

하다마드 변환의 적응적 적용을 이용한 고속 움직임 예측 알고리즘

준회원 이 혁*, 정회원 김 중 호*, 준회원 진 순 중*, 정회원 정 제 창*

Fast Motion Estimation Algorithms Through Adaptive Application of the Hadamard Transform

Hyuk Lee* *Associate Member*, Jongho Kim* *Regular Member*,
Soonjong Jin* *Associate Member*, Jechang Jeong* *Regular Member*

요 약

본 논문에서는 4x4 단위의 하다마드 변환을 이용한 새롭고 효과적인 움직임 예측 알고리즘을 제안한다. 하다마드 변환은 덧셈과 뺄셈으로만 이루어져 있기 때문에 고속 알고리즘에 적합하다는 장점이 있다. 제안하는 알고리즘은 세 단계로 이루어져 있다. 먼저 하다마드 DC (Direct Current) 계수를 이용하여 움직임 추정의 조기종결여부를 블록의 특성에 적응적으로 적용하고 이전 프레임의 움직임 벡터를 이용할지를 결정한 다음, 하다마드 AC (Alternating Current) 계수를 이용하여 영상의 복잡도를 구하여 정합 스캔 순서를 적응적으로 결정한다. 마지막으로 제안하는 알고리즘을 MVFAST와 PMVFAST 알고리즘에 적용했다. 모의 실험 결과 제안하는 알고리즘은 계산 속도와 PSNR (Peak Signal-to-Noise Ratio) 면에서 MVFAST와 PMVFAST 알고리즘에 비해 매우 효율적임을 나타냈다.

Key Words : Fast Motion Estimation, Hadamard Transform, Adaptive Matching Scan, MPEG-4 Video

ABSTRACT

In this paper, we propose a new, effective, fast motion estimation algorithms using 4x4 pixels Hadamard transform. The Hadamard transform has the advantage of simplicity because it uses only addition and subtraction. Motion estimation is composed of three stages. First, it should be decided whether to terminate the search early and use a previous motion vector with DC (Direct Current) coefficients. Then the adaptive matching scan order for motion estimation should be determined according to the image complexity using AC (Alternating Current) coefficients. Experimentally, we adapted this algorithms to MVFAST and PMVFAST algorithms, and the proposed algorithms turns out to be very efficient in terms of computational speed while remaining almost the same in terms of PSNR (Peak Signal-to-Noise Ratio) compared to MVFAST and PMVFAST algorithms.

I. 서 론

컴퓨터와 통신망의 발달로 인해 디지털 영상 매체의 서비스가 급속하게 발전하고, 소비자들에게 VOD

(Video on Demand) 서비스나 화상 전화 등의 서비스가 보편적으로 제공되고 있다. 최근에는 디지털 방송 분야에서도 HD 방송이나 DMB (Digital Multimedia Broadcasting) 등의 서비스가 사용자들

* 본 연구보고서는 정보통신부 출연금으로 ETRI, SoC산업진흥센터에서 수행한 IT SoC 핵심설계인력양성사업의 연구결과입니다.

* 한양대학교 전자통신컴퓨터공학과 영상통신 및 신호처리 연구실(yomuto@ece.hanyang.ac.kr)

논문번호 : KICS2007-04-162, 접수일자 : 2007년 4월 6일, 최종논문접수일자 : 2007년 7월 10일

에게 제공되고 있다. 이러한 디지털 영상 통신 서비스를 위해서, 제한된 채널을 통해 더 많은 영상 데이터를 안정적으로 전송할 수 있는 기술이 요구되는데 그 핵심기술이 동영상 압축 기술이고, 현재 MPEG, H.26x와 같은 동영상 압축 표준들이 제정되어 있다. 동영상 압축의 목표는 더 적은 데이터량으로 더 나은 화질을 구현하고, 다양한 응용을 위한 부/복호시 복잡도 감소에 있다.

동영상 압축에 있어 데이터량을 획기적으로 줄이는 기술 중 하나인 움직임 예측은 현재 프레임의 매크로블록과 가장 유사한 이전 프레임의 매크로블록을 찾아 그 움직임 정보만을 부호화 하는 방법이다. 움직임 예측의 방법으로 가장 널리 쓰이고 있는 방법은 전역 탐색 방법 (FS; Full Search)이다. 전역 탐색 방법은 주어진 탐색 영역 내의 모든 후보 지점에서 정합 에러를 갖는 위치를 찾는 방법이다. 전역 탐색 방법은 모든 후보 지점을 탐색하기 때문에 최소 정합 에러를 갖는 위치를 가장 정확하게 찾아내고, 하드웨어로의 구현이 간편하다는 장점을 지닌다. 그러나 전역 탐색 방법은 그 계산량이 막대하다는 단점을 지니고 있고, 이는 동영상의 실시간 부호화에 큰 어려움을 주고 있다. 이로 인하여 다양한 고속 움직임 예측 알고리즘들이 개발되어 왔다 [11-14].

