

論文2002-39SP-5-3

# 이산여현변환 공간에서의 영상 축소를 위한 움직임 벡터 재추정

## (A Motion Vector Re-Estimation Algorithm for Image Downscaling in Discrete Cosine Transform Domain)

金雄熙\*, 吳承均\*\*, 朴玄旭\*\*

(Woong-Hee Kim, Seung-Kyun Oh, and HyunWook Park)

### 요 약

이 논문에서는 이산여현변환 공간에서 영상의 크기를 줄이는 변환부호화 과정에서의 움직임 벡터 (motion vector) 재추정을 위한 방법을 제안하였다. 제안된 방법에서는 영상 사이의 SAD (Sum of Absolute Difference)와 매크로 블록의 에지 정보를 이용하여 커널 함수를 정의하고, 그것을 이용해서 축소된 영상에서의 새로운 움직임 벡터를 재추정하였다. 변환부호화기에서는 속도의 효율성을 위해서 변환부호화기의 입력의 화면 간 (inter-frame) 정보에서 얻을 수 있는 기존의 움직임 벡터를 이용하는 방법을 택하고 있는데 본 논문에서도 속도의 효율성을 위해서 기존의 움직임 벡터들을 이용하는 방법을 선택하였다. 제안한 방법을 이용해서 실영상에 대해서 실험을 해 본 결과 연산량을 고려한 PSNR (Peak Signal to Noise Ratio)의 측면에서 우수한 성능을 보여주는 것을 확인할 수 있었다.

### Abstract

A motion vector re-estimation algorithm for image downscaling in discrete cosine transform domain is presented. Kernel functions are defined using SAD (Sum of Absolute Difference) and edge information of a macroblock. The proposed method uses these kernel functions to re-estimate a new motion vector of the downscaled image. The motion vectors from the incoming bitstream of transcoder are reused to reduce computation burden of the block-matching motion estimation, and we also reuse the given motion vectors. Several experiments in this paper show that the computation efficiency and the PSNR (Peak Signal to Noise Ratio) and better than the previous methods.

**Key Words**: 변환부호화 (Transcoding), 움직임 벡터 (Motion vector), 움직임 재추정 (Motion re-estimation), 이산여현변환 (Discrete Cosine Transform), 해상도 변환 (Image resizing)

\* 正會員, 韓國電子通信研究院 附設 國家保安技術研究所  
(National Security Research Institute, ETRI)

\*\* 正會員, 學生會員, 韓國科學技術院 電子電算學科 電氣 및 電子工學

(Department of Electrical Engineering and Computer Science, Division of Electrical Engineering, Korea Advanced Institute of Science and Technology)

接受日字:2002年2月25日, 수정완료일:2002年8月12日

### I. 서 론

최근에 화상 회의, 웹 카메라, VOD (Video On Demand)와 같은 많은 디지털 비디오 서비스들에 대한 관심이 증가하고 있다. 디지털 비디오를 저장하거나 전송하기 위해서는 디지털 비디오의 많은 자료량으로 인해서 디지털 비디오 압축이 요구된다. 디지털 비디오를 압축한 후에도 압축된 비트율 (bitrate)보다 더 낮은 비

트율이 필요한 경우가 발생하게 된다. 예를 들어서 전송하고자 하는 압축된 디지털 비디오의 비트율보다 전송 네트워크의 채널 대역 폭이 작은 경우, 압축된 디지털 비디오의 비트율 감소가 필요하게 된다.<sup>[1]</sup>

비트율을 줄이기 위한 많은 변환부호화 방법들이 제안되었다.<sup>[2~5]</sup> 영상의 크기를 줄이는 방법은 압축된 디지털 비디오의 비트율을 줄이는 한 방법이다. 영상의 크기를 줄여서 압축된 디지털 비디오의 비트율을 줄이기 위한 간단한 방법으로는 우선 압축된 디지털 비디오를 복호화를 거친 후, 공간 영역에서 영상의 크기를 줄이는 연산을 하고, 크기가 줄어든 영상을 다시 부호화를 하는 것이다. H.261<sup>[6]</sup>, H.263<sup>[7]</sup>, MPEG-1<sup>[8]</sup>, MPEG-2<sup>[9]</sup> 등과 같은 디지털 비디오 압축을 위한 국제 표준에서는 시간방향 중복성을 줄여서 높은 압축 효율을 얻기 위해 움직임 추정 및 보상 기법을 채택하여 사용하고 있다. 이러한 영상 압축 표준들에 적합한 부호화를 하기 위해서 우리는 축소된 영상의 움직임 벡터를 재추정해야 하며 지금까지 많은 공간 영역에서의 움직임 추정 및 보상 기법들이 연구되었다.<sup>[10]</sup> 움직임 추정 및 보상 기법은 비디오 부호화기의 전체 연산량의 적어도 60%를 차지한다.<sup>[10]</sup> 변환부호화기에서의 공간영역에서의 움직임 추정 및 보상 기법은 이러한 많은 연산량으로 인해서 실시간 응용에 적용하기에는 적합하지 않다. 이러한 이유로 인해서 변환부호화 과정에서는 변환부호화기의 입력으로 들어오는 이미 압축된 디지털 비디오로부터 얻어진 움직임 벡터들을 이용해서 축소된 영상을 부호화하기 위한 새로운 움직임 벡터를 찾고 이러한 연구들이 지금까지 수행되었다.<sup>[3,11,12]</sup>

전술한 국제 비디오 압축 표준에서는 공간 방향 중복성을 줄이기 위해서 이산여현변환 (Discrete Cosine Transform, DCT)을 사용한다. 복호화기와 부호화기가 순서대로 연결된 변환부호화기에서 복호화 과정에서 역이산여현변환이 필요하고, 부호화하는 과정에서는 이산여현변환이 필요하다. 이러한 일련의 과정은 변환부호화기의 연산의 측면에서 복잡성을 증가시킨다. 그러므로 연산량 측면에서 효과적인 변환부호화기를 구현하기 위해서 우리는 변환부호화 과정을 공간 영역에서 행한 것이 아니라 역이산여현변환과 이산여현변환이 필요 없는 변환 영역에서 행하였다.

