VTOC-트리와 비디오 모델링 연산자

오승[O] 김혁만
국민대학교 전산과학과 멀티미디어 데이터베이스 연구실
{shareguy77[O], hmkim}@cs.kookmin.ac.kr

VTOC-Tree and video modeling operators

Seung Oh[0] Hyeokman Kim
Multimedia Database Lab., Dept. of Computer Science, Kookmin University

요 약

디지털 방송이나 디지털 저장 매체 또는 인터넷을 통하여 제작, 전송되는 비디오의 내용을 좀더 효과적으로 브라우징할 필요성이 대두되고 있다. 또한 비디오가 내포하고 있는 복잡한 정보들을 기술하고 그 구조를 표현하는 저작 도구들의 필요성이 날로 증대되고 있다. 이렇게 비디오의 내용은 브라우징하거나 복잡한 정보들을 기술하기 위해 트리 형태의 계층구조로 표현된다. 본 논문에서는 세그먼트 계층구조를 갖는 VTOC-트리를 정의하고 세그먼트 경계 검출 알고리즘을 통해 계층구조를 자동적으로 생성한 후, 이 구조를 사용자가 수작업을 통해 원하는 형태로 전환시키는 모델링 연산자를 제안한다.

1. 서론

비디오의 환경이 급속하게 변화되고 있는 지금 비디오의 내용을 파악하는 방법들이 날로 다양진 가운데 하드웨어 기능을 뒷받침할 내용 파악 기법으로 브라우징을 주로 사용한다. 이러한 브라우징 기법들은 크게 시간 연속 구조를 보여주는 스토리보드 기법[1], 트리 구조를 이용한 계층 브라우징 기법[2], 그리고 좀더 복잡한 관계의 표현이 가능한 그래프 브라우징 기법[3]으로 나눌 수 있는데 모두 비디오의 단편적인 구조에 의존하여 완전한 내용 파악에는 무리가 있다. 결국 직관적인 해석이 가능하고, 복잡한 내용의 표현이 가능하며, 사용자의 의도에 맞는 구조를 요약 및 정리, 브라우징해 갈 수 있는 계층 브라우징 기법이 가장 보편적으로 사용되는 추세이다.

계층 브라우징을 위해서는 비디오의 내용을 트리 형태의 계층구조로 표현해야 하며, 이를 위해서는 트리 구조를 편집하는 연산이 필요하다. 트리 구조를 편집하는 가장 간단한 방법은 MovieTool[4]과 같이 윈도우의 탐색기와 같은 인터페이스를 제공하여, 탐색기에서 폴더를 생성하고 파일을 옮기는 것과 비슷한 방법으로 비디오 세그먼트와 서브세그먼트를 정의하는 것이다. 이 방법은 계층구조 편집시 복잡한 구조 변환을 원할 경우 여러 단계를 거쳐야 한다. Wright 대학에서 개발한 계층구조 편집기[5]는 계층구조 변환을 위한 몇 가지 연산을 제공한다. 그러나 이들 연산에 대한 정형화된 표현이 없으며, 제공하고 있는 연산도 매우 간단한 것으로 계층구조 편집시 자주 발생하는 복잡한 구조 변환을 하기 위해서는 몇 가지 연산을 연속적으로 적용해야 하는 번거로움이 있다. 또한 비디오 계층구조의 제약조건을 고려하지 않고 있다. 본 논문에서는 계층구조 편집을 위한 중요 연산들을 정형화하여 제안한다. 제안한 연산들은 비디오 계층구조 제약조건을 충분히 고려하여 브라우징하기 쉬운 최대한 간단한 계층구조를 생성한다.

2. 비디오의 세그먼트 계층구조

비디오의 내용에 대한 의미 정보를 기술하는 기본 단위는 세그먼트(segment)이다. 세그먼트는 연속된 프레임의 집합으로, 비디오 전체를 하나의 세그먼트로 볼 수 있다. 세그먼트는 몇 개의 겹치지 않는 서브세그먼트(subsegment)로 나뉠 수 있다. 이 경우 세그먼트와 그것의 서브세그먼트들의 종속 관계를 트리 형태로 표현하면, 이 트리를 세그먼트 계층구조(hierarchical segment structure)라 한다. 특정 비디오에 대해 세그먼트 계층구조를 만드는 과정을 계층 비디오 모델링(hierarchical video modeling)이라 한다. 세그먼트 계층구조의 각 세그먼트에 세그먼트 타이틀을 부여하거나 세그먼트에 속한 프레임 중 하나를 대표화면으로 선택하면, 세그먼트 타이틀 혹은 대표화면으로 표현되는 세그먼트 계층구조를 통해 그 비디오의 내용을 편하게 브라우징하고, 세그먼트별로 재생할 수 있다. 또한 세그먼트 계층구조 자체가 비디오의 전체 내용 요약으로 사용될 수도 있다. 그림 1은 세그먼트 계층구조의 예를 보여준다.

