

비디오 검색과 시각적 요약을 위한 장면 기반 계층적 브라우징 기법

(A Scene-based Tree Browsing Technique for Video Retrieval and Visual Summary)

임 동 혁 [†] 이 석 룡 ^{**} 정 진 완 ^{***}

(Tong Hyeok Lim) (Seok-Lyong Lee) (Chin-Wan Chung)

요약 디지털 비디오의 사용이 일반화되어감에 따라 비디오 브라우징에 대한 연구가 더욱 요구되어지고 있다. 과거에 연구된 바 있는 VCR식 순차탐색기법은 아날로그 비디오 브라우징에서 사용되던 고전적 방식을 디지털 비디오에 적용한 것이며, 키프레임 정적표현법은 비디오를 구성하는 프레임들을 보이는 방식이다. 이 두 방식이 디지털 비디오의 특성을 충분히 사용하지 못하기 때문에, 최근에는 비디오 샷간의 계층적 관계를 기반으로 하는 계층적 브라우징과 장면 간의 관계를 기반으로 하는 장면 기반 브라우징이 주목을 받고 있다. 본 논문에서는 기존의 비디오 브라우징에 대한 연구들을 자세히 알아보고, 계층적 브라우징과 장면 기반 브라우징의 장점을 합한 ST 브라우징 기법을 제안하고 구현하였다. ST 브라우징은 장면 기반 브라우징에서처럼 비디오 내의 각 장면에 바로 접근할 수 있을 뿐만 아니라, 계층적 브라우징처럼 비디오의 전체구조를 알기 쉽게 보여준다. 또한 브라우징의 결과는 시각적 요약으로 사용될 수 있다.

Abstract Video browsing, which enables to retrieve certain scenes in a video data stream, is becoming important as the digital video becomes widely used. The VCR-like sequential scanning, which is a traditional video browsing scheme, and the static presentation of keyframes have been studied in the past. Recently, the hierarchical video browsing based on the temporal relationship among video shots and the scene-based browsing have been actively studied. In this paper, we first investigate the existing browsing techniques in detail. Then, we propose and implement the ST browsing scheme which is a hybrid of the hierarchical and the scene-based video browsing, to retrieve directly a certain scene in video clips, and at the same time to see the overall shot flow like the hierarchical browsing. In addition, the result of browsing can be stored as a visual summary.

1. 서론

아날로그 비디오에 비해 다양한 장점을 가진 디지털 비디오에서도, 여러 비디오들 중 원하는 비디오를 찾는 다거나 같은 비디오 내에서 원하는 부분을 찾는 일은 (1) 비디오에 내재되어 있는 메타데이터를 다른 형태의 매체로 변환하기 어렵다는 점과 (2) 비디오를 탐색하여

원하는 부분을 찾는 일이 처리시간이 높은 작업이라는 점에서 여전히 어렵다. 그러나, 이러한 부담에도 불구하고, 비디오에 대한 검색시스템이나 효율적인 브라우징 기법에 대한 요구는 점차 높아져 가고 있다.

비디오 내의 원하는 부분을 찾는 문제에 대한 연구는 크게 두 가지로 나눌 수 있다: 검색(search)과 브라우징(browsing)[1]. 비디오 검색은 사용자가 자신이 찾고자 하는 부분에 대한 분명한 정보를 가지고 있어서 검색용 질의를 구성할 수 있는 경우로, 데이터베이스 질의에 비유될 수 있다. 이에 비해 비디오 브라우징은 사용자 자신도 원하는 부분을 접하기 전까지는 그 부분에 대해 묘사할 수 없는 경우로 흔히 I-know -it-when-I-see-it 형태로 표현되어진다. 따라서 검색시스템이 사용자 질의에 대한 정확한 결과를 돌려주는데 초점을 맞

[†] 학생회원 : 한국과학기술원 전산학과
tong@islab.kaist.ac.kr

^{**} 비 회 원 : 한국과학기술원 전산학과
sillee@islab.kaist.ac.kr

^{***} 중신회원 : 한국과학기술원 전산학과 교수
chungcw@islab.kaist.ac.kr

논문접수 : 2000년 4월 24일

심사완료 : 2001년 2월 15일

추는데 비해, 브라우징 시스템은 사용자가 원하는 방향의 선택 가능한 후보들을 최대한 제공하는데 중점을 두고 있다.

