

초기 매칭 에러를 통한 적응적 고속 움직임 예측 알고리즘

정 태 일^{*}

요 약

본 논문에서는 기존의 PDE방법의 계산량을 줄이고 동시에 동일한 예측화질을 얻기 위해, 초기 매칭블록의 서브블록별 SAD값의 비교를 통한 고속 움직임 예측 알고리즘을 제안한다. 제안한 방법은 최초 매칭 블록의 서브블록별 매칭 에러의 복잡도의 비교를 통해 후보블록들의 매칭순서를 변경하여 불필요한 계산량을 줄이는 방법을 제안한다. 제안한 알고리즘은 예측 화질의 저하 없이 기존의 PDE (partial distortion elimination) 알고리즘을 이용한 전영역 탐색 방법에 비해 45%의 계산량을 줄였으며, MPEG-2 및 MPEG-4 AVC를 이용하는 비디오 압축 응용분야에 유용하게 사용될 수 있을 것이다.

An Adaptive and Fast Motion Estimation Algorithm using Initial Matching Errors

Tae-Il Jeong^{*}

ABSTRACT

In this paper, we propose a fast motion estimation algorithm using initial matching errors by sorting square sub-blocks to find complex sub-block area adaptively based on partial calculation of SAD(sum of absolute difference) while keeping the same prediction quality compared with the PDE(partial distortion elimination) algorithm. We reduced unnecessary calculations with square sub-block adaptive matching scan based initial SAD calculation of square sub-block in each matching block. Our algorithm reduces about 45% of computations for block matching error compared with conventional PDE(partial distortion elimination) algorithm without any degradation of prediction quality, and our algorithm will be useful to real-time video coding applications using MPEG-4 AVC or MPEG-2.

Key words: Motion Estimation(움직임 추정), Block Matching(블록매칭), Partial Distortion elimination (PDE), Successive elimination algorithm(SEA)

1. 서 론

비디오 코딩 기술에서 사용되는 표준은 움직임 추정, 움직임 보상, 이산 직교변환(DCT)을 이용한 변환, 양자화, 엔트로피 부호화 등으로 구성된다. 이들 가운데 움직임 보상은 이전프레임 또는 이후 프레임에서부터 예측의 효율을 향상시키는데 사용되어진다. 움직임 추정은 인접 프레임간의 움직임을 측정

하는 과정이다. 움직임 추정은 인코딩 시간에 가장 많은 영향을 미친다. 블록매칭 방법은 이러한 움직임 추정과정에서 가장 많이 사용되는 방법으로 각각의 블록 당 이전프레임에서 현재 프레임의 블록을 예측하기 위해 가장 유사한 블록을 찾는 과정이다. 주어진 블록마다 이러한 과정을 거쳐 움직임 벡터로 나타낸다. 전영역 탐색 방법은 주어진 탐색 영역에서 최소의 매칭 에러를 갖는 후보 지점을 찾는 것이

※ 교신저자(Corresponding Author): 정태일, 주소: 부산광역시 남구 용당동 535번지(608-711), 전화: (051)610-8383, FAX: (051)610-8846, E-mail: tijeong@hanmail.net

