

웨이블릿 계수 특성을 이용한 고속 움직임 추정 기법

선 동 우*, 배 진 우*, 정희원 유지 상*

A Fast Motion Estimation using Characteristics of Wavelet Coefficients

Dong-Woo Sun*, Jin-Woo Bae*, Ji-Sang Yoo* *Regular Members*

요 약

본 논문에서는 MRME(multiresolution motion estimation)의 처리속도를 개선하고자, 웨이블릿 계수의 각 대역별 특성을 이용하여 영상의 화질을 유지하면서 계산량을 줄일 수 있는 효율적인 움직임 추정 기법을 제안하였다. 일반적으로 동영상의 연속한 프레임간에는 유사성이 높기 때문에, 기저대역에서 블록간 MAD(mean absolute difference) 값을 임계치와 비교하여 먼저 움직임 벡터의 존재 여부를 파악한다. 움직임이 없는 영역으로 판단되어지면, 웨이블릿 변환의 특성에 따라 고주파 부대역에 대해서도 움직임이 없는 것으로 판단한다. 반대로, 움직임이 있는 영역으로 판단되어질 경우는 고주파 부대역의 에너지 성분을 비교하여 선별적으로 움직임을 추정함으로써 계산량을 줄일 수 있다. 모의 실험 결과 제안된 기법에서 MRME와 비교하여 거의 대등한 PSNR을 유지하면서 30~50%의 처리속도 향상을 보였다.

Key Words : fast motion estimation; wavelet; MRME; .

ABSTRACT

In this paper, we propose an efficient motion estimation algorithm which can reduce computational complexity by using characteristics of wavelet coefficient in each subband while keeping about the same image quality as in using MRME(multiresolution motion estimation). In general, because of the high similarity between consecutive frames, we first decide whether the motion exists or not by just comparing MAD(mean absolute difference) between blocks with threshold in the lowest subbands of consecutive two frames. If it turns out that there is no motion in the lowest subband, we can also decide no motion exists in the higher subband. This is due to the characteristics of wavelet transform. Conversely, if we find any motion in the lowest subband, we can reduce computational complexity by estimating high subband motion vectors selectively according to the amount of energy in that subband. Experimental results are shown that algorithm suggested in this paper maintains about the same PSNR as MRME. However, the processing time was reduced about 30-50 % compared with the MRME.

I. 서론

멀티미디어 정보 중 영상데이터는 대용량 때문에 한정된 대역폭을 통하여 전송되거나 저장하기 위해

서는 압축이 필요하다. 일반적으로 동영상은 시간적인 중복성(temporary redundancy)과 공간적인 중복성(spatial redundancy)을 가지고 있다. 공간적인 중복성은 변환 부호화, 예측 부호화, 벡터

* 광운대학교 전자공학과 디지털 미디어 연구실(sdw76@image.gwu.ac.kr, bjw8751@image.gwu.ac.kr, jsyoo@daisy.kw.ac.kr), 논문번호: 020457-1017, 접수일자: 2002년 10월 17일

※ 본 연구는 한국과학재단 목적기초연구(R01-2001-000-00350-0)지원으로 수행되었습니다.

양자화 등과 같은 기법들에 의해 제거될 수 있으며, 시간적인 중복성은 움직임 추정과 보상을 이용하여 제거한다.

이런 공간적인 중복성을 제거하기 위해 기존의 MPEG이나 H.263의 경우 DCT(discrete cosine transform)을 이용한다. 그러나 DCT를 사용하여 영상을 압축할 경우 뛰어난 압축 성능에도 불구하고, 8×8 크기의 화소 블록의 DCT 계수를 부호화하므로 압축율이 높을 경우 블록의 경계면이 눈에 띄게되는 블록화 현상이 발생하여 화질이 많이 떨어지는 단점을 가지고 있다.

웨이블릿 변환(wavelet transform)을 이용하면 블록화 현상이 일어나지 않을 뿐만 아니라, 부대역 부호화와 다해상도 분석기능도 가능하다. 따라서 현재 웨이블릿 변환을 이용하여 공간적인 중복성과 시간적인 중복성을 제거하려는 연구가 활발히 진행되고 있다⁽¹⁾⁻⁽³⁾.

웨이블릿 변환을 이용한 움직임 추정과 보상 방법 중에 대표적인 방법은 1992년에 Ya-Qin Zhang 등이 제안한 MRME(multiresolution motion estimation)이다⁽³⁾. 이 기법은 웨이블릿 변환 영역내의 저해상도 계층에서 전역 움직임(global motion)을 추정하고, 이 움직임 벡터를 이용하여 고해상도 계층에서는 국부 움직임(local motion)을 추정함으로써 부드러운 움직임 벡터장(motion vector field)을 만들 수 있다. 또한 웨이블릿 역변환을 통하여 구획화 현상을 제거할 수 있다. 그러나 MRME 기법은 모든 블록에 대해 움직임을 추정하므로 많은 계산량을 필요로 하고, 전송되어질 데이터가 증가하는 단점을 가지고 있다.