최근까지 연구된 브라우징 기법은 크게 4가지로 나눌 수 있다: VCR식 순차탐색 기법, 키프레임 정적표현법, 계층적 브라우징 기법, 장면 기반 브라우징 기법. VCR식 순차탐색기법은 고전적 비디오 브라우징 기법으로 사용자가 이미 잘 알고 있는 방식으로 친근감이 있다는 장점이 있는 반면, 디지털 비디오의 임의접근이 가능한 특성을 충분히 활용하지 못하며 많은 시간이 소요된다는 단점이 있다. 키프레임 정적표현법은 디지털 비디오를 구성하고 있는 각 샷(Shot)당 하나의 키프레임을 추출하여 사용자에게 보임으로써 사용자가 원하는 샷에서부터 비디오를 플레이할 수 있도록 하는 기법이다. 그러나 디스플레이 모니터의 공간적 제한으로 사용자가 충분한 후보 프레임을 볼 수 없다는 단점을 가지고 있다. 이를 개선한 기법이 계층적 브라우징 기법과 장면 기반 브라우징 기법인데, 이 두 기법에 대해서는 2장에서 자세히 살펴보겠다.

하나의 비디오 내에서 원하는 장면을 찾는 브라우징과는 별도로 다수의 비디오에서 원하는 비디오를 빨리 찾을 수 있도록 돕는 방법도 연구되고 있는데, 그 중의 하나는 메타데이터(비디오의 내용)를 요약하는 것이다. 비디오에 내재된 메타데이터를 표현하는 방법에 대한 연구는 계속 있어 왔으나, 시각적 매체의 내용을 나타내는 가장 좋은 방법은 영화의 예고편과 같은 시각적 요약(시각적 요약)로 표현하는 것[2]이다. 매체 변환에 따른 정보 손실을 막을 수 있고, 메타데이터 분석에 따른 부담을 줄일수 있다는 점에서 시각적 요약의 제작은 가장 효율적인 요약방식이다.

본 연구는 효율적인 비디오 브라우징 기법과 이를 통한 시각적 요약의 작성에 대해 다루고 있다. 제안하는 ST 브라우징 기법은 계층적 브라우징 기법과 장면 기반 브라우징 기법을 하나로 취합하였으며, 브라우징 결과를 시각적 요약으로 저장할 수 있으므로, 비디오의 요약된 메타데이터를 작성할 때도 사용할 수 있다.

2. 관련연구

2.1 디지털 비디오의 구조

비디오 브라우징에서 원시 비디오 데이터(raw video data, RVD)는 프레임(frame)이라 불리는 엔티티(entity)의 집합으로 본다. 따라서 RVD의 한 프레임은 "비디오 필름의 한 컷"이 된다. RVD의 프레임들은 카메라 작동을 판단할 수 있는 함수 λ 에 의해 그룹지워

질 수 있는데, 이를 샷(shot)이라고 한다. 샷은 "비디오 카메라의 녹화 시작점에서 녹화 종료까지를 일컫는 단위"[3]라고 정의할 수 있다. 샷을 구성하는 프레임 중 그 샷을 가장 잘 나타낼 수 있는 프레임을 특별히 키프레임(keyframe) 혹은 대표 프레임(representative frame, r-frame)이라고 하는데, 본 연구에서는 각 샷의 첫 프레임을 키프레임으로 정했다. 샷들을 특정 조건에 따라 하나로 묶는 함수 σ 가 존재한다면, 함수 σ 에 의해 그룹지워진 샷들을 장면(scene)이라 한다. 장면이란 "위치 또는 즐거리 상의 사건에 따라 하나로 묶을 수 있는 비교적 적은 수의 관련된 샷의 모임"을 의미한다.

2.2 계층적 브라우징 기법

계층적 브라우징은 키프레임 정적표현법을 발전시킨 기법이다. 비디오 클립을 구성하고 있는 기본 단위인 샷으로부터 키프레임을 추출하여 사용자에게 표시해주는 키프레임 정적표현법은 전체 키프레임을 한 눈에 살펴볼 수 있다는 장점을 가지고 있다. 비록 각 샷을 가장 잘 표현하는 프레임을 찾아 키프레임으로 하는데는 많은 어려움이 있지만[4], 비디오 클립의 내용 전체를 한 눈에 살펴볼 수 있다는 점은 이 전의 VCR식 순차탐색 기법보다 진보된 기법이라 할 수 있겠다.

