

일반논문-11-16-4-08

# 이진 블록 매칭 움직임 예측을 위한 효율적인 탐색 알고리즘

임진호<sup>a)†</sup>, 정제창<sup>a)</sup>

## An Efficient Search Method for Binary-based Block Motion Estimation

Jinho Lim<sup>a)†</sup> and Jechang Jeong<sup>a)</sup>

### 요 약

1비트 변환 (one-bit transform) 및 2비트 변환 (two-bit transform)을 이용하는 이진 블록 매칭 움직임 예측 (motion estimation) 방법은 전역 탐색 (full search) 움직임 예측 방법에 비해 블록 매칭 연산의 복잡도를 감소시키지만 PSNR (Peak Signal-to-Noise Ratio) 성능 저하를 야기한다. 이러한 이진 블록 매칭 움직임 예측 방법의 정확도를 개선하기 위해 조건부 국부 탐색 (conditional local search)이 더해져 보완된 1비트 변환 (modified one-bit transform) 및 보완된 2비트 변환 (modified two-bit transform) 방법이 제안되었다. 그러나 이와 같이 추가적인 국부 탐색은 움직임이 빠른 영상에 대한 16×16 블록 크기의 움직임 예측에 있어서 많은 수의 추가적인 탐색을 필요로 한다. 본 논문은 기존의 조건부 국부 탐색 방법 대신 탐색 영역내의 각 후보 블록들의 (candidate blocks) NNMP (Number of Non-Matching Points)를 기반으로 한 효율적인 탐색 방법을 제안한다. NNMP 기반 탐색 방법을 통하여 작은 NNMP 값을 가지는 후보 블록들을 쉽게 탐색하여 최종 움직임 벡터(motion vector)를 효율적으로 찾을 수 있다. 실험을 통하여 제안하는 알고리즘이 기존의 방법들보다 복잡도 및 정확도 측면에서 좋은 성능을 보여주는 것을 확인하였다.

### Abstract

Motion estimation using one-bit transform and two-bit transform reduces the complexity for computation of matching error; however, the peak signal-to-noise ratio (PSNR) is degraded. Modified 1BT (M1BT) and modified 2BT (M2BT) have been proposed to compensate degraded PSNR by adding conditional local search. However, these algorithms require many additional search points in fast moving sequences with a block size of 16×16. This paper provides more efficient search method by preparing candidate blocks using the number of non-matching points (NNMP) than the conditional local search. With this NNMP-based search, we can easily obtain candidate blocks with small NNMP and efficiently search final motion vector. Experimental results show that the proposed algorithm not only reduces computational complexity, but also improves PSNR on average compared with conventional search algorithm used in M1BT, M2BT and AM2BT.

Keyword : motion estimation, binary motion estimation, one-bit transform, two-bit transform.

## 1. 서 론

영상 압축은 제한된 대역폭과 제한된 용량을 가진 멀티

미디어 환경에서 필수적이며, 움직임 예측은 영상 압축에 있어서 가장 핵심적인 기술이다. 가장 보편적이고 간단한 움직임 예측 방법은 블록 매칭 알고리즘 (block matching algorithm)이다. 영상의 프레임들은 중복되지 않는 정사각형의 블록들로 분할되고, 이 분할된 매크로 블록 (macro block) 단위로 움직임 예측을 위한 부호화가 수행된다. 움

a) 한양대학교 전자컴퓨터통신공학과

Department of Electronics Computer Engineering, Hanyang University

† 교신저자 : 임진호 (zzinho1015@gmail.com)

· 접수일(2011년4월18일), 수정일(2011년6월28일), 게재확정일(2011년7월4일)

직접 예측을 위해서는 먼저 참조 프레임 (reference frame)의 탐색 영역 (search range)내의 분할된 후보 블록들이 (candidate blocks) 현재 프레임의 부호화 할 매크로 블록과 매칭을 통하여 탐색된다. 탐색된 후보 블록들 중 가장 유사한 블록의 위치가 움직임 벡터 (motion vector)이며, 이렇게 각 매크로 블록에 대한 움직임 벡터를 제공하는 것이 블록 매칭 움직임 예측 방법이다. 블록 매칭을 위한 오차 기준 (matching error criterion)으로는 SAD (Sum of Absolute Difference)가 사용된다. 참조 프레임의 탐색 영역내의 모든 후보 블록들의 SAD를 계산하여 움직임 벡터를 찾는 전역 탐색 방법 (full search algorithm)이 가장 정확한 움직임 예측 방법이나 높은 복잡도를 요구한다.

