효율적인 브라우징 환경을 위한 비디오 색인
(Video Indexing for Efficient Browsing Environment)

고 병 철 † 이 해 성 † 변 혜 란 ††
(ByongChul Ko) (Hae-Sung Lee) (Hyeran Byun)

요 약 최근 멀티미디어에 대한 관심이 증가하면서 그에 따른 기술 또한 매우 빠른 속도로 증가하고 있다. 특히 비디오 영상 검색 기능에 대한 사용자들의 욕구는 비디오에 대한 수동적인 접근 방식에서, 자신이 원하는 부분만을 선택적으로 검색할 수 있는 보다 편리한 환경을 요구하고 있다. 이를 위해서는 대용량의 비디오 데이타를 의미 있는 단위로 나누기 위한 비디오 파싱(Parsing)과 클러스터링(Clustering), 브라우징(Browsing)등을 포함하는 비디오 인덱싱 시스템의 구현이 필요하다. 본 논문에서는 우선 비디오 시퀀스를 히스토그램과 화소단위 비교법을 혼합한 하이브리드 방법을 통해서 자동 인덱싱을 위한 기본 단위인 샷(shot)으로 나눈다. 비디오 분할 후에 각 샷들로부터 대표 프레임을 검출한다. 대표 프레임은 사용자로 하여금 비디오의 전체적인 내용을 이해할 수 있도록 도와줌으로써 그 중요성이 크다고 할 수 있다. 따라서 본 논문에서는 웨이블릿 변환을 이용하여 우선적으로 샷 내에 포함된 카메라의 움직임을 분석하고, 각 프레임들의 변화량을 측정하여 샷의 복잡성에 따라 각기 다른 수의 대표 프레임을 선출하도록 하였다.

마지막으로 카메라 움직임중 패닝, 혹은 틸팅이 포함된 샷에 대해서 파노라마 영상을 합성함으로써 사용자에게 보다 편리하고 이해하기 쉬운 브라우징 환경을 제공할 수 있도록 하였다.

Abstract There is a rapid increase in the use of digital video information in recent years. Especially, user requires the environment which retrieves video from passive access to active access, to be more efficiently. we need to implement video retrieval system including video parsing, clustering, and browsing to satisfy user's requirement.

In this paper, we first divide video sequence to shots which are primary unit for automatic indexing, using a hybrid method with mixing histogram method and pixel-based method.

After the shot boundaries are detected, corresponding key frames can be extracted. Key frames are very important portion because they help to understand overall contents of video. In this paper, we first analyze camera operation in video and then select different number of key frames depend on shot complexity.

At last, we compose panorama images from shots which are containing panning or tilting in order to provide more useful and understandable browsing environment to users.

1. 서 론

최근 멀티미디어에 대한 관심이 증가하면서 그에 따른 기술 또한 매우 빠른 속도로 증가하고 있다. 특히 비디오 영상 검색 기능에 대한 사용자들의 욕구는 비디오에 대한 수동적인 접근 방식에서, 자신이 원하는 부분만을 선택적으로 검색할 수 있는 보다 편리한 환경을 요구하고 있다. 이를 위해서는 대용량의 비디오 데이타를 의미 있는 단위로 나누기 위한 비디오 파싱(Parsing)과 클러스터링(Clustering), 브라우징(Browsing)등을 포함하는 비디오 인덱싱(indexing) 시스템의 구현이 필요하다[1]. 기존의 비디오 데이타 분석은 사용자가 비디오를 직접 보면서 적절한 내용을 텍스트 형식으로 직접 입력함으로써 비디오를 분류했다. 이와 같은 방법은 좋은 성능을 가질 수 있는 시스템을 구성할 수 있지만 사용자에게 많은 시간과 노력을 요구한다. 따라서 최근에는 이

† 학생회원 : 연세대학교 컴퓨터과학과
soccer1@aipiri.yonsei.ac.kr
geneel@aipiri.yonsei.ac.kr
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
hrbyun@aipiri.yonsei.ac.kr
논문접수 : 1999년 5월 17일
심사완료 : 1999년 10월 4일

러한 사용자의 노력을 최소화 하기 위해 컴퓨터에 의한 자동 비디오 분류 기술이 연구 되고 있다[2]. 비디오 자동 인덱싱을 위한 기본 단위인 샷(shot)은 1대의 카메라로 기록되는 프레임들의 연속을 뜻하는 것으로 샷 경계면이 물리적으로 존재 하므로 연속적인 프레임들간의 물리적 특징(색상,화소,물체의 움직임)들을 이용하여 분할 할 수 있다.

비디오 분할 후에 각 샷으로부터 대표 프레임(Key frame)을 검출한다. 대표 프레임은 각 샷의 중요한 내용을 포함하고 있고, 샷의 내용의 복잡성에 따라 한 개 혹은 여러 개의 대표 프레임이 한 개의 샷으로부터 검출된다. 대표 프레임은 사용자로 하여금 비디오의 전체적인 내용을 이해할 수 있도록 도와줌으로 그 중요성이 크다고 할 수 있다[3]. 즉, 사용자로 하여금 몇 개의 하일라이트된 프레임만을 보여줌으로써 전체 비디오에 대해 빠른 브라우징 환경을 제공할 수 있고, 비디오 인덱싱을 위한 속도 또한 현저하게 감소 시켜 줄 수 있다[4].

하지만 패닝(Panning), 줌잉(Zooming)과 같은 카메라 움직임이 포함된 샷의 경우는 대표 프레임만으로는 그 의미를 사용자에게 전달하기가 부적절함으로 샷의 의미를 적절히 표현 할 수 있는 수의 대표 프레임을 추출하도록 하는 알고리즘이 필요하다. 아울러 패닝이 포함된 샷으로부터는 파노라마 영상을 합성함으로써 사용자에게 보다 편리하고 이해하기 쉬운 브라우징(Browsing) 환경을 제공 할 수 있다.

