움직임 방향 지향적인 고속 블록정합 알고리즘

오정수*

Motion Direction Oriented Fast Block Matching Algorithm

Jeong-su Oh*

요 약

블록정합에서 방대한 계산을 줄이기 위해, 본 논문은 탐색영역에서 탐색점을 제한하는 고속 블록정합 알고리즘을 제안한다. 대부분의 움직임 벡터가 탐색영역 중심부에 위치하고, 정합오차가 최적의 유사블록을 향해 단조감소한다는 사실에 근거하여 제안된 알고리즘은 단계 사이에 정합패턴을 1 화소 단위로 이동하고, 이전 단계들에서 결정된 유사블록들로부터 최적의 유사블록을 향한 움직임을 예측하고, 탐색점들의 움직임을 움직임 방향에 대해 ±45°로 제한한다. 그 결과 불필요한 탐색점을 제거할 수 있었고 블록정합 계산을 줄일 수 있었다. 기존 유사 고속 알고리즘과 비교하여 제안된 알고리즘은 큰 움직임을 갖는 영상에서 미미한 화질 저하를 발생시키지만 보통 움직임을 갖는 영상에서 동등한 화질을 유지하고, 반면에 그들의 블록정합 계산을 적게는 20% 많게는 67%를 줄여 주었다.

ABSTRACT

To reduce huge computation in the block matching, this paper proposes a fast block matching algorithm which limits search points in the search area. On the basis of two facts that most motion vectors are located in central part of search area and matching error is monotonic decreasing toward the best similar block, the proposed algorithm moves a matching pattern between steps by the one pixel, predicts the motion direction for the best similar block from similar blocks decided in previous steps, and limits movements of search points to ±45° on it. As a result, it could remove the needless search points and reduce the block matching computation. In comparison with the conventional similar algorithms, the proposed algorithm caused the trivial image degradation in images with fast motion but kept the equivalent image quality in images with normal motion, and it, meanwhile, reduced from about 20% to over 67% of the their block matching computation.

키워드

블록 정합, 움직임 추정, 움직임 벡터, 움직임 방향

Key word

block matching, motion estimation, motion vector, motion direction

* 종신회원 : 부경대학교 이미지시스템공학과(ojs@pknu.ac.kr)

접수일자 : 2011. 04. 18
심사완료일자 : 2011. 06. 07

Ⅰ. 서 론

오늘날 정보전달 매체로 활발히 사용되고 있는 동영상은 방대한 정보량으로 인해 저장하거나 전송하기 위해서는 압축이 필수적이다. 동영상 압축을 위해 다양한 기법들이 개발되고 있고, 이들의 핵심은 영상의 시간적 중복성을 제거하는 움직임 추정/보상(motion estimation/compensation)이다. 블록정합 (block matching)은 움직임 추정을 위한 대표적인 알고리즘으로 현 프레임을 일정 크기의 정합블록(matching block)으로 나누고, 정합블록을 이전 프레임의 탐색영역 내 탐색점 블록들과 비교하여 가장 유사한 블록을 찾는 것이다. 특히 최적의 유사블록을 찾기 위해 탐색영역 내 모든 탐색점 블록들과 비교하는 알고리즘을 전역 탐색(full search:FS) 알고리즘이라 한다. 이는 가장 이상적인 블록정합 알고리즘이나 방대한 계산이 요구되어 시스템 구현에 어려움을 준다. 그래서 정합 계산을 줄이기 위한 다양한 고속 알고리즘들에 대해 연구되고 있다. 고속 알고리즘들은 다양한 탐색패턴을 이용해 제한된 탐색점들만을 탐색하는 알고리즘과 모든 탐색점을 탐색하나 불필요한 정합 계산을 생략하는 알고리즘으로 나눌 수 있다. 3 단계 탐색(three step search: TSS)[1], 4 단계 탐색(four step search:4SS)[2], 다이아몬드 탐색(diamond search:DS)[3], 육각형 탐색(hexagonal search:HEXA)[4] 등이 전자의 대표적인 예이고 FS와 비교하여 화질 저하는 있지만 계산량을 크게 줄일 수 있다. PDE (partial difference elimination)[5], SEA (successive elimination algorithm)[6], MSEA (multi-level SEA)[7] 등이 후자의 대표적인 예이고 전자보다 계산량의 감소는 적지만 화질 저하는 없다.

