

전영역 탐색의 고속 움직임 예측에서 기울기 크기와 부 블록을 이용한 적응 매칭 스캔 알고리즘

김 중 남 (金鍾男), 최 태 선 (崔泰善)
광주과학기술원 기전공학과
전화 : (062) 970-2419 / 팩스 : (062) 970-2384

Adaptive Matching Scan Algorithm Based on Gradient Magnitude and Sub-blocks in Fast Motion Estimation of Full Search

Jong-Nam Kim, Tae-Sun Choi
Dept. of Mechatronics, Kwangju Institute of Science and Technology
E-mail : jnkim@sip1.kjist.ac.kr, tschoi@pia.kjist.ac.kr

Abstract

Due to the significant computation of full search in motion estimation, extensive research in fast motion estimation algorithms has been carried out. However, most of the algorithms have the degradation in predicted images compared with the full search algorithm. To reduce an amount of significant computation while keeping the same prediction quality of the full search, we propose a fast block-matching algorithm based on gradient magnitude of reference block without any degradation of predicted image. By using Taylor series expansion, we show that the block matching errors between reference block and candidate block are proportional to the gradient magnitude of matching block. With the derived result, we propose fast full search algorithm with adaptively determined scan direction in the block matching. Experimentally, our proposed algorithm is very efficient in terms of computational speedup and has the smallest computation among all the conventional full search algorithms. Therefore, our algorithm is useful in VLSI implementation of video encoder requiring real-time application.

1. 서론

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. 전영역 탐색 방식의 계산량을 줄이기 위해 많은 고속 알고리즘들이 연구 되어져 왔다. 이들 고속 알고리즘들은 크게 두 그룹으로 나누어 질 수 있다. 하나는 전영역 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]. 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 그 것의 변형된 알고리즘들 [2]-[5], 고속의 2차원 FIR 필터링 방법을 이용한 고속 알고리즘 [6], 기준 블록과 후보 블록에 대해 수직, 수평, 블록 전체의 투영 (vertical, horizontal and massive projection) 을 이용한 알고리즘 [7], 그리고 부분 매칭 에러값을 이용한 후보 제거 (partial distortion elimination--PDE) 알고리즘 및 그 변형 알고리즘 [8]-[9] 등이 있다.

본 논문에서는 전영역 움직임 예측의 계산량을 줄이기 위한 방법으로서, 가변 매칭 스캔을 이용한 고속 매칭 알고리즘을 제안한다. 이 알고리즘의 속도 향상에서 중요한 사항은 매칭 에러를 계산할 때 불필요한 계산을 가능한 빨리 감지해서 그 불필요한 계산을 감축하는 것이다. 따라서 본 논문에서는 기준 블록의 기울기 크기 정보로부터 가변 매칭 스캔을 적용함으로써 불필요한 계산을 더 많이 감축할 것이다. 먼저, 기준 블록의 기울기 크기와 매칭 에러의 관계를 유도할 것이다. 이 결과로부터 가변 매칭 스캔을 갖는 고속 움직임 예측 알고리즘이 제안된다.

본 논문은 다음과 같이 구성 되어진다. 2 장에서는 기준 블록의 기울기 크기와 매칭 에러의 관계를 유도할 것이며, 이것이 제안하는 알고리즘에 어떻게 사용될 수 있는가를 보일 것이다. 그리고 유도된 결과를 가지고 가변 매칭 스캔을 갖는 고속 움직임 예측 알고리즘을 제안 할 것이다. 3 장에서는 본 알고리즘의 검증을 위한 실험 결과 및 분석이 기술 될 것이다. 마지막 4 장에서 결론으로 마무리를 할 것이다.

2. 제안되는 움직임 추정 기법

움직임 예측기의 VLSI 구현을 위하여, SAD 를 매칭 기준으로 하는 전영역 탐색 방식이 간단한 구현과 우수한 예러 성능으로 인하여 널리 사용되어져 왔다. 하지만 탐색 영역이 크고 화면 크기가 커질 때, 방대한 계산량이 심각한 문제점으로 지적되었다. 이 문제점 때문에, SEA 와 PDE 알고리즘 및 이들의 변형된 방법들이 제안되었다. 이들 변형된 알고리즘들은 시/공간적으로 인접한 움직임 벡터의 사용, Minkowski's inequality, 나선형 탐색방법, 그리고 이들 개념을 합한 방법들이다.

