TRANSACTIONS ON ELECTRICAL AND ELECTRONIC MATERIALS
Vol. 18, No. 6, pp. 320-322, December 25, 2017

Regular Paper

pISSN: 1229-7607 eISSN: 2092-7592
DOI: https://doi.org/10.4313/TEEM.2017.18.6.320
OAK Central: http://central.oak.go.kr

High-Performance and Low-Complexity Image Pre-Processing Method Based on Gradient-Vector Characteristics and Hardware-Block Sharing

Woo Suk Kim
Electrical, Electronic, and Control Engineering, Han-Kyong University, Anseong 17579, Korea

Juseong Lee
Center of Human-centered Interaction for Coexistence, Seoul 02792, Korea

Ho-Myoung An
Department of Electronics, Osan University, Osan 18119, Korea

Jooyeon Kim[†]
School of Electrical Electronics Engineering, Ulsan College, Ulsan 44610, Korea

Received July 12, 2017; Revised August 20, 2017; Accepted August 23, 2017

In this paper, a high-performance, low-area gradient-magnitude calculator architecture is proposed, based on approximate image processing. To reduce the computational complexity of the gradient-magnitude calculation, vector properties, the symmetry axis, and common terms were applied in a hardware-resource–shared architec-ture. The proposed gradient-magnitude calculator was implemented using an Altera Cyclone IV FPGA (EP4CE115F29) and the Quartus II v.16 device software. It satisfied the output-data quality while reducing the logic elements by 23% and the embedded multipliers by 76%, compared with previous work.

Keywords : Edge map generation, Approximate image signal processing, Low-complexity hardware architecture

1. INTRODUCTION

Recently, a low-power, high-performance vision chip has received attention owing to its high-resolution display and image-sensor technology [1]. Following this trend, low-complexity very-large-scale integration (VLSI) Canny edge detection has been studied as an image-pre-processing algorithm for object tracking, recognition, and similar applications [2,3]. For the hardware implementation of a low-complexity Canny edge detector, previous works have concentrated on reducing the computational complexity of the gradient-magnitude calculator module.

The gradient magnitude is one of the fundamental blocks of an image because it represents a directional change in the intensity or color. The formula for the gradient magnitude consists of complex operations, e.g., square roots, multiplication, and addition. However, approximate image signal processing has been adopted by many related works [4–6].

S.-L. Chen et al. [4] proposed an absolute-value summation of the horizontal and vertical gradients. Because the diagonal magnitude was not considered, the data quality was quite degraded . To obtain reasonable data quality, a diagonal Sobel filter architecture [5] and a piecewise planar approximation [6] have been proposed to consider the diagonal directions. However, the gradient-magnitude quality for

[†] Author to whom all correspondence should be addressed:
E-mail: joo@uc.ac.kr

high-level computer-vision applications is still insufficient.

In this paper, a high-performance gradient-magnitude hardware architecture is proposed, using an approximate image-processing technique. The proposed algorithm reduces the complexity of the gradient-magnitude calculator while preserving the image-data accuracy. In the proposed method, the gradient magnitude can be efficiently approximated, based on vector projections to pre-determined directions on a two-dimensional vector plane. Moreover, using the symmetric characteristic of two-dimensional vectors, a folding architecture is applied to increase the energy efficiency.

2. MAIN CONCEPT

The proposed gradient-magnitude calculator is based on selecting the maximum value from among five specific orthogonal projection vectors. The original equation for the gradient-magnitude (G) calculation is as follows [7]:

$$G = \sqrt{{G_X}^2 + {G_Y}^2} \quad (1)$$

where G_X and G_Y are the horizontal and vertical image gradients, respectively. Owing to the high computational complexity of (1), an approximate signal processing technique is needed that also satisfieg the output data quality.

The gradient-magnitude calculation can be simplified to eight sectors, considering the eight neighboring pixels in the digital image format. In addition, the gradient magnitude has a unit-vector characteristic. With these properties, (1) can be approximated to three directions (0°, 45°, 90°). However, border lines can be ambiguous, resulting in degraded output quality; therefore, we consider two additional directions (22.5°, 67.5°). The calculation of the proposed gradient magnitude follows:

$$G = \max(|G_X| \times \cos(i) + |G_Y| \times \sin(i)) \quad (2)$$

where $i = 0°$ to $90°$ (22.5° per step).

The gradient magnitude can be efficiently approximated using (2). For 0° or 90°, G_X or G_Y can directly substitute for the projection-vector calculation. In addition, because cos (45°) and sin (45°) have the same value, one of the multipliers can be removed by the associative law. Figure 1(a) shows the characteristics of projection vectors. For $|G_X| \geq |G_Y|$, we can remove tw- vector directions (67.5°, 90°) because the 45° direction is the symmetry axis. For $|G_X| < |G_Y$d, the 0°, 22.5°, and 45° vector directions can bedomitted, as shown in Fig. 1(b).

Figure 2 shows the projection vector calculation process for 22.5°and 67.5°. As shown in the figure, constant values (0.92 and 0.38) overlad for calculating each projection vectos. In this regard, two multipliers (Mult.) and an adder (Add.) can be shared using a multiplexer (MUX) with the symmetry-axis property. The hardware architecture for the proposed gradient-magnitude calculator is presented in Fig. 3. With the proposed hardware architecture, the computational complexity of the gradient-magnitude calculation is significantly reduced.

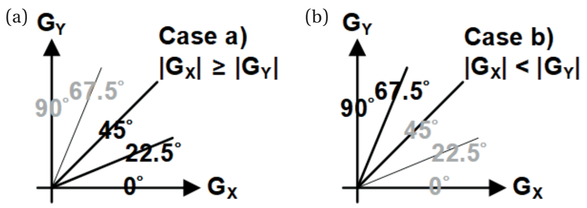

Fig. 1. Projection-vector characteristics. (a) $|G_X| \geq |M_Y|$ and (b) $|M_X| < |M_Y|$.

