Journal of Korea Multimedia Society Vol. 20, No. 2, February 2017(pp. 163-169)
https://doi.org/10.9717/kmms.2017.20.2.163

근사 임계값 추정을 통한 Otsu 알고리즘의 연산량 개선

이영우†, 김진헌††

A Computational Improvement of Otsu's Algorithm by Estimating Approximate Threshold

Youngwoo Lee†, Jin Heon Kim††

ABSTRACT

There are various algorithms evaluating a threshold for image segmentation. Among them, Otsu's algorithm sets a threshold based on the histogram. It finds the between-class variance for all over gray levels and then sets the largest one as Otsu's optimal threshold, so we can see that Otsu's algorithm requires a lot of the computation. In this paper, we improved the amount of computational needs by using estimated Otsu's threshold rather than computing for all the threshold candidates. The proposed algorithm is compared with the original one in computation amount and accuracy. we confirm that the proposed algorithm is about 29 times faster than conventional method on single processor and about 4 times faster than on parallel processing architecture machine.

Key words: Otsu's Algorithm, Image Binarization, Image Segmentation, Image Thresholding, High Speed Implementation

1. 서 론

영상분할은 영상분석, 패턴인식 등의 전처리를 위해 자주 사용되는 방법 중 하나이다. 대표적인 영상분할 방법들에는 임계값(threshold) 방법, 에지(edge) 검출 기법, 영역 성장 법, 텍스처(texture) 특징 값을 이용한 방법 등이 있다. 그중에서 임계값 방법은 주어진 영상에 대해 히스토그램(histogram)을 구하고 각 방법들에 맞게 임계값을 설정하여 영상의 객체와 배경을 분리하는 방법이다.

현재 영상분할을 하기 위해 임계값 방법을 활용하는 연구가 활발히 진행되고 있으며 B. Sankur 등[1]은 임계값 방법을 6가지로 분류하였다. 각각 히스토그램을 활용하는 방법[2,3], 군집화를 활용하는 방법[4-6], 엔트로피를 활용하는 방법[7,8], 객체 속성을 활용하는 방법[9-11], 공간 속성을 활용하는 방법[12, 13], 지역 적응적 방법[14,15] 등이 있다.

히스토그램을 활용하는 방법은 히스토그램의 모양을 보고 임계값을 설정하는 방법이다. 여기에는 볼록 헐(convex-hull)을 활용하는 방법[2], 스무딩(smoothing)을 이용하여 정상(peak)과 계곡(valley)을 찾는 방법[3] 등이 있다. 군집화를 활용하는 방법은 그레이 영상을 두 개의 군집으로 분리하고 각각의 정상 사이의 중앙지점을 찾으려한다. 두 클래스(class)를 갖는 가우시안 혼합 모델을 이용하여 반복연산을 통해 찾는 방법[4], 클래스 간 분산이 최대가

※ Corresponding Author : Jin Heon Kim, Address: (02713) Bukakwan 513, Seogyeong-ro 124, Seongbuk-gu, Seoul, Korea, TEL : +82-2-940-7747, FAX : +82-2-940-7114, E-mail : jinheon@skuniv.ac.kr
Receipt date : Jan. 13, 2017, Approval date : Jan. 19, 2017

† Graduate School of Electronic & Computer Eng., Seokyeong University
(E-mail : lywwkd@naver.com)
†† Graduate School of Electronic & Computer Eng., Seokyeong University

