

유한 오토마타를 이용한 비디오 브라우징 시스템

서명필, 김기원, 이팔진
초당대학교 컴퓨터학과

Video Browsing System using Finite Automata

Myung-Pil Seo, Gi-Weon Kim, Pal-Jin Lee
Dept. of Computer Science, Chodang University

요약

본 논문에서는 문서 처리와 비디오 스트림 처리에 이용되는 이론적 배경을 고찰하고, 제시된 이론을 바탕으로 하나의 비디오 스트림을 결정적 유한 오토마타 형태의 논리적인 구조로 재구성하여 사용자의 요구나 수준에 따라 가변적인 재생 시간을 갖는 비디오 브라우징 시스템을 구현하였다.

1. 서론

최근 컴퓨터 기술의 발달과 함께 다양한 매체의 출현으로 기존의 문자 정보뿐만 아니라 정지영상, 동영상, 음성 데이터 등과 같은 다양한 멀티미디어 정보들의 효율적인 저장과 인덱싱 및 검색 등이 중요한 핵심 기술로 부각되고 있다. 하지만 멀티미디어 정보는 선형 또는 공간 형태의 데이터 구조를 가지고 있으며, 또한 데이터의 종류가 다양하고 비정형화된 데이터의 형태로 존재하여 기존의 데이터 모델링 기법으로는 효율적으로 표현할 수 없는 복잡하고 다양한 관계성을 포함하고 있다. 따라서, 멀티미디어 데이터를 검색하기 위해서는 기존의 텍스트를 위한 정보 검색 시스템과는 다른 접근 방법이 필요하다. 특히 멀티미디어 시스템에 있어서 핵심 목표는 정보손실을 최소화하면서 보다 편리하게 의사정보를 전달하기 위한 것이기 때문에 인간에 가까운 사용자 인터페이스 구축과 함께 데이터를 속성별로 분류하는 일이 우선적으로 이루어져야 한다. 또한 분류된 데이터에 자유롭게 접근하여 필요한 부분만을 검색할 수 있는 사용자 인터페이스가 제공되어야 한다. 이에 본 논문에서는 멀티미디어 데이터처리 시스템을 구현할 때 요구되는 기술 중에서 데이터 분류 및 시간 제약에 따른 비디오 스트림의 부분 재생에 관한 기술적 측면에 대하여 논의하였다.

2. 비디오 컷 검출

비디오 컷 검출의 기본 원리는 같은 장면내의 비디오 프레임들간에는 유사성이 강하고, 장면의 전환이 이루어지는 부분의 프레임들간에는 유사성이 적다는 것을 이용한다. 프레임들간의 유사도를 측정하는 방법으로는 일반적으로 화소단위의 비교법과 유사율 비교, 히스토그램 비교 등이 사용된다. 화소단위의 비교는 이웃한 프레임에서 대응하는 화소간의 차이를 계산하여 특정 임계값을 넘는지 알아보는 것이며, 유사율 측정은 프레임에서 일정 영역의 통계치를 비교하는 방법이다. 히스토그램 비교는 프레임의 화소의 세기, 색상성분을 히스토그램으로 표현하여 프레임들간의 히스토그램 차이를 이용하여 컷을 검출하는 방법이다.

2.1 프레임간의 밝기 히스토그램의 차이를 이용

프레임에서의 밝기 분포는 이미지가 급격히 변화되면 이에 비례하여 변화되나, 한 프레임 내에서 특정 이미지가 이동을 하는 경우에는 이 같은 변화량은 적어진다. 결과적으로 프레임의 밝기 분포는 하나의 장면 내에서는 적은 편차를 가지게 되므로, 이를 컷 검출에 이용할 수 있다. 프레임간의 밝기 히스토그램의 절대 차를 이용하는 $HDsum$ 은 다음과 같이 계산된다 [9].

$$HDsum(t) = \sum_{b=0}^R |h(t+\Delta t, b) - h(t, b)|$$

$$h(t, b) = \sum_{n=0}^N 1, \quad \text{if } Lb \ni In(t)$$

$$Lb = \{In(t) | \frac{b}{B} \leq In(t) < \frac{b+1}{B}\}, \quad B: \text{최대밝기}$$

HDsum은 프레임 N의 밝기 히스토그램과 프레임 N-1의 밝기 히스토그램과의 차를 구하는 것이 되는데, 이를 위해 우선 한 프레임의 밝기 히스토그램을 생성한 뒤, 이전 프레임의 히스토그램과 비교하여 각 밝기 레벨에 대한 절대 차를 구한 뒤 이를 모두 합하는 과정을 거쳐 구해진다. 계산된 HDsum 값이 특정 임계값을 넘는 경우 이 부분을 컷으로 구분한다.

