Extended Abstracts of the 2007 International Conference on Solid State Devices and Materials, Tsukuba, 2007, pp. 1062-1063

G-9-3

A Moving-Object-Localization Hardware Algorithm Employing OR-Amplification of Pixel Activities

Yusuke Niki, Yasuo Manzawa, Satoshi Kametani and Tadashi Shibata

Department of Frontier Informatics, School of Frontier Sciences, The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8651, Japan
Phone: +81-4-7136-3853 Fax: +81-4-7136-3855
E-mail: {niki, manzawa, kametani}@else.k.u-tokyo.ac.jp, shibata@ee.t.u-tokyo.ac.jp

1. Introduction

Analyzing the movement of objects in a scene is quite essential in video surveillance, robot control, gesture understanding and so forth. The optical flow is most frequently used for motion analysis [1], but its computation is very expensive. To achieve real-time performance of the system, hardware implementation of optical flow processing has been studied very extensively [2].

In order to understand the motion of objects, we must, first of all, find out moving objects in the scene and localize them. If the subject is limited to only such moving object localization, there is an opportunity to implement the algorithm directly in a very simple circuitry. The frame difference in the pixel intensity can be utilized as the sign of motion, which is very compatible to implementation in the focal plane processing architecture. The technique was applied to motion picture compression by identifying the area of motion and sending the pixel data only in the region [3]. The purpose of this study is to develop a hardware algorithm of detecting and localizing moving objects based on the pixel activity in which a new adaptive threshold determination method has been explored.

The most difficult issue in this approach is the setting of the threshold in the pixel activity to decide whether the motion exists or not as illustrated in Fig. 1. If the threshold is too low, so many noises are detected from the background, while the moving object is almost disappearing when the threshold is too high. How to determine the threshold adaptively to the environmental conditions is of critical importance [4]. We have developed the concept of "OR-amplification," in which the activity flags from moving object is differentiated from those due to other origins by dynamically observing the appearance of the activity flags as a function of the threshold value. After presenting the basic algorithm, the design of prototype chip and the results of the circuit simulation are presented.

Fig. 1 Importance of adaptive threshold setting: original image (a); frame difference images after binarization with low threshold (b) and high threshold (c).

2. Moving Object Localization Algorithm

The proposed algorithm for moving object localization is illustrated in Fig. 2. Firstly, two images are taken with a time interval Δt, and the differential image is obtained by taking the frame difference in the pixel intensities. Then, the frame difference image is binarized using the threshold adaptively determined. Then, the binarized pixel data (pixel activities) are projected onto x- and y- axes and activity histograms are generated. Finally, a moving object is localized by binaraizing the activity histograms with fixed thresholds.

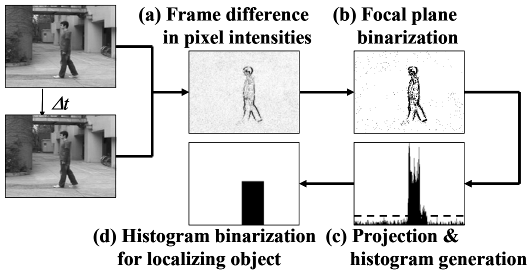

Fig. 2 Proposed algorithm for moving object localization.

The key issue in this algorithm is how to determine the threshold for binarizing the frame difference image. When a differential image is binarized at high threshold, the pixel activity bits are concentrated mostly in the area of the moving object (Fig. 3). If the pixel activity bits are dilated by taking OR with their nearest neighbors, the total number of bit "1" increases. However, the increment in sum of bit "1" is small at such high threshold values because the increment of bit "1" due to dilation mostly occurs only at the object contour. When the binarization is carried out at near noise levels, the activity bits occur not only in the moving object area but also in the area of background. However, in the region of background, the pixel activities are sparsely distributed. Therefore, the increment in sum of bit "1" by dilation becomes very large as illustrated in Fig. 3. Consequently, by observing the variation in the increment in sum of bit "1" with varying threshold, it is possible to detect the noise level. We produce the plot of increment in sum of bit "1" vs. threshold, and the optimum threshold is determined as the value yielding the 10% of the peak value in the plot.

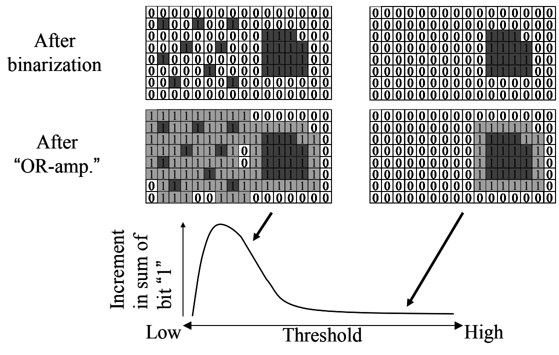

Fig. 3 "OR-amplification" for highlighting the background noises.

The determination of the time interval Δt is also essential in this algorithm. We observe the sum of activity bits in every frame while fixing the reference frame and Δt is determined so that the sum reaches a certain threshold, thus making the frame difference image intensity approximately constant.

The computer simulation results are shown in Fig. 4. In Fig. 4(a), a walking person is successfully enclosed, but stripes and gaps appear due to error bits occurring in the histogram binarization. A majority voting filter is introduced, in which it is decided if each location x (or y) belongs to the object region or not based on the majority bit within a certain window. A window size of 32 pixels with sampling at every 16 pixels has been employed and the results demonstrating the successful enclosure are shown in Fig. 4(b).

