

# 칼라임계값을 이용한 동영상의 컷 검출에 관한 연구

이명주\*, 김형균\*, 정기봉\*, 오무송\*, 김태성\*\*  
조선대학교 컴퓨터공학과\*, 전남대학교 전기공학과\*\*

## A Study on Cut Detection of Video Retrieval using the Color Threshold

Myung-Ju Lee\*, Hyeng-Gyun Kim\*, Ki-Bong Joung\*,  
Moo-Song Oh\*, Tae-Seung Kim\*\*

Dept. of Computer Eng. Chosun Univ.\* , Dept. of Electrical Eng. Chonnam Univ.\*\*

### 요 약

동영상 정보는 영상정보뿐만이 아니라 음성정보, 문자정보 및 각종 의미있는 정보들을 포함하고 있어서 기존의 검색 방법으로는 사용자가 원하는 이미지를 찾는 데 어려움이 따른다. 따라서, 본 연구에서는 동영상 정보의 효율적인 활용을 위한 색인 방법으로 칼라 임계값을 이용한 컷 검출 방법을 제안하였다. 이것은 frame간의 유사도를 측정해서 이 값이 주어진 임계값 보다 작을 경우, 장면의 전환이 일어나는 곳을 컷 지점으로 검출하는 것인데, 동영상의 장면에 따른 유사도가 다를 수 있기 때문에, 컷을 구성하는 프레임들 간의 칼라 임계값에 변동을 주어 최적의 컷 검출율을 구하고자 했다. 초기의 칼라 임계값은 '80'을 사용했고, 이후 frame의 유사도가 임계값보다 클 경우 즉, 장면전환이 일어나지 않았을 경우 일정한 상수 값을 초기 임계값에서 감소토록 하였다. 이러한, 과정을 거쳐 추출된 frame을 가지고 원하는 이미지를 검색하게 되면 사용자의 노력 및 검색 시간이 단축되고 동영상 정보의 관리가 용이해 진다.

### 1. 서 론

정보통신 기술의 비약적인 발전은 멀티미디어 자료의 활용을 적극적으로 요구하게 되었다. 이러한 멀티미디어 정보 중에서도 동영상 정보는 인간에게 많은 정보를 매우 자연스럽게 제공해 줌으로써 관심의 대상이 되고 있다.

동영상 정보는 영상정보뿐만이 아니라 음성정보, 문자정보 및 각종 의미있는 정보들을 포함하고 있어서 기존의 검색 방법으로는 사용자가 원하는 이미지를 찾는 데 어려움이 따른다. 따라서, 최근에는 내용기반 검색 기법을 사용하고 있는데, 이것은 기존의 문자기반 검색의 단점을 보완하고자 물체의 모양, 질감, 색

과 같은 영상의 특징을 이용하여 검색하는 방법이다. 동영상의 내용기반 검색을 위해서는 video indexing 이 중요한 기술의 하나이다. 이러한 방법중의 하나로 히스토그램 방법이 제안되었지만, 히스토그램의 임계값이 영상 값의 분포도에 따라 변경되는 문제점을 가지고 있어서, 이 문제를 해결하기 위해 컷 검출 기법이 제안되었다. 이 기법의 기본 원리는 frame간의 유사도를 측정해서 이 값이 주어진 임계값 보다 작을 경우, 장면의 전환이 일어나는 곳을 컷 지점으로 검출하는 것이다.

따라서, 본 연구에서는 frame의 유사도를 측정하기 위하여 픽셀의 칼라 값을 임계값으로 사용하였고, 동

영상의 장면에 따른 유사도가 다를 수 있기 때문에, 컷을 구성하는 프레임들 간의 칼라 임계값에 변동을 주어 컷을 검출해 보았다.

본 논문의 구성으로는, 2장에서는 동영상 검색에 사용되는 기법에 관하여 설명하고, 3장에서는 본 연구에서 제안한 프레임간의 칼라 임계값 변동을 이용한 컷 검출 알고리즘에 대하여 설명하고, 실험 결과를 살펴보고, 마지막 4장에서는 결론을 다루게 된다.