그러나, 키프레임 정적표현법은 너무 많은 키프레임이 존재한다는 문제가 있다. 너무 많은 키프레임은 모니터 화면에서 표현이 어렵기 때문에, 전체 키프레임을 화면이나 용지 등의 디스플레이 매체에 모두 보이지 않고, 그것들 중 대표성을 가진 일부만을 표시한 후, 현재 나타나 있는 키프레임으로부터 보이지 않는 키프레임으로 확장할 수 있도록 도와주는 계층적 브라우징 기법[1]이 제시되었다.

계층적 브라우징 기법에서는 전체 키프레임 중 선택된 키프레임의 프레임 아이콘을 시간적인 순서대로 배열하여 보여줌으로써, 사용자가 긴 비디오의 내용을 모두 보지 않더라도 빠른 시간 내에 비디오의 개략을 살펴보는 데 도움을 준다.

그러나, 이러한 장점에도 불구하고 계층적 브라우징은 몇 가지 문제를 가지고 있다. 그 중 하나는 계층적 브라우징이 비디오 클립을 구성하는 샷들의 위치만을 중요시하고 의미를 무시하여, 비효율적인 브라우징을 초래할 수 있다는 점이다. 실제로 대부분의 관객은 자신이 감상한 영화의 중요 부분을 묘사할 때, 전체 영화 중 절대적이거나 상대적인 위치보다는 배경이나 배경음악(사운드)를 자주 언급한다는 점만 보아도, 비디오 클립 전체를 임의의 등분으로 나누어 브라우징하는 방법은 개선의 여지가 있음을 알 수 있다.

2.3 장면 기반 브라우징 기법

비슷한 배경의 샷을 모아 장면을 구성하고, 이를 기반으로 원하는 샷을 찾아나가는 장면 기반 브라우징 기법 [5]이 연구되었다. 계층적 브라우저가 비디오 클립을 몇 등분한 뒤, 해당 위치의 샷으로부터 키프레임을 추출, 사용자에게 선택토록 제공하는데 비해, 장면 기반 브라우저는 비디오 클립을 구성하는 샷들을 몇 개의 장면으로 분할한 후, 각 장면 당 첫 번째 샷으로부터 키프레임을 추출, 사용자가 선택할 수 있도록 제공한다.

비디오 클립을 분석하여 장면을 구분하는 비디오 세그멘테이션 알고리즘에서 주로 사용되는 방법은 비교 대상인 두 샷간의 비슷한 점을 찾는 것이다. 샷 간의 비슷한 점으로는 색, 형태, 이미지간의 관계성 등을 들 수 있다. 이러한 샷간의 속성들을 비교할 때는 주로 키프레임을 사용하게 된다.

비슷한 속성을 가진 샷들을 하나의 그룹으로 모으면 하나의 장면이 될 수 있는데, 이러한 장면들을 사용자에게 보여주면, 사용자는 그 중 자신이 원하는 샷이 속한 장면을 선택하여 브라우징할 수 있다. 장면 기반 브라우징 기법은 키프레임 정적표현법이나 계층적 브라우징 기법에 비해 사용자의 장면 선택이 더욱 명료해지고, 한번에 펼쳐 볼 수 있는 후보(키프레임)의 수가 적어 더욱 효과적이다.

장면 기반 기법 중에서도 장면을 후보로 제공하는데 머물지 않고 장면 간의 순서를 사용자에게 보임으로써 전체적인 흐름을 더욱 알기 쉽도록 제시한 기법 [5]이 특히 주목받고 있다. 이 기법에서는 장면을 나타내는 키프레임 사이에 비디오 클립 내의 상영순서에 따른 화살표 선분을 첨가한 장면 전환 그래프(Scene Transition Graph)를 생성하여 제공한다. 장면 전환 그래프는 방향성 그래프인데, 장면을 구성하는 샷과 다른 장면내의 샷과의 시간적 순서에 따라 장면 간에 화살표 선분을 그리는 것이다.

그러나 장면 전환 그래프는 장면 내의 샷이 연속적이지 않기 때문에 화살표 선분이 너무 많아져서 전체적인 흐름을 알기 어려워졌다. 이는 계층적 브라우징 기법에서의 순서상의 명료함이 오히려 퇴색한 느낌이고, 너무 복잡한 그래픽을 제공하여 사용자에게 혼란을 줄 수 있다. 따라서 본 논문에서는 장면 기반 브라우징을 행하면서도 계층적 브라우징 기법에서처럼 전체적인 흐름을 알기 쉽도록 한 새로운 브라우징 기법을 제안하고자 한다.