본 논문에서는 2장에서 히스토그램과 화소 단위 비교법의 장단점을 보완한 하이브리드(Hybrid) 방법을 이용하여 샷 경계면을 분할하고, 3장에서 웨이블릿 변환을 통한 카메라 움직임 분석과 , 각 샷의 프레임들간의 컬러 히스토그램 비교를 통해 최적의 대표 프레임을 추출하고 샷의 특성을 분석하여 패닝, 트랙킹과 같은 카메라 움직임이 포함된 영상에 대해서 파노라마 영상을 구성하는 방법을 제안한다. 마지막으로 4장에서는 본 논문에 대한 결론을 제시한다.

2. 샷 경계면 분할 기법

샷 경계면 분할 기법은 비디오 인덱싱을 위한 가장 기본적인 단계로서 현재 여러 가지 알고리즘들이 제안되고 있다. 이러한 분할의 목적은 비디오 스트림을 의미있고, 다루기 쉬운 샷들의 집합들로 나눔으로써 인덱싱을 위한 기본 단위를 제공하기 위함이다. 기존에 발표된 대표적인 샷 경계면 분할 기법을 소개하면 다음과 같다.

우선 히스토그램 비교법(Histogram Comparison)은 샷 경계면 분할 기법중 가장 일반적으로 사용되는 방법으로서, 비디오 영상에서 이웃하는 프레임들 사이의 색상 히스토그램의 차를 계산하여 장면 변화의 발생 여부를 판단한다[5] [6].

히스토그램 비교법은 화소단위 비교법에 비해 물체의 움직임이나 카메라 이동에 덜 민감하지만, 프레임 사이에 섬광이나 선명한 조명이 있을 경우 오류를 발생시킬 수 있다[6].

화소 단위 비교법(Pixel-wise comparison)은 연속적인 한 쌍의 프레임에서 대응하는 화소의 세기를 비교하여 얼마나 많은 변화가 발생하였는가를 측정하는 방식으로 매우 작은 움직임도 감지해낼 수 있지만, 물체의 이동 또는 잡음에 민감한 단점을 가지고 있다[7] [8]. 본 논문에서는 위의 두 가지 방법을 혼합하여 새로운 하이브리드 (Hybrid)방법을 제안한다[1].

2.1 하이브리드 비교법(Hybrid Comparison)

하이브리드 비교법은 히스토그램 비교법과 화소단위 비교법의 단점을 보완하기 위하여 제안하는 방법으로, 히스토그램 비교를 위한 임계값과 화소단위 비교를 위한 2개의 임계값을 사용하여 이 임계값들을 동시에 초과할 경우만을 샷 경계면으로 선언한다.

$$Diff\ f_{k,k+1} = \sum_{b=1}^{N_b} Dist(Hist_{r,g,b}(k), Hist_{r,g,b}(k+1)) > t_1 \quad \&\&- \quad DP_i(k,l) = \sum_{l=0}^{N\times N} |p_i(k,l) - p_{i+1}(k,l)| > t_2 \tag{1}$$

N_b: 히스토그램 범위, N: 영상 크기

t_1: 히스토그램 비교 임계치, t_2: 화소단위 비교 임계치

$Diff\ f_{k,k+1}$: k번째 프레임과 k+1번째 프레임간의 색상 히스토그램 차이

$DP_i(k,l)$: k번째 프레임과 k+1번째 프레임간의 화소값의 차이

하이브리드 방법에서는 히스토그램과 화소단위 비교법에서 사용된 임계값 보다 낮은 임계값이 적용된다. 이렇게 함으로써 각각의 방법에서 놓친 샷을 검출해 낼 수 있고 또한, 히스토그램 방법의 밝기 값에 의해 잘못 인식되는 샷 경계면 검출의 오류를 방지함으로써 보다 정확한 샷 경계면을 검출 결과를 얻을 수 있었다. 본 논문에서는 다양한 영화 비디오에 대하여 본 알고리즘을 적용하였다.

실험 결과는 표 1과 같다. 표에서 A 는 정적인 영화에 대한 결과이고 B는 동적인 영화에 대한 결과이다. 마지막으로 C는 카메라 패닝과 줌잉, 빠른 물체의 움직

임 등을 포함하도록 편집된 영상이다.

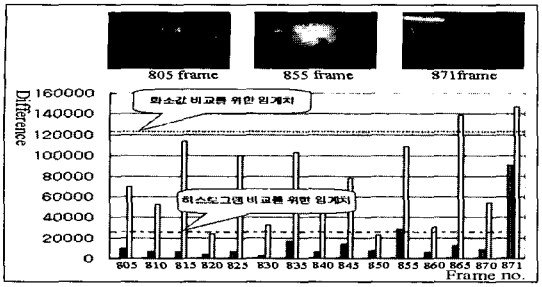


그림 1 하이브리드 비교법

표 1 하이브리드 방법을 이용한 샷 경계면 검출 결과
A-쇼생크 탈출 (2020 frame), B-컷스로트 아일랜드 (1800 frame),
C-편집된 비디오 (561-frame)

종류	검출방법	총샷	Missed	False Positive	Recall (%)	Precision (%)
A	히스토그램 비교법	14	0	0	100	100
	화소단위 비교법		1	8	86	43
	하이브리드 방법		0	0	100	100
B	히스토그램 비교법	11	1	7	88	50
	화소단위 비교법		0	5	100	64
	하이브리드 방법		1	2	90	81
C	히스토그램 방법	12	0	1	100	92
	화소단위 비교법		0	4	100	67
	하이브리드 방법		0	0	100	100

$$Recall = \frac{Detect - FalsePositive}{Detect - FalsePositive + Missed},$$

$$\Pr ecision = \frac{Detect - FalsePositive}{Detect} \quad (2)$$

본 논문에서는 평가기준으로 리콜(recall)과 프리시전(precision) 방법을 사용하였다. 리콜이 크다는 것은 올바른 샷 경계를 잃어버리지 않았다는 것을 뜻하며, 반대로 프리시전이 높다는 것은 잘못 검출된 샷 경계가 많지 않다는 것을 뜻한다. 따라서 두 값 중 하나가 높고 다른 하나가 상대적으로 낮다면 오히려 정확도가 떨어진다고 볼 수 있다[18]. 표1에서, 하이브리드 방법은 A와 같은 정적인 영화에서는 히스토그램과 동일한 정확도를 보여주지만, B와 같은 동적 영화에서는 다른 2가지 방법에 비해 리콜과 프리시전이 고르게 높은 결과를 보여주고 있다. 또, C와 같이 카메라움직임이 포함된 경우도 다른 2가지 방법에 비해 월등히 좋은 결과를 보여주고 있다. 결론적으로, 본 논문에서 제안하는 하이브리드 방식은 2개의 낮은 임계값을 사용하여, 여러 종류의 영화에 대해 강건성(robustness)을 갖도록 하는 것이다.

