

동영상데이터의 프레임간 색상차의 자승합을 이용한 컷 검출시스템

The Cut Detection System using Sum of Square Difference of Color between Frames of Video Image Data

김 병 철*
Byung-Chul Kim

정 창 렬**
Chang-Ryul Jung

고 진 광***
Jin-Gwang Koh

요 약

컴퓨터의 발전과 정보통신기술의 발달은 멀티미디어 기술 확산과 대용량의 멀티미디어 동영상데이터 사용용 증가시켰다. 동영상데이터는 전체적인 데이터를 파악하고, 원하는 동영상을 바로 재생 가능할 뿐만 아니라 동영상 데이터의 정보가 요약된 리스트제공이 필요하다. 그리고 효과적인 동영상 검색을 위해서는 동영상 데이터의 색인과정이 필수적이며 꼭 필요한 기술이다.

따라서 본 연구는 동영상 데이터 내용기반 색인에 기초가 될 프레임의 컷 검출의 효과적인 방법을 제안한다. 제안된 방법은 프레임의 대각선 방향으로 나누어 일정한 픽셀 색상 정보 값을 추출하였다. 동영상 데이터의 각각 프레임에서 추출된 칼라색상의 픽셀값은 $A(i, j)$ 행렬로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장한다. 저장되어진 픽셀값은 프레임간의 색상차의 자승합을 이용하여 프레임간의 특정 값의 차이를 임계 값보다 클 경우 빠르고 정확하게 컷을 검출하였다. 프레임 컷 검출에 대한 실험을 포괄적으로 하기 위해 여러 종류의 동영상 데이터를 실험 대상으로 하여 컷 검출 시스템의 성능을 비교 분석하였다.

Abstract

The development of computer technology and the advancement of the technology of information and communications spread the technology of multimedia and increased the use of multimedia data with large capacity. Users can grasp the overall video data and they are able to play wanted video back. To grasp the overall video data it is necessary to offer the list of summarized video data information. In order to search video efficiently an index process of video data is essential and it is also indispensable skill. Therefore, this thesis suggested the effective method about the cut detection of frames which will become a basis of an index based on contents of video image data. This suggested method was detected as the unchanging pixel color intelligence value, classified into diagonal direction. Pixel value of color detected in each frame of video data is stored as $A(i, j)$ matrix - i is the number of frames, j is an image height of frame. By using the stored pixel value as the method of sum of squared difference of color two frames I calculated a specified value difference between frames and detected cut quickly and exactly in case it is bigger than threshold value set in advance. To carry out an experiment on the cut detection of frames comprehensively, I experimented on many kinds of video, analyzing and comparing efficiency of the cut detection system.

1. 서 론

* 정 회 원 : 순천대학교 공과대학 정보통신공학부 정교수
kbc@sunchon.ac.kr

** 정 회 원 : 순천대학교 대학원 컴퓨터과학과 박사과정
chari7@mail.knou.ac.kr

*** 종신회원 : 순천대학교 공과대학 정보통신공학부 정교수
kjg@sunchon.ac.kr

이 논문은 2001년도 순천대학교 공과대학 학술재단연구비에 의하여 연구되었음.

컴퓨터를 사용하는 정보통신의 기술의 발전으로 인하여 디지털화된 대용량의 멀티미디어 자료를 대중적으로 이용하게 되었다. 때문에 동영상자료 검색, 전송 그리고 데이터를 처리하는 환경의 수요가 해마다 증가하고 있다. 멀티미디어 자료 중에서 동영상 자료는 데이터의 양이 방대하며, 정보 전달 매체로 통신 네트워크를 이용하는 경우 빠른 시간 내에 정확한 정보검색과 처리 할 수 있는 기술이 매우 중요하다.

기존의 멀티미디어 데이터 검색 방법은 미리 입력된 키워드를 사용한 텍스트를 기반으로 하는 질의 및 접근 방법이 일반적이다. 이러한 접근 방식은 텍스트주석을 각 영상에 포함하고 있기 때문에 텍스트를 검색 질의어로 사용한다. 그러므로 사용자는 영상을 기술하는 키워드 범위를 미리 알고 있어야 하고, 나라마다 다양한 언어와 문화적 특성에 종속되는 문제점을 가지고 있다. 텍스트를 기반으로 하는 영상 검색 시스템의 문제를 극복하기 위해 최근에는 영상의 내용(content)을 기반[14,16]으로 하는 여러 방법들이 연구되고 있다.

이러한 방법은 기존의 텍스트 기반의 단점을 보완하고자 물체의 모양, 질감, 색상과 같은 영상의 특징을 이용하여 검색하는 방법이다[16]. 동영상의 내용기반 검색을 위해서는 동영상 인덱싱이 중요한 기술이며, 사용자가 전체 동영상 정보를 한눈에 파악하고, 필요한 동영상을 원하는 지점부터 직접 재생할 수 있도록 하는 동영상의 대표 프레임 리스트가 제공되어야 한다. 대표프레임 리스트를 작성하기 위해서는 정확한 프레임 컷 검출 필요하다. 컷 검출 알고리즘으로는 Nagasaka가 제안한 칼라 쌍(color-pair)의 방법은 영상내의 경계선 사이에서 변환되는 칼라 성분의 차로 히스토그램을 형성하는 방식[18]이 있다. 그리고 Hampapur[2] 및 Shahraray[5] 등이 제안한 모델기반 장면 변환기법과 Grsky가 제안한 검색 대상이 되는 객체의 윤곽선을 벡터화하여 화상 데이터의 특징 값으로 이용하는 방법 [3]이 있다. Chua[18]는 Nagasaka가 제안한 칼라 쌍에 유사도를 첨가하는 방법을 제안하였다.

이들 연구방법들은 전체 동영상에 대한 검색 속도를 향상시키기 위하여 공간적, 시간적 표본화 기법등의 연구였다. 그러나 Nagasaka가 제안한 방법은 히스토그램의 임계값이 영상의 값의 분포에 따라 변경되는 문제점을, Hampapur가 제안한 방법은 영상데이터가 압축되지 않을 때만 제안한 방법이 적용가능 하므로 MPEG로 압축된 동영상 데이터는 먼저 디코딩을 해야 함으로 프레임 컷 검출을 위한 데이터를 처리하는데 시간이 오래

걸리고, 메모리 오버헤드가 발생하는 문제점등을 가지고 있다[18]. 최근에는 이 문제를 해결하기 위해 내용을 직접 분석하여 컷 검출에 필요한 의미 있는 특징 정보를 추출하여 동영상의 컷 검출에 활용하려는 연구가 많이 이루어지고 있다.

