

윤곽선과 컬러 분포를 이용한 비디오 분할과 비디오 브라우징

허 승[†] · 김 우 생^{††}

요 약

본 논문에서는 비디오 프레임들의 윤곽선과 컬러 분포를 사용한 비디오를 분할 하는 방법을 제안하며 분할된 장면의 정보를 사용하여 비디오 브라우징을 구현하였다. 비디오를 분할하기 위한 방법으로는 HSV 162개의 색상을 가진 히스토그램과 자동 임계값으로 산출된 윤곽선을 사용하였고 각 장면들의 객체 위치와 색상 분포 등의 특성을 고려하였다. 검출된 장면들을 계층적인 브라우저와 장면 기반 브라우저를 사용해 비디오를 브라우징 할 수 있도록 하였다. 또한 본 논문에서는 제안하는 장면 변화 검출 방법이 기존의 색상 분포만을 사용하는 히스토그램의 방법에 비해 움직임에 보다 견고하고, 빛의 영향을 최소화 할 수 있음을 다양한 종류의 비디오 데이터를 통해 보였다.

Video Segmentation and Video Browsing Using the Edge and Color Distribution

Seoung Heo[†] · Woo Saeng Kim^{††}

ABSTRACT

In this paper, we propose a video data segmentation method using edge and color distribution of video frames and also develop a video browser by using the proposed algorithm. To segment a video, we use a 64-bin HSV color histogram and the edge information which generated with automatic threshold method. We consider scene's characteristics by using positions and color distributions of object in each frame. We develop a hierarchical and a shot-based browser for video browsing. We also show that our proposed method is less sensitive to light effects and more robust to motion effects than previous ones like a histogram-based method by testing with various video data.

1. 서 론

최근의 멀티미디어 기술은 매우 빠른 속도로 발전

※본 연구는 1997학년도 광운대학교 교내 연구비에 의해 부분적으로 지원되었음.

† 정 회 원: 고합정보통신연구소 연구원

†† 중 심 회 원: 광운대학교 전자계산학과

논문접수: 1997년 2월 18일, 심사완료: 1997년 5월 28일

하고 있으며, 실생활의 많은 부분에서 적용되고 있다. 특히 비디오 데이터에 대한 필요한 기술로 비디오 브라우징(video browsing) 기술을 들 수 있다. 비디오 브라우징은 사용자의 비디오 데이터에 대한 피동적인 접근방식에서 비디오의 원하는 부분을 직접적으로 선택함으로써 보다 쉽고, 편리하게 접근할 수 있는 환경을 제공한다. 기존에는 비디오 데이터를 분류하

기 위해서 사용자가 비디오를 직접 보면서 적절한 내용을 텍스트 형식으로 직접 입력함으로써 비디오를 분류했다. 이와 같은 방법은 좋은 성능을 가질 수 있는 시스템을 구성할 수는 있지만 사용자의 많은 시간과 노력을 필요로 한다. 따라서 최근에는 이러한 사용자의 노력을 최소화하기 위해 컴퓨터에 의해 자동화할 수 있는 방법들이 연구되고 있다.

비디오 데이터를 자동적으로 분류하기 위해서 많이 쓰이는 기법은 장면 변화 현상을 사용하는 방법이다. 장면 변화 현상이란 연속적인 동화상 이미지의 어떤 한 장면에서 다른 장면으로 전이하는 것을 일컫는 말로 전이되는 특성에 따라 통상 컷(cut), 페이드(fade), 디졸브(dissolve), 와이프(wipe) 등으로 분류한다. 컷은 한 장면에서 다른 장면으로 변화하는 순간을 말하고, 페이드는 한 장면과 다른 연속적인 장면 간의 점차적으로 변화하는 것을 나타내는 것으로 변화하는 명암에 따라 페이드 인/아웃(fade in/out)의 두 가지가 있다. 디졸브는 한 장면에서 다른 장면으로의 점진적인 변화를 나타내는 말로 처음 장면에서는 페이드 아웃이 발생하고, 다음 장면에서는 페이드 인이 일어날 때의 현상을 말한다. 와이프는 한 라인에 따라서 화면이 움직이는 것으로 그 라인 뒤에서는 새로운 화면이 나타나고 그 라인의 앞에서는 전의 화면이 사라지는 현상을 말한다. 이러한 페이드, 디졸브, 와이프같은 현상을 점진적인 변화(gradual transition)라 한다.

비디오를 분할하기 위한 연구는 크게 주석을 기반으로 분할하는 방법[18, 19]과 비디오의 내용 기반으로 분류하는 방법이 있다. 주석을 기반으로 하는 방법에서는 사용자가 직접 비디오를 보면서 내용을 텍스트 형식으로 입력하는 방법으로 여러 가지의 응용에서 손쉽게 사용할 수 있고, 비디오의 의미들을 보다 정확하게 모델링하고 검색 할 수 있다는 장점을 가지고 있으나 이러한 방법은 사용자의 많은 노력과 시간을 필요로 한다는 단점을 가진다. 비디오의 내용을 기반으로 하는 방법은 비디오를 이루는 이미지들의 특성 값들을 추출하고 이러한 특성 값들을 비교함으로써 비디오를 분류하고자 하는 방법으로 주석 기반 방법보다는 부정확하게 비디오를 분류하지만 자동적으로 분류할 수 있다는 장점을 가진다. 비디오의 내용을 기반으로 비디오를 분류하는 방법은 주로 비

디오 장면들을 검출함으로써 분류하는 방법으로 크게 픽셀간의 색상을 비교하는 방법[6, 12], 히스토그램의 차이를 비교하는 방법[1, 8, 11], 압축 계수의 차이를 비교하는 방법[4, 5, 14], 움직임 벡터[6, 8], 그리고 형태 기반 검출[3, 7]을 사용하는 방법 등으로 분류할 수 있다. 기존의 여러 연구에서 공통적으로 사용되었던 전역 히스토그램을 이용한 장면 변화 현상의 검출 방법은 객체의 움직임에는 비교적 견고하나 이미지 내 객체의 위치는 전혀 고려하고 있지 않기 때문에 거의 같은 색상 분포를 가지는 두 장면을 구분하지 못한다는 단점을 가지고 있다. 또한 이러한 단점을 해결하기 위해 사용되었던 영역 히스토그램 방법은 색상의 변화와 객체의 위치를 고려함으로써 이러한 문제는 해결할 수 있었으나 움직임에서는 매우 민감한 반응을 일으킨다는 단점이 있다[1].