되는 지점을 찾는 방법[5], 등분산 가우시안 밀도 함수를 이용하여 오분류된 군집을 최소화하는 방법[6] 등이 있다. 엔트로피를 활용하는 방법은 원영상과 그레이 영상의 엔트로피를 활용하여 임계값을 설정하는 방법이다. 영상의 객체와 배경을 분리하여 두 클래스의 엔트로피의 합이 최대가 되게 하는 방법[7], 교차 엔트로피를 최소화하는 방법[8] 등이 있다. 객체 속성을 활용하는 방법은 원영상과 이진화영상의 유사성을 이용하여 임계값을 설정한다. 모멘트(moment)를 보존하는 방법[9], 에지영역을 비교하는 방법[10], 위상공간(topological) 안전성을 활용한 방법[11] 등이 있다. 공간 속성을 활용하는 방법은 그레이 값 분포와 주변 화소의 상관관계를 활용하여 임계값을 설정한다. 동시 발생 확률을 이용한 방법[12], 영상의 객체로부터 최대-최소의 거리를 활용하는 방법[13] 등이 있다. 지역 적응적 방법은 지역적 영상 특성에 따라 임계값을 설정하는 방법이다. 지역 평균과 분산을 활용한 방법[14], 주변 화소의 밝기 평균와 비교하는 방법[15] 등이 있다.

이와 같이 현재 영상분할을 위해 수많은 방법들이 연구되었다. 그중에서도 Otsu 알고리즘이 자동차[16], 얼굴인식[17], 의료[18,19] 등과 같은 분야에서 널리 사용되고 있다. 또한 Otsu를 개선하기 위해 표준편차를 활용하는 방법[20] 등 많은 연구도 진행되고 있다. 하지만 Otsu 알고리즘이 모든 그레이 레벨에 대해 반복적인 계산[21,22]을 해야 하기에 속도 개선의 여지가 있다. GP-GPU(General Purpose Graphic Processor Unit) 혹은 SIMD(Single Instruction Multi Data) 구조의 특성을 갖는 현대의 고성능 PC 환경에서는 이러한 중복 연산이 연산량에 비해서는 빠른 속도로 실행되고 있으나 이러한 구조의 보조 연산 장치가 없는 임베디드 시스템과 같은 환경에서나 특히 실시간 구현을 위해 전용 H/W 연산 유니트(unit)를 만들고자 하는 FPGA(Field Programmable Gate Array)와 같은 연산자의 H/W 구현에서는 Otsu 알고리즘에 대한 개선이 필요한 상황이다.

본 논문에서는 Otsu 알고리즘의 연산량을 줄여 다양한 분야에서 활용 가능하게 하고자 한다. 기존의 Otsu 알고리즘은 모든 그레이 값에 대해서 분산을 구해야 했지만 제안된 방법은 미분 가중치를 활용하여 근사 Otsu 임계값을 추정해나간다. 따라서 모든 그레이 값에 대해 연산할 필요 없이 몇 개의 값만 연산해보면 Otsu 임계값을 찾을 수 있다.

본 논문의 구성은 아래와 같다. 제2장에서는 Otsu 알고리즘이 간단히 소개되고 제3장에서는 제안된 방법을 설명한다. 제4장에서는 두 알고리즘의 수행 시간을 비교하여 실험 결과를 확인하고 제5장에서 결론을 내린다.

2. 이론적 배경

Otsu 알고리즘[5]은 영상의 히스토그램을 구한 후 두 개의 클래스로 이 분할하여 클래스 간 분산이 최대가 되는 지점을 찾아 임계값으로 설정하는 방법이다.

주어진 영상이 L개의 그레이 레벨 $[1, 2, \cdots, L]$ 을 갖고 있다고 하자. 그레이 레벨이 i 인 화소의 수를 n_i이라고 할 때, 총 화소 수는 $N = n_1 + n_2 + \cdots + n_L$ 이다.

$$p_i = n_i / N, \quad p_i \geq 0, \quad \sum_{i=1}^{L} p_i = 1 \tag{1}$$

영상을 배경과 객체 두 개의 클래스로 나누면, $[1, \cdots, k]$ 의 그레이 레벨을 갖는 클래스 C_0와 $[k+1, \cdots, L]$ 의 그레이 레벨을 갖는 클래스 C_1를 얻을 수 있다. 각 클래스의 확률분포는

$$w_0 = \Pr(C_0) = \sum_{i=1}^{k} p_i = w(k) \tag{2}$$

$$w_1 = \Pr(C_1) = \sum_{i=k+1}^{L} p_i = 1 - w(k) \tag{3}$$

이고, 각 클래스 C_0 와 C_1의 평균은

$$\begin{aligned} \mu_0 &= \sum_{i=1}^{k} iPr(i|C_0) \\ &= \sum_{i=1}^{k} ip_i / \omega_0 = \mu(k)/\omega(k) \end{aligned} \tag{4}$$

$$\begin{aligned} \mu_1 &= \sum_{i=k+1}^{L} iPr(i|C_1) \\ &= \sum_{i=k+1}^{L} ip_i / \omega_1 = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \end{aligned} \tag{5}$$

