

# MPEG으로 압축된 비디오에서 최소 복호화에 의한 빠른 장면전환검출 알고리즘

김 강 욱<sup>†</sup> · 이 재 승<sup>††</sup> · 김 종 훈<sup>†††</sup> · 황 찬 식<sup>††††</sup>

## 요 약

비디오 데이터의 장면전환검출은 연속되는 두 개 샷(shot) 사이의 경계인 컷(cut)을 검출하는 것으로 비디오 검색과 색인에 있어서 매우 중요한 과정이다. 본 논문에서는 압축된 비디오에서 최소 복호화에 의한 빠르고 정확한 장면전환검출 알고리즘을 제안한다. 제안한 방법은 먼저 I-프레임간의 DC 영상을 비교해서 장면전환을 포함한 GOP(group of picture)를 찾고 GOP 내에서 장면전환이 발생한 정확한 위치는 B-프레임에서 매크로블록의 부호화 형태에 관한 정보를 이용해서 찾는다. 실험결과 제안한 방법이 모든 DC 영상을 사용한 기존의 방법보다 Precision과 Recall 측면에서 더 좋은 성능을 나타내었다. 그리고 제안한 방법은 빠르고 정확하며 작은 저장공간을 필요로 한다는 장점을 가진다.

## Fast Scene Change Detection Algorithm in MPEG Compressed Video by Minimal Decoding

Kang Wook Kim<sup>†</sup> · Jae Seung Lee<sup>††</sup> · Jong Hoon Kim<sup>†††</sup> · Chan Sik Hwang<sup>††††</sup>

## ABSTRACT

A scene change detection which involves finding a cut between two consecutive shots is an important step for video indexing and retrieval. This paper proposes an algorithm for fast and accurate detection of abrupt scene changes in an MPEG compressed domain with minimal decoding requirements and computational effort. The proposed method compares two successive DC images of I-frames for finding the GOP (group of picture) which contain a scene change and uses macroblock-coded type information contained in B-frames to detect the exact frame where the scene change occurred. The experiment results demonstrate that the proposed algorithm has better detection performance, such as precision and recall rate, than the existing method using all DC images. The algorithm has the advantage of speed, simplicity and accuracy. In addition, it requires less amount of storage.

키워드 : 장면전환검출(scene change detection), MPEG, DC 영상(DC image)

## 1. 서 론

최근 멀티미디어 기술의 발달과 다양한 하드웨어의 발전에 힘입어 여러 형태의 데이터들이 기하급수적으로 생겨나고 있을 뿐만 아니라 디지털화됨으로 인해서 멀티미디어 정보에 대한 이용과 가공이 용이해지고 있다. 하지만, 대용량화에 따른 효율적인 데이터의 저장 및 관리가 필요하게 되었으며, 더불어 사용자의 요구를 만족시킬 수 있는 데이터의 검색이나 색인이 중요한 문제로 대두되고 있다. 특히 비디오, 영화, TV 프로그램 같은 동영상 정보들이 MPEG (moving picture expert group)과 같은 압축 기술에 의해

VOD(video on demand) 형태로 저장 및 제공되고 있다. 따라서 비디오 데이터의 내용을 기반으로 한 검색, 색인, 구조화 등에 관한 많은 연구가 진행되고 있다. 일반적인 비디오 데이터의 색인 및 검색을 위해서는 먼저 비디오 시퀀스를 시간적으로 동일한 성질을 갖는 부분 즉 샷(shot)으로 분할해야 하며 그 다음은 분할된 샷에서 몇 개의 대표되는 프레임 즉 키 프레임(key frame)을 뽑아낸다. 그리고 마지막으로 키 프레임의 유사성을 이용해서 분할된 샷의 클러스터링(clustering)이나 구조화(organization)를 수행하게 된다[1, 2].

비디오 데이터의 장면전환검출은 연속되는 두 개 샷(shot) 사이의 경계인 컷(cut)을 검출하는 것으로 비디오 검색과 색인의 첫 번째 단계로서 매우 중요한 역할을 한다. 장면전환검출 기법은 크게 비 압축영역에서의 방법과 압축영역에서의 방법으로 나눌 수 있다. 비 압축영역에서의 방

† 정 회 원 : 삼성전자 무선통신연구소 선임연구원  
 †† 정 회 원 : 한국항공우주연구원 연구원  
 ††† 정 회 원 : 동양대학교 정보통신공학부 교수  
 †††† 정 회 원 : 경북대학교 전기전자공학부 교수  
 논문접수 : 2001년 9월 21일, 심사완료 : 2002년 4월 25일

법은 실제 복호화된 프레임을 사용하므로 정확한 장면전환을 검출할 수 있지만 복호화 시 소요되는 시간상의 문제점과 복호화된 프레임을 저장하기 위한 공간상의 문제점 등이 있다. 뿐만 아니라 대부분의 비디오 데이터는 주로 압축된 상태로 저장 및 전송되므로 압축 영역에서 장면전환을 검출하는 것이 계산량 및 저장공간 측면에서 더욱 효율적이라고 할 수 있다. 일반적으로 압축된 영역에서 대부분의 장면전환검출은 주로 DC 영상을 이용한 방법들이 많이 제안되었다[3, 4, 14]. 이 방법은 실제 영상보다 축소된 크기를 갖는 DC 영상의 화소값이나 히스토그램 등을 이용하므로 실제 영상을 얻기 위해 전체 프레임을 복호화하는 경우보다 시간을 절약할 수 있고 계산량도 감소하게 된다. 그러나, 압축된 비디오에서 근사식을 이용해 DC 영상을 빠르게 추출해서 사용한다 할지라도 모든 프레임의 DC 영상을 추출하는 것은 여전히 계산량 및 수행시간 측면에서 문제점이 많다. 더구나 P, B-프레임의 경우에는 추출된 DC 영상의 화질 열화 때문에 실제 프레임이 갖는 영상 특징을 많이 잃어버려 잘못된 장면전환을 검출할 수 있다.