따라서 본 논문에서는 움직임에 비교적 견고한 전역 히스토그램과 객체의 정확한 위치를 표현하기 위한 객체의 윤곽선 정보를 조합해 사용함으로써 객체의 움직임과 빛의 영향을 최소화해 장면 변화 현상을 보다 정확하게 검출할 수 있도록 하였다. 그리고 검출된 장면 변화 현상을 동화상 정보 데이터 구조에 저장하고 이를 이용해 비디오를 브라우징할 수 있도록 하였다. 본 시스템에서 사용하는 비디오 브라우저로는 비디오의 전체에서 계층적으로 점차 자세하게 접근을 가능하게 하는 계층적인 브라우저와 동화상 정보 데이터의 각 장면 아이템으로 구성된 장면 기반 브라우저의 두 가지를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 히스토그램과 윤곽선을 사용한 프레임간의 차이를 계산하는 과정과 이러한 차이 값을 이용한 장면 변화를 검출하는 방법을 기술한다. 3장에서는 본 시스템의 구현과 제안하는 방법이 전역 히스토그램과 영역 히스토그램에 비해 보다 정확하게 검출할 수 있음을 다양한 종류의 데이터에 실험함으로써 보이고, 마지막으로 4장에서는 결론과 향후 연구 과제 등을 기술한다.

2. 장면 변화 현상의 검출

본 장에서는 프레임간의 차이를 계산하는 과정과 계산된 값을 이용해 장면 변화 현상을 검출하는 과정을 기술한다.

2.1 프레임간의 차이 계산

장면 변화 현상의 검출을 위한 프레임간의 차이는 각각 프레임의 전역 히스토그램과 윤곽선을 계산함으로써 산출되어진다. 전역 히스토그램의 차이는 통상 컴퓨터에서 사용하는 RGB 색상 모델보다는 사람의 시각에 보다 민감한 HSV 색상 모델로 변환 후 162가지의 색상으로 표본화(sampling)한 후 계산되어진다. 또한 윤곽선의 차이는 먼저 각각의 프레임 이미지에 맞는 임계값을 자동으로 산출하고 산출된 임계값을 사용함으로써 윤곽선 이미지를 추출하고 추출된 윤곽선 이미지를 Dilation한 이미지[7]를 생성하고 이 두 이미지에 대해서 소멸 윤곽선(exiting edge), 출현 윤곽선(entering edge)을 계산함으로써 차이 값이 얻어진다.

2.1.1 HSV 히스토그램 생성 및 차이 계산

연속된 비디오 데이터에서 여러 가지의 장면 변화 현상을 분류하기 위해서 기본적으로 히스토그램 방법을 사용한다. 컴퓨터에서 통상 사용하는 RGB 색상 모델은 인식적인 측면에서 균일한 분포를 가지지 못하기 때문에 이러한 단점을 개선하기 위해 CIE-LAB, CIE-LUV, Munsell등이 제안되었다[2]. 일반적으로 이러한 색상 모델은 보통 색깔에 대한 표현으로 사용하는 색상(Hue), 채도(Saturation), 명도(Brightness)등의 세 가지 요소에 대해 균일한 분포를 가진다. 그러나 이러한 색상 모델은 RGB 색상 모델과의 변환이 비선형적인 연산을 포함하기 때문에 매우 많은 연산이 요구된다. 따라서 균일한 분포, 비선형적인 연산의 특성을 가지면서, 쉽게 변환이 가능한 HSV 색상 모

델을 사용한다. 변환된 HSV 색상은 다시 히스토그램을 생성하기 위해 HSV 값은 각각 18:3:3으로 다음(그림 1)과 같은 표본화 과정을 거치게 된다.

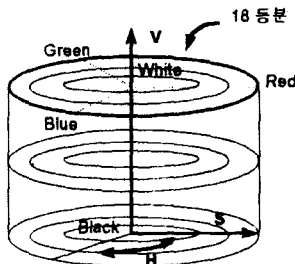
위의 표본화 과정은 색상 값(H)은 대체로 그대로 유지하는 편이나 밝기 값(V)과 채도 값(S)을 작게 함으로써 빛의 영향이 최소화 될 수 있도록 하는 과정이다. 이것은 연속적인 이미지에서 같은 장면의 프레임들이라 할지라도 원근 또는 빛의 영향으로 근소한 차이를 보이면서 변하게 되기 때문에, 이것들은 장면 변화 현상 검출에서는 불필요한 정보로 최소화되어야 하는 것이다. 히스토그램의 차이값은 다음과 같이 정의된다.

$$h_{dk} = \frac{\sum_{i=0}^{161} (h_k(i) - h_{k-1}(i))}{\sum_{i=0}^{161} (h_k(i))} \tag{1}$$

h_{dk} 는 연속적인 비디오에서 산출한 k번째 히스토그램과 k-1번째 히스토그램과의 차이의 합을 k번째 히스토그램의 전체 합으로 나눈 비율이다. 두 프레임 k와 k-1이 서로 다른 장면이라면 차이값은 매우 큰 값이 되고, 같은 장면에 포함된 프레임일 경우에는 작은 값을 산출하게 된다.