## 2. 동영상 검색 기법

동영상 포맷으로 주요 사용되는 것으로는 .mov, .avi 가 있으며, 이와 조금은 구분되는 실시간 스트리밍을 해주는 포맷인 streamwork MPEG 포맷과 VDOLive 의 AVI 포맷등이 있다. mov 파일은 매킨토시 플랫폼을 이용하고 동영상등 시간적으로 변화하는 일련의 데이터를 취급하기 위해서 만든 포맷이며, avi 포맷 파일은 윈도우즈 환경에서 동영상과 애니메이션을 위한 소프트웨어 독자방식의 동영상 압축알고리즘을 이용한다. 이러한 avi나 mov의 포맷을 사용할 경우에는 동영상 플레이를 하는데 초점을 맞춘 데이터이므로 압축 효율이 MPEG 포맷에 비해서 떨어진다. MPEG 포맷을 사용할 경우 작게는 두배에서 많게는 4배까지 데이터를 압축해 준다. 이러한 고효율의 압축방식을 이용하여 인터넷상에서 모델을 이용한 동영상 실시간 전송 프로그램은 Xing tech 사의 StreamWorks 와 VDOnet 사의 VDOLive 가 대표적이다.

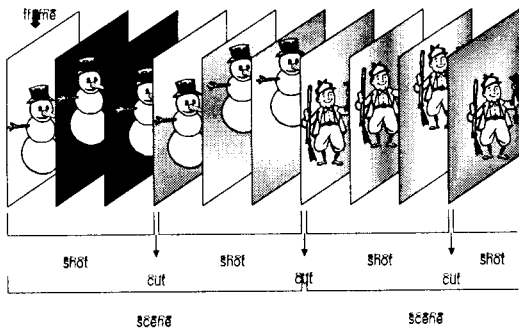


그림 1. 동영상의 구조

동영상은 약간씩 다른 화상을 빨리 겹쳐 보므로써 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 shot 이 하나의 주 대상을 연속적으로 촬영한 집합을 scene 이라고 한다.

cut은 shot과 shot 사이의 경계에 해당하는 장면 전환점을 말하는 것으로 동영상 검색의 단위로 쓰인다.

동영상은 연속된 frame의 집합이므로 연속된 장면에서는 인접한 frame사이의 유사도가 높고 장면의 전환이 발생하는 부분에서는 frame간의 유사도가 상대적으로 낮다. 따라서 동영상내에서 컷을 추출하기 위해서는 동영상을 구성하는 프레임간의 특징값을 이용하여 유사도를 측정하여 동영상의 연속성을 계산하고 유사도가 급격히 낮아지는 지점을 동영상의 불연속 지점으로 생각하고 이를 컷으로 판단 추출하게 된다.

frame의 특징값을 구하는 방법으로는 히스토그램의 차이 비교, 픽셀간의 차이 비교, 에지 변화 비교, 압축상관 계수 비교, 유사율 측정법 등이 있다.

본 연구에서는 픽셀간의 차이를 비교하는 기법을 사용했는데, 이것은 frame을 구성하는 픽셀들이 동일한 장면내에서 변화가 적다는 성질을 이용하여 인접 frame들간의 동일 위치의 픽셀들을 비교해 유사도를 측정해서 이 값이 임계값 보다 작을 경우 장면 전환이 있다고 보는 방법이다.

동영상의 각 frame을 바둑판 모양으로 잘게 잘랐을 때 각각의 작은 구역을 픽셀이라고 한다. 픽셀은 color 값을 가지게 되는데, 모든 색은 빛의 삼원색인 Red, Green, Blue 세 가지 색의 조합이므로, 픽셀의 값은 이 세 성분의 값을 기술함으로써 표기가 가능하다. cut 검출시 인접한 frame 간의 유사도를 측정하기 위하여 두 frame의 동일 지점 픽셀간의 color 값을 비교하였다.

## 3. 컷 검출 알고리즘

동영상 자료의 내용 기반 검색을 위해서는 데이터를 색인하기 위한 비디오 파싱기법과 사용자 인터페이스, 비디오 데이터 압축 및 저장 방법등의 기술들이 필요하다. 이중에서도 동영상 데이터의 색인 작업이 중요한 부분을 차지하고 있는데, 색인의 Key 가 되는 자료로 영상자료를 일반적으로 사용한다.

본 연구에서는 먼저, 영상자료의 색인을 위하여 컷 검출 기법을 사용하였는데, 이것은 프레임 간의 유사도를 측정해서 이 값이 주어진 임계값 보다 작을 경우, 장면의 전환이 일어나는 곳을 컷 지점으로 검출하는 것이다. 또한, 동영상의 장면에 따른 유사도가 다를 수 있기 때문에, 컷을 구성하는 프레임들 간의 칼라 임계값에 변동을 주어 최적의 컷 검출율을 구하고자 했다.

