

히스토그램 기반 Otsu 알고리즘을 활용한 엣지 검출 하드웨어 가속기

강주완¹, 정인수¹, 조승준¹, 문병인²

¹ 경북대학교 대학원 전자전기공학부 석사과정

² 경북대학교 전자공학부/대학원 전자전기공학부 교수

jkwang1231@knu.ac.kr, insu8944@knu.ac.kr, jo73012@knu.ac.kr, bihmoon@knu.ac.kr

An Edge Detection Hardware Accelerator Using a Histogram-Based Otsu Algorithm

Joowan Kang¹, Insu Jeong¹, Seungjun Jo¹, Byungin Moon^{1,2}

¹Graduate School of Electronic and Electrical Engineering, Kyungpook National University

²School of Electronics Engineering, Kyungpook National University

요 약

Otsu 알고리즘은 픽셀 값의 분포를 분석해 이미지를 두 개의 클래스로 나누는 최적의 임계값을 산출하는 기법으로, 주로 의료 영상 처리, 산업 자동화 등 다양한 분야에서 객체를 구분 지을 때 활용되고 있다. 이미지마다 적합한 임계값을 계산하는 Otsu 알고리즘을 통해 객체 구분에 널리 활용되는 엣지 검출 가속기의 정확도 또한 향상시킬 수 있다. 그러나 Otsu 알고리즘은 복잡한 연산으로 인해 하드웨어 사용량이 제한된 조건에서 구현하기 위해서는 효율적인 설계가 필요하다. 이에 본 논문에서는 Otsu 알고리즘 연산을 하드웨어에 효율적인 방식으로 구현한 히스토그램 기반 연산을 활용한 엣지 검출 하드웨어 가속기를 제안한다.

1. 서론

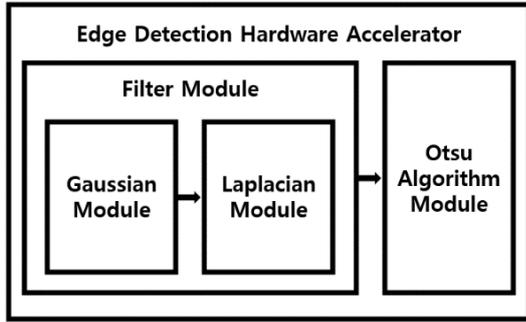
엣지 검출은 CT 및 MRI 와 같은 의료 영상 분석뿐만 아니라 산업 자동화와 같은 다양한 분야에서 중요한 역할을 하고 있다[1]. 엣지 검출은 적절한 임계값으로 인해 경계가 아닌 영역이 엣지로 잘못 인식되는 문제가 발생한다. 이러한 문제를 극복하여 정확도를 향상시키기 위한 방법 중 하나는 Otsu 알고리즘을 통해 이미지마다 적절한 임계값을 산출하는 것이다[2]. Otsu 알고리즘은 이미지가 변환됨에 따라 해당 이미지를 두 개의 클래스로 분리하기 위한 최적의 임계값을 계산하는 알고리즘이다[3]. 하지만 복잡한 연산으로 이루어진 Otsu 알고리즘을 엣지 검출 가속기에 구현하기 위해서는 Otsu 알고리즘 연산의 효율적인 설계가 필요하다. Otsu 알고리즘을 하드웨어로 구현할 때 대부분은 로그 근사화 기법이 사용되며 [4][5] 연산 속도와 정확도 면에서 이점을 제공한다. 하지만 세분화된 값을 처리할 때 LUT(Look-Up Table) 사용량이 증가해 하드웨어 자원 소모가 커지는 단점이 있다. 이에 본 논문은 Otsu 알고리즘을 간소화한

식(1)[6]을 기반으로 반복 연산을 통해 하드웨어 사용량을 줄인 히스토그램 기반 연산의 엣지 검출 가속기를 제안한다. 또한 이를 로그 근사화 기반의 엣지 검출 가속기와 비교 분석하였다.

$$\sigma_B^2 = (\sum_{i=0}^t n_i) * (\sum_{i=t+1}^{L-1} n_i) * \left(\frac{\sum_{i=0}^t i n_i}{\sum_{i=0}^t n_i} - \frac{\sum_{i=t+1}^{L-1} i n_i}{\sum_{i=t+1}^{L-1} n_i} \right)^2 \quad (1)$$

2. 히스토그램 기반 연산 방식의 Otsu 알고리즘 하드웨어 가속기 설계 및 분석

본 논문에서는 그림 1 과 같이 가우시안 필터, 라플라시안 필터, 그리고 Otsu 알고리즘을 입력 이미지에 순차적으로 적용하는 엣지 검출 하드웨어 가속기를 설계하였다. 가우시안 필터는 이미지의 잡음을 제거하여 엣지 검출의 효율성을 높여주며, 라플라시안 필터는 이미지의 엣지를 강조하며 이미지의 픽셀 값은 전체적으로 감소시킨다. Filter Module 에서 출력된 이미지는 엣지 영역이 엣지가 아닌 영역에 비해 픽셀 값이 상대적으로 크며, 이러한 픽셀 값의 차이를 활용해 Otsu 알고리즘을 적용하여 두 영역을 분리