본 논문에서는 압축된 비디오에서 최소 복호화에 의한 빠르고 정확한 장면전환검출 알고리즘을 제안한다. 제안한 방법은 I-프레임의 DC 영상만을 이용해서 장면전환을 포함한 GOP(group of picture)를 찾고 B-프레임에서 매크로블록의 부호화 형태만을 이용해서 GOP 내에서 장면전환이 발생한 정확한 위치를 찾는다. 제안한 방법은 P, B-프레임의 DC 영상을 모두 이용하는 기존 방법보다 간단하고 P, B-프레임의 DC 영상에서 발생하는 화질 열화를 고려할 필요가 없다. 본 논문의 구성은 2장에서 비 압축영역에서의 장면전환검출 기법을 논하고 3장에서 압축영역에서의 장면전환검출 기법 및 DC 영상을 이용한 방법에 관한 서술, 그리고 4장에서는 새로운 장면전환검출 기법을 제안한다. 5장에서는 실험을 통해 기존방법과 제안한 방법과의 성능 비교 및 검토를 하고 6장에 결론을 맺는다.

## 2. 비 압축영역에서 장면전환검출

비 압축영역에서의 장면전환검출 기법들은 픽셀간 차[5], 통계적 차[6], 히스토그램 비교[7]를 주로 사용한다. 픽셀차를 이용한 방법은 직접 프레임간 차를 계산하거나 값이 변한 픽셀의 개수를 조사해서 그 값을 미리 정한 임계치와 비교해서 장면전환여부를 판단한다. 그러나 이 방법의 문제점은 카메라의 움직임이 빠르거나 개체가 이동하는 경우에 매우 민감하다는 것이다. 통계적인 차를 이용한 방법은 픽셀간의 차를 특정 영역으로 확장한 방법이다. 즉 프레임을 블록으로 나누어서 블록내의 밝기값에 2차 통계적 특성을 근거로 비교하는 방법으로 변한 블록의 개수를 계산해서 이 값이 어떤 임계치 이상일 때 장면전환으로 검출이 된다. 이러한 통계적 방법은 잡음에 강한 장점이 있지만 비교적

많은 연산을 필요로 한다. 히스토그램 비교 방법은 샷 경계 검출을 위해 일반적으로 가장 많이 사용되는 방법으로 컬러나 밝기 분포가 영상내의 내용변화에 따라 프레임간에서 서로 다르다는 특성을 이용한 방법이다. 이 방법은 이웃하는 프레임의 컬러나 밝기 히스토그램의 차를 계산해서 비디오 시퀀스에서 장면전환을 검출한다. 히스토그램 기반의 방법들은 정확성과 계산량 간에 trade-off 관계가 있으며 물체의 움직임이 있는 경우에도 장면전환을 비교적 잘 검출하지만 카메라의 움직임에는 비교적 둔하다는 단점이 있다.

## 3. 압축영역에서 장면전환검출

프레임에서 장면전환이 일어날 때 컬러나 밝기가 이전 프레임과는 매우 큰 차이가 나타나게 된다. 인트라로 부호화된 프레임에서  $8 \times 8$  DCT(discrete cosine transform) 블록의 계수들은 공간영역에서  $8 \times 8$  블록의 휘도와 색차 성분과 관련이 있다. 그러므로, 압축영역에서 DCT 계수는 두 프레임간 휘도와 색차 신호의 차이를 나타내는 이상적 파라미터이다. 특히 DC 계수는  $8 \times 8$  블록내의 모든 픽셀값의 평균이 되므로 DC 계수를 사용하면 장면전환검출 과정을 간략화시킬 수 있다. 일반적으로 같은 샷 내에 있는 인트라 프레임의 DC 계수 값의 평균은 거의 비슷하며 샷 경계에서 크게 변한다.

MPEG 비디오에서 장면전환검출을 위한 다른 중요한 파라미터로는 세 가지 매크로블록 타입에 관한 정보들이며 P-프레임과 그것의 참조 프레임간의 관계 혹은 B-프레임과 참조 프레임간의 관계를 나타낸다. FMB(forward macroblock), BMB(backward macroblock), BMB(bidirectional macroblock)의 수는 현재 프레임과 그것의 과거, 미래 또는 과거와 미래 프레임들에 얼마나 가까운가를 나타낸다. 이런 매크로블록의 수가 많아질수록 현재 프레임이 참조 프레임과 더 유사하다는 것을 의미하며 압축영역에서 이러한 정보는 쉽게 얻을 수가 있다. 최근에 MPEG으로 압축된 데이터로부터 부분 복호화에 근거한 여러 장면전환검출 알고리즘이 제안되고 있으며 사용되는 정보 형태에 따라 크게 세 가지로 분류된다. DCT 계수에 근거한 알고리즘, 움직임 벡터에 근거한 알고리즘, 둘 모두를 이용한 방법 등이 있다.

### 3.1 DCT 계수를 이용한 장면전환검출

Arman et al. [8]은 I-프레임에서 DCT 계수를 이용한 장면전환검출 기법을 제안하였다. 압축된 각 프레임  $f_m$ 에 대해서 연속된 R개의 영역에서 B 개의 블록들을 뽑아낸 다음 각 블록에서 임의로 분포된 여러 개의 계수들을 선택한다. 각 블록에서 선택한 계수들의 집합을 연결함으로써 벡터  $\bar{v}_{f_m}$ 이 만들어진다. 벡터  $\bar{v}_{f_m}$ 은 변환 영역에서  $f_m$ 을 의미한다. 이렇게 구한 벡터간의 정규화 된 내적(inner pro-

duct)을 프레임  $f_m$ 과 프레임  $f_n$ 간의 유사도의 척도로 사용한다.

Zhang et al. [9]은 DCT 계수를 사용해서 Motion-JPEG에서 장면전환을 검출하는 방법으로 개별 비교방법을 제안하였다. 이 방법은 두 프레임  $f_m, f_n$ 에서  $(i, j)$  블록의 정규화 된 절대차를 유사도의 척도로 사용하였다.  $D(f_m, f_n, i, j)$ 이 임계치 이상일 때  $(i, j)$  블록은 변화된 블록으로 판단하며 이 변화된 블록의 개수가 주어진 임계치를 넘을 때 장면전환이 일어났다고 한다. MPEG 시퀀스에서 I-프레임의 시간적 해상도는 보통 12~15 프레임 크기의 GOP를 가지므로 이 방법은 장면전환의 정확한 위치를 찾아낼 수는 없다.

