

패턴매칭을 위한 고속 스레쉬홀딩법

A Fast Thresholding Method For Pattern Matching

이철학*, 김상운**
(Zhe-Xue Li, Sang-Woon Kim)

Abstract-For pattern matching, an object image should be segmented and analyzed for the first time. Thresholding is a fundamental approach to segmentation that utilizes a significant degree of pixel popularity or intensity. Otsu's thresholding is one of the most well-known methods proposed in the literature. However, the method has a disadvantage of repeatedly searching the optimal thresholds for the entire region. To overcome this problem, a number of methods have been proposed. In this paper, we propose a simple and fast thresholding method of finding multi-level threshold values by extending the Otsu's method. Our experimental results for the benchmark images show a possibility that the proposed method could be used efficiently for pattern matching.

key word: 판별식, 스레쉬홀드, 패턴매칭, 영상분할, Otsu의 방법

1. 서 론

얼굴인식을 포함한 영상인식처리 분야에서 원 영상으로부터 관심 객체의 특성을 분할 추출하여 비교하는 작업을 하게 된다. 이를 위해 많이 사용하는 방법에는 영상의 에지를 이용한 방법과 스레쉬홀드를 이용하여 2진화영상으로 변환하거나, 그들을 결합하여 연구하는 방법 등이 있다. 스레쉬홀드를 구하는 방법들 가운데서 Otsu의 방법은 비교적 유효한 방법이다. 그러나 여러 객체로 이루어진 영상에서 클래스사이의 분산값이 가장 큰 최적의 스레쉬홀드를 찾기 위하여 그레이 레벨 전 구간에 대해 모든 가능한 분산값을 반복적으로 계산해야 하므로 계산시간이 많이 걸리고 비효율적이다.

지금까지 Otsu 방법의 계산시간을 축소하기 위한 연구들이 많이 진행되어 왔다. 문헌 [1]에서는 이 분야 연구에 대한 포괄적인 분석을 하고 스레쉬홀딩 기술을 6가지로 분류하였다. 본 논문에서 제안한 방법은 Otsu방법[2]을 확장한 방법으로, 영상의 전체 그레이레벨 구간 $[1, L] = \{1, 2, \dots, L\}$ 을 2개의 클래스로 나누고, 다시 4개, 8개, 16개 등으로 나누는 방법이다. 즉, 영상 히스토그램을 6개 클래스로 나누려고 할 때 두 번에 걸쳐 그레이레벨을 4개 클래스로 나누고, 얻어지는 스레쉬홀드 3개를 저장한다. 그리고 다시 각 클래스를 2개로 나눈다. 4개의 클래스에서 2개로 나눌 때, 얻은 클래스사이 분산과 그 클래스 전체 분산의 비율이 큰 것으로 클래스 2개를 선택하고 그 클래스에서 얻은 스레쉬홀드 2개를 이미 얻은 3개의 스레쉬홀드에 추가하여 구하려는 멀티레벨 스레쉬홀드값을 구한다.

실제 영상에 대한 분할실험결과, 대상영상을 특성에 맞게 고속으로 잘 분할 수 있었음을 확인하였으며, 패턴매칭을 이용한 얼굴인식에서 실용적임을 알 수 있었다.

2. 기존의 방법들

2.1 Otsu의 방법

Otsu의 방법[2]은 영상 밝기에 대하여, 분할할 두 클래스의 분리를 클래스사이의 분산이 최대화하는 스레쉬홀드를 구하는 방법이다.

영상이 L 개 그레이레벨 $[1, 2, \dots, L]$ 의 화소로 되어 있다고 한다. 그리고 밝기 값이 i 인 화소의 수를 f_i , 영상의 총 화소 수를 N 이라고 할 때

$$N = \sum_{i=1}^L f_i$$

이고 화소가 i 밝기 값을 갖는 확률은

$$p_i = f_i / N$$

이 된다. 그리고

$$\sum_{i=1}^L p_i = 1, p_i \geq 0$$

이다. 입력 영상이 배경과 객체 두 개로 나누어 졌다면 화소들은 그레이레벨 $[1, \dots, t]$ 에 속하는 두 클래스로 나눌 수 있다.

두 클래스에 대하여 클래스내의 분산(within-class variance)을 v_w^2 , 클래스사이의 분산(between-class variance)을 v_B^2 , 그리고 전체 구간에서의 분산(total variance)을 v_T^2 라고 하면 최적의 스레쉬홀드는 다음의 판별 기준식을 최대화하는 t 로 선택할 수 있다.

$$\eta = v_B^2 / v_T^2 \tag{1}$$

여기서

$$v_B^2 = \omega_1(\mu_1 - \mu_T)^2 + \omega_2(\mu_2 - \mu_T)^2 \tag{2}$$

$$v_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p(i) \tag{3}$$

이다. 또한, 이 방법을 쉽게 멀티레벨 스레쉬홀딩으로 확장할 수 있다.