임계값 설정시 데이터의 종류에 따라 임계값의 차이가 매우 크다. 따라서 비디오 데이터의 모든 종류를 사용자가 모르는 상황에서 아래의 식과 같이 임의의 임계값을 설정해야 된다.

$$|HDsum_{i+1} - HDsum_i| > t$$

따라서 본 논문에서 제시한 임계값(t)은 우선 전체 프레임에 대한 히스토그램 차이에 대한 절대값의 합을 구한다. 다음으로 구한 값을 전체프레임의 수-1로 나누어 평균을 구한다. 마지막으로 이 값을 이용하여 히스토그램 차이가 b배 이상이면 장면 변화가 있다고 설정하는 방법이다.

$$|HDsum_{i+1} - HDsum_i| > \frac{\sum_{i=0}^{Frame-1} |HDsum_i - HDsum_{i+1}|}{Frame - 1} \times b$$

여기서 HDsum_i는 i 번째 히스토그램의 차이 값이고, HDsum_{i+1}은 i+1 번째 히스토그램의 차이 값, 그리고 b는 일정 상수이다. 본 논문에서 제시한 임계값을 이용하면 사용자가 비디오 데이터마다 임계값 설정시에 부딪히는 어려움을 해결할 수 있다.

3. 비디오 스트림 저장

본 논문에서 제안한 비디오 스트림 저장은 결정적 유한 오토마타(Deterministic Finite Automata)를 저장 모델로 이용하여 입력된 비디오를 비디오 구조화에 의해 구조화된 논리적 비디오로 재구성하여 저장 관리하였다. 이에 의해 하나의 비디오 에피소드에 대해 다음의 그림 1과 같은 상태를 생성한다. 몇 개의 장면으로 구성된 하나의 에피소드가 포함되며, 하나의 상태 내에는 에피소드내의 전체 장면 재생에 소요되는 처리시간, 장면들의 대표 키워드들과, 에피소드의 내용 중요도 등을 상태 정보로 추가한다.

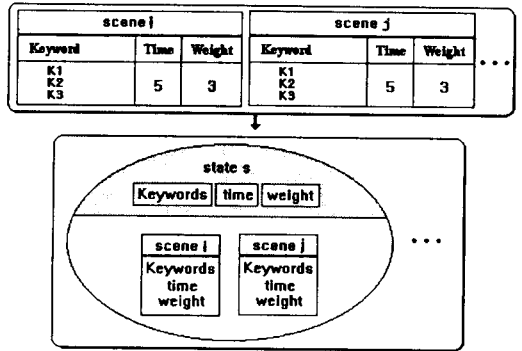


그림 1. 비디오 에피소드에 대한 상태 표현

또한 하나의 장면 내에는 장면 재생에 소요되는 처리시간, 대표 키워드와 몇 개의 기타 키워드, 장면의 내용 중요도 등을 상태 정보로 추가한다. 그후, 에피소드들의 내용에 대한 중요도를 기준으로 의미 가중치를 부여하며 의미적으로 상호 연관이 있는 상태들을 가중치가 레이블(label)인 지시선(directed arc)으로 연결한다. 이와 같은 과정을 거쳐 최종적으로 그림 2와 같은 비디오 스트림에 대한 오토마타를 생성한다.

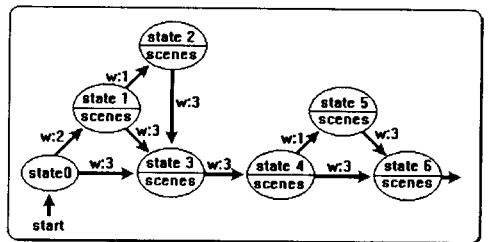


그림 2. 비디오 에피소드에 대한 오토마타

4. 비디오 스트림의 인덱싱

4.1 인덱싱 요소의 저장 구조

비디오를 인덱싱하여 얻은 결과는 장면 전환과 자막 정보이다. 이들 정보를 그림 3과 같은 구조로 저장한다. 저장된 인덱싱 정보는 비디오 검색 시스템에서 질의를 이용하여 검색 할 수 있으며, 검색 결과에 대한 정보를 제공할 수 있다. 비디오 인덱싱 결과를 샷과 자막의 인덱싱구조로 분할하여 저장한다.