하지만, SAD (Sum of Absolute Difference) 에 기초한 고속 전영역 탐색 알고리즘들에서 두 손실의 고속 매칭에 대한 시도가 아직 발표 되지 않고 있다. PDE 알고리즘에서 중요한 것은 불가능한 후보들을 얼마나 빨리 검출해서 이를 제거 하느냐에 있다. 제안하는 알고리즘은 이렇게 불가능한 후보들을 더 빨리 검출해서 제거하는 효율적인 알고리즘이다. 그렇게 하기 위해, 기준 블록의 기울기 크기 (gradient magnitude) 와 블록 매칭 에러와의 관계를 밝혀서 그것을 이용할 것이다. 먼저 블록 매칭 에러가 기준 블록의 기울기 크기에 비례한다는 것을 보일 것이다. 검증된 이 관계를 이용하여, 기존에 고정된 매칭 스캔 대신에 기준 블록의 기울기 크기에 따른 적응적인 고속 매칭 알고리즘을 제안한다. 그 결과, 불가능한 후보 벡터들을 더 빨리 제거한다. 제안하는 적응적인 고속 매칭 알고리즘은 기준 블록의 기울기 크기에 따라 수행된다. 따라서, 매칭 스캔 방식은 기존의 좌측 상단에서 우측 하단으로 향하는 것이 될 수도 있고, 다른 방향이 될 수도 있다. 이를 적절히 이용함으로써, 기존의 PDE 알고리즘보다 더 빠르며, 동시에 예측화질의 저하가 없는 알고리즘을 얻을 것이다.

먼저, Taylor series expansion 을 이용하여 매칭 에러와 기준 블록의 기울기 크기와의 관계를 보이기로 한다. 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 $mv=(mvx, mvy)$ 이라고 하자. 식 (1) 에서 처럼, 이전 프레임과 현재 프레임 사이의 관계를 기술할 수 있다.

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

$$\begin{aligned} d_{t+1}(p) &= |f_{t+1}(p) - f_t(p+cmv)| \\ &= |f_t(p+mv) - f_t(p+cmv)| \\ &\approx \left| \frac{\partial f_t(p+mv)}{\partial x} (cmvx - mvx) + \frac{\partial f_t(p+mv)}{\partial y} (cmvy - mvy) \right| \\ &\approx \left| \frac{\partial f_{t+1}(p)}{\partial x} (cmvx - mvx) + \frac{\partial f_{t+1}(p)}{\partial y} (cmvy - mvy) \right| \end{aligned} \quad (2)$$

변형된 Taylor series expansion 을 사용하여 매칭 에러와 기준 블록의 기울기 크기간의 관계를 식 (2) 에서 처럼 나타낼 수 있다. 여기서, $cmv=(cmvx, cmvy)$ 는 그 매칭 에러에 해당하는 후보 벡터를 나타낸다. 식 (2) 으로부터, 탐색 영역 안에서 기준 블록과 후보블록의 매칭 에러는 기준 블록의 기울기 크기에 비례한다는 것을 알 수 있다.

이제, 위에서 유도한 결과를 가지고 고속 알고리즘을 위한 가변 매칭 알고리즘을 제안할 것이다. 종래의 PDE 알고리즘은 상단에서 하단으로의 매칭 스캔을 사용한다. PDE 알고리즘에서의 속도 향상은 불필요한 후보 벡터들을 얼마나 빨리 제거 할 수 있는가에 달려 있다. 이를 위해, 주어진 탐색 영역에서 나선형 탐색 방법을 사용한다. 나중에 실험 결과에서도 보듯이, 나선형 탐색 방식을 갖는 PDE 알고리즘이 단순한 PDE 알고리즘보다 더 많은 계산량을 줄임을 알 수 있다. 따라서, 제안하는 알고리즘에서는 나선형 탐색을 함께 사용할 것이다.

계산된 기울기 크기를 가지고, 상단에서 하단으로의 매칭 스캔 대신 새로운 가변 매칭 스캔으로써 블록 매칭 에러를 계산한다. 제안하는 알고리즘의 궁극적인 목적은 불가능한 후보 벡터를 제거함으로써 가능한 빨리 움직임 벡터를 찾는 것이다. 식 (2) 에서 보여지는 것처럼, 블록 매칭 에러는 기준 블록의 기울기 크기에 비례함을 이용하였다. 기울기 크기가 큰 영역부터 매칭 에러를 계산함으로써, 더 많은 계산 감축을 얻을 수 있다.