본 논문은 블록정합에서 최적의 유사블록이 탐색영역 중심인 (0,0)에 압도적으로 많이 위치하고, 최적의 유사블록에 접근할수록 유사도 평가를 위한 정합오차가 단조 감소한다는 사실에 근거해 움직임 방향 지향적인 고속 블록정합 알고리즘을 제안한다. 제안된 알고리즘은 초기에 (0,0) 주변 블록의 유사도를 평가하고, 움직임의 방향을 예측하여 제한된 탐색점의 후보블록들에 대해서만 블록정합을 수행한다. 초기 움직임 방향 예측은 4 근방 예측을 수행하고 나서 부분적인 8 근방 예측을 수행하고, 이후 움직임 방향 예측은 움직임 진행 방향에 대해 ±45°로 제한하고 있다. 또한 적은 움직임이 지배적이므로 단계별 정합패턴을 1 화소 단위로 움직인다. 그 결과 탐색점의 수를 크게 줄일 수 있었다. 제안된 알고리즘의 성능 평가를 위해 유사 고속 알고리즘들인 4SS, DS, HEXA와 비교되고, 모의실험 결과는 영상과 알고리즘에 따라 다소 차이가 있지만 화질을 거의 유지하면서 계산량을 19.80%에서 67.46%를 감소시켜 주고 있다.

Ⅱ. 기존 블록정합 알고리즘

본 장에서는 블록정합을 이해할 수 있는 전역 탐색 알고리즘과 기존 유사 고속 블록정합 알고리즘인 4SS, DS, HEXA를 기술한다.

2.1 전역 탐색 알고리즘

블록정합은 일정 크기의 정합블록들로 나누어진 현재 프레임의 각 정합블록에 대해 그림 1과 같이 직전 프레임의 탐색영역 내 후보블록들과 비교하여 가장 유사한 블록을 찾는 것이다. 이때 최적의 유사블록 위치를 움직임 벡터라 하고, 블록간의 유사성은 SAD(sum of absolute difference) 혹은 MSE (mean square error)를 이용하여 평가되고, 본 논문에서는 식 (1)의 SAD를 이용한다.

$$SAD(x,y)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|f_t(i,j)-f_{t-1}(i+x,j+y)\right| \qquad (1)$$

여기서 $f_t(\cdot)$와 $f_{t-1}(\cdot)$는 각각 현재와 직전 프레임이고, N은 정합블록의 크기이고, (i,j)와 (x,y)는 각각 정합블록과 탐색점의 위치이다. 이때 블록정합을 위해 M^2의 탐색점에서 SAD가 계산되므로 M^2N^2의 뺄셈과 $M^2(N^2-1)$의 덧셈이 요구된다.

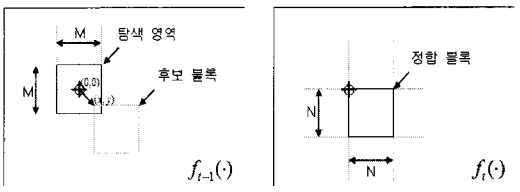


그림 1. 정합블록
Fig. 1 Block matching

2.2 기존 고속 알고리즘

전역 탐색 알고리즘의 방대한 계산을 줄이기 위한 한 부류가 탐색영역의 일부 탐색점에서만 정합을 수행하는 알고리즘으로 4SS, DS, HEXA가 대표적인 예이다.

4SS는 그림 2(a)의 사각형 정합패턴을 이용해 다음과 같이 수행된다.

1 단계 : 원점 중심의 정합패턴에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 원점이면 3 단계를 수행하고, 그렇지 않으면 2 단계를 수행한다.