본 논문에서는 각 대역별 특성을 이용하여 영상의 화질은 유지하면서 계산량을 줄일 수 있는 효율적인 움직임 추정 기법을 제안하였다. 기존의 고속 움직임 추정 기법⁽⁴⁾과는 달리, 블록단위로 기저대역에서의 움직임의 존재 여부를 파악한 후, 고주파 대역의 에너지를 분석하여 중요한 블록만을 정제하므로 계산량을 현저히 줄일 수 있다. 즉, 제안된 기법에서는 각 프레임에 대해 웨이블릿 변환 후, 두 프레임(이전 프레임과 현재 프레임)의 기저 대역에서 블록간의 MAD 값을 이용하여, 움직임의 존재 여부를 파악한다. 기저대역에서 움직임이 존재하지 않는 경우, 웨이블릿 변환의 특성상, 고주파 부대역의 블록 또한 움직임이 거의 존재하지 않게 된다⁽¹⁾. 또한 움직임이 존재하는 경우라도, 최상위 고주파 부대역에서 블록의 에너지 성분을 이용하여 선별적으로 움직

임 벡터를 정제한다. 이러한 특성을 이용하여 움직임이 없는 블록에서의 계산량을 감소시킬 수 있을 뿐만 아니라, 움직임이 많은 블록이라 하더라도 선별적으로 움직임 벡터를 계산하기 때문에 효율적으로 계산량을 감소시킬 수 있다. 모의실험을 통하여 성능 평가를 한 결과, 제안한 기법이 기존의 MRME와 비교하여 화질 저하는 거의 없고, 계산량은 상당히 감소함을 확인할 수 있었다.

본 논문의 2장에서는 리프팅을 이용한 웨이블릿 변환과 MRME에 대한 기본 이론을 설명한다. 3장에서는 제안하는 기법에 대하여 설명하고, 4장에서 모의실험 결과를 보이고 5장에서 결론을 내린다.

II. 웨이블릿 변환과 MRME 기법

1. 리프팅을 이용한 웨이블릿 변환

웨이블릿 변환은 DCT 변환의 블록화 현상을 개선하고, 점진적 전송에 있어서도 매우 적합한 방식이다. 웨이블릿 변환은 Morlet에 의해 소개된 이후에 효율이 높고 계산량은 적은 웨이블릿에 관한 연구가 많이 진행되었다^{(5),(6)}. 그 중에 하나가 리프팅 방식이다. 리프팅은 기존의 필터뱅크 방식에 비해 계산량이 절반 정도로 줄어 속도가 빠르고 메모리를 적게 사용하는 장점을 가지고 있다⁽⁷⁾.

리프팅은 쌍직교(biorthogonal) 웨이블릿 변환을 공간축 상에서 구현하는 방법론이다⁽⁸⁾. 리프팅을 이용한 정방향 웨이블릿 변환은 크게 3 단계로 구성된다. 각 단계를 분할(split), 예측(predict), 갱신(update)이라 한다.

분할은 입력 신호를 이웃한 두 성분으로 분할한다. 예를 들어, 입력 신호 $x[n]$ 을 짝수 번째 신호 $x_e[n]$ 과 홀수 번째 신호 $x_o[n]$ 으로 분할한다.

$$\begin{aligned} x_e[n] &= x[2n] \\ x_o[n] &= x[2n+1] \end{aligned} \quad (1)$$

예측 단계에서는 예측 연산자 P를 이용하여 $x_e[n]$ 로부터 $x_o[n]$ 를 예측하는 과정으로서 예측을 한 후의 오차인 웨이블릿 계수 $d[n]$ 를 구한다. 이 $d[n]$ 이 고주파 대역의 웨이블릿 계수에 해당한다.

$$d[n] = x_o[n] - P(x_e[n]) \quad (2)$$

갱신은 입력 신호 $x[n]$ 을 근사화하여 나타내는 스케일링(scaling) 계수 $d[n]$ 를 얻기 위해 $x_e[n]$ 과 $d[n]$ 을 결합한다. 이것은 웨이블릿 계수에 갱신 연산자 U 를 적용한 다음 $x_e[n]$ 를 더하여 구한다. 이 $d[n]$ 이 기저 대역의 웨이블릿 계수에 해당한다.

$$d[n] = x_e[n] + U(d[n]) \quad (3)$$

그림 1은 리프팅을 이용한 정방향 웨이블릿 변환을 나타낸다.

리프팅에 대한 역방향 웨이블릿 변환은 부호와 연산 순서만 바뀌면 쉽게 얻을 수 있다. 그림 2는 역방향 리프팅 구조를 나타내고 있다. 정방향과 반대로 결합(merge) 단계에서는 짝수 항과 홀수 항을 하나의 스트림으로 결합하게 된다⁽⁵⁾.