이다. 원 영상의 전체 평균은

$$\mu_T = \mu(L) = \sum_{i=1}^{L} ip_i \tag{6}$$

이다. 또한, 전체 평균은

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T, \quad \omega_0 + \omega_1 = 1 \tag{7}$$

로 표현할 수 있다. 각 클래스 C_0 와 C_1의 분산은

$$\sigma_0^2 = \sum_{i=1}^{k}(i-\mu_0)^2 \Pr(i|C_0) = \sum_{i=1}^{k}(i-\mu_0)^2 p_i/\omega_0 \quad (8)$$

$$\sigma_1^2 = \sum_{i=k+1}^{L}(i-\mu_1)^2 \Pr(i|C_1) = \sum_{i=k+1}^{L}(i-\mu_1)^2 p_i/\omega_1 \quad (9)$$

이다. 최적의 임계값은 아래의 판별식들 중 하나를 선택하여 그 값이 가장 크게 나오는 k를 선택하면 된다.

$$\lambda = \sigma_B^2/\sigma_W^2, \quad \kappa = \sigma_T^2/\sigma_W^2, \quad \eta = \sigma_B^2/\sigma_T^2 \quad (10)$$

$$\sigma_W^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \quad (11)$$

$$\sigma_B^2 = \omega_0(\mu_0-\mu_T)^2 + \omega_1(\mu_1-\mu_T)^2 = \omega_0\omega_1(\mu_1-\mu_0)^2 \quad (12)$$

$$\sigma_T^2 = \sum_{i=1}^{L}(i-\mu_T)^2 p_i \quad (13)$$

위의 σ_W^2는 클래스 내 분산, σ_B^2는 클래스 간 분산, σ_T^2는 전체 분산을 의미한다. 식 (10)의 판별식에서 어떠한 판별식을 사용하더라도 같은 결과를 얻을 수 있다. η를 선택하여 계산을 하면 k가 σ_T^2 계산에 영향을 안 줌으로 계산상의 이점이 있다.

즉, 아래의 식 (15)을 최대화하는 k^*를 찾으면 Otsu의 임계값을 얻을 수 있다.

$$\eta(k) = \sigma_B^2(k)/\sigma_T^2 \quad (14)$$

$$\sigma_B^2(k) = \frac{[\mu_T\omega(k)-\mu(k)]^2}{\omega(k)[1-\omega(k)]} \quad (15)$$

$$\sigma_B^2(k^*) = \max\ \sigma_B^2(k),\ 1 \le k \le L \quad (16)$$

3. 제안한 방법

기존의 Otsu 알고리즘은 영상의 모든 그레이 레벨에 대해 반복적으로 임계값을 설정하고, 그 임계값을 기준으로 두 개의 클래스를 나눈 후, 클래스 간 분산을 구하는 방법으로 많은 연산량을 필요로 한다. 따라서 본 논문에서는 영상의 모든 그레이 레벨에 대해 임계값을 설정하지 않고 근사 Otsu 임계값을 추정하여 반복 횟수를 줄임으로써 전체적인 연산량을 줄였다.

제안된 방법은 크게 3단계로 구성된다. Stage 1에서는 초깃값과 진행 방향을 설정하고, Stage 2에서는 미분 가중치에 따라 Otsu 임계값의 근사 위치를 찾아간다. Stage 3에서 정확한 Otsu 임계값을 찾는다. 실제 입력 영상을 갖고 제안된 방법으로 Otsu의 임계값을 찾아가는 모습을 사진과 함께 설명함으로써 제안된 방법의 알고리즘을 설명하고자 한다. 설명에 사용되는 영상은 63빌딩 전경으로써 5312x2988의 고해상도 영상이다.

