

움직임 벡터 추정을 위한 새로운 빠른 알고리즘 (New Fast Algorithm for the Estimation of Motion Vectors)

정 수 목(Soo-Mok Jung)¹⁾

요 약

본 논문에서는 움직임 벡터(motion vector)를 추정하는 새로운 고속 알고리즘을 제안하였다. 제안된 알고리즘은 블록 합 피라미드 알고리즘의 움직임 추정 정확도를 유지하면서 연산량을 감소시키기 위하여 Efficient Multilevel Successive Elimination Algorithm의 diamond mesh기법을 개선한 후, 탐색점에서의 MAD추정기법을 적용하여 블록 합 피라미드 알고리즘에 적용한 것이다. 실험을 통하여 제안된 기법이 블록 합 피라미드 알고리즘의 연산량을 효과적으로 줄일 수 있음을 확인하였다.

Abstract

In this paper, a very fast block matching scheme was proposed to reduce the computations of Block Sum Pyramid Algorithm for motion estimation in video coding. The proposed algorithm is based on Block Sum Pyramid Algorithm and Efficient Multi-level Successive Elimination Algorithm. The proposed algorithm can reduce the computations of motion estimation greatly with 100% motion estimation accuracy. The efficiency of the proposed algorithm was verified by experimental results.

Key words: Block Sum Pyramid Algorithm, Motion vector, Motion estimation, Triangular inequality

1) 정희원 : 삼육대학교 컴퓨터과학과 부교수

I. 서론

H.261, H.263, MPEG등과 같은 동영상은 데이터량이 매우 크기 때문에 동영상을 압축하여 데이터량을 대폭 줄임으로 저장과 전송에 적합하게 한다. 동영상 압축에는 움직임 추정(motion estimation)기법이 사용되는데, 움직임 추정기법에서는 압축하고자 하는 현재 프레임상의 블록과 이전 프레임 상에 있는 블록 중 가장 유사한 블록을 찾아, 구해진 가장 유사한 블록(최적 정합 블록)과의 상대적인 변위인 움직임 벡터(motion vector)를 구한다. 이러한 움직임 추정 기법은 블록 정합 알고리즘(BMA: block matching algorithm)과 화소순환 알고리즘(PRA: pel-recursive algorithm)으로 나누어지는데 구현의 간단함 때문에 블록 정합기법이 H.261, H.263, MPEG등의 비디오 표준으로 채택되어 널리 사용되고 있다.

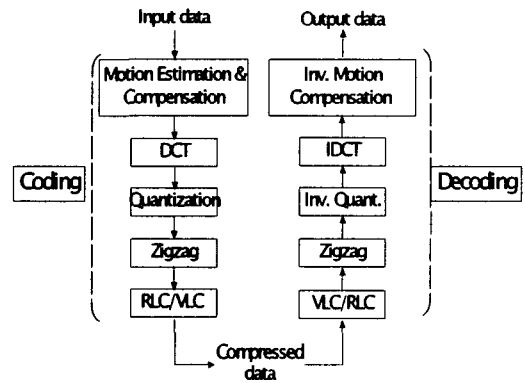
블록 정합 기법에서는 현재 프레임을 일정한 크기의 작은 블록으로 분할한 후 각 블록(template block)에 대하여 이전 프레임 상에 설정된 탐색 윈도우(search window)내의 탐색점들을 좌측 상단점으로 하는 정합 후보 블록들에 대하여 특정한 정합 척도(MAD: mean absolute difference)를 사용하여 현재 블록과의 유사성을 검사하여 최적 정합 블록을 찾아 움직임 벡터를 구한다.

현재 프레임의 각 블록은 블록 정합기법에서 구해진 움직임 벡터와 최적 정합 블록과의 화소차 값들로 변환된 후 [그림 1]에 표시된 동영상 데이터 압축 절차들이 수행된다. 이러한 움직임 추정기법은 동영상의 연속 프레임 상에 존재하는 시간적인 중복성(temporal redundancy)을 제거함으로써 높은 압축율을 얻는데 핵심적인 역할을 하는 기법이다.

