

전영역 탐색 방식의 움직임 예측에서 매칭 스캔 방법을 이용한 불가능한 후보 벡터의 고속 제거 알고리즘

김 종 남[†]

요 약

움직임 예측의 전영역 탐색 (full search)에서 방대한 계산량은 실시간 비디오 압축에 큰 장애물이 되어 왔으며, 이는 최근 MPEG-4 AVC (Advanced Video Coding) 표준이 기존의 MPEG-2 보다 움직임 예측에서 더 많은 계산량을 요구하고 있다. 전영역 움직임 예측의 계산량을 줄이기 위해, 본 논문에서는 기존의 전영역 움직임 예측에 비하여 예측 화질의 저하가 없는 새로운 고속 매칭 알고리즘을 제안한다. 불가능한 후보의 움직임 벡터를 빨리 제거함으로써 예측 화질의 저하 없이 계산량만 줄일 수 있게 되는 것이다. 본 논문에서는 영상의 복잡한 영역의 효율적인 매칭 단위와 디더링 (dithering) 순서에 기초한 매칭 방식을 통하여 불가능한 후보 벡터를 더 빨리 제거한다. 제안된 알고리즘은 예측 화질의 저하 없이 기존의 PDE (partial distortion elimination) 알고리즘을 이용한 전영역 탐색 방법에 비해 30%의 계산량을 줄였으며, MPEG-2 및 MPEG-4 AVC를 이용하는 비디오 부호화 응용분야에 유용하게 사용될 수 있을 것이다.

A Fast Elimination Algorithm of Impossible Candidate Vectors Using Matching Scan Method in Motion Estimation of Full Search

Jong-Nam Kim[†]

ABSTRACT

Significant computations for full search (FS) motion estimation have been a big obstacle in real-time video coding and recent MPEG-4 AVC (advanced video coding) standard requires much more computations than conventional MPEG-2 for motion estimation. To reduce an amount of computation of full search (FS) algorithm for fast motion estimation, we propose a new and fast matching algorithm without any degradation of predicted images like the conventional FS. The computational reduction without any degradation in predicted image comes from fast elimination of impossible candidate motion vectors. We obtain faster elimination of inappropriate motion vectors using efficient matching units from localization of complex area in image data and dithering order based matching scan. Our algorithm reduces about 30% of computations for block matching error compared with the conventional partial distortion elimination (PDE) algorithm, and our algorithm will be useful in real-time video coding applications using MPEG-4 AVC or MPEG-2.

Key words: Data Compression(데이터 압축), Motion Estimation(움직임추정), Full Search(전영역 탐색), Block Matching(블록 매칭), Matching Scan(매칭 스캔)

※ 교신저자(Corresponding Author): 김종남, 주소: 부산시 남구 대연3동(608-737), 전화: 051)620-6494, FAX: 051)620-6450, E-mail: jongnam@pknu.ac.kr
접수일: 2005년 2월 7일, 완료일: 2005년 4월 8일

[†] 부경대학교 전자컴퓨터정보통신공학부 교수

※ 본 연구는 한국학술진흥재단에서 주관하는 “차세대물류IT기술연구사업단”과 “신진교수과제” 및 한국산업기술재단에서 주관하는 “지역혁신인력양성사업”에 의해 지원 받은 연구임.

1. 서 론

병진 움직임 모델에 기반한 블록 매칭 알고리즘에서 전영역 탐색 방식은 주어진 탐색 영역에서 최소의 매칭 에러를 갖는 후보 지점을 찾는 것이다. 이 탐색 방식은 간단하고 쉬운 하드웨어 구현 때문에 비디오 데이터 부호화에서 널리 사용되어져 왔다. 하지만 큰 탐색 영역을 가질 때, 이 방식의 방대한 계산량은 실시간 부호화 응용 분야에서 심각한 문제점으로 남겨져 왔다. 지난 십 수년간 전 영역 탐색 방식의 계산량을 줄이기 위해 많은 고속 알고리즘들이 연구 되어져 왔는데, 이들 고속 알고리즘들은 크게 두 그룹으로 나누어 질 수 있다. 하나는 전 영역 탐색 방식에 비해 예측 화질의 손실을 갖는 것이고, 다른 하나는 예측 화질의 손실을 갖지 않는 방식이다. 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 서브샘플링(subsampling) 기법 등이 있다[1-3]. 그리고 후자의 무 손실 움직임 예측 기법에는 다음과 같은 것들이 있는데, 기준 블록과 후보 블록의 블록합을 이용한 후보 제거 알고리즘 (successive elimination algorithm--SEA) 및 그 것의 변형된 알고리즘들[4-10], 고속의 2차원 FIR 필터링 방법을 이용한 고속 알고리즘[11], 기준 블록과 후보 블록에 대해 수직, 수평, 블록 전체의 투영 (vertical, horizontal and massive projection) 을 이용한 알고리즘[12,13], 그리고 부분 매칭 에러값을 이용한 후보 제거 (partial distortion elimination - PDE) 알고리즘 및 그 변형 알고리즘[14-16] 등이 있다. 이들 방법 중에서 실시간 하드웨어 구현을 고려한다면 SEA와 PDE 방법이 그 후보가 될 수 있을 것이다.

