

# MPEG에서 B 프레임의 특징을 이용한 급진적 장면전환 검출에 관한 연구

김 중 헌<sup>†</sup> · 장 종 환<sup>\*\*</sup>

## 요 약

일반적인 장면전환 검출방법은 연속적인 두 영상의 특징 값을 비교하여 일정한 임계값 이상일 경우 장면전환으로 판단한다. 그러나 기존의 장면전환을 검출하는 알고리즘은 장면전환을 검출하는데 있어서 프레임의 특징 값을 추출하기 위하여 복호화 과정에서 많은 시간이 소비되었고 단지 연속적인 두 영상의 특징 값을 비교하기 때문에 빛의 변화나 물체의 움직임에 따른 오검출 문제를 나타내었다. 본 논문에서는 MPEG 압축 영역에서 매크로블록 정보를 직접 추출 및 이용하여 효과적인 장면 전환 검출을 위한 알고리즘을 제안한다. 제안한 알고리즘은 MPEG에서 매크로블록 정보를 직접 추출하고 이용하므로 기존의 알고리즘의 문제점인 많은 연산량 문제를 개선하였고, 연속된 프레임과의 비교를 통한 장면전환 검출이 아닌 이전 또는 이후 영상과의 연관성을 분석하여 장면전환된 프레임을 검출함으로써 빛의 변화나 물체의 움직임과 같은 오검출 문제를 해결할 수 있는 알고리즘을 제안한다. 제안한 알고리즘은 MPEG 데이터를 대상으로 한 실험을 통해 기존의 히스토그램을 이용한 알고리즘보다 빠르고 정확하게 검출할 수 있음을 보이고, 실험 결과를 통해 성능을 분석하였다.

키워드 : 장면전환 검출, B 프레임, 매크로 블록 타입, 모션 예측

## A Study on the Abrupt Scene Change Detection Using the Features of B frame in the MPEG Sequence

Joong-Heon Kim<sup>†</sup> · Jong-Whan Jang<sup>\*\*</sup>

### ABSTRACT

General scene change detection determines the changes of a scene by using feature comparison of two continuous images that are above the fixed threshold. But existing algorithm detects scene change that was used in comparing the features of two images continuously, it usually takes a lot of time in decrypting the image data and false-detection problem occurs when there is an object motion or a change of illumination. In this paper, macroblock were used to extract the information directly from the MPEG compression area and suggests algorithm that will detect scene changes more effectively. Existing algorithm have shown numerous arithmetic problems that were improved in the proposed algorithm. The existing algorithm cannot detect the changes of a scene after analyzing the relationship of the previous and future images while the algorithm being proposed can detect the changes of a scene continuously and resolves the problem of false-detection. To this end, the data used in general were tested to prove that this algorithm would be able to detect the scene changes faster and more correctly than the existing ones. The performance of the suggested algorithm was analyzed based on the results of the experiment.

Key Words : Scene Change Detection, B Frame, Macroblock Type, Motion Estimation

### 1. 서 론

동영상은 인간에게 많은 정보를 매우 자연스럽게 제공해 줌으로써 많은 관심의 대상이 되고 있다. 최근 들어 컴퓨터와 통신기술의 발달로 멀티미디어 데이터를 생성, 저장 및 처리하는 기술이 급속히 발달함에 따라 인터넷 방송, 주문형 비디오(VOD; Video On Demand), 의료 비디오 시스템, 동영상 편

집, 무인 카메라에 의한 물체 인식 등 멀티미디어 서비스 요구가 증대되었고, 이러한 요구를 충족시키기 위해서 빠른 시간 내에 정확하게 필요한 정보를 검색, 처리 할 수 있는 방법의 개발이 매우 중요하여 이에 대한 많은 연구가 수행되고 있다. 일반적으로 동영상 검색 시스템에서는 사용자가 전체 동영상 정보를 한 눈에 파악하는 것이 필요하다. 동영상을 파악하려면 내용을 요약해 놓은 대표 프레임 리스트가 필요한데 대표 프레임 리스트를 작성하기 위하여 장면전환을 정확하게 검출할 필요성이 발생한다. 이러한 기능을 구현하기 위해서

<sup>†</sup> 준 회원 : 배재대학교 정보통신공학과 박사과정

<sup>\*\*</sup> 중신회원 : 배재대학교 정보통신공학과 교수

논문접수 : 2005년 2월 7일, 심사완료 : 2005년 7월 25일

비디오의 각 장면 별로 색인과 특징을 부여하는 비디오 분할 작업이 선행 되어야 한다. 장면 전환 검출(scene change detection)은 동영상의 내용기반 검색을 위해 필요한 기초 작업으로써 동영상으로부터 장면들을 구분하기 위해 컷(cut)이라 불리는 전환 지점들을 자동으로 찾아내는 작업을 말하며 이를 바탕으로 동영상 데이터에 대한 색인과 검색이 이루어지게 되므로 효과적인 검출 방안이 반드시 필요하다 [1]-[5].

장면전환은 크게, 급격한 장면전환과 점진적인 장면전환으로 구분할 수 있다. 급격한 장면전환은 한 프레임 만에 순간적으로 장면이 변하는 것이고, 점진적인 장면전환은 여러 프레임에 걸쳐 서서히 장면이 전환되는 것이다. 두 영상의 픽셀 및 히스토그램 차를 이용한 장면전환 방법은 구현하기 간단하나 움직임은 물체가 존재하거나 빛의 변화에 따른 오 검출이 많은 단점이 있다. 이런 문제점을 해결하기 위해 모델기반 장면전환 검출 기법을 제안하였다[2-3]. 이 방법들은 화소값을 이용하여 장면전환을 검출하므로 압축된 영상에 적용할 수 없기 때문에 복호화에 많은 시간이 소비되는 문제점이 있다. 이러한 단점들을 해결하기 위해 압축된 비트열의 내용을 직접 분석하여 특징 정보를 추출하여 동영상의 장면전환 검출에 활용하였다[4-6]. 위의 알고리즘은 상대적으로 계산시간이 빨라지긴 했지만 여기에서 사용되는 정보가 실제 영상의 내용과 정확하게 일치하지 않는 약간의 오류를 포함하는 정보이기 때문에 비압축 도메인에서의 알고리즘에 비하여 검출 효율이 저하된다.

또한 두 부류의 알고리즘들은 대부분 카메라나 물체의 움직임, flashlight에 의한 효과, 영상이 전체적으로 어둡거나 밝은 경우에 대한 효과들을 고려하지 않았다. 실제로 각 알고리즘들은 이러한 카메라나 물체의 움직임, flashlight, 영상의 전체적인 밝기에 대하여 상당히 민감한 성능을 보이고 있다.

본 논문에서는 복호화에 소비되는 시간을 단축하기 위하여 압축 비트 열에서 프레임의 특징을 직접 추출 및 분석하였고, 물체의 움직임 및 빛의 변화에 따른 오 검출을 감소시키기 위하여 이전 또는 이후 프레임과의 연관성을 분석하여 장면전환을 검출하였다.

MPEG동영상은 모션 예측이라는 과정을 통하여 압축률을 높이는데 이 과정에서 I 또는 P 프레임, P 또는 P 프레임 사이에 존재하는 B 프레임들의 매크로블록들은 I 또는 P 프레임으로 대체된다. 즉, B 프레임들은 이전 또는 이후의 프레임 신호를 포함하고 있고 이것은 매크로블록 타입 정보에 나타나게 된다. 본 논문에서는 이러한 특징을 이용하여 장면전환 시 모션 예측하는 프레임들에 나타나는 매크로블록 타입의 변화를 이용하여 장면전환을 효과적으로 검출할 수 있는 알고리즘을 제안한다.

본 논문의 구성은 2장에서 기존의 알고리즘에 대하여 설명하고 3장에서는 MPEG동영상의 구성에 대하여 설명하였다. 4장에서는 양 방향 모션예측을 수행하여 구성되는 B 프레임을 분석하여 장면전환을 검출하는 알고리즘을 제안한다. 5장은 제안한 알고리즘을 실험을 통해 성능을 분석하고, 6장에서 결론을 기술한다.

## 2. 기존의 장면전환 검출 알고리즘

기존의 장면 전환 검출 알고리즘은 크게 나누어 보면, 압축 도메인에서의 알고리즘과 비압축 도메인에서의 알고리즘으로 나누어 질 수 있다. 이 중에서 오늘날 비디오 정보의 효율적인 저장과 전송을 위한 영상 압축 기술의 급속한 발달과 더불어 계산량을 줄이고자 하는 목적 때문에 압축 도메인에서의 알고리즘에 대한 연구가 더욱 많이 이루어지고 있다. 이들 두 종류의 알고리즘 중 대표적인 몇 가지에 대하여 살펴보면 다음과 같다.

### 2.1 비압축 도메인에서의 알고리즘

비압축 도메인에서의 장면 전환 검출 알고리즘은 많은 경우에 영상의 히스토그램이나 에지 정보를 이용한 방법 또는 픽셀값의 차이를 이용하는 방법을 사용하고 있다.

먼저 픽셀의 차를 이용하는 방법은 동일한 위치에 있는 두 프레임의 픽셀 값들의 차의 절대값의 합이 일정한 임계 값을 초과할 경우 장면 전환으로 판단하는 방법으로 간단하고 수행시간도 적게 드는 장점이 있다. 하나의 장면 내에는 거의 유사한 장면들로 이루어져 있기 때문에 프레임간의 픽셀 차이는 매우 작은 값을 갖는다. 그러나 장면전환이 일어날 경우 다른 장면내의 프레임이 유사하지 않고 서로 다른 내용을 나타내기 때문에 이들 픽셀 값들의 차이는 크게 나타난다. 이러한 특성을 이용하여 장면전환 점을 찾아 낼 수 있다. 이 방법은 간단하고 수행시간이 적게 드는 장점이 있지만 객체의 이동이나 프레임의 잡음에 매우 민감하다는 단점을 가지고 있다[1].

다음으로 히스토그램 차를 이용하는 방법은 픽셀의 차를 이용한 방법과 같이 구현하기 간단하고 널리 사용되고 있는 방법으로 프레임간의 히스토그램의 차이를 이용하는 방법이다. 이 방법은 객체의 움직임에 대해 효과적이고 잡음에 덜 민감한 장점이 있지만 조명 변화가 다양한 영상에서는 비 효율적이다[1]-[3].

### 2.2 압축 도메인에서의 알고리즘

MPEG 압축 데이터를 대상으로 한 장면전환검출 알고리즘으로 Yeo[7] 등은 MPEG 데이터로부터 DC 성분만을 추출하여 원래 영상 크기를 1/64로 줄인 DC 시퀀스를 구성하였고 히스토그램 비교법을 적용하여 장면전환을 검출하였다. I 프레임으로부터 DC성분을 추출하여 내는 것은 쉬운 일이나, P 프레임이나 B 프레임에서는 움직임 벡터를 이용하여 DC 성분을 따로 계산하여 내는 과정이 필요하다.