움직임 추정 기법 중, 이전 프레임상의 탐색영역내의 모든 정합 후보 블록들을 탐색하는 전역탐색 알고리즘(FSA: full search algorithm)은 최적의 정합블록을 찾을 수 있지만 연산 집중한 블록 정합이 모든 후보 정합블록에 대하여 수행되기 때문에 연산량이 매우 많아 속도가 느려지는 단점이 있다.

FSA의 단점을 줄이기 위하여 움직임 추정을

고속으로 수행하는 여러 가지 움직임 추정기법들이 제안되었는데 이러한 기법들은 다음과 같은 3가지 그룹으로 나뉘어 진다. FSA와 동일한 움직임 추정 정확도를 유지하면서 연산량을 줄이는 방법[1]~[5], 탐색영역 내에 있는 탐색점 중 일부의 탐색점들에 대하여서만 블록 정합을 행함으로써 움직임 정확도는 감소하지만 연산량을 보다 크게 줄이는 방법[6]~[11] 그리고 블록 정합에 사용되는 화소 수를 줄이는 방법[12][13] 등 세 가지의 부류가 있다.



[그림 1] 동영상 압축 및 복원과정

본 논문에서는 FSA와 동일한 움직임 추정 정확도를 유지하면서 연산량을 줄이는 효과적인 방법을 제안한다. FSA와 동일한 움직임 추정 정확도를 유지하면서 연산량을 획기적으로 감소시키는 다단계 연속 제거 알고리즘(SEA: successive elimination algorithm)[1]이 Li와 Salrari에 의해서 제안되었다. 그 후 SEA의 성능을 개선하기 위하여 Lee와 Chen 등이 블록합 피라미드 알고리즘(BSPA: Block Sum Pyramid Algorithm)[2]을 제안하였다. 그 후 다단계 연속 제거 알고리즘[3]과 이웃 탐색점에서의 평균 절대치 오차 및 탐색영역 줄임을 이용한 고속 블록 정합 알고리즘[4]등이 제안되었고, 저자에 의하여 Efficient Multi-level Successive Elimination Algorithm(EMSEA)[5]이 제안되었다. 본 논문에서는 [4]에서 유도된 수식과 [5]에서 제안된 방안을 개선하여 결합한 후, 블록 합

피라미드 알고리즘에 적용하여 블록 합 피라미드 알고리즘과 동일한 움직임 추정 정확도를 가지면서 블록 합 피라미드 알고리즘의 연산량을 효율적으로 감소시키는 고속 움직임 추정 기법을 제안하였다. 본 논문은 구성은 다음과 같다. 본 논문의 기초가 되는 블록 합 피라미드 알고리즘에 대하여 기술하고 제3장에서 제안된 알고리즘을 기술하고 제4장에서 제안된 알고리즘을 동영상압축에 적용한 실험 결과에 대하여 기술하고 제5장에서 결론을 맺는다.

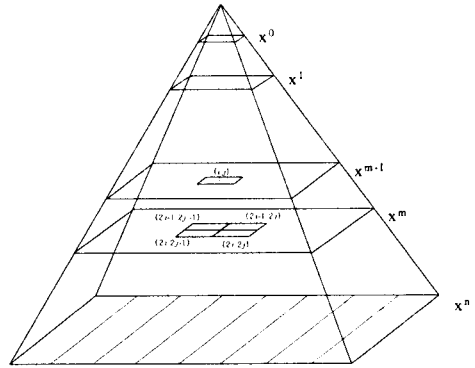


그림 2] 피라미드 데이터 구조

II. Block Sum Pyramid Algorithm(BSPA)

BSPA는 SEA보다 매우 적은 연산량을 필요로 하면서 SEA와 동일한 움직임 추정 정확도를 가진다. BSPA는 최적 정합 블록이 될 수 없는 블록들을 보다 효율적으로 제거하기 위하여 블록의 데이터에 대하여 합 피라미드 데이터 구조(sum pyramid data structure)를 사용한다.

