Mean Shift 알고리즘을 이용한 효율적인 문자 추출

정기철°* 김광인** 한정현*
*성균관대학교 전기· 전자· 컴퓨터 공학부
**한국 과학 기술원
{kjung, han}@ece.skku.ac.kr, kimki@ai.kaist.ac.kr

An Efficient Text Location using Mean Shift Algorithm

Keechul Jung°* Kwang In Kim** JungHyun Han*
*School of Electrical and Computer Engineering, Sungkyunkwan University
**Korea Advanced Institute of Science & Technology

요 약

영상내의 문자 정보는 색인에 필요한 유용한 정보를 제공하므로, 이를 이용한 멀티미디어 데이터의 인덱싱 기법이 최근 많이 연구되고 있다. 본 논문은 mean shift 알고리즘을 이용한 텍스춰 기반의 문자 영역 추출 방법을 제안한다. 다양한 크기와 모양의 문자에 적응성을 가지는 필터를 만들기 위해 신경망을 이용한다. 문자 영역의 위치와 크기는 문자 확률 영상상에서 mean shift 알고리즘을 이용하여, 국소 탐색만으로 별도의 후처리 과정 없이 기존의 문자 추출 방법보다 우수한 성능을 보인다.

1. 서 론

영상 내의 문자 추출은 비디오 인덱싱, 문서 구조 분석, 우편 영상 내의 주소 영역 추출, 자동차 번호판 추출 등의 다양한 응용 분야에서 활발히 연구되고 있다 [1-5].

기존의 문자 추출 방법은 크게 연결 성분 (connected component) 방법과 텍스춰 (texture) 방법으로 나눌 수 있다. 연결 성분 방법은 구현이 쉬운 반면, 문자 크기와 문자간의 거리 등에 대한 사전 지식이 필요하며, 비디오 영상 등의 잡음이 많은 저해상도 영상에는 적합하지 않다 [2]. 문자 영역의 텍스춰 성질을 이용한 gabor filter, wavelet, spatial variance 등의 방법 또한, 응용 분야의 문자의 모양이나 크기에 적합한 필터의 제작, 문자 추출 필터의 전역탐색 (e.g. exhaustive convolution)에 의한 속도 저하, 텍스춰 분석 후 문자 영역 획득을 위한 별도의 후처리 과정 수행 등의 단점이 있다 [1,4].

본 논문에서는 mean shift algorithm(MSA)을 이용한 텍스춰 기반의 문자 추출 방법을 제안한다. 문자 추출 필터의 생성을 위해 신경망을 사용하고, 문자 확률 영상 (text probability image)상에서 지정된 다수의 시작 노드 (node)에서 MSA를 수행함으로써 기존의 텍스춰 기반 방법의 전역 탐색으로 인한 속도 문제를 극복한다. 본 방법은 입력 영상의 전역 탐색과 명시적인 후처리 과정 없이 효율적으로 보다 정확한 문자 영역을 추출할 수 있다.

2. Mean Shift

본 절에서는 문자 영역 추출을 위한 MSA 기반 방법을 기술한다. 영상내의 문자 추출을 위해서 사용하는 MSA는 특징값의 확률 분포 (probability distribution) 상의 경사면을 반복 수행에 의해 탐색하는 방법으로, 최고점 (peak, mode)을 통계적 측면에서 효과적으로 찾아주며, 직관적이고 단순하며 잡음에 강한 면을 지닌다 [7]. MSA는 초기에 패턴 분류 등에서 많이 사용되었으며, 최근에서 영상 분할, 얼굴 추적과 같은 다양한 컴퓨터 비젼 관련 분야에서 사용되고 있다 [6]. 이러한 응용 분야에서는 주로 색상정보를 이용한 대상 물체의 확률 분포를 이용한다. 그러나 문자 영역은 색상 공간 상에서 다양한 분포를 나타내기 때문에, 본 논문에서는 신경망을 이용하여 만든 문자 확률 영상 상에서 mean shift를 반복 수행함으로써 문자 영역을 찾는다.

