2000年度 第13回 信號處理合同學術大會 論文集 第13卷 1號

Nearest Neighbor 클러스터링 방법을 이용한 비디오 스토리 분할

이해만, 최영우*, 정규식
숭실대학교 정보통신전자공학부
숙명여자대학교 전산학과*

Video Story Segmentation using Nearest Neighbor Clustering Method

Haeman Lee, Yeongwoo Choi*, Kyusik Chung
School of Electronic Engineering, Soongsil Univ.
Computer Science Dept., Sookmyung Women's Univ.*

요약

비디오 데이터의 효율적인 검색, 요약 등에 활용하기 위해서 대용량의 비디오 데이터를 프레임(Frame), 샷(Shot), 스토리(Story)의 계층적인 구조로 표현하는 방법들이 요구되고 있으며, 이에 따라 비디오를 샷, 스토리 단위로 분할하는 연구들이 수행되고 있다. 본 논문은 비디오가 샷 단위로 분할되어 있다고 가정한 후, 인접한 샷들을 결합하여 의미 있는 최소 단위인 스토리를 분할하는 방법을 제안한다. 제안하는 방법은 각 샷에서 추출된 대표 프레임들을 비교하기 위한 CCV(Color Coherence Vector) 영상 특징을 추출한다. CCV 특징의 시각적인 유사도의 초기 임계값과 일정한 시간 안에 반복되는 프레임들을 찾기 위한 시간적인 유사도의 시간 임계값을 설정하여 NN(Nearest Neighbor) 클러스터링 방법을 이용하여 클러스터링을 한다. 클러스터링된 정보와 같은 장면이 한번이상 반복되는 스토리의 특성을 이용해 비디오를 스토리로 분할한다. 영화 비디오 데이터를 이용한 실험을 통해 제안하는 방법의 유효성을 검증하였다.

1. 서론

최근 인터넷의 발달로 네트워크를 통한 멀티미디어 서비스에 관한 관심이 고조되고 있다. 그런데 이에 접속하는 사용자들의 환경은 천차만별이다. 따라서 각각의 사용자 환경에 적합한 비디오 데이터를 전달 할 수 있는 방법이 필요한데, 이를 위해서는 비디오 데이터를 분석하고 분석된 정보들을 적합한 형태로 저장하며, 이를 각각의 사용자 환경에 맞게 표현하는 방법들이 필요하다[1].

비디오 데이터는 그 양이 방대하기 때문에 좀더 관리하기 쉬운 형태로 분할하여 구조화할 필요가 있는데 기존의 연구에 사용된 비디오는 대부분 프레임, 샷, 스토리, 비디오의 4계층으로 구성되는 경우가 대부분 이었다[2]. 이러한 구조를 바탕으로 비디오 분할에 관한 많은 연구들 중 초기연구들은 하나의 카메라 동작이 끝나고 다른 카메라로 넘어가기까지의 프레임의 집합인 샷을 검출하는 연구들이 대부분 이었다. 하지만 샷 자체만으로는 비디오의 내용을 파악하는 것은 어렵기 때문에 비디오의 내용을 빠르게 파악하기 위해서 의미를 가지는 최소 단위인 스토리를 분할하는 연구가 필요하게 되었다.

스토리는 각 샷에 포함된 공간적인 정보와 시간적인 정보를 이용하기 때문에 압축된 데이터보다는 압축되지 않은 데이터에서 클러스터링을 이용한 연구들이 대부분이었다. Xuesheng[3]등이 제안한 방법은 비디오를 샷으로 분할 한 후 각 샷마다 대표 프레임(Key Frame)을 추출해 이들을 나열한 후 공간적인 정보 측정 방법인 Similarity Length를 구해 가장 작은 값을 가질때 최적화된 경우로 보고 이때의 배열을 찾아내는 방법이다. 이 방법은 시간적인 정보를 고려하기 않고 단지 공간적인 정보만을 사용하였다. Yeung[4,5]등은 Time Constrained Clustering을 위한 방법으로 모든 샷의 대표 프레임에서 가장 가까운 프레임을 찾아 이 두 대표 프레임의 시간적인 거리가 임계치 이하이면 같은 클래스로 묶고, 이 정보를 이용해 스토리를 찾는 방법으로서 클러스터링에 사용될 공간적인 임계치와 시간적인 임계치가 적응적이지 못하다는 단점이 있다. Kender[6]등이 제안한 방법은 샷의 모든 프레임을 이용해 시간적으로 떨어진 정도에 따라 두 샷의 관련정도를 Short term memory model을 통해 나타낸 방법으로 공간적인 임계치와 시간적인 임계치가 적응적인 반면에 샷의 모든 프레임을 사용함으로써 처리 시간이 오래 걸리고, 스토리의 경계인 Local Minimum값을 찾아야 하는 번거로움이 있다.