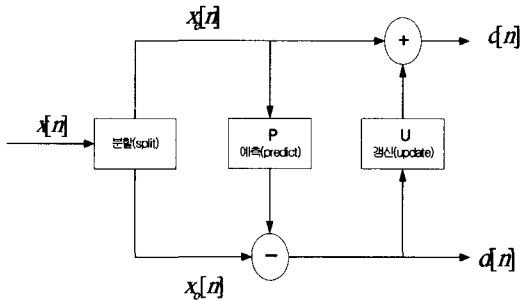


그림 1. 정방향 리프팅 구조도
Fig1. Forward lifting block diagram

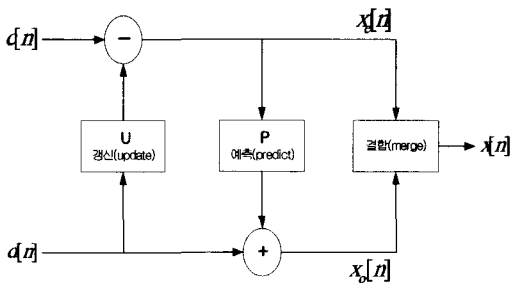


그림 2. 역방향 리프팅 구조도
Fig 2. Inverse lifting block diagram

2. MRME(multiresolution motion estimation)

MRME 기법은 웨이블릿 변환된 영역 내의 기저대역에서 추정된 움직임을 나머지 고주파 부대역들의 초기 움직임으로 사용하여 움직임

벡터를 추정하는 방법이다. 웨이블릿 변환된 영역 내의 최저 해상도를 갖는 M 계층의 부대역들을 $p \times p$ 크기의 화소블록으로 나눈다. 여기서 p 는,

$$p = S \div 2^M \quad (4)$$

이다. 식(4)에서 S 는 원영상에서의 블록 크기이다.

M 계층의 기저대역에서 블록들에 대하여 움직임 추정을 행한다. 그 후 M 계층에 있는 고주파 부대역들은 기저 대역에서 구한 움직임을 기준으로 작은 탐색 영역에서 미소 움직임 변위를 추정한다. 이러한 과정을 정제(refinement)라 한다.

그리고 $M-1$ 계층의 고주파 부대역들을 블록으로 분할한다. 그때 블록의 크기는 $M-1$ 계층이 M 계층에 비해 가로, 세로 해상도가 2배이므로 M 계층에 비해 2배의 크기로 블록을 설정한다. 예를 들어, M 계층의 블록의 크기가 2×2 라면, $M-1$ 계층의 블록은 4×4 의 크기로 설정한다. 즉, 일반적으로 m 계층의 블록의 크기는 $p \cdot 2^{M-m}$ 이다. $M-1$ 계층에서의 정제는 M 계층의 움직임의 크기를 2배하여 이를 기준으로 추정한다.

다음 식은 m 계층 고주파 부대역에서의 움직임 추정을 나타낸 것이다.

$$V_{i,j}(x,y)^{(m)} = V_0(x,y) \cdot 2^{M-m} + \Delta(\delta x, \delta y) \quad (5)$$

식(5)에서 i 는 고주파 부대역의 방향 성분으로 $i=H, V, D$ 가 되며, j 는 스케일 변수를 나타내며 $j=2, 4, 8, \dots$ 의 값을 가진다.

$V_0(x,y)$ 는 좌표 (x,y) 에서의 움직임 벡터이고 $\Delta(\delta x, \delta y)$ 는 정제를 통한 미소 변위를 나타낸다. $\Delta(\delta x, \delta y)$ 는 다음과 같이 표현할 수 있다.

$$\begin{aligned} \Delta(\delta x, \delta y) &= \arg \text{Min} \left[\frac{1}{XY} \sum_{p=-X/2}^{X/2} \sum_{p=-Y/2}^{Y/2} \right. \\ &\quad \left. | I_i(x_1+p, y_1+p) - I_{i-1}(x_1+p+x+\delta x, y_1+p+y+\delta y) | \right] \end{aligned} \quad (6)$$

식 (6)에서 $I_i(x_1, y_1)$ 는 i 번째 프레임의 (x_1, y_1) 위치에서의 화소값을 나타낸다^[3].

MRME 기법은 기존의 공간 영역상에서의 BMA(block matching algorithm)에 비해 탐색과 정합시간이 적게 소모되지만, 모든 부대역에서 움직임을 추정하기 때문에 여전히 많은 계산량과 비트량을 필요로 하는 단점을 가지고 있다.

