

블록 유형 분류 알고리즘 기반 고속 특징추출 시스템 구현에 관한 연구

이주성*, 안호명**

A Study on Implementation of the High Speed Feature Extraction System Based on Block Type Classification

Juseong Lee* and Ho-Myoung An**

요약 본 논문은 고속 특징추출 알고리즘의 구현 방법을 제안한다. 제안하는 방법은 블록 유형 분류 알고리즘을 기반으로, 블록 유형 분류 알고리즘 적용 시, 영상 특징 정보가 발생하지 않는 스무스 블록에서 연산을 생략하여 영상 특징 검출에 필요한 연산시간을 감소시킬 수 있다. 200장의 표준 테스트 이미지를 활용해 매크로 블록의 크기를 64×64 로 나누어 스무스 블록의 발생 빈도를 측정할 결과 전체의 29.5%만큼 발생하는 것을 정량적으로 확인했다. 이 의미는 다양한 영상 정보를 포함하고 있는 표준 테스트 이미지 내에서는 29.5%에 해당하는 만큼 연산의 복잡도를 감소시킬 수 있다는 의미를 나타낸다. 제안된 방법을 케니 윤곽선 검출 알고리즘에 적용하면 이차원 미분 필터, 그라디언트 크기 및 방향 연산, 비최대 억제, 적응형 임계값 연산, 히스테리시스 임계 처리와 같은 총 다섯 단계의 영상처리에 필요한 지연시간을 완전히 제거할 수 있다. 이와 같은 방법으로 다양한 특징 검출 알고리즘에 블록 유형 구분 알고리즘을 적용해, 연산에 필요한 시간을 감소할 수 있을 것을 기대한다.

Abstract In this paper, we propose a implementation approach of the high-speed feature extraction algorithm. The proposed method is based on the block type classification algorithm which reduces the computation time when target macro block is divided to smooth block type that has no image features. It is quantitatively identified that occurs at 29.5% of the total image using 200 standard test images with 64×64 macro block size. This means that within a standard test image containing various image information, 29.5% can reduce the complexity of the operation. When the proposed approach is applied to the Canny edge detection, the required latency of the edge detection can be completely eliminated, such as 2D derivative filter, gradient magnitude/direction computation, non-maximal suppression, adaptive threshold calculation, hysteresis thresholding. Also, it is expected that operation time of the feature detection can be reduced by applying block type classification algorithm to various feature extraction algorithms in this way.

Key Words : Block type classification, Canny edge detection, feature detection, high performance signal processing, low-power image processing

1. 서론

패턴인식 알고리즘은 다양한 시스템에 적용되는 영상처리 과정이다. 최근 자율주행 자동차 관련 연구가 활발하게 진행되면서 고성능 센서, 영상처리 모듈 등

컴퓨터 비전을 기반으로 하는 물체 인식 관련 연구가 주목받고 있다. 실시간 물체 인식 알고리즘은 차선이탈 경고, 제한속도 인식, 전방 추돌경고, 자동긴급제동 등 비전 기반 자율주행 자동차 시스템 구축에 필수적이다. 시스템의 실시간 동작 가능성을 증가시키기 위

This work was supported by the Academic Research Funding of Osan University in 2018.

*School of Electrical Engineering, Korea University, Korea.

**Corresponding Author : Department of Electronics, Osan University, Korea.(callname@osan.ac.kr)

Received February 28, 2019

Revised April 08, 2019

Accepted April 17, 2019

해 물체 인식 알고리즘에서 가장 높은 연산 복잡도를 갖는 특징추출 알고리즘의 최적화가 필수적이다 [1-4]. Canny edge detector [5]은 대표적인 특징추출 알고리즘이다 [6]. 기존 연구들에선 알고리즘 중 가장 복잡도가 높은 adaptive thresholding [7]