Fig. 1은 제안된 방법을 설명하기 위해 사용되는 영상으로써 63빌딩의 전경을 보여주고 있다.

Fig. 2는 63빌딩 전경 영상의 히스토그램을 나타내고 있다. 또한 영상의 평균 밝기(빨간선)와 Otsu 알고리즘의 임계값(초록선)을 나타내고 있다.

Fig. 3은 제안한 방법의 첫 번째 단계이다. 위의 그림은 Otsu 알고리즘의 클래스 간 분산(σ_B^2)을 모든 그레이 레벨에 대해서 나타내고 있다. σ_B^2을 구하는 방법은 식 (15)를 이용하였다. 분산은 평균을 기준으로 하향 곡선을 나타낸다. 따라서 σ_B^2의 그래프 또한 위와 같이 나타나게 된다. Otsu 임계값은 영상의 평균 밝기 근처에 존재할 확률이 높다[4]. 그래서 stage 1의 초깃값 설정은 영상의 밝기 평균을 기준으로 좌측, 우측을 임계값으로 설정한다. 이후 좌측 임계값(T_L), 우측 임계값(T_R), 평균 밝기 값(T)의 σ_B^2을

Fig. 1. The 63building scenery image.

Fig. 2. The histogram of 63building scenery image.

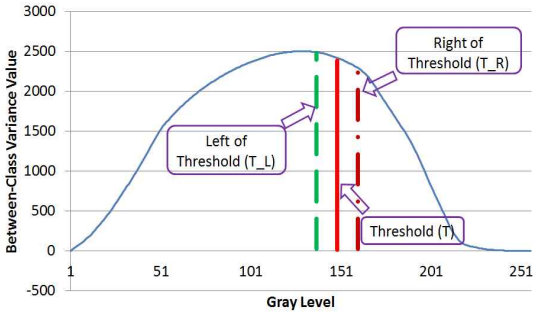


Fig. 3. The set of first threshold and progress direction with mean (stage 1).

구한다. T_L과 T_R을 정하는 기준은 T를 중심으로 사용자가 정하는 상숫값을 이용한다. 본 논문에서는 T를 기준으로 ±10을 사용하였다. T_L과 T_R의 σ_B^2을 비교하여 값이 큰 쪽으로 T를 이동시킨다. 위의 그림에서는 T_L의 σ_B^2(2445.591)이 T_R의 σ_B^2(2178.911)보다 큼으로 T가 T_L로 이동할 것이다.

Fig. 4는 제안된 방법의 두 번째 단계이다. 새로 정해진 T를 기준으로 새로운 T_L를 설정하게 된다. 설정 방법은 stage 1의 T의 σ_B^2(2349.341)과 T_L의 σ_B^2(2445.591), 두 값을 갖고 미분(9.62)을 구하여 미분 가중치(9.62*1)만큼 σ_B^2이 큰 쪽 방향으로 이동하게 된다. 미분 가중치는 구한 미분 값과 임의의 상수를 곱하여 구해진다. 이 방법은 PID 제어의 P(proportion)와 D(derivative) 제어의 개념이 내포되어있다. P는 결과 값을 피드백(feedback) 받아 다음 설정값을 조절한다. 제안된 방법에서는 P의 방법으로 다음 임계값을 설정한다. D는 다음 임계값을 설정하기 위한 이동량을 조절하는 지표로 사용되어 오버슈트(overshoot)가 생기는 것을 줄여준다. 미분의 크기를

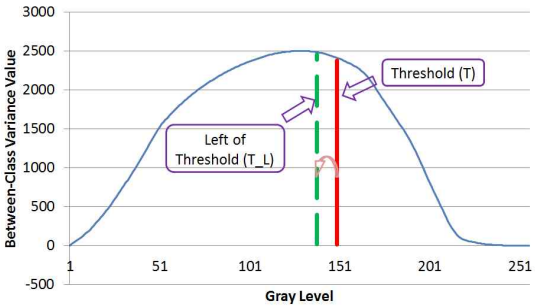


Fig. 4. The search of approximate Otsu's threshold (stage 2).

