

# 압축 비디오에서 비디오 구조화를 고려한 장면 전환 검출

°이재승\*, \*김강욱, \*황찬식  
\*경북대학교 전자·전기공학부

## Scene Change Detection on Compressed Video Considering Video Organization

°Jae Seung Lee\*, Kang Wook Kim\*, Chan Sik Hwang\*  
\*School of Electronic & Electrical Eng., Kyungpook National University  
jaeseung@palgong.knu.ac.kr

### 요약문

최근 정보통신의 급속한 발달로 비디오의 내용기반 검색은 많은 응용분야에서 중요성이 증가하고 있다. 자동적인 비디오 검색에 있어서 장면전환의 검출은 없어서는 안될 필수적인 과정이다. 그래서, 압축 영역이나 비압축 영역에서의 장면전환검출 기법들이 많이 제안되었다. 특히, 비디오가 대용량화됨에 따라 압축 영역에서의 검출 기법의 연구가 활발히 진행되고 있다. 본 논문에서는 압축비디오에서 I-프레임의 DC 영상과 B-프레임의 매크로 블록 타입만을 이용하여 정확한 컷의 위치를 찾아내고자 한다. 그리고, 비디오 구조화의 수행에 적합한 성능과 정보를 얻을 수 있는 방법을 제안한다.

### I. 서론

장면전환검출은 연속되는 두 개 샷(shot)사이의 경계인 컷(cut)을 검출하는 것으로 검색이나 색인을 위한 비디오 구조화 및 해석의 첫 번째 단계로서 매우 중요한 과정이다. 장면전환검출 기법은 크게 압축영역에서의 방법과 비 압축영역에서의 방법으로 나눌 수 있다. 비 압축영역에서 장면전환을 찾는 방법에는 DCT 계수나 움직임벡터를 이용하는 방법[1-2]이 있으며 축소된 DC 영상을 이용한 방법[3-4]도 많이 사용되고 있다. 대부분의 비디오 데이터는 주로 압축된 상태로 저장 및 전송되므로 압축 영역에서 장면전환을 검출하는 것이 계산량 및 저장공간의 절약 등 많은 장점이 있다.

그러나, 압축영역에서 사용되는 방법이라 할지라도

기존의 방법들은 실제로 상당히 긴 비디오 데이터에 적용하기에는 수행시간이 길며 임계값에 많이 의존하므로 비디오의 종류에 따라 검출결과가 달라지는 단점들이 있다. 그리고, 축소된 DC 영상을 이용한 방법은 비디오 구조화 과정에서 DC 영상을 다시 사용할 수 있는 장점이 있지만 모든 프레임에 대해서 DC 영상을 추출하는 것은 계산량이 많아지고 특히 B-프레임의 경우는 정확한 DC 영상을 추출하기 어렵다는 단점들이 있다.

본 논문에서는 인트라 프레임의 DC 영상만 복호화하여 컷이 존재하는 GOP를 먼저 찾고 GOP 내부는 B-프레임의 매크로 블록 타입만을 이용해서 정확한 컷의 위치를 찾아내는 새로운 장면전환검출 방법을 제안한다.

### II. 기존의 장면전환검출 방법

압축된 비디오에서의 장면전환검출 기법을 살펴보면 크게 DC 영상을 이용한 방법과 압축된 비디오로부터 간단히 얻을 수 있는 DCT 계수나 움직임 벡터 정보 등을 이용하는 방법으로 나눌 수 있다.

전자의 경우는 주로 비디오 구조화와 관련해서 많이 사용되고 있고, 후자는 컷 검출만을 다루는 경우에 많이 사용되어 진다.

먼저 DC 영상을 추출하는 방법을 살펴보고 대표적인 두 가지 장면전환검출 방법에 대해 알아보도록 한다.

