

구조화된 논리적 비디오를 이용한 비디오 브라우징 및 검색 시스템

권 성 복[†] · 조 영 우^{††} · 김 영 모^{††}

요 약

본 논문에서는 비디오를 체계적으로 관리하고 비디오 브라우징, 검색, 편집 등의 비디오 응용 분야에 효과적으로 적용할 수 있도록 기존의 비디오를 구조화된 형태의 논리적 비디오로 재구성한 구조화된 논리적 비디오를 제안하였다. 제안한 구조화된 비디오는 비디오의 물리적 정보뿐만 아니라 비디오의 논리적 정보를 포함하며 이러한 정보를 이용하여 기존의 비디오가 제공하지 못하는 비디오의 논리적 단위 편집 등의 다양한 기능을 제공할 수 있고, 비디오 브라우징, 검색, 비디오 편집 등의 여러 비디오 응용 분야에 효과적으로 적용할 수 있으므로 비디오의 재사용성을 높일 수 있다. 본 논문에서는 기존의 비디오를 구조화된 논리적 비디오로 재구성하여 관리하는 비디오 데이터베이스 관리 시스템과 이를 이용하여 비디오 브라우징 및 검색을 효과적으로 수행하는 비디오 응용 시스템을 설계 및 구현하였다.

Video Browsing and Retrieval System using Structured Logical Video

Sung-Bok Kwon[†] · Young-Woo Cho^{††} · Young-Mo Kim^{††}

ABSTRACT

In this paper, we propose a new video structuring method for reconstructing a 'structured logical video' which adds logical video information to a conventional structured video. Logical video information is composed of background, object, and situation. The proposed structured video is reconstructed in logical unit like episodes as well as in physical unit like shots. It can support many video functions and may be reusable in many video applications-video browsing, video retrieval, video editing, etc. We also present the video database management system in order to manage the proposed structured video. In the proposed video database management system, the information in the proposed structured video is modeled on object-oriented concept and hierarchical structure. Also, we design and implement a video browsing and retrieval system using the proposed structured video.

1. 서 론

※이 연구는 1995년 정보통신부 연구비 지원에 의한 결과임.

† 준 회 원: 경북대학교 전자공학과

†† 정 회 원: 경북대학교 전자공학과

논문접수: 1996년 10월 21일, 심사완료: 1997년 5월 15일

현재 비디오는 산업이나, 교육, 그 외의 많은 분야에서 중요한 정보의 교환 수단으로 발전하고 있으며 멀티미디어 기술의 핵심 분야로 취급되고 있다. 최근에 이루어진 카메라 및 디스플레이 장치나 동영상 압축 기술 등의 발달로 인해서 문자나 정지 영상뿐 아

니라 비디오를 처리하는 분야의 연구가 활발하게 이루어지고 있다.^[1] 그리고 디지털 비디오는 기존의 아날로그 비디오가 지원하지 못했던 사용자의 요구를 다양하게 지원할 수 있으며 컴퓨터를 이용하여 비디오 정보를 효과적으로 구성, 편집, 저장할 수 있다.

그러나 비디오를 이용하는 데에는 여러 가지 문제점이 있다. 첫째는 기존의 문자나 정지 영상에 비해 데이터의 크기가 대단히 크므로 데이터를 저장하는데 대용량의 저장 매체가 필요하고 비디오의 내용을 확인하거나 전송하는데 많은 시간이 소요된다. 둘째, 비디오는 공간과 시간의 두 가지 차원을 동시에 가지므로 지연(delay)에 민감하고 처리 시간의 제약을 받는다. 마지막으로 비디오는 단순히 연속된 프레임들의 배열로 구조화된 형태를 가지지 못하므로 정보를 효과적으로 관리하는데 어려움이 있다.

이러한 비디오의 문제점을 해결하기 위해서 기존의 비디오를 구조화된 형태의 비디오로 재구성하는 비디오 구조화(video structuring) 과정이 필요하며 이를 이용한 비디오 브라우징(browsing), 검색(retrieval) 및 편집(editing) 시스템이 많이 개발되어졌다.^{[2][3][4]} Devenport^[1]는 비디오 검색을 위해서 장면 단위로 색인어를 부여하는 비디오 주석(annotation) 시스템을 제안하였다. 그러나 색인어간의 연관성이 없으므로 인해서 비디오를 체계적으로 관리하지 못하였다. Zhang^[2]과 Yositaka^[3] 등은 내용-기반(content-based)의 비디오 인덱싱을 이용하여 비디오 브라우징 및 비디오 검색을 효과적으로 수행하는 방법을 제안하였고, Weiss^[4]는 비디오 대수 연산(video algebra operation)을 이용한 비디오 검색 및 편집 방법을 제안하였다. 그러나 비디오를 단순한 시간적인 연속 매체(linear temporal medium)로 취급하였고,^[4] 비디오를 프레임 단위로만 구조화하였으므로^{[2][3]} 비디오의 논리적인 구조나 비디오 단편들(video segments) 사이의 상호연관성, 계층적 관계 등의 이점을 충분히 활용하지 못하였다.

본 논문에서는 비디오를 체계적으로 관리하고 비디오 브라우징, 검색, 편집 등의 비디오 응용 분야에 효과적으로 적용할 수 있도록 기존의 비디오를 구조화된 형태의 논리적 비디오로 재구성한 구조화된 논리적 비디오(structured logical video)를 제안하였다. 구조화된 논리적 비디오는 비디오의 단편들 사이의

상호 연관성에 따른 계층적 구조로 구성이 되며 각 비디오 단편은 배경, 물체, 상황 등의 장면 구성 정보와 특수 효과, 카메라 효과 등의 장면 특성 정보를 포함한다. 즉, 제안한 구조화된 비디오는 비디오의 물리적 정보뿐만 아니라 비디오의 논리적 정보를 포함하며 이러한 정보를 이용하여 기존의 비디오가 제공하지 못하는 비디오의 논리적 단위 편집 등의 다양한 기능을 제공할 수 있고, 비디오 브라우징, 검색, 비디오 편집 등의 여러 비디오 응용 분야에 효과적으로 적용할 수 있으므로 비디오의 재사용성을 높일 수 있다.