본 논문에서는 영상의 화질을 열화 시키지 않으면서 고속 움직임 예측을 위해서 [5]의 내용을 기초로 하여 4x4 블록 단위의 하다마드 변환을 이용한 알고리즘을 제안한다. 즉, 이전 프레임의 움직임 벡터를 이용하고 별도의 움직임 탐색을 종료할 것인지 아닌지의 판단을 위해서 4x4 블록 단위 하다마드 변환의 DC 계수를 이용한다. 별도의 움직임 탐색을 수행해야 할 경우, AC 계수를 이용한 영상의 복잡도를 이용하여 움직임 예측을 위해 계산되는 행의 수를 줄인다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 고속 움직임 예측 방법들에 대해 간단히 살펴보고, III장에서는 제안하는 알고리즘에 대해 자세히 기술한다. IV장에서 실험 결과를 기존의 고속 움직임 예측 방법들과 비교하고, V장에서 결론을 맺는다.

II. 기존의 고속 움직임 예측 방법

2.1 전역 탐색 방법

전역 탐색 방법이란 움직임 예측을 할 때 주어진 탐색 범위 내의 모든 탐색점에서의 정합 에러를 검

사하여 최소의 에러를 가지는 점을 선택하는 방법이다. 정합 에러를 구할 때 PDE (Partial Distortion Elimination) 방법을 통해 계산량을 줄일 수 있는데, PDE는 정합 에러의 부분합을 이용하여 블록의 정합 에러를 전부 계산하기 전에 미리 불가능한 후보들을 제거하는 방법이다. 이러한 PDE 방식을 적용하면 소위 나선 구조 (Spiral Structure)로 움직임 추정을 실행할 때 성능이 극대화 된다. 나선 구조는 대부분의 자연 영상의 움직임 벡터가 원점 근처에 위치하기 때문에 탐색할 영역의 중심부터 원을 그리듯이 움직임 벡터를 추정하므로 빠른 속도로 움직임 추정을 할 수 있다. 즉 초기에 구한 정합 에러가 최소 정합 에러가 될 가능성이 높기 때문에 전체 탐색 범위에서 정합 에러 계산량을 많이 줄일 수 있다.

2.2 영상 복잡도와 다양한 매칭 스캔을 이용한 고속 전역 움직임 예측 알고리즘

전역 탐색 방법과 탐색점의 개수는 동일하지만 각 탐색점에 대해 정합 에러를 계산할 때 계산량을 줄여 고속 탐색을 구현하기 위하여 [5]에서 영상 복잡도와 다양한 매칭 스캔을 이용한 고속 전역 움직임 예측 알고리즘을 제안하였다.

테일러급수 전개 방법을 이용한 정합 에러와 기준 블록, 즉 현재 부호화 하려고 하는 블록의 기울기 크기와의 관계를 정리할 수 있다. t 번째 프레임의 위치 $p = (x, y)$ 에서의 화소값을 $f_t(x, y)$, 움직임 벡터를 $mv = (mv_x, mv_y)$ 라 하면, 움직임 보상에 의해 재구성된 $(t+1)$ 번째 프레임은 식 (1)과 같이 표현할 수 있다.

$$f_{t+1}(x, y) = f_t(x + mv_x, y + mv_y) \quad (1)$$

또한, 각 후보 벡터 $cmv = (cmv_x, cmv_y)$ 에 대해서 정합 에러를 $d_{t+1}(x, y)$ 라 하면 다음과 같은 식이 성립한다.

$$\begin{aligned} d_{t+1}(x, y) &= |f_{t+1}(x, y) - f_t(x + cmv_x, y + cmv_y)| \\ &= |f_t(x + mv_x, y + mv_y) - f_t(x + cmv_x, y + cmv_y)| \\ &\approx \left| \frac{\partial f_t(x + mv_x, y + mv_y)}{\partial x} (mv_x - cmv_x) \right. \\ &\quad \left. + \frac{\partial f_t(x + mv_x, y + mv_y)}{\partial y} (mv_y - cmv_y) \right| \\ &\approx \left| \frac{\partial f_{t+1}(x, y)}{\partial x} (mv_x - cmv_x) \right| \\ &\quad + \left| \frac{\partial f_{t+1}(x, y)}{\partial y} (mv_y - cmv_y) \right| \end{aligned} \quad (2)$$

식 (2)로부터, $p = (x, y)$ 에서의 정합 에러는 현재 프레임의 기준 블록의 기울기 크기에 비례한다는 사실을 알 수 있다. 즉, 영상의 복잡도에 따라 정합 에러가 비례한다는 것을 의미한다.