유클리디언 공간 X 상의 데이터 집합을 S, K를 커널 (kernel) 함수라고 할 때, 샘플 평균(sample mean, $m(x)$, $x \in X$)은 다음과 같다.

$$m(x) = \frac{\sum_{s \in S} K(\|s - x\|)s}{\sum_{s \in S} K(\|s - x\|)} \quad (1)$$

이때 $m(x)$-x를 mean shift라고 하고, MSA는 이러한 mean shift를 계산하고, 현재의 mean을 $m(x)$로 이동하는 과정을 반복하는 것을 말한다.

문자 영역 추출을 위해서는 문자 영역의 위치와 크기를 구해야 한다. 입력 영상의 신경망을 거쳐

나온 2차원 문자 확률 영상을 $I(x,y)$라 하면,

$$M_{00} = \sum_x \sum_y I(x,y) \quad (2)$$

$$M_{10} = \sum_x \sum_y xI(x,y) \quad (3)$$

$$M_{01} = \sum_x \sum_y yI(x,y) \quad (4)$$

와 같이 0,1차 모멘트를 정의할 수 있고, 문자 영역의 중심 좌표(sample mean position)는 flat kernel을 사용할 때

$m(x) = \frac{M_{10}}{M_{00}}$, $m(y) = \frac{M_{01}}{M_{00}}$로 설정할 수 있다 [6].

이렇게 구해진 중심 좌표 값에 따라서, 입력 영상에서의 문자 영역을 MSA를 이용하여 탐색하는데, mean shift 값이 정해진 임계값 (θ_x, θ_y) 이하일 때까지 반복 수행한다. 문자 영역의 크기를 구하기 위해, 2차 모멘트를 다음과 같이 정의할 때,

$$M_{20} = \sum_x \sum_y x^2 I(x,y), \quad M_{02} = \sum_x \sum_y y^2 I(x,y) \quad (5)$$

문자 영역의 폭(w)과 높이(l)는 다음과 같이 계산할 수 있다.

$$w = (1/\sqrt{2})\sqrt{(a+c) + \sqrt{b^2 + (a-c)^2}},$$

$$l = (1/\sqrt{2})\sqrt{(a+c) - \sqrt{b^2 + (a-c)^2}} \quad (6)$$

$$a = M_{20}/M_{00} - \mathrm{m(x)}^2, \quad b = 2(M_{20}/M_{00} - \mathrm{m(x)} \times m(y)),$$

$$c = M_{02}/M_{00} - \mathrm{m(y)}^2.$$

3. 문자 확률 영상

본 연구에서는 문자 확률 영상을 만들기 위하여 multi-layer perceptron (MLP)을 사용한다 [1,4]. 이는 MLP가 텍스춰 분류기로 사용될 수 있고, 또한 패턴 분류 문제에서 신경망의 출력 결과를 출력 클래스의 사후 확률값(*a posteriori*)으로 간주할 수 있다는 사실에 기반한다 [8].

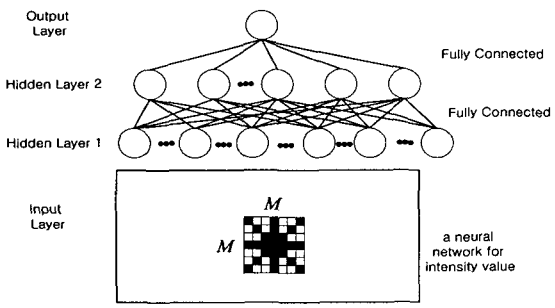


그림 1. 신경망의 개략적인 구조

신경망은 다층 퍼셉트론으로 2개의 은닉층, 1개의 출력 노드로 구성되며, 인접층의 노드들은 모두 연결되어있고 입력층은 입력 영상에서 M×M 크기의 윈도우 내의 특정 위치의 점들의 256-밝기값을 사용한다 (그림1). 신경망 웨이트는 백프로퍼게이션 알고리즘을 이용하여 학습하며, 비-문자 클래스의 학습 샘플을 효율적으로 구하기 위해서 bootstrap 방법을 사용한다 [4].