본 논문에서는 기존의 비디오를 구조화된 논리적 비디오로 재구성하여 관리하는 비디오 데이터베이스 관리 시스템과 이를 이용하여 비디오 브라우징 및 검색을 효과적으로 수행하는 비디오 응용 시스템을 설계 및 구현하였다. 2장에서는 기존의 비디오 구조화 방법을 설명하였고, 3장에서는 제안한 구조화된 논리적 비디오의 구조 및 구조화 방법을 설명하였다. 4장에서는 제안한 구조화된 비디오를 재구성 및 관리하는 비디오 데이터베이스 관리 시스템에 대해 설명하였고, 5장에서는 이것을 이용한 비디오 응용 시스템으로 비디오 브라우징 시스템과 비디오 검색 시스템에 대해 설명하였다. 마지막으로 6장에서는 본 논문의 결론을 보였다.

2. 기존의 비디오 구조화 모델

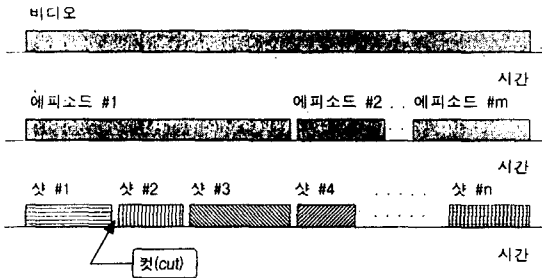
2.1. 용어 정의

비디오에서 장면의 전환이 이루어지는 부분을 컷(cut)이라고 하고 컷으로 구분된 작은 비디오 단위를 샷(shot), 장면(scene), 또는 비디오 단편(video segment)이라고 한다. 즉, 샷은 장면 전환이 없는 비디오 단위를 의미하며 전체 비디오는 여러 개의 연속된 샷으로 구성된다. 그리고 비디오를 샷으로 구분하는 작업을 비디오 단편화(video segmentation)라고 한다. 에피소드(episode)는 하나의 논리적 의미를 가지는 비디오 단위로서 비디오의 논리적인 내용이 같은 연속된 샷으로 이루어진다.

따라서 구조화된 비디오는 그림 1에서와 같이 내용 전환으로 구분되는 연속된 에피소드로 구성되고 각 에피소드는 장면 전환의 단위인 샷으로 구성된다.

그리고 비디오 단편화를 위해 장면의 전환점인 컷

을 검출하는 작업을 컷 검출(cut detection)이라고 한다.



(그림 1) 구조화된 비디오 데이터
(Fig. 1) Structured video data

2.2. 비디오 구조화

비디오 구조화(video structuring)란 비디오를 효과적으로 관리하기 위해 순차적으로 저장된 비디오 데이터를 구조화된 형태의 비디오 데이터로 재구성하는 작업을 말한다. 즉, 비디오를 장면 단위로 구분하

고 각 장면의 특징을 인덱싱하여 비디오 데이터를 재구성함으로써 비디오 검색, 편집 등의 기능을 빠르고 효과적으로 수행할 수 있다.^[2] 이렇게 비디오 구조화에 의해 재구성된 비디오를 구조화된 비디오(structured video)라고 한다.^[5]

비디오 구조화의 전체적인 구성은 그림 2에서와 같이 비디오 단편화와 비디오 인덱싱의 2단계로 이루어진다.^[5]

2.3. 비디오 단편화

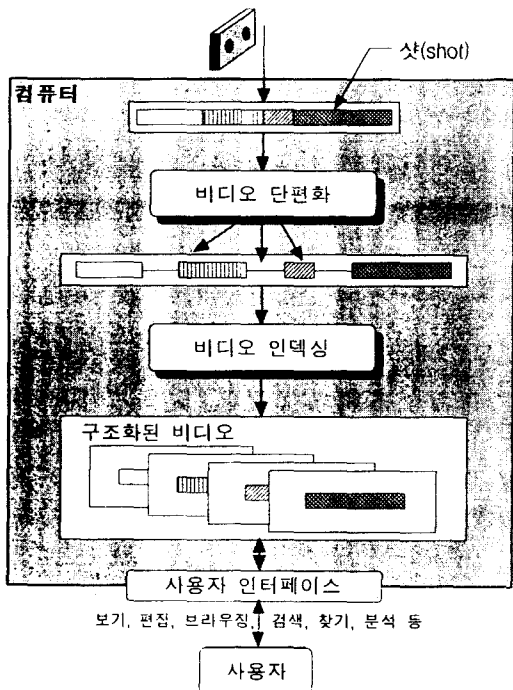
비디오 단편화는 입력 비디오에서 장면의 전환점인 컷을 추출하여 비디오를 장면 단위인 샷 단위로 구분한다. 비디오는 연속된 프레임의 집합이므로 연속된 장면에서는 인접한 프레임 사이의 유사성이 강하고 장면의 전환이 이루어지는 부분에서는 프레임 사이의 유사성이 상대적으로 약하다. 그러므로 컷을 추출하기 위해서는 특정 비디오 요소의 프레임간 차이를 이용하여 그 요소의 연속성을 계산하고 불연속 지점을 컷으로 정의한다. 이러한 비디오 요소로는 프레임의 밝기^[6], RGB 색상^[6], 또는 움직임 벡터(motion vector)^[7] 등이 이용된다.