이제, 계산된 기울기 크기에 기초한 가변 매칭 스캔을 적용한다. 첫번째 제안 하는 알고리즘 (알고리즘 1) 에서는, 그림 1 (a) 에서 보는 것처럼 $16*16$ 화소 크기의 블록을 네 부분으로 나눈다. 그리고 나서 그림 1 (b)-(e) 에서 처럼 각 부 블록에 대해 기울기 크기를 계산한다. 마지막으로 기울기 크기의 순서대로 적당한 매칭 스캔의 방향을 선택하고 블록 매칭을 수행하게 된다. 실험 결과에서도 보여 주듯이, 기존의 상단에서 하단으로의 매칭 스캔 보다 더 많은 후보 점들을 제거 할 수 있음을 알 수 있다.

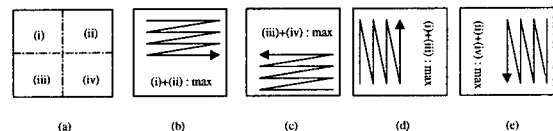


그림 1. 알고리즘 1: 기준 블록의 기울기 크기에 근거한 $16*16$ 블록 크기의 가변 매칭 스캔.

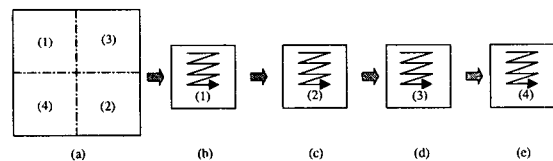


그림 2. 알고리즘 2: 기준 블록의 기울기 크기에 따른 $8*8$ 블록에서 상단에서 하단으로의 가변 매칭 스캔.

불가능한 후보 벡터들을 더 빨리 제거하여 계산을 줄일 수 있는 또 하나의 개념은 그림 2 (a)에서 처럼 $16*16$ 블록 크기대신 네 개의 $8*8$ 블록으로 나누어서 계산하는 것이다. 이렇게 하는 것은 하나의 부 블록의 기울기 크기가 다른 부 블록들의 기울기 크기보다 훨씬 우세할 수 있다는 사실 때문이다. 만약 부 블록의

기울기 크기의 순서가 (1)<(4)<(3)<(2) 처럼 되었을 때, 부 블록 (2) 를 가장 먼저 계산하고, 여기서 계산이 중단되지 않으면 (3), (4), (1)의 순서대로 계산해 나간다. 그림 2 에서는 부 블록의 기울기 크기 순서가 (4)<(3)<(2)<(1) 처럼 되었다고 가정 한 것이다. 이렇게 함으로써, 두 번째 알고리즘인 알고리즘 2 가 알고리즘 1 보다 더 많은 계산 감축을 얻을 수 있다. 그리고 각각의 계산 감축의 결과를 비교하기 위하여 두 번째 알고리즘에서는 가변 매칭 스캔 대신에 고정된 상단에서 하단으로의 매칭 스캔만 사용하였다.

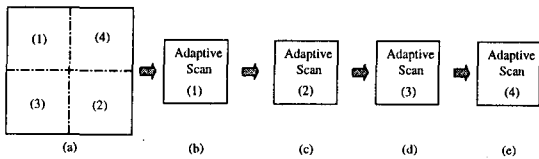


그림 3. 알고리즘 3: 기준 블록의 기울기 크기에 근거한 8*8 블록 크기의 가변 매칭 스캔.

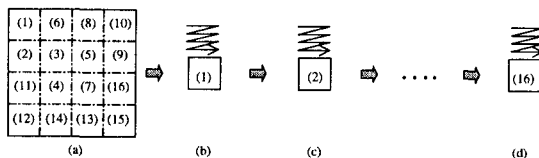


그림 4. 알고리즘 4: 기준 블록의 기울기 크기에 따른 4*4 블록에서 상단에서 하단으로의 가변 매칭 스캔.