본 논문에서 우리는 압축된 디지털 비디오 입력 영상 ( $N \times N$ )을 변환 영역에서 축소된 크기 ( $N/2 \times N/2$ )의 압축된 디지털 비디오로 바꾸는데 필요한 움직임 벡터

를 재추정하는 방법을 제안하였다. II장에서는 변환부호화기에서 움직임 벡터를 재추정하기 위한 기존의 연구들에 대해서 살펴보겠다. 그리고 III장에서는 커널 함수 (kernel function)를 비롯하여 본 논문에서 제안하는 움직임 벡터 재추정 방법에 대해서 기술하고, IV장에서는 제안된 방법을 이용해서 구현된 변환부호화기를 통한 실험 결과를 살펴보고, 마지막으로 V장에서는 결론을 언급하였다.

## II. 변환부호화기에서의 움직임 벡터 재추정

변환부호화는 이미 부호화가 된 디지털 비디오를 다른 표준의 형식으로 재부호화를 하거나 비트율을 줄여서 재부호화하는 것을 의미한다. 본 논문에서는 입력 비디오의 영상 크기를 줄여서 비트율을 줄이는 변환부호화 과정에서 필요한 움직임 벡터를 재추정하는 방법을 살펴보고자 한다.

변환부호화기를 포함하고 있는 비디오 시스템의 기본적인 구조는 그림 1과 같다. 입력 비디오는  $R_1$ 의 비트율로 부호화가 되어 있고, 변환부호화 과정을 거친 후  $R_2$ 의 비트율로 부호화가 된다. 마지막으로 최종 사용자가 변환부호화 과정을 거친 디지털 비디오를  $R_2$ 의 비트율로 복호화를 할 수 있다.<sup>[3,11]</sup>

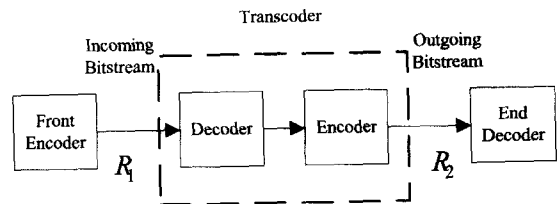


그림 1. 변환부호화기를 포함한 기본적인 비디오시스템  
Fig. 1. Conventional Basic video system including a transcoder.

우리가 관심을 가지고 있는 변환부호화기는 그림 2에서와 같이 복호화기와 부호화기가 연속적으로 연결되어 있고 그 사이에 영상의 크기를 줄이는 연산을 하는 부분이 삽입되어 있는 구조를 가지고 있다. 변환부호화기의 입력으로 들어오는 디지털 비디오는 먼저 높은 해상도의 복호화기 (higher resolution decoder)에서 복호화 과정을 거치면서 영상의 크기를 줄이는데 필요한 정보를 추출한 후, 영상의 크기를 이산여현변환 영

역에서 줄이고, 낮은 해상도의 부호화기 (lower resolution encoder)를 통해서 재부호화가 된다. 이 과정에서 재부호화에 필요한 새로운 움직임 벡터는 부호화가 된 영상의 화질에 중요한 영향을 미친다.

공간 영역에서 영상의 크기를 줄이는 것은 변환부호화기에서의 역이산여현변환, 이산여현변환과정을 거쳐야 하므로 이산여현변환 영역에서의 영상의 크기를 줄이는 방법보다 더 많은 연산량을 요구한다. 이산여현변환 영역에서 영상의 크기를 줄이는 방법과 그것을 이용해서 변환부호화기에 적용한 연구는 지금까지 많이 진행되어 왔다.[3, 5, 13, 14, 15] 본 논문에서는 지금까지의 이산여현변환 영역에서 영상크기를 줄이는 방법보다 PSNR의 측면과 연산량 측면에서 우수한 성능을 보여주고 있는 Park[16]이 제안한 방법을 사용하였다. 그림 3은 이산여현변환 영역에서 영상의 크기를 줄이는 연산을 하는 변환부호화기의 구조를 보여주고 있다.

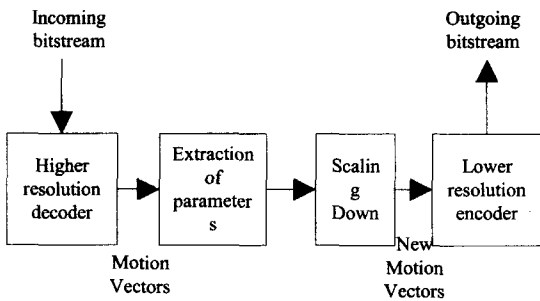


그림 2. 변환부호화기의 구조  
Fig. 2. Architecture of video transcoder.

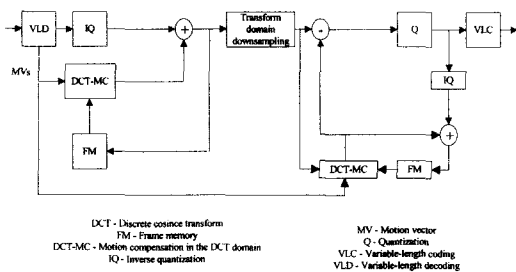


그림 3. 이산여현변환 영역에서 영상의 크기를 줄이는 전형적인 변환부호화기의 구조  
Fig. 3. Conventional transcoder with DCT-domain downsampling.