Yeo et al. [10]은 Motion-JPEG이나 MPEG에서 12~15 프레임 크기가 아닌 한 프레임간 장면전환도 검출할 수 있는 DC 계수 기반의 알고리즘을 제안하였다. I-프레임에서 DC 계수의 추출뿐만 아니라 P와 B-프레임의 DC 계수를 근사적으로 처리하여 추출하였는데 이러한 형태로 얻어진 시퀀스를 DC 시퀀스라고 한다. 두 프레임간 유사도의 척도는 DC 영상  $f_m^{DC}$ 와  $f_n^{DC}$ 간 차의 합을 이용한다.

### 3.2 움직임 벡터를 이용한 장면전환검출

P-프레임과 B-프레임은 매크로 블록 어리의 DCT 계수와 움직임 벡터 정보를 포함하고 있으므로 이러한 파라미터는 예측된 프레임이 그것의 참조 프레임과 얼마나 비슷한가를 직접적으로 나타낸다. Liu et al. [11]은 에러신호의 에너지와 P와 B-프레임의 움직임 벡터의 개수를 사용하여 움직임 기반의 샷 분할기법을 제안하였다. 이 방법은 FMB 에러신호의 평균이 어떤 임계치 이상일 때 P-프레임에서 장면전환이 일어났다고 하며 B-프레임에서는 FMB와 BMB간 차를 계산해서 샷 경계를 판단한다.

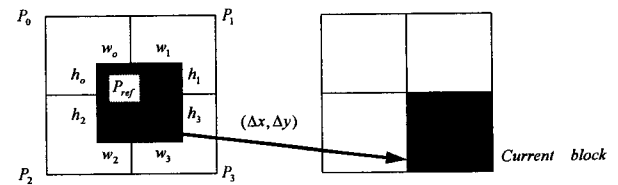
### 3.3 DCT 계수와 움직임벡터를 이용한 장면전환검출

Meng et al. [12]과 Tse et al. [13]은 DCT DC 계수와 움직임 정보를 기반으로 한 알고리즘을 제안하였다. 이 방법은 I-프레임, P-프레임, B-프레임에서 서로 다른 검출 알고리즘을 사용하였다. I-프레임에 발생하는 장면전환을 검출하기 위해서 연속되는 두 I-프레임의 컬러 히스토그램을 비교한다. P-프레임에 발생하는 장면전환을 검출하기 위해서 FMB에 대한 IMB 개수의 비  $R_p$ 가 사용된다.  $R_p$ 값이 매우 클 때 장면전환으로 판단한다. B-프레임에서 장면전환검출은 FMB에 대한 BMB 개수의 비  $R_b$ 를 사용되며  $R_b$ 가 매우 클 때 장면전환으로 판단한다. DC 영상을 사용한 Yeo의 알고리즘은 인접한 프레임을 비교하는데는 적합하지만 DC 영상을 얻는데 비교적 많은 연산이 필요하다. Meng et al.의 방법은 I-프레임에서 장면전환을 찾기 위해서 많은 양의 히스토그램 계산을 필요로 한다. 이 알고리즘은 장면

전환검출 과정을 간략화하기 위해 움직임 벡터의 비를 사용하지만 BMB 파라미터는 사용하지 않는다. 예를 들어 B-프레임에서 장면전환이 발생했을 때는 B-프레임과 그의 과거 참조 프레임간의 관계가 없다. 이런 경우에 FMB와 BMB의 개수 둘 모두가 매우 작게 된다. 만약 이러한 관계가 FMB의 개수만을 사용하게 된다면 잘못된 장면전환을 검출하게 될 수도 있다.

### 3.4 DC 영상을 이용한 방법

Bede Liu et al. [4, 14]은 MPEG으로 압축된 비디오로부터 축소된 DC 영상을 추출하고 추출된 DC 영상간의 밝기 값의 변화를 측정해서 장면전환의 위치를 찾아내는 알고리즘을 제안하였다. MPEG으로 압축될 경우 각 프레임들은 I, P, B-프레임 가운데 한가지 형태로 부호화되는데 참조 프레임을 가지지 않는 I-프레임의 경우는  $8 \times 8$  블록 단위로 DCT를 수행하여 계수값들에 대해 가변장 부호화를 수행하므로 DCT 계수의 DC 값으로부터 간단히 DC 영상을 만들 수 있지만 참조 프레임을 가지는 P, B-프레임의 경우는 움직임 보상에 의해 이전 프레임과 현재 프레임의 차가 DCT되어 가변장 부호화 됨으로 복호화 과정이 복잡해진다. (그림 1)은 MPEG에서 매크로블록 단위의 움직임 보상을 할 경우 일반적인 형태를 나타낸다.



(그림 1) P, B-프레임에서 DC 영상 추출

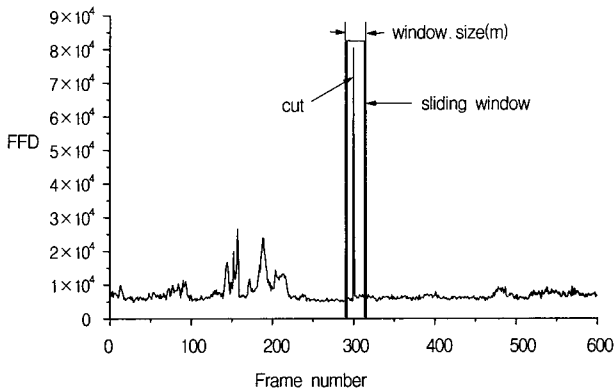
(그림 1)에서 볼 수 있듯이 움직임 보상이 행해질 경우 현재 블록이 참조할 블록 ( $P_{ref}$ )은  $P_0 \sim P_3$  블록의 전체 또는 일부분과 겹치게 된다. 이때  $h_i$ 와  $w_i$ 는 참조할 블록이 참조 프레임의 블록과 겹쳐진 부분의 수직, 수평 화소 수를 나타낸다. 실제로 현재 블록의 화소값은 참조할 블록과 겹치는 블록들의 모든 DCT 계수들에 의해 영향을 받으므로 현재 블록의 평균값은 식 (1)과 같이 겹치는 블록의 DCT 계수들 각각에 적절한 가중치를 둔 합으로 표현된다.