2.3.1. 프레임간 밝기의 차이를 이용한 비디오 단편화^[6]

비디오는 보통 초당 15에서 30 프레임이 화면상에 출력되므로 장면의 전환이 없는 경우 인접한 프레임 사이의 밝기의 변화 정도가 상대적으로 작다. 그러므로 비디오 스트림(stream)에서 각 프레임간 밝기의 차이를 계산하고 그 값을 특정 임계값(threshold value)과 비교하여 임계값보다 클 경우 두 프레임을 컷으로 구분한다. 즉, 시간 t 인 프레임의 (x, y) 위치에 있는 픽셀의 밝기를 $I(x, y, t)$ 라고 했을 때 프레임간 밝기의 차이 $\Delta I(t)$ 는

$$\Delta I(t) = \frac{1}{N \times M} \sum_{x=1}^N \sum_{y=1}^M |I(x, y, t) - I(x, y, t+1)| \quad (1)$$

와 같이 정의된다. 여기서 N 과 M 은 각각 프레임의 가로 크기와 세로 크기이다. 프레임간 밝기의 차이 $\Delta I(t)$ 는 장면이 전환되는 위치에서 상대적으로 큰 값을 가지므로 $\Delta I(t)$ 가 특정 임계값 이상일 때의 시간을 컷의 위치로 정의한다.



(그림 2) 비디오 구조화 모델
(Fig. 2) Video structuring model

2.3.2. 움직임 벡터를 이용한 비디오 단편화^[8]

비디오에서 장면이 전환되는 부분에서는 두 프레임 사이의 움직임 벡터의 상관도(correlation)가 연속된 장면의 프레임에 비해 상대적으로 약하다. 그러므로 프레임간 움직임 벡터를 이용한 상관도를 계산하여 이 값이 특정 임계값 이하일 경우 그 지점을 컷으로 정의한다.

움직임 벡터의 상관도를 계산하기 위해 먼저 비디오의 각 프레임에서의 움직임 벡터는 블록 정합 알고리즘(BMA: block matching algorithm)을 이용하여 계산한다. 비디오 데이터를 $I(r, t)$ 라고 나타낼 수 있는데, 여기서 r 은 공간 좌표인 (x, y) 를 의미하며 t 는 시간을 의미한다. 이때 움직임 벡터 $m(r, t)$ 와 인접 프레임간의 관계식은 다음과 같다.

$$I(r, t + 1) = I(r + m(r, t), t) \tag{2}$$

이때 움직임 벡터를 계산하기 위해서 먼저 비디오를 $m \times m$ 의 크기를 가지는 중첩되지 않는 블록 b 로 나눈 다음에 각 블록에 대해서 미리 정의된 정합 평가자를 최소화하는 움직임 벡터를 찾는다. 정합 평가자는 다음과 같은 거듭 제곱 평균 오차를 이용한다.

$$\sum_{r \in b} \{I(r + m(r, t), t + 1), I(r, t)\}^2 \tag{3}$$

움직임 벡터를 이용하여 컷을 검출하기 위해서 인접 프레임간 움직임 벡터 사이의 상관도를 의미하는 평균 프레임간 상관도 계수(average inter-frame correlation coefficient)를 이용한다. 프레임내의 모든 블록 b 에 대해서 평균 프레임간 상관도 계수는 아래 식(4)와 같이 정의된다.

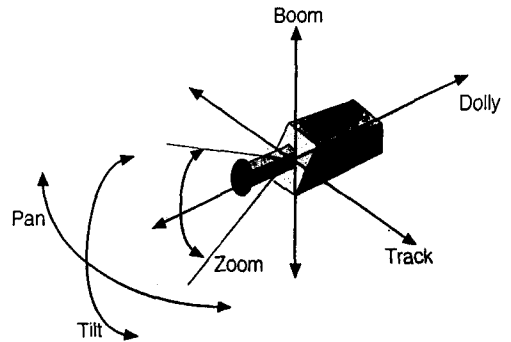
$$\frac{\sum [I(r + m(r, t), t + 1) - \bar{I}(r + m(r, t), t + 1)][I(r, t) - \bar{I}(r, t)]}{\sqrt{\sum [I(r, m(r, t), t + 1) - \bar{I}(r, m(r, t), t + 1)]^2 \sum [I(r, t) - \bar{I}(r, t)]^2}} \tag{4}$$

여기서 $\bar{I}(r, t)$ 는 $I(r, t)$ 의 평균값을 의미한다. 식(4)는 프레임간 움직임 벡터의 유사성을 나타내는데 컷의 위치에서는 상대적으로 작은 값을 가진다. 따라서 이 값을 임계값과 비교하여 임계값보다 작은 값을 가지는 시간 t 를 컷의 위치로 정의한다.

2.4. 비디오 인덱싱

비디오 인덱싱은 비디오 단편화에 의해 구분되어진 비디오에서 샷 단위로 비디오의 특징을 추출하여 인덱싱함으로써 구조화된 비디오를 구성한다. 이때 추출되는 비디오의 특징을 비디오 인덱스(video index)라고 한다.^[5] 이러한 비디오 인덱스는 비디오 검색, 장면 찾기 등의 사용자와의 상호 작용을 지원한다. 비디오 인덱스는 카메라 효과^{[5][6]}와 대표 색상^[5]이 이용된다.

