

논문 2011-48SP-1-16

# 매크로 블록 정보와 시공간 히스토그램을 이용한 빠른 장면전환 검출

(Fast Scene Change Detection Using Macro Block Information and  
Spatio-temporal Histogram)

진 주 경\*, 조 주 희\*, 정 재 협\*, 정 동 석\*\*

(Ju-Kyong Jin, Ju-Hee Cho, Jae-Hyup Jeong, and Dong-Suk Jeong)

## 요 약

기존의 많은 장면 전환 검출 알고리즘은 점진적 장면 전환을 검출하기보다는 급격한 장면 전환 검출에 중점이 맞추어졌다. 일반적으로 점진적 장면 전환 검출에 중점을 둔 알고리즘은 많은 연산량을 필요로 한다. 또한 장면 전환 검출에 오류 요소인 플래쉬 라이트, 카메라 움직임 및 특수효과 등의 다양한 오류 요소를 고려하지 못하는 경우가 많다. 또한 기존의 많은 방법들은 히스토그램 기반의 알고리즘을 제시하였지만 좋은 성능에 비해 처리속도에서 취약하다. 본 논문에서는 저장된 동영상으로부터 수직과 수평 블록의 시간적 슬라이스 영상과 슬라이스 영상 내 매크로 블록에 해당되는 정보를 이용한 빠르고 정확한 장면 전환 검출 알고리즘을 제안한다. 슬라이스 영상으로부터 시, 공간 상관관계의 히스토그램을 구성하고, 이를 그래프 컷 분할 알고리즘에 적용하였다. 처리속도 향상을 위해 영상 전체가 아닌 각각 영상 내 수직, 수평 방향의 중심 부분의 해당되는 위치의 블록에서만 시공간 정보를 추출하여 히스토그램을 구성하였다. 또한 카메라, 물체의 움직임 및 특수효과 변화 등을 효과적으로 검출할 수 있도록 매크로 블록의 움직임과 형태 정보를 이용하여 상당한 변별력 향상을 보였다.

## Abstract

Most of the previous works on scene change detection algorithm focus on the detection of abrupt rather than gradual changes. In general, gradual scene change detection algorithms require heavy computation. Some of those approaches don't consider the error factors such as flashlights, camera or object movements, and special effects. Many scenes change detection algorithms based on the histogram show better performances than other approaches, but they have computation load problem. In this paper, we proposed a scene change detection algorithm with fast and accurate performance using the vertical and horizontal blocked slice images and their macro block informations. We apply graph cut partitioning algorithm for clustering and partitioning of video sequence using generated spatio-temporal histogram. When making spatio-temporal histogram, we only use the central block on vertical and horizontal direction for performance improvement. To detect camera and object movement as well as various special effects accurately, we utilize the motion vector and type information of the macro block.

**Keywords:** scene change detection, shot boundary detection, video indexing, graph cut partitioning

## I. 서 론

최근 저장된 비디오를 이용한 내용 기반 검색, 색인, 브라우징 및 데이터베이스 구축 등 다양한 멀티미디어 시스템이 등장하고 있다. 이러한 기술들은 장면전환 기술 기반의 알고리즘이 주류를 이룬다. 이는 비디오 데이터를 보다 계층적이고 구조적으로 재구성하는데 장면

\* 학생회원, \*\* 정회원, 인하대학교 전자공학과  
(Dept. of Electronic Engineering, Inha University, Korea)

※ 이 논문은 인하대학교 교내학술연구비의 지원을 받아 수행된 연구임

접수일자: 2010년9월6일, 수정완료일: 2010년10월13일

전환 기술이 매우 효과적이기 때문이다. 비디오 장면전환 검출기술은 크게 압축된 영역에서의 장면전환 검출과 비 압축된 영역에서의 장면전환 검출로 구분할 수 있다.

압축된 영역에서의 장면전환 검출은 부분 복원된 영상의 데이터만을 이용하므로 빠른 검출기술이 가능하다. 기존에 연구된 방법들은 움직임 벡터와 매크로 블록의 DCT(Discrete Cosine Transform)계수를 많이 사용하였다<sup>1~2)</sup>. 그러나 최근 비디오 압축 기술에서는 I 프레임 외에는 DCT 계수를 완전한 디코딩 없이 이용할 수가 없다<sup>3)</sup>. 근래의 비디오 저장 기술에서는 매크로 블록의 형태와 움직임 벡터만을 사용할 수 있다. 이 정보만을 이용한 장면전환 검출이<sup>3~4)</sup> 최근 발표되고 있으나 대부분의 실험이 제한된 실험환경으로 실제 적용이 불가능한 경우가 많다.

비 압축 영역에서의 장면전환 검출은 화소비교나 히스토그램을 사용하는 알고리즘이 주로 이용된다. 이 검출기술은 급진적 장면 전환뿐만 아니라 장시간에 걸쳐서 발생하는 점진적 장면 전환 검출에도 많은 시도가 있었다<sup>5)~8)</sup>. 장면전환 검출기술의 오류 요소인 플래시라이트 및 카메라, 물체의 움직임 등에는 많은 성능 향상을 보였으나, 과도한 연산량을 요구하는 단점이 생겨 이 기술의 사용을 기피하는 결과를 가져왔다.