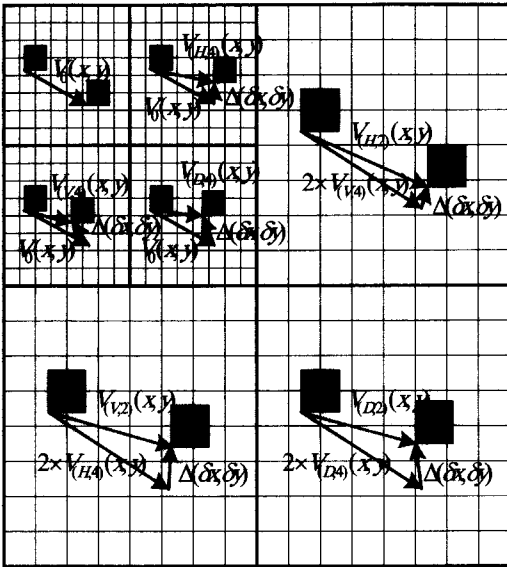


그림 3. 가변 블록 크기의 다해상도 움직임 추정
Fig 3. Multiresolution motion estimation of variable block size

III. 제안한 움직임 추정기법

본 논문에서는 MRME 기법이 계산량이 많다는 문제점을 해결하기 위하여 웨이블릿 계수의 각 대역별 특성을 이용하여 영상의 화질은 유지하면서 계산량을 줄일 수 있는 효율적인 움직임 추정 기법을 제안하였다. 제안된 기법에서는 최저 해상도의 기저대역에서 움직임이 있는 영역을 미리 설정하게 된다. 움직임이 있는 영역으로 설정된 블록은 고주파 부대역에서 블록의 에너지를 비교하여 선별적으로 움직임을 추정하게 된다. 이 과정을 2 단계로 구분하여, 움직임이 있는 영역을 미리 설정하는 단계와 고주파 부대역의 에너지를 비교하여 선별적으로 움직임을 추정하는 단계로 나누었다.

1. 기저 대역의 움직임 영역 설정

동영상은 연속한 프레임간에 높은 유사성이 존재한다. 뉴스 장면을 담고 있는 영상의 경우는 배경의 변화는 거의 없고 아나운서와 같은 객체에만 움직임이 존재한다고 할 수 있다. 이러한 영상의 경우에는 먼저 기저대역에서 움직임 벡터의 존재 여부를 파악하고, 움직임이 없다고 판정된 영역에서는 움직임 추정을 하지 않음으로서 계산량을 줄일 수 있다. 따라서 기저대역에서 움직임이 있는 영역과 없는 영역으로 분리하는 과정이 먼저 수행되어야 한다.

움직임 영역을 추정하는 방법은 다음과 같다. 연속한 두 프레임에서 블록간의 MAD(mean absolute difference)를 구한다.

$$MAD =$$

$$\frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |I_C(x, y) - I_R(x, y)| \quad (7)$$

식 (7)에서 N 은 블록의 크기이고, $I_C(x, y)$ 는 현재 프레임의 블록의 화소, $I_R(x, y)$ 은 이전 프레임의 블록의 화소를 나타낸다.

계산된 MAD 값이 미리 설정된 임계값(threshold) 보다 크면 움직임이 있는 영역으로 설정하고 이와 반대로 임계값보다 작으면 움직임이 없는 영역으로 설정하며 동시에 전체층, 전 대역의 동일 위치의 블록에 대해서 움직임이 없는 것으로 설정한다. 즉 정리하면 다음과 같다.

- i) 만약 $MAD \geq T_0$
→ 움직임이 있는 영역으로 설정
- ii) 만약 $MAD < T_0$
→ 움직임이 없는 영역으로 설정
→ $MV(x, y)_{(i,j)} = 0$

여기에서 T_0 는 임계값이고, $MV(x, y)_{(i,j)}$ 는 각 블록의 움직임 벡터 값이며 $i=H, V, D$ 이고 $j=2, 4, 8$ 이다. 그림 4는 움직임이 없는 영역으로 설정되었을 경우에 각 대역별로 움직임 벡터값이 0으로 할당되는 것을 보여준다.