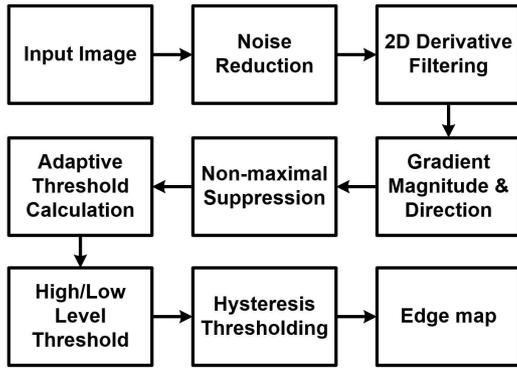


그림. 1. Canny edge detection 알고리즘의 블록 다이어그램.
Fig. 1. Block diagram of Canny edge detection algorithm.

중 가장 복잡도가 높은 adaptive thresholding [7]에 대한 연구가 집중적으로 수행되어왔다. 알고리즘은 영상에서 특징점이 될 수 있는 edge 정보를 추출하기 위한 기준값을 구하는 과정으로 구성되어 있고, histogram을 그려 그 안에서 기준이 될 수 있는 값을 찾아내는 과정 때문에 값 도출까지 상당한 지연시간이 요구된다. 이런 문제점을 해결하기 위해 최근 연구 [8]에선 block type classification [9] 기반의 distributed architecture 을 적용했다. 영상을 macro block 단위로 나눈 후, 병렬로 computing unit을 구축하여 연산 출력까지 걸리는 시간을 줄이는데 이바지했다 [8]. 하지만 구현 복잡도가 연산 속도 증가에 비례해 상승하는 단점이 있다.

본 논문에서는 기존 feature detection에 요구되는 연산 속도 및 복잡도 문제를 해결하기 위해 다수의 이미지를 포함하는 테스트 이미지 세트 [10]에서 발생하는 블록 유형의 수를 분석하여 결과를 도출해내는데 필요 없는 연산을 생략하도록 한다. 제안하는 방법은 다섯 개의 블록 유형 중 edge 정보가 포함되지 않

는 smooth 블록을 블록 유형 구분 알고리즘을 사용해 분류해 내고, 그 블록에 필요한 연산을 생략하여 연산에 필요한 지연시간을 절감할 수 있다.

2. 기존 연구

Canny edge detection [5]은 물체 및 배경의 윤곽을 가장 잘 찾아내는 방법의 한 종류이다. 알고리즘

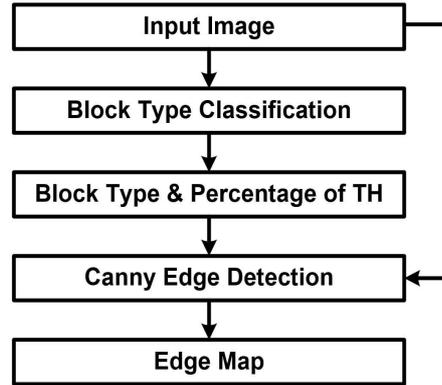


그림. 2. 블록 유형 구분 알고리즘 기반 Canny edge detection 수행과정.
Fig. 2. Process of Canny edge detection based on block type classification algorithm.

A. Pixel classification
 if $(\text{Var}(x,y) \leq T_u)$
 Pixel type: uniform
 else if $(T_u < \text{Var}(x,y) \leq T_e)$
 Pixel type: texture
 else if $(T_e < \text{Var}(x,y))$
 Pixel type: edge

B. Block classification

Block type	No. of pixel type	
	N_{uniform}	N_{edge}
Smooth	$\geq 0.3 \times TP$	0
Texture	$< 0.3 \times TP$	0
Edge/texture	$< 0.65 \times (TP - N_{\text{edge}})$	$(> 0) \ \& \ (< 0.3 \times TP)$
Medium edge	$\geq 0.65 \times (TP - N_{\text{edge}})$	$(> 0) \ \& \ (< 0.3 \times TP)$
Strong edge	$\leq 0.7 \times TP$	$\geq 0.3 \times TP$

$\text{Var}(x,y)$: the local 3×3 variance at pixel (x,y) ;
 T_u and T_e : two thresholds as in [7];
 TP: the total number of pixels in the block;
 N_{uniform} : the total number of uniform pixels in the block;
 N_{edge} : the total number of edge pixels in the block;

그림. 3. 블록 유형 구분 알고리즘의 의사 코드 [8].
Fig. 3. Pseudo-codes of the block type classification algorithm [8].