더 많은 계산 감축을 위하여 위 두 개념을 혼합하여 사용 할 것이다. 그림 3에서 보여지는 것처럼, 알고리즘 3은 네 개의 부 블록에 대해 각각 가변 매칭 스캔을 적용한다. 이렇게 하기 위해서는 네 개의 부 블록 각각에 대해 다시 네 개의 부 블록으로 나누어서 기울기 크기 계산을 해야 한다. 여기서 주목할 사항이 하나 있다. 기울기 크기를 계산할 때 계산의 중복을 피하기 위하여 가장 작은 4*4 블록부터 기울기 크기 계산을 하여야 한다는 것이다. 열 여섯 개의 부 블록에 대한 기울기 크기 계산이 완료된 후, 네 개씩 영역에

맞게 더해주면 계산의 중복 없이 기울기 크기를 계산할 수 있다. 이것 역시 알고리즘 2 보다 더 많은 계산 감축을 얻을 수 있음을 실험 결과로부터 확인 가능 하였다.

마지막 네 번째 알고리즘은 두 번째 알고리즘과 마찬가지로 더 작은 부 블록에 대해 고정된 매칭 스캔을 사용한다. 알고리즘 3과 유사하게 열 여섯 개의 부 블록에 대한 기울기 크기를 정렬하여 그 순서대로 4*4 블록에 대해 매칭 에러를 계산한다. 이를 더 확장하면, 최종적으로는 화소 하나씩에 대해서 매칭 에러를 계산하여 PDE를 적용할 수는 있다. 하지만 이렇게 하면 너무 많은 비교문이 수행 되어야 하며, 이것으로 인한 시간 지연을 무시할 수 없기 때문에 이를 적절히 조정하는 것이 중요하다. 결론적으로, 나선형 탐색 방식과 가변 매칭 스캔 및 부 블록 매칭을 통하여 다른 알고리즘들 보다 불가능한 후보 벡터를 더 빨리 찾아서 계산을 더 많이 감축 할 수 있음을 실험 결과를 통해서 확인할 수 있다.

3. 모의 실험 결과 및 분석

제안된 알고리즘과 기존의 알고리즘의 성능을 비교하기 위해, "foreman", "car phone", "trevor", "claire", "akio", "grandmother"의 비디오 시퀀스를 가지고 실험을 하였다. 이들 시퀀스에서, "foreman", "car phone"은 다른 시퀀스에 비해 비교적 큰 움직임을 가지고 있다. 한편 "claire", "akio", "grandmother"는 이전 두 시퀀스보다 비교적 작은 움직임을 갖는다. "trevor" 시퀀스는 이들 두 부류의 중간 정도 움직임을 갖는 것이라고 할 수 있다. 제안된 알고리즘은 SEA [2], PDE (Notspiral PDE) [8], PDE (Spiral PDE) [9] 등과 비교 되어진다. 매칭 블록의 크기는 16*16 이며, 탐색 영역의 범위는 ±7 화소로 선택했다. 프레임의 크기는 QCIF (176 by 144) 이다. 실험 결과는 계산된 평균 행의 수와 계산의 감축 비율, 그리고 PSNR 로 나타내었다. 계산된 평균 행의 수는 한 블록에서 계산 되는 평균 수치이다. SEA 의 계산량은 전체 계산량에서 계산된 평균 행의 수로 변환시켜 산출하였다.

표 1. 계산된 평균 행의 수에 대한 실험 결과

Sequences Algorithms	Foreman	Car phone	Trevor	Claire	Akio	Grandmother
SEA	4.1219	4.0346	3.6514	2.6080	1.4248	2.5129
Notspiral PDE	6.9227	7.3701	8.1059	7.8274	6.6575	7.8934
Spiral PDE	4.1264	3.9211	3.1599	3.8764	1.5833	3.0525
Algorithm 1	3.2601	3.2611	2.6840	3.3128	1.4704	3.1063
Algorithm 2	2.8925	3.0550	2.4213	3.0555	1.0688	2.8851
Algorithm 3	2.6633	2.7959	2.2289	2.9326	0.9294	2.6878
Algorithm 4	2.4268	2.5713	2.0477	2.6689	0.7294	2.5427

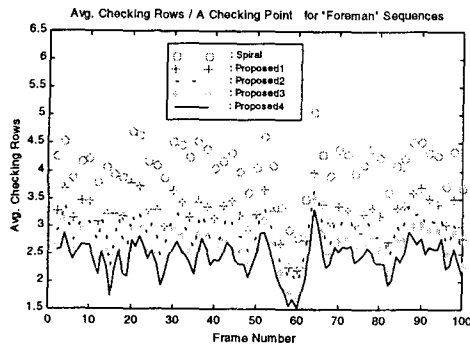


Fig. 5. 'Foreman' 시퀀스에 대한 계산된 평균 행의 수.

