MPEG-2 비트열로부터 객체 기반 MPEG-4 응용을 위한 고속 정보 추출 알고리즘

정회원 양 종 호*, 원 치 선**

Fast information extraction algorithm for object-based MPEG-4 application from MPEG-2 bit-streamaper

Jong-Ho Yang* , Chee-Sun Won** *Regular Members*

요 약

본 논문에서는 MPEG-2 비트열로부터 객체 기반 MPEG-4로의 고속 변환을 위한 정보 추출 알고리즘을 소개한다. 객체 기반 MPEG-4로의 변환을 위한 정보로써 객체 영상과 형상 정보, 매크로블록 움직임 벡터, 헤더정보가 MPEG-2로부터 추출된다. 추출된 정보를 이용하면 객체 기반 MPEG-4로의 고속 변환이 가능하다. 가장 중요한 정보인 객체 영상 추출은 MPEG-2의 움직임 벡터와 워터쉐드 알고리즘을 이용하여 이루어진다. 사용자의 인지정보를 이용하여 프레임 내에서 객체를 추출하고, 추출된 객체로 연속된 프레임에서 객체를 추적하게 된다. 수행 중 객체의 빠른 움직임으로 만족스럽지 못한 결과를 내더라도, 사용자가 개입하여 다시 좋은 결과를 얻을 수 있도록 하였다. 객체 추적 과정은 크게 두 단계로 객체 추출 단계와 객체 추적 단계로 나누어져 있다. 객체 추출 단계는 블록분류와 워터쉐드 알고리즘으로 자동 분할된 영상에서 사용자가 직접 객체를 추출하는 단계이다. 사용자가 개입하는 단계이기 때문에, 번거로울 수 있으나 손쉽게 추출할 수 있도록 구현하였다. 객체 추적 단계는 연속된 프레임에서 객체를 추적하는 단계로, MPEG-2 움직임 벡터와 객체 모양 정보를 이용하여 고속으로 구해지고 워터쉐드 알고리즘으로 윤곽선 보정작업을 하였다. 실험 결과 MPEG-2 비트스트림으로부터 객체 기반 MPEG-4로의 고속 변환이 가능함을 알 수 있었다.

ABSTRACT

In this paper, a fast information extraction algorithm for object-based MPEG-4 application from MPEG-2 bit-stream is proposed. For object-based MPEG-4 conversion, we need to extract such information as object-image, shape-image, macro-block motion vector, and header information from MPEG-2 bit-stream. If we use the extracted information, fast conversion for object-based MPEG-4 is possible. The proposed object extraction algorithm has two important steps, namely the motion vector extraction from MPEG-2 bit-stream and the watershed algorithm. The algorithm extracts objects using user's assistance in the intra frame and tracks then in the following inter frames. If we have an unsatisfactory result for a fast moving object, the user can intervene to correct the segmentation. The proposed algorithm consist of two steps, which are intra frame object extracting processing and inter frame tracking processing. Object extracting process is the step in which user extracts a semantic object directly by using the block classification and watersheds. Object tracking process is the step of the following the object in the subsequent frames. It is based on the boundary fitting method using motion vector , object-mask, and modified watersheds. Experimental results show that the proposed method can achieve a fast conversion from the MPEG-2 bit-stream to the object-based MPEG-4 input.

* 동국대학교 전자공학과 영상처리연구실 (taruyang@dongguk.edu), ** 동국대학교 전자공학과 부교수 (cswon@dongguk.edu)
논문번호: 010232-0831, 접수일자: 2001년 8월 31일

Ⅰ. 서 론

영상 내의 의미 있는 객체의 추출을 위한 영상 분할과 객체의 동영상 내에서 추적은 대부분의 내용기반 디지털 비디오 어플리케이션을 위한 중요한 작업이다. 예로써, MPEG-4 비디오 인코더는 객체 기반 영상 압축에 기반하고 있으며, 전처리 과정으로써 객체 영상을 필요로 한다[1]. 영상이나 비디오 검색 어플리케이션을 위한 MPEG-7의 서술자중의 일부인, 모양과 움직임 서술자는 객체 분할을 필요로 한다[2]. 또한 응용 어플리케이션으로써 내용 기반 편집, 특수처리 등 어플리케이션에 영상 분할과 객체 추적 기술이 적용될 수 있다. 그러나 현재의 컴퓨터로는 완전 자동 의미적 객체 분할과 추적은 불가능하다. 대신에 사용자 개입을 통한, 즉 반자동에 의한 의미 있는 객체의 추출이 가능하다[3][4][5].
반자동에 의한 의미 있는 객체 추출/추적에 대한 연구는 비 압축 영역과 압축 영역에 대해서 이루어져 왔다. 대표적인 비 압축 영역 반자동 객체 추적 알고리즘은 인트라 프레임 화면 내 객체 정의 과정과 인터 프레임 화면간 추적 과정의 2단계로 이루어져 있다[4][5]. 최초 화면에서 반자동으로 객체를 정의하게 되고, 정의된 객체는 화면 간 과정을 통해 동영상 내에서 추적되고 경계가 보정된다. 비압축영역 객체 추적 알고리즘은 화면 간 과정에서 영상 간 객체의 움직임 변위를 계산하는 움직임 예측 과정을 필요로 한다. 사용된 움직임 예측 방법은 어파인 모션 모델이나, 투영 정보를 이용한 방법이다[4][5]. 비압축영역의 기존 반자동 객체 추적 알고리즘은 어느 정도 정확한 객체의 추적을 이루나, 추적 계산 시 많은 비용이 드는 단점이 있다. 특히 객체 추적 과정 중 움직임 변위 계산은 전체 연산 중 가장 많은 비용을 필요로 한다.

사용목적에 따라 압축 영역에서의 선택 객체 추적 방법이 연구되었다. 이와 같은 알고리즘은 압축 영역의 정보를 이용함으로써 비압축영역에서의 움직임 예측 과정에서 나타나는 계산 복잡도를 대폭 줄인다. 대표적인 알고리즘은 MPEG-2에서 매크로 블록 단위 객체 마스크와 움직임 벡터를 이용한 방법이다[6]. 그러나 이 방법은 객체 주위의 부정확한 영역에 하이퍼텍스트를 넣기 위한 특수 목적용으로 객체 기반 MPEG-4 비디오 부호화 등과 같이 정확한 객체의 추출이 필요한 여러 경우에는 쓸 수 없다. 더욱이 추적 단계에 있어 MPEG-2의 움직임 벡터만을 사용함으로써 객체의 움직임 변위가 크거나 변형이 클 시 추적을 못하는 단점이 있다.

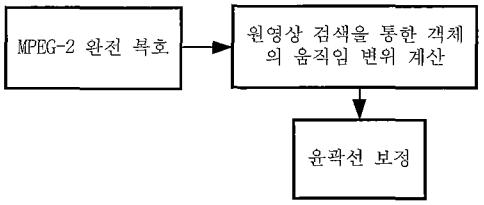


그림 1. 기존의 객체 추적 알고리즘을 이용한 MPEG-2 영상 내 객체 추적

비압축영역과 압축영역의 객체추적은 각각 장단점을 가지고 있지만 사용자가 정확한 객체의 추출을 필요로 하는 다양한 어플리케이션의 목적에 쓰고자 할 때는 기존의 비 압축영역에서 이루어지는 알고리즘이 보다 적합하다. 그러나 비압축영역 객체 추적을 현재 유통되어지는 MPEG-1,2의 동영상 압축 영상에 적용하여 객체 추적 응용 어플리케이션에 사용하기 위해서는 그림 1과 같이 많은 비용을 필요로 하는 압축 영상의 완전 복호화 과정과 화소 영역에서의 객체 추출 과정이 필요하다. 이는 많은 처리비용을 요구하며, 특히 여러 개의 객체를 추적할 경우 연산 비용은 더욱 상승한다.

