

# 비디오 시퀀스에서 움직임 객체 분할과 VOP 추출을 위한 강력한 알고리즘

## (A Robust Algorithm for Moving Object Segmentation and VOP Extraction in Video Sequences)

김 준 기 \* 이 호 석 \*\*  
(Junki Kim) (Ho Suk Lee)

**요 약** 비디오 객체 분할은 MPEG-4와 같은 객체기반 비디오 코딩을 위한 중요한 구성 요소이다. 본 논문은 비디오 시퀀스에서 움직임 객체 분할을 위한 새로운 알고리즘과 VOP(Video Object Plane)추출 방법을 소개한다. 본 논문의 핵심은 시간적으로 변하는 움직임 객체 에지와 공간적 객체 에지 검출 결과를 효율적으로 조합하여 정확한 객체 경계를 추출하는 것이다. 이후 추출된 에지를 통하여 VOP를 생성한다. 본 알고리즘은 첫 번째 프레임을 기준영상으로 설정한 후 두 개의 연속된 프레임 사이의 움직임 픽셀 차이 값으로부터 시작된다. 차이영상을 추출한 후 차이영상에 Canny 에지 연산과 수리형태 녹임 연산(erosion)을 적용하고, 다음 프레임의 영상에 Canny 에지 연산과 수리형태 녹임 연산을 적용하여 두 프레임 사이의 에지 비교를 통하여 정확한 움직임 객체 경계를 추출한다. 이 과정에서 수리형태학 녹임 연산은 잘못된 객체 에지의 검출을 방지하는 작용을 한다. 두 영상 사이의 정확한 움직임 객체 에지(moving object edge)는 에지 크기를 조절하여 생성한다. 본 알고리즘은 픽셀 범위까지 고려한 정확한 객체의 경계를 얻음으로서 매우 쉬운 구현과 빠른 객체 추출을 보였다.

**키워드** : 비디오 객체 분할, MPEG-4, 차이 영상, 캐니 에지, 수리형태학 연산

**Abstract** Video object segmentation is an important component for object-based video coding scheme such as MPEG-4. In this paper, a robust algorithm for segmentation of moving objects in video sequences and VOP(Video Object Planes) extraction is presented. The points of this paper are detection of an accurate object boundary by associating moving object edge with spatial object edge and generation of VOP. The algorithm begins with the difference between two successive frames. And after extracting difference image, the accurate moving object edge is produced by using the Canny algorithm and morphological operation. To enhance extracting performance, we apply the morphological operation to extract more accurate VOP. To be specific, we apply morphological erosion operation to detect only accurate object edges. And moving object edges between two images are generated by adjusting the size of the edges. This paper presents a robust algorithm implementation for fast moving object detection by extracting accurate object boundaries in video sequences.

**Key words** : video object segmentation, MPEG-4, difference image, Canny edge, morphology operation

### 1. 서 론

ISO/IEC SC29 WG11 동영상기술의 국제 표준화 그

룹에서는 MPEG-1, MPEG-2의 성공에 힘입어 통신, 방송, 컴퓨터 응용분야에 공통적으로 적용될 수 있는 새로운 영상기술을 개발하고 있다. 최근 고성능 멀티미디어 기기들의 활용과 무선 디지털 통신, 데이터베이스 등의 급속한 신장으로 인하여 기존 영상기술의 핵심기능인 압축부호화 기능뿐 아니라 고도의 새로운 기능들(영상분할, 내용인식, 3차원 편집)을 지원할 수 있도록 해주는 새로운 압축 방식의 필요성이 대두되고 있다. 그러나 이전의

\* 비 회 원 : 호서대학교 컴퓨터공학부  
kjk73@hanmail.net

\*\* 정 회 원 : 호서대학교 컴퓨터공학부 교수  
hslee@office.hosco.ac.kr

논문접수 : 2001년 10월 22일  
심사완료 : 2002년 4월 9일

MPEG-1, MPEG-2와 같은 압축기술은 영상에 담긴 내용과는 무관하게 프레임 단위로 압축하는 방식이다. 영상에 담긴 객체의 내용에 대한 이해와 구별 없이 프레임 단위로 처리하는 방법은 각 객체의 비트스트림에 대한 접근, 객체 기반 비트스트림의 조작, 객체와의 다양한 상호작용, 장면조합에 의한 영상 재사용등의 기능상 제한을 주기 때문에 멀티미디어 통신, 방송, 영화 등이 요구하는 다양한 기능을 충족시키지 못한다. 이처럼 다양한 요구를 충족시키기 위하여 동영상기술의 국제 표준화 그룹에서는 새로운 MPEG-4 기술을 채택하게 되었다. MPEG-4는 영상의 객체를 바탕으로 부호화하는 내용기반 부호화(content-based coding)를 통하여 효율적인 압축을 제공한다.

MPEG-4 기능을 크게 셋으로 나누면 객체지향 대화형, 고비율 압축, 범용 액세스 등이다. 객체 지향 상호작용 기능은 멀티미디어(주로 비디오나 오디오) 데이터 접근에 있어서 화면이나 음향을 자유로이 구성할 수 있는 기능을 말한다. 예를 들어 화면에서 배경을 그대로 둔 채 주인공만을 교체하는 등의 처리가 이전까지는 프로덕션 단계에서만 가능했으나 MPEG-4에서는 사용자 단계에서 가능해진다. 고비율 압축에 있어서는 기존의 방식보다 개선된 객체 기반 압축을 수행하기 때문에 높은 압축률을 제공한다. 또 범용 액세스에 있어서는 무선통신 환경을 고려해 채널 에러가 많은 환경에서도 내성이 강하도록 해야 한다. 이러한 기능들을 모두 만족시키는 단일 알고리즘은 사실상 불가능하므로 MPEG-4에서는 많은 기능요소들을 표준 메뉴형식으로 제공해 사용자의 응용에 따라 선택해 사용하도록 하고 있다.

MPEG-4 표준의 내용기반 비디오 부호화는 매우 낮은 비트율에서 더욱 정확한 비디오 표현을 가능하게 해준다[1]. 내용기반 기능을 제공하기 위해 MPEG-4에서는 우선적으로 비디오 시퀀스를 의미있는 객체 또는 VOP(Video Object Plane)로 분할하여야 한다. 즉 MPEG-4 비디오 부호화는 VOP 단위로 처리된다. VOP는 비디오 영상에서 객체에 대한 하나의 장면을 나타내며 각각의 VOP는 움직임(motion), 텍스처(texture), 모양정보(shape information)에 따라 분리되어 비트스트림으로 부호화 된다. 즉 VOP의 개념이 내용기반 비디오 부호화의 기능을 위해서는 배경으로부터 정확한 움직임 객체를 분리하는 비디오 객체 분할방법이 매우 중요한 문제이다. 그러나 비디오 시퀀스에서 움직임 객체를 분할하는 문제는 매우 난해하다. 우선 움직임 객체를 추출하기 위한 고려사항으로는 객체와 배경

이 단순히 고정되어 있는 경우, 배경은 고정되어 있고 객체가 움직이는 경우, 배경에는 움직임이 있고 객체가 고정되어 있는 경우, 가장 일반적인 영상으로 배경과 객체가 움직이는 3가지 경우를 고려하여야 한다. 또한 움직임은 객체 추출의 핵심이므로 영상에서 움직이는 객체를 추출하기 위해서는 먼저 영상의 움직임 특성, 명암도, 물체, 구조, 크기, 형태의 변화를 발견하고 움직임의 속도에 따라 움직임 특성을 결정하는 많은 객체 분할 방법이 사용된다[2][3][4].

