

저 전송률 비디오 압축을 위한 새로운 BC-ABBM 움직임 추정 알고리즘에 관한 연구

정회원 이 완 범*, 김 환 용*

A Study on the New BC-ABBM Motion Estimation Algorithm for Low Bit Rate Video Coding

Wan-bum Lee*, Hwan-yong Kim* *Regular Members*

요 약

고속 탐색 및 기존의 이진 연산 움직임 추정 알고리즘은 연산량 및 처리시간을 대폭 줄일 수 있지만 전역 탐색 움직임 추정 알고리즘에 비하여 성능이 떨어지는 단점이 있다. 따라서 본 논문에서는 하드웨어 구현이 용이하고 움직임 추정을 고속으로 수행 할 수 있는 새로운 BC-ABBM 알고리즘을 제안하였다. BC-ABBM 알고리즘은 움직임 추정시 필요한 연산을 이진 연산으로만 수행하면서 전역 탐색에 근접한 성능을 나타낸다. BC-ABBM 알고리즘의 움직임 추정 성능은 QCIF와 CIF 포맷의 100프레임 영상을 이용하여 분석하였다. BC-ABBM 알고리즘의 PSNR 성능은 전역 탐색 알고리즘보다 약 0.04dB 정도 떨어지지만, 고속 탐색 알고리즘 및 기존의 이진 연산 알고리즘보다는 약 0.6~1.4dB 정도 우수함을 모의실험을 통해 확인하였다.

key Words : Video coding; Block matching; Motion estimation.

ABSTRACT

Fast search and conventional boolean matching motion estimation algorithms reduce computational complexity and data processing time but this algorithms have disadvantages that is difficult of implementation of hardware because of high control overhead and that is less performance than Full search Algorithm(FA).

This paper present new all binary block matching algorithm, called Bit Converted All Binary Block Matching(BC-ABBM). Proposed algorithm have performance closed to the FA by boolean only block matching that may be very efficiently implemented in hardware for low bit rate video communication. Simulation results show that the PSNR of the proposed algorithm is about 0.04dB loss than FA but is about 0.6~1.4dB gain than fast search algorithm and conventional boolean matching algorithm

1. 서 론

저 전송률 비디오 압축 기법은 공중 전화망(PSTN)이나 무선통신망으로 영상 회의, 영상 전화 등 실시간 영상 통신 서비스를 제공하기 위하여 제안되었으며, ISO/IEC MPEG-IV 그룹과 ITU-T/LBC 그룹에서 계속 연구되고 있다[1]. 이러

한 저 전송률 비디오 압축 기법에서 핵심 적인 기능으로 움직임 추정(Motion estimation) 및 보상(Motion Compensation), DCT(Discrete Cosine Transform), 엔트로피 부호화(Entropy coding) 기술이 있다[1,2]. 특히 프레임 간에 유사한 부분을 찾아 차이성분만을 전송하는 움직임 추정(ME)은 고 화질의 영상을 유지하면서 상당히 큰 압축률을 제

* 원광대학교 전자공학과 회로 및 시스템 연구실(lwbwon@wonkwang.ac.kr, hykim@wonkwang.ac.kr)

논문번호 : 040099-0302, 접수일자 : 2004년 3월 2일

※ 이 논문은 2002년도 원광대학교의 교비 지원에 의해서 수행됨.

공하는 역할을 수행한다. 움직임 추정은 블록단위로 움직임을 찾아 그 움직임 벡터를 블록 전체 화소에 적용하는 블록 정합 알고리즘(BMA : Block Matching Algorithm)이 가장 많이 사용된다. 블록 정합 알고리즘 중에서 전역 탐색 알고리즘(FSBMA : Full Search BMA)은 기준 블록을 탐색영역 내의 모든 후보 블록과 비교하는 방법으로 성능이 우수하고 데이터 흐름과 제어회로가 비교적 간단하여 하드웨어 구현이 용이하다는 장점을 갖고 있다.[3] 하지만 탐색 영역이 커질 경우 연산량이 급증하여 실시간 통신 시스템에 적용이 어렵다는 단점을 가지고 있다.