본 논문에서는 PDE 방법에서 최적화된 매칭 (matching) 단위와 그 순서를 결정하여 기존의 PDE 방법들에 비하여 계산량을 현저히 줄였다. 또한, 최적화된 매칭 단위와 그 순서를 결정할 때 부가적인 계산이 필요한데, 이런 부가적인 계산이 필요 없이 기존의 매칭 단위와 순서에 비하여 계산량 감소를

얻을 수 있는 디더링 (dithering) 순서를 이용하여 계산량을 줄였다. 기준 블록과 후보 블록들에서 블록 매칭 에러가 기준 블록의 기울기 크기에 비례한다는 것을 이용하여 종래의 순차적인 매칭 스캔 (scan) 방법에 비해 복잡도와 디더링 순서에 기초한 매칭 스캔 방법을 가지고 불필요한 계산을 효율적으로 줄인 것이다. 그리고 영상의 복잡도를 이용할 때 행 벡터의 단위를 가지고 수행하는 것 보다 정사각형의 서브블록을 가지고 하는 것이 더 효율적임을 확인하였다. 위에서 언급했듯이, 디더링 순서에 기초한 매칭 스캔 방법은 매칭 스캔 순서를 결정하기 위해 어떠한 부가적인 계산도 요구하지 않는다. 실험적으로, 제안된 알고리즘은 예측 화질과 계산 감축의 견지에서 아주 효율적임과 동시에 기존의 PDE 알고리즘을 이용한 전영역 탐색 방법들에 비해 약 30%의 계산량 감소를 얻었다.

본 논문은 다음과 같이 구성 되어 있는데, 2장에서는 본 논문의 배경 연구로서 기존의 고속 전영역 탐색 알고리즘을 정리할 것이다. 3장에서는 기준 블록의 기울기 크기와 매칭 에러의 관계를 제시하며, 영상의 복잡도에 근거한 매칭 스캔 알고리즘과 서브블록을 이용한 복잡도 계산 및 매칭 스캔, 그리고 부가적인 계산을 필요치 않는 디더링 순서에 의한 매칭 스캔 알고리즘을 제안 할 것이다. 그리고 4장에서는, 본 논문의 알고리즘 검증을 위한 실험 결과 및 분석이 기술될 것이다. 마지막 5장에서 본 논문의 알고리즘 및 실험 결과 요약으로 마무리를 할 것이다.

2. 기존의 고속 전영역 탐색 알고리즘

고속 움직임 예측 알고리즘의 대부분은 정확하지 못한 움직임 벡터 때문에 예측 화질의 열화를 초래한다. 특히, 비트율이 제한된 응용 분야에서, 고속 알고리즘으로 인한 부정확한 움직임 벡터는 증가된 예측 에러로 인해 심각한 문제를 야기시킬 수 있다. 따라서, 이러한 문제점을 배제할 수 있고 하드웨어 구현이 간단하면서도 쉬운 전영역 탐색 방법이 널리 사용되었다. 움직임 예측을 위한 대부분의 연구들은 전영역 탐색에 비해 화질 저하를 초래하는 손실 움직임 예측 방법들이고 일부의 방법들만 무손실 움직임 예측방법이며 이는 다음과 같다.

