

## 2-모드 선택 기반의 압축비디오 신호의 움직임 객체 블록 추출

김 동 욱\*

# Moving Object Block Extraction for Compressed Video Signal Based on 2-Mode Selection

Dongwook Kim\*

### 요 약

본 논문에서는 압축된 비디오 신호의 움직임 벡터 및 DCT 계수로부터 움직임 객체를 추출하는 새로운 기법을 제시한다. 움직임 객체 추출에 관한 기술은 내용 기반 검색, 타겟트래킹 등 다양한 분야에서 필요로 한다. 움직임 객체 블록의 추출을 위해서 움직임 벡터와 DCT계수가 선택적으로 이용되는 2-모드 방식의 기법이 제시된다. 또한, 제시된 기법은 DCT 변환 영역상의 계수들만을 이용하기 때문에 완전히 복호화된 정보를 필요로 하지 않는 장점을 갖는다. 제시된 기법을 바탕으로 몇 가지 테스트 영상에 대해 모의 실험을 실시한 결과 양호한 결과를 얻을 수 있었다.

### Abstract

In this paper, We propose a new technique for extraction of moving objects included in compressed video signal. Moving object extraction is used in several fields such as contents based retrieval and target tracking. In this paper, in order to extract moving object blocks, motion vectors and DCT coefficients are used selectively. The proposed algorithm has a merit that it is no need of perfect decoding, because it uses only coefficients on the DCT transform domain. We used three test video sequences in the computer simulation, and obtained satisfactory results.

▶ Keyword : video signal, moving object extraction, DCT coefficient

---

• 제1저자 : 김동욱  
• 접수일 : 2007. 11.14, 심사일 : 2007. 11.19, 심사완료일 : 2007. 11.26.  
\* 전주대학교 전기전자정보통신공학부

## I. 서론

오늘날 MPEG-1, 2, 4 등에서 기본적으로 적용하고 있는 기법은 블록기반 움직임 추정 및 예측 기술이다 [1]. 비디오 데이터는 이를 통해 얻어진 블록 움직임 벡터와 예측오차를 부호화하여 전송된다. 따라서, 압축된 비디오 데이터로부터 원하는 객체를 추출하기 위해서는 블록 움직임 벡터로부터 얻는 것이 좋은 방법 중의 하나이다 [2][3][4]. 연속 영상을 이용하여 실시간으로 움직임 객체를 추출하고 추적하기 위해 배경분리 기법이 도입되기도 한다 [5].

대부분의 객체 분할 알고리즘은 변화영역 검출로부터 출발한다 [6][7]. 비디오 데이터의 임의의 프레임에서 움직이는 객체의 추출은 배경과 움직이는 객체의 분할이며, 대부분의 분할 기법은 인접한 프레임간의 변화된 화소 검출(change detection)에 의존한다 [8]. 이러한 기법들은 움직이는 물체들은 보통 연속된 프레임 간에 밝기의 변화를 수반한다는 점을 이용하는데, 밝기의 차를 구하고 이로부터 움직이는 객체 영역을 찾아낸다. 이러한 기법은 화소간의 차 연산에 의존하기 때문에 계산이 간단한 반면, 움직이는 내부는 배경으로 처리되는 등의 단점을 가지고 있다. 또한, 화소단위로 움직임 필드를 구성하고 이를 바탕으로 움직이는 객체를 구하는 방법은 계산적으로 매우 복잡하며, 지역 최소값 문제에 빠질 위험을 가지고 있다. 객체 추출의 또 다른 방법으로서 영역 분할 기법을 이용한다 [9][10]. 그러나 이러한 영역 분할 기법은 영역 분할 알고리즘의 문턱치 등의 설정에 따라 과분할되거나 잘못 분할되는 결과를 초래하기 쉽다.