$$[DCT(P_{ref})]_{00} = \sum_{i=0}^3 \left( \sum_{(m, l)=(0,0)}^{(7,7)} w_{ml}^i [DCT(P_i)]_{ml} \right) \quad (1)$$

$[DCT(P_i)]_{xy}$ 는  $P_i$  블록에서  $(x, y)$  위치의 DCT 계수값을 나타내고  $w_{xy}^i$ 는  $P_i$  블록에서  $(x, y)$  위치의 DCT 계수에 대한 가중치를 나타낸다. 추출된 DC 영상을 이용해서 프레임간의 변화 정도를 나타내는 FFD(frame-to-frame difference)를 식 (2)와 같이 정의한다.

$$FFD(k) = \sum |f_{DC}(k) - f_{DC}(k-1)| \quad (2)$$

식 (2)에 의해 계산된 FFD를 이용해서 장면전환을 판단할 때는 (그림 2)와 같이  $m$ 개의 연속된 FFD 값을 검사하는 슬라이딩 윈도우를 이용한다. 만약  $k$ 번째 FFD 값이 슬라이딩 윈도우 내에서 가장 큰 값이고 두 번째 큰 값보다  $n$ 배 이상이면  $k$ 번째 프레임과  $k-1$ 번째 프레임 사이에서 장면전환이 일어난 것으로 판단한다. 일반적으로 윈도우 크기  $m$ 은 프레임율을 사용하며  $n$ 은 2.0~3.0의 값을 사용한다[14].



(그림 2) 슬라이딩 윈도우를 이용한 장면전환검출

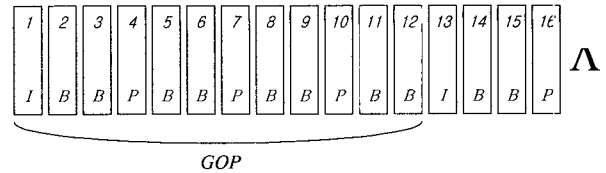
DC 영상의 경우 실제 영상을 얻기 위해 전체를 복호화할 경우에 비해 훨씬 적은 계산량이 소요되고, 슬라이딩 윈도우를 이용함으로써 카메라플래시가 있는 장면이나 급격한 움직임이 있는 장면을 컷으로 잘못 판단하는 경우가 없도록 해준다. 그러나, P, B-프레임에서 특히 B-프레임의 DC 영상 추출시 실제 영상과의 에러를 최소화하더라도 움직임이 큰 영상에서는 에러가 커져서 FFD 값이 실제와 달리 크게 증가하기 때문에 정확한 컷 검출이 어려워진다. 슬라이딩 윈도우의 경우 1초 이내에 두 번 이상의 장면전환이 일어나지 않는다고 가정하지만 최근 많이 나오고 있는 뮤직비디오나 뉴스와 같은 비디오에서는 그러한 경우가 자주 발생하기 때문에 올바른 컷 검출이 이루어지지 못한다.

#### 4. 제안한 장면전환검출 알고리즘

정확한 장면전환을 검출하기 위해서는 완전하게 복호화된 실제 프레임을 사용하는 것이 가장 좋은 성능을 얻을 수 있다. 하지만 역 DCT와 움직임 보상을 수행하는 데 많은 시간이 걸리므로 압축된 비디오에서 쉽게 추출할 수 있는 DCT 계수, 움직임 벡터, 매크로블록의 부호화 형태 같은 정보를 이용해서 장면전환을 검출하는 것이 효율적이다. 압축된 대용량 비디오를 처리하기 위해서는 복호화 과정을 최대한으로 줄이는 것이 중요하므로 제안한 방법은 복호화 과정을 최소화하기 위해 I-프레임의 DCT 계수와 B-프레임의 매크로블록 부호화 형태만을 이용해서 장면전환을 검출한다.

#### 4.1 장면전환이 있는 GOP 검출

MPEG으로 압축된 비디오는 연속된 GOP들로 이루어진다. GOP는 압축된 비디오의 임의 접근을 위해 마련된 단위이다. 각 GOP는 (그림 3)과 같이 하나의 I-프레임으로 시작되어 여러 개의 P와 B-프레임들로 구성된다.



(그림 3) 디스플레이 순서에 따른 GOP 구조

이러한 GOP 구조에서 I-프레임간 정보를 비교함으로써 장면전환이 존재하는 GOP를 찾을 수 있다. I-프레임간 비교는 DCT 계수의 DC 값으로부터 간단하게 추출할 수 있는 DC 영상을 이용하는 것이 가장 효율적이다.  $f(x, y)$ 가 I-프레임의  $(x, y)$  위치에 있는 화소값 이라면  $8 \times 8$  블록에서 DCT 계수의 DC 값  $c(0, 0)$ 은 식 (3)으로 표현 가능하다.

$$c(0, 0) = \frac{1}{8} \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \quad (3)$$

식 (3)에서 DCT 계수의 DC 값은  $8 \times 8$  블록의 평균 휘도값의 8배가됨을 알 수 있다. 따라서, I-프레임에서 각 블록의 평균값은 단순히 DCT 계수의 DC 값을 8로 나누면 되므로 원 영상의 1/64 크기인 DC 영상을 쉽게 구할 수 있다. 일단 I-프레임의 DC 영상들이 추출되면 DC 영상들 간 유사도를 조사해야한다. 본 논문에서는 DC 영상들간의 내용 변화를 나타내는 값으로 휘도투영(luminance projection) 방법[2]을 사용하였다. 휘도투영 방법은 작은 물체나 카메라 움직임 같은 영향에 상대적으로 강한 성질을 가진다. 주어진 DC 영상  $f(j, k)$  ( $j=1, 2, \dots, J, k=1, 2, \dots, K$ )에 대해서  $n$ 번째 행에 대한 휘도투영은 식 (4)로 나타낼 수 있다.  $J$ 와  $K$ 는 각각 DC 영상의 세로와 가로 화소 수를 나타낸다.