고속 전영역 탐색 알고리즘들 중에 하나는 기준 블록의 합과 후보 블록의 합 그리고 그 때 까지의

최소의 매칭 에러를 이용하여 후보 지점들을 제거하는 것이다[4]. 이 알고리즘 (successive elimination algorithm-SEA)은 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)로부터 정해진 후보점의 초기 매칭 에러를 구한 후, 불필요한 후보점들을 제거한다. 그 이후에, SEA에 기초한 몇몇의 변형 알고리즘들이 발표되어 왔다. SEA의 속도 문제는 초기 매칭 에러를 어떻게 잘 선택하느냐에 따라 크게 좌우된다. 이런 맥락에서, G.C. de Oliveira[5]는 이웃하는 움직임 벡터로부터 더 작은 매칭 에러를 갖는 변형된 알고리즘을 제안했다. J.Y. Lu[6]은 다섯 개의 레벨을 갖는 민코우스키 부등식 (Minkowski's inequality)을 이용하여 후보 점의 개수를 더 많이 줄였다. M.Z. Coban[7]는 최적화된 비트율-외곡을 갖는 움직임 벡터를 결정하기 위해 아래 SEA의 개념을 이용하였으며, 방대한 계산을 줄이기 위해 가중 스케일을 갖는 비트율 항을 덧붙였다. H.S. Wang[8]은 SEA에 PDE 알고리즘과 제곱근과 제곱항을 이용하여 계산을 줄이는 시도를 하였다. 한편, X.Q. Gao와 T.M. Oh는 SEA개념을 일반화하며 매칭 블록을 서브 블록화하여 부등식의 경계조건을 더 좁게 설정함으로써 불가능한 후보 벡터들을 더 많이 줄였다[9,10].

Y. Naito [11]는 곱셈기와 덧셈기를 갖는 프로그래머블 프로세서 (programmable processor)를 위한 고속 전영역 탐색 알고리즘을 제안하였다. 이 방법에서는 매칭 기준으로서 평균 자승 에러 (mean square error: MSE)를 사용하였다. 하지만 이 알고리즘은 매칭 기준으로서 MSE를 사용하기 때문에 VLSI 구현에서는 적합하지 않으며, 이는 MSE의 곱셈으로 인해서 VLSI로 구현시 많은 계산량과 복잡성 때문이다. Y.C. Lin[12]는 제공한 열의 투영의 합 (vertical projections), 제공한 행의 투영의 합 (sum of squared horizontal projections), 제공한 전체 투영의 합 (sum of squared massive projection)을 가지고 고속 매칭 기준을 이용하여 고속 전영역 탐색 알고리즘을 제안 하였다. 하지만 이 방법 역시 MSE의 곱셈 계산 때문에 VLSI 구현에는 부적합한 알고리즘이라고 할 수 있다.

위의 알고리즘들 외에 계산을 효율적으로 줄일 수 있는 또 다른 알고리즘이 부분 에러 제거 (partial

distortion elimination: PDE)이다[2,3,14-16]. 이 방법은 하나의 블록의 완전한 계산을 하기 전에 부분 매칭 에러를 가지고 최소 에러와 비교함으로써 불필요한 계산을 줄이는 것이다. 즉, 매칭 에러의 중간 합이 그때까지의 최소 에러보다 크다면 나머지 계산을 할 필요가 없다는 것이다. (1)은 일반적인 SAD에서 각 행마다 중간 계산을 점점하도록 고친 것이다.

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

(1)에서 N 은 블록 크기를 의미한다. 앞에서 언급했듯이, $k < N$ 일때 부분 매칭 에러 합이 그때까지의 최소 에러보다 크다면 $k+1$ 부터 N 까지의 행에 대한 매칭 에러 계산을 절약 할 수 있다. J.N. Kim [2]에서는 기존의 순차적인 Top-to-bottom방식의 매칭 스캔 대신 영상의 기울기 크기에 따라 순차적으로 Top-to-bottom, Bottom-to-top, Left-to-right, Right-to-left 의 네 방향을 가지고 매칭 에러를 구해서 불필요한 계산을 줄였다. 그리고 J.N. Kim[3]에서는 가변적인 순차적 매칭 스캔 대신 기울기의 크기대로 매칭 스캔 순서를 재정렬 해서 매칭 에러를 구해 계산 감축을 얻었다. B. Erol[14]은 실시간 MPEG 소프트웨어 코덱을 만드는데 있어 PDE 알고리즘을 이용하였다. PDE 방법은 전 영역 탐색 방법 뿐만 아니라 다른 고속 탐색 방법에서도 계산을 효율적으로 줄이기 위해 사용되어질 수 있다. PDE 알고리즘을 가지고 움직임 벡터를 구할 때 계산의 감소는 주어진 탐색영역에서 전체 최소 에러를 가능한 빨리 찾는 것과 해당 매칭 블록에서 매칭 에러가 큰 영역을 먼저 계산함으로써 얻을 수 있다.

