

실시간 비디오 통신에 적합한 새로운 이진 연산 움직임 추정 알고리즘에 관한 연구

A Study on the New Motion Estimation Algorithm of Binary Operation for Real Time Video Communication

이완범* · 심병섭* · 김환용*

Wan-Bum Lee, Byoung-Sup Shim, Hwan-Yong Kim

* 원광대학교 전자공학과

요 약

블록 정합 움직임 추정 알고리즘은 국제 표준인 MPEG, H.26x 등 대부분의 표준 비디오 압축 기법에 채택되어 시간적 중복성을 제거하여 압축률을 높이는 핵심적인 기술로 사용되고 있다. 이러한 블록 정합 알고리즘 중 전역 탐색은 탐색영역이 증가하는 경우 막대한 양의 연산이 필요하다는 단점을 갖고 있다. 반면에 고속 탐색 및 이진 연산 알고리즘은 전역 탐색보다 연산량과 속도는 개선할 수 있지만, 성능이 크게 떨어지는 단점을 가지고 있다. 따라서 본 논문에서는 하드웨어 구현이 용이하고 움직임 추정을 고속으로 수행 할 수 있는 새로운 BCBM(Bit Converted Boolean Matching) 알고리즘을 제안한다. BCBM 알고리즘은 움직임 추정시 필요한 연산을 이진 연산으로만 수행하면서 전역 탐색에 근접한 성능을 나타낸다. 움직임 추정 성능은 CIF 포맷의 영상 100프레임을 이용하여 분석하였다. BCBM 알고리즘의 PSNR 성능은 전역 탐색 알고리즘보다 약 0.08dB 떨어지지만, 고속 탐색 알고리즘 및 기존의 이진 연산 알고리즘보다 0.96~2.02dB 정도 우수함을 실험을 통해 확인하였다.

Abstract

The motion estimation algorithm based block matching is a widely used in the international standards related to video compression, such as the MPEG series and H.26x series. Full search algorithm(FA) ones of this block matching algorithms is usually impractical because of the large number of computations required for large search region. Fast search algorithms and conventional binary block matching algorithms reduce computational complexity and data processing time but this algorithms have disadvantages that is less performance than full search algorithm. This paper presents new Boolean matching algorithm, called BCBM(Bit Converted Boolean Matching). Proposed algorithm has performance closed to the FA by Boolean only block matching that may be very efficiently implemented in hardware for real time video communication. Simulation results show that the PSNR of the proposed algorithm is about 0.08dB loss than FA but is about 0.96~2.02dB gain than fast search algorithm and conventional Boolean matching algorithm.

Key words : Image compression, Video coding, Motion estimation, Block matching algorithm

1. 서 론

실시간 비디오 통신에서 핵심적인 기술은 영상내에 존재하는 시간적, 공간적, 통계적 중복성을 효율적으로 제거하는 것이다. 일반적으로 시간적 중복성은 움직임 추정 및 보상을 사용하여 제거하고, 공간적 중복성은 DCT(Discrete Cosine Transform), 확률적 중복성은 엔트로피 부호화(Entropy coding)를 사용하여 제거한다[1-2].

특히 프레임 간에 유사한 부분을 찾아 차이성분만을 전송하는 움직임 추정(Motion Estimation)은 고화질의 영상을 유지하면서 상당히 큰 압축률을 제공하는 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 제어 회로가 비교적 간단하여 하드웨어 구현이 용이하다[3]. 하지만 탐색 영역이 커질 경우 연산량이 급증하여 실시간 통신 시스템에 적용이 어렵다. 고속 탐색 알고리즘(FBMA : Fast BMA)은 탐색영역 내의 여러 블록 중 일부분에 대해서만 탐색하거나 기준블록의 여러 화소 중 일부분만을 비교하기 때문에 연산량을 대폭 줄일 수 있다[4-9]. 하지만 국부 극소(Local Minimum)에 빠질 수 있어 전역 탐색 알고리즘에 비해 성능이 크게 떨어지고 하드웨어 구현시 제어회로가 복잡해지는 문제점을 갖고 있다.

최근에는 그레이 영상에서 움직임 추정시 필요한 산술 연산들을 이진 논리 연산으로 처리 할 수 있는 알고리즘들이 제안되고 있다[10-12]. 이진 연산 알고리즘의 장점은 고속으로 처리할 수 있으면서 하드웨어 구현이 용이하다는 것이다.

