

뉴스 비디오 데이터에서의 가중치를 이용한 효율적 장면변환 검출 알고리즘

정 영 은[†] · 이 동 섭^{††} · 신 성 윤^{†††} · 전 근 환^{††††} · 배 석 찬^{†††††} · 이 양 원^{††††††}

요 약

뉴스 비디오 데이터베이스 시스템을 구축하기 위해서는 장면변환 검출 기법이 매우 중요하다. 일반적인 분할기법으로 컬러히스토그램, χ^2 히스토그램, 그리고 Bin_to_Bin Difference(B2B)기법이 주로 이용되고 있다. 본 논문에서는 NTSC 표준에 따른 가중치를 적용하여 보다 효율적인 장면변환 검출 알고리즘을 제안한다. 제안된 방법은 컬러 히스토그램에서 RGB의 값들을 각각 따로 계산함으로써 히스토그램 구성 및 비교에 필요한 계산시간을 단축하며, χ^2 히스토그램에서의 차이 값을 강조하여 임계치 선택을 용이하게 하는 장점을 제공한다. 이 알고리즘의 일반적인 성능 평가를 위하여 본 논문에서는 KBS, MBC, SBS, CNN 및 NHK의 뉴스 비디오들을 실험 도메인으로 하여 이용한다. 실험 결과에서, 제안한 방법은 기존 방법들 보다 장면변환 검출에 더 효율적임을 알 수 있고, 임계값들을 자동적으로 할 수 있는 기반을 제시한다.

The Efficient Cut Detection Algorithm Using the Weight in News Video Data

Yung-Eun Jung[†] · Dong-Seop Lee^{††} · Sung-Yoon Shin^{†††} ·
Keun-Hwan Jeon^{††††} · Seok-Chan Bae^{†††††} · Yang-Weon Lee^{††††††}

ABSTRACT

In order to construct the News Video Database System, cut detection technique is very important. In general, the color histogram, χ^2 histogram or Bin-to-Bin Difference(B2B) techniques are mainly using for the scene partitioning. In this paper, we propose the efficient algorithm that is applied the weight in terms of NTSC standard to cut detection. This algorithm is able to reduce the time of acquiring and comparing histogram using by separate calculation of R, G and B for the color histogram technique. And it also provide the efficient selection method of threshold value by emphasis of the difference in the χ^2 histogram technique. For the performance evaluation of this algorithm, we select and use the news videos of KBS, MBC, SBS, CNN and NHK as experimental domains. By the result of experiment, we present the proposed algorithm is more efficient for cut detection than the previous methods, and that the basis for the automatic selection of threshold values.

† 정 회 원 : 한국전자통신연구원 선임연구원
†† 준 회 원 : 군산대학교 대학원 컴퓨터학과
††† 정 회 원 : 군산대학교 기초과학연구소 연구원
†††† 정 회 원 : 군장대학 경영전산학부 교수
††††† 정 회 원 : 군산대학교 컴퓨터과학과 교수
†††††† 종신회원 : 군산대학교 컴퓨터과학과 교수
논문접수 : 1998년 7월 13일, 심사완료 : 1998년 11월 19일

1. 서론

최근 멀티미디어 자료의 활용이 적극적으로 요구되면서 다양한 자료를 저장, 관리, 검색하는 데이터베이스 기술이 요구되고 있다. 특히 동영상자료인 비디오는 인간에게 많은 정보를 매우 자연스럽게 제공해 줌으로써 많은 관심의 대상이 되고 있으며, 특정한 문제 영역에 맞도록 저장하고 이용하는 내용 기반 멀티미디어 데이터베이스의 구축에 대한 연구가 활발히 진행 중이다.

이러한 비디오 자료 중에서도 뉴스에 관한 비디오 자료는 사회 전반의 중요한 정보를 함축적으로 짧은 시간대에 표현한 것으로, 그 나라의 정치, 경제, 문화, 사회 등의 전반적인 상황을 분석할 수 있으며, 향후의 상황 예측이 가능한 귀중한 자료로 내용 기반에 의한 비디오 검색의 주요 연구 대상이 되어 왔다.

본 논문에서는 뉴스 비디오 데이터베이스에서 검색과 결과의 대상이 되는 장면을 분류하기 위한 여러 기법들을 살펴보고, 이러한 기법들보다 나은 새로운 알고리즘을 제안하고, 주어진 일반적인 실험환경에서 파생되는 결과를 비교 분석하여 제안한 방법을 평가한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 현재까지 진행되어 온 관련 연구와 본 논문이 궁극적으로 접근하고자하는 시스템의 전반적인 구성과 일반적인 용어에 대해 설명한다. 제 3장에서는 본 논문에서 제안한 방법을 설명하고, 제 4장에서는 현재 국내외의 대표적인 몇몇 뉴스를 실험 도메인으로 하여, 기존의 방법들과 제안한 방법을 적용한 후, 실험 결과에 대해서 비교 분석한다. 제 5장에서는 본 논문의 결론 및 향후 연구 방향을 제시한다.