초기의 칼라 임계값은 '80'을 사용했고, 이후 frame의 유사도가 임계값보다 클 경우 즉, 장면전환이 일어나지 않았을 경우 일정한 상수 값을 초기 임계값에서 감소토록 하였다.

이는 동영상의 특성상 장면전환이 일어나기 전까지는 근접한 frame간의 유사도가 점진적으로 작아진다는 점에 근거를 두었다.

다음 그림은 본 연구에서 제안한 동영상 색인 시스템에서 동영상 정보의 각 프레임들 비교하여 불연속적인 부분을 감지해 컷으로 검출하는 알고리즘이다.

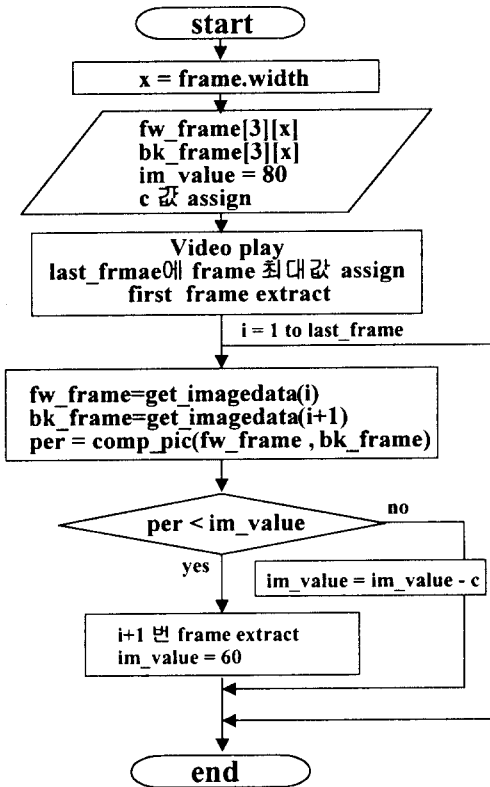


그림 2. 컷 검출 알고리즘

여기서 get\_imagedata함수는 load된 각 프레임들을 픽셀별로 비교하기 위하여 3차 형식으로 위치를 설정하고, 여기에 해당하는 칼라 값을 추출하여 메모리에 저장하는 기능을 한다. 이때, 비교를 위해 지정된 위치는 해당 프레임 상단, 중단, 하단의 배경 및 이미지의 변화를 감지하기 위한 설정이다.

```

get_imagedata(f_num)
{
  int f_num, cnt, img_x, img_y, img_m;
  img_x에 f_num frame의 Width속성 assign;
  img_y에 f_num frame의 Height속성 assign;
  img_m에 (img_y/2)정수화 값 assign;
  int clr_value[3][img_x];
  for cnt=1 to img_x
  {
    clr_value[1][cnt] = frame.Point[1][cnt];
    clr_value[2][cnt] = frame.Point[img_m][cnt];
    clr_value[3][cnt] = frame.Point[img_y][cnt];
  }
  return clr_value;
}
  
```

이렇게 추출된 칼라 값은 다음의 comp\_pic 함수를 사용해서, 인접한 두 프레임 간의 대응되는 픽셀을 비교하여 그 값을 리턴한다. 이때, 그 값이 일치하면 카운트를 증가시켜 유사한 픽셀 수를 구한다. 비교가 끝나면 유사한 픽셀 수의 평균을 구해 유사도를 산출하고 그 값을 리턴한다.

```

comp_pic(s_value, t_value)
{
  int cmp_x, cmp_y, cmp_m, cmp_cnt, cmp_loop, cmp_value;
  cmp_x에 frame의 Width속성 assign;
  cmp_y에 frame의 Height속성 assign;
  cmp_m에 (cmp_y/2)정수화 값 assign;
  int s_value[3][cmp_x], t_value[3][cmp_x];
  for cmp_loop=1 to cmp_x
  {
    if(s_value[1][cmp_loop]=t_value[1][cmp_loop])
      cmp_cnt=cmp_cnt+1;
    if(s_value[2][cmp_loop]=t_value[2][cmp_loop])
      cmp_cnt=cmp_cnt+1;
    if(s_value[3][cmp_loop]=t_value[3][cmp_loop])
      cmp_cnt=cmp_cnt+1;
  }
  cmp_value=(cmp_cnt/(cmp_x*3))*100;
  return cmp_value;
}
  
```

유사도 값 per가 산출되면 임계값 im\_value 와 비교를 한다. 이때 유사도가 초기 임계값('80') 보다 적지 않으면 초기 임계값에서 일정 상수값 C를 뺀다.