이제 영상의 복잡도에 따라 정합 에러가 비례한다는 사실을 토대로 PDE 정합 에러를 계산할 정합 스캔 순서를 결정한다. 먼저 16x16 단위의 매크로블록을 4x4 단위의 서브블록으로 나눈 뒤 각각의 서브블록의 화소당 복잡도를 식 (3)과 같이 구한다.

$$C_t = |f_t(x, y) - f_t(x - 1, y)| + |f_t(x, y) - f_t(x, y - 1)| \quad (3)$$

$S_{m,n}$ 을 (m, n) 에 위치하는 서브블록의 복잡도라고 하면, 식 (3)으로부터 $S_{m,n}$ 는 다음과 같이 구한다.

$$S_{m,n} = \sum_{x=1}^4 \sum_{y=1}^4 C_t(x, y) \quad (4)$$

식 (4)에서 구한 서브블록 단위의 복잡도 순서대로 PDE 정합 순서를 재정렬한다. 이렇게 4x4 단위의 서브블록 단위의 정합 순서로 정합 에러를 계산하면, 영상의 복잡도를 더 잘 영역화할 수 있다. 영상의 복잡도를 잘 영역화하면 PDE 정합 에러가 최소 정합 에러에 다다르는 시간을 앞당길 수 있어 불필요한 계산을 더 많이 줄일 수 있다. 이 알고리즘은 전역 탐색 방법의 나선 구조 움직임 추정에 비해 계산량을 더욱 줄일 수 있다.

2.3 MVFAST와 PMVFAST

MVFAST 알고리즘은 MPEG-4에서 고속 움직임 예측을 위한 핵심 기술로 권고하고 있다⁶⁾⁸⁾. MVFAST와 PMVFAST는 전역 탐색 방법이 가지는 엄청난 계산량을 줄이는 동시에 거의 동등한 수준의 영상 화질을 유지한다.

MVFAST 알고리즘은 초기 탐색점으로 (0, 0)을 검사한다. 만약 (0, 0)에서의 SAD (Sum of Absolute Difference) 값이 임계치 T보다 작다면, 움직임 탐색을 중지하고 움직임 벡터를 (0, 0)으로 사용한다. 이를 움직임 탐색의 조기 종결 (Early Termination) 조건이라고 한다. 움직임 탐색이 단 하나의 탐색점으로 끝날 수 있다는 점이 MVFAST 알고리즘이 지니는 가장 강력한 장점이다. 만약 움직임 벡터가 (0, 0)으로 정해지지 않는다면, 움직임 활동도 (Motion Activity)를 계산한다. MVFAST는 매크로블록의 움직임의 종류를 구분하기 위하여 움직임 활동도를 높음, 중간, 낮음의 세 가지로 정의한다. 움

직임 활동도는 초기 탐색 중심점과 탐색 전략을 결정한다. 만약 움직임 활동도가 낮거나 중간이면, 탐색 중심점은 원점이 된다. 그렇지 않다면, 최소의 SAD를 이끌어내는 집합 V에 속하는 벡터가 초기 탐색 중심점으로 선택된다. V는 상, 우상, 좌에 위치하는 매크로블록의 움직임 벡터들을 포함한다. 움직임 활동도가 낮거나 높다면, 탐색 전략은 SDS (Small Diamond Search)가 된다. 움직임 활동도가 중간이라면, 탐색 전략은 LDS (Large Diamond Search)로 한다.

PMVFAST는 MVFAST에서 파생된 알고리즘이다. PMVFAST는 MVFAST에서 몇 개의 임계치들을 혼합하여 메모리의 크기와 대역폭, 부가적인 복잡도에 있어서 손해를 보는 대신 속도를 향상시키는 알고리즘이다.

III. 제안하는 알고리즘

3.1 하다마드 변환

H.264/AVC에서도 이용하고 있는 하다마드 변환은 오직 덧셈과 뺄셈으로만 이루어진 가장 간단한 직교 변환이다.

$n \times n$ 하다마드 행렬은 식 (5)를 만족하는 1과 -1만을 가지는 행렬 H_n 이다¹⁰⁾.

$$H_n H_n^T = I_n \quad (5)$$

2x2 하다마드 행렬은 식 (6)과 같다. 이를 핵심 행렬 (Core Matrix)이라고 한다.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (6)$$

확장된 하다마드 행렬은 H_2 를 크로네커 곱 (Kronecker product)을 이용하여 만든다. 즉, H_{2n} 은 다음 식에 의해 구한다.

$$H_{2n} = H_2 \otimes H_n \quad (7)$$

식 (7)에서 \otimes 는 크로네커 곱을 의미하는 기호이다. 식 (7)에 의해 본 논문에서 이용하는 4x4 하다마드 행렬은 식 (8)과 같이 구할 수 있다.

$$H_4 = H_2 \otimes H_2 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (8)$$

현재 하다마드 계수를 구하고자 하는 서브블록을 S 라고 하면 그 블록에 대한 하다마드 계수 F 는 식 (9)과 같이 구한다.

$$F = H_4 S H_4^T = H_4 S H_4 \quad (9)$$

Hermitian을 취하지 않은 이유는 하다마드 행렬이 실수로 이루어져 있고, 하다마드 행렬에 전치를 취한 결과가 자기 자신과 같기 때문이다. 이러한 성질은 하다마드 변환이 널리 쓰이는 또 다른 이유이기도 하다.