Shen[11]는 DC성분과 움직임 벡터를 이용하여 컬러 히스토그램 차이로 장면전환을 검출하였다. 이 알고리즘은 DC 성분 추출을 위한 복호화 과정이 필요하다.

Meng[9]에 의해 제안된 알고리즘은 움직임 벡터를 이용한다. I frame에서 장면 전환 여부를 판단하는 데에는 다음과 같은 방식을 이용한다. 먼저, I frame 앞에 있는 B frame을 이용한다. B frame이 디스플레이상으로 앞에 위치한 I/P frame 보다 뒤에 위치한 I frame과 유사하다면 그 B frame은 앞 I/P

frame의 블록보다 뒤 I frame의 블록으로 많이 만들어질 것이다. 이는 달리 말해서 B frame을 만들 때 순방향 움직임 벡터의 수가 역방향 움직임 벡터의 수보다 더 많다고 할 수 있다. 따라서 순방향 움직임 벡터와 역방향 움직임 벡터 수의 비를 구하여 I frame에서의 장면 전환을 검출할 수 있다. 그러나 이 알고리즘의 경우 수학적 모델링이 제시되지 않았으며, B 프레임의 매크로 블록 타입 중 양방향 움직임 벡터를 고려하지 않았으므로 인코더에 독립적이지 못한 단점을 나타낸다. Kuo[13]등은 I, P, B 픽처 각각에 컷이 발생하는 때를 컷 마스크(cut mask)로 구성하였고 데이터 시퀀스와 마스크 매칭을 통하여 장면전환 검출을 하였다. 압축 도메인에서의 알고리즘은 상대적으로 계산시간이 빨라지긴 했지만 여기에서 사용되는 정보가 실제 영상의 내용과 정확하게 일치하지 않는 약간의 오류를 포함하는 정보이기 때문에 비압축 도메인에서의 알고리즘에 비하여 검출 효율이 약간 저하된다.

### 3. MPEG 동영상의 구조

#### 3.1 MPEG 구조

MPEG에서는 동영상의 시공간적인 중복성을 제거하기 위하여 MC-DCT(Motion Compensation-Discrete Cosine Transform) 압축 방식을 이용한다. 간단히 MPEG 구조를 살펴보면 가장 상위 레벨에는 비디오 시퀀스(Video Sequence)가 있고, 그 아래 동영상 편집을 고려한 GOP(Group Of Picture)가 있다. GOP는 프레임들의 집합을 나타내며 각각 I(Intra Picture), P(Predictive Picture), B(Bidirectionally Predictive Picture)로 구분되어 있고 각 프레임마다 다른 방식으로 부호화된다. 프레임은 16×16 크기의 매크로블록으로 구성되고, 매크로블록은 DCT 변환 단위인 8×8 블록으로 구성된다. 한 매크로블록은 밝기 정보를 포함하고 있는 4개의 Y 블록과 색 정보를 포함하고 있는 2개의 Cb, Cr 블록을 가지고 있다. MC-DCT 압축 방식에 따라 압축된 MPEG 비트열은 각 계층적인 구조들을 나타내는 헤더정보, 각종 파라미터, DCT 계수, 움직임 벡터 등이 VLC(Variable Length Code)된 결과이다. 장면전환 검출은 프레임간의 컷이나 점진적인 변환(Gradual Transition)이 존재하는지를 판단하는 것이므로 프레임 단위에 대한 이해가 필요하다. 프레임은 I, P, B 프레임 세 종류로 되어 있는데 각각 다른 부호화 기술이 적용된다. I 프레임은

디코딩 시 기준이 되는 프레임인데, P, B 프레임으로 인해 발생하는 에러를 줄이기 위해서 DCT 와 양자화를 통해 공간 중복성만을 제거한다. P, B 프레임은 참조 프레임과의 차영상(Residual Error Image)과 움직임 벡터(Motion Vector)로 시간 중복성을 제거한다.

#### 3.2 MPEG에서 프레임의 구조

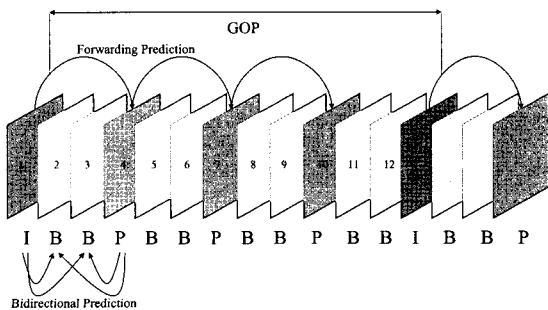
MPEG에서는 모션예측을 통한 압축을 수행하는데 이는 영상 전체에 대하여 수행하지 않고 매크로블록 단위로 분할하여 수행한다. 다시 말하면 모션 예측을 통하여 얻어진 매크로블록 또는 원 영상 자체의 매크로블록이 모여서 하나의 영상을 구성하게 된다.

MPEG에서 B 프레임에 매크로블록은 I 또는 P 프레임과 인코딩되는 B 프레임간의 차영상을 구하여 그 차영상의 분산을 이용하여 매크로블록 타입이 결정된다. MPEG 에서 프레임은 16×16의 매크로블록들의 집합으로 이루어진다. 임의의 프레임을  $F$ 라고 정의하고,  $F$ 에서  $(i, j)$ 위치의 16×16 크기의 매크로블록을  $f^{i,j}(x, y)$ 라고 하였을 경우 (그림 2)에서 보는 바와 같이  $F$ 는 식 (1)과 같이 나타낼 수 있다.

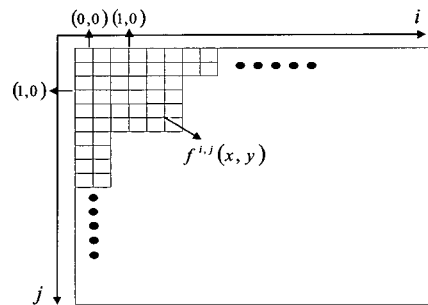
$$F = \{f^{i,j}(x, y) | i = 0, 1, \dots, M-1 \text{ and } j = 0, 1, \dots, N-1\} \quad (1)$$

(그림 2)에서  $M$ 은  $i$ 축 방향의 매크로블록의 개수이고,  $N$ 은  $j$ 축 방향의 매크로블록의 개수이다. 또한  $(x, y)$ 는 매크로블록 내의 픽셀의 좌표가 된다.

일반적으로 MPEG에선 I 또는 P 프레임들 사이에 2개의 B 프레임이 존재하고 B 프레임들은 forward 또는 backward 프레임들을 참조하여 모션 예측을 통한 압축을 수행한다. 즉 B 프레임은 양쪽 방향에 걸쳐 모션 예측을 수행함으로써 양쪽 프레임들간의 연관성이 포함되어 있으므로 양쪽 프레임들간의 연관성이 매크로블록 타입의 종류를 결정한다. 참조하는 프레임들 사이에 2개의 B 프레임이 있다고 가정하고, 첫 번째 B 프레임이  $t$ 시점이라고 하였을 경우 forward 방향의 참조 프레임은  $t-1$ 에, backward 방향의 참조 프레임은  $t+2$ 에 존재하게 된다. 즉 B 프레임은 양쪽 방향에 걸쳐 모션 예측을 수행함으로써 양쪽 프레임들간의 연관성이 매크로블록 타입의 종류를 결정한다. 참조하는 프레임들 사이에 2개의 B 프레임 중 첫 번째 B 프레임을  $F_t$ 라고 하였을 경우 forward/backward 방향의 참조 프레임을 다음과 같이 정의할 수 있다.



(그림 1) I, P, B 프레임 화면의 예측방법



(그림 2) MPEG 프레임의 구조

$$F_i = \{f_i^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (2)$$

$$F_{i-1}^{i,j} = \{f_{i-1}^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (3)$$

$$F_{i+2}^{i,j} = \{f_{i+2}^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (4)$$

$F_{i-1}$ 은 forward 방향의 참조 프레임이고,  $F_{i+2}$ 는 backward 방향으로 참조 프레임이다.

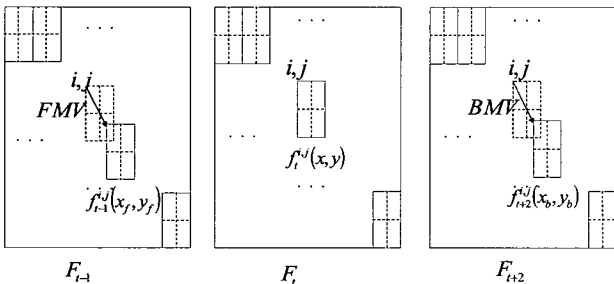
(그림 3)은  $F_i$ 에서  $F_{i-1}$ 와  $F_{i+2}$ 를 이용하여 모션예측을 수행하는 모습을 보여주고 있다.

(그림 3)에서  $f_{i-1}^{i,j}(x_f, y_f)$ 은  $F_{i-1}$ 에서 참조하는 매크로블록이고,  $f_{i+2}^{i,j}(x_b, y_b)$ 은  $F_{i+2}$ 에서 참조하는 매크로블록이다. 또한  $x_f, y_f$ 는  $x, y$ 가 forward 모션벡터인  $FMV = (FMV_x, FMV_y)$ 만큼 이동된 좌표이고,  $x_b, y_b$ 는  $x, y$ 가 backward 모션벡터인  $BMV = (BMV_x, BMV_y)$ 만큼 이동된 좌표이다.

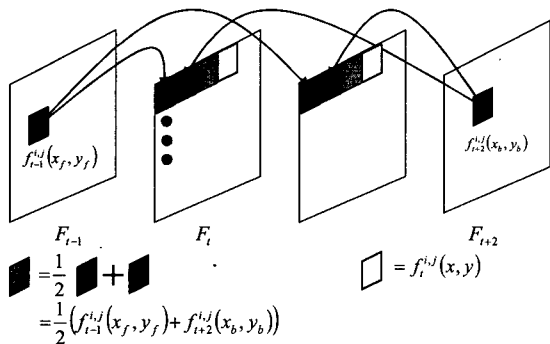
$$x_f = x + FMV_x, \quad y_f = y + FMV_y \quad (5)$$

$$x_b = x + BMV_x, \quad y_b = y + BMV_y \quad (6)$$

MPEG의 B 프레임은 일반적으로 4종류의 매크로블록 타입이 존재하는데 참조하는 영상의 방향에 따라서 매크로블록 타입이 *F Type*(forward 방향의 영상을 참조), *B Type*(backward 방향의 영상을 참조) 그리고 *Inter Type*(forward와 backward 양방향 영상의 평균을 참조)타입과 마지막으로 *Intra Type*(자기 자신의 매크로블록으로 인코딩 되는)으로 정의된다.