현재 존재하는 객체 분할 방법은 다음과 같은 방법을 포함한다. 컬러 동질성을 이용한 지역 기반 접근 방법[5][6], 움직임 표준을 이용한 객체기반 접근 방법[7][8][9], 객체 추적을 이용한 접근 방법[5][10][11] 이외 많은 방법이 있다. 영상에서 의미있는 객체는 많은 움직임과 컬러 사이에 공통적인 지역을 포함한다. 그래서 단일 컬러나 기본 움직임은 객체 분할에 있어서 만족한 결과를 이끌 수 없다. 이러한 컬러나 움직임 같은 의미있는 객체의 다중 특성은 정확한 객체 분할을 위하여 최종적으로 통합되어야 한다. 즉 정확한 객체 추출에 있어 시간-공간 통합은 중요한 문제로 대두되고 있다. 많은 연구 작업은 인터랙티브(interactive)한 객체 추출 작업과 정확한 객체 경계를 찾는 방법에 관심이 모여져 있다[10][11][12]. 분할 방법중 반자동 객체 추출 방법은 적어도 첫 번째 프레임의 분할을 위하여 사용자와의 상호작용이 요구된다. 이러한 방법은 융통성 있고 상대적으로 정확하다. 그러나 사용자의 상호작용은 사용자에게 부담이 될 수 있다. 그리고 이것은 또한 실시간 처리 시스템이나 컨퍼런스 시스템에 대해서는 부적절하다. 또 다른 종류의 방법은 자동분할 방법이다. 대부분 이러한 형태의 분할 결과는 거칠고 조잡한 객체를 나타내므로 일반적으로 사용하기에 적당하지 않다. 비디오 시퀀스의 모든 종류에 대하여 일반적인 자동 분할 알고리즘의 개발은 의미있는 분할의 어려움 때문에 현재 최적화되어 있지 않다. 그래서 우리는 움직임 객체 추출을 위한 영상 범위를 줄여 정확한 객체 경계를 찾는 효율적인 자동, 반자동 분할 알고리즘에 초점을 두어 개발하였다. 움직임 분할의 결과는 움직임 측정의 정확성에 의해 영향을 미친다. 이 움직임 측정은 프레임안의 노이즈 때문에 정확한 결과를 나타내지 못하는 단점이 있다. 또한 다른 공간적 분할중 컬러를 이용하는 분할은 에지를 이용하는 방법보다 더욱 정확한 경계를 찾을 수 있지만 일반적으로 분할된 지역이 많이 발생하는 단점이 있다. 그렇기 때문에 다중 분할지역을 하나로 모으는 지역 병합 알고리즘이 추가되어진다. 그러므로 일반적으로 만족할

만한 결과를 위해서는 시간-공간 분할의 효율적인 조합이 필요하다. 본 논문에서는 시간-공간 객체의 효율적인 조합을 위하여 시간적인 객체 에지를 추출하고 여기에 수리형태학 녹입 연산을 적용하였다. 또한 다음 프레임의 공간적 에지를 검출하여 검출된 에지에 수리형태학 녹입 연산을 적용하여 두 프레임 사이의 에지를 비교하여 정확한 객체 경계를 나타내었다. 즉 정확하게 비디오 객체를 추출하기 위하여 우리는 움직임 분할과 공간적인 분할의 결과에 수리형태학 불림 녹입 연산을 직접적으로 조합하였다.

본 논문에서는 일반적인 MPEG-4 실험 비디오 시퀀스를 사용하여 실험하였다. 개발된 알고리즘은 전체 실험 시퀀스를 통하여 정확한 객체의 경계를 보여주었다. 본 논문의 구조는 다음과 같다. 2장에서는 관련 연구분야를 소개하며 3장에서는 개발된 알고리즘을 소개하고, 4장에서는 시스템 구현과 실험 결과를, 마지막으로 5장에서는 결론을 맺었다.

## 2. 관련 연구

### 2.1 현재의 VOP 추출 방법

최근 VOP추출 방법에 대하여 많은 제안이 발표되었다[2][3][4][13][14]. MPEG-4 표준 버전(1)에서 권장하는 객체 분할 구조는 시간적 분할, 공간적 분할, 시간-공간적 분할 결합의 세 부분으로 나누어져 있다. 시간적 분할이라는 것은 영상안에 있는 객체의 움직임 위치를 추출하는 작업이고 공간적 분할은 주어진 기준에 따라서 정확한 객체의 경계와 영상안에 있는 의미있는 객체의 지역을 추출하는 작업이다. 이 두 가지의 작업을 통합하여 마지막으로 영상으로부터 각 객체를 추출하는 것이다. MPEG-4 비디오 부분에 적용된 시간-공간 객체 분할을 이용한 VOP 추출 과정은 <그림 1>과 같다.

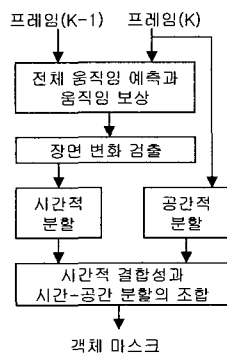


그림 1 시간-공간 객체 분할 단계

<그림 1>에서와 같이 움직임 객체를 추출하는 단계에서 가장 중요한 단계는 시간-공간의 조합이다. 적절한 방법을 통하여 두 객체를 조합에 따라서 정확한 움직임 객체를 구별 할 수 있다.