한편, 영상 데이터는 매우 높은 계산적 복잡도를 요구하기 때문에 가능한 한 낮은 연산을 통해 검색을 달성해야 한다. 또한, 영상 데이터는 용량의 크기 때문에 압축된 형태로 저장하거나 전송하는 경우가 대부분이다. 따라서, 원영상이 아닌 압축된 영상을 대상으로 처리하는 것이 계산적 복잡도를 줄이는 측면에서 바람직하다. 본 논문에서는 이러한 압축 비디오 데이터로부터 움직임 객체를 효과적으로 추출하기 위한 새로운 기법을 제시하고자 한다.

대부분의 동영상 압축 포맷은 움직임 벡터와 예측오차에 대한 DCT 계수를 기반으로 한다. 이는 시간적 상관성을 제거하는 가장 효과적인 방법 중의 하나로서 움직임 추정으로 통한 움직임 벡터를 얻고, 추정된 결과를 바탕으로 추정 오차, 즉 예측 오차에 대해 DCT를 적용하여 예측오차간의 상관성을 줄이는 결과를 얻는다. 본 논문에서는 압축 비디오 데이터의 움직임 벡터를 기반으로 전역움직임 정보를 추출

하고, 이를 바탕으로 움직임 정보 및 DCT 계수를 이용하여 객체를 추출 기법을 제시하고자 한다. 제2장에서는 압축영역에서 DCT 계수를 효과적으로 이용하기 위한 방안을 고찰하고, 제3장에서는 객체추출 기법을 제시하고, 제4장에서는 이를 바탕으로 행한 모의 실험 결과를 보이며, 마지막으로 제5장에서 결론을 맺는다.

## II. DFD와 DCT 계수의 고찰

### 2.1 움직임 벡터와 DCT 계수

일반적으로 비디오 신호의 압축은 시간적으로는 움직임 추정에 바탕을 둔 움직임 보상 기법에 의해 중복성을 제거하고, 공간적으로는 DCT변환에 의해 중복성을 제거함으로써 달성된다. 본 논문에서는 움직임 보상을 위한 블록크기로서 16 대상으로 하는 포맷은 MPEG-2 비트 스트림이다. MPEG-2 포맷의 비트스트림에서 움직임 벡터는 매크로블럭(16×16) 당 움직임 벡터를 구성한다.

DFD(displaced frame difference) 기반의 블록기반의 움직임 추정은 영상의 각 블록에 대한 DFD 함수를 최소화하는 과정이다. 블록 움직임 벡터는 정해진 탐색 범위내에서 블록 정합법(block matching algorithm)에 의해 얻을 수 있다. 블록정합을 달성하는 방법으로서 전역탐색법, 고속블럭정합법 등 여러 가지 방법이 있다.

한편, 움직임 추정 결과 얻어지는 움직임 벡터를 바탕으로 움직임 보상을 행하고, 보상 후 남은 예측 오차에 대해 공간적 상관성을 제거하기 위해 DCT를 행한다. DCT의 정의는 다음과 같다.

$$F(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \left[ \frac{\pi(2x+1)u}{2N} \right] \cos \left[ \frac{\pi(2y+1)v}{2N} \right], \dots (1)$$

여기서,  $u, v = 0, 1, 2, \dots, N-1$  이고,  $\alpha(u)$ 와  $\alpha(v)$ 는 다음과 같다.

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0 \end{cases} \dots \dots \dots (2)$$

식 (1)에서  $u = 0, v = 0$  일 때의 첫 번째 변환계수(transform coefficient)는 이 2차원 블록의 평균값을 의미하며, 이 값은 일반적으로 DC 계수(DC coefficient)라

부른다. 나머지의 모든 계수들은 AC 계수(AC coefficient) 들이다. DCT 기법은 여러 가지 영상 압축 기법에 널리 사용되는데, 이것은 공간적으로 인접한 화소들 간의 중복성을 제거하여 비상관화(decorrelation)를 달성하고, 적은 수의 계수로 영상을 표현하는 에너지 집중성(energy compaction)을 얻을 수 있기 때문이다.