전역 탐색 방법의 복잡도를 줄이기 위한 많은 기술들이 제안되고 있으며, 그 중 하나가 이진 블록 매칭 움직임 예측 방법이다. 이진 블록 매칭 움직임 예측 방법은 영상의 8비트 픽셀 (pixel)을 낮은 비트수로 변환한 후 움직임을 예측하기 때문에 블록간의 매칭을 위해 불리언(Boolean) 연산을 사용할 수 있으며, 이러한 연산은 하드웨어 구현시 고속 움직임 예측을 가능하게 한다. 이진 표현을 위한 방법으로는 8비트 픽셀을 1비트로 표현하는 1BT<sup>[1]</sup> (one-bit transform) 방법과 2비트로 표현하는 2BT<sup>[2]</sup> (two-bit transform) 방법이 있다. 그러나 이러한 방법들은 낮은 비트로 변환하는 과정에서 정보 손실로 인하여 움직임 예측의 정확도가 저하되는 결과를 야기한다. 이러한 움직임 예측 정확도의 성능을 개선하기 위해서 기존의 1BT와 2BT 움직임 예측 방법에 조건부 국부 탐색 (conditional local search) 방법이 더해진 M1BT<sup>[3]</sup> (modified 1BT), M2BT<sup>[4]</sup> (modified 2BT)과 AM2BT<sup>[5]</sup> (Adaptive M2BT) 방법들이 제안되었다. 1BT의 결과로 얻어진 움직임 벡터의 대부분은 전역 탐색 방법에 의해 얻어진 움직임 벡터와 가까운 지역에 위치하는 특성을 보이기 때문에 1BT의 움직임 벡터 주변에 조건부 국부 탐색 수행하면 기존 1BT 보다 더 정확한 움직임 벡터를 구할 수 있다<sup>[3]</sup>.

이러한 보완된 방법들은 적은수의 추가 탐색으로 기존의 1BT와 2BT 방법에 정확성을 많이 보완했다. 하지만 일부 영상의 16×16 블록 크기 움직임 예측에서 상당히 많은 수의 추가적인 탐색이 요구되는 단점이 있다. 실제로

AM2BT<sup>[5]</sup>의 “Football” 영상에서 16×16 블록 크기에 대한 실험결과를 보면, 하나의 매크로 블록당 평균적으로 M2BT에서 66번, AM2BT에서 48번의 추가 탐색이 요구되며, 이러한 결과는 7번의 추가 탐색 수를 요구하는 “Foreman”의 실험결과와 비교했을 때 상당히 높은 복잡도를 가짐을 의미한다. 본 논문에서는 이러한 보완된 이진 블록 매칭 움직임 예측을 위한 새로운 추가 탐색 알고리즘을 제안한다. 2SS (two-step search) 및 전역 탐색 (full search)을 사용하는 기존의 추가 탐색 방법들과는 달리 참조 프레임의 모든 후보 블록들 간의 NNMP (number of non-matching points) 값의 차이를 추가적인 탐색을 위해 사용한다. 실험결과를 보면, 제안하는 알고리즘이 복잡도를 감소시킬 뿐 아니라 PSNR도 향상시키는 것을 알 수 있다.

본 논문은 구성은 다음과 같다. 제 2장에서는 종래의 이진 블록 매칭 움직임 예측 알고리즘에 대한 개요를 설명하고, 제 3장에서는 본 논문에서 제안하는 NNMP 기반 추가 탐색 알고리즘에 대해 소개한다. 제 4장에서는 종래의 알고리즘들과 복잡도 및 정확성을 비교하고, 제 5장에서 결론을 맺는다.

## II. 종래의 알고리즘들

### 1. 1비트 변환을 사용하는 움직임 예측

이진 블록 매칭 움직임 예측은 영상의 화소를 적은 비트수로 변환하기 때문에 여러 비트를 한꺼번에 처리할 수 있으며, XOR나 OR 연산과 같은 불리언 연산을 사용할 수 있다. 이러한 특징은 하드웨어 구현시, 움직임 예측을 위한 계산의 고속처리를 가능하게 한다. 이진 블록 매칭을 위한 방법으로 1BT와 2BT를 이용한 움직임 예측 방법이 제안되었다. 먼저 1BT는 영상의 화소를 1비트로 표현하는 것으로, 영상 프레임  $I$ 는 (1)에 정의된 17×17 대역 통과 필터 커널  $K$ 에 의해 통과되고, 1비트 변환된 프레임은 대역 통과 필터 적용된 프레임  $I_F$ 와 원본 영상 프레임  $I$ 의 각 화소를 식 (2)와 같이 비교하여 만들어 낸다.

$$K(i,j) = \begin{cases} 1/25, i,j \in [0,4,8,12,16] \\ 0, otherwise \end{cases} \quad (1)$$

$$B(i,j) = \begin{cases} 1, I(i,j) \geq I_F(i,j) \\ 0, otherwise \end{cases} \quad (2)$$

영상의 모든 프레임은 식 (1), (2)에 의해 1비트 프레임으로 변환되며, 현재 및 이전에 변환된 프레임 사이의 움직임 예측을 위한 매칭 오차 기준인  $NNMP_{1BT}$ 는 식 (3)과 같이 정의된다.

$$NNMP_{1BT}(m,n) = \sum_{i=0}^{N-1N-1} \sum_{j=0}^{N-1N-1} B^t(i,j) \oplus B^{t-1}(i+m,j+n) \quad (3)$$

$$-s \leq m, n \leq s-1$$

여기서  $B^t(i,j)$ 는 현재 프레임의 변환된 프레임,  $B^{t-1}(i,j)$ 는 이전 프레임의 변환된 프레임을 말한다. 또한  $\oplus$ 는 배타적 논리합 (XOR) 연산,  $N \times N$ 은 블록 크기,  $-s \leq m, n \leq s-1$ 은 탐색 범위를 의미한다.