참고문헌[2]는 시간-공간 정보를 기반으로 객체 추출 방법을 소개한다. 제안된 알고리즘은 비디오 시퀀스에서 움직이는 객체의 위치를 측정하기 위하여, 시간적 방향으로 연속된 두 이미지 차이에 대한 조사와 두 연속적인 이미지로부터 이미지 편차를 측정 비교함으로써 시작된다. 즉 시간적 분할 방법은 명암도 차이와 편차 비교를 기반으로 통계적 가정 테스트를 포함하여 객체를 추출하고 공간적 분할은 watershed 검출 기반 지역 성장 알고리즘(region growing algorithm)과 수리 형태학 필터(morphological filter)를 사용한다. 논문에서는 움직임 객체의 위치측정을 위해 전처리 과정을 수행하는데 전체적 움직임 추정과 보상은 장면 변환 검출 알고리즘을 사용한다. 움직임 추정기법은 6 파라미터 affine 움직임 모델을 사용한다. 장면 변환 검출은 장면 변환을 추출하기 위해 MAD(Mean Absolute Difference)와 SDMAD(Sign Difference MAD) 방법을 사용한다. 공간적 분할단계는 이미지 표준화, 기울기 근사치, watershed 검출, 지역 병합으로 이루어진다. 이미지 표준화는 수리 형태학 필터의 기능을 사용한다. 이것은 주어진 영상 요소에 기하학적인 특징은 보존하면서 불필요한 요소들을 제거함으로써 영상을 단순화시킴과 동시에 유용한 기하학 정보를 추출하는 뛰어난 기능을 가지고 있기 때문이다. 논문에서는 비디오 객체 분할을 위하여 시간적 분할단계에서 affine 움직임 모델을 사용하였다. affine 움직임 모델은 객체의 움직임 보다는 영상에서 카메라의 움직임 벡터를 측정하기 위해 많이 사용되는 알고리즘으로 움직임 벡터를 추출하기 위해서는 각각의 픽셀마다 움직임 거리를 계산하여야 하는 계산의 복잡도가 추가되어진다. 또한 공간적 분할은 기울기를 기반한 watershed 알고리즘을 사용하였다. Watershed 알고리즘은 분할 중복의 문제점이 발생하므로 추가적으로 지역 병합 연산을 필요로 하는 단점이 있다.

참고문헌[3]은 움직임 정보를 기반으로 배경(background)으로부터 실제 객체(foreground)를 분리한다. 전체적인 알고리즘은 움직임 객체 추출, 모델 초기화, 모델 업데이트, VOP 추출의 4단계로 구성되어있다. 움직임 추출을 위한 방법으로는 수리 형태학 움직임 필터(mathematical morphology motion filter)와 변환 탐지 마스크(change detection mask)기법을 사용한다[5]. 객체 추적을 위한 매칭 표준은 Hausdorff distance 방

법을 사용한다. 이것은 계산적 효율과 모양 변화와 잡음에 강력한 특징을 가진다. 움직임 추출을 위해 두 가지 방법을 사용한다. 수리 형태학 움직임 필터는 이미지의 다른 부분은 정확하게 보존하면서 전체 움직임에 따르지 않는 부분을 제거하는 기능을 가지고 있다. 이 움직임 추출기법은 작은 움직임이나 정확한 움직임에 사용되고, 또 다른 방법으로 변화 탐지 마스크는 보다 빠른 움직임에 적용한다. 변화 탐지 마스크의 사용은 배경이 움직일 경우 문제가 발생한다. 그 이유는 배경의 움직임에 대하여 정확한 표시가 불가능하기 때문이다. 따라서 변환 탐지 마스크는 고정된 배경을 기반으로 제한되어 사용된다. 모델 초기화는 처음 객체의 위치와 모델의 존재를 발견하기 위하여 사용되어 진다. 모델 업데이트는 다음 프레임의 존재에 따라서 수행된다. VOP 추출은 객체 경계를 결정하기 위하여 2 단계로 이루어진다. 처음 단계는 filling-in technique을 사용하고 차후 boundary postprocessor를 사용하여 VOP를 추출한다. 변환 탐지 마스크는 객체 내부의 움직임이 존재할 경우 움직임을 탐지 할 수 없는 단점이 있다. 우리가 제안한 논문은 변환 탐지 마스크를 사용하지 않고 이러한 단점을 없애기 위하여 수리형태학 연산을 적용한 것이다. 수리형태학 연산을 적용함으로써 움직임 객체의 경계를 뚜렷하게 찾을 수 있었다.

참고문헌[13]에서는 joint marker 추출, 경계 결정, 움직임 기반 지역 융합 방법을 사용하였다. Joint marker 추출은 휘도와 움직임 사이에서 동일한 지역의 존재를 나타낸다. 다음으로 시간-공간적 경계는 watershed 검출을 사용하여 결정되어진다. 마지막 단계로 움직임 기반 지역 병합 알고리즘을 사용하여 중복된 지역을 제거함으로써 객체를 추출한다. 제안된 방법도 공간적 객체 분할을 위하여 watershed 검출 방법을 사용하였다. Watershed 검출 방법은 영상의 휘도에 따라 지역을 검출하지만 이전에 언급하였듯이 지역 병합 알고리즘이 추가되는 단점이 있다.

참고문헌[14]에서는 컬러 정보와 움직임 정보, 움직임 추적을 통한 객체 분할을 소개한다. 컬러 영상분할은  $L \times u \times v$  칼라 공간에서 수행되며, 컬러 영상 분할을 위한 알고리즘은 1975년 Fukunaga, Hosteler에 의해 제안되었던 non-parametric 기율기 기반 컬러 클러스터링 알고리즘, 즉 mean shift algorithm을 사용하였다. 이것은 컬러 유사성에 따른 동일한 컬러 지역의 분할과 경계를 비교함으로써 객체를 추출한다. 전체적인 mean shift algorithm의 수행과정은 다음과 같다.

- 1) 탐색 윈도우의 반지름  $r$ 를 선택
- 2) 윈도우의 초기 위치 선택

3) mean shift vector 와 이동된 탐색

4) 중심점까지 반복하여 계산

즉 mean shift algorithm은 전체 영상을 유사한 색으로 모으는 작업이다. 칼라 성분의 명암도 분포는 확률 밀도 함수에 의해 나타내질 수 있다. 즉 확률 밀도 함수에 의해 높은 칼라 밀집지역을 찾을 수 있다. 움직임 검출은 지역 기반 유사 움직임 모델과 광학적 흐름(optical flow)를 사용하였다. 이후 유사 움직임 모델을 통하여 다음 프레임의 움직임을 추적하는 방법을 사용하였다. 위 알고리즘의 가장 큰 단점은 칼라 클러스터링 알고리즘에서 영상에 너무 많은 빛이 존재하면 많은 분할 지역으로 나누어지므로 병합하는 과정이 추가적으로 필요하다는 점이다. 또한 광학적 흐름 알고리즘은 두 영상사이에 휘도 차이가 아주 작을 경우 객체의 움직임을 찾기 힘든 단점이 있다.