3.2 하다마드 AC 계수를 이용한 영상 복잡도의 정의

[5]의 고속 전역 움직임 예측 방법에서는 16x16 단위의 매크로블록을 4x4 단위의 서브블록으로 나눈 뒤 각 서브블록에 대해 복잡도를 식 (4)를 이용하여 구했다. 제안하는 알고리즘에서는 4x4 단위의 서브블록에 대해 하다마드 변환을 수행하고, 그 AC 계수를 이용하여 영상의 복잡도를 판단한다. 즉, 각 서브블록의 가장 좌측 상단의 화소 위치를 (m, n) 이라고 할 때 하다마드 변환된 4x4 단위의 서브블록의 복잡도 $Sh_{m,n}$ 을 다음과 같이 정의한다.

$$Sh_{m,n} = \sum_{i=1}^{15} |AC_i| \quad (10)$$

식 (10)에서 AC_i 는 4x4 서브블록의 각 AC계수이다. 그림 1은 4x4 단위의 서브블록의 하다마드 변환 계수의 위치를 그림으로 나타낸 것이다. 그림 1에서 하나하나의 사각형은 각각의 하다마드 변환 계수를 나타내고, 가장 큰 사각형은 하나의 서브블록을 나타낸다.

DC	AC ₁	AC ₂	AC ₃
AC ₄	AC ₅	AC ₆	AC ₇
AC ₈	AC ₉	AC ₁₀	AC ₁₁
AC ₁₂	AC ₁₃	AC ₁₄	AC ₁₅

그림 1. 4x4 단위의 서브블록의 하다마드 변환 계수의 위치

식 (4)에서 구한 영상의 복잡도와 식 (10)에서 제안하는 영상의 복잡도는 이론적으로 큰 차이가 없기 때문에 거의 비슷한 효율을 보인다. 그렇지만 본 논문에서 제안하는 알고리즘은 DC 계수를 효율적으로 이용할 수 있는 가능성을 제공한다는 장점이 있다.

또한 본 논문에서는 한 매크로블록 내에 존재하는 16개의 $Sh_{m,n}$ 의 합을 $ACsum$ 이라고 정의한다. 즉 $Sh_{m,n}$ 가 한 서브블록에서 영상의 복잡도를 나타냈다면, $ACsum$ 은 한 매크로블록에서 영상의 복잡도를 나타낸다. $ACsum$ 은 이후에 현재의 매크로블록에 대한 움직임 예측을 할 것인가 하지 않을 것인가에 대한 판단을 내릴 때 이용한다.

3.3 하다마드 변환의 DC 계수를 이용한 시간적 블록 오프셋 검출

장면 전환을 검출할 수 있는 기준으로 [11]에서는 $MAFD$ (Mean Absolute Frame Differences)를 제안하고 있다. $M \times N$ 크기의 n 번째 프레임에서의 $MAFD$ 는 식 (11)과 같이 구한다.

$$MAFD_n = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f_n(x,y) - f_{n-1}(x,y)| \quad (11)$$

여기서, $f_n(x,y)$ 는 (x,y) 위치에서의 화소의 휘도를 나타낸다. 하지만 $MAFD$ 는 프레임 전체를 위한 장면 전환 검출 기준이고, 복잡도가 매우 크기에 제안하는 알고리즘에서는 사용하지 않는다. 대신, 본 논문에서는 DC 계수를 이용하여 매크로블록 단위로 영상의 시간적 블록 오프셋 (Temporal Block-offset)을 이용하는 방법을 제안한다. 즉, 매크로블록에 포함된 4x4 단위의 서브블록의 DC 계수를 더한 $DCsum$ 을 식 (12)에 의해서 구한다.

$$DC_{sum}^{(n)} = \sum_{i=1}^{16} DC_i \quad (12)$$

여기서, DC_i 는 매크로블록 내의 각 서브블록의 DC 계수를 의미하고, $DC_{sum}^{(n)}$ 은 n 번째 프레임에 대한 $DCsum$ 을 의미한다.

식 (12)에서 구한 $DCsum$ 을 이용하여 제안하는 시간적 블록 오프셋 $offset_{TB}$ 는 식 (13)과 같이 구한다.

$$offset_{TB}^{(n)}(p,q) = |DC_{sum}^{(n)}(p,q) - DC_{sum}^{(n-1)}(p,q)| \quad (13)$$

여기서, $offset_{TB}^{(n)}(p,q)$ 은 n 번째 프레임의 (p,q) 위치의 매크로블록에 대한 시간적 블록 오프셋을 의미하고, $DC_{sum}^{(n)}(p,q)$ 는 n 번째 프레임의 (p,q) 위치의 매크로블록에 대한 DC_{sum} 을 의미하며 식 (12)에 의해서 구한다.

