

프레임간의 상대적인 차이를 이용한 비디오의 셔트 검출 기법

정인식, 권오진

세종대학교 전자공학과

전화 : 02-3408-3828 / 핸드폰 : 016-396-5843

Shot Boundary Detection Using Relative Difference between Two Frames

In-Shik Jeong, Oh-Jin Kwon
Dept. of Electronics Engineering, Sejong University
E-mail : always@image.sejong.ac.kr

Abstract

This paper proposes a unique shot boundary detection algorithm for the video indexing and/or browsing. Conventional methods based on the frame differences and the histogram differences are improved. Instead of using absolute frame differences, block by block based relative frame differences are employed. Frame adaptive thresholding values are also employed for the better detection. For the cases that the frame differences are not enough to detect the shot boundary, histogram differences are selectively applied. Experimental results show that the proposed algorithm reduces both the "false positive" errors and the "false negative" errors especially for the videos of dynamic local and/or global motions.

I. 서론

최근 비디오 콘텐츠의 대용량화 및 활용도 증가 추세에 따라 효과적인 비디오 인덱싱과 브라우징의 중요성이 증대되고 있다. 이와 함께 비디오 인덱싱과 브라우징을 위해서 비디오의 내용물을 분석하여 의미가 있는 몇 장의 이미지로 표현하기 위해 셔트(shot)의 경계를

검출하는 기법은 각광을 받고 있으며 활발한 연구가 이루어지고 있다.

본 논문의 II장에서는 지금까지 제시된 방법들의 대략적인 설명과 이들 중에서 대표적인 몇 가지 방법을 소개하고 이러한 방법들이 가지고 있는 문제점들을 파악한다. III장에서는 새로운 알고리즘을 설명한다. IV장에서는 기존의 방법들과 대비하여 개선된 실험결과를 제시하고, 결론을 V장에 서술한다.

II. 지금까지 제시된 방법들

비디오의 셔트를 검출하는 방법들을 살펴보면, 가장 쉽게 사용할 수 있는 방법으로써 두 인접한 프레임에서 화소간의 차이를 구하고 이 값이 임계값보다 큰 화소들의 개수를 계산한 후 이 개수가 일정 임계값보다 많은 경우 셔트의 경계로 검출하는 방법이 널리 사용되어 왔다. 또한 이 방법은 카메라 모션에 대하여 민감하게 작용하여 검출 오류를 범하는 경우가 많이 발생하는데 이 단점을 보완하기 위하여 두 프레임 간의 화소 차이를 구하기 전에 3x3 average filter를 사용하여 카메라 모션을 상쇄시킨 후 화소간의 차이를 구하는 방법이 제기되기도 하였다[1].

이 방법보다 개선된 방법으로는 흑백 또는 칼라 영상의 히스토그램을 이용하는 방법이 대표적인 방법인데 지금까지 가장 광범위하게 사용되고 있는 방법이다.

이것은 각각의 프레임마다 히스토그램을 구하고 그 히스토그램의 차이가 임계값을 넘으면 서트의 경계로 보는 방법이다. 이 방법을 전체영상에 적용하여 사용하기도 하지만 전체를 4x4 서브-블록으로 분할한 후 그 서브-블록들에 대하여 적용함으로써 로컬 모션에 대해서 성능을 개선하고자 하는 시도가 제시되기도 하였다 [2].

그 외에도 두 프레임간에 각각 에지의 변화 정도를 측정하여 서트의 경계를 검출하는 방법이 있다[3]. 하지만 이 방법은 정확한 에지를 구하는 것이 어렵고 정확한 에지를 구한다 해도 전체의 비디오물에 대하여 에지를 구하는 것은 엄청난 연산을 요구하는 단점이 있다[4].

