

# P13 Focal Plane Processing for a Fast Detection of 2D Motion Vectors

Zheng Li, Kiyoharu Aizawa

Dept. of Electrical Engr., University of Tokyo  
 Bunkyo-ku, Tokyo 113-8656, Japan  
 E-mail: liz@hal.t.u-tokyo.ac.jp, aizwa@hal.t.u-tokyo.ac.jp

## Abstract

This paper proposed and designed a CMOS imager with very fast detection of 2D motion vectors on the focal plane. The chip consists of a parallel pixel array and column parallel block-matching processors. Each pixel in the pixel array contains a photo detector, an edge detector and 4 bits of memory. In the fast detection of motion vectors, first, the gray level image is binarized by the edge detector and subsequently the binary edge data is used in the block matching processor. The block-matching takes place locally in pixel and globally in column. The chip can create a dense field of motion where a vector is assigned to each pixel by overlapping  $2 \times 2$  target blocks. A prototype with  $16 \times 16$  pixels and four block-matching processors has been implemented. Preliminary results obtained by the prototype are shown.

## 1 Introduction

Motion vector detection is one of the key techniques in image processing. The traditional way of designing a motion vector detector with a large searching area requires complex computations. When the design of the pixel array on the vision chip is determined, the resource of the focal plane becomes very limited. Thus, the conventional 2D motion vector detection and the improved design of hardware algorithms [1] will help little when they are integrated with the photo diode, edge detector and memory arrays on the focal plane. Because of these difficulties, focal plane processing of motion detection always requires other measures.

Thinking of an imaging process at a high frame rate, such as from 100 to 1000 frames/sec., it will make the changes very small between two successive frames. By making use of this feature, we can realize a very fast detection of motion vectors on the focal plane by a small-size block-matching within a small searching area. The computational cost is also reduced by binary operation after an edge detection.

The VLSI architecture of the very fast detection of 2D motion vectors is shown in Fig. 1, which includes the inputs (reset, bias, threshold, shift registers),  $16 \times 16$  pixel array, LPGCP processor's shifting window, controls, the output of motion vector detection and images/edges.

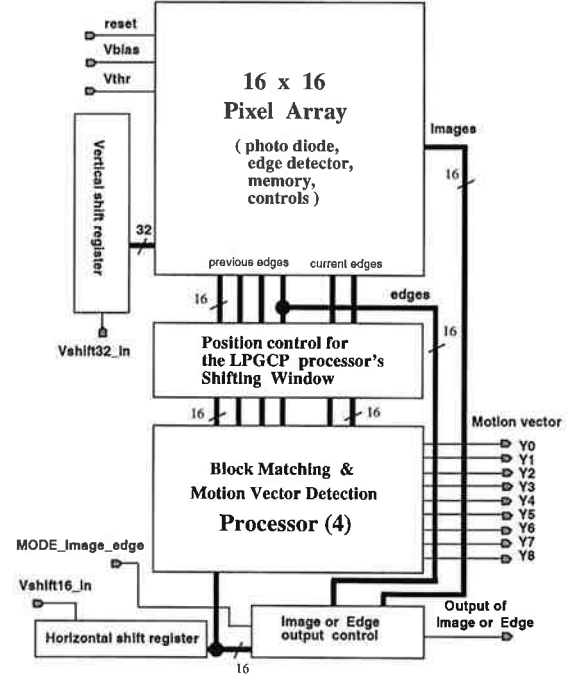


Figure 1: The VLSI construction of a very fast detection of 2D motion vectors on CMOS sensor focal plane. It includes inputs, outputs, and controls for  $16 \times 16$  pixel array, LPGCP processor's shifting window, block matching, the output of motion vector detection and images/edges.