고속 탐색 알고리즘(FBMA : Fast BMA)은 탐색 영역 내의 여러 블록 중 일부분에 대해서만 비교하거나 기준블록의 여러 화소 중 일부만을 비교하기 때문에 연산량을 대폭 줄일 수 있다[4-7]. 하지만 국부 최소(Local Minimum)에 빠질 수 있어 전역 탐색 알고리즘에 비해 성능이 크게 떨어지고 하드웨어 구현시 제어회로가 복잡해지는 단점을 갖고 있다. 최근에는 그레이 영상에서 움직임 추정시 필요한 산술 연산들을 이진 논리 연산으로 처리 할 수 있는 알고리즘들이 제안되고 있다[8-10]. 이러한 이진 연산 알고리즘은 하드웨어 구현이 용이하고 데이터를 고속으로 처리할 수 있다는 장점을 가지고 있다. 하지만 이진 영상을 생성하거나 변환하는 과정에서 많은 데이터 손실이 발생하기 때문에 성능이 전역 및 고속 탐색 알고리즘보다 떨어지는 문제점을 갖고 있다.

따라서 본 논문에서는 움직임 추정을 이진 연산으로 수행하면서 전역 탐색에 근접한 성능을 갖고 하드웨어 구조는 이진 연산 알고리즘에 근접하는 BC-ABBM (Bit Converted All Binary Block Matching) 알고리즘을 제안한다. BC-ABBM은 영상의 윤곽 및 세부 정보가 상위 4비트에 가장 많이 포함되어 있다는 영상의 특성을 이용한 알고리즘이다. BC-ABBM의 특징은 별도의 이진화 및 변환 과정 없이 영상의 상위 4비트를 조합 논리식을 이용하여 15비트로 변환한 후 XOR 연산으로 움직임을 추정한다. 따라서 전역 및 고속 탐색 알고리즘의 산술연산들을 단순한 이진 논리 연산으로 처리함으로써 움직임 추정에 소요되는 계산량 및 처리 속도를 개선할 수 있고 이진 연산 알고리즘의 단점인 성능저하를 개선할 수 있다.

II. 블록정합 알고리즘

블록 정합 알고리즘은 블록 내의 모든 화소가 같은 움직임 벡터를 갖는다는 조건하에 이전 프레임으로부터 현재 프레임을 블록 단위로 추정하는 방법이다.[1][3] 즉 프레임을 일정한 크기의 블록(Block)으로 나누고 현재 부호화 하고자 하는 블록과 가장 유사한 블록을 이전 프레임에서 찾아 그 블록으로 현재의 블록을 추정하는 방법이다. 그림 1과 같이 $N \times N$ 크기의 블록에 대해 프레임당 최대 움직임 속도를 W 화소라 하면 상하, 좌우로 W 화소만큼 움직임이 가능하므로 데이터 검색영역의 크기는 $(2W+N) \times (2W+N)$ 이 된다.

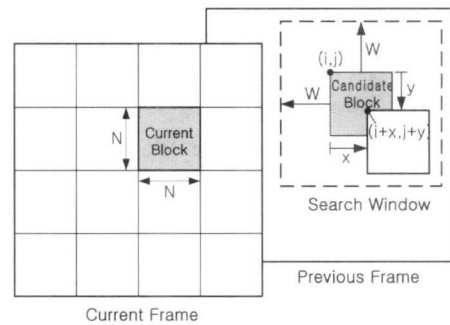


그림 1. 블록 정합 알고리즘(BMA)의 구조

이 때, 블록간의 일치 정도를 평가하는 기준으로 는 MSE(Mean Squared Error), MAD(Mean Absolute Difference), SAD(Sum Absolute Difference)가 있는데, 계산이 간편하고 하드웨어 구현이 용이한 SAD가 주로 사용된다.

$$MSE(x, y) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N [C(i, j, t) - P(i+x, j+y, t-1)]^2 \quad (1)$$

$$MAD(x, y) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N |C(i, j, t) - P(i+x, j+y, t-1)| \quad (2)$$

$$SAD(x, y) = \sum_{i=1}^N \sum_{j=1}^N [C(i, j, t) - P(i+x, j+y, t-1)] \quad (3)$$

식(1),(2),(3)에서 기준 블록과 후보 블록의 크기는 $N \times N$ 이고, 탐색 영역은 수평 수직 각각 $-W \sim +W$ 사이 값을 가진다. $C(i, j, t)$ 와 $P(i+x, j+y, t-1)$

각각 현재 프레임의 기준 블록과 탐색 영역 내 후보 블록의 (i,j)번째 위치의 화소 값을 나타낸다. 그리고 움직임 벡터는 SAD를 정합 기준으로 사용할 때 SAD(x,y)가 최소가 되는 벡터 (x,y)로 식(4)와 같이 표현 할 수 있다.