은 낮은 검출 에러율을 보이지만, 구현이 복잡하고 실행 시간이 다른 알고리즘들에 비해 상당히 긴 단점이 있다 [6]. 그림 1의 블록 다이어그램과 같이 Canny edge detection 알고리즘이 구성된다. 먼저 gray scale 영상이 입력되면, 영상에 포함된 noise를 제거하기 위해 Gaussian smoothing을 적용한다. 다음 단계로 one-dimensional direction과 norm 값을 구하기 위해 Sobel mask를 그림 2의 이용해 vertical 및 horizontal derivative filtering을 수행한다. 모든

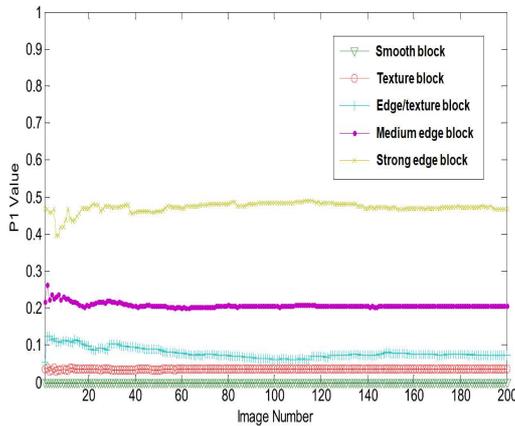


그림 4. 각 블록 유형의 high level threshold (P_1) 값의 백분율 [8].
Fig. 4. Percentage of high level threshold (P_1) values for each block type [8].

방향을 고려할 수 있는 gradient magnitude 및 direction을 연산하고, Non-maximum suppression을 통해 local maxima에 해당하는 candidate를 선정한다. Double level thresholding을 위한 high/low threshold 값을 구하고, hysteresis thresholding을 통해 edge를 tracking 해 정확한 윤곽 정보를 추출 해낸다. High/low threshold를 찾아내는 과정에서 이미지 전체에 대한 gradient magnitude를 도출하고, 미리 선정한 percentage에 부합하는 특정한 값을 찾아내기 위해 누적 히스토그램 생성하는 것이 요구된다. 계산된 gradient magnitude 값을 픽셀 값별로 카운트하여 누적 히스토그램을 생성하고, 전체 픽셀에 대한 누적 히스토그램이 생성될 동안 gradient magnitude 값은 메모리

에 저장되어야 한다. 하지만 영상 전체에 대한 gradient magnitude 값을 메모리에 저장하려면 복잡도가 상당히 높아지는 문제점이 있다.

그림 2는 블록 유형 구분 기반 Canny edge detection의 수행과정을 나타낸다. 블록 유형 분류 알고리즘은 [9] 영상 내에서 macro block으로 영상을 나누고, 영역마다 갖는 특징을 나타낼 수 있는 알고리즘이다. 블록 유형 구분 알고리즘은 그림 3과 같은 의사 코드에 의해 구성되어 있다 [8]. 알고리즘에서는 macro block의 특징에 따라 각각 smooth, texture, edge/texture, medium edge, strong edge 블록으로 분류한다. 알고리즘은 이미지를 미리 설정한 macro block 크기로 나누고, 3x3 영역마다 분산

```

Let  $P_1$  be the percentage of pixels, in a block, that would be
classified as strong edges.
Step 1: If smooth block type
         $P_1 = 0;$  /* No edges*/
    else if texture block type
         $P_1 = 0.03;$  /* Few edges*/
    else if texture/edge block type
         $P_1 = 0.1;$  /* Some edges*/
    else if medium edge block type
         $P_1 = 0.2;$  /* Medium edges*/
    else
         $P_1 = 0.4;$  /* Many edges*/
Step 2: Compute the 8-bin non-uniform gradient magnitude
        histogram and the corresponding cumulative distribution
        function  $F(G)$ .
Step 3: Compute High_threshold as  $F(\text{High\_threshold}) = 1 - P_1$ 
Step 4: Compute Low_threshold =  $0.4 * \text{High\_threshold}$ 
    
```