각 블록은 크기가 $N \times N$ 이고 $N=2^n$ 이라고 가정하면, 각 블록 X 에 대한 피라미드는 $\{X^0, \dots, X^{m-1}, X^m, \dots, X^n\}$ 과 같은 블록들의 시퀀스(sequence)로 정의된다. 여기서 X^{m-1} 은 크기가 $2^{m-1} \times 2^{m-1}$ 이고 X^m 으로부터 해상도가 축소된 것(reduced-resolution version)이다. X^0 는 하나의 픽셀을 가지게 된다.

[그림 2]와 같이 피라미드 데이터 구조는 이미지 코딩을 위하여 처음 개발된 계층적인 데이터 구조로서 상위 레벨에 있는 인접한 2×2 크기의 픽셀들을 반복적으로 조작하여 만들어진다. 즉, 레벨 $m-1$ 상의 $X^{m-1}(i,j)$ 는 레벨 m 상의 4개의 점 $X^m(2i-1, 2j-1)$, $X^m(2i-1, 2j)$, $X^m(2i, 2j-1)$, and $X^m(2i, 2j)$ 로부터 만들어진다. 즉 $X^{m-1}(i,j) = f\{X^m(2i-1, 2j-1), X^m(2i-1, 2j), X^m(2i, 2j-1), X^m(2i, 2j)\}$ 이다. 여기서 f 는 연산 함수(operating function)이다.

블록 합 피라미드 구조에서 연산 함수 f 는 합 연산을 하는 함수이다. 즉, 각 픽셀의 값은 식 (1)과 같이 다음 레벨의 2×2 픽셀의 값을 더함으로 구해진다.

$$X^{m-1}(i,j) = X^m(2i-1, 2j-1) + X^m(2i-1, 2j) + X^m(2i, 2j-1) + X^m(2i, 2j) \quad (1)$$

X 와 Y 의 두 개의 블록에 대하여 $MAD^m(X,Y)$ 를 $MAD(X^m, Y^m)$ 으로 두면 식(2)와 같이 표현된다.

$$MAD^m(X,Y) = \sum_{j=1}^{2^m} \sum_{i=1}^{2^m} |X^m(j,h) - Y^m(j,h)| \quad (2)$$

여기서 $X^m(j,h)$ 와 $Y^m(j,h)$ 는 X^m 과 Y^m 에서의 (j,h) 번째의 픽셀의 값을 각각 나타낸다. 그러므로 최상위 레벨에서 $MAD^0(X,Y) \geq |S_X - S_Y|$ 가 성립한다. 위의 식들로부터 식(3)이 성립됨을 Lee 등이 [2]에서 증명하였다.

$$MAD(X,Y) \geq MAD^{n-1}(X,Y) \geq MAD^{n-2}(X,Y) \geq \dots \geq MAD^0(X,Y) \quad (3)$$

BSPA는 이전 프레임상의 탐색점에 대한 각 블록에 대하여 합 피라미드를 구성한다. 템플릿 블록 T 에 대한 정합 블록을 찾기 위하여 T 에 대한 합 피라미드를 구성한다. 그 후 템플릿 블록 T 와 변위 벡터가 $(0,0)$ 인 이전 프레임상의 블록과 MAD 를 계산한다. 이 값을 현재 최소 $MAD(MAD_{min})$ 값으로 한다. 이전 프레임상의 탐색 윈도우 내에 존재하는 다른 블록 X 에 대하여 최상위 레벨에 대한 $MAD^0(T,X)$ 를 구한 후 MAD_{min} 과 비교한다. $MAD^0(T,X)$ 가

