

블록 정합오차 예측을 이용한 고속 PDE 알고리즘

준회원 신 세 일*, 정회원 오 정 수*

Fast PDE Algorithm Using Block Matching Error Prediction

Se-ill Sin* Associate Member, Jeong-Su Oh* Regular Member

요 약

본 논문은 고속 PDE 알고리즘을 제안한다. 기존 PDE 알고리즘이 부분 정합오차를 이용해 후보블록의 나머지 정합 과정을 생략할 수 없을 때 제안된 알고리즘은 계산된 부분 정합오차로부터 예측된 블록 정합오차를 이용해 다시 한번 나머지 정합 과정을 생략하는 것을 평가한다. 예측된 블록 정합오차는 부분 정합오차보다 항상 크므로 제안된 알고리즘은 기존 PDE보다 일찍 불가능한 후보 블록을 제거할 수 있다. 모의 실험의 결과들은 제안된 알고리즘이 기존 PDE 정도의 화질을 유지하면서 계산량을 크게 줄여주는 것을 보여준다.

Key Words : Motion Estimation, Motion Compensation, Block Matching, PDE, Prediction

ABSTRACT

This paper proposes the fast partial difference elimination (PDE) algorithm. When the conventional PDE cannot skip the rest of matching procedure in a candidate block using a partial matching error, the proposed algorithm estimates to skip it again using the block matching error predicted from the computed partial matching error. The proposed algorithm can eliminate impossible candidate blocks earlier than the conventional PDE since the predicted block matching error is always bigger than the partial matching error. The simulation results show that the proposed algorithm can significantly reduce the computations while keeping image quality as good as the conventional PDE.

1. 서 론

동영상의 시간적 중복성을 제거하는 움직임 추정(motion estimation) 및 움직임 보상(motion compensation)은 가장 효율적인 동영상 압축 방법 중 하나이다. 움직임 추정을 위한 가장 기본적인 알고리즘인 전역 탐색 알고리즘(full search algorithm)은 블록을 기반으로 이전 프레임의 탐색 영역 내에 존재하는 모든 후보 블록에서 현재 프레임의 정합블록과 가장 유사한 블록을 찾는다. 그러나 전역 탐색법은 최적의 유사 블록을 찾을 수는 있지만 많은 계산량이 문제가 되어 고속 알고리즘에 대한 연구가 활발히 진행되고 있으며 대표적인 고속 전역 탐

색법으로는 PDE(partial difference elimination), SEA(successive elimination algorithm), MSEA(multi-level SEA) 등이 있다¹⁻⁶⁾. 특히 PDE는 블록을 일정 크기로 나누어 부분 정합오차를 계산하고 단계별로 유사성을 비교하여 블록 전체의 정합오차를 계산하기 전에 유사 블록으로 선택이 불가능한 후보 블록을 찾아내어 계산량을 줄이는 매우 유용한 고속 알고리즘이다. 하지만 불가능한 후보 블록인지 판단을 하기 위해서는 일정 수 이상의 부분 정합오차 계산이 반드시 필요하다. 만약 일부 계산된 부분 정합오차를 이용해 블록 정합오차를 예측할 수 있다면 기존 알고리즘보다 빨리 불가능한 후보 블록을 판단할 수 있고 추가적으로 계산량을 줄일 수

* 부경대학교 화상정보공학부 이미지시스템공학과 디지털영상처리 연구실 (ojs@pknu.ac.kr)

논문번호 : KICS2006-10-426, 접수일자 : 2006년 10월 13일, 최종논문접수일자 : 2007년 2월 28일

있을 것이다.

본 논문에서는 예측된 블록 정합오차를 이용하여 기존의 PDE보다 적은 계산량으로 빨리 유사 블록으로 선택이 불가능한 후보 블록들을 판단하는 고속 PDE를 제안한다. 제안된 알고리즘은 PDE 수행 중 부분 정합오차에 의해 불가능한 후보 블록을 판단하지 못한 경우, 계산된 부분 정합오차를 이용하여 후보 블록의 블록 정합오차를 예측하고 이를 근거로 불가능한 후보 블록인지 재 판단하는 알고리즘이다. 예측된 블록 정합오차는 부분 정합오차보다 항상 크므로 기존 PDE보다 일찍 불가능한 후보 블록을 판단할 수 있어 계산량을 감소시킨다. 모의실험의 결과는 제안된 알고리즘이 화질은 유지하면서 계산량을 감소시켜 고속 PDE를 구현할 수 있음을 보여 준다.