이렇게 DC 계수를 이용하여 시간적 블록 오프셋을 구하면 픽셀 단위로 장면 전환을 검출할 때와 비교하여 정확도는 다소 떨어지게 되지만, 복잡도가 훨씬 감소하는 장점이 있다.

3.4 시간적 블록 오프셋을 이용한 조기 움직임 탐색 중지 (Early Termination) 방법

그림 2는 제안하는 알고리즘을 순서도로 나타낸 것이다. 먼저 현재 프레임의 매크로블록을 4x4 단위로 나눈 뒤, 하다마드 변환을 통해 영상의 복잡도를 구한다. 그 후 현재 매크로블록에서의 $offset_{TB}$ 가 임계치 α 보다 작으면 움직임 예측 과정이 생략되고 이전 프레임의 같은 위치의 매크로블록의 움직임 벡터를 그대로 사용한다. 이는 그림 2에서의 Terminate를 의미한다. 만약 현재 매크로블록의 AC_{sum} 이 임계치 β_1 (예를 들면 4096)보다 크면 그 매크로블록은 복잡한 블록으로 판단하고, α 는 작은 값 (예를 들면 16)으로 설정한다. 만약 AC_{sum} 이 β_2 (예를 들면 1024)보다 작으면, 그 매크로블록은 평탄한 블록으로 판단하고, α 는 큰 값 (예를 들면 128)으로 설정한다. 그 외의 경우에는 α 를 중간 값 (예를 들면 64)으로 설정한다. 이 조건들을 만족하지 않으면

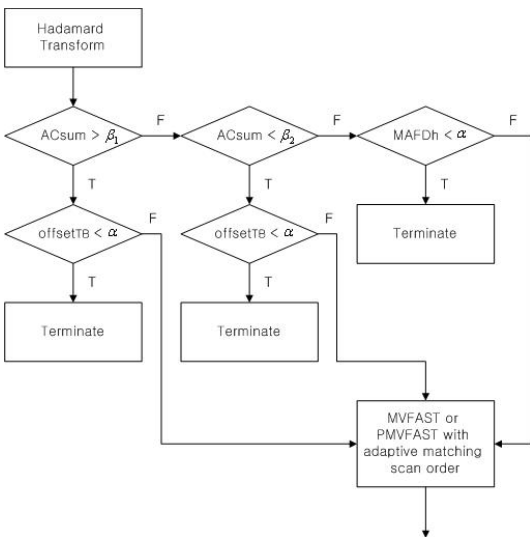


그림 2. 제안하는 알고리즘의 순서도

MVFAST나 PMVFAST 알고리즘을 수행한다. 즉 MVFAST나 PMVFAST 알고리즘을 이용하여 탐색하려는 점을 찾고, 블록 정합 에러는 $Sh_{m,n}$ 의 크기순으로 구하여 PDE 정합 에러가 최소 정합 에러에 다른 시간을 앞당긴다.

이 알고리즘은 영상이 크게 변하지 않는 경우를 검출하여 이전 움직임 벡터를 그대로 사용함으로써 화질 손상이 거의 없이 움직임 예측 시간을 크게 단축시킨다. 이렇게 이전 프레임과 차이가 적은 경우 움직임 추정 과정을 생략하여 계산량 감소를 취하는 방법은 다소 부정확한 움직임 벡터를 찾게 만든다는 단점이 있지만, 이전 프레임의 같은 위치의 움직임 벡터를 사용함으로써 움직임 벡터 간의 상관도를 증가시켜 움직임 벡터를 부호화하는 데 드는 비트를 줄인다. 그렇기 때문에 제안하는 방법을 적용한 부호화 후의 화질 열화는 최소화된다.

IV. 실험 결과

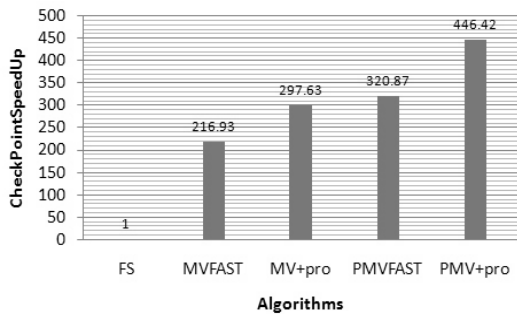
성능평가를 위하여 MPEG-4 비디오 부호화가 참조 소프트웨어에 제안하는 알고리즘을 구현하고 이를 기존의 움직임 예측 알고리즘들과 비교하였다. 율-제어 알고리즘은 TM5를 사용하고, 실험 조건과 테스트 시퀀스는 MPEG-4에서 권고하는 실험 조건 [12]을 따랐다. 전역 탐색 방법에서는 나선 추정 구조를 이용하였다. 표 1에 제안하는 알고리즘의 실험 결과를 나타내었다.