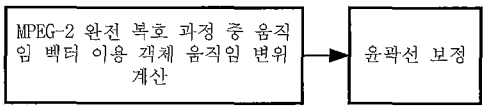


그림 2. 제안된 알고리즘을 이용한 MPEG-2 내 객체 추적

본 논문에서는 기존의 비압축영역에서의 알고리즘과 같은 객체 추적 성능을 가지면서, MPEG-2 압축 영역의 정보를 이용하여 연산 비용을 대폭 줄인 알고리즘을 제안하다. 그림 2는 제안 방법의 객체 추출 과정을 나타내고 있다. 그림으로부터 제안된 방법이 객체 움직임 변위 계산 시 그림 1과 같은 추가 비교/검색 계산이 없이 객체의 움직임 변위가 계산됨을 보여준다.

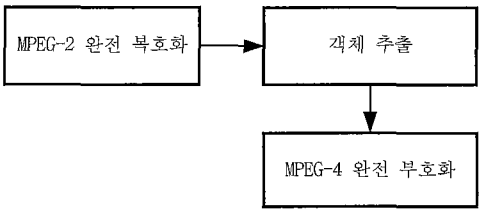


그림 3. MPEG-2에서 MPEG-4로의 변환

화면간 객체의 움직임 예측 과정 후 정확한 객체의 경계선 추출을 위해 경계 주위 일정 영역에 대해서 재분류 작업을 한다[4][5]. 본 논문에서는 객체의 예측 에러를 줄이는 효율적인 경계선 설정을 위한 탄력적 미결정영역 결정 방법을 제안한다. 결정된 미결정 영역에 대해서는 고속 워터쉐드 알고리즘을 이용하여 보정한다.

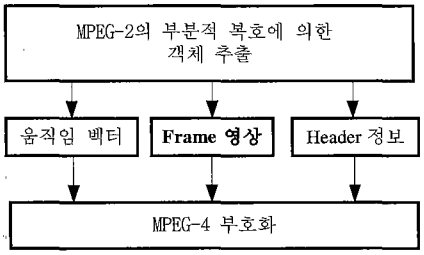


그림 4. 프레임 기반 MPEG-2에서 MPEG4로의 변화

객체 추출 과정을 통해서 추출된 객체 정보는 객체 기반 MPEG-4 영상 입력에 사용된다. MPEG-4는 다양한 멀티미디어 환경에서 사용가능 하도록 개발되어 사용되고 있다. 그러나 기존의 대부분의 영상은 MPEG-1,2로 압축되어 관리되고 있으므로 목적에 따라서는 MPEG-1,2에서 MPEG-4로 변환이 필요하다. MPEG-2에서 객체 기반 MPEG-4로의 변환을 위한 가장 기본적인 방법은 그림 3과 같다. MPEG-1,2의 완전 복호화를 거친 후, 비 압축 영역에서 객체 움직임 예측과 추출과정, MPEG-4로 완전 부호화의 3단계를 거치는 방법이다. 그러나 이와 같은 방법은 변환에 많은 비용이 필요하다. 특히 객체 추출과 MPEG-4 부호화에 있어 움직임 예측과정은 많은 비용을 필요로 한다. 이와 같은 문제를 해결하기 위해 최근에 움직임 벡터를 재 사용함으로써 MPEG-2로부터 MPEG-4로의 변환에 필요한 비용을 줄이는 방법들이 연구되고 있다[6][7]. [6][7]의 MPEG-2에서 MPEG-4로의 고속 변환을 위한 알고리즘은 그림 4에 나타나듯이 MPEG-2의 완전 복호화가 필요하다. 그러나 MPEG-2 복호화에서 얻은 움직임 벡터를 MPEG-4 움직임 예측과정에 재 사용함으로써 MPEG-4의 부호화 시간 단축을 한다. 이상의 알고리즘이 공통적으로 MPEG-2의 완전 복호화를 필요로 하는 이유는 MPEG-4가 저 수준의 처리 방법이 MPEG-2와 다르며 MPEG-4는 MPEG-2에 대해서 하위 호환성을 가지지 않기 때문이다.

[6][7]의 방법은 프레임 기반 MPEG-4로의 변환에 사용되었다. 본 논문에 제안된 MPEG-2에서의

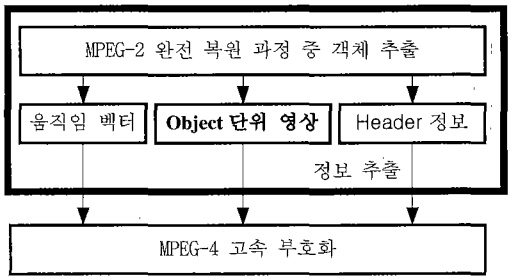


그림 5. 본 논문에서 사용된 MPEG-2에서 MPEG-4로의 변환 정보 추출

정보 추출 방법을 적용하면, 그림 3에서 설명된 방법과 유사한 성능을 가지면서 MPEG-2에서 객체 기반 MPEG-4로의 고속변환이 가능하다. MPEG-2로부터 객체 기반 MPEG-4로의 변환 블록도는 그림 5와 같다. 그림 5에서 보듯이 제안된 알고리즘은 MPEG-2로부터 정보를 고속으로 추출하고, 추출된 정보는 MPEG-4 부호화에 재 사용됨으로써 객체 기반 MPEG-4로의 고속 변환이 된다. 본 논문에서는 그림 5의 질은 사각 박스에 의해 둘러싸여진 객체 기반 MPEG-4 고속 변환에 필요한 고속 정보 추출 단계까지 제안한다. 제안 방법과 MPEG-2 움직임 벡터를 재 사용하는 [6][7]의 방법을 같이 사용하면 그림 3에서의 움직임 예측 과정을 대폭적으로 줄일 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서 객체 추출에 대해 설명하고, 3장에서 객체 추적 단계, 4장에서 MPEG-4로의 고속 변환을 위한 정보 추출, 5장에서 실험을 통한 결과를 보이며, 마지막으로 6장에서 결론을 맺는다.

Ⅱ. 반자동 추적 객체 추출

화면 내에서 객체를 추출하기 위해서는 화면 내의 영상 분할 작업이 수행되어야 한다. 완전 자동 객체 분할인 경우는 화면 내에서 다중 형태학적 경사도(multiscale morphological gradient)를 이용하여 분할 한 후 움직임 예측을 적용하여 움직이는 객체를 추출하는 방법 등이 있다[8]. 이 같은 완전 객체 분할 알고리즘은 연산에 많은 시간이 걸리고 움직임이 없는 객체의 추출이 매우 어렵다.