지금까지 SAD를 사용하는 고속 전영역 탐색 방법에서, 고속 매칭으로부터 계산 감축을 향상 시킬 수 있음이 간과 되어져 왔다. 여기서 고속 매칭이라는 것은 고정된 좌측상단에서 우측 하단으로의 매칭 스캔 대신에 더 많은 계산 감축이 가능한 다른 매칭 스캔 방법을 지칭하는데, 이전의 방법에서는 이에 대한 연구의 여지를 많이 남기고 있다. 최적화된 매칭 순서를 얻기 위해서는 부가적인 계산이 필요한데 부가적인 계산이 필요 없으면서 향상된 매칭 순서에 대한 연구 또한 기존의 연구에서는 없었다. 이런 관점에서, 새로운 고속 매칭 방법이 3장에서 제안되어질 것이다.

3. 제안된 알고리즘

PDE의 변형된 알고리즘들은 시/공간적으로 인접한 움직임 벡터의 사용, 나선형 탐색방법, 그리고 이들 개념을 복합한 방법들이다[14-16]. 탐색 방법에 있어서는, 주어진 탐색 영역에서 나선형 탐색 방법을 사용한다. 이전의 실험 결과에서도 보듯이, 나선형 탐색 방식을 갖는 PDE 알고리즘이 단순한 PDE 알고리즘보다 더 많은 계산량을 줄임을 알 수 있다[2,3]. 따라서, 제안하는 알고리즘에서는 나선형 탐색을 함께 사용 할 것이다.

SAD (Sum of Absolute Difference) 에 기초한 고속 전영역 탐색 알고리즘들에서 PDE 알고리즘 기반의 무 손실의 고속 매칭에 대한 시도는 참고 문헌 [2,3]. 이 전부일 정도이다. J.N. Kim[2]에서는 기존의 순차적인 Top-to-bottom 방식의 매칭 스캔 대신 영상의 기울기 크기에 따라 순차적으로 Top-to-bottom, Bottom-to-top, Left-to-right, Right-to-left의 네 방향의 가지고 매칭 에러를 구해서 불필요한 계산을 줄였다. 그리고 J.N. Kim[3]에서는 가변적인 순차적 매칭 스캔 대신 기울기의 크기대로 매칭 스캔 순서를 재정렬 해서 매칭 에러를 구해 계산 감축을 얻었다. PDE 알고리즘에서 중요한 것은 불가능한 후보 벡터들을 얼마나 빨리 검출해서 이를 제거 하느냐에 있다.

제안하는 알고리즘은 이렇게 불가능한 후보들을 더 빨리 검출해서 제거하는 방법을 제시한다. 그렇게 하기 위해, 기준 블록의 영상 복잡도와 이를 통해 매칭 스캔 순서의 재정렬, 그리고 디더링 순서대로 매칭 스캔을 재정렬 등을 이용하여 불필요한 계산을 효율적으로 줄일 것이다. 먼저 블록 매칭 에러가 기준 블록의 기울기 크기에 비례한다는 것은 참고 문헌 [2,3]에서 보였다. 이렇게 검증된 관계를 이용하여, 먼저 복잡도에 근거한 매칭 스캔의 새로운 알고리즘을 제안하고, 복잡도를 측정하지 않고 미리 정해진 순서로 매칭 스캔의 순서를 조정하는 알고리즘을 제안하며, 이들을 통해 계산 감축을 얻을 것이다. 이를 적절히 이용함으로써, 기존의 PDE 알고리즘보다 더 빠르며, 동시에 예측화질의 저하가 없는 움직임 예측을 얻을 것이다.