본 논문에서는 연산 양을 줄이는 효율적인 접근을 위해 디코딩한 영상에서 수직, 수평의 중심 블록 영상만을 이용하여 시공간 히스토그램을 구성하였다. 이는 영상 전체를 사용하지 않기 때문에 따라 시스템의 부하를 줄일 수 있는 효과적인 방법이다. 구해진 시공간 히스토그램에 기반을 두어 단일 영상 내부 분할 및 군집화에서 주로 사용되는 그래프 컷 분할(Graph Cut Partitioning) 알고리즘을 시간적 분할 영역으로 적용하였고, 선택된 시간적 분할 경계영역을 동영상 디코딩에서 얻는 매크로 블록 정보들을 이용하여 최종 판단 과정을 거치는 장면전환 검출 알고리즘을 제안한다.

본 논문은 구성은 다음과 같다. II장 본론에서는 시공간 히스토그램 구성 방법과, 구성된 히스토그램을 이용한 그래프 컷 분할 그리고 최종 움직임 벡터 및 매크로 블록 형태 정보를 이용한 최종 판단 방법에 대하여 설명한다. III장에서 실험을 통해 제안된 알고리즘을 효율성을 검증하며, IV장에서는 결론 및 향후 연구 과제에 대해서 언급한다.

## II. 장면 전환 검출 알고리즘

### 1. 시간적 슬라이스 영상 정보 추출

기존 알고리즘 중 속도개선을 위해 전체 영상에서 정보를 얻어내는 방식에서 영상에 일부만을 취한 정보만을 이용한 장면전환 검출을 시도하였다<sup>6~7)</sup>. 그중 가장 효과적인 방법의 하나로 그림 1. 과 같이 수직, 수평 중심부분을 연속된 영상에서 계속적으로 영상을 취하여

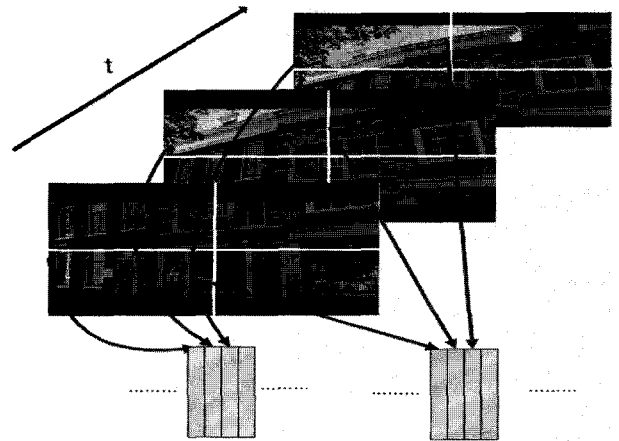


그림 1. 시공간 수직, 수평 슬라이스 영상 패턴 구성  
Fig. 1. Construction of temporal spatial vertical and horizontal slice image pattern.

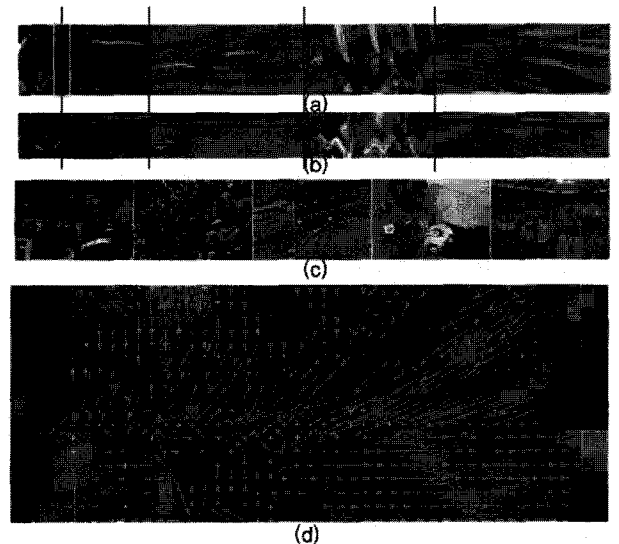


그림 2. 시공간 슬라이스 영상 구성  
(a)수평 (b)수직 (c)슬라이스영상의 각 장면마다의 대표영상들 (d) 첫 번째 장면 변화 부근의 매크로 블록 정보

Fig. 2. An illustration of generating spatio-temporal slice image (a) horizontal (b)vertical (c) representative image formed per each scene (d) information of macro block at the first scene change region.

