보색개념을 도입한 Video Cut의 검출†

김재학° 박종승 한준희
포항공과대학교 전자계산학과 컴퓨터비젼 연구실

Video Cut Detection Using Complementary Color

J. H. Kim J. S. Park J. H. Han
Computer Vision Lab, Dept. of Comp. Sci. & Eng., POSTECH

요 약

Video영상을 의미있는 부분으로 나누는 video segmentation을 위해서는 video cut의 검출이 필요하다. 본 논문에서는 video cut의 검출을 위하여 신경망을 이용하였으며, cut의 측정방법으로 보색(complementary color)의 개념을 도입하였다. 이 방법을 이용하여, 여러개의 video data로 부터 학습을 한 뒤 새로운 video에 대해 테스트한 결과 좋은 성능을 보였다.

1. 서론

Video를 의미있는 조각으로 나누는 것은 지능적인 비디오 검색을 위해 매우 중요한 일이다. 조각(segment)은 하나의 끊기지 않는 카메라 샷(shot)으로 정의되고 비디오 컷(cut)은 조각들의 경계로 정의된다. 컷을 검출하는 것은 dissolve, wipe, fade, zooming, panning등의 카메라 기법들 때문에 더욱 어렵다. 그림 1은 video cut의 예를 보여주고 있다.

기존의 video cut방법으로는 영상 화소의 밝기 변화, 영상의 히스토그램 변화, 기타 통계적인 차이 등을 비교하는 방법을 사용하였다[1, 2, 3]. 이 방법들에 있어서는 영상에서의 물체의 영역이 배경의 영역보다 넓은 경우에 문제점들이 발견된다. 그리고 물체의 움직임이 통계치의 크기를 바꿀 경우에 잘못된 video cut이 발견된다. 또한 노이즈에 약하다는 문제점이 여전히 남아있다. 영상의 경계선 방향을 이용하는 경우에는 특정 threshold값을 주어야 하므로 비디오 영상마다 다른 결과가 나올 가능성이 있다. 정확한 video cut을 표현하는 수학적인 모델이 존재하지 않기 때문에 현재에도 다양한 측정모델들이 제시되고 있다.

본 논문에서는 신경망을 이용하여 주어진 비디오로부터 video cut위치를 찾는 방법을 기술한다.

2. 신경망을 이용한 video cut의 검출

먼저 주어진 비디오로부터 간단한 영상처리 기법을 사용하여 몇가지 특징을 추출한다. 특징으로는 연속되는 두 프레임 간의 밝기차, 히스토그램 분포차, 그리고 보색차를 사용하였다. 얻어진 특징들은 정규화된 후 신경망 모델의 입력노드에 각각 할당된다. 신경망 모델은 다층 퍼셉트론을 사용하였다. 출력노드는 2개로 첫번째 출력노드는 현재의 프레임에서 video cut이 발생되었을 경우 활성화되며 다른 출력노드는 그렇지 않은 경우에 활성화된다.

특징 추출 과정과 정규화 과정을 설명한다. 세가지 특징은 밝기차, 히스토그램 분포차, 보색차이다. 밝기차는 연속된 두 프레임간의 밝기변화가 클수록 이전 프레임에 대해 현재 프레임은 video cut일 가능성이 높다. 영상내의 각 화소는 3×3 평활화 필터를 사용하여 노이즈에 대한 영향을 줄였다. 밝기차에 대한 식은

$$P = \frac{1}{N}\sum_{i=1}^{N}|S_2(i) - S_1(i)|$$

이다. 여기서 $S_1(i)$는 이전 프레임에서 i번째 화소의 평활화된 밝기이며 $S_2(i)$는 현재 프레임에서의 i번째 화소의 평활화된 밝기이다.

히스토그램이란 한 영상내에서의 밝기에 대한 빈도수를 뜻한다. 영상내의 물체가 이동되더라도 video cut이 아니면 히스토그램의 변화는 적다. 히스토그램 분포차에 대한 식은

$$H = \frac{1}{N}\sum_{i=1}^{N}|H_2(i) - H_1(i)|$$

이다. 이기서 $H_1(i)$는 이전 프레임에서 히스토그램의 i번째 밝기의 빈도수이며 $H_2(i)$는 현재 프레임에서의 히스토그램의 i번째 밝기의 빈도수이다.

Video cut의 검출을 위해서 밝기만을 고려한다면 상당한 부분의 실제 video cut이 검출되지 않을 수 있다. 이것은 다른 색이지만 같은 밝기를 가질 수 있기 때문이다. 이의 해결을 위해서 RGB에 대한 여러 히스토그램들을 만들기도 하는데 이는 너무 많은 비용이 들며, 같은 계통의 색이라도 히스토그램에서 차이가 크게 나타날 수 있는 단점이 있다. 이의 극복을 위해서 본 논문의 방법에서는 색의 보색성질을 이용한다. 보색은 인간이 느끼는 가장 반대되는 색을 말한다. 예를들어 빨강색에 대한 보색은 청록색이다. 보색의 계산은 HSV 칼라모델[4]에서 hue의 값이 180도 반대인 것이다. RGB 영상을 HSV 영상으로 변환하여 한 화소에 대해 hue의 각도값이 얼마나 변화가 되었는가를 측정하여 video cut의 위치를 알 수 있다. 이것을 수식화하면 다음과 같다

$$C = \frac{1}{N}\sum_{i=1}^{N}|C_2(i) - C_1(i)| \quad (1)$$

2.1 특징값의 정규화 및 데이타 획득

계산된 특징값들이 균일하게 네트웍에 기억하도록 하기 위하여 다음의 식으로 정규화를 시킨다 [5].