* 이철학: 明知大 컴퓨터工學科 博士課程

** 김상운: 明知大 컴퓨터工學科 教授 · 工博

“이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임”
(MOEHRD-KRF-2005-042-D00265).

2.2 LIAO의 방법

Otsu의 방법에서는 최적의 스레쉬홀드를 구하기 위해 그레이 레벨 값 $[1, 2, \dots, L]$ 범위 내에서 모든 가능한 스레쉬홀드 $T=(t_1, t_2, \dots, t_{M-1})$ 을 취하고, 각각의 경우에 대하여 클래스사이의 분산값을 계산해야 한다. 이 과정에서 직접 식 (1)을 이용할 경우 같은 구간에 대한 클래스사이의 분산을 반복적으로 계산하게 된다. LIAO 등[3]은 이런 반복계산을 고려하여 식 (1)을 동등한 효과를 가지는 간단한 식으로 수정하여 처리했으며, 각 구간에서의 값을 미리 계산하여 테이블에 저장해 놓았다가 그 값을 한 번의 검색으로 처리하는 방법을 이용하였다.

2.3 WU의 방법

WU등[4]의 연구에서는 Otsu의 방법을 단순하게 확장하였다. 먼저 분할 판별계수를 구하고 나누어진 클래스들의 판별계수가 미리 정한 값 보다 작을 때, 나누어진 구간들에서 분산값이 큰 구간에 대하여 Otsu의 방법을 이용하여 그 구간을 2 개의 구간으로 나눈다. 이렇게 하나씩 스레쉬홀드를 증가하면서 판별계수값이 미리정한 값 보다 크게 될 때까지 진행한다. 판별계수(separability factor) SF 를 다음과 같이 정의한다.

$$SF = v_B^2(T) / v_T^2 = 1 - v_w^2(T) / v_T^2$$

여기서 $T=(t_1, t_2, \dots, t_{M-1})$ 은 이미 정해진 스레쉬홀드이다.

3. 제안 방법

본 논문의 방법은 Otsu방법을 확장한 방법으로, 먼저 영상의 전체 그레이레벨 구간 $[1, L] = \{1, 2, \dots, L\}$ 을 2개 클래스로 나누고, 2개 클래스를 다시 4개, 8개, 16개 등으로 나누며, 요구하는 스레쉬홀드의 개수에 따라 마지막으로 얻은 스레쉬홀드에서 클래스사이의 분산과 클래스분산의 비가 큰 것으로 모자라는 부분을 선택하여 취하는 방법을 이용하였다. 예를 들면, 영상 히스토그램을 6개 클래스, 즉 5개의 스레쉬홀드값으로 나누려고 할 때 두 번에 걸쳐 그레이레벨을 4개 클래스로 나누고, 얻어지는 스레쉬홀드 3개를 저장한다. 그리고 다시 각 클래스를 2개로 나눈다. 4개의 클래스에서 2개로 나눌 때, 얻은 클래스사이 분산과 그 클래스 전체 분산의 비율이(이 비율을 R 로 표시한다) 큰 클래스 2개를 선택하고 그 클래스에서 얻은 스레쉬홀드 2개를 이미 얻어진 3개의 스레쉬홀드에 추가하여 구하려는 멀티레벨 스레쉬홀드값을 구성한다.

본 논문에서 영상의 멀티레벨 스레쉬홀드값 찾는 방법은 다음과 같다. 여기서 C_0, C_1, \dots, C_q 는 분할하여 얻은 클래스, m 은 구하려는 스레쉬홀드 수, n 은 스레쉬홀드를 구하기 위한 순환회수($n=1, 2, \dots$), T_0 는 n 번째 순환에서 얻은 스레쉬홀드의 집합, $q_0=2^{n-1}$ 은 T_0 의 개수, $T=(t_1, t_2, \dots, t_q)$ 는 n 번 순환하여 얻은 스레쉬홀드의 집합, $q=2^n-1$ 은 T 의 개수이다.

Step 1: 영상 I 의 각 화소 그레이 값을 이용하여 그레이의

도수를 계산하고 초기 클래스 C_0 에 할당한다.

초기값으로 $q_0=0; q=0; n=1; m=0$ 을 취한다. T_0 과 T 는 공집합이다.