시간적으로 구성된 영상 슬라이스로 구성하여 분석하는 방법이 있다<sup>[6]</sup>. 본 논문에서도 이와 유사하게 연속된 영상에서 수직, 수평 중심 영역의 정보만을 취하여 구성하였다. 그림 1은 시간적 슬라이스 영상을 구성하는 방법을 나타낸다. 그림 2(a), (b)는 구성된 시간 수평, 수직 슬라이스 영상이다. 그림 2(c)는 각 장면마다의 대표 영상이다. 슬라이스 영상을 관찰하면 각각 장면전환 및 특정 이벤트마다 특징적인 패턴이 나타난다. 이를 이용하여 다양한 방법으로 장면 전환 검출하는 연구가 있다<sup>[4]</sup>. 그러나 카메라 및 물체의 움직임을 와이프(Wipe)나 디졸브(Dissolve) 같은 특수 효과의 장면전환과 구분하기는 수직, 수평 시간적 슬라이스 영상만으로 부족하다. 그림 2(d)는 (a)의 첫 번째 장면 전환 지점의 박스영역에 확대 영상이다. 영상위에 십자가는 인트라 블록을 의미하고 화살표는 움직임 벡터를 의미한다. 장면 전환이 일어나는 경우 경계 부근에서 인트라 블록이 거의 전 라인에서 나타냄을 알 수 있다. 카메라 움직임을 블록의 움직임 벡터로 예측하여 특수 효과 전환과 구분할 수도 있다. 본 논문에서 움직임 벡터와 형태를 이용하여 점진적 장면전환과 카메라 움직임을 구분하는 해결책으로 제시한다.

## 2. 시공간 히스토그램

본 논문에서는 연속된 프레임 사이에 차이의 정도를 얻기 위하여 기존의 알고리즘들과 다르게 시공간 정보를 이용한 히스토그램을 제시한다. 히스토그램은 검출 효과를 높이기 위하여 명암정보와 컬러정보인 Hue를 이용해 구성한다. 기존의 히스토그램 구성은 매 프레임마다 해당 영상 내에서만 히스토그램을 얻어낸다<sup>[6]</sup>. 이는 히스토그램의 문제인 공간적 관계 정보를 사용하지 않기 때문에 영상을 기술하는데 한계가 있다. 예로 바다나 하늘의 풍경 영상의 경우 유사한 배경에서 장면 전환이 일어나더라도 하늘의 색상이 지배적이기 때문에 유사한 히스토그램을 구성함에 변별력이 낮다. 이를 해결하기 위해서 현재 블록의 이웃하는 블록과의 상관관계를 이용하여 히스토그램 구성한다. 이는 중복되는 지배적인 영상 패턴에 대해 히스토그램 구성 시 비중을 줄이는 방법의 하나이다. 또한 블록의 시간적 상관관계까지 확장시켜 해당 프레임에 앞, 뒤 영상까지 포함한 현재 블록의 이웃하는 3차원 공간상에 26개 블록을 이용하여 시공간 히스토그램을 구성한다.

시공간 히스토그램 구성은 방법은 첫 번째로 모든 영

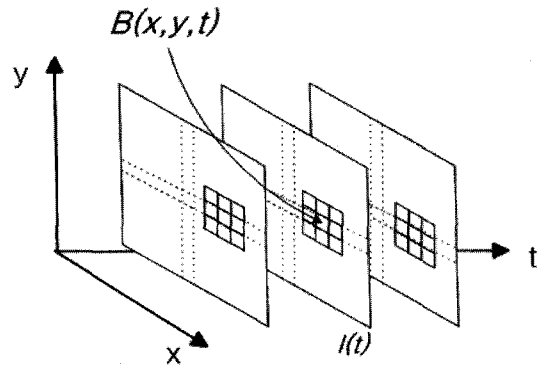


그림 3. 시공간 히스토그램 구성  
Fig. 3. Construction of spatio-temporal histogram.

상을 16x16 화소의 매크로블록마다의 명암과 Hue 평균값을 이용하여 재구성하였다. 그림 3은 시공간 좌표상의 한 블록의 위치와 이웃하는 블록을 보여준다. 식 (1)과 같이 해당 블록과 윈도우내에 이웃 블록의 평균값  $m$ 을 이용하여 식 (2)에서 평균값에 해당되는 히스토그램 위치  $bin$  을 1을 할당한다.  $B(x,y,t)$ 는 해당 블록의 명암과 Hue값을 나타낸다.  $T$ 는  $m$ 값에 해당하는 히스토그램 위치 값을 나타내는 변환 표를 의미한다,  $bin$ 은 히스토그램의 크기 값인 0부터  $N$ 값을 가지며,  $Bin$ 은 바이너리 값인 0, 1 가진다.  $N$ 은 히스토그램의 좌표의 크기이며, 본 논문에서는  $N$ 은 32로 사용하였다.

$$m = \frac{1}{2}(B(x,y,t) + B(x+a,y+b,t+c)) \quad (1)$$

$$-1 \leq a,b,c \leq 1$$

$$Bin(bin) = 1, bin = T(m) \quad (2)$$

한 블록과 시공간의 주변 블록에 대한 모든 연산을 통해  $Bin$ 을 1과 0로 작성한다. 구성된  $Bin$ 을 이용하

표 1. 시공간 히스토그램  $H_i$ 를 구성하는 유사코드  
Table 1. Pseudocode for constructing spatio-temporal histogram  $H_i$

```

WHILE  $x, y, t \in I(t)$  DO
  FOR  $a, b, c = -1$  to 1 DO
     $m = (B(x,y,t) + B(x+a,y+b,t+c)) / 2$ 
     $Bin(T(m)) = 1$ 
  ENDFOR
  FOR  $n=1$  to  $N$  DO
     $H_i(n)++$ , if  $Bin(n) = 1$ 
  ENDFOR
ENDWHILE
    
```