3. ST 브라우징 기법

본 논문에서 제안하는 ST 브라우징(Scene-based Tree browsing) 기법은 세그멘테이션 단계에서 얻은 샷 정보와 장면 정보를 토대로, 장면을 재구성하여 구해

진 장면을 노드로 하는 브라우징 트리를 생성한 후, 트리의 운영방식과 마찬가지로 노드의 하위계층을 확장(expand)하거나 축소(reduce)해가며 장면을 브라우징하게 된다(그림 1). 따라서 ST 브라우징 기법에서는 장면을 재구성하는 방법(클러스터링)과 트리의 생성과 운영이 중요하다.

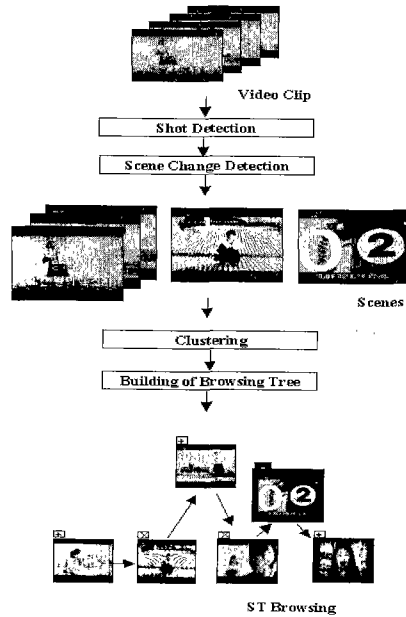


그림 1 ST 브라우징의 순서

3.1 클러스터링

세그멘테이션에서 행해지는 장면 검출은 대체로 각 샷의 컬러 정보에 따라 샷을 그룹화하였다. 따라서 장면을 구성하는 샷이 다양하게 존재할 수 있다. 이러한 장면을 그대로 가져다가 장면 전환 정보를 표시하면, 장면과 장면 사이에 다수의 선분이 발생하여 사용자가 장면 간의 순서나 비디오 클립 내에서의 시간적인 위치를 파악하는데 어렵게 된다. 이를 해결하기 위해서는 장면을 구성하는 샷들 중 연속한 샷의 모임만을 재그룹하여 별도의 장면으로 만들면, 장면 전환 정보를 보여줄 때 하나의 선분만이 남게 된다. 이렇게 장면 내의 연속한 샷만을 추출하여 별도의 장면으로 재구성하는 과정이 클러스터링이다. 본 논문에서는 클러스터링 과정을 통해 얻을 수 있는 연속된 샷의 모임을 장면이라 하고, $[i, j]$ 로 표시한다.

정의 1(RVD) 프레임의 집합 RVD는 임의의 정수 n

에 대한 집합 $\{1, \dots, n\}$ 이다. 이 때 1은 비디오를 구성하는 첫번째 프레임이며 n 은 마지막 프레임이다. 따라서 $n \geq 1$ 이다.

정의 2(shot) 샷은 (i, j) 로 나타낼수 있는데, 이때 $1 \leq i, \leq j \leq n$ 인 관계가 성립한다. 즉 (i, j) 는 i 번째 프레임으로부터 j 번째 프레임까지의 모든 프레임으로 이루어진 집합 $\{k \mid i \leq k \leq j\}$ 이다. 따라서 샷 (i, j) 의 길이는 $j-i+1$ 이 된다. 이때, i 는 샷 (i, j) 의 first이고 j 는 샷 (i, j) 의 last이다.

정의 3(keyframe) 샷 (i, j) 의 키프레임 k 는 샷 (i, j) 의 first이다.

정의 4(partial ordering) 두 개의 샷 (i_1, j_1) 과 (i_2, j_2) 가 $i_1 \leq j_1 \leq i_2 \leq j_2$ 의 조건을 만족하면 두 샷은 부분적으로 순서가 있다(partial ordering)고 하고, $(i_1, j_1) \sqsubseteq (i_2, j_2)$ 로 표현한다. 이 때, $(i_1, j_1) \sqsubseteq (i_2, j_2)$ 를 만족하고 $j_1 \neq i_2$ 인 경우를 특별히 $(i_1, j_1) \subset (i_2, j_2)$ 로 표현할 수 있다.