$$l_N^R = \sum_{j=1}^J Lum\{f(j, n)\} \quad (4)$$

$$l_m^c = \sum_{k=1}^K Lum\{f(m, k)\} \quad (5)$$

휘도투영을 사용해서 두 DC 영상간의 상관성을 나타내는 척도로 사용한다. 두 개의 DC 영상  $f_i^{DC}$ 와  $f_j^{DC}$ 간 유사성은 휘도투영 값들의 절대 차의 합  $d_b(i, j)$ 으로 정의하며 이 값은 0과 1사이의 정규화 된 값으로 계산된다.

$$d_b(i, j) = \frac{1}{255(J+K)} \left\{ \frac{1}{J} \sum_{n=1}^J |l_n^R(i) - l_n^R(j)| + \frac{1}{K} \sum_{m=1}^K |l_m^c(i) - l_m^c(j)| \right\} \quad (6)$$

I-프레임은 GOP 마다 한 프레임씩 존재하기 때문에 일반적으로 12-15 프레임 정도 떨어져 있다. 그러므로 I-프레임의 DC 영상간 단순한 차를 사용할 경우 물체의 움직임에 의한 영향으로 프레임간의 차가 갑작스럽게 커질 수 있는데 휘도투영은 행과 열의 평균을 이용함으로써 기존의 다른 방법에 비해 이러한 영향을 크게 줄일 수 있다.  $d_{ip}(i, j)$  값이 모두 계산되면 이 값들을 가지고 장면전환을 판단한다. 기존의 방법들은 슬라이딩 윈도우 내에서 임계값을 사용해서 장면전환을 판단하는데 이 방법은 비디오 종류에 따라 검출 성능이 달라지는 단점이 있다. 제안한 방법은 임계값을 사용하지만 다양한 비디오에 적용할 수 있도록 하기 위해  $d_{ip}(i, j)$  값의 평균 ( $m$ )과 분산 ( $\sigma^2$ )을 이용해서 임계값이 자동적으로 설정되도록 하였다. 대부분의  $d_{ip}(i, j)$  값은 평균값 주변의 값을 가지게 되며 장면전환이 있을 경우에는 평균보다 큰 값을 가지게 된다. 장면전환이 없더라도 움직임이 많은 경우 평균값보다 큰 값은 가질 수 있으므로 임계값은 평균보다는 커야하지만 얼마나 더 크냐에 따라서 검출 결과가 크게 달라진다. 임계값을  $m + \sigma^2$ 로 설정하면 장면전환이 존재하지 않는 GOP를 검출 할 수도 있다. 그러나  $m + \sigma^2$  이상을 사용하면 오 검출 비율은 낮아지지만 실제로 장면전환이 존재하는 GOP를 많이 놓칠 수 있다. 본 논문에서는 검출되지 않는 장면전환의 수를 최소화하기 위해 임계값으로  $m + \sigma^2$ 로 설정하고 만약 잘못된 장면전환을 검출한 경우를 대비해서 GOP 내에서 실제 장면전환인지 아닌지를 한번 더 검사하도록 하였다.

4.2 GOP 내에서의 장면전환검출

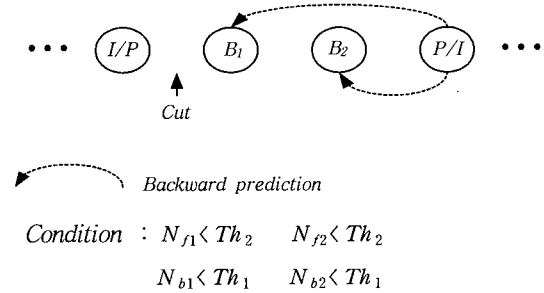
장면전환이 있다고 판단된 GOP 내에서 정확한 컷의 위치를 찾기 위해서 B-프레임과 그것의 참조 프레임과의 관계를 이용한다. 즉 현재 B-프레임을 부호화하기 위해 참조 프레임으로 이전 프레임을 사용하는 비율 ( $N_f$ )과 이후 프레임을 사용한 매크로블록의 비율 ( $N_b$ )을 계산한다.

$$N_f = \frac{\text{number of forward predicted MB}}{\text{total number of MB in a frame}} \quad (7)$$

$$N_b = \frac{\text{number of backward predicted MB}}{\text{total number of MB in a frame}} \quad (8)$$

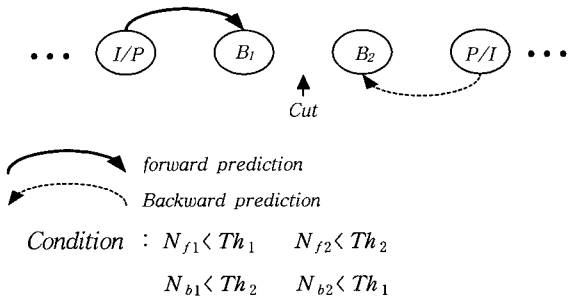
MPEG으로 압축된 비디오의 경우 I와 P-프레임 또는 P와 P-프레임 사이에는 일반적으로 2개의 B-프레임이 존재한다. 따라서 장면전환이 존재할 수 있는 위치는 이웃한 2개의 B-프레임의 앞, 중간 그리고 뒤의 3가지 경우가 있다. 장면전환이 발생한 경우 장면전환된 프레임을 경계로 첫 번째 B-프레임은 주로 이전의 I, P-프레임을 참조하게 되고 두 번째 B-프레임은 주로 이후의 I, P-프레임을 참조하게 된다. (그림 4)와 (그림 5)와 (그림 6)은 장면전환이 존재할 수 있는 3가지 경우와 각 경우에 대해 장면전환을 판

단하는 기준을 나타내고 있다. 그림에서  $Th_1$ 과  $Th_2$ 는 장면전환을 판단하기 위한 상위 임계값과 하위 임계값이다.  $N_{b1}$ 과  $N_{b2}$ 는 각각 B<sub>1</sub>-프레임과 B<sub>2</sub>-프레임의  $N_b$ 값을 나타내고,  $N_{f1}$ 과  $N_{f2}$ 는 각각 B<sub>1</sub>-프레임과 B<sub>2</sub>-프레임의  $N_f$ 값을 나타낸다.