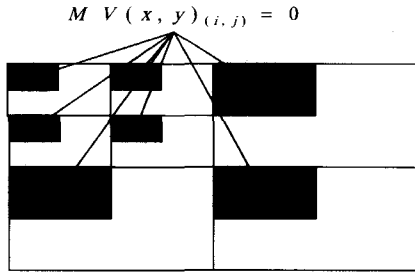


그림 4. 움직임 벡터값 설정
Fig 4. Assignment of motion vector

2. 고주파 부대역의 선별적 움직임 추정 기법

이 과정은 기저 대역의 움직임 영역 설정 후, 움직임이 있는 영역으로 판단된 블록에 대하여 행해지는 과정이다. 즉, 고주파 부대역의 에너지 성분을 고려하여 움직임 벡터의 정제여부를 판단하게 된다. 고주파 부대역의 에너지는 각 부대역 별로 해당 블록내 계수들의 합으로 결정된다. 고주파 부대역에서 에너지 성분이 높은 블록의 경우 영상의 화질에 많은 영향을 미치게 된다. 따라서 높은 에너지 성분을 가지는 부대역에 대해서는 미소 움직임 추정을 통하여 움직임 벡터를 정제하게 된다. 움직임이 있는 블록으로 판단되었어도 해당 블록의 고주파 부대역이 가지는 에너지 성분이 작은 경우에는 기저대역의 움직임 벡터를 그대로 사용하게 된다. 이러한 경우에는 고주파 부대역의 계수가 영상의 화질 열화에 영향을 거의 미치지 않기 때문이다. 즉, 고주파 부대역의 선별적 움직임을 통하여 화질 열화는 최소로 하면서도 계산량을 최대한도로 줄일 수 있게 되는 것이다.

계수값의 합이 실험적으로 결정한 임계값보다 크면 정제하여 움직임 벡터값을 추정하고, 그와 반대로 임계값보다 작으면 정제하지 않고 기저 대역에서 구한 움직임 벡터값을 고주파 부대역에서 그대로 적용한다.

LL	HL2	HL1
LH2	HH2	
LH1		HH1

그림 5. 2단계 웨이블릿 변환
Fig 5. 2-scale wavelet transform

그림 5는 2 계층 웨이블릿 변환후 각 부대역을 나타낸 것이다. 기저대역인 LL영역에서 움직임이 있는 영역으로 설정된 블록에 대해 HL1, LH1, HH1에서 동일한 위치에 있는 블록의 계수들의 합을 구한다.

$$S_{(HL1, k)} = \sum_{x=0}^{R-1} \sum_{y=0}^{R-1} |C_{(HL1, k)}(x, y)|$$

$$S_{(LH, k)} = \sum_{x=0}^{R-1} \sum_{y=0}^{R-1} |C_{(LH, k)}(x, y)| \quad (7)$$

$$S_{(HH, k)} = \sum_{x=0}^{R-1} \sum_{y=0}^{R-1} |C_{(HH, k)}(x, y)|$$

식 (7)에서 $S_{(HL1, k)}$ 는 HL1 영역에서 k 번째 블록내 계수들의 합을 말한다. 그리고 $C_{(HL1, k)}(x, y)$ 는 HL1 영역에서 k 번째 블록내 (x, y) 좌표의 계수값을 말하며, R 은 블록의 크기를 말한다. 계수들의 합이 실험적으로 정한 임계값보다 크면 그 블록에 대해 미소 움직임을 추정하고, 반대로 임계값보다 작으면 기저대역인 LL 영역에서 구한 움직임 벡터 값을 그 블록의 움직임 벡터 값으로 정한다. 이 내용을 c 코드 형태로 정리하면 다음과 같다.

$$\text{if } (S_{(HL1, k)} \geq T_1),$$

$$MV_{(HL2, k)} = MV_{(LL, k)} + \Delta(\delta x, \delta y)$$

$$MV_{(HL1, k)} = 2 \times MV_{(HL2, k)} + \Delta(\delta x, \delta y) \quad (8)$$

else,

$$MV_{(HL2, k)} = MV_{(LL, k)}$$

$$MV_{(HL1, k)} = 2 \times MV_{(LL, k)}$$

LH1, HH1영역에서도 동일하다.

여기서 T_1 은 미소 움직임 추정에 대한 임계값을 말하고, $MV_{(HL1, k)}$ 은 HL1 영역에서 k 번째 블록의 움직임 벡터값을 말한다.

IV. 실험 결과 및 고찰

본 논문에서 제안한 움직임 추정 기법의 성능을 평가하기 위하여 컴퓨터 모의 실험을 행하였다. 본 실험에서는 실험 영상을 352×288의 Salesman, Football 영상과 352×240의 Susi, Table tennis 영상을 각각 10프레임씩 사용하였다. 웨이블릿 변환에 사용된 필터는

5-3 필터이고, 변환은 2 계층까지 분할하였다. 기저대역의 블록 크기는 4×4 로 하였고, 탐색 영역은 -4~4 로 하였다. 고주파 부대역의 미소 움직임 추정시 탐색 영역은 -2~2 로 하였다. 블록 정합의 척도로는 MAD(mean absolute difference)를 사용하였다. 성능 평가를 위해 PSNR(peak signal-to-noise ratio)을 이용하였고 10개 프레임 전체의 움직임 추정시 소용된 계산량은 CPU 처리 시간 및 블록들에 대해 움직임 추정 과정에서 비교되는 후보 블록 총개수의 비교로 측정하였다.