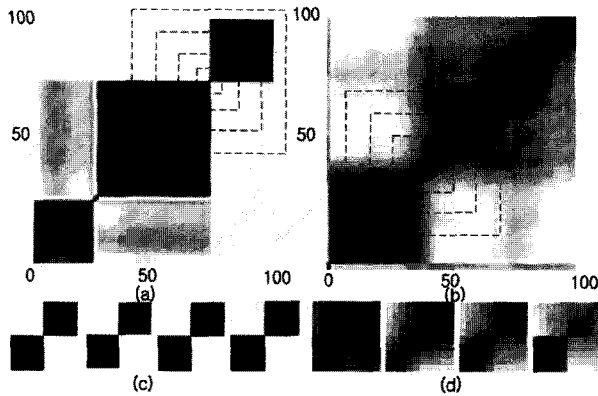


그림 4. 100 프레임끼리 모든 쌍의 차이 값을 영상화 (a) 급진적 장면 분할 (b) 점진적 장면 변화 (c),(d) 점선의 사각형별 추출 후 10x10으로 샘플링 영상( $\sigma = 1, 2, 3, 5, 7$ )

Fig. 4. Imagination of dissimilarity of 100 frame pairs. (a) abrupt scene change (b) gradual scene change (c),(d) 10x10 sampling image from each red box( $\sigma = 1, 2, 3, 5, 7$ )

여 1로 설정된 히스토그램 좌표  $H(n)$ 에만 1을 추가한다. 즉 현재 블록과 주변 블록에 상관관계를 살펴 중복을 제거하여 히스토그램을 구성한다. 표 1은 시공간 히스토그램을 구하는 유사코드(Pseudocode)를 나타낸다.  $I(t)$ 는 현재  $t$ 프레임의 수직, 수평 중심에 해당되는 영역을 나타낸다.

모든 프레임 영상에서 히스토그램을 구한 후 이를 이용하여 프레임간의 차이 값은 식 (3)로 계산한다.  $i, j$ 는 프레임 번호,  $H_i$ 는 전체의 합이 1로 정규화된 히스토그램이며, 매 프레임마다 명암, 색상 값 2개의 히스토그램을 가진다.

$$w(i, j) = \sum_n^N |H_i(n) - H_j(n)| \quad (3)$$

### 3. 그래프 컷 분할을 이용한 검출

그래프 컷 분할(Graph Cut Partitioning)은 영상 처리에서 영상 분할, 군집(Image Segmentation and Clustering)분야에서 많이 사용되는 방법 중 하나이다. 이를 동영상으로 적용한 방법 또한 동영상 분할에서 각 광범는 접근법이다<sup>[5]</sup>. 간단하면서도 급진 장면 변화 및 점진적 장면 전환 모두에서 효과적이다. 즉, 영상 내에서 군집화를 시간적 차원에서 수직, 수평 슬라이스 영상에서의 군집화의 문제로 변환하는 것이다.

그림 4는 100 프레임내 모든 프레임 쌍끼리 히스토그

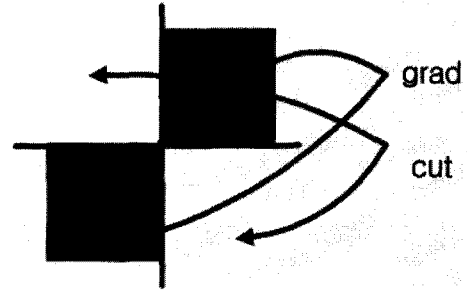


그림 5. 차이값 샘플링 영상의 영역  
Fig. 5. Region of frame dissimilarity sampling image.

램 차이 값을 영상화 하였다. 영상의 한 화소마다 해당 프레임간의 시공간 히스토그램 차이 값을 0에서 255로 표시하였다. 어두운 부분이 차이 값이 적은 부분이고, 밝은 부분은 차이 값이 큰 부분이다. 그림 4(a)는 급진적 전환이 25, 70 프레임 번호 두 곳에서 발생하였다. 그림에서 보인 듯 명확하게 두 개의 검은 정사각형의 모서리가 이웃하는 두 곳을 확인할 수 있다. 그림 4(b)는 40 프레임 번호에서 점진적 전환이 발생하였다. 또한 카메라 움직임 및 빠른 물체의 움직임 또한 이와 유사하게 나타난다. 모든 프레임에서 프레임을 중심으로 앞, 뒤로  $\sigma$ 간격으로 샘플링 하여 그림 4(c),(d)와 같은 영상으로 재구성한다. 이는 점진적 장면 전환도 넓은 영역에서 재구성해보면 급진적 장면 전환과 유사한 형태의 영상으로 취급할 수 있음을 보인다. 그림 5와 같이 각각의 샘플링한 영상마다 영상을 균등하게 4등분하여 좌상과 우하는 전환구역(*cut*), 좌하와 우상은 점진적 전환(*grad*)구역으로 나누어 각 구역의 값의 비율을 식(4)와 같이 구한다.