본 논문에서 사용한 객체 추출 방법은 선택 영역의 자동 분류/분할, 관질 영역 정의, 사용자에 의한 객체 선택의 3단계로 이루어져 있다[5]. 최초 단계로, MPEG-2 비트스트림 중 최초 I 프레임의 영상을

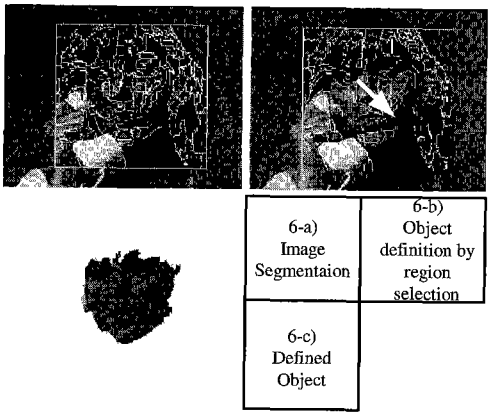
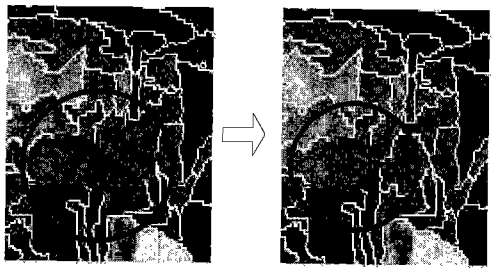


그림 6. 화면 내 분할을 통한 객체 추출 단계

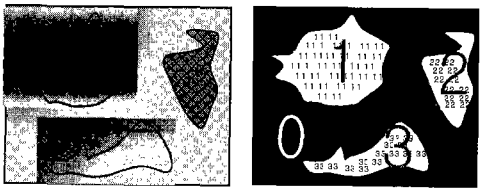
복호하여 원 영상을 복원한다. 사용자가 복원 영상에서 추적하고자 하는 객체를 포함하는 박스 영역을 마우스 끌기를 이용해 설정하면 선택 영역은 블록 기반 영상 분할 기법을 통하여 컴퓨터에 의해서 자동적으로 분할된다[5]. 사용자는 주어진 선택 박스 내 분할 정보에 대해 간단한 마우스 선택을 통해서 추적 객체를 정의하게 된다. 객체 정의 과정에 있어 영상의 분할 정보가 주어지므로 사용자는 보다 쉽게 의미적 객체 정의를 할 수 있다. 그림 6은 의미 있는 객체를 추출하는 과정을 보여준다. 그림 6-a)는 의미적 객체를 포함하는 선택 영역에 대해 블록 기반 영상 분할을 통하여 균질 영역을 얻는 과정을 보여준다, 그림 6-b)는 마우스 끌기를 통한 객체를 정의 과정을 보여주고, 그림 6-c)는 최종 의미적 객체 선택을 보여준다.



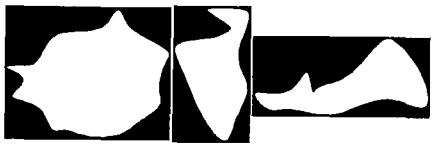
a)객체와 배경의 병합 b) 컷에 의한 재분류

그림 7. 사용자 개입에 의한 영역 재분류

분할 과정에서 객체와 배경의 그레이 밝기 값이 유사한 특정 경우에 있어서는 그림 7-a)와 같이 객체와 배경의 경계 부분이 같은 영역으로 판단 병합되는 경우가 발생한다. 이와 같은 경우 그림 7-b)와 같이 마우스 클릭과 컷의 개입에 의해 다른 영역으로 재분류한다[5]. 이와 같은 일련의 반복 과정을 거쳐 영상 내에서 추적하고자 하는 객체를 여러 개 선택할 수 있다.



8-a) 영상 내 의미적 객체. 8-b) α-plane 영상



8-c) α-plane내 객체 마스크 영상

그림 8. 객체와 객체 마스크, α-plane 영상

한 영상 내 선택된 여러 객체를 구분하기 위해서 각 객체에는 객체 식별 번호가 부여되고, 객체 식별 정보의 처리를 위해 복원 영상과 같은 크기를 가지는 배열이 생성된다. 생성된 배열에는 객체 식별 번호를 통한 객체의 식별 정보가 포함된다. 선택된 추적 객체마다 서로 다른 임의의 정수의 객체 식별 번호가 주어진다. 예로서 그림 8-a)와 같이 영상 내 객체가 3개 정의되었다고 가정하면, 각 객체에는 그림 8-b)와 같이 임의의 서로 다른 3개의 정수 값의 객체 식별 번호가 주어진다. 그림 8-b)의 설명에서는 객체 식별 번호로 정수 값 1,2,3을 사용되었으나, 실제로는 임의의 양의 정수의 값이 주어진다. 객체에 따라 주어진 식별번호는 영상 객체 내부 위치와 같은 위치의 배열에 값이 주어진다. 즉 객체 식별 번호는 화소 단위로 주어지며 의미적 객체 내부의 모든 화소 위치의 배열은 그림 8-b)에서 보듯 같은 객체 식별 번호를 갖는다. 배경에 대해서는 0의 값이 주어진다. 복원 영상에 대한 객체 식별 정보를 가지게 되는 배열을 α-plane이라 부른다. α-plane 내에서 같은 식별 번호를 가지는 영역을 포함하는 최소 사각 영역을 객체 마스크라 부른다. 그림 8-c)는 객체마스크를 보여주며, 8-b)는 α-plane이 된다. 그림 8-b)에서 보듯이 α-plane내에는 의미적 객체 개수와 같은 수의 객체 마스크가 존재한다.

Ⅲ. 고속 객체 추적

2장을 통해서 선택된 객체는 동영상의 시간 축을 통해서 자동으로 추적이 된다. 본 논문에서 사용된 객체 추적 방법을 본 절을 통해서 설명한다.

3.1 객체 추적 단계의 개괄

그림 9는 객체 추적 과정의 전체적인 순서도를 보여주고 있다. 객체 추적 단계의 최초 작업으로 영상간 객체의 움직임 변위를 구하기 전 한 장의 MPEG-2 프레임이 복호된다. 복호화 과정에서 얻어진 정보를 이용하여 현재 복호중인 프레임의 α-plane을 생성한다. 현재 복호 프레임의 대략적인 α-plane은 프레임 복원에 사용된 참조 프레임의 α-plane과 복호 과정에서 얻어진 MPEG-2의 움직임 벡터를 이용한 움직임 예측을 통해서 생성되고, 이는 각 객체의 이전 영상과의 움직임 변위 계산에 이용된다. α-plane정보를 이용하여 객체의 움직임 변위를 계산하기 이전에 객체에 따른 참조 프레임을 결정한다. 움직임 변위를 구하고자 하는 객체에 대한 객체 마스크와 가장 유사한 객체 마스크를 포함하는 MPEG-2 참조 프레임을 객체 참조 프레임이라 부른다. 현재 추적 위치 계산 중인 객체가 포함된 MPEG-2 프레임 타입이 P 프레임인 경우