본 실험에서는 움직임 영역 설정시 이용되는 임계값 T_0 를 변화시키면서 제안된 기법의 성능을 기존의 기법과 비교해 보았다. 즉, T_0 를 상수로 고정시키는 경우와 블록의 MAD값에 따라 적응적으로 설정하는 경우를 실험하였다. 그 결과 상수로 고정시킨 경우에는 움직임이 많은 영상의 경우에는 처리 속도의 향상이 저조하였다는 실험 결과를 얻었다. 적응적으로 설정한 방법은 기저 대역에서 블록간의 MAD 값을 구해 그 중간값을 T_0 로 하는 방법과 MAD 값의 평균값을 이용하는 방법 두 가지를 적용하였다. 이 경우 평균값을 이용하는 방식은 다음과 같다.

$$\begin{aligned}
 &\text{if } 0 \leq MAD_{Avr} < 3 \quad , D = 1.3 \\
 &\text{if } 3 \leq MAD_{Avr} < 8 \quad , D = 2.0 \\
 &\text{if } 8 \leq MAD_{Avr} < 15 \quad , D = 3.0 \quad (9) \\
 &\text{if } 15 \leq MAD_{Avr} < 20 \quad , D = 4.2 \\
 &\text{else} \quad , D = 9.0 \\
 &T_0 = MAD_{Avr} \div D
 \end{aligned}$$

식 (9)에서 MAD_{Avr} 는 두 프레임의 기저 대역에서 블록간 MAD 값의 평균값을 나타내며, D 는 실험에 의해 정한 값이다. 실험결과 적응적으로 임계값을 정하면 영상의 특성에 관계없이 일정하게 처리 속도를 향상시킬 수 있었다. 특히, 임계값을 중간값으로 설정한 경우보다 평균값을 이용한 경우가 처리속도 개선에 더 효율적이라는 결과를 얻을 수 있었다. 그 이유는 각 블록에 해당하는 임계값이 적응적으로 적용되어 움직임이 없는 영역에 대하여 효율적으로 판단할 수 있었기 때문이다.

표 1은 기존의 MRME 기법과 3 가지 방법에 의해 정해진 T_0 의 값을 제안한 기법에 적용하여 각각의 평균 PSNR 값과 CPU 처리 시간을 나타낸 것이다. 표에서 보는 바와 같이 MRME 기법과 비교하여 PSNR은 크게 저하되지 않으며 처리 속도는 30~50%정도 향상되는 것을 알 수 있다. 또한 앞서 설명한 바와 같이 전체적으로 T_0 를 적응적으로 구한 경우 가장 효율적인 결과를 얻을 수 있었다. 그리고 표 2는 MRME 기법과 제안한 기법의 계산량을 비교한 것이다. 계산량의 비교는 블록들에 대해 움직임 추정 과정에서 비교되는 후보 블록의 총개수를 비교하여 백분율로 나타내었다.

MRME의 경우,

$$N_{MRME} = N_{LL} \times N_S + N_H \times N_R \quad (10)$$

와 같다.

식(10)에서, N_{MRME} 는 MRME 기법을 사용하여 블록들에 대해 움직임 추정 과정에서 비교되는 후보 블록의 총개수를 말하며, N_{LL} 은 LL영역에서의 블록 개수를 말한다. 그리고, N_S 는 LL영역에서의 탐색 구간내의 블록 개수를 말하고, N_H 는 LL영역을 제외한 나머지 대역의 총 블록 개수를 나타내며, N_R 는 정제 구간내의 블록 개수를 말한다.

제안한 논문의 경우,

$$\begin{aligned}
 N_P = &((N_{LL} - N_{LT}) \times N_S + N_{LL}) \\
 &+ ((N_H - N_{HT}) \times N_R + N_{LT} \times 3) \quad (11)
 \end{aligned}$$

와 같다.

식(11)에서, N_P 는 제안한 기법을 사용하여 각 블록에 대해 움직임 추정 과정에서 비교되는 후보 블록의 개수를 말하며, N_{LT} 는 LL영역에서의 임계값보다 작은 블록수를 말한다. 그리고, N_{HT} 는 LL영역을 제외한 나머지 대역에서 정제되지 않는 블록 개수를 말한다. 이 식에서 N_{LL} 와 $N_{LT} \times 3$ 를 각각 더한 이유는 LL영역에서 블록의 움직임 여부 및 고주파 대역에서의 에너지 성분을 비교하여 중요 블록을 설정하기 위해 필요한 비교 횟수이다.

식(12)는 백분율을 나타낸다.

$$P = \frac{N_P}{N_{MRME}} \times 100 \quad (\%) \quad (12)$$

표 (2)에서 보는 바와 같이 임계값을 적응적으로 설정한 경우에 블록들에 대해 움직임 추정 과정에서 비교되는 후보 블록 총개수가 MRME에 비해 20~40%정도에 그친다.