$$M(i, \sigma) = \frac{cut(i, \sigma)}{grad(i, \sigma)} \quad (4)$$

$i$ 는 프레임 번호이다. 매 프레임마다 각 샘플링  $\sigma$ 별

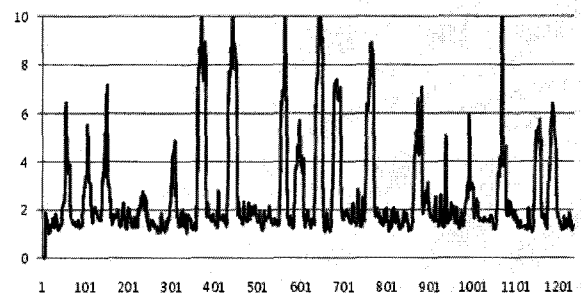


그림 6.  $M(i, \sigma)$  값들의 분포  
Fig. 6. Distribution of  $M(i, \sigma)$

로 값을 가진다.  $cut(i, \sigma)$ 는 전환구역의 화소 값의 합,  $grad(i, \sigma)$ 는 점진적 전환 구역의 화소 값의 합을 나타낸다. 각각의  $\sigma$ 마다 연속된 값들에서 국소 최댓값(Local Maxima)을 갖는 프레임을 장면전환 후보로 지정한다. 그림 6은 매 프레임마다  $M(i, \sigma)$ 값 중 최댓값을 보여준다.

4. 움직임 벡터를 이용한 검증

저장된 동영상에 디코딩 과정에서 자동으로 얻어지는 움직임 벡터와 매크로 블록 형태를 이용하여 장면전환 여부를 최종 판단한다. 영상에서 I, B 및 P 프레임 중 B 프레임은 제외하고 바로 이전과 현재 프레임의 변화 정도를 움직임 벡터와 매크로 블록 형태를 가지고 두 프레임의 차이 값을 추출해 낸다. I 프레임의 경우 모두 인트라 매크로 블록을 가지는 관계로 그림 7과 같이 바로 전, 후의 P 프레임에서 같은 위치에 매크로 블록으로부터 움직임 벡터를 예측해 낸다. 식(5)은 움직임 벡터 예측 식을 나타낸다.

$$m_I(x, y) = (\alpha m_{pre-p}(x, y) + \beta m_{nxt-p}(x, y)) \quad (5)$$

$m_I$ 는 I 프레임에서  $x, y$ 블록 위치에서 움직임 벡터,  $pre-p, nxt-p$ 는 가장 가까운 전, 후에 P프레임을 나타낸다.  $\alpha, \beta$ 는 현재 프레임과 전, 후 프레임의 거리 가중치를 나타낸다.  $\alpha, \beta$ 의 합은 1이다. 현재 프레임  $i$ 에서 전 프레임과의 차이 값  $dist(i)$ 은 식(7)과 같이 구한다. 블록의 차이를 나타내는 상태  $D_{blk}$ 는 매크로 블록의 조건에 따라 1, 0으로 구성되고 1이 되는 경우는 다음 표 2와 같다.

$M, N$ 은 각각 수직, 수평의 블록 수. I 프레임에 경우는 복원된 움직임 벡터( $x', y'$ )를 이용하여 식(6)과 같이 전 P 프레임에서 추적된 위치에 블록의 합산 값과 비교하여  $thres_{blk}$ 보다 높으면  $D_{blk}$ 를 1로, 낮으면 0으로 설정한다. P 프레임에서 제로 움직임 벡터에도 식(6)

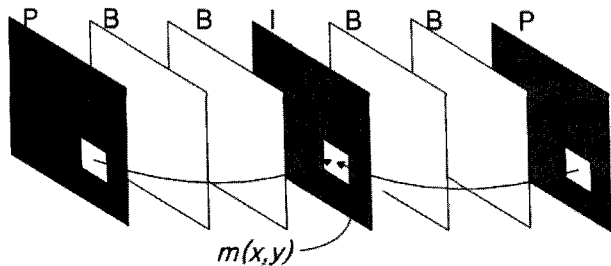


그림 7. I 프레임에서 움직임 벡터 예측  
Fig. 7. Prediction of motion vector at I-frame.

표 2. 매크로 블록별  $D_{blk}$ 값 조건

Table 2. Condition of  $D_{blk}$  per each macro block.

매크로블록 타입	조건	$D_{blk}$
Direct, Skip	None	0
$mv = (0, 0)$	$D_t < thres_{blk}$	0
	others	1
Intra in P	Interlaced이면 매크로블록 타입 $mv = (0, 0)$ 가정하여 해당조건 확인, Progressive 이면 1	1 or 0
Intra in I	$(x', y')$ 이용하여 $mv$ 복원 후 $D_t < thres_{blk}$	1 or 0

을 이용하여 판단한다. 이는 매크로 블록이 전 프레임과 상당한 변화가 발생하지만 인트라 블록대신 제로 움직임 벡터와 큰 차분 값을 두는 경우를 판단하기 위함이다. 본 논문 실험에서  $thres_{blk}$ 는 7로 하였다.

$$D_t = \sum_{x', y'}^{16} I(x', y', t-1) - \sum_{x, y}^{16} I(x, y, t) \quad (6)$$

$$x' = x + mv_x, y' = y + mv_y$$

$$dist(i) = \left\{ \frac{\sum_x^N D_{blk}(x, M/2) + \sum_y^M D_{blk}(N/2, y)}{M+N} \right\} \quad (7)$$