정의 5(well-ordered) 샷의 집합 X 의 모든 샷이 다음의 조건을 만족하면, 집합 X 는 잘 정돈되어있다(well-ordered)고 한다.

1. 집합 X 는 유한하다. 즉 정수 r 에 대해 $X = \{(i_1, j_1), \dots, (i_r, j_r)\}$ 이다. (단, $r \geq 1$)
2. $(i_1, j_1) \subset (i_2, j_2) \subset \dots \subset (i_r, j_r)$

정의 6(general scene) 보통장면(general scene)은 잘 정돈된 샷의 집합 $\{(i_1, j_1), \dots, (i_r, j_r)\}$ 이며, $[(i_1, j_1), \dots, (i_r, j_r)]$ 로 나타낸다.

정의 7(solid) 샷의 집합 X 의 모든 샷이 다음의 조건을 만족하면, 집합 X 는 완전하다(solid)고 한다.

1. 집합 X 는 잘 정돈되어있다.
2. 연속된 두 샷 (i_1, j_1) 과 (i_2, j_2) 사이에는 어떠한 샷도 존재하지 않는다. 즉 j_1 과 i_2 는 연속된 프레임이다.

정의 8(scene) 정의 6의 보통장면 중 완전한 샷의 집합으로 구성된 $\{(i_1, j_1), \dots, (i_r, j_r)\}$ 을 장면이라하며, $[(i_1, j_1), \dots, (i_r, j_r)]$ 로 나타낸다.

3.2 브라우징 트리

브라우징 트리 T 는 장면을 나타내는 노드 v_1, v_2, \dots, v_n 의 집합으로, 다음과 같은 일반적인 트리의 규칙을 따른다.

1. Root(T)로 표기되는 특별한 노드(v_1)가 있다.
2. 그 외의 노드들(v_2, \dots, v_n)은 T_1, T_2, \dots, T_m ($m \geq 0$)인 서로 겹치지 않는 집합들로 묶여지는데, 이 때 T_i 는 그 자체가 브라우징 트리이다.

일반적인 트리의 규칙에 다음과 같은 브라우징 트리만의 규칙이 존재한다.

3. 노드의 차수(degree) k 는 0, 1, 또는 2이다($k \leq 2$).
4. 따라서, n 개의 노드로 구성된 브라우징 트리의 깊이는 $\log_2(n+1)$ 이다.
5. 브라우징 트리의 계층 h 에 존재하는 최대 노드수는 $2^{(h-1)}$ 개 이다.
6. 자식 노드가 없는 경우에는 null값을 저장한다. 따라서, 브라우징 트리의 계층 h 의 노드들은 메모리의 $s + (2^{(h-1)} - 1) * \text{length}$ 위치에 순차적으로 저장된다. 이때 s 는 브라우징 트리를 저장하는 배열의 메모리 내 시작 위치이며, length 는 노드의 크기이다.
7. 반대로, 브라우징 트리를 저장하는 배열 내의 위치 i 는 계층 $\log_2(i+1)$ 의 노드 중의 하나이다.
8. 브라우징 트리 내의 임의의 노드 v_i 의 왼쪽 자식 노드는 $v_{(i-2)}$ 이다.
9. 또한, 브라우징 트리 내의 임의의 노드 v_i 의 오른쪽 자식 노드는 $v_{(i+2)}$ 이다.

위의 모든 규칙을 만족하는 브라우징 트리를 생성하기 위해, 트리 클래스에서는 $\log_2(n+1)$ -1개의 배열로 구성된 2차원 배열 levels 의 h 번째 1차원 배열에 브라우징 트리의 계층 h 의 노드들을 순차적으로 저장한다. 이때 2차원 배열 levels 의 h 번째 1차원 배열은 각각 2^h 개의 원소로 구성된다. 배열의 원소에 노드를 저장하는 과정은 재귀적 호출을 통해 실행되고, 모든 저장이 끝나면 1차원 배열 nodes 에 2차원 배열 levels 의 1번째 1차원 배열의 1번째 원소부터 $\log_2(n+1)$ -1번째 1차원 배열의 마지막 원소까지 순서대로 옮겨진다.