2. 기존 연구

2.1 용어 정의

비디오를 대상으로 하는 연구에서 사용되는 용어는 서로 다른 의미를 가지고 있는 경우가 많다. 본 연구에서는 일반적으로 많이 사용되는 용어를 기본적으로 이용하였으며 이들에 관한 일반적인 정의는 다음과 같다.

- 프레임(frame)

프레임은 비디오 자료를 구성하는 최소단위이다. 필름 한 장에 해당하는 하나의 정지영상으로, 이 프레임들을 빠르게 진행시키면 연속된 동작에 의해 움직이는 동영상이 된다.

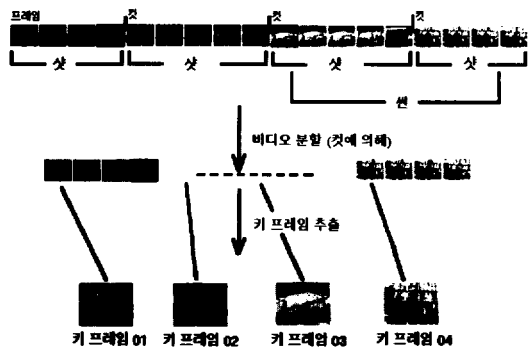
- 샷(shot)

비디오를 분할 할 때에 기본 단위가 되는 일련의 프레임들의 모임이다. 보통 하나의 카메라로 촬영되며, 필름이 끊기지 않고 연결된 프레임으로 구성된다.

- 장면(scene)

장면은 하나의 주 대상을 촬영한 연속된 샷(shot)들의 집합이다. 즉, 필름의 내용이 연결되지 않는다 하더라도 촬영되는 대상이 변하지 않거나 그 의미가 변하지 않는 범위 내에서의 샷들을 의미한다.

(그림 1)은 비디오 데이터를 구성하는 요소들의 관계를 그림으로 표현한 것이다.



(그림 1) 비디오 자료의 계층적 구성
(Fig. 1) Hierarchical structure of Video Data

일반적으로 장면 변화는 크게 두 가지로 나눌 수 있다. 갑작스러운 장면 변환과 점진적인 장면의 변환인데, 갑작스러운 장면 변환에 해당하는 것은 추출이 쉽지만, 점진적인 장면의 변환에 해당하는 것은 찾기가 어렵다. 본 논문에서 실험에 이용되는 뉴스 영역은 갑작스런 장면 변환에 해당된다. 뉴스 영역은 일반적으로 다음과 같은 물리적인 구조 계층으로 분류되며, 이들은 아래의 용어로 설명된다.

- 컷(cut)

장면과 장면 사이의 경계에 해당하는 장면 전환점에 해당한다.

● **컷 프레임(cut frame)**

장면에서 장면으로 전환되는 최초의 프레임을 컷 프레임이라고 한다.

● **키 프레임(key frame)**

샷(shot)의 프레임들을 대표하는 하나의 프레임을 키 프레임이라고 한다. 본 논문의 연구에서는 컷 프레임을 키 프레임으로도 정의하여 이용한다.

2.2 기존의 장면 분할 기법

장면을 분할하기 위한 방법에는 여러 가지가 있다. 프레임간의 차이를 이용하는 방법과 칼라 히스토그램을 이용하는 방법, 객체의 이동을 계산하여 사용하는 방법, 에지를 추출하여 이용하는 방법과 전체적인 웨이블릿(wavelet)을 측정하여 이용하는 방법 등 많은 방법이 사용되었다. 이러한 방법 중 뉴스 도메인처럼 갑작스런 장면의 검출에는 주로 히스토그램 차이를 계산하여 사용하는 방법(식 1)이 보편화 되어있으며 아래와 같다[1][2][3][4].

$$d(I_i, I_{i+1}) = \sum_{j=0}^{N-1} |H_i(j) - H_{i+1}(j)| \quad (\text{식 1})$$

$d(I_i, I_{i+1})$ 는 이웃 프레임과의 유사성을 측정하는 함수이며, $H_i(j)$ 는 i 번째 프레임의 j 번째 히스토그램 값이라는 의미이다. 주 대상이 되는 사건의 장면에서는 각 프레임들이 유사한 특징을 가지므로 에지 히스토그램의 모양이 거의 비슷하다. 그러므로 히스토그램의 차이 값이 크게 나타나는 곳은 한 사건 장면에서 또 다른 사건장면으로 넘어가는 것이라고 할 수 있다. 사건장면에서는 일반적으로 장면의 변화가 많아 이웃한 프레임과의 차이가 크게 나타나는 경향이 있는데, 이럴 경우 장면변화가 많은 연속된 프레임들 중에서 첫 프레임만을 컷 프레임으로 선택한다[1][2][4].