접수일자 : 2004년 5월 25일

완료일자 : 2004년 6월 24일

감사의 글 : 이 논문은 2003년도 원광대학교의 교비 지원에 의해서 수행됨.

하지만 영상을 이진화 하는 과정에서 많은 데이터를 잃어 버리기 때문에 성능이 전역 및 고속 탐색 알고리즘에 비하여 크게 떨어진다. 따라서 본 논문에서는 움직임 추정을 이진 연산으로 수행하면서 전역 탐색에 근접한 성능을 갖고, 하드웨어 구조는 이진 연산 알고리즘에 근접하는 BCBM (Bit Converted Boolean Matching) 알고리즘을 제안한다. BCBM은 영상의 윤곽 및 세부 정보가 상위 4비트에 가장 많이 포함되어 있다는 영상의 특성을 이용한 알고리즘이다. BCBM의 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 논리 연산으로 처리함으로써 움직임 추정에 소요되는 계산량 및 처리 속도를 개선할 수 있고, 기존의 이진 연산 알고리즘 및 고속 탐색 알고리즘의 성능저하를 개선할 수 있다.

2. 블록 정합 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 가능하다. 따라서 데이터 검색영역의 크기는 $(2W + N) \times (2W + N)$ 이 된다.

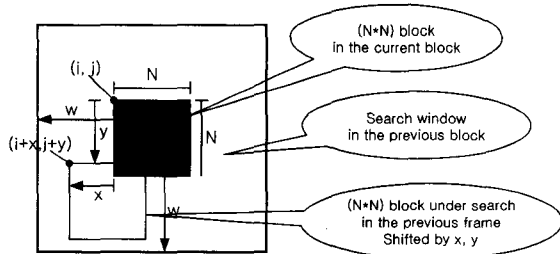


그림 1. 블록 정합 알고리즘(BMA)의 구조
Fig. 1 Structure of Block Matching Algorithm

이때, 블록간의 일치 정도를 평가하는 기준으로는 MSE(Mean of Squared Error), MAD(Mean of Absolute Difference), SAD(Sum of Absolute Difference) 등이 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 $C(i, j, t)$ 와 $P(i+x, j+y, t-1)$ 는 각각 현재 프레임의 기준 블록과 탐색 영역 내 후보 블록의 (i, j) 번째 위치의 화소 값을 나타내고 블록의 크기는 $N \times N$

이다. 그리고 움직임 벡터를 SAD를 정합 기준으로 사용할 때 $SAD(x, y)$ 가 최소가 되는 벡터 (x, y) 로 식(4)와 같이 표현 할 수 있다.

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

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. 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.

이진 연산 블록 정합 알고리즘은 그레이 영상에서 움직임 추정시 필요한 산술 연산들을 이진 논리 연산으로 처리할 수 있는 알고리즘으로 하드웨어 구현이 쉽고, 속도가 빠르다는 장점을 가지고 있다. 이러한 움직임 추정 기법에는 8비트 그레이 영상으로부터 한 비트만을 선택하여 영상을 이진화한 다음에 움직임을 추정하는 알고리즘과, 8비트 그레이 레벨 값을 이진 연산에 적합하도록 전처리 부호화한 후 움직임을 추정하는 알고리즘이 제안되었다[10-12].

이진 연산 알고리즘 중 BPM(Bit Plane Matching) 움직임 추정 알고리즘은 k 비트 그레이 레벨 해상도의 영상에서 한 비트만을 추출하여 영상을 이진화 한 다음 움직임을 추정하는 알고리즘이다.

$$f(i, j) = b_{k-1} 2^{k-1} + b_{k-2} 2^{k-2} + \dots + b_1 2^1 + b_0 2^0 \quad (5)$$

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 입력되는 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[10].

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

BPM의 특징은 움직임 정합에 사용할 이진 영상을 비트 플레인으로부터 바로 생성하기 때문에 이진 영상을 생성하기 위한 추가적인 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.