이산여현변환 영역에서 영상의 크기를 축소하므로 움직임 추정 및 보상도 이산여현변환 영역에서 이루어져야 한다. 이산여현변환 영역에서 움직임 추정 및 보

상을 하는 방법과 그것을 이용한 부호화기에 대한 연구가 되었다.[17~19] 이산여현변환 영역에서 움직임 추정 및 보상을 하기 위해서는 참조 영상 (reference frame)의 모든 부분에 위치한 블록의 이산여현변환 계수를 알고 있어야 한다. 현재 영상 (current frame)에서의 블록을 대상 블록 (target block)이라고 하고, 참조 영상에서의 블록을 후보 블록 (candidate block)이라고 부르며, 대상 블록은 정렬 블록 (aligned block)이고, 후보 블록은 그림 4에서 나타난 것과 같이 정렬 블록이나 비정렬 블록 (non-aligned block)일 수도 있다. 비정렬 블록의 이산여현변환 계수는 비정렬 블록의 주위에 있는 정렬 블록을 이용해서 계산할 수 있다.[19,20]

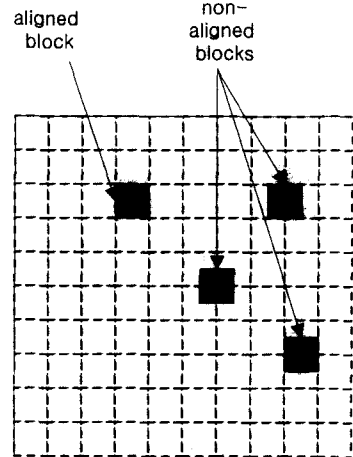


그림 4. 정렬 블록과 비정렬 블록  
Fig. 4. Aligned and non-aligned blocks.

그림 5는 비정렬 블록  $b_{na}$  와 그것의 주변에 있는 정렬 블록  $b_1, b_2, b_3, b_4$  를 나타내고 있고, 여기에서 비정렬 블록  $b_{na}$ 의 이산여현변환 계수는 주위의 정렬 블록  $b_1, b_2, b_3, b_4$ 의 이산여현변환 계수를 이용해서 식 (1)과 같이 구해진다.[20]

$$2DCT(b_{na}) = \sum_{i=1}^4 DCT(H_{a_i})DCT(b_i)DCT(H_{b_i}) \quad (1)$$

여기에서  $H_{a_i}$ 과  $H_{b_i}$ 는 windowing과 shifting 행렬을 의미하고,  $DCT(b_i)$ 는 블록  $b_i$ 의 이산여현변환 계수이다.

변환부호화기의 입력에서 얻어진 원래 영상의 움직임 벡터를 이용해서 변환부호화기내의 부호화기에서 필요한 새로운 움직임 벡터를 추정하는 방법들이 제안되었다.[3,5]

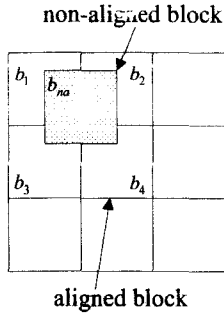


그림 5. 비정렬 블록과 이웃하는 정렬 블록과의 위치 관계

Fig. 5. Configuration of a non-aligned block and its neighboring aligned blocks.

변환부호화기의 입력에서 추출한 네 개의 움직임 벡터 중에서 중간 값에 해당하는 움직임 벡터를 선택하는 방법이 있다.  $MV_1, MV_2, MV_3, MV_4$ 를 네 개의 이웃한 블록의 움직임 벡터들의 집합이라고 하면 식 (2)을 이용하여 각각의 벡터와 나머지 벡터들 사이의 거리를 계산하고 식 (3)으로 중간 값에 해당하는 움직임 벡터  $MV_{med}$ 를 선택한다.<sup>[5]</sup>

$$d_i = \sum_{j=1, j \neq i}^4 \|MV_j - MV_i\| \quad (2)$$

$$MV_{med} = MV_k \text{ such that } d_k = \underset{i}{med}(d_i) \quad (3)$$

여기에서  $med()$ 는 median operator를 나타낸다.

인접한 네 개의 움직임 벡터들의 산술적 평균을 구해서 새로운 움직임 벡터를 구하는 방법이 있다. 새로운 움직임 벡터는 식 (4)로 구해질 수 있다.

$$MV_{avg} = \frac{1}{4} \sum_{i=1}^4 MV_i \quad (4)$$

이 방법은 하나의 움직임 벡터가 나머지 움직임 벡터와 그 크기의 차이가 클 경우 좋지 않은 결과를 낼 수 있다.<sup>[5]</sup> 이 방법을 확장한 방법으로 가중치 합계를 이용한 방법을 생각할 수도 있다.<sup>[12]</sup>

인접한 네 개의 움직임 벡터들 중에서 영상 축소 전에 가장 SAD (Sum of Absolute Difference)가 작은 매크로 블록의 움직임 벡터를 선택하거나 반대로 가장 큰 SAD를 가지고 있는 매크로 블록의 움직임 벡터를 선택하는 방법이 있다. 본 논문에서 계산한 SAD는 공간 영역에서 계산한 SAD가 아니라, 이산여현변환 공간에서 계산된 SAD이다.

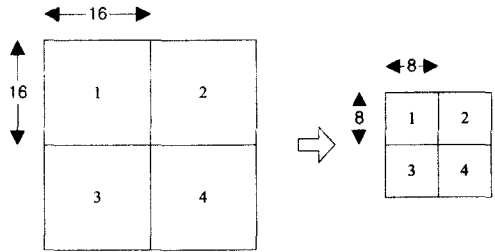


그림 6. 매크로 블록들의 크기 변화

Fig. 6. Resolution reduction of macroblocks.

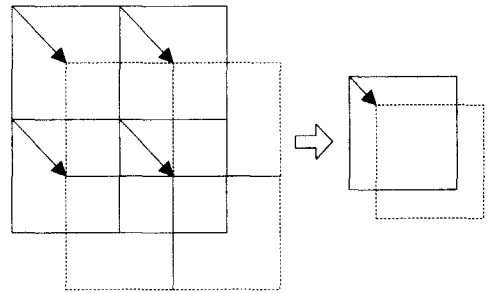


그림 7. 유사한 방향과 크기의 움직임 벡터를 가진 매크로 블록의 축소

Fig. 7. Resizing of macroblocks with similar direction and magnitude.