II. PDE 알고리즘

PDE는 블록을 라인 또는 서브 블록 형태로 나누고 각 부분에 대한 정합 오차를 단계별로 누적시킨 부분 정합오차를 구하고, 현재까지의 최소 정합 오차와 비교하여 해당 후보 블록이 유사 블록이 불가능한지를 판단한다. 부분 정합오차가 이전 최소 정합오차를 초과할 경우 해당 후보 블록은 유사 블록으로 선택될 수 없으므로 나머지 부분에 대한 정합 과정은 생략한 뒤 다음 후보 블록으로 이동하고, 부분 정합오차가 이전 최소 정합오차보다 작을 경우에는 다음 부분 정합오차를 계산하고 후보 블록의 유사성을 판단하는 과정을 반복한다. 본 논문에서는 블록을 라인 단위로 나누고, 정합오차를 위해 SAD(sum of absolute difference)를 사용하였으며^[7], k번째 부분 정합오차를 Row_SAD_k 로 표기하고 식 (1)과 같이 계산된다.

$$Row_SAD_k = \sum_{i=0}^k \sum_{j=0}^{N-1} |f_i(i, j) - f_{i-1}(i+x, j+y)| \quad (1)$$

$$k = 0, 1, \dots, N-1$$

여기서 N 은 정합 블록의 크기이며 f_i 와 f_{i-1} 은 각각 현재 프레임과 이전 프레임이고 (i, j) 와 (x, y) 는 각각 정합 블록의 화소 위치와 후보 블록의 위치를 나타낸다.

PDE는 매카니즘의 특성상 초기에 최소 정합오차를 찾거나 큰 부분 정합오차를 찾으면 성능이 향상된다. 이러한 특성은 고속 PDE 알고리즘에 사용되

는데, 전자의 특성을 이용한 예로는 최적의 유사 블록일 확률이 큰 위치의 후보 블록부터 나선형으로 이동하며 PDE를 수행하는 spiral PDE가 있고 후자의 특성을 이용한 경우는 영상 복잡도를 기반으로 블록의 라인 또는 서브 블록들을 부분 정합오차가 클 확률이 높은 순서로 정렬한 후 PDE를 수행하는 방법 등이 있다^[8].

III. 제안된 알고리즘

본 장에서는 계산된 부분 정합오차를 이용해 블록 정합오차를 예측하고 이를 이용하여 유사 블록으로 선택이 불가능한 후보 블록을 초기에 판단하는 고속 PDE 알고리즘을 제안한다.

3.1 블록 정합오차의 예측

PDE에서 후보 블록의 유사성을 판단하는데 사용되는 부분 정합오차는 블록에서 정합이 진행된 라인들의 누적 정합오차이고 최종적인 부분 정합오차는 블록 전체 정합오차와 같다. 그러므로 부분 정합오차와 블록 정합오차간에는 상관성이 존재하고, 부분 정합오차를 기반으로 대략적인 블록 정합오차를 예측하는 것이 가능하다. 예측된 블록 정합오차는 항상 부분 정합오차보다 크기 때문에 기존 PDE보다 유사 블록으로 선택이 불가능한 후보 블록을 초기에 판단할 수 있다.

블록 정합오차의 예측에서 가장 중요한 부분은 잘못된 예측이 발생하지 않도록 고려하는 부분이다. 블록 정합오차는 라인 단위 정합오차의 누적인 부분 정합오차의 최종값이므로 식 (2)와 같이 부분 정합오차의 변화율을 이용하여 예측하고 잘못된 예측을 방지하기 위해 가중치 w 를 고려한다.

$$p_SAD_k = Row_SAD_k + \{avg_g_k \times (16-k) \times w\} \quad (2)$$

$$avg_g_k = Row_SAD_k + k$$

여기서 p_SAD 는 예측된 블록 SAD를 의미하고, avg_g_k 는 라인 단위의 SAD들의 평균 증가량을 나타낸다.