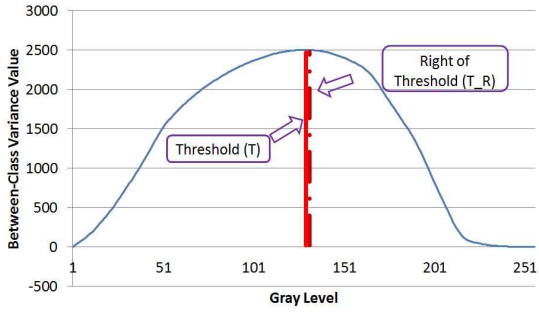


Fig. 5. The search of accurate Otsu's threshold (stage 3).

확인하여 기울기가 크면 많이 이동하고 적으면 적게 이동함으로써 찾고자 하는 σ_B^2의 최댓값에 정확하게 도달할 수 있다.

Fig. 5는 제안된 방법의 세 번째 단계이다. 진행 방향의 두 임계값의 σ_B^2을 비교하여 앞선 임계값의 σ_B^2이 작아지게 되면 진행 방향을 바꾸어 하나의 그레이 레벨씩 이동하여 정확한 Otsu 임계값을 찾아간다. 위 그림에서는 T_L의 σ_B^2(2504.373)이 더 이상 T의 σ_B^2(2504.642)보다 크지 않아서 왼쪽으로 진행하지 않고 오른쪽의 T_R(2504.837)을 설정한다. 이유는 왼쪽에 더 이상 현재의 σ_B^2보다 큰 σ_B^2이 존재하지 않기 때문이다. 위의 과정을 통해 정확한 Otsu 임계값(T=129)을 찾은 것을 확인할 수 있다.

4. 실험 결과 및 고찰

제안한 방법의 연산량 개선을 확인하기 위해 저화질 영상(SVGA급)과 고화질 영상(UHD급) 각각 평균 밝기를 기준으로 세 그룹으로 분류하였다. 영상의 평균 밝기가 64이하는 어두운(dark), 96이상 160이하는 중간(middle), 192이상은 밝은(bright)으로 정하였다, 각 밝기마다 영상 5장씩 총 30장의 영상으로 기존의 Otsu 알고리즘과 비교 실험을 하였다. 실험 환경은 Intel i3 3.70 Ghz 프로세서, 8GB 메모리, 운영체제는 Window 10, 개발 툴은 Matlab 2014버전을 사용하였다.

4.1 제안한 방법의 정확성

Fig. 6에서는 Otsu 알고리즘의 클래스 간 분산과 제안한 방법의 클래스 간 분산의 최댓값을 비교하였

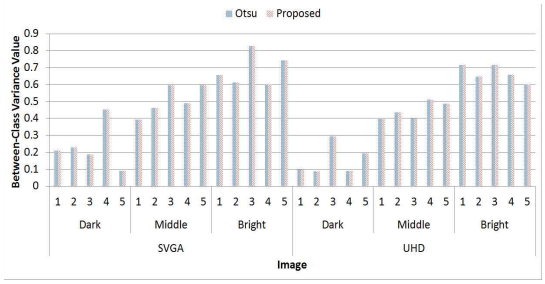


Fig. 6. The comparison of Otsu and Proposed algorithm's between-class variance.

다. Otsu 알고리즘에서는 클래스 간 분산이 최대가 될 때 그 위치를 임계값으로 설정한다. 위 그림에서 두 알고리즘의 결과 값이 정확히 같음을 확인할 수 있다.

4.2 두 알고리즘의 연산 속도 비교

Fig. 7에서는 Otsu 알고리즘(parallel, loop)과 제안한 방법의 수행 시간을 나타내고 있다. 실험에 사용된 Otsu 알고리즘은 Matlab에서 지원하는 Graythresh 함수로써 매트릭스 연산을 수행한다. 즉 병렬처리 방법을 이용하고 있다. 제안한 방법이 병렬처리를 이용한 Otsu 알고리즘에 비해 평균 4배 빠르고, loop 문을 이용한 Otsu 알고리즘보다는 평균 29배 빠른 것을 확인할 수 있다. 또한 메모리 사용 측면에서도 더 좋은 효율을 나타낸다.