### 3. 제안된 비디오 객체 분할 알고리즘

#### 3.1 전체 구조

제안된 비디오 분할 알고리즘의 블록 다이어그램은 <그림 2>에서 보여준다. 본 논문에서 우리는 시간-공간영역의 복잡도를 줄이고 빠르고 적용성이 강한 자동, 반자동 분할 알고리즘에 초점을 맞추었다. 대부분 존재하는 영상 분할 기법은 객체와 배경을 분리하기 위하여 비디오 시퀀스안에 시간-공간 상관 정보와 전처리, 후처리 등 추가적인 알고리즘을 적용하여 객체의 경계를 찾는다. 시간적 분할을 위한 알고리즘에는 크게 변환 검출 마스크(change detection mask), 유사 움직임 검출(affine motion detection), 광학적 흐름 알고리즘(optical flow algorithm)[15], 전 탐색 블록 매칭 알고리즘(full search block matching algorithm)등이 있다. 변환 검출 마스크는 쉽게 구현되고 빠르게 움직임을 찾을 수 있는 장점에 비하여 아주 미세한 움직임에 대해서는 정확한 측정이 불가능한 단점이 있다. 본 논문에서는 이러한 단점을 보완하기 위하여 수리형태학 녹임 연산을 적용하였다. 또한 유사 움직임 검출 방법은 각 파라미터의 설정(6 파라미터, 8 파라미터, 12 파라미터)에 따라 전체적인 카메라 움직임을 측정하는 것으로 미세한 움직임을 찾는데 유리하지만 다른 움직임 검출 알고리즘의 도움 없이는 정확한 움직임 객체를 찾을 수 없는 단점과 각 픽셀에 대한 벡터를 찾기 때문에 많은 시간적 낭비를 갖는다. 광학적 흐름 알고리즘은 카메라 움직임이 아닌 비디오 영상안의 객체에 대한 움직임을 찾을 수 있는 알고리즘이며 보통 영상의 명암도와 기율기에 의하여 움직임을 찾는다. 그러나 전체 영상의 명암도에 따

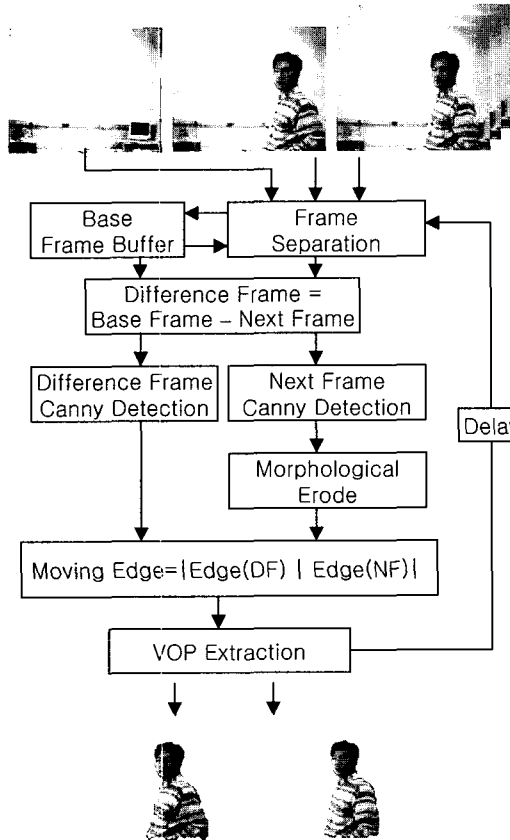


그림 2 전체 구조와 흐름도

라 픽셀의 기울기와 명암도 차이를 구하는 만큼 명암도 차이가 심하면 정확한 움직임을 구할 수 없고, 계산에 따른 시간적 낭비가 심하다. 이에 광학적 흐름 알고리즘은 다른 공간적 알고리즘의 추가적인 작업 없이는 정확한 움직임 객체를 찾기 힘들다. 공간적 분할은 배경으로부터 칼라 또는 그레이 명암도를 이용하여 객체의 경계를 결정한다. 공간적 분할을 위한 알고리즘에는 크게 영상 자체의 에지를 찾는 에지 추출 알고리즘, 칼라 클러스터링 알고리즘(color clustering algorithm), 지역 병합 알고리즘(region merging algorithm) 등이 있다. 본 논문에서는 공간적 분할에 Canny 에지 알고리즘을 사용하였다. 이 알고리즘은 노이즈에 강하며 컨벌루션 가우시안 함수에 따라 각 영상에 대한 미세한 에지를 찾을 수 있는 장점이 있다. 칼라 클러스터링 알고리즘은 영상의 칼라 유사성을 이용하여 비슷한 칼라를 흡수하여 영상의 분할 지역을 찾는 것으로 영상의 색이 단순한 경우는 정확한 영상을 분리하나 복잡한 영상은 너무

많은 분할 지역을 보이는 단점이 있다. 마찬가지로 지역 병합 알고리즘의 경우는 너무 많은 지역으로 나누어지는 현상이 발생하므로 차후 각 지역에 대한 병합을 수행하는 작업이 필요하다.

본 논문에서는 추가적인 알고리즘 필요 없이 시간적, 공간적으로 분할된 객체의 에지에 수리형태학 알고리즘을 적용하여 정확한 객체의 경계를 찾았다. 본 알고리즘의 전체 구조는 매우 간단하며 쉽게 객체의 정확한 경계를 찾는다. 알고리즘 구성은 5 단계로 이루어져 있다. 처음 단계는 분할을 위한 기준 프레임의 설정이다. 두 번째 단계는 현 프레임과 다음 프레임 사이의 차이값을 찾는다. 즉 두 프레임 사이의 차이값을 검출하여 두 객체사이의 움직임 위치를 검출하는 것이다. 세 번째 단계는 객체 검출이 이루어지는 프레임의 에지 연산과 차이영상에 에지 연산을 수행한다. 네 번째 단계는 차이영상과 검출 영상에 수리형태학 녹임 연산을 적용하여 에지 크기를 조절하여 두 프레임 사이의 움직임이 있는 에지를 정확하게 찾는다. 다섯 번째 단계는 추출된 움직임 에지 영상을 통하여 VOP를 추출한다. 위와 같은 알고리즘을 사용하면 우선 알고리즘의 복잡도를 줄일 수 있으며, 복잡한 계산을 필요하지 않고 간단하게 정확한 객체 경계를 찾을 수 있다. 또한 거친 경계면이나 다중 분할(over-segmentation)을 막을 수 있으며 영상에서 움직임이 있는 곳만을 빠르고 정확하게 추출 할 수 있는 장점이 있다.

VOP 분할 알고리즘의 가장 중요한 점은 어떻게 영상에서 의미있는 객체를 검출할 수 있는지에 대한 방법의 문제이다. 즉 움직임은 물리적 객체를 식별하는 가장 널리 사용되어지는 핵심 요소이다. 그러므로 VOP 분할을 위한 또 다른 중요한 문제는 정확한 움직임 측정을 수행하는 것이다. 우리의 분할 알고리즘은 움직임 정보를 기반으로 배경으로부터 객체를 분리하여 수행되어진다. 각 단계의 자세한 내용은 다음과 같다.