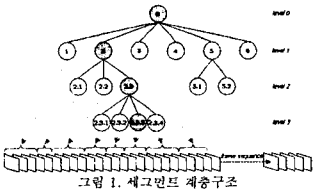

그림 1. 세그먼트 계층구조

세그먼트 계층구조에서 하위 레벨의 세그먼트들은 상위 레벨의 특정 세그먼트에 포함되며, 동일한 세그먼트에 포함된 모든 서브세그먼트들은 시간 순서에 따라 정렬된다. 세그먼트 계층구조에서 루트 세그먼트의 레벨을 0이라 하고, 그것의 프레임 개수를 t 라 할 때, 세그먼트 계층구조의 모든 세그먼트는 다음에 정의를 만족한다.

[정의 1] i 번째부터 $j-1$ 번째 프레임까지의 연속된 프레임의 집합은 세그먼트 $S_{[i,j)}$ 로 표현한다. 여기서 i 와 j 는 $0 \le i < j \le t$ 를 만족하는 정수이며, i 는 세그먼트 $S_{[i,j)}$ 의 시작 프레임, $j-1$ 은 마지막 프레임을 의미한다.

[정의 2] 세그먼트 $S_{[i_1,j_1)}$과 $S_{[i_2,j_2)}$가 $i_1 \le i_2 < j_2 \le j_1$를 만족하면, 세그먼트 $S_{[i_1,j_1)}$이 $S_{[i_2,j_2)}$를 포함(segment-inclusion)한다 하고 $S_{[i_1,j_1)} \supseteq S_{[i_2,j_2)}$으로 표현한다. 세그먼트 $S_{[i_1,j_1)}$이 $S_{[i_2,j_2)}$를 포함할 때 $i_1 \neq i_2$ 혹은 $j_1 \neq j_2$를 만족하면, 세그먼트 $S_{[i_1,j_1)}$이 $S_{[i_2,j_2)}$를 완전 포함(complete segment-inclusion)한다 하고 $S_{[i_1,j_1)} \supset S_{[i_2,j_2)}$로 표현한다.

[정의 3] 세그먼트 $S_{[i_1,j_1)}$과 $S_{[i_2,j_2)}$가 $i_1 < j_1 \le i_2 < j_2$를 만족하면, 두 세그먼트는 부분 정렬(partial ordering)되었다 하고 $S_{[i_1,j_1)} \sqsubseteq S_{[i_2,j_2)}$으로 표현한다 [6]. 부분 정렬된 두 세그먼트 $S_{[i_1,j_1)}$과 $S_{[i_2,j_2)}$가 $j_1 = i_2$을 만족하면, 두 세그먼트는 연속(successive) 부분 정렬되었다 하고 $S_{[i_1,j_1)} \cong S_{[i_2,j_2)}$으로 표현한다.

[정의 4] 세그먼트들의 집합 S가 다음 두 조건을 만족하면, 이 집합은 정-순서화(*well-ordered*)되었다고 한다 [6].

1. S는 유한하다. (즉, $S = \{S_{[i_1,j_1)},...,S_{[i_r,j_r)}\}$, r은 임의의 정수).
2. $S_{[i_1,j_1)} \sqsubseteq S_{[i_2,j_2)} \sqsubseteq ... \sqsubseteq S_{[i_r,j_r)}$.

만일 S의 세그먼트들이 모두 연속 부분 정렬되었다면(즉 $S_{[i_1,j_1)} \cong S_{[i_2,j_2)} \cong ... \cong S_{[i_r,j_r)}$), 이 집합 S는 완전 정-순서화(completely well-ordered)되었다고 한다.

어떤 세그먼트들의 집합이 정-순서화되었다면 세그먼트들이 시간순으로 정렬된 것을 의미하며, 완전 정-순서화되었다면 연속적인 세그먼트들이 시간순으로 정렬된 것을 의미한다. 따라서 어떤 세그먼트의 모든 서브세그먼트들이 완전 정-순서화되었다고 하면, 세그먼트가 겹치지 않게(disjoint) 서브세그먼트들로 분할되어 시간순으로 정렬된 것을 의미한다.