$$V(x, y) = \arg \min_x \sum_{y=-W}^W \sum_{x=-W}^W SAD(x, y) \quad (4)$$

2.1 전역 탐색(Full Search)

전역 탐색은 이전 화면의 탐색 영역 내 전체를 일일이 탐색하는 방법으로서 블록 정합 알고리즘으로 얻을 수 있는 가장 정확한 움직임을 추정할 수 있다.[3] 탐색 범위가 수직, 수평 방향 모두 $-W \sim +W$ 이고, 정합 기준으로 SAD를 사용할 경우, 전역 탐색의 계산량을 살펴보면 다음과 같다. 하나의 기준블록에 대해서 전역 탐색 알고리즘을 수행하려면 $(2W+1)2$ 번 차의 절대값 합 계산이 필요하므로, 프레임 전송 속도가 m frame/sec이고 한 프레임이 n 개의 기준블록으로 구성되어 있을 때 차의 절대값 합 계산은 초당 $(2W+1)2 \times n \times m$ 번 수행되어야 한다.

기준블록의 크기가 $N \times N$ 화소일 때 차의 절대값 합 계산은 각각 $N \times N$ 번씩의 뺄셈과 절대값, 합 계산으로 구성되므로, 전역 탐색 알고리즘의 뺄셈과 절대값, 합 계산은 각각 $(2W+1)2 \times n \times m \times N^2$ 번이다. 따라서 전역 탐색 기법에서 소요되는 총 연산량은 $3(2W+1)2 \times n \times m \times N^2$ 이 된다. 이처럼 전역 탐색 방법은 알고리즘이 단순하여 하드웨어 구현이 용이하고 정확한 움직임 벡터를 찾을 수 있지만 영상의 크기나 탐색 범위가 증가하는 경우 발생하는 많은 계산량 때문에 실시간 시스템 구현에 많은 어려움이 있다.

2.2 이진 연산 블록 정합 알고리즘

이진 연산 블록 정합 알고리즘은 그레이 영상에서 움직임 추정시 필요한 산술 연산들을 이진 논리 연산으로 처리 할 수 있는 알고리즘으로 하드웨어 구현이 쉽고, 속도가 빠르다는 장점을 가지고 있다. 이러한 움직임 추정 기법에는 8비트 그레이 영상으로부터 한 비트만을 선택하여 영상을 이진화 한 다음에 움직임을 추정하는 알고리즘과, 8비트 그레이 레벨 값을 이진 연산에 적합하도록 전처리 부호화 한 후 움직임을 추정하는 알고리즘이 제안되었다 [8-10].

1) BPM

비트 플레인 정합을 이용한 움직임 추정 알고리즘은 k 비트 그레이 레벨 해상도의 영상에서 한 비트만을 추출하여 영상을 이진화 한 다음 움직임을 추정하는 BPM(Bit Plane Matching) 알고리즘이 있다[8].

$$f(i, j) = b_{k-1}2^{k-1} + b_{k-2}2^{k-2} + \dots + b_12^1 + b_02^0 \quad (5)$$

영상의 해상도가 2^k 의 그레이 레벨 값을 가진다고 할 때, 한 프레임 내의 (i, j) 번째 화소는 식(5)와 같이 k 비트로 표현할 수 있다. 비트 플레인 영상은 각 화소를 나타내는 k 비트의 데이터들 중에서 동일한 위치의 비트들만을 모아서 만든 이진 영상이다. 따라서 2^k 의 그레이 레벨 해상도를 갖는 영상에서는 k 개의 비트 플레인 영상이 생성 될 수 있다. BPM 알고리즘은 프레임 메모리로부터 입력 되는 현재 프레임과 이전 프레임에서 특정 비트 플레인을 선택하여 비트 플레인 영상을 생성하고 생성된 영상으로 움직임 추정을 하게 된다. 움직임 추정의 정합 기준은 식(6)과 같이 k 번째 비트 레벨에서 XOR 연산의 합을 이용하였다. 따라서 BPM의 성능은 몇 번째 비트를 선택하느냐에 따라 크게 달라질 수 있다. 각 비트레벨에서 움직임 추정 성능을 비교 하였을 때 가장 안정적인 성능을 나타낸 것은 일곱 번째 비트 플레인 영상($k=7$)이다.

$$C(x, y) = \sum_{i=1}^N \sum_{j=1}^N b_{k-1}(i, j, t) \oplus b_{k-1}(i+x, j+y, t-1) \quad (6)$$

BPM의 특징은 움직임 정합에 사용할 이진 영상을 비트 플레인으로부터 바로 생성하기 때문에 이진 영상을 생성하기 위한 부가적인 변환과 연산이 필요 없고, 산술연산이 아닌 이진 연산으로 움직임을 추정할 수 있다는 것이다. 따라서 BPM은 하드웨어 구현이 용이하고 고속으로 동작 할 수 있다는 장점을 가지고 있다. 하지만 성능이 전역 탐색뿐만 아니라 고속 탐색보다 많이 떨어지는 단점을 가지고 있다.