본 논문에서는 스토리를 빠르게 분할하기 위해 샷에서 대표 프레임을 추출하고 추출된 대표프레임들을 주위에서 가장 비슷한 것과 같은 클래스로 묶는 NN(Nearest Neighbor) Clustering 방법으로 클러스터링을 한다. 그리고 클러스터링 된 정보와 스토리의 반복적인 특성을 이용하여 의미 있는 최소단위인 스토리를 효과적으로 분할하는 방법을 제안하고자 한다.

2. 제안하는 방법

일반적으로 비디오 데이터의 양이 많으므로 스토리를 찾는 속도를 빠르게 하기 위해서 비디오에서 대표 프레임들만을 가지고 처리했다. 비디오의 각 샷에 속한 프레임들을 보면 대부분 거의 비슷한 프레임으로 구성되어 있다. 하지만 카메라 또는 물체가 움직인 경우에는 프레임간의 차이가 크다. 따라서 본 논문에서는 비디오가 샷으로 분할 되어있고 샷에서 카메라나 물체의 움직임 정보도 주어졌다고 가정하여 카메라나 물체의 움직임이 있는 부분에서는 여러 개의 대표프레임을 추출하고, 그 외의 샷에서는 각 샷의 처음 프레임을 대표 프레임으로 추출해 스토리를 찾았다. 이렇게 추출된 대표 프레임들을 가지고 시간적인 유사도와 공간적인 유사도를 고려하여 NN(Nearest Neighbor) Clustering방법을 이용해 클러스터링 했다. 공간적인 유사도는 CCV특징을 이용하였고, 시간적인 유사도는 Time Window를 사용했다.

2.1 특징 추출

2.1.1 색 줄임

영상에 쓰이는 색의 수는 매우 많다. 24bit인 경우에는 2^{24}개의 CCV의 bin이 생겨 메모리를 낭비하게 되고 CCV들간의 차이를 구할 때도 시간이 많이 걸리게 된다. 그래서 RGB 모델에서 각각에 대해 양자화에 의해 색을 줄인다. 하위보다는 상위의 비트들이 더 많은 색 정보를 포함하고 있으므로 RGB각각의 상위 3bit, 3bit, 2bit의 정보만을 사용하여 256개의 색으로 줄인 영상을 가지고 CCV를 구한다. 양자화 공식은 식 1에 나와 있다.

$$\text{Color} = (R,G,B) \rightarrow \text{Color}_q = (\left\lfloor \frac{R}{64} \right\rfloor, \left\lfloor \frac{G}{32} \right\rfloor, \left\lfloor \frac{B}{32} \right\rfloor) \quad \text{(식 1)}$$

2.1.2 시간적인 유사도(Time Similarity)

같은 장면이라 하더라도 시간적으로 떨어져 있을 경우 이는 다른 내용 즉 다른 스토리가 된다. 따라서 클러스터링 할 때 현재 대표 프레임의 이전 T개의 대표 프레임만을 비교하는 것으로써 같은 내용이 반복될 수 있는 최대 시간적인 정도를 나타내는 것이다.

2.1.3 공간적인 유사도(Visual Similarity)

비디오 프레임간 색의 비교로 주로 사용되는 것이 Pixel-by-Pixel difference와 Histogram difference가 있다. Pixel-by-Pixel difference는 두 프레임의 같은 위치의 색 값의 차이를 구하는 방법으로 움직임이 작은 영상에서는 잘 적용되지만 물체나 카메라의 움직임이 큰 경우에는 성능이 저하된다. 그리고 Histogram difference는 각각의 영상의 같은 색의 화소 개수를 비교하는 방법으로 빠른 움직임이나 회전등에는 강하지만 전환된 두 장면이 비슷한 칼라 분포나 밝기 분포를 가지면 성능이 저하되는 단점을 가지고 있다. 이와 같은 Histogram difference의 전역적인 특징과 Pixel_by_Pixel difference의 지역적인 특징을 모두 포함하는 방법으로 CCV[7](Color Coherence Vector)방법이 있다. 이는 각 Color bin에 대해 같은 색을 가지는 영역의 화소의 개수가 임계치보다 크면 Coherence부분으로 작으면 Incoherence 부분으로 나눔으로서 Pixel-by-Pixel difference의 지역적인 특성과 Histogram의 전역적인 특성을 모두 지니게 된다. 두 대표 프레임 n 과 n+1에

서 CCV차이를 구하는 방법은 아래의 식 2와 같다.