히스토그램 차이 값을 이용하는 방법은 자료의 성격에 따라 다음과 같은 방법들이 사용된다.

● **템플릿 매칭(Template Matching)**

이 방법은 동일한 위치의 두 영상의 픽셀을 비교하는 방법으로 방법이 매우 간단하고 수행 시간이 적게 드는 장점이 있으나 객체의 이동이나 zoom에 매우 민감하여 불만족스러운 결과를 산출할 수도 있다[4].

$$d(I_i, I_j) = \sum_{x=0}^{x^c} \sum_{y=0}^{y^c} |I_i(x, y) - I_j(x, y)| \quad (\text{식 2})$$

● **컬러 히스토그램**

컬러 히스토그램은 일반적으로 (식 1)번 방법이 많이 쓰이지만 (식 3)번과 같이 RGB 컬러 공간대로 따로 계산하여 사용할 수도 있다.

$$d_{RGB}(I_i, I_j) = \sum_k (|H_i^r(k) - H_j^r(k)| + |H_i^g(k) - H_j^g(k)| + |H_i^b(k) - H_j^b(k)|) \quad (\text{식 3})$$

● χ^2 히스토그램

이 방법은 많은 연구에서 사용되며, 컬러 히스토그램이나 템플릿 매칭보다도 좋은 결과를 나타내는 방법이다[4].

$$d(I_i, I_j) = \sum_{k=1}^K \frac{(H_{i(k)} - H_{j(k)})^2}{H_{j(k)}} \quad (\text{식 4})$$

$d(I_i, I_j)$ 는 i 번째 Image와 j 번째 Image의 difference를 의미하고 $H_i(k)$ 는 i 번째 Image의 히스토그램에서 k 번째 bin의 값을 의미한다. 즉, 이전 프레임의 히스토그램과 다음 프레임의 히스토그램의 동일한 bin을 뺀 후 그 값을 제곱하여 차이값을 강조한다. 그리고 제곱한 값을 다음 프레임 값으로 나누어 정형화한다.

위의 방법에서 변형된 방법이 (식 5)이다.

$$d(I_i, I_j) = \sum_{k=1}^K \sqrt{H_{i(k)}^2 - H_{j(k)}^2} \quad (\text{식 5})$$

● **Bin to bin difference (B2B)**

$$f_{d_{bb}}(h_1, h_2) = \frac{1}{N^2} \sum_i |h_1[i] - h_2[i]| \quad (\text{식 6})$$

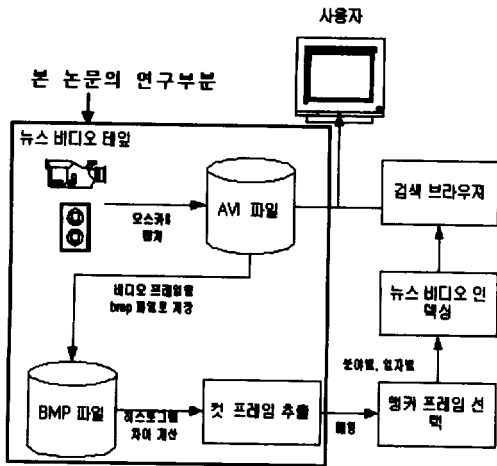
(식 6)에서는 히스토그램의 각 요소의 차이값을 모두 더한 후 bin의 제곱으로 나누어서 계산한다.

3. 본 논문에서 제안한 알고리즘

본 논문에서 제안하는 알고리즘은 (그림 2)와 같은

시스템 구조로 뉴스 비디오 데이터베이스를 구축하기 위한 것이다.

입력되는 뉴스 비디오는 우선적으로 각 장면이 분할되어야 한다. 각 장면의 분할은 컷 프레임 추출하는 단계와, 키 프레임에서 사전지식을 이용하여 앵커 프레임을 추출하는 두 단계로 구성된다. 제안한 알고리즘을 이용하여 컷 프레임을 추출한 후에 앵커 프레임을 선택하고, 앵커 프레임과 컷 프레임을 이용하여 전체적인 뉴스 비디오 자료를 색인화 한다. 그리고 사용자가 원하는 뉴스는 즉각적으로 검색이 가능하게끔 검색 브라우저를 구성한다.



(그림 2) 시스템 구조
(Fig. 2) System Architecture

본 논문에서는 (그림 2)에서 제시된 큰 사각형안을 연구 도메인으로 하여 컷 프레임 추출에 가장 효율적인 알고리즘의 제안을 목적으로 한다.