2) PCIPM

PCIPM(Pre Coded Image Plane Matching) 알고

리즘은 원영상의 k 개 비트와 Gray code로 변환한 k 개 비트를 이용하여 부호화 한 후 움직임 추정을 하는 알고리즘이다. 움직임 추정 원리는 식(5)의 계수 $b_{k-1}, b_{k-2}, \dots, b_1, b_0$ 와 이 계수를 Gray code로 변환한 $g_{k-1}, g_{k-2}, \dots, g_1, g_0$ 를 이용하여 27비트로 부호화 한 후에 이진 연산으로 수행하게 된다. 이때 사용된 Gray code 변환은 식(7)을 이용하였다[9,10].

$$g_{k-1} = b_{k-1} \tag{7}$$

$$g_m = b_m \oplus b_{m-1}, \quad 0 \leq m \leq k-2$$

27비트의 부호화는 XOR 연산의 차가 실제 픽셀 간의 차에 근접할 수 있도록 픽셀에 비트를 추가하여 부호하는 구조로 그림 2와 같이 나타 낼 수 있다. 그림 2는 0-255 그레이 레벨의 8비트 영상을 27비트로 부호하여 다시 표현한 것이다.

255										0										
[원영상]																				
1	1	1		1	1	0		0	1	0		0	0	0		0	0	0		0
11				10				00				01				11				00
10		11		01				00		10		11			01				01	00
11		10		00		01		11		10		00		01		11			01	11
10	11	01	00	10	11	01	00	10	11	01	00	10	11	01	00	10	11	01	00	
:				:				:				:				:				:

그림 2. 8비트 영상의 27비트 부호화

PCIPM 알고리즘은 비트 변화를 근접한 픽셀 간에는 0~1비트 변화가, 보다 먼 픽셀 간에는 2~3 비트 변화가 이루어지게 함으로써 XOR 연산의 차로 SAD 정합 기준에 근접하는 성능을 나타낼 수 있도록 하였다. 따라서 성능은 기존의 BPM 알고리즘에 비하여 개선할 수 있었다. 하지만 비트 변화가 화소값의 가중치에 따라 이루어지지 않기 때문에 성능이 전역 탐색보다 떨어지고, XOR 연산을 27비트로 확장하여 수행해야 하기 때문에 하드웨어 구현시 기존의 BPM 알고리즘 보다 매우 복잡해지는 단점을 가지고 있다.

III. 새로운 BC-ABBM 움직임 추정 알고리즘

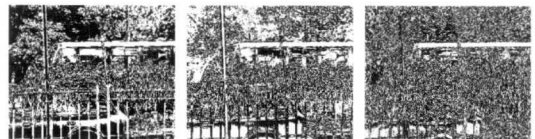
본 논문에서는 고속 탐색 알고리즘, BPM의 성능 저하 문제와 전역 탐색 및 PCIPM의 복잡도를 개선

할 수 있는 새로운 이진 연산 알고리즘을 제안한다. 제안된 알고리즘은 전역 탐색에 근접한 성능을 갖 으면서 BPM에 근접한 하드웨어 복잡도를 갖는 것으로 BC-ABBM 이라고 명명하였다.