지금까지의 예들은 모두 압축이 되어있지 않은 비디오에 대하여 연산을 수행하는 방법인 반면, MPEG 압축 기법으로 압축되어 있는 상태에서 DCT계수만을 이용하여 검출하는 방법이 있고[5], 전체의 DCT계수에서 다른 부분은 제외하고 DC계수만을 사용하여 서트의 경계를 찾아내는 방법도 제시되었다[6]. DC계수만을 사용하는 방법은 속도 면에선 탁월한 성능을 보인다.

본 논문에서 제시된 알고리즘을 다음의 대표적인 알고리즘들과 비교한다.

· Histogram intersection을 이용한 방식[7]:

$$D_h = \sum_j \frac{\min(H_{t+1}(j), H_t(j))}{\max(H_{t+1}(j), H_t(j))} \quad (1)$$

$H_t(j)$ 는 t 번째 프레임의 히스토그램에서 j 번째 레벨로 정의된다.

· Yakimovsky likelihood ratio test를 이용한 방식[7]:

$$D_y = \left(\frac{\sigma_0^2}{\sigma_t^2} \right) \left(\frac{\sigma_0^2}{\sigma_{t+1}^2} \right) \quad (2)$$

σ_t^2 은 t 번째 프레임의 분산값이다. 반면, σ_0^2

은 t 와 $t+1$ 번째 프레임의 히스토그램을 구하고 두 히스토그램이 중첩되는 부분의 화소들의 분산값이다.

· Kolmogorov-Smirnov test를 이용한 방식[7]:

$$D_k = \max |C_t(x) - C_{t+1}(x)| \quad (3)$$

$C_t(x)$ 는 t 번째 프레임의 화소값들의 누적 분포이다.

Histogram intersection과 Yakimovsky likelihood ratio test는 결과 값이 임계값 이하로 나올 때, Kolmogorov-Smirnov test는 결과 값이 임계값 이상일 때 서트의 경계로 검출한다.

III. 서트의 검출

지금까지 제시되어온 알고리즘들은 II장에서 살펴 본 바와 같이 인접한 프레임 간의 화소값의 차이를 이용하거나 히스토그램의 차이를 이용하는 방법이 가장 널리 사용되어 왔다. 이들은 서트 내에서의 화면 변화는 적고 서트와 서트 사이의 경계 부분에서는 화면의 변화가 급격한 비디오물에 대해서 상당한 검출 능력을 갖는 반면, 한 서트 내에서 글로벌 및 로컬 모션이 많을 경우 검출하지 않아야 할 서트들을 잘못 검출하는 "false positive" 오류를 자주 발생시키며 서트와 서트 사이의 경계에서 화면의 변화가 적은 상태로 변화되는 서트들은 구분해내지 못하는 "false negative" 오류를 자주 발생시키는 단점이 있다.

본 논문에서는 1) 인접한 프레임간의 화소값의 차이를 전체 프레임에 대하여 일괄적으로 계산하여 비교하지 않고 블록 단위로 비교하고, 2) 이와 함께 프레임간 히스토그램의 차이를 적절히 배합하여 사용함과 동시에, 3) 프레임별로 서트 검출 여부를 판단하기 위해 사용되는 임계값을 가변적으로 적용하는 독특한 방법을 사용함으로써 기존 알고리즘에 비해 "false positive" 오류와 "false negative" 오류를 현저하게 줄일 수 있는 새로운 알고리즘을 제시한다.

서트의 경계를 찾는 알고리즘은 대략 다음과 같다.

- 1) 전체 이미지를 MxM 서브-블록으로 분할한다.
- 2) t 와 $t+1$ 번째 프레임에서 각 서브-블록의 화소들간의 차이의 합을 다음 식을 사용하여 구한다.

$$B_t(k) = \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} (I_t(i, j, k) - I_{t+1}(i, j, k)) \quad (4)$$

여기에서 $I_t(i, j, k)$ 는 t 번째 프레임에서 k 번째 블록의 i, j 번째 화소이다.