먼저, Taylor series expansion을 이용하여 매칭 에러와 기준 블록의 기울기 크기와의 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 움직임 벡터를 $mv=(mv_x, mv_y)$ 이라고 하자. (2)에서 처럼, 이전 프레임과 현재 프레임 사이의 관계를 기술 할 수 있다.

$$f_{t+1}(p) = f_t(p+mv) \tag{2}$$

$$\begin{aligned} d_{t+1}(p) &= |f_{t+1}(p) - f_t(p+cmv)| \\ &= |f_t(p+mv) - f_t(p+cmv)| \\ &\approx \left| \frac{\partial f_t(p+mv)}{\partial x} (cmv_x - mv_x) + \frac{\partial f_t(p+mv)}{\partial y} (cmv_y - mv_y) \right| \\ &\approx \left| \frac{\partial f_{t+1}(p)}{\partial x} (cmv_x - mv_x) \right| + \left| \frac{\partial f_{t+1}(p)}{\partial y} (cmv_y - mv_y) \right| \end{aligned} \tag{3}$$

변형된 Taylor series expansion을 사용하여 매칭 에러와 기준 블록의 기울기 크기간의 관계를 (3)에서 처럼 나타낼 수 있다. 여기서, $cmv=(cmv_x, cmv_y)$ 는 그 매칭 에러에 해당하는 후보 벡터를 나타낸다. (3)으로 부터, 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.

여기서 중요한 문제는 영상의 복잡도가 매칭 블록의 행 벡터 또는 열 벡터의 형태일 때가 최적이라는 것이다. 영상의 복잡도를 더 잘 측정함으로써 계산 감축을 더 얻을 수 있는데, 영상의 평균적인 성질을 볼 때 매칭 블록의 전체 길이에 걸쳐 있기 보다는 어떤 부분 영역으로 걸쳐있는 경우가 더 많다는 것이다. 본 논문에서는 먼저, 기존의 1×16의 행 벡터로서 매칭 에러를 구하지 않고 4×4의 정사각형의 서브 블록의 형태로 복잡도를 측정하여 매칭 에러를 구해 나갈 것이다. 우선 기존의 행 벡터를 가지고 PDE를 구현하는 식은 (1)처럼 나타낼 수 있다. 여기서 N 은 매칭 블록의 크기를 나타내며, k 가 증가할수록 계산이 늘어난다.

기존의 1×16의 행 벡터로서 매칭 에러를 구하지 않고 4×4의 정사각형의 서브 블록의 형태로 복잡도를 측정하여 PDE를 구현한다면, (4)처럼 나타낼 수 있다. 이는 위에서 언급한 복잡도에 근거한 매칭 스캔에서 4×4 서브 블록들을 이용한 방법이 1×16행 벡터들을 이용한 방법보다 영상의 복잡도를 더 잘 영역화 한다고 볼 수 있기 때문이다. 여기서 s 는 서브 블록의 크기를 나타내며, m 은 4×4의 정사각형의 서브 블록 형태의 복잡도로 부터 재정렬된 매칭 스캔의 순서를 나타낸다.

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

$$k = 1, 2, \dots, N/s * N/s,$$

$$m \in local_complexity_order[] \quad (4)$$

위의 복잡도에 근거한 매칭 스캔 알고리즘은 각 블록의 복잡도를 측정하기 위해 추가적인 계산을 필요로 한다. 검색할 후보점들이 많을 경우, 이 방법이 효율적이지만 참고 문헌[9] 처럼, 최종 검색 후보점의 개수가 적을 때는 부적합한 방법일 수가 있다. 그리고 PDE 알고리즘에서 보통 매칭 에러가 큰 하나의 매칭 단위에서 불가능한 후보 벡터의 진위 여부가 결정된다. 이런 관점에서 볼 때, 순차적인 매칭 스캔보다는 어느 정도 간격을 두고 매칭 에러를 구해서 큰 매칭 에러를 가지는 영역을 만날 확률을 높이는 매칭 스캔 방식이 효율적이라고 볼 수 있다. 이런 일정 간격을 두고 매칭 스캔 순서를 정하는 방법 중에서 디더링 행렬의 순서를 매칭 스캔 순서로 매핑시키는 알고리즘을 제안한다. 이를 이용하여 PDE를 구현하는 알고리즘은 (5)처럼 나타낼 수 있다. 이 식은 (4)와 비슷하며, 단지 다른 것은 매칭 스캔의 순서를 나타내는 p 이다. 이 p 는 (6)과 (7)의 반복과정을 통하여 얻을 수 있다[17]. (7)에서 $D_{i,j}$ 는 D 행렬의 i 번째 행과 j 번째 열의 원소를 지칭한다.