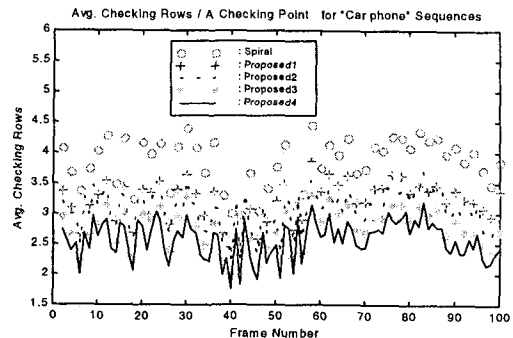


Fig. 6. 'Car phone' 시퀀스에 대한 계산된 평균 행의 수.

표 1에서, 제안된 알고리즘이 기존의 고속 전영역 탐색 알고리즘들 보다 더 많은 계산을 줄인다는 것을 보여 준다. 이들 표에서, 기존의 탐색 방법의 PDE (Notsrial) 이 가장 많은 계산을 요구하며, SEA 와 PDE (Spiral) 는 거의 비슷한 계산량을 필요로 함을 알 수 있다. 기준 블록의 기울기 크기에 따라 가변 매칭 스캔을 갖는 본 알고리즘은 위의 세 알고리즘보다 더 적은 계산을 수행함을 알 수 있다. 그림 5~6 은 각 프레임에 대해 줄여진 계산량을 나타내기 위해 계산되어진 평균 행의 수를 나타낸다. 실험 결과를 통하여, 적응 매칭 스캔 알고리즘이 활동적인 또는 비활동적인 영상 모두에 대해서 우수한 성능을 얻을 수 있다.

4. 결론

본 논문에서는 매칭 에러와 기준 블록의 기울기 크기 사이의 관계를 유도했고, 이를 통하여 후보 블록과 기준 블록 사이의 매칭 에러가 기준 블록의 기울기 크기에 비례 한다는 것을 보였다. 이 사실을 바탕으로 고속 전영역 탐색 방식의 가변 매칭 스캔 알고리즘을 제안 하였다. 본 알고리즘은 기존의 전영역 탐색 알고리즘에 비하여 예측 화질의 저하 없이 불필요한 후보 벡터를 더 빨리 제거하여 더 많은 계산을 감축 할 수 있다. 제안하는 알고리즘은 가변 매칭 스캔과 더 작은 부 블록 매칭으로 구성 되어져 있다. 전자는 네 개의 매칭 스캔을 갖고 있으며, 이는 상단에서 하단으로의 스캔, 하단에서 상단으로의 스캔, 좌측에서 우측으로의 스캔, 우측에서 좌측으로의 스캔이다. 한편, 후자에서는 네 개의 8*8 부 블록과 열여섯 개의 4*4 부 블록을 사용한다. 그리고 이 두 가지 방법을 함께 사용하여, 각 개별 방법을 사용하는 것보다 더 많은 계산을 감축함을 알 수 있다. 얻어진 실험 결과로부터, 제안하는 알고리즘은 원래 처음의 전영역 탐색 방법에 비하여 예측화질의 저하 없이 약 84 ~ 95 %의 계산을 감축함을 볼 수 있다. 따라서 제안하는 알고리즘은 실시간 부호화를 위한 VLSI 구현에 유용하게 사용 될 수 있을 것이다.

참고 문헌

- [1] J.N. Kim and T.S. Choi, "Adaptive matching scan algorithm based on gradient magnitude for fast full search in motion estimation," *IEEE Trans. Consumer Elect.*, vol.45, no.3, pp.762-772, Aug. 1999.
- [2] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, pp. 105-107, Jan. 1995.
- [3] G.C. de Oliveira and A. Alcain, "On fast motion compensation algorithms for video coding," *Proc. PCS*, pp. 467-472. 1997.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," *Proc. SPIE*, vol. 2419, pp. 100-118, 1995.
- [9] ITU-T Recommendation H.263 software implementation, Digital Video Coding Group at Telenor R&D, 1995