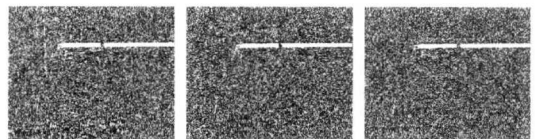
새로운 BC-ABBM 움직임 추정 알고리즘은 256 레벨의 화소값을 표현할 수 있는 8비트 중 상위 4 비트에 가장 많은 영상의 정보가 포함되어 있다는 특징을 이용한다. 그림 3은 BUS 영상에 대한 8개의 비트 플레인 영상을 나타내고 있는데, 영상의 전체적인 윤곽은 상위 비트 플레인 영상에 포함되어 있음을 알 수 있다. 하위 비트 플레인에는 영상의 세세한 정보가 포함되어 있어 영상의 윤곽을 찾아 볼 수가 없다. 따라서 보다 효율적인 움직임 추정을 하기 위해서는 하위 비트 플레인 영상들보다 영상의 윤곽 정보와 세부 정보를 동시에 포함하고 있는 상위 비트 플레인 영상들을 이용하는 것이 효율적 이라는 것을 알 수 있다. 이러한 영상의 특징을 이용하여 제안된 BC-ABBM 알고리즘은 상위 4비트 플레인 영상을 이용하였다.



(a) 원영상 (b) 1번째 비트 영상 (c) 2번째 비트 영상



(d) 3번째 비트 영상 (e) 4번째 비트 영상 (f) 5번째 비트 영상



(g) 6번째 비트 영상 (h) 7번째 비트 영상 (i) 8번째 비트 영상

그림 3. 8개의 비트 플레인 영상

BC-ABBM의 움직임 추정은 그림 4와 같이 이전 영상과 현재 영상의 상위 4비트를 15비트로 변환한 다음 이진연산(XOR)으로 수행된다. 비트 변환은 픽셀의 차이에 따라 1비트에서 최대 15비트까지 균 등하게 하였다. 이웃하는 픽셀 간에는 1비트의 변화

만 있게 하고 먼 거리의 픽셀 간에는 거리에 따른 비트 변화가 이루어지게 하였다.

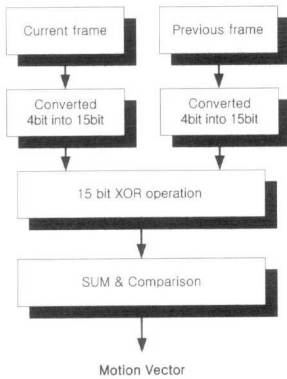


그림 4. 제안된 BC-ABBM 알고리즘 구조

그레이 레벨 영상의 상위 4비트를 15비트로 변환하는 것은 식(8)을 이용하여 수행하였다.

$$\begin{aligned}
 I_0 &= b_{k-1} + b_{k-2} + b_{k-3} + b_{k-4} \\
 I_1 &= b_{k-1} + b_{k-2} + b_{k-3} \\
 I_2 &= (b_{k-1} + b_{k-2}) + (b_{k-3} \cdot b_{k-4}) \\
 I_3 &= b_{k-1} + b_{k-2} \\
 I_4 &= b_{k-1} + b_{k-2} \cdot (b_{k-3} + b_{k-4}) \\
 I_5 &= b_{k-1} + (b_{k-2} \cdot b_{k-3}) \\
 I_6 &= b_{k-1} + (b_{k-2} \cdot b_{k-3} \cdot b_{k-4}) \\
 I_7 &= b_{k-1} \\
 I_8 &= b_{k-1} \cdot (b_{k-2} + b_{k-3} + b_{k-4}) \\
 I_9 &= b_{k-1} \cdot (b_{k-2} + b_{k-3}) \\
 I_{10} &= b_{k-1} \cdot (b_{k-2} + (b_{k-3} \cdot b_{k-4})) \\
 I_{11} &= b_{k-1} \cdot b_{k-2} \\
 I_{12} &= (b_{k-1} \cdot b_{k-2}) \cdot (b_{k-3} + b_{k-4}) \\
 I_{13} &= b_{k-1} \cdot b_{k-2} \cdot b_{k-3} \\
 I_{14} &= b_{k-1} \cdot b_{k-2} \cdot b_{k-3} \cdot b_{k-4}
 \end{aligned} \tag{8}$$

식(8)에서 $b_{k-1}, b_{k-2}, b_{k-3}, b_{k-4}$ 는 그레이 레벨 영상의 상위 4비트를 의미하며, '0' 또는 '1'의 이진수 값을 가진다. 그리고 '+', '.', '·'는 이진 연산의 OR와 AND 연산을 의미하고, $I_0 \sim I_{14}$ 는 변환된 15비트를 나타낸다. BC-ABBM 알고리즘은 변환된 15비트 데이터를 이용하여 XOR 연산으로 정합 정도를 측정하게 되는데, 블록 단위로 XOR 연산을 하여 이진수 "1"의 수가 가장 적은 벡터가 최종 움직임 벡터가 된다. 식으로 표현하면 (9), (10)과 같다.