우리는 영상의 크기를 가로, 세로로 각각 1/2로 축소하는 변환부호화기에서는 그림 6에서 보이는 것과 같이 네 개의 인접한 매크로 블록이 한 개의 매크로 블록으로 바뀌게 된다.

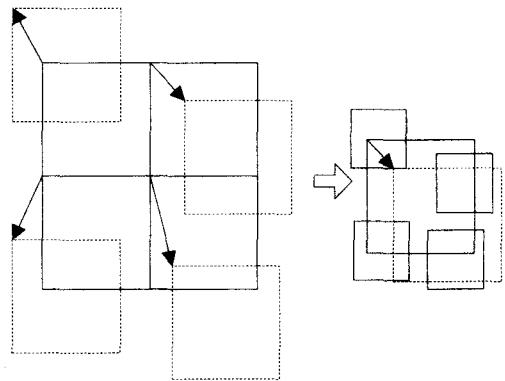


그림 8. 상이한 방향과 크기의 움직임 벡터를 가진 매크로 블록의 축소

Fig. 8. Resizing of macroblocks with different direction and magnitude.

그림 7에서와 같이 만약 네 개의 매크로 블록이 가지고 있는 움직임 벡터들의 방향과 크기가 같거나 유사하다면 새로운 움직임 벡터를 추정하는 것은 쉬운 일이다. 하지만 그림 8과 같이 네 개의 움직임 벡터들의 방향과 크기가 서로 유사하지 않을 경우에는 새로운 최적의 움직임 벡터를 추정하는 방법이 필요하게 된다.<sup>[12]</sup>

### III. 제안한 움직임 벡터 재추정 방법

변환부호화기의 입력으로 들어오는 화면 간 (inter-frame) 부호화 비트스트림에서 우리는 두 가지 정보를 얻을 수 있다. 하나는 움직임 벡터이고 다른 하나는 오류 신호의 이산여현변환 계수이다. 오류 신호의 이산여현변환 계수를 이용해서 우리는 쉽게 축소되기 전 영상의 매크로 블록의 SAD에 해당하는 값을 구할 수 있다. 제안된 방법은 축소하기 전 영상의 매크로 블록의 SAD 값과 움직임 벡터, 그리고 이산여현변환 영역에서 얻을 수 있는 에지 정보를 이용해서 새로운 움직임 벡터를 추정한다.

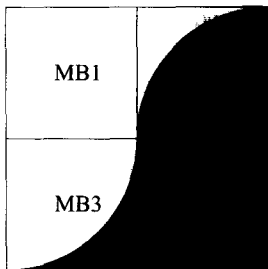


그림 9. 에지를 가진 매크로 블록들  
Fig. 9. Macroblocks with edges.

우리는 약한 에지를 가지는 매크로 블록보다 강한 에지를 가지고 있는 매크로 블록이 잘못된 움직임 벡터를 가질 경우 더 큰 오류를 발생할 수 있다고 가정하였다. 다시 말해서, 그림 9에서 MB2와 MB3가 잘못된 움직임 벡터를 가질 경우 더욱 큰 오류를 발생할 수 있다고 가정하였다.

그림 10에서 우리는 이러한 가정의 타당성을 확인할 수 있다. 그림 10은 움직임 벡터가 1 화소 (pixel) 잘못 되었을 경우 에지 정보와 SAD 사이의 관계를 나타내는 것으로, 에지 정보의 크기가 클수록 SAD의 크기가 커지는 것을 알 수 있다. 이 논문에서는 이산여현변환

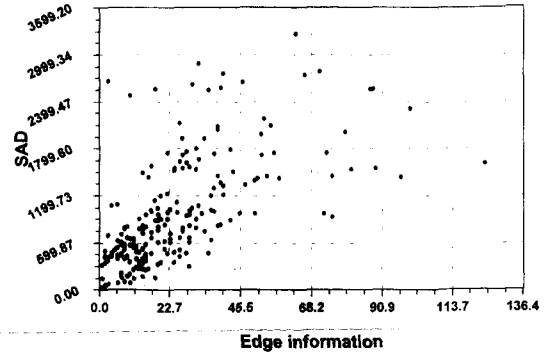


그림 10. 에지 정보와 SAD의 관계  
Fig. 10. Relationship between edge information and SAD.

공간에서 가로, 세로 방향으로의 높은 주파수 성분에 에지 정보라 보고, 각 블록에서 높은 주파수 성분에 해당하는 이산여현변환 계수 4개의 합을 각 방향의 에지 정보라 정의하였다. 이를 식으로 표현하면,  $i$ 번째 매크로 블록의 에지 정보는 다음과 같이 나타낸다.

$$E_x = \sum_{k=1}^4 \sum_{j=1}^7 |B_{i,j}(k,0)| \tag{5}$$

$$E_y = \sum_{k=1}^4 \sum_{j=1}^7 |B_{i,j}(0,k)| \tag{6}$$

여기에서  $B_{i,j}(x,y)$ 는  $i$ 번째 매크로 블록의  $k$ 번째 블록을 구성하는  $8 \times 8$  이산여현변환 계수를 의미하며  $E_x$ 와  $E_y$ 는 각각  $x$  방향과  $y$  방향의 에지 정보를 대표한다.