이진 연산 알고리즘 중 가장 좋은 성능을 나타내는 PCIPM (Pre coded Image Plane Matching)은 8비트 그레이 레벨 값을 이진 연산에 적합하도록 전처리 부호화한 후 움직임을 추정하는 알고리즘이다[12]. 전처리 부호화는 원영상의 k 개 비트와 Gray code로 변환한 k 개 비트를 이용한다. PCIPM 알고리즘의 움직임 추정은 식(3)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 수행한다. 27비트의 부호화는 XOR 연산의 차가 실제 픽셀간의 차에 근접할 수 있도록 픽셀에 비트를 추가하여 부호화하는 구조로 그림 2와 같이 나타 낼 수 있다. 그림 2는 0-255 그레이 레벨의 8비트 영상을 27비트로 부호화하여 다시 표현한 것이다.

PCIPM 알고리즘은 비트 변화를 근접한 픽셀 간에는 0~1비트 변화가, 보다 먼 픽셀 간에는 2~3비트 변화가 이루어 지게 함으로써 XOR 연산의 차로 정합 정도를 나타낸다. 따라서 성능은 기존의 이진화 알고리즘에 비하여 개선할 수 있다. 하지만 화소 값에 상관없이 비트 변화를 수행하기 때문에 XOR 연산 결과가 부정확하다. 예를 들어 입력 화소 값이 각각 "1"하고 "255" 일 경우 가장 큰 차를 나타내야 되는데 "11"이 출력되고, 반면에 입력 화소 값이 각각 "167"과 "126" 일 때는 "17" 출력되어 오히려 더 큰 오차를 출력한다. 따라서 화소값의 가중치에 따라 비트 변환이 이루어지지 않기 때문에 성능이 전역 탐색보다 떨어짐을 알 수 있다. 또한, XOR 연산을 27비트로 확장하여 수행해야 하기 때문에 하드웨어 구현시 기존의 BPM 알고리즘 보다 매우 복잡해지는 단점을 가지고 있다.

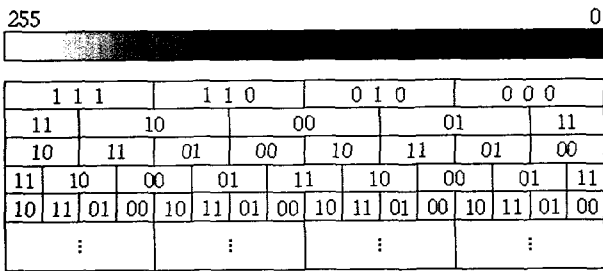


그림 2. 8비트 영상의 27비트 부호화
Fig. 2. 27 bits coded of 8 bits image

3. 새로운 이진 연산 움직임 추정 알고리즘

본 논문에서는 고속 탐색 알고리즘, BPM의 성능 저하 문제와 전역 탐색 및 PCIPM의 복잡도를 개선할 수 있는 새로운 이진 연산 알고리즘을 제안한다. 제안된 알고리즘은 전역 탐색에 근접한 성능을 갖으면서 BPM에 근접한 하드웨어 구조를 갖는 것으로 BCBM(Bit Converted Boolean Matching)이라고 명명하였다. 새로운 BCBM 움직임 추정 알고리즘은 256레벨의 화소값을 표현할 수 있는 8비트 중 상

위 4비트에 가장 많은 영상의 정보가 포함되어 있다는 특징을 이용한다.

그림 3은 "Stefan" 영상에 대한 8개의 비트 플레인 영상을 나타내고 있는데, 영상의 전체적인 윤곽은 상위 비트 플레인 영상에 포함 되어 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 효율적이라는 것을 알 수 있다. 이러한 영상의 특징을 이용하여 제안된 BCBM 알고리즘은 그레이 레벨 영상을 표현하는 8비트 중에 상위 4비트를 사용하였다.



그림 3. 8개의 비트 플레인 영상
Fig. 3 Eight bit-plane images

BCBM의 움직임 추정은 그림 4와 같이 이전 영상과 현재 영상의 상위 4비트를 15비트로 변환 한 다음 이진연산(XOR)으로 수행된다.