#### 2.1 DC 영상 추출 방법

MPEG으로 압축된 비디오 스트림으로부터 계산량과

화질의 열화를 최소화하면서 DC 영상을 추출하는 방법 [5]이 많이 연구되어왔다.

I-프레임의 경우는 DCT 계수의 DC 값으로부터 간단히 DC 영상을 만들 수 있지만 P/B-프레임은 움직임 보상이 수행되므로 계산량이 많아지게 된다. 그림 1은 MPEG에서 움직임 보상의 일반적인 형태를 나타낸다.

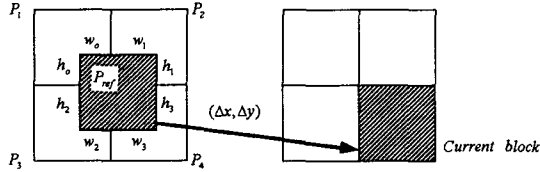


Fig 1. Extraction of DC image in P/B frame  
그림 1. P/B-프레임에서 DC 영상 추출

현재 블록이 참조할 블록( $P_{ref}$ )은  $P_1 \sim P_4$  블록의 전체 또는 일부분과 겹치게 된다.  $h_i$  와  $w_i$ 는 각각 겹쳐진 부분의 수직, 수평 화소 수를 나타낸다. 현재 블록의 평균 값을 구하는 계산량을 줄이기 위해 제안된 근사식을 식 (1)에 나타내었다.

$$[DCT(P_{ref})]_{00} \approx \sum_{i=1}^4 \frac{h_i w_i}{64} [DCT(P_i)]_{00} \quad (1)$$

$\frac{h_i w_i}{64}$ 은 참조 프레임의 각 블록에서 겹쳐진 부분의 비율을 나타낸다.

## 2.2 DC 영상을 이용한 방법

비디오 구조화 방안의 하나인 클러스터링 수행을 위해 장면전환을 검출하는 경우 압축된 비디오로부터 DC 영상을 추출하는 방법이 주로 사용된다. DC 영상을 추출함으로써 비디오 구조화 과정에서 필요한 대표프레임 추출이나 샷간의 유사성을 계산하기 용이하기 때문이다.

추출된 DC 영상간의 화소값의 차나 히스토그램의 차를 이용하여 프레임간의 변화 정도를 나타내는 FFD (Frame-to-Frame Difference)를 구하고, 슬라이딩 윈도우를 FFD 값 위로 이동시키면서 윈도우 내의 최대값과 두 번째 큰 값 사이의 차가 임계값이상이면 컷으로 판단하게 된다. 1초 이내에 두 번 이상의 장면전환이 발생하지 않는다고 가정하고, 윈도우의 크기는 프레임율 (일반적으로 30 frames/sec)와 동일한 값을 사용한다.

DC 영상의 경우 실제 영상을 얻기 위해 전체를 디코딩할 경우에 비해 훨씬 적은 계산량이 소요되고, 슬라이딩 윈도우를 이용함으로써 카메라플래시가 있는 장면이나 급격한 움직임이 있는 장면을 컷으로 잘못 판단하

는 경우가 없도록 해준다.

그러나, P/B-프레임에서 특히 B-프레임의 DC 영상 추출시 실제 영상과의 에러를 최소화하더라도 움직임이 큰 영상에서는 에러가 커져서 FFD 값이 실제와 달리 크게 증가하기 때문에 정확한 컷 검출이 어려워진다. 슬라이딩 윈도우의 경우, 1초 이내에 두 번 이상의 장면전환이 일어나지 않는다고 가정하지만 뮤직비디오에서는 그러한 경우가 자주 나오기 때문에 올바른 컷 검출이 이루어지지 못한다.

## 2.3 DCT 계수와 MB의 타입을 이용한 방법

DCT 계수나 움직임 보상 정보를 이용하면 다른 방법에 비해 수행속도를 빠르게 할 수 있기 때문에 많은 연구들이 행해지고 있다.