(그림 3) 참조 프레임을 결정하는 방법



(그림 4) 모션예측을 통한 B 프레임의 구성

(그림 4)는 MPEG 영상에서 B 프레임을 구성하는 매크로블록들의 집합을 보여주고 있다. 매크로블록 타입을 결정하는 것을 살펴보면 입력영상과 모션예측을 통하여 구한 영상의 차에 분산을 이용하여 결정하게 된다. (그림 4)는 첫 번째 B 프레임인  $F_i$ 가 영상을 구성하게 되는 인코딩 과정을 설명한다. 첫 번째, 두 번째, 세 번째 매크로블록은 각각 *F Type*, *B Type* 그리고 *Inter Type*이다.

(그림 4)에서 보는 바와 같이 각 방향의 참조 매크로블록 중 B 프레임에 대처하기 위하여 일반적으로 *SAD*(Sum of Absolute Difference)를 이용한다.

$$SADf_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} |f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) - f_i^{i,j}(x,y)| \quad (7-1)$$

$$SADb_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} |f_{i+2}^{i,j}(x+BMV_x, y+BMV_y) - f_i^{i,j}(x,y)| \quad (7-2)$$

$$SADI_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} \left[ \frac{1}{2} (f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) + f_{i+2}^{i,j}(x+BMV_x, y+BMV_y)) - f_i^{i,j}(x,y) \right] \quad (7-3)$$

$$SAD_i^{i,j}(x,y) = \min(SADf_i^{i,j}(x,y), SADb_i^{i,j}(x,y), SADI_i^{i,j}(x,y)) \quad (7-4)$$

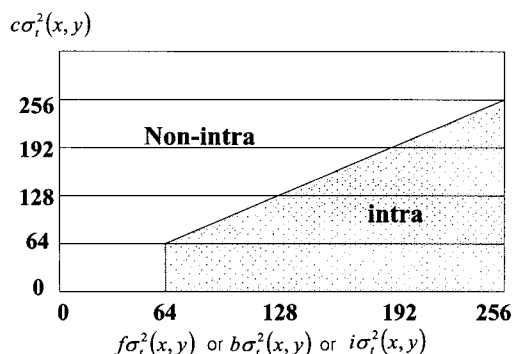
식(7-1)에서 식(7-3)는 현재 프레임과 이전/이후 그리고 양방향의 참조 프레임과의 *SAD*를 구하는 식이다. 식(7-1)에서 식(7-3)까지 구한 *SAD*중 최소가 되는 값을 식(7-4)처럼 구하게 되는데, 현 프레임과의 가장 적은 오차를 갖는 참조 매크로블록을 구하게 된다. 그러나 단지 *SAD*가 최소가 되는 매크로블록을 참조하는 것이 아니라 현 매크로블록과 참조 매크로블록과의 분산을 이용하여 참조할 것인지 아닌지를 판단하게 된다.

$$c\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} f_i^{i,j}(x,y)^2}{16 \times 16} - \left( \frac{\sum_{x=0,y=0}^{15} f_i^{i,j}(x,y)}{16 \times 16} \right)^2 \quad (8-1)$$

$$f\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} (f_i^{i,j}(x,y) - f_{i-1}^{i,j}(x+FMV_x, y+FMV_y))^2}{16 \times 16} \quad (8-2)$$

$$b\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} (f_i^{i,j}(x,y) - f_{i+2}^{i,j}(x+BMV_x, y+BMV_y))^2}{16 \times 16} \quad (8-3)$$

$$i\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} (f_i^{i,j}(x,y) - [f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) + f_{i+2}^{i,j}(x+BMV_x, y+BMV_y)])^2}{16 \times 16} \quad (8-4)$$



(그림 5) Intra/Non-Intra 결정

식(8-1)은 현 매크로블록의 분산을 구하고, 식(8-2), (8-3), (8-4)는 각각의 참조 매크로블록과의 차의 분산을 구하게 된다. 만약 식(7-4)의  $SAD^{i,j}(x, y)$ 가  $SADf^{i,j}(x, y)$ 일 경우  $f_i^{i,j}(x, y)$ 와  $f_{i-1}^{i,j}(x_f, y_f)$ 의 차가 가장 적은 값을 갖게 되는데, 참조하는 매크로블록을 결정하기 위하여 식(8-1)의  $c\sigma_i^2(x, y)$ 와 식(8-2)의  $f\sigma_i^2(x, y)$ 의 값을 비교한다. 즉,  $f_i^{i,j}(x, y)$ 의 분산값이  $f_{i-1}^{i,j}(x_f, y_f)$ 과 차의 분산값보다 작고, 차의 분산값이 임계값 이상이면 모션벡터를 이용하지 않는다. 다시 말해서  $SAD^{i,j}(x, y)$ 가 최소가 되는 매크로블록이 현 매크로블록과의 차이가 크게 되면  $f_i^{i,j}(x, y)$ 는 *IntraType*으로 인코딩 된다.

(그림 5)는 매크로블록의 타입 중 *IntraType*을 결정하는 임계치를 그래프로 나타내고 있다. 만약 식(8-1)의  $c\sigma_i^2(x, y)$ 이 식(8-2)의  $f\sigma_i^2(x, y)$ 보다 작을 경우 (그림 5)에서 보느냐와 같이  $f_i^{i,j}(x, y)$ 는 *IntraType*으로 인코딩되고  $c\sigma_i^2(x, y)$ 가  $f\sigma_i^2(x, y)$ 보다 클 경우 *Non-IntraType(FType)*으로 인코딩 된다.

#### 4. B 프레임의 매크로블록 개수를 이용한 급진적 장면전환 검출 알고리즘

B 프레임을 구성하는 매크로블록들의 타입은 참조하는 프레임에 따라서 식(9)과 같이 네 종류로 구분할 수 있다.

$$if \quad f_i^{i,j}(x, y) = \begin{cases} f_{i-1}^{i,j}(x_f, y_f), & FType \\ f_{i+2}^{i,j}(x_b, y_b), & BType \\ \frac{1}{2}(f_{i-1}^{i,j}(x_f, y_f) + f_{i+2}^{i,j}(x_b, y_b)), & InterType \\ f_i^{i,j}(x, y), & IntraType \end{cases} \quad (9)$$

$F_i$ 의 매크로블록인  $f_i^{i,j}(x, y)$ 가 *FType*으로 인코딩 되면,  $f_i^{i,j}(x, y) = f_{i-1}^{i,j}(x_f, y_f)$ 이므로  $F_i$ 에서 매크로블록 타입 중 *FType*의 개수는  $F_i$ 의 매크로블록 중  $F_{i-1}$ 에서 참조한 매크로블록인  $f_{i-1}^{i,j}(x_f, y_f)$ 의 개수가 된다.

*BType* 역시  $f_i^{i,j}(x, y)$ 가 *BType*으로 인코딩되면,

$f_i^{i,j}(x, y) = f_{i+2}^{i,j}(x_b, y_b)$ 이므로 *BType*의 개수는  $F_i$ 에서  $F_{i+2}$ 에서 참조한 매크로블록인  $f_{i+2}^{i,j}(x_b, y_b)$ 의 개수가 된다.

*InterType*으로 인코딩 되는 경우 (그림 4)에 의해 알 수 있듯이  $\frac{1}{2}f_{i-1}^{i,j}(x_f, y_f) + \frac{1}{2}f_{i+2}^{i,j}(x_b, y_b)$ 이므로,  $\frac{1}{2}f_{i-1}^{i,j}(x_f, y_f)$ 는  $F_{i-1}$ 에 포함되고,  $\frac{1}{2}f_{i+2}^{i,j}(x_b, y_b)$ 는  $F_{i+2}$ 에 포함되므로, *InterType*으로 인코딩 된 매크로블록은  $F_{i-1}$ 와  $F_{i+2}$ 에 각각 포함 된다.

*IntraType*의 경우 인코딩 되는 매크로블록이 forward나 backward 방향의 영상으로부터 참조를 하지 않고 자신의 영상 값으로 인코딩 된다.

(그림 6)에서 볼 수 있듯이 인코딩 되는 B 프레임은 forward 방향의 영상  $F_{i-1}$ 의 성분인  $f_{i-1}^{i,j}(x_f, y_f)$ 와 backward 방향의 영상  $F_{i+2}$ 의 성분인  $f_{i+2}^{i,j}(x_b, y_b)$ 의 조합으로 이루어지거나 어떤 방향으로도 참조하지 않은  $f_i^{i,j}(x, y)$ 의 조합으로 이루어진다. 그러므로  $F_i$ 는 식(10-1)에서 식(10-4)의 집합들로 식(11)처럼 나타낼 수 있다.

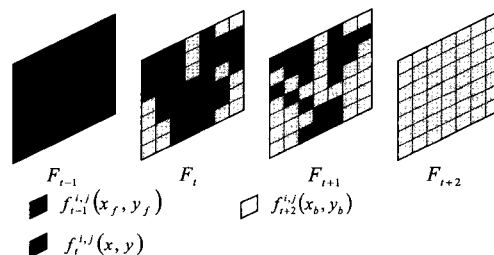
$$F_i^{sub1} = \{f_i^{i,j}(x, y) \mid f_i^{i,j}(x, y) = f_{i-1}^{i,j}(x_f, y_f)\} \quad (10-1)$$

$$F_i^{sub2} = \{f_i^{i,j}(x, y) \mid f_i^{i,j}(x, y) = f_{i+2}^{i,j}(x_b, y_b)\} \quad (10-2)$$

$$F_i^{sub3} = \left\{ f_i^{i,j}(x, y) \mid f_i^{i,j}(x, y) = \frac{1}{2}(f_{i-1}^{i,j}(x_f, y_f) + f_{i+2}^{i,j}(x_b, y_b)) \right\} \quad (10-3)$$

$$F_i^{sub4} = \left\{ f_i^{i,j}(x, y) \mid f_i^{i,j}(x, y) \notin F_i^{sub1}, f_i^{i,j}(x, y) \notin F_i^{sub2}, f_i^{i,j}(x, y) \notin F_i^{sub3} \right\} \quad (10-4)$$

$$F_i = F_i^{sub1} \cup F_i^{sub2} \cup F_i^{sub3} \cup F_i^{sub4} \quad (11)$$



(그림 6) 각 방향의 매크로블록의 집합으로 이루어진 B 프레임

식(11)의  $F_i$  집합에서 원소의 개수, 즉 매크로블록의 개수를  $n(F_i)$ 라고 정의하면,  $n(F_i)$ 는 식(12)과 같이 정의할 수 있다.

$$n(F_i) = n(F_i^{sub1}) + n(F_i^{sub2}) + n(F_i^{sub3}) + n(F_i^{sub4}) \quad (12)$$

식(10-3)의  $F_i^{sub3}$ 은  $\frac{1}{2}f_{i-1}^{i,j}(x_f, y_f)$ 과  $\frac{1}{2}f_{i+2}^{i,j}(x_b, y_b)$ 으로 이루

어져 있는데,  $\frac{1}{2}f_{i-1}^{i,j}(x_f, y_f)$ 는  $F_{i-1}$ 의 성분이고,  $\frac{1}{2}f_{i+2}^{i,j}(x_b, y_b)$ 는  $F_{i+2}$ 의 성분이므로,  $F_{i-1}$ 의 성분으로 이루어진 매크로블록의 개수를  $n(F_i^{subF})$ ,  $F_{i+2}$ 의 성분으로 이루어진 매크로블록의 개수를  $n(F_i^{subB})$ 라고 정의 하던  $n(F_i^{subF})$ 과  $n(F_i^{subB})$ 는 다음과 같이 나타낼 수 있다.

$$n(F_i^{subF}) = n(F_i^{sub1}) + \frac{1}{2}n(F_i^{sub3}) \quad (13-1)$$

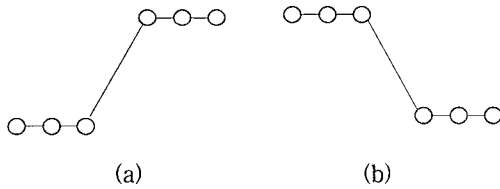
$$n(F_i^{subB}) = n(F_i^{sub2}) + \frac{1}{2}n(F_i^{sub3}) \quad (13-2)$$

식(13-1)과 식(13-2)를 이용하여 식(12)를 다음과 같이 표현할 수 있다.