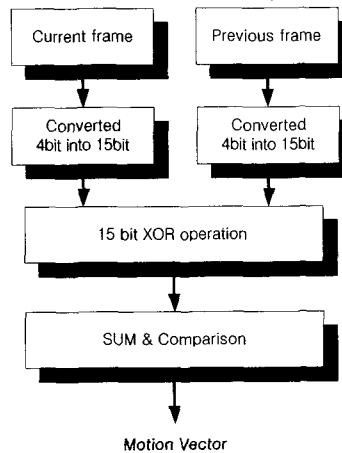


그림 4. 제안된 BCBM 알고리즘 구조
Fig. 4. Structure of proposed BCBM algorithm

비트 변환은 픽셀의 차이에 따라 1비트에서 최대 15비트 까지 균등하게 하였다. 이웃하는 픽셀 간에는 1비트의 변화 만 있게 하고 먼 거리의 픽셀 간에는 거리에 따른 비트 변화 가 이루어지게 하였다. 그레이 레벨 영상의 상위 4비트를 15 비트로 변환은 표 1과 같이 나타낼 수 있다. 표 1에서 $b_{k-1} \sim b_{k-4}$ 는 그레이 레벨 영상의 상위 4비트를 의미하며, '0' 또는 '1'의 이진수 값을 가진다. $I_0 \sim I_{14}$ 는 변환된 15 비트를 나타낸다.

표 1. 비트 변환
Table 1. Bit conversion

| Input | Output |
|---------------------------|---|
| b_{k-1}, \dots, b_{k-4} | $I_{14}, I_{13}, I_{12}, \dots, I_2, I_1, I_0$ |
| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| 0 0 0 1 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 |
| 0 0 1 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 |
| 0 0 1 1 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 |
| 0 1 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 |
| 0 1 0 1 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 |
| 0 1 1 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 |
| 0 1 1 1 | 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 1 0 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 1 0 0 1 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 1 0 1 0 | 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 1 0 1 1 | 0 0 0 0 1 |
| 1 1 0 0 | 0 0 0 1 |
| 1 1 0 1 | 0 0 1 |
| 1 1 1 0 | 0 1 |
| 1 1 1 1 | 1 |

그레이 레벨 영상의 상위 4비트를 15비트로 변환하는 표 1을 논리식으로 표현 하면 식(7)과 같이 나타낼 수 있다.

$$\begin{aligned}
 I_0 &= b_{k-1} + b_{k-2} + b_{k-3} + b_{k-4} \\
 I_1 &= b_{k-1} + b_{k-2} + b_{k-3} \\
 I_2 &= (b_{k-1} + b_{k-2}) + (b_{k-3} \cdot b_{k-4}) \\
 I_3 &= b_{k-1} + b_{k-2} \\
 I_4 &= b_{k-1} + b_{k-2} \cdot (b_{k-3} + b_{k-4}) \\
 I_5 &= b_{k-1} + (b_{k-2} \cdot b_{k-3}) \\
 I_6 &= b_{k-1} + (b_{k-2} \cdot b_{k-3} \cdot b_{k-4}) \\
 I_7 &= b_{k-1} \\
 I_8 &= b_{k-1} \cdot (b_{k-2} + b_{k-3} + b_{k-4}) \\
 I_9 &= b_{k-1} \cdot (b_{k-2} + b_{k-3}) \\
 I_{10} &= b_{k-1} \cdot (b_{k-2} + (b_{k-3} \cdot b_{k-4})) \\
 I_{11} &= b_{k-1} \cdot b_{k-2} \\
 I_{12} &= (b_{k-1} \cdot b_{k-2}) \cdot (b_{k-3} + b_{k-4}) \\
 I_{13} &= b_{k-1} \cdot b_{k-2} \cdot b_{k-3} \\
 I_{14} &= b_{k-1} \cdot b_{k-2} \cdot b_{k-3} \cdot b_{k-4}
 \end{aligned} \tag{7}$$

BCBM 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. 이러한 과정을 식으로 표현하면 식(8), (9), (10)과 같다.

$$\begin{aligned}
 M(x, y) &= \sum_{i=1}^N \sum_{j=1}^N [I_0(i, j, t) \oplus I_0(i+x, j+y, t-1)] \\
 &+ \sum_{i=1}^N \sum_{j=1}^N [I_1(i, j, t) \oplus I_1(i+x, j+y, t-1)] \\
 &\vdots \\
 &+ \sum_{i=1}^N \sum_{j=1}^N [I_{14}(i, j, t) \oplus I_{14}(i+x, j+y, t-1)]
 \end{aligned} \tag{8}$$

식(8)을 다시 정리하여 표현 하면 식(9)와 같이 나타낼수 있다.