카메라 효과는 카메라의 7가지 기본 동작을 규정한다. 카메라의 기본 동작은 그림 3에서와 같이 ① 고정(fixed), ② 패닝(panning):수평 방향 회전, ③ 트래킹(tracking):수평 방향 횡단, ④ 틸팅(tilting):수직 방향 회전, ⑤ 부밍(booming):수직 방향 횡단, ⑥ 주밍(zooming):초점거리 변경, ⑦ 돌링(Dolling):수평 방향 전후진이다.^{[5][7]}

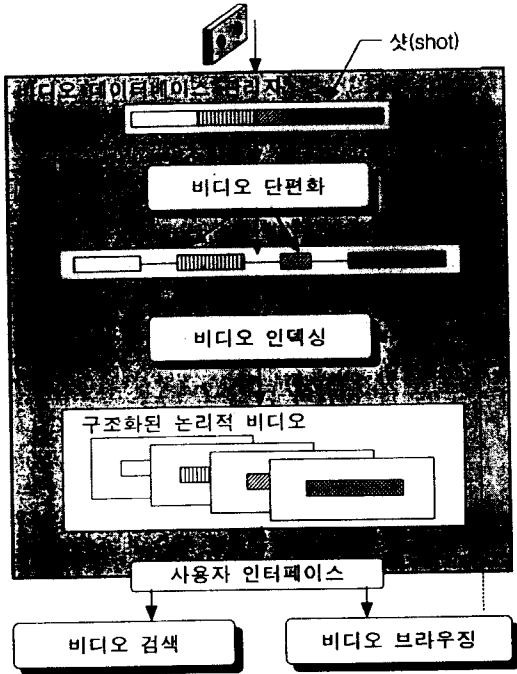


(그림 3) 카메라의 기본 동작
(Fig. 3) Basic operation of camera

대표 색상은 비디오에서 샷의 내용을 반영하는데 이용된다. 원경(遠景) 샷에서는 대표 색상이 배경의 색상을 반영하며, 근경(近景) 샷에서는 대표 색상이 물체의 색상을 반영한다. 이러한 대표 색상은 보통 한 샷 내에서 한정되므로 샷을 구분하는데 사용된다.^[5]

3. 구조화된 논리적 비디오

본 논문에서 제안한 구조화된 논리적 비디오는 기존의 구조화된 비디오에 논리적 정보를 포함한 비디오이다.

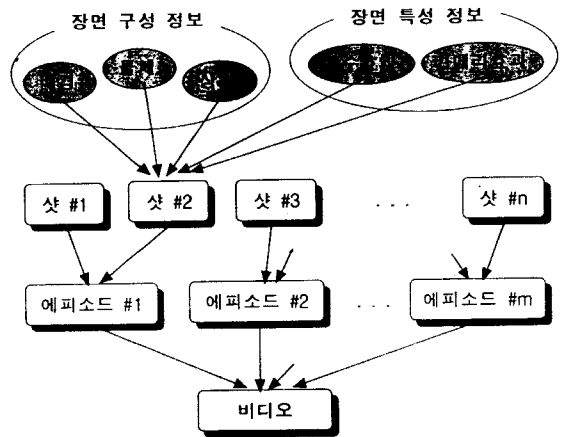


(그림 4) 제안한 비디오 구조화 모델
(Fig. 4) Proposed video structuring model

구조화된 논리적 비디오는 그림 4에서와 같이 비디오 구조화에 의해서 재구성되며 비디오 구조화는 비디오 단편화와 비디오 인덱싱의 2 단계로 이루어진다. 입력된 비디오는 비디오 단편화에 의해서 장면 단위의 샷으로 구분되고 비디오 인덱싱 부분에서는 각 샷의 물리적 정보 및 논리적 정보를 추출하여 샷 단위로 인덱싱한다. 샷의 물리적인 정보는 카메라 효과와 장면의 특수 효과로 구성되며 샷의 논리적인 정보는 장면의 배경, 물체, 상황으로 이루어진다. 그리고 논리적인 내용상의 연관성이 있는 연속된 샷들을 연결하여 에피소드를 구성한다. 즉, 비디오는 내용상의 구분인 에피소드 단위와 장면 구분인 샷 단위의 계층적 구조로 재구성된다. (그림 5)

기존의 구조화된 비디오 모델은 비디오의 특징을 프레임 단위로 인덱싱하고⁶⁾, 장면의 물리적 정보만을 인덱싱함으로써⁷⁾ 비디오의 내용 전개 등과 같은 비디오의 논리적 특성을 관리하는데 한계가 있었다. 본 논문에서 제안한 구조화된 논리적 비디오 모델은 비디오의 특징을 규정하는 비디오 인덱스로 장면의

특수 효과⁶⁾나 카메라 효과⁵⁾ 등의 물리적 정보뿐만 아니라 장면의 논리적 특성을 나타내는 장면의 배경, 움직이는 물체, 상황 등의 정보를 이용하므로 비디오의 논리적 표현이 가능하다. 그리고 비디오의 장면 전환의 단위인 샷 단위뿐만 아니라 비디오의 논리적 구분 단위인 에피소드 단위로 비디오를 재구성하여 관리하므로 비디오 브라우징, 검색, 편집 등의 비디오 응용 분야에서 그 기능이 비디오의 프레임 단위나 장면 단위로 제한되지 않으며 비디오의 내용 전개상의 논리적 단위로 확장할 수 있다.