3.3 ST 브라우징

ST 브라우징은 트리의 노드가 자식 노드를 확장(expand)하거나 축소(reduce)하는 운영방식을 브라우징에 적용한 것이다. 3.1에서 설명한 클러스터링 결과로 구한 장면을 가지고 3.2에 기술한 조건을 만족하는 브라우징 트리를 생성한다. 이때 각 장면의 위치는 트리 상의 위치에 대응하지만, 노드간의 선분은 생략하고 장면의 비디오 내의 순서에 따라 선분을 새로 그린다.

브라우징 트리 상의 각 노드는 장면 하나와 대응된다. 따라서 사용자가 원하는 장면을 화면의 노드 중에서 찾으면, 브라우징은 끝나게 된다. 원하는 장면이 노드 중에서 보이지 않는 경우에는 노드를 확장한다. 노드가 확장되어 하위 계층이 펼쳐지면 2개의 노드가 트리에 추가되는데, 선택한 노드(장면)의 앞, 뒤 장면을 나타내는 노드가 된다. 사용자는 원하는 장면이 발견될 때까지 노드를 확장해나가고, 경우에 따라서는 더 이상 필요없는 노드를 축소시킬 수도 있다. 이때 노드와 노드를 잇는 선분은 비디오 클립내에서의 장면의 순서에 맞춰 일정

한 순서를 이루도록 계속 업데이트된다.

ST 브라우징을 좀 더 명확하게 설명하기 위해서 여기서는 작은 비디오 클립을 예로 사용하겠다. 사용되는 비디오 클립은 유명한 텔레비전 광고물로 상영시간이 약 19초 정도 된다. 비디오 클립의 크기는 2,662 킬로바이트이며, 프레임의 수는 571개이다. 세그멘테이션 결과 전체 샷의 개수는 13개이고, 이 중 같은 배경인 것들을 모아 5개의 장면을 검출하였다. 그림 2는 검출된 샷의 키프레임들을 보인 것이다.



그림 2 예제 클립에서 구한 샷들

샷이 검출되고나면 결정된 키프레임을 비교하여 장면을 찾는데, 키프레임을 비교하는 방법에 따라 다양한 장면 검출법이 있을 수 있다. 여기서는 주로 컬러값을 기준으로 인터랙티브하게 그룹을 지어 장면을 찾았다. 장면을 대표하는 키프레임은 장면 내의 첫 샷의 키프레임을 사용한다. 예제 비디오 클립에서는 5개의 보통장면이 검출되었는데, 이 보통장면 내의 샷을 대상으로 1에서 기술한 바와 같이 클러스터링을 하면 그림 3과 같은 장면들을 얻을 수 있다. 보통장면들과 클러스터링한 그림 3의 장면들이 다른 점은, 보통장면들이 배경의 컬러만으로 그룹되어진 것에 반해 장면들은 인접한 샷들로 재구성했다는 점이다. ST 브라우징 기법은 장면과 더불어

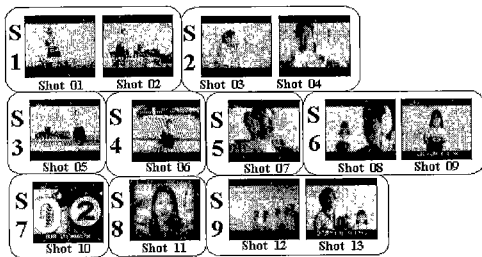


그림 3 클러스터링 결과로 얻은 장면들

샷의 순서도 반영하므로, 인접하지 않은 샷은 서로 다른 장면으로 취급한다.

모든 노드를 사용자에게 보이는 것은, 예제 비디오 클립과 같은 적은 수의 장면들 간의 관계에서는 무리가 없어 보이지만, 대단히 많은 수의 장면으로 구성된 비디오 클립에서는 STG에서와 같이 복잡도가 증가하여 사용자가 원하는 장면을 찾는 것을 어렵게 될 수 있다. 따라서 실제 사용자와 인터랙티브하게 동작하는 브라우저에서는 브라우징 트리의 Root(T)를 중심으로 브라우징을 행해나가고, 노드 사이의 선분(edge)는 장면 간의 시간적인 순서를 의미하게 된다.