## 2 Edge Detector Array

For an image pixel represented by  $f(i, j)$ , the edges can be simply extracted by the first order differential operation between 2 neighboring pixels in the horizontal and vertical directions, respectively.

$$Edge_{horizontal} = \begin{cases} 1 & Thr \leq |f(i, j) - f(i + 1, j)| \\ 0 & otherwise \end{cases}$$

$$Edge_{vertical} = \begin{cases} 1 & Thr \leq |f(i, j) - f(i, j + 1)| \\ 0 & otherwise \end{cases}$$

On the focal plane, the implementation of the above algorithm can be described by Fig. 2.

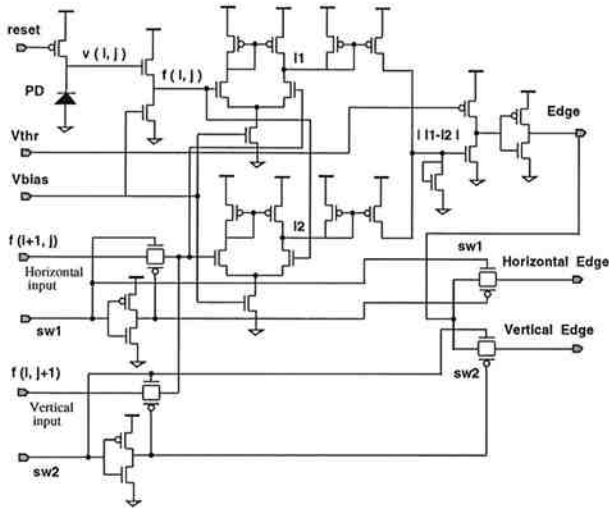


Figure 2: The photo diode and edge detector circuit used in the prototype design.

The photo diode and edge detector can form an array by a proper connection with 4-bit memory cell designed parallel in each pixel. The memory cell is developed from transmission gate<sup>[2]</sup>.

### 3 Fast Detection of Motion Vectors by Small Size Block Matching

Block matching is performed to search for the best motion vectors. In our prototype design, we applied a  $2 \times 2$  block matching and a small searching area of  $(\pm 1, \pm 1)$  pixels around the intentional area. We can obtain a dense motion field in which a motion vector is assigned to each pixel by overlapping  $2 \times 2$  target blocks.

#### 3.1 Fast block matching

The minimum-absolute difference(MAD)<sup>[3]</sup> is taken as the criterion for the block matching. The operation in MAD form can be written as

$$D_{x,y} = \sum_{i=-m}^m \sum_{j=-n}^n |T(x,y) - S(x+i,y+j)|.$$

$D_{x,y}$  is the Hamming distance when T and S are binary images, which represents the similarity between a block pattern T in present frame and a shifting block S in the previous frame.

$(-m \leq i \leq m, -n \leq j \leq n)$  is the searching area around the position  $(x,y)$ .

The simulations in Fig. 3 and Fig. 4 show that the motion trails detected by small-size block matching of binary edges show cleaner motion vectors than those achieved by using small-size gray pixels while also greatly reducing the arithmetic and logic operations. Researches<sup>[4]</sup> shown that it can improve the

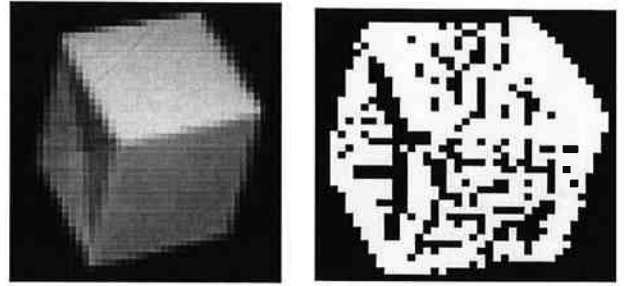


Figure 3: (a) a 2D moving object; (b) the motions detected by block matching with pixel values between 2 successive frames. The bright blocks indicate that motion vectors are detected.