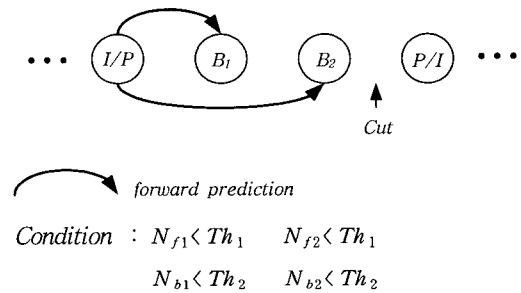
(그림 4) 이전 참조 프레임과 B<sub>1</sub>-프레임 사이의 장면전환

(그림 4)의 경우는 B<sub>1</sub>과 B<sub>2</sub> 프레임의  $N_{f1}$ 과  $N_{f2}$  모두  $Th_2$ 보다 작고  $N_{b1}$ 과  $N_{b2}$ 는 모두  $Th_1$ 보다 큰 경우로 두 B-프레임 모두가 이후 프레임에서 대부분의 참조했으므로 컷은 이전의 I, P-프레임과 B<sub>1</sub> 프레임 사이에 존재한다.



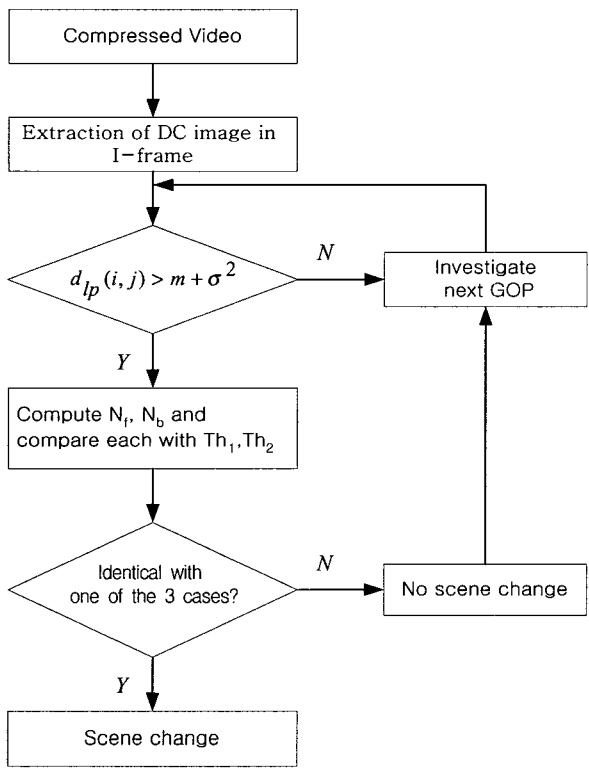
(그림 5) 이웃한 두 B-프레임 사이의 장면전환검출

(그림 5)는 B<sub>1</sub> 프레임에서는  $N_{b1}$ 이  $Th_2$ 보다 작고  $N_{f1}$ 이  $Th_1$ 보다 크며, B<sub>2</sub> 프레임에서는 반대의 결과가 나오는 경우로 B<sub>1</sub> 프레임은 대부분 이전 프레임을 참조했으며 B<sub>2</sub> 프레임은 대부분 이후 프레임을 참조 프레임으로 한 경우이므로 컷은 B<sub>1</sub>과 B<sub>2</sub> 프레임 사이에 존재하게 된다.



(그림 6) B<sub>2</sub>-프레임과 이후 참조 프레임 사이의 장면전환검출

마지막으로 (그림 6)은 B<sub>1</sub>과 B<sub>2</sub> 프레임의 N<sub>b1</sub>과 N<sub>b2</sub> 모두 Th<sub>2</sub>보다 작고 N<sub>f1</sub>과 N<sub>f2</sub> 모두 Th<sub>1</sub>보다 큰 경우로 두 B-프레임 모두가 이전 프레임을 많이 참조했으므로 컷은 B<sub>2</sub> 프레임과 이후의 I/P-프레임 사이에 존재한다. 만약 GOP 내에서 위의 3가지 조건에 해당하는 경우가 없다면 장면전환이 일어나지 않은 것으로 판단하기 때문에 잘못된 판단한 GOP를 제거시켜 줄 수 있다. 임계값으로 사용되는 Th<sub>1</sub>과 Th<sub>2</sub>는 전체 매크로블록 개수에 대해 순방향 및 역방향 매크로블록 개수의 비율이 각각 70%와 15%가 되도록 설정하였다. (그림 7)은 제안한 장면전환검출 알고리즘의 순서도를 나타낸다. 제안한 방법의 장점은 간단하면서도 정확한 장면전환 검출이 가능하다는 것이다. 제안한 방법은 압축된 비디오 스트림으로부터 I-프레임에서는 DCT DC 계수, B-프레임에서는 매크로블록의 부호화 형태만을 가변 길이 복호화하므로 장면전환을 빠르게 검출할 수 있다.



(그림 7) 제안한 알고리즘의 순서도

5. 실험결과 및 고찰



제안한 장면전환검출 알고리즘의 성능을 평가하기 위해서 일반적으로 많이 사용되는 Precision rate와 Recall rate [15, 16]를 성능 평가 척도로 사용하였다.

$$Recall\ rate = \frac{number\ of\ detected\ cuts}{number\ of\ detected\ cuts + number\ of\ missed\ cuts} \times 100\ [%] \quad (9)$$

$$Precision\ rate = \frac{number\ of\ detected\ cuts}{number\ of\ detected\ cuts + number\ of\ false\ alarms} \times 100\ [%] \quad (10)$$