3. 웨이블릿 변환을 통한 카메라 움직임 분석과 색상 히스토그램을 이용한 최적의 대표 프레임 추출

일반적으로 샷 경계면이 검출된 후에 각 샷들은 대표 프레임에 의해 나타내어지게 된다. 하지만, 대부분의 대표 프레임은 각 샷의 첫 번째, 혹은 마지막 프레임으로 결정되므로 매우 긴 패닝(Panning)이나 줌(Zoom)이 포함된 샷의 경우, 한 개나 두 개의 대표 프레임만으로 전체의 샷을 표현하는 것은 무리이다[9].

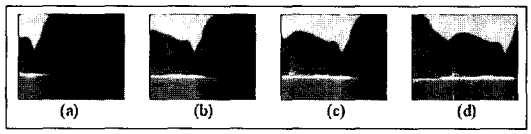


그림 2 패닝이 포함된 샷

그림2는 패닝이 포함된 샷으로, 첫 번째 프레임이나 마지막 프레임을 대표 프레임으로 선정할 경우, 샷의 동적인 특성을 효율적으로 표현하지 못하게 된다. 따라서 샷의 특성을 분석하고, 그 특성에 따라 각각 다른 기준을 적용하여 대표 프레임을 선출하는 기술이 필요하다. 이를 위해서는 우선적으로 샷 내에 포함된 카메라의 움직임을 분석하고, 각 프레임들의 변화량을 측정하여 샷의 복잡성에 따라 각기 다른 수의 대표 프레임을 선출해야 한다.

3.1 기존의 대표 프레임 검출 기법

H. Zhang[10]은 샷의 마지막 프레임을 기준으로, 연속적인 프레임들의 컬러 값을 비교하여, 두드러진 변화가 발생할 경우 해당 프레임을 새로운 대표 프레임으로 선출하는 방식을 택하고 있다. 하지만 이 방법의 경우 움직임 정보에 대해서는 고려하지 않고 있으므로, 카메라 움직임이 포함된 샷에 대해서는 의미적으로 정확한 대표 프레임을 검출하기가 힘들다.

Wolf[11]는 움직임에 기반한 대표 프레임 검출을 제안하고 있다. 이 방법은 우선 각 프레임에 대해서

optical flow를 계산하고 시간에 따른 움직임 값을 분석한 뒤 움직임 값이 작은 부분의 프레임을 대표 프레임으로 추출한다. 이것은 중요한 인물이거나, 등장인물의 중요성을 강조하기 위해 특정 자세를 유지하는 동안 카메라는 해당 인물에 대해 오랫동안 멈춰 있으므로, 움직임 값이 작게 된다는 가정을 두고 있다.

이 방법의 경우 매 프레임마다 optical flow를 계산하여야 하므로 상당한 계산 시간을 필요로 할뿐만 아니라, 기본 가정인 움직임 값이 작은 부분이 항상 의미적으로 중요한 대표 프레임이 아닐 경우가 발생 할 수 있다.

따라서 본 논문에서는 우선 각 샷의 특성을 분석하여, 카메라 움직임이 포함된 샷과 일반적인 샷에 대해 다른 방법을 적용함으로써 의미적으로 샷의 내용을 정확히 표현 할 수 있는 최적의 대표 프레임 추출 알고리즘을 제안한다. 또한, 패닝 혹은 틸팅과 같은 카메라 움직임이 포함된 샷에 대해서는 효율적인 브라우징 환경을 제공하기 위해 파노라마(Panorama)영상을 합성하는 알고리즘을 제안한다.

3.2 웨이블릿 변환을 이용한 카메라 움직임 검색

카메라 움직임은 크게 카메라 고정 여부에 따라 패닝(Panning), 틸팅(Tilting), 줌잉(Zooming)과 트랙킹(Tracking),부밍(Booming), 돌링(Dolling)등으로 나뉘어진다.

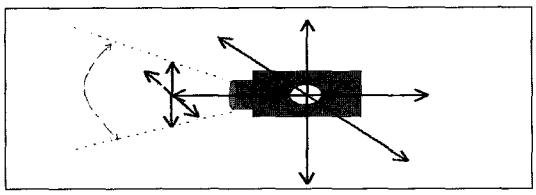


그림 3 기본적인 카메라 동작

H. Zhang[12]은 각 프레임들의 움직임 벡터를 분석하여 카메라 움직임을 구분하는 움직임 벡터 분석법을 제안하였다. 카메라 패닝의 경우 움직임의 방향이 일정하므로 특정방향의 벡터(modal vector)성분을 많이 갖게 되고, 따라서 해당 프레임에서의 움직임 벡터들과 모달벡터의 차이 값이 작을수록 한 방향으로 일정한 움직임이 발생한 것이므로 이때의 카메라 움직임을 패닝 혹은 틸팅등으로 판단하였다.

또한 줌의 경우, 줌이 발생 할 때 움직임 벡터 값이 0이 되는 초점의 중앙이 프레임 안에 위치하고, 프레임의 극 상하단 수직 벡터 값과, 극 좌우(left,right-most) 수평 벡터 절대값의 차이가 각 벡터들의 최대값 보다 큰, 위 2가지 성질들을 모두 만족할 경우 최종적으로 줌으로 판단한다.

Akutsu et al.[4]은 우선 연속적인 프레임들 사이에 움직임 벡터를 구한 뒤 이것을 다시 Hough Transform을 이용한 극좌표계로 나타내어 직선 성분들에 대한 각도 분포를 통해 패닝과 줌을 판단하였다.