3. VTOC-트리와 비디오 모델링

3.1 VTOC-트리의 정의

비디오의 세그먼트 계층구조와 그 계층구조가 갖는 제약조건을 반영하기 위해 VTOC-트리를 정의한다.

[정의 5] t개 프레임으로 이루어진 비디오 $S_{[0,t)}$에 대한 VTOC-트리는 다음 성질을 만족하는 다원 순차 트리(n-ary ordered tree)이다.

1. $0 \le i < j \le t$를 만족하는 세그먼트 $S_{[i,j)}$는 노드로 표시한다.
2. 세그먼트 $S_{[i,j)}$는 완전 정-순서화된 2개 이상의 서브세그먼트로 분할(segmentation)될 수 있으며 (즉 $S_{[i,j)} = \{S_{[i,k_1)}, S_{[k_1,k_2)}, S_{[k_2,k_3)},..., S_{[k_r,j)}\}$, 단 $r \ge 1$), 세그먼트와 서브세그먼트의 완전 포함관계는 포함하는 세그먼트에서 포함되는 서브세그먼트로의 링크로 표시한다.

정의에 의해 VTOC-트리의 모든 노드들은 자식 노드가 없거나, 있으면 두 개 이상 존재한다. 즉 모든 비단말 노드의 차수(degree)는 2 이상이다. 그리고 VTOC-트리의 임의의 레벨에 있는 모든 노드들은 정-순서화되어 있으며, 모든 단말 노드들은 완전 정-순서화되어 있다. 또한 VTOC-트리의 임의의 서브트리도 VTOC-트리이다.

3.2 비디오 모델링

비디오 모델링은 임의의 비디오 프로그램으로부터 사용자가 원하는 비디오의 세그먼트 계층구조를 생성하는 과정을 의미한다. 입력 비디오를 하나의 세그먼트로 하여, 그 세그먼트 내의 서브세그먼트간의 포함관계를 정의하면서 원하는 계층구조를 생성한다. 비디오는 편집될 때 이미 편집의 단위인 샷(shot)이 정의되므로, 세그먼트의 최소 단위로는 일반적으로 샷이 사용된다. 따라서 샷 경계검출 알고리즘의 결과를 이용하여 비디오 모델링을 수행하면 매우 편하게 수작업으로 모델링을 수행할 수 있다. 샷 경계 검출 알고리즘을 사용하지 않고 모든 모델링을 수작업으로 수행할 수도 있다.

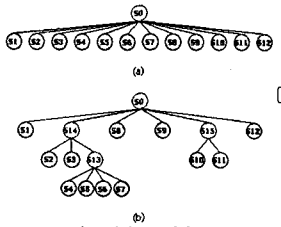

그림 2. 비디오 모델링

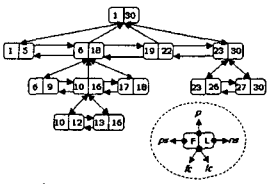

그림 3. VTOC-트리의 자료구조

그림 2.(a)는 샷 경계를 검출한 결과 검출된 12개의 샷을 전체 비디오 세그먼트의 서브세그먼트로 표현한 트리이다. 트리에서 12개 서브세그먼트는 완전 정-순서화되어 있으므로 이 트리는 VTOC-트리이다. 그림 2.(b)는 이 트리로부터 비디오 모델링 과정을 거쳐 사용자가 원하는 최종 계층구조를 구성한 결과이다. 이 트리는 그림 2.(a)의 트리에서 노드 S4, S5, S6, S7를 서브세그먼트로 하는 노드 S13, 노드 S2, S3, S13를 서브세그먼트로 하는 노드 S14, 노드 S10, S11을 서브세그먼트로 하는 노드 S15를 새로 정의하여 얻은 결과이다. 새로 정의한 노드 S13, S14, S15를 루트로 하는 서브트리도 VTOC-트리이므로, 그림 2.(b)의 트리 또한 VTOC-트리이다. 이와 같이 세그먼트들의 의미를 파악하여 입력 VTOC-트리로부터 새로운 형태의 VTOC-트리로 변환하는 연산이 필요하며, 이런 연산자를 모델링 연산자(modeling operator)라 한다. 체계적인 모델링 연산자가 제공되면, 비디오 모델링은 입력 VTOC-트리에 여러 가지 모델링 연산자를 연속적으로 적용하여 사용자가 원하는 형태의 VTOC-트리로 변환하는 과정으로 볼 수 있다.

