

효과적인 움직임 예측을 위한 정수형 리프팅 기반의 저대역 이동법에 관한 연구

존회원 백정은*, 정회원 장선봉**, 신중홍***, 종신회원 지인호*

A Study on Low-Band-Shift with Integer Lifting for Effective Motion Estimation

Jung-eun Baek* *Associate Member*, Sun-bong Jang**, Jong-hong Shin*** *Regular Members*,
Inn-ho Jee* *Lifelong Member*

요 약

본 논문에서는 정수형 리프팅 기반의 저대역 이동법을 이용한 움직임 추정과 보상방식을 제안하였다. 저대역 이동법(Low-Band-Shift method)은 웨이블릿 계수들의 이동-변환 성질을 극복하기 위해 제안된 방법으로 동영상 부호화 시 참조 프레임에 적용하면 정확한 움직임 추정이 가능하여 일반적인 방법보다 압축 대비 우수한 성능을 가지지만, 사용되는 메모리량과 계산량이 많아지는 단점을 가진다. 이를 개선하기 위해 본 논문에서는 웨이블릿에 비해 계산량을 1/2이상 줄일 수 있고 사용되는 메모리량을 약 7배 이상 줄일 수 있는 정수형 리프팅을 제안하여 저대역 이동법에 적용해 움직임을 추정하였다.

Key words : Integer Lifting, Low-Band-Shift(LBS), Shift Variant, Motion Estimation

ABSTRACT

In this paper, we propose motion estimation and compensation using Low-Band-Shift(LBS) with Integer Lifting Shift property. To overcome shift-variant property on wavelet coefficients, the LBS was previously proposed. This method which is applied to reference frame in video coding technique has superior performance in terms of rate-distortion characteristic. However, this method needs more memory and computational complexity. For the enhancement of those disadvantages, we propose motion estimation using Low-Band-Shift with Integer Lifting. The Integer Lifting implementation gives us good efficiency of calculation and storage.

I. 서론

영상신호를 디지털 형태로 처리할 경우 전송과 저장에 상당히 넓은 대역폭과 메모리를 필요로 하므로 영상신호를 충실하게 재생하면서 영상의 시간적, 공간적 중복성을 효과적으로 제거하기 위한 데이터 압축 기법들이 많이 연구되어 왔다^[1]. 그 중에

서 웨이블릿 변환(wavelet transform)은 다해상도 대역분할 방식으로 움직임 예측이 많은 곳에서 응용되어 지고 있다. 그러나 웨이블릿 변환 영역에서의 움직임 보상 부호화는 웨이블릿 변환 영역에서의 계수값이 공간 영역에서 화소 값 이동시에 그 값들이 변하는 이동-변환 성질(Shift-Variant Property)로 인하여 정확한 움직임 추정을 할 수가 없기 때

※ 본 논문은 서울시 전략사업 혁신클러스터 육성 지원사업에 의하여 지원되었습니다.

* 홍익대학교 컴퓨터정보통신공학과 (ijee@wou.hongik.ac.kr), ** 홍익대학교 전자공학과 (sunbjj@won.hongik.ac.kr)

*** 한국사이버대학교 컴퓨터정보통신학부 (sigs@mail.kcu.ac)

논문번호 : KICS2006-06-272, 접수일자 : 2006년 6월 13일, 최종논문접수일자 : 2006년 9월 25일

문에 공간 영역 움직임 보상 부호화 방법보다 화질과 부호화 등의 측면에서 효율이 떨어지게 된다. 이를 극복하기 위해 저대역 이동법(Low-Band-Shift Method)이 제안되었다^[2]. 저대역 이동법은 움직임 추정시 사용되는 참조 영상을 웨이블릿 변환 할때 가로, 세로, 대각선 방향으로 한 화소단위로 이동한 후 한 단계 웨이블릿 변환을 하는 부분을 추가하고 그 변환된 영상의 저주파 대역에 대해 반복적으로 화소 단위 이동과 웨이블릿 변환을 수행하여 움직임 추정을 위한 참조 영상을 만든다. 이 방식은 임의의 위치에 있는 화소가 이동시 이동되기 전의 계수 값들에 대하여 움직임 추정에 대한 참조 영상을 만들어 낼 수 있기 때문에 보다 정확한 움직임 보상 부호화가 가능하여 개선된 결과를 가져오지만 계산량과 메모리가 증가하게 된다. 본 논문에서는 메모리의 계산량을 줄일 수 있는 방법으로 리프팅을 제안한다. 웨이블릿의 장점을 모두 가지고 있는 리프팅은 웨이블릿의 컨벌루션 연산을 덧셈과 쉬프트 연산으로 대체하기 때문에 계산량이 적어져 속도가 빨라지고 대체 연산이 가능하기 때문에 메모리를 줄일 수 있는 장점을 가지고 있다^[3]. 특히 정수형 리프팅은 기존의 리프팅보다 계산량과 메모리를 더 줄일 수 있는 장점을 갖기 때문에 정수형 리프팅 기반의 저대역 이동법을 제안한다.