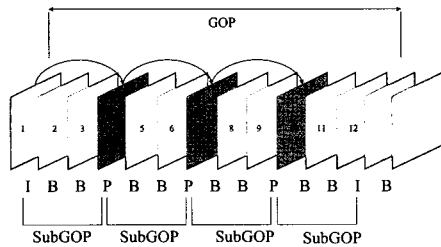
$$n(F_i) = n(F_i^{subF}) + n(F_i^{subB}) + n(F_i^{sub4}) \quad (14)$$

식(14)에서의  $n(F_i^{subF})$ 는 식(9)에 의하여  $F_i$ 에서  $FType$ 의 개수와  $\frac{1}{2}InterType$  개수의 합이 되고,  $n(F_i^{subB})$ 는  $BType$ 의 개수와  $\frac{1}{2}InterType$  개수의 합이 된다. 또한  $n(F_i^{sub4})$ 는  $IntraType$ 의 개수가 된다. 두 번째 B 프레임인  $F_{i+1}$ 도 같은 방법으로 해석한다.

(그림 7)과 같이 장면이 전환 되는 경우를 두 가지로 정의할 수 있다. (그림 7)의 (a)처럼 이전 장면보다 인텐서티의 급격한 증가로 인하여 장면이 전환되는 경우와 (그림 7)의 (b)처럼 이전 장면보다 인텐서티의 급격장 감소로 인하여 장면이 전환되는 경우로 정의할 수 있다. 또한 장면이 전환되기 이전 프레임과 장면전환 된 프레임과는 연관성이 매우 적게 나타나고 장면전환 된 프레임 이후의 프레임들은 장면전환 된 프레임을 포함한 장면이기 때문에 많은 연관성을 가지고 있다.



(그림 7) 장면전환 모델



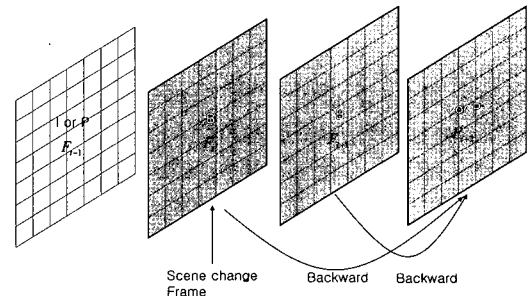
(그림 8) SubGOP의 정의

제안한 알고리즘을 이용하여 장면전환을 검출하기 위해 MPEG 시퀀스에서 I 또는 P 프레임부터 다음 I 또는 P 프레임까지는 (그림 8)처럼 SubGOP로 분할 하였다. 또한 I 또는 P 프레임 사이에 B 프레임은 2개가 존재 한다고 가정하였다.

본 논문에서는 SubGOP내에 장면전환이 발생한 경우를 3가지로 정의할 수 있다.  $F_i$ 를 SubGOP내의 첫번째 B 프레임이라 가정하면, 첫 번째는  $F_i$ 에서 장면전환이 발생한 경우이고, 두 번째로 두 번째 B 프레임인  $F_{i+1}$ 에서 장면전환이 발생한 경우이고, 마지막으로 참조 프레임인  $F_{i+2}$ 에서 장면전환이 발생한 경우이다.

4.1  $F_i$ 에서 장면전환이 발생 한 경우

(그림 9)은 첫 번째 B 프레임인  $F_i$ 에서 장면전환이 발생한 모습을 보이고 있다.



(그림 9)  $F_i$ 에서 장면전환이 발생

$F_i$ 에서 장면 전환이 발생하였다고 가정 하였으므로 장면 전환 발생 프레임인  $F_i$ 와 이전 프레임인  $F_{i-1}$ 과 장면전환이 발생한 이후 프레임인  $F_{i+1}$ ,  $F_{i+2}$ 는 다음과 같이 정의 할 수 있다.

$$F_{i-1}^{i,j} = \{f_{i-1}^{i,j}(x, y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (15-1)$$

$$F_i = \{f_i^{i,j}(x, y) + \delta_i^{i,j}(x, y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (15-2)$$

$$F_{i+1} = \{f_{i+1}^{i,j}(x, y) + \delta_{i+1}^{i,j}(x, y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (15-3)$$

$$F_{i+2} = \{f_{i+2}^{i,j}(x, y) + \delta_{i+2}^{i,j}(x, y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (15-4)$$

$F_i$ 는 이전 프레임인  $F_{i-1}$ 에서  $\delta_i^{i,j}(x, y)$ 만큼 intensity의 변화로 인하여 장면전환이 발생하였으므로  $\delta_i^{i,j}(x, y)$ 는 매우 큰 값을 갖고 장면전환이 발생한 이후 프레임인  $F_{i+1}$ ,  $F_{i+2}$ 는  $F_i$ 와 같은 장면에 존재하는 프레임이므로  $\delta_i^{i,j}(x, y)$ 만큼 intensity가 변화된 상태로 나타난다. 또한 같은 장면에 포함되어 있는 프레임들은 intensity의 변화가 매우 적다고 가정한다. 즉, 식(15-2), (15-3), (15-4)의  $\delta_i^{i,j}(x, y)$ 와 참조되는 매크로블록은 다음과 같은 특성을 갖는다.

$$\delta_i^{i,j}(x,y) \cong \delta_{i+1}^{i,j}(x,y) \cong \delta_{i+2}^{i,j}(x,y) \quad (16-1)$$

$$f_i^{i,j}(x,y) \cong f_{i+1}^{i,j}(x,y) \cong f_{i-1}^{i,j}(x_f, y_f) \cong f_{i+2}^{i,j}(x_b, y_b) \quad (16-2)$$

위에서  $F_i$ 는 첫 번째 B 프레임이라 가정하였으므로  $F_i$ 는  $F_{i-1}$ 과  $F_{i+2}$ 에서 식(7-1)에서 식(7-3)까지 이용하여 SAD를 구한다.

$$SADf_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} |f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) - (f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y))| \cong \sum_{x=0,y=0}^{15} |\delta_i^{i,j}(x,y)| \quad (17-1)$$

$$SADb_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} (f_{i+2}^{i,j}(x+BMV_x, y+BMV_y) + \delta_{i+2}^{i,j}(x,y)) - (f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y)) \cong 0 \quad (17-2)$$

$$SADi_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} \left| \frac{1}{2} [f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) + (f_{i+2}^{i,j}(x+BMV_x, y+BMV_y) + \delta_{i+2}^{i,j}(x,y))] - (f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y)) \right| \cong \sum_{x=0,y=0}^{15} \left| \frac{1}{2} \delta_{i+2}^{i,j}(x,y) - \delta_i^{i,j}(x,y) \right| \quad (17-3)$$

식(17-1)에서 식(17-3)까지 구한 SAD를 식(7-4)를 이용하여 최소가 되는  $SAD_i^{i,j}(x,y)$ 를 구하게 되면 다음과 같다.

$$SAD_i^{i,j}(x,y) = \min(SADf_i^{i,j}(x,y), SADb_i^{i,j}(x,y), SADi_i^{i,j}(x,y)) = SADb_i^{i,j}(x,y) \quad (18)$$

식(18)의  $SAD_i^{i,j}(x,y)$ 가  $SADb_i^{i,j}(x,y)$ 이므로  $f_i^{i,j}(x,y)$ 와 가장 적은 오차를 갖는 매크로블록은  $f_{i+2}^{i,j}(x_b, y_b)$ 가 된다. 위에서 구한  $f_{i+2}^{i,j}(x_b, y_b)$ 와  $f_i^{i,j}(x,y)$ 를 식(8-1)과 식(8-3)을 이용하여 분산을 구하여 보면 다음과 같다.

$$c\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y)]^2}{16 \times 16} - \left( \frac{\sum_{x=0,y=0}^{15} [f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y)]}{16 \times 16} \right)^2 \quad (19-1)$$

$$b\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y)] - [f_{i+2}^{i,j}(x+BMV_x, y+BMV_y) + \delta_{i+2}^{i,j}(x,y)]^2}{16 \times 16} \cong 0 \quad (19-2)$$

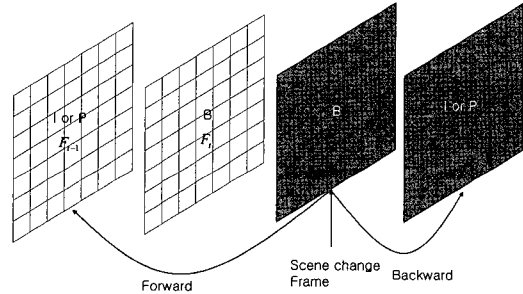
식(19-1)과 식(19-2)에서 구한 분산을 이용하여  $f_i^{i,j}(x,y)$ 을 이용할 것인지  $f_{i+2}^{i,j}(x_b, y_b)$ 를 이용할 것인지 결정하게 되는데

$c\sigma_i^2(x,y) > b\sigma_i^2(x,y)$ 이므로  $f_{i+2}^{i,j}(x_b, y_b)$ 으로 대체한다. 즉  $f_i^{i,j}(x,y)$ 는 B Type으로 인코딩된다. 위에서 증명한 바와 같이 첫 번째 B 프레임에서 장면전환이 발생되면 장면전환이 발생된  $F_i$ 는 대부분  $F_{i+2}$ 의 성분인  $f_{i+2}^{i,j}(x_b, y_b)$ 로 대체됨을 알 수 있다. 두 번째 B 프레임에서의 특징을 검출하기 위하여 식(15-3)을 식(17-1)에서 식(17-3)까지 각각 대입하여  $SAD_{i+1}^{i,j}(x,y)$ 를 구하여 보면 첫 번째 B 프레임과 같은 특징을 갖는다. 즉,  $SAD_{i+1}^{i,j}(x,y)$  역시  $SADb_{i+1}^{i,j}(x,y)$ 로 나타나고, 식(8-1)과 식(8-3)의 분산 역시  $c\sigma_{i+1}^2(x,y) > b\sigma_{i+1}^2(x,y)$ 의 특징을 나타낸다. 그러므로, SubGOP의 B 프레임들의 매크로블록 개수가 다음과 같이 나타날 경우 첫 번째 B 프레임( $F_i$ )에서 장면전환을 검출할 수 있다.

$$n(F_i) - n(F_i^{sub4}) \cong n(F_i^{subB}) \quad \text{and} \quad n(F_{i+1}) - n(F_{i+1}^{sub4}) \cong n(F_{i+1}^{subB}) \quad (20)$$

식(20)에서  $F_i^{sub4}$ 의 개수는 이전 또는 이후 프레임과의 연관성이 존재하지 않는 매크로블록의 집합이므로 장면전환을 검출하는데 있어서 에러로 가정할 수 있다.