우리는 변환부호화기의 입력으로 들어오는 압축된 디지털 비디오에서 계산한 SAD를 새로운 움직임 벡터

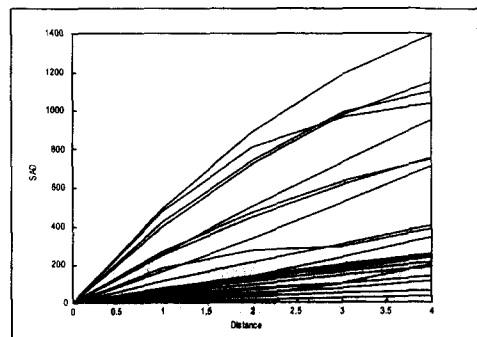


그림 11. 움직임 벡터를 중심으로 한 SAD의 변화  
Fig. 11. Profile of SAD from the motion vector edge information

를 추정하는데 사용하였다. SAD가 작다는 것은 이전 단계의 부호화 과정에서 움직임 벡터를 이용한 추정으로 인한 오류가 작다는 것을 알 수 있다. 이것은 이전 단계의 부호화 과정에서 움직임 벡터가 잘 예측되었다는 것을 의미한다.

그림 11는 움직임 벡터를 중심으로 해서 벡터 값을 증감시키면서 SAD의 변화량을 나타낸다. 우리는 움직임 벡터,  $MV=(mv_x, mv_y)$  를 중심으로 SAD의 변화량을 나타내고 있는 이 곡선을 식 (7)과 식 (8)와 같이 모델링하고 그것을 커널 함수라고 정의하였다.

$$f_x(m_x) = g(E_x) \left| m_x - \frac{mv_x}{2} \right|^{\frac{1}{2}} + SAD_i \quad (7)$$

$$f_y(m_y) = g(E_y) \left| m_y - \frac{mv_y}{2} \right|^{\frac{1}{2}} + SAD_i \quad (8)$$

여기에서  $(mv_x, mv_y)$ 는 입력 영상에서  $i$  번째 매크로 블록의 움직임 벡터이며  $SAD_i$ 는  $i$  번째 매크로 블록의 움직임 보상된 오류 값이다.

커널 함수의 폭은 변환부호화기에서 복호화된 원 영상의 매크로 블록의 에지 정보(식 (5), 식 (6))에 의해서 결정된다. 만약 원 영상의 매크로 블록이 약한 에지를 가지고 있을 경우 커널 함수의 폭은 넓어지고 강한 에지를 가지고 있을 경우에는 폭이 좁아지도록 커널 함수를 정의하였다. 그림 12는 이러한 특성을 보여주는 커널 함수의 예를 보여주고 있다. 그림 13에서 알 수 있듯이 원 영상의 SAD가 클 경우 커널 함수의 높이는 전체적으로 SAD가 작을 경우보다 높아지는 것을 알

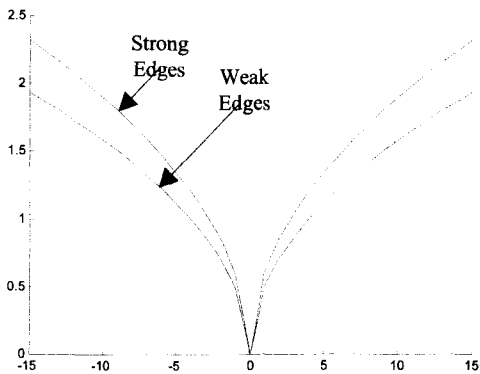


그림 12. 에지 정보에 따른 커널 함수 폭의 변화  
Fig. 12. Kernel functions with different edge information.

수 있다. 그림 12와 그림 13에서 가로축은  $m_x, m_y$ , 즉 새롭게 추정할 움직임 벡터의 범위이다.

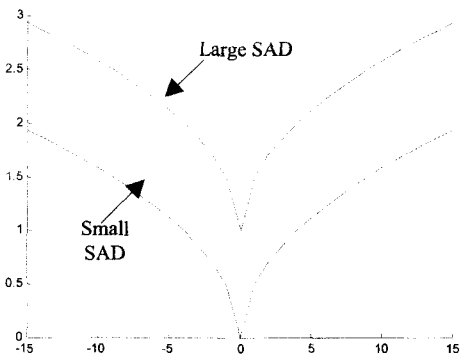


그림 13. SAD에 따른 커널 함수 높이의 변화  
Fig. 13. Kernel functions with different SAD.

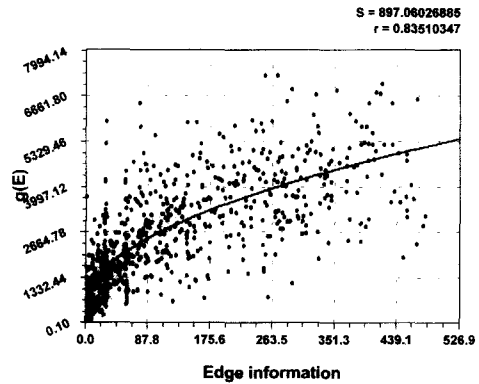


그림 14. 에지 정보와  $g(E)$ 의 상관 관계 및 곡선 적합을 수행한 곡선  
Fig. 14. Correlation between  $g(E)$  and edge information and its fitted curve

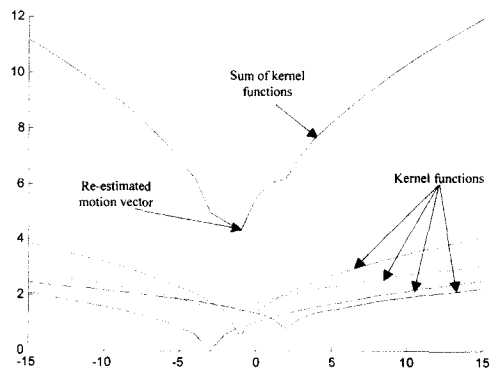


그림 15. 커널 함수와 새로운 움직임 벡터의 선택  
Fig. 15. Kernel functions and re-estimation of a new motion vector.