표 1에서 BR은 Bitrate (kbps), SR은 Search Range를 나타낸다. CheckPoint는 탐색점의 개수를 나타낸다. LineSAD란 SAD가 계산된 행의 수를 나타낸다. 즉 LineSAD는 PDE 방법을 사용하여 블록의 SAD를 계산할 때 현재의 SAD가 최소 SAD보다 클 경우 조기 중단이 얼마나 일찍 되는가를 나타낸다. 조기 중단이 먼저 일어날수록 CheckPoint의 개수에는 변화가 없지만 LineSAD는 더 작아진다. CheckPointSpeedUp은 전역 탐색 방법과 제안하는 알고리즘과의 CheckPoint 비율을 의미하고, LineSADSpeedUp은 LineSAD 비율을 의미한다. FS는 전역 탐색 방법, MVFAST는 MVFAST 알고리즘, PMVFAST는 PMVFAST 알고리즘을 나타낸다. FS+HT는 제안하는 하다마드 변환의 AC 계수를 이용하여 적응적으로 정합 스캔 순서를 정한 전역 탐색 방법을 의미한다. MV+pro는 제안하는 알고리즘을 MVFAST 알고리즘에 적용한 결과를 나타내고, PMV+pro는 PMVFAST 알고리즘에 적용한 결과를

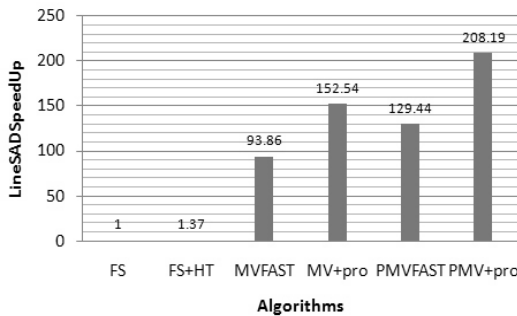
나타낸다.

표 1에서 볼 수 있듯이 FS와 FS+HT 사이의 CheckPoint 차이는 없다. 그렇지만 FS+HT가 LineSAD 측면에서 FS에 비해 이득을 보고 있다. 이것은 제안하는 알고리즘이 최소 SAD에 더욱 일찍 다다를 수 있음을 의미한다.

그림 3은 News 시퀀스에서 BR이 112kbps, SR이 16일 때의 CheckPointSpeedUp과 LineSADSpeedUp을 도표로 나타낸 것이다. 이는 제안하는 알고리즘이 FS에 비해 속도향상이 얼마나 이루어지는지를 나타낸다.



(a) CheckPointSpeedUp



(b) LineSADSpeedUp

그림 3. News, BR 112kbps, SR 16일 때의 각 알고리즘에 대한 복잡도의 비교

실험 결과에서 알 수 있듯이 전체적인 PSNR은 전역 탐색 방법과 비교했을 때 큰 차이가 없다. 즉, 제안하는 알고리즘은 영상의 화질을 크게 열화시키지 않는다. 이런 결과는 크기가 작은 영상과 큰 영상, 저비트율과 고비트율에 공통적으로 나타난다. 제안하는 알고리즘의 전체적인 SpeedUp은 MVFAST나 PMVFAST에 비해 언제나 더 향상됨을 알 수 있다. 때로는 제안하는 알고리즘의 PSNR이 전역 탐색 방법이나 MVFAST, PMVFAST에 비해 나은 경우가 있는데, 이는 움직임 벡터 간의 상관도가 증가하여 움직임 벡터를 부호화하는 데 드

는 비트가 줄었기 때문이다.

이러한 결과는 제안하는 알고리즘이 하다마드 변환의 DC 계수를 이용하여 효과적으로 장면 전환을 검출하는데 성공했다는 것과 AC 계수를 이용하여 영상의 복잡도를 잘 나타내어 정합 스캔 순서를 효과적으로 바꾸는데 성공했다는 것을 의미한다. 특히 DC 계수를 이용한 알고리즘은 CheckPoint를 감소하는데, AC 계수를 이용한 알고리즘은 LineSAD를 감소하는데 효과적이었다. 이러한 시도는 때로 전역 탐색 방법에 비해 잘못된 움직임 벡터를 찾는 결과를 가져오기도 하지만, 움직임 벡터들의 상관도를 높이기도 하여 전체적인 영상의 화질은 거의 일정하게 유지한다.

V. 결론

본 논문에서는 하다마드 변환을 이용하여 영상 복잡도를 재정의 하고, 장면 전환을 검출하는 고속 움직임 예측 알고리즘을 제안하였다. 실험 결과, 제안하는 알고리즘은 전역 탐색 방법과 MVFAST, PMVFAST 알고리즘에 비해 CheckPoint와 LineSAD 두 측면에서 큰 속도 향상을 보이면서 PSNR 감소는 미미함을 보인다. 제안하는 알고리즘은 현존하는 디지털 방송이나 DMB 등을 위한 비디오 부호화에 유용하게 사용할 수 있을 것이다. 그리고 제안하는 알고리즘은 본 논문의 실험에 사용된 MPEG-4 비디오 부호화기 이외에도 H.264/AVC나 차세대 동영상 압축 표준인 SVC 등의 비디오 부호화기에도 효과적으로 사용할 수 있을 것이다. 특히 H.264/AVC를 기반으로 하고 있는 동영상 압축 표준은 4x4 단위의 하다마드 변환을 사용하고 있어 하다마드 변환기 하드웨어의 부담을 덜 받을 수 있다는 장점이 있다.