2.1.2 자동 임계값을 사용한 윤곽선의 추출과 차이 계산

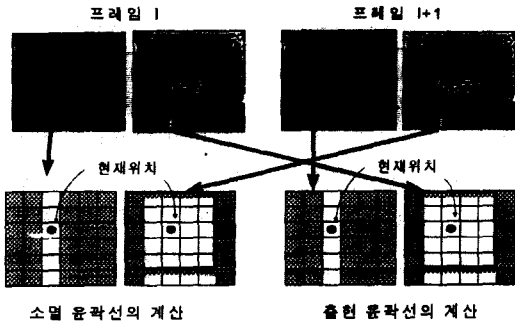
장면 변화 현상을 구분하기 위한 또 다른 매개변수로는 윤곽선의 정보를 사용한다. 비디오는 정지 화상과는 달리 어두운 이미지가 연속되는 경우도 존재하고, 밝은 이미지가 연속해서 나오는 경우도 많다. 기존의 고정된 임계값을 사용해 윤곽선을 찾는다면, 이러한 경우에는 정확한 윤곽선에 대한 정보를 산출할 수 없다. 따라서 본 논문에서는 고정된 임계값 대신 각 프레임의 특성에 맞는 임계값을 자동으로 계산해 이러한 문제점을 해결한다[14]. 이때 생성된 윤곽선 이미지는 배경의 복잡성 등의 영향으로 많은 잡음을 내포하게 되기 때문에 잡음을 최소화하기 위해 윤곽선 이미지에 세션화[19]를 적용해 불필요한 잡음을 제거하였다. 잡음이 제거된 윤곽선 이미지는 다시 윤곽선간의 비교를 위해 Dilation 이미지[15]를 생성한다. 이러한 과정을 통해 계산된 윤곽선 이미지와 Dilation



$$Q_c = 9 * (H / (360 / 18)) + 3 * (S / (1 / 3)) + (V / (1 / 3))$$

(그림 1) HSV 히스토그램의 표본화 (Fig. 1) Sampling of HSV Histogram

이미지는 프레임간 윤곽선 사이의 차이 계산을 위해 사용된다. 윤곽선의 차이 계산은 (그림 2)와 같은 방법으로 구해진다.



(그림 2) 프레임간 윤곽선 차이의 계산

(Fig. 2) Computation of Edge Difference between frames.

프레임간의 윤곽선을 비교하는 방법은 그림에서 보는 바와 같이 비교할 프레임의 이미지에서 윤곽선이 발견되는 경우, 발견되는 좌표와 비교대상 이미지에 같은 좌표를 중심으로 5x5 템플릿을 적용함으로써, 템플릿 내에 윤곽선이 존재하면 윤곽선 수에 0을, 존재하지 않는다면 1을 더함으로써 윤곽선간의 비교를 하게 된다. 윤곽선의 비교는 전 프레임에서 추출된 윤곽선 이미지 E_k 을 비교할 이미지로 두고 다음 프레임의 Dilation 이미지 ED_{k+1} 를 비교대상 이미지로 두는 경우와 다음 프레임에서 추출된 윤곽선 이미지 E_{k+1} 을 비교할 이미지로 두고 전 프레임에서 추출된 Dilation 이미지 ED_k 을 비교대상 이미지로 두는 경우 사이에서 이루어진다. 전자의 경우는 전 프레임에서 발견되었던 것이 다음 프레임에서 어떻게 되었나를 조사하는 것으로 소멸된 윤곽선만을 조사하게 된다. 이러한 윤곽선을 소멸 윤곽선이라 정의하고, 후자의 경우는 이와 반대로 새롭게 출현된 윤곽선을 조사하는 것으로 출현 윤곽선이라 정의한다. 이러한 두 값은 윤곽선의 변화를 표현하는 것으로써 컷 검출시 변화율의 비교를 위해 하나의 대표값을 정의한다. 윤곽선의 변화를 대표하는 값은 두 값중에서 더 큰 값으로 두고 이를 윤곽선의 차이값으로 사용한다.

2.2 전 처리 과정

비디오에서 장면 변화 현상은 일반적으로 컷과 점

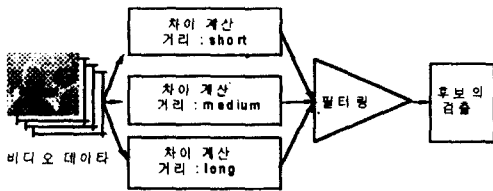
진적인 변화의 두 가지로 분류되는데, 컷은 인접한 두 개 프레임간의 비교를 통해서 구할 수 있으나 점진적인 변화는 이러한 두개의 프레임간의 차이 비교를 통해서 계산해내기가 어렵다. 점진적인 변화를 보이는 곳에서 컷을 검출하는 방법을 사용하는 경우, 비디오 생성 시에 페이드, 디졸브등과 같은 추가되는 여러 가지의 특수 효과들로 인해 잘못된 컷들을 검출하게 된다. 또한 비디오 내의 많은 부분을 차지하고 있는 객체의 빠른 움직임과 같은 현상들 또한 잘못된 컷을 검출하게 된다. 따라서 제안하는 방법에서는 이러한 잘못된 컷을 검출하게 되는 많은 움직임과 점진적인 변화의 후보들을 미리 산출하여 분류함으로써 보다 정확하게 장면 변화를 검출할 수 있도록 하였다. 본 논문에서는 이와 같은 움직임이 많은 것과 점진적인 변화, 컷 등을 판단하기 위해 다음 <표 1>와 같은 4가지의 임계값을 사용한다. HHThreshold와 HLThreshold는 히스토그램의 높은 임계값과 낮은 임계값을 말하며, 이 중 HHThreshold는 두개의 프레임을 다른 장면으로 분류하기 위해서는 최소한 이 값을 초과해야 한다는 값으로 컷을 검출하는데 사용되고, HLThreshold는 점진적인 변화의 후보가 되기 위해서는 적어도 이 값을 초과해야 하는 값을 의미한다. EHThreshold와 ELThreshold도 윤곽선의 차이에 대해 같은 내용과 같은 의미를 갖는다.