샷 인덱스	시작 프레임번호	끝 프레임번호	자막해디	자막 인덱스데이터
			자막 인덱스	자막 인덱스

(a)

자막인덱스	시작 프레임 번호	끝 프레임 번호	자막 문자 데이터(문자열)
-------	--------------	-------------	-------------------

(b)

그림 3. 인덱스 저장 구조

4.1.1 샷 인덱스 구조

비디오 인덱싱 결과로 검출된 장면 전환을 저장하기 위한 구조이다. 샷 인덱스 구조에서 시작과 끝 프레임 번호는 하나의 샷이 시작하고 끝나는 프레임 번호이다. 자막 헤더는 샷에 포함된 자막 종류(내용)의 개수이다. 자막 인덱스 데이터에 포함되어 있는 자막 인덱스는 자막 인덱스 구조와 매칭을 위해서 이다. 장면 전환 인덱스 구조는 그림 3의 (a)이다.

4.1.2 자막 인덱스 구조

자막 인덱스 구조는 자막 프레임 검출, 자막 문자 추출, 자막 문자 인식 과정을 통해 얻어진 자막 정보를 저장하기 위한 것이다.

시작과 끝 프레임 번호는 자막이 나타나고 사라지거나 다른 자막으로 바뀌는 프레임 번호이다. 자막 문자 데이터는 인식된 자막 문자들의 문자열이다. 자막 인덱스 구조는 그림 3의 (b)이다. 그림 4는 인덱스 목록 중에서 하나를 선택하여 클릭하면 해당 인덱스 파일에 대한 정보를 가지고와 브라우저에 해주는 프로그램이다.

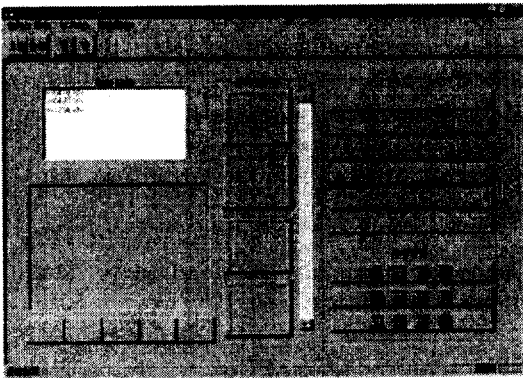


그림 4. 목록 선택 화면

5. 비디오 스트림 재생

본 논문에서는 장면들간의 상호 관련성을 고려하는 모델로 결정적 유한 오토마타를 이용하여 스트림을 저장하였기 때문에 사용자의 다양한 재생시간 요구에 따라 가변적인 재생시간을 제공할 수 있게 된다. 상태전이를 통한 재생 알고리즘은 다음과 같다.

Algorithm Video Browsing1

begin

current_state := *initial_state*;

state_queue := *initial_state*;

get(*weight* *i*);

while *current_state* is not *final_state* do

if exist arc of *weight* *i* then

current_state := δ (*current_state*, *weight* *i*);

else if exist arc of *weight* *i*+1 then

current_state := δ (*current_state*, *weight* *i*+1);

else if exist arc of *weight* *i*+2 then

current_state := δ (*current_state*, *weight* *i*+2);

end if

append *current_state* to *state_queue*;

end while

for each state in the *state_queue* do

play the scenes of state;

end for

end

그림 5는 특정 교육에 관련된 비디오 스트림에 적용한 경우를 보이고 있는데, 수강자의 수준에 따라 난이도를 3단계(W=1,2,3)로 조정하여 난이도에 따라 각기 처리 시간이 다른 서비스를 제공할 수 있다. 즉, 초급과정의 경우 우선적으로 가중치 레이블이 1인 지시선으로 상태전이(state transition)를 하고 만약 해당 전이가 존재하지 않으면 레이블 2, 3으로 전이를 하며, 중급과정의 경우에는 가중치 레이블이 2또는 3인 상태로 전이를 한다. 그리고 고급과정의 경우 유한 오토마타 입력 스트링을 w=3,3,3,... 로 하여 상태전이를 실행한다.