#### 4.2 $F_{i+1}$ 에서 장면전환이 발생 한 경우



(그림 10)  $F_{t+1}$ 에서 장면전환이 발생

두 번째로 SubGOP내의 두 번째 B 프레임인  $F_{i+1}$ 에서 장면전환이 발생하였다고 가정하면, 식(15-1)에서 식(15-4)까지의 SubGOP내 프레임들이 다음과 같이 정의된다.

$$F_{i-1} = \{f_{i-1}^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (21-1)$$

$$F_i = \{f_i^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (21-2)$$

$$F_{i+1} = \{f_{i+1}^{i,j}(x,y) + \delta_{i+1}^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (21-3)$$

$$F_{i+2} = \{f_{i+2}^{i,j}(x,y) + \delta_{i+2}^{i,j}(x,y) | i=0,1,\dots,M-1 \text{ and } j=0,1,\dots,N-1\} \quad (21-4)$$

첫 번째 B 프레임인  $F_i$ 를 식(7-1), 식(7-2), 식(7-3)에 적용하여 식(7-4)의  $SAD_i^{i,j}(x,y)$ 를 구하여 보면 다음과 같다.

$$SADf_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} |f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) - (f_i^{i,j}(x,y))| \cong 0 \quad (22-1)$$

$$SADb_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} (f_{i+2}^{i,j}(x+BMV_x, y+BMV_y) + \delta_{i+2}^{i,j}(x,y)) - (f_i^{i,j}(x,y)) \cong \sum_{x=0,y=0}^{15} \delta_{i+2}^{i,j}(x,y) \quad (22-2)$$

$$SADi_i^{i,j}(x,y) = \sum_{x=0,y=0}^{15} \left[ \frac{1}{2} [f_{i-1}^{i,j}(x+FMV_x, y+FMV_y) + (f_{i+2}^{i,j}(x+BMV_x, y+BMV_y) + \delta_{i+2}^{i,j}(x,y))] - (f_i^{i,j}(x,y)) \right] \cong \sum_{x=0,y=0}^{15} \left[ \frac{1}{2} \delta_{i+2}^{i,j}(x,y) - \delta_i^{i,j}(x,y) \right] \quad (22-3)$$

$$SAD_i^{i,j}(x,y) = SADf_i^{i,j}(x,y) \quad (23)$$

식(23)의  $SAD_i^{i,j}(x,y)$ 가  $SADf_i^{i,j}(x,y)$ 이므로  $f_i^{i,j}(x,y)$ 와 가장 적은 오차를 갖는 매크로블록은  $f_{i-1}^{i,j}(x_f, y_f)$ 가 된다. 위에서 구한  $f_{i-1}^{i,j}(x_f, y_f)$ 와  $f_i^{i,j}(x,y)$ 를 식(8-1)과 식(8-2)를 이용하여 분산을 구하여 보면 다음과 같다.

$$c\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [f_i^{i,j}(x,y) + \delta_i^{i,j}(x,y)]^2}{16 \times 16} - \left( \frac{\sum_{x=0,y=0}^{15} [f_i^{i,j}(x,y)]}{16 \times 16} \right)^2 \quad (24-1)$$

$$f\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} (f_i^{i,j}(x,y) - [f_{i-1}^{i,j}(x+FMV_x, y+FMV_y)])^2}{16 \times 16} \cong 0 \quad (24-2)$$

식(24-1)과 식(24-2)에서 구한 분산을 이용하여  $f_i^{i,j}(x,y)$ 을 이용할 것인지  $f_{i-1}^{i,j}(x_f, y_f)$ 를 이용할 것인지 결정하게 되는데  $c\sigma_i^2(x,y) > f\sigma_i^2(x,y)$ 이므로  $F_i$ 의 매크로블록인  $f_i^{i,j}(x,y)$ 는  $f_{i-1}^{i,j}(x_f, y_f)$ 으로 대체되므로  $f_i^{i,j}(x,y)$ 는  $FType$ 으로 인코딩된다. 즉,  $F_i$ 의 대부분은  $F_{i-1}$ 의 성분인  $f_{i-1}^{i,j}(x_f, y_f)$ 로 이루어진다. 다음으로 장면전환이 발생된 프레임인  $F_{i+1}$ 의 성분을 검출하기 위해 식(21-3)을 식(7-1)에서 식(7-3)까지 대입하여  $SAD_{i+1}^{i,j}(x,y)$ 를 구하여 보면,  $SAD_{i+1}^{i,j}(x,y)$ 는 첫 번째 B 프레임에 장면전환이 발생한 경우와 마찬가지로 식(18)처럼  $SADb_{i+1}^{i,j}(x,y)$ 로 구할 수 있다.  $SADb_{i+1}^{i,j}(x,y)$ 로 구한  $f_{i+2}^{i,j}(x_b, y_b)$ 를 식(8-1)과 식(8-3)을 이용하여 분산을 구하여 보면, 식(25-1)과 식(25-2)처럼 구할 수 있다.

$$c\sigma_{i+1}^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [f_{i+1}^{i,j}(x,y) + \delta_{i+1}^{i,j}(x,y)]^2}{16 \times 16} - \left( \frac{\sum_{x=0,y=0}^{15} [f_{i+1}^{i,j}(x,y)]}{16 \times 16} \right)^2 \quad (25-1)$$

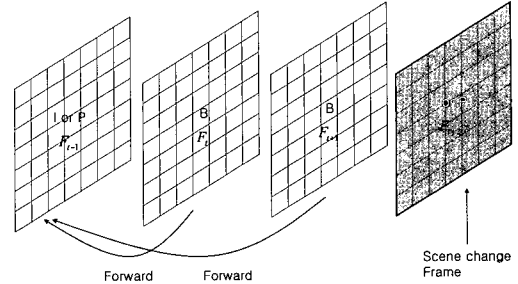
$$b\sigma_{i+1}^2(x,y) \cong 0 \quad (25-2)$$

식(25-1)과 식(25-2)의 분산은  $c\sigma_{i+1}^2(x,y) > b\sigma_{i+1}^2(x,y)$ 이므로  $F_{i+1}$ 의 매크로블록인  $f_{i+1}^{i,j}(x,y)$ 는  $f_{i+1}^{i,j}(x_b, y_b)$ 으로 대체되므로

$f_{i+1}^{i,j}(x,y)$ 는  $BType$ 으로 인코딩된다. 다시 말하면,  $F_{i+1}$ 의 대부분은  $F_{i+1}$ 의 성분인  $f_{i+1}^{i,j}(x_b, y_b)$ 로 이루어진다. 그러므로 SubGOP의 B 프레임들의 매크로블록 개수가 다음과 같이 나타날 경우 두 번째 B 프레임( $F_{i+1}$ )에서 장면전환을 검출할 수 있다.

$$n(F_i) - n(F_i^{sub4}) \cong n(F_i^{subF}) \quad \text{and} \quad n(F_{i+1}) - n(F_{i+1}^{sub4}) \cong n(F_{i+1}^{subB}) \quad (26)$$

### 4.3 $F_{i+2}$ 에서 장면전환이 발생한 경우



(그림 11)  $F_{i+2}$ 에서 장면전환이 발생

(그림 11)은 참조 프레임인  $F_{i+2}$ 에서 장면전환이 발생한 모습을 보이고 있다.

$F_{i+2}$ 에서 장면 전환이 발생하였다고 가정 하였으므로 장면 전환 발생 프레임인  $F_{i+2}$ 와 이전 프레임인  $F_{i-1}$ ,  $F_i$ 는 다음과 같이 정의 할 수 있다.

$$F_{i-1} = \{f_{i-1}^{i,j}(x,y) | i=0,1,\dots, M-1 \text{ and } j=0,1,\dots, N-1\} \quad (27-1)$$

$$F_i = \{f_i^{i,j}(x,y) | i=0,1,\dots, M-1 \text{ and } j=0,1,\dots, N-1\} \quad (27-2)$$

$$F_{i+1} = \{f_{i+1}^{i,j}(x,y) | i=0,1,\dots, M-1 \text{ and } j=0,1,\dots, N-1\} \quad (27-3)$$

$$F_{i+2} = \{f_{i+2}^{i,j}(x,y) + \delta_{i+2}^{i,j}(x,y) | i=0,1,\dots, M-1 \text{ and } j=0,1,\dots, N-1\} \quad (27-4)$$

참조 프레임에서 장면전환이 발생한 경우는 식(27-2)는 식(15-2)와 식(27-3)는 식(15-3)과 동일한 특징을 나타낸다. 그리고 첫 번째 B 프레임에서 장면전환이 발생한 1)의 경우와 반대의 특징을 나타내는데 즉, 이후 방향의 참조 프레임인  $F_{i+2}$ 에서 장면전환이 발생하였으므로 식(27-4)처럼  $F_{i+2}$ 에서  $\delta_{i+2}^{i,j}(x,y)$ 만큼 증가된 인텐서티를 나타낸다. 그러므로 SubGOP 내의 B 프레임들인  $F_i$ 와  $F_{i+1}$ 에서  $SAD_i^{i,j}(x,y)$ 와  $SAD_{i+1}^{i,j}(x,y)$ 를 구하여 보면 다음과 같다.

$$SAD_i^{i,j}(x,y) = SADf_i^{i,j}(x,y) \quad (28)$$

$$SAD_{i+1}^{i,j}(x,y) = SADf_{i+1}^{i,j}(x,y) \quad (29)$$

식(28)의  $SAD_i^{i,j}(x,y)$ 와 식(29)의  $SAD_{i+1}^{i,j}(x,y)$ 가 각각  $SADf_i^{i,j}(x,y)$ ,  $SADf_{i+1}^{i,j}(x,y)$ 이므로  $f_i^{i,j}(x,y)$ 와  $f_{i+1}^{i,j}(x,y)$ 은  $f_{i-1}^{i,j}(x_f, y_f)$ 와 가장 적은 오차를 갖는다. 식(8-1)과 식(8-2)를 이용하여 각각의 분



산을 구하여 보면 다음과 같다.

