

장면 전환점 검출을 위한 프레임의 평균오차 비교에 관한 연구

°김단환*, 김형균*, 오무송**

*조선대학교 컴퓨터공학과

e-mail:danhwan@hanmail.net

A Study on Frame of MSE Comparison for Scene Chang Detection Retrieval

Dan-Hwan Kim*, Hyeong-Gyun Kim*, Moo-Song Oh**

*Dept of Computer Engineering, Cho-sun University

요 약

대용량의 동영상 데이터 이용에 있어 사용자가 전체 동영상 데이터를 한눈에 파악할 수 있고, 필요한 경우 원하는 지점부터 동영상을 재생할 수 있도록 하기 위하여 동영상 데이터의 정보를 요약해 놓은 프레임 리스트를 제공하며, 효과적인 동영상 검색을 위해서는 동영상 데이터의 색인과정이 필요하다. 본 논문은 내용기반 색인에 기초가 될 동영상의 장면 전환점 검출에 관한 효과적인 방법을 제안하고자 한다. 제안된 방법은 동영상 데이터를 대각선 방향으로 일정 픽셀의 칼라 값을 추출하여 동영상의 전체 구조를 파악할 수 있도록 정지영상으로 샘플링 하였으며, 샘플링 된 데이터는 장면전환점을 한눈으로 파악할 수 있었다. 각각의 프레임에서 추출한 픽셀의 칼라 값은 행렬A에 $i \times j$ 행렬로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장하고 MSE(Mean Square Error) 도입하여 각 프레임의 평균 오차를 계산한다. 평균오차와 일정 임계값을 초과하면 그 프레임을 장면 전환점으로 검출하고자 한다.

ABSTRACT

User in video data utilization of high-capacity can grasp whole video data at a look. Offer frame list that summarize information of video data to do so that can remake video from branch that want when need. Need index process of video data for effective video retrieval. This treatise wishes to propose effective method about scene change point detection of video that is been based on contents base index. Proposed method video data so that can grasp whole structure of video detection color value of schedule pixel for diagonal line direction in image sampling do. Data that get into sampling could grasp scene change point on one eye. Color value of pixel that detection in each frame is i frame number by $i \times j$ procession to procession A, j stores to reflex height of frame. Introduce MSE and calculate mean error of each frame. If exceed mean error and schedule critical value, wish to detect the frame for scene change point.

1. 서 론

컴퓨터를 사용하는 통신 및 멀티미디어 기술의 발전과 인터넷의 대중화로 현 사회 전반에 있어서 디지털화된 멀티미디어 정보 전달에 대한 효율적인 관리의 중요성이 증대되고 있다. 멀티미디어 자료 중에서 특히 동영상 자료는 데이터의 양이 방대하며 대역이 제한된 통신네트워크를 정보 전달 매체로 사용도가 증폭되고 있어 그 정보에 대한 검색, 전송 및 처리 할 수 있는 방법의 개발이 매우 중요하며 이에 대한 많은 연구가 수행되고 있다.

본 연구에서는 동영상 데이터를 한눈에 확인할 수 있도록 각 프레임 별로 영상데이터의 특정영역을 샘플링 하였다. 대용량 동영상 데이터를 소용량 정지영상으로 변환하는 것으로 연속된 각 프레임들의 대각선 방향 일정 픽셀 값을 추출하여 정지영상으로 샘플링함으로써 한눈에 장면전환점을 볼 수 있도록 하였다. 각각의 프레임에서 추출한 픽셀 값은 행렬A에 $A(i, j)$ 행렬로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장하고 MSE (Mean Square Error) 도입하여 각 프레임의 평균 오차를 계산한다. 평균오차는 일

정 임계값을 초과하면 그 프레임을 장면 전환점으로 추출하는 방법으로 기존에 연구된 방법보다 우수한 장면 전환 지점을 검색하는 방법을 제시 하고자 한다.