본 논문에서는 기존의 히스토그램 차이값을 계산하는 알고리즘 중에서 컬러 히스토그램과 χ^2 히스토그램을 이용하여 프레임간의 차이값을 계산하는 식을 제안한다[5]. 두 알고리즘의 장점을 복합하여 변형한 (식 7)은 컬러 히스토그램의 각각 RGB의 값들을 독립적으로 각각 계산함으로써 히스토그램 구성 및 비교에 필요한 계산시간을 단축하며, χ^2 컬러 히스토그램 방법의 차이 값을 강조하는 장점을 이용하여 프레임간의 차이값 계산으로 분할을 용이하게 할 수 있게 하였다.

$$d(I_i, I_j) = \sum_{k=1}^n \left(-\frac{(H_i^r(k) - H_j^r(k))^2}{H_i^r(k)} \times 0.299 + \frac{(H_i^g(k) - H_j^g(k))^2}{H_i^g(k)} \times 0.587 + \frac{(H_i^b(k) - H_j^b(k))^2}{H_i^b(k)} \times 0.114 \right) / 3 \quad (\text{식 7})$$

(식 7)에서 곱해진 가중치 0.299, 0.587, 0.114는 영상을 RGB 컬러 모델에서 YIQ 모델의 Y요소로 바꿀 때 사용하는 값으로서 NTSC 표준에서 비롯된 것이며, 컬러 모델을 명암도(gray) 등급으로 변환하기 위한 것으로 RGB 컬러 공간은 서로 가산될 수 있는 삼원색으로 구성되어 컬러분광 정도에 따라 색이 결정되는 것이다. 특히 초록은 명암도 등급 중에서 상당히 큰 요소로, 간혹 초록 요소만을 명암도 단계 데이터로 사용하기도 한다.

수식을 자세히 설명하면 다음과 같다. $d(I_i, I_j)$ 는 i 번째 Image와 j 번째 Image의 difference를 의미하고, $H_i^r(k)$ 는 i 번째 Image의 R에 대한 히스토그램의 k 번째 bin의 값을 의미한다. 즉, 이전 프레임의 R에 대한 히스토그램과 다음 프레임의 R에 대한 히스토그램의 동일한 bin을 뺀 후 그 값을 제공하여 차이값을 강조한다. 그리고 제공한 값을 다음 프레임의 값으로 나누어 정형화한다. 이런 식으로 G와 B에 대한 값을 구한 후 weighted sum 방식으로 값을 통일한다.

컷 프레임의 선택은 얻어진 히스토그램 차이 값이 임계값보다 클 경우이다. 또한, 임계값을 선택하는데 있어서는 주변의 히스토그램 차이 값을 참조하는 동적 임계값을 이용하였다.

4. 실험

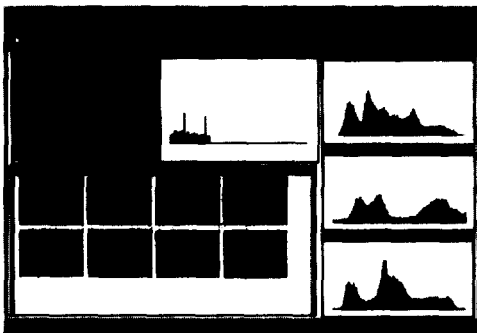
본 논문의 실험은 Windows '98 환경에서 visual C++5.0을 이용하여 구현하였으며, 실험에 사용한 비디오 자료들은 오스카II를 이용하여 AVI 압축 형태로 저장한 국내 KBS, MBC, SBS와 국외의 CNN, NHK의 뉴스 비디오를 <표 1>과 같이 실험 비디오를 택하였

으며, 1998년 5월 29에서 31일 사이의 뉴스를 기본으로 하였다.

〈표 1〉 실험을 위한 뉴스 비디오 자료
 〈Table 1〉 News Video Data for experiment

A	KBS	300초	5	400×300
B	KBS	278초	5	400×300
C	MBC	300초	5	400×300
D	MBC	210초	5	400×300
E	SBS	280초	5	400×300
F	SBS	260초	5	400×300
G	CNN	300초	5	400×300
H	NHK	200초	5	400×300

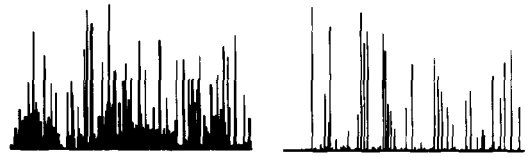
(그림 3)은 해당 비디오 프레임에서 히스토그램 차이 값에 따라 컷 프레임이 추출되게 하는 프로그램의 운영을 보여준다. 오른쪽에는 해당 프레임의 RGB 값에 따라 누적된 히스토그램을 보여준다. 상단의 중앙에는 획득된 비디오의 자료에서 얻어진 한 프레임과 이웃한 프레임과의 차이 값을 나타낸다. 하단에는 주어진 임계값을 적용하여 추출한 컷 프레임을 제시하였다.