<표 1> 제안하는 방법에서 사용하는 임계값들
<Table 1> Thresholds for the proposed method

임계값	의미
HHThreshold	컷을 검출하기 위한 히스토그램의 높은 임계값
HLThreshold	점진적인 변화를 검출하기 위한 히스토그램의 낮은 임계값
EHThreshold	컷을 검출하기 위한 윤곽선의 높은 임계값
ELThreshold	점진적인 변화를 검출하기 위한 윤곽선의 낮은 임계값

(그림 3)은 객체의 빠른 움직임과 점진적인 변화의 가능성이 있는 후보를 검출하는 방법을 설명한다. 비디오 데이터에 대해 프레임간의 거리 간격을 3단계로 나누어서 각각 프레임간의 히스토그램과 윤곽선의

차이를 분석해 낸다. 이러한 차이는 같은 장면으로 분류될 프레임들에서는 각 단계의 차이의 값은 작게 나오나, 다른 장면으로 분류될 프레임들에서는 매우 큰 값을 가진다. 이러한 속성을 사용해 각 단계에서 변화가 많이 발생하는 프레임들을 객체의 빠른 움직임과 점진적인 변화의 후보로 두고자 하는 방법으로 히스토그램을 사용해서 점진적인 변화를 검출하고자 제안하는 방법[6]에 윤곽선의 차이 정보까지를 포함시킨 방법이다.



(그림 3) 후보의 검출
(Fig. 3) Detection of candidate frames

분석하고자 하는 비디오 데이터는 위와 같은 계산 과정을 전 프레임에 걸쳐 행함으로써 객체의 빠른 움직임과 점진적인 변화를 가지는 후보의 범위를 산출해 낼 수 있다. 점진적인 변화를 일으키는 후보의 경우에는 후보의 범위는 단순히 몇 개의 프레임으로 구성되어지지 않고, 여러 많은 프레임들에 걸쳐서 일어난다. 여러 개의 프레임들로 구성되는 것들만 점진적인 변화의 후보로 두고, 단지 몇 개의 프레임의 후보는 움직임이 많은 것으로 분류한다. 그러나 점진적인 변화의 후보로 둔 것들 모두는 점진적인 변화의 후보는 되지 않는다. 이 중에는 객체의 빠른 움직임이 여러 개의 프레임에 걸쳐서 되는 경우도 있으므로 객체의 빠른 움직임과 점진적인 변화의 가능성이 있는 후보에 대해서는 가장 짧은 거리의 프레임간의 차가 $HHThreshold$ 와 $EHThreshold$ 를 초과하는 경우가 있는지를 조사하여 초과하는 경우가 발생할 때에는 그 후보에 대해서는 객체의 빠른 움직임이 포함되어 있는 것으로 분류한다. 이것은 움직임이 많은 프레임들의 윤곽선 차이 값보다 점진적인 장면 변화 프레임들에서의 윤곽선의 차이 값이 훨씬 더 작게 나오기 때문이다. 일반적으로 점진적인 변화는 두 화면이 거의 변화가 없는 상태에서 일어나기 때문에 두 프레임에

서 윤곽선의 차이는 거의 없다. 그러나 움직임을 많이 내포하는 프레임간에는 윤곽선의 차이가 매우 크게 나오게 된다.

2.3 컷의 검출

컷은 두 개의 프레임간 경계점으로 두 프레임간의 차이 값을 계산함으로써 검출되는 장면 변화 현상으로 제안하는 방법에서는 히스토그램 차이 값과 윤곽선 차이 값을 동시에 사용함으로써 검출한다. 컷을 검출하기 위한 분석 대상으로는 잘못된 컷을 최소화하기 위해 전처리 과정에서 움직임이 많은 것들과 점진적인 후보를 뺀 나머지 프레임들에 대해 행해진다. 식 (2)는 히스토그램과 윤곽선의 차이 값에 대해 컷의 가능성을 조사하기 위한 식이다. $H_{decision}$ 은 히스토그램의 차이 값에 대해서는 $HHThreshold$ 와의 비교를 통해 가능성을 조사하고, $E_{decision}$ 은 윤곽선 차이 값에 대해서는 $EHThreshold$ 를 사용해 가능성을 타진한다. 이 값들은 각각의 차이 값들이 적어도 이 값을 초과해야지만 컷으로 분류될 수 있다는 것을 의미한다.

$$H_{decision} = \begin{cases} TRUE & \text{if } h_{dk} > HHThreshold \\ FALSE & \text{else} \end{cases}$$

$$E_{decision} = \begin{cases} TRUE & \text{if } \rho > EHThreshold \\ FALSE & \text{else} \end{cases} \quad (2)$$

〈표 2〉는 식(2)에 의해 계산된 값에 대해 컷을 판단하기 위한 도표이다 두 개의 반환 값이 모두 참을 반환하는 경우에는 컷으로 분류하고, 두 값 모두 거짓 값을 가질 때는 컷으로 분류하지 않는다. 그리고 히스토그램의 비교 반환 값이 참이고 윤곽선의 비교 반환 값이 거짓일 때에는 위의 잘못된 컷의 검출의 한 경우였던 빛의 영향으로 인해서 히스토그램의 차이는 많으나, 윤곽선에서는 차이가 없는 경우로 컷으로 분류하지 않는다. 또한 이와 반대로 히스토그램의 비교 반환 값은 거짓이나, 윤곽선의 비교 반환 값은 참일 경우에는 같은 색상 분포로 인해 히스토그램에서는 차이 값이 별로 나지 않는 경우로 본 방법에서는 컷으로 인식한다. 이러한 방법을 사용함으로써 기존의 히스토그램을 사용하는 방법에서 가장 큰 문제점인 빛의 영향과 다른 장면임에도 불구하고 같은 색상 분포로 인해 분류하지 못한다는 단점을 해결함으로써

보다 정확한 것을 검출할 수 있도록 하였다.