그림. 8은 H264, MPEG2 및 XVID 주로 사용되는 3가지 비디오 저장 포맷별 식(7)의  $dist(i)$  값을 비교하는 그래프이다. 식(7)의 값이 큰 경우에는 거의 유사하나 작은 값에서 차이는 다소 보이나 전반적으로 포맷별 값의 차이는 크지 않다고 볼 수 있다. 불필요한 연산량을 줄이기 위해서 모든 프레임에서  $dist(i)$ 을 구하는

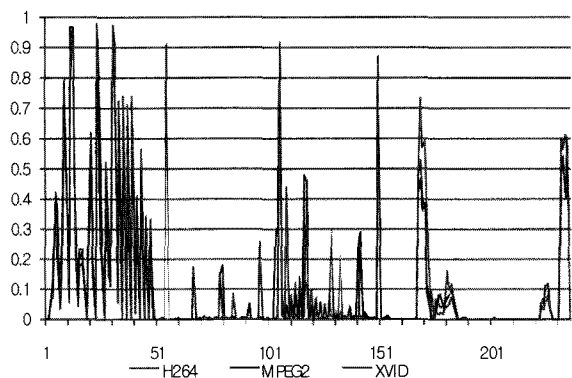


그림 8. 비디오 저장 포맷별  $dist(i)$  값의 분포  
Fig. 8. Distribution of  $dist(i)$  from various video storage format.

것이 아니라 2. 3절에서 제시한  $M(i, \sigma)$ 의 국부 최댓값 주변 영역에서만 연산이 이루어지기 때문에  $dist(i)$ 를 구하는 프레임은 극히 일부분이다.

5. 제안된 장면 전환 알고리즘 과정

제안된 알고리즘은 다음 Step을 거친다.

Step1. 디코딩된 영상에서 각각 수직, 수평 중심에서 영상정보와 매크로블록 관련정보(움직임 벡터, 형태)를 얻어온다.

Step2. 시공간 슬라이스 이용하여 시공간 히스토그램을 구한다.

Step3. 프레임별 히스토그램을 이용하여 이웃 프레임과 차이값 영상을 제작하여  $M(i, \sigma)$ 를 구한다.

Step4. 연속된  $M(i, \sigma)$ 에서 국소 최댓값 위치를 찾는다.

Step5. 찾은 최댓값 위치에서 해당  $\sigma$ 에 해당되는 구간에  $dist(i)$ 값을 구한다.

Step 6.  $dist(i)$ 값이 0.40 이상이면 장면전환 검출이라고 간주한다.  $dist(i)$ 의 임계치 0.40는 실험값이다.

III. 실험 및 결과

본 논문에서 실험을 위해 실제 방송되는 뉴스, 다큐멘터리, 스포츠, 영화 등에 다양한 장르에 동영상을 이용하였다. 또한 공정한 실험을 위해 TRECVID2001 실험 영상을 포함하였다. 비디오 형식은 H.264 및 MPEG2를 사용하였다. 실험 평가는 일반적으로 가장 많이 사용되는 Recall, Precision 및 F-Measure 방법을 사용하였다. 정의는 식(8)과 같다. 모든 실험 데이터를 통해 Recall, Precision 값을 구하고, 그 결과를 이용하여 F-Measure를 구한다. 최대 F-Measure 값을 가질 때에 설정한 실험 변수 값을 본 논문의 실험 치로 이용하였다.

$$\begin{aligned}
 Recall &= \frac{N_c}{N_c + N_m} \times 100 \\
 Precision &= \frac{N_c}{N_c + N_f} \times 100 \\
 F-Measure &= \frac{2 \times Recall \times Precision}{Recall + Precision}
 \end{aligned}
 \tag{8}$$

$N_c$ 는 검출된 장면전환 수,  $N_m$ 은 검출되지 못한 장면전환 수,  $N_f$ 는 잘못 검출된 장면전환 수를 나타낸다.

표 3은 실험에서 사용된 비디오의 제원을 나타내며, 표 4는 실험에서 얻어진 비디오 성능평가를 보여준다.

표 3. 실험 영상의 조건

Table 3. Condition of test video set.

Clip	Dur.(sec)	Size (WxH)	Cut	Grad.	contents
1	300	640 x 480	60	13	다큐멘터리 영화
2	290	720 x 384	49	18	애니메이션 영화
3	387	624 x 352	126	2	드라마
4	360	448 x 252	44	2	다큐멘터리
5	360	448 x 280	76	22	자동차경주
6	402	704 x 396	51	18	TV 쇼
7	354	544 x 480	53	11	피겨스케이팅
8	360	328 x 240	80	5	뉴스
9	420	704 x 396	101	22	TV광고
10	480	640 x 352	77	31	TV 쇼
11	330	640 x 352	85	11	TV 쇼
12	247	448 x 336	17	29	dissolve 홈 비디오
13	574	720 x 480	173	24	스포츠 하이 라이트

표 4. 제안된 알고리즘 실험 결과

Table 4. Result of proposed algorithm.