접수일: 2007년 9월 21일, 완료일: 2007년 11월 22일

^{*} 정회원, 동명대학교 정보통신공학과

다. 이 탐색 방법은 간단하고 쉬운 하드웨어 구현 때문에 비디오 데이터 부호화에서 널리 사용되어져 왔다. 하지만 큰 탐색 영역을 가질 때, 이 방법의 방대한 계산량은 실시간 부호화 응용 분야에서 심각한 문제점으로 남겨져 왔다. 지난 십 수년간 전 영역 탐색 방법의 계산량을 줄이기 위해 많은 고속 알고리즘들이 연구 되어져 왔는데, 이들 고속 알고리즘들은 크게 두 그룹으로 나누어 질 수 있다. 하나는 전 영역 탐색 방법에 비해 예측 화질의 손실을 갖는 것이고, 다른 하나는 예측 화질의 손실을 갖지 않는 방법이다. 전자는 다음의 세부 그룹으로 다시 나누어 질 수 있다. 단일 에러 표면 가정(unimodal error surface assumption-UESA) 기법, 다해상도(multiresolution) 기법, 움직임 벡터의 시/공간 상관관계를 이용한 가변 탐색 영역 기법, 매칭 에러의 문턱값을 이용한 중간 멈춤 기법, 매칭 블록의 행/열의 투영(integral projection technique) 기법, 더 낮은 비트 해상도(low bit resolution) 기법, 매칭 블록의 서브 샘플링(subsampling) 기법 등이 있다[1-4]. 그리고 후자의 무 손실 움직임 예측 기법에는 다음과 같은 것들이 있다. 기준 블록과 후보 블록의 블록합을 이용한 후보 제거 알고리즘(successive elimination algorithm-SEA) 및 그것의 변형된 알고리즘들[5-10], 고속의 2차원 FIR필터링 방법을 이용한 고속 알고리즘[11], 기준 블록과 후보 블록에 대해 수직, 수평, 블록 전체의 투영(vertical, horizontal and massive projection)을 이용한 알고리즘[12-14], 그리고 부분 매칭 에러값을 이용한 후보 제거(partial distortion elimination-PDE) 알고리즘 및 그 변형 알고리즘[15-19] 등이 있다.

본 논문에서는 PDE에 기초한 초기 매칭에러의 비교를 통한 적응적 후보 벡터의 고속 제거 알고리즘을 제안한다. 제안된 방법은 기준 블록과 후보 블록들에서 블록 매칭 에러가 기준 블록의 기울기 크기에 비례한다는 것을 이용하여 초기 매칭에러의 차이에 기초한 적응적 매칭 스캔 방법을 가지고 불필요한 계산을 효율적으로 줄였다. 제안된 알고리즘은 기존의 SEA 기반 알고리즘들에 적용하면 보다 효율적으로 계산량을 줄일 수 있을 것이다. 제안된 알고리즘은 기존의 PDE 알고리즘을 이용한 전 영역 탐색 방법에 비해 약 45%의 계산량 감소를 보였다.

2. 관련연구

2.1 전 영역 탐색 방법(Full Search algorithm)

전 영역 탐색 방법은 탐색영역 내 모든 후보 블록들을 체크한 후 매칭 에러가 가장 작은 후보블록을 최선의 매칭 점으로 선택한다. 오차값을 측정하는 많은 방법들이 제안되었고, 이들 가운데 블록 매칭 에러(sum of absolute difference-SAD)가 가장 많이 쓰인다. $N \times N$ 의 크기를 갖는 MB(Macro Block)에서 $SAD(x,y)$ 는 식(1)과 같다.

$$SAD(x,y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_n(i,j) - f_{n-1}(i+x,j+y)| \quad (1)$$

식 (1)에서, $f_n(i,j)$ 는 현재 MB의 픽셀(i,j)값을 나타내고 $f_{n-1}(i+x,j+y)$ 는 이전 프레임의 후보블록의 픽셀 값을 나타낸다.

2.2 SEA 방법

SEA는 부등식 $|a+b| \leq |a|+|b|$ ($a, b \in R$)에 기초하여 기준 블록의 합과 후보 블록의 합 그리고 그때까지의 최소의 매칭 에러를 이용하여 후보 지점들을 제거하는 것이다[2]. 이 알고리즘은 연속 제거 알고리즘이라 지칭되는데, 먼저, 기준 블록의 합과 후보 블록합의 중복 계산을 줄이기 위해 행의 합 또는 열의 합을 계산한다. 탐색 영역의 중앙점 또는 움직임 벡터의 상관관계(correlation)로 부터 정해진 후보 점의 초기 매칭 에러를 구한 후, 식 (2)를 가지고 예측 화질의 열화 없이 불필요한 후보점들을 제거한다. 식 (2)에서 R 은 현재 프레임의 기준블록을 의미하고 $C(x,y)$ 는 이전 프레임에서 후보 벡터를 갖는 후보블록의 합을 나타낸다. SAD_{min} 은 그때까지의 최소의 매칭 에러를 나타낸다. 이전 프레임의 후보 블록 합이 식 (2)의 조건을 만족하면 두 블록간의 SAD를 계산하며, 그렇지 않으면 그 후보점은 무시하고 다음 후보점을 조사한다. 이 방법은 초기에 매칭된 에러를 잘 구하느냐에 따라 부호화 속도가 결정된다.