참고 문헌

- [1] W. Li and E. Salari, "Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105-107, Jan 1995.
- [2] Renxiang Li, Bing Zeng and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438-442, Aug, 1994.

표 1. 실험 결과.

Sequence	Frame	BR	SR		FS	FS+HT	MVFAST	MV+pro	PMVFAST	PMV+pro	
Container QCIF	300	112	16	PSNR(dB)	27.67	27.67	27.98	27.92	27.96	27.91	
				CheckPointSpeedUp	1.00	1.00	218.96	256.78	285.33	344.06	
				LineSADSpeedUp	1.00	1.27	110.18	147.02	140.85	189.11	
			32	PSNR(dB)	27.84	27.84	27.903	27.82	27.85	27.73	
				CheckPointSpeedUp	1.00	1.00	753.08	886.69	1007.49	1225.69	
				LineSADSpeedUp	1.00	1.24	294.44	392.95	389.32	516.58	
Mother&Daughter QCIF	300	24	16	PSNR(dB)	33.73	33.73	33.54	33.62	33.64	33.61	
				CheckPointSpeedUp	1.00	1.00	208.21	233.18	298.11	337.56	
				LineSADSpeedUp	1.00	1.30	67.63	85.86	92.32	114.69	
			32	PSNR(dB)	33.65	33.65	33.60	33.63	33.70	33.59	
				CheckPointSpeedUp	1.00	1.00	724.00	805.32	1039.23	1168.74	
				LineSADSpeedUp	1.00	1.22	191.59	241.33	261.76	321.78	
Coast-guard QCIF	300	48	16	PSNR(dB)	28.52	28.52	28.51	28.51	28.51	28.48	
				CheckPointSpeedUp	1.00	1.00	101.82	103.03	158.41	160.23	
				LineSADSpeedUp	1.00	1.14	56.47	62.46	83.99	90.95	
			32	PSNR(dB)	28.49	28.49	28.53	28.50	28.52	28.48	
				CheckPointSpeedUp	1.00	1.00	351.18	358.99	539.97	549.26	
				LineSADSpeedUp	1.00	1.12	169.73	188.99	249.20	270.84	
Coast-guard CIF	300	112	16	PSNR(dB)	26.95	26.95	26.98	26.98	26.97	26.99	
				CheckPointSpeedUp	1.00	1.00	98.20	99.24	111.25	113.62	
				LineSADSpeedUp	1.00	1.11	60.05	63.89	68.39	73.62	
			32	PSNR(dB)	26.97	26.97	27.00	27.00	27.03	27.01	
				CheckPointSpeedUp	1.00	1.00	365.45	368.93	416.88	423.23	
				LineSADSpeedUp	1.00	1.11	198.74	211.18	228.35	244.42	
News CIF	300	48	16	PSNR(dB)	29.62	29.62	29.72	29.66	29.55	29.67	
				CheckPointSpeedUp	1.00	1.00	171.96	224.30	247.19	302.64	
				LineSADSpeedUp	1.00	1.33	80.62	117.95	111.11	153.00	
			32	PSNR(dB)	29.67	29.67	29.77	29.61	29.65	29.70	
				CheckPointSpeedUp	1.00	1.00	629.83	810.87	916.33	1077.70	
				LineSADSpeedUp	1.00	1.29	227.22	328.00	318.03	417.79	
		112	16	PSNR(dB)	32.06	32.06	32.10	32.07	31.88	31.84	
				CheckPointSpeedUp	1.00	1.00	216.93	297.63	320.87	446.42	
				LineSADSpeedUp	1.00	1.37	93.86	152.54	129.44	208.19	
			32	PSNR(dB)	32.10	32.10	32.11	31.98	32.01	31.93	
				CheckPointSpeedUp	1.00	1.00	799.94	1100.24	1186.89	1650.92	
				LineSADSpeedUp	1.00	1.31	265.70	431.38	368.51	589.79	
Foreman CIF	300	112		PSNR(dB)	29.32	29.32	29.27	29.24	29.26	29.25	
				CheckPointSpeedUp	1.00	1.00	84.52	85.79	103.33	104.62	
				LineSADSpeedUp	1.00	1.17	44.75	48.56	55.23	59.47	
				PSNR(dB)	29.53	29.53	29.43	29.43	29.45	29.43	
				CheckPointSpeedUp	1.00	1.00	299.58	304.93	367.42	372.91	
				LineSADSpeedUp	1.00	1.17	124.13	135.08	154.41	166.97	
			512		PSNR(dB)	34.97	34.97	34.95	34.95	35.00	35.00
					CheckPointSpeedUp	1.00	1.00	98.51	100.90	139.87	142.77
					LineSADSpeedUp	1.00	1.21	44.90	51.17	64.29	72.50
				PSNR(dB)	35.17	35.17	35.11	35.10	35.17	35.16	
				CheckPointSpeedUp	1.00	1.00	360.01	368.60	516.01	526.36	
				LineSADSpeedUp	1.00	1.20	126.99	145.23	185.48	210.68	
		1024		PSNR(dB)	35.87	35.87	35.84	35.84	35.89	35.88	
				CheckPointSpeedUp	1.00	1.00	127.71	131.96	205.57	214.26	
				LineSADSpeedUp	1.00	1.26	52.44	62.48	82.96	97.96	
				PSNR(dB)	35.91	35.91	35.87	35.87	35.91	35.92	
				CheckPointSpeedUp	1.00	1.00	470.22	486.72	755.59	787.49	
				LineSADSpeedUp	1.00	1.23	150.89	180.11	238.82	282.41	