잘못된 예측은 큰 라인 SAD에 의해 avg_g_k 가 의도하지 않게 커지면서 발생한다. 또한 큰 라인 SAD는 복잡한 영상에서 발생할 가능성이 높고, 복잡한 블록 영상에서는 SAD가 크게 나타난다. 그래서 본 논문에서는 영상의 복잡도에 따라 w 를 조정하고 있다. 그럼 1은 w 에 따른 예측 결과의 변화를

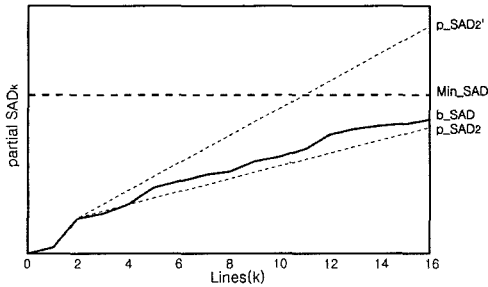


그림 1. w가 적용된 정합오차 예측

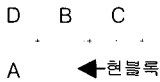


그림 2. 복잡도 결정을 위한 현재 블록과 주변 블록의 위치

나타낸다. 그림에서 Min_SAD 와 b_SAD 는 각각 이전까지의 최소 정합오차와 블록 정합오차를 의미한다. p_SAD_2' 는 w 없이 avg_gk 만 이용한 예측 블록 정합오차이며, p_SAD_2 는 적절한 w 를 사용한 예측 블록 정합오차이다. 2번째 라인의 큰 SAD에 의해 잘못된 예측된 블록 정합오차를 w 를 이용하여 방지할 수 있음을 알 수 있다.

본 논문에서는 영상에 따라 적용적으로 w 를 결정하기 위해 먼저 그림 2와 같이 현재 블록과 주변 블록들의 평균 SAD를 이용해 정합 블록의 영상 복잡도를 판단하는 복잡도(complexity)를 계산한다. 단, 현재 블록의 SAD가 생성되지 않았으므로 (0,0) 위치의 SAD를 먼저 계산하여 초기 min_SAD 와 현재 블록의 SAD로 사용하고, C 블록이 없을 경우에는 D 블록으로 대체한다.

w 는 위에서 계산된 복잡도를 이용하여 그림 3과 같이 0.1과 0.8 사이의 범위 내에서 적용적으로 선택되도록 하였다. 복잡도에 의해 해당 블록의 영상이 복잡하다고 판단되면 큰 라인 SAD가 존재하고 잘못된 예측이 발생할 가능성이 있으므로 작은 가중치를 주고, 블록 영상이 단순하다고 판단되면 큰 라인 SAD의 존재 가능성이 적으므로 큰 가중치를 주어 효율적인 예측을 수행한다[9].

3.2 정합오차 예측을 이용한 고속 PDE 알고리즘

기존의 PDE는 부분 정합오차가 이전 최소 정합오차보다 작으면 불가능한 후보 블록으로 판단할 수 없어 다음 부분 정합오차를 계산하는데, 제안된 알고리즘은 다음 부분 정합오차를 계산하기 전에 계

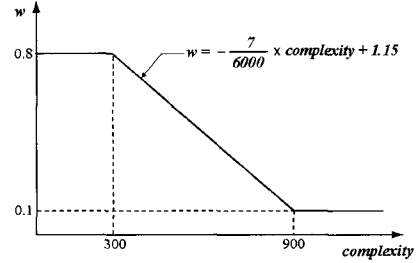


그림 3. 가중치 결정 함수

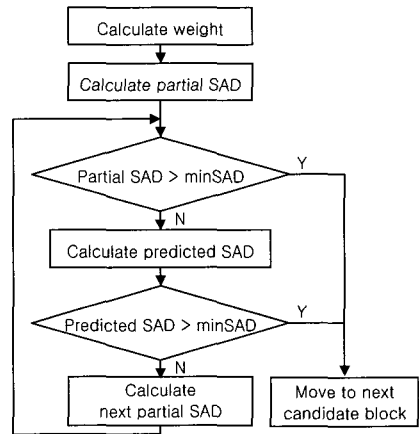


그림 4. 제안된 알고리즘의 블록도

산된 부분 정합오차를 이용해 블록 정합오차를 예측하고 예측된 블록 정합오차를 이용하여 다시 불가능한 후보 블록인지를 판단한다. 유사 블록이 불가능한 후보 블록이면 이후 정합 과정을 생략하고, 가능한 후보 블록은 정합 과정을 계속 수행한다. 제안된 알고리즘에 대한 블록도를 그림 4에 보여주고 있다. PDE는 불가능한 후보 블록을 판단하기 위해 부분 정합오차를 현재 최소 정합오차와 라인 단위로 한번씩 비교하지만 제안된 알고리즘은 추가적으로 예측된 블록 정합오차와 현재 최소 정합오차와 비교하므로 PDE보다 빨리 불가능한 후보 블록을 판단하고 있다.