MAD_{min}보다 크다면 블록 X는 제거되고 그렇지 않으면 1번째 레벨에서 구해진 MAD[MAD¹(T,X)]와 비교되어진다. 만약 MAD¹(T,X)이 MAD_{min} 보다 크다면 블록 X는 제거될 수 있다. 이러한 과정이 최상위 레벨에 도달 할 때까지 반복된다. 만약 최상위 레벨에 도달하였다면 MAD(T,X)는 계산되어지고 확인 되어진다. 만약 MAD(T,X)가 MAD_{min} 보다 적다면 현재의 최소 왜곡 MAD_{min}은 MAD(T,X)로 교체된다.

BSPA는 “coarse to fine” 기법이다. 이것은 많은 탐색 블록들을 그들의 MAD들을 계산하지 않고 제거할 수 있는 알고리즘이다. 두 블록간의 MAD를 계산하는 것은 피라미드의 최상위 레벨에서의 MAD[MAD⁰(T,X)]를 계산하는 것보다 연산량이 훨씬 크기 때문에 시간이 많이 걸린다. 따라서 BSPA는 블록들의 MAD들을 계산하기 전에 블록들을 제거할 수 있으므로 탐색 시간을 많이 절약 할 수 있는 알고리즘이다.

III. 움직임 벡터 추정을 위한 새로운 빠른 알고리즘

블록 합 피라미드 알고리즘의 연산량을 감소시키기 위하여 [그림3]과 같은 다이아몬드 그물 패턴을 적용하여 흰색 원으로 표시된 탐색점에서는 BSPA를 수행하고, 검은색 원으로 표시된 탐색점에서는 [4]에서 인근 탐색점에서의 MAD를 이용하여 추정된 최소 MAD인 식(4)를 이용하여 계산 집중적인 연산을 포함하는 블록 정합을 수행하지 않고도 탐색점을 제거할 수 있어 움직임 추정에 소요되는 연산량을 효과적으로 감소시킬 수 있다

$$Esti_MAD_{min}(x+1, y) = | \|D'(x,y)\| - \|C'\| | \quad (4)$$

여기서 D'(x,y)는 템플릿 블록의 첫 번째 열과 정합 후보 블록의 첫 번째 열의 차인 오차 블록의 첫 번째 열을 템플릿 블록의 N 번째 열과 정합 후보 블록의 첫 번째 열 간의 차인 열로 대체 한 새로운 오차블록이다. C'는 템플릿 블록 내, 화소들의 이웃 화소간의 수평방향으로

의 화소값의 차를 나타낸다.

즉, 식 (4)를 바탕으로 상하좌우 인근 탐색점에서의 MAD를 사용하여 검은색 원으로 표시된 탐색점에서의 MAD를 추정 한 후, 계산집중 연산인 블록 정합을 수행할지 여부를 조사한다. [그림 3]에서 흰색 원으로 표시된 탐색점은 탐색 윈도우 내의 전체 탐색점 중 절반에 해당하고 나머지 절반이 검은색 원으로 표시된 탐색점들이다. 흰색 원으로 표시된 탐색점에서는 BSPA를 수행함으로써 연산량 감소는 발생하지 않는다. 제안된 고속 블록 정합 기법에서는 탐색 윈도우 내에 있는 탐색점 중 절반에 해당하는 검은색 원으로 표시된 탐색점에서의 블록 정합 수행 여부를 판단하여 블록 정합이 불필요할 경우 계산 집중 연산인 블록 정합을 수행하지 않고 탐색점을 제거할 수 있음으로 움직임 추정에 필요한 연산량을 매우 효과적으로 감소시킬 수 있다. [4]에서 유도된 부등식은 삼각 부등식의 성질을 이용하여 유도된 식임으로 움직임 추정 정확도는 100%로 유지된다.

BSPA에 기초한 고속 블록 정합 기법의 절차는 아래와 같다.