본 연구에서는 동영상 데이터 내용기반 색인에 기초가 될 프레임의 컷 검출을 위한 효과적인 방법을 제안한다. 제안한 방법은 프레임을 대각선 방향으로 나누어 일정한 픽셀 색상 정보 값을 추출하여 정지된 영상데이터로 변환하여 변환된 동영상데이터의 전체적인 구조를 파악하기 위해 동영상데이터의 컷을 한 눈에 파악 할 수 되도록 표본추출 하였다. 급격한 장면이 전환되는 동영상 데이터와 점진적으로 장면이 전환되는 동영상데이터의 컷을 검출한다. 이 컷 검출 알고리즘은 픽셀값을 추출하여 칼라색상의 픽셀값을 $A(j, i)$ 행렬로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장함으로써 저장되어진 픽셀값의 프레임간 색상차의 지승합(DDS)을 이용하여 전환프레임간의 특정 값의 차이를 구해 미리 정해진 임계값보다 클 경우 컷을 검출한다. 검출된 컷은 실험을 포괄적으로 하기 위해 여러 종류의 동영상 데이터를 실험 대상으로 하여 성능을 비교 분석한다.

본 연구의 구성은 다음과 같다. 먼저 2장에서는 동영상데이터의 컷 검출 기법을 소개하고 제 3장에서는 본 연구에서 제안하는 동영상데이터의 컷 검출시스템을 기술한다. 제 4 장에서 제안하는 기법을 실제의 동영상데이터를 이용하여 실험하여 결과를 기술하며 마지막으로 5장에서 결론과 향후 연구에 대해 기술한다.

2. 동영상데이터의 컷 검출 기법

2.1 동영상의 구조

동영상의 일반적인 데이터의 구조를 살펴보면, 동영상의 데이터는 다단계의 계층적 구조의 형태로 표현할 수 있다. 동영상 데이터의 상위 단계는

스토리(story) 계층으로 전체적인 이야기의 줄거리이다. 스토리 계층의 바로 아래 단계는 장면(Scene)이라는 동영상의 이벤트(event) 단위로 나눌 수 있다. 동영상을 장면 단위로 나누는 것이 가장 이상적인 방법이지만, 장면이라는 것은 의미그룹으로서 자동으로 장면변화가 되었다는 것을 검출하는 것이 매우 어려운 작업이다. 따라서 물리적 단위인 컷을 검출하는 작업이 수행된다.

일반적으로 동영상 데이터의 구조는 다단계 계층적 구조로 표현하여 프레임(Frame)은 동영상 데이터를 구성하는 최소 단위로 각각의 픽셀로 이루어진 필름 한 장에 해당하는 하나의 정지영상이다. 샷(Shot)은 동영상을 구분 할 때의 기본단위로서 필름이 끊기지 않고 연결된 프레임 집합의 구성이다. 장면(Scene)은 하나의 촬영대상이 연속된 샷들의 집합으로 이루어진 공간상, 시간상의 의미적 분리의 구성이다. 컷은 샷과 샷사이의 경계이며 장면이 변화된 프레임이다.

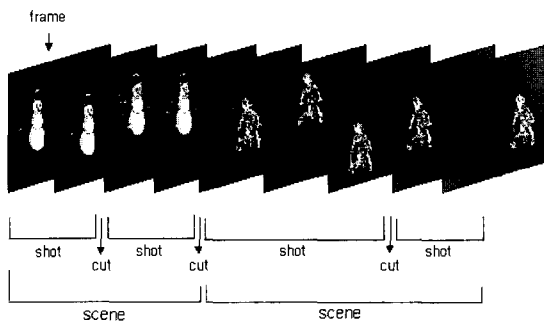
계층적 구조화된 동영상 데이터는 내용전환으로 구분되는 연속적인 스토리로 구성되고, 각 스토리는 장면전환의 단위인 샷으로 구성된다. 동영상 데이터를 샷으로 구분하는 작업을 비디오 분할(Video Segmentation)이라 하고, 비디오 분할을 위해 컷 검출 작업을 하여야 하고, 컷 검출은 연속적인 프레임으로 이루어진 동영상데이터의 특징을 이용한다. 연속되는 프레임들은 인접 프레임 사이는 유사성이 매우 강하자만 장면 변화가 이루어진 부분의 프레임은 상대적으로 프레임간의 유사성이 약

하다. 컷 검출을 위해서는 동영상데이터의 프레임간 특징 값 차이를 이용하여 각 프레임의 연속성을 계산하여 불연속 지점으로 인식되어진 지점을 컷으로 간주한다.

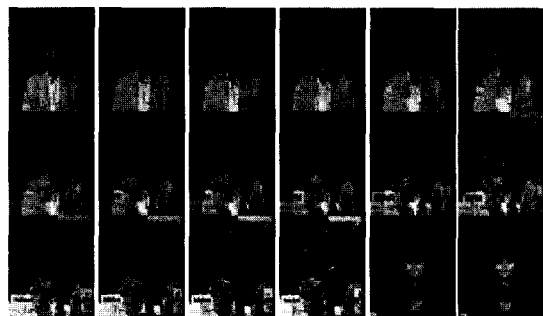
2.2 동영상 데이터의 컷(cut) 검출 기법

동영상데이터는 내용이 바뀔 때마다 해당 내용을 분류하고, 인덱싱(indexing)해야만 동영상데이터 정보를 체계적으로 데이터베이스화하고 효율적으로 검색할 수 있다. 기본적으로 주로 카메라 샷(camera shot)에 의해 구분되어지는 컷 검출은 여러 가지 편집효과에 의한 페이드-인(fade-in), 페이드-아웃(fade-out), 디졸브(dissolve)와 같은 점진적 변화[14]와 주밍(zooming), 패닝(panning)과 같은 카메라의 움직임으로 발생하는 점도 고려해야 한다[7].