I-프레임에서는 DCT 계수들의 값을 이용해서 I-프레임간의 차( $d(i,j)$ )를, P-프레임에서는 인트라로 코딩되는 MB의 비율( $T_p$ )을, B-프레임에서는 참조 프레임이 이전 프레임인 MB의 비율( $T_i$ )과 이후 프레임인 MB의 비율( $T_b$ )을 구해서 각각에 주어진 임계값과 비교하여 컷을 판단한다.

그림 2는 이러한 방법들이 공통적으로 사용하는 컷 검출 알고리즘을 나타낸다.

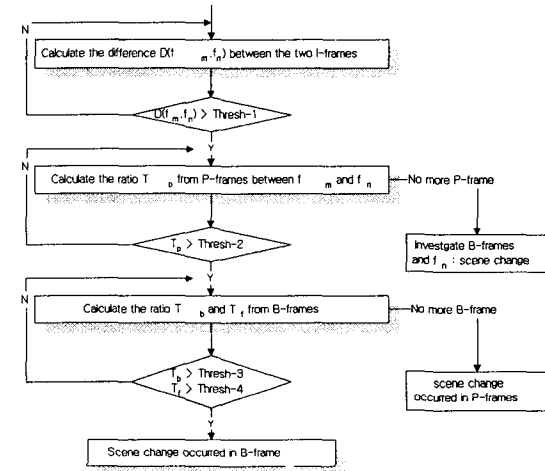


Fig 2. Cut detection using DCT coefficients of I-frame and MB type of P/B-frames  
그림 2. DCT 계수와 MB의 타입을 이용한 장면전환검출

2.2절에 제시된 방법에 비해 수행시간은 상당히 줄일 수 있지만 그림 2에 나타나 있듯이 사용되는 임계값이 4개나 되기 때문에 수행하는 비디오의 종류에 따라 각 임계값을 적절히 조절해 주지 않으면 적절한 장면전환

검출이 이루어지지 못한다.

### 2.4 실제 영상을 이용한 방법

완전하게 디코딩된 실제 프레임들을 이용[6]하면 다른 방법들에 비해 훨씬 더 정확한 컷 검출이 이루어질 수 있다. 그러나, 동영상 정보들이 점점 대용량화 되어 가고 있고 종류도 다양해지고 있어서 공간적·시간적 문제들로 인해 비디오 구조화에 적용하기는 어렵다.

## III. I-프레임의 DC 영상과 B-프레임의 MB 타입을 이용한 장면전환검출 방법

### 3.1 장면전환이 있는 GOP(Group of Picture) 검출

I-프레임에서 DC 영상을 얻는 방법은 간단하다. DCT 계수의 DC 값  $c(0,0)$ 은 화소값  $f(i,j)$ 와 식 (1)과 같은 관계가 있다.

$$c(0,0) = \frac{1}{8} \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \quad (2)$$

식 (2)로부터 DC 값은 8x8 블록의 평균 휘도값의 8배가 됨을 알 수 있다. 따라서, DC 영상은 실제 영상의 블록 평균값  $c(0,0)/8$ 로 만들 수 있다. 이렇게 추출된 DC 영상들간의 내용 변화를 나타내는 값으로 식(3)~(5)에 나타낸 식을 이용한다.

$$I'_n = \sum_{j=1}^J Lum\{f(j,n)\} \quad (3)$$

$$I'_m = \sum_{k=1}^K Lum\{f(m,k)\} \quad (4)$$

$$d_p(i,j) = \frac{1}{255(J+K)} \left\{ \frac{1}{J} \sum_{n=1}^J |I'(i)_n - I'(j)_n| + \frac{1}{K} \sum_{m=1}^K |I'(i)_m - I'(j)_m| \right\} \quad (5)$$

$Lum(f(x,y))$ 는 주어진 영상  $f(x,y)$ ,  $x=1,2,\dots,J$ ,  $y=1,2,\dots,K$ ,의 x행, y열에 있는 화소의 휘도값을 나타낸다. 즉, 식 (3)와 (4)은 각각 주어진 영상의 n번째 열과 m번째 행에 있는 화소값들의 합을 계산하는 식이다. 식 (5)는 앞의 두 식에서 계산된 값들의 차의 절대값을 정규화한 값으로 두 프레임간의 차를 나타낸다.