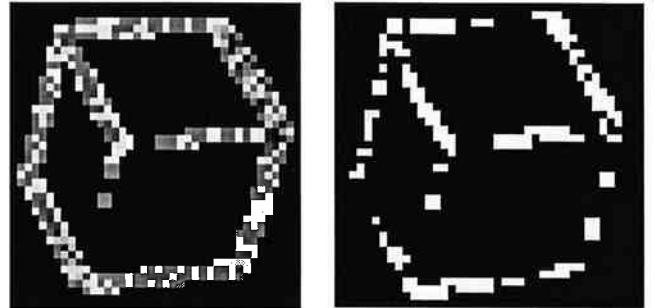


Figure 4: (a) Motion is detected between 2 successive frames by block matching with the binary values; (b) Motion vectors are detected between 2 successive frames by block matching with the binary values after the priority decision. The bright blocks indicate that motion vectors are detected.

hardware implementation by transforming a gray-scale pixel(8 bits) into a binary edge (1 bit); the result of block-matching does not have a significant impact on the quality of the motion compensation<sup>[4]</sup>.

The fast block matching equation can be changed into a logic operation form:

$$ACC_k = \left( \sum_{i=1}^4 c_i \oplus p_i \right)_{even-phase} + \left( \sum_{j=5}^8 c_j \oplus p_j \right)_{odd-phase}$$

$$i = 1, 2, 3, 4; \quad j = 5, 6, 7, 8$$

Since 9 accumulators are used for block matching and motion vector detection, as shown in Fig. 5,

therefore  $k = 1, 2, 3, 4, 5, 6, 7, 8, 9$ ;  $\oplus$  is the XOR operator for binary information. Binary edge coming from the current frame is represented by "c"; Binary edge coming from the previous frame, represented by "p".

### 4 Decision with Priority

A local pixel parallel and global column parallel (LPGCP) processor<sup>[5]</sup> is proposed for the fast block

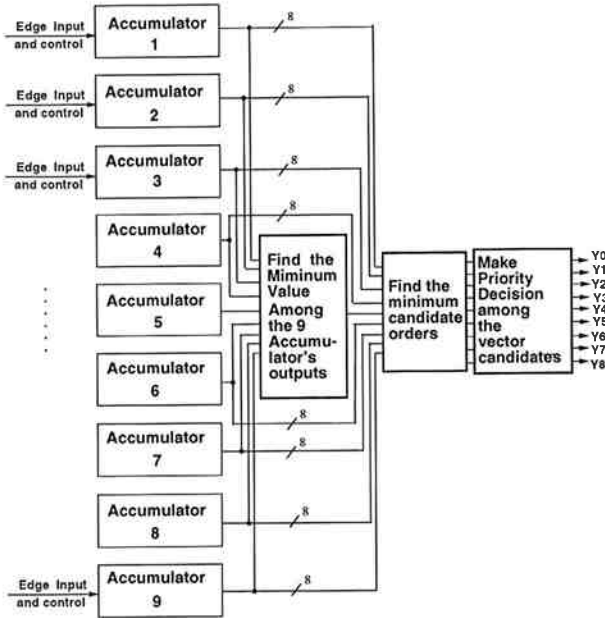


Figure 5: Block-matching and vector detection logic circuit. 9 block-matching operations are taking place at same time according to 9 groups of edge inputs for 9 ACCs .

matching and motion vector estimation.

There are 9 results coming out of every accumulator(ACC) as the vector candidates after the blockmatching operations, an example is given in Fig. 6. Among them, the most smallest one is chosen as the best vector.

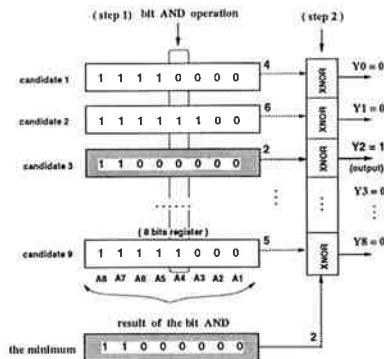


Figure 6: The operation of searching for the minimum and the best vector among 9 candidates.

When multiple minimum values come up at the same time, an arbitration logic<sup>[2]</sup> is needed to select the one as the final vector which has the smallest distance to the center of the reference block. When the distance is the same, we can name the output index order by ourselves. The arbitration logic can ensure a unique vector detected.