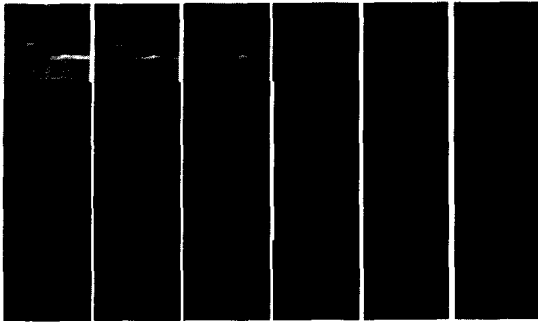
비디오 분할은 동영상데이터에 따라 비압축 비디오 스트림에 대한 분할과 압축 비디오 스트림에 대한 분할로 구분된다. 기본이 되는 비압축 비디오 스트림에 대한 컷 기법으로는 단순히 두 프레임간의 대응 픽셀을 비교하여 밝기 값이 변화한 픽셀을 양으로 표현하여 일정한 임계치를 초과할 경우 장면이 전환한 것으로 판단하는 화소 비교(pixel comparison)방법[5]이 있으나 영상내의 부분적인 변화에 약한 단점을 가지고 있다. 이 문제를 문제 해결을 하기 위하여 프레임간의 히스토그램의 변화가 일정한 임계치를 초과할 경우 장면이 변환되었다고 판단하는 히스토그램 비교(comparison)



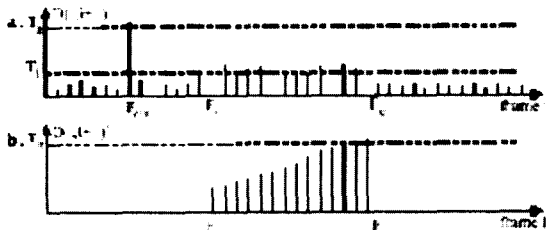
(그림 1) 동영상의 구조



(그림 2) 디졸브



(그림 3) 페이드-인, 페이드-아웃



(그림 4) 히스토그램 비교 - a.연속적인 경우
b.히스토그램 차이 합

방법[13]이 제안되었다. 히스토그램 비교의 또 다른 방법으로 컷이 발생할 경우 픽셀값의 변화가 랜덤하게 분포함으로 프레임간 같은 위치의 픽셀 밝기값 차이에 대한 히스토그램을 구하여 이것의 분산이 일정한 임계치 이하일 경우 컷으로 판단하는 방법이다. 이들 방법들은 주로 영상 내 밝기값의 분포에만 의존하므로 모양(shape), 색상(color), 질감(texture)과 같은 영상의 내용정보를 반영하지는 못하고 있다.

동영상데이터의 내용정보를 장면변환이 이루어진 컷 검출에 이용하기 위하여 윤곽 추출 필터(edge retrieval filter)를 이용하여 에지를 추출한 후 프레임간의 에지의 개수를 비교하여 개수의 차이가 일정한 임계치를 초과할 경우 장면 변화로 판단하는 윤곽 픽셀 비교 방법이 있다[1,5].

또한, 각 프레임을 블록으로 분할하여 각각 움직임 벡터(motion vector)를 구한 후 움직임 연속성(motion continuity)을 이용한 방법[4]과 압축 비디오 스트림에 대한 컷 검출 방법에는 DCT DC 계수를 이용한 컷 검출방법[5,6], 움직임 정보를

이용한 컷 검출 방법이 있다. 그 외 영상색상대비를 특징으로 하는 방법, 블록비교와 움직임 패턴 추적을 통한 와이프 검출 방법[12]등이 있다.

2.3 동영상데이터의 컷 검출 알고리즘

(1) 픽셀차이(pixel difference)를 이용한 방법

두 프레임의 동일한 위치의 픽셀 강도 값의 차이들을 비교하여 두 프레임사이의 변화를 검출한다. 이 알고리즘은 변화된 픽셀의 수를 카운트하여 만약 전체 픽셀에 대한 변화된 픽셀의 퍼센트가 임계치를 넘으면 장면 변화가 발생하는 것으로 보고 컷을 추출한다[4,7,9]. 픽셀의 차이와 임계치 계산을 수학적으로 식(1)과 (2)와 같이 나타낼 수 있다. 식 (1)에서 $F_i(x,y)$ 는 프레임 i 의 좌표점 (x,y) 에서 픽셀의 강도 값으로 만약 두 개의 연속적인 프레임에서 대응하는 픽셀 사이의 차이가 최소 강도 값보다 크다면 다른 그림으로 판단하고 $DPI(x,y)$ 는 1로 설정한다. 식 (2)에서 두 프레임의 픽셀 사이의 퍼센트 차이는 프레임에서 픽셀값 차이의 합을 전체 픽셀수로 나눈 것에 의해 계산한다.

$$DP_i(x,y) = \begin{cases} 0 & \text{if } |F_i(x,y) - F_{i+1}(x,y)| > t \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

$$\frac{\sum_{x,y=1}^{X*Y} DP_i(x,y)}{X*Y} * 100 > t \quad (2)$$

이러한 픽셀값 비교는 카메라의 움직임과 물체의 움직임에 매우 민감하다는 단점을 가지고 있어 컷 검출에 어려움이 있다. Pan이나 Zoom같은 카메라 이동은 많은 수의 픽셀 변화 효과를 가져와서 장면 변화된 것으로 인식되며, 빠르게 이동하는 물체 역시 똑같은 픽셀의 변화효과를 가짐으로 픽셀의 변화를 픽셀의 값과 그 픽셀과 연결된 픽셀을 비교한다면 카메라 효과와 물체움직임에서의 컷 검출이 줄어들게 되는 방법이다.