$$c\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [f_t^{i,j}(x,y) + \delta_t^{i,j}(x,y)]^2}{16 \times 16} - \left( \frac{\sum_{x=0,y=0}^{15} [f_t^{i,j}(x,y)]}{16 \times 16} \right)^2 \quad (30-1)$$

$$f\sigma_i^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [(f_t^{i,j}(x,y)) - (f_{t-1}^{i,j}(x+FMV_x, y+FMV_y))]^2}{16 \times 16} \cong 0 \quad (30-2)$$

$$c\sigma_{i+1}^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [f_{t+1}^{i,j}(x,y) + \delta_{t+1}^{i,j}(x,y)]^2}{16 \times 16} - \left( \frac{\sum_{x=0,y=0}^{15} [f_{t+1}^{i,j}(x,y)]}{16 \times 16} \right)^2 \quad (31-1)$$

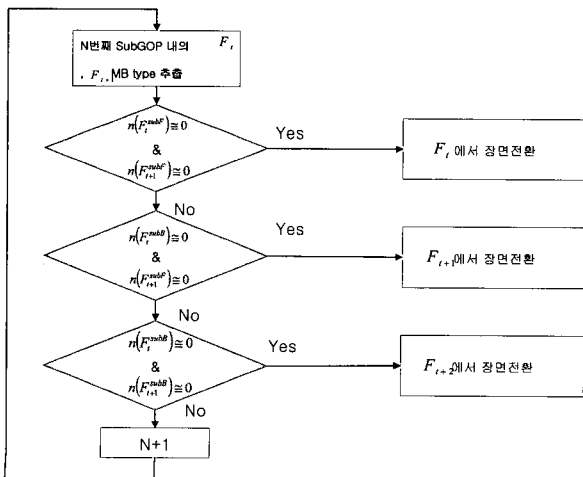
$$f\sigma_{i+1}^2(x,y) = \frac{\sum_{x=0,y=0}^{15} [(f_{t+1}^{i,j}(x,y)) - (f_{t+1}^{i,j}(x+FMV_x, y+FMV_y))]^2}{16 \times 16} \cong 0 \quad (31-2)$$

식(30-1)에서 식(30-2)와 식(31-1)과 식(31-2)에서 구한 분산을 살펴 보면,  $c\sigma_i^2(x,y) > f\sigma_i^2(x,y)$  이고  $c\sigma_{i+1}^2(x,y) > f\sigma_{i+1}^2(x,y)$  이므로  $F_i$ 의 매크로블록인  $f_t^{i,j}(x,y)$ 와  $F_{i+1}$ 의 매크로블록인  $f_{t+1}^{i,j}(x,y)$ 은  $f_{t-1}^{i,j}(x_f, y_f)$ 으로 대체되므로  $f_t^{i,j}(x,y)$ 과  $f_{t+1}^{i,j}(x,y)$ 는  $F_{i-1}$  Type으로 인코딩된다. 즉,  $F_i$ 와  $F_{i+1}$ 의 대부분은  $F_{i-1}$ 의 성분인  $f_{t-1}^{i,j}(x_f, y_f)$ 로 이루어 진다. 그러므로 SubGOP의 B 프레임들의 매크로블록 개수가 다음과 같이 나타날 경우 이후 참조 프레임( $F_{i+2}$ )에서 장면전환을 검출할 수 있다.

$$n(F_i) - n(F_i^{sub4}) \cong n(F_i^{subF}) \quad \text{and} \quad n(F_{i+1}) - n(F_{i+1}^{sub4}) \cong n(F_{i+1}^{subF}) \quad (32)$$

$F_{i-1}$  프레임은 이전 SubGOP의  $F_{i+2}$  프레임이므로 위의 3가지 경우가 SubGOP 내에서 발생할 수 있는 장면전환의 모든 종류가 된다.

(그림 12)에 제안한 급진적 장면전환 검출 알고리즘을 나타내었다.



(그림 12) 제안한 급진적 장면전환 검출 알고리즘

### 5. 실험 결과

본 논문에서는 Windows 2000 환경에서 C++ 6.0을 사용하여 제안한 알고리즘을 구현하였다. 동영상의 장면전환 검출에 대한 성능을 대비하기 위하여 일반적인 비압축 영역에서의 검색과 제안한 압축 영역에서의 검색 방법을 실험을 통하여 비교 분석하였다. 실험대상 동영상은 MPEG으로 압축된 video 5개를 선정하였고, 영상 길이는 각 영상의 종류별 방영시간의 특성을 고려하여 가장 보편적인 값으로 정하였다.

<표 1> 실험 대상 동영상의 특성

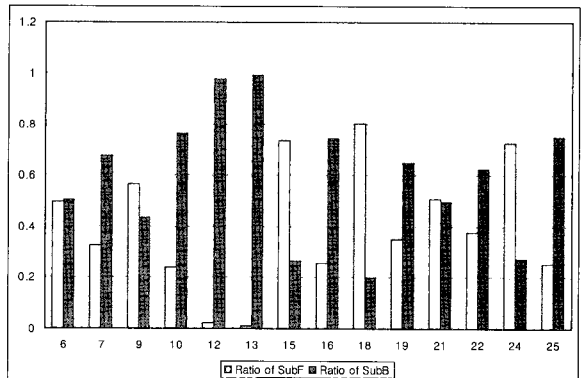
	video 1	video 2	video 3	video 4	video 5
총 프레임수(N)	449	440	807	837	418
장면전환 수	10	8	12	18	8

<표 1>은 실험 video의 특성을 보인 것이다. 실제 장면전환의 위치와 검출된 장면전환의 비교를 위하여 우선 사람의 눈으로 각 동영상에서 정확한 장면전환 위치를 검출하였으며, 이 데이터를 실제 장면전환의 근거로 사용하였다. 제안한 급진적 장면전환 검출 알고리즘의 실험을 위하여 비압축 영역의 히스토그램 차를 사용하는 방법과 압축 영역의 기존 매크로 블록 타입의 개수를 이용하는 방법을 제안한 알고리즘과 비교 실험하였다. 또한  $n(F_i^{subF})$ 의 비율을 검출하기 위하여 식(33-1)의  $R_i^{subF}$ 를 이용하였고  $n(F_i^{subB})$ 의 비율을 검출하기 위하여 식(33-2)의  $R_i^{subB}$ 를 이용하였다. 각각의 임계값으로 0.8을 적용하여 검출하였다.

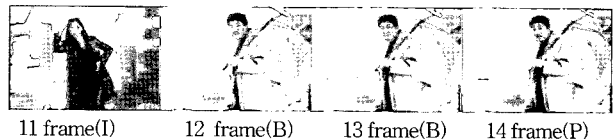
$$R_i^{subF} = \frac{n(F_i^{subF})}{n(F_i) - n(F_i^{sub4})} \quad (33-1)$$

$$R_i^{subB} = \frac{n(F_i^{subB})}{n(F_i) - n(F_i^{sub4})} \quad (33-2)$$

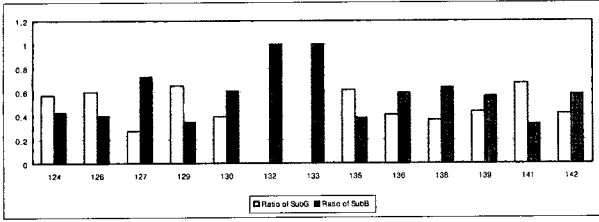
#### 5.1 $F_i$ 에서 장면전환 검출



(그림 13) video1의 6에서 25 프레임까지의  $R_i^{subF}$ 와  $R_i^{subB}$ 의 변화



(그림 14) video1의  $F_i$ 에서 장면전환 프레임



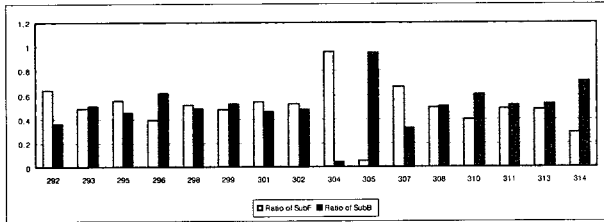
(그림15) video2의 124에서 142 프레임까지의  $R_i^{subF}$  와  $R_i^{subB}$  의 변화



131 frame(P) 132 frame(B) 133 frame(B) 134 frame(P)  
(그림 16) video2의  $F_i$ 에서 장면전환 프레임

(그림 13)의 12번째 프레임, 13째 프레임 (그림 15)의 132프레임, 133프레임의  $R_i^{subF}$  와  $R_{i+1}^{subF}$  의 값이 임계치 보다 크게 나타난다. 즉, 4장의 1)의 경우처럼  $F_i$ (그림14)의 11번째 프레임과 (그림 16)의 131번째 프레임)에서 장면전환을 검출할 수 있다. (그림 14)와 (그림16)는 실제 장면전환된 프레임의 모습을 나타내었다.

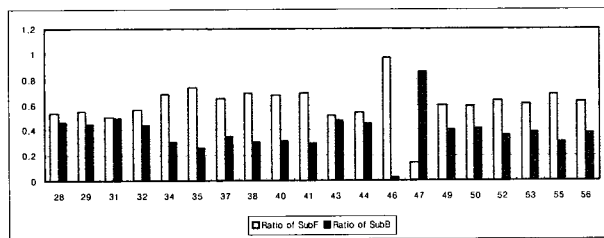
5.2  $F_{i+1}$ 에서 장면전환 검출



(그림17) video3의 292에서 314 프레임까지의  $R_i^{subF}$  와  $R_i^{subB}$  의 변화



303 frame(P) 304 frame(B) 305 frame(B) 306 frame(I)  
(그림 18) video3의  $F_{i+1}$ 에서 장면전환 프레임



(그림19) video4의 28에서 56 프레임까지의  $R_i^{subF}$  와  $R_i^{subB}$  의 변화



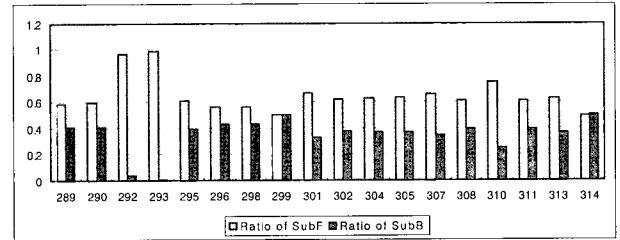
45 frame(I) 46 frame(B) 47 frame(B) 48 frame(P)  
(그림 20) video 4의  $F_{i+1}$ 에서 장면전환 프레임

(그림 17)의 304번째 프레임과 (그림 19)의 46프레임 경우  $R_i^{subF}$  의 값이 매우 크게 나타나고, (그림17)의 305번째 프레임과 (그림 19)의 47번째 프레임 경우  $R_i^{subB}$  의 값이 매우 크게 나타나고 있다. 4장의 2)의 경우처럼  $n(F_i) - n(F_i^{sub4}) \cong n(F_i^{subF})$ ,  $n(F_{i+1}) - n(F_{i+1}^{sub4}) \cong n(F_{i+1}^{subB})$ 의 모습을 나타내고,  $R_i^{subF} \cong 1$ ,  $R_{i+1}^{subB} \cong 1$  이므로 video3의  $F_{i+1}$  프레임인 205번째 프레임과 video4의  $F_{i+1}$  프레임인 47번째 프레임에서 장면전환이 검출된다. (그림 18)과 (그림 20)은 실제 장면전환된 프레임의 모습을 나타내었다.