만일, 밝기 또는 칼라 값이 아주 느리게 변하는 영상에 대해 DCT를 적용할 경우, DC의 계수 값을 제외한 대부분의 AC값은 0에 가까울 것이다. 반면에 화소간의 밝기 값이나 칼라값이 반복해서 변하거나 그 변화 값이 큰 경우, 앞의 경우에 비해 큰 AC 값을 가질 것이다.

### 2.2 DFD(displaced frame difference)와 DCT 계수와의 관계

MPEG-2, 4 등에서는 비디오 신호 압축을 위해 I-frame에 대해 프레임내 압축을 하고 P, B-frame에 대해 프레임간 압축을 한다. 특히, P, B-frame의 경우 움직임 예측 및 보상 기법을 이용하여 얻어진 움직임 벡터와 예측 오차 성분을 전송한다. I-프레임의 경우 정해진 블록 크기에 대해 직접 DCT를 행하고 그 변환계수를 양자화하여 부호화하며, P, B-프레임은 움직임 예측 후 얻어지는 예측오차를 DCT 변환하여 얻어진 변환계수를 양자화하여 부호화한다.

압축된 비디오 데이터는 원영상에 비해 이미 처리되어 얻어진 여러 가지 의미있는 정보들을 포함하고 있다. 그 대표적인 정보가 움직임 벡터와 예측오차이다. 움직임 벡터의 경우 비트스트림으로부터 움직임 벡터부분을 별도로 쉽게 복호화하여 사용할 수 있으나, 예측 오차 성분의 경우 DCT 계수를 복호화하고, 역변환하여 예측오차를 구할 수 있다. 따라서, 움직임 벡터에 비해 역변환과정을 한 단계 움직임 벡터와 또한 영상의 재구성과정을 거쳐 원영상을 재구성할 수 있다.

대부분의 압축 기법에서 DCT는 8×8 크기의 블록에 대해서 수행된다. 이 때, DCT 계수로써 예측오차를 표현하기 위한 간략한 표현으로서 다음과 같이 나타낼 수 있다. 즉,

$$C_j = DC_{j=1} + \sum_{j=2}^K |AC_j| \dots\dots\dots (3)$$

식(3)에서  $DC_{j=1}$ 은 DCT 계수 중 DC 성분을 나타내며,  $AC_j$ 는 63개의 AC 성분을 나타낸다.

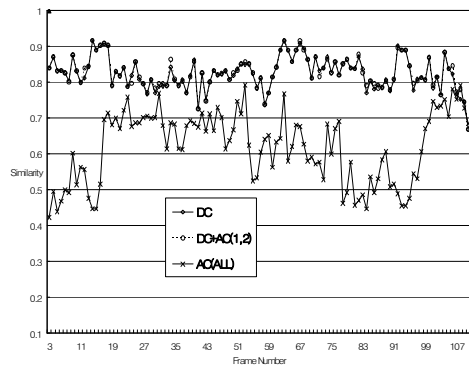


그림 1. DFD와 DCT계수의 유사도  
Fig. 1. Similarities between DFD and DCT coefficients

그림 1은 DFD와 DCT 계수와의 관계를 나타낸 것이다. 본 실험 데이터를 얻은 과정은 다음과 같다. 먼저 블록매칭 알고리즘을 수행하여 각 블록별로 최적의 매칭 블록을 찾고, 그 때 DFD를 얻는다. 그리고, 각 블록에 대해 얻은 DFD에 대해 이치화를 시도한다. 즉, 미리 정한 문턱치를 넘을 경우 1을, 그렇지 않을 경우 0을 부여한다. 이 경우, 일반적으로 움직임이 있는 부분에서는 일반적으로 높은 값의 DFD를 갖고, 움직임이 없거나 배경인 경우 낮은 값의 DFD를 갖는다. 만일 카메라의 이동이 있을 경우 이 또한 에지 부분에서 높은 값의 DFD를 갖기 쉽다. 다음으로 각 블록의 DFD에 대해 DCT를 취하여 각각 DC, DC+AC(1,2), AC(ALL)에 대해 이치화를 시도한다. 여기서 AC(1,2)는  $AC_1 + AC_2$ 를 의미하며, AC(ALL)은 모든 AC 계수를 합한 값을 의미한다.