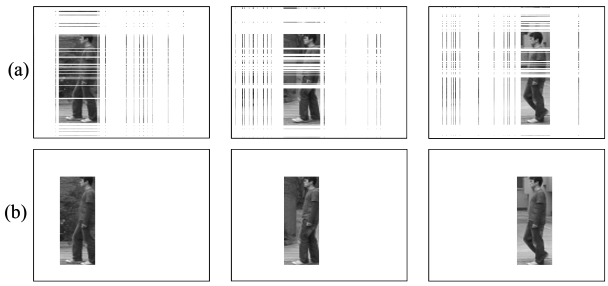

Fig. 4 Computer simulation results of enclosing a walking person before (a) and after (b) the majority voting filtering.

3. Hardware Architecture

Image acquisition, frame difference computation, focal plane binarization, and dilation are all carried out at each pixel site, and other processing is performed in the peripheral circuits. The configuration of the pixel circuits is shown in Fig. 5, which consists of a photo diode, a sample and hold circuit, a bump circuit for frame difference computation, and a comparator. The bump circuit calculates the frame difference of pixel intensities utilizing the short cut current in a CMOS inverter, whose *I-V* characteristic (SPICE simulation) is given in the insert of Fig. 5. The circuit utilized for calculating the increment in sum of bit "1" is shown in Fig. 6(c).

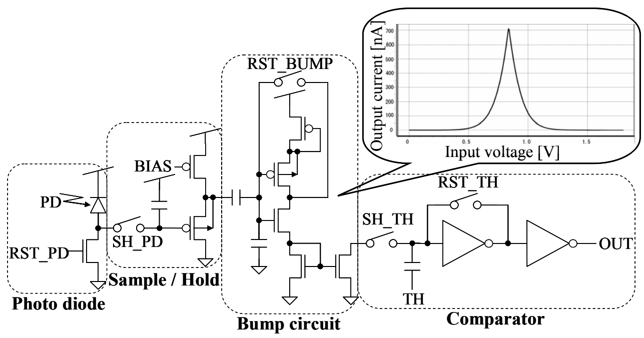

Fig. 5 Configuration of pixel circuitry.

A proof-of-concept test chip having an 8 x 8 pixel array was designed in a 0.18μm single-poly quintuple-metal CMOS technology as shown in Fig. 6. The threshold-voltage generator and the circuit for sum of bit "1" increment calculation are placed at the lower part of the test chip in Fig. 6(a). Final object localization circuits including the majority voting filter have not yet been implemented. However, it is easy to implement using floating-gate MOS technology. The cell size for one pixel is 58.80μm x 43.38μm as shown in Fig. 6(b). Instead of integrating photo diodes on the test chip, PMOS current sources are provided for verifying circuit operation.

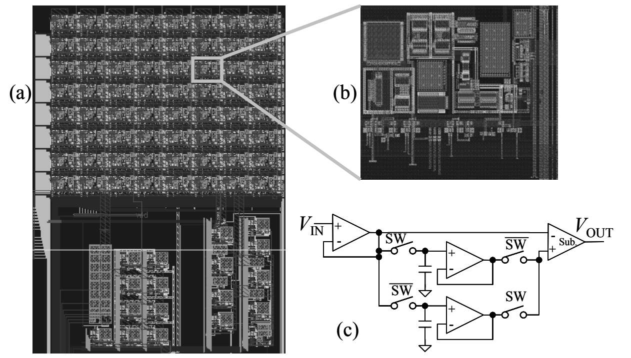

Fig. 6 Test chip layout: (a) 8 x 8 pixel array; (b) pixel circuit; (c) circuit for sum of bit "1" increment calculation.

The SPICE simulation results of entire test chip operation are demonstrated in Fig. 7. Fig. 7(a) and (b) are inputs of 8 x 8 images. Fig. 7(c) is differential image of Fig. 7(a) and (b). Fig. 7(d) is binary image produced by the adaptively determined threshold. The increment in sum of bit "1" vs. threshold is shown in Fig. 7(e).

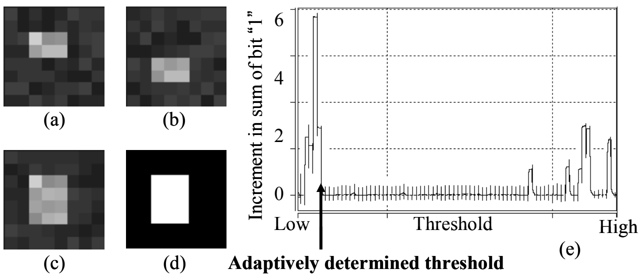

Fig. 7 SPICE simulation results: (a) input image #1; (b) input image #2; (c) differential image of #1 & #2; (d) binary image; (e) plot of increment in sum of bit "1" vs. threshold.

The measurement results have not been available yet, for the chip is under fabrication, but we hope they are ready at the time of conference.

Acknowledgements

The VLSI chip in this paper has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), The University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation.

References

[1] B. K. P. Horn and B. G. Schunck, *Artif. Intell.* **17** (1981) 185.

[2] S. Mehta and R. Etienne-Cummings, *IEEE Trans. Circuits Syst. I* **53** (2006) 1223.

[3] K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori, M. Abe, H. Maruyama and H. Otake, *IEEE Trans. Electoron Devices* **44** (1997) 1724.

[4] P. L. Rosin, *Comput. Vision Image Understand.* **86** (2002) 79.