1-1. 동영상의 구조

동영상 데이터의 일반적인 구조를 살펴보면 그림 1과 같다. 동영상은 약간씩 다른 화상을 빨리 겹쳐 봄으로써 구현을 하게 되는데, 1초당 약 50~60개의 정지영상 필름을 뿌려 주게 된다. 이때 이 필름 한 장에 해당하는 하나의 정지영상을 frame이라 하고, 이것은 동영상 자료를 구성하는 최소 단위이다. 동영상을 구분할 때의 기본 단위로 사용하는 shot은 필름이 끊기지 않고 연결된 frame으로 구성되어 있다. 그리고 이러한 shot이 하나의 주 대상을 연속적으로 촬영한 집합을 scene 이라고 한다. cut은 shot과 shot 사이의 경계에 해당하는 장면 전환점을 말하는 것으로 동영상 검색의 단위로 쓰인다.

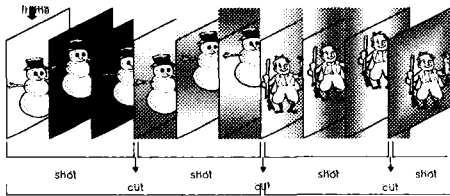


그림 1. 동영상의 구조

- Frame : 동영상 데이터를 구성하는 최소 단위로 각각의 픽셀로 이루어진 일종의 정지영상. 필름 한 장에 해당하는 하나의 정지영상
- Shot : 동영상을 구분 할 때의 기본단위로서 필름이 끊기지 않고 연결된 프레임들의 집합으로 구성
- Scene : 하나의 주 대상을 촬영한 연속된 Shot들의 집합으로 이루어진 공간상이나 시간상의 의미적 분리로 구성
- Cut : Shot과 Shot사이의 경계에 해당하는 장면 전환점

1-2. 일반적인 장면 전환지점 검출 기법

① 픽셀차이(pixel difference)를 이용한 방법
 Zhang, Kankanjili, and Smoliar등[1]은 동영상 장면 전환 검출을 위해서 각 프레임의 픽셀값을 이용하여 프레임간의 픽셀값의 차이는 매우 낮은 값을 갖는다. 그러나 장면 전환이 있는 곳은 픽셀값들의 차이가 두드러지게 커지므로 이를 이용하면 장면 전환점을 찾아 낼 수가 있다. 이러한 방법은 프레임의 잡음(noise)에 민감

하고 특히 점진적인 장면 전환 기법은 검출이 거의 되지 않는 취약점을 보인다.

② 명도히스토그램(histogram)을 이용한 방법

이 방법은 가장 널리 사용되는 장면 전환 기법 중의 하나로 프레임의 명도 히스토그램을 찾아 프레임간의 히스토그램의 차이를 비교하여 일정한 임계값 이상이면 장면전환으로 인식하는 기법으로 픽셀차이를 이용하는 방법보다는 잡음(noise)에 강한 성질을 갖는다. 하지만 이 방법 또한 점진적 장면 전환을 찾는 것에는 취약점을 보인다.

③ DCT 계수를 이용한 방법

압축영상(MPEG)의 일반적 방법인 DCT 변환의 계수를 이용하는 방법이 있다. 이는 압축영상의 스트림에 존재하는 계수의 차이를 이용하는 방법으로 압축영역에서 복원에 필요한 많은 계산을 거치지 않고 바로 장면 전환을 검출하는 특징을 갖고 있다