그림 1은 이렇게 처리된 이치화 결과에 대해 1로 처리된 블록의 위치가 일치할 경우의 블록 개수를 각 프레임별로 정리한 것이다. 위의 결과에서 알 수 있듯이 특히 DC 및 DC+AC(1,2)의 결과는 DFD의 분포에 약 83% 정도 근접한다. 이로부터 알 수 있는 점은 압축된 비디오 신호의 움직임 블록을 구하고자 할 때, 근사적으로 DFD 대신 DC 성분을 이용할 수 있음을 암시한다. 본 논문에서는 이 결과를 바탕으로 복원된 신호의 DFD 대신에 DC 계수만을 이용하여 거의 동일한 결과를 얻고자 한다.

### III. 2-모드 선택기반의 객체추출기법

압축 영상에서 정지된 배경, 움직임이 적은 비디오에서

DCT계수를 이용한 움직임 영역 추출은 매우 효과적이다. 이는 정지된 배경에서 움직임이 발생될 경우 움직임이 발생된 영역의 프레임간 차가 크고, 이 때 발생하는 예측 오차에 대해 DCT를 행하여 얻어지는 DCT 계수는 DC와 AC계수 공히 큰 값을 가질 확률이 높다. 따라서, 이러한 성질을 이용하면 움직임 객체 영역을 얻을 수 있다.

반면에, 이러한 DCT 계수 기반의 움직임 추출 기법은 카메라의 움직임이 존재하거나 배경이 움직이는 경우 비효과적이다. 이는 카메라 또는 배경의 움직임으로 인하여 배경 부분에서 큰 값의 프레임차가 얻어지기 때문이다. 따라서, 객체와 배경영역 모두에서 큰 값의 DCT 계수가 얻어지기 때문에 이 경우 DCT 계수를 이용할 수 없다. 이러한 경우에는 움직임 벡터를 이용하는 것이 효과적이다.

따라서, 본 논문에서는 전역움직임 또는 배경의 움직임 유무에 따라 움직임 객체 추출을 위하여 DCT 계수와 움직임 벡터 중 선택적으로 이용하여 움직임 객체를 추출하는 방법을 제시하고자 한다.

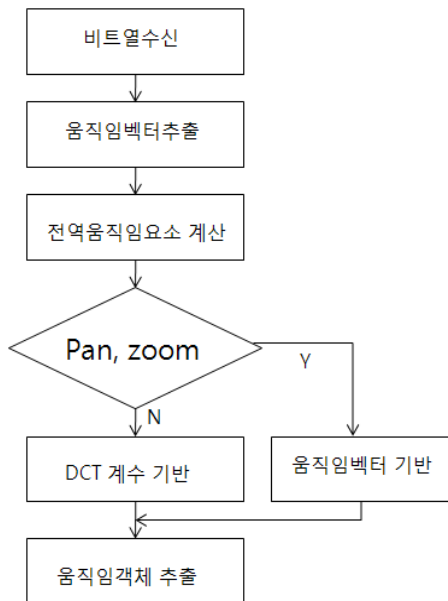


그림 2. 제안된 알고리즘의 블록도  
Fig. 2. Block diagram of the proposed algorithm

그림 2는 제안된 알고리즘에 대한 전체적인 흐름도를 나타낸 것이다. 먼저, 비트열을 수신하거나 입력되면, 압축데이터로부터 움직임 벡터를 추출하여 전역움직임 요소를 계산한다. 다음으로 팬 혹은 줌 요소가 있는지 조사하고, 전역

움직임 존재 여부에 따라 움직임 객체 추출을 DCT 계수를 기반으로 할 것인지, 움직임 벡터를 기반으로 할 것인지 판단하고, 그에 따른 처리를 행하여 움직임 객체를 추출한다. 각 단계별 처리는 다음과 같다.