처음 단계는 프레임 분리와 기준 프레임의 설정이다. 즉 이것은 기준 프레임을 설정하는 과정이다. 왜냐하면 영상에서는 배경과 객체의 조합에 따라 최소한 3 가지 타입의 영상 형태를 고려해야 하기 때문이다. MPEG-4 분할 과정에서 가장 널리 사용하는 영상 구성은 첫째 감시 카메라에서와 같이 배경은 고정되어 있고 전경에 움직임이 있는 영상이다. 이러한 영상은 본 논문에서 제안된 방법으로 아주 정확한 움직임 객체를 얻을 수 있다. 또한 배경이 움직이는 경우와 전경이 고정되어 있는 경우는 전자에서 설명한 과정의 반대 형태로 분할이 가능하다. 또한 전경과 배경이 움직이는 형태의 영상은 아직까지 많은 연구가 진행되고 있으며 차후 다른 알고리즘

을 사용하여 분할을 수행할 것이다. 즉 위에서 설명하였듯이 첫 번째 과정은 움직임 검출에서 가장 중요한 단계로 움직임의 기준이 되는 프레임을 분리하여 기준 영상을 프레임 버퍼에 저장하는 과정이다. 일반적으로 영상 시퀀스에서 첫 번째 영상을 기준 프레임으로 설정한다.

두 번째 단계는 영상 시퀀스의 움직임을 검출하는 단계로 프레임 차이를(frame difference) 이용하여 움직임을 검출한다. 본 논문에서 차이값을 측정하는 방법으로 는 다음과 같은 [식 1]을 사용하였다.

[식 1]

$$DFn = |F(\text{baseframe}) - Fn(\text{nframe})|$$

baseframe : 기준 프레임

nframe : 다음 프레임

차이 영상  $DFn$ (Difference Frame)은 원 영상인 기본 프레임(base frame)에서 다음 프레임  $Fn$ (next frame)의 차이값을 계산하여 구한다. 영상 차이값을 구하는 소스 코드는 <표 1>과 같다.

표 1 영상 차이값 추출 코드

```

for(register int x=0; x<width; x++)
for(register int y=0; y<height; y++)
{
    int v;
    int index = (x*height)+y;
    v=(baseframe[index] - newframe[0][index]);
    Df[index] = abs(v);
}
    
```

<표 1>의 코드를 적용하여 차이값을 구하고 차이 영상으로부터 움직임 변화를 측정할 수 있다. 즉 처음 영상을 기본 프레임으로 버퍼에 저장하고 다음 영상과의 차이를 구한다. [식 1]을 적용한 결과를 <그림 3>이 보여준다.

세 번째 단계는 차이영상에 대한 에지와 다음 프레임에 대한 에지를 추출하는 과정으로 각 영상에 대하여 Canny 에지 검출 알고리즘을 사용하였다. 다음 [식 2]는 Canny 에지 검출식이다.

[식 2]

$$CannyEdge[fn] = \mathcal{O}(\nabla G * |fn|)$$

Canny 에지 검출 방법은 기울기 연산  $\nabla$ 에 가우시안 함수  $G*fn$ 을 수행하는 것이다. Canny 알고리즘은 3.2.1절에 소개한다.

네 번째 단계는 움직임 에지(moving edge) 추출 단계이다. 움직임 에지는 기준 프레임 에지와 다음 프레임 에지, 즉 두 영상에서 움직임이 있는 객체의 에지를 말한다.



[차이 영상 1]



[차이 영상 2]

그림 3 DF(Difference Frame) 영상

움직임 에지 추출 방법은 세 번째 단계에서 추출된 차이 영상의 에지와 다음 프레임 영상의 에지를 비트단위 논리곱 알고리즘을 사용하여 추출한다. 다음 [식 3]은 움직임 에지 추출 방법에 대한 수식이다.

[식 3]

$$ME(MovingEdge) = (\nabla G * |F(\text{baseFrame})|) \wedge ((\text{morphology}) \nabla G * |Fn|)$$

[식 3]은 기준 프레임의 에지 영상과 다음 프레임의 에지 영상에 비트단위 논리곱을 수행하는 것을 보여준다. 여기서 움직임 객체의 경계 에지를 정확하게 찾기 위하여 <그림 9>에서와 같이 다음 프레임에 수리 형태 불림 연산을 적용하였다. <그림 4>는 검출된 움직임 에지를 보여준다.

만약 움직임 에지 검출 단계에서 수리형태학 연산이 추가되지 않았다면 추출된 영상은 많은 노이즈로 인해 움직임 객체와 배경의 구분이 어렵다.

마지막 단계는 추출된 움직임 에지의 VOP(Video Object Plane)를 생성하는 단계이다. 추출된 에지를 수평과 수직으로 각각 에지 영역안에 픽셀 값을 채우고 두 수평과 수직에 대하여 논리곱 연산을 수행하였다. VOP

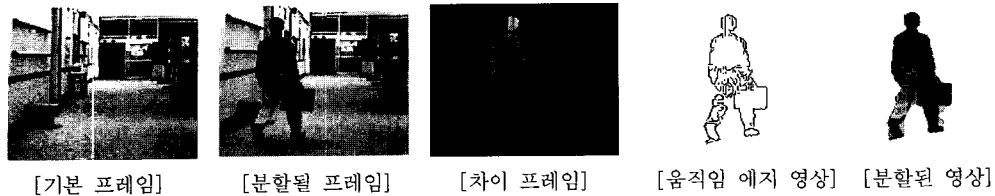


그림 4 움직임 에지

추출 단계에서도 정확한 객체를 얻기 위하여 움직임 에지를 찾는 방법으로 추출된 객체 에지에 다시 수리형태학 연산을 추가하였다. 즉 기본 프레임 영상과 다음 프레임 영상에 Canny 에지를 적용시키고 수리형태학 연산을 수행한 다음 두 프레임사이의 논리곱 연산을 수행하여 변화된 움직임 에지를 측정하였다. <그림 5>는 전체 알고리즘 수행 후의 VOP 추출 결과를 보여준다.

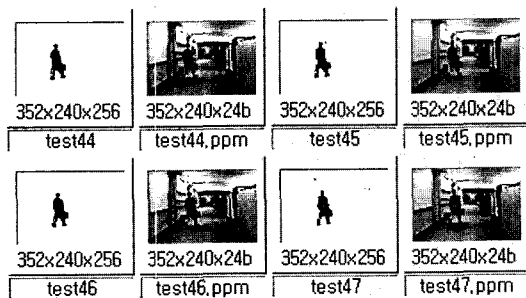


그림 5 VOP 추출 결과

### 3.2 공간적 분할

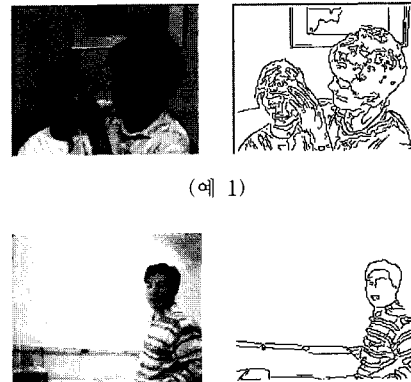
#### 3.2.1 Canny 에지 검출

Canny 에지는 움직임 객체의 윤곽을 얻기 위하여 수행된다. Canny 에지 검출방법의 특징은 다른 에지검출 방법보다 노이즈(noise)에 강하며 회선(convolution)하는 가우시안 함수의 조작을 통해 필요에 따라 적절한 이미지를 얻을 수 있다. 본 논문에서 Canny 에지 검출은 다음과 같은 단계로 수행하였다[16].