II. 다해상도 영역에서의 움직임 예측

이미지 압축에 널리 사용되는 웨이블릿 변환은 이산 웨이블릿(discrete wavelet) 함수에 해당하는 계수 값을 갖는 필터들의 집합에 기반을 둔다. 이산 웨이블릿 변환의 기본적인 동작은 N개의 표본치를 갖는 이산 신호에 적용된다. 신호에 대해 한 쌍의 필터를 적용시켜 저주파 대역(L)과 고주파 대역(H)으로 분리한다. 각 대역은 2라는 요소로 부 표본화 되었으므로 N/2개의 표본치를 포함한다. 완전재생 필터를 사용한다면 이 과정은 역변환이 가능하다. 그림 1은 1-level의 2차원 웨이블릿 변환을 나타내고 있다. 여기서, $H_0(z)$ 는 저주파 통과 필터이고, $H_1(z)$ 은 고주파 통과 필터이다. 2차원 웨이블릿 변환의 경우 입력을 수평축으로 웨이블릿 변환하고 그 결과를 다시 수직축으로 웨이블릿 변환하여 LL, LH, HL, HH의 주파수 대역으로 분할된다. 이러한 부 표본화 과정으로 인하여 웨이블릿 변환은 이동 변환 성질을 갖게 된다. 웨이블릿 변환의 대역 분할 방식과 이동 변환 성질로 인하여 움직임 예측 오류

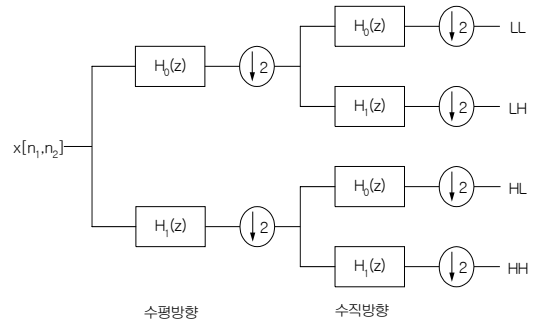


그림 1. 2차원 웨이블릿 변환

가 최저 주파수 대역에서 발생하면 움직임 예측 오류는 다음 단계로 누적되고 각각의 단계에서 계산에 소요되는 시간과 데이터 량이 증가하는 문제점을 가지게 된다.

이러한 문제점은 모서리 부분에서 자주 일어난다. 웨이블릿 영역의 움직임 추정방법은 이러한 문제점으로 공간영역 움직임 추정에 비해 효율이 떨어지는 것으로 알려져 있다^[4].

III. 저대역 이동법을 이용한 움직임 예측

3.1 저대역 이동법

저대역 이동법은 웨이블릿 변환영역에서 움직임 추정시 이동-변환 성질을 극복하기 위해 제안된 방법이다^[5]. 저대역 이동법은 입력 영상을 웨이블릿 변환시 가로, 세로, 대각선 방향으로 한 화소단위로 이동한 후 웨이블릿 변환하는 것을 추가하고, 그 변환된 영상의 저주파 대역에 대해 반복적으로 화소 단위 이동과 웨이블릿 변환을 수행하게 된다. 그림 2는 저대역 이동법을 이용한 1차원 신호의 3단계 웨이블릿 분해에 대한 그림이다. 여기서, d 를 공간 영역에서 이동한 수, x 를 부대역에 위치한 신호라 할 때 $H_f^{(i)}(d, x)$ 는 입력 $f(n)$ 의 i 번째 고대역 신호를 나타낸다. 첫 번째 단계에서 원 신호와 한 화소 이동한 신호가 저대역과 고대역으로 분해되고, 거기서 원 신호의 고대역 신호 $H_f^{(1)}(0, x)$ 와 한 화소 이동된 신호의 고대역 신호 $H_f^{(1)}(1, x)$ 가 생성된다. 계층적 분해를 위해 저대역 신호는 첫 번째 단계와 같은 방법으로 계속 분해되어 진다. 만약 3단계까지 분해가 된다면, 총 8개의 저대역과 14개의 고대역 신호가 생성된다. 예를 들어 $f_s(n)$ 을 $f(n)$ 의 d 화소 이동된 신호라 놓으면 $f(n)$ 은 식 (1)과 같이 놓을 수 있다.