### 3.1 전역 움직임의 추출

일반적으로 카메라 시스템은 그림 1과 같이 모델링할 수 있다 [1].

먼저 정지된 배경에서 카메라의 영향을 생각해 보면,  $u = (x, y, z)$ 와  $u' = (x', y', z')$ 를 각각 카메라 움직임 전후의 카메라 좌표라고 두고,  $U = (X, Y)$  및  $U' = (X', Y')$ 를 대응되는 영상 공간의 좌표라고 하자.

#### 가. 카메라 줌 (zoom)

줌은 카메라의 초점거리의 변화에 의해 유발되는데, 이것은 단지 카메라의 이동은 없기 때문에 카메라의 좌표는 변하지 않은 채 영상공간의 좌표만 변화된다. 즉,

$$U' = \frac{F'}{F} U \dots\dots\dots (4)$$

로 나타낼 수 있는데, 식 (4)에서  $F$ 와  $F'$ 는 줌이전과 이후의 카메라의 초점거리이다. 식 (4)에서,  $f_x = \frac{F'}{F}$ 를 줌요소(zoom factor)라 정의한다. 또한,  $U' - U$ 은  $U$ 에 비례해서 증가한다.

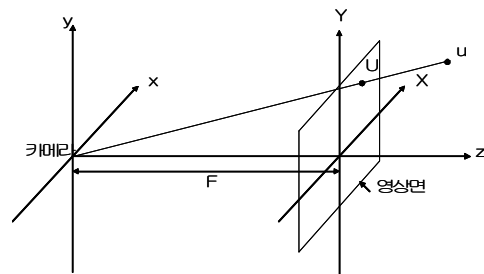


그림 3. 카메라 모델  
Fig. 3. Camera model

- (x,y,z) : 카메라 좌표
- (X,Y) : 영상면 좌표
- u : 대상체의 카메라 좌표상의 위치
- U : 대상체의 좌표상의 위치
- F: 초점거리

나. 카메라 팬 (pan)

팬은 영상면에 평행한 축에 대한 카메라의 회전에 의해 유발된다.  $x$ -축에 대한 카메라의 회전은 수직 팬을 유발한다 ( $\theta_x$ 각도만큼). 이것은 동일한 축에 대해 완전한 장면을  $-\theta_x$ 만큼 회전하는 효과와 동일하다. 작은  $\theta_x$ 에 대하여

$$X' = X, Y' = F\theta_x \dots\dots\dots (5)$$

즉, 전체 장면이  $F\theta_x$ 만큼 수직으로 이동된다. 따라서, 식(5)는

$$U' = U + p \dots\dots\dots (6)$$

로 나타낼 수 있고, 식 (6)의  $p = (p_x, p_y)$ 를 팬 벡터라고 부른다.

카메라 줌과 팬을 함께 조합하면,

$$U' = f_x U + p \dots\dots\dots (7)$$

로 나타낼 수 있다.