$$R - SAD_{min} \leq C(x,y) \leq R + SAD_{min} \quad (2)$$

2.3 MSEA(Multilevel SEA)

X.Q. Gao 는 식 (2)를 일반화하며 매칭 블록을

다음과 같이 서브 블록화 하여 표현한다. Level 0에서의 SAD값을 $SSAD_0$ 로 표현하고 level 1에서 SAD값을 $SSAD_1$ 로 표현한다. 이런식으로 부등식의 경계조건을 좁게 설정함으로써 불가능한 후보 벡터들이 많이 줄일 수 있었다[6]. 이 식을 일반화하면

$$SSAD_L = \sum_{u=0}^{2^L-1} \sum_{v=0}^{2^L-1} |R_L^{(u,v)}(i,j) - C_L^{(u,v)}(i+x, j+y)| \quad (3)$$

$$SSAD_0 \leq SSAD_1 \leq \dots \leq SSAD_L \leq \dots \leq SSAD_L \leq SAD_{min} \quad (4)$$

2.4 PDE 방법

이 방법은 하나의 블록의 완전한 계산을 하기 전에 부분 매칭 에러를 가지고 최소 에러와 비교함으로써 불필요한 계산을 줄이는 것이다[7-8]. 즉, 매칭 에러의 중간 합이 그때까지의 최소 에러보다 크다면 나머지 계산을 할 필요가 없다는 것이다. 식 (5)는 일반적인 SAD에서 각 행마다 중간 계산을 점검하도록 고친 것이다.

$$\sum_{i=1}^k \sum_{j=1}^N |f_t(i, j) - f_{t-1}(i+x, j+y)| \quad k=1, 2, \dots, N \quad (5)$$

앞에서 언급했듯이, $k < N$ 일 때 부분 매칭 에러 합이 그때까지의 최소 에러보다 크다면 $k+1$ 부터 N 까지의 행에 대한 매칭 에러 계산을 절약 할 수 있다. PDE 방법은 전 영역 탐색 방법 뿐만 아니라 다른 고속 탐색 방법에서도 계산을 효율적으로 줄이기 위해 사용되어질 수 있다. PDE 알고리즘의 계산의 감소는 주어진 탐색영역에서 전체 최소 에러를 가능한 빨리 찾는 것과 해당 매칭 블록에서 매칭 에러가 큰 영역을 먼저 계산함으로써 얻을 수 있다.

3. 제안한 알고리즘

PDE의 변형된 알고리즘들은 시/공간적으로 인접한 움직임 벡터의 사용, 나선형 탐색방법, 그리고 이들 개념을 복합한 방법들이다[2-3]. PDE알고리즘에서 중요한 것은 불가능한 후보 벡터들을 얼마나 빨리 검출해서 이를 제거 하느냐에 있다. 탐색 방법에 있어서는, 주어진 탐색 영역에서 나선형 탐색 방법을 사용한다. 나선형 탐색 방법을 갖는 PDE 알고리즘이 단순한PDE 알고리즘보다 더 많은 계산량을 줄임을

알 수 있다[2-3]. 따라서 제안하는 알고리즘에서는 나선형 탐색을 함께 사용 할 것이다.