### 2. 2비트 변환을 사용하는 움직임 예측

2BT를 이용한 움직임 예측은 1BT를 이용한 움직임 예측의 정확성을 향상시키기 위하여 제안되었다. 2BT을 위해 식 (4)와 같이 국부 주변 (local surrounding) 화소들의 평균 ( $\mu$ ), 표준편차 ( $\sigma$ )와 근사치 (approximate) 표준편차 값들이 사용된다.

$$\begin{aligned} \mu &= E[I_{tw}] \\ \sigma^2 &= E[I_{tw}] - E^2[I_{tw}] \\ \sigma_a &= 15 + 0.0125\sigma^2 \end{aligned} \quad (4)$$

여기서  $I_{tw}$ 는 2비트로 변환되는  $8 \times 8$  크기 변환 블록 주변의  $40 \times 40$  크기 국부 문턱 윈도우 (local threshold win-

dow)안의 화소값들을 말한다. 2BT을 위한 첫 번째 비트 평면은 식 (5)와 같이 윈도우안의 화소들의 평균값과 비교하여 얻어지며, 두 번째 비트 평면은 평균 및 표준편차 값과 비교하여 얻어진다.

$$\begin{aligned} B_1(i,j) &= \begin{cases} 1, I(i,j) \geq \mu \\ 0, otherwise \end{cases} \\ B_2(i,j) &= \begin{cases} 1, I(i,j) \geq \mu + \sigma_a \text{ or } I(i,j) \leq \mu - \sigma_a \\ 0, otherwise \end{cases} \end{aligned} \quad (5)$$

여기서  $B_1(i,j)$ 은 첫 번째 비트 평면,  $B_2(i,j)$ 는 두 번째 비트 평면을 표현한다. 2BT 기반 움직임 예측을 위한 오차 기준인  $NNMP_{2BT}$ 는 다음 식 (6)과 같이 주어진다.

여기서  $B_{1,2}^t(i,j)$ 는 현재 변환된 2비트 평면,  $B_{1,2}^{t-1}(i,j)$ 는 이전 변환된 2비트 참조 평면을 말한다. 또한  $\oplus$ 는 배타적 논리합 (XOR) 연산,  $\parallel$ 는 논리합 (OR) 연산,  $N \times N$ 은 블록 크기,  $-s \leq m, n \leq s-1$ 은 탐색 범위를 의미한다. 2BT의 실험 결과에서 보는 것처럼 2BT 기반 움직임 예측이 1BT 기반 보다 PSNR이 향상되기는 했으나 여전히 전역 탐색보다는 부정확한 결과를 보인다.

### 3. 보완된 이진 블록 매칭 움직임 예측

보완된 방법들은 이진 블록 매칭 움직임 예측에 조건적 국부 탐색을 추가하여 움직임 예측의 정확도를 향상시켰다. 1BT 과정 후 2SS와 전역 탐색을 추가하여 움직임 예측 정확도를 향상시키는 1BT를 위한 보완된 접근법<sup>[3]</sup>이 먼저 제안되었고, 동일한 방법이 2BT에 적용되었다<sup>[4]</sup>. 보완된 이진 블록 매칭 움직임 예측의 첫 번째 단계는 기존의 이진 블록 매칭 움직임 예측을 동일하게 수행하여  $NNMP$  값이 가장 작은 블록의 움직임 벡터 ( $mv_1$ )와 두 번째로 작은 블록의 움직임 벡터 ( $mv_2$ )를 구한 후, 두 개의 움직임 벡터가 최종 움직임 벡터가 될지 여부를 판단하는 것이다. 최종 움직임 벡터 판단을 위한 블록 왜곡 측정은 식 (7)과 같이

$$NNMP_{2BT}(m,n) = \sum_{i=0}^{N-1N-1} \sum_{j=0}^{N-1N-1} B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n) \parallel B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n) \quad (6)$$

$$-s \leq m, n \leq s-1$$

SSAD (sub-sampled sum of differences)를 사용한다.

$$D = \sum_{(i \bmod 2 = j \bmod 2)} |I_t(i, j) - I_{t-1}(i - mv_x, j - mv_y)| \quad (7)$$

여기서  $I_t(i, j)$  과  $I_{t-1}(i, j)$ 는 각각 변환되지 않은 현재 프레임과 참조 프레임의 매크로 블록 내의 화소들을 말한다. 먼저  $D_{mv_1}$ 이 구해지면 문턱값  $T$ 와 비교하여  $T$  이하이면  $mv_1$ 이 최종 움직임 벡터가 되고, 그렇지 않으면  $D_{mv_2}$ 를 구해  $T$ 와 비교한다. 마찬가지로  $D_{mv_2}$ 가  $T$  이하이면  $mv_2$ 가 최종 움직임 벡터가 되고, 그렇지 않으면 다음 두 번째 단계로 넘어간다.