그림6은 4개의 실험 영상에 대한 10개 프레임의 PSNR을 그래프로 표현한 것이다. 그림에서 보는 바와 같이 전 프레임에 걸쳐 PSNR의 저하가 크게 생기지 않음을 알 수 있다. 그리고, 그림 7-(a)는 Salesman 10번째 영상을 나타낸다. (b)는 MRME를 이용해서 움직임 보상된 영상이며, (d)는 제안된 기법으로 움직임 보상된 영상을 나타낸다. 영상에서 보는 바와 같이 화질에서 큰 차이가 없음을 알 수 있다. 그리고 (c)는 (a)와 (b)를 뺀 영상이고, (e)는 (a)와 (d)를 뺀 영상을 보여준다. 이 차 영상 역시 큰 차이를 볼 수 없다.

V. 결론

본 논문에는 MRME의 단점 중에 하나인 많은 계산량을 줄이고자 웨이블릿 계수의 각 대역별 특성을 이용하여 효율적인 움직임 추정 기법을 제안하였다. 제안된 기법에서는 웨이블릿 다해상도 분할의 특성을 이용하여 기저 대역에서 움직임이 있는 영역과 움직임이 없는 영역을 설정하고, 움직임이 있는 영역으로 설정된 블록에 대해서는 선별적으로 미소 움직임을 추정하므로써 계산량을 감소시켰다. 특히, 블록의 움직임 유무를 판별하는 과정에서 적응적인 방법을 이용함으로써 영상의 특성에 관계없이 성능이 일정하게 향상 되도록 하였다. 또한 고주파 부대역에서 영상의 화질에 영향을 많이 주는 주요 블록들에 대하여 미소 움직임을 추정함으로써 화질 열화를 최소화 하였다. 실험 결과에서도 알 수 있듯이 기존 MRME 기법과 비교한 결과 화질 저하는 최소화하며 계산량을 30~50%정도 줄일 수 있었다. 제안된 기법은 DVR 환경이나, 핸드폰이나 PDA와 같은 각종 모바일 환경에서 필요시 되고 있는 실시간 동영상 부호화에 응용될 수 있을 것으로 기

대된다.

표 1. MRME 기법과 제안한 기법의 평균 PSNR과 처리시간

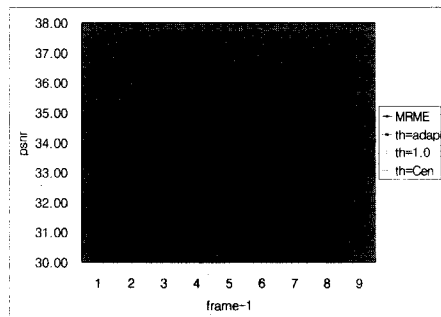
Table 1. Average PSNR and processing time for MRME and the proposed method

		image	Sales-man	Foot-ball	Table tennis	Susi
MRME	평균 PSNR(db)		34.39	24.62	30.49	33.87
	processing time(sec)		6.208	6.218	5.157	5.177
제안한 기법	Thres = 1.0	평균 PSNR(db)	34.22	24.53	30.57	33.84
		processing time(sec)	3.715	5.608	3.174	3.114
	Thres = Cen	평균 PSNR(db)	34.28	24.53	30.57	33.83
		processing time(sec)	4.005	3.955	3.254	2.944
	Thres = adap	평균 PSNR(db)	34.19	24.50	30.58	33.83
		processing time(sec)	3.414	3.815	2.543	2.794

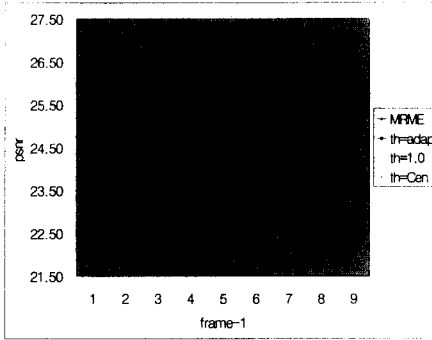
표 7. 움직임 추정 대상 블록의 개수

Table 2. Number of blocks required for motion estimation

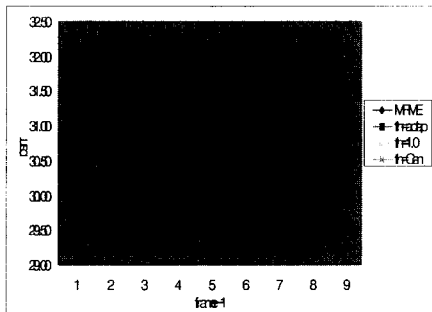
		image	Sales-man	Foot-ball	Table tennis	Susi
제안한 기법	Thres=1.0		34.2%	78.1%	35.9%	35.0%
	Thres=Cen		41.9%	38.7%	38.4%	31.7%
	Thres=adap		30.4%	39.0%	20.9%	30.5%