식 (9)의 Recall rate는 실제 컷이 존재함에도 불구하고 검출하지 못한 컷이 얼마나 많은 가를 나타내고, 식 (10)의 Precision rate는 실제 컷은 아니지만 컷으로 오인해서 잘못 검출한 컷이 얼마나 많은 가를 나타내는 척도이다. 일반적으로 비디오 색인을 위해 구조화나 클러스터링 과정을 수행하면 유사한 내용의 샷을 하나의 논리적인 단위인 이벤트(event)로 묶어주게 된다. 그러므로, 컷이 아닌 부분을 컷으로 잘못 검출하더라도 클러스터링에 의해 같은 내용 단위로 묶일 수 있다. Precision rate가 너무 낮을 경우 클러스터링 수행 횟수가 많아지게 된다. 반면에 검출하지 못한 컷이 있을 경우 클러스터링 과정에서 샷 간 유사도를 계산할 때 잘못된 값이 나오게 되어 적절한 비디오 구조화가 이루어지지 못한다. 따라서, 효율적인 비디오 검색 및 색인을 위해서는 Precision rate와 Recall rate가 모두 높아야 한다. 실험 비디오 시퀀스로는 최근 인터넷을 통해 많이 유포되는 뮤직 비디오를 사용하였다. 일반적으로 뮤직 비디오는 다양한 내용 전개로 구성되며 많은 카메라 효과가 있기 때문에 장면전환검출 성능을 평가하기에 적합하다. <표 1>은 실험에 사용된 비디오 시퀀스에 대한 세부정보를 나타내고 있다. 참고로, <표 1>에서 실제 장면전환의 개수는 해당 시퀀스를 한 프레임씩 차례로 보면서 판단한 수치이므로 어느 정도 객관성을 갖는 값이다.

<표 1> 실험에 사용된 비디오 시퀀스

	Test sequence ①	Test sequence ②
Display time	6 min. 55 sec.	8 min. 23 sec.
No. of frames	12391 frames	15086 frames.
GOP	15	15
Frame rate	30 frames/sec	30 frames/sec
Bit rate	1.394 Mbits/sec	1.496 Mbits/sec
No. of real cuts	256 cuts	324 cuts
Specific frame		
	10th frame	10th frame

성능평가로써 제안한 방법을 기존의 DC 영상을 이용하는 Bede Liu의 방법 및 상용 비디오편집 툴(tool)인 “Webflix”[17]와 비교를 하였다. 참고로, Webflix는 DC 영상간 비교가 아닌 완전 복호화된 프레임간 비교를 통해서 장면전환을 검출하는 구조를 갖고 있다. Bede Liu의 방법에서 파라미터설정은 m=30, n=2.3로 하였으며 “Webflix” 편집 툴에서는 “generating shot index”라는 메뉴를 사용해서 장면전환검출을 수행하였다. 기존연구에 의하면 비 압축 영역에서의 검출 성능은 약 90% 이상이 되고 압축 영역에

서의 검출 성능은 약 80~90%로 나와있다. 그러나, 압축 영역인 경우에는 갑작스런 카메라 효과가 있거나 움직임이 많은 비디오에서는 성능이 비 압축 영역의 경우보다 검출 성능이 급격히 떨어지는 경향이 있다. 테스트 시퀀스 ①과 ②에 대해서 검출된 장면전환의 수, 검출하지 않은 장면전환의 수, 잘못 검출한 장면전환의 수 및 Recall, Precision rate를 계산해서 <표 2>와 <표 3>에 각각 나타내었다.

#### ● 실험환경

- O/S : Windows 98
- CPU : Pentium II 350 MHz
- Main Memory : 64 Mbyte
- Compiler & Library : Visual C++ 6.0, MFC

<표 2> 테스트 시퀀스 ①의 장면전환검출 결과

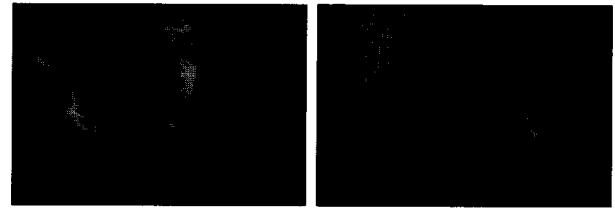
	Bede Liu's method using all DC image	Method using fully decoded images(WebFlix)	Proposed method
No. of detected cuts	246	249	232
No. of missed cuts	48	27	26
No. of falsely detected cuts	38	6	2
Recall rate	83.67 %	90.21	<b>89.92 %</b>
Precision rate	86.62 %	97.65	<b>99.15 %</b>
Processing time	18 sec.	110 sec.	13 sec.

<표 3> 테스트 시퀀스 ②의 장면전환검출 결과

	Bede Liu's method using all DC images	Method using fully decoded images(WebFlix)	Proposed method
No. of detected cuts	325	331	308
No. of missed cuts	195	65	61
No. of falsely detected cuts	196	51	44
Recall rate	62.50 %	83.58	<b>83.47 %</b>
Precision rate	62.38 %	86.87	<b>87.50 %</b>
Processing time	25 sec.	156 sec.	18 sec.

<표 2>와 <표 3>의 결과로부터 제안한 방법이 Bede Liu의 방법보다 시퀀스 ①, ②에 대해서 약 6%와 20% 더 높은 Recall rate를 보이고 처리시간도 적음을 알 수 있다. 그리고, WebFlix와 비교해서는 Recall과 Precision 측면에서는 비슷한 성능을 보이면서 처리시간은 8배 이상 빠르다는 것을 보인다. 특히 물체의 움직임이 많고 화면에 잔상이 오래 남는 실험 영상 ②의 경우에는 DC 영상 기반의 방법이 비디오 구조화나 색인에 적용하기에 검출 성능이 매우 낮는데 반해 제안한 방법은 우수한 성능을 보이고 있다. 그러나, 기존의 방법과 제안한 방법 모두 (그림 8)과 같이 전체적으로 어두운 배경을 바탕으로 휘도 정보가 비슷한 상황에서 발생하는 장면전환은 검출할 수 없었다. 그러므로, 휘도 정보가 아닌 물체의 윤곽 정보 등의 프레임 사이의

차를 효과적으로 나타낼 수 있는 방법의 연구가 있어야 할 것으로 생각된다.