우선 영상의 복잡도와 매칭에러의 관계를 정리 하기 위해 Taylor 급수전개(Taylor Series Expansion)을 이용하여 매칭 에러와 기준 블록의 기울기 크기와 관계를 정리 할 것이다. 가령, (t+1) 번째 프레임의 어떤 위치 $p=(x,y)$ 에서 화소값을 $\{f_{t+1}(p), p=(x,y)\}$ 이라 하자. 그리고 그 위치의 움직임벡터를 $mv=(m_{vx}, m_{vy})$ 라고 하자. 식 (6)에서처럼 이전 프레임과 현재프레임 사이의 관계를 기술 할 수 있다. 매칭 에러와 기준 블록의 기울기 크기간의 관계를 식 (7)에서처럼 나타낼 수 있다. 여기서, $cmv=(cm_{vx}, cm_{vy})$ 는 그 매칭 에러에 해당하는 후보 벡터를 나타낸다. 식 (7)로부터, 탐색 영역 안에서 기준 블록과 후보 블록의 매칭 에러는 기준 블록의 기울기 크기에 비례한다는 것을 알 수 있다. 즉, 영상의 복잡도에 따라 매칭 에러가 비례한다는 것을 의미한다.

$$f_{t+1}(p) = f_t(p+mv) \quad (6)$$

$$\begin{aligned} d_{t+1}(p) &= |f_{t+1}(p) - f_t(p+cmv)| \quad (7) \\ &= |f_t(p+mv) - f_t(p+cmv)| \\ &\approx \left| \frac{\partial f_t(p+mv)}{\partial x} (cm_{vx} - m_{vx}) + \frac{\partial f_t(p+mv)}{\partial y} (cm_{vy} - m_{vy}) \right| \\ &\approx \left| \frac{\partial f_{t+1}(p)}{\partial x} (cm_{vx} - m_{vx}) \right| + \left| \frac{\partial f_{t+1}(p)}{\partial y} (cm_{vy} - m_{vy}) \right| \end{aligned}$$

복잡도의 최적화를 위해 제안한 방법은 매칭 스캔에 영상의 복잡도를 잘 적용하기 위해 4x4서브 블록들을 이용한 방법이 1x16 벡터들을 이용한 방법보다 영상의 복잡도를 더 잘 영역화 할 것이다.

불필요한 후보블록을 미리 제거하기 위한 확률을 높이기 위해서는 효율적인 스캔방법을 찾아야 한다. 매칭에러가 큰 영역을 우선적으로 만날 확률을 높이기 위한 매칭 스캔 순서를 정하기 위해 최초의 SAD를 계산할 때 각 4x4서브블록의 부분 SAD를 계산하고 각 서브블록들 중 SAD값이 큰 순서대로 정렬하여 서브블록들 중에서 매칭에러가 큰 값이 우선적으로 나올 확률을 높여 SAD_{min} 값과 비교하여 고속으로 제거하는 알고리즘을 제안한다. 제안한 방법은 최초 후보블록의 SAD를 구할 때 부가적인 계산량 없이

4×4형태의 서브블록의 SAD값만을 가지고 이들을 비교하여 각 블록마다 적응적으로 매칭에러를 정렬함으로써 추가적인 계산량 없이 적응적인 매칭 스캔 순서를 구할 수 있다. 기존의 변형된 PDE알고리즘에서 제시된 방법들은 4×4 서브 블록으로 나누어 순차적으로 스캔하는 방법, 후보블록간의 복잡도를 이용한 방법 그리고 4×4서브블록에 dithering order를 적용한 방법이 있다. 순차적인 방법은 매칭 에러를 구하는데 있어 매칭에러의 크기와 상관없이 순차적으로 비교하므로 후보블록간의 특성을 적용할 수 없는 단점이 있다. 후보블록간의 복잡도를 이용한 방법은 각 블록의 복잡도를 측정하기 위해 부가적인 계산을 필요로 한다. 검색할 후보점이 많을 경우 이 방법이 효율적이지만 최종 후보점의 개수가 적을 때는 부적합하다. Dithering order를 이용한 방법 역시 후보블록간의 특징을 파악해서 적용할 수 없는 단점이 있다. 따라서 후보블록간의 특징을 파악하여 매칭에러가 큰 영역을 우선적으로 만날 확률을 높이는 스캔방법이 효율적이라 볼 수 있다. 이러한 매칭 스캔 순서를 정하는 방법으로 나선형 스캔을 적용하여 최초의 SAD를 계산할 때 4×4 형태의 서브블록의 매칭에러를 계산하고 각 서브블록들 중 에러가 큰 순서대로 정렬하여 후보블록들 중에서 매칭에러가 큰 값이 우선적으로 나올 확률을 높여 고속으로 제거하는 알고리즘을 제안한다. 제안한 방법은 최초 후보블록의 SAD를 구할 때 부가적인 계산량 없이 4×4 형태의 서브블록의 SAD값만을 가지고 이들을 비교하여 각 블록마다 적응적으로 매칭에러를 정렬함으로써 추가적인 계산량 없이 적응적인 매칭 스캔순서를 구할 수 있다.