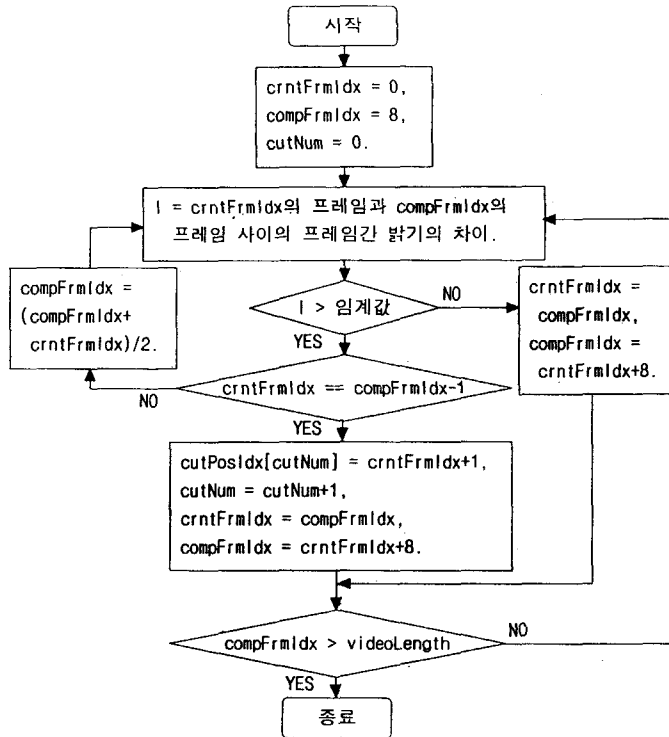


(그림 5) 구조화된 논리적 비디오의 구조
(Fig. 5) The structure of structured logical video

3.1. 비디오 단편화

비디오 단편화는 기존의 프레임간 밝기의 차이를 이용하는 방법을 이용하였다.⁶⁾ 그러나 전체 비디오 스트림의 프레임간 밝기의 차이를 구하기 위해서는 많은 시간이 소요된다. 그러므로 제안한 비디오 구조화의 비디오 단편화 부분에서는 비디오 스트림의 모든 프레임간 밝기의 차이를 계산하는 것이 아니라 빠른 비교를 이용하여 계산량과 계산 시간을 개선하였다. 본 논문에서 사용된 알고리즘은 그림 6에서와 같다.

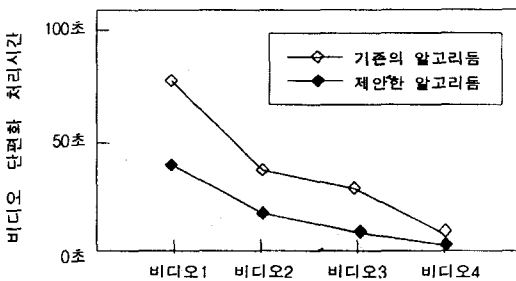
제안한 비디오 단편화 알고리즘과 기존의 프레임간 밝기의 차이를 이용한 비디오 단편화 알고리즘⁶⁾을 펜티엄 120MHz PC에서 실험한 결과는 표 1과 그림 7과 같다. 비디오 단편화에 의해 구분된 각 샷들은 비디오 브라우징, 검색, 편집 등의 비디오 응용 분



(그림 6) 제안한 비디오 단편화 알고리즘
(Fig. 6) Proposed video segmentation algorithm

<표 1> 비디오 단편화 처리 시간
(Table 1) The processing time in video segmentation

| | 기존의 비디오 단편화 알고리즘 | 제안한 비디오 단편화 알고리즘 |
|----------------|------------------|------------------|
| 비디오 1(816 프레임) | 73.00초 | 38.62초 |
| 비디오 2(442 프레임) | 36.09초 | 15.27초 |
| 비디오 3(306 프레임) | 26.80초 | 10.55초 |
| 비디오 4(105 프레임) | 9.00초 | 5.71초 |



(그림 7) 비디오 단편화 처리 시간
(Fig. 7) The processing time in video segmentation

야에 기본 단위로 이용된다.

3.2. 비디오 인덱싱

비디오 인덱싱은 비디오, 에피소드, 샷 단위로 정보를 인덱싱한다. 각각의 비디오 인덱스는 문자열로 기록하며 비디오의 물리적 특성뿐 아니라 논리적 특성을 포함한다. 비디오, 에피소드, 샷 단위별로 각각의 비디오 인덱스를 살펴보면 표 2와 같다.

비디오는 비디오 제목과, 비디오 설명, 상영 시간의 비디오 인덱스로 인덱싱된다. 비디오 제목과 비디오

<표 2> 비디오 인덱스
(Table 2) Video indices

| | 비디오 인덱스 |
|------|--------------------------------|
| 비디오 | 비디오 제목, 비디오 설명, 상영 시간. |
| 에피소드 | 에피소드 설명, 대표 샷 ID. |
| 샷 | 샷 ID, 샷 설명, 장면 구성 정보, 장면 특성 정보 |

설명은 문자열로 기록하며 상영 시간은 비디오의 전체 상영 시간을 나타낸다. 에피소드는 에피소드의 설명과 대표 샷 정보로 인덱싱된다. 대표 샷 정보란 에피소드의 내용을 가장 잘 나타내는 샷의 ID를 의미하며 에피소드의 첫 번째 샷 또는 사용자의 선택에 의해 정의된다. 마지막으로 샷의 비디오 인덱스는 샷 ID, 샷 설명, 장면 구성 정보, 장면 특성 정보로 구성된다. 샷의 ID는 샷을 구분하는 고유 번호이고 샷 설명은 샷의 특징을 나타내는 부분이다. 장면 구성 정보는 장면을 구성하는 배경, 물체 그리고 상황으로 표현된다. 특히, 장면 구성 정보는 항목 상호간의 연관성을 이용하여 계층적 구조로 관리되므로 이를 이용해서 비디오 검색을 효과적으로 수행 할 수 있다. (표 3) 장면 특성 정보는 페이드-인(fade-in), 페이드-아웃(fade-out) 등의 비디오 편집상의 특수 효과와 주밍(zooming), 패닝(panning) 등의 카메라 효과로 구성된다.