frame 10528                      frame 10529  
(그림 8) 어두운 배경으로 이루어진 장면전환

## 6. 결 론

본 논문에서는 압축된 비디오에서 최소 복호화에 의한 빠르고 정확한 장면전환검출 알고리즘을 제안하였다. 제안한 방법은 I-프레임의 DC 영상만을 이용해서 장면전환을 포함한 GOP를 찾고 B-프레임에서 매크로블록의 부호화 형태만을 이용해서 GOP 내에서 장면전환이 발생한 정확한 위치를 찾는다. 제안한 방법은 P, B-프레임의 DC 영상을 모두 이용하는 기존 방법보다 간단하고 P, B-프레임의 DC 영상에서 발생하는 화질 열화를 고려할 필요가 없다는 장점이 있다. 그리고, GOP 단위로 건너뛰면서 장면전환 여부를 조사하기 때문에 수행속도가 빠르며 임의의 GOP 형태를 갖는 비디오에도 쉽게 적용할 수 있다. 실험결과 제안한 방법이 기존 방법에 비해 Precision과 Recall rate 측면에서 좋은 성능을 나타냄을 알 수 있었다. 특히 물체의 움직임이 많고 화면에 잔상이 오래 남는 비디오의 경우 기존 DC 영상을 이용한 방법은 검출 성능이 매우 낮는데 반해 제안한 방법은 우수한 성능을 보이고 있다. 제안한 방법은 I-프레임간 유사도 척도로써 해당 DC 영상에서 휘도 투영 방법을 사용하였다. 그러나 만약 비슷한 휘도 정보를 가지는 프레임이 연속되는 경우에는 이 방법이 취약할 수 있기 때문에 유사도 계산시 영상의 윤곽정보를 함께 이용한다면 이러한 단점을 보완할 수 있다고 생각된다.

## 참 고 문 헌

- [1] M. K. Mandal, F. Idris, and S. Panchanathan, "Image and video indexing in Compressed Domain : A critical review," VCCL Lab., Dep. of Elec. Eng., University of Ottawa, Paper submitted to *Image and Vision Computing Journal*.
- [2] M. M. Yeung and B. Liu, "Efficient matching and clustering of video shots," *IEEE International Conference on Image Processing*, Vol.1, pp.338-341, 1995.
- [3] B. L. Yeo, and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.5, No.6, pp.533-544, Dec., 1995.

[4] J. Song and B. Liu, "Fast extraction of spatially reduced image sequences from MPEG-2 compressed video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No.7, pp.1100-1114, Oct., 1999.

[5] H. J. Zhang, A. Kankanhalli, S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, Vol.1, No.1, pp.10-28, 1993.

[6] R. Kasturi, and R. Jain, "Dynamic Vision," Computer Vision : Principles, *IEEE Computer Society Press*, Washington, 1991.

[7] D. Swanberg, C. F. Shu, and R. Jain, "Knowledge guided parsing and retrieval in video databases," Storage and Retrieval for Image and Video Databases, *Proc. SPIE* 1908, pp.173-187, Feb., 1993.

[8] F. Arman, A. Hsu, and M-Y. Chiu, "Image processing on compressed data for large video databases," *SPIE Proceedings : Storage and Retrieval for Image and Video Databases*, pp.267-272, 1993.

[9] H. J. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools and Application*, Vol.1, No.1, pp.89-111, Mar., 1995.

[10] B. -L. Yeo, and B. Liu, "A unified approach to temporal segmentation of motion JPEG and MPEG compressed videos," *Proc. International Conference on Multimedia Computing and Systems*, pp.81-88, May, 1995.

[11] H. C. Liu, and G. L. Zick, "Scene decomposition of MPEG compressed video," *SPIE Proceeding : Digital Video Compression : Algorithms and Technol.*, Vol.2419, pp.26-37, Feb., 1995.

[12] J. Meng, Y. Juan, and S. F. Chang, "Scene change detection in MPEG compressed video sequence," *SPIE Digital Video Compression : Algorithms and Technol.*, Vol.2419, pp.14-25 Feb., 1995.

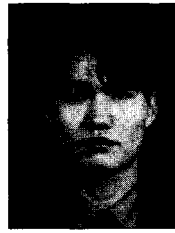
[13] K. Tse, J. Wei, and S. Panchanathan, "A scene change detection algorithm for MPEG compressed video sequence," *CCECE'95 Montreal*, Vol.2, pp.827-830.

[14] B. L. Yeo, and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.5, No.6, pp.533-544, Dec., 1995.

[15] G. Salton, *Automatic Information Organization and Retrieval*, McGraw-Hill, New York, 1968.

[16] J. S. Boreczsky and L. A. Rowe, "Comparison of video shot boundary detection techniques," *Proc. Storage and Retrieval for Image and Video Databases IV*, Vol. SPIE-2670, pp.170-179, 1996.

[17] <http://www.mediaware.com>.



**김 강 욱**

e-mail : ekans99@hanmail.net  
 1996년 경북대학교 전자공학과 졸업 (학사)  
 1998년 경북대학교 대학원 전자공학과 졸업(공학 석사)  
 2002년 경북대학교 대학원 전자공학과 졸업(공학 박사)

2002년~현재 삼성전자 무선통신연구소 선임연구원  
 관심분야 : 영상신호처리, 비디오 브라우징, VOD 단말 등



**이 재 승**

e-mail : jslee@kari.re.kr  
 1999년 경북대학교 전자공학과 졸업 (학사)  
 2001년 경북대학교 대학원 전자공학과 졸업(공학석사)

2001년~현재 한국항공우주연구원 연구원  
 관심분야 : 영상신호처리, 비디오구조화



**김 종 훈**

e-mail : kjh@phenix.dyu.ac.kr  
 1980년 경북대학교 전자공학과 졸업 (학사)  
 1984년 경북대학교 대학원 전자공학과 졸업(공학석사)  
 1993년 경북대학교 대학원 전자공학과 졸업(공학박사)

1981년~1982년 LG 정밀 중앙연구소 연구원  
 1984년~1993년 연암공업대학 전자과 부교수  
 1994년~현재 동양대학교 정보통신공학부 교수  
 관심분야 : 영상압축, 영상부호화, 디지털통신



**황 찬 식**

e-mail : cshwang@ee.knu.ac.kr  
 1977년 서강대학교 전자공학과 졸업 (학사)  
 1978년 한국과학기술원 전자공학과 졸업(공학석사)  
 1996년 한국과학기술원 전자공학과 졸업(공학 박사)

1991년~1992년 UTA 방문교수  
 1979년~현재 경북대학교 전기전자공학부 교수  
 관심분야 : 영상통신, 암호통신, 초고속망