웹 상에서의 비디오 세그멘테이션

박종암, 권영빈
중앙대학교 컴퓨터 공학과
japark@eudoramail.com, ybkwon@visionnet.cse.cau.ac.kr

Video Segmentation Techniques on Web

JongAm Park, Young-Bin Kwon
Dept. of Computer Science & Engineering, ChungAng University

요 약

이 논문은 웹에서 찾을 수 있는 비디오 포맷들에 대한, 간단하고 개선된 비디오 세그먼테이션 방법을 다룬다. 2개의 임계 값을 이용해서 효과적인 비디오 프레임간의 차이를 비교한다. 또한 개체의 이동과 같은 이유로 프레임 사이에 차이가 날 경우, 하지만 다른 비디오 세그먼테이션이라고 볼 수 없을 경우를 위해, 공간 정보를 이용한 방법과 전역 정보를 이용한 방법간의 균형점을 찾아 본다. 그렇게 하기 위해 전역적인 히스토그램은 적절한 크기의 작은 히스토그램으로 나뉘어 진다. 웹에서는 여러 가지의 비디오 포맷이 존재하기 때문에, 각 포맷과의 독립성을 위해 상위 수준에서의 프로세싱을 위주로 한다.

1. 서론

정지 영상에서의 세그먼테이션과는 달리 동영상에서의 세그멘테이션은 흔히 Camera Break 추출, 컷 추출 혹은 shot 추출 등으로 불린다. Apostolos Dailianas등의 정의[1]에 의하면 shot은 "카메라로 촬영을 시작해서 끝날 때까지 일련의 프레임"이다. 영상 DB가 점차 누적되어 그 양이 많아짐에 따라, 추후에 DB를 효율적으로 찾는 것이 필요해 진다. 지금까지의 연구를 보면 주로 MPEG이나 최근에는 MPEG2 상에서 이루어지며 그 방법도 기존의 단순한 픽셀 비교나 히스토그램 비교에서부터, 통계를 이용한 방법[2], 영상 내 객체들의 이동을 고려한 방법[3] 등 MPEG의 Macroblock을 이용한 방법까지 나와 있는 상태이다.

이상의 기존 연구들은 MPEG을 기반한 방법으로서, 인코더 특성에 의존을 한다. 웹에는 다양한 코덱의 동영상이 존재하며 상당수는 비공개된 방법을 사용하므로 코덱에 독립적인 방법을 사용하는 것이 좋다.

본 논문에서는 기존의 방법들을 웹에서 얻을 수 있는 동영상에 대해 적용해 보고, Sliding Window를 통한 비교에 비해 적은 비교를 해도 되는 방법을 제안하고, 전역 히스 토그램 법이 프레임간의 단순한 픽셀 비교법의 장점을 가질 수 있는 적절한 프레임 영상 조각의 개수를 휴리스틱한 방법으로 알아 보고자 한다. 이는 모션 감지와 같은 보다 발전된 방법을 이용하지 않고도 어느 정도 비슷한 결과를 얻을 수 있을 것이라는 생각과 실제로 인코더들이 interframe 압축을 하면서 key frame을 설정할 때 복잡한 방법을 쓸 필요가 없다는 것을 생각한다면 합리적인 방법이 될 것이다.

2. 웹 기반 동영상의 특징

웹 기반의 동영상은 다음과 같은 특징과 그에 따른 장단점을 가지고 있다.

- 코덱과 파일 구조의 다양성

MPEG을 비롯하여 sorenson, cinepak, indeo등 다양한 코덱이 사용되고 있으며, 파일 구조도 AVI, QuickTime등 다양하다. 또한 코덱은 비공개로 되어 있어, 해당 코덱에 최적화된 개발을 하기 어렵다.

- 상대적으로 적은 데이터량

웹에서의 상영을 위한 것이기 때문에 Low bit rate 를 갖는 경우가 많고, 프레임 이미지의 크기가 방송용 MPEG에 비해 작다

- 인코딩 방법에 따른 특성 변화

Intracoding에 의한 방법은 Intracoding에 의해

만들어진 영상보다 높은 임계값을 가지는 특성이 있다. 그러므로 영상에 따라 사용자가 임계값을 정해 주거나 자동으로 감지해 낼 수 있어야 한다. MPEG2 비디오도 1/4이 I frame만으로 구성되어 있다[4]고 알려져 있다.