〈표 2〉 컷의 판단
(Table 2) Decision of cut

$H_{decision}$	$E_{decision}$	컷의 판단
○	○	cut
○	×	×
×	○	cut
×	×	×

2.4 점진적인 변화의 검출

점진적인 변화라 함은 점차적으로 변화를 일으키는 프레임들의 연속으로, 컷을 검출하는 방법에서와 같이 두 개 프레임간의 비교를 통해서만 제대로 검출하지 못한다. 점진적인 변화를 검출하기 위해서는 여러 개의 프레임들을 고려해야 하나 움직임이 많은 프레임들과 구분하기 매우 어렵다는 문제를 가진다. 따라서 본 연구에서는 전 처리 과정에서 점진적인 변화의 후보와 움직임이 많은 것들을 미리 분류해냄으로써 움직임이 많은 것들은 점진적인 변화의 검출 방법에서는 사용되지 않도록 하였다. (그림 4)는 점진적인 변화 현상을 검출하기 위해 윤곽선의 분포를 계산하는 과정을 보여주는 그림으로 먼저 점진적인 변화의 후보에 대해서 범위를 산출하고, 범위를 3개의 부분으로 분류한다. 3개의 부분으로 분류하는 것은 한 장면에서 다른 장면으로의 점진적인 변화는 거의 같은 분포를 가지면서 변해가기 때문이다. 예를 들어 A라는 장면에서 B라는 장면으로 변해 가는 디졸브 현상은 A 장면은 거의 일정한 비율로 사라져가고, B 장면은 거의 일정한 비율로 나타나게 된다. 그렇기 때문에 3부분으로 분류한다는 것은 A의 사라지는 부분이

첫 번째 부분으로, A장면과 B장면이 겹치면서 변하는 부분은 두 번째 부분으로, B장면이 나타나는 부분이 세 번째 부분으로 분류되게 된다. 앞절에서 기술한 윤곽선의 차이 값인 소멸 윤곽선과 출현 윤곽선의 분포의 합을 계산해 차이 합의 값이 어떤 부호를 가지는가를 이용해 현상을 분류한다.

〈표 3〉는 3부분으로 분리된 변화의 후보에 대해 계산된 소멸 윤곽선과 출현 윤곽선의 분포에 따른 점진적인 변화 현상을 판단하기 위한 도표이다. 3개의 부분에서 모두 양의 값을 갖는 경우 즉, 소멸 윤곽선이 더 많은 분포를 이루는 경우에는 페이드 아웃으로 분류하고, 이와 반대로 각 부분에서 음의 값을 가지는 경우, 즉 출현 윤곽선이 더 많이 발견되는 경우에는 페이드 인으로 분류한다. 그리고 1 Part에서는 소멸 윤곽선이 더 많이 발생하는 즉, 양의 값을 갖고, 3 Part 부분에서는 출현 윤곽선이 더 많이 발생하는 경우 즉, 음의 값을 가지는 경우에는 디졸브 현상으로 분류한다. 그러나 위의 모든 경우에 포함되지 않는다면 UnKnown을 발생함으로써 사용자가 나중에 직접 내용을 봄으로써 발생하는 현상을 입력할 수 있도록 하였다.

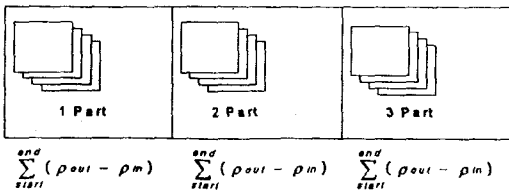
〈표 3〉 윤곽선 분포에 따른 점진적인 변화의 판단
(Table 3) Decision of Gradual Transition by Edge Distribution

1 part	2 part	3 part	판단
+	+	+	Fade out
-	-	-	Fade in
+	+/-	-	Dissolve
그외의 경우들			Unknown

3. 구현 및 결과 분석

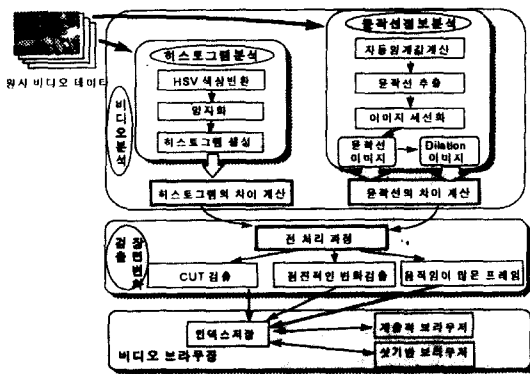
본 장에서는 비디오를 분할하는 과정과 비디오 브라우저등의 구현 방법을 기술한다. 또한 장면 변화 검출의 결과 분석을 통해 제안하는 방법이 보다 정확하게 검출할 수 있음을 보인다. 본 시스템은 Windows 95상에서 Visual C++ 언어를 사용해 구현되었으며, 사용하는 비디오 데이터는 AVI(Audio Video Interleave)[16]를 사용하였다. (그림 5)는 본 시스템의 개략적인 구조를 표현한 그림으로 비디오 분석을 통한 장

점진적인 변화의 후보



(그림 4) 윤곽선의 분포에 따른 계산
(Fig. 4) Computation of Edge distribution

면 변화 검출, 비디오 브라우징의 두 가지로 분류해서 구현했다. 장면 변화 검출은 비디오를 분류하기 위해 장면 변화 현상을 검출하는 과정으로 비디오 데이터의 분석을 통한 윤곽선과 히스토그램의 차이 값을 적절하게 사용함으로써 분류했다. 비디오 브라우징은 이러한 과정으로 계산된 장면 변화 현상을 동화상 정보 데이터에 저장하고 이를 사용함으로써 비디오를 손쉽게 접근할 수 있는 환경을 조성했다.



(그림 5) 구체적인 시스템 구성도
(Fig. 5) Overview of the Detailed System

3.1 비디오 분석을 통한 장면 변화 검출의 구현

본 절에서는 2장에서 기술한 두 가지의 특성 값을 사용해 비디오의 장면을 분류하는 과정을 설명한다. 장면 변화의 검출은 인접한 두 프레임간의 차이 또는 몇 개의 프레임간의 차이를 통해서 이루어진다. 프레임간의 차이 값들은 short, medium, long의 세 거리에 대해서 각각 계산되어지고 전처리 과정을 거치게 된다. 컷과 점진적인 변화의 장면 변화 현상은 다음과 같은 과정으로 검출된다.