현재 프레임에서  $U'_i, i = 1, 2, \dots, N$ 의 각 위치에서 변이 벡터  $d_i$ 를 생각해보자. 이전 프레임에서  $U' = f'_z U'_i + p'$ 이며, 따라서,

$$\hat{d}_i = \hat{U}_i - U'_i = (f'_z - 1)U'_i + p' \dots\dots\dots (8)$$

이다. 또한

$$E(f'_z, p') = \sum_i \|\hat{d}_i - d_i\|^2 = \sum_i \|\hat{U}_i - U_i\|^2 \dots\dots\dots (9)$$

식 (9)에서  $U_i = U'_i + d_i$ 는 이전 프레임에서  $U'_i$ 의 실제 위치를 나타낸다. 따라서, 이제 추정은 아래의 최소화문제에 귀결된다.

$$\min E(f'_z, p') \dots\dots\dots (10)$$

미분 후 그 결과가 영이 되도록 두면, 식 (11)과 같은 결과가 얻어진다. 즉,

$$f'_z = \frac{\sum_i \langle U_i, U'_i \rangle - \frac{1}{N} \langle \sum_i U_i, \sum_i U'_i \rangle}{\sum_i \langle U'_i, U'_i \rangle - \frac{1}{N} \langle \sum_i U'_i, \sum_i U'_i \rangle} \dots\dots\dots (11)$$

$$p' = \frac{1}{N} (\sum_i U_i - f'_z \sum_i U'_i) \dots\dots\dots (12)$$

식 (11)과 식(12)를 바탕으로 전역움직임을 검출한다. 즉, 줌 요소와 팬 요소를 구하여 줌요소의 경우 1이 아닌 경우, 그리고, 팬요소의 경우 영이 아니면 전역움직임이 존재하는 것으로 판단한다.

3.2 2-모드 선택 기반의 움직임 객체 추출

비디오 데이터 압축에서 움직임 벡터를 얻기 위해 영상을 일정한 크기, 예를 들어  $16 \times 16$ 로 나누고 각 매크로 블록에 대해 움직임 추정을 하여 이전 프레임에서 가장 정합이 잘 되는 위치를 찾는다. 이 때 처리하고자 하는 블록을 기준블록이라고 하고, 찾고자 하는 대상의 범위를 탐색 범위라고 한다. 단일 기준블록이 움직임 객체 부분이 아닌 배경이라고 할 때, 탐색 범위 내에서 얻어지는 최적 매칭 블록과의 차를 구하면 예측오차는 적은 값을 가짐과 동시에, 각 화소별 오차 역시 변동폭이 적다. 왜냐하면, 카메라의 움직임 등 전역 움직임이 없다면, 밝기 변화 및 카메라 노이즈 등만이 존재하기 때문이다.

가. 전역움직임이 없을 경우

전역 움직임이 없을 경우 객체블록의 추출은 DC계수에 의해 이루어진다. 전역 움직임의 유무는 팬 및 줌 요소를 기준으로 판단하는데, 본 논문에서는 전역 움직임으로서 팬 (pan)에 의한 전역 움직임 성분만을 대상으로 한다. 즉, 다음 조건을 만족하면 전역움직임이 없는 것으로 간주된다.

$$|p'| < \alpha \dots\dots\dots (13)$$

여기서  $\alpha$ 는 사전에 설정된 임계치이며, 전형적으로 영(0)의 값을 갖는다. 이 경우 DCT 계수를 기반으로 움직임 객체를 추출한다.

임의의  $i$  번째 블록에 대한 움직임 객체 여부는 다음 식에 의해 판단된다.

$$block_i = \begin{cases} 1, & \text{if } C_i > T \\ 0, & \text{otherwise} \end{cases} \dots\dots\dots (14)$$

식 (14)에서  $C_i$ 번째 블록의 DC 계수값이며, T는 사전에 설정된 문턱치이다.

나. 전역움직임이 존재할 경우

만일 식 (13)의 조건을 만족하지 않을 경우 전역움직임이 존재하는 것으로 판정된다. 전역 움직임이 존재하는

경우, 먼저 전역움직임을 제거하고, 이를 바탕으로 움직임 객체를 추출한다. 이때, 움직임 객체의 대표 벡터  $V_p$ 는 다음과 같이 구해진다.

$$V_p = \operatorname{argmax}_{x,y} \sum_{|p,q| < R} (h(x+p,y+q)) \dots\dots\dots (15)$$

식 (15)에서  $h(a,b)$ 는 움직임 벡터에 대한 2-D 히스토그램을 구하고, 히스토그램 성분 중 벡터  $(a,b)$ 를 갖는 움직임 벡터의 누적을 나타내며,  $R$ 은 사전에 설정된 크기를 갖는다. 위의 계산을 통해 가장 높은 누적 값을 갖는 벡터를 객체의 움직임 벡터로 간주한다.

