

# 오토마타를 이용한 가변재생 비디오 브라우징 시스템

김기원\*

\*초당대학교 컴퓨터학과  
e-mail:kwkim@chodang.ac.kr

## A Variable Playback Video Browsing System using Automata

Gi-Weon Kim\*

\*Dept of Computer Science, Cho-Dang University

### 요약

본 논문의 연구 목적은 효율적인 비디오 브라우징에 관한 연구이다. 비디오 스트림을 효율적으로 저장 및 재생하는 브라우징 시스템의 기본 구조로는 결정적 유한 오토마타를 사용하였다. 인덱싱 단계에서는 기존의 비디오 정보 또는 오디오 정보 한가지만을 사용하여 분야별로 특수하게 모델링하는 방법을 개선하여 비디오 스트림과 오디오 스트림 모두를 이용하는 새로운 인덱싱 방법을 제시하였다. 즉, 일차적으로 비디오 정보를 이용하여 동영상 데이터를 장면 단위로 분할한 후, 오디오 정보를 이용하여 에피소드 단위로 분할하여 인덱싱을 실행하였다. 이를 결정적 유한 오토마타 형태의 논리적인 구조로 에피소드들을 재구성하여 사용자의 요구에 따라 가변적인 재생 시간을 갖는 비디오 브라우징 시스템을 설계하였다.

### 1. 서론

오늘날 멀티미디어 시스템에 있어서 핵심 목표는 어떻게 대량의 비디오, 오디오 데이터를 실시간에 검색, 재생하느냐는 것이기 때문에 대용량의 비정형 데이터에 대한 미디어간의 시간 관계성 표현 및 효율적인 검색 및 재생 분야에 대한 연구가 필수적으로 요구된다. 이는 비디오 내용을 기반으로 하는 자동화된 인덱싱 기술, 검색, 재생 기술 등이 필요하게 됨을 의미한다. 따라서 이 분야에 대한 연구가 활발히 진행되고 있으나, 다양한 동영상에 대한 정확한 인덱싱 방법은 존재하지 않으며 비디오 정보 또는 오디오 정보 한가지만을 사용하여 분야별로 특수하게 모델링하여 개발하고 있는 실정이다. 또한 현재의 연구방향은 장면경계로 구분되는 장면단위의 검색 방법으로 제한되고 있으며, 장면들간에 의미적으로 연관성이 있는 에피소드 단위의 인덱싱 및 검색 방법에 대한 연구는 아직 미비한 실정이다.

따라서 멀티미디어 데이터의 효율적인 이용을 위해서는 비디오와 오디오 두 가지 데이터의 특징을 고려한 적절한 인덱싱 기법이 필요하며, 나아가 보

다 효과적이고 융통성 있는 검색 및 브라우징을 위해서는 에피소드 단위에 기반하는 검색 방법이 제안되어야 한다.

본 논문에서는 위에서 기술한 데이터 처리 시스템들이 갖는 문제점을 해결하는 방안으로 비디오 및 오디오 정보를 이용하여 에피소드를 검출하여 이들을 의미적으로 분류하는 방법을 제시하며, 이를 기반으로 사용자의 요구에 따라 재생 시간이 각기 다른 비디오 브라우징 시스템을 설계하여 현재의 VOD가 갖는 단점인 고정된 재생시간 문제를 해결하는 방안을 제시한다.

### 2. 시스템의 설계

#### 2.1 비디오 정보를 이용한 장면 경계 검출

본 논문에서는 밝기 히스토그램 법을 이용하여 장면 경계 검출을 하며, 처리 시간의 단축을 위한 방법을 제시한다. 프레임간의 밝기 히스토그램의 절대 차를 이용하는  $III_{diff}$ 은 다음의 (식-1)과 같이 계산된다[2,3,4].

$$IH_{diff}(t) = \sum_{b=0}^{B-1} |h(t+\Delta t, b) - h(t, b)| \quad (\text{식-1})$$

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

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

장면 경계 검출시 처리속도를 향상시키기 위하여, 동영상 데이터가 갖는 기본 특성을 사용한다. 즉, 하나의 장면 경계가 검출된 후에는 일정 시간동안에는 다음 장면이 나타나지 않는다는 경험적 지식을 장면 경계 검출에 이용한다. 이 방법은 프레임 개수를 나타내는 하나의 값을 두어 이 값만큼의 프레임을 건너뛰어 경계를 검출하는 방법이다. 즉, 장면 경계가 검출된 이후 다음 장면 경계 검출시, 비교적 많은 수의 프레임을 건너뛰고 나서 다음 처리로 들어간다.