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

$$k = 1, 2, \dots, N/s * N/s,$$

$$p \in dithering_order[] \quad (5)$$

$$D^2 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix},$$

$$U^N = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & \dots & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \dots & 1 & 1 \end{bmatrix} \quad (6)$$

$$D^N = \begin{bmatrix} 4D^{N/2} + D_{0,0}^{N/2}U^{N/2} & 4D^{N/2} + D_{0,1}^{N/2}U^{N/2} \\ 4D^{N/2} + D_{1,0}^{N/2} & 4D^{N/2} + D_{1,1}^{N/2}U^{N/2} \end{bmatrix} \quad (7)$$

4. 실험 결과 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 움직임을 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. 매칭 블록의 크기는 16*16이며, 탐색 영역의 범위는 ±7 화소로 선택했다. 프레임의 크기는 QCIF (176 by 144)이다. 실험 결과는 계산된 평균 행의 수와 PSNR로 나타내었다. 계산된 평균 행의 수는 하나의 스캔 블록에서 계산 되는 평균 행의 수치이다. 표 1~표 2의 모든 수치는 복잡도를 측정하기 위한 모든 추가적인 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 10 Hz의 ‘foreman’과 ‘car phone’ 영상 시퀀스에서 4*4 서브 블록과 1*16행 벡터에 대해 다양한 매칭 스캔의 계산된 평균 행의 수를 보여 준다. 이 그림들로부터 우리는 몇 가지 주목할 만한 점들을 알 수 있다. 첫째, 영상의 복잡도에 근거한 매칭 스캔 알고리즘은 종래의 순차적인 방법에 비해 매칭 에러를 계산하는데 있어 불필요한 계산을 현저히 줄인다. 둘째, 종래의 순차적인 방법에 비해 디더링 순서에 근거한 매칭 스캔 알고리즘은 매칭 에러를 계산하는데 있어 불필요한 계산을 약간 더 줄인다. 이 알고리즘의 장점은 매칭 스캔의 순서를 정하기 위해 다른 추가적인 계산을 필요치 않는다는 것이다. 셋째, 복잡도에 근거한 매칭 스캔 알고리즘에서 4*4 서브 블록들을 이용한 방법이 1*16행 벡터들을 이용한 방법보다 불필요한 계산을 더 줄일 수 있다는 것이다. 이 결과는 순차적인 방법이나 디더링 순서에 근거한 매칭 스캔에서는 그렇게 현저하게 나타나지 않는다. 복잡도에 근거한 매칭 스캔에서만 분명히 나타나는데, 이는 4*4 서브 블록들을 이용한 방법이 1*16행 벡터들을 이용한 방법보다 영상의 복잡도를 더 잘 영역화 한다고 볼 수 있다.

표 1. 30 Hz 영상 시퀀스에서 4x4 서브 블록과 1x16행 벡터에 대해 다양한 매칭 스캔의 계산된 평균 행의 수

Algorithms \ Sequences	Foreman	Car hone	Trevor	Claire	Akio	Grand
Original FS	16.0	16.0	16.0	16.0	16.0	16.0
Top_to_bottom (4x4)	4.1	3.9	3.2	4.0	1.7	3.6
Dithering ordered (4x4)	3.7	3.6	3.0	3.6	1.6	3.4
Complexity localized (4x4)	2.8	3.0	2.4	3.1	1.3	2.9
Top_to_bottom (1x16)	4.1	3.9	3.1	3.8	1.5	3.5
Dithering ordered (1x16)	3.5	3.6	2.9	3.4	1.4	3.2
Complexity localized (1x16)	3.0	3.2	2.6	3.2	1.3	3.0

표 2. 10 Hz 영상 시퀀스에서 4x4 서브 블록과 1x16행 벡터에 대해 다양한 매칭 스캔의 계산된 평균 행의 수

Algorithms \ Sequences	Foreman	Car hone	Trevor	Claire	Akio	Grand
Original FS	16.0	16.0	16.0	16.0	16.0	16.0
Top_to_bottom (4x4)	5.1	4.6	4.6	4.6	2.1	4.2
Dithering ordered (4x4)	4.7	4.4	4.4	4.2	2.0	4.0
Complexity localized (4x4)	3.7	3.6	3.7	3.5	1.5	3.4
Top_to_bottom (1x16)	5.1	4.6	4.6	4.5	2.0	4.0
Dithering ordered (1x16)	4.6	4.4	4.3	4.0	1.9	3.8
Complexity localized (1x16)	4.0	3.9	4.0	3.7	1.6	3.5