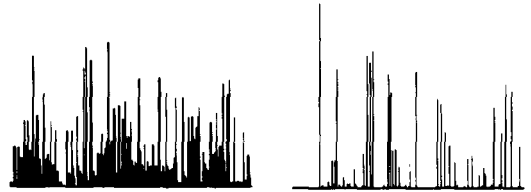
(그림 3) 컷 프레임 추출의 전체화면
 (Fig. 3) Full screen of extracting Cut Frame

4.1 히스토그램 산출

(그림 4)에서 (그림 7)까지는 획득된 여러 방송사의 비디오 자료 중에서 KBS 방송사의 300초 짜리 자료 A에 일반적인 컷 프레임 추출 식과 제안 식에 의해서 얻어진 히스토그램 값의 실례이다.

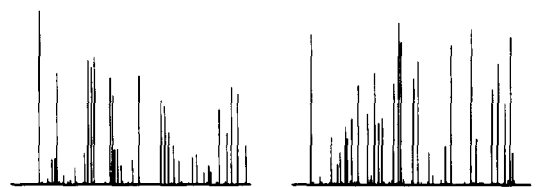


(그림 4) (식 3) 적용결과 (그림 5) (식 4) 적용결과
 (Fig. 4) Result of applied (Eq. 3) (Fig. 5) Result of applied (Eq. 4)

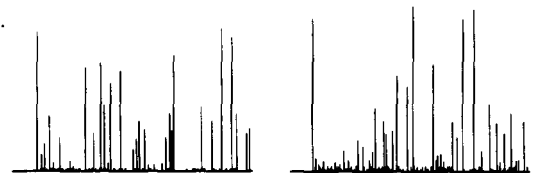


(그림 6) (식 6) 적용결과 (그림 7) (식 7) 적용결과
 (Fig. 6) Result of applied (Eq. 6) (Fig. 7) Result of applied (Eq. 7)

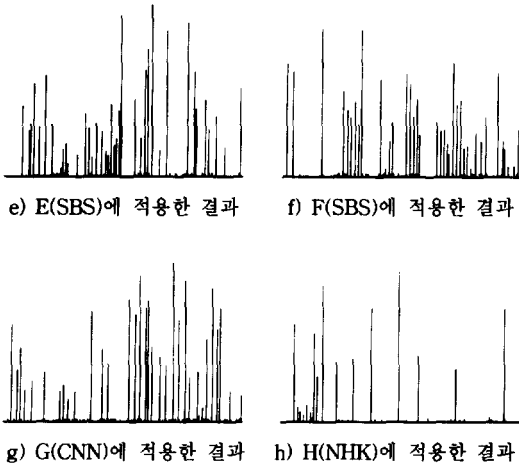
(식 3)과 (식 6)의 결과로 나타나는 히스토그램에서, 임계값의 설정에 따라 추출되는 컷 프레임의 결과는 매우 민감하게 반응한다. 적정 임계값의 설정이 어려운 것은 물론 임계값의 작은 변화에도 그 결과 값은 크게 달라진다. 그러나 (식 4)와 (식 7)을 이용하면 장면 변환이 대부분 히스토그램 차이가 큰 곳에 위치하여 임계값 선택을 쉽게 할 수 있는 장점을 가진다. 즉, 히스토그램 차이 값이 어느 정도 크지 않으면 히스토그램 상에 나타나지 않고, 값의 차이가 크게 나타나는 장면들이 대부분 히스토그램에 나타나므로 임계값을 자동 설정할 수 있는 기반을 제공한다.



a) A(KBS)에 적용한 결과 b) B(KBS)에 적용한 결과



c) C(MBC)에 적용한 결과 d) D(MBC)에 적용한 결과

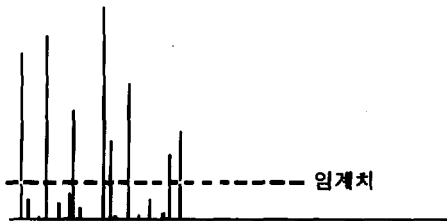


(그림 8) 실험 자료에 (식 7)을 적용한 결과
(Fig. 8) Results of applied (Eq. 7)

(그림 8)은 각각의 실험 자료에 대하여 본 논문에서 제안한 식을 적용하여 얻은 히스토그램의 예를 보여준 것이다.

4.2 컷 프레임 추출

(그림 9)는 얻어진 히스토그램 차이 값에 임의의 임계값 기준을 적용하는 그림이다. 임계값을 결정하는 방법은 사용자가 임의의 값을 전체 비디오에 적용하는 방법과 프로그램 자체적으로 일정한 간격의 범위 내에 있는 차이 값들을 비디오의 특성에 따라, 서로 감안하여 임계값을 적용하는 방법의 두 부류가 대표적이다. 전자의 경우는 정적 임계값 설정이라고 부르며, 후자의 경우는 동적 임계값의 설정이라 한다. 본 논문에서는 기존의 방법 중에서 동적 임계값 설정 방법을 사용하여 각 비디오마다 다른 임계값을 결정하여 사용하였다.