- 1) 영상 입력.
- 2) 가우시안 필터(Gaussian filter)를 이용하여 이미지를 평활화(smoothing).
- 3) 편미분에 적용한 유한 차이 근사값을 이용하여 기울기의 크기와 방향을 계산.
- 4) 기울기 크기에 nonmaxima suppression을 적용.
- 5) 검출과 에지 연결을 위하여 이중 임계값 알고리즘을 적용.

#### 6) 에지 검출된 영상 출력.

즉, Canny 에지 검출은 신호대 잡음 비율과 위치를 최적화하는 연산과 흡사하다. 본 논문에서 적용한 Canny 에지 검출 알고리즘의 예는 <그림 6>의 예 1 그리고 예 2와 같다. 다음 그림에서 컨벌루션 가우시안 함수의 값은 ( $\sigma=1.2$ )을 사용하였다.



(예 1)

(예 2)

그림 6 Canny 에지 검출

#### 3.2.2 수리 형태학(Mathematical Morphology)

수리 형태론 영상처리 분야에서 영상의 구성 추출 및 물체의 외곽선이나 경계, 골격 등의 특징을 표현하는 기법이다. 또한 전처리나 후처리에서 영상을 가늘게 하거나 가지치기효과를 얻기 위하여 수리 형태 필터를 이용한다. 이러한 수리 형태는 집합이론을 이용하며 가장 기본적인 연산은 불림(dilation)과 녹임(erosion) 연산으로 본 논문에서는 에지 객체의 비교를 위하여 적용된 알고리즘의 핵심 부분이다.

수리 형태 불림과 녹임 연산은 정확한 움직임 에지와 VOP 추출을 얻기 위해 적용하였다. 다음 <그림 7>은 수리 형태 불림 연산을 적용한 움직임 에지 추출 과정을 보여준다.

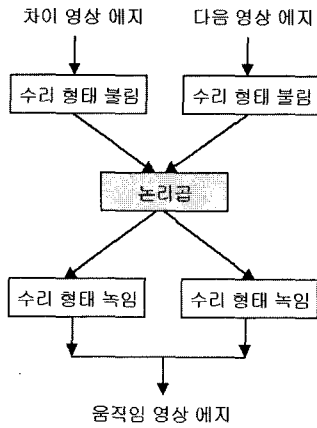


그림 7 움직임 에지 추출 다이어그램

<그림 8>의 좌측 영상은 다음 영상에 Canny에지 검출을 적용하여 에지를 추출한 결과이며, 우측 영상은 추출된 에지 영상에 수리 형태 불림 연산을 적용한 결과 영상이다.

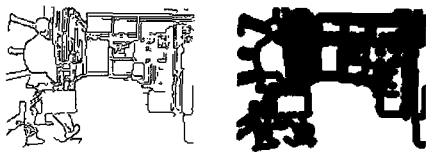


그림 8 다음 영상과 수리형태 불림 적용 영상

<그림 9>에서 좌측 영상은 [식 3]에 의하여 기준 프레임에서 다음 프레임을 뺀 차이 영상이고 우측 그림은 차이 영상에 대하여 추출된 에지를 보여준다.



그림 9 차이 영상과 에지 영상

<그림 10>은 <그림 8>의 우측 영상에 <그림 9>의 우측 영상을 비트단위 논리곱에 의해 추출된 움직임 에지를 보여준다. 또한 그 반대 형태로 <그림 9>의 우측 영상에 수리형태 연산을 적용하고 <그림 8>의 좌측 영상에 알고리즘을 적용하여 움직임 에지를 찾을 수 있다. 만약 <그림 10>의 결과에서 수리 형태 불림 연산을 적

용하지 않는다면 움직임 에지의 영상은 많은 노이즈로 인하여 식별하기가 어렵게 된다. 따라서 수리 형태 불림 연산을 적용함으로써 추가적인 알고리즘이 제외되고 복잡한 계산을 줄이며 효율적으로 객체 경계와 움직임 에지를 정확하게 추출할 수 있다.



그림 10 비트단위 논리곱 연산으로 추출된 움직임 에지 영상

### 3.3 시간적 분할

#### 3.3.1 움직임 검출 및 측정

시간적 분할을 위한 알고리즘은 두 영상의 차이값을 이용하는 변환 검출 마스크 알고리즘을 사용하였다. 움직임은 화면 사이의 상대적인 위치 변화로부터 발생하는 것으로, 관심 대상 물체를 배경으로부터 추출하는데 이용한다. 즉 변환 검출은 시간  $t_i$ 와  $t_j$ 에서 두 영상 프레임  $f(x,y,t_i)$ 와  $f(x,y,t_j)$ 사이의 변화를 검출하는 것으로, 가장 간단한 방법은 두 영상의 모든 대응하는 화소를 비교하여 차이 영상(difference image)을 생성하는 것이다. 만약 정적인 성분만을 가진 기준 영상이 존재한다면 기준영상과, 같은 환경에서 움직이는 물체를 포함한 다음 영상을 비교하면 두 영상의 차이에서, 정적 성분 제거되고 비정적 성분에 해당하는 영(0)이 아닌 요소만이 남는다. 일반적으로 차이 영상을 구하는 공식은 [식 4]와 같다.

[식 4]

$$d_{ij}(x, y) = 1 \quad \text{if } (|f(x, y, t_i) - f(x, y, t_j)| > \theta)$$

$\theta$  : 임계값

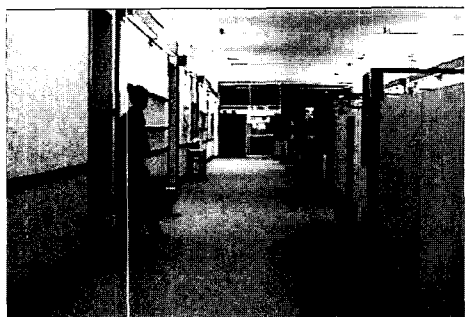
$$d_{ij}(x, y) = 0 \quad \text{otherwise}$$

$d_{ij}(x, y)$ 는 공간 좌표  $(x, y)$ 에서 두 영상간의 명암도 차이가 상당히 다를 경우에만 임계값에 의해 1로 결정하고 다른 경우에는 0 값으로 결정 된다. 일반적인 변환 검출에 따른 결과는 <그림 11>의 c와 같다. <그림 11>의 a와 <그림 11>의 b는 첫 번째와 두 번째 프레임이다.