위에서 열거한 비디오 인덱스들은 비디오 검색 시에 질의어로 이용된다.

4. 비디오 데이터베이스 관리 시스템

본 논문에서 제안한 비디오 데이터베이스 관리 시스템은 입력된 비디오를 비디오 구조화에 의해 구조화된 논리적 비디오로 재구성하여 관리한다.

비디오 데이터베이스 관리 시스템은 그림 5에서와 같이 비디오 단편화에 의해 장면 단위로 구분된 비디오의 각 샷과 비디오 데이터베이스 관리 시스템에서 정의한 비디오 인덱스를 연결하여 각 샷의 비디오 인덱스 정보를 정의하고 비디오 브라우징 및 비디오 검색 시스템에서 이러한 정보를 활용할 수 있도록 관리한다.

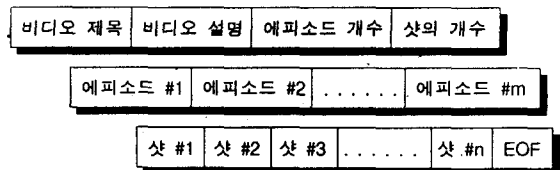
비디오 데이터베이스 관리 시스템에서 정의하는 비디오 인덱스는 샷 단위의 비디오 인덱스인 장면 구성 정보와 장면 특성 정보이다. 장면 구성 정보는 장면의 논리적 정보를 포함하며 배경, 물체, 상황으로 구성되며 이들 각 항목간의 상호 연관성을 이용하여 계층적 구조로 관리된다. 장면 특성 정보는 장면의 물리적 특성을 나타내며 특수 효과, 카메라 효과로 구성된다. 표 3은 비디오 데이터베이스 관리 시스템에서 정의하는 비디오 인덱스의 예를 나타낸다. 표 3

에서처럼 장면 구성 정보는 항목의 연관성을 이용하여 계층적 구조로 관리되는데 점(.)의 왼쪽 항목이 상위 계층이고 오른쪽 항목이 하위 계층을 나타낸다.

〈표 3〉 비디오 인덱스의 예
 〈Table 3〉 Example of video indices

| No. | 장면 구성 정보(배경) | 장면 특성 정보(카메라 효과) |
|-----|--------------|------------------|
| 1 | 학교 | Fixed |
| 2 | 학교. 강의실 | Panning |
| 3 | 학교. 도서관 | Tracking |
| 4 | 학교. 실험실 | Tilting |
| 5 | 학교. 운동장 | Booming |
| 6 | 학교. 운동장. 축구장 | Zooming |
| 7 | 학교. 운동장. 농구장 | Dolling |

비디오 데이터베이스 관리 시스템은 비디오를 논리적 단위로 구조화하여 재구성하고 구조화된 비디오에 대한 정보를 저장한다. 정보의 저장은 비디오 단위로 이루어지며 저장되는 구조는 그림 8에서처럼 비디오, 에피소드, 샷 단위로 정보를 저장한다.



(그림 8) 비디오 데이터베이스의 저장 구조
 (Fig. 8) The file structure of video database

비디오 단위의 정보는 비디오 제목, 설명, 비디오를 구성하는 에피소드의 개수, 샷의 개수로 구성되며 이러한 정보는 비디오 단위의 브라우징 및 검색 등에 이용된다. 에피소드 단위의 정보는 에피소드 설명, 에피소드를 구성하는 샷의 개수 및 위치 정보, 대표 샷 ID로 구성되며 이들 정보는 에피소드 단위의 브라우징 및 검색 등에 이용된다. 마지막으로 샷 단위의 정보는 샷의 위치 정보, 샷의 ID, 샷의 설명, 장면 구성 정보와 장면 특성 정보로 구성이 되며 이들 정보는 샷 단위의 비디오 브라우징 및 검색 등에 이용된다.

본 논문에서 제안한 비디오 데이터베이스 관리 시스템은 그림 5에서와 같은 구조로 비디오, 에피소드, 샷 단위의 관리를 객체 지향 데이터베이스 모델을 이용하여 설계하였다. 아래의 리스트는 샷, 에피소드, 비디오의 정보를 정의 및 관리하는 C++ 타입의 클래스이다.

```

class CVideo // 비디오의 정보를 정의하는 클래스
{
private:
    char* m_szTitle, m_szDescription; // 비디오 제목, 설명
    int m_nEpisodeNum; // 에피소드의 개수
    CEpisode m_Episode[DEF_MAX_NUM]; // 에피소드의 정보
    int m_nShotNum; // 샷의 개수
    CShot m_Shot[DEF_MAX_NUM]; // 샷의 정보
};

class CEpisode // 에피소드의 정보를 정의하는 클래스
{
private:
    char* m_szDescription; // 에피소드의 설명
    int m_nShotNum; // 포함되는 샷의 개수
    CShot m_Shot[DEF_MAX_NUM]; // 포함되는 샷의 정보
    int m_nRepShotID; // 대표 샷 ID
};

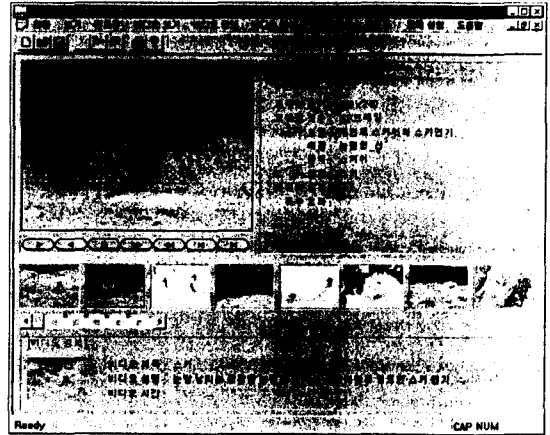
class CShot // 샷의 정보를 정의하는 클래스
{
private:
    long m_nStartFrameIdx, m_nEndFrameIdx; // 샷의 위치
    int ID; // 샷의 ID
    char* m_szDescription; // 샷의 설명
    int m_nBackIdx, m_nObjIdx, m_nSitIdx; // 장면 구성 정보 - 배경, 물체, 상황
    int m_nEffIdx, m_nCmrIdx; // 장면 특성 정보 - 특수 효과, 카메라 효과
};
    
```