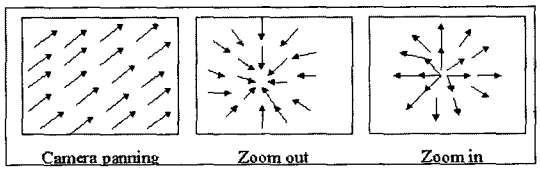


그림 4 카메라 동작에 따른 optical flow

이와 같은 기존의 방법들은 모두 연속적인 프레임으로부터 움직임 벡터를 추출하여 일정한 식에 의해 최종적으로 카메라 움직임을 판단하지만, 움직임 벡터는 노이즈에 민감하고, 상당한 계산 시간을 요구함으로 속도의 측면에서 비효율적이다. 또한, 이러한 이론들은 비디오 영상에 두드러진 물체의 움직임이 없다는 가정 하에 유도되었기 때문에 , 만약 커다란 물체가 빠른 속도로 움직이는 장면이 포함된 영상의 경우 잘못된 결과를 나타낼 수도 있다[13]. 기존의 이러한 한계들을 극복하기 위하여 본 논문에서는 웨이블릿 변환(Wavelet transform)에 기반한 알고리즘을 사용하여, 움직임 벡터의 계산시간을 줄일 수 있었다. 또한 블록간의 절대값 차이를 이용하여 물체의 움직임과 카메라 움직임을 구별해 내는 알고리즘을 개발하였다.

3.2.1 웨이블릿 변환에 기반한 빠른 움직임 벡터 산출 기법

패닝이나 줌과 같은 카메라 움직임을 분석하기 위해, 본 논문에서는 웨이블릿 변환을 이용하여 얻어진 lowpass부분에 블록매칭(block matching) 방법중 FSA(*Full Search Algorithm*)를 사용하였다[14].

그림 5 웨이블릿 변환된 영상(level 2)

FSA는 지역적 극소(local minimum) 문제의 영향을 받지 않는 가장 정확한 움직임 측정방법으로 알려져 있다.

그림5에서와 같이 웨이블릿 변환을 이용하여 생성되는 각 sub-image에서 lowpass영역만으로 움직임 벡터를 구했을 때 총 계산 시간은 Pentium-Ⅱ 233 환경에서, 전체 영상을 사용했을 경우(3.09초) 보다 level-2단계의 웨이블릿 변환 과정을 포함(0.25초)하여 약 12배 향상되었다.

또한, 그림 6,7에서 보는 것과 같이 전체 영상으로부터 구해진 움직임 벡터 값과 lowpass영역으로부터 구해진 움직임 벡터 값도 큰 차이를 보이지 않았다.

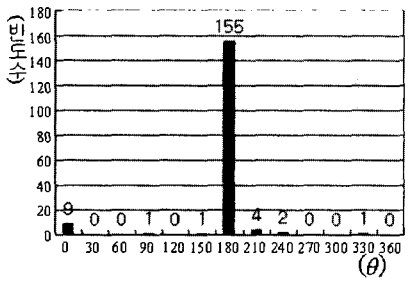


그림 6 전체 영역으로부터 구해진 움직임 벡터

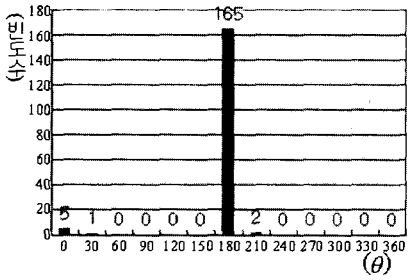


그림 7 Lowpass영역으로부터 구해진 움직임벡터

따라서, 본 논문에서 제안하는 방법을 사용 할 경우 비록 FSA를 사용하더라도 상당한 시간을 절약할 수 있고, 카메라 움직임을 찾아내기에 정확한 움직임 벡터를 구할 수 있다.

본 논문에서는 블록간 유사도 평가함수(Cost function)로 MAD(Mean Absolute Difference) 방법을 사용하였고, 블록의 크기를 4x4, 탐색 윈도우(Search Window)의 크기를 8x8로 설정하였다. 입력 영상의 크기는 320x210 이고 블록매칭을 위해 레벨 2단계(4배 축소)의 lowpass 부분(80x53)만을 사용하였다.

$$MAD(d_x,d_y)=\frac{1}{mn}\sum_{i=-\frac{n}{2}}^{\frac{n}{2}}\sum_{j=-\frac{m}{2}}^{\frac{m}{2}}|F(i,j)-G(i+d_x,j+d_y)| \quad (3)$$

$F(i,j)$: 현재 프레임에서 $m\times n$ 크기의 블록

$G(i,j)$: 이전 프레임에서 $m\times n$ 크기의 블록

d_x, d_y : 검색 위치를 나타내는 벡터

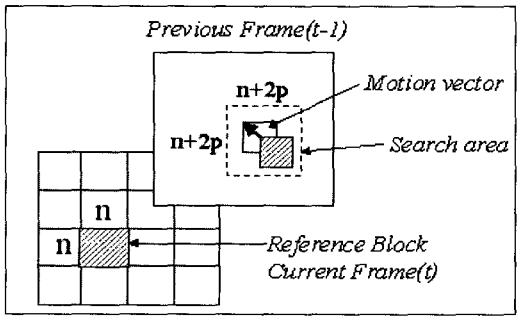


그림 8 블록정합 움직임 추정과정

3.2.2 카메라 움직임과 물체 움직임 구별을 위한 기법

기존의 카메라 움직임 분석방법의 대부분은 영상에 물체의 움직임이 없다는 가정에 기초하지만, 실제로 대부분의 비디오에는 커다란 물체의 움직임을 포함하는 샷들이 많이 존재한다. 이 경우 움직임벡터는 물체의 운동 방향으로 크게 나타나므로, 기존의 방법들을 사용할 경우 잘못된 결과를 나타낼 수 있다. 따라서 본 논문에서는 이미지의 각 부분에 9개의 NxN 크기를 갖는 고정된 대표 블록을 만들고 프레임간의 차이 값을 측정하여, 그 값들을 크기순으로 정렬한 뒤, 가장 값이 작은 3개 블록들의 평균 차이 값이 기준치 이상이 될 경우에만 카메라 움직임 중 패닝의 후보 프레임으로 선언한다[14].

