

디지털 비디오내의 문자영역 추출 및 추적

장재식 김은이 김항준
경북대학교 컴퓨터공학과 인공지능연구실
jschang@ailab.knu.ac.kr

Text Region Extraction And Tracking In Digital Video

Jea-Sig Chang Eun-Yi Kim Hang-Joon Kim
Dept. of Computer Engineering, Kyungpook National University

요 약

영상내의 문자정보는 색인에 필요한 유용한 정보를 제공하기 때문에 이를 이용한 멀티미디어 데이터의 색인 기법이 최근 많이 연구되고 있다. 본 논문에서는 칼라동영상에서 실시간으로 문자를 추출하고 추적하는 방법을 제안하였다. 제안된 방법은 연결성분(connected component)방법을 이용한 문자추출 모듈과 SSD(a Sum Of Squared Difference)를 이용한 문자추적모듈로 구성되어져 있다. 실제 TV영상에 대하여 제안된 방법을 테스트 해본 결과 빠른 문자추출과 추적시간을 가졌다.

1. 서론

최근 영상처리 문자 인식 등의 기술발전과 더불어 영상내의 문자 자동 추출에 대한 연구가 진행되고 있다. 내용기반의 비디오 색인을 위해서는 비디오프레임에 있는 글자를 인식하는 것이 필요하다[3]. 이를 위해서는 글자의 위치를 아는 것이 선행되어야 하기 때문에 최근까지 많은 방법들이 제안되어 왔다. 하지만 양질의 문서의 자동문자인식과는 달리 동영상에서는 영상 내 문자의 해상도와 배경영역의 복잡도에 상당한 영향을 받기 때문에 문자영역의 추출과 자동 문자 인식은 상당히 어렵고 많은 계산시간이 필요한 문제로 인식되고있다.

본 논문에서는 칼라동영상에서 실시간으로 문자를 추출하고 추적하는 방법을 제안하였다. 본 논문에서 제안된 방법은 연결성분방법을 이용한 문자추출모듈과 SSD(a Sum Of Squared Difference)를 이용한 문자추적모듈로 구성되어져 있다. 문자추출모듈은 다섯 프레임마다 실행되어 이미지에 새로 나타나는 문자를

추출한다. 추출된 영역들에 대해 문자추적모듈이 적용된다[1]. 본 논문의 구성은 다음과 같다. 2절에서는 문자를 추출하기 위한 모듈에 대해 살펴보고, 3절에서는 추출된 문자를 추적하기 위한 모듈에 대해 설명한다. 실험결과는 4절에서 보여지고,마지막에 결론이 이어진다.

2. 문자추출

2.1 개요

제안된 시스템은 크게 문자추출 모듈과 문자추적 모듈로 나누어진다. 그림 1은 문자추출모듈의 개요를 보여준다. 디지털 비디오로부터 문자를 추출하기 위해서 입력영상은 연결 성분들로 분할된다.

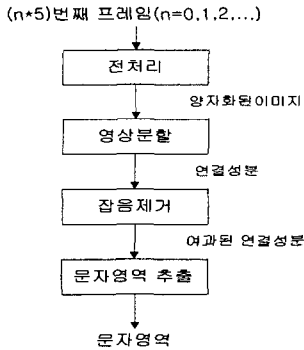


그림 1. 문자 추출 모듈

대부분의 칼라영상분할 방법은 좋은 결과를 얻기 위해 많은 시간이 소요되고 정확한 파라메타의 추정을 필요로 한다. 따라서 계산시간을 감소시키기 위해서 입력칼라영상은 그레이 레벨 영상으로 전환되고, 그 다음에 양자화된다[4]. 양자화된 영상에 대해서 연결 성분방법이 적용된다. 그리고 그 결과들은 글자가 아닌 성분들을 잡음 제거 모듈에서 제거하고 남겨진 성분들에 대해 프로파일분석을 함으로써 글자영역이 얻어진다.

2.2 영상분할

영상분할은 기본적인 연결성분 알고리즘으로 실행된다. 연결성분은 3*3 이웃들 상에서 동일한 색상을 가진 픽셀들을 모아서 생성된다.

그림 2(a), (b), (c)에서 각각 입력영상과 양자화된 영상, 그리고 영상분할에 의해 생성된 연결 성분들을 보여준다.