보완된 접근법의 두 번째 단계는 2SS를 이용한 추가 탐색을 수행하는 것이다. 먼저 그림1과 같이  $mv_1$ 을 중심으로 사각형 표시된 다섯 개의 움직임 벡터들 중 가장 작은 SSAD 값을 가진 움직임 벡터를 구한 후, 다시 그 주변에 위치한 여덟 개의 원형 표시된 움직임 벡터들 중 가장 작은 SSAD 값을 갖는 움직임 벡터를 구한다. 즉  $mv_1$ 을 중심으로 2SS를 통해 탐색된 13개의 후보 블록들 중 가장 왜곡이 작은 블록을 구한 후, 해당 벡터의 SSAD 값이  $2 \times T$  이하이면 최종 움직임 벡터로 결정한다. 그렇지 않으면  $mv_2$ 를 가지고 동일한 과정을 수행한다. 만약  $mv_2$  위치에서도 2SS를 통해 최종 움직임 벡터를 구하지 못하면 마지막 세 번째 단계로써 전역 탐색을 실시한다. 전역 탐색은 탐색 범위 내 모든 후보 블록들을 SSAD를 계산하기 때문에 첫 번째, 두 번째 단계에서 최종 움직임 벡터를 찾지 못하면 복잡도는 매우 높아지게 된다.

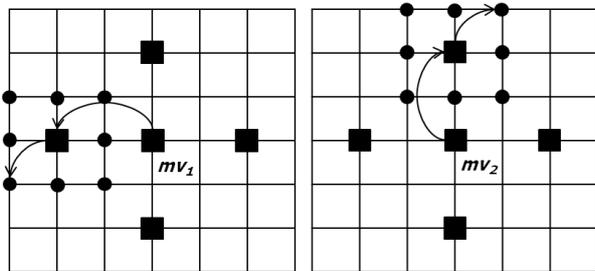


그림 1.  $mv_1$ 과  $mv_2$ 를 위한 두 단계 탐색의 예  
Fig. 1. An example of the two-step search path for  $mv_1$  and  $mv_2$

보완된 방법에서 사용하는 문턱값  $T$ 는 추가 탐색의 복잡도와 정확성 사이의 트레이드 오프 (trade-off)에 따라 결정된다.  $T$ 가 커지면 첫 번째, 두 번째 단계에서 최종 움직임 벡터가 결정될 확률이 높아지기 때문에 복잡도는 많이 증가하지 않겠지만, 정확성의 많은 향상 또한 기대하기 힘들다.  $T$ 가 너무 낮으면 마지막 단계인 전역 탐색의 수가 많아져서 정확성은 많이 향상되겠지만, 복잡도 또한 높아진다. 최적의  $T$ 의 값을 위한 AM2BT의 실험 결과를 보면, 고정된 문턱값을 사용한 M2BT에서는  $T=6$ , 현재 매크로 블록의 표준편차 ( $\sigma$ )에 의한 적응적인 문턱값을 사용한 AM2BT에서는  $T = 5 \times (4 + (\sigma/100))$ 를 사용했으며, AM2BT에서 가장 최상의 결과를 보여주는 것을 볼 수 있다.

### III. 제안하는 알고리즘

#### 1. 참조 프레임 후보 블록들의 NNMP 차이

종래의 M2BT 및 AM2BT를 사용하는 움직임 예측 방법을 보면,  $mv_1$ 과  $mv_2$ 를 구하는 단계에서 참조 프레임의 모든 후보 블록들의 NNMP 값들은 이미 계산되어진다. 제안하는 알고리즘은 이와 같은 후보 블록들의 NNMP 값들을 활용하는 것으로써, 참조 프레임의 탐색 범위내의 모든 후보 블록들의 NNMP 차이 ( $NNMP_{diff}$ )는 식 (9)와 같이 정의된다.

$$NNMP_{diff}(m, n) = NNMP_{2BT}(m, n) - NNMP_{min} \quad (9)$$

$$-s \leq m, n \leq s-1$$

여기서  $-s \leq m, n \leq s-1$ 은 탐색 범위를,  $NNMP_{min}$ 은 후보 블록들의 NNMP 값 중 가장 작은 NNMP 값을 의미한다. 즉  $NNMP_{diff}$ 는 자신의 NNMP 값과 가장 작은 NNMP 값의 차이를 말한다. 본 논문에서는 이진 블록 매칭 움직임 예측 방법으로 1BT보다 정확도 측면에서 성능이 우수한 2BT 방법을 사용했고, NNMP 또한 2BT 과정에 의해 얻어진  $NNMP_{2BT}$ 를 사용했다.