이론적 가정을 설명하자면, 카메라 움직임이 일어날 경우에는 화면 전체가 동시에 움직이므로, 가장 값이 작은 3개 블록의 변화량일지라도 모두 기준치 이상이 되고, 반대로 정지된 화면에 커다란 물체의 움직임만 일어났을 경우에는 가장 값이 작은 3개 블록의 변화량은 0에 가깝게 될 것이므로 이를 카메라 움직임과 구별 할 수 있다는 것이다.

줌의 경우는 움직임 벡터가 초점의 중앙부분에서 '0' 값을 갖고, 주변 부분에서는 분산된 형태의 값을 갖기 때문에 그림9에서와 같이 점진적 장면 전환인 디졸브(Dissolve)를 줌으로 잘못 판단할 경우가 발생한다. 따라서 각 영상에 9개의 대표 블록(그림10)을 설정하여 각 블록간의 변화량 차이 값을 계산하고 이들의 평균, 분산을 구한 뒤에 분산 값이 기준치 이상이 될 경우를 줌의 후보 프레임으로 선언한다.

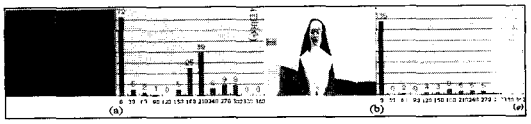
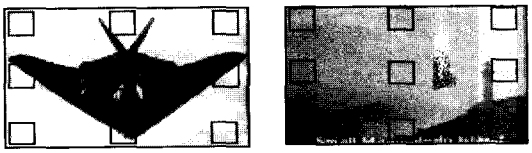


그림 9 (a) 디졸브의 경우 움직임 벡터 (b) 줌의 경우 움직임 벡터

이것은 줌의 경우 9개 대표 블록 중 카메라의 초점이 위치하는 부분에서는 블록 변화량이 작은 반면, 디졸브 발생시는 영상 전체가 같은 비율로 변하므로 줌과는 다르게 9개 블록의 변화량이 균일한 값을 가지게 되고, 각 블록의 분산 값을 계산했을 때 '0'에 가까운 값을 갖는 특성을 이용한 것이다.



(a) (b)

그림 10 물체의 움직임(a)과 카메라 움직임(b)이 포함된 영상 구분을 위한 9개의 블록

$$Diff=\frac{1}{N\times N}\sum_{(x,y)}^{N\times N}|f(x,y)-f'(x,y)| \quad (4)$$

$Diff$: 블록간의 차이값 , N : 블록의 픽셀 수

$f(x,y)$: 현재 프레임 블록의 (x,y)위치 픽셀

$f'(x,y)$: 다음 프레임 블록의 (x,y)위치 픽셀

3.2.3 카메라 움직임 검출

웨이블릿 변환을 이용하여 움직임 벡터를 산출하고, 영상의 특정 블록들간의 변화량을 측정한 뒤에 이 정보들을 이용하여 카메라 움직임을 판별할 수 있다.

우선, 첫 번째 단계에서 웨이블릿 변환과 FSA방법에 의해 얻어진 움직임 벡터 값이 0이 아닌 블록들의 수를 계산하여, 이 값이 정해진 임계값(본 논문에서는 총 블록 수의 1/2로 정했다) 이상이 되고, 블록간의 변화량에서 가장 작은 3개의 평균값이 또 다른 임계값 이상이 될 경우 후보 카메라 움직임, 즉 패닝으로 판단하고 동시에 각 블록의 분산 값이 임계값 이상이 될 경우 후보 줌으로 판단한다.

이렇게 얻어진 후보 카메라 움직임 영상에 H. Zhang이 제안한 식을 개선한 식(5)를 적용한다. 이때 프레임 사이의 차이 값이 0에 가깝고 일정 프레임동안 지속되는 특성을 이용하여, 차이 값이 0에 가까운 프레임들이 주어진 시간적 간격(Time Duration)이상 동안 연속적으로 나타나면 최종적인 패닝으로 판단한다. 줌의 경우는 개선된 식(6)를 적용하여 차이 값이 주어진 임계값 이하이고 역시 연속적인 시간적 간격을 만족시킬 경우 최종적인 줌으로 판단한다. 여기서 시간적 간격을 두는 것은 카메라의 흔들림 등으로 인해 발생 할 수 있는 잘못된 패닝과 줌을 방지하기 위한 것이다.

$$\sum_{b=1}^{N}|\theta_b-\theta_m|\le\Theta_p \quad (5)$$

θ_b : 해당 프레임에서의 움직임벡터

θ_m : 해당 프레임에서의 모달벡터

Θ_p : 카메라 움직임 감지를 위한 임계치

N : 총 Block 수

$$|v_k^t|-|v_k^b|\le T_1 \quad (6)$$
$$|v_k^l|-|v_k^r|\le T_2$$

T_1, T_2: 줌 판단을 위한 수직, 수평 방향 임계값

v_k^t, v_k^b: k^{th} *frame*의 수직방향 최상위, 최하위 y 벡터

v_k^l, v_k^r: k^{th} *frame*의 수평방향 극좌, 극우 x 벡터

이 과정을 요약하면 그림11과 같다.