(그림 1) 엣지 검출 하드웨어 가속기 블록도

하는 최적의 임계값을 계산한다. 그 후 계산된 임계값을 기반으로 이진화를 수행하여 최종적으로 엣지 검출된 이미지를 출력한다. Otsu 알고리즘을 간소화한 식(1)에서는 σ^2 값이 최대일 때 임계값 t 에 의해 나누어지는 두 픽셀 분포 그룹의 분산이 최소가 되며, 이때의 t 가 최적의 임계값을 의미한다. $\frac{\sum_{i=0}^t m_i}{\sum_{i=0}^t n_i}$ 와 $\frac{\sum_{i=t+1}^{L-1} m_i}{\sum_{i=t+1}^{L-1} n_i}$ 는 입력 이미지에 대한 픽셀 분포 그룹을 임계값 t 기준으로 두 영역으로 나누었을 때, 각각의 평균 픽셀 값을 의미한다. 이때 평균값 계산에서, 각 영역의 가장 작은 픽셀 값을 가진 픽셀의 개수와 가장 큰 픽셀 값을 가진 픽셀의 개수를 순차적으로 뺄셈하여 평균값을 도출하는 히스토그램 기반 연산을 적용하였다.

보편적으로 쓰이는 로그 근사화 기법을 활용한 엣지 검출 가속기와 본 논문에서 제안하는 히스토그램을 활용한 엣지 검출 가속기는 HDL 로 작성하였으며, Xilinx 사의 xc7z020clg484-1 chip 에 synthesis 를 수행한 결과는 표 1 과 같다.

로그 근사화 기법을 활용한 엣지 검출 하드웨어 가속기는 정밀도에 비례해 하드웨어 사용량이 증가하였다. 히스토그램 기반의 엣지 검출 가속기는 연산 시간이 다소 길어지며 연산 오차율이 증가했지만, 로그 근사화 기법 엣지 검출 가속기와 비교했을 때 하드웨어 사용량은 대폭 감소하였다.

<표 1> 연산 기법에 따라 설계된 엣지 검출 가속기 하드웨어 사용량 및 성능 비교

	로그 근사화 기법을 활용한 엣지 검출 하드웨어 가속기 (high precision)	로그 근사화 기법을 활용한 엣지 검출 하드웨어 가속기 (precision match)	히스토그램을 활용한 엣지 검출 하드웨어 가속기
Slice LUT 개수	42,674	27,068	19,993
Slice Register 개수	29,053	28,835	17,578
평균 Latency (ms)	2.085	2.085	2.139
연산 오차율 (%)	0.39	2.47	2.60

3. 결론

본 연구에서는 Otsu 알고리즘을 히스토그램 기반의 연산 방식을 사용한 새로운 구조의 엣지 검출 가속기를 설계했으며, 로그 근사화 기법을 활용한 엣지 검출 가속기와 비교 분석하였다. 로그 근사화 기법 기반의 설계는 빠른 처리 속도와 높은 정확도를 요구하는 환경, 히스토그램 기반의 설계는 하드웨어 자원이 제한된 환경에서 효율적인 대안을 제공한다는 것을 확인하였다. 이러한 결과는 실시간 응용 분야에서 엣지 검출을 최적화하기 위한 연구에 기여할 것이다.

ACKNOWLEDGMENT

본 연구는 IDEC 에서 EDA Tool 을 지원받아 수행하였습니다.

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단-시스템반도체융합전문인력육성사업의 지원을 받아 수행된 연구임(2020M3H2A107804521)

참고문헌

- [1] G. Wang, C. Lopez-Molina and B. De Baets, "Automated blob detection using iterative laplacian of gaussian filtering and unilateral second-order gaussian kernels," *Digital Signal Processing*, vol. 96, pp. 102592, 2020.
- [2] A. M. A. Talab, Z. Huang, F. Xi and L. HaiMing, "Detection crack in image using OTSU method and multiple filtering in image processing techniques," *Optik*, vol. 127, no. 3, pp. 1030-1033, 2016.
- [3] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.
- [4] J. G. Pandey, A. Karmakar, C. Shekhar and S. Gurunaranayan, "A Novel Architecture for FPGA Implementation of Otsu's Global Automatic Image Thresholding Algorithm," *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, Mumbai, India, 2014, pp. 300-305
- [5] H. Tian, S. K. Lam and T. Srikanthan, "Implementing Otsu's thresholding process using area-time efficient logarithmic approximation unit," *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, Bangkok, Thailand, 2003, pp. IV-IV
- [6] Wang Jianlai, Yang Chunling, Zhu Min and Wang Changhui, "Implementation of Otsu's thresholding process based on FPGA," *2009 4th IEEE Conference on Industrial Electronics and Applications*, Xi'an, China, 2009, pp. 479-483