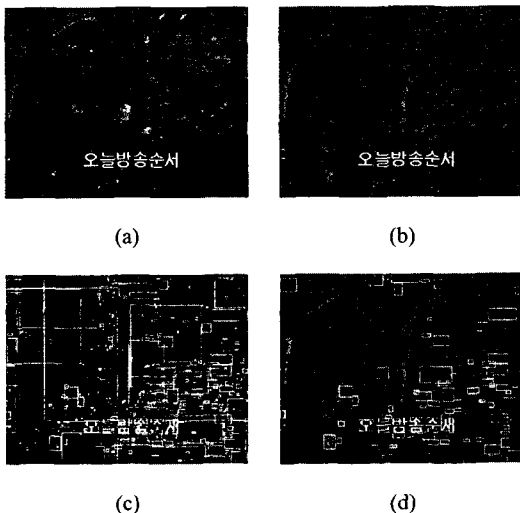


그림 2. 영상분할 결과

- (a) 원 영상
- (b) 양자화된 영상
- (c) 연결 성분들
- (d) 여과된 연결 성분들

2.3 잡음제거

생성된 연결 성분들은 크기, 색깔, 성분 간의 위치 정보를 이용해서 여과된다. 큰 연결 성분들은 큰 그래픽영상을 나타내고, 작은 연결 성분들은 그래픽영상이나 잡음을 나타낸다고 가정한다. 다시 말해서, 어떤 성분들은 문자라고 보기에는 너무 크거나 작다. 다음 조건들을 만족하지않는 성분들은 제거된다.

- 1) 연결성분의 크기가 15이상이고 800미만 이어야 한다.
- 2) 연결성분의 폭이 5이상이거나 높이가 5이상 이어야 한다.
- 3) 연결성분의 폭과 높이가 각각 입력영상의 폭과 높이의 절반보다 작아야 한다.
- 4) 연결성분의 폭과 높이의 비가 3이하이고 1/3이상 이어야 한다.

그림2(d)에서 보는 바와 같이 잡음성분 여과 과정을 거쳤음에도 불구하고 여전히 글자가 아닌 성분들이 남아있다. 만약 너무 작거나 큰 크기의 성분을 제거한다면 많은 글자 성분들 또한 제거될 것이다.

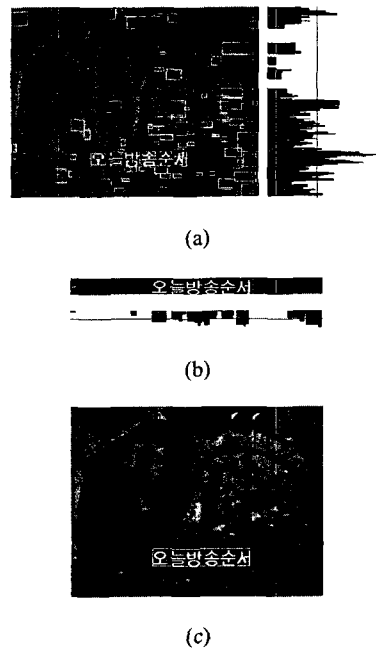


그림 3. 문자영역 검출

- (a) x축으로 프로젝션한 결과
- (b) y축으로 프로젝션한 예
- (c) 최종적으로 추출된 결과

2.4 문자영역검출

대부분의 문자열들은 수평으로 놓여져 있다. 반면 어떤 것 들은 수직으로 또는 비선형적으로 놓여져 있다. 같은 영상에서 다른 배열을 가진 문자열들이 존재 할 수 있다. 본 논문에서는 문자열들이 수평 하게 놓여져 있다고 가정한다. 이 과정동안 분할된 영역의 프로파일분석을 통해 문자 영역들이 구분된다.

우선 x-축으로 선택된 영역의 픽셀수를 더하여 얻은 히스토그램이 threshold값 보다 큰 영역을 선택한다(그림3a).다시 선택된 영역들을 y-축으로 히스토그램을 구하여 값이 threshold값보다 큰 영역을 선택한다(그림3b)[2]. Threshold값은 실험적으로 적당한 값을 사용하였다. 선택된 영역들 중 같은 수평선상에서 근접하게 위치한 영역들을 하나의 영역으로 합친다.

그림 3(c)에서 추출된 문자영역을 보여주고 있다.

3. 문자영역의 추적

3.1 개요

문자검출 모듈에서 문자 영역들이 검출되었을 때 각각의 영역에 대하여 추적모듈이 실행된다. 연속된 두 프레임간의 영역들의 유사성을 비교하기 위해서 일반적인 SSD(a Sum Of Squared Difference)를 이용하였다.