Step 2: 스레쉬홀드 $T=(t_1, t_2, \dots, t_q)$ 를 이용하여 얻은 영상의 그레이도수 각 클래스에 대하여 식 (2)과 식 (3)를 계산하고 얻은 스레쉬홀드를 T_0 에 할당한다. 그 스레쉬홀드의 개수를 q_0 에 할당한다.

Step 3 : $m > q + q_0$ 이면 T_0 를 T 에 추가하여 새로운 T 를 얻는다. $n=n+1$ 로 하고 Step 2로 간다.

$m = q + q_0$ 이면 T_0 를 T 에 추가하여 새로운 T 를 얻고 Step 4로 간다.

$m < q + q_0$ 이면 T_0 의 각 스레쉬홀드를 얻을 때 비율 R 값이 가장 큰 것으로 $m-q$ 개를 취하여 T 에 추가한다. 그리고 Step 4로 간다.

Step 4 : 얻은 $T=(t_1, t_2, \dots, t_q)$ 를 출력한다. 분할된 $q+1$ 개 클래스 (C_0, C_1, \dots, C_q)를 얻고 종료한다.

4. 실험

영상의 그레이레벨 $[1, 2, \dots, L]$ 이 스레쉬홀드 $T=(t_1, \dots, t_i, \dots, t_{M-1})$ 에 의하여 구간 $[1, t_1], \dots, [t_{i-1}+1, t_i], \dots, [t_{M-1}+1, L]$ 으로 나누어진다면 Otsu 원래 방법의 계산 복잡도는 $(5L+M+2) \cdot L! / M! (L-M)!$ 이다. M 이 L 에 비교해 많이 작으므로 이를 $O(L^{M+1})$ 로 볼 수 있다. LIAO는 원래 Otsu의 방법에서 식 (1)와 동등한 효과를 가지는 정리된 판별식을 사용하였다. 그러나 순환으로 인한 $O(L^M)$ 의 계산은 여전히 남아 있다.

WU[4]는 스레쉬홀드를 구할 때 매번 그 전의 스레쉬홀드를 기준하여 한 구간씩 선택하여 그 클래스에서 얻은 스레쉬홀드를 기존의 것에 하나씩 추가하는 방법을 사용하였다. 이렇게 M 단계 스레쉬홀드를 구하는데 $(M-1)(5L+M+2)$ 번 계산이 필요하므로 결국 $O(L)$ 의 계산이 된다.

제안 방법을 이용할 경우, WU의 방법보다 시간을 더 축소할 수 있다. WU의 방법에서 M 단계 스레쉬홀드, 즉 $M-1$ 개 스레쉬홀드를 구하는데 $M-1$ 번의 순환을 해야 하는데 반하여, 제안 방법에서는 $\log_2(M)$ 번 순환하면 된다.

영상의 실제적 분할 효과를 비교하기 위하여 본 논문에서는 그림 1과 같은 영상 "lena.jpg", "house.jpg", 그리고 "auroras.jpg"를 시험영상으로 이용하였다. 프로그램은 펜티엄 4 컴퓨터에서 Matlab 7 버전을 이용하여 구현하였다.

실험에서 방법들 사이의 비교를 위하여, WU의 방법을 구현할 때 판별계수를 이용한 스레쉬홀드수 자동선택방법을 사용하지 않고, 요구에 따라 반복수를 고정하여 실험하였다. 그리고 제안 방법과 WU의 방법에서 시간을 측정할 때, 100 번 순환을 실행시켜 얻은 결과시간을 100으로 나누어 얻었다.

표 1은 실험에서 얻은 수치이다. 원 Otsu의 방법과 Liao의 방법은 같은 스레쉬홀드값을 얻었다. 여기서는 WU의 방법과 제안 방법도 같은 스레쉬홀드값을 얻었다. 영상 분할

에서 4 단계 분할과 5 단계 분할사이의 실행시간을 비교하기 위해 여기서는 5 단계 즉 스톱스레쉬홀드 개수를 4개로 정하여 실험하였는데, Otsu의 방법에서 5 단계 분할 방법은 4 단계 분할 방법보다 놀라울 정도로 많은 실행시간이 걸리었다. Liao의 방법은 원래 Otsu의 방법보다 현저히 적은 실행시간이 걸리었다. 그럼에도 불구하고 제안 방법은 이들 방법중 가장 적은 실행시간이 걸리었다.