### 2.2 오디오 정보를 이용한 에피소드 검출

일반적으로 내용의 변화가 있는 에피소드 경계에서는 오디오의 데이터들도 같이 바뀌게 된다. 따라서 에피소드의 경계 지점에서는 앞쪽 장면의 오디오 데이터와 뒤쪽 장면의 오디오 데이터에는 많은 차이가 발생하게 된다. 그러므로 서로 의미있는 장면들의 모임인 에피소드를 검출하는 데에 오디오 정보를 이용할 수 있다.

본 논문에서는 오디오 데이터 특징 파라미터로 에너지와 영교차를 기초로 하며[7], 실험에 사용된 AVI 파일 형식에 적합한 형태로 변형하여 에피소드 구간을 검출한다. 오디오의 특징을 나타내는 파라미터들은 PCM 부호화 과정에서 사용된 샘플링 값과 모노 또는 스테레오를 결정하는 채널값에 따라 각기 다른 계산 방식을 갖게 된다.

(식-2)는 오디오 데이터의 특징 파라미터인 변형된 형태의 에너지를 구하는 식이다. 이 식은 채널이 모노인 경우에 해당하며, 만약 스테레오인 경우에는 이와 같은 과정을 두 채널에 적용시켜 평균을 구하면 된다.

$$E_{diff} = \left| \frac{e_{before}}{length} - \frac{e_{after}}{length} \right| > T \quad (\text{식-2})$$

$$e_{before} = \sum_{i=cut-length}^{cut-1} e_{f(i)}, \quad length : \text{No. of frame}$$

$$e_{after} = \sum_{i=cut}^{cut+length-1} e_{f(i)}$$

$$e_{f(i)} = \sum_{j=frame\_start}^{frame\_end} e(j), \quad \text{if } h \leq e(j) < h+\Delta h$$

여기서 e(j)는 하나의 프레임에서의 시간 j에 대한 에너지를 나타내며, e<sub>f</sub>(i)는 i번째 프레임에 대한 구간 h와 Δh 사이에 있는 에너지의 총합을 구하는 식이다. 따라서 e<sub>after</sub>는 장면 경계를 기준으로 경계 프레임 cut부터 시작하여 length-1 길이 만큼의 프레임에 대한 구간 h와 Δh 사이에 있는 에너지 e<sub>s</sub>(i)의 합을 구하는 것이며, 유사하게 e<sub>before</sub>는 경계 이전의 cut-1 프레임부터 계산에 적용할 범위인 프레임 길이 length까지에서 구간 h와 Δh 사이에 있는 에너지 e<sub>f</sub>(i)의 합을 구하는 것이다. E<sub>diff</sub>는 경계 이전의 구간과 이후의 구간에 대한 에너지의 평균의 차를 구하는 것이 되는데, 이를 위해 우선 두 개의 구간에 대한 에너지 합을 구한 뒤 이를 장면 개수로 각각 나누어 절대 차를 구하게 된다.

(식-3)은 본 논문의 오디오 파라미터인 변형된 형태의 에너지를 구하는 식이다.

$$Z_{diff} = \left| \frac{z_{before}}{length} - \frac{z_{after}}{length} \right| > T \quad (\text{식-3})$$

$$z_{before} = \sum_{i=cut-length}^{cut-1} zcr_f(i)$$

$$z_{after} = \sum_{i=cut}^{cut+length-1} zcr_f(i)$$

$$zcr_f(i) = \sum_{j=frame\_start}^{frame\_end} \frac{|s(e(j)) - s(e(j+1))|}{2}$$

s(x)는 에너지 x 값이 최대 샘플값의 1/2보다 큰 경우 1이 되며, 작은 경우에는 -1 값이 된다. zcr<sub>f</sub>(i)는 현재의 프레임 i에 대한 영교차를 구한다. 따라서 z<sub>before</sub>는 장면 경계를 기준으로 경계 이전의 프레임들에 대해 영교차를 구하는 것이며, z<sub>after</sub>는 장면 경계부터 시작하여 일정 길이인 length 만큼의 프레임에 대한 영교차를 구한 것이 된다. Z<sub>diff</sub>는 경계 이전의 구간과 이후의 구간에 대한 영교차의 평균의 차를 구하는 식이 된다.