$$M(x, y) = \sum_{t=0}^{14} \sum_{m=1}^N \sum_{n=1}^N [I_t(m, n, t) \oplus I_t(m+x, n+y, t-1)] \tag{9}$$

식(9)을 이용하여 정합 정도를 측정하고, 최종 움직임 벡터는 식(10)을 이용하여 최종 결정하게 된다.

$$V(x, y) = \arg \min_x \sum_{y \in W} \sum_{y \in W} M(x, y) \tag{10}$$

제안된 알고리즘의 특징은 비트 변화를 균등하게 하여 PCIPM의 단점인 성능 저하와 27비트가 아닌 15비트로 움직임 추정을 하여 하드웨어 구현의 복잡도를 개선할 수 있도록 하였다. 그리고 한 비트만을 이용하여 움직임 추정을 하는 BPM 알고리즘보다 움직임 추정 성능은 크게 개선할 수 있었다.

IV. 모의실험 및 성능 분석

제안된 움직임 추정 알고리즘의 성능 분석은 저전송률 비디오 압축 표준 영상인 QCIF, CIF 포맷의 영상을 이용하여 수행하였다. QCIF 포맷의 비디오 영상은 "Stefan", "Tempete", "Table tennis"를 CIF 포맷의 영상은 "Bus", "Foreman", "Weather"로 각각 100프레임씩을 사용하여 모의실험을 수행하였다. 모의실험에서 탐색영역의 크기는 수평 수직 방향으로 CIF의 경우 각각 -15에서 +15로 하였고, QCIF의 경우 각각 -7에서 +7로 설정하였다.

BC-ABBM 알고리즘의 성능은 전역 탐색 알고리즘, 고속 탐색 알고리즘(TSS : Three Step Search, DS : Diamond Search) 및 이진 연산 알고리즘(BPM, PCIPM)과 MSE 및 PSNR 측면에서 비교분석 하였다. MSE과 PSNR를 식으로 표현하면 식 (11)과 식(12)로 표현 할 수 있다.

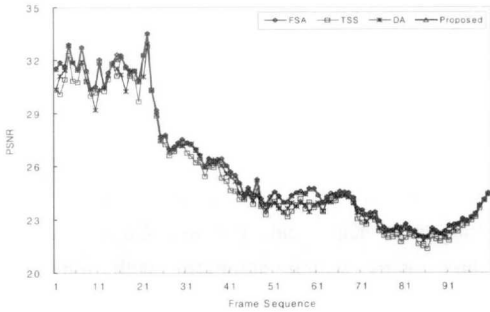
$$MSE = -\frac{1}{VH} \sum_{i=1}^V \sum_{j=1}^H [I(i, j) - R(i, j)]^2 \tag{11}$$

$$PSNR = 10 \log_{10} \left[\frac{2^k - 1}{\sqrt{MSE}} \right] \tag{12}$$

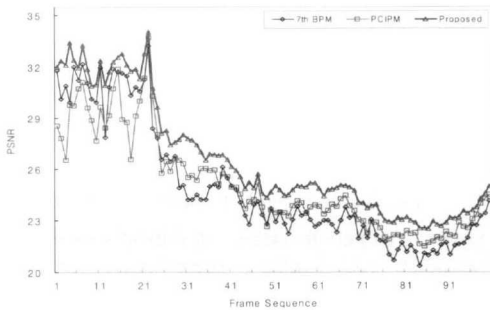
식(11), (12)에서 영상의 크기는 $V \times H$ 개의 픽셀로 QCIF 경우 176×144 , CIF 경우 352×288 이 된다. 그리고 I, R 은 각각 원영상과 복원 영상을 의미하며, k 는 화소당 비트수로 보통 8비트로 표현한다.

그림 5, 6은 각각 QCIF 형태의 영상 "Table tennis"와 CIF 형태의 "Foreman" 영상에 대한 모의

실험결과이다. 모의실험은 100 프레임의 영상을 가지고 수행하였다. 그림 5, 6에서 보는 바와 같이 BC-ABBM 알고리즘의 성능이 전역 탐색에 가장 근접하고 고속 탐색 알고리즘 및 이진 연산 알고리즘 보다는 우수함을 확인하였다.