[3] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313-317, June, 1996.

[4] Shan Zhu and Kai-Kuang Ma, "A New Diamond Search Algorithm for Block-Matching Motion Estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287-290, Feb, 2000.

[5] 김종남, "영상 복잡도와 다양한 매칭 스캔을 이용한 고속 전영역 움직임 예측 알고리즘", *정보과학회논문지 : 소프트웨어 및 응용 제 32권 제 10호*, pp. 949-955, 2005. 10.

[6] *Optimization Model Version 1.0*, ISO/IEC JTC1/SC29/WG11 N3324, Mar 2000.

[7] Alexis M. Tourapis, Oscar C. Au, and Ming L. Liou, "Highly Efficient Predictive Zonal Algorithms for Fast Block-Matching Motion Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 934-947, Oct 2002.

[8] Ishfaq Ahmad, Weiguo Zheng, Jiancong Luo, and Ming Liou, "A Fast Adaptive Motion Estimation Algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 420-438, Mar 2006.

[9] 신세일, 오정수, "블록 정합오차 예측을 이용한 고속 PDE 알고리즘", *한국통신학회논문지 '07-4 Vol. 32 No. 4*, pp. 396-400.

[10] Moon and Todd K, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, Inc., 2000.

[11] Xiaquan Yi and Nam Ling, "Fast Pixel-Based Video Scene Change Detection," *Proc. Of IEEE International Symposium on Circuits and Systems (ISCAS 2005)*, vol. 4, pp. 3443-3446, May 2005.

[12] *Experimental Conditions for Evaluating Encoder Motion Estimation Algorithms*, ISO-IEC/ JTC1/SC29/WG11 N3141, Dec 1999.

[13] 정제창 역, *H.264/AVC 비디오 압축 표준*, 홍릉과학출판사.

이 혁 (Hyuk Lee)

준회원



2006년 2월 : 한양대학교 전자전
기컴퓨터공학부 졸업
2006년 3월~현재 : 한양대학교
전자통신컴퓨터공학과 석박사
통합과정
<관심분야> 영상 처리, 고속 움
직임 예측, SVC

김 종 호 (Jongho Kim)

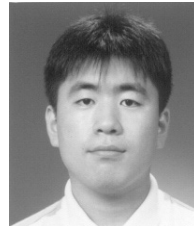
정회원



1998년 2월 : 한양대학교 전자
통신공학과 졸업
2000년 2월 : 한양대학교 전자
통신공학과 석사
2000년 3월~현재 : 한양대학교 전
자통신컴퓨터공학과 박사과정
<관심분야> 영상처리, 영상압축

진 순 중 (Soonjong Jin)

준회원



2004년 2월 : 한양대학교 전자컴
퓨터공학부 졸업
2006년 3월 : 한양대학교 전자통
신컴퓨터공학과 석사
2006년 3월~현재 : 한양대학교
전자통신전파공학과 박사과정
<관심분야> Image Compression,

Image Processing, H.264 Enhancement, Transcoding, MVC, SVC

정 제 창 (Jechang Jeong)

정회원



1980년 2월 : 서울대학교 전자
공학과 졸업
1982년 2월 : KAIST 전기전자
공학과 석사
1990년 : 미국 미시간대학 전기
공학과 공학박사
1980년~1986년 : KBS 기술연구소

연구원 (디지털 TV 및 뉴미디어 연구)

1990년~1991년 : 미국 미시간대학 전기공학과 연구교
수 (영상 및 신호처리 연구)

1991년~1995년 : 삼성전자 멀티미디어 연구소
(MPEG, HDTV, 멀티미디어 연구)

1995년~현재 : 한양대학교 전자통신컴퓨터공학과 교수
(영상통신 및 신호처리 연구실)

1998년 11월 : 과학기술자상 수상

1998년 12월 : 정보통신부장관상 표창

<관심분야> 영상처리, 영상압축