(2) 명도 히스토그램을 이용한 방법

카메라와 움직이는 물체의 감도는 두 프레임의 명도 히스토그램의 비교에 의해 감소되는데[4,7], 이것은 두 프레임의 배경에 많은 차이가 없어 약간의 물체 움직임은 거의 유사한 히스토그램을 갖는다는 사실에 기인한다. 히스토그램은 프레임에서 각각의 명도값에 해당하는 픽셀들의 수를 나타낸 것이다.

$$\sum_{j=0}^g |H_i(j) - H_{i+1}(j)| > t \quad (3)$$

식(3)에서 g 는 명도값의 개수이고, j 는 명도값, i 는 프레임 번호, $H_i(j)$ 는 명도값 j 에 대한 히스토그램이다. 만약 합계가 주어진 임계값 t 보다 크다면 변화가 있는 것으로 인식한다. 칼라 코드를 이용한 히스토그램 비교도 제안되었는데 칼라 코드값은 세 가지 칼라 값으로부터 유도되었다. bin 크기를 줄이기 위해 24비트로 나타내고 각각의 칼라의 상위 2비트는 칼라코드를 나타내는데 이용한다. 위의 식에서 j 는 코드 값으로 대체된다. 칼라 히스토그램 좌표는 칼라값을 그레이 레벨로 변환하는 작업이 필요하지 않기 때문에 보다 견고하다고 할 수 있다.

$$\sum_{j=0}^g \frac{|H_i(j) - H_{i+1}(j)|^2}{H_{i+1}(j)} > t \quad (4)$$

```

For cmp_loop2 = 1 to lastFrame - 1
{
for cmp_loop2 = 1 to lastFrame - 1
{
cmp_hap = cmp_hap + (s_value[cmp_loop1]
[cmp_loop2] - t_value[cmp_loop1]
[cmp_loop2+1])
}
}
cmp_value = cmp_hap;
return cmp_value;
    
```

(그림 5) 히스토그램을 이용한 컷 검출 알고리즘

χ^2 -test 칼라 히스토그램 비교법은 Nagasaka와 Tanaka에 의해 제안[6]되었으며, 카메라 정지와 카메라와 물체의 이동에 의한 작은 변화에서도 차이를 확장[7]시킴으로 그레이 또는 칼라 히스토그램 비교 기법보다 효율적이지 못할 수도 있는 한계를 가지고 있다.

(3) DCT 계수를 이용한 방법

컷을 추출하기 위해 압축 비디오 영상을 이용[9,11]하기 때문에 처리해야할 데이터의 양이 상당히 줄어든다. 비디오의 압축은 8×8 픽셀 블록으로 이미지를 나누어 수행한다. 블록내의 픽셀은 64개의 DCT(Discrete Cosine Transform)계수로 변환하고 이것을 양자화와 Huffman 엔트로피 부호화한다.

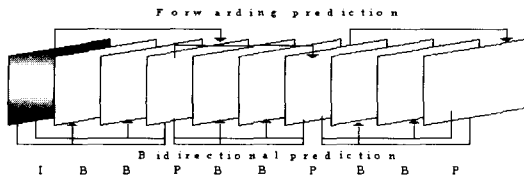
DCT 계수는 어떤 프레임에서 장면변환이 일어나는지 찾는데 이용된다. 빈도에서의 계수는 어떤 수학적으로 공간 영역과 연관되어 있으므로 연속된 비디오에서 전환점을 찾는데 이용할 수 있다. 단일 DCT기반 부호화된 비디오 프레임 f 의 8×8 블록이 주어진다면 블록의 부분집합은 쉽표로 분리한다. 각각의 프레임에 n 개의 연결된 지역으로부터 선택되어 진다. 또 각각의 블록을 위해 64개의 부분집합이 선택되어 계수의 선택은 블록의 AC계수 사이에서 임의로 할당한다. 각각의 프레임으로부터 가져오는 계수의 벡터는 다음과 같다.

$$V_f = \{c_1, c_2, c_3, \dots\}$$

벡터 V_f 는 DCT 공간에서 연속된 비디오의 f 프레임을 나타내며 내부의 요소들은 두 프레임 사이의 차이를 찾아내는데 이용한다.

$$\varphi = \frac{\vec{V}_j \cdot \vec{V}_{j+1}}{|\vec{V}_j| \cdot |\vec{V}_{j+1}|} \quad (5)$$

식 (5)에서 V_j 는 비교할 프레임의 벡터이고, V_{j+1} 은 완성된 프레임의 벡터를 나타낸다. 임계값이 t 인 곳에서 $1-|\varphi| > t$ 일 때 변환점이 검출된다.



(그림 6) GOP의 구조

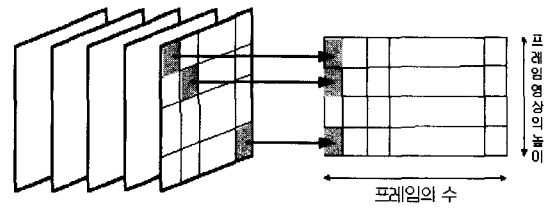
Zhang은 DCT계수보다 MPEG압축 영상에서 움직임 벡터를 이용한 움직임 기반 장면분할, Meng은 MPEG/MPEG-2 압축 영상에서 보다 정확한 동작을 하도록 개념을 확장 시켰다. 확장된 압축 영상은 I, P, B 프레임으로 구성되었으며 프레임의 구성을 도식화하면 그림 6과 같다. 압축된 영상의 I-프레임은 완전한 내부 프레임 코드이므로 움직임에 대한 보상이 수행되지 않는다. P-프레임은 I 또는 P 프레임으로부터 움직임 보상을 하는 예측 코드이다. 이 두 프레임은 비순차적 움직임 보정이 된 B 프레임을 위한 기반으로 이용된다.

만약 장면전환의 전환이 B-프레임에서 일어난다면 움직임 벡터의 대부분은 다음 프레임으로부터 오고 극소의 것은 이전 프레임으로부터 올 것이다. 전후방으로부터의 움직임 벡터의 비율에 의하여 장면 전환점은 발견된다. P-프레임들은 단지 전 방향 움직임 보정을 가지고 있기 때문에 이곳에서 프레임의 장면전환의 변화가 일어난다면 엔코더는 움직임 보정을 위한 이전 프레임으로부터의 매크로 블록을 이용할 수 없다. 움직임 보정이 있는 매크로 블록에 대한 움직임 보정이 없는 매크로 블록의 비율에 의존하여 장면전환의 전환점이 검출된다. I-프레임은 움직임 벡터를 가지지 않는 완전한 내부코드이기 때문에 기존의 방법 중 어떠한 것을 이용하여도 장면전환에 대한 전환점인 프레임 컷을 검출하는데 이용할 수 있다.