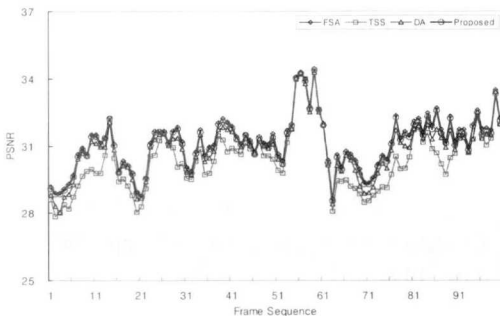


(a) 제안된 알고리즘과 FSA, TSS, DA의 비교

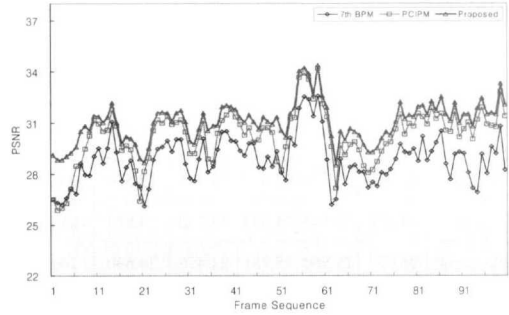


(b) 제안된 알고리즘과 BPM, PCIPM의 비교

그림 5. QCIF 형태의 "Table tennis" 영상에 대한 PSNR



(a) 제안된 알고리즘과 FSA, TSS, DA의 비교



(b) 제안된 알고리즘과 BPM, PCIPM의 비교

그림 6. CIF 형태의 "Foreman" 영상에 대한 PSNR

BC-ABBM의 성능을 더 자세하게 분석하기 위하여 여러 가지의 영상들을 이용하여 모의실험을 수행하였다. 모의실험 결과 값 MSE와 PSNR는 각각 표 1과 2에 나타내었다. 표 1은 움직임 추정 알고리즘으로 복원한 영상과 원영상과의 에러를 나타내는 것이다. 표에서 평균 MSE는 6가지의 영상에 대한 MSE의 평균을 의미한다. 에러가 가장 작은 알고리즘은 평균 MSE가 187.97인 전역 탐색 알고리즘이었다. BC-ABBM 움직임 추정 알고리즘은 평균 에러가 190.23으로 고속 탐색 알고리즘 및 이진 연산 알고리즘 보다 6.6~28.7%정도 에러가 적게 나타났다으며 전역 탐색에 가장 근접함을 확인하였다.

표 1. 모의실험에 사용된 비디오 영상에 대한 평균 MSE

비디오 영상	FSA	TSS	DA	BPM	PCIPM	Proposed
Stefan	290.000	315.659	352.114	357.065	326.014	299.321
Tempete	266.861	278.038	299.973	311.750	292.094	267.332
Table tennis	204.779	229.943	219.276	302.611	266.342	206.038
Bus	303.494	486.142	616.781	517.923	366.070	304.677
Foreman	51.518	62.298	55.406	89.436	66.188	52.747
Weather	11.180	12.405	12.302	18.818	13.478	11.289
평균 MSE	187.97	203.75	259.31	266.67	221.70	190.23

표 2는 모의실험에 사용된 100 프레임 영상에 대한 평균 PSNR을 나타낸 것으로 제안된 알고리즘의 성능이 전역 탐색 알고리즘 보다는 약 0.04dB 정도 떨어지는 반면에 고속 탐색 알고리즘 보다는 약 0.6~0.7dB 정도 개선되었고 이진 연산 알고리즘 보다는 0.7~1.4dB 정도 개선됨을 확인하였다. 특히 탐색 범위가 증가하고, 영상의 움직임이 빠르고 활동적일수록 성능차이가 크게 나타남을 실험을 통해

확인하였다.

표 2. 모의실험에 사용된 비디오 영상에 대한 평균 PSNR

비디오 영상	FSA	TSS	DA	BPM	PCIPM	Proposed
Stefan	23.919	23.720	23.519	23.275	23.550	23.915
Tempete	24.208	24.146	24.053	23.726	23.823	24.200
Table tennis	26.077	25.560	25.731	24.455	24.630	26.044
Bus	23.389	21.410	20.325	21.115	22.558	23.372
Foreman	31.171	30.388	30.866	28.825	30.242	31.066
Weather	39.110	38.785	38.963	37.855	38.617	39.068
평균 PSNR	27.98	27.33	27.24	26.54	27.24	27.94