4.3 미분 가중치에 따른 연산 횟수 비교

Fig. 8은 미분 가중치에 따라 달라지는 연산 횟수를 나타내고 있다. (a)에서는 stage 2의 연산 횟수를 나타낸다. 가중치가 커질수록 연산 횟수가 줄어드는 것을 확인할 수 있다. 이유는 한 번에 이동하는 거리

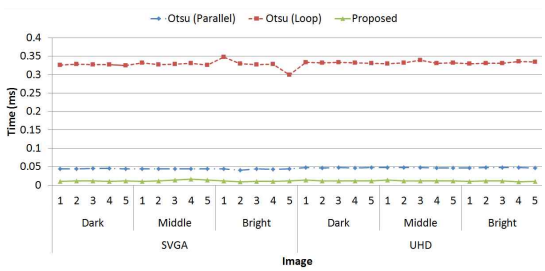
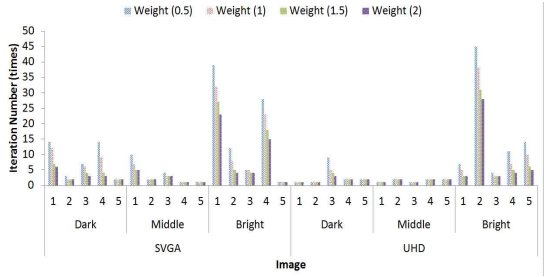
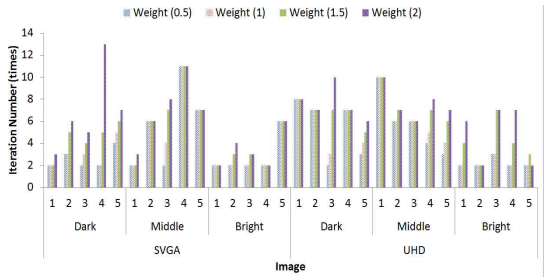


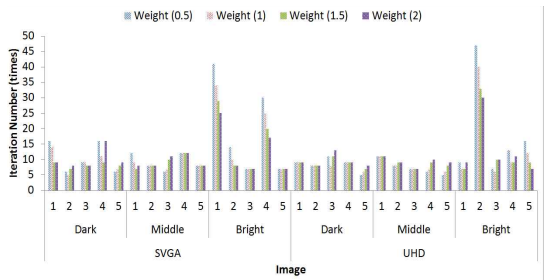
Fig. 7. The comparison of Otsu and Proposed algorithm's execution time.



(a)



(b)



(c)

Fig. 8. The comparison of iteration times depending on differential weight. (a) stage 2, (b) stage 3, and (c) total.

가 길어지기 때문이다. (b)에서는 stage 3의 연산 횟수를 나타낸다. 가중치가 커질수록 연산 횟수가 늘어나는 것을 확인할 수 있다. 이유는 stage 2에서 오버슈트가 더 크게 발생했기에 정확한 Otsu 임계값을 찾아가는 연산 횟수가 증가한 것이다. (c)에서는 미분의 가중치에 따라 달라지는 총 연산 횟수를 나타낸다. 가중치가 낮은 순으로 평균 12번, 11번, 10번, 11번의 연산 횟수를 나타낸다.

4.4 히스토그램 연산을 포함한 두 알고리즘의 수행 시간 비교

Fig. 9에서는 Otsu 알고리즘과 제안한 방법의 수행 시간을 히스토그램을 구하는 과정부터 계산하였

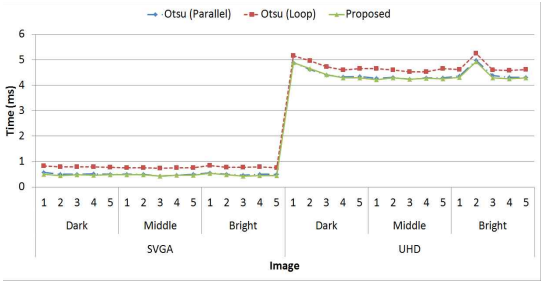


Fig. 9. The comparison of Otsu and Proposed algorithm's execution time including histogram computation.