$$\sum_{k=1}^k \sum_{i=1}^{N/s} \sum_{j=1}^{N/s} |f_t(i, j) - f_{t-1}(i+x, j+y)| \quad (8)$$

$k = 1, 2, \dots, N/s \times N/s, s \in \text{sorting_order} []$

식(8)의 sorting_order는 SAD를 계산할 때 스캔 순서를 형태의 서브 블록 별로 구해 큰 값을 기준으로 정렬하는 방법을 사용한다. sorting_order를 사용하여 정렬하여야 할 형태의 서브블록의 개수는 16개이다. 이들 16개의 후보블록들의 스캔순서를 각 블록 별로 적응적으로 구하여 매칭에러가 큰 값이 우선적으로 나올 수 있는 확률을 높였다. 형태의 서브 블록의 부분SAD값들을 정렬하기 위해 사용되는 sorting_order에 사용되는 계산량은 무시할만한 아주 작

은 부분이므로 무시하여도 좋을 정도이다. 제안한 방법과 같은 PDE 방법은 전 영역 탐색 방법 뿐만 아니라 다른 고속 탐색 방법에서도 계산을 효율적으로 줄이기 위해 사용되어질 수 있으므로 제안한 방법으로 SEA기반의 두 방법(SEA, Multilevel SEA)에 적용하여 계산량을 줄이고자 한다.

SEA에 제안한 방법을 적용하는 경우는 우선 SEA를 통하여 식(2)의 부등식의 경계조건을 사용하여 우선적으로 이 부등식을 만족하지 못하는 후보블록을 제거하고 경계조건을 만족하는 후보블록들 가운데 매칭에러가 큰 서브블록이 나올 확률을 높이기 위해 각 블록별로 적응적 스캔 방법을 사용하여 보다 빠르게 후보 블록을 제거하였다.

Multilevel SEA의 경우 블록의 크기가 이고 서브블록의 크기가 이므로 식(3) 처럼 연속해서 부등식의 레벨들을 적용해 나가면 서브블록의 크기보다 작아지게 된다. 따라서 제안한 방법은 Level 0과 Level 1까지 Multilevel SEA의 부등식의 경계조건을 만족하는 후보 블록들 가운데 매칭에러가 큰 서브블록이 나올 확률을 높이기 위해 각 블록별로 적응적 스캔 방법을 사용하여 보다 빠르게 후보 블록을 제거하였다. 또한 형태의 서브블록의 SAD값을 정렬하기위해 필요한 최초의 SAD값을 계산하기 위해 영상의 시작일 경우에는 MV(motion vector)를 (0, 0)으로 지정하고 이후부터는 이전블록의 MV를 기준으로 SAD를 구하였다.

4. 실험결과 및 토의

제안된 알고리즘과 기존의 알고리즘의 성능을 비교하기 위해, "foreman", "car phone", "trevor", "akio", "claire", "grandmother"의 비디오 시퀀스를 가지고 실험을 하였다. 이들 시퀀스에서, "foreman"과 "car phone"은 다른 시퀀스에 비해 비교적 큰 움직임 을 가지고 있다. 한편 "akio", "claire", "grandmother"는 이전 두 시퀀스보다 비교적 작은 움직임을 갖는다. "trevor" 시퀀스는 이들 두 부류의 중간 정도 움직임을 갖는 것이라고 할 수 있다. 매칭블록의 크기는 16×16이며, 탐색 영역의 범위는 ±7 화소로 선택했다. 프레임의 크기는 QCIF(176×144)이다. 실험 결과는 계산된 평균 행의 수와 PSNR로 나타내었다. 계산된 평균 행의 수는 하나의 스캔 블

록에서 계산 되는 평균 행의 수치이다.