#### IV. 모의실험

본 논문에서 제시된 기법의 성능을 평가하기 위해 비디오 시퀀스들에 대해 실험을 행하였다. 본 실험에서는 블록 움직임 벡터를 얻기 위해 전역 탐색 기반의 블록 매칭 알고리즘(full search block matching algorithm)을 이용하여 블록 벡터를 얻었으며, 이때 사용한 블록의 크기는  $16 \times 16$ 이다. 또한 실험에 사용된 영상은 'hall monitor'를 포함한 3개의 영상열이다 (그림 4(a), 5(a), 6(a)).

그림 4에서 살펴보면, 그림 4(b)는 DCT 계수를 이용한 결과이며, 반면에, 그림 4(c)는 움직임벡터를 이용한 것이다. 그림에서 볼 수 있듯이 전역움직임이 없을 때, 움직임 벡터를 기반으로 한 추출 결과는 좋지 않다는 것을 알 수 있다. 특히, 배경 부분에서 오정합으로 인해 영이 아닌 크기의 벡터를 갖는 것으로 판정된 블록을 움직임 객체 블록으로 분류하는 잘못된 결과를 얻는다. 반면에 DCT 계수를 기반으로 할 경우 거의 정확하게 움직임 객체를 추출할 수 있음을 볼 수 있다. 그림 5의 경우에는 카메라의 이동으로 전역 움직임이 발생한 경우이다. 본 실험 영상의 경우 오른쪽으로 2 화소 가량 팬 움직임이 발생한 상태이다. 이 경우 그림 5(c)의 경우 DCT 계수를 이용한 경우이며, 그림 5(b)의 경우 움직임 벡터 기반으로 추출한 결과이다. 그림에서 알 수 있듯이 전역 움직임이 발생한 경우 DCT 계수보다는 움직임 벡터를 기반으로 추출하는 것이 효과적임을 알 수 있다.

또한, 그림 6의 경우도 전역 움직임이 존재하는 경우인데(pan 요소 2), 그림 6(c)의 경우 DCT 계수를 기반으로 처리한 결과인데, 전역 움직임 때문에 배경부분이 움직임 객체로 잘못 검출됨을 알 수 있다. 반면에, 그림 6(b)의 경우 전역움직임이 존재하기 때문에 움직임 벡터를 기반으로

처리한 것이다. 이 경우 움직임 객체가 비교적 정확하게 추출된 것을 볼 수 있다.