V. 결 론

본 논문에서는 이진 연산으로 움직임 추정하면서도 전역 탐색 알고리즘에 근접한 성능을 나타내는 BC-ABBM 움직임 추정 알고리즘을 제안하였다. BC-ABBM 알고리즘은 하드웨어 구현이 용이하고 처리속도가 빠른 이진 연산 알고리즘의 장점과, 산술 연산으로 가장 좋은 성능을 나타내는 전역 탐색 알고리즘의 장점을 갖고 있다. BC-ABBM 알고리즘의 성능은 MSE와 PSNR 측면에서 QCIF 및 CIF 포맷의 비디오 영상 100프레임을 이용하여 분석하였다. 실험 결과, BC-ABBM 움직임 추정 알고리즘은 평균 에러가 190.23으로 고속 탐색 알고리즘 및 이진 연산 알고리즘 보다 6.6~28.7% 정도 에러가 적게 나타났고, 전역 탐색 알고리즘보다는 1.19% 정도 크게 나타남을 확인하였다. PSNR 측면에서 분석한 결과, BC-ABBM 알고리즘의 성능이 전역 탐색 알고리즘 보다 0.04dB 정도 떨어지는 반면에 고속 탐색 알고리즘 보다 0.6~0.7dB 정도 개선되었고 이진 연산 알고리즘보다는 0.7~1.4dB 정도 개선됨을 확인하였다. 특히 BC-ABBM 알고리즘의 성능은 탐색 범위가 증가하고, 영상의 움직임이 빠르고 활동적일수록 고속 탐색 및 이진 연산 알고리즘과 성능차이가 크게 나타남을 실험을 통해 확인하였다. 따라서 BC-ABBM 움직임 추정 알고리즘은 저 전송률 영상 통신뿐만 아니라 실시간 비디오 압축 시스템 및 디지털 영상 안정화 시스템 등에 응용할 수 있을 것으로 사료된다.

참 고 문 헌

- [1] Iain E. G. Richardson, "Video Codec Design", John Wiley & Sons Ltd, 2002.
- [2] K. Sayood, "Introduction to Data Compression", 2nd edition, Morgan Kaufmann, 2000.
- [3] K. M. Yang, M .T. Sun, and L. Wu, "A family of VLSI architectures for the full-search block-matching algorithm", *IEEE Trans. Circuits Syst.*, vol. 36, no. 10, pp. 1317-1325, Oct. 1989.
- [4] M. Ghanbari, "The cross search algorithm for motion estimation", *IEEE Trans. Commun.*, Vol.COM-38, pp. 950-953, July 1990.
- [5] Jong-Nam Kim and Tae-Sun Choi, "A fast three steps search algorithm with minimum checking points using Unimodal error surface assumption", *IEEE Transactions on Consumer Electronics*, vol.44, No.3, August 1998.
- [6] Shan Zhu, and Kai-kuang, Ma, "A new Diamond Search algorithm for fast block matching motion estimation", *IEEE Transaction on Image Processing*, vol.9, No.2, pp387-525, Feb 2000.
- [7] J. Chalidabhongse and J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," *IEEE Trans. Circuits and Systems for Video Tech.*, vol.7, no.3, pp.477-488, June 1997.
- [8] Young-Ki Ko and Sung-Jea Ko et al., "VLSI Architecture for Fast Motion Estimation Based on Bit-Plane Matching," *Journal of the Korean Physical Society*, vol. 37, no. 6, pp. 934-944, Dec. 2000.
- [9] S.-J. Ko, S.-H. Lee, S.-W. Jeon, and E.-S. Kong, "Fast digital image stabilizer based on Gray-coded bit-plane matching", *IEEE Trans. Consumer Electronics.*, vol. 45, pp. 598-603, Aug. 1999.
- [10] Erturk, S. "Motion estimation by pre-coded image plane matching", *Int. Conf. Image Proc.*, vol 2, pp.347-350, Sept. 2003.

이 완 범(Wan-Bum Lee)

정회원



1995년 2월 : 원광대학교
전자공학과 공학사
1997년 8월 : 원광대학교
전자공학과 공학석사
1997년 8월 ~ 현재 : 원광대학
교 전자공학과 박사과정

<관심분야> 디지털 신호처리, 영상 통신 및 압축,
VLSI 회로 설계

김 환 옹(Hwan-Yong Kim)

정회원



1973년 2월 : 전북대학교
전기공학과 공학사
1978년 2월 : 전북대학교
전기공학과 공학석사
1984년 8월 : 전북대학교
전기공학과 공학박사
1979년 3월 ~ 현재 : 원광대학
교 전기전자 및 정보공학부 교수

<관심분야> 디지털 신호처리, VLSI 회로 설계