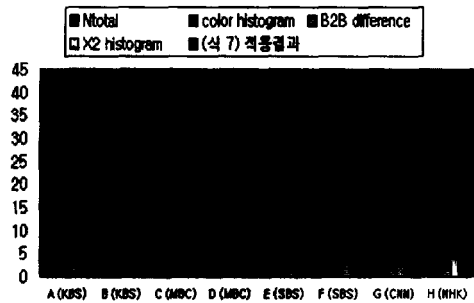


(그림 9) 히스토그램 차이 값에 임계값 적용
(Fig. 9) Applying the threshold to difference of histogram

(그림 10)은 각각의 입력 자료에 따른 실험 결과를

비교한 것으로, 제시된 실험 결과는 모든 컷 프레임을 찾을 수 있는 기준의 임계값을 적용했을 때에 발생하는 오류를 상호 비교한 것이다.

N_{total} 은 실험 자료에서 실질적으로 나타난 컷 프레임의 전체 수를 의미하고, 나머지는 각 방법에서 추출되지 않아야 할 오류 정보가 추출된 경우의 수를 의미한다. 각 방법에 의해 추출된 컷 프레임의 총수는 N_{total} 에 오류 정보가 합쳐진 개수이다.



(그림 10) 자료 A~H의 실험결과 비교
(Fig. 10) The comparison of the experimental results of A~H

(그림 10)은 <표 2>의 자료에 의한 것이다.

실험결과 χ^2 히스토그램과 본 논문에서 제안한 방식의 경우에는 장면의 변화가 발생하는 프레임이 출현하는 경우에 히스토그램 차이 값이 높게 나타났으며 반면에 변화가 심하지 않는 경우에는 낮게 나타나 임계값 선택을 쉽게 하였다.

컬러 히스토그램과 B2B를 적용한 경우에는 앵커 프레임의 차이 값이 다른 컷 프레임의 차이 값과 거의 비슷했으며 컷 프레임의 추출 성공률이 높은 대신 컷 프레임이 아닌 프레임을 컷 프레임으로 추출하는 오류가 많았다. 또한, 앞서 지적한 것처럼 임계값의 기준을 찾기가 여러 번의 시행착오로 수행되었다.

실험에서 나타나듯이 χ^2 히스토그램과 제안한 방법은 같은 결과 또는 보다 나은 결과를 보여주고 있다. 즉, B, E와 H에서는 같은 결과가 산출되었지만, A, C, D, F와 G 자료의 실험에서는 제안한 방법이 더욱 우수한 결과를 보여주고 있다.

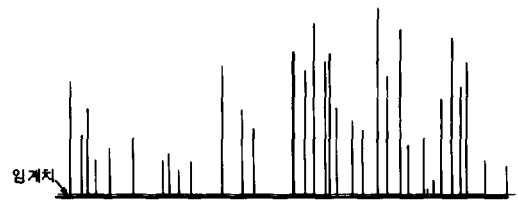
모든 컷 프레임을 찾기 위해서 임계값에 대한 기준을 설정했을 때, 사용되는 계산식에 의해 추출되지 않아야 할 오류정보의 수가 결정되어지며, 오류정보는 추후에 색인을 구성하기 위해 제거되어야 하는 문제를

파생한다. 그러므로 오류정보가 적게 추출되는 것이 시스템의 효율을 높이는 것이 된다.

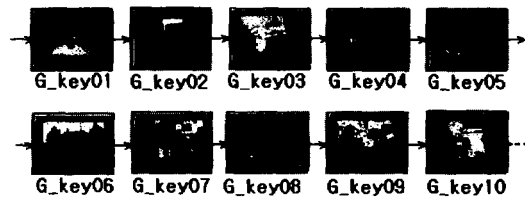
〈표 2〉 자료 A~H의 실험결과
(Table 2) The Experimental result of A~H

A (KBS)	Color Histogram	38	14
	χ^2 Histogram	38	13
	Bin-to-Bin Difference	38	14
	(식 7) 적용 결과	38	10
B (KBS)	Color Histogram	33	20
	χ^2 Histogram	33	13
	Bin-to-Bin Difference	33	18
	(식 7) 적용 결과	33	13
C (MBC)	Color Histogram	34	16
	χ^2 Histogram	34	13
	Bin-to-Bin Difference	34	17
	(식 7) 적용 결과	34	12
D (MBC)	Color Histogram	39	19
	χ^2 Histogram	39	15
	Bin-to-Bin Difference	39	21
	(식 7) 적용 결과	39	13
E (SBS)	Color Histogram	40	21
	χ^2 Histogram	40	13
	Bin-to-Bin Difference	40	24
	(식 7) 적용 결과	40	13
F (SBS)	Color Histogram	40	14
	χ^2 Histogram	40	8
	Bin-to-Bin Difference	40	13
	(식 7) 적용 결과	40	6
G (CNN)	Color Histogram	36	19
	χ^2 Histogram	36	11
	Bin-to-Bin Difference	36	18
	(식 7) 적용 결과	36	6
H (NHK)	Color Histogram	18	6
	χ^2 Histogram	18	4
	Bin-to-Bin Difference	18	5
	(식 7) 적용 결과	18	4

χ^2 와 제안 방법은 (그림 11-a)에 제시한 것과 같이 한번에 임계값을 설정할 수 있는 장점을 보여준다. 히스토그램 차이 값의 변화가 여타의 값과는 현격한 차이를 제공하기 때문이다. 이것은 임계값의 자동 설정이 가능함을 예시하여 (그림 11-b)에서 보여준 것과 같은 추출된 컷 프레임들의 시간적인 연계를 자동적으로 구성할 수 있다. (그림 11-b)는 자료 G의 실험결과로 추출된 컷 프레임의 시간적인 연계를 보여준다.