## 2. NNMP 차이값에 의해 분류된 후보 블록들

이진 블록 매칭 움직임 예측에서 예측 오차 기준으로 사용하는 NNMP 값들의 범위는 실제 전역 탐색 방법에서 사용하는 예측 오차 기준인 SAD 값의 범위보다 훨씬 작기 때문에 후보 블록들이 같은 NNMP 값을 가질 확률이 매우 높다. 즉 다수의 후보 블록들이 같은  $NNMP_{diff}$  값을 가질 수 있으며, 이를 가지고  $NNMP_{diff}$  히스토그램 표를 작성할 수 있다. 표 1은 실제 “Football” 영상의  $NNMP_{diff} < 8$  인 범위안의 후보 블록들의 히스토그램 표의 예를 보여준다. 625 번째 매크로 블록을 보면, 39개의 후보 블록들이 0의  $NNMP_{diff}$  값을 가지는 것을 볼 수 있다. 이때 같은  $NNMP_{diff}$  값을 가지는 후보 블록들은 나선형 탐색에 의해 저장되어 현재 매크로 블록과 가까운 후보 블록들을 먼저 탐색할 수 있게 한다.

## 3. NNMP 차이 기반 추가 탐색 알고리즘

앞에서 설명한 각 후보 블록들의  $NNMP_{diff}$  값과  $NNMP_{diff}$  값에 의해 분류된 후보 블록들을 이용하여 본 논문에서는 보완된 이진 블록 매칭 움직임 예측을 위한 새로운 탐색 알고리즘을 제안하며, 이것을  $NNMP_{diff}$  기반 추가 탐색 알고리즘이라 한다.

제안하는 알고리즘의 첫 번째 단계는 2BT 단계에서 도출된 움직임 벡터 중 매우 작은 SSAD 왜곡을 가지는 움직임 벡터를 최종 움직임 벡터로 걸러내는 단계이다. 매우 작은 SSAD 왜곡을 가지는 움직임 벡터는 최종 움직임 벡터가 될 확률이 높기 때문에 본 단계를 통하여 2BT의 움직임 벡터들 중  $NNMP_{diff}$  기반 추가 탐색 알고리즘을 적용할 필요가 없는 움직임 벡터들을 걸러낸다. 즉 2BT 단계에서 도

출된 움직임 벡터인  $mv_{2BT}$ 의 SSAD 값이 문턱값인  $T_p$  이하이면  $mv_{2BT}$ 는 추가 탐색 없이 바로 최종 움직임 벡터가 되고, 그렇지 않으면 본 논문에서 제안하는  $NNMP_{diff}$  기반 추가 탐색을 위한 다음 단계로 넘어간다. 문턱값  $T_p$ 는 매우 작은 왜곡값을 가진  $mv_{2BT}$ 을 걸러내기 위해서 M2BT나 AM2BT에서 사용하는 문턱값  $T$ 보다 작은 값으로 실험에 의해 결정되었다.

제안하는 알고리즘의 두 번째 단계인  $NNMP_{diff}$  기반 추가 탐색 방법은 작은  $NNMP_{diff}$  값을 가지는 블록들을 후보 블록들로 고려하는 것으로 시작한다. 즉,  $NNMP_{diff} < t_c$  이한 블록들만이 최종 움직임 벡터 유무를 판단하기 위한 후보 블록들로서 탐색된다. 여기서  $t_c$ 는  $NNMP_{diff}$  값에 따라 후보 블록들을 제한하기 위한 문턱값을 말한다. 그림 2는 “Football” 영상을 가지고 전역 탐색 방법을 수행하여 얻어진 실제 움직임 벡터 (true motion vector)들의 프레임 당 평균  $NNMP_{diff}$  값들의 분포를 보여준다. 그림 2를 보면, 평균적으로 실제 움직임 벡터의 98% 이상이 15보다 작은  $NNMP_{diff}$  값을 가진다. 즉, 작은  $NNMP_{diff}$  값을 가지는 움직임 벡터들이 SSAD 전역 탐색을 통한 최종 움직임 벡터가 될 확률이 많기 때문에 제안하는 알고리즘에서는  $t_c$  이하의 작은  $NNMP_{diff}$  값을 가지는 후보 블록들만을 추가적으로 탐색한다. 표 1에서는  $t_c=8$  ( $NNMP_{diff} < 8$ )의 문턱값이 사용되었다. 2번째 매크로 블록을 보면, 해당 범위 안에 단지 두 개의 후보 블록들이 존재하는 반면, 625번째 매크로 블록의 예에서와 같이  $NNMP_{diff} < t_c$  이한 범위 안에 100개 이상의 후보 블록들을 가지는 경우도 있다. 첫 번째 경우에는 추가적으로 적은 탐색 수를 필요로 하지만, 두 번째와 같이 작은  $NNMP_{diff}$  값을 가지는 후보 블록들이 많이 존재할 경우에는 범위 안의 모든 후보 블록들을 탐색하는 것은 비효율적이며 불필요한 복잡도를 증가시키게 될 것이다.