3. 제안된 방법

3.1 조각화 된 히스토그램

히스토그램을 이용한 다양한 방법은 전역적인 정보를 이용한다. 그러므로 사람의 육안으로 관찰할 때, 다른 영상임에도 히스토그램으로 보면 같게 보일 수 있다.[5] 반면에 Pixel 차이를 이용한 방법은 같은 장면으로 사람은 인식하지만 크기가 큰 객체가 영상 안에서 이동을 했을 경우에 Pixel의 차이가 많아지게 되므로 다른 장면으로 인식될 수 있다는 문 제가 있다.[5] 그러므로 전역적인 히스토그램이 물체의 이동에 민감한 Pixel 간의 비교를 이용한 방법의 장점을 얻을 수 있도록 John Chung-Mong Lee 등이 시도한 방법[5]과 유사하게 주어진 프레임들을 가로 X개, 세로 Y개의 영역으로 나누어 각 영역들의 히스토그램을 비교하고, 그 중 주어진 임계값 보다 큰 결과를 보이는 영역의 개수를 또 다른 임계값과 비교하는 추가적인 이중의 비교를 통해, 두 방법의 장점을 통합하고 단점을 줄였다. 앞 2절의 2번째 조건으로 히스토그램의 단점인 처리 데이터량이 많다[6]는 것은 그 비중이 줄어든다. 하지만 통계적으로 의미 있는 한 영역 당 최소 픽셀 수 확보를 위해 적정선 이하로 영상을 분할하는 것은 바람직하지 않다. 이 과정은 다음과 같은 알고리즘으로 표현할 수 있다.

```
Do until no more frame remained
  Compare Each Corresponding Histogram Slice
  If ( result > threshold value )
    Then  Current  Region  that  Current
Histogram Slice represents is marked as
"changed" or "different"
```

3.2 프레임간 2차 비교(Smart Method)

프레임간의 변화가 긴 시간에 걸쳐 점차적으로 이루어질 때, 육안으로는 다른 장면으로 인식될 수 있는 장면이 나올 수 있다. 하지만 이전 프레임간의 차이만을 비교한다면 i번째 프레임과 i+1번째 프레임간의 차이는 미미하여 i+1이 다른 장면으로 검지 되지 않을 수 있다. 이를 해결하기 위해 Sliding Window 방법[7]이나 Twin comparison[8] 등이 사용되고 있다. 본 논문은 1차 비교는 이전 프레임과 이루어지고, 2차비교는 가장 최근에 감지된 장면과의 비교만으로 방법을 단순화하였다.

현재의 고려 대상 프레임인 i+5 프레임은 우선 i+4와 비교된 후, 그 차이가 임계값보다 작을 경우 가장 최근에 감지된 장면 프레임인 i와 비교된다. i+3, i+2, i+1과 비교가 될 필요가 없는 이유는, 그들과 i+5와 비교를 했을 때의 결과는 i+5 프레임이 주어진 임계값 차이 보다 크거나 작게 되는 데, 어느 한 경우라도 클 경우는 i 프레임과의 비교만으로도 크다. 이를 수식으로 일반화하여 나타내면 다음과 같다.

Where $\Delta(i+m, i+k) > threshold$,

$$\sum_{j=0}^{m-1} \Delta(i+j, i+j+1) = \Delta(i+m, i) > threshold$$

Where $\Delta(i+m, i+k) < threshold$, for all k,

$$\sum_{j=0}^{m-1} \Delta(i+j, i+j+1) = \Delta(i+m, i) > threshold$$

or < threshold

When $\Delta(i+m, i) < threshold$,

frame i + m is not a scene cut

When $\Delta(i+m, i) > threshold$,

frame i + m is detected as scene cut

by second trial

$\Delta(a, b)$는 프레임 a, b 간의차이

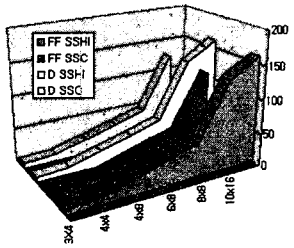
k는 0부터 m -1까지 임의의 정수

즉 모든 경우에 임계값보다 작을 경우, 장면 전환된 것이 아니거나, 서서히 장면 전환이 되어 i 프레임과의 차이가 나는 두가지를 모두 감지해 낸다. 그러므로 twin threshold를 사용하는 효과를 냄과 동시에, 경우에 따라 임계값을 조금 부적절하게 선택을 하여도 한번 더 해 시도를 해서 추출해 낼 수 있는 여지를 제공하는 장점이 있다.