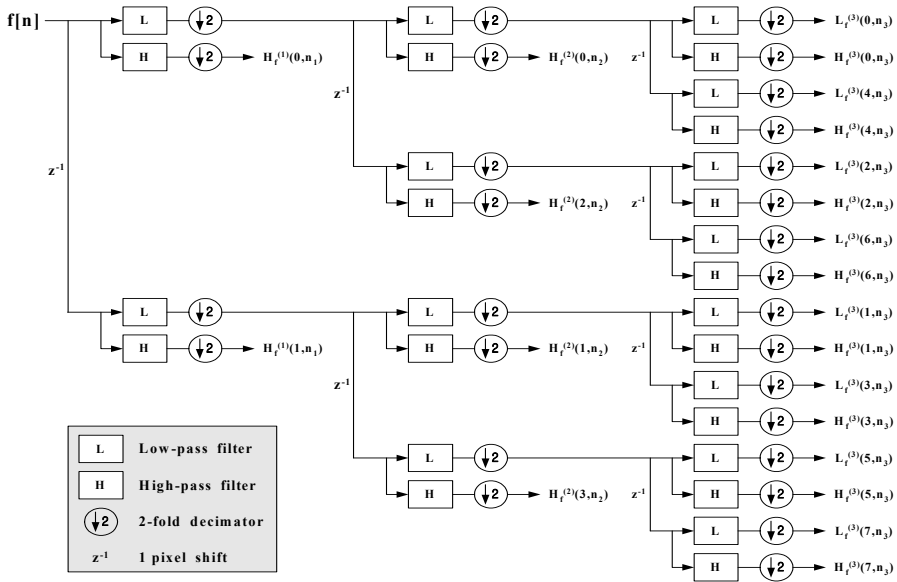


그림 2. 1차원 신호에 대한 저대역 이동법 (3단계 분해)

$$f_s(n) = f(n + d) \tag{1}$$

이동된 신호 $f_s(n)$ 의 각각의 부대역을 $f(n)$ 의 변환으로 표현하면 다음과 같다.

$$H_{f_s}^{(1)}(n_1) = H_f^{(1)}\left(d\%2, n_1 + \left\lfloor \frac{d}{2} \right\rfloor\right) \tag{2}$$

$$H_{f_s}^{(2)}(n_2) = H_f^{(2)}\left(d\%2^2, n_2 + \left\lfloor \frac{d}{2^2} \right\rfloor\right) \tag{3}$$

$$H_{f_s}^{(3)}(n_3) = H_f^{(3)}\left(d\%2^3, n_3 + \left\lfloor \frac{d}{2^3} \right\rfloor\right) \tag{4}$$

$$L_{f_s}^{(3)}(n_3) = L_f^{(3)}\left(d\%2^3, n_3 + \left\lfloor \frac{d}{2^3} \right\rfloor\right) \tag{5}$$

여기서, %는 모듈로(modulo)연산, $x[n]$ 는 x 보다 작거나 같은 최대 정수, n_1, n_2, n_3 는 각각 1번, 2번, 3번 부 표본화된 신호 좌표, 그리고 n 은 공간영역 좌표를 나타낸다. 또한, $H_{f_s}^{(1)}(n_1), H_{f_s}^{(2)}(n_2), H_{f_s}^{(3)}(n_3)$ 과 $L_{f_s}^{(3)}(n_3)$ 은 각각 첫 번째, 두 번째, 세 번째 단계의 고대역 신호와 세 번째 단계의 저대역 신호이다. 식 (2)~식(5)와 그림 2를 보면 알 수 있듯이, 모든 이동된 신호의 웨이브렛 계수 $f_s(n)$ 은 원 신호 $f(n)$ 의 저주파수 대역을 이동함으로써 얻어질 수 있음을 알 수 있다. 이 방식을 2차원 영상에 적용시키려면 1차원 방식을 가로, 세로에 대해서 수행시킴으로써 쉽게 적용이 가능하다.

3.2 저대역 이동법을 이용한 움직임 추정과 보상^[2]