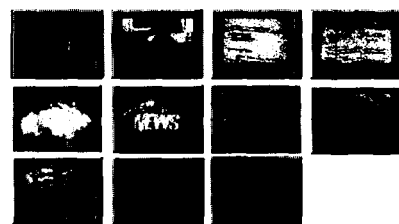
(a) 히스토그램에 임계치 적용



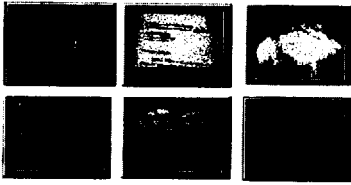
(b) 임계값에 의해 추출된 컷 프레임 예

(그림 11) G로부터 추출된 컷 프레임의 예
(Fig. 11) Example of extracted Cut Frame from G

CNN 뉴스에서 (그림 11-a)와 같이 임계값을 선택했을 때, 추출되는 컷 프레임들 중에서 추출되지 않아야 할 프레임들(N_i)이 χ^2 히스토그램은 11개, 제안한 방법에서는 6개가 포함되어 있다. 이들은 χ^2 와 제안 방법에서 컷 프레임이라 인식된 것으로, 오류정보라 할 수 있으며 (그림 12)와 같다.



(a) χ^2 히스토그램에서의 오류 프레임



(b) 제안방법에서의 오류 프레임

(그림 12) G로부터 추출된 오류 프레임의 예
(Fig. 12) Example of extracted Error Frame from G

(그림 12-a)는 x^2 히스토그램에서, (그림 12-b)는 제안한 방법에서 추출된 컷 프레임 중 잘못 추출된 프레임들의 실례를 나타낸다.

(그림 12-a)에서 오류 프레임으로 추출되는 원인은 여러 가지가 있을 수 있다. 먼저 화면의 전환방식이 복잡한 과정을 거쳐 다음 장면으로 넘어가는 경우, 갑작스런 조명의 밝기 변화, 뜻하지 않은 객체의 출현 등이다. 이러한 원인에 의해 추출되지 않아야 할 정보가 추출되며, 오류 프레임을 제거하기 위한 많은 방법이 연구되고 있다.

x^2 히스토그램에서 추출된 오류 프레임((그림 12-a))과 제안방법에서 추출된 오류 프레임((그림 12-b))을 보면, x^2 히스토그램의 5개 오류 프레임이 제안한 방법의 결과에서는 제거되었다는 것을 알 수 있다. 또한, x^2 히스토그램 방법에서 발생되지 않는 프레임은 제안방법에서도 발생될 수 없다는 것을 나타내기도 한다. 이것은 기존 방법보다 적게 오류가 포함된다는 것을 의미하며, 기존 방법보다 정밀해진 것을 의미한다.

기존 방법의 계산시간과 본 논문에서 제안한 방법의 계산시간은 다음과 같다.

〈표 3〉 계산 시간의 비교
(Table 3) The comparison of the calculating time

Color Histogram	0.4 sec/frame
x^2 Histogram	1.2 sec/frame
Bin-to-Bin Difference	1.1 sec/frame
(식 7)의 방법	0.6 sec/frame

sec/frame은 한 영상을 처리하는데 걸리는 시간을 의미한다. 계산 시간의 측정은 작성된 프로그램 내에 타이머(timer)를 작동시켜 프레임 당 계산시간을 측정하였다.

5. 결론 및 향후 연구방향

본 논문은 뉴스 비디오 데이터베이스를 구축하기 위한 장면 분할 기법 중 컬러 히스토그램에서 각각의 RGB의 값들을 각각 따로 계산함으로써 히스토그램 구성 및 비교에 필요한 계산시간을 단축하는 장점과, x^2 히스토그램에서 차이 값을 강조하는 장점을 이용하여 NTSC 표준에 따른 가중치를 적용한 새로운 장면 분할 방법을 제안하였다.

본 논문에서 제안한 방법은 비디오 자료의 자동 색인화 작업에 요구되는 컷 프레임의 정확한 추출 면에서 기존 방법보다 일보 진전된 것으로 실험에서 입증되었다. 또한, 동적으로 임계값의 선택이 프로그램으로 가능한 것이 기존 방법에 비해 임계값의 설정에서 보다 높은 자동화 할 가능성을 제시하였다. 기존의 방법보다도 효율적이지만, 아직까지도 포함되는 오류 프레임을 정확히 제거하기 위한 방법이 계속적으로 연구되어야 할 것이다.