Clip	$N_c$	$N_m$	$N_o$	Recall	Precision	F-Mea.
1	55	18	7	75.34	88.71	81.48
2	46	21	4	68.66	92	78.63
3	111	17	9	86.72	92.5	89.52
4	45	1	6	97.83	88.24	92.78
5	71	27	10	72.45	87.65	79.33
6	50	19	7	72.46	87.72	79.37
7	41	23	12	64.06	77.36	70.09
8	70	15	9	82.35	88.61	85.37
9	91	32	11	73.98	89.22	80.889
10	84	24	6	77.78	93.33	84.85
11	77	19	4	80.21	95.06	87.01
12	18	28	0	39.13	100	56.25
13	155	42	19	78.68	89.08	83.56

비교 대상 알고리즘으로는 TSS(Temporal spatial slice)<sup>[6]</sup>, Adaptive<sup>[8]</sup>, TSI(TSIghua)<sup>[5]</sup>들을 선택하였다. TSS는 시공간 슬라이스를 이용한 방법이고, Adaptive의 경우는 이웃하는 영상간의 히스토그램 기반 접근 방법으로 장면전환 임계값을 실험치를 따로 사용하지 않고 적응 기법을 이용한 알고리즘이다. 마지막으로 TSI 경우 그래프 컷을 장면전환 검출에 적용하였으나 최종 판단을 학습을 통해서 얻는 알고리즘이다.

그림 9는 비교대상 과 제안된 알고리즘의 성능 평가 그림이다. 그림 9(a)는 표 3에서 나타내는 실제 방송되

는 영상으로 구성된 데이터에서 F-measure를 이용한 성능 결과이고 그림 9(b)는 Trecvid2001 동영상에서 선택된 7가지 영상에 대한 성능 결과이다. 이렇게 영상을 2가지로 구분한 이유는 Trecvid2001 영상들이 많은 연구에서 실험 영상으로 사용되고 있으나 오래된 영상이기 때문에 오늘날의 다양하고 복잡한 장면 전환 형태에 비해 비교적 전환 형태가 단순하며 화질 면에서도 열화가 심한 편이기 때문이다. 그림 9(a)가 좀 더 현실성 있

는 영상의 실험 결과로 볼 수 있다. 그림 9에서 제시 하듯 모든 실험 데이터에서 제안된 알고리즘이 우수하게 나옴을 알 수 있다. 비교 대상 TSI 알고리즘의 경우 학습과정이 들어감에 따라 제안한 실험 데이터 외에 4배 이상의 데이터를 이용하여 학습하여 얻은 결과이다. 성능 실험에 사용한 데이터에서만 학습할 경우 높은 성능을 발휘하지만 다른 데이터가 들어감에 따라 성능이 떨어짐을 알 수 있다.

그림 10에서는 처리 속도 실험 결과를 보인다. 보이는 것과 같이 제안된 알고리즘이 우수한 처리 속도를 보인다. 처리 시간을 측정할 때 디코딩 시간을 제외한 알고리즘 연산 시간만 측정하였다. 특히 중요한 점은, 다른 알고리즘과 다르게 영상의 사이즈에 크게 영향 받지 않는다는 것이다.

#### IV. 결 론

본 논문에서는 기존 알고리즘의 단점을 보완하기 위해 프레임의 수직, 수평 중심의 시공간 및 매크로 블록의 정보를 통합하여 장면전환 검출하는 알고리즘을 제안하였다. 제안된 알고리즘은 기존의 장면 알고리즘과 비교하여 성능 및 처리 속도에서 효과적으로 개선되었다. 이는 장면전환이 이용되는 많은 분야에서 활용 가능할 정도의 속도개선이 괄목할 만한 속도 개선을 이룬다. 그러나 그림 9(a)에서 12번 비디오 데이터 경우와 같이 디졸브 영상에서 성능이 낮음을 알 수 있다. 이는 디졸브는 매크로 블록의 변화가 영상 전반에서 급격하게 변화하지 않고 순차적으로 장시간에 변화하는 과정으로 인해 움직임 벡터를 이용한 최종 판단에서 미 검출되는 경우가 일어난다. 그러나 별도의 디졸브 알고리즘을 추가로 제안한 TSS알고리즘의 경우보다는 높음을 알 수 있다. TSS알고리즘의 경우 Fade Out/In의 경우에서만 국한한 결과이다. 실제 디졸브는 움직이는 두 영상의 점진적 교차가 다수를 이룬다. 반면 학습 알고리즘인 TSI의 경우 디졸브 영상에서 좋은 성능을 보인다. 그러나 학습의 특성상 어떤 원인에서 잘 검출되는지는 쉽게 파악할 수 없다.

본 논문에서는 오직 수직, 수평만의 정보를 사용하였는데 이는 전체 영상을 이용함에서 오는 처리속도 저하 때문이다. 반대로 더 많은 영상 영역에서 정보를 이용할 경우 검출 성능의 향상을 가져올 수도 있다. 하지만 향상이 괄목할 정도의 향상은 아니다. 또한 움직임 벡