동영상 정보는 공간적인 중복성과 시간적인 중복성이 있으며, 공간적 중복성을 줄이는 부호화 기법을 프레임내 부호화라고 하고, 시간적 중복성을 제거하는 것을 프레임간 부호화라 한다. 동영상은 특히 시간적인 중복성을 효율적으로 줄여야 하는데, 이전 프레임의 영상 정보를 그대로 활용하면서 변화된 부분, 즉 움직임이 발생한 부분만을 찾아서 그 정보만을 보내 주면 뛰어난 부호화 효율을 얻을 수 있다. 이때 현재 영상에서 움직임이 발생한 영역과 가장 비슷한 부분을 이전 프레임 영상에서 찾는 과정을 움직임 추정이라고 하며, 이 추정된 정보를 현재 프레임에 예측(prediction)하여 예측 오차만을 부호화 하게 된다. 변위 차분 프레임(Displacement Frame Difference: DFD)은 움직임 추정 과정에서 얻어진 움직임 벡터에 의해 재구성된 참조 프레임과 현재 프레임간의 예측 오차를 의미하며 두 프레임간의 시간적 중복성이 매우 높다면 이 변위 차분 프레임의 정보량은 매우 낮아지게 된다. 일반적으로 동영상 부호화 시스템에서는 이러한 움직임 보상 오차 프레임인 변위 차분 프레임과 움직임 벡터를 부호화 하는 방식을 사용하므로 효율적인 부호화를 위해서는 정확한 움직임 추정이 필요하다. 웨이브렛 변환 영역에서의 움직임 추정은 이동-변환 성질로 인하여 프레임간에 움직임이 일어났을 때, 그 계수 값이 공간영역에서 이동된 위치에 따라 값이 달라

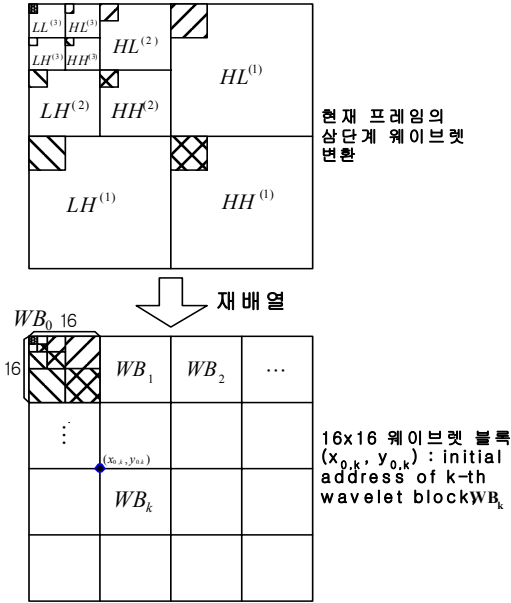


그림 3. 웨이블릿 계수들의 재배열

지게 되므로 정확한 움직임 추정이 어렵다. 저대역 이동법을 사용한다면 현재 프레임의 임의의 화소에서 이동된 화소의 변화된 웨이블릿 계수값을 참조할 수 있기 때문에 일반적인 웨이블릿 변환 영역에서의 움직임 추정보다 부호화 면에서 더 효율적인 움직임 추정을 할 수 있다. 기존의 저대역 이동법을 이용한 부호화기에서는 움직임 추정을 위해 그림 3과 같이 공간영역에서 같은 위치의 계수들끼리 재배열을 하게 된다.

재배열된 웨이블릿 블록에 대하여 움직임 추정을 하기 위해 MPEG 이나 H.263등과 같은 동영상 부호화 기법에서 널리 쓰이는 전 영역 블록 정합 알고리즘(full-search matching algorithm)을 사용한다 [6]. 여기서 사용되는 정합 기준으로는 일반적으로 평균 절대 오차(MAD)를 사용하여 MAD가 최소화 되도록 한다. 평균 절대 오차 함수에 의한 움직임 벡터 (dx, dy) 의 결정은 식(6)과 같다.

$$(dx, dy) = \operatorname{argmin}_{p=-\frac{M}{2}, q=-\frac{N}{2}}^{\frac{M}{2}, \frac{N}{2}} \sum_{x=p}^{\frac{M}{2}} \sum_{y=q}^{\frac{N}{2}} |I_{cur}(x+p, y+q) - I_{ref}(x+p+dx, y+q+dy)| \quad (6)$$

여기서, $I_{cur}(x+p, y+q)$: 현재 프레임의 블록
 $I_{ref}(x+p+dx, y+q+dy)$: 참조 프레임의 블록
 M, N : 블록의 가로, 세로 크기이다.