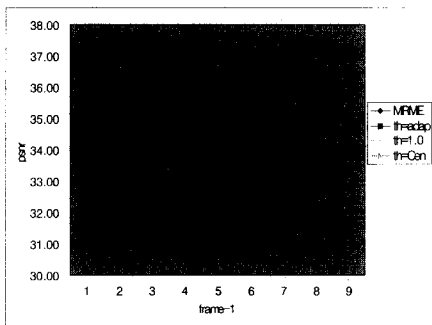
(a) Salesman



(b) Football

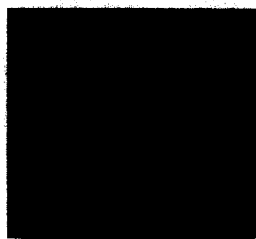


(c) Table tennis

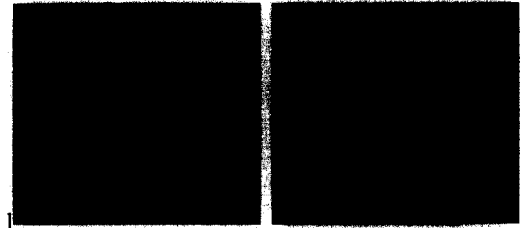


(d) susi

그림 6. 실험 영상에 대한 PSNR
Fig 6. PSNR of test sequences



(a) 원영상



(b) MRME

(c) (a)와 (b)의 차영상



(d) 제안한 기법

(e) (a)와 (d)의 차영상

그림 7. 움직임 보상된 영상과 차영상(Saleman 10번째 영상)
Fig 7. Motion compensated image and residual image(Salesman 10th image)

참 고 문 헌

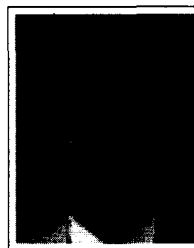
- [1] S.Mallat, "A Theory for Multiresolution Signal Decomposition : The Wavelet Representation", IEEE Trans. Pattern Analysis Mach. Intell, vol. 11, pp.674-693, July, 1989
- [2] S.Mallat, "Multifrequency Channel Decomposition of Images and Wavelet Models", IEEE Trans. ASSP, vol.37, pp.2091-2110, Dec, 1989
- [3] Ya-Qin Zhang and Sohail Zafar, "Motion Compensated Wavelet Transform Coding for Color Video Compression", IEEE Tr. on Circuits and Systems for Video Technology, vol. 2, no. 3, September, 1992
- [4] Tingrong, ZHAO, Masao Yanagisawa, and Tatsuo OHTSUKI, "A High Performance Embedded Wavelet Video Coder", IEICE Trans. Fundamentals, vol.E83-A, no.6, June, 2000
- [5] 신철, 김호식, 유지상, "저 메모리를 갖는 제로트리 기반 영상 압축," 한국 통신학회 논문지, 제 27권 제8호, pp 526-532, 20

02년 8월.

- [6] O.Rioul and M.Vetterli, "Wavelets and Signal Processing", IEEE Signal Processing Magazine, pp.14-38, Oct. 1991
- [7] W.Sweldens and P.Schröder. "Building your own wavelets at home", Wavelets in Computer Graphics, p p.15-87, ACM SIGGRAPH Course notes, 1996
- [8] W.Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets", J.Appl.Comp. Harm.Anal, vol.3, no.2, pp.186-200, 1996

유 지 상(Ji-Sang Yoo)

정회원

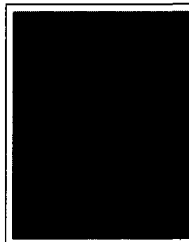


1985년 2월 : 서울대학교 전자공학과 졸업(공학사)
 1987년 2월 : 서울대학교 대학원 전자공학과 졸업(공학석사)
 1993년 5월 : Purdue 대학교 전기공학과 졸업(Ph.D.)
 1993년 9월 ~ 1994년 8월 : 현대전자산업(주) 산전연구소 선임연구원
 1994년 9월~1997년 8월 : 한림대학교 전자공학과 조교수
 1997년 9월~2001년 8월 : 광운대학교 전자공학과 조교수
 2001년 9월~현재 : 광운대학교 전자공학과 부교수

<주관심분야> : 웨이블릿 기반 영상처리, 영상압축, 영상인식, 비선형 신호처리

선 동 우(Dong-Woo Sun)

준회원

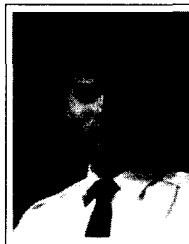


2002년 2월 : 광운대학교 전자공학과 졸업
 2003년 3월~현재 : 광운 대학교 전자 공학과 석사과정

<주관심분야> 웨이블릿기반영상처리, 영상압축, 스테레오 영상압축

배 진 우(Jin-Woo Bae)

준회원



1998년 2월 : 순천대학교 전자공학과 졸업
 2003년 2월 : 광운대학교 전자공학과 석사
 2003년 3월~현재 : 광운 대학교 전자 공학과 박사과정

<주관심분야> 웨이블릿기반영상처리, 동영상 코덱, 영상통신