4. 모델링 연산자

본 장에서는 비디오 모델링을 쉽고 편하게 수행할 수 있는 몇 가지 비디오 모델링 연산자를 제안한다. 제안하는 연산자의 기능을 정의하기 위해, 먼저 VTOC-트리의 자료구조를 설명하고, 그 자료구조를 이용하여 제안하는 연산자들의 알고리즘을 설명한다.

4.1 VTOC-트리의 자료구조

그림-3은 VTOC-트리의 세그먼트 데이터베이스를 구성하는 자료구조를 도식화한 것이다. 각 세그먼트는 특정 비디오 안에 시작 프레임과 마지막 프레임을 정의(*F*와 *L*)하고 그것의 내용을 요약(타이틀 혹은 대표화면)하는 추가 정보를 포함할 수 있다. 각 세그먼트는 링크를 통해 부모 노드, 자식 노드를 완전 포함 관계로 연결하고 형제 노드를 완전 정-순서화되도록 연결한다. 이것은 VTOC-트리의 정의에 따라 재귀적인 계층구조의 자료 표현을 허용한다.

VTOC-트리의 모든 세그먼트를 S로 표현할 때, 세그먼트 *S*의 부모 세그먼트를 *p*[*S*], 처음, 마지막 자식 세그먼트를 *fc*[*S*], *lc*[*S*], 앞, 뒤 형제 세그먼트를 지정하는 링크를 *ps*[*S*], *ns*[*S*]로 정의한다.

4.2 GROUP/UNGROUP 연산자

GROUP 연산자는 연속 부분 정렬된 2개 이상의 세그먼트들을 서브세그먼트로 하는 새로운 세그먼트를 생성하는 연산자이다. UNGROUP 연산자는 GROUP 연산자의 역으로, 특정 세그먼트를 삭제하고 그것의 서브세그먼트를 삭제된 세그먼트의 부모 세그먼트의 자식으로 만드는 연산자이다. 이때 연산을 전후로 전체 VTOC-트리는 항상 유효하다. 그림 4.(a)는 VTOC-트리 S2와 S4사이의 모든 세그먼트들을 묶어 새로운 세그먼트 S10을 생성하는 GROUP 연산자, 그리고 세그먼트 S10를 삭제하여 그것의 서브세그먼트인 S2와 S4 사이의 모든 서브세그먼트들을 루트의 서브세그먼트로 만드는 UNGROUP 연산자의 적용 예를 보여주고 있다. 특히 그림 4.(a)은 두 연산자가 서로 역 관계에 있다는 것을 보여준다.

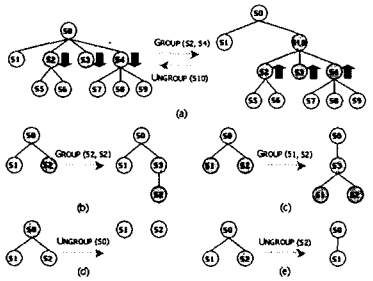

그림 4. GROUP/UNGROUP 연산자

그림 4.(b)~(c)는 GROUP 연산자를 적용할 때 발생하는 예외를 보여주고, 그림 4.(d)~(e)는 UNGROUP 연산자를 적용할 때 발생하는 예외를 보여준다. 그림 4.(b)~(e)의 경우를 허용하면, 연산자를 적용하여 변환된 트리가 VTOC-트리의 정의를 위반하게 되므로 이러한 경우를 막아야 한다. 예를 들어 그림 4.(b)의 GROUP 연산을 허용하면 S3이 단 하나의 자식 노드를 갖게 되어 VTOC-트리의 정의를 위반하게 되며, 그림 4.(d)의 UNGROUP을 허용하면 하나의 VTOC-트리가 두개의 VTOC-트리로 변환된다. 그림 5는 설명한 예외 처리를 고려한 GROUP/UNGROUP 연산자의 알고리즘을 보여준다.