표 1~2는 30 Hz, 10 Hz의 모든 영상 시퀀스에서 4x4 서브 블록과 1x16행 벡터에 대해 순차적, 디터링 순서에 근거한, 그리고 복잡도에 크기에 기초한 다양한 매칭 스캔의 계산된 평균 행의 수를 보여 준다. 위에서 언급한 결과들을 이들 표에서도 동일하게 확

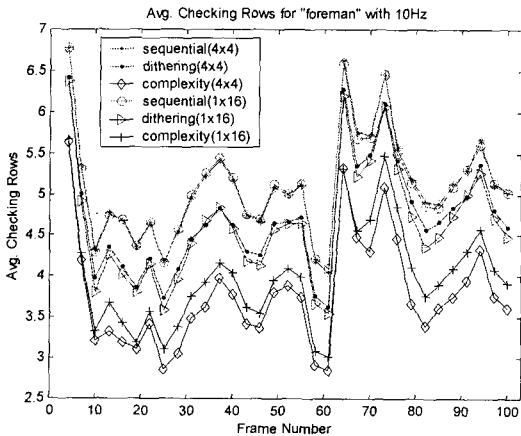


그림 1. 10 Hz의 'foreman' 영상 시퀀스에서 4x4 서브 블록과 1x16행 벡터에 대해 다양한 매칭 스캔의 계산된 평균 행의 수

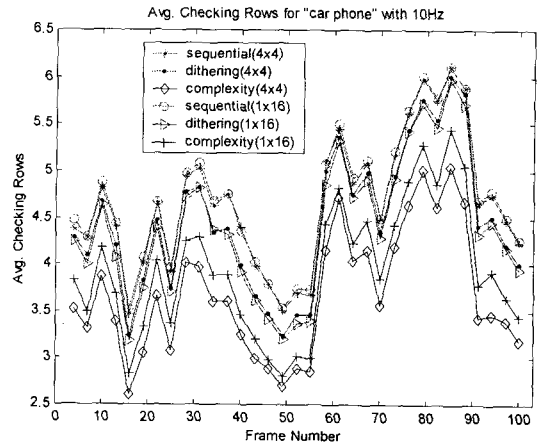


그림 2. 10 Hz의 'car phone' 영상 시퀀스에서 4x4 서브 블록과 1x16행 벡터에 대해 다양한 매칭 스캔의 계산된 평균 행의 수

인할 수 있다. 고속 알고리즘을 전혀 사용하지 않은 원래의 전영역 탐색 알고리즘은 표에서 보는 바와 같이 평균 계산 행의 수가 16이 된다. 위에서도 나선형 탐색을 언급하였는데, 참고 문헌[2,3]에서는 PDE 알고리즘에서 나선형 탐색 방법의 효율성과 중요성

표 3. 30 Hz, 10 Hz 영상 시퀀스에서 모든 매칭 스캔 알고리즘에 대한 평균 PSNR

Sequences Frequency	Foreman	Car phone	Trevor	Claire	Akio	Grand
30	34.4	33.4	33.2	41.2	42.3	42.2
10	31.8	30.8	28.1	35.6	36.8	37.5

을 보여 준다. 4×4 서브 블록과 1×16행 벡터에 대해 계산된 평균 행의 수의 계산 감축 비율을 구해 보면 다음과 같다.

우선 표 1의 'foreman' 영상 시퀀스에서 4×4 서브 블록에 대해 복잡도에 기초한 매칭 스캔 알고리즘은 순차적인 매칭 스캔 알고리즘에 비해 31% ((4.1-2.8)*100/4.1)의 계산 감축을 얻을 수 있다. 그리고 1×16행 벡터에 근거한 방법과도 비교한다면, 4×4 서브 블록에 대해 복잡도에 기초한 매칭 스캔 알고리즘이 9% ((3.0-2.8)*100/3.0)의 계산 감축을 더 얻을 수 있음을 볼 수 있다.