$$CCV\_D_{(n,n+1)} = \sum_{i=1}^{B} \begin{pmatrix} |ccv_{n+1}(i,\alpha) - ccv_n(i,\alpha)| \\ +|ccv_{n+1}(i,\beta) - ccv_n(i,\beta)| \end{pmatrix} \quad \text{.....(식 2)}$$

이 차이 값은 매우 큰 값을 가지므로 0과 1사이 값을 가지도록 정규화 시킨다. CCV difference 에서 가질 수 있는 최대 값은 두 프레임의 CCV에서 공통적인 bin이 존재하지 않는 경우이므로 최대 값은 $2 \cdot N \cdot M$이 되며 이 값을 이용해 정규화 시킨다.

$$NCCV\_D_{(n,n+1)} = \frac{CCV\_D_{(n,n+1)}}{2 \bullet N \bullet M} \quad \text{.....(식 3)}$$

2.2 시간·공간 유사도를 이용한 NN Clustering

클러스터링을 통해 비슷한 내용을 포함하는 대표 프레임끼리 하나의 클래스로 분류한다. 이 때 사용되는 클러스터링 방법은 NN(Nearest Neighbor) Clustering 방법으로 현재의 대표 프레임과 이전의 T개의 대표 프레임들과 비교하여 가장 CCV차이가 적은 대표 프레임을 선택하고, 그 대표 프레임이 속한 Class의 CCV와 현재 대표 프레임의 CCV차이를 계산한다. 이 때의 CCV차이가 설정된 임계치보다 크면 현재 프레임을 포함하는 새로운 클래스를 만들고, 임계치보다 작은 경우에는 저장된 클래스가 속한 클래스에 현재의 대표 프레임을 포함하는 방법이다. 여기에서 T는 시간적으로 같은 내용이 반복될 수 있는 최대 길이를 의미하는 것으로 같은 장면이라 하더라도 시간적으로 멀리 떨어져 있으면 다른 스토리임을 나타낸다.

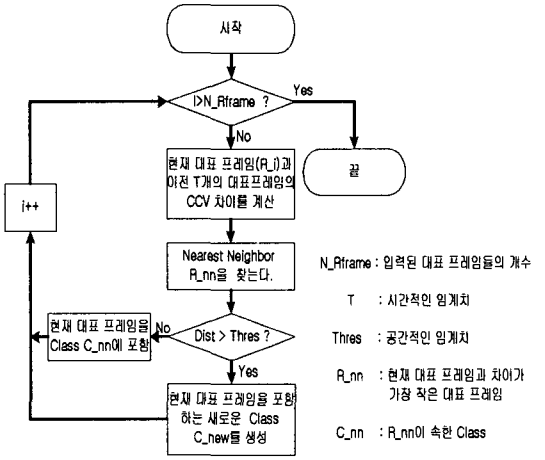

그림 1 NN Clustering

2.3 스토리 추출

Story Unit은 주로 같은 장소에서 같은 인물 혹은 같은 물체가 촬영되므로, 같은 Class가 반복적으로 나타나는 경우가 많다. 이와 같은 Story Unit의 반복적인 특성과 클러스터링 결과를 이용하여 반복적인 Story Unit을 검출한다. 반복적인 알고리즘은 Yeung[5]이 제시한 알고리듬을 사용했다.

< Algorithm I. Find Story Unit >

1) Set l ← m
   Set e ← last(L_1,m)
2) while l ≤ e do
   if(last(L_l,m) > e) e← last(L_1,m)
   l ← l+1
3) Shots s_m, s_{m+1}, ..., s_{m+e} constitute a story unit

그림 2 에서 처음 대표 프레임이 속한 Class가 A이므로 마지막으로 Class A가 나타난 지점 그림 3.3.에서 $e \leftarrow last(a,m)$이라 표시된 부분까지 하나의 스토리가 되는 것이다. 또 다음 대표 프레임을 보면 Class 가 B이므로 역시 마지막으로 Class B가 나타난 지점까지 하나의 스토리가 된다. 그런데 두 번째 대표 프레임은 첫 번째 스토리에 포함되므로 첫 번째 스토리가 Class B가 나타난 마지막 지점까지 되는 것이다.

그림 2 반복적인 스토리 검출

이외에 점층적인 Story 즉, 폭발 장면이나 자동차 추적 장면 같은 경우는 카메라가 사람이나 물체를 따라 다니며 긴 샷들이 나열되거나, 카메라가 계속 움직인 샷들이 나열되는 경우가 많다. 이 경우에는 Story Unit의 클러스터링 결과가 연속적인 경우가 많다. 이러한 연속 적인 특성을 이용해 반복적인 Story Unit를 찾은 후에 점층적인 Story Unit을 검출한다.