(a) 검출하고자 하는 현재 프레임이 점진적인 변화의 후보로 분류되었다면 (e) 단계로 가서 점진적인 변화의 현상을 검출하도록 하고, 객체의 움직임이 많은 프레임으로 분류되었다면 검출할 다음 프레임의 위치를 현재 프레임의 값에 1을 더한 값을 할당하고 다시 (a)의 단계를 수행한다.

(b) 검출하고자 하는 현재 프레임과 이전 프레임과의 히스토그램과 윤곽선의 차이 값을 계산하고 차이

값들이 각각의 임계값을 초과하는지의 여부를 판단한다.

(c) 각각의 판단 값이 참의 값을 가지거나 히스토그램의 판단값이 거짓이나 윤곽선의 차이의 값이 참일 경우에는 컷으로 분류하고, 그렇지 않는 경우에는 컷으로 분류하지 않는다.

(d) 현재 프레임이 컷으로 분류되는 경우에는 전에 장면 변화 현상으로 분류되었던 프레임과의 차이를 계산해서 장면 변화를 허용하는 최소 프레임간의 거리값 이상일 때는 컷으로 분류하고 동화상 정보 데이터에 저장한다. 그리고 다음에 처리할 프레임 번호에 1을 더하고 다시 (a)의 단계로 가서 다음 프레임을 처리한다.

(e) 점진적인 변화의 후보들에 대해 각 범위를 세 부분으로 분리하고, 분리된 각 부분에 대해서 소멸 윤곽선과 출현 윤곽선의 차이의 합을 구한다.

(f) 계산된 차이의 합을 사용해 각 부분의 윤곽선의 차이의 합이 모두 양의 값을 가질 때는 페이드 아웃으로 분류하고, 모두 음의 값을 가질 때는 페이드 인으로 분류한다. 그리고 처음 부분에서는 양의 값을 갖고 마지막 부분에서는 음의 값을 가지는 경우에는 디졸브로 분류한다. 그리고 나머지의 경우에 대해서는 UnKnown으로 분류한다.

(g) 분류된 장면 변화 현상을 동화상 정보 데이터에 저장하고, 다음에 검출할 프레임의 위치를 점진적인 변화 후보의 종료 프레임 번호에 1을 더한 값으로 둔다. 그리고 다시 (a) 단계로 가서 다음 장면 변화를 검출할 수 있도록 한다.

이러한 과정으로 검출된 장면 변화 현상들은 하나의 동화상 정보 데이터를 산출하게 되는데 동화상 정보 데이터는 비디오를 구성하는 각각의 장면들이 갖는 특성 값을 표현하기 위한 데이터로 (그림 6)과 같이 구성된다. 그림에서 장면 번호는 각각의 장면을 구별하기 위한 식별자로 사용되고 시작 프레임과 종료 프레임의 번호는 각 장면의 시작 프레임과 종료 프레임의 번호를 뜻한다. 장면 변화 현상은 연속된 프레임에서 어떠한 현상으로 인해 다른 장면으로 분류되었나를 표현하는 것으로 컷, 페이드 인, 페이드 아웃, 디졸브등으로 분류한다. 차이 값은 장면 변화 현상을 일으킬 때의 히스토그램과 윤곽선의 차이 값이다. 분류하고자 하는 비디오 데이터마다 하나의 이

러한 동화상 정보 데이터를 생성함으로써 비디오의 브라우저를 가능하게 한다.

장면 번호	시작 프레임 번호	종료 프레임 번호	장면 변화 현상	차이 값
-------	-----------	-----------	----------	------

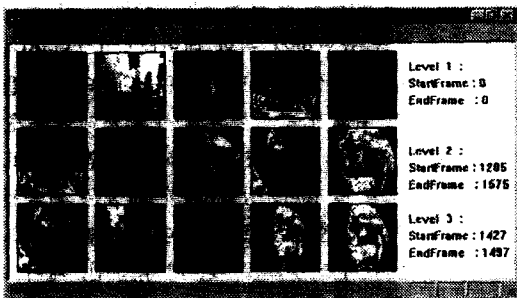
(그림 6) 동화상 정보 데이터의 구조
(Fig. 6) Structure of information data for moving pictures

3.2 비디오 브라우저의 구현

본 절에서는 검출된 장면들을 저장하고 있는 동화상 정보 데이터를 사용함으로써 비디오 데이터에 연산을 수행하는 비디오 브라우저를 기술한다. 본 시스템에서는 계층적인 브라우저와 장면 기반 브라우저의 2가지 비디오 브라우저를 제공한다.

3.2.1 계층적인 브라우저

계층적인 브라우저는 비디오 데이터를 계층적으로 접근할 수 있는 브라우저로 전체의 비디오를 대표할 수 있는 몇 개의 프레임을 제일 상위 레벨에 표현한다. 그리고 사용자가 보기를 원하는 대표 프레임을 선택하면 다음 레벨에서는 그 부분을 대표하는 몇 개의 대표 프레임을 다시 생성해서 보여준다. 그래서 사용자가 전체적인 보기에서 보다 자세한 부분으로 계층적으로 접근할 수 있도록 한다. 이 브라우저는 4.1절에서 구한 동화상 정보 데이터를 읽어 들여 가장 많은 차이 값을 대표 프레임으로 선택함으로써 구현된다. (그림 7)은 '순수의 시대'라는 비디오 데이터를



(그림 7) 계층적인 브라우저
(Fig. 7) hierarchical browser

계층적인 브라우저를 통해 접근하는 것을 보여주는 예이다. 이 비디오는 5개의 대표 프레임을 생성하고 레벨 1에서는 4번째 대표 프레임이 선택되었고 레벨 2에서도 4번째의 프레임이 선택된 모습이다.