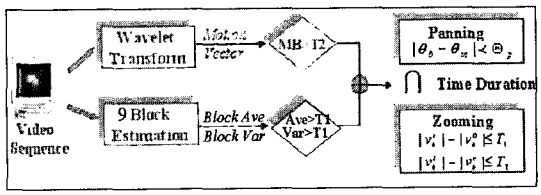


그림 11 카메라 움직임 검출 단계

3.3 카메라 움직임과 색상 히스토그램을 이용한 대표 프레임 추출

3.3.1 카메라 움직임이 포함된 샷에서의 대표 프레임 선출 기법

샷 경계면 분할 작업이 끝나고, 각 샷에 대한 카메라 움직임을 추정하여, 카메라 움직임이 발생한 샷과 일반 샷에 따라 다른 알고리즘을 적용하여 대표 프레임을 선출한다. 일반적으로 패닝 영상은 상대적으로 중요한 인물이나 배경을 따라 카메라가 한쪽 방향으로 이동한 것으로, 샷의 첫 프레임만으로는 그 의미를 파악하기 힘들기 때문에 첫 번째, 중간, 마지막 프레임을 대표 프레임으로 선출했다. 줌의 경우는 제작자가 주인공의 감정이나 현재 상황을 강조하기 위해서 사용된다[15]. 따라서 시작 프레임과 가장 많이 확대/축소된 마지막 프레임을

대표 프레임으로 선정함으로써 샷의 의미적 내용을 잘 표현하도록 하였다.

3.3.2 일반적인 샷에 대한 대표 프레임 선출 기법

카메라 움직임이 없는 일반적인 샷에 대해서는 연속적인 프레임들 사이에 컬러 히스토그램의 비유사성 정도를 측정하고 두드러진 변화가 발생하는 프레임을 대표 프레임으로 선택하는 방법을 사용하였다[10]. 이 경우 변화가 적은 비디오 샷이라면, 1개의 대표 프레임만이 선출될 것이고 , 반대로 긴 비디오 샷이거나 변화량이 많은 비디오 샷일 경우는 더 많은 프레임들을 대표 프레임으로 선출하게 된다. 이 과정을 요약하면 그림 9와 같다.

그림12와 같이 각 샷의 첫 번째 프레임이 우선적으로 후보 대표 프레임으로 선출되고, 연속적인 다음 프레임들과 R,G,B 공간상의 히스토그램을 계산한 뒤 이들간의 비유사성(Dissimilarity)을 계산하여 이 값이 미리 정의된 임계값을 초과할 경우 또 다른 후보 대표 프레임으로 인정하고 이 프레임을 기준으로 다시 연속적인 프레임간의 비유사성 정도를 측정한다. 이때 비유사성 측정 방법은 식(7)을 사용하였다.

$$Diss = \frac{\sum_{i=0}^{255}(|R_i^* - R_i''| + |G_i^* - G_i''| + |B_i^* - B_i''|)}{256} \quad (7)$$

$Diss$: 비유사성 측정치

R_i^*, G_i^*, B_i^*: 기준 프레임의 색상값,

R_i'', G_i'', B_i'': 후속 프레임의 색상값

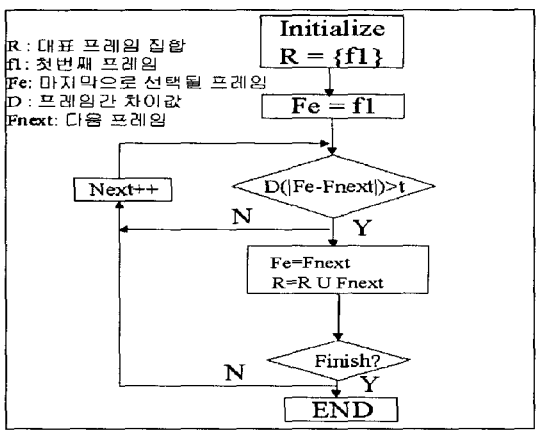


그림 12 대표 프레임 선출을 위한 알고리즘

이때 한 샷에서 추출될 수 있는 최대 대표 프레임의 수를 4개로 제한하였다. 이것은 추출된 대표 프레임들간의 유사성 측정을 통해 하나의 장면(Scene)을 구성 할 때 시간적인 효율을 고려하기 위해서다.

최종적으로 본 논문에서 제안하는 알고리즘과 일반적인 색상 변화량만을 고려하는 알고리즘과의 비교 결과는 그림 13, 14와 같다.

3.4 효율적인 브라우징 환경을 위한 파노라마 영상 구성

최근 연속적인 이미지들로부터 파노라마(Panorama) 영상을 구성하는 기술이 큰 관심을 모으고 있다. 파노라마 영상이란 자동으로 여러 개의 이미지를 정렬시키거나 겹치게 함으로써 만들어지는 영상을 말한다[16].

일반적으로 샷 경계면이 검출된 후에 각 샷들은 대표 프레임에 의해 나타내어지게 된다. 하지만, 대부분의 대표 프레임은 각 샷의 첫 번째, 혹은 마지막 프레임으로 결정되므로 매우 긴 패닝(Panning)이 포함된 샷의 경우, 장면 클러스터링(Clustering)을 위한 사용이외에 단지 사용자 브라우징을 목적으로 할 경우, 한 개나 두 세 개의 대표 프레임만으로 전체의 샷을 표현하는 것은 무리이다[14]. 따라서 파노라마 영상을 구성함으로써 사용

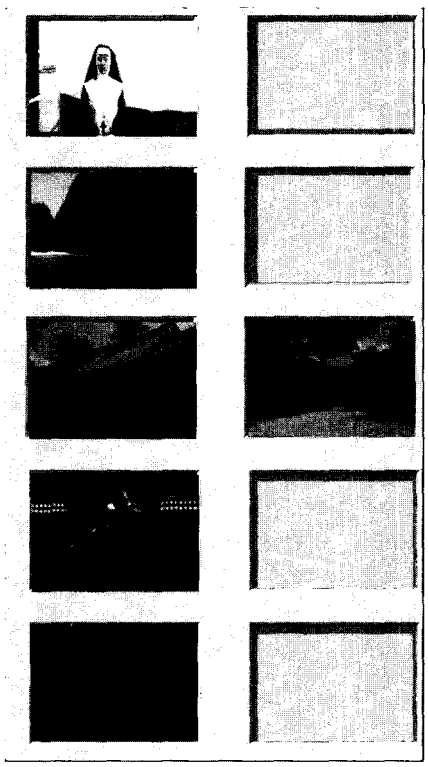


그림 13 일반적인 컬러 값 변화만을 고려했을 경우 대표 프레임 추출 결과

자에게 보다 편리한 비디오 브라우징 환경을 제공할 수 있다.