저대역 이동법을 이용한 움직임 추정에서, 3단계 웨이블릿 분해시 k 번째 블록에 대한 블록 정합 움직임 추정 알고리즘의 적용은 식(7)과 같다.

$$\begin{aligned} MAD_k(dx, dy) &= \sum_{i=1}^3 \sum_{x_i=x_{0,k}}^{M/2^i-N/2^i-1} \sum_{y_i=y_{0,k}} \left\{ \left| HL_{cur}^{(i)}(x_i, y_i) - HL_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \right| \right. \\ &\quad + \left| LH_{cur}^{(i)}(x_i, y_i) - LH_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \right| \\ &\quad \left. + \left| HH_{cur}^{(i)}(x_i, y_i) - HH_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \right| \right\} \\ &+ \sum_{x_3=x_{3,k}}^{M/2^3-N/2^3-1} \sum_{y_3=y_{3,k}} \left\{ \left| LL_{cur}^{(3)}(x_3, y_3) - LL_{ref}^{(3)}\left(dx\%2^3, dy\%2^3, x_3 + \left\lfloor \frac{dx}{2^3} \right\rfloor, y_3 + \left\lfloor \frac{dy}{2^3} \right\rfloor\right) \right| \right\} \quad (7) \end{aligned}$$

여기서 $LH_{cur}^{(i)}, HL_{cur}^{(i)}, HH_{cur}^{(i)}$ 는 현재 프레임에서 i 번째 웨이블릿 분해 단계의 저-고, 고-저, 고-고대역을 나타내며, $LL_{cur}^{(3)}$ 은 현재 프레임의 최저 주파수 대역을 나타낸다. 마찬가지로 $LH_{ref}^{(i)}, HL_{ref}^{(i)}, HH_{ref}^{(i)}$ 는 참조 프레임에서 i 번째 웨이블릿 분해 단계의 저-고, 고-저, 고-고대역을 나타내며, $LL_{ref}^{(3)}$ 는 참조 프레임의 최저 주파수 대역이다. i 번째 웨이블릿 분해 단계의 부대역에서 k 번째 웨이블릿 블록의 시작점 $(x_{i,k}, y_{i,k})$ 은 식(8)과 같이 정의된다.

$$x_{i,k} = \frac{x_{0,k}}{2^i}, y_{i,k} = \frac{y_{0,k}}{2^i} \quad (8)$$

이때, $(x_{0,k}, y_{0,k})$ 는 공간 영역에서 k 번째 웨이블릿 블록의 시작점이다. 오차가 최소가 되는 최적 움직임 벡터는 다음 식에 의해서 결정된다.

$$(dx_{\min}, dy_{\min}) = \min_{(dx, dy) \in W} MAD_k(dx, dy) \quad (9)$$

여기서, W 는 움직임 추정 블록의 탐색영역이다. 움직임 벡터를 (dx, dy) 라 놓을 때, 움직임 보상 블록에 대한 식은 식(10)~식(13)과 같다.

$$\begin{aligned} HL_{mc}^{(i)}(x_i, y_i) &= \\ HL_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) & \quad (10) \end{aligned}$$

$$LH_{mc}^{(i)}(x_i, y_i) = LH_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \quad (11)$$

$$HH_{mc}^{(i)}(x_i, y_i) = HH_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \quad (12)$$

$$LL_{mc}^{(3)}(x_3, y_3) = LL_{ref}^{(3)}\left(dx\%2^3, dy\%2^3, x_3 + \left\lfloor \frac{dx}{2^3} \right\rfloor, y_3 + \left\lfloor \frac{dy}{2^3} \right\rfloor\right) \quad (13)$$

단, $i = 1, 2, 3, x_{i,k} \leq x_i \leq x_{i,k} + \frac{M}{2^i} - 1,$

$$y_{i,k} \leq y_i \leq y_{i,k} + \frac{M}{2^i} - 1.$$

여기서, $LH_{mc}^{(i)}, HH_{mc}^{(i)}, LL_{mc}^{(i)}$ 는 i 번째 웨이브렛 분해 단계의 저-고, 고-저, 고-고대역의 움직임 보상 블록이고, 마찬가지로 $LL_{mc}^{(3)}$ 은 3번째 단계의 저대역에 대한 움직임 보상 블록이다.

IV. 제안 방식

4.1 정수형 리프팅

리프팅 변환은 부대역 부호화 과정에 noble identity 적용하여 부 표본화 과정이 필터를 통과하기 이전에 실행되도록 한다. 이렇게 하여 $\tilde{P}(z^{-1})$ 의 다위상 행렬을 얻게 되고 $\tilde{P}(z^{-1})$ 을 Euclidean 알고리즘을 사용하여 인수분해 된다.