식 (7)과 식 (8)에서 에지 종속 함수  $g(E)$ 를 구하기 위해서 실제 영상에서 SAD를 구하여 곡선 적합을 한 후, 에지 정보와  $g(E)$ 의 분포를 나타내면 그림 14와 같다. CurveExpert™ Version 1.37을 이용하여 그림 14가 나타내는 자료의 정보를 식 (9)와 같은 형태의 곡선 적합을 해서  $a, b$ 를 결정하였다.

$$g(E) = a\sqrt{E} + b \quad (9)$$

새로운 움직임 벡터는 식 (11)와 식 (12)를 이용해서 구할 수 있다. 각 성분의 커널 함수를 더해서 가장 작은 값을 가지는 인자를 새로운 움직임 벡터의 성분으로 결정하였고 그림 15는 이것을 설명해 주고 있다.

$$MV_{new} = (mv_x, mv_y) \quad (10)$$

$$mv_x = \arg \min_{m_x} \sum_{i=1}^4 f_i(m_x) \quad (11)$$

$$mv_y = \arg \min_{m_y} \sum_{i=1}^4 f_i(m_y) \quad (12)$$

#### IV. 실험 결과

H.263<sup>[7]</sup>으로 부호화가 된 몇 가지의 디지털 비디오들을 이용해서 본 논문에서 제안한 움직임 벡터 재추정 방법을 이용한 변환부호화를 실험해 보았다. 본 실험에서 우리는 공개 소프트웨어 (public domain software) H.263<sup>[7,21]</sup>을 수정하여서 제안된 움직임 벡터 재추정 방법을 포함하고 있는 변환부호화기를 구현하였다. 본 실험에서, CIF (352×288) 크기의 영상을 QCIF (176×144) 크기로 변환하였고 첫 번째 영상은 화면 내 부호화 (intra-frame coding)을 하였고 그 다음 49개의 영상은 화면 간 부호화를 하였다.

제안된 방법과 비교를 하기 위해서 움직임 벡터를 재추정하는 다른 방법들을 구현하였다. 우선 가장 간단한 "Average" 방법으로 인접한 4개의 매크로 블록의 움직임 벡터들의 평균을 새로운 움직임 벡터로 추정하는 것을 구현하였다. 변환부호화기에서 원 영상의 매크로 블록은 영상의 크기를 가로, 세로로 1/2했을 경우 블록에 해당하게 되고 원 영상에서 매크로 블록의 SAD는 축소된 영상에서는 매크로 블록의 1/4에 해당하는 부분의 SAD가 되는 것을 그림 6에서 확인할 수 있었다. 그림 16에서 보이듯이 축소된 영상에서 하나의 매크로 블록은 축소 전 영상의 4개의 매크로 블록으로

구성되는데 축소 전 매크로 블록의 SAD를 이용해서 새로운 움직임 벡터를 구하는 방법을 두 가지 구현하였다. 하나는 "SAD\_MAX" 방법으로 원 영상의 매크로 블록들 중에서 가장 SAD가 큰 매크로 블록의 움직임 벡터를 축소된 매크로 블록의 움직임 벡터로 선택하는 것이고, 다른 하나는 "SAD\_MIN"으로 원 영상의 매크로 블록들 중에서 가장 SAD가 작은 매크로 블록의 움직임 벡터를 새로운 움직임 벡터로 선택하는 것이다.

그리고 "Optimal Selection" 방법을 구현하였는데 전술한 것과 같이 원 영상에서 매크로 블록의 SAD는 축소된 영상에서 매크로 블록의 1/4에 해당하는 정보만을 주는데 별도의 연산을 통해서 축소 전 영상의 매크로 블록들이 가지고 있는 움직임 벡터들에 해당하는 16×16 SAD를 구해서 가장 작은 SAD를 가지는 원 영상의 매크로 블록의 움직임 벡터를 새로운 움직임 벡터로 선택하였고, 그림 17은 그것을 설명해 주고 있다. 또한 제안된 방법을 "Full Search" 방법과도 비교를 하였다. 실험 환경은 CPU가 Pentium® III 1 GHz이고 메인 메모리가 512MB인 PC이고 운영 체제는 Microsoft™ Windows™ 2000이다.

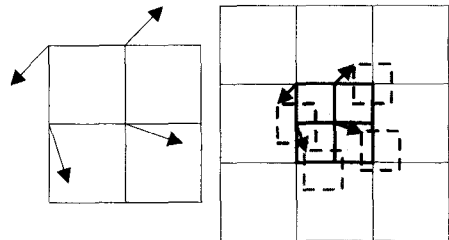


그림 16. 축소된 매크로 블록들과 움직임 벡터들  
Fig. 16. Resized macroblocks and motion vectors.

표 1. PSNR 비교

Table 1. Comparison of averaged PSNR.

	carphone	paris	tempe	news	foreman
Full Search	33.22	28.80	27.34	32.59	30.72
Optimal Selection	32.66	28.31	27.08	32.58	29.78
Average	31.48	27.72	26.56	32.07	29.08
SAD_MAX	31.47	27.62	26.54	32.12	29.01
SAD_MIN	31.44	27.71	26.67	32.14	29.08
Proposed Algorithm	32.18	28.16	27.01	32.51	29.24

그림 18과 그림 19는 제안된 방법과 실험을 하기 위해서 구현한 다른 방법들의 PSNR (peak signal to

noise ratio)을 보여주고 있다. 제안된 방법이 Optimal Selection과 Full Search보다는 PSNR 측면에서 좋지 않은 성능을 보여주고 있으나 Optimal Selection은 별도의 연산을 필요로 하기 때문에 고속의 변환부호화기에는 적합하지 못하다. 이 두 가지 방법을 제외하고는 제안된 방법이 전체적으로 좋은 PSNR을 보여주고 있고, 표 1은 그 결과를 보여주고 있다.