여기서는 설명 상의 편의를 위해, 지난 브라우징의 결과로 장면2, 장면4, 장면5, 장면7, 장면8의 다섯 개의 장면이 보인다고 가정한다(그림 4). 이때 각 장면의 위치는 브라우징 트리 상의 위치와 같고, 장면 간의 선분은 장면 전환 순서(Scene Transition Order)가 된다. 사용자가 다섯 장면 중 원하는 장면을 찾지 못하면, 브라우징을 계속한다. 브라우징 방법은 두 가지로 장면을 보고 찾는 방법과 위치를 보고 찾는 방법이다. 2장에 기술된 바와 같이 두 방법은 모두 장단점을 가지고 있다. ST 브라우징은 두 가지 방법을 모두 지원한다. 사용자는 화면에 이미 표시되어 있는 장면들 중에서 원하는 장면을 찾거나 화면 상단의 눈금자에서 적당하다고 생각하는 위치를 선택할 수 있다. 화면의 나오는 장면을 하나 선택하면 해당하는 장면을 확장하거나 축소할 수 있는데, 장면을 확장하면 해당하는 장면의 앞, 뒤 장면이 화면에 나타나고 장면 전환 순서를 나타내는 선분은 업데이트 된다. 따라서 사용자는 자신이 원하는 장면으로부터 가장 가까운 위치의 장면으로부터 원하는 장면까지 확장시켜 나갈 수 있다. 그림 5는 그림 4의 장면5를 확장시킨 것이다. 이 때, '+' 기호가 붙은 노드는 확장 가능한 하위 노드가 숨겨져 있는 노드이며, '-' 기호가 붙은 노드는 모든 노드가 화면에 표시되었으며 축소 가능하다는 표시이다. 또한 'x' 기

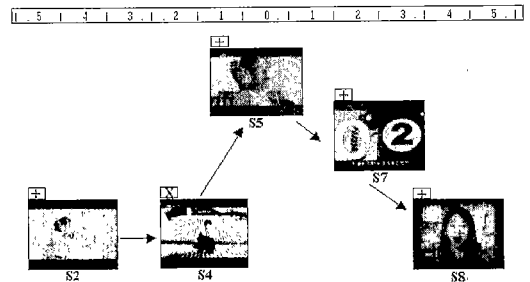


그림 4 브라우징 시작화면

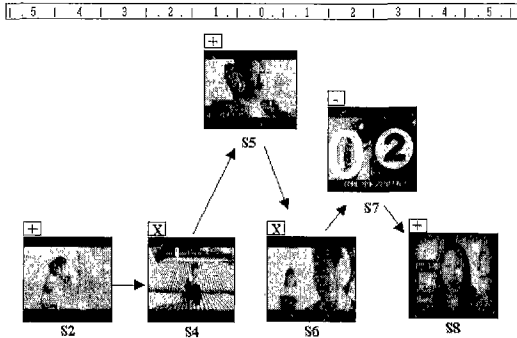


그림 5 장면5를 확장시킨 결과

호가 붙은 노드는 제거할 수 있는 노드이다.

브라우징 트리는 비디오 클립의 중간을 Root(T)로 하므로, 이런 방식의 브라우징은 장면의 위치를 쉽게 알 수 있어 매우 효율적이다. 또 다른 방법으로는 눈금자로부터 한 번에 원하는 장면을 찾는 방법이다. 이 방법은 사용자가 비디오 클립 전체로부터 원하는 장면의 절대 위치를 정확히 알고 있을때 도움이 된다. 장면을 확장하는 것 외에 축소시키거나 제거 할 수도 있다. 너무 많은 장면을 확장시켜 화면이 복잡해지면, 사용자는 스크롤바를 이용하여 화면을 계속 조정해야만 한다. 효율적인 공간사용에 위배되는 것이다. 한정된 모니터 화면에서 불필요한 많은 장면은 사용자를 혼란스럽게 하여 브라우징의 성공률을 크게 떨어뜨리고, 심지어는 목적하는 장면의 형태를 잇게 할 수도 있다. 불필요한 장면을 축소시키면, 해당하는 장면의 서브트리를 화면에서 제거한다. 이때 선택된 장면은 제거되지 않는데, 이는 사용자가 실수로 서브트리를 제거했을때, 다시 펼칠 수 있도록 제거된 서브트리의 실마리를 제공하기 위해서이다. 만약 선택된 장면마저 없애고 싶을 때는 축소가 아닌 제거를 사용할 수도 있다.