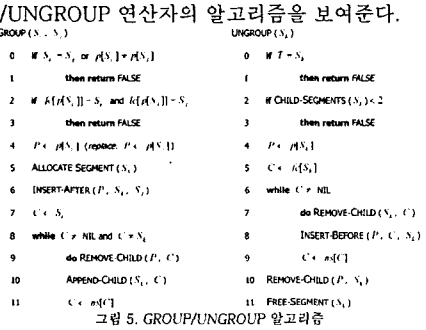

그림 5. GROUP/UNGROUP 알고리즘

4.3 MERGE/SPLIT 연산자

MERGE 연산자는 연속 부분 정렬된 2개 세그먼트을 하나로 합쳐 새로운 세그먼트를 생성하는 연산자이다. 이때 두 세그먼트의 모든 서브세그먼트들은 새롭게 생성되는 세그먼트의 서브세그먼트로 종속된다. SPLIT 연산자는 MERGE 연산자의 역으로, 특정 세그먼트를 연속 부분 정렬된 두개의 세그먼트로 분할하는 연산자이다. 이때 연산을 전후로 전체 VTOC-트리는 항상 유효하다. 그림 6.(a)는 세그먼트 S2와 S3를 합한 새로운 세그먼트 S9를 만들고 세그먼트 S2와 S3의 서브세그먼트들을 결합된 세그먼트 S9에 포함시키는 MERGE 연산자, 그리고 세그먼트 S9을 둘로 나누어 세그먼트 S2와 S3를 만들고 S9의 서브세그먼트들은 S6과 S7를 기준으로 나누어 새롭게 생성한 S2와 S3의 서브세그먼트로 각각 포함시키는 SPLIT 연산자를 보여주고 있다. 특히 그림 6.(a)은 두 연산자가 서로 역 관계에 있다는 것을 보여준다.

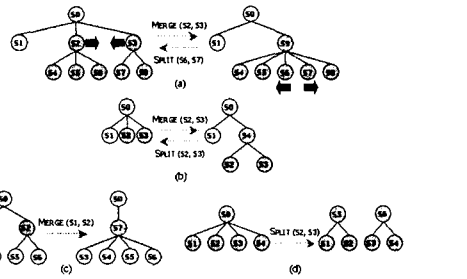

그림 6. MERGE/SPLIT 연산자

그림 6.(b)는 MERGE 연산자와 SPLIT 연산자를 적용할 때 발생할 수 있는 예외를 보여주고, 그림 6.(c)와 6.(d)는 각각 MERGE 연산자와 SPLIT 연산자를 적용할 때 발생하는 예외를 보여준다. 그림 6.(b)~(d)의 경우를 허용하면, 연산자를 적용하여 변환된 트리가 VTOC-트리의 정의를 위반하게 되므로 이러한 경우를 허용하지 말아야 한다. 예를 들어, 그림 6.(c)의 MERGE 연산을 허용하면 S0가 단 하나의 자식 노드를 갖게 되어 VTOC-트리의 정의를 위반하게 되며, 그림 4.(d)의 SPLIT 연산을 허용하면 하나의 VTOC-트리가 두개의 VTOC-트리로 변환된다. 그림 7은 설명한 예외 처리를 고려한 MERGE/SPLIT 연산자의 알고리즘을 보여준다.

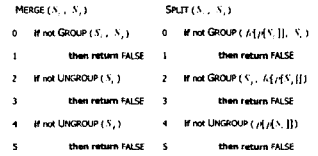

그림 7. MERGE/SPLIT 알고리즘

5. 결론 및 향후 과제

본 논문에서는 비디오의 내용에 대한 의미 정보를 기술하는 세그먼트 계층구조의 한계를 극복하기 위해 VTOC-트리를 정의하였다. 또한 VTOC-트리에서 비디오 모델링을 위한 모델링 연산자와 알고리즘을 정형화하여 제안하였다. 제안한 연산들은 비디오 계층구조 제약조건을 충분히 고려하고 있어 VTOC-트리의 계층구조를 변환하기에 효율적이다. 본 논문의 VTOC-트리와 모델링 연산자들은 비디오 내용 분석을 통해 비디오 계층구조를 자동으로 생성하는 도구 혹은 수작업에 의한 편집을 쉽게 해주는 편집기, 자동화 작업과 수작업을 병행해서 사용할 수 있는 저작 도구 등에서 사용될 수 있다.

참고문헌

1. F. Arman, et al., " Content-based browsing of video sequences" , Proc. ACM Multimedia, pp.97-103, Oct. 1994.
2. D. Zhong, et al., " Clustering methods for video browsing and annotation" , Proc. Storage and retrieval for image and video databases IV, pp.239-246, Jan. 1996.
3. B.-L. Yeo, M. M. Yeung, " Classification, simplification and dynamic visualization of scene transition graphs for video browsing" , Proc. Storage and retrieval for image and video databases VI, pp.60-70, Jan. 1998.
4. ISO/IEC JTC1/SC29/WG11(MPEG)/N5833, " MPEG-7 Industry Use" , ISO/IEC, July 2003.
5. F. Quek, R. Bryll, X. Ma, " Content-based video access" , VISLab Report, Wright State University, March 2000.
6. V. S. Subrahmanian, Principles of multimedia database systems, Morgan Kaufmann, 1998.