표 1~표 2 의 모든 수치는 복잡도를 측정하기 위해 모든 부가적인 계산을 고려한 값들이다. 여기서 테스트되는 모든 매칭 스캔 알고리즘은 탐색 영역에서 나선형 탐색 방법을 사용한 것이다. 이는 움직임 벡터들의 분포상태를 효율적으로 이용하는 방법으로서 PDE 및 SEA 알고리즘들에서는 널리 사용하고 있다.

그림 1~그림 2는 10Hz의 'foreman' 과 'grand

mother' 영상 시퀀스에서 서브 블록에 대해 다양한 방법으로 실험된 순차적 매칭 스캔과 최초의 후보 블록의 SAD값을 기준으로 한 적응적 매칭 스캔으로 계산된 평균 행의 수를 보여준다. 실험은 기존의 PDE방법과 제안한 적응 매칭 스캔 방법의 비교와 기존의 SEA와 MSEA방법과 제안한 적응스캔 알고리즘을 SEA방법들에 적용하였을 경우 나타난 결과를 그림으로 나타내었다. 적응적 매칭 스캔 알고리즘은 종래의 순차적인 방법에 비해 매칭 에러를 계산하

표 1. 30Hz 영상 시퀀스에서 서브 블록에 대해 다양한 매칭 스캔의 계산된 평균 행의 수.

Algorithms \ Sequences	Foreman	Car phone	Trevor	Claire	Akio	Grand
Original FS	16.00	16.00	16.00	16.00	16.00	16.00
PDE(sequential, 4x4)	4.13	3.91	3.21	4.04	1.72	3.65
PDE(proposed, 4x4)	2.86	3.16	2.43	3.26	1.36	3.45
SEA+ PDE (sequential, 4x4)	2.48	2.51	1.77	1.57	0.78	2.39
SEA+PDE(proposed, 4x4)	1.55	1.72	1.12	1.24	0.56	1.85
MSEA+PDE(sequential, 4x4)	1.42	1.66	1.01	1.12	0.59	1.82
MSEA+PDE(proposed, 4x4)	1.19	1.43	0.85	1.03	0.57	1.71

표 2. 10Hz 영상 시퀀스에서 서브 블록에 대해 다양한 매칭 스캔의 계산된 평균 행의 수.

Algorithms \ Sequences	Foreman	Car phone	Trevor	Claire	Akio	Grand
Original FS	16.00	16.00	16.00	16.00	16.00	16.00
PDE(sequential, 4x4)	5.11	4.67	4.68	4.65	2.19	4.23
PDE(proposed, 4x4)	3.64	3.75	3.53	3.54	1.54	3.80
SEA+ PDE (sequential, 4x4)	3.12	2.98	2.75	1.88	1.07	2.80
SEA+PDE(proposed, 4x4)	2.10	2.12	1.84	1.44	0.69	2.10
MSEA+PDE(sequential, 4x4)	1.88	1.99	1.60	1.29	0.71	1.99
MSEA+PDE(proposed, 4x4)	1.58	1.71	1.32	1.16	0.63	1.83

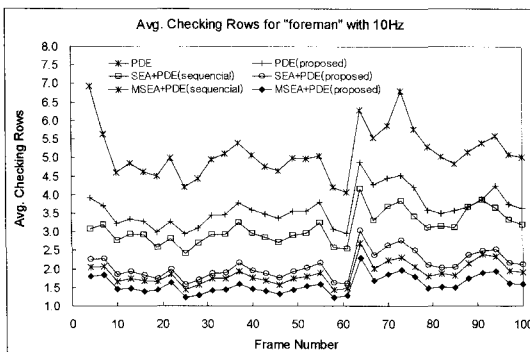


그림 1. 10Hz의 'foreman' 영상 시퀀스에서 서브 블록을 순차적인 매칭순서와 적응적인 매칭순서에 대한 다양한 방법으로 계산된 평균 행의 수.