II. 제안한 검색 시스템

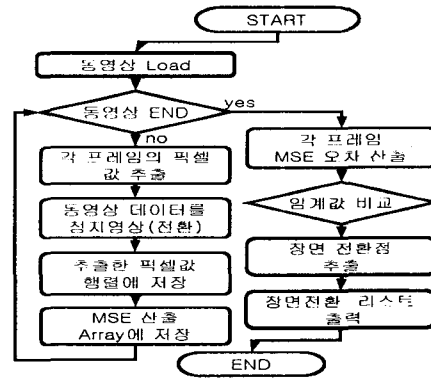


그림 2. 장면 전환검색의 Flowchart

동영상은 장시간에 연속적인 프레임들이 서로 유사한 배경 구성과 이미지들의 변형으로 이루어진다. 따라서, 영상의 공간정보는 연속성이 끊어지는 부분이 검색 대상이 되며, 그 영상이 장면 전환점이 될 수 있다. 본 연구에서는 일반적인 장면 전환 기법과 장면 전환의 특수효과인 점진적인 장면 전환 기법을 검출하는데 중점을 둔다. 이를 위하여 동영상 데이터의 각 프레임들은 대각선 방향으로 추출한 픽셀의 칼라 값을 행렬A에 ixj 행렬로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장한 내용은 그림3과 같다.

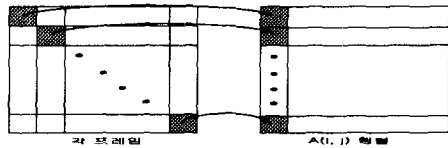


그림 3. 프레임의 추출한 픽셀값 저장

동영상의 각 프레임에서 추출된 데이터를 소용량의 정지영상으로 샘플링한 결과는 그림 4에서와 같이 샘플링된 영상에서 장면전환점은 수직선상의 픽셀 값들의 확연한 변화가 나타나게 되며, 점진적인 장면전환점은 픽셀 값들의 점진적인 변화로 나타나게 된다. 이러한 영상 데이터는 여러 프레임간의 간격으로 샘플링 함으로써 그 크기를 줄일 수가 있다. 또한 동영상 데이터의 전체 구조를 파악할 수 있다는 특징을 가진다.



그림 4 동영상 데이터 샘플링 결과

2-1. 프레임의 평균오차

동영상의 각 프레임들의 대각선 방향으로 추출한 값들을 행렬 A의 열성분 a_i 와 a_{i+1} 의 차이가 큰 i 열이 장면 전환이 일어나는 곳이다. 행렬 A의 열성분 a_i 과 a_{i+1} 의 차이를 정의하기 위하여 MSE를 이용한다.

$$E_1 = \sqrt{(a_{11} - a_{12})^2 + (a_{21} - a_{22})^2 + \dots + (a_{i1} - a_{i2})^2} \quad (1)$$

식 (1)에서 E_1 은 영상 데이터 행렬 A의 1열 a_1 과 2열 a_2 의 MSE이다.

$$E_k = \sqrt{(a_{1(k-1)} - a_{1k})^2 + (a_{2(k-1)} - a_{2k})^2 + \dots + (a_{i(k-1)} - a_{ik})^2} \quad (2)$$

식 (2)에서 E_k 는 영상 데이터 행렬 A의 $k-1$ 열 a_{k-1} 과 k 열 a_k 의 MSE이다. 그러면 식(2)에서 얻어진 E_k 을 이용하여 장면 전환점 f 를 찾을 수가 있다.

$$|E_{f-1} - E_f| \geq t \quad (3)$$

식(3)에서 t 는 장면전환점을 찾는 임계값이고, E_f 는 전환이 일어나는 프레임 번호를 나타낸다.

2-2. 제안한 검색 알고리즘

```

; 동영상 데이터 추출
for(i=1 to frame_x축 값)
  for(k=1 to frame_y축 값) {
    StoreA[i][k]= //frame의 대각선 방향의
    데이터 픽셀 값 추출;
    Image[i][k]= //frame의 대각선 방향의
    데이터 샘플링;
  }
; 추출한 데이터의 MSE 구함
for(f=1 to frame_end) {
  for(g=1 to frame_end){
    hap = hap + (StoreA[g][f] - StoreA[f][g])^2
  }
  MSE_Tab[f]=SQRT(hap); //E1의 MSE 구
  함
; 임계값과 비교
for(f=1 to frame_end) {
  Chang[f]=MSE_Tab[f]
  MSE_Tab[f+1]=
    //각 프레임의 오차
  if(Chang[f])= 임계값)
    Chang_Image[] = f; //동영상 데이터저장
}
    
```