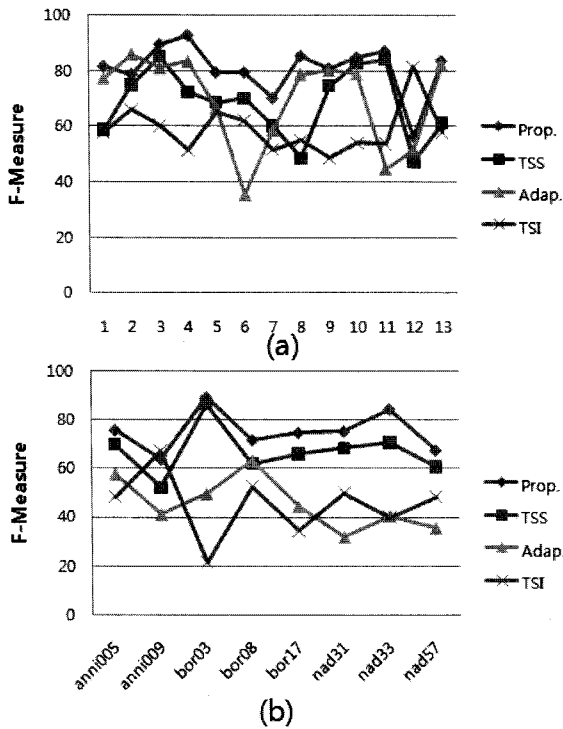


그림 9. 성능 실험 결과 (a) 제안된 실제 방송 실험 데이터 (b) TrecVid2001

Fig. 9. Experimental result on performance. (a) proposed real broadcasting test data (b) Trecvid2001.

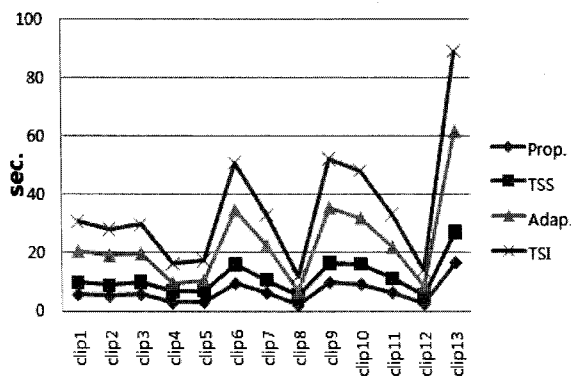


그림 10. 처리 속도 실험 결과

Fig. 10. Experimental result on processing time.

터와 매크로블록 정보는 엔코더에 특성에 민감한 것도 무시할 수 없다. 추후 연구에서는 이러한 것들을 개선하기 위한 많은 실험과 노력이 필요하다.

### 참고 문헌

- [1] J. Besos, "Real-time shot change detection over online MPEG-2 video", *IEEE Trans. Circuits Systems Video Technol.* 14 (4) (2004) 475-484.
- [2] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?", *IEEE Transaction on circuits and Systems for Video Technology*, vol. 12, pp. 90-105, Feb, 2002.
- [3] S. De Bruyne et al., "A compressed-domain approach for shot boundary detection on H.264/AVC bit stream", *Signal Processing: Image Communication*, vol. 23, pp.473-489, Aug. 2008.
- [4] Kim, S.M., Byun, J., Won, C.: "A Scene Change Detection in H.264/AVC Compression Domain" *Proceedings of PCM '05.*(2005) 1072-1082
- [5] Yuan J, Li J, Lin F, Zhang B "A unified shot boundary detection framework based on graph partition model." *In Proceedings of the 13th annual ACM international conference on multimedia*, pp 539-542
- [6] C. W. Ngo, T. C. Pong, and R. T. "Video partitioning by temporal slice coherency." *IEEE Trans. on Circuits and Systems for Video Technology*, 11(8), Aug 2001.
- [7] C. W. Ngo, T. C. Pong, and H. J. Zhang. "Motion analysis and segmentation through spatio-temporal slices processing." *IEEE Trans. on Image. Processing*, vol 12. issue 3. pp.341-355 March. 2003.
- [8] Dugad R, Ratakonda K, Ahuja N. "Robust video shot change detection", *IEEE workshop on Multimedia Signal Processing*, Redondo Beach, CA, December 1998. p.376-81.

### 저자 소개



진 주 경(학생회원)  
2003년 인하대학교 전자공학과  
학사 졸업.  
2005년 인하대학교 전자공학과  
석사 졸업  
2005년~현재 인하대학교  
전자공학과 박사과정

<주관심분야 : 영상처리, 멀티미디어 신호처리,  
패턴인식, 내용 기반 검색>



조 주 회(학생회원)  
2009년 인하대학교 전자공학과  
학사 졸업.  
2009년~현재 인하대학교  
전자공학과 석사 과정

<주관심분야 : 영상처리, 컴퓨터 비전, 패턴인식,  
내용 기반 검색>



정 재 협(학생회원)  
2009년 인하대학교 전자공학과  
학사 졸업.  
2009년~현재 인하대학교  
전자공학과 석사 과정  
<주관심분야 : 영상처리, 컴퓨터  
비전, 멀티미디어 신호처리, 내용  
기반 검색>



정 동 석(정회원)  
1977년 서울대학교 전기공학과  
학사 졸업  
1985년 Virginia 주립 공과대학  
전자공학과 공학 석사  
1988년 Virginia 주립 공과대학  
전자공학과 공학 박사  
1988년~현재 인하대학교 전자공학과 교수  
<주관심분야 : 영상처리, 컴퓨터 비전, 패턴인식,  
멀티미디어 정보처리>