표 1. “Football” 영상 3번째 프레임, 8×8 블록 크기에서  $NNMP_{diff} < 8$  인 조건의 후보 블록들을 위한 히스토그램 테이블의 예  
Table 1. An example of the histogram table for candidate blocks in the range of  $NNMP_{diff} < 8$  in the 3rd frame of the “Football” sequence with an 8×8 block size

$NNMP_{diff}$ Value	0	1	2	3	4	5	6	7	Total
The Number of Blocks (2 <sup>nd</sup> MB)	1	0	1	0	0	0	0	0	2
The Number of Blocks (625 <sup>th</sup> MB)	39	25	25	19	16	13	15	13	165



표 3. "Football" 영상 3번째 프레임 625번째 매크로 블록, 8×8 블록 크기에서의 NNMP 기반 탐색의 예

Table 3. An example of the  $NNMP_{diff}$ -based search for 625th MB in the 3rd frame of the "Football" sequence with an 8×8 block size

$NNMP_{diff}$ Value	0	1	2	3	4	5	6	7	Total
The Number of Candidate Blocks	39	25	25	19	16	13	15	13	<b>165</b>
<b>The Number of Final Candidate Blocks</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>5</b>	0	0	<b>30</b>

표 4. "Football" 영상 3번째 프레임 1088번째 매크로 블록, 8×8 블록 크기에서의 NNMP 기반 탐색의 예

Table 4. An example of the  $NNMP_{diff}$ -based search for 1088th MB in the 3rd frame of the "Football" sequence with an 8×8 block size

$NNMP_{diff}$ Value	0	1	2	3	4	5	6	7	Total
The Number of Candidate Blocks	2	1	4	12	3	7	10	7	<b>46</b>
<b>The Number of Final Candidate Blocks</b>	<b>2</b>	<b>1</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>5</b>	<b>5</b>	<b>30</b>

또한  $\beta$  문턱값에 따라 총 수집된 후보 블록들의 개수가 30개가 되면 탐색은 종료된다. 표 3의 625번째 매크로 블록을 위한 추가 탐색에서는  $NNMP_{diff}$  값이 5인 그룹까지 총 30번의 탐색이 수행되었고, 표 4의 1088번째 매크로 블록을 위한 추가 탐색은  $NNMP_{diff}$  값이 7인 그룹까지 마찬가지로 30번의 추가 탐색이 수행되었다. 각 후보 블록의 추가 탐색마다 SSAD 왜곡이 계산되고, 가장 작은 SSAD 값을 가지는 움직임 벡터가 갱신된다.

#### IV. 실험 결과

실험은 움직임 예측 알고리즘만의 성능 측정을 위해 독

립적으로 작성된 Visual C++ MFC 응용 프로그램에 의해 수행되었으며, 성능 비교를 위해 종래의 방법들에 대한 실험도 함께 수행되었다. 4개의 SIF (352×240) 영상들과 6개의 CIF (352×288) 영상들이 8×8과 16×16의 블록 크기를 위한 실험을 위해 사용되었다. 표 5, 6의 처음 6개 영상들은 실험의 객관적인 성능 비교를 위하여 M2BT과 AM2BT에서 사용했던 동일한 실험 영상들이다. 성능 측정 방법 또한 AM2BT에서 사용했던 동일한 방법을 사용하였다. 즉, 움직임 예측의 정확도 비교를 위해 PSNR을 측정하였고, 추가 탐색의 복잡도 비교를 위해 MB당 평균 탐색의 수 (average number of search points per MB)를 측정하였다. 여기서 ‘f’는 SSAD를 사용하는 8비트 기반 평균 탐색 수를 의미하며, ‘b’는 NNMP를 사용하는 이진기반 평균 탐색 수를 의미한다<sup>[5]</sup>.

표 5. 16×16 블록 크기에서의 PSNR과 평균 추가탐색 수의 결과 비교

Table 5. PSNR (dB) and average search points results with a block size 16×16 and search range 16 pixels

Sequences	Format, Frames	FSA	1BT	2BT	M2BT (T=6)		AM2BT		Proposed (t=30)	
		PSNR	PSNR	PSNR	PSNR	Av. search points	PSNR	Av. search points	PSNR	Av. search points
Football	SIF, 125	22.88	21.83	22.08	22.68	1024b+60f	22.68	1024b+53f	22.76	1024b+9f
Foreman	CIF, 299	32.08	30.50	30.64	31.04	1024b+4f	31.05	1024b+4f	31.50	1024b+3f
Tennis	SIF, 112	29.87	28.73	28.92	29.42	1024b+13f	29.41	1024b+13f	29.66	1024b+5f
Flowergarden	SIF, 112	23.79	23.29	23.41	23.67	1024b+25f	23.67	1024b+21f	23.73	1024b+3f
Mobile	SIF, 140	22.99	22.66	22.70	22.84	1024b+26f	22.83	1024b+22f	22.91	1024b+5f
Coastguard	CIF, 299	30.48	29.83	29.94	30.23	1024b+2f	30.22	1024b+2f	30.39	1024b+2f
Bus	CIF, 150	25.01	23.77	24.34	24.71	1024b+25f	24.68	1024b+20f	24.89	1024b+6f
Tempete	CIF, 260	26.88	26.15	26.42	26.68	1024b+10f	26.68	1024b+11f	26.76	1024b+3f
Children	CIF, 299	29.78	28.94	28.98	29.52	1024b+13f	29.50	1024b+10f	29.63	1024b+2f
News	CIF, 299	36.91	35.44	35.83	36.27	1024b+3f	36.26	1024b+3f	36.62	1024b+1f
Average		<b>28.07</b>	<b>27.11</b>	<b>27.33</b>	<b>27.71</b>	<b>1024b+18f</b>	<b>27.70</b>	<b>1024b+16f</b>	<b>27.89</b>	<b>1024b+4f</b>