인수분해 된 대수식은 각각의 필터링 과정에서 반올림 과정을 거쳐 정수형태의 변환 계수를 얻을 수 있다. 그림 4는 정수형 리프팅 변환 과정을 보

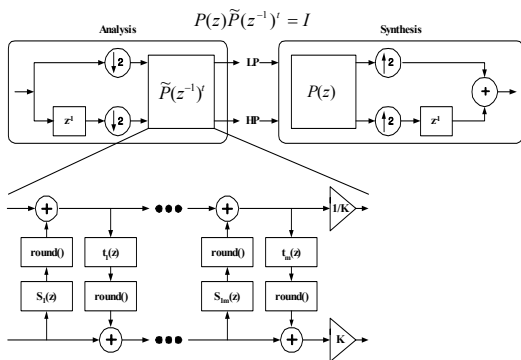


그림 4. 정수형 리프팅 변환의 다위상 표현

여준다. 웨이브렛 변환의 경우 실수 형태의 필터를 사용하여 변환 계수를 얻기 때문에 정수 형태의 변환 계수를 얻기 어렵다. 하지만 리프팅 변환의 경우 정수 형태의 데이터를 쉽게 유지 할 수 있으며, 쉽게 변환 과정과 역변환 과정을 얻을 수 있는 장점이 있다. 분석단에서 반올림 처리로 인하여 발생하는 계수의 손실은 합성단에서 보상된다. 정수형 리프팅은 리프팅에 비해 계산을 단순하게 해주어 속도를 빠르게 하고 메모리를 줄여주는 효과가 있다.

4.2 제안된 방법을 이용한 움직임 추정 및 보상

제안한 방법은 기존의 저대역 이동법(LBS)의 다 해상도 움직임 추정 개념을 이용한 방법으로써 저대역 이동법의 단점인 계산량이 많고, 많은 메모리가 요구되는 점을 보완하고자 하였다.

동영상 부호화에서는 무엇보다 실시간 처리가 중요하다. 또한 동영상은 많은 메모리를 요구하므로 메모리를 줄일 수 있는 방법이 요구된다. 저대역 이동법은 계산량이 많기 때문에 시간이 다소 오래 걸리는 단점이 있다. 리프팅은 저대역 이동법에서 사용된 웨이브렛에 비해 계산량이 거의 절반에 가깝고 대체 연산이 가능하여 메모리 요구량이 줄어든다. 그 중에서도 정수형 리프팅은 일반적인 리프팅에 비해 소숫점 연산이 아닌 정수 연산을 실행하기 때문에 연산의 속도 및 메모리의 활용 측면에서 리프팅 변환에 비해 장점이 있다. 본 논문에서는 저대역 이동법의 단점을 극복하고자 계산속도가 빠르고 대체 연산이 가능한 특징을 가지는 정수형 리프팅을 이용한 저대역 이동법을 제안한다. 그림 5는 정수형의 리프팅 변환 계수를 사용하여 움직임 추정과 보상을 실행하기 위한 블록 다이어그램을 나타내고 있다.

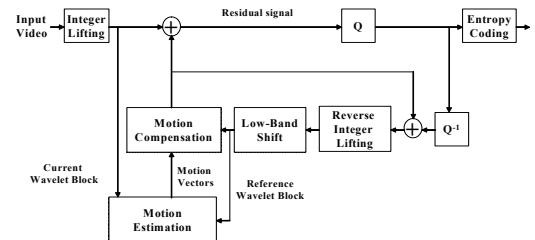


그림 5. 제안된 방법의 움직임 추정과 보상된 블록 다이어그램

V. 실험

제안된 알고리즘의 성능 평가를 위해 동영상 모

의 실험에 널리 사용되고 있는 QCIF(176X144) 크기를 갖는 Akiyo, Carphone, Clair 영상의 Y성분을 실험에 사용하였다. 리프팅 변환은 3단계까지만 수행하였고 사용된 필터는 Daubechies 9/7 필터이다. Daubechies 9/7 리프팅 필터를 사용하는 웨이브렛의 평균 연산량은 각 계수당 4개이 덧셈과 2개의 곱셈 연산을 필요로 한다. 이것은 컨벌루션을 사용하는 9/7 쌍직교 필터를 사용하는 웨이브렛 변환이 평균적으로 8개의 덧셈과 4.5개의 곱셈 연산을 사용하는 것과 비교할 때 거의 1/2의 연산량을 갖게 된다¹³⁾. 움직임 추정에서 블록은 16X16 픽셀 블록으로 하였고 탐색영역은 [-16, 16]으로 하였다. 성능 평가를 위해 제안된 방식과 기존에 제시된 3가지 방식을 비교하여 실험하였다. 실험한 방법들은 다음과 같다.