그림 5. 블록 유형 구분 기반 Canny edge detection의 High level threshold (P_1) 값 설정 과정 [8].
Fig. 5. Process of high level threshold value (P_1) for Canny edge detection based on block type classification [8].

값을 구한 후, 알고리즘에서 미리 정한 기준 값에 따라 분산 값이 100 이하이면 uniform 픽셀, 100 과 900 사이 값이면 texture 픽셀, 900 이상이면 edge 픽셀로 구분한다. 로컬 영역 내에서 세 가지 유형으로 분류된 각 픽셀 값의 총 개수를 카운트하고, 그 수에 따라 블록을 미리 설정된 다섯 가지의 블록 유형(Smooth, Texture, Edge/texture, Medium edge, Strong edge)으로 나누는 과정으로 구성된다 [8]. 이는 기존 연구에서 영상 압축과 특징추출 알고리즘의 복잡도를 줄이는데 적용됐다 [9]. 특히 블록 유형

구분 알고리즘은 그림 2에 나타난 것과 같이 Canny edge detection에 적용되어 연산 복잡도를 절감시킬 수 있다 [8]. 알고리즘은 macro block 단위의 각 블록의 블록 유형을 구분한 결과를 기반으로 Canny edge detection의 adaptive threshold calculation에 요구되는 누적 히스토그램에서의 백분율 값을 선정하는 데 사용됐다 [8]. 이는 Canny edge detection이 macro block 단위로 수행될 수 있도록 하고, 병렬 처리 될 수 있도록 설계되어 throughput을 증가시켰다. 하지만 알고리즘을 수행하는데 요구되는 연산 복잡도가 높아 여전히 최적화가 요구된다.

3. 본론

기존 연구에서는 특징추출 시스템에 주로 사용되는 Canny edge detection의 처리속도를 향상하기 위



그림. 6. Berkeley Segmentation Dataset [10]의 예시.
Fig. 6. Example of the Berkeley Segmentation Dataset [10].

해 블록 유형 구분 알고리즘 기반 Canny edge detection을 제안했다 [8]. 본 논문에서는 기존 연구에서 문제가 되는 높은 연산량 문제를 해결하기 위해 블록 유형 구분 알고리즘을 기반으로 하는 불필요한 연산 제거 기법을 제안한다. 그림 3은 기존 연구[8]에서 각 블록 유형별 high level threshold (P_1) 값의 백분율을 분석한 결과를 나타낸다. 그림의 결과와 같이

블록 유형별로 high level threshold 값의 백분율이 같은 블록 유형에서는 유사하게 나타나는 것을 확인할 수 있다 [8].

그림 5는 그림 4의 결과를 바탕으로 High level threshold 및 low level threshold를 결정하는 과정을 나타낸다 [8]. 이때 smooth block에서 high threshold 값이 1 (100%)로 설정되어, edge가 발생하지 않는 것을 확인할 수 있다. 이 점에 착안해 본 논문에서는 edge가 발생하지 않는 smooth 블록의 비율을 분석하고, 이에 따라 이후 과정에 포함되는 Canny edge detection 연산을 생략하는 방법을 제안한다. 제안하는 방법은 블록 유형 구분 알고리즘을 먼저 수행하여, 블록 유형이 smooth 블록일 경우 Canny edge detection 과정인 noise reduction, dimensional (2D) derivative filtering, gradient magnitude and direction, non-maximal suppression, hysteresis thresholding 과정을 생략할 수 있다.