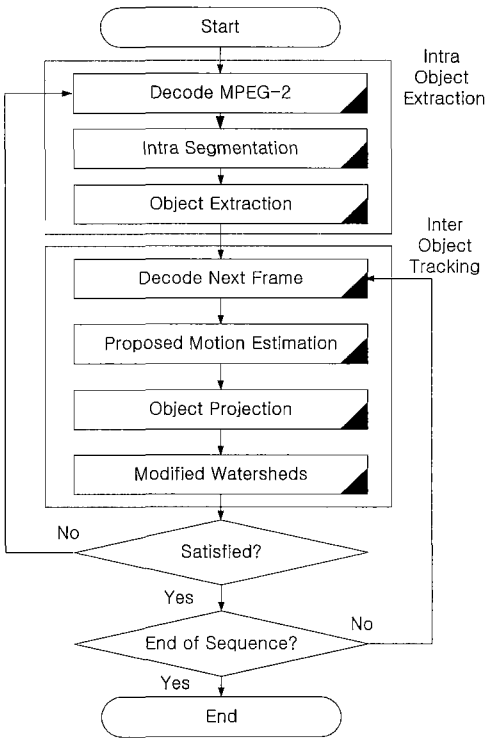


그림 9. 객체 추적 단계 전체 순서도

MPEG-2 복원 순서 상 바로 이전에 복원된 I나 P 프레임이 현재의 객체 마스크와 가장 유사한 모양을 가지는 객체 참조 프레임이다. B 프레임인 경우는 다른데 B 프레임 복원에 사용된 I와 P 두 프레임 중에서 현재의 객체 마스크와 가장 유사한 마스크를 가지는 프레임이 객체 참조 프레임이다. 각 객체 마스크는 하나의 객체 참조 프레임을 가진다. 따라서 같은 영상 내 객체라 하더라도 객체 참조 프레임이 다를 수 있다. 현재 복호 프레임 내의 객체와 객체 참조 프레임내의 객체 사이의 움직임 변위는 3.2절에서 설명할 움직임 예측 방법을 통해 계산된다. 이 후 객체 참조 프레임의 객체 마스크를 계산된 움직임 변위를 이용 현 프레임에 투영한다. 그리고 투영 에러는 경계 재 설정 과정을 통해서 보정된다. 경계 재 설정 과정은 3.3절에서 자세히 설명한다. 마지막으로 현 영상의 객체가 객체 참조 프레임의 객체 영상과 유사한지 판단되고, 만약 유사하지 않다고 판단되면 객체의 추적을 중지하고, 다음 P 나 I 프레임에서 다시 사용자의 객체 정의를 요구한다. 사용자에 의해 객체가 재 정의되면 같은 방법으로 다시 객체를 추적한다.

3.2 제안된 객체의 움직임 변위 계산

동영상 내 객체의 추적을 위해서는 영상 간 객체의 움직임 변위를 구하는 작업이 필요하다. 본 논문에서는 고속 움직임 변위 계산 알고리즘으로 객체 마스크의 가로와 세로 축에 대해 각각 1차원 투영된 객체 모양 히스토그램을 사용한 방법을 제안한다. 참조 객체 프레임 내의 객체와 현 영상 내의 객체의 움직임 변위를 구하는 방법을 설명하기 앞서 앞 절들을 통해 설명된 용어를 수학적 표기를 이용해 정의하고, 이를 사용하여 객체 추적 방법에 대해서 상세히 설명한다.

MPEG-2를 복호하여 얻은 원 영상을 R_k라 표기한다. 여기서 k는 MPEG-2 비트스트림상의 복호화 순서로써 현재 복호 프레임을 나타낸다. 현재의 영상 R_k에 대한 객체 식별 정보를 가지게 되는 2차원 배열인 α-plane은 A_k라 표기하고, k프레임 내 i번째 객체의 객체 마스크는 $OB_{k,i}$라 표기한다. 현재 프레임과 객체 참조 프레임간의 거리 즉 복호화 순서상의 차이는 t이다. 따라서 객체 참조 프레임은 R_{k-t}, 객체 참조 α-plane은 A_{k-t}이다. 그리고 객체 참조 α-plane내에서 객체 마스크 $OB_{k,i}$와 가장 유사 모양을 가지는 마스크인 객체 참조 마스

크는 $OB_{k-1,i}$가 된다. 또한 영상의 수평 좌표 x와 수직 좌표 y에서의 값은 $R_k(x,y)$, $A_k(x,y)$, $OB_{k,i}(x,y)$이다.

객체 추적에 필요한 영상 간 객체의 움직임 정보를 부호화에 최적화된 MPEG-2의 움직임 벡터를 통해 바로 얻을 수는 없다. 객체의 움직임 변위를 얻기 위한 최초 작업으로 현재 프레임의 대략적인 α-plane인 A_k를 생성한다. A_k 생성 시 사용된 방법은 MPEG-2의 움직임 예측과정과 같다. 현재 생성하고자 하는 α-plane이 P 프레임의 α-plane인 경우 I 프레임의 α-plane과 MPEG-2 움직임 벡터를 이용하여 생성되고, B프레임의 경우 I 프레임과 P 프레임의 α-plane과 움직임 벡터를 이용하여 생성된다. 이 과정을 통해 생성된 α-plane은 정확하지는 않지만 대략적인 객체의 모양과 위치 정보를 가진다. 보다 정확한 모양을 위한 후처리 과정으로 인트라 블록에 대한 처리가 필요하다. 현재 복원 영상 R_k의 매크로 블록의 코딩타입이 인트라인 경우 같은 위치의 α-plane 값은 0의 값을 가지게 된다. 이에 대한 처리로 매크로 블록이 객체 내부에 있으면 주위와 같은 객체 식별 번호를 부여한다. 인트라 매크로블록의 상, 하, 좌, 우 4방향 매크로블록이 같은 객체 식별 번호를 포함 시 매크로블록 내부 위치의 화소에 주위 매크로블록과 같은 객체 식별번호 i가 부여된다.

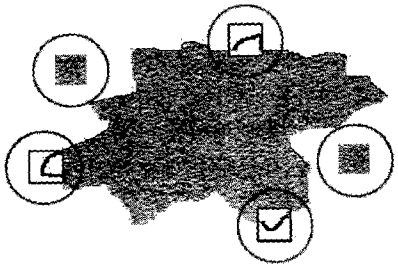


그림 10. 부호화에 최적화된 MPEG-2 움직임 예측으로 인한 객체 마스크 영상의 에러

현재까지 생성된 A_k를 통해서 복원 중인 영상 내의 객체 예상 지역은 알 수 있지만 이 정보만을 통해서 두 영상사이에 객체의 움직임 변위를 알 수 없다. 또한 MPEG-2 움직임 벡터만을 통해서 생성된 α-plane내의 객체 마스크 $OB_{k,i}$는 부호화에 최적화된 MPEG-2 부호기의 특성으로 인해 그림 10과 같이 객체 외부 지역에 객체 식별 번호 i를 가지는 영역이 발생하거나, 경계 영역의 모양변화로 인한 인트라 블록의 생성, 또는 비교적 큰 16×16 매크로블록 단위의 예측으로 인해 실제 객체와의 차이와 같은 에러가 많이 생긴다. 따라서 1차적으로 생성된 $OB_{k,i}$정보만을 이용해서는 대략적으로 비슷한 객체 모양과 위치 정보는 얻을 수 있으나, 정확한 객체의 모양 정보는 얻을 수 없다.