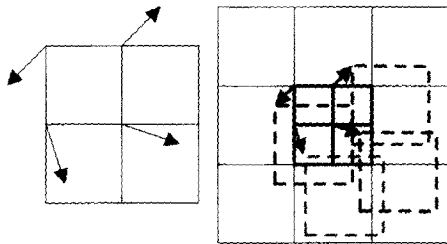


그림 17. Optimal Selection 방법  
Fig. 17. Optimal Selection method.

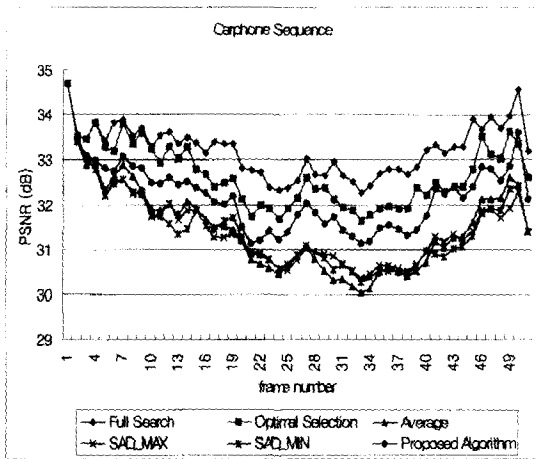


그림 18. 200kbps의 영상을 100kbps로 변환부호화한 carphone 영상을 이용한 다른 영상과의 PSNR비교

Fig. 18. Quality comparison of different motion vector re-estimation algorithms("carphone" test sequence). Test sequence is encoded at 200kbps and then transcoded at 100kbps.

그림 20은 제안된 방법보다 좋은 PSNR을 보여주는 방법들과 제안된 방법과의 연산량을 비교한 것을 보여주고 있다. 연산량은 Compuware®의 NUMEGA™을

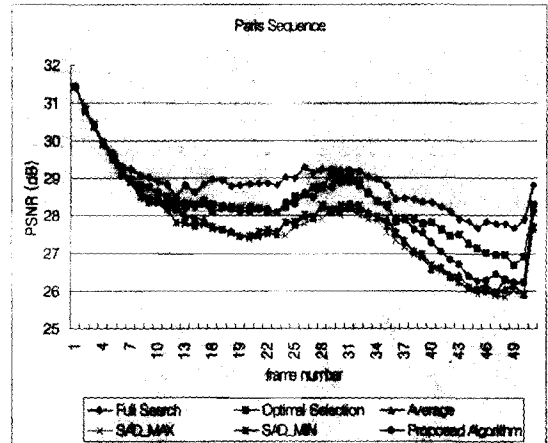


그림 19. 200kbps의 영상을 100kbps로 변환부호화한 paris 영상을 이용한 다른 영상과의 PSNR비교

Fig. 19. Quality comparison of different motion vector re-estimation algorithms ("paris" test sequence). Test sequence is encoded at 200kbps and the transcoded at 100kbps.

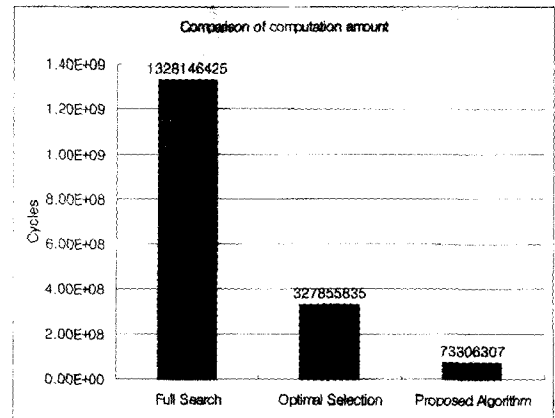


그림 20. 연산량 비교

Fig. 20. Comparison of computational complexity.

사용하여서 측정하였다. 그림 21에서 알 수 있듯이 제안된 방법이 Optimal Selection의 22.34%의 연산량을 보여주고 있다. 그리고 그림 22은 실제로 변환부호화 과정을 거친 비트스트림을 복호화한 영상으로써 50번째 영상을 나타내고 있다. 실험 결과에서 알 수 있듯이 제안된 방법은 영상의 화질을 크게 손상시키지 않으면서 연산량 측면에서 우수한 성능을 보여주고 있다.



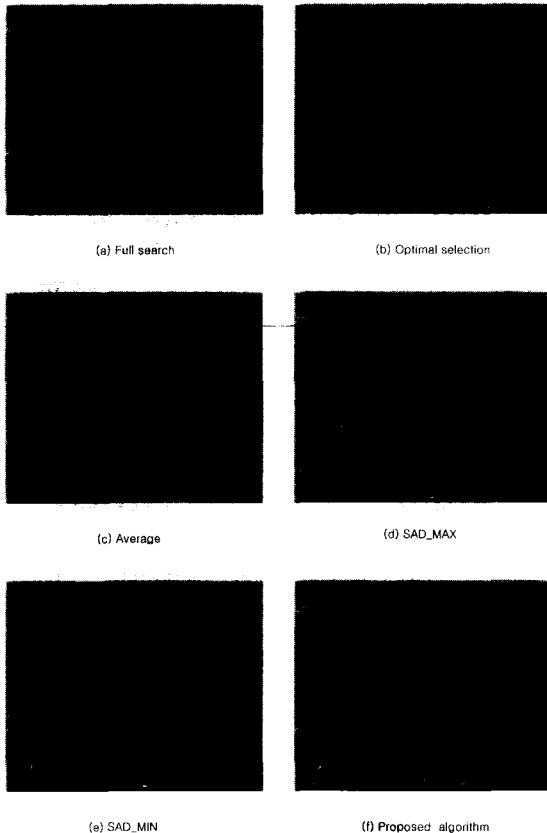


그림 21. 변환부호화된 50번째 tempete 테스트 영상  
Fig. 21. Transcoded image of tempete sequence.