2 단계 : 유사블록 중심의 정합패턴에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 그대로면 3 단계를 수행하고, 그렇지 않으면 2 단계를 반복 수행한다.

3 단계 : 현 정합패턴의 내부 8 근방 탐색점에서 블록정합을 수행하여 최적의 유사블록을 결정한다.

DS는 그림 2(b)의 다이아몬드형 정합패턴을 이용해 다음과 같이 수행된다.

1 단계 : 원점 중심의 정합패턴에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 원점이면 3 단계를 수행하고, 그렇지 않으면 2 단계를 수행한다.

2 단계 : 유사블록 중심의 정합패턴에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 그대로면 3 단계를 수행하고, 그렇지 않으면 2 단계를 반복 수행한다.

3 단계 : 현 정합패턴의 내부 4 근방 탐색점에서 블록정합을 수행하여 최적의 유사블록을 결정한다.

HEXA는 그림 2(c)의 육각형 정합패턴으로 이용해 다음과 같이 수행된다.

1 단계 : 원점 중심의 정합패턴에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 원점이면 3 단계를 수행하고, 그렇지 않으면 2 단계를 반복 수행한다.

2 단계 : 유사블록 중심의 정합패턴에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 변화가 없으면 3 단계를 수행하고, 그렇지 않으면 2 단계를 수행한다.

3 단계 : 현 정합패턴의 내부 4 근방 탐색점에서 블록정합을 수행하여 최적의 유사블록을 결정한다.

상기 알고리즘들의 공통적인 처리는 최적의 유사블록이 결정되면 블록정합이 종료되고, 이미 수행된 탐색점의 블록정합은 생략된다. 한편 이들은 최적의 유사블록들이 탐색영역 중심인 (0,0)에 집중되어 있다는 사실에 근거하여 (0,0)에 탐색점을 집중하면서 외곽으로는 탐색점을 제한하고, 외곽으로 이동 시 2 화소 단위로 움직이다 최적의 유사블록을 결정시 1 화소 주변을 탐색한다. 전자는 국부 최소(local minimum)에 빠지는 원인으로 모든 고속 알고리즘이 갖는 문제이고, 후자는 큰 움직임에 대해서 빠른 접근을 유도하나 중간이하 움직임에 대해서는 불필요한 계산이 추가된다.

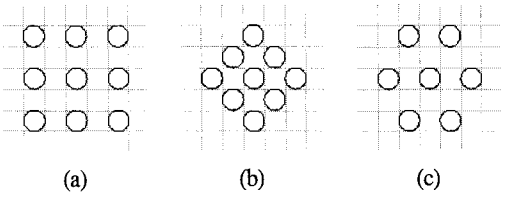


그림 2. 정합패턴 (a) 4SS (b) DS (c) HEXA
Fig. 2 Matching pattern (a) 4SS (b) DS (c) HEXA

III. 제안된 블록정합 알고리즘

본 장에서는 블록정합이 갖는 특성을 분석하여 기술하고, 이에 근거한 움직임 방향 지향적인 고속 블록정합 알고리즘을 제안한다.

표 1. 최대 움직임 벡터의 분포
Table 1. Distribution of maxmum motion vector

max(MV_X,MV_Y)	0	1	2	3	4	5	6	7
%	55.02	27.95	7.25	3.46	2.50	1.73	1.10	0.98

3.1 블록정합 특성

4SS, DS, HEXA처럼 탐색점을 제한하는 고속 블록정합 알고리즘들은 두 사실에 근거한다. 첫번째는 그림 3과 표 1에서 볼 수 있듯이 움직임 벡터가 (0,0)에 집중되어 있다는 것이다. 그림 3과 표 1은 실험에 사용될 7 개의 QCIF 포맷 연속영상들에서 통계된 움직임 벡터 (MV_X, MV_Y)의 분포와 x, y 방향의 최대 움직임 벡터의 분포이다. 두 자료를 통해 움직임 벡터가 (0,0)인 정합블록이