참고 문헌

- [1] E. Ardizzone, M. L. Cascia, "Automatic Video Database Indexing and Retrieval," *Multimedia Tools and Applications*, Vol.4, No.1, pp.29-56, January. 1997.
- [2] J. C. LEE, Q. LI, W. XIONG, "VIMS : A Video Information Management System," *Multimedia Tools and Applications*, Vol.4, No.1, pp.7-28, January. 1997.
- [3] B. Furht, S. W. Smoliar, H. J. Zhang, "Video and Image Processing in Multimedia System," Kluwer academic Publishers, pp.335-356, 1995.
- [4] S.J. Dennis, R. Kasturi, U.Gargi, S.Antani, "An Evaluation of Color Histogram Based Methods in Video Indexing," *Research Progress Report CSE-96-053 for the contract MDA 904-95-C*

2263, 1995.

- [5] Y. Gong et. al. "An image database system with content capturing and fast image indexing abilities," Proceedings of the International Conference on Multimedia Computing and Systems, Boston, MA, pp.121-130, May. 1994.
- [6] S. Devadiga, D. Kosiba, U. Gargi, S. swald, P.Kasturi, "A semiautomatic video database system," SPIE conference on Storage & Retrieval for Image & Video Databases, Vol.2420, pp.262-267, 1995.
- [7] U. Gargi, S. Oswald, D. Kosiba, S. Devadiga, and R. Kasturi "Evaluation of video sequence indexing and hierarchical video indexing," SPIE conference on Storage & Retrieval for Image & Video Databases, Vol.2420, pp.144-151, 1995.
- [8] A. Akutsu(ed.), "Video indexing using motion vector," Visual Communications and Image Processing '92, Boston, MA, pp.1522-1530, November. 1992.
- [9] W.Wolf, "Multimedia Information on the Internet," '97 International Conference Multimedia database on internet, pp.3-23, 1997.
- [10] 이동섭, 김재원, 배석찬, 이양원, "비디오 장면변환 검출 알고리즘", 한국 퍼지 및 지능시스템 학회 '98 춘계 학술대회 발표 논문집 제8권 1호, 98년 6월.

정 영 은

e-mail : yejung@etri.re.kr
 1987년 경희대학교 물리학과(이학사)
 1987년~현재 한국전자통신연구원
 선임연구원
 관심분야 : 멀티미디어, 영상처리,
 데이터베이스

이 동 섭

e-mail : leeds@ai.kunsan.ac.kr
 1997년 군산대학교 컴퓨터학과
 (이학사)
 1997년~현재 군산대학교 컴퓨터
 과학과 석사과정
 관심분야 : 영상 데이터베이스, 패
 턴인식, 멀티미디어 데이터베이스

신 성 윤

e-mail : syshin@cs.kunsan.ac.kr
 1994년 군산대학교 컴퓨터학과
 (이학사)
 1997년 군산대학교 정보통신공학
 과(공학석사)
 1998년~현재 군산대학교 기초과
 학연구소 연구원, 군산대
 학교 컴퓨터과학과 시간강
 사, 군장대학 경영전산학
 부 겸임교수
 관심분야 : 데이터베이스, 영상처리, 패턴인식, 멀티미디
 어 데이터베이스

전 근 환

e-mail : khjeon@kunjang.ac.kr
 1993년 군산대학교 전자계산학과
 (이학사)
 1995년 충북대학교 전자계산학과
 (이학석사)
 1997년 충북대학교 전자계산학과
 박사과정 수료
 1997년~현재 군장대학 경영전산학부 전임강사
 관심분야 : 시간지원 데이터베이스, 멀티미디어 데이
 터베이스, 전문가 시스템 셸



배 석 찬

e-mail : scbae@cs.kunsan.ac.kr

- 1983년 전남대학교 전자계산학과 (이학사)
- 1988년 전남대학교 전자계산학과 (이학석사)
- 1995년 전남대학교 전자계산학과 (이학박사)

1993년~95년 서남대학교 전산통계학과 학과장

1995년~현재 군산대학교 컴퓨터과학과 조교수

관심분야 : 분산처리, 객체지향데이터베이스, 데이터베이스



이 양 원

e-mail : ywrhee@ai.kunsan.ac.kr

- 1979년 숭실대학교 전자계산학과 (공학사)
- 1983년 연세대학교 산업대학원 전산전공(공학석사)
- 1994년 숭실대학교 전자계산학과 (공학박사)

1979년~1986년 국방 연구원

1986년~현재 군산대학교 컴퓨터과학과 교수

관심분야 : 영상 데이터베이스, 패턴인식, 인공지능, 멀티미디어 데이터베이스