'Lena' and (b) 'Baboon' images captured by the chip, and $\mathbf{G}'_s$ on the on-chip pre-processed versions.
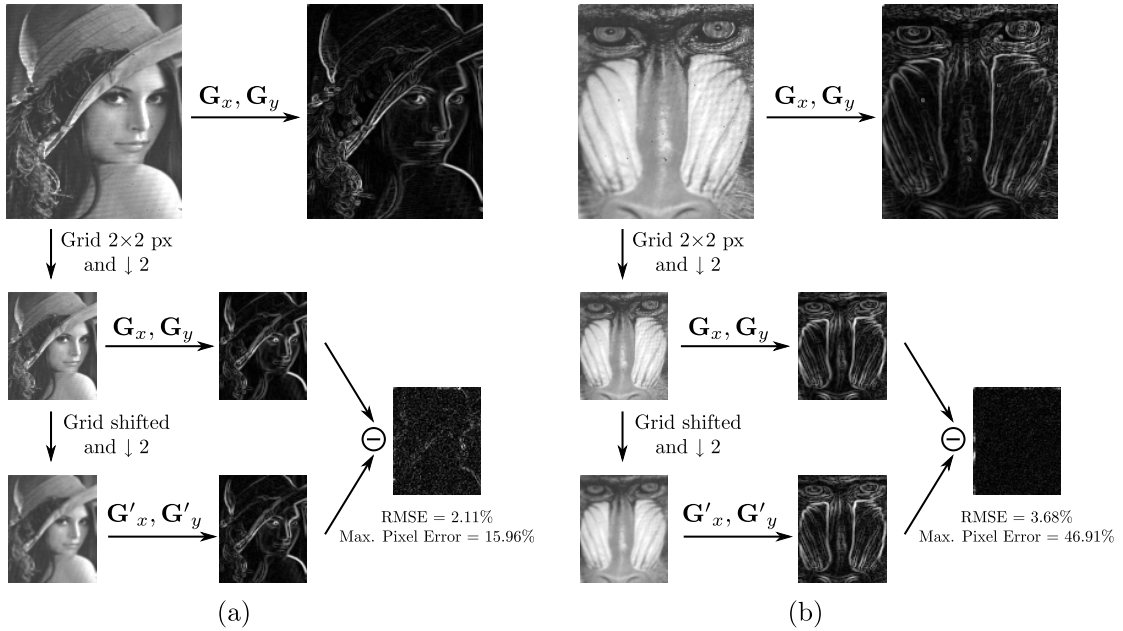

Fig. 5. Results obtained with $\mathbf{G}_x$ and $\mathbf{G}_y$, and with $\mathbf{G}'_x$ and $\mathbf{G}'_y$ on the on-chip pre-processed versions.