† 본 연구는 한국과학재단 지정 지능자동화 연구센터로부터 연구비의 일부를 지원받았음

$\bar{x}_i = \frac{1}{N}\sum_{t=1}^{N} x_i^n, \sigma_i^2 = \frac{1}{N-1}\sum_{t=1}^{N}(x_i^n - \bar{x}_i)^2, \tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i}$

실험을 위한 동영상은 MOV(QuickTime Movie For Window) 또는 MPEG 양식으로 저장이 되어있다. 동영상으로 부터 각 프레임을 온라인 획득하는 프로그램을 SGI의 *DMultimedia Library*를 사용하여 제작하였다.

2.2 학습

입력노드가 3개, 출력노드가 2개이며 은닉노드는 10개인 MLP를 구현하였다. 구현은 SNNS패키지를 사용하였으며 은닉층의 활성화 함수로는 시그모이드 함수를 사용하였다. 그림 2는 학습모델의 도시이다.

실험에 이용된 동영상 데이타는 다음과 같다.

내용	프레임크기	frame/sec	frame 개수
오토바이경주	320 × 240	29.97	2998
에니메이션1	320 × 240	30.00	2701
에니메이션2	316 × 200	8.00	1137
축구경기1	320 × 240	29.97	1393
뮤직비디오	320 × 240	24.00	5205
축구경기2	320 × 240	29.97	11004
인덱스모음	169 × 120	25.00	100
TV 캡쳐	320 × 240	29.97	7111

네트웍을 학습하기 위한 목표값은 동영상을 플레이어로 보면서 supervisor가 video cut이 발생한 프레임을 지정하여 얻었다.

수집된 데이타는 video cut이 발생한 프레임에서의 3가지 특징값들과 목표값으로 이루어 진다. 일반적으로 한 동영상 내에는 video cut의 개수보다는 video cut이 아닌 프레임의 개수가 더 많기 때문에 네트웍을 위한 데이타로는 이들의 균형를 맞추어 주어야 한다. 먼저 video cut에 해당하는 프레임들의 특징값들을 뽑아낸후, video cut이 아닌 프레임들 중에서 video cut 개수만큼을 임의로 뽑아서 균형를 맞추었다.

SNNS에서 학습을 시켰을 때의 MSE 오차곡선이 감소하다가 갑자기 증가하는 순간까지를 epoch으로 결정하였다. 약 200 사이클 정도까지 학습을 하여 네트웍의 일반화(generalization)를 시켰다.

다음은 학습에 사용된 파라메타들이다.

사이클수	학습율	초기 가중치
180	0.01	-1.0...1.0

3. 실험및 결과

3.1 학습결과

학습결과 학습에러 MSE는 0.0238 이다. 테니스 경기영상을 사용하여 테스트를 하였다. Video cut의 검출 결과가 그림 3에 있다

학습에러 MSE	테스트에러 MSE
0.0238	0.0068

테니스 영상에서는 총 2477 프레임 중에서 11 프레임의 video cut이 존재하며 나머지 2465 프레임은 video cut이 아니다. 실험 결과 11개의 video cut이 검출 되었는데 이중 2개는 실제 video cut이 아니었다. 그리고 2개의 실제 video cut은 검출이 되지 않았다.

영상에서 video cut이 일어날때 영상들이 overlap되는 현상이 있다. 따라서 학습시에는 video cut이라고 여거지는 명백한 위치를 목표값으로 주었지만 네트웍이 학습한 뒤에는 video cut의 주위에서도 cut의 검출이 일어나는 현상이 나타난다. 이것은 자연스러운 현상으로 보통 연속적으로 2-3 프레임에 걸쳐 나타난다.

프레임 2220과 2271은 검출이 되지 않았다. 또한 완전한 에러로는 557번 프레임이 video cut으로 검출이 되었는데, 이것은 잘못된 결과이지만 프레임 2090번째와 2091번째에서는 볼 보이가 빠르게 카메라 앞을 달려가는 순간으로 이 경우가 video cut으로 인식이 되었다는 점은 주목할 필요가 있다.