ST 브라우징의 또 다른 장점은 시각적 요약 제작물로 사용될 수 있다는 점이다. 브라우징 결과물은 디지털 라이브러리 등에서 비디오 검색을 위한 키프레임으로 사용되거나 비디오의 내용을 요약한 시각적 요약로 사용될 수 있다.

4. 결론

본 논문에서 제안하고 구현한 ST 브라우징 기법은 트리의 구조와 운영방식을 이용하여 효과적으로 비디오 클립을 브라우징할 수 있다. 사용자는 계층적 브라우징 기법에서와 마찬가지로 위치정보를 효율적으로 이용할

수 있을 뿐만 아니라, 하위 계층으로 장면을 확장하거나 상위 계층으로 장면을 축소하여 원하는 장면을 효율적으로 검색할 수 있다. 또한 장면 기반 브라우징에서처럼 장면정보와 장면 전환 순서를 명확히 알 수 있고, 임의의 장면으로부터 원하는 장면까지 찾아 나갈 수 있으므로 위치정보가 불분명한 경우에도 도움을 줄 수 있다. 장면 탐색 결과는 연속적인 애니메이션으로 저장하여 비디오 검색에 유용한 시각적 요약로 사용될 수 있다.

또한, 본 논문에서 제시된 클러스터링 방식을 통해 세그멘테이션한 장면을 재구성할 수 있으며, 제시된 조건에 따른 브라우징 트리를 만들 수 있다. 제시된 클러스터링 기법은 컬러정보를 통해 그룹 지어진 장면에 순서정보를 포함시키므로 위치정보에 의한 브라우징을 가능하게 한다. 브라우징 트리는 비디오 클립의 중간에 위치하는 장면을 Root(T)로 하는 트리이므로 확장·축소에 의해 브라우징하는 것을 가능하게 한다. ST 브라우저에서는 브라우징 트리를 통해 얻은 노드에 장면 전환 순서를 선분으로 추가하여, 장면 기반 브라우징을 가능하게 한다.

참고 문헌

- [1] HongJiang Zhang, Stephen W. Smoliar and Jian Hua Wu. Content-Based Video Browsing Tools. SPIE Vol.2417, p. 389-398, 1995.
- [2] Amarnath Gupta, Simone Santini, and Ramesh Jain. In Search of Information in Visual Media. Communications of the ACM, Vol.40, No.12, p. 34-42, December 1997.
- [3] Boon-Lock Yeo and Minerva M. Yeung. Retrieving and Visualizing Video. Communications of the ACM, Vol.40, No.12, p. 43-52, December 1997.
- [4] Yukinobu Taniguchi, Akihito Akutsu, Yoshinobu Tonomura. PanoramaExcerpts: Extracting and Packing Panoramas for Video Browsing. Proc. of ACM Multimedia 1997, p. 427-436, 1997.
- [5] Minerva M. Yeung, Boon-Lock Yeo, Wayne Wolf, and Bede Liu. Video Browsing using Clustering and Scene transitions on Compressed Sequences. Proc. of SPIE, Vol.2417, p. 399-413, 1995.



임 동 혁

1994년 고려대학교 전산학과 졸업(학사).
1999년 한국과학기술원 정보통신학과 컴퓨
터공학전공 졸업(석사). 1999년 ~ 현재
한국과학기술원 전자전산학과 전산학전공
박사과정. 관심분야는 멀티미디어 브라우징,
멀티미디어 데이터베이스, 객체지향 데이터

베이스, 데이터마이닝



이 석 룡

1984년 연세대학교 기계공학과 졸업(학
사). 1993년 연세대학교 산업대학원 전자
계산과 졸업(석사). 1984년 ~ 1995년 한
국IBM(주) 소프트웨어 연구소. 1995년
~ 현재 한국과학기술원 정보통신공학과
컴퓨터공학전공 박사과정. 관심분야는 멀
티미디어 데이터베이스, 객체지향 데이터베이스.



정 진 완

1973년 서울대학교 공과대학 전기공학과
(학사). 1983년 University of Michigan
컴퓨터공학과(박사). 1983년 ~ 1993년
미국 GM 연구소 전산학과 선임연구원
및 책임연구원. 1993년 ~ 현재 한국과
과학기술원 전산학과 부교수 및 교수. 관심
분야는 XML, 멀티미디어 데이터베이스, GIS.

분야는 XML, 멀티미디어 데이터베이스, GIS.