- 3) 2)의 결과인 $B_t(k)$ 로부터 각 프레임별로 다음 식을 사용하여 R_t 를 구한다.

$$R_t = \sum_k |B_t(k)| \quad (5)$$

- 4) t 번째 프레임의 전 2개 프레임과 후 2개 프레임에 대하여 같은 연산을 수행하여 다음의 R_{MAX} 를 계산한다.

$$R_{MAX} = \max(R_{t-2}, R_{t-1}, R_{t+1}, R_{t+2}) \quad (6)$$

- 5) 임의 임계값 h_2 에 대하여 $R_t \geq h_2 * R_{MAX}$ 이면 서트의 경계로 결정한다.

- 6) 다른 임계값 h_1 에 대하여 $R_t < h_1 * R_{MAX}$ 이면 서트의 경계가 아닌 것으로 결정한다.

프레임간의 상대적인 차이를 이용한 비디오의 서트 검출 기법

7) $h_1 * R_{MAX} \leq R_t < h_2 * R_{MAX}$ 인 경우에는 다음과 같이 인접 프레임 간 히스토그램의 차이를 계산한다.

$$D_t = \sum_j |H_t(j) - H_{t+1}(j)| \quad (7)$$

8) $D_{MAX} = \max(D_{t-1}, D_{t+1})$ 값을 계산하고 임의 임계값 h_3 에 대하여 $D_t \geq h_3 * D_{MAX}$ 이면 서트의 경계로 결정하고 그렇지 않으면 서트의 경계가 아닌 것으로 결정한다.

과정 2)의 $B_t(k)$ 계산식에서 화소값의 차이를 구할 때 절대값을 취하지 않았음에 유의하여야 한다. 이는 두 프레임간의 화소값의 차이를 살펴보면 t 프레임이 서트의 경계인 경우에는 화소값의 차이가 대부분 한 블록 내에서 양수만의 값을 갖거나 음수만의 값을 갖는 경향이 있는 반면, 로컬 또는 글로벌 모션이 있는 프레임의 경우에는 화소값의 차이가 양수 값과 음수 값을 동시에 블록 내에 갖게 되는 현상이 고려되었다. 이러한 현상은 특히 로컬 또는 글로벌 모션에 의하여 변화가 있는 영역이 한 프레임 내에서 단위 면적 당 화소값의 변화가 많은 "texture" 영역인 경우 더욱 두드러지게 나타났다. 이렇게 함으로써 절대값을 사용하는 경우보다 $B_t(k)$ 값이 로컬 및 글로벌 모션에 대하여서는 양수 값과 음수 값이 상쇄되어 서트의 경계 대비 상대적으로 작은 $|B_t(k)|$ 값을 보이게 된다. 이는 결과적으로 로컬 및 글로벌 모션이 있는 프레임들과 서트의 경계에 해당하는 프레임들을 과정 3)의 R_t 계산 시에 더욱 확연히 구분하는 결과를 보이게 되어 "false positive" 오류를 줄이는 효과로 나타났다.

과정 5)와 6)에서는 절대적인 임계값을 사용하지 않고 상대적 임계값인 R_{MAX} 를 사용하여 성능을 향상시켰다. 절대적인 임계값을 적용하면 로컬 및 글로벌 모션이 발생할 경우 두 프레임간의 차이가 비교적 크기 때문에 다수의 "false positive" 오류가 발생할 확률이 높다. 그러나 상대적 임계값을 사용하면 주위의 값들을 기준으로 서트의 경계 여부를 판단하므로 로컬 및 글로벌 모션에서 발생하는 "false positive" 오류를 대부분 줄일 수 있다.

과정 7)은 R_t 만으로는 서트의 경계 여부를 판단하기가 어려운 경우, 추가로 히스토그램의 차이를 분석하는 과정이다. 이런 경우는 주로 비디오가 전체적으로 글로벌 모션이 많은 경우 자주 나타나게 되는데 이때 글로벌 모션에 해당하는 프레임들과 서트의 경계에 해당하는 프레임들을 구분하기 위한 과정이다.