IV. 실험 및 결과

본 연구의 실험은 영상처리시간의 단축을 위해서도 사용자의 간섭 없이 특징 추출 전과정이 빠르게 진행되어야 하기 때문이다. windows 2000 Sever 환경에서 RAM은 256Mb, VGA 카드는 GeForce2MX, 언어는 visual C++ 6.0을 이용해 구현하였으며, 실험에 사용한 자료는 사회에서 사용자가 접할 수 있는 동영상을 분류하면 무한히 많다. 이렇게 많은 동영상 중에서 본 논문에서 광고, 뉴스, 드라마 세 가지 데이터를 동일한 환경에서의 성능 측정을 하기 위하여 AVI 파일 포맷을 이용하였다.

동영상 종류	프레임 크기	프레임 수	장면전환점의 수	파일 포맷
광고	320*320	480	95	AVI
뉴스	320*320	625	196	
드라마	320*240	236	13	

그림 5는 검색 시스템 초기화면으로 화면 좌측에 검색을 위한 동영상이 로드 되고, 임계값을 입력한 후 검색 시작 버튼을 누르게 되면 장면전환점 검색이 시작된다. 이렇게 구해진 프레임은 중간화면에 검색 프레임의 정보와 리스트

를 출력한다. 또한 임계값은 검출된 장면전환점의 수가 바뀌므로, 장르에 따라 최적화된 임계값으로 변화시켜 사용하면 더 나은 결과를 얻을 수 있다. 실험에서 사용된 임계값은 최대 크기를 1로 정규화 시킨 수치이다.

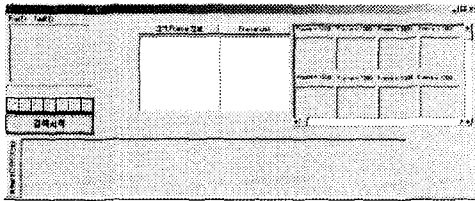


그림 5 검색 시스템 초기화면

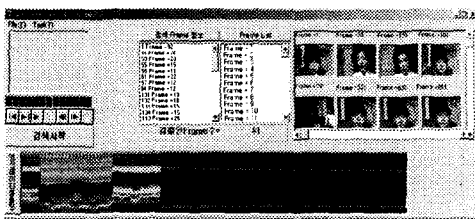


그림 6 검색 결과 화면

본 연구에서는 세 가지 유형의 동영상 데이터에 대하여 실험을 하였다. 광고, 뉴스, 드라마 각기 화면 구성에 특성을 지니고 있기 때문에 본 연구가 모든 유형의 동영상에 효율적인지 판단하기 위하여 다양한 데이터를 이용하였다. 먼저 각각의 동영상에 대한 장면전환점을 검색하여 비교해 본 후 정확도까지 비교한다. 다음 표는 동영상 데이터를 장면전환점 검색에 대한 정확도와 재현율을 나타내고 있다.

$$Precision(정확도) = \frac{\text{검출된장면전환점}}{\text{총존재하는장면전환점}}$$

정확도	픽셀 값	히스토그램	MSE 이용
광고	63.16 (60/95)	61.05 (58/95)	84.21 (80/95)
뉴스	80.61 (158/196)	88.78 (174/196)	91.84 (180/196)
드라마	61.53 (8/13)	76.92 (10/13)	84.61 (11/13)

$$Recall(재현율) = \frac{\text{정확하게검출된장면전환점}}{\text{검출된장면전환점}}$$

재현율	픽셀 값	히스토그램	MSE 이용
광고	81.67 (49/60)	77.59 (45/58)	88.75 (71/80)
뉴스	89.24 (141/158)	77.01 (134/174)	97.22 (175/180)
드라마	60.00 (6/10)	70.00 (7/10)	81.81 (9/11)

본 연구에서 비교의 대상으로 선정한 픽셀에 의한 검출, 히스토그램에 의한 검출은 현재 장면전환점 검출에 보편적으로 이용하는 방법이기 때문에 본 연구에서도 비교에 대상으로 선정하였다. 결과에서 확인할 수 있듯이 광고영상에서는 다른 방법에 비해 7% 정도 우수한 검출율을 보여주고 있다. 단지 영화 화면에서의 검출율이 다른 방법과 비슷한 결과를 보이고 있다. 검출 비율은 전체 장면전환점 수를 검출된 프레임 수로 나누어서 구하였다.