약 55%이고, 움직임 벡터가 1이하인 정합블록이 약 83%이고, 움직임 벡터가 2이하인 정합블록이 약 90%를 차지하고 있다. 두 번째는 그림 4에서 볼 수 있듯이 탐색점이 최적의 유사블록에 멀어질수록 정합오차가 단조 증가한다는 것이다. 그림 4는 (1,0)를 움직임 벡터로 갖는 정합블록의 탐색영역에서 정합오차를 보여주고 있다. 탐색점 (1,0)에서 최소 정합오차를 갖고 외곽을 향해 전방향으로 정합오차가 단조 증가하고 있다. 그러나 이는 절대적이지 않아 일부 정합블록에서 국부 최소에 빠져 잘못된 유사블록을 찾기도 한다. 잘못된 최적의 유사블록은 영상에 따라 움직임이 큰 영상에서 최대 10%가, 움직임이 작은 영상에서 최소 1%가, 평균적으로 5% 정도가 발생하고 있고, 이는 탐색점을 제한하는 고속 블록정합 알고리즘들에서 피할 수 없는 문제이다.

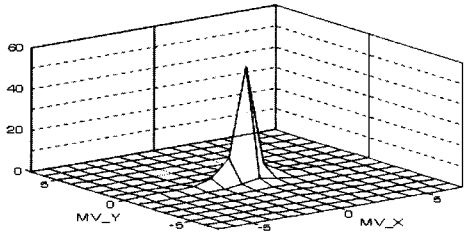


그림 3 탐색영역에서 움직임 벡터의 분포
Fig. 3 Distribution of motion vector in the search range

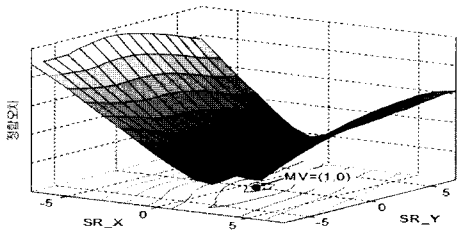


그림 4 탐색점에 따른 정압오차
Fig. 4 Matching error in the search point

3.2 제안된 고속 블록정합 알고리즘

기존 알고리즘들처럼 본 논문 역시 최적의 유사블록이 탐색영역의 중심에 집중되어 있고, 정합오차가 최적의 유사블록을 향해 단조 감소한다는 사실에 근거한다. 이들을 기존 알고리즘들과 달리 해석하면 전자는 작은 움직임이 지배적임이므로 단계사이 정합패턴 이동이 2화소보다 1화소가 더 효율적임을 의미하고, 후자는 정합의 진행 방향 즉 최적의 유사블록을 향한 움직임 방향이 후진되지 않음을 의미한다. 따라서 제안된 알고리즘은 최적의 유사블록을 향한 움직임을 예측하여 정합진행 방향을 ±45°로 제한하고, 정합패턴을 1 화소 단위로 이동하도록 하여 불필요한 탐색점을 줄이고 있다. 제안된 알고리즘은 다음과 같이 수행된다.

1 단계 : 그림 5(a)와 같이 탐색영역의 중심과 1 화소 거리 4 근방 탐색점에서 블록정합을 수행하여 유사블록을 결정한다. 유사블록이 탐색영역의 중심이면 그를 최적의 유사블록으로 결정하고, 그렇지 않으면 2 단계를 수행한다.

2단계 : 유사블록에 따라 그림 5(b)와 같이 유사블록의 양 옆의 탐색점에서 블록정합을 수행하여 유사블록과 움직임 방향을 결정한다.

3단계 : 유사블록에 따라 그림 5(c)와 같이 정합진행 방향을 ±45°로 제한되어 지정된 유사블록 주변 탐색점들에서 블록정합을 수행한다. 유사블록이 변화가 없으면 그를 최적의 유사블록으로 결정하고, 그렇지 않으면 3 단계를 수행한다.