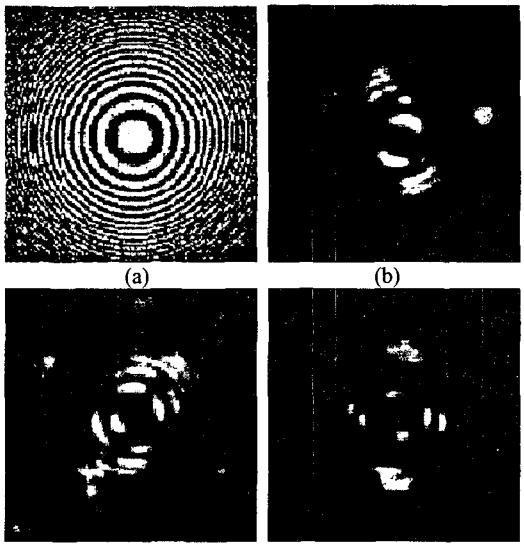


그림 2. 은닉 노드 출력: (a)입력 영상, (b-d)은닉 노드 출력

그림 2는 특정한 주파수와 방향을 가지는 영상에 대해 신경망의 은닉 노드들이 반응하는 모습을 보여준다 (높은 그레이 레벨이 높은 출력값에 대응). 그림 2(a)는 입력 영상이고, (b-d)는 첫 번째 은닉층 중 2개의 노드의 특성을 보여준다. 높은 출력을 내는 부분이 좁은 부분에 집중되어 있는 것으로 보아 특별한 주파수와 방향 선택성 (frequency and orientation selectivity)을 가짐을 알 수 있다. 그림 3(a)는 문자 클래스의 학습 데이터 예, (b)는 비-문자 클래스의 학습 데이터의 예, (c)는 부트스트랩 과정에 오 인식된 예, 그리고 (d)는 신경망에 비-문자 학습 데이터로 재사용되는 예이다.

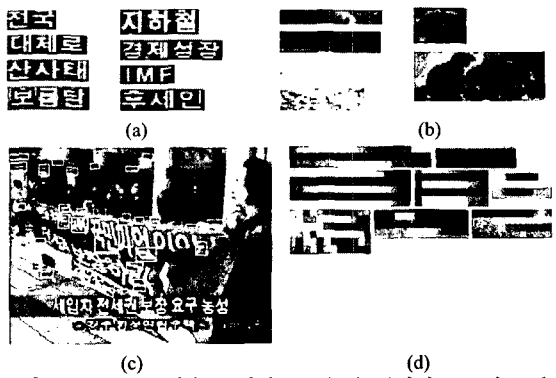


그림 3. 신경망의 학습 데이터: (a) 문자 데이터, (b) 비-문자 데이터, (c) 오인식 예, (d) 재학습되는 비-문자 데이터

4. 실험 및 결과

실험을 위하여 MBC 및 KBS의 뉴스 비디오 방송 화면을 사용하였으며, 총 12개의 비디오 클립 중 320×240 크기의 300개의 키 프레임을 추출하여, 문자 영역 추출 방법을 적용하였다. 신경망의 학습을 위하여 50개의 프레임 (57000개의 학습 패턴)을 사용, 나머지 프레임은 테스트에 사용되었다. 또한 영어권의 데이터를 실험하기 위해서 MoCA 프로젝트의 실험 영상을 사용하였다 [3].