### 2.3 에피소드의 검출

장면의 경계 지점에서 앞쪽 장면의 오디오 데이터와 뒤쪽 장면의 오디오 데이터의 정보를 이용하여 에피소드를 검출할 수 있다.

에피소드를 검출하는 과정은 다음과 같다. 앞에서 계산된  $E_{diff}$  값과  $Z_{diff}$  값이 특정 임계값을 넘는 경우 두 장면은 동일한 에피소드에 포함시키지 않으며, 임계값보다 작은 경우 두 장면을 하나의 에피소드에 포함시킨다. 논 논문에서 제시한 에피소드 검출 알고리즘은 다음과 같다.

*Algorithm Episode\_Detection*

*(long cut, int frame\_length)*

*begin*

```

long before_energy, before_zcr;
long after_energy, after_zcr;
long temp_energy=0, temp_zcr=0;
long frame;
long start, end;
long episode[100];
static int idx=0;
start=cut - frame_length;
end=cut;
for (frame=start; frame<end; frame++) do
    temp_energy+=energy(frame);
    temp_zcr+=zcr(frame);
end for
before_energy=temp_energy/frame_length;
before_zcr=temp_zcr/frame_length;
start=cut;
end=cut + frame_length;
for (frame=start; frame<end; frame++) do
    temp_energy+=energy(frame);
    temp_zcr+=zcr(frame);
end for
after_energy=temp_energy/frame_length;
after_zcr=temp_zcr/frame_length;
if (abs(before_energy-after_energy) >
    energy_threshold) & (before_zcr-after_zcr
    > zcr_threshold) then
    episode[idx++]=cut-1;
    episode[idx++]=cut;
end if
return(episode);

```

*end*

#### 2.4 오토마타를 이용한 에피소드 저장 구조

현재의 VOD 시스템은 재생되는 전체시간이 고정되어 있기 때문에 사용자가 임의로 스트림의 재생 시간을 변화시킬 수가 없다. 이때 다양한 사용자의 요구를 충족시키기 위해서는 재생 시간이 각기 다른 동일한 내용의 비디오 스트림들을 중복 저장하여 몇 개의 스트림을 제공하여야 한다. 한 예로 특정 교육에 관련된 VOD 시스템의 경우 수강자의 수준이나 요구에 따라 난이도 및 재생시간이 각기 다른 재생 서비스를 제공해야 하기 때문에 동일한 교과 과정 콘텐츠에 대해서도 몇 개의 콘텐츠를 제공해야 하는 비효율적인 결과를 초래한다.

본 논문에서는 이와 같은 단점을 보완하여 한편의 비디오 스트림을 장면들간의 상호 연관성을 기준으로 재구성한 뒤, 사용자의 재생시간 요구에 따라 가변적인 재생 시간을 갖는 비디오 브라우징 모델을 제안한다. 재생시간 요구에 따라 융통성 있게 스트림을 재생하는 모델을 구성하기 위해서, 다음과 같은 형태의 결정적 유한 오토마타 DFA (Deterministic Finite Automata)를 이용한다.

$$DFA M = (Q, \Sigma, \delta, q_0, F)$$

$Q$ : 연관성이 있는 장면들의 유한 집합(에피소드)

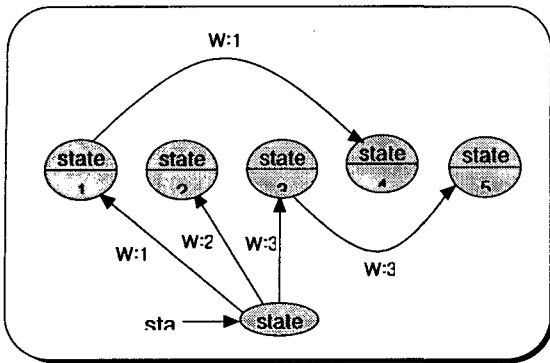
$\Sigma$ : 상태들의 중요도에 따른 가중치

$\delta$ : 가중치에 따른 전이 함수

$q_0$ : 시작상태

$F$ : 종결상태

위의 오토마타 정의를 기초로 다음과 같은 과정을 거쳐 [그림-1]과 같은 결정적 유한 오토마타를 구성한다. 우선 하나의 동영상 내에 존재하는 각각의 독립된 에피소드를 검출한다. 그후, 각각의 에피소드들에 대해 하나의 상태(state)를 생성한다. 그후, 각 상태내의 에피소드 내용 중요도를 나타내는 가중치를 이용하여 가중치가 레이블(label)인 지시선(directed arc)으로 연결한다. 이때 현재 상태의 가중치와 동일한 가중치 값을 갖는 상태가 다수 발생할 수 있다. 이 경우 결정적 유한 오토마타의 형태를 갖도록 하기 위해서 동일한 가중치 값을 갖는 상태들 중, 첫 번째 나타나는 상태에 대해서만 지시선으로 연결한다.