- 방식 1: 표준인인 MPEG의 블록 정합 알고리즘 기반의 움직임 예측⁶⁾
- 방식 2: 웨이브렛 변환 기반의 저대역 이동법을 이용한 움직임 예측²⁾
- 방식 3: 리프팅 기반의 저대역 이동법을 이용한 움직임 예측
- 방식 4: 정수형 리프팅 기반의 저대역 이동법을 이용한 움직임 예측

표 1. 평균 PSNR [dB]

Method \ Test image	방식 1	방식 2	방식 3	방식 4
Akiyo.qcif	44.3	44.9	44.9	44.9
Claire.qcif	40.1	41.2	41.0	40.8
Carphone.qcif	30.5	30.3	29.7	29.8

표 1은 각 방식들의 평균 PSNR을 나타낸다. 이 표에 나타난 결과를 보면 본 논문에서 제안한 방식의 성능이 다른 방식들과 비슷하거나 다소 떨어지는 것을 볼 수 있다. 다음 표는 각 방식들의 사용된 메모리량을 비교한 표이다. 여기에서는 리프팅을 이용한 정수형 이동법을 기준으로 각 방식의 비를 나타내었다. 표 2에서는 우리가 제안한 정수형 리프팅을 이용한 저대역 부호화 방식이 실수 계수를 이용한 다른 방식들에 비해 메모리가 약 7배 이상 적게 사용되는 것을 알 수 있다. 방식 4에서 사용된 실험영상들은 8-비트로 양자화 과정을 수행하였다. 따라서 실수 형태로 처리되는 방식 3과 방식 2에 비해 사용되는 메모리의 양이 7~8배 정도 감소하게 된다. 정수형 데이터를 사용함으로써 인해 적은 비트

표 2. 각 방법들에 사용된 메모리량 비교

Method \ Test image	방식 4	방식 3	방식 2
Akiyo.qcif	1	8	8
Claire.qcif	1	8	8
Carphone.qcif	1	7	7

수로 양자화 과정을 수행하여도 영상의 품질 저하를 막을 수 있다.

다음의 영상은 실험 영상을 가지고 제안된 방법으로 움직임 예측과 보상을 하였을 때 PSNR이 가장 높은 영상과 가장 낮은 영상의 원본 영상과 움직임 보상한 영상이다.

• Akiyo 영상 중 PSNR이 가장 높은 영상의 원본 영상과 움직임 보상한 영상



그림 6. Akiyo 2번째 프레임 원영상

그림 7. 재생된 영상 PSNR = 54.01dB

• Akiyo 영상 중 PSNR이 가장 낮은 영상의 원본 영상과 움직임 보상한 영상

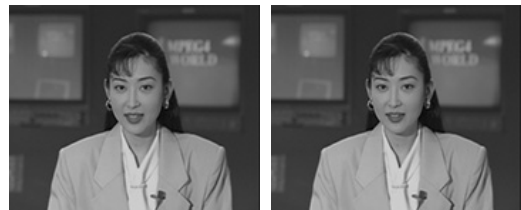


그림 8. Akiyo 27번째 프레임 원영상

그림 9. 재생된 영상 PSNR = 39.00dB

• Claire 영상 중 PSNR이 가장 높은 영상의 원본 영상과 움직임 보상한 영상



그림 10. Claire 12번째 프레임 원영상

그림 11. 재생된 영상 PSNR = 47.65dB

• Claire 영상 중 PSNR이 가장 낮은 영상의 원본 영상과 움직임 보상한 영상



그림 12. Claire 17번째 프레임원영상

그림 13. 재생된 영상 PSNR = 32.58dB

• Carphone 영상 중 PSNR이 가장 높은 영상의 원본 영상과 움직임 보상한 영상



그림 14. Carphone 5번째 프레임 원영상

그림 15. 재생된 영상 PSNR = 35.35dB

• Carphone 영상 중 PSNR이 가장 낮은 영상의 원본 영상과 움직임 보상한 영상



그림 16. Carphone 19번째 프레임 원영상

그림 17. 재생된 영상 PSNR=25.45dB

VI. 결론

웨이브렛 변환의 이동 변환 성질을 극복하기 위해 제안된 저대역 이동법은 연산량이 너무 많고 메모리 사용량이 너무 많은 단점을 보였다. 이를 보완하기 위해 본 논문에서는 웨이브렛에 비해 연산량이 적고 메모리 사용량이 감소하는 정수형 리프팅 기반의 저대역 이동법을 제안하였다. 정수형 리프팅 기반의 저대역 이동법을 가지고 움직임 추정과 움직임 보상을 실행해 본 결과 PSNR은 웨이브렛에 비해 다소 떨어지지만 리프팅 기반의 저대역 이동법과는 거의 비슷함을 보였다. 우리가 사용한 Daubechies 9/7 필터를 이용한 리프팅은 웨이브렛

변환의 계산량을 절반으로 줄일 수 있고 대체 연산이 가능하기 때문에 사용되는 메모리도 줄일 수 있다. 정수형 리프팅 변환은 변환 과정에서 반올림 과정에 의해 오류성분이 포함되는 변환과정이다. 따라서 변환계수의 에러에 매우 민감한 특징이 있다. 하지만, 정수형 리프팅은 일반적인 리프팅에 비해 소숫점 연산이 아닌 정수 연산을 실행하기 때문에 사용된 메모리량이 실수형에 비해 약 1/8 정도이다. 그러므로 정수형 리프팅을 이용한 저대역 이동법은 동영상 부호화에서 중요시 하는 연산의 속도 및 메모리의 활용 측면에서 웨이브렛은 물론이고 리프팅 변환에 비해서도 장점이 있다. 이러한 점에서 정수형 리프팅 기반의 저대역 이동법을 이용한 움직임 추정과 보상은 효과적이다.