5. 비디오 응용 시스템

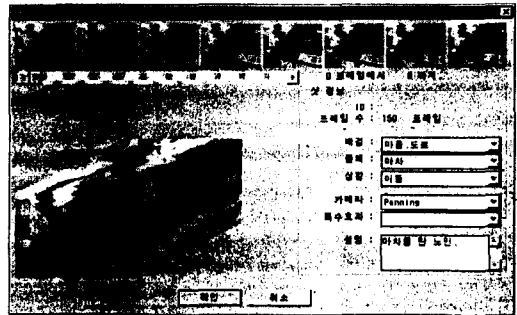
5.1. 비디오 브라우징 시스템

비디오 브라우징은 비디오의 전체 내용을 요약하여 사용자에게 보여줌으로써 사용자가 비디오의 내용을 빠르게 확인할 수 있도록 한다.^[25]

본 논문에서 제안한 비디오 브라우징 시스템은 먼저 비디오 단편화에 의해 구분된 비디오의 각 샷으로부터 샷의 내용을 대표하는 프레임을 선택하여 대표 프레임으로 정의한다. 대표 프레임은 샷의 첫 번째 프레임 또는 사용자가 지정한 특정 프레임으로 정의



(그림 9) 비디오 브라우징 시스템의 사용자 인터페이스
(Fig. 9) The user interface of video browsing system



(그림 10) 선택된 샷 정보 편집
(Fig. 10) Editing of selected shot information

된다. 비디오 브라우징 시스템은 정의된 대표 프레임들을 시간 순서로 화면에 나열함으로써 사용자는 대략적인 비디오의 내용을 확인할 수 있다.(그림 9)

사용자는 원하는 샷의 대표 프레임 아이콘을 선택하여 각 샷의 정보를 확인하거나 편집할 수 있다. 이 과정에서 비디오 브라우징 시스템은 프레임 보기(frame viewer)와 비디오 스트리머(video streamer)^[8] 기능을 제공하여 사용자가 샷의 내용을 빠르게 파악할 수 있도록 한다.(그림 10)

그리고 본 논문에서 제안한 비디오 브라우징 시스템은 에피소드 단위로 비디오를 구성할 수 있는 환경을 제공한다. 그러므로 사용자는 비디오를 논리적인 단위로 구성할 수 있으며 이것을 비디오 검색이나 논

리적 비디오 편집 등에 이용할 수 있다. 그리고 비디오 이야기 문서(video story document) 등을 편집할 수 있고 이것을 이용하여 영화의 예고편이나 교육용 비디오 등을 작성할 수 있다. 본 논문에서 제안한 비디오 브라우징 시스템은 기존의 VCR에서 제공하는 비디오 플레이, 빨리 보기, 역순 플레이 등의 기능도 포함하고 있다.

5.2. 비디오 검색 시스템

비디오 검색은 전체 비디오 중에서 사용자가 요구하는 장면들만을 빠르게 찾아서 사용자에게 보여 주는 것이다.^{[2][3][5]}

본 논문에서 제안한 비디오 검색은 비디오의 샷 단위와 에피소드 단위로 이루어진다. 비디오 검색은 사용자의 질의(query)에 의해서 이루어지는데 질의는 비디오의 논리적 정보에 해당하는 설명, 장면 구성 정보와 비디오의 물리적 특성을 나타내는 장면 특성 정보, 비디오 상영 시간 등 4가지 부분으로 구성된다. 그리고 각각의 질의는 부울(bool) 연산으로 연결되어 질 수 있다.

비디오의 장면 구성 정보의 각 항목은 계층적인 구조로 구성되므로 상위 항목을 검색 시에 하위 항목까지 자동적으로 검색이 가능하다. 즉, 사용자가 보기를 원하는 장면의 정확한 정보를 알지 못하더라도 추상적인 개념의 상위 계층을 검색함으로써 원하는 장면의 정보를 쉽게 접근할 수 있다. 예를 들면, 표 3의 장면 구성 정보의 예에서처럼 공간적인 배경의 항목이 구성되어 있을 때 사용자는 학교를 배경으로 가지는

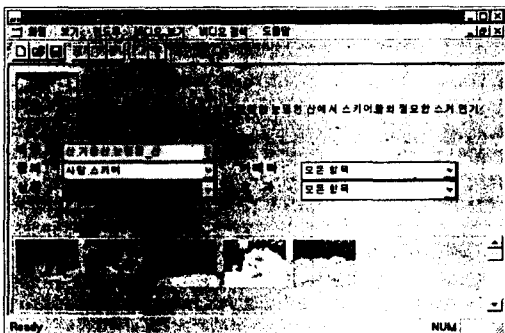
장면을 검색하면 학교 항목의 하위 계층인 강의실, 도서관, 실험실, 운동장을 배경으로 가지는 장면까지도 자동적으로 검색이 가능하다.