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

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

$$V(x, y) = \arg \min_{x=-W}^W \sum_{y=-W}^W M(x, y) \tag{10}$$

제안된 알고리즘의 특징은 탐색 범위가 증가 할수록 연산 량이 급증한다는 전역 탐색의 단점과 국부 극소에 빠져 성능 이 떨어지고 하드웨어 구현이 어려운 고속 탐색 알고리즘의 단점을 이진 연산 정합으로 개선하였다. 그리고 변환 및 이진화 과정에서 데이터 손실이 많아 움직임 추정 성능이 떨어 지는 기존 이진 연산 알고리즘의 단점은 움직임 정보를 많이 포함하고 있는 상위 4비트를 이용함으로써 개선할 수 있다.

4. 모의실험 및 성능 분석

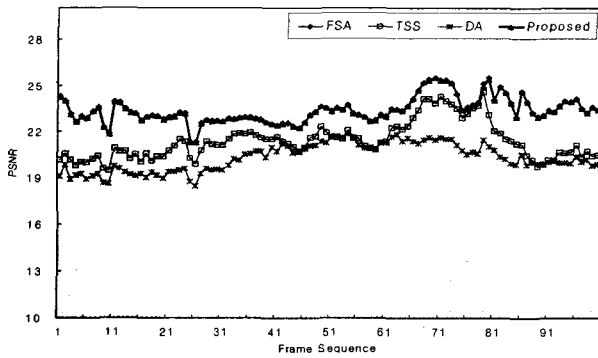
연산량 측면에서 가장 빠른 것은 고속 탐색 알고리즘이다. 탐색 범위가 ±15일 때 고속 탐색 알고리즘의 연산량은 프레임 당 11 MOPS(Mega operations per second)로 가장 빠른 것으로 분석 되었다. 하지만 성능이 크게 떨어지고 하드웨어 구현시 제어가 복잡해지는 단점을 가지고 있다. 반면에 제안 된 BCBM 알고리즘은 이진 연산으로 처리하기 때문에 연산 량 측면에서는 고속 탐색 알고리즘보다 많지만 하드웨어 구현이 용이하여 고속 탐색에 근접한 속도로 데이터를 처리 할 수 있다.

제안된 움직임 추정 알고리즘의 성능 분석은 CIF 포맷의 영상 "Bus", "Stefan", "Foreman", "Carphon", "Weather"로 각각 100프레임씩을 사용하여 수행하였다. 모의실험에서 움직임 추정은 16×16 매크로 블록단위로 수행했으며, 탐색영역의 크기는 수평 수직 방향으로 각각 -15에서 +15로 하였다. BCBM 알고리즘의 성능은 전역 탐색 알고리즘(FS), 고속 탐색 알고리즘(TSS : Three Step Search, DS : Diamond Search) 및 이진 연산 알고리즘(BPM, PCIPM)과 비교 분석 하였다. 성능 평가는 MSE (Mean Square Error) 및 PSNR(Peak to Signal to Noise Ratio)을 이용하였다.

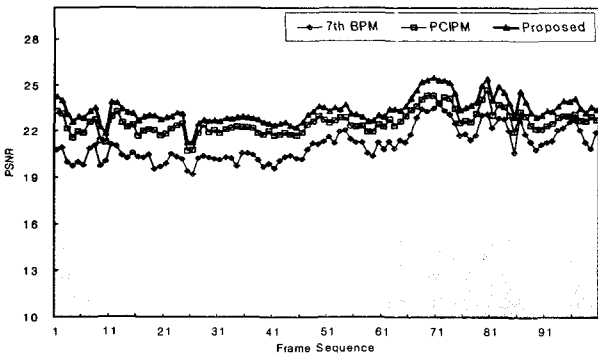
$$MSE = \frac{1}{VH} \sum_{i=1}^V \sum_{j=1}^H [I(i, j) - R(i, j)]^2 \tag{11}$$