Table 1: summary of the prototype chip with the function of 2D motion vector detection on the focal plane.

number of pixels	16 × 16 pixels
size of PD (photo diode)	30.9 × 30.9 μm
size of PD+edge detector+mem	241 × 221 μm
size of the die chip	6499 × 8218 μm
size of block matching	2 × 2 pixels
size of search area	(±1, ±1) pixels
size of 1 LPGCP processor	1334.6 × 2924.4 μm
imaging frame rate	100 - 1000
number of trans.	PD & edge: 30 /pixel mem+control: 100 /pixel 1 LPGCP processor: 2486
fill factor	1.79%
power supply	single +5 V
power dissipation	90 mW / chip

## 5 Prototype Chip Design

By using 1-poly 2-metals 0.7μm CMOS process, a prototype chip for imaging, edge detection and motion vector detection is implemented.

In the prototype, only 4 LPGCP processors are adopted for the motion vector detection on the whole focal sensor plane by shifting the LPGCP processor window. The shifting window can locate on any one of 3 positions which covers the whole focal plane.

The chip with 4 LPGCP processors is shown in the left side of Fig. 7. The technical descriptions for it is summarized in Table 1.



Figure 7: The prototype chip. And the obtained images(middle and right) when "T" is moving up and down and projected on the focal plane.

## 6 Experiment and Preliminary Results

### 6.1 Imaging

When we projected a pattern, such as T with a movement of up and down, on the vision chip, we can obtain the recovered pictures shown in the middle and right of Fig. 7. Although the prototype has only 16 × 16 pixels, we can see the moving T in the images. Notice that the output is inverted in such a way that the brighter area is shown darker, and vice versa.

### 6.2 Estimate the Motion Vectors

Block matching operation is examined in detail using a test chip made together with the array. According to the LPGCP architecture, the function of motion vector detection needs 4 signal lines for the edges in the previous frame and 2 signal lines for the edges in the current frame. All the input signals in Fig. 8 and in Fig. 9(a) are represented by 5 groups, i.e.

{Pre - 4th - h1, Pre - 3rd - h1, Pre - 2nd - h1, Pre - 1st - h1} in Fig. 8(a),

{Pre - 4th - h2, Pre - 3rd - h2, Pre - 2nd - h2, Pre - 1st - h2} in Fig. 8(b),

{Pre - 4th - h3, Pre - 3rd - h3, Pre - 2nd - h3, Pre - 1st - h3} in Fig. 8(c),

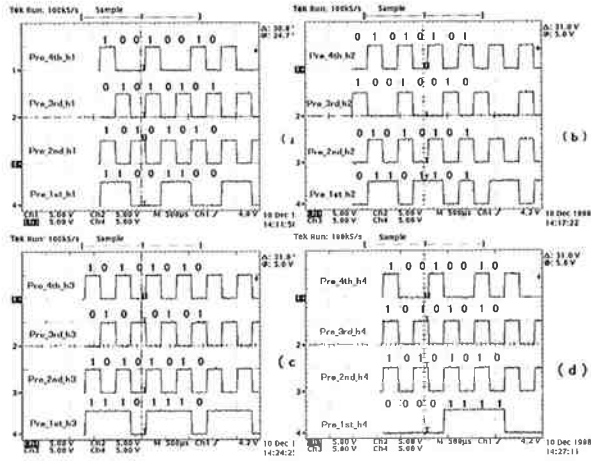


Figure 8: These 4 figure are the inputs of the previous frame edge image, which is combined together with the inputs of the current frame edge image to perform block matching.