3.2.2 장면 기반 브라우저

장면 기반 브라우저에서 표현되는 각 프레임 이미지들은 비디오에서 각각 하나의 장면을 대표하는 것으로서 동화상 정보 데이터의 각각의 요소를 사용해서 표현된다. (그림 8)은 '순수의 시대' 비디오 데이터를 장면 기반 브라우저로 나타낸 모습이다. 그림 내의 각 이미지들은 비디오를 구성하는 장면들을 대표하는 이미지로 사용자는 이러한 이미지들을 한 눈에 볼 수 있도록 하여 비디오 전체의 내용에 대한 대략적인 이해를 가능하게 한다. 이 브라우저에서 보기를 원하는 프레임이 선택되면 다음 절에서 설명할 비디오 재생기가 실행되면서 원하는 부분을 볼 수 있다.



(그림 8) 장면 기반 브라우저
(Fig. 8) shot-based browser

3.2.3 비디오 재생기

비디오의 브라우저를 통해 원하는 부분을 선택했다면, 원하는 부분을 실제적으로 볼 수 있는 재생기가 필요하다. 위의 계층적인 브라우저와 장면 기반 브라우저에서 원하는 부분을 선택하게 되면 (그림 9)와 같은 재생기가 실행되면서 부분적인 재생이 가능하게 한다.

3.3 결과 분석

제안하는 비디오 분할 방법의 효율성을 보이기 위



(그림 9) 비디오 재생기
(Fig. 9) Video Player

해 기존의 연구에서는 가장 좋은 성능을 보이는 전역 히스토그램을 사용하는 방법과 영역 히스토그램을 사용하는 방법들과 비교 하였다. 히스토그램을 사용하는 두 가지의 방법은 3장에서 설명한 HSV히스토그램을 사용해서 검출한다. 전역 히스토그램 방법은 두 프레임간의 차이 값에 일정 임계값을 적용해 이러한 임계값을 초과하는 경우에는 컷으로 검출하고, 점진적인 변화에 대해서는 [20]의 방법을 사용해 검출한다. 그리고 영역 히스토그램을 사용하는 방법에서는 이미지의 영역을 4x4으로 분할한 후 각 영역에 대해서 162개의 색상의 HSV 히스토그램을 구하고, 프레임간 같은 영역에 대해서 각각 히스토그램의 차이 값을 계산한다. 차이 값이 일정 임계값을 초과하는 경우를 조사해 50%(=8)이상이면 프레임간의 차이가 많은 것으로 간주해 장면으로 인식하는 방법으로 [8]의 변형된 형태를 사용한다.

기존의 방법과 본 논문에서 제안하는 방법과의 성능을 비교, 평가하기 위한 매개변수로는 다음과 같은 세 가지 CR(Correct detection Rate), MR(Mis-detection Rate), FR(False detection Rate)를 사용한다.

$$CR = \frac{D_c}{C_t}, MR = \frac{D_m}{D_t}, FR = \frac{(D_t - D_c - D_m)}{D_t} \quad (4.2)$$

CR은 비디오 분할의 정확도를 측정하는 매개변수로 C_t 는 원시 비디오에서 포함된 전체 장면의 수를

의미하고, D_c 는 각 방법에 의해 정확히 검출된 장면의 수를 의미하는 것으로 전체 비디오에 포함된 장면이 얼마만큼 정확히 검출되고 있는가를 의미한다. 또한 D_m 은 장면임에도 불구하고 검출하지 못하는 프레임들의 수를 나타내고, D_t 는 검출된 전체 장면의 수를 의미한다. MR은 검출하지 못하는 컷을 측정하는 매개변수로 실제로는 장면 변화를 일으키나, 검출하지 못하는 경우를 뜻하는 것으로써 전체 검출된 것으로부터 검출하지 못한 것의 비율로 계산한다. 그리고 FR은 장면 변화현상이 일어나지 않았음에도 불구하고 잘못된 검출한 것에 대한 비율로 전체 검출된 것에서 정확히 검출된 것과 검출하지 못한 것을 뺀, 나머지에 검출된 전체 장면 변화수의 비율로 나타낸다. 다음 <표 4.1>은 제안하는 방법과 전역 히스토그램을 사용하는 방법, 그리고 영역 히스토그램을 사용하는 방법을 비교하기 위해 3가지의 비디오(A video: 순수의 시대, B video: 뱀파이어의 인터뷰, C Video: good times 뮤직 비디오)에 대해 실험한 결과이다. 위의 표에서 알 수 있듯이 전역 히스토그램과 영역 히스토그램을 사용하는 방법에서는 많은 차이를 보이지 않지만, 본 시스템에서 사용하는 방법과는 많은 차이를 보인다. B 비디오는 비디오 자체가 대부분이 어두운 이미지로 구성되어 있기 때문에, 색상의 분포는 거의 변화 없지만, 실제로는 장면이 변화되는 프레임들이 많이 발견된다. 이것은 검출하지 못한 컷(FR)들을 일으키는 요인으로 전역 히스토그램에서 특히 많이 발견되나, 본 시스템에서는 비교적 견고함을 보여주고 있다. 또한 C 비디오의 경우에 있어서는 빛의 영향으로 인해 컷이 아닌데도 불구하고, 컷으로 잘못된 검출이 많은 비디오로 특히 전역 히스토그램에서 가장 많이 발생하고, 영역 히스토그램도 많이

<표 4> 각 방법에 따른 실험 결과
<Table 4> Result of Experience Applied each methods 단위 : %

비디오	전역 히스토그램			영역 히스토그램			제안하는 방법		
	CR	MR	FR	CR	MR	FR	CR	MR	FR
A video	93.1	5.26	22.8	93.1	5.35	23.2	98.7	1.37	8.5
B video	89.8	7.95	31.6	92.7	6.90	32.3	98.1	1.40	10.8
C video	90.2	5.43	41.3	92.5	4.34	36.9	97.5	2.17	12.8

발생하고 있다. 그러나 본 시스템에서 사용하는 방법은 빛의 영향에 대해서도 비교적 견고함을 보여주고 있다.