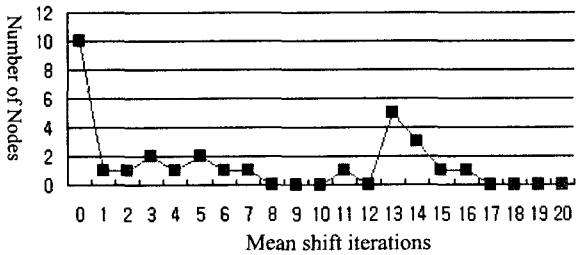
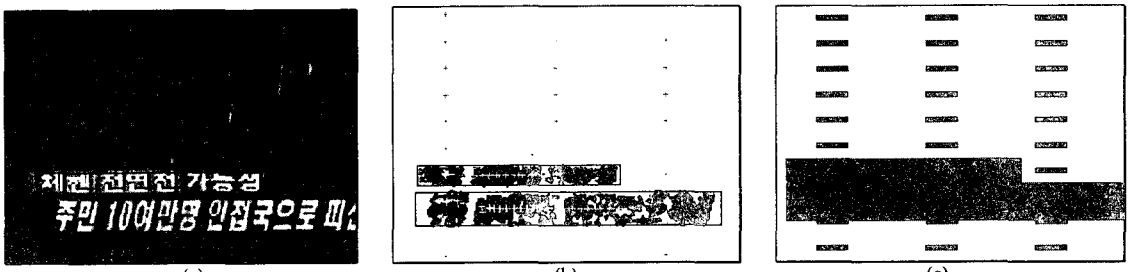
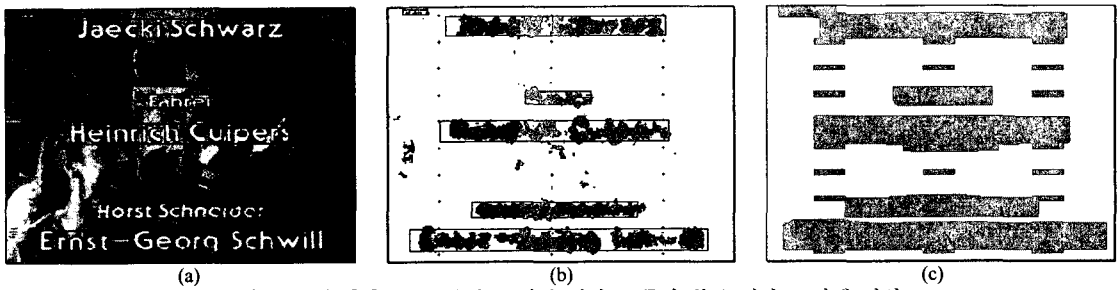


그림 4. Mean shift 알고리즘의 평균 반복 횟수



(a) (b) (c)

그림 5. 문자 추출 세부 단계: (a)입력 영상, (b)문자 확률 영상, (c)탐색 영역



(a) (b) (c)

그림 6. 문자 추출 세부 단계: (a)입력 영상, (b)문자 확률 영상, (c)탐색 영역

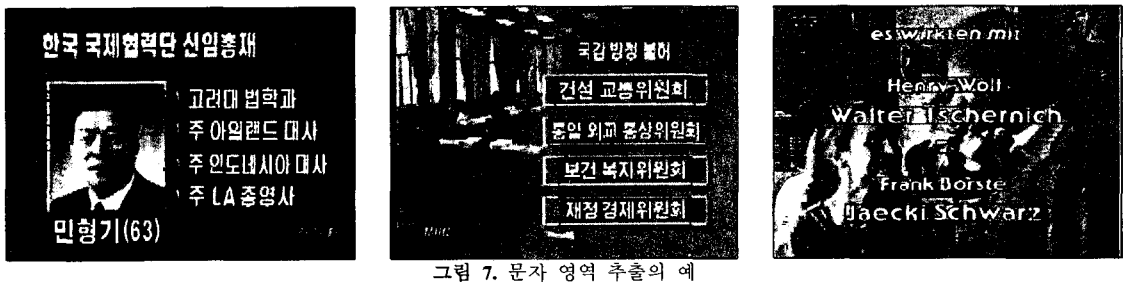
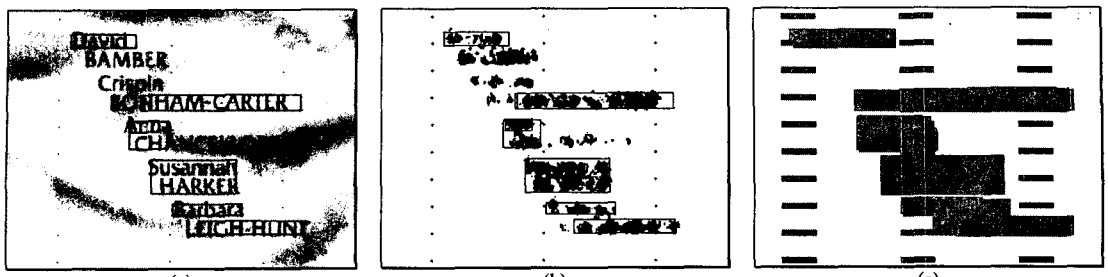


그림 7. 문자 영역 추출의 예

Mean들의 초기의 노드 위치는 *mean(x,y) = {(12+24×(x), 50+100×(y)), for 0 x<10 and 0 y<3}*와 같이 정했다. 본 실험에서는 입력 영상 폭의 1/3, 높이의 1/10 이상의 수평 문자열만을 추출 대상으로 한다. MSA의 종료 조건은 $\theta_x=2, \theta_y=1$로 하고, 각 노드의 mean shift의 값이 임계치 이하로 내려가면 수행을 멈춘다. 각 반복 수행 단계마다 윈도우의 폭은 10 pixels, 높이는 4 pixels 만큼 커진다.