MPEG-2 움직임 예측을 통해서 생긴 α-plane내의 각각의 객체에 대한 참조 객체 프레임의 객체와의 움직임 변위를 계산하기 앞서 우선 객체에 따른 참조 객체 프레임이 결정 방법을 설명하겠다. 본 논문에서는 객체 마스크 구성에 쓰인 MPEG-2 움직임 벡터의 개수를 이용하여 객체 참조 프레임을 결정하였다. 우선 현재 복원 중인 MPEG-2 영상 프레임이 B 프레임이고 예측에 사용된 프레임이 I와 P 프레임이라 가정하자. B 프레임 내 객체 마스크의 구성에 사용된 매크로블록의 개수를 계산한 결과 I프레임보다 P 프레임으로부터의 예측이 더 많은 경우, P 프레임이 객체 참조 프레임이 된다. 객체 참조 프레임이 결정되고 객체간의 움직임 변위가 계산된다.

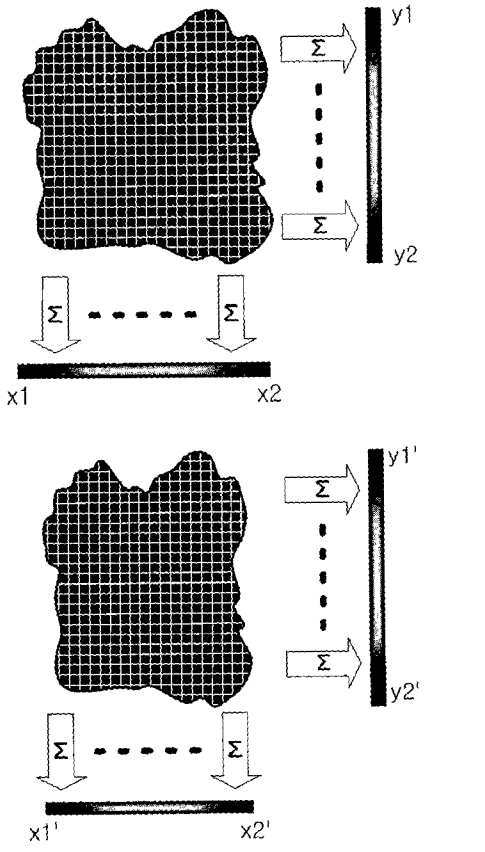


그림 11. 현재 프레임과 참조 객체 프레임 객체 마스크의 2차원 투영

본 논문에서 제안하는 움직임 변위 계산은 MPEG-2 움직임 예측을 통해 생긴 대략적인 객체 마스크인 $OB_{k,i}$와 이전 프레임을 통해 보정 과정을 거친 $OB_{k-t,i}$를 이용하여 계산된다. 최초 작업으로 움직임 변위를 구하고자 하는 현재 프레임과 객체 참조 프레임의 객체에 대한 마스크 $OB_{k,i}$와 $OB_{k-t,i}$의 값을 가지는 화소의 개수를 그림 11과 같이 가로 방향, 세로 방향의 2차원으로 투영한다. 객체마스크 $OB_{k,i}$에 대한 y축에 대한 투영을 통해 $Y_{k,i}$, x축 투영을 통해 $X_{k,i}$가 생성되고, $OB_{k-t,i}$의 y축, x축 투영을 통해서 $Y_{k-t,i}$, $X_{k-t,i}$가 생성된다. 현 영상과 객체 참조 프레임의 객체 마스크가 그림 11과 같은 좌표에 분포할 때 $OB_{k,i}$의 x열과 y행에서의 투영 히스토그램 $Y_{k,i}(y)$와 $X_{k,i}(x)$는 식 (1)에 의해 생성한다.

$$Y_{k,i}(y) = \sum_{x=x1}^{x2} C(x,y) \ , \ y1 \le y \le y2 \qquad (1)$$

$$X_{k,i}(x) = \sum_{y=y1}^{y2} C(x,y) \ , \ x1 \le x \le x2$$

여기서

$$C(x,y) = \begin{pmatrix} 1, & \text{for} \quad OB_{k,i}(x,y) > 0 \\ 0, & \text{for} \quad OB_{k,i}(x,y) = 0 \end{pmatrix}$$

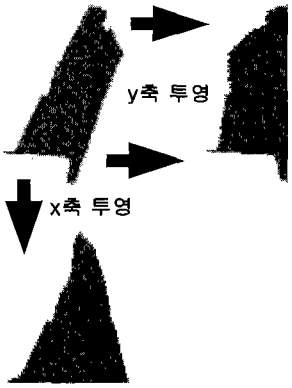


그림 12. 객체 마스크를 1차원 투영한 영상

그림 12는 실제 객체 마스크를 x축과 y축에 대해 투영을 한 영상이다. 생성된 투영 히스토그램은 객체에 대한 모양 정보를 포함하고 있다. 이와 같이 1차원 투영을 함으로써 나타나는 특징은 객체 마스크를 1차원 확률 분포 함수로 볼 수 있다는 것이다. 투영에 따른 장점으로 객체의 모양을 서술하는 분포 함수의 모양을 평균과 분산으로 나타남으로써 보다 간단한 서술과 비교가 가능해진다. 그리고 간단한 연산을 통해 MPEG-2 움직임 예측을 통해 나타나는 객체 경계 영역과 외부에서의 에러가 제거될 수 있다. 즉, 에러에 대한 처리가 간단하고 경계 주위의 모양변화에도 강인하다.

식 (1)을 통해서 구해진 히스토그램 $Y_{k,i}$, $X_{k,i}$와 $Y_{k-t,i}$, $X_{k-t,i}$를 1차원 확률 분포 함수로 생각한다. 각각의 분포 함수에 대한 대표 값으로 평균을 사용한다. 각각의 $Y_{k,i}$, $X_{k,i}$와 $Y_{k-t,i}$, $X_{k-t,i}$에 대한 평균 $\overline{X}_{k,i}$, $\overline{Y}_{k,i}$와 $\overline{X}_{k-t,i}$, $\overline{Y}_{k-t,i}$ 는 식 (2)을 통해서 계산된다. 단 여기서 $\overline{X}_{k,i}(x)$의 x축에 대한 분포 범위는 $x1 \le x \le x2$, $\overline{X}_{k-t,i}(x)$는 $x1' \le x \le x2'$이고 $\overline{Y}_{k,i}(y)$의 y축에 대한 분포 범위는 $y1 \le y \le y2$, $\overline{Y}_{k-t,i}(y)$는 $y1' \le y \le y2'$이다.

$$\overline{X}_{k,i} = \frac{\sum_{x=x1}^{x2} x \times X_{k,i}(x)}{\sum_{x=x1}^{x2} X_{k,i}(x)}, \quad \overline{Y}_{k,i} = \frac{\sum_{y=y1}^{y2} y \times Y_{k,i}(y)}{\sum_{y=y1}^{y2} Y_{k,i}(y)}$$

$$\overline{X}_{k-t,i} = \frac{\sum_{x=x1'}^{x2'} x \times X_{k-t,i}(x)}{\sum_{x=x1'}^{x2'} X_{k-t,i}(x)},$$

$$\overline{Y}_{k-t,i} = \frac{\sum_{y=y1'}^{y2'} y \times Y_{k-t,i}(y)}{\sum_{y=y1'}^{y2'} Y_{k-t,i}(y)} \qquad (2)$$