비디오의 검색 시스템은 사용자로부터 질의를 입력받아서 비디오 데이터베이스 관리 시스템에 질의를 전달하고 검색을 요청하게 된다. 비디오 데이터베이스 관리 시스템은 전달된 질의어의 항목 즉, 설명, 장면 구성 정보, 장면 특성 정보 및 비디오의 상영 시간을 조사하여 해당하는 샷의 ID와 개수 또는 에피소드의 위치 인덱스와 개수를 비디오 검색 시스템에 전달한다. 비디오 검색 시스템은 전달된 정보를 이용하여 사용자에게 원하는 샷 또는 에피소드의 대표 프레임을 출력한다.

검색의 결과로는 질의에 해당하는 샷이나 에피소드의 대표 프레임이 화면상에 배열되고 사용자는 검색된 내용을 다시 비디오 브라우징 시스템을 통해서 브라우징 할 수 있다. (그림 11)

6. 결 론

본 논문에서는 비디오를 체계적으로 관리하고 비디오 브라우징, 검색 편집 등의 비디오 응용 분야에 효과적으로 적용할 수 있도록 기존의 비디오를 구조화된 형태의 논리적 비디오로 재구성한 구조화된 논리적 비디오를 제안하였다. 구조화된 논리적 비디오는 비디오 단편들 사이의 상호 연관성에 따른 계층적 구조를 가지며 각 비디오 단편은 배경, 물체, 상황 등의 장면 구성 정보와 특수 효과, 카메라 효과 등의 장면 특성 정보를 포함한다. 즉, 구조화된 비디오는 비디오의 물리적 정보뿐 아니라 논리적 정보를 포함하며 이렇게 재구성된 비디오는 기존의 비디오로 제공하지 못하는 다양한 기능을 제공할 수 있고, 비디오의 재사용성을 높일 수 있다. 그리고 본 논문에서는 기존의 비디오를 구조화된 논리적 비디오로 재구성하여 관리하는 비디오 데이터베이스 관리 시스템과 구조화된 논리적 비디오를 이용하여 비디오 브라우징과 검색을 효과적으로 수행하는 시스템을 설계 및 구현하였다. 비디오 데이터베이스 관리 시스템은 비디오 단편들 사이의 시간적인 관계와 상호 연관성을 이용하여 비디오를 계층적 구조로 관리하며 비디오, 에피소드, 샷 단위의 관리를 객체 지향 데이터베이스



(그림 11) 비디오 검색의 결과
(Fig. 11) The result of video retrieval

모델로 설계하였다.

본 논문에서 제안한 구조화된 논리적 비디오를 이용한 비디오 브라우징 및 검색 시스템은 오디오나 문자가 포함된 비디오 문서의 작성 및 편집, 교육, VOD, CATV, 광고 제작, 멀티미디어 매일 등의 여러 비디오 응용 분야에 이용될 수 있다.

참고 문헌

- [1] W. E. Mackay and G. Devenport, "Virtual Video Editing in Interactive Multimedia Applications," *Communications of ACM*, vol. 32, no. 7, pp. 802-810, July, 1989.
- [2] S. W. Smoliar and H. J. Zhang, "Content-Based Video Indexing and Retrieval," *IEEE Multimedia*, vol. 1, no. 2, pp. 62-72, Summer 1994.
- [3] A. Yoshitaka, S. Kishida, M. Hirakawa, and T. Ichikawa, "Knowledge-Assisted Content-Based Retrieval for Multimedia Databases," *IEEE Multimedia*, vol. 1, no. 4, pp. 12-21, Winter 1994.
- [4] R. Weiss, A. Duda, and D. K. Gifford, "Composition and Search with a Video Algebra," *IEEE Multimedia*, vol. 2, no. 1, pp. 12-25, Spring 1995.
- [5] Y. Tonomura, A. Akutsu, Y. Taniguchi, and G. Suzuki, "Structured Video Computing," *IEEE Multimedia*, vol. 1, no. 3, pp. 34-43, Fall 1994.
- [6] H. J. Zhang, A. Krukanhali, and S. W. Smoliar, "Automated Temporal Segmentation of Video Sequences," *IEEE Visual Signal Processing and Communications*, 21-22 September, pp. 191-194, 1993.
- [7] A. Akusu, Y. Tonomura, H. Hashimoto, and Y. Ohba, "Video Indexing using Motion Vectors," *SPIE Communications and Image Processing*, vol. 1818, pp. 1524-1530, 1992.
- [8] H. J. Zhang, A. Krukanhali, and S. W. Smoliar, "Automatic Partitioning of Full-Motion Video," *ACM Multimedia Systems*, ACM Press, New York, vol. 1, pp. 10-28, 1993.



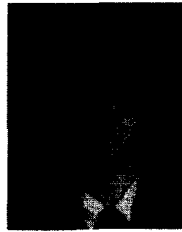
권성복

1994년 경북대학교 전자공학과 졸업(공학사)
 1996년 경북대학교 대학원 전자공학과(공학석사)
 1996년~현재: 경북대학교 대학원 전자공학과(박사과정)
 관심분야: 화상처리, 멀티미디어 시스템, 멀티미디어 검색 기술 등



조영우

1992년 경북대학교 전자과 졸업(학사)
 1994년 경북대학교 대학원 전자과 졸업(공학석사)
 현재: 경북대학교 전자과 박사과정 재학중
 주관심분야: 멀티미디어 시스템, 영상 처리



김영모

1980년 경북대학교 전자과 졸업(학사)
 1983년 KAIST 대학원 졸업(공학석사)
 1989년 KAIST 대학원 졸업(공학박사)
 현재: 경북대학교 전자과 부교수
 주관심분야: 컴퓨터 그래픽스, 영상 처리