5.3  $F_{i+2}$ 에서 장면전환 검출

(그림 21)의 292번째 프레임, 293째 프레임 (그림 23)의 232프레임, 233프레임의  $R_i^{subF}$  값이 매우 크게 나타남을 알 수 있다. 4장의 3)의 경우처럼  $n(F_i) - n(F_i^{sub4}) \cong n(F_i^{subF})$ ,  $n(F_{i+1}) - n(F_{i+1}^{sub4}) \cong n(F_{i+1}^{subF})$  이므로 I 또는 P 프레임( $F_{i+2}$ )에서 장면 전환을 검출할 수 있다. 그러므로 video4의 294번째 프레임과 video5의 234 프레임에서 장면전환이 검출된다.

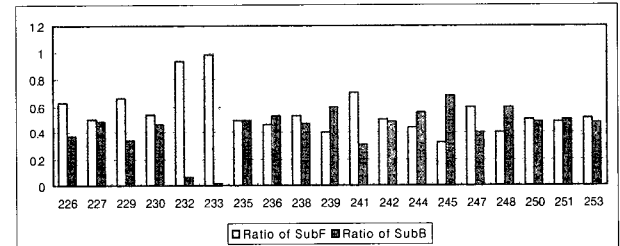
(그림 22)과 (그림 24)는 실제 장면전환된 프레임의 모습을 나타내었다.



(그림21) video4의 289에서 314 프레임까지의  $R_i^{subF}$  와  $R_i^{subB}$  의 변화



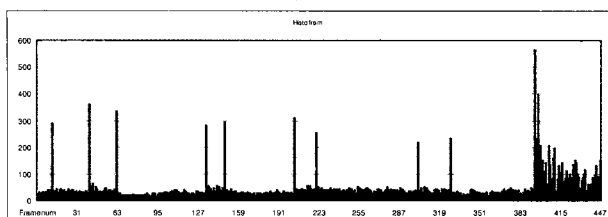
291 frame(P) 292 frame(B) 293 frame(B) 294 frame(P)  
(그림 22) video 4의  $F_{i+2}$ 에서 장면전환 프레임



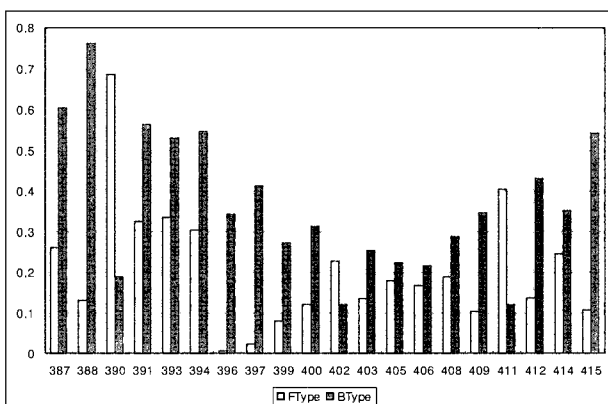
(그림23) video5의 226에서 253 프레임까지의  $R_i^{subF}$  와  $R_i^{subB}$  의 변화



231 frame(P) 232 frame(B) 233 frame(B) 234 frame(P)  
(그림 24) video 5의  $F_{i+2}$ 에서 장면전환 프레임



(그림 25) video1의 히스토그램 차의 분포



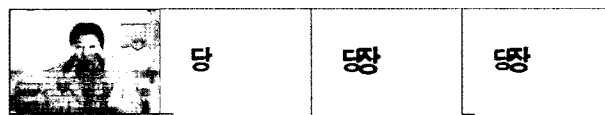
(그림 26) video1의 387에서 415 프레임까지의 FType와 BType 비율의 변화

(그림 26)는 기존의 매크로 블록 타입을 이용한 장면전환 검출 알고리즘에서 forward 타입과 backward 타입의 변화를 보여준다. (그림 26)에서 알 수 있듯이 기존 매크로블록 타입

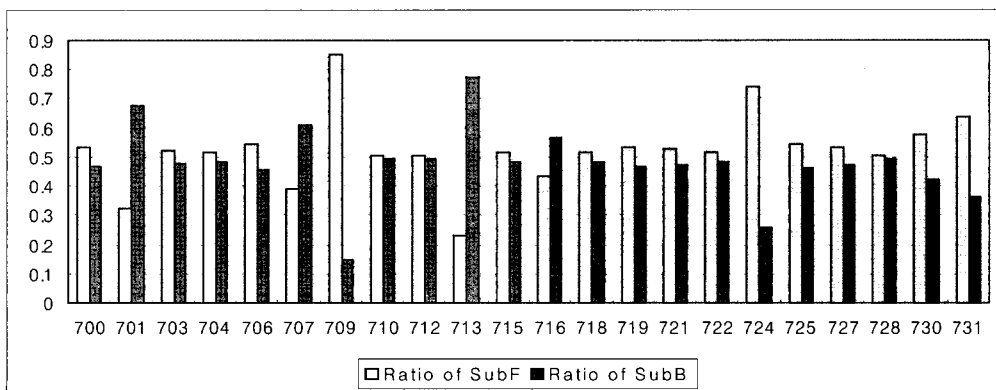
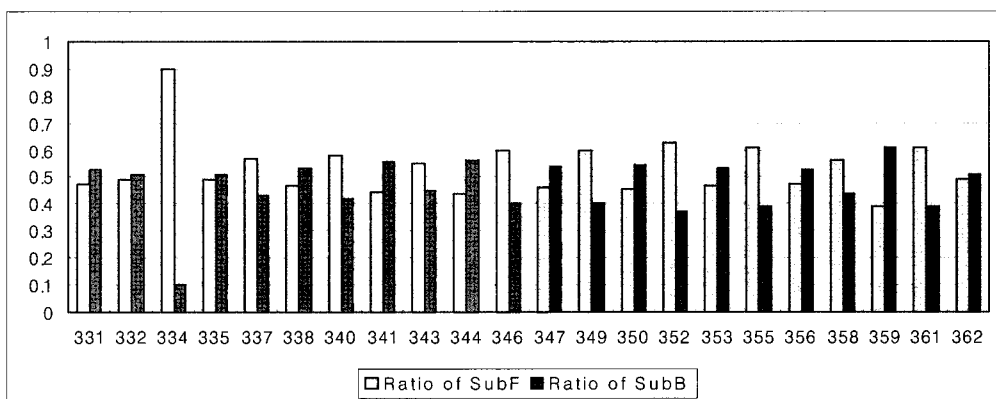
을 이용한 알고리즘을 적용하였을 경우 임계값을 0.6으로 하였을 경우 387번째 프레임과 388번째 프레임에서 backward 타입의 비율이 임계치 이상을 나타내므로 388번째 프레임에서 오검출이 발생하게 되고, 396번째 프레임과 397번째 프레임의 backward 비율은 임계치 이하로 나타나므로 장면전환을 검출하지 못한다. 즉, 기존 매크로블록 타입의 개수를 이용한 알고리즘은 영상의 변화에 의하여 InterType나 IntraType의 개수가 증가 할 경우 임계값 설정의 어려움으로 인하여 오검출 및 미검출이 발생한다. (그림 27)와 (그림 28)은 오검출된 프레임 및 미검출된 프레임을 보여준다. 또한 히스토그램 차를 이용하여 장면전환을 검출할 시 연속된 두 프레임과의 히스토그램 차의 비교를 통하여 차가 임계값 이상이 되면, 장면



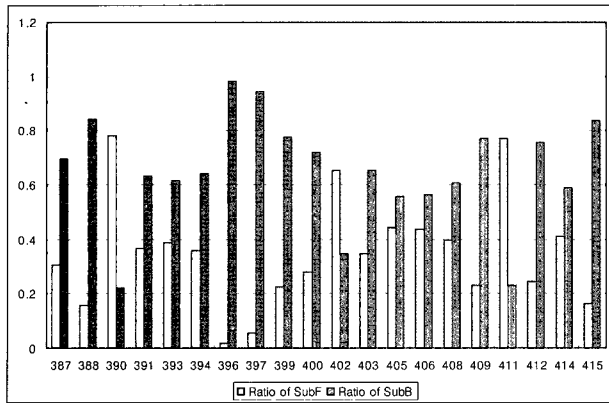
(그림 27) 기존 매크로블록 타입개수를 이용시 오검출된 프레임



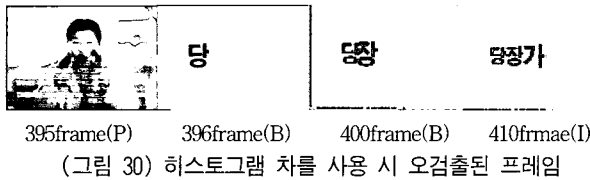
(그림 28) 기존 매크로블록 타입개수를 이용시 미검출된 프레임



(그림 31) 제안한 알고리즘 적용시 video3의 미검출 구간의  $R_{subF}$ 와  $R_{subB}$ 의 변화

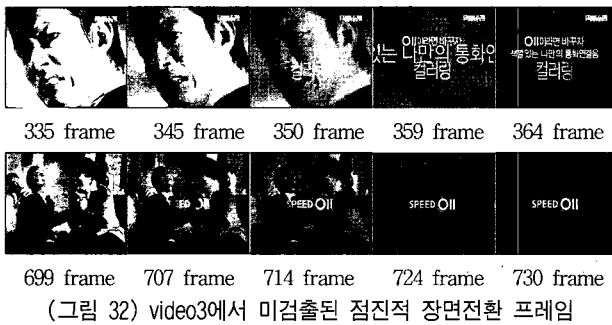


(그림 29) video1의 390에서 415 프레임까지의  $R_i^{subF}$  와  $R_i^{subB}$  의 변화



(그림 30) 히스토그램 차를 사용 시 오검출된 프레임

전환으로 인식하는데, video1 같이 동영상 내에 물체의 움직임이 발생하게 되면 (그림 25)에서 보는 바와 같이 395번째 프레임 이후부터 연속된 두 프레임간의 히스토그램 차가 크게 발생된다. 그러므로 실제 장면전환이 아닌 프레임들을 장면전환으로 검출하는 오검출이 발생하게 된다. 그러나 제안한 알고리즘을 적용하였을 경우 이전/이후 프레임들과의 비교를 통하여 (그림 29)에서처럼  $F_i$ 인 396프레임과  $F_{i+1}$ 인 397 프레임에서  $R_i^{subF}$  과  $R_{i+1}^{subF}$  이 임계치 이상으로 나타나므로  $F_{i+2}$ 인 396 프레임에서 장면전환을 검출 할 수 있고, 그 이후의



(그림 32) video3에서 미검출된 점진적 장면전환 프레임

SubGOP내의 B 프레임들에서는 위에서 설명한 3 가지 경우에 만족하는 부분들이 나타나지 않으므로 제안한 알고리즘을 적용할 경우 히스토그램 차를 이용한 알고리즘에서 발생한 오검출이 발생하지 않는다.