그림 4의 평균 mean shift 반복 횟수 그래프에서 좌측의 피크는 비 문자 영역에 해당하며, 대부분의 노드가 13번 반복 후에 수렴함을 알 수 있다.

그림 5와 6은 문자 추출의 세부 단계이다. (a)의 입력 영상의 사각형은 문자 영역을 나타내며, 그림 (b)는 MLP의 출력 결과인 문자 확률 영상으로써, 회색의 작은 십자가 기호는 초기 노드의 위치와 연속된 반복 수행중의 mean의 위치를 나타내며, 그림



(a) (b) (c)

그림 8. 문자 추출 오류: (a)입력 영상, (b)문자 확률 영상, (c)탐색 영역

(c)는 최종적으로 검색된 부분을 나타내는데, 문자 영역을 검출하기 위해서 전체 영상을 탐색할 필요 없이 제한된 부분만을 탐색함으로써, 전체 시스템의 수행 시간을 30% 정도로 줄일 수 있었다. 그림 7은 문자 추출 예로써 비-한글 문자에 대해서도 적용 가능함을 보인다. 제안된 문자 추출 방법이 수행 시간과 성능면에서 기존의 방법들에 비해 우수함을 알 수 있다 (표 1).

표 1. 수행 시간과 성능 비교

	Mean shift	Full scan [4]	연결 성분 [2]
시간(sec.)	1.2	4.0	1.2
추출률(%)	93.2	92.2	73.8

그림 8은 문자 추출의 오류를 보인다. 초기의 노드들의 위치를 설정함에 있어서, 추출 대상 문자열의 길이와 높이를 가정하였기 때문에, 보다 짧은 문자열들은 추출할 수 없는 경우가 있다.

5. 결 론

본 논문에서는 mean shift 알고리즘을 이용한 텍스춰 기반의 문자 추출 방법을 제안하였다. 제안한 방법은 다음의 몇 가지 점에서 다른 방법들과 구분된다: (1) 신경망을 이용하여 다양한 응용 분야에서의 필터를 자동으로 구현할 수 있다; (2) mean shift 알고리즘을 문자 검출에 사용함으로써, 영상 전체 영역을 탐색하지 않고도, 더욱 정확한 문자 영역을 구할 수 있다; (3) 별도의 잡음 제거나 프로젝션과 같은 명시적인 후처리 과정이 필요 없다. 향후의 연구 과제로, 압축된 동영상에서의 수행 속도 개선을 위한 문자 추적, 더욱 자연스러운 모양의 문자열의 추출 등이 있다.

참고 문헌

[1] Anil K. Jain, and Kalle Karu, "Learning Texture Discrimination Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 2, 1996.

[2] E.Y. Kim, K.Jung, K.Y.Jeong, and H.J.Kim, "Automatic Text Region Extraction Using Cluster-based Templates," *ICAPRDT*, pp. 418-421, 2000.

[3] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg, "Video Abstracting," *Communications of the ACM*, December, Vol. 40, No. 12, 1997.

[4] Jeong, K. Y., Jung, K., Kim, E. Y., and Kim, H. J., "Neural Network-based Text Locating for news video Indexing," *Proceedings of International Conference of Image Processing*, 1999.

[5] Huiping Li, David Doerman, and Omid Kia, "Automatic Text Detection and Tracking in Digital Video," *IEEE Transactions on Image Processing*, Vol. 9, No. 1, pp.147-156, January 2000,

[6] Gary R. Bradski and Vadim Pisarevsky, "Intel's Computer Vision Library: Application in calibration, stereo, segmentation, tracking, gesture, face and object recognition," *Proceedings of CVPR*, Vol. 2, pp. 796 –797, 2000.

[7] Yizong Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, August 1995.

[8] M.D.Richard and R.P.Lippmann, "Neural network classifiers estimates Bayesian a posteriori probabilities," *Neural Computation*, No. 3, pp.461-483, 1991.