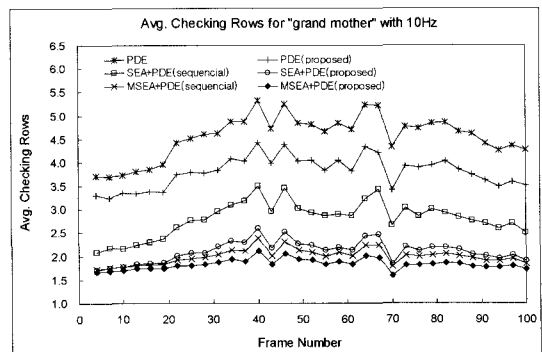


그림 2. 10Hz의 'grand mother' 영상 시퀀스에서 서브 블록의 순차적인 매칭순서와 적응적인 매칭순서에 대한 다양한 방법으로 계산된 평균 행의 수.

는데 있어 매칭에러가 큰 서브블록이 우선적으로 나올 확률을 높여 불필요한 계산을 현저히 줄이는 것이다. 그리고 이러한 적응적 매칭 스캔을 기존의 고속 알고리즘인 SEA기반방법에 적용하였을 경우 기존 방법보다 불필요한 계산을 더욱 더 줄일 수 있다는 것을 나타내고 있다.

표 1~2는 30 Hz, 10 Hz의 모든 영상 시퀀스에서 서브 블록에 대해 다양한 방법에서의 순차적인 방법과 최초의 후보블록에서의 각 서브블록의 초기 SAD 값의 초기 매칭에러를 기준으로 한 적응적 매칭 스캔으로 계산된 평균 행의 수를 보여 준다. 위에서 언급한 결과들을 이들 표에서도 동일하게 확인할 수 있다. 고속 알고리즘을 전혀 사용하지 않은 원래의 전영역 탐색 알고리즘은 표에서 보는 바와 같이 평균 계산 행의 수가 매칭 블록의 크기인 16이 된다. 위에서 나선행 탐색을 언급하였는데, 참고 문헌 [11,12]에서는 PDE 알고리즘에서 나선행 탐색 방법의 효율성과 중요성을 보여준다. 서브 블록에 대해 계산된 평균 행의 수의 계산 감축 비율을 구해 보면 다음과 같다.

우선 표 1의 'foreman' 영상 시퀀스에서 제안한 서브블록의 초기 매칭에러를 통한 적응적 스캔 알고리즘은 순차적인 매칭 스캔 알고리즘에 비해 SEA방법에 적용 하였을 때 37%의 계산 감축이 있었고, MSEA방법에 적용하였을 때 16%의 계산감축을 얻을 수 있었다.

5. 결 론

본 논문에서는 전영역 탐색에서 발생하는 많은 계산량을 최초 후보블록의 초기 매칭에러에 기반한 적응적 움직임 예측 방법을 사용하여 고속으로 불가능한 벡터를 미리 제거하는 방법으로 화질의 저하가 없는 새로운 방법을 제안한다. 제안한 방법은 영상의 복잡한 영역의 효율적인 매칭 단위와 최초 후보블록의 SAD를 계산할 때 최초의 후보블록의 매칭 오차에 따라 적응적 매칭 스캔을 이용하여 불필요한 계산을 효율적으로 줄였다. 제안된 알고리즘은 예측 화질의 저하 없이 기존의 PDE 알고리즘을 이용한 전영역 탐색 방법에 비해 45%의 계산량을 줄였으며, MPEG-2 및 MPEG-4 AVC를 이용하는 비디오 압축 응용분야에 유용하게 사용될 수 있을 것이다.