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

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

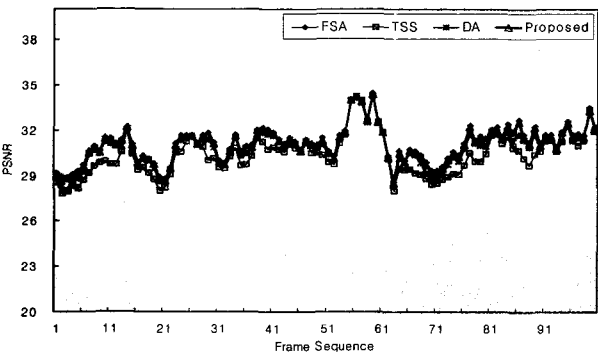


(a) 전역 및 고속 탐색 알고리즘과 비교

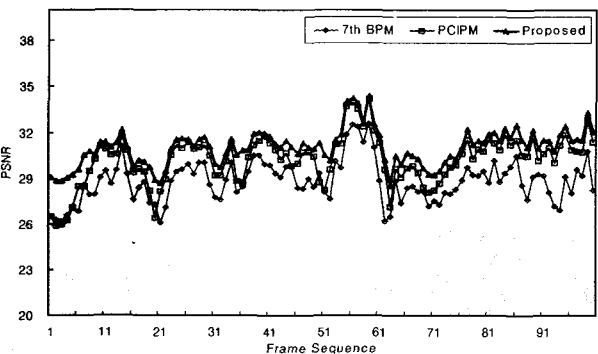


(b) 기존 이진 연산 알고리즘과 비교

그림 5. "Bus" 영상에 대한 PSNR
Fig. 5. PSNR for "Bus" video sequence



(a) 전역 및 고속 탐색 알고리즘과 비교



(b) 기존 이진 연산 알고리즘과 비교

그림 6. "Foreman" 영상에 대한 PSNR
Fig. 6. PSNR for "Foreman" video sequence

그림 5와 6은 각각 "Bus"와 "Foreman" 영상에 대한 모의 실험결과이다. 그림 5, 6에서 보는 바와 같이 BCBM 알고리즘의 성능이 전역 탐색에 가장 근접하고 고속 탐색 알고리즘 및 이진 연산 알고리즘 보다는 우수함을 확인할 수 있다. BCBM의 성능을 더 자세하게 분석하기 위하여 여러 가지의 영상들을 이용하여 실험을 수행하였다. 실험 결과 값 MSE와 PSNR는 각각 표 2와 3에 나타내었다.

표 2에서 평균 MSE는 6가지의 영상에 대한 MSE의 평균을 의미한다. 예러가 가장 작은 알고리즘은 평균 MSE가 119.28인 전역 탐색 알고리즘(FSA)이었다. BCBM은 평균 예러가 120.44로 고속 탐색 알고리즘(TSS, DS) 및 이진 연산 알고리즘(BPM, PCIPM) 보다 16.2~45.7% 정도 예러가 적게 나타났으며 전역 탐색에 가장 근접함을 확인하였다.

표 2. 실험 비디오 영상에 대한 평균 MSE
Table 2. Average MSE of test image sequence

| 실험 영상 | FSA | TSS | DA | BPM | PCIPM | BCBM |
|---------|--------|--------|--------|--------|--------|---------------|
| Bus | 303.49 | 486.14 | 616.78 | 517.92 | 366.07 | 304.68 |
| Carphon | 45.21 | 53.30 | 49.52 | 97.33 | 61.92 | 47.59 |
| Foreman | 51.52 | 62.30 | 55.41 | 89.44 | 66.19 | 52.75 |
| Stefan | 184.99 | 332.10 | 375.81 | 252.80 | 210.91 | 185.91 |
| Weather | 11.18 | 12.40 | 12.30 | 18.82 | 13.48 | 11.29 |
| 평균 MSE | 119.28 | 189.25 | 221.96 | 195.26 | 143.71 | 120.44 |

표 3은 모의실험에 사용된 100 프레임 영상에 대한 평균 PSNR을 나타낸 것으로 제안된 알고리즘의 성능이 전역 탐색 알고리즘 보다는 약 0.08dB정도 떨어지는 반면에 고속 탐색 알고리즘(TSS, DS) 보다는 약 0.96~1.07dB 정도 개선되었고 이진 연산 알고리즘(BPM, PCIPM) 보다는 0.74~2.02dB 정도 개선됨을 확인하였다. 특히 탐색 범위가 증가하고, 영상의 움직임이 빠르고 활동적일수록 성능차이가 크게 나타남을 실험을 통해 확인하였다.