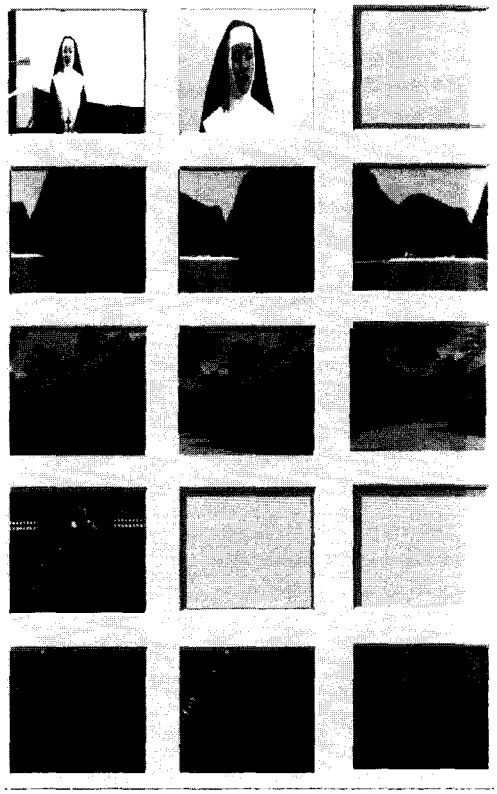


그림 14 카메라 움직임 분석 및 컬러 값 변화량을 고려했을 경우 대표 프레임 추출 결과(최 상단 : 줌, 2,3,5행 : 패닝)

3.4.1 움직임 벡터 보정

본 논문에서는 연속적인 비디오 프레임으로부터 움직임 벡터를 계산해 내기 위해서 원 영상을 웨이블릿 변환시키고 lowpass부분의 정보만을 이용하여 움직임 벡터 값을 추정하는 MRME (Multire-solution Motion Estimation)방법을 사용하였다. 이 경우 움직임 벡터 계산 시간을 현저하게 줄일 수 있다. 하지만, 계산된 움직임 벡터 값은 낮은 해상도(Resolution)에서의 값이므로 파노라마 영상을 구성하기 위해서는 원래의 움직임 벡터 값이 필요하다.

본 논문에서는 웨이블릿 변환을 레벨 2 단계까지 실시하였으므로, 아래와 같은 식을 이용하여 원 영상에서의 움직임 벡터 값을 보정 할 수 있다.

$$V_x = 4V_4^L(x) + \Delta_x \quad (8)$$

$V_y = 4V_4^L(y) + \Delta_y$

V_x, V_y : 실제영상에서의 움직임 벡터값

Δ_x, Δ_y : 오차 벡터

$V_4^L(x), V_4^L(y)$: *lowpass*부분에서의 움직임 벡터값

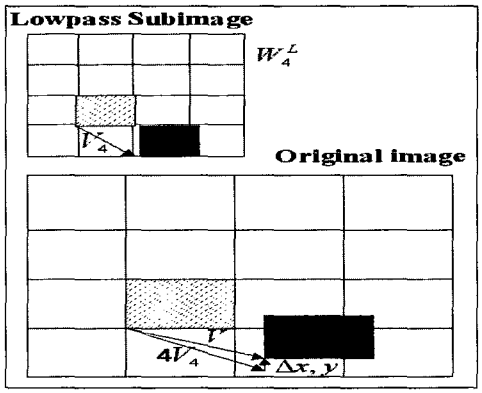


그림 15 lowpass subimage로부터 움직임 벡터 보정

3.4.2 파노라마 영상 합성

웨이블릿 변환을 이용하여 계산된 움직임 벡터 값을 원 영상에 대해 보정한 후, 움직임 벡터 값에 따라 해당 프레임들을 연속적으로 위치시킴으로써 패닝, 트랙킹 등이 포함된 샷에 대한 파노라마 영상을 얻을 수 있다. 이 때 합성될 연속적인 프레임들의 위치는 식(9)에 의해 구해진다[17].

$$(x', y') = (x, y) + (d_x, d_y) \quad (9)$$

(x', y') : 다음 프레임의 위치

(x, y) : 현재 프레임의 위치

d_x, d_y : 움직임 벡터들의 평균값

3.4.3 파노라마 영상 합성 결과

카메라 움직임중 패닝, 틸팅 등이 포함된 샷을 검출하고, 각 샷에 대하여 파노라마 영상을 합성한 결과는 그림16, 17 ,18 와 같다

4. 결 론

본 논문에서는 지금까지 비디오를 샷 단위로 분할하고 카메라 움직임 검출을 통한 의 미 있는 대표 프레임 선출 및 파노라마 영상의 합성 방법에 대하여 살펴보았다.

샷 경계면 검출의 경우 기존의 히스토그램 비교법과

그림 16 좌에서 우로 찍은 패닝 영상(# 24 frame)

그림 17 우에서 좌로 약 15 °각도로 촬영한 패닝 영상(# 30 frame)

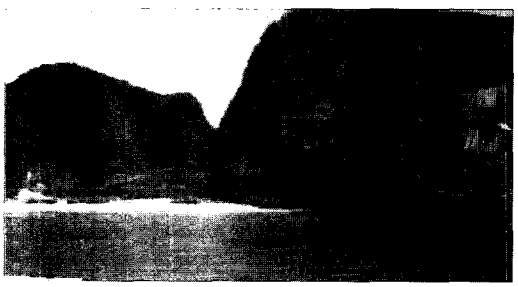


그림 18 우에서 좌로 찍은 패닝 영상(# 68 frame)

화소단위 비교법을 혼합한 하이브리드 방식을 통해 빛과 물체의 움직임에 덜 민감한 샷 경계면 검출을 할 수 있었다. 하지만 보다 정확한 샷 검출을 하기 위해서는 페이드(Fade), 와이프(Wipe), 디졸브(Dissolve)와 같은 특수 효과도 쉽게 검출 할 수 있는 알고리즘이 더 연구되어야 한다.