그림4 (a) Hall monitor 영상  
Fig. 4 (a) Hall monitor image



그림 4(b) DCT 계수를 이용한 객체검출  
Fig. 4(b) Object extraction based on DCT coefficients



그림 4(c) 움직임벡터를 이용한 객체검출  
Fig. 4(c) Object extraction based on motion vectors



그림 5(a) Coastguard 영상  
Fig. 5 (a) Coastguard image

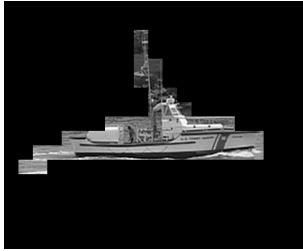


그림 5(b) 움직임벡터를 이용한 객체검출  
Fig. 5(b) Object extraction based on motion vectors



그림 6(c) DCT 계수를 이용한 객체검출  
Fig. 6(c) Object extraction based on DCT coefficients

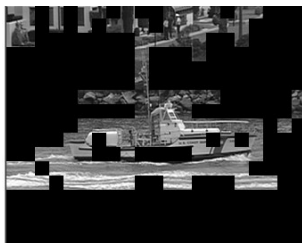


그림 5(c) DCT 계수를 이용한 객체검출  
Fig. 5(c) Object extraction based on DCT coefficients



그림 6(a) foreman 영상  
Fig. 6(a) Foreman image



그림 6(b) 움직임벡터를 이용한 객체검출  
Fig. 6(b) Object extraction based on motion vectors

## V. 결 론

본 논문에서는 움직임 객체 추출을 위한 새로운 기법을 제안하였다. 움직임 객체 추출 기법은 내용 기반 검색분야를 포함하여 다양하게 적용될 수 있다. 제안된 기법은 움직임 객체 추출시 움직임 벡터와 DCT 계수 중 적응적으로 적용되는 2-모드 선택 기법이 이용된다. 또한, DCT계수는 완전한 복호화를 필요로 하지 않으며, DCT 영역에서 계수값들이 직접적으로 사용됨으로써 계산적 복잡도를 낮출 수 있다. 실험결과에서 확인할 수 있듯이, 전역움직임이 있을 경우에는 움직임 벡터를 기반으로, 그리고 전역움직임이 없을 경우 예측오차를 기반으로 하는 DCT 계수를 바탕으로 객체를 추출하는 것이 효과적임을 확인할 수 있다.

## 참고문헌

- [1] F. Pereria, T Ebranhimi, The MPEG-4 Book, Prentice-Hall PTR, Englewood Cliffs, NJ, 2003.
- [2] L. Favalli, A. Mecocci, and F. Moschetti, "Object tracking for retrieval applications in MPEG-2," IEEE Trans. Circuits Syst. Video Technology," vol. 10, pp. 427-432, Apr. 2000.
- [3] D. Zhong and S. F. Chang, "An Integrated approach for content-based video object segmentation and retrieval," IEEE Trans. Circuits System Video Technology, vol. 9, pp. 1259-1268, Dec. 1999.
- [4] R. V. Baru, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: a compressed domain approach," IEEE Trans.

- Circuits System Video Technology, vol. 14, pp. 462-474, April 2004.
- [5] 김영주, 배경 분리 기반의 실시간 객체 추적을 위한 개선된 적응적 배경 혼합 모델, 한국컴퓨터정보학회 논문지 제10권 제6호, 2005.
  - [6] N. Paragios and G. Tziritas, "Adaptive detection and localization of moving objects in image sequences," Signal Processing: Image Comm., vol. 14, pp. 277-296, Feb. 1999.
  - [7] I. Kompatsiaris, G. Mantzaras, and M. G. Strintzis, "Spatiotemporal segmentation and tracking of objects in color image sequence," Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2000), vol. 5, Geneva, Switzerland, May 2000, pp. 29-32
  - [8] C. Stiller, "Motion-estimation for coding of moving video at 8 kbit/s with Gibbs modeled vectorfield smothing," SPIE vol. 1360 Visual Comm. and Image Proc. '90., pp. 468 - 476, 1990.
  - [9] Derin, Cole: "Segmentation of textures images using Gibbs Random Fields," Computer Vision, Graphics and Image Processing 35, pp. 72-98, 1986.
  - [10] P. Salembier and F. Marques, "region-based representations of image and video: segmentaiton tools for multimedia services," IEEE Trans. Circuits and Video Technol. vol 9, pp. 1147-1169, Dec. 1999.

### 저 자 소 개



**김동욱**

1987년 성균관대학교 전자공학과 졸업  
 1992년 중앙대학교 대학원 전자 공학  
 과 졸업(공학석사)  
 1996년 중앙대학교 대학원 전자공학과  
 졸업(공학박사)  
 현재 전주대학교 전기전자정보통신공학  
 부 부교수 재직  
 <관심분야> 신호처리, 영상처리,  
 네트워크