3.2 SSD을 기반으로 한 문자추적

현재프레임에서 이전프레임의 문자영역에 대응되는 영역을 찾기 위해서 SSD값을 사용하였다[1]. SSD값이 가장 작은 영역을 대응되는 영역으로 선택하였다. 비교된 값은 그레이레벨의 값을 사용하였다. SSD값은 다음식으로 주어진다.

$$\epsilon = \iint_W [J(D\bar{x} + \vec{d}) - I(\bar{x})]^2 dx \tag{1}$$

- D : 2*2 deformation 행렬
- \vec{d} : 2*1 displacement 벡터
- J : 현재프레임
- I : 이전 프레임의 문자영역
- W : 탐색영역

여기서 프레임간의 움직임이 매우 작다고 가정하면 deformation 행렬을 무시함으로써 위의 식을 간단히 할 수 있다.

$$\epsilon = \iint_W [J(\bar{x} + \vec{d}) - I(\bar{x})]^2 dx \tag{2}$$

4. 실험

제안된 시스템은 Pentium II 400MHz,128MB RAM 시스템에서 visual C++ 6.0을 사용하여 구현되었다. 실험에서 사용된 영상은 실제 TV에서 캡처된 300*230 칼라 이미지이다.



그림 4. 문자 추출 실험 결과

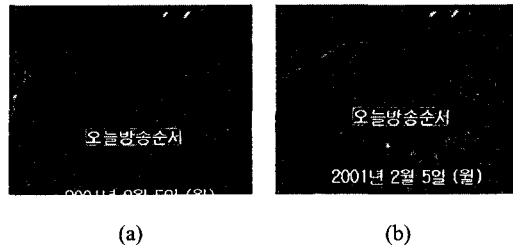


그림 5. 문자 추적의 예

제안된 방법의 효율성을 입증하기위해서 실제 동영상 sequence에 대해서 실험이 이루어졌다. 제안된 시스템은 크게 문자추출 모듈과 문자추적 모듈로 구성되어 있다. 따라서 실제로 문자추출 과 문자추적 각각의 모듈의 정확성에 대한 고찰이 수행되었다.

그림 4는 제안된 시스템을 사용하여 추출된 문자의 결과를 보여준다. 그림4에서 보여지듯이, 제안된 방법으로 다양한 크기의 문자영역을 추출할 수 있다.

그림 5는 문자영역의 추적결과를 보여준다. 그림 5(a)에서 검출된 문자영역은 아래에서 위로 서서히 이동하고 있다. 그때 그림(b)는 검출된 영역의 움직임이 정확히 추적되고 있음을 보여준다.

제안된 방법의 정확도는 Accuracy와 Miss-detection-rate를 이용하여 평가되었다. Accuracy와 Miss-detection-rate는 각각 식(3),(4)에 정의하였다.

$$A_i(t) = \frac{num\{R_i(t)\}}{num\{G_i(t)\}} * 100 \tag{3}$$

$$M_i(t) = \left| \frac{num\{R_i(t)\} - num\{G_i(t)\}}{num\{G_i(t)\}} \right| * 100, \tag{4}$$

$R(t) = \{ R_i(t) \mid i = 1, 2, \dots, \text{문자영역의 수} \}$
 : t번째 프레임에서 제안된 방법으로 추출된 문자영역

$G(t) = \{ G_i(t) \mid i = 1, 2, \dots, \text{문자영역의 수} \}$
 :t번째 프레임에서 수동으로 추출한 문자영역으로 일반적으로 ground truth 라고 불리워진다.

num(●): 해당 문자영역의 픽셀수

제안된 방법은 94%의 Accuracy 와 9%의 Miss-detection-rate를 가진다. 그리고 문자 추출에는 프레임당 평균 0.14초의 추출시간이 소요되었고 문자추적은 영역의 크기에 따라 많은 차이를 보였으나 영역당 평균 0.06초가 소요되었다. 따라서 제안된 방법은 real-application에 적용할수 있음이 보여진다.

5. 결론

본 논문에서는 연속영상에서 실시간으로 문자를 추출하고 추적하는 방법을 제안하였다. 제안된 시스템은 연속성분 방법을 이용한 문자추출 모듈과 SSD(a Sum of Squared Difference)를 사용한 문자 추적모듈로 구성되어 있다. 다양한 디지털 영상에 대해 실험해 본 결과 제안된 방법의 효율성이 검증되었다.

향후 연구는 더 많은 영상에 대해서 제안된 방법을 테스트 하고 문자의 실시간 추출을 위한 연구를 계속해 나갈 것이다.

6. 참고 문헌

- [1]Huiping Li,David Doermann,and Omid Kia,“Automatic Text Detection and Tracking in Digital Video,”IEEE Transaction on Image Processing,Vol.9,No.1,pp147-156 January 2000.
- [2]K.Sobottka,H.Kronenberg,T.Perroud,H.Bunke,“Text extraction from colored book and journal covers,” IJDAR(2000) 2: 163-176
- [3]정기철,김항준,“텍스춰를 이용한 영상 내의 문자 추출,”문자인식 워크샵 pp49-58 2000년 12월
- [4]Anil K. Jain and Bin Yu, “Automatic Text Location in Images and Video Frames”, *Pattern Recognition*, Vol. 31, No. 12, pp. 2055-2076, 1998.