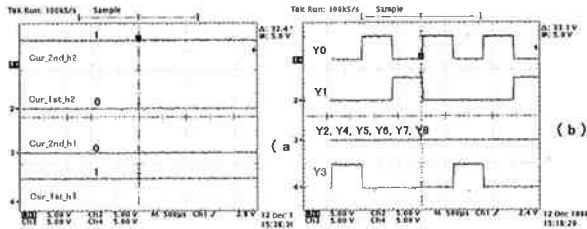


Figure 9: The input signals and evaluation results for the motion vector detection. Figure(a) is an input of the current frame edge image which is constant in this experiment. In finger(b), the motion vectors are detected and output by 9 output lines from Y0 to Y8. In this test, the results are { Y3, Y0, Y1, Y0, Y3, . . . }.

{Pre - 4th - h4, Pre - 3rd - h4, Pre - 2nd - h4, Pre - 1st - h4} in Fig. 8(d),  
 {Cur - 2nd - h1, Cur - 1st - h1, Cur - 2nd - h2, Cur - 1st - h2} in Fig. 9(a).

Among the 4 signal lines for the edges of previous frame, if one of the signal lines has the bit stream of {10010010}, the horizontal edge stream is its odd position's digit {1001} and the vertical edge stream is its even position's digit {0100}, which is illustrated in Fig. 8(a). The other inputs in Fig. 8 (b)(c)(d) have a similar arrangement for the bit streams.

Fig. 9(a) is an input of edges in the current frame, i.e. {1, 0, 0, 1} which is constant in this experiment. Fig. 9(b) shows the result of the motion vector detection. According to the priority decision rule, Y3 is firstly detected, and then Y0, Y1, Y0, Y3, etc.

The proposed method intends to detect a motion vector for every pixel and results in a motion field of vectors.

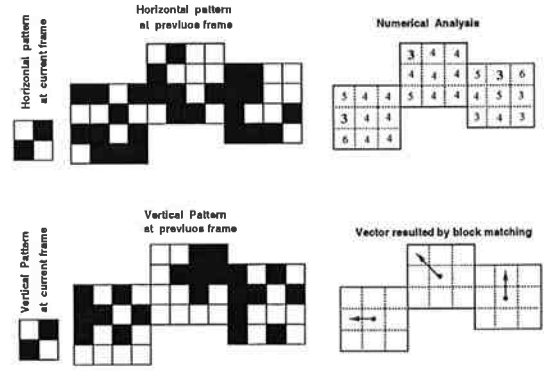


Figure 10: The theoretical analysis is consistent to the experiment test in Fig. 8 and Fig. 9. When tracing with the motion vectors, we can get an equivalent long vector with any possible degree of angles.

The experimental results in Fig. 8 and Fig. 9 are identical to the analysis by using of the edge images in Fig. 10. The 1st search indicates Y3 (index<sup>[2],[5]</sup> No. 2), the 2nd indicates Y0 (index No. 5), and the 3rd Y1 (index No. 1), and so on.

## 7 Conclusion

In this paper, on  $16 \times 16$  pixels' CMOS image sensor plane, very fast detection circuit for 2D motion vectors is implemented. Time multiplexed edge detection (TMED) scheme and 4 local pixel parallel and global column parallel (LPGCP) processors with a shifting window are applied on the focal plane. The proposed method intends to detect a motion vector for every pixel and results in a motion field of vectors. The experiment shows that the motion vectors detected by our prototype detection circuit are correct and stable.

## References

- [1] H.Gharavi and M. Mills, "Block-matching Motion Estimation Algorithms-New Results," *IEEE Trans. on Circuits and Systems*, Vol.37, No. 5, page 649-651, May 1990
- [2] Zheng Li, Kiyoharu Aizawa, Mitsutoshi Hatori, "Motion Vector Estimation on Focal Sensor Plane by Block Matching", *Proceedings of SPIE* Vol. 3410, page 215-221, 1998
- [3] B. Liu, A. Zaccarin, *New Fast Algorithms for the Estimation of Block motion Vectors*, page 148-157. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 3, No.2, April 1993
- [4] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-Complexity Block-based Motion Estimation via One-bit Transforms", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, No.4, pp.702-706, Aug., 1997
- [5] Zheng Li, "Focal plane processing for very fast detection of motion vectors", Ph.D. thesis at the University of Tokyo. March. 1999