표 1: 실험 결과 수치

실험 영상	실험 방법	스레쉬홀딩 수	스레쉬홀딩 값	시간 (sec)
lena.jpg	Otsu	5	50, 93, 130, 170	23651
	Liao	5	50, 93, 130, 170	20.33
	Wu	5	43, 81, 141, 176	0.034
	Ours	5	43, 81, 141, 176	0.031
house.jpg	Otsu	4	83, 113, 158	438.1
	Liao	4	83, 113, 158	1.203
	Wu	4	67, 96, 147	0.026
	Ours	4	96, 147, 205	0.024
auroras.jpg	Otsu	4	22, 65, 137	455
	Liao	4	22, 65, 137	1.203
	Wu	4	64, 136, 188	0.029
	Ours	4	22, 64, 136	0.025

다른 실험 영상들에 대해서는 실험의 편리를 위해 4 단계 스톱스레쉬홀딩하여 검사하였다. 그림 2는 실험 영상들을 스톱스레쉬홀드에 의해 분할하고, 얻은 영상에 대하여 에지를 구한 영상이다. 분할 결과, 영상에서 볼 수 있는 것과 같이 "lena.jpg" 영상에 대해서는 그이 비슷한 결과를 얻었고, "lauroras.jpg"의 영상에 대해서는 Otsu의 방법과 제안 방법은 비교적 좋은 결과를 얻었으며, "house.jpg" 영상에서는 제안 방법이 가장 좋은 결과를 얻었다.

5장. 결론

영상처리에서 멀티스레쉬홀딩법을 이용하면 영상을 객체와 배경 두 레벨로 표시하는 것보다 영상의 특성을 더 잘 반영할 수 있다. 본 논문에서는 Otsu의 방법을 확장하여 멀티레벨 스톱스레쉬홀딩을 효율적으로 수행할 수 있는 간단한 방법을 제안하였다. 이 방법에서는 전체 영역을 대상으로 멀티레벨 스톱스레쉬홀드를 구하는 기존의 방법과는 달리, 간단한 스톱스레쉬홀딩법을 구간별 반복하여 적용하는 방법으로 고속화를 시도하였다. 실험결과 컴퓨터 처리상 가장 짧은 시간으로 효과적인 분할 처리를 수행하였다. 고속화된 멀티레벨 스톱스레쉬홀딩법을 패턴매칭과 얼굴인식을 포함한 영상인식에 도입하는 것이 급후의 연구과제이다.

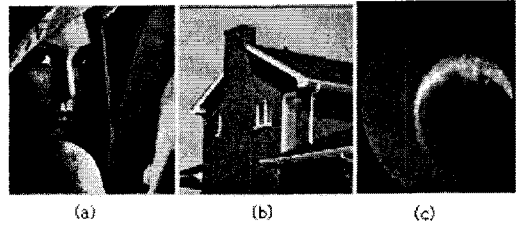


그림 1. 실험에 이용한 영상. (a) lena.jpg, (b) house.jpg, (c) lauroras.jpg

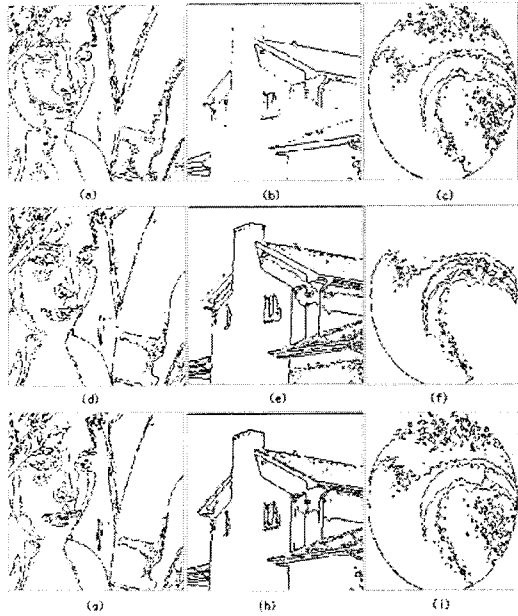


그림 2: Otsu방법, Wu방법, 제안 방법으로 그림 1의 영상에 대하여 분할하고 에지를 구한 결과

참고 문헌

- [1] B. Sankur, M. Sezgin, "Survey over image thresholding techniques and quantitative and quantitative performance evaluation", *J.Electron. Imaging*, 13(1), 2004, 146-165.
- [2] N.Otsu, "A threshold selection method from gray-level histogram", *IEEE Transactions on System Man Cybernetic*, SMC-9 (1), 1979, 62-66.
- [3] P.-S. Liao, T.-S. Chen, P.-C.Chung. "A Fast Algorithm for Multilevel Thresholding", *Journal of Information Science and Engineering*, 17, 2001, 713-727.
- [4] X.-J.Wu et al, "A fast recurring two-dimensional entropic thresholding algorithm", *Pattern Recognition*, 32, 1999, 2005-2061.