3. 실험 및 분석

본 논문에서 제안한 클러스터링을 이용한 스토리 추출 방법은 Pentium Ⅲ 450Mhz상에서 Visual C++6.0을 이용하여 구현하였다. 실험 데이터는 영화 '메리에겐 뭔가 특별한 것이 있다' DVD에서 Capture한 동영상을 샷으로 분할 한 후 각 샷에서 앞에 제시한 방법으로 대표

프레임을 추출한 후 이 대표 프레임 영상만으로 이루어진 avi를 실험에 사용했다. 실험에 사용된 영화의 길이는 1시간 1분 28초이며 여기에 포함된 샷의 개수는 613개 였고, 여기에서 카메라와 물체의 움직임 정보까지 이용해 추출된 대표 프레임의 개수는 713개였다. 실험에 사용된 스토리 추출 방법은 본 논문에서 제안한 세 가지 단계로 이루어져 있다. 1)색을 줄인 영상에서 CCV특징을 추출한다. 2) 추출된 CCV특징을 사용하여 대표 프레임들을 클러스터링한다. 3) 클러스터링된 결과를 이용해 스토리를 추출한다. 실제 실험에서 클러스터링에서 사용된 Visual Threshold와 Time Threshold는 각각 0.22 , 6이었다. 여기서 Visual Threshold는 두 대표 프레임의 차이를 나타내고, Time Threshold는 시간적인 거리를 나타낸다. 실험결과는 Correct, Split, Merge의 세 가지 경우로 나누어 살펴보았다. Correct는 정확히 맞은 것, Split은 두 개 이상의 스토리가 결합되어 분할되어야하는 것, Merge는 하나의 스토리가 여러 개로 나누어져 하나로 합쳐져야 하는 경우를 각각 나타낸다.

Correct	Merge	Split	합계
24	12	4	40

표 1 스토리 분할 결과

그림 3 정확히 분할된 스토리

그림 3의 경우가 Merge되어야할 경우인데 카메라의 위치가 급격히 변해 물체의 배경이 급격히 바뀜으로서 서로 다른 스토리로 분할된 경우이다.

그림 4 합쳐져야 할 스토리

또 그림 4에서는 두 개의 스토리가 하나로 묶여 Split되어야 하는데, 이는 어두운 장면이나 거의 비슷한 배경색을 가지는 부분들이 가까이 위치했기 때문이었다.

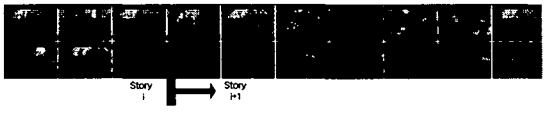

그림 5 분할 되어야할 스토리

4. 결론

본 논문에서는 빠르고 정확히 스토리를 분할하기 위한 알고리즘을 제안하고 있다. 처리 시간을 줄이기 위해 샷에서 대표 프레임만을 사용했고, 이 대표프레임들을 중심으로 CCV특징을 추출해 클러스터링의 특징으로 사용했으며, 이 결과를 가지고 스토리를 분할했다. 제안하는 방법의 빠르기와 정확성은 실험을 통해 확인 할 수 있었다. 향후 연구는 클러스터링에 적용되는 시간적 공간적인 임계치를 보다 적응적으로 바꾸기 위한 특징을 찾는 것이다.

5. 참고 문헌

[1] P. Aigrain, H. Zhang and D. Petkovic, "Content-based Representation and Retrieval of Visual Media: A State-of-the-Art Review", Multimedia Tools and Application 3, pp179-202 1996

[2] R. Lienhart, S. Pfeiffer and W. Effelsberg, "Video Abstracting" Communications of the ACM , V.40 N.12 ,pp 54-62 , 1997

[3] B. Xuesheng, X. Guangyou and S. Yuanchun, "Similarity sequence and its application in shot organization", IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases vii 1999.

[4] M. M.Yeung, "Time-Consrained Clustering for Segmentation of Video into Story Units ", Proceedings of the 13th International Conference on Pattern Recognition - Volume 3 , 1996

[5] M. M. Yeung and Boon-Lock Yeo, "Video visualization for compact presentation and fast browsing of pictorial content",IEEE Transactions on Circuits & Systems for Video Technology , V.7 N.5 , , 1997

[6] J. R. Kender and B. L. Yeo ,"Video Scene Segmentation Via Continuous Video Coherence", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 367-373 , 1998

[7] G. Pass, R. Zabih and J. Miller ,"Comparing images using color coherence vectors", Proceedings of the fourth ACM international multimedia conference on Proceedings ACM Multimedia 96 , pp 65-73, 1996.