참고 문헌

- [1] Y. Wang, J. Ostermann and Y. Zhang, "Video processing and communications", Prentice Hall, 2002.
- [2] H. Park and H. Kim, "Motion estimation using Low-Band-Shift Method for Wavelet-Based Moving-Picture Coding", IEEE Transactions on Image Processing, Vol. 9, Issue 4, pp.577-587, April, 2000.
- [3] Daubechies, I. and W. Sweldens, "Factoring Wavelet transforms into lifting steps," Asilomar Conference on Signals, Systems, and Computers, 1997.
- [4] Iain E. G. Richardson, "H.264 and MPEG-4 video compression", Wiley, 2003.
- [5] 조재만, 김현민, 고희화 "웨이브렛 변환 영역에서 저대역 이동법에 적합한 다해상도 움직임 추정에 관한 연구" 한국통신학회논문지, Vol. 29, No. 2, 2003.
- [6] F. Pereira and T Ebrahimi, "The Mpeg-4 Book", Prentice Hall, N.J., 2002.
- [7] M. Vetterli, "Wavelet and Filter Banks: Theory and Design," IEEE Trans. on Signal processing. Vol. 40, No 9. pp. 2207-2232, Sept. 1992.
- [8] Y. Q. Zhang, S. Zafar, "Motion-compensated wavelet transform coding for color video compression," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 2, No. 3, pp. 285-296, Sept. 1992.

- [9] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Trans. Signal Process., vol. 41, no. 12, pp. 3445-3462, Dec. 1993.
- [10] W. Sweldens, "The lifting scheme : A construction of second generation wavelets," SIAM J. Math. Anal, Vol. 29, No. 2, 1997.
- [11] Jonghong Shin, Innho Jee, "Efficient motion estimation in the wavelet transform domain for moving image codecs," Proc. of KSPC, Vol. 12, No. 1, pp. 785-788, Sept 1999.
- [12] Jonghong Shin, Sangjun Kim, and Innho Jee, "A Fast Multiresolution Motion Estimation Algorithm in the Adaptive Wavelet Transform Domain," Journal of Broadcast engineering, Vol. 7, No. 1, pp. 55-65, June 2002.
- [13] Lin Luo, YunnanJin Li and Ya-Qin Zhang, "3-D Wavelet Compression and Progressive Inverse Wavelet Synthesis Rendering of Concentric Mosaic," IEEE Trans. on Image Processing, Vol. 11, pp. 802- 816, July 2002.

백 정 은 (Jung-eun Baek)

준회원



2004년 2월 홍익대학교 컴퓨터
정보통신 공학과 졸업
2006년 2월 홍익대학교 전자전
산공학과 석사
<관심분야> 영상처리, MPEG

장 선 봉 (Sun-bong Jang)

정회원



1999년 2월 홍익대학교 컴퓨터
정보통신공학과 졸업
2001년 2월 홍익대학교 전자전
산공학과 석사
2003년 2월~현재 홍익대학교 전
자공학과 박사과정
<관심분야> 이동통신, 3D Image
processing, DTV

신 종 홍 (Jong-hong Shin)

정회원



1997년 2월 홍익대학교 전기공
학과 졸업
1999년 2월 홍익대학교 전기공
학과 석사
2002년 6월 홍익대학교 전기공
학과 박사
2003년 2월~현재 한국사이버대

학교 교수

<관심분야> 이동통신, 영상처리

지 인 호 (Inn-ho Jee)

종신회원



1980년 2월 서울대학교 전자공
학과 졸업
1983년 2월 서울대학교 전자공
학과 석사
1995년 Polytechnic Univ (New
York City), 전기 및 컴퓨터공
학과 공학박사

1982년~1988년 국방과학연구소 선임연구원

2004년~2005년 Dept. of Electrical and Computer
Engineering, University of Maryland at College
Park, USA. 연구교수

1995년~현재 홍익대학교 컴퓨터정보통신공학과 교수

<관심분야> CDMA/OFDM, 3D Image Processing,
MPEG, Multimedia security