2와 3 단계에서 탐색점은 단계사이 유사블록의 변화에 의해 최적의 유사블록을 향한 움직임 방향을 결정하고 그 방향에 의존하여 결정된다.

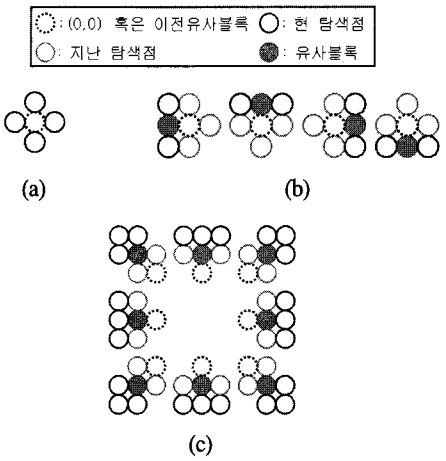


그림 5. 제안된 알고리즘의 탐색점
(a) 1 단계 (b) 2 단계 (c) 3 단계
Fig. 5 Search points in the proposed algorithm
(a) step 1 (b) step 2 (c) step 3

Ⅳ. 모의 실험 결과

제안된 알고리즘의 성능 평가를 위해 FS를 포함한 유사 고속 알고리즘인 4SS, DS, HEXA들과 비교하고 있다. 모의 실험영상은 144x176인 QCIF 해상도에 30Hz 100 프레임의 연속 영상인 'carphone', 'foreman', 'stefan', 'table', 'mobile', 'mother& daughter', 'hall', 'coastguard'가 사용되었다. 이때 정합블록은 16x16 크기이고, 탐색영역은 정합블록을 중심으로 ±7이다.

표 2는 100 프레임의 실험 영상들에서 알고리즘에 따른 화질과 계산량을 비교하고 있다. 화질은 100 프레임의 평균 PSNR이고, 계산량은 100 프레임의 평균 탐색점을 FS의 평균 탐색점에 대한 백분율로 나타내었다. 화질 측면에서 FS와 비교하면 움직임 작은 'mobile', 'mother& daughter', 'hall' 영상들과 움직임이 다소 있는 'carphone', 'foreman', 'coastguard' 영상들에서는 각각 0.1dB와 0.2dB 이하의 미세한 화질 저하가 발생하고, 움직임 큰 'stefan', 'table' 영상에서는 0.8dB 정도의 화질 저하가 발생하고 있다. 그러나 기존 고속 알고리즘들과 비교하면 움직임이 큰 영상들에서 0.2dB 정도 화질 저하가 발생뿐 나머지 영상에서는 ±0.1dB 정도로 동등하다. 하지만 계산량 측면에서는 크게 개선되고 있다. FS에 비해 최소 2.59%, 최대 5.07%, 평균 3.73% 정도만 사용되었고, 기존 고속 알고리즘과 비교해 최소 19.80%, 최대 67.46%, 평균 45.78% 정도를 줄여주고 있다.

표 2. 성능 비교
Table 2. Performance comparison

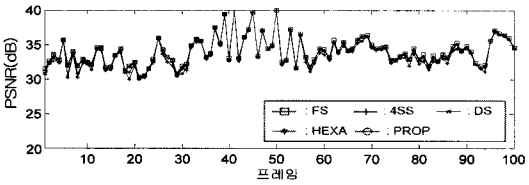
	carphone		foreman		stefan		table	
	화질	계산량	화질	계산량	화질	계산량	화질	계산량
FS	34.01	100.00	32.55	100.00	25.34	100.00	27.29	100.00
4SS	33.82	8.44	32.33	8.58	24.85	9.02	26.73	8.92
DS	33.93	7.04	32.34	7.18	24.84	8.02	26.90	7.51
HEXA	33.60	5.60	31.92	5.71	24.76	6.32	26.42	6.07
PROP	33.85	3.81	32.25	4.00	24.52	5.07	26.50	4.40
	mobile		mother&daughter		hall		coastguard	
	화질	계산량	화질	계산량	화질	계산량	화질	계산량
FS	26.11	100.00	40.62	100.00	35.72	100.00	30.86	100.00
4SS	26.11	7.95	40.56	8.09	35.72	7.99	30.74	8.38
DS	26.11	6.21	40.56	6.41	35.72	6.27	30.79	6.86
HEXA	26.11	5.24	40.45	5.34	35.69	5.25	30.69	5.63
PROP	26.11	2.59	40.52	2.82	35.69	2.67	30.72	4.49