1. select initial candidate matching block within the search window in the previous frame.
2. calculate MAD at the selected search point, calculated MAD becomes the current minimum MAD(*curr_MAD_{min}*)
3. select another search point among the rest of the search points
4. if(selected search point is white circled search point?) goto 7.0
5. estimate the minimum MAD at the search point using the MADs of 4 neighbor black circled search points
6. if(estimated minimum MAD(*Esti_MAD_{min}*) is greater than *curr_MAD_{min}*) goto 9
- 7.0 calculate the MAD⁰(T,X) at the selected search point
- if(*curr_MAD_{min}* ≤ MAD⁰(T,X)) goto 9
- 7.1 calculate the MAD¹(T,X) at the selected

search point
 if($curr_MAD_{min} \leq MAD^1(T,X)$) goto 9

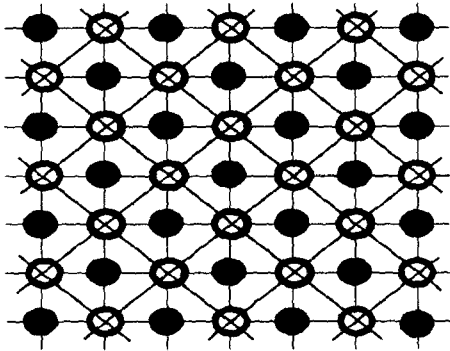
7.n calculate the $MAD(T,X)$ at the selected search point

if($curr_MAD_{min} \leq MAD(T,X)$) goto 9

8 $curr_MAD_{min}=MAD(T,X)$

9. if(all the search points in the search window are not tested?) goto 3

10. minimum $MAD=curr_MAD_{min}$, calculate motion vector. calculated motion vector becomes the optimum motion vector for the temporary block



[그림 3] 다이아몬드 그물 패턴

IV. 실험결과

본 연구의 실험에 사용한 영상은 176x144 pixel 크기를 가지는 miss_america.qcif, grandma.qcif, salesman.qcif의 100 프레임(frame)에 대하여 실험을 수행하였다. BSPA의 실험환경과 같이 블록의 크기는 16x16 pixel($N=16$)로 하였고 탐색윈도우의 크기는 31x31 pixel($M=15$)이며 정수 움직임 벡터 값만을 고려하였다.

본 논문에서 제안한 고속 블록 정합 기법에 대한 실험 결과는 <표 1>과 같다. 제안된 고속 블록 정합 기법은 탐색 윈도우 내의 탐색점 중 절반에 대하여는 BSPA를 수행하고 나머지 탐색점에서는 상하좌우 인근 탐색점에서의 MAD

값을 사용하여 계산 집중적인 연산인 블록 정합의 수행 여부를 결정함으로써 연산량이 매우 효과적으로 줄어든다.

miss_america, grandma, salesman 비디오 시퀀스에 대하여 BSPA의 연산량이 각각 6.5%, 5.3% 6.2% 감소하였다.

<표 1> 개선된 블록 합 피라미드 알고리즘의 움직임 추정 연산량

Algo- rith- m	비디오 시퀀스	평균정합 평가회수 / frame	평균ro- w/ 정합 평가	오버헤드 (in rows)	전체 연산량	연산량 감소율 (%)
BSPA	miss_am.	369.7	14.89	28.281.3	33,786.1	
	grandma	351.5	14.12	29,768.5	34,731.7	
	salesman	242.3	14.39	29,510.6	32,997.3	
제안 기법	miss_am.	287.2	14.15	27538.1	31602.0	6.5%
	grandma	279.3	13.79	29052.4	32903.9	5.3%
	salesman	183.5	14.10	28,364.2	30,951.6	6.2%