IV. 실험 결과

본 장에서는 II장에서 소개된 기존의 기법과 III장에서 소개된 제안된 기법을 비교한다. 다양한 종류의 비디오물에 대하여 우리가 제안한 방식과 histogram intersection을 이용한 방식, Yakimovsky likelihood ratio test를 이용한 방식, Kolmogorov-Smirnov test를 이용한 방식을 비교한다. 표 1은 그 결과를 보이고 있다.

실험에 사용된 비디오물은 s1~s6으로 표시하였다. 비디오물 s1~s3는 상대적으로 로컬 및 글로벌 모션이 적은 정적인 비디오물로 구성하였고, s4~s6은 로컬 및 글로벌 모션이 많은 비디오물로 구성하였다. s1은 총 36개의 서트를 가진 심야 토크쇼 내용으로서 모션이 거의 없고 화질도 비교적 좋은 비디오물이다. s2는 상업용 광고물로서 10개의 서트로 구성되어 있으며, 특이한 화면으로는 내용 중 객체의 인공적인 흔들림이 일정 시간 진행되는 화면이 포함되어 있다. s3 역시 자동차 광고용 비디오로서 14개의 서트로 구성되어 있고 내용중 진행이 매우 빠른 서트의 변화가 있다. s4는 항공모함에서 비행기가 이착륙하는 장면으로서 50개의 서트로 구성되어 있다. 비행기 이륙 시에 발생하는 연기에 의해서 "false positive" 오류가 발생할 확률이 매우 높은 비디오물이다. s5는 51개의 서트로 이루어진 "트루라이즈" 영화 예고편이다. 매우 다양하게 변화하는 로컬 및 글로벌 모션이 전반에 포함되어 있고 폭발 및 사격 장면에서는 순간적으로 화면이 밝아져 다수의 "false positive" 오류가 발생하는 요인이 된다. s6은 "리셀웨폰4" 영화의 예고편으로 144개의 서트로 구성되어 있다. s6은 s5와 같이 매우 다양한 모션이 포함되어 있고, 또한 전체적인 카메라의 흔들림과 그림 1에 예시된 화면처럼 등장인물의 이름이 소개될 때 글씨가 전면으로 순간적으로 확대되었다가 축소되는 효과를 사용하였기 때문에 서트를 검출하기가 매우 까다로운 비디오물이다.

제안된 기법의 성능은 "A"로 표시하였고 비교된 기법들인 histogram intersection을 이용한 방식, Yakimovsky likelihood ratio test를 이용한 방식, Kolmogorov-Smirnov test를 이용한 방식은 각각 "H", "Y", "K"로 표시하였다.

제안된 기법을 구현할 때 사용된 임계값은 다양한 종류의 비디오물에 대하여 수행한 실험을 통하여 최적화된 값을 사용하였다. 블록의 크기인 $M \times M$ 값은 2×2 , 4×4 , 8×8 , 16×16 4가지 경우에 대하여 테스트하였는데, 4×4 가 최적의 결과를 보였다. h_1, h_2, h_3 값들은 $h_1 = 1.5$ $h_2 = 4.0$ $h_3 = 2.0$ 결정되었다.

비교된 기법들 또한 임계값을 사용하는데 이들의 값

들은 실험에 사용된 6개의 비디오물에 대하여 최적화된 값들을 사용하였다.

표 1에서 “f”는 사용된 각 비디오의 전체 프레임의 수를 표시한다. “r”은 실제로 존재하는 서트의 수와 이중 제대로 검출된 서트 수의 비율(recall)을 표시한다. “p”는 각 기법이 검출한 총 서트 수와 이중 제대로 검출한 서트 수의 비율(precision)을 표시한다.