그림 5는 예측을 통해 고속 PDE를 구현하는 원리를 보여준다. 그림에서 p_SAD_1 , p_SAD_2 , p_SAD_3 은 각각 부분 정합오차 Row_SAD_1 , Row_SAD_2 , Row_SAD_3 을 이용하여 예측된 블록 SAD이다. PDE의 경우 부분 SAD가 Min_SAD 를 초과하는 ②지점에서 불가능한 후보 블록임을 판단하여 나머지 정합 과정을 생략하지만, 제안된 알고리즘은 예측된 블록 정합오차인 p_SAD_3 이 Min_SAD 를 초과하는

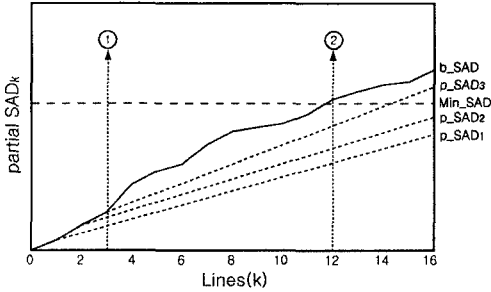


그림 5. 예측을 통한 계산량 감소 원리

①지점에서 불가능한 후보 블록임을 판단하여 나머지 정합 과정을 생략할 수 있어 고속 PDE 구현이 가능한 것을 확인할 수 있다.

IV. 모의 실험 및 결과

모의 실험은 30Hz의 QCIF 해상도 비디오 시퀀스 'foreman', 'stefan', 'akiyo', 'mobile', 'news', 'container', 'silent voice', 'table tennis'의 각 300 프레임을 대상으로 실험하였다. 탐색 영역은 현재 블록에서 ±7, 정합 블록의 크기는 16x16을 사용하였다. 제안된 알고리즘의 성능 평가를 위해 영상 별로 전역 탐색 알고리즘(FSA)과 spiral PDE와 비교한다. 제안된 알고리즘 역시 spiral 탐색을 수행한다.

표 1은 제안된 알고리즘의 화질과 예측의 정확도를 확인하기 위해 PSNR과 움직임 벡터(MV)의 불일치도를 조사하였다. 표 1에서 MV 불일치도는 프레임당 존재하는 99개의 움직임 벡터 중 제안된 알고리즘이 spiral PDE와 다른 움직임 벡터를 검출한 블록 수를 백분율로 나타낸 것이다. 'foreman'이나 'stefan' 같은 복잡한 영상에서 프레임당 1, 2개의 다른 움직임 벡터를 검출하고 있으나, 나머지 영상에 대해서는 2개에서 4개의 프레임당 1개 정도에 불과하다. 결국 MV 불일치도를 통해 알 수 있는 것처럼 높은 예측 정확도는 화질과 연관되어 제안된 알고리즘의 PSNR이 모든 영상에 대해 소수점 셋째 자리 이하의 미미한 화질 저하가 있을 뿐이므로 기존의 spiral PDE의 화질을 유지한다고 할 수 있다.

표 2는 계산량 관점의 결과로서 하나의 정합 블록에 대해 움직임 벡터를 찾는 데 사용되는 평균 블록당 라인 수(#line/block)와 제안하는 알고리즘 수행 시 블록당 평균 예측 수를 조사하였다. spiral

표 1. 제안된 알고리즘의 화질과 정확도 비교

	#line/block			average predictions
	FSA	spiral PDE	P_PDE	
foreman	16.000	4.671	2.938	2.641
stefan	16.000	5.865	4.061	3.779
akiyo	16.000	1.461	1.117	0.217
mobile	16.000	4.047	2.147	1.982
container	16.000	2.758	1.213	0.705
silent voice	16.000	2.406	1.622	0.923
news	16.000	2.244	1.510	0.752
table tennis	16.000	4.075	2.541	2.205
average	16.000	3.441	2.144	1.651

표 2. 제안된 알고리즘의 계산량 비교

	FSA, spiral PDE		P_PDE	
	PSNR	PSNR	difference	disagreed MV
foreman	32.0635	32.0599	-0.0036	1.18%
stefan	23.8256	23.8229	-0.0027	2.94%
akiyo	45.8185	45.8199	0.0013	0.04%
mobile	26.0091	26.0090	-0.0001	0.03%
container	43.0272	43.0225	-0.0046	0.39%
silent voice	35.0025	34.9994	-0.0032	0.19%
news	36.2349	36.2360	0.0011	0.15%
table tennis	31.0417	31.0387	-0.0030	0.46%
average			-0.0018	0.67%