3.2 성능비교

기존의 방법중에서 영상의 밝기 차이로만 계산하는 경우에는 신경망을 이용한 방법과 커다란 차이를 보이지는 않았다. 물론 때로는 기존의 방법을 사용하는 것이 약 93%를 추출을 하는데 신경망의 경우에는 90%에 못미치지는 경우도 있었다. 기존의 방법은 성능을 높이는 threshhold 값을 필요로 하므로 프로그램의 실행을 하면서 사람이 직접 좋은 성능의 값을 찾아야 하는 단점이 있을 뿐만 아니라, 한 동영상에서의 threshhold 값이 다른 동영상에 적용될 수가 없기 때문에 자동화를 하기에는 문제점이 있다. 신경망은 다수의 동영상으로 부터 학습하였기에 비록 성능이 떨어지는 경우가 있더라도 자동화에는 쉽게 적용 될 수 있다. 실험에서는 대부분의 video cut은 찾지만 잘못 추출하는 경우도 가끔 발생하였다.

4. 결론 및 토의

신경망을 이용하여 video cut 검출 시스템을 구현하여 보았다. 새롭게 도입한 측정방법으로 보색(complementary color)을 사용하여 인간이 느끼는 색의 변화에 대해서도 장면이 전환되는 상황을 설명하는 변수를 제시하였다. 또한, 기존의 방법에서는 변수가 문제에 의존되어 다른 영상에는 적용이 불가한 반면에 신경망의 특성상 여러 다수 영상에서 학습이 가능하므로 문제의 독립적인 video cut검출 방법이 된다.

현재 지적되는 문제점으로서 개인의 차나 환경에 따라 잘못된 데이타를 넣을 가능성이 있다. 또한 zooming, panning등의 카메라 기법을 video cut으로 간주하여 목표데이타를 입력하는 등의 오류에 대한 대책이 필요하다.

여러가지의 실험 예제를 웹페이지
http://falcon.postech.ac.kr/ipawb/demo/ 에서 자세히 볼 수 있다.

참고문헌

[1] B. Furht and M. Milenkovic, *A Guided Tour for Multimedia Systems and Applications* IEEE Computer Society Press, 1995.

[2] H Zhang, A. Kankanhalli, and S W. Smoliar, "Automatic partitioning of full-motion video," *in Multimeida Systems*, vol. 1, pp. 10–28, 1993.

[3] J. S. Boreczky and L. A. Rowe, "Comparision of video shot boundary detection techniques," in *Proceedings of SPIE*, pp. 170–179, 1996.

[4] F. at al., *Computer Graphics PRINCIPLES AND PRACTICE* Addision-Wesley, 2nd ed.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford.

그림 1: Video Cut: 축구경기 비디오

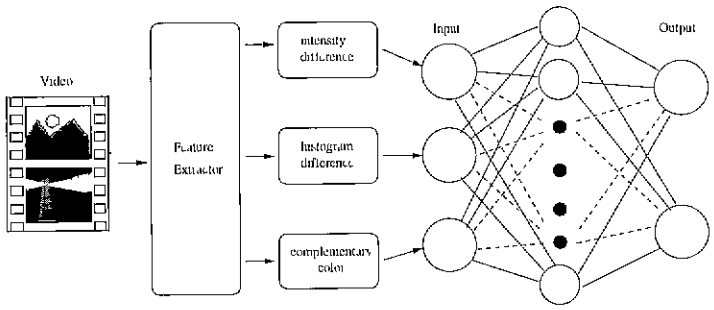

그림 2: 신경망 모델(MLP)

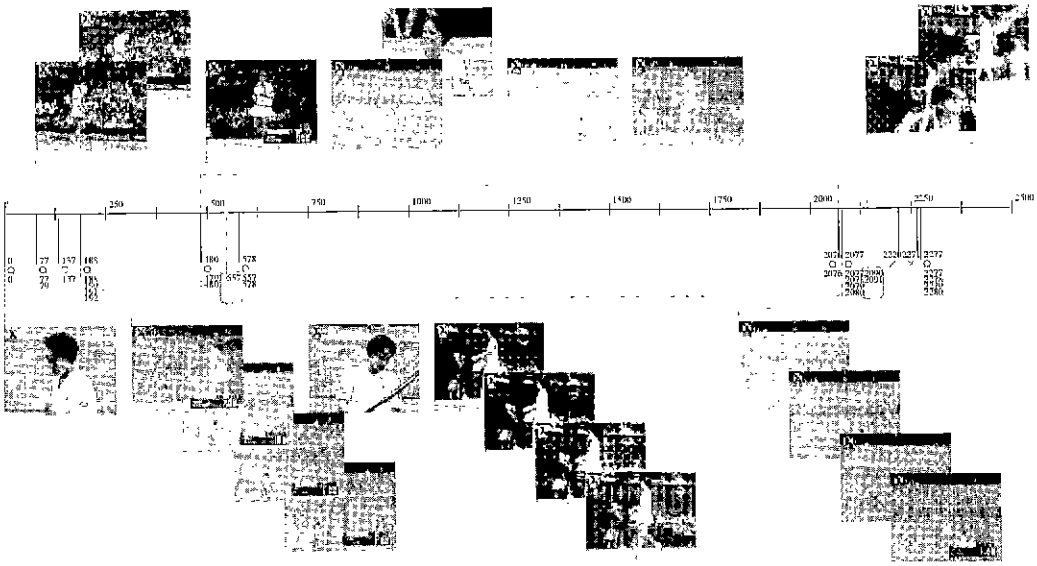

그림 3: Video Cut의 검출 결과