$d(i,j)$ 값이 모두 계산되면 이 값들의 평균(m)과 분산( $\sigma^2$ )을 구하여  $m + \sigma^2$ 을 임계값으로 하여 이보다 큰 값을 가지는 GOP는 장면전환이 발생한 GOP로 판단한다. 이러한 임계값을 사용하면 움직임이 큰 장면이 있는 GOP를 컷이 존재하는 것으로 잘못 판단할 수 있지만 3.2절에서 제시하는 방법을 사용하여 정확한 컷을 검출

할 수 있게된다.

이 과정에서 얻어진 DC 영상은 이후의 비디오 구조화 과정에서 대표 프레임이나 샷 사이의 유사성을 구하기 위해 사용되어진다.

### 3.2 GOP 내에서의 장면전환검출

장면전환이 있다고 판단된 GOP내에서 정확한 컷을 찾기 위해서 B-프레임에서 움직임 보상을 위한 참조 프레임으로 이전 프레임을 사용하는 MB의 비율( $N_b$ )과 이후 프레임을 사용하는 MB의 비율( $N_f$ )만을 이용한다. 컷이 존재할 수 있는 위치는 3가지 경우로 발생하며 경우에 따른 판별 조건은 그림 3에 나타나 있다. 만약 GOP 내에서 조건을 만족하는 경우가 없다면 장면전환이 일어나지 않은 것으로 판단한다.

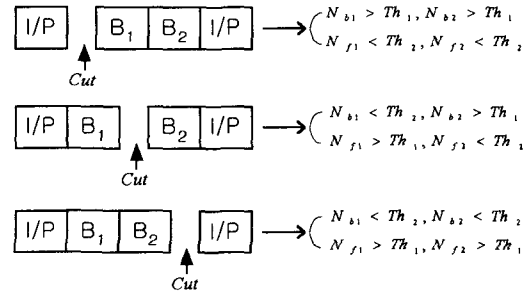


Fig 3. Decision of scene change using MB types of B-frame

그림 3. B-프레임의 MB 타입을 이용한 컷 판별

임계값으로 사용되는  $Th_1$ 과  $Th_2$ 는 각각 70%와 15%를 사용하였을 때 대부분의 비디오에서 가장 좋은 성능을 나타내었다.

### 3.3 성능 평가 방법

실험 결과를 비교하기 위한 성능 평가 방법으로는 일반적으로 많이 사용되어지는 Precision과 Recall을 사용하였다. 식(6)와 (7)은 Precision과 Recall의 계산방법을 나타낸다.

$$Recall = \frac{\text{num. of detected cuts}}{\text{num. of detected cuts} + \text{num. of missing cuts}} \quad (6)$$

$$Precision = \frac{\text{num. of detected cuts}}{\text{num. of detected cuts} + \text{num. of false alarms}} \quad (7)$$

Recall 값은 검출하지 못한 컷이 얼마나 많은가를 나타내고, Precision 값은 컷으로 오인하고 잘못 검출한 컷이 얼마나 많은가를 나타내는 척도이다.

비디오 구조화의 클러스터링을 수행하게 되면 유사한 내용의 샷을 하나의 논리적인 유닛으로 묶어주게 된다. 그러므로, 컷이 아닌 부분을 컷으로 잘못 검출하더라도 클러스터링에 의해 같은 scene으로 묶일 수 있다. 하지만 검출하지 못한 컷이 있을 경우 클러스터링 과정에서 샷 사이의 차를 계산할 때 잘못된 값이 나오게 되어 적절한 비디오 구조화가 이루어지지 못한다. 따라서, Recall 값이 너무 낮아지면 비디오 구조화가 제대로 수행될 수 없다.