4. 실험 및 결과

적절한 히스토그램 조각의 개수를 찾아내고, 최소 통계적 의미를 갖는 넓이의 조각을 알아보기 위하여 실험은 히스토그램 조각의 개수를 변화시켜 행해졌다. 단 사용하는 히스토그램 방법에 따라 결과가 많이 달라지고, 또 그 특성도 다르므로 각 방법에 대한 적절한 임계값을 달리해야 한다. 이 실험에서는 X^2은 Nagasaka와 Tanaka의 방법[9]을 사용하였다. 하지만 젯수를 H_{i+1}대신 max(H_i, H_{i+1})을 사용하는 결과가 더 좋다는 Apostolos Dailianas 등의 논문[1]을 참고하였다. 이를 이용하여 젯수가 0이 되지 않게 되는 추가적인 효과가 있었다.

실험에 사용된 영상은 줌, 페이드인 등 다양한 장면이 나오며 급격한 영상 변화와 점진적인 영상 변화가 같이 있는 Final Fantasy(FF)의 예고편(480x260), 그리고 비슷한 색 기조로 이루어지고, 점진적인 변화 위주로 구성된 Dinosaur(D) 예고편(280x272)이다. 또한 데이터 량이 많아 실험의 시간을 줄이기 위하여, MPEG의 I 프레임과 유사한 Sync 프레임의 특성 상 전체 프레임 대신 Sync 프레임만을 이용해도 무방하다는 결론 하에 Sync 프레임을 대상으로 하였다. 또한 Final Fantasy와 같은 비디오에서는 화질의 향상을 위해 Sync 프레임을 많이 가지고 있어, Sync 프레임 내에서도 점진적인 변화, 줌 인, 페이드 인/아웃 등의 관계가 있는 프레임이 많이 있으므로 실험의 목적에 위배되지는 않았다. 그림 1에서 보이는 바와 같이 영상을 나눌수록 프레임간 편차가 커지는 이유는 나누어진 영역에서만



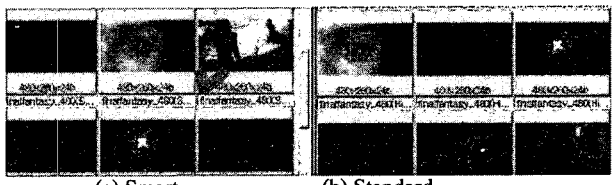
SSHI : Smart Sliced Histogram Intersection
SSC : Smart Sliced Chi Square