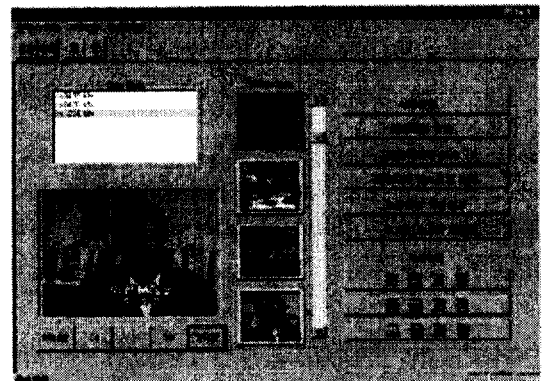


그림 5. 특정 교육 비디오 스트림 재생

또한 그림 6은 영화 비디오 스트림에 적용한 경우를 보이고 있는데 사용자에게 따라 상영 시간을 입력하게 되면 사용자의 요구 재생 시간에 맞게 에피소드 단위로 비디오를 구성하여 빠른 시간내에 영화의 흐름을 파악할 수가 있다. 본 논문에서 제안한 비디오 브라우징 시스템은 에피소드 단위로 비디오를 구성할 수 있는 환경을 제공하므로 사용자는 비디오를 논리적인 단위로 구성할 수 있으며 이것을 비디오 이야기 문서(video story document)등이나, 영화의 예고편이나 교육용 비디오 등에 효과적으로 이용할 수 있다.

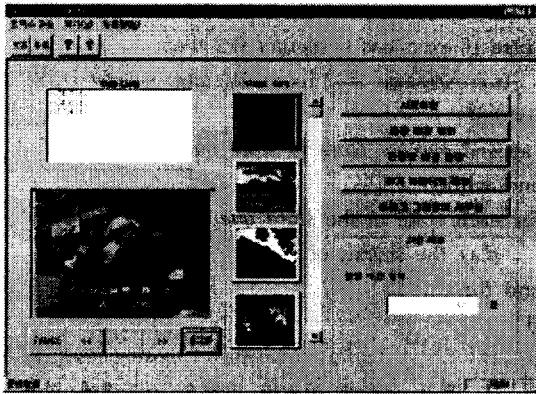


그림 6. 영화 비디오 스트림 재생

6. 결론

본 논문에서는 유한 오토마타를 비디오 스트림의 처리 모델에 적용하였다. 장면 검색과 재생시간 요구에 따라 융통성 있게 스트림을 재생하는 모델을 구성하기 위해서, 결정적 유한 오토마타(Deterministic Finite Automata)를 장면들의 저장모델로 이용하였으며 비디오 브라우징 단계에서 현재의 VOD가 갖는 단점인 고정된 재생시간 문제를 해결하는 방안으로, 하나의 비디오 스트림을 결정적 유한 오토마타 형태의 논리적인 구조로 재구성하여 사용자의 수준에 따라 가변적인 재생 시간을 갖는 비디오 브라우징 시스템의 모델을 구현하였다.

본 논문에서 제안한 결정적 유한 오토마타를 이용하여 논리적 비디오를 비디오 브라우징 및 검색 시스템은 VOD, CATV, 광고 제작 등의 여러 비디오 분야에 이용될 수 있다.

[참고문헌]

- [1] Claudio L. Lucchesi, "Application of finite Automata Representing Large Vocabularies", SOFTWARE-PRACTICE and EXPERIENCE, vol. 23(1), pp. 15-30, Jan, 1993.
- [2] 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.
- [3] S.W Smoliar and H.J. Zhang, "Content-Based Video Indexing and Retrieval," IEEE Multimedia, vol. 1, no. 2, pp. 62~72, Summer 1994.
- [4] A. Yoshitaka, S. Kishida, M. Hirakawa and T. Ichikawa, "Knowledge-Assisted Content-Based Revival Multimedia Databases," IEEE Multimedia, vol. 1, no. 4, pp. 12~21, winter 1994.
- [5] R. Weiss, A. Duda and D. K. Difford, "Composition and Search with a Video Algebra," IEEE Multimedia, vol. 2, no. 1, pp. 12~25, Spring 1995.
- [6] Y. Tonomura, A. Akutsu, Y. Taniguchi and G. Suzuki, "Structured Video Computing," IEEE Multimedia, vol. 1, no. 3, pp. 34~43, Fall 1994.
- [7] H. J. Ahang, 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.
- [8] 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.
- [9] H. J. Zhang, A. Krukanhali and S. W. Smoliar, "Automated Partitioning of Full_Motion Video," ACM Multimedia Systems, ACM Press, New York, vol. 1, pp. 10~28, 1993.
- [10] Chua, Tat-Seng, Issues in Hypermedia Research, Multimedia Technology and Applications, ELLIS HORWOOD LIMETED, 1989.