3. 제안한 동영상데이터 컷 검출시스템

3.1 동영상 데이터의 특정영역 표본추출

동영상 데이터는 장시간에 연속적인 프레임들이



(그림 7) 프레임의 추출한 픽셀 값 저장 구조

서로 유사한 배경 구성과 이미지들의 변형으로 이루어짐으로 연속성이 끊어지는 부분이 본 연구에서 검색 대상이 되며 그 영상이 컷이 된다.

각 프레임 별로 동영상 데이터의 대각선 방향으로 특정영역을 표본추출 하여 대용량 동영상 데이터를 소용량의 정지영상으로 전환과 (i, j) 행렬 A로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장한 내용을 그림으로 도식화하면 그림 7과 같다.

본 연구에서는 일반적인 장면 전환의 프레임의 컷 검출과 장면 전환이 점진적으로 변화는 장면 전환 프레임을 검출하는데 중점을 둔다. 이를 위하여 동영상 데이터 A를 정의하는데 $A(i, j)$ 행렬로 i 는 프레임의 수, j 는 프레임 영상의 높이로 나타낸다. 각 프레임의 픽셀의 RGB값들을 대각선으로 하여서 대표 값들로 표본추출된 행렬 A의 열 a_i 과 a_{i+1} 의 차이가 큰 i 열이 장면 전환이 일어나는 프레임의 컷으로 검출한다.

동영상의 각 프레임에서 추출된 데이터를 소용량의 정지영상으로 표본추출한 결과를 그림 8과 같이 행렬 구조로 픽셀값이 저장되며, 행렬 구조로 저장되어 있는 표본추출 영상에서 일반적인 영상데이터는 프레임 컷이 수직선상에 배열한다. 배열되어진 픽셀 값들은 차이가 나타나 픽셀의 변화가 있음을 알 수 있다. 동영상의 각 프레임에서 데이터를 추출하는 방법은 동영상의 카메라의 초점을 항상 표적이 되는 영상의 중앙에 두고 있다는 점에서 기인하여 대각선방향으로 픽셀을 나누어 점진적으로 변화는 장면전환이 발생한 프레임의 컷은 픽셀의 값들이 점진적인 변화로 나타나게 된다. 이러한 영상 데이터는 여러 프레임간

의 간격으로 표본추출 함으로써 그 크기를 줄일 수가 있다. 또한 동영상 데이터의 전체 구조를 파악할 수 있다는 특징을 가지고 있다.

3.2 영상 데이터의 색상 차의 자승합 추출 알고리즘

동영상의 각 프레임들의 RGB값을 대각선 방향으로 추출한 값들은 행렬 A의 열 a_i 와 a_{i+1} 의 차이가 큰 i열이 장면 전환이 일어나는 곳이다. 행렬 A의 열 a_i 과 a_{i+1} 의 차이를 정의하기 위하여 다음 알고리즘을 이용한다.

$$E_1 = \sqrt{(a_{11} - a_{12})^2 + (a_{21} - a_{22})^2 + \dots + (a_{n1} - a_{n2})^2} \quad (6)$$

식 (6)에서 E_1 은 영상 데이터 행렬 A의 1열 a_1 과 2열 a_2 의 색상 차의 자승합을 의미한다.

```

; 동영상 데이터 추출
for (j=1 to frame x축 값)
    Store A[i][j]=image[j][i]; //i frame의 대각선 방향의
    데이터 값
    Pict_Image[i][j]=image[j][i]; //i frame의 대각선 방향의
    데이터 표본추출;

; 추출데이터의 SSD 구함
for(f=1 to frame_end) {
    for(g=1 to j) {
        hap = hap + (Store [g][f] - StoreA[g][f+1])^2
    }
    SSD_Tab[f]=SQRT(hap);
    hap = 0;
};
k=0
for(f=1 to frame_end) {
    Chang[f]=SSD_Tab[f] - SSD_Tab[f+1]; //각 프레
    임의 오차
    if (Chang[f]>=t)
        Chang_Image[]=f; //동영상 데이터 저장
        k=k+1; //k는 장면변화 수
};
    
```

(그림 8) 동영상데이터의 프레임간의 색상차의 자승합 추출 알고리즘

$$E_k = \sqrt{(a_{1(k-1)} - a_{1k})^2 + (a_{2(k-1)} - a_{2k})^2 + \dots + (a_{n(k-1)} - a_{nk})^2} \quad (7)$$

식 (7)에서 E_k 은 영상 데이터 행렬 A의 k-1열 a_{k-1} 과 k열 a_k 의 SSD이다. 이렇게 색상차의 자승합을 이용하여 식(7)에서 얻어진 E_k 을 이용하여 프레임의 컷 f 를 찾을 수가 있다.

$$|E_{f-1} - E_f| \geq t \quad (8)$$

식(8)에서 t 는 프레임 컷을 찾는 임계값[4]이고, E_f 는 전환이 일어나는 프레임 번호를 나타낸다.

4. 구현 및 실험 결과

사용자가 입력한 질의영상에 대해서 유사한 여러 후보영상을 검색하는 내용기반 영상검색시스템의 성패여부는 영상의 특징을 빠르고 효과적이면서 정확하게 추출해내는가 하는 특징추출 부분이다. 따라서 사용자 인터페이스를 통해 입력된 원래의 영상에 대해 시스템은 특징 추출하는 모든 과정을 자동으로 수행해야 한다. 이는 통상적으로 일정시간이 소요되는 영상처리시간의 단축을 위해서도 사용자의 간섭 없이 특징의 추출 전 과정이 빠르게 진행되어야 하기 때문이다. 본 연구에서 사용된 시스템 CPU는 Pentium III 700Mhz 이고 RAM은 512MB, VGA는 GeForce2MX이다. 사용되는 운영체제는 Windows 2000 Server에 Visual C++ 6.0을 이용하였다.

동영상데이터를 일상에서 접할 수 있는 기회는 무수히 많다. 이렇게 많은 동영상 중에서 광고, 뉴스, 영화, 드라마를 본 연구에서 이용하였다. 이들 동영상자료들 중에서 광고데이터의 경우는 특수효과를 많이 사용하기 때문에 검색의 효율을 측정해 볼 수 있는 좋은 데이터이다. 뉴스데이터는 빠른 장면 변화에 민감한 반응을 하는 것을 알아 볼 수 있고, 영화와 드라마데이터는 사물에 대해 정확하고 민감하게 반응할 수 있는지 알아

볼 수 있게 하는 좋은 동영상데이터이다. 본 연구에서 사용되는 데이터는 광고, 뉴스, 영화, 드라마 네 가지 데이터를 동일한 환경에서 AVI 파일 포맷에 맞춰 성능 측정을 하였다.