4. 결 론

최근의 멀티미디어 처리 기술이 빠른 속도로 발전하고 있고, 그에 따른 많은 요구들이 필요로 하게 되었다. 멀티미디어 데이터를 다루는 데 있어서 현재의 가장 큰 문제는 비디오 등과 같은 멀티미디어 데이터의 브라우징과 인덱싱의 어려움이다. 기존의 정형화된 데이터와 달리 비디오 데이터와 같은 멀티미디어 데이터를 어떤 특성 데이터의 형태로 분류하기란 거의 불가능하기 때문이다. 따라서 본 논문에서는 비디오를 분류하고자 하는 많은 연구에서 사용하는 히스토그램을 사용하는 비디오 분할 방법의 문제점을 알아보고, 이를 해결하기 위해 윤곽선 정보를 조합해서 사용함으로써 효과적으로 비디오를 장면 단위로 분류하는 방법을 제안했다. 그래서 기존의 방법에서의 가장 큰 문제점이었던 객체의 움직임과 빛의 영향을 최소화 함으로써 보다 정확히 비디오 데이터를 분류할 수 있도록 했다. 그리고 몇 가지의 비디오 데이터에 실제로 적용함으로써 보다 정확히 검출할 수 있음을 보였다.

향후 연구과제로는 본 논문에서 제안하는 방법은 많은 윤곽선간의 계산으로 말미암아 실시간 처리를 필요로 하는 TV방송과 같은 응용에서는 적합하지 못하다. 따라서 이러한 객체의 위치와 색상을 고려하는 보다 빠른 방법에 대한 보다 많은 연구가 필요하다. 그리고 고려하지 않았지만 와이프 현상의 검출에 대한 연구와 카메라의 연산들(줌, 패닝)을 고려한다면 보다 정확하게 비디오를 분할할 수 있을 것이다.

참 고 문 헌

- [1] Akio Nagasaka, and Yuzuru Tanaka, "Automatic video indexing and full-video search for object appearances", Visual Database Systems II, pp 113-127, October 1991.
- [2] Dana H. Ballard, Christopher m. Brown "Computer Vision", Prentice-hall, pp31-34, 1982.
- [3] Daniel P.Huttenlocher, Gregory A.Klanderman, William J.Rucklidge, "Comparing Images Using the Hausdorff Distance", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 15, No 9, pp850-863, 1991.
- [4] Farshid Arman, Arding Hsu, and Ming-Yee Chiu, "Image processing on Encoded Sequences", Multimedia system Vol 1. No 5, pp211-219, 1994.
- [5] Farshid Arman, Arding Hsu, and Ming-Yee Chiu, "Image processing on compressed data for large video databases", Multimedia Conference, pp267-272, ACM, 1993.
- [6] H. Zhang, A. Kankanhalli, and S. Smoiler. "Automatic partitioning of full-motion video", Multimedia systems Vol 1, pp10-28, 1993.
- [7] Henning Bassmann, Philipp W.Besslich, "Ad Oculos, Digital Image Processing professional version 2.0", International Thomson Publishing, pp218-223, 1995.
- [8] Hirotsada Ueda, Takafumi Miyatake and Satoshi Yoshizawa. "IMPACT:An interactive natural-motion picture dedicated Multimedia authoring system", CHI 91 Conference Proceeding, pp343-350, 1991.
- [9] Ioannis pitas, "Digital image processing alorithms", prentice hall, pp331-333, 1993.
- [10] John S. Boreczky and Lawrence A.Rowe, "comparison of video shot boundary detection techniques", in Storage & Retrieval for Image and Video Databases IV, Proc of SPIE 2670, pp. 170-179, 1996.
- [11] Kiyotaka Otsuji and Yoshinobu Tonomura. "Projection-detecting filter for video cut detection", Multimedia Systems vol 1, pp205-210, 1994.
- [12] Kiyotaka Otsuji, Yoshinobu Tonomura and Yuji Ohba, "video Browsing using brightness data". Proc of SPIE VCIP91, pp980-989, 1991.
- [13] Kyoji HIRATA, Toshikazu KATO, "Query by Visual Example-content based image Retrieval", Lecture Notes in Computer Science Vol. 580,

Advances in Database Technology-EDBT 92., pp56-71, 1992.

- [14] Little T.D.C, Ahanger G., Folz R.J, Gibbon J.F, Reeve F.W, Schelleng D.H, and Venkatesh D., A "Digital On-demand video service Supporting Content-based Queries", ACM Multimedia 93, pp427-436, 1993.
- [15] Peter Burt, James Bergen, Rajesh Hongorani, R.Kolczynski, W.Lee, A, Leung, J, Lubin, Shvayser. "Object Tracking with a moving camera", Proc IEEE Workshop on Visual Motion, pp2-12, 1989.
- [16] Phillip E.Mattison, "Practical Digital Video with Programming Examples In C", Wiley Professional computing, pp248-263, 1994.
- [17] Ramin Zabih, Justin Miller, and Kevin Mai, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", ACM Multimedia, pp189-200, 1995.
- [18] Rune Hjelsvold. "Video Information Contents and Architecture", 4th International Conference on Extending Database Technology, March 1994.
- [19] Rune Hjelsvold and Roger Midtstraum. "Modeling and Querying Video Data". Proc of 20th International Conference on VLDB, pp 686-694, October 1994.
- [20] Yoshinobu Tonomura, Akihito Akutsu, Yukinobu Taniguchi, Gen Suzuki, "Structured Video Computing", IEEE Multimedia FALL 94, pp34-43, 1994.



허 승

1995년 광운대학교 전자계산학과 학사
 1997년 광운대학교 전자계산학과 석사
 1997년~현재 고희정보통신연구소 연구원
 관심분야: 멀티미디어, 데이터베이스



김 우 생

1985년 University of Texas at Austin 전산학 학사
 1987년 University of Minnesota 전산학 학사
 1987년~1988년 현대전자, Zeus Computer Co. 과장
 1991년 University of Minnesota 전산학 박사
 1992년~현재 광운대학교 전자계산학과 부교수
 관심분야: 멀티미디어, 화상처리, 데이터베이스