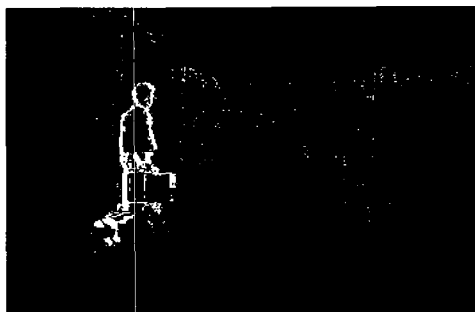




(a) 프레임 1



(b) 프레임 2



(c) 움직임 변환 검출 영상  
그림 11 움직임 검출 영상

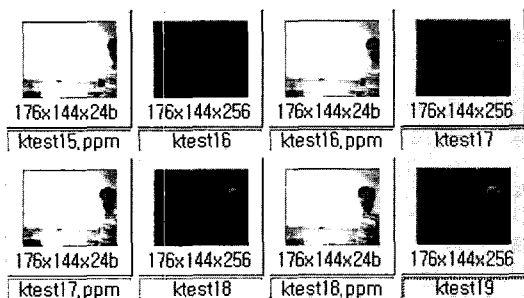


그림 12 움직임 검출 결과

즉 동적인 객체를 추출하기 위해서는 먼저 영상의 움직임 특성, 명암도, 불체, 구조, 크기, 형태의 변화와 움직임의 속도에 따라 움직임 특성을 결정하여야 한다. 본 논문에서 적용된 움직임 검출은 이전에 설명한 [식 1]에 의해 두 프레임 사이의 차이 값 추출을 계산하여 수행된다. [식 1]에 의해 추출된 움직임 검출은 <그림 12>와 같다.

### 3.4 움직임 객체 VOP 추출

VOP(Video Object Plane) 추출은 이전 단계에서 추출된 움직임 에지 영상에 <그림 13>의 단계를 적용한다.

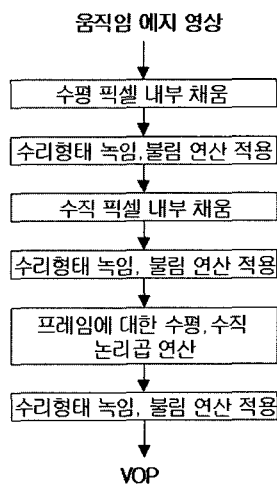


그림 13 VOP 추출 블록 다이어그램

원 영상의 에지와 차이 영상의 에지 사이에서 추출된 움직임 에지에 대하여 VOP 추출 과정을 수행한다. VOP 추출 과정은 수평과 수직 에지 사이의 내부에 픽셀을 채우는 과정으로 시작한다. 다음에 내부가 채워진 두 영상 사이에 논리곱 연산을 수행한다. 마지막으로 VOP 추출 과정에서 정확한 VOP 추출을 위하여 수리형태 녹입 연산과 불림 연산을 적용한다. <그림 14>는 추출된 움직임 에지 영상을 보여준다.



그림 14 움직임 에지 영상

다음 단계로 <그림 15>는 수평 에지 픽셀을 채운 형태의 영상을 보여준다.

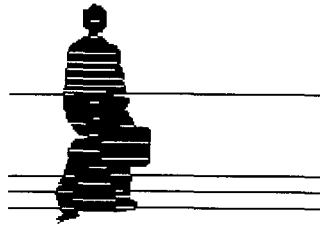


그림 15 수평 에지 픽셀 채운 영상

수평 에지 픽셀을 채운 영상은 <그림 15>처럼 영상 내부에 픽셀이 채워지지 않는 부분이 발생하므로 수리 형태 불림 연산과 녹임 연산을 통하여 <그림 16>처럼 채워지지 않은 내부 픽셀을 병합한다.

<그림 17>은 수직 에지 픽셀을 채운 영상을 보여준다.

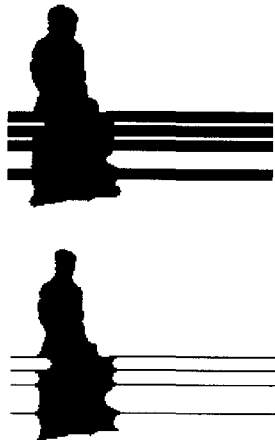


그림 16 수리 형태 불림(위), 녹임 연산(아래)



그림 17 수직 에지 픽셀 채운 영상

<그림 17>의 수직 에지 픽셀 채운 영상도 수평 에지 픽셀 채운 영상과 같은 방법으로 수리 형태 불림과 녹임 연산을 수행한 다음 마지막으로 수평과 수직에지 영상에 대하여 비트단위 논리곱 연산을 수행한다. 수행된 영상에 수리 형태 불림과 녹임 연산을 적용한 결과를 <그림 18>에서 보여준다.



그림 18 수평과 수직 에지에 대한 수리 형태 녹임, 불림 연산 후 비트단위 논리곱을 수행한 결과

마지막 단계로 추출된 VOP 영상에 원 영상과 추출 영상 사이의 객체를 매칭하여 VOP 영상을 추출한다.



그림 19 VOP 추출 영상

#### 4. 시스템 구현 및 결과

제안된 알고리즘은 “Miss America”와 “Test man”의 176\*144 사이즈의 QCIF와 CIF포맷 형태의 동영상을 MPEG-2 복호기를 이용하여 PPM 영상과 PGM 영상으로 프레임을 나누어 수행하였다. 본 논문에서 제안된 알고리즘은 분리된 PPM 프레임 영상을 입력받아 사용자의 선택에 의해 프레임 수를 10~15프레임 정도를 수행하였고 수행 결과는 펜티엄 500Mhz PC에서 프레임당 약 0.5~0.6초로 수행되었다. <그림 20>은 구현된 사용자 인터페이스를 보여준다.

구현된 시스템을 실험하기 위한 실험 영상은 다음 <그림 21>과 같다.