표 6. 8×8 블록 크기에서의 PSNR과 평균 추가탐색 수의 결과 비교

Table 6. PSNR (dB) and average search points results with a block size 8×8 and search range 8 pixels

Sequences	Format, Frames	FSA	1BT	2BT	M2BT (T=6)		AM2BT		Proposed ( $t_c=8$ )	
		PSNR	PSNR	PSNR	PSNR	Av. search points	PSNR	Av. search points	PSNR	Av. search points
Football	SIF, 125	24.70	22.67	23.31	24.31	256b+14f	24.29	256b+13f	24.40	256b+7f
Foreman	CIF, 299	32.81	29.73	30.43	31.19	256b+2f	31.21	256b+3f	31.85	256b+4f
Tennis	SIF, 112	31.25	29.16	29.89	30.57	256b+4f	30.57	256b+4f	30.84	256b+4f
Flowergarden	SIF, 112	25.26	24.09	24.56	24.98	256b+9f	24.97	256b+7f	25.06	256b+3f
Mobile	SIF, 140	23.82	22.57	22.87	23.36	256b+13f	23.33	256b+11f	23.52	256b+5f
Coastguard	CIF, 299	31.61	29.15	30.46	30.85	256b+1f	30.84	256b+1f	31.36	256b+3f
Bus	CIF, 150	26.70	24.37	25.25	26.10	256b+7f	26.04	256b+6f	26.33	256b+5f
Tempete	CIF, 260	27.87	26.25	26.90	27.42	256b+4f	27.43	256b+4f	27.62	256b+3f
Children	CIF, 299	31.56	29.70	30.18	31.15	256b+3f	31.12	256b+3f	31.22	256b+1f
News	CIF, 299	38.62	35.85	36.69	37.38	256b+1f	37.35	256b+1f	37.92	256b+1f
Average		<b>29.42</b>	<b>27.35</b>	<b>28.05</b>	<b>28.73</b>	<b>256b+5.8f</b>	<b>28.72</b>	<b>256b+5.3f</b>	<b>29.01</b>	<b>256b+3.6f</b>

표 7. "Football" 영상, 16×16 블록 크기에서의 블록 매칭을 위한 매크로 블록 당 평균 연산량 비교

Table 7. Average computational complexity for matching in number of operations per MB in the "Football" sequence with a 16×16 block size

Operation	Av. operations per MB for matching					
	FSA	1BT	2BT	M2BT (T=6)	AM2BT	Proposed
Addition (8 bit)	262144	-	-	7714	6817	<b>1296</b>
ABS (8 bit)	262144	-	-	7714	6817	<b>1296</b>
Boolean (1 bit)	-	262144	786432	786432	786432	786432
Comparison	1024	1024	1024	1084	1077	<b>1048</b>

제안하는 알고리즘에서 매우 작은 왜곡을 가지는 움직임 벡터를 걸러내기 위한 문턱값으로는 실험적으로 결정된 최적의 값인  $T_p=2+(4 \times (\sigma/100))$ 를 사용하였다. 만약 AM2BT 방법이 아닌 M2BT 방법을 사용한다면  $T_p=3$ 을 사용한다.  $NNMP_{diff} < t_c$  범위의 블록들을 최초 후보 블록들로 결정하기 위한  $t_c$  값은 16×16 블록크기 움직임 예측을 위해  $t_c=30$ 을, 8×8 블록크기 움직임 예측을 위해  $t_c=8$ 을 각각 사용하였다. 또한 같은  $NNMP_{diff}$  값을 가지는 그룹 당 최대 탐색 개수를 위한 문턱값인  $\alpha=5$ , 총 탐색의 개수를 위한 문턱값인  $\beta=30$ 을 사용하였다. 세 개의 문턱값인  $t_c$ ,  $\alpha$ 와  $\beta$  값은 추가 탐색을 위한 복잡도와 움직임 예측의 정확도 사이에 최상의 결과를 도출하기 위해 실험에 의해 결정된 값들이다.

표 5, 6, 7은 제안한 알고리즘의 최종 성능 측정의 결과를

보여준다. 표 5의 16×16 블록크기 움직임 예측을 위한 결과를 보면, "Football" 영상에서 MB 당 평균 SSAD 추가 탐색의 수는 53f에서 9f로 현저하게 감소하였고, PSNR은 심지어 더욱 증가하였다. 표 6은 8×8 블록크기 움직임 예측을 위한 실험 결과를 보여준다. 마찬가지로 8×8 크기에서도 종래의 방법들보다 추가 탐색을 위한 정확성은 증가하였고, 복잡도는 감소되었다. 제안하는 알고리즘의 10개 실험 영상들에 대한 평균 결과를 보면, AM2BT에 비해 16×16 블록크기에서 추가된 평균 탐색 수는 75%까지 감소되었고, PSNR도 0.19dB 증가하였다. 8×8 블록크기에서도 PSNR이 0.29dB까지 증가하였고, 평균 탐색 수는 약 30% 감소되었음을 볼 수 있다. 표 7에서는 제안하는 알고리즘의 연산 복잡도 비교에 대한 보다 정확한 결과를 볼 수 있다. 16×16 블록크기에 대한 실험에서 M2BT와 AM2BT는 MB당 평균

적으로 각각 7714, 6817번의 절대값 및 덧셈 연산을 필요로 하는 반면, 제안하는 알고리즘은 단지 1296번의 연산을 요구하며, 이것은 MB당 절대값 및 덧셈 연산의 약 80%가 감소됨을 의미한다.