에러에 대한 처리 과정은 다음과 같다. 현 영상에서 객체 경계 주위의 큰 변화로 인해 MPEG-2 부호화 과정 중 매크로블록이 인트라블록으로 처리되었을 경우, 이로 인하여 객체 마스크의 경계에서의 모양은 변화하게 된다. 투영 분포 함수의 값의 분포 범위 또한 줄어든다. 그러나 이와 같은 경계에서의 에러는 분포함수의 평균에 적은 영향을 미친다. 값들이 객체의 중심 영역에 많은 분포를 하고 경계 영역에는 값이 적게 분포하기 때문이다. 또한 부호화에 최적화된 MPEG-2 움직임 예측에 의한 객체 외부 지역에서 나타나는 에러는 투영된 1차원 함수에서 보았을 때 함수의 평균값에서 멀리 위치한 곳에 나타난다. 이와 같은 특징을 활용하면 움직임 예측에 의해 잘 못 나타난 에러 즉 아웃라이어(outlier)는 함수의 표준 편차를 이용하여 간단히 제거 될 수 있다. 아웃라이어는 평균값에서 멀리 떨어진 양끝 주변에 나타나므로 확률 분포 함수에서 아웃라이어를 제거하는 방법과 같은 방법으로 제거된다. 식 (2)을 통해 구해진 평균 $\overline{X}_{k,i}$와 $\overline{Y}_{k,i}$ 에

의한 $Y_{k,i}$와 $X_{k,i}$의 표준 편차를 $\sigma_{x,k,i}$, $\sigma_{y,k,i}$라 할 때, 평균에서 표준편차의 2배 이상 떨어진 곳에 위치하는 값은 아웃라이어로 계산되고 제거된다. 제거 과정을 통해 갱신된 $Y_{k,i}(y)$와 $X_{k,i}(x)$의 평균을 식 (2)에 의해 재 계산하여 $\overline{X}_{k,i}$와 $\overline{Y}_{k,i}$를 얻는다. 두 객체 사이의 움직임 변위 계산은 식 (3)을 통해서 구해진다. 구해진 dx, dy를 통해 객체 참조 프레임의 객체 마스크를 그림 13과 같이 현재의 영상에 투영한다

$$dx = \overline{X}_{k,i} - \overline{X}_{k-t,i}$$
$$dy = \overline{Y}_{k,i} - \overline{Y}_{k-t,i} \quad (3)$$

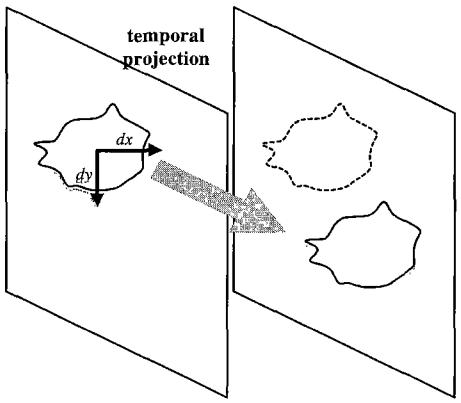


그림 13. 움직임 변위 dx, dy에 의한 객체 마스크의 시간축에 대한 투영

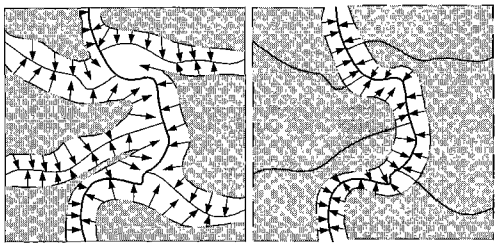
그러나 항상 아웃라이어가 존재하는 것은 아니다. MPEG-2 움직임 벡터를 사용해 생성된 1차 객체 마스크가 아웃라이어를 가지지 않는 재대로 예측이 된 경우라 하자. 이와 같은 경우 아웃라이어를 계산하고 다시 갱신하는 연산은 불필요한 과정이 된다. 따라서 아웃라이어가 발생하지 않았을 경우 제거 연산을 하지 않고 다음 객체의 움직임 예측을 해야 한다. 본 논문에서 아웃라이어가 발생했는지를 판단 결정 시 사용한 방법은 다음과 같다. 우선 $\overline{X}_{k-t,i}$와 $\overline{Y}_{k-t,i}$는 이전 단계에서 보정을 거친 객체에 대해 정확한 정보를 가지고 있는 마스크의 투영 값이므로 아웃 라이어 제거연산이 불필요하다. $\overline{X}_{k,i}$와 $\overline{Y}_{k,i}$ 에 대해서만 판단 계산을 한다. 아웃 라이어 발생 판단 연산은 분포 함수가 분포하는 영역 대 평균의 위치를 사용하여 결정된다. 아웃라이어가 생겼을 경우의 특징은 평균에서 멀리 떨어진 곳에 값이 존재한다는 것이다. 예로서 투영에 의한 x축 분포 함수의 경우로 $X_{k,i}$의 x축에 대한 분포 범위가 $x1 \le x \le x2$ 이고 $X_{k-t,i}$의 분포 범위가 $x1' \le x \le x2'$ 이라 가정하자. $X_{k,i}$에 아웃라이어가 발생한 경우 객체 참조 마스크의 투영 함수 $X_{k-t,i}$에 비해 값의 분포 범위가 넓어진다. 그러나 x의 범위가 넓어지는 경우는 아웃라이어가 발생한 경우와 객체의 크기가 변하는 경우도 있으므로 이의 구분이 필요하다. 객체가 커지는 경우는 x의 범위가 넓어지지만 분포함수의 양끝 점으로부터의 평균값까지의 거리의 비율은 유사하다. 따라서 크기가 확장되는 경우 이전 프레임의 $\frac{\overline{X}_{k-t,i}-x1'}{x2'-x1'}$과 $\frac{\overline{X}_{k,i}-x1}{x2-x1}$ 사이의 값은 변화가 적다. 그러나 아웃라이어가 발생한 경우 x의 범위는 크게 늘어난 반면 평균의 위치의 변화는 적다. 따라서 $\frac{\overline{X}_{k-t,i}-x1'}{x2'-x1'}$과 $\frac{\overline{X}_{k,i}-x1}{x2-x1}$ 사이에 값의 차이가 생긴다. $\frac{\overline{X}_{k-t,i}-x1}{x2-x1}$과 $\frac{\overline{X}_{k,i}-x1}{x2-x1}$의 값을 비교하여 값의 차이가 나면 아웃라이어가 있는 것으로 제거 연산을 수행한다. 아웃라이어가 있다고 판단되는 축에 대해서만 아웃 라이어 제거 연산을 한다.

현재 영상의 객체 마스크에 대해 구해진 $\overline{X}_{k,i}$, $\overline{Y}_{k,i}$ 는 다음에 나타나는 영상 내 객체 마스크와의 거리 계산에도 쓰인다. 따라서 현재 계산 된 객체 참조 마스크가 복호화 순서 상 다음에 나오는 영상 내 객체 마스크의 참조 마스크인 경우 이미 계산, 보정된 $\overline{X}_{k,i}$, $\overline{Y}_{k,i}$를 이용하므로 추가로 참조 객체의 평균값을 구하는 연산은 불필요하다.