[그림-1] 비디오 스트림에 대한 DFA의 표현

### 2.5 비디오 스트림의 가변시간 재생

본 논문에서는 에피소드들의 내용 중요도를 기반으로 상태전이를 하는 결정적 유한 오토마타 구조로 비디오 스트림을 저장하였기 때문에 사용자의 다양한 재생시간 요구에 따라 가변적인 재생시간을 제공할 수 있게 된다.

만약 이를 교육용 비디오에 적용하면, 수강자의 수준에 따라 난이도를 단계별로(weight=1,2,3)로 조정하여 난이도에 따라 각기 재생 내용이 다른 서비스를 제공할 수 있다. 즉, 초급과정의 경우 입력 가중치를 1로 주게 되는데, 이 경우 가중치 레이블이 1인 지시선으로 상태전이(state transition)를 한다. 중급과정의 경우에는 가중치 레이블이 2인 상태로 상태 전이를 한다. 그리고 고급과정의 경우 입력 가중치를 3으로 하여 상태전이를 실행한다.

사용자의 난이도 요구에 따라 각기 다른 상태 전이를 하는 비디오 스트림 재생 알고리즘은 다음과 같다.

*Algorithm Video\_Browsing1(int weight)*

```

begin
  queue state_queue=initial_state;
  queue current_state=initial_state;
  while (current_state is not final_state) do
    if (exist arc of weight ) then
      current_state= δ(current_state, weight);
    else
      break;
    end if
    append current_state to state_queue;
  end while

```

```

for (each state in the state_queue) do
  playback the episodes of the state;
end for
end

```

### 3. 결론

본 연구에서는 기존의 비디오 정보 또는 오디오 정보 한가지만을 사용하여 분야별로 특수하게 모델링하는 방법을 개선하여 비디오 스트림과 오디오 스트림 모두를 이용하는 새로운 인덱싱 방법을 제시하였다. 또한 검출된 에피소드들을 내용 중요도에 따른 오토마타 형태로 저장함으로써 사용자의 다양한 재생시간 요구에 따라 가변적인 재생시간을 제공할 수 있게 된다.

위의 연구결과로 비디오와 오디오 정보를 이용하여 의미적으로 유사한 장면들을 하나의 에피소드로 연결하는 방법을 제시하였으며, 가변 재생시간을 갖는 비디오 브라우저의 모델을 제시하였다. 이러한 연구는 현재 연구중이거나 상용화되고 있는 멀티미디어 콘텐츠 개발이나 주문형 비디오 시스템, 인터넷 방송 서비스 분야에 이용될 수 있다.

### 참고문헌

[1] A.G. Hauptmann and M.J Witbrock, "Story Segmentation and Detection of Commercial in Broadcast News Video", proceedings of the Advances in Digital Libraries Conferences, 1998.

[2] Kien A. Hua and JungHwan Oh, "Detecting Video Shot Boundaries up to 16 Times Faster", Proceedings ACM Multimedia, ACM press, pp.385-387, 2000.

[3] M. R. Naphade, A. M. Feman and et al, "A high performance algorithm for Shot boundary detection using multiple cues," Proceedings of ICIP, Oct. 1998.

[4] M. Yeung, B. L. Yeo and B. Liu, "Segmentation of Video by Clustering and Graph Analysis," Computer Vision and Image Understanding, June, 1998.

[5] J. You and H. Shen, "Fast Indexing and Searching for Content-based Image Retrieval", Visual Information Processing VII, SPIE Vol. 3387, 1998.

[6] 김재홍, 남종호, 박수용, "MPEG 시스템 스트림 상에서 오디오 정보를 이용한 장면 경계 검출 방법", 정보과학회논문지, 제 27권 8호, pp864-876, 2000

[7] 송정길, 이극, "고도 지능망용 Intelligent Peripheral의 음성 인식에 관한 연구", 한국전자통신 연구소, 최종연구보고서, pp. 18-19, 1994