의미적으로 연관성 있는 대표 프레임을 선출하기 위해서 각 샷의 색상 변화량을 측정하고, 아울러 카메라 움직임을 분석하여 그 종류에 따라 다른 수의 대표 프레임을 선출하도록 함으로써 대표 프레임 브라우징(Browsing) 및 장면구축(Scene Clustering)의 효율성을 높일 수 있었다.

카메라 움직임을 빠른 시간에 검출하기 위하여 웨이블릿 변환을 이용하여 low-pass 부분에서만 움직임 벡터를 계산하므로 그 동안 움직임 벡터 계산 과정에서 커다란 문제점이 되었던 속도 문제를 상당 부분 해결할 수 있었고, 또한 대표 블록간의 화소 변화량 측정을 통해 카메라 움직임과 물체의 움직임을 구분해 냄으로써 잘못된 카메라 움직임을 방지 할 수 있었다

마지막으로 이렇게 측정된 카메라 움직임이 포함된 샷에 대해, 보다 이해하기 쉬운 사용자 브라우징 환경을 제공하기 위하여 파노라마 영상을 합성하였다.

참 고 문 헌

[1] 고병철, 변혜란, "비디오 샷 경계면 분할기법 비교를 통한 대표 프레임 추출" *'98 정보과학회 가을 학술발표 논문집 (II)* ,pp 512 - 514

[2] 허승, 김우생, "윤곽선과 컬러 분포를 이용한 비디오 분할과 비디오 브라우징", *정보처리학회논문지 제4권 제9호 ('97.9)*,pp 2197-2207, 1997

[3] Anita Komlodi, Gary Marchionini, "Key frame preview techniques for video browsing," in *Proc.* ACM conference on Digital Libraries(Digital Libraries '98), pp 118 -125, 1998

[4] HongJiang Zhang, J.Y.A Wang and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," in *Proc. IEEE Int. Conf. on Image Proc.*, 1997

[5] Yong Rui,Thomas S. Huang and Sharad Mehrotra, "Exploring Video Structure Beyond the Shots," *IEEE Multimedia System '98,* pp237-240, 1998

[6] Thomas S. Huang, Yong Rui, Trausti Kristjansson, Milind Naphande and Yueting Zhuang, "Video Analysis and Representation," *Proposal of University of Illinois at urbana-champaign.* 1998

[7] 이재현, 장옥배 "움직임 벡터를 사용한 점진적 장면전환 검출", *정보과학회 논문지(C) 제 3 권 제2호(97.4),* 1997

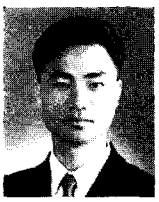
[8] John S. Boreczky and Lawrence A. Rowe, "Comparison of Video Shot boundary Detection Techniques," *Ph.D. dissertion, University of Califonia at Berkeley,* 1996.

[9] Myron Flicker *et al.* "A Query by Image and Video Content: The QBIC System" *Intelligent Multimedia Information Retrieval,* pp. 7-22, 1997

[10] H. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognition, vol.30*, no. 4, pp. 643-658, 1997

[11] W. Wolf, "Key frame selection by motion analysis," *in Proc. IEEE Int. Conf. Acoust.,Speech, and Signal*

Proc., 1996
[12] HongJiang Zhang, Chein Yong Low and Stephen W.Smoliar, "Video Parsing and Browsing Using Compressed Data," *Multimedia Tools and Application,* vol 1,pp 89-111, 1995
[13] F. Idris and S. Panchanathan, "Review of Image and Video Indexing Techniques". *Journal of Visual Communication and Image Representation* Vol. 8, No 2, June, pp. 146-166, 1997
[14] 고병철,이해성,변혜란, "웨이블릿 변환을 이용한 비디오 파노라마 영상 구성" *HCI '99 학술대회 발표 논문집*,pp 187-190 1999. 2
[15] 이용현, 강행봉, 박용진, "영역 정보를 이용한 비디오 데이타의 카메라 모션 검출 및 대표 프레임 선택 방법" *'98 가을 학술 발표 논문집(II) 제25권 2호* pp, 315-317. 1998
[16] Brown, L.G. "A Survey of Image Registration Techniques," *ACM Computing Surveys*, Vol. 24, No. 4, pp 325-376, 1992
[17] Yukinobu Taniguchi , Akihito Akutsu , Yoshinobu Tonomura, "PanoramaExcerpts: Extracting and Packing Panoramas for Video Browsing," *ACM Multimedia-Electronic Proceeding 97*, 1997
[18] 이재연, 전병태, 배영래, "복수의 검출기를 사용하는 동영상의 샷 경계 자동 검출 알고리즘," 정보과학회 논문지(B), 제 25권 제9호(98.9) p.1352-1360



고 병 철
1998년 경기대학교 전자계산학과 졸업(이학사). 1998년 ~ 현재 연세대학교 대학원 컴퓨터과학과 석사과정 재학중. 관심분야는 영상검색, 비디오 인덱싱, 컴퓨터 비젼, 인공지능

이 해 성
1995년 연세대학교 물리학과 졸업(이학사). 1995년 ~ 1997년 삼성코닝(주) 연구소 근무(연구원). 1997년 ~ 1999년 연세대학교 대학원 인지과학과 졸업(공학석사). 1999년 ~ 현재 연세대학교 대학원 컴퓨터과학과 박사과정 재학중. 관심분야는 웨이블릿 이론 및 응용, 컴퓨터 그래픽스, 영상처리, 컴퓨터 비젼, 신호처리, 인공지능

변 혜 란
1980년 연세대학교 수학과 졸업(이학사). 1983년 연세대학교 대학원 수학과 졸업(이학석사). 1987년 Univ. of Illinois, Computer Science(M.S.). 1993년 Purdue Univ., Computer Science (Ph.D.). 1994년 ~ 1995년 한림대학교 정보공학과 조교수. 1995년 ~ 1998년 연세대학교 컴퓨터과학과 조교수. 1998년 ~ 현재 연세대학교 컴퓨터과학과 부교수. 관심분야는 인공지능, 영상인식, 영상처리