본 연구에 이용된 동영상 데이터의 동영상 종류별 프레임크기와 프레임의수, 그리고 컷의 개수는 표 1과 같다. 동영상데이터의 컷 검출 시스템의 초기화면은 화면 좌측에 검색에 이용될 동영상데이터를 오픈 로드하여 본 연구에서 제안한 컷 검출과 기존 연구의 컷 검출 알고리즘과 비교할 수 있도록 픽셀의 차이를 이용한 알고리즘 검색과 명도 히스토그램을 이용한 알고리즘 검색을 그림 9와 같이 제안검색, 픽셀검색, 히스토그램 3개의 버튼을 두어 검색하고자하는 버튼을 누르면 원하는 검색이 자동으로 이루어지도록 구현하였다. 이렇게 구해진 프레임은 중간화면에 검색 프레임의 정보와 리스트가 출력 되도록 하여 검색된 프레임 리스트정보에선 검색된 프레임번호와 SSD의 알고리즘에 의해 얻어진 결과값을 나타내었다. 예를 들어 15 frame=64는 컷검출이 15프레임에서 발생하였고 SSD의 값이 64를 의미한다.

제안한 컷 검출 시스템의 임계값은 검출된 컷의 수가 동영상데이터에 따라 차이가 있으므로

(표 1) 컷 검출에 이용한 동영상 데이터의 포맷

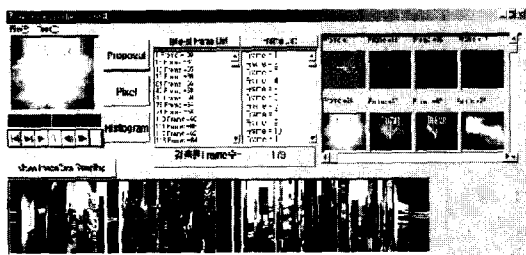
동영상 종류	프레임 크기	프레임수	컷의 개수	파일포맷
광고	320×320	480	95	AVI
뉴스	320×320	625	196	
영화	320×240	805	217	
드라마	320×240	98	13	

동영상데이터에 따라 Yeof[4]의해 제안된 슬라이드 윈도우법을 적용한 임계값으로 변화시켜 결과를 얻었다. 그림 9, 10, 11에서 보여지는 동영상자료는 영화데이터로 총 805개의 프레임으로 시각적으로 확인 할 수 있는 프레임의 컷의 수는 170개이다. 이러한 동영상데이터를 이용하여 컷을 검출하였다. 검출된 컷은 precision rate와 recall rate를 기존연구의 알고리즘과 본 연구에서 제안한 알고리즘과 비교하였다. 비교된 자료는 표 2와 표 3에서 나타나고 있다.

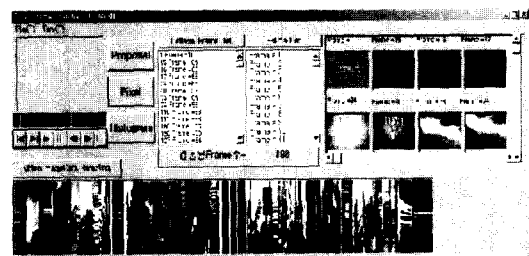
기존에 연구된 검색 시스템에서 프레임의 컷을 추출한 결과인 그림 9는 픽셀의 차이를 이용한 검색이며 그림 10은 명도 히스토그램을 이용한 결과이다. 제안한 검색 시스템에서 추출한 결과의 화면은 그림 11과 같고 동영상 데이터의 컷 검출이 끝나면 추출된 프레임 리스트가 화면에 디스플레이 되어 프레임리스트 중에서 원하는 프레임을 클릭하면 선택된 프레임부터 동영상데이터를 원하는 만큼 재생 할 수 있다. 그림 9, 10, 11의 결과 화면은 본 연구에서 실험데이터로 이용한 영화, 뉴스, 드라마, 광고의 동영상데이터 중 영화 데이터이다.

그림 10은 각 프레임의 픽셀 값을 이용하여 프레임간의 픽셀 값의 차이를 가지고 장면 전환점을 검색하는 방법으로 프레임의 잠움에 민감하게 반응을 한다. 특히 드라마와 같은 동영상 데이터의 컷 검출에서는 거의 검출되지 않는 취약점을 보인다.

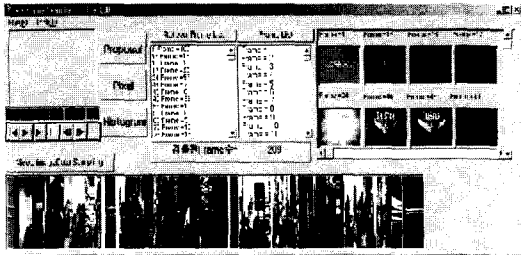
그림 10은 프레임의 명도 히스토그램을 찾아 프레임간의 히스토그램의 차이를 비교하는 방법



(그림 9) 픽셀차이를 이용한 방법의 결과



(그림 10) 히스토그램을 이용한 방법의 결과



(그림 11) 제안한 알고리즘에 의한 컷 검출 결과

으로 이 방법은 히스토그램 차이가 일정한 임계값 이상이면 컷이 발생한 것으로 인식하는 기법으로 픽셀차이를 이용하는 방법보다는 잡음에 강한 성질을 갖는다. 하지만 이 방법 또한 드라마와 장면전환이 미묘한 동영상데이터에서는 컷을 검출하는데 있어서 취약하다.