## V. 결론

본 논문에서는 보완된 이진 블록 매칭 움직임 예측 방법의 효율적인 추가 탐색 방법을 위해  $NNMP_{diff}$  기반 탐색 알고리즘을 제안 하였다. 첫 번째 단계에서 매우 작은 SSAD 값을 가지는 2BT 움직임 벡터가 최종 움직임 벡터로 결정되고, 두 번째 단계에서는  $NNMP_{diff}$  기반 추가 탐색 방법에 의해 움직임 벡터를 결정한다.  $NNMP_{diff}$  값에 의해 분류된 탐색 범위안의 모든 후보 블록들 중 세 개의 문턱값 ( $t_c, \alpha, \beta$ ) 범위 안의 후보 블록들이 최종 움직임 벡터를 결정하기 위해 추가적으로 탐색된다. 실험 결과는 이러한  $NNMP_{diff}$  기반 추가 탐색 방법이 종래의 방법보다 더 낮은 복잡도를 제공할 뿐만 아니라 더 정확한 움직임 예측을 수행하는 것을 보여준다. 특히  $16 \times 16$  블록크기를 위한 움직임 예측에서, 제안하는 알고리즘이 추가 탐색을 위한 연산량을 줄이는데 있어서 매우 좋은 성능을 보이는 것을 볼 수 있다. 즉  $NNMP_{diff}$  기반 추가 탐색 알고리즘을 통하여 약간의 추가 탐색으로 전역 탐색 움직임 예측 대비 열화된 이진 블록 매칭 방법의 PSNR을 상당히 증가시킬 수 있으며, 이는

$NNMP_{diff}$  기반 탐색이 이진 기반 움직임 예측의 추가 탐색을 위한 하나의 좋은 기준 (criterion)으로 유용하게 활용될 수 있음을 보여준다.

## 참고 문헌

- [1] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-Complexity Block-based Motion Estimation via One-Bit Transforms," IEEE Trans. Circuits and Systems for Video Tech., vol. 7, no. 5, pp. 702-706, Aug. 1997.
- [2] A. Erturk and S. Erturk, "Two-Bit Transform for Binary Block Motion Estimation," IEEE Trans. Circuits and Systems for Video Tech., vol. 15, no. 7, pp. 938-946, July 2005.
- [3] P. H. W. Wong and O. C. Au, "Modified one-bit transform for motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 9, no. 7, pp. 1020-1024, Oct. 1999.
- [4] B. Demir and S. Ertürk, "Block motion estimation using modified two bit transform," Lect. Notes in Computer Science, vol. 4263, pp. 522-531, 2006.
- [5] B. Demir and S. Ertürk, "Block Motion Estimation Using Adaptive Modified Two-Bit Transform," IET Image Process., vol. 1, no. 2, pp. 215-222, June 2007.
- [6] Z. He, C. Tsui, K. Chan, and M. Liou, "Low-power VLSI design for motion estimation using adaptive pixel truncation," IEEE Trans. Circuits and Systems for Video Tech., vol. 10, no. 5, pp. 669-678, Aug. 2000.
- [7] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit-plane matching," in IEEE Int. Conf. Image Processing, Washington, DC, 1995, pp. 496 - 499.
- [8] Sarp Erturk, "Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation," IEEE Signal Processing Letters, vol. 14, no. 2, 109-112, Feb. 2007.

---

## 저 자 소 개



### 임진호

- 2003년 2월 : 서강대학교 전자공학과 졸업
- 2003년 2월 ~ 현재 : 삼성전자 무선사업부 책임연구원
- 2010년 3월 ~ 현재 : 한양대학교 전자통신컴퓨터공학과 석사과정
- 주관심분야 : 영상처리, 영상압축, 영상화질개선

---

저 자 소 개

---



정 제 창

- 1980년 2월 : 서울대학교 전자공학과 졸업
- 1982년 2월 : KAIST 전기전자공학과 석사
- 1990년 : 미국 미시간대학 전기공학과 공학박사
- 1980년 ~ 1986년 : KBS 기술연구소 연구원(디지털 및 뉴미디어 연구)
- 1990년 ~ 1991년 : 미국 미시간대학 전기공학과 연구교수(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
- 2007년 : IEEE Chester Sall Award 수상
- 2008년 : ETRI Journal Paper Award 수상
- 2011년 5월 : 제46회 발명의 날 녹조근정훈장 수훈
- 주관심분야 : 영상처리, 영상압축, 3DTV