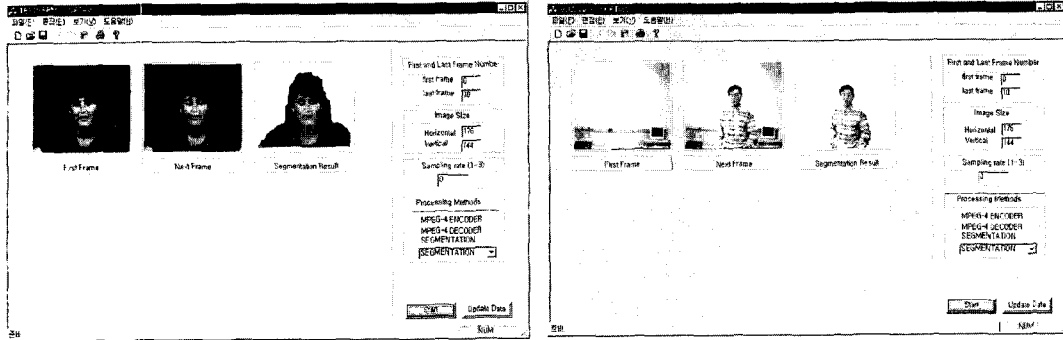


그림 20 사용자 인터페이스

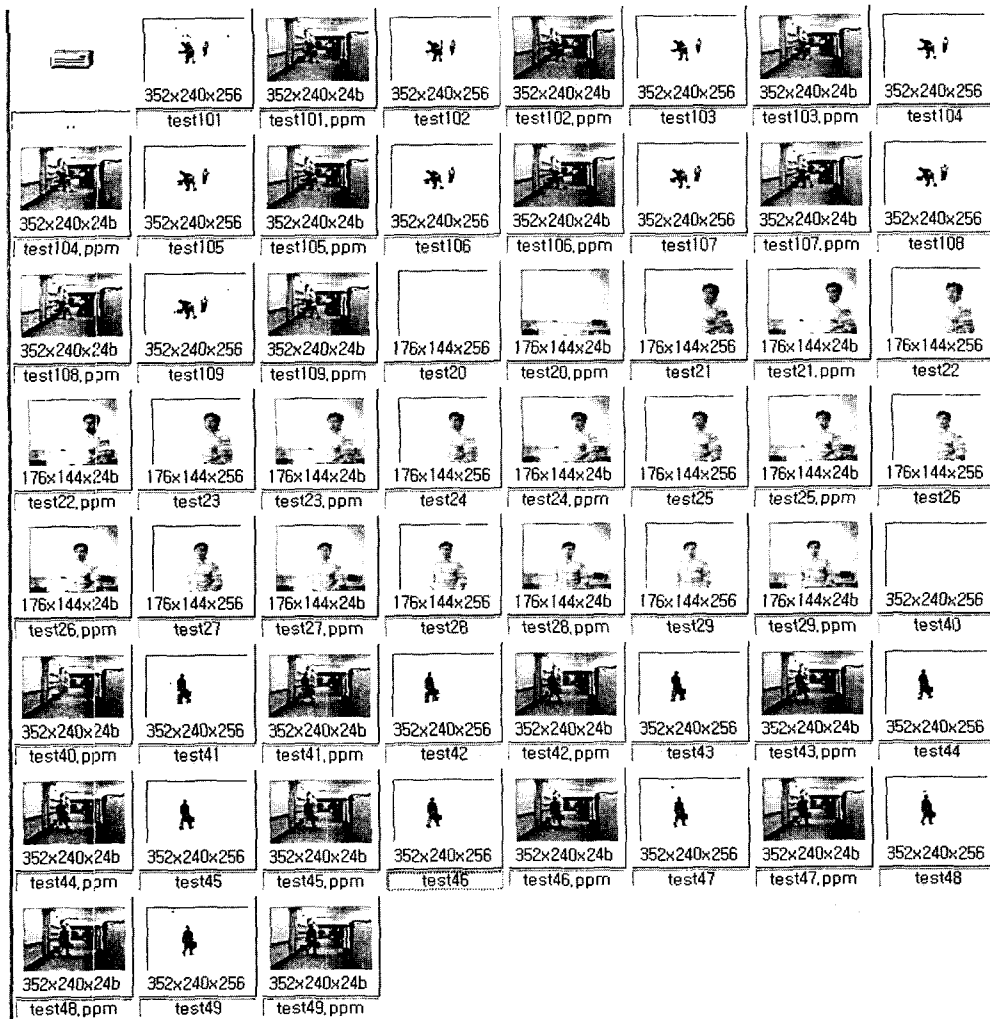


그림 21 실험 영상

5. 결 론

본 논문은 MPEG-4 객체기반 코딩을 위한 빠르고 잡음에 강력한 움직임 객체 추출 알고리즘을 소개하였다. 또한 제안된 객체 분할 기법은 프레임 차이를 이용한 알고리즘으로서 빠른 움직임이 있는 부분과 모양 변화에 강력하다. 특별히 제안된 알고리즘은 MPEG-4 부호기에 적당하며 자동적으로 움직이는 객체를 분할할 수 있다. 차후 움직이는 배경과 움직이는 객체에 적용할 수 있는 알고리즘이 제시되어야 할 것이다. 그리고 움직임 객체의 의미 해석과 정량적 방법의 개발이 필요하다.

참 고 문 헌

[1] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, pp. 19-31, Feb., 1998.

[2] M.C. Kim, J.G. Choi, and D.H. Kim et al., "A VOP Generation Tool : Automatic Segmentation of Moving Objects in Image Sequences based on Spatio-Temporal Information," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 8, Dec., 1999.

[3] T. Meier and K.N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 15, pp. 525-538, Sept., 1998.

[4] P. Bouthemy and E. Francois, "Motion segmentation and qualitative dynamic scene analysis from a image sequence," *Int. J. Computer Vision*, Vol. 10, No. 2, pp.157-182, 1993.

[5] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, No. 5, Sept., 1998.

[6] J. Scharcanski and A.N.Venetsanopoulos, "Edge detection of color image using directional operator," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, No. 2, April 1997.

[7] A.M. Tekalp, *Digital Video Processing*, Prentice Hall, Inc, 1996.

[8] M.M. Chang, A.M. Tekalp, and M.I. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Trans. on Image Processing*, Vol. 6, pp.1326-1333, Sept., 1997.

[9] J.Y.A. Wang and E.H. Adelson, "Representing moving images with layers," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 3, No. 5, Sept., 1994.

[10] C. Cuand M.-C. Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, No. 5, Sept., 1998.

[11] C. Cu and M.-C. Lee, "Tracking of Multiple Semantic Video Objects for Internet Applications," *SPIE, Visual Communications and Image Processing'99*, Vol. 3653.

[12] R. Castango, T. Ebrahimi and M. Kunt, "Video Segmentation based on Multiple Features for Interactive Multimedia Application," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, No. 5, Sept., 1998.

[13] J.G. Choi, M. Kim, M.H. Lee, and C. Ahn, "Automatic segmentation based on spatio-temporal information," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, pp. 525-538, Sept., 1998.

[14] Ju Cuo et al., "Fast and accurate moving object extraction technique for MPEG-4 object-based video coding," *SPIE*, Vol. 3653, pp. 1210-1221, January 1999.

[15] B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artifi. Intell.*, Vol. 17, pp. 185-203, 1981.

[16] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. PAMI-8, pp. 679-698, Nov., 1986.



김 준 기

1992년 3월 ~ 1998년 2월 호서대학교 전자계산학과 이학사 졸업. 1998년 3월 ~ 2000년 2월 호서대학교 컴퓨터공학부 대학원 석사 졸업. 2000년 3월 ~ 2001년 2월 호서대학교 컴퓨터공학부 대학원 박사 수료. 2002년 ~ 현재 호서대학교 컴퓨터공학부 시간 강사. 관심분야는 영상부호화, 영상분할, MPEG-4



이 호 석

1979년 3월 ~ 1983년 2월 서울대학교 전자계산기공학과 공학사 졸업. 1983년 3월 ~ 1985년 2월 서울대학교 컴퓨터공학과 대학원 석사 졸업. 1985년 3월 ~ 1989년 2월 한국전기통신공사 근무. 1989년 3월 ~ 1993년 8월 서울대학교 컴퓨터공학부 대학원 박사 졸업. 1994년 ~ 현재 호서대학교 컴퓨터공학부 근무. 관심분야는 영상부호화, 영상해석, 웨이블릿응용, 프로그래밍 언어