(그림 30)의 400frame, 410frame은 히스토그램 차를 이용할 경우 발생하는 오검출된 프레임들이다.

(그림 31)의 그래프를 살펴보면 위에서 제안한 3가지의 장면전환 경우에 적용되는  $R_i^{subF}$  와  $R_i^{subB}$  가 존재하지 않으므로 제안한 알고리즘을 적용하였을 경우 장면전환을 검출하지 못한다. (그림 32)에서 보는 바와 같이 점진적으로 장면이 변화하는 프레임들이 존재하지만, 점진적인 장면전환의 경우  $R_i^{subF}$  와  $R_i^{subB}$  가 뚜렷한 특징이 없이 변화하므로 검출하지 못하였다.

<표 2>에서 보면 video2과 video3의 경우 점진적 장면전환이 존재하는 영상으로 제안한 알고리즘을 이용하였을 경우 점진적 장면전환 구간을 검출하지 못하는 문제가 발생하였다. video3은 기존 알고리즘을 이용하였을 경우 점진적 장면전환 구간의 히스토그램 차가 크게 발생하여 장면전환 프레임은 검출하였으나 이후에 나오는 프레임들도 장면전환을 검출하는 오검출이 발생하였다. 그러나 video2의 경우 점진적 장면전환 구간의 히스토그램 차가 적게 나타나므로 히스토그램 차를 이용하였을 경우 역시 이 구간을 검출하지 못하는 문제를 발생하였다. 또한 video2의 경우 점진적 장면전환이 이루어지면서 영상 내의 플래쉬에 의하여 영상내에 빛의 변화가 급격하게 나타나므로 오검출이 발생하였다. 제안한 알고리즘은 기존의 히스토그램을 이용한 검출방법보다 물체 움직임 및 빛의 변화에 의한 오검출에 강건한 모습을 나타내고 있고 간단한 연산만으로 장면전환을 검출함으로써 검출 소용시간도 매우 단축됨을 알 수 있다.

## 6. 결론 및 향후 과제

멀티미디어 데이터베이스 구축에 있어서 동영상을 의미있는 장면으로 분할하는 장면 전환 검출이 필요하게 되고, 검출된 대표 프레임을 사용자에게 제공하는 서비스가 필수적이기 때문에 향후 동영상 분할에 관한 연구는 활발히 진행되고 있다. 동영상 데이터들의 크기가 방대하기 때문에 장면전환 검출기법에 있어서 검출속도 및 정확한 검출은 매우 중요한 일

<표 2> 기존 알고리즘과의 장면전환 검출 비교.

Video Data	실제 Cut 개수	기존 알고리즘 (히스토그램)				기존 알고리즘 (매크로블록타입)				제안한 알고리즘			
		컷 검출	미 검출	오 검출	검출 시간	컷 검출	미 검출	오 검출	검출 시간	컷 검출	미 검출	오 검출	검출 시간
Video1	10	21	0	11	15초	10	1	1	7초	10	0	0	7초
Video2	11	8	3	0	16초	13	4	2	8초	12	2	1	8초
Video3	14	18	0	12	28초	16	2	2	10초	12	2	0	10초
Video4	18	18	0	0	59초	19	0	1	21초	18	0	0	21초
Video5	8	10	0	2	19초	9	0	1	9초	8	0	0	9초

이다.

일반적으로 장면전환을 검출 하는데 있어서 중요한 문제는 얼마나 빠르고 정확하게 장면전환 부분을 검출하는가에 있다. 기존 알고리즘의 대부분은 장면전환을 검출하는데 있어서 디코딩 과정을 거치므로 많은 연산량을 필요로 하거나 연속되는 두 프레임간의 특징 값의 변화만을 이용하므로 빛의 변화나 물체의 움직임에 관한 오검출에는 약한 모습을 나타내고 있다. 본 논문에서는 동영상 내용기반 검색을 위하여 가장 기본적인 장면전환 검출에 있어서 복호화 과정시 소요되는 시간을 절약하여 기존의 알고리즘보다 더 빠른 장면전환 검출을 할 수 있도록 하였다. 급진적 장면전환 검출에 있어서 B 프레임의 매크로블록타입의 변화만을 가지고 장면전환 구간을 검출할 수 있는 기법을 제안하였고, 장면전환 구간의 수학적 모델을 제시하였다. 제안한 기법은 장면전환을 검출하기 위한 부가적인 연산이 적기 때문에 구현이 쉽고 빠른 처리가 가능한 장점을 갖는다. 제안된 방법은 일반적인 PC에서도 실시간으로 동작할 수 있어 내용기반 정보검색 시스템에 필요한 장면분할에 효율적으로 활용할 수 있을 것으로 기대된다. 특히 광고와 같이 영상내에 빛의 변화나 물체의 움직임이 빠르게 일어나는 영상에 매우 강한 모습을 나타낸다.

기존의 급진적 장면전환을 검출하는 알고리즘은 연속된 두 프레임 사이의 특징 값을 비교하여 일정 임계값 이상일 경우 장면전환으로 검출하였는데, 제안한 알고리즘에서는 하나의 B 프레임이 이전 프레임과 이후 프레임의 정보를 포함하고 있으므로, 이전 프레임과의 특징 값 비교뿐만 아니라 이후 프레임까지도 특징 값을 비교할 수 있으므로, 프레임내의 물체 움직임이나 빛의 변화에 의한 오검출에 매우 강건한 모습을 나타내었다. 앞으로는 카메라 움직임에 의한 오검출 및 미검출을 줄이고 점진적인 장면전환 구간을 검출하기 위한 연구가 필요하다.

## 참 고 문 헌

- [1] A. Nagasaka and Y. Tanaka, "Automatic Video Indexing and Full Motion Search for Object Appearance," Proc. Of IFIP on Visual Database System, pp.113-127, Sep., 1991.
- [2] Fernando, W.A.C., Canagarajah, C.N., and Bull, D.R. "Fade-in and fade-out detection in video sequences using histograms," The 2000 IEEE International Symposium on Circuits and Systems., Proceedings. ISCAS 2000 Geneva, Vol.4, pp.709-712, 2000.
- [3] Xinying Wang, and Zhengke Weng, "Scene abrupt change detection," Electrical and Computer Engineering, 2000 Canadian Conference on, Vol.2, pp.880-883, 2000.
- [4] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, "Automatic Partitioning of Full-Motion Video," Multimedia Systems, Vol.1, pp.10-28, 1993.
- [5] A. Hampapur, R. Jain, and T. Weymouth, "Production Model Based Digital Video Sequentionation," Multimedia Tools and Applications, Vol.1, No.1, pp.9-46, Mar., 1995.
- [6] B.Shahary, "Scene Change Detection and Content-Based Sampling of Video Sequentionation. Multimedia Tools and Applications," Vol.1, No.1, pp.9-46, Mar., 1995.
- [7] B.L Yeo and B. Lie, "Rapid scene analysis on compressed video," IEEE Trans. On Circuits and Systems for Video Technology, Vol.5, No.6. pp.533-544, Dec., 1995.
- [8] H. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," Multimedia Tools Applicat., Vol.1, pp.89-111, 1995.
- [9] Jianhao Meng, Y. Juan, S. F. Chang, "Scene Chang Detection in a MPEG Compressed Video Sequences," Proc. of SPIE, Vol.2419, pp.14-25, 1995.
- [10] V. Kobla, D. S. Doermann, and K. I. Lin, "Archiving, indexing, and retrieval of video in the compressed domain," Proc. SPIE: Multimedia Storage and Archiving Systems, Vol.2916, pp.78-89, 1996.
- [11] Shen, Bo, Li, Donge, Sethi, Ishwar K., "HDH Based Compressed Video Cut Detection," IEEE '97 Second Int. Conf. on Visual Information Systems, pp.149-156, 1997.
- [12] V. Kobla, and D. Doermann, "Extraction of features for indexing MPEG compressed video," IEEE First Workshop on Multimedia Signal Processing, pp.337-342, 1997.
- [13] Wan, Xia and Kuo, C.-C. Jay, "A New Approach to Image Retrieval with Hierarchical Color Clustering," IEEE Trans. on Circuits and Systems for Video Technology, Vol.8, No.5, pp.628-643, 1998.
- [14] Soo-Chang Pei and Yu-Zuonng Chou, "Efficient MPEG Compressed Video Analysis Using Macroblock Type Information," IEEE Trans. On Multimedia, Vol.1, No.4, pp.473-492, December, 1999.
- [15] S. W. Lee, Y. M. Kim and S. W. Choi "Fast Scene change Detection using Direct Feature Extraction from MPEG Compressed Videos," IEEE Trans. Multeimedia, Vol.2, No.4, pp.240-254, 2000.
- [16] Lelescu D., and Schonfeld D, "Real-time scene change detection on compressed multimedia bitstream based on statistical sequential analysis," IEEE International Conference on Multimedia and Expo, 2000. ICME 2000, Vol.2, pp.1141-1144, 2000.
- [17] W.A.C. Frenando, C.N. Canagarajah and D.R. Bull. "Scene change detection algorithms for content-based video indexing and retrieval," Electronics & Communication Engineering Journal, pp.117-126. June, 2001.
- [18] Tianming Liu, Hong-Jiang Zhang, and Feihu Qi "A Novel video Key-Frame-Extraction Algorithm Based on Perceived Motion Energy Model," IEEE Trans. On Circuits and Systems for Video Tech., Vol.13, No.10, pp.1006-1013. Oct., 2003.



**김 중 헌**

e-mail : joongheon@mail.pcu.ac.kr  
2000년 배재대학교 정보통신공학과(학사)  
2002년 배재대학교 정보통신공학과  
(공학석사)  
2005년~현재 배재대학교 정보통신공학과  
박사과정

관심분야: 동영상처리, 멀티미디어 내용기반 검색, 비디오 편집



**장 종 환**

e-mail : jangjw@mail.pcu.ac.kr  
1979년 한양대학교 전자통신공학과  
(공학사)  
1986년 North Carolina 주립대학교 전기  
및 컴퓨터공학과(공학석사)  
1990년 North Carolina 주립대학교 전기  
및 컴퓨터공학과(공학박사)

1990년~현재 배재대학교 정보통신공학과 교수

1998년~현재 정보통신부 지정 정보통신창업지원센터 소장

2003년~현재 배재대학교 정보통신대학원 원장

관심분야: 영상처리, 멀티미디어 검색, 비디오편집, 컴퓨터 비전