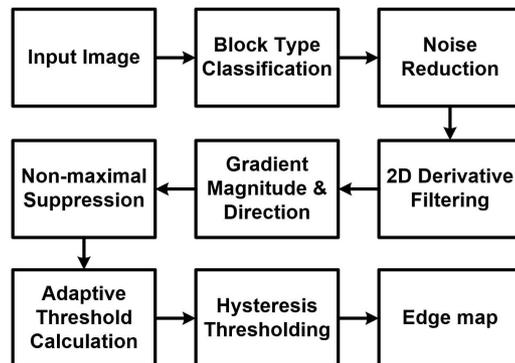


그림. 7. 제안된 블록 유형 구분 알고리즘 기반 고속 특징추출 시스템의 블록 다이어그램.
Fig. 7. Proposed high speed feature extraction system based on block type classification algorithm.

4. 실험 결과

블록 유형 구분 기반 영상 특징추출 알고리즘에서 feature detection을 위한 threshold 값을 누적 히스토그램을 기준으로 Smooth 블록의 경우 100%의 값, Texture 블록의 경우 97%의 값, Edge/texture 블록

의 경우 90%의 값, Medium edge 블록의 경우 80%의 값, Strong edge 블록의 경우 60%의 값으로 설정해 사용하고 있다 [8]. 이때 Smooth 블록의 100%의 경우 누적 히스토그램에서 가장 큰 값을 의미하며, 이 값보다 큰 값만을 feature로 설정하기 때문에, 이 의미는 해당 블록 내에 feature가 없다는 것을 의미한다.

먼저 그림 6에 나타난 다수의 테스트 이미지 (Berkeley segmentation dataset [10])를 사용해 블록 유형 구분 알고리즘에서 발생할 수 있는 각 블록 유형의 비율을 확인하도록 한다. 이때 이미지는 8bit color image, 512×512이고, 블록의 크기는 64×64이다. 한 이미지 당 64개의 블록이 발생하고, 200개의 이미지를 고려하면 총 12,800개의 블록이 발생한다. 그림 2는 200장의 테스트 이미지 중 일부를 나타낸다 [10]. 알고리즘을 적용한 결과, Smooth 블록은 3,786개 (29.5%), Texture 블록은 3개 (0%), Edge/texture 블록은 3,418개 (26%), Medium edge 블록은 5,284개 (41.2%), Strong edge 블록은 309개 (2%)만큼 발생하는 것을 확인할 수 있다. 이 결과는 feature가 전혀 발생하지 않는 smooth 블록의 경우 전체 비율에서 29.5%나 발생하고 있다는 점을 의미한다. 분석 결과를 바탕으로 feature detection 알고리즘의 연산 에너지 효율을 증가시키기 위해 블록 다이어그램을 그림7 와 같이 변경한다. 그림 7에 나타난 것과 같이 영상이 입력되면 가장 먼저 블록 유형 구분 알고리즘을 수행한다. 블록 유형이 smooth 블록으로 구분된 경우, 영역 내에 유효한 edge 정보가 전혀 없는 것을 의미하기 때문에 이후 Canny edge detection을 위해 연산해야 하는 노이즈 제거, 이차원 미분 필터, 그라디언트 크기 및 방향, 비최대 억제, 적응형 임계값 연산, 히스테리시스 임계 처리의 총 여섯 단계의 연산을 생략할 수 있게 된다.