표 3. 부가 계산량

	+연산	-연산	*연산	/연산
w 결정	4	0	1	1
p_SAD 결정	1	1	2	1

PDE와 비교할 때, 제안된 알고리즘이 'akiyo'와 같이 극히 단순한 영상을 제외하고는 모든 영상에 대해 평균 1개 이상의 라인에 대한 계산량을 감소시켰다. 이는 모든 후보 블록에 대해 1개의 라인을 감소 시킨 것이므로 전체적인 계산량에는 더욱 큰 이득이 있다.

단, 제안된 알고리즘은 w는 정합 블록당, p_SAD는 후보 블록당 표 3과 같이 부가 계산량이 요구된다. 하지만 이러한 부가 계산량은 모든 후보 블록에 대해 한 라인의 정합오차 계산량 즉, 15번의 +연산과 16번의 -연산에 비해 상대적으로 무시될 수 있는 계산량이다.

V. 결론

본 논문에서는 기존 PDE의 화질은 거의 유지하면서 블록 정합오차의 예측을 통해 계산량을 감소시키는 고속 PDE 알고리즘을 제안하였다. 제안된 알고리즘은 PDE에서 계산된 부분 정합오차만으로 불가능한 후보 블록을 판단하지 못해 다음 부분 정합

오차를 계산하기 전에 이미 계산된 부분 정합 오차와 블록 영상의 복잡도를 고려하여 블록 정합오차를 예측한다. 예측된 블록 정합오차는 부분 정합오차보다 크므로 기존의 PDE보다 빨리 불가능한 후보 블록을 판단하여 전체적인 계산량을 감소시킨다. 모의 실험을 통하여 제안된 알고리즘이 기존 PDE의 화질을 유지하면서도 많은 계산량을 감소시키는 고속 PDE 알고리즘이라는 것을 보여 준다.

참 고 문 헌

[1] ITU-T Recommendation H.263 software implementation, Digital Video Coding Group at Telenor R&D, 1995.

[2] S. Eckart and C. Feig, "ISO/IEC MPEG-2 software video codec," *Proc. SPIE*, vol. 2419, pp.100-118, 1995.

[3] J. N. Kim and T. S. Choi, "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching scan," *IEEE Trans. On CSVT*, vol. 10, pp.1040-1048, Oct. 2000.

[4] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 8, pp. 435-438, Mar. 1999.

[5] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. on Image Processing*, vol. 9, pp. 501-504, Mar. 2000.

[6] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. On Image Processing*, vol. 4, no. 1, pp.105-107, Jan. 1995.

[7] F. Dufaux and F. Moscheni, "motion Estimation Techniqu for Digital TV : A Review and a New Contribution," *Proceedings of the IEEE Trans. Commun.*, vol. 83, pp. 858-876, Jun. 1995.

[8] B. Montrucchio and D. Quaglia, "New Sorting-Based Lossless Motion Estimation Algorithm and a Partial Distortion Elimination Performance Analysis," *IEEE Trans. on CSVT*, vol. 15, no. 2, pp 210- 220, Feb, 2005.

[9] J. N. Kim, Y. H. Kim and B. H. Ahn, "Fast full search motion estimation algorithm using early detection of impossible candidate vectors," *IEEE Trans. on Signal Process*, vol. 50, no. 9, pp. 2355-2365, Sep. 2002.

신 세 일 (Se-ill Sin)

준회원



2005년 2월 : 부경대학교 화상정보공학부 졸업
 2005년 3월~현재 : 부경대학교 대학원 이미지시스템공학과 석사과정 재학
 <관심분야> 디지털 영상 처리, 신호 처리

오 정 수 (Jeong-Su Oh)

정회원



1990년 2월 : 중앙대학교 전자공학과 졸업
 1992년 8월 : 중앙대학교 대학원 전자공학 석사
 1993년 7월~1997년 1월 : 신도리코 기술연구소
 2001년 8월 : 중앙대학교 첨단영상대학원 영상공학과 박사

2002년~현재 : 부경대학교 화상정보공학부
 <관심분야> 디지털 영상 처리, 적외선 신호처리