실험에 이용한 네 가지 유형의 동영상 데이터는 각각의 데이터가 지니는 특성이 있어 본 연구에서 실험한 동영상 데이터의 컷 검출 결과가 효율적인가를 판단하기 위해 성능평가 척도로 Precision Rate과 Recall Rate[19]를 사용한다.

$$Precision\ rate = \frac{\text{number of detection cuts}}{\text{number of total frame cuts}} \quad (9)$$

$$Recall\ rate = \frac{\text{number of precision cuts}}{\text{number of detection cuts}} \quad (10)$$

precision rate는 전체 동영상데이터에서 표본추출된 전체 프레임의 컷에서 실제 컷이 발생한 부분을 검출하는 비율을 의미한다. 이는 실제 컷은 아니지만 컷으로 오인해서 잘못 검출된 컷이 얼마나 많은가를 나타낸다. recall rate는 실제 컷이 존재하는데 검출하지 못하는 컷이 얼마나 되는가를 의미하는 척도를 나타내는 것이다. 일반적으로 동영상 비디오 색인을 위한 구조화나 클러스터링 과정을 수행하면 유사한 내용의 샷을 하나의 논리적인 단위인 이벤트로 묶어주게 된다. 그

러므로 컷이 아닌 부분을 컷으로 잘못 검출하더라도 클러스터링에 의해 같은 내용 단위로 묶을 수 있다. precision rate가 너무 낮을 경우 클러스터링 수행 횟수가 많아지게 된다. 반면에 검출하지 못한 컷이 있을 경우 클러스터링 과정에서 샷간 유사도를 계산할 때 잘못된 값이 나오게 되어 적절한 비디오 구조화가 이루어지지 못한다. 따라서 효율적인 비디오 동영상데이터 검색 및 색인을 위해서는 precision rate와 recall rate가 모두 높아야 한다.

현재 비교의 대상으로 선정된 픽셀차에 의한 검출, 히스토그램에 의한 검출은 컷 검출에 보편적으로 이용하는 방법이므로 제안한 알고리즘과의 비교를 표 2와 표 3과 같이 확인할 수 있다. 광고영상에서는 다른 방법에 비해 7%정도 우수한 검출율을 보여주고 있다.

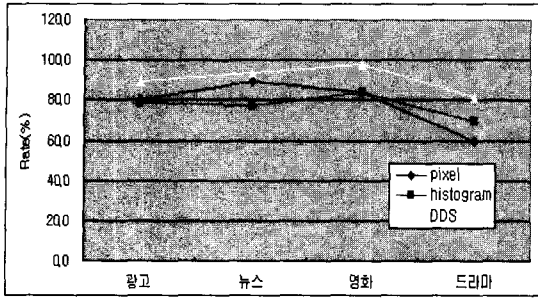
검출율의 비교에서 드라마처럼 정면의 변화가 적은 동영상데이터에서 장면변화가 많은 광고 테

(표 2) precision rate 비교

Precision rate	pixel	histogram	DDS
광고	63.16 (60/95)	61.05 (58/95)	84.21 (80/95)
뉴스	80.61 (158/196)	88.78 (174/196)	91.84 (180/196)
영화	82.49 (179/217)	91.24 (198/217)	96.31 (209/217)
드라마	82.95 (10/13)	88.94 (10/13)	95.29 (11/13)

(표 3) recall rate 비교

Recall cate	pixel	histogram	DDS
광고	81.67 (49/60)	77.59 (45/58)	88.75 (71/80)
뉴스	89.24 (141/158)	77.01 (134/174)	97.22 (175/180)
영화	84.36 (151/179)	81.82 (162/198)	90.43 (189/209)
드라마	60.00 (6/10)	70.00 (7/10)	81.81 (9/11)



(그림 12) 검출율의 비교

(표 4) 동영상데이터 실험 결과

동영상 데이터종류	급격한 장면 변화		점진적인 장면 변화	
	detection	missing	detection	missing
광 고	31	5	11	0
드라마	7	0	2	0
영 화	112	12	72	5
뉴 스	121	8	53	2

이러가지 14%에서 21%의 검출 결과 차이를 보였다. 검출율은 precision rate를 전체 검출된 프레임 수로 나누어진 비율을 의미하며, 전체적으로 본 연구에서 제안한 알고리즘이 19% 정도의 효율적인 검출결과를 보였다. 급격한 장면변화가 있는 동영상 데이터와 점진적인 장면변화가 있는 동영상 데이터의 프레임간 색상차를 이용한 컷을 검출하여 비교한 결과데이터는 표 4와 같다. 나머지 검출하지 못한 컷들은 직접 확인하여 보면 특수 효과 부분이나 식(8)의 두 프레임간의 차이가 양수를 나타내고 있으나 해당구간이 프레임의 변화가 모호한 경우일 때 일어난 것으로 나타났다.

점진적 장면변화는 검출해야 할 대상이 장면 변화 효과 그 자체가 아닌 변화가 일어나기 전과 후의 장면이 되므로 어떠한 형태의 점진적인 전환이라 해도 전 후의 장면에서 추출되는 특징값의 변화가 안정적이라면 두개의 장면을 확실하게 구별하는 장점이 있다. 본 연구에서 제안한 알고리즘은 점진적인 장면 변화가 있는 동영상데이터를 검출하는 데 대체로 효율적이다.

5. 결 론

멀티미디어 기술과 멀티미디어데이터를 이용한 자료이용이 보편화되고 있는 현재의 정보화 환경에서 동영상으로 이루어진 데이터의 정보를 효과적으로 이용하고 관리하기 위해서 내용기반으로 한 동영상데이터의 요약된 정보와 검색 그리고 인덱싱하는 과정에서 기초가 되는 프레임의 컷 검출이 필수적이다. 본 연구에서는 동영상데이터의 컷 검출에 대한 새로운 알고리즘을 제안하였고, 제안한 알고리즘은 프레임의 픽셀의 색상차를 이용한 방법을 적용하여 각각의 프레임을 대각선 방향으로 영상 데이터를 검출하여 대용량의 동영상 데이터를 정지영상으로 표본추출하여 동영상의 전체 구조를 파악할 수 있도록 하였으며, 각각의 프레임에서 추출한 픽셀의 칼라 값은 $A(i, j)$ 행렬로 i 는 프레임 수로 j 는 프레임의 영상 높이로 저장하여 색상차의 합을 이용하여 장면변화가 발생하는 컷을 신속하고 정확한 컷 검출이 이루어지도록 컷 검출 시스템을 구현하였다.