3.3 경계선의 재 검출

3.2 절에서 설명된 객체 마스크의 그림 13과 같은 투영과정을 통해 현 영상에 생긴 객체 마스크와 실제 객체와는 차이가 있다. 이는 추적하는 의미적 객체의 다양한 움직임에 의한 에러로써, 대부분 객체의 경계에서 발생한다. 이 같은 에러를 제거하기 위하여 시간 축을 통해 투영된 객체 마스크의 경계선 내부와 외부의 일정 부분의 픽셀에 대해서 재분류 작업을 한다. 픽셀 단위 재분류는 워터쉐드 알고리즘을 사용한다. 기본적인 워터쉐드 알고리즘은 거리 측정 값으로 모든 영역에 대한 영역의 평균 밝기 값을 사용한다. 그림 14-a)에서 보여지는 기본적인 워터쉐드 알고리즘은 균질 영역의 마커(영역의 평균 밝기)와 미결정영역의 화소간의 거리를 계산하여

갱신하는 방법을 사용한다. 미결정영역의 화소와 마커와의 계산을 통해 가장 밝기 차이가 적은 영역에 화소가 포함되고 다시 마커 값이 갱신된다. 그러나 이와 같이 미결정영역의 모든 화소에 대해서 주위 영역과 거리를 계산하고 밝기를 갱신하는 방식은 많은 컴퓨터의 연산을 필요로 한다. 대신 본 논문에서는 각 픽셀의 기울기 값을 이용하여 영역의 라벨에 상관없는 방법을 사용한다[5]. 화소의 기울기 정보를 통하여 일정 임계 치에 도달할 때까지 물을 채우는 방법으로 주위 마커와의 비교 연산이 필요하지 않다. 그럼으로써 그림 14-b)에서 보여지듯이 경계주위의 적은 영역에 대한 재분류 작업이 가능하다. 또한 충전 기법과 큐를 사용한 수정된 워터쉐드 알고리즘을 사용하여 컴퓨터 수행시간을 줄였다[5].



a) 마커에 기반한 영역 확장 b) 사용된 영역 확장

그림 14. 영역 확장 방법의 비교

워터쉐드 알고리즘을 적용하기에 앞서 객체의 경계선 주위 일정 부분 미결정영역을 선정해야 한다. 그러나 모든 영상에 대해서 같은 범위 값의 미결정 영역을 주는 것은 비효율적이다. 만약 객체의 미결정 영역 설정 시, 부정확한 영역의 설정에 의해 객체 윤곽선이 미결정 영역 밖으로 나간다면, 정확한 객체 윤곽선을 찾아내지 못할 것이다. 이를 해결하기 위해 미결정 영역을 넓게 설정하여 계산하면 가능하지만, 처리 속도가 느려지게 된다. 따라서 상황에 맞는 미결정 영역을 탄력적으로 설정하면 더 빠른 처리 속도를 얻을 수 있다. 탄력적 미결정 영역 설정의 기존 방법은 움직임이 심한 경우 움직임 벡터 값에 의존하여 비례적인 미결정 영역을 결정한다[5]. 그러나 본 논문에서는 실제적인 객체의 모양 정보를 이용하여 미결정영역을 결정한다. 실제로 객체의 움직임 변위 값과 모양의 변화는 비례한다고 볼 수 없기 때문이다. 따라서 실제적인 크기 변화를 관찰함으로써 미결정 영역을 정하였다. 객체의 아웃라이어의 제거 과정에서 얻은 객체 마스크들의 투영 정보 히스토그램의 표준 편차 값의 차이를 이용하여 미결정영역을 설정한다. 표준 편차의 차이가 큰 경우 넓은 미결정 영역을 선택하고 차이가 적은 경우 작은 미결정 영역을 설정한다. 그림 15를 보면 객체가 축소되면서 움직이고 있음을 알 수 있다. 객체의 모양 변화가 이와 같이 탄력적인 때 미결정영역을 작게 잡아주거나 크게 잡아주면 오차를 보정할 수 있고 연산 속도 또한 올릴 수 있다. 미결정 영역 설정은 여러 번의 실험을 통해 실험 치에 의해 설정하였다.

3.4 정지 조건

객체의 모양에 있어 많은 변화가 있을 시 객체의 추적을 중지하고 사용자의 객체 재 설정을 요구한다. 객체의 추적 정지 조건은 객체 참조 프레임의 객체 마스크의 평균에 대한 분산과 현 영상의 객체 마스크의 평균값에 대한 분산 값의 차이가 크게 날 때이다. 차이 값으로는 실험에 의한 값이 사용되었다. 분산의 차이가 크게 나온 것은 객체 마스크에 있어 큰 변형이 일어난 것이다. 정지 조건이 만족되면 추적을 중단하고 이후에 나타나는 P 프레임이나 I 프레임을 복호하여 사용자에 객체 재 정의를 요구한다. 사용자에 의해 객체의 재 정의가 이루어진 이후에 이전과 같은 방법으로 객체의 추적이 재 수행된다.

Ⅳ. MPEG-4를 위한 정보 추출

본 논문에서 제안된 방법을 사용하면 고속 객체 기반 MPEG-4로의 변환에 적용할 수 있는 객체 영상과 객체 내부의 MPEG-2의 매크로블록단위의 움직임 벡터, 형상 영상, 헤더 정보를 추출할 수 있다. 헤더 정보는 MPEG-2의 헤더 정보로부터 추출된다. MPEG-2의 움직임 벡터를 이용하여 고속 MPEG-4 부호화를 할 수 있는 이유는 MPEG-4의 움직임 예측 방법과 MPEG-2의 움직임 예측 방법이 유사하기 때문이다. MPEG-4는 동영상 부호화에 있어 두 가지 방법이 있는데 그 중 하나는 프레임 기반 영상 부호화이고, 다른 하나는 객체 기반 영상 부호화이다. 그러나 두 가지 모두 같은 방식의 영상 신호 부호화기법을 사용한다. 따라서 기존에 제안된 MPEG-2 움직임벡터의 MPEG-4 부호화 과정 중 움직임 예측에 재 사용하는 기법과 제안된 방법을 같이 사용하면, MPEG-2로부터 객체 기반 MPEG-4로의 변환 시간을 대폭 줄일 수 있다. 실험에서는

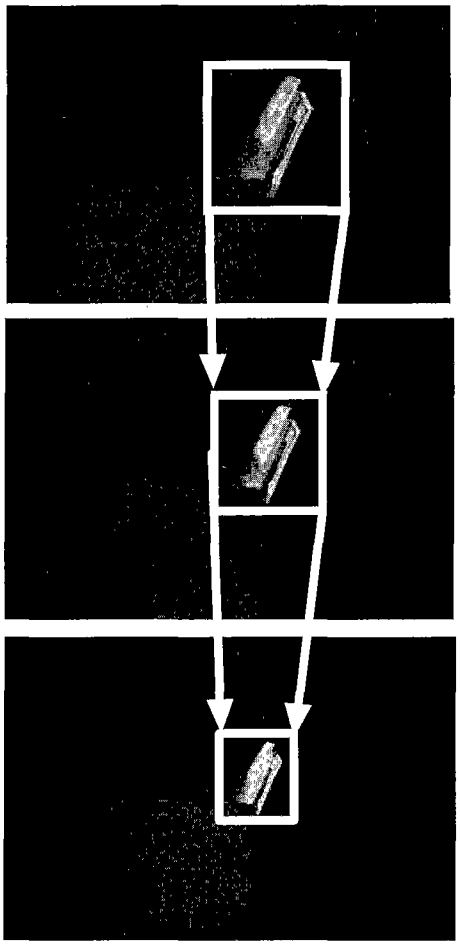


그림 15. 탄력적 미결정 영역의 설정 예

MPEG-2에서 추출된 정보를 파일로 만들어 저장하였다. 추출 정보 중 MPEG-2 움직임 벡터는 객체 외부의 영역은 제외되고, 객체부분을 포함하는 매크로 블록의 움직임 벡터 정보만 추출되어 저장된다. 형상 영상은 현재 프레임의 보정된 α-plane이 사용된다.