참 고 문 헌

- [1] F. Dufaus and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proceedings. IEEE*, Vol.83, pp. 858-876, Jun. 1995.
- [2] J.N. Kim and T. S. choi, "A fast motion estimation for software based real-time video coding," *Proc. ISCE*, pp. TPB1-16-TPB1-21, Oct. 1998.
- [3] X. Lee and Y.Q. Zhang, "A fast hierarchical motion-compensation scheme for video coding using block feature matching," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.6, pp. 627-635, Dec. 1996.
- [4] J.b. Xu, L.M. Po, and C.K. Cheung "A new prediction model search algorithm for fast block motion estimation," *Proc ICIP*, pp. 659-666, Aug. 1998.
- [5] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, Vol.4, pp. 105-107, Jan. 1995.
- [6] G.C. de Oliveira and A. Alcain, "On fast motion compensation algorithms for video coding," *Proc. PCS*, pp. 467-472. 1997.
- [7] J.Y. Lu, K.S. Wu, and J.C. Lin, "Fast full search in motion estimation by hierarchical use of Minkowski's inequality (HUMI)," *Pattern Recog.*, Vol.31, pp. 945-952, 1998.
- [8] M.Z. Coban and R.M. Mersereau, "A fast exhaustive search algorithm for rate-constrained motion estimation," *IEEE Trans. Image Processing*, Vol.7, pp. 769-773, May 1998.
- [9] J.N. Kim, D.K. Kang, S.C. Byun, I.L. Lee, and B.H. Ahn, "A fast full search motion estimation algorithm using sequential rejection of candidates from hierarchical decision structure," *IEEE Transactions on Broadcasting*, Vol.48, pp. 43-46, Mar. 2002.
- [10] X.Q. Gao, C.J. Duanmu, and C.R. Zou, "A

- multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. Image Processing*, Vol.9, pp. 501-504, Mar. 2000.
- [11] Y. Naito, T. Miyazaki, and I. Kuroda, "A fast full-search motion estimation method for programmable processors with a multiply-accumulator," *Proc. ICASSP*, pp. 3221-3224, 1996.
- [12] Y.C. Lin and S.C. Tai, "Fast full-search block-matching algorithm for motion-compensated video compression," *IEEE Trans. Communication*, Vol.45, pp. 527-531, May 1997.
- [13] V.L. Do and K.Y. Yun, "A low-power VLSI Architecture for full-search block-matching motion estimation," *IEEE Trans. Circuits Syst. for Video Technol.*, Vol.8, pp. 393-398, Aug. 1998.
- [14] J.N. Kim, et al., "Adaptive matching scan algorithm based on gradient magnitude for fast full search in motion estimation," *IEEE Trans. Consumer Electronics*, Vol.45, pp. 762-772, Aug. 1999.
- [15] J.N. Kim, et al., "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan," *IEEE Trans. Circuits Syst. for Video Technol.*, Vol.10, pp. 1040-1048, Oct. 2000.
- [16] J.N. Kim, S.C. Byun, Y.H. Kim, and B.H. Ahn, "Fast Full Search Motion Estimation Algorithm Using Early Detection of Impossible Candidate Vectors," *IEEE Trans. Signal Processing*, Vol.50, pp. 2355-2365, Sept. 2002.
- [17] B. Erol, F. Kossentini, and H. Alnuweiri, "Efficient Coding and mapping algorithms for software-only real-time video coding at low bit rates," *IEEE Trans. Circuits Syst. for Video Technol.*, Vol.10, pp. 843-856, Sept. 2000.
- [18] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," *Proc. SPIE*, Vol.2419, pp. 100-118, 1995.
- [19] ITU-T Recommendation H.263 software implementation, Digital Video Coding Group at Telenor R&D, 1995.



정 태 일

1994년 부경대학교 전자공학과 졸업(공학사)
 1997년 부경대학교 대학원 전자공학과 졸업(공학석사)
 2001년 부경대학교 대학원 전자공학과 졸업(공학박사)
 2000년~현재 동명대학교 정보통신공학과 전임

관심분야 : 신호처리, 워터마킹, 멀티미디어 통신