본 연구에서는 각각의 동영상의 내용에 따라 표현하는 방법이 다른 네 가지 종류의 동영상 데이터를 비교대상으로 사용하였다. 기존의 방법에서 많이 이용되고 있는 픽셀값에 의한 방법, 히스토그램에 의한 검출법과 비교한 결과에서 detection rate에서 20%정도의 효과적인 검출을 하였으며, precision rate면에서는 10%정도 향상됨을 확인하였다. 기존의 방법들은 점진적인 장면 변화가 있는 부분에서는 민감하지 반응하지 못하였으나 제안한 알고리즘에서는 대체로 효율적이었다.

따라서 본 연구에서 제안한 시스템이 의료기관 의 의료 영상데이터처럼 프레임의 변화가 미세한 부분에 이용하였을 때 매우 효과적이다. 앞으로 영상데이터를 인덱싱하기 위해서는 단순한 프레임 전환뿐만 아니라 특수효과를 내는 카메라 등의 이미지와 같은 복잡한 프레임 변화에도 검출되도록 하는 연구가 계속되어야 할 것이다.

참고 문헌

- [1] A. Akutsu et al, "Video Indexing using motion vectors", Visual Communications and Image Processing '92, pp. 1522-1530, Boston, MA, SPIE November 1992.
- [2] A. Hampapur, R. Jain and T. Weymouth, "Production Model based Digital Video Sequention", Multimedia Tools and Application, Vol. 1, No. 1, pp.9-46, Mar. 1995.
- [3] Adnam M. Alattar, "Wipe Scene Change Detector for Segmenting Uncompressed Video Sequences", ISCAS '98, The Informational Symposium on Circuits and System, Monterey, California, May. 1998.
- [4] B. L. Yeo, b. Liu, "Rapid Scene Analysis on Compressed Video," IEEE Tran. on Circuit and System for Video Technology, Vol.5, No.6. pp. 533~544, December 1995.
- [5] B. Shahraray, "Scene Change Detection and Content-Based Sampling of Video Sequences", Proc. IS&T/SPIE, Vol. 2419, CA, February 1995.
- [6] C. Gu, and M. C. Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects", IEEE CSVT, Vol. 8, No. 5, pp. 572~584, September 1998.
- [7] C. W. Neo, T. C. Pong, R. T. Chin, "Video Partitoning by Temporal Slice Coherency", IEEE Tran. on Circuit and System for Video Technology, Vol.11, No.8, Aug., 2001.
- [8] D. Li, H. Lu, "Model based Video Segmentation", The IEEE Workshop on Signal Processing Systems, pp. 120-129, 2000.
- [9] F. Arman, A. Hsu, and M.-Y. Chiu, "Image Processing on Compressed Data for Large Video Databases", Proc. 1st ACM Intl. Conf. on Multimedia, Anaheim CA, pp. 267~272, August 1993.
- [10] James. D. Murray & William Varrayper, 'Graphics file format', 2nd Ed., O'Reilly & associates Inc, 1996.
- [11] J. Meng, Y. Juan, S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence", Proc. IS&T/SPIE 2419, February 1995.
- [12] M. G. Chung, H. K. Kim, M. H. Song, "A Scene Boundary Decton Method", Image Processing 2000, Proceeding 2000 International Conference, Japan, July. pp. 933~936, 2000.
- [13] R. Lienhart, A. Zaccarin, "A System for Reliable Dissolve Detection in Videos", Proceeding. 2001 International Conference on Image Processing, Vol.3, Japan, pp. 406~409, August, 2001.
- [14] R. John, et al. "Integrated Spatial and Feature Image Systems : Retrieval, Analysis and Compression", Submitted in partial fulfillment of the requirements for the degree of Doctor of philsofhy in the Graduate School of Arts and Sciences, Columbia University, 1997.
- [15] R. Zabih, J. Miller, K. Mai, "A Feature based Algorithm for Detection and Classifying Scene Breaks", Proc. ACM Multimedia 95, pp. 189~200, San Francisco, CA, Nov. 1995.
- [16] S. Berretti, A. Del Bimbo, P. Pala, "Sensations and Psychological Effects in Color Image Database", IEEE International Conference on Images Processing, NewYokrk USA July, pp. 560~563, 1997.
- [17] S. Loncaric, "A survey of shape analysis techniques", Pattern Recognition, Vol. 31, No. 8, pp. 983~1001, 1998.
- [18] T. S. Chua, S. K. Lim and H. K. Pung, "Content-Based Retrieval of Segment Images", ACM, pp. 211~218, Oct. 1994.
- [19] J. S. Boreczsky and L. A. Rowe, "Comparison of Video Shot Boundary Detection Techniques," Proc. Storage and Retrieval for Image and Video Database IV, Vol. SPIE-2670, pp. 170~179, 1996.

◎ 저자 소개 ◎



김 병 철

1967년 서울대학교 수학과 졸업(학사)
1982년 성균관대학교 경영행정대학원 전자·자료처리학과 졸업(석사)
1984년 3월~1987년 2월 제주대학교 조교수
1995년 3월~1997년 2월 순천대학교 전자계산소장
1987년~현재 : 순천대학교 공과대학 정보통신공학부 정교수
관심분야 : 전산수학, Image Processing, 이산구조
E-mail : kbc@sunchon.ac.kr



정 창 렬

1995년 광주대학교 전자계산학과 졸업(학사)
1999년 순천대학교 교육대학원 컴퓨터교육학과 졸업(석사)
2000년~현재 : 순천대학교 대학원 컴퓨터과학과 박사과정
관심분야 : 정보보안, Image processing, Mobile Agent, Authentication, Electronic Commerce
E-mail : chari7@mail.knou.ac.kr



고 진 광

1982년 홍익대학교 컴퓨터공학과 졸업(학사)
1984년 홍익대학교 대학원 컴퓨터공학과 졸업(석사)
1997년 홍익대학교 대학원 컴퓨터공학과 졸업(박사)
1984년 3월~1998년 2월 송원전문대학 전자계산과 전임강사
1988년 3월~현재 : 순천대학교 공과대학 정보통신공학부 정교수
1992년 3월~1993년 2월 홍익대학교 공과대학 컴퓨터공학과 국내교류교수
1997년 8월~1998년 2월 미국 Oregon State Univ. 컴퓨터공학과 방문교수
2000년 12월~2001년 2월 일본 류큐대학 정보공학과 방문교수
2001년 3월~현재 : 순천대학교 정보전산원장
관심분야 : 데이터베이스, 전자상거래, Image processing
E-mail : kjg@sunchon.ac.kr