표 3은 전영역 탐색 방법을 이용해서 얻은 예측 영상을 원래의 영상과 비교해서 얻은 PSNR이다. 물론, 이 값은 표 1~2에서의 모든 알고리즘에 대해서 동일한 값을 얻었다. 여기서 시도된 알고리즘들은 원래의 전영역 탐색 방법으로부터 얻은 예측 화질과는 똑 같으면서 단지 불필요한 계산을 어떻게 더 효율적으로 줄일 수 있는가에 대해 접근한 방법이기 때문이다. 본 논문에서 새로 얻은 결과를 정리하면 다음과 같다. 복잡도에 기초한 매칭 스캔 방법의 우수성, 디더링 순서에 기초한 매칭 스캔 방법의 효율성, 1×16행 벡터에 대한 4×4 서브 블록의 우수성 등을 들 수 있다.

5. 결 론

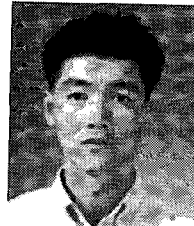
본 논문에서는 매칭 에러와 기준 블록의 기울기 크기 사이의 관계를 이용하여 두 가지 매칭 스캔 알고리즘을 제안 하였다. 후보 블록과 기준 블록 사이의 매칭 에러가 기준 블록의 기울기 크기에 비례 한다는 것을 보임으로써, 기존의 순차적 매칭 방식이 아닌 영상의 복잡도에 기초한 매칭 스캔 알고리즘을 제안하였다. 4×4 서브 블록을 이용해서 복잡도에 근거한 매칭 스캔 알고리즘을 가지고 30 Hz의 'foreman' 영상 시퀀스에 대해 예측 화질의 저하 없이 순차적인 방법에 비해 약 30 %정도 까지 더 많은 계산 감축을 얻을 수 있었다. 영상의 복잡도를 측정할 때

1×16의 행 벡터 형태보다 4×4의 서브 블록으로 나누어서 구하는 것이 더 효율적이라는 것을 실험을 통해 보였으며, 이로부터 약 9%의 계산 감축을 더 얻었을 수 있었다. 그리고 기존의 순차적 매칭 스캔의 방법에 비해 디더링 순서에 기초한 매칭 스캔 방법이 계산 감축을 더 많이 얻을 수 있었다. 따라서 제안하는 알고리즘은 MPEG-2 및 MPEG-4 AVC를 이용한 실시간 비디오 압축 응용 분야에 유용하게 사용 될 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 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.
- [3] J.N. Kim, and 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.
- [4] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, Vol. 4, pp. 105-107, Jan. 1995.
- [5] G.C. de Oliveira and A. Alcaim, "On fast motion compensation algorithms for video coding," *Proc. PCS*, pp. 467-472. 1997.
- [6] 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.

- [7] 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.
- [8] H.S. Wang and R.M. Mersereau, "Fast algorithms for the estimation of motion vectors," *IEEE Trans. Image Processing*, Vol. 8, pp. 435-438, Mar. 1999.
- [9] 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.
- [10] T.M. Oh, Y.R. Kim, W.G. Hong, and S.J. Ko, "A fast full search motion estimation algorithm using the sum of partial norms," *Proc. ICCE*, pp. 236-237, 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. Commun.*, Vol. 45, pp. 527-531, May 1997.
- [13] N. Roma and L. Sousa, "Efficient and configurable full-search block-matching processors," *IEEE Trans. Circuits Syst. for Video Technol.*, Vol. 12, pp. 1160-1167, Dec. 2002.
- [14] 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.
- [15] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," *Proc. SPIE*, Vol. 2419, pp. 100-118, 1995.
- [16] ITU-T Recommendation H.263 software implementation, Digital Video Coding Group at Telenor R&D, 1995.
- [17] C.K. Cheung and L.M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. for Video Technol.*, Vol. 10, pp. 417-422, Apr. 2000.



김 종 남

2001년 8월 광주과학기술원 박사 졸업
 2001년 7월 ~ 2004년 2월 KBS 기술연구소 연구원
 2004년 3월 ~ 현재 부경대학교 전자컴퓨터정보통신공학부 교수

2005년 3월 ~ 현재 ㈜홈캐스트 사외이사
 관심 분야: 멀티미디어 데이터 압축, 멀티미디어 통신 및 신호처리, 콘텐츠 보안, MPEG 2/4/7/21.