그림 1. 평균 프레임차

히스토그램이 구해지므로 평균화되는 정도가, 더 넓은 영역보다는 작기 때문이다. 평균 히스토그램 변화를 기준 임계값으로 주고 실험했을 때, 결과는 3x4에서 6x8정도와 10x16이상으로 잘라 주었을 때는 별 차이가 없었다. Final Fantasy를 3x4로 잘랐을 때, clear cut은, 비조각화된 히스토그램 교집합법의 평균 프레임차 52를 주어 나온 결과와 비교 해 보았을 때, 추가 7개 zoom/panning/blending은 추가 11개 오인식은 전체 4개로 12x16의 7,22,6과 크게 차이가 없었다. Dinosaur와 같은 경우 평균 64에서의 결과 대비 3x4에서 31,7,0, 12x16에서 33,7,0이고 오히려 중간이 8x8에서 35, 13, 0개가 각각 나왔다. Chi Square인 경우 Final Fantasy의 경우 비조각 평균 22 대비 3x4에서 5,22,10, 10x16에서 5,26,12이고, Dinosaur인 경우 평균 39 결과 대비 3x4에서 34,6,1, 10x16에서 34,6,1였다. 3x4와 12x16사이는 사람이 판단하는 것과 비슷한 수준의 컷 영상을 찾아 내는 결과를 보이며, 그보다 더 잘랐을 때는 수행시간은 $256(xy - x'y')$ 만큼 길어지지만 결과는 큰 차이가 없었다. 실험시 invariance를 찾기 위해 1차와 2차 임계값을 고정해 주어 조각을 나눌 때 마다 각 조각들의 히스토그램 값의 의미가 달라 지므로 단순히 좋은 결과를 얻기 위해서는 1차 2차 임계값을 그때마다 다르게 해 주어야 하지만, 실험에서는 조각을 나누는 효과를 보기 위해 고정 시켰다. 결과를 분석해 보면 그렇게 많이 조각을 나누지 않아도 3x4와 6x8사이로만 나누어도 좋은 결과를 보이며, 1차 2차 임계값까지 적절하게 주어 준다면 더 좋은 결과를 보일 수 있음을 알 수 있었다. 그 이유는 의도적이지 않는 한 다른 영상이지만 유사한 히스토그램 분포를 갖는 경우가 적기 때문이다. Dinosaur에서는 천둥이 치는 장면에서 같은 영상이지만 천둥 때문에 히스토그램이 많이 변화를 가져 영상을 작게 나누거나 나누지 않더라도 컷으로 구분될 수 밖에 없는 경우가 있었다. 이것은 Spatial 방법과 Global 방법의 장점이 모두 효과가 없는 경우로써, pattern matching을 이용해야 할 것이라는 결론을 얻었다.

그림 2 blending에 의한 장면 전환(조각화)

프레임간의 2차 비교는 임계값을 부적절하게 큰 정도로 주어도 그림 3의 (a)와 같이 어느 정도 보정을 해 주는 효과가 있었으며, 이것으로 점진적인 변화, Zoom, blending효과 등을 검지해 내는데에 그림 2에서 보이는 조각화의 효과와 더불어 도움을 주었다.



(a) Smart (b) Standard

그림 3. Histogram Intersection 결과
(threshold t = 60, avg. diff = 52)

프로그램은 Windows상에서 QuickTime SDK v.4.1.2를 이용하여 구현되었으며, avi, mov 등의 구조와 해당 파일 구조에서 지원하는 각종 코덱 그리고 DV, OpenDML, SDP, Macromedia Flash, AutoDesk Animator(FLC)등을 지원한다.

5. 참고 문헌

[1]. Robert B. Allen, Paul England, A. Dailianas, "Comparison of automatic video segmentation algorithms," In Integrations Issues in Large Commercial Media Delivery Systems, volume SPIE 2615, pages 2--16, 1995.

[2]. Ishwar K. Sethi and Nilesh V. Patel "A statistical Approach to Scene Change Detection,"SPIE proc., Storage and Retrieval for Image and Video Databases III, Vol. 2420, Feb. 1995, San Jose, California

[3]. Nilesh V. Patel and Ishwar K. Sethi "Video Shot Detection and Characterization for Video Databases,"Pattern Recognition: Special Issue on multimedia, 1996

[4]. Ramin Zabih, Justin Miller, Kevin Mai "A Feature Based Algorithm for Detecting and Classifying Scene Breaks," Proc. ACM Multimedia 95, San Francisco, CA, pp. 189-200, Nov. 1995

[5] John Chung-Mong Lee, Dixon Man-Ching IP "A Robust Approach for Camera Break Detection in Color Video Sequence,"Technical Report HKUST-CS95-14, 1995

[6] Farshid Arman, Arding Hsu, Ming-Yee Chiu "Image Processing on Compressed Data for Large Video Databases,"ACM Multimedia, 1993

[7] www.cse.chuhk.edu.hk/~csc5280/segment/vs.html

[8] H.J. Zhang, A. Kankanhalli, S.W. Smoliar "Automatic Partitioning of Full-Motion Video," ACM/Springer Multimedia Systems, Vol. 1, No. 1, 1993, pp. 10-28.

[9] A. Nagasaka, and Y. Tanaka "Automatic Video Indexing and Full-Video Search for Object Appearances,'" *Visual Database Systems, II* , Eds. E. Knuth, and L.M. Wegner, Elsevier Science Publishers B.V., 1992 IFIP, pp. 113-127