다. 그 결과 두 알고리즘 간의 수행 시간 차이가 거의 나지 않는 것을 확인할 수 있었다. 이유는 히스토그램을 구하는 과정이 전체 연산 시간에 많은 부분을 차지하고 있기 때문이다. 히스토그램은 Otsu의 임계값을 구하기 위한 필수조건이다. 따라서 히스토그램을 구하는 시간을 단축시키는 것 또한 연산 속도 개선 측면에서 중요한 역할을 한다. FPGA를 활용하면 도움이 될 수 있다. 기존의 PC 환경에서는 입력 영상을 메모리에 저장한 후 각 화소 값의 개수를 세야만 히스토그램을 구할 수 있다. 반면 FPGA를 이용하면 영상이 입력되는 순간마다 계속 화소 값을 누적시켜 실시간으로 히스토그램을 구현할 수 있다. 따라서 Otsu 임계값을 구하는데 시간이 대폭 단축될 것이라 예상된다.

5. 결 론

본 논문에서는 Otsu 알고리즘의 연산량을 줄임으로써 연산 시간을 단축시키는 방법을 제안하였다. 제안된 방법은 기존의 모든 그레이 레벨에 대해 연산을 하던 방법에서 근사 Otsu의 임계값을 추정하는 방법으로 연산 횟수를 줄였다. 또한 Otsu 알고리즘과 제안된 방법의 비교 실험을 통해 그 타당성을 입증하였다.

현재의 고성능 PC 환경에서는 Otsu 알고리즘의 속도 개선이 두드러지게 체감되지는 않는다. 하지만 병렬 연산이 불가능한 환경, 임베디드 시스템, 저용량 FPGA의 경우에서는 기존의 Otsu 알고리즘의 연산량 문제가 걸림돌이 될 수 있다. 향후 H/W 구현을 위한 최적화 문제에 대한 연구가 더 필요한 것으로 기대되며 위의 경우들에 대해서 제안한 방법을 적용하여 비교 실험 및 초깃값 설정과 미분 가중치 지표 설정에 대한 고찰이 필요할 것이라고 생각된다.

REFERENCE

[1] B. Sankur, and M. Sezgin, "Survey over Image Thresholding Techniques and Quantitative and Quantitative Performance Evaluation," *Journal of Electronic Imaging*, Vol. 13, No. 1, pp. 146-165, 2004.

[2] A. Rosenfeld, and P. De la Torre, "Histogram Concavity Analysis as an Aid in Threshold Selection," *IEEE Transactions on Systems Man Cybernetics*, Vol. SMC-13, Issue 2, pp. 231-235, 1983.

[3] M.I. Sezan, "A Peak Detection Algorithm and Its Application to Histogram-Based Image Data Reduction," *Journal of Computer Vision, Graphics, and Image Processing*, Vol. 49, No. 1, pp. 36-51, 1990.

[4] T.W. Riddler, and S. Calvard, "Picture Thresholding Using an Iterative Selection Method," *IEEE Transactions on Systems Man Cybernetics*, Vol. SMC-8, No. 8, pp. 630-632, 1978.

[5] N. Otsu, "A Threshold Selection Method from Gray Level Histograms," *IEEE Transactions on Systems Man Cybernetics*, Vol. SMC-9, No. 1, pp. 62-66, 1979.

[6] D.E. Lloyd, *Automatic Target Classification Using Moment Invariant of Image Shapes*, Technical Report, RAE IDN AW126, Farnborough, UK, 1985.

[7] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram," *Graph Models Image Processing*, Vol. 29, Issue 3, pp. 273-285, 1985.

[8] H. Li, and P.K.S. Tam, "An Iterative Algorithm for Minimum Cross-Entropy Thresholding," *Pattern Recognition Letters*, Vol. 19, Issue 8, pp. 771-776, 1998.