	f	H		Y		K		A	
		r	p	r	p	r	p	r	p
s1	7762	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.97
s2	449	0.88	1.0	1.0	0.81	0.88	1.0	1.0	1.0
s3	452	0.92	1.0	0.76	1.0	0.76	0.83	1.0	0.92
s4	1304	0.83	0.65	0.93	0.77	0.73	0.69	0.97	0.96
s5	1451	0.92	0.54	0.86	0.63	0.7	0.46	0.9	0.95
s6	4037	0.77	0.6	0.73	0.56	0.78	0.48	0.99	0.94

표 1. 기존의 방법과 제안한 방법의 비교

표 1에서 보는 바와 같이 제안된 알고리즘이 s4~s6의 비디오물에서 기존의 방법보다 매우 좋은 결과를 보인다는 것을 알 수 있다. 특히 s6 비디오의 경우 그림 1의 장면에서는 비교된 모든 알고리즘들이 “false positive” 오류를 발생시키는 반면, 제안된 알고리즘은 이 오류를 발생시키지 않는 결과를 보였다.

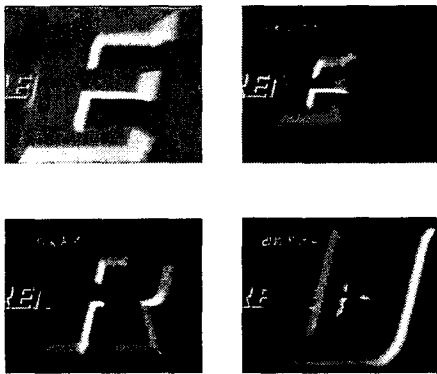


그림 1. 기존 방법들의 “false positive” 오류 프레임

V. 결론

본 논문에서는 서트의 경계를 검출하는 새로운 방법을 제시하였다. 기존에 사용되어 온 인접 프레임간의 화소간의 차이를 이용하는 방법과 히스토그램의 차이

를 이용하는 방법은 로컬 및 글로벌 모션에서 “false positive” 오류를 다수 발생시키는 단점을 가지고 있다.

그러나 본 논문에서는 1) 화소간의 차이를 이용하는 방법을 블록단위로 적용하고, 2) 화소간의 차를 구할 때 절대적인 차이 대신에 상대적인 차이를 사용하고, 3) 프레임 별로 가변적인 임계값을 사용하였으며, 4) 화소간의 차이만으로는 서트의 경계 검출이 어려운 경우 히스토그램의 차이를 선별적으로 적용시킴으로써, 실험결과에서 보듯이 대부분의 “false positive” 오류를 제거할 수 있었다.

향후에는 우리가 제시한 방법을 fade 및 dissolve에 적용할 수 있는 추가 연구가 요구된다.

참고문헌

- [1] Zhang, H.J., Kankanhalli, A., and Smoliar, S.W., “Automatic partitioning of Full-motion Video”, *Multimedia Systems* Vol. 1, No. 1, pp. 10-28, 1993
- [2] Nagasaka, A. and Tanaka, Y., “Automatic Video Indexing and Full-video Search for Object Appearances”, in *Visual Database Systems II*, Knuth, E., Wegner, L., Editors, Elsevier Science Publishers, pp.113-127, 1992
- [3] R. Zabih, J. Miller, and K. Mai. “A Feature-Based Algorithm for Detecting and Classifying Scene Breaks” *Proceedings of ACM Multimedia 95*, San Francisco, CA pp.189-200, 1995.
- [4] J. Canny. “A computational Approach to Edge Detection” *IEEE Transaction of Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp.34-43, 1986.
- [5] F. Arman, A. Hsu, and M.-Y. Chiu, “Image processing on encoded video sequences,” *Multimedia Systems*, Vol. 1, no. 5, pp. 211-219, 1994.
- [6] B.-L. Yeo and B. Liu, “Rapid scene analysis on compressed video” *IEEE Transaction of Circuits, System, Video Technology*, Vol. 5, pp.533-544, 1995
- [7] R. Brunelli, O. Mich, and C. M. Modena, “A survey on the automatic indexing of video data,” *Journal Visual Communication and Image Representation*, vol. 10, no. 2, pp.78-112, June 1999