V. 결론

컴퓨터를 사용하는 통신기술의 발전과 인터넷의 대중화로 인하여 디지털화된 멀티미디어 자료에 대한 검색, 전송 및 처리 환경의 수요가 증가하고 있다. 그러므로 멀티미디어 정보에 대한 검색, 전송 및 처리 할 수 있는 방법의 개발이 매우 중요하다.

본 논문에서 각 프레임을 대각선 방향으로 영상 데이터를 추출하고, 대용량의 동영상 데이터를 정지영상으로 샘플링하여 동영상의 전체 구조를 파악할 수 있고 장면 전환점을 한눈에 확인할 수 있다. 각 프레임에서 추출한 픽셀 값은 행렬A에 ixj 행렬로 i 는 프레임 수, j 는 프레임의 영상 높이로 저장하고 MSE(Mean Square Error)도입하여 각 프레임의 평균 오차를 계산한다. 평균오차와 일정 임계값을 초과하면 그 프레임을 장면 전환점으로 검출하였다.

본 실험에서는 세 가지 종류의 동영상 데이터를 이용했는데 동영상도 각각의 내용에 따라 표현하는 방법이 틀리기 때문에 비교의 대상으로 설정하였다. 기존의 방법(픽셀 값에 이용한 방법, 히스토그램에 이용한 방법)과 비교해 보았을 때 검출율에서 20%정도의 효율적인 결과를 보였으며, 정확도면에서 10%정도 향상됨을 확인할 수 있었다.

향후 연구 방향은 비디오 인덱싱을 하기 위해서는 단순한 장면전환뿐만 아니라 카메라 특수효과 등과 같은 복잡한 장면전환도 완벽하게 검출하는 방안에 대한 연구가 계속되어야 할 것이다.

참고문헌

- [1] C.L. Fennema, and W.B. Thompson, "Velocity Determination in Scene Containing Several Moving Objects", Computer Graphics and Image Processing, Vol. 9, No. 4, 1979, pp. 310-315.
- [2] W. Niblack et al. The Qbic project Querying Images, by content using color, texture, and shape. In SPIE 1908, Storage

- and Retrieval for Image and Video Databases, Feb. 1993.
- [3] Hampapur, A., Jain, R., and Weymouth, T., "Digital Video Segmentation", Proc. ACM Multimedia 94, San Francisco, CA, pp.357-364, October, 1994,.
- [4] A. Nagasaka and Y. Tanaka, "Automatic Video Indexing and Full Motion Search for Object Appearance", Visual Database System, II, Eds, E. Knuth, and L.M Wegner, Elsevier Science Publishers B.V., 1992 IFIP, pp. 113-127.
- [5] F. Arman, A. Hsu, and M.-Y. Chiu, "Image Processing on Compressed Data for Large Video Databases", Proc. 1st ACM Intl. Conf. on Multimedia, Anaheim CA, August 1993, pp. 267-272.
- [6] J. Meng, Y. Juan, S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence", Proc. IS&T/SPIE, Vol. 2419, February 1995.
- [7] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar, "Automatic Partitioning of Full-Motion Video", ACM/Springer Multimedia Systems, Vol. 1, No. 1, 1993, pp. 10-28
- [8] John. R. Smith, "Integrated Spatial and Feature Image Systems : Retrieval, Analysys and Compression", Submitted in partial fulfillment of the requirements for the degree of Doctor of philsophy in the Graduate School of Arts and Sciences, Columbia University, 1997.
- [9] S. Berretti, A. Del Bimbo, P. Pala, "Sensations and Psychological Effects in Color Image Database", IEEE International Conference on Images Processing, pp.560-563, 1997.