상수값 C는 { 0, 0.5, 1, 1.5, 2 } 다섯가지 경우로 테스트해 보았다.

다음 표는 실험자료를 대상으로 임계값에서 일정 상수 값을 뺀 경우 각각의 컷 검출 결과이다.

총 frame 수	C 값	Cut frame 수	Cut 검출율
975	0.0	37	3.8%
	0.5	42	4.3%
	1.0	48	4.9%
	1.5	60	6.2%
	2.0	65	6.7%

표 1. c값의 변동에 따른 cut frame 수

실험 자료를 눈으로 확인한 결과 cut frame은 약 51개로 5.2%의 cut 검출율을 보였다. 표 1에서 보듯이 C값을 '1.0'을 주었을 경우 확인된 결과와 비슷한 검출율을 보였다.

본 연구는 windows 98 환경에서 visual C++ 6.0을 이용해 구현하였으며, 실험에 사용한 자료는 애니메이션 영상으로 320\*240 RGB 영상이다. 총 frame 수는 975, 플레이 타임은 65sec로 frame율은 15이다.

다음 그림은 사용자가 선택한 동영상에서 컷으로 검출된 프레임의 리스트 박스 모습이다.



그림 3 . 컷 검출 프레임 리스트

이러한 동영상 검색의 전처리 과정인 색인 작업을 거친 후 추출된 frame을 가지고 영상검색을 하게 되면 사용자의 노력 및 검색 시간이 단축되고, 동영상 정보의 관리가 용이해 질 것이다.

#### 4. 결 론

본 연구에서는 동영상 정보의 효율적인 활용을 위한 색인 방법으로 칼라 임계값을 이용한 컷 검출 방법을

제안하였다. 이것은 frame간의 유사도를 측정해서 이 값이 주어진 임계값 보다 작을 경우, 장면의 전환이 일어나는 곳을 컷 지점으로 검출하는 것인데, 동영상의 장면에 따른 유사도가 다를 수 있기 때문에, 컷을 구성하는 프레임들 간의 칼라 임계값에 변동을 주어 최적의 컷 검출율을 구하고자 했다.

초기의 칼라 임계값은 '80'을 사용했고, 이후 frame의 유사도가 임계값보다 클 경우 즉, 장면전환이 일어나지 않았을 경우는 일정한 상수 값을 초기 임계값에서 감소토록 하였다.

이는 동영상의 특성상 장면전환이 일어나기 전까지는 근접한 frame간의 유사도가 점진적으로 작아진다는 점에 근거를 두었다.

이러한, 과정을 거쳐 추출된 frame을 가지고 원하는 이미지를 검색하게 되면 사용자의 노력 및 검색 시간이 단축되고 동영상 정보의 관리가 용이해 진다. 추후에는 여기에 자동 색인 생성 기능을 추가하여 보다 효율적인 동영상정보 관리시스템을 구축하고자 한다.

#### [참 고 문 헌]

- [1] Rajiv Mehrotra, James E. Gray, "Featurebased Retrieval of similar Shapes", 9th International Conference on Data Engineering, pp108-115, 1993
- [2] James D. Murray & William Varrayper, "Graphics file format", second edition, O'Reilly & Associates Inc, 1996.
- [3] Hong Jiang Zhang, Atreyi Kankanhalli, and Stephen W. Smoliar, "Automatic partitioning of full-motion video": Multimedia system, Vol1.No. 1, pp10-28, 1993.
- [4] 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.
- [5] Farshid Arman, Arding Hsu, and Ming-Yee Chiu, "Image processing on encoded video sequences": Multimedia Systems, pp211-219, 1994
- [6] Ioannis Pitas, "Digital Image Processing Algorithms", Prentice Hall, 1993.
- [7] A. Nagasaka and Y. Tanaka, "Automatic Video indexing and full video search for object Appearances": Visual database system, volII Oct1991
- [8] William I. Grosky, Peter Neo, Rajiv Mehrotra, "A Pictorial Index Mechanism for Model based Matching", Data and Knowledge Engineering, vol.8, pp309-327, 1992.