[9] W.H. Tsai, "Moment-Preserving Thresholding: A New Approach," *Graph Models Image Processing*, Vol. 19, pp. 377-393, 1985.

[10] L. Hertz, and R.W. Schafer, "Multilevel Thresholding Using Edge Matching," *Computer*

Vision, Graphics, and Image Processing, Vol. 44, Issue 3, pp. 279-295, 1988.

[11] J.C. Russ, "Atomatic Discrimination of Features in Gray-Scale Images," *Journal of Microscopy*, Vol. 148, No. 3, pp. 263-277, 1987.

[12] B. Chanda, and D.D. Majumder, "A Note on the Use of Gray Level Co-Occurrence Matrix in Threshold Selection," *Signal Processing*, Vol. 15, Issue 2, pp. 149-167, 1988.

[13] N. Friel, and I.S. Molchanov, "A New Thresholding Technique Based on Random Sets," *Pattern Recognition*, Vol. 32, Issue 9, 1999.

[14] W. Niblack, *An Introduction to Image Processing*, Prentice-Hall, Englewood Cliffs, pp. 115-116, 1986.

[15] J.M. White, and G.D. Rohrer, "Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction," *IBM Journal of Research Development*, Vol. 27, No. 4, pp. 400-411, 1983.

[16] J.H. Kim, and G. Kim, "A Binarization Technique Using Histogram Matching for License Plate with a Shadow," *Journal of Broadcast Engineering*, Vol. 19, No. 1, pp. 56-63, 2014.

[17] H.J. Lee, and J.H. Chung, "Face Recognition Robust to Brightness, Contrast, Scale, Rotation and Translation," *Journal of the Institute of Electronics Engineers of Korea*, Vol. 40, No. 6, pp. 149-156, 2003.

[18] K.S. Kim, S.W. Shin, S. Lee, J.S. Jeong, W. Park, and K.D. Kim, "Color Image Segmentation for Extracting Dental Plaque," *The Transactions of the Korean Institute of Electrical Engineers*, Vol. 60, No. 6, pp. 1183-1189, 2011.

[19] G.H. Jang, H.H. Park, S.L. Lee, D.H. Kim, and M.K. Lim "An Effective Extraction Algorithm of Pulmonary Regions Using Intensity-Level Maps in Chest X-Ray Images," *Journal of Korea Multimedia Society*, Vol. 13, No. 7, pp. 1062-1075, 2010.

[20] J.M. Sung, H.G. Ha, and B.Y. Choi, "Image Thresholding Based on Within-Class Standard Deviation," *Journal of the Institute of Electronics Engineers of Korea*, Vol. 50, No. 7, pp. 1844-1852, 2013.

[21] Z.X. Li, and S.W. Kim, "A Multi-Thresholding Approach Improved with Otsu's Method," *Journal of the Institute of Electronics Engineers of Korea*, Vol. 43, No. 5, pp. 407-415, 2006.

[22] B.M. Singh, R. Sharma, A. Mittal,, and D. Ghosh, "Parallel Implementation of Otsu's Binarization Approach on Graphics Processing Unit," *International Journal of Computer Applications*, Vol. 32, No. 2, pp. 16-21, 2010.

이 영 우

2015년 서경대학교 컴퓨터공학과 졸업(공학사)
2015년 현재 서경대학교 전자컴퓨터공학과 재학(석사과정)
관심분야 : 영상처리, 시스템 설계

김 진 헌

1982년 고려대학교 공과대학 전기공학과 공학사
1984년 고려대학교 대학원 전기공학과 공학석사
1983년 동양정밀공업(OPC) 중앙연구소 연구원
1986년 삼성종합기술원 선임연구원
1989년 ZyMOS 한국지사 FAE
1990년 고려대학교 대학원 전기공학과 공학박사
1995년 현재 서경대학교 컴퓨터공학과 부교수
관심분야 : 디지털영상처리, 영상신호처리(ISP), 영상/비디오시스템