#### IV. 실험 결과 및 고찰

##### 4.1 실험 영상

실험에 사용된 스트림은 디스플레이 시간이 6분 55초인 뮤직비디오로써 총 12391개의 프레임으로 구성되어 있다. 그리고 압축된 비디오는 초당 30 프레임 (GOP=15)의 프레임 율을 가지며 주관적으로 확인된 실제 컷은 총 256 개임이 확인되었다.

##### 4.2 실험 결과

제안한 방법과 비교할 기존의 방법으로는 DC 영상을 이용한 방법을 사용한다. 비디오 구조화와 관련해서 컷을 검출하는 대부분의 방법이 DC 영상을 이용하고 있기 때문이다.

두 방법을 이용한 컷 검출 결과는 표 1과 같다.

Table 1. Results of scene change detection

표 1. 장면전환 검출 결과

	Detected	Missed	False Alarm	Recall	Precision
Previous Method	246	48	38	83.67 %	86.62 %
Proposed Method	243	26	2	90.33 %	99.18 %

표 1로부터 기존의 방법에 비해 Recall과 Precision 성능이 크게 향상됨을 볼 수 있다. 기존의 DC 영상만을 이용한 방법은 P/B-프레임에서 에러에 의해 검출되지 못한 컷들이 제안한 방법에서 검출되며 1초 내에 두 번 이상의 컷이 존재할 경우에도 검출할 수 있는 장점이 있다.

#### V. 결 론

영상 정보가 대용량화됨에 따라 압축된 비디오에서 장면전환을 검출하는 방법들이 많이 제안되었다. 특히 비디오 구조화를 수행하기 위하여 DC 영상을 추출하는

기법이 주로 사용되어지지만 기존의 방법들은 움직임이 많고 장면전환이 급격히 일어나는 비디오에서는 잘못된 컷 검출이 높아지고, 많은 임계값에 의존하므로 다양한 종류의 비디오에 적용하기 어려운 문제점들이 있다.

본 논문에서 제안한 방법은 I-프레임의 DC 영상만을 추출함으로써 수행시간을 줄이고, P/B-프레임에서 주로 발생하는 검출 에러를 없애 기존 방법보다 더 높은 Recall과 Precision 성능을 얻을 수 있었다.

이러한 결과가 유용하게 사용되기 위해서는 I-프레임에서 추출된 DC 영상과 검출된 컷을 이용한 클러스터링 기법에 관한 연구가 있어야 할 것이다.

#### 참고문헌

- [1] N. Gamaz, X. Huang, S. Panchanathan, "Scene Change Detection on MPEG Domain", Proceedings of Image Analysis and Interpretation, IEEE, 1998, pp. 12-17.
- [2] A.M Dawood, M.Ghanbari, "Clear Scene Cut Detection Directly from MPEG Bit Streams", Image Processing and its Applications, IEE, 1999, pp. 285-289
- [3] B.L. Yeo, B. Liu, "Rapid Scene Analysis on Compressed Video", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, No. 6, 1995, pp. 533-544.
- [4] H. Jiang, A. Helal, A.K. Elmagarmid, A. Joshi, "Scene Change Detection Techniques for Video Database System", Multimedia Systems, Vol. 6, No. 3, 1998, pp. 186-195.
- [5] Junehwa Song, B.L Yeo, "Fast Extraction of Spatially Reduced Image Sequences from MPEG-2 Compressed Video", IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No. 7, 1999, pp. 1100-1114.
- [6] G. Lupatini, C. Saraceno, R. Leonardi, "Scene Break Detection: a Comparison", Proceedings of Continuous-Media Databases and Applications, IEEE, 1998, pp. 34-41.