V. 실험 결과

본 논문에서 실험은 352*288 CIF 사이즈의 영상에 적용하였다. 실험 영상으로는 Cactus and Comb, flower and garden, mobile and calendar, Akiyo, Susi를 사용하였다. 본 논문에서 제안한 방법과 기존 방법인 BMA 3단계 고속 탐색 방법과 투영기법[5]를 사용한 방법의 움직임 벡터 계산 연산량을 표1에서 비교하였다. 객체의 크기가 N x N이고 탐색에 의한 알고리즘의 경우 탐색범위는 -15 ~ 15이다. 제안된 알고리즘은 표에서 알 수 있듯이 움직임 변위 계산 연산 량이 매우 적음을 알 수 있다. 특히 비교 연산이 기존 알고리즘과 달리 전혀 필요 없다.

그림 16에는 선택 객체의 추적 과정을 보여주고 있다. 선택 객체는 프레임이 진행됨에 따라 크기와 위치에 있어 변화가 있다. 그림 16에 나타난 영상의 경우 수평 이동, 수직 이동, 객체 축소, 복합 움직임으로 이루어져 있는데 제안된 알고리즘을 이용하여 연속된 영상에서 고속으로 추적됨을 알 수 있다. 실험 영상에 대해 기존의 알고리즘을 적용하여 객체 추출을 하였을 경우와 비교하였을 때 같은 결과가 발생한다. 따라서 제안된 방법은 기존의 비압축영역의 방법과 같은 성능을 내면서 속도와 구현상에서 이득을 얻을 수 있다.

표 1. 객체 움직임 변위 연산량 비교

사용 알고리즘	덧셈 횟수	비교 횟수
BMA 3단계 탐색	$6N^2$	33
투영 기법[5]	$4N^2$	66
제안 방법	$2N^2$	0

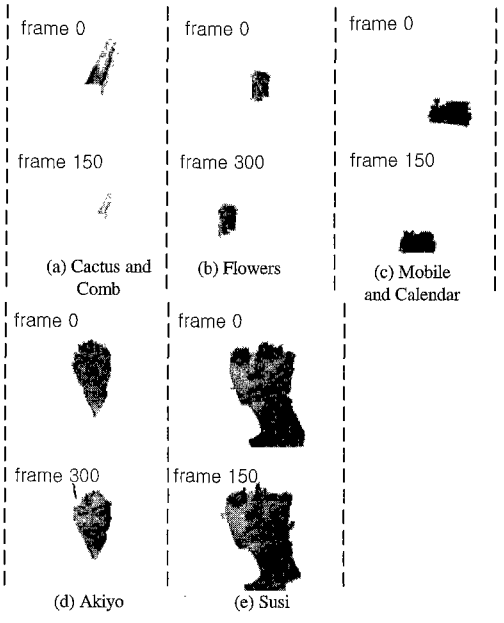


그림 16. 객체 추적 결과

Ⅵ. 결 론

MPEG-2의 영상을 객체기반 MPEG-4로 변환하고자 할 때 객체 추출/추적에 있어 기존에 제안된 비압축영상에 대한 탐색에 의한 객체 추적 방법을 적용하면 많은 비용이 든다. 이에 비해 본 논문에서는 제안된 알고리즘을 통해서 최소의 연산 량으로 정확한 객체를 추출할 수 있었고, MPEG-4로의 변환을 위한 정보를 고속으로 뽑아낼 수 있었다. 추출된 정보 중 MPEG-2로부터 추출된 움직임벡터를 MPEG-4 부호화에 재 사용함으로서 고속으로 MPEG-4 변환이 가능하다. 이를 통해서 기존의 MPEG-2에서 객체 기반 MPEG-4로 변환 시 객체 추출과 부호화 과정에 필요한 복잡한 움직임 예측 과정을 대폭 단순화하였다. 빠르고 정확한 객체의 추출을 통해 MPEG-2상에서 객체기반의 다양한 어플리케이션에 응용이 가능하다.

참 고 문 헌

[1] S. Fukunaga, Y. Nakaya, S. H. Son, and T. Nagumo, MPEG-4 Video Verification Model 15.0, ISO/IEC JTCI/SC29/WG11, Maui, Dec. 1999.

[2] ISO/IEC JTC1/SC29/WG11/W3703 : "CD 15938-3 MPEG-7 Multimedia Content Description Interface Part 3 Visual" October 2000 (La Baule)

[3] Chuang Gu, Ming-Chieh Lee, Semiautomatic Segmentation and Tracking of Semantic Video Objects, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 572-584, Sept. 1998.

[4] Munchurl Kim, J. G. Jeon, J. S. Kwak, M. H. Lee and C. Ahn, "Moving object segmentation in video sequences by user interaction and automatic object tracking," *Image and Vision Computing Journal*, vol. 19, no. 5, pp. 245 ~ 260, April 2001.

[5] Dong Kwon Park, Ho Seok Yoon, Chee Sun Won, "Fast Object Tracking in Digital Video", *IEEE Trans. Consumer Electorics*, Vol. 46, No. 3, pp.785-790, Aug. 2000

[6] Lorenzo Favalli, Alessandro Mecocci, and Fulvio Moschetti Object Tracking for Retrieval Applications in MPEG-2, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 10, No. 3, April 2000

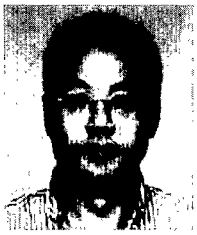
[7] Kuniaki Takahashi, Kazushi Satoh, Teruhiko Suzuki and Yoichi Yagasaki. "Motion Vector Synthesis Algorithm for MPEG2-to-MPEG4 Transcoder", *SPIE Visual Communicatons and Image Processing 2001*.Vol. 4310 pp.872-882

[8] Randy Crane, "A Simplified Approach to Image Processing," Prentice Hall, pp. 206-211, 1997

양 종 호(Jong-Ho Yang) 정회원

2000년 2월 : 동국대학교 전자공학과 졸업
2000년 2월~현재 : 동국대학교 전자공학과 석사과정
<주관심 분야> MPEG-2,7 응용 분야

원 치 선(Chee-Sun Won) 정회원



1982년 2월 : 고려대학교 전자공학과 졸업
1986년 : University of Massachusetts/Amherst 전자공학 석사과정 졸업
1990년 : University of Massachusetts/Amherst 전자공학 박사과정 졸업
1989년~1992년 : 금성사 가전연구소(현, LG전자멀티미디어 연구소) 선임연구원
1992년~현재 : 동국대학교 전자공학과 부교수
<주관심 분야> Image segmentation, digital video coding, digital video browsing and retrieval, and video securities.