표 3. 실험 비디오 영상에 대한 평균 PSNR
Table 3. Average PSNR of test image sequence

| 실험 영상 | FSA | TSS | DA | BPM | PCIPM | BCBM |
|---------|-------|-------|-------|-------|-------|--------------|
| Bus | 23.39 | 21.41 | 20.32 | 21.11 | 22.56 | 23.37 |
| Carphon | 31.68 | 31.01 | 31.32 | 28.36 | 30.39 | 31.45 |
| Foreman | 31.17 | 30.39 | 30.87 | 28.82 | 30.24 | 31.07 |
| Stefan | 25.75 | 24.30 | 23.89 | 24.45 | 25.18 | 25.72 |
| Weather | 39.11 | 38.79 | 38.96 | 37.85 | 38.62 | 39.07 |
| 평균 PSNR | 30.22 | 29.18 | 29.07 | 28.12 | 29.40 | 30.14 |

5. 결 론

본 논문에서는 이진 연산으로 움직임을 추정하면서도 전역 탐색 알고리즘에 근접한 성능을 나타내는 BCBM 움직임 추정 알고리즘을 제안한다. BCBM 알고리즘은 전역 탐색의 단점과 고속 탐색 알고리즘의 단점을 이진 연산 기법을 적용하여 개선하였다. 그리고 기존의 이진 연산 알고리즘(BPM, PCIPM)의 성능 저하는 영상의 상위 4비트를 사용하여 추정

함으로써 개선할 수 있었다.

BCBM 알고리즘의 성능은 MSE와 PSNR 측면에서 CIF 포맷의 비디오 영상 100프레임을 이용하여 분석 하였다. 실험 결과, BCBM 움직임 추정 알고리즘은 고속 탐색 알고리즘(TSS, DS) 및 이진 연산 알고리즘(BPM, PCIPM) 보다 16.2~45.7% 정도 에러가 적었다. PSNR 측면에서 분석한 결과, BCBM 알고리즘의 성능이 전역 탐색알고리즘 보다 0.08dB 정도 떨어지는 반면에 고속 탐색 알고리즘(TSS, DS) 보다 0.96~1.07dB 정도 개선되었고 이진 연산 알고리즘(BPM, PCIPM) 보다는 0.74~2.02dB 정도 개선됨을 확인 하였다. 탐색 범위가 증가할수록, 영상의 움직임이 빠르고 활동적일수록 BCBM 성능이 더욱 증가하는 것을 실험을 통해 확인하였다. 따라서 BCBM 움직임 추정 알고리즘은 실시간 비디오 통신 시스템 뿐만 아니라 저 전송률 영상 통신 및 디지털 영상 안정화 시스템 등에 응용할 수 있을 것으로 사료 된다.

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] Humaira Nisar, and Tae-Sun Choi, "An new four step search algorithm with minimum checking points for block motion estimation", *Proceedings of MICC and ISCE'99*, Melaka, Malaysia, pp.211-215, 1999.
 [7] 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.
 [8] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low complexcity block based motion estimation via one bit transform", *IEEE Trans. Circuits. Syst.*, Video Technol., vol. 7, no. 4, pp. 702-706, Aug. 1997.
 [9] 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.
 [10] 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.

[11] 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.
 [12] Erturk, S. "Motion estimation by pre-coded image plane matching", *Int. Conf. Image Proc.*, vol 2, pp. 347-350, Sept. 2003.

저 자 소 개



이완범(Wan-Bum Lee)

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

관심분야 : 신경회로망, 디지털 신호처리, 영상 통신 및 압축, VLSI 회로 설계

Phone : 063-850-6740
Fax : 063-857-3999
E-mail : lwbwon@wonkwang.ac.kr



심병섭(Byoung-Sup Shim)

1998년 2월 : 원광대학교 전자공학과
공학사
2000년 2월 : 원광대학교 전자공학과
공학석사
2000년 3월 ~ 현재 : 원광대학교
전자공학과 박사과정

관심분야 : 신경회로망, 디지털 신호처리, VLSI 회로 설계

E-mail : shim@wonkwang.ac.kr



김환용(Hwan-Yong Kim)

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

관심분야 : 신경회로망, 디지털 신호처리, VLSI 회로 설계
E-mail : hykim@wonkwang.ac.kr