5. 결론

본 논문은 Smooth 블록에 edge와 같은 feature가 포함되지 않는 점을 이용해 feature detection 연산에 요구되는 latency를 절감하는 방법을 제안한다. Berkeley 테스트 이미지 [10]를 활용해 실험한 결과

200장의 이미지, 총 12,800개의 블록 중에서 29.5%의 블록이 smooth 블록임을 확인했다. 이는 Macro block을 기본 단위로 하는 특징추출 알고리즘에서 feature가 발생하지 않는 local 영역의 연산을 생략할 수 있는 것을 의미한다. 알고리즘을 직렬로 수행하는 일반적인 환경에 적용하면, Canny edge detection의 경우, 다섯 단계의 연산 과정을 생략할 수 있다. 이후 연구에서 다양한 feature detection 알고리즘에 적용할 수 있는 방향을 고안하도록 한다.

REFERENCES

- [1] W. Kim, J. Lee, H. An, and J. Kim, "High-Performance and Low-Complexity Image Pre-Processing Method Based on Gradient-Vector Characteristics and Hardware-Block Sharing," *Trans. Electr. Electron. Mater. (TEEM)*, vol. 18, no. 6, pp. 320-322, Dec. 2017.
- [2] W. Kim, J. Lee, and H. An, "Gradient Magnitude Hardware Architecture based on Hardware Folding Design Method for Low Power Image Feature Extraction Hardware Design," *Journal of Korea institute of information, electronics, and communication technology (KIIECT)*, vol. 10, no. 2, pp. 141-146, Apr. 2017.
- [3] W. Kim, J. Lee, H. An, and B. Kim, "Image Filter Optimization Method based on common sub-expression elimination for Low Power Image Feature Extraction Hardware Design," *Journal of Korea institute of information, electronics, and communication technology (KIIECT)*, vol. 10, no. 2, pp. 192-197, Apr. 2017.
- [4] W. Kim, J. Lee, and H. An, "Low Complexity Gradient Magnitude Calculator Hardware Architecture Using Characteristic Analysis of Projection Vector and Hardware Resource Sharing," *Journal of Korea institute of information, electronics, and communication technology (KIIECT)*, vol. 9, no. 4, pp. 414-418, Aug. 2016.

[5] J. Canny, "A Computational Approach to Edge Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, Nov. 1986.

[6] M. Nixon and A. Aguado, "Low-level feature extraction," in *Feature Extraction & Image Processing for Computer Vision*, 3rd ed., Academic Press, Sep. 2012.

[7] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62-66, Jan. 1979.

[8] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A Distributed Canny Edge Detector: Algorithm and FPGA Implementation," *IEEE Trans. on Image Processing*, vol. 23, no. 7, pp. 2944-2960, Jul. 2014.

[9] J. K. Su and R. M. Mersereau, "Post-processing for artifact reduction in JPEG-compressed images," in *Proc. IEEE ICASSP*, vol. 3, pp. 2363-2366, May. 1995.

[10] P. Arbelaez, C. Fowlkes, and D. Martin, (2015). The Berkeley Segmentation Dataset and Benchmark [Online]. Available: <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

Author Biography

이 주 성 (Juseong Lee)

[정회원]



- 2013년 2월 : 서울과학기술대학교 전자IT미디어공학과 (공학사)
- 2015년 8월 : 고려대학교 전기전자공학과 (공학석사)
- 2018년 11월 : (재)실감교류인체감응솔루션연구단 (연구원)
- 2019년 1월 ~ 현재 : 고려대학교 전기전자공학과 (연구원)

〈Research Interests〉 반도체 설계, 저전력 영상처리

안 호 명 (Ho-Myoung An)

[종신회원]



- 2001년 2월 : 광운대학교 전자재료공학과 (공학사)
 - 2003년 2월 : 광운대학교 전자재료공학과 (공학석사)
 - 2009년 2월 : 광운대학교 전자재료공학과 (공학박사)
 - 2009년 3월 ~ 2013년 2월 : 고려대학교 전기전자전파공학과 연구교수
 - 2013년 3월 ~ 현재 : 오산대학교 전자과 조교수
- 반도체 소자, 비휘발성메모리, 광소자, 산화물반도체

〈Research Interests〉