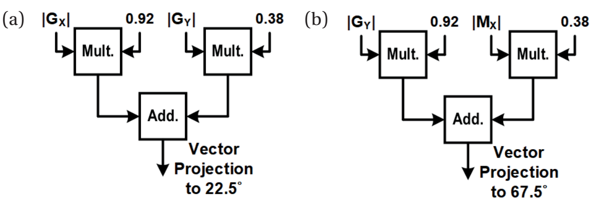

Fig. 2. Projection-vector calculation process. (a) Vector projection for 22.5° and (b) vector projection for 67.5°.

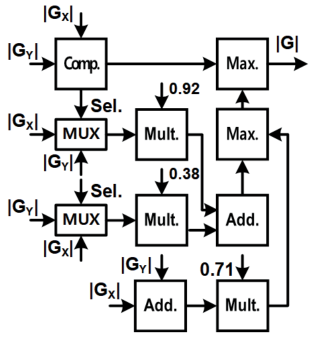

Fig. 3. Hardware architecture for the proposed gradient-magnitude calculator.

3. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental results of the proposed Canny edge detector using our hardware implementation. Figure 4 shows the edge-map results of the Canny edge detection. Figure 4(a) is the original rail image, Fig. 4(b) shows the edge-detection result based on the original Canny edge-detection algorithm, and Fig. 4(c) shows the result of the proposed Canny edge detector. As shown in

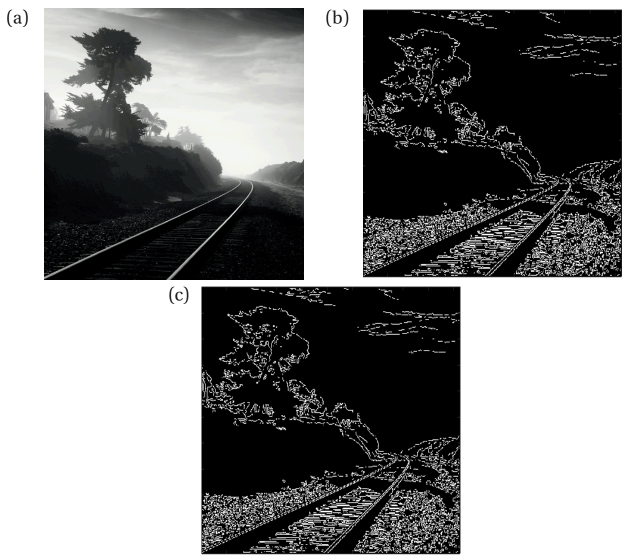

Fig. 4. Canny edge-detection results. (a) Original rail image, (b) canny edge-detection result of original algorithm, and (c) canny edge-detection result of proposed approach.

Table 1. FPGA implementation results.

	Conventional [7]	This work
Registers	32	32
Logic elements	179	137
Embedded multipliers	13	3

the figure, the proposed algorithm can successfully approximate the original output-data accuracy.

The proposed high-performance, low-area gradient magnitude calculator was implemented using an Altera FPGA (field-programmable gate array, Cyclone IV, EP4CE115F29C7N) and the Quartus II v.16 device software . Table 1 lists the synthesis results of the FPGA (field-programmable gate array) implementation. The proposed hardware architecture reduces the logic elements by 23% and the embedded multipliers by 76%, compared to the conventional Canny edge detector [7].

4. CONCLUSIONS

In this paper, a high-performance and low-area gradient-magnitude calculator architecture was proposed. We presented a low computational-complexity algorithm and a hardware architecture based on a maximum-value selection among pre-defined projection vectors. The experimental results of the implementation showed that the proposed approach reduced hardware area while also satisfying the output-data quality of the original Canny edge detection. Our hardware-design approach can be used to construct high-level computer-vision applications.

ACKNOWLEDGMENT

This work was supported by the 2017 Research Fund of Ulsan College.

REFERENCES

[1] A. Alaghi, C. Li, and J. P. Hayes, *Proc. DAC '13 the 50th Annual Design Automation Conference* (ACM, Austin, USA, 2013) p. 1. [DOI: https://doi.org/10.1145/2463209.2488901]

[2] P. Zhao, H. Zhu, H. Li, and T. Shibata, *IEEE Trans. Circuits Syst. Video Technol.*, **23**, 503 (2013). [DOI: https://doi.org/10.1109/TCSVT.2012.2210665]

[3] S. L. Chen and E. D. Ma, *IEEE Trans. Circuits Syst. Video Technol.*, **24**, 1982 (2014). [DOI: https://doi.org/10.1109/TCSVT.2014.2317890]

[4] P. R. Possa, S. A. Mahmoudi, N. Harb, C. Valderrama, and P. Manneback, *IEEE Trans. Comput.*, **63**, 2376 (2014). [DOI: https://doi.org/10.1109/TC.2013.130]

[5] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, *IEEE J. Solid-State Circuits*, **23**, 358 (1988). [DOI: https://doi.org/10.1109/4.996]

[6] B. Geelen, F. Deboeverie, and P. Veelaert, *Proc. 2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* (IEEE, Como, Italy, 2009) p. 1. [DOI: https://doi.org/10.1109/ICDSC.2009.5289349]

[7] J. Canny, *IEEE Trans. Pattern Anal. Mach. Intell.*, **PAMI-8**, 679 (1986). [DOI: https://doi.org/10.1109/TPAMI.1986.4767851]