V. 결론

본 논문에서는 효율적인 움직임 추정기법으로 널리 알려진 블록 합 피라미드 알고리즘의 연산량을 줄이기 위하여, 고속 블록 정합 기법을 제안하였다. 제안된 알고리즘은 계산 집중적인 블록 정합 과정을 탐색윈도우 내에 있는 모든 탐색점에서 수행하지 않고 다이아몬드 그물 패턴의 흰색 원으로 표시된 탐색점에서만 수행한다. 그 이외의 탐색점에 대한 블록 정합 수행 여부는 상하좌우의 인근 탐색점에서의 MAD를 이용하여 판단한다. 인근 탐색점에서의 MAD값을 이용하여 검은색 원으로 표시된 현재 탐색점에서의 MAD값을 예측하여 현재 탐색점에서의 MAD값이 현재까지의 최소 MAD보다 크면 계산 집중적인 연산인 블록 정합을 수행하지 않고도 탐색점을 제거할 수 있어 연산량을 효과적으로 줄일 수 있다. 이 때 움직임 추정의 정확도는 100% 정확하게 유지된다. qcif format을 갖는 miss_america, grandma, salesman 비디오 시퀀스에 대한 실험을 통하여 본 논문에서 제안된 블록 정합 기법은 블록 합 피라미드 알고리즘의 연산량을 최대 6.5% 줄였다.

참 고 문 헌

- [1] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. Image Processing, vol. 4, No.1, pp. 105-107, Jan. 1995.
- [2] C. H. Lee, and L. H. Chen, "A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid," IEEE Trans. Image Processing, Vol. 6, No.11, pp. 1587-1591, Nov. 1997.
- [3] X. Q. Gao, C.J. Duanmu, and C.R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," IEEE Trans. Image Processing, Mar. 2000, Vol. 9, No.3, pp.501-504
- [4] W.S. Cheong, B.K. Lee etc., "A fast Block Matching Algorithm Using Mean Absolute Error of Neighbor Search Point and Search Region Reduction," The Journal of Korean Institute Communication Science, Vol. 25, No.1B, Jan. 2000.
- [5] S. M. Jung, S. C. Shin, H. Baik and M. S. Park, "Efficient multilvel successive elimination algorithms for block matching motion estimation," IEE Proc.-Vis. Image Signal Process., Vol. 149, No.2, April 2002.
- [6] J. R. Jain, and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., Dec. 1981, vol. COMM-29, pp. 1799-1808
- [7] T. Koga, K. Iinuma, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proc. Nat. Telecommunications Conf., Nov. 1981. pp. G5.3.1- G.5.3.5
- [8] A. Puri, H.-M. Hang, and D. L. Schilling, "An efficient block matching algorithm for motion compensated coding," Proc. Int. Conf. Acoust., Speech, Signal Processing, 1987, pp. 25.4.1-25.4.4
- [9] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., July 1990, vol. 38, no. 7, pp. 950-953
- [10] M. J. Chen, L. G. Chen and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," IEEE Trans. Circuits Syst. Video Technol., Oct. 1994, vol. 4, pp. 504-509
- [11] J. Y. Tham, S. Ranganath, M. Ranganath, and A. Ali Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Aug. 1998, vol. 8, no. 4, pp. 369-377
- [12] Y. L. Chan and W. C. Siu, "New adaptive pixel decimation for block motion vector estimation," IEEE Trans. Circuit and Systems for Video Technology, vol. 6, no. 1, pp. 113-118, Feb. 1996
- [13] B. Liu, and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Syst. Video Technol., Apr. 1993, vol. 3, no. 2, pp. 148-157

정 수 목 (Soo-Mok Jung)



1984. 2. 경북대학교
전자공학과졸업(공학사)
1986. 2. 경북대학교 대학원
전자공학과 졸업(공학석사)
1986. 1. ~ 1991. 2.
LG정보통신 연구소 연구원

2002. 2. 고려대학교 대학원 컴퓨터 학과
졸업(공학박사)
현재 삼육대학교 컴퓨터 과학과 부교수
관심분야 : 멀티미디어, 영상처리,
컴퓨터아키텍처