## V. 결 론

이 논문에서는 변환 영역에서 영상의 크기를 줄이는 변환부호화기에서 움직임벡터를 효과적으로 재추정하는 방법을 제안하였다. 제안한 방법은 입력된 디지털 비디오 영상의 크기를 1/4로 줄이는데 사용되었다.

새로운 움직임 벡터를 추정하기 위해서 본 논문에서는 복호화된 매크로블록의 SAD와 에지 정보를 이용하였다. 본 연구에서 제안한 방법을 이용한 실험결과에서 기존의 방법보다 연산량을 고려한 PSNR의 측면에서 우수한 성능을 보여주는 것을 확인할 수 있었다.

## 참 고 문 헌

[1] N. Chaddha, "A software only scalable video delivery system for multimedia applications over heterogeneous networks", *Proc. Int. Conf.*

*Image Processing*, Vol. 3, pp. 404~407, Washington, DC, Oct. 1995.

- [2] H. Sun, W. Kwok and J.W. Zdepski, "Architectures for MPEG compressed bitstream scaling", *IEEE Trans. on Circuit Syst. Video Technol.*, Vol. 6, No. 2, pp. 19~4199, Apr. 1996.
- [3] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding", *IEEE Trans. on Consumer Electronics*, Vol. 44, No.1, pp. 88~98, Feb. 1998.
- [4] P.A.A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit stream", *IEEE Trans. on Circuits Syst. Video Technol.*, Vol. 8, No. 8, pp. 95~4967, Dec. 1998.
- [5] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolution and different encoding formats", *IEEE Trans. on Multimedia*, Vol. 2, No. 2, pp. 101~110, June 2000.
- [6] ITU-T Recommendation H.261, *Video codec for audiovisual services at p×64 kbits*, Mar. 1993.
- [7] ITU-T Recommendation H.263, *Video coding for low bit rate communication*, Feb. 1998.
- [8] CCITT Recommendation MPEG-1, *Coding of moving pictures associated audio for digital storage media at up to about 1.5 Mbit/s*, ISO/IEC 11172, Geneva, Switzerland, 1993.
- [9] CCITT Recommendation MPEG-2, *Generic coding of moving pictures and associated audio*, ISO/IEC 13818, Geneva, Switzerland, 1994.
- [10] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards, Algorithms and Architectures*. Boston, MA : Kluwer Academic, 1995.
- [11] J. Youn, M. Sun and C.W. Lin, "Motion vector refinement for high-performance transcoding", *IEEE Trans. on Multimedia*, Vol. 1, No. 1, pp. 30~40, Mar. 1999.
- [12] B. Shen, I.K. Sethi and B. Vasudev, "Adaptive motion-vector re-sampling for compressed

- video downscaling", *IEEE Trans. on Circuits Syst. Video Technol.*, Vol. 9, No. 6, pp. 929~936, Sept. 1999.
- [13] N. Merhav and V. Bhaskaran, "Fast algorithms DCT domain image down-sampling and for inverse motion compensation", *IEEE Trans. on Circuits Syst. Video Technol.*, Vol. 7, No. 3, pp. 468~476, June 1997.
- [14] K.D. Kim and J.K. Kim, "Fast motion vector refinement for MPEG-1 to MPEG-4 transcoding with spatial down-sampling in DCT domain", *Proc. Int. Conf. Image Processing*, Vol. 1, pp. 469~472, 2001.
- [15] W. Zhy, K.H. Yang and M.J. Beacken, "CIF-to-QCIF video bitstream down conversion in the DCT domain", *Bell Labs Technical Journal*, Vol. 3, No. 3, pp. 21~29, July-Sept. 1998.
- [16] H.W. Park, Y.S. Park and S.K. Oh, "L/M-fold image resizing in block-DCT domain using symmetric convolution", *submitted to IEEE Trans. on Image Processing*, Sept. 2001.
- [17] H. Li, A. Lundma<sup>7</sup> and R. Forchheimer, "Image sequence coding at very low bitrates : a review", *IEEE Trans. on Image Processing*, Vol. 3, No. 5, pp. 589~609, Sep. 1994.
- [18] U.V. Koc and K.J.R. Liu, "DCT-based motion estimation", *IEEE Trans. on Image Processing*, Vol. 7, No. 7, pp. 468~476, June 1997.
- [19] M. Song, A. Cai and J.A. Sun, "Motion estimation in DCT domain", *ICCT 96*, Vol. 2, pp. 670~674, 1996.
- [20] S.F. Chang and D.G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video", *IEEE J. Select. Areas Commun.*, Vol. 13, No. 1, pp. 1~11, 1995.
- [21] ITU-T SG-15, "Video codec test model, TMN5", Jan. 1995.

---

 저 자 소 개
 

---

## 金 雄 熙(正會員)

2000년 고려대학교 전기전자전공공학부 졸업(공학사).  
 2002년 한국과학기술원 전자전산학과 졸업(공학석사).  
 2002년~현재 전자통신연구원부설 국가보안기술연구소(NSRI) 연구원. <주관심분야: 동영상 부호화, 영상코덱 구현 등>

## 吳 承 均(學生會員)

1998년 한국과학기술원 전기 및 전자공학과 졸업(공학사). 2000년 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 2000년~현재 한국과학기술원 전자전산학과 전기 및 전자공학전공 박사과정 재학 중. <주관심분야: 동영상 코덱, 트랜스코딩, 고속영상처리 등>

## 朴 玄 旭(正會員)

1981년 서울대학교 전기공학과 졸업(공학사). 1983년 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1988년 한국과학기술원 전기 및 전자공학과 졸업(공학박사). 1989년~1992년 University of Washington 연구원. 1992년~1993년 삼성전자 수석연구원. 1993년~현재 한국과학기술원 전자전산학과 전기 및 전자공학전공 교수. <주관심분야: 영상처리, 영상압축, 의료영상 시스템, 멀티미디어 시스템>