그림 6과 7은 각각 'carphone'과 'stepfan'의 100 프레임에서 화질과 상대적인 계산량을 보여주고 있다. 화질은 PSNR이고, 계산량은 FS 알고리즘에 대한 백분율이다. 평범한 움직임을 갖는 영상에서 화질은 그림 6(a)에 보이는 것처럼 FS나 기존 고속 알고리즘이나 차이를 구별하기 어렵지만 계산량은 그림 6(b)에 보이는 것처럼 고속 알고리즘이 FS보다 매우 적은 계산량이 요구되고, 거기에 제안된 알고리즘은 모든 프레임에서 기존 고속 알고리즘보다 더 적은 계산량이 요구된다. 그림 7은 제안된 알고리즘은 물론 기존 고속 알고리즘이 큰 움직임을 갖는 영상에서 화질 저하가 발생하는 것을 보여주고 있다. 그림 7(a)는 그림 6(a)와 달리 일부 프레임에서 화질이 크게 저하되고 있는데 이는 객체의 아주 빠른 움직임과 카메라의 급격한 움직임에 의해 찾고자 하는 대응블록이 탐색영역을 벗어나 발생한다. FS는 대응블록이 탐색영역을 벗어날지라도 최소 정합오차를 발생하는 차선의 대응블록을 찾아 그나마 화질을 유지한다. 그러나 대응블록이 탐색영역을 벗어나면 최적의 유사블록을 향해 정합오차가 단조 감소한다는 가정이 위배될 가능성이 매우 커져 제안된 알고리즘은 물론 기존 고속 알고리즘은 국부 최소에 빠지면서 화질 저하가 발생한다.

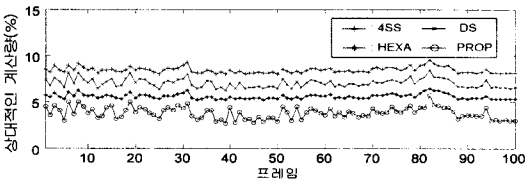
Ⅴ. 결 론

본 논문에서는 FS의 방대한 계산량을 줄이기 위해 탐색영역의 탐색점을 줄이는 고속 블록정합 알고리즘을 제안하였다. 제안된 알고리즘은 기존 고속 알고리즘의 전제 조건인 움직임 벡터가 탐색영역 중심에 집중한다는 것과 최적의 유사블록을 향해 정합오차가 단조 감소한다는 것에 각각 블록정합 단계 사이에 2 화소보다 1 화소 정합패턴 이동이 효율적이고 블록정합 진행 방향이 후진하지 않는다는 의미를 부여하였다. 따라서 제안된 알고리즘은 이전 단계들에서 결정된 유사블록들을 이용하여 최적의 유사블록을 향한 움직임을 예측하여 블록정합 진행방향을 ±45°로 제한하고, 단계 사이에 정합패턴을 1 화소 단위로 이동하여 불필요한 탐색점을 제거하여 블록정합 계산량을 줄이고 있다. 제안된 알고리즘의 성능 평가를 위해 4SS, DS, HEXA들과 비교하였다. 실험 결과 제안된 알고리즘은 큰 움직임을 갖는 영상에서 미미한 화질 저하가 발생하지만 평이한 움직임을 갖

는 영상에서 동등한 화질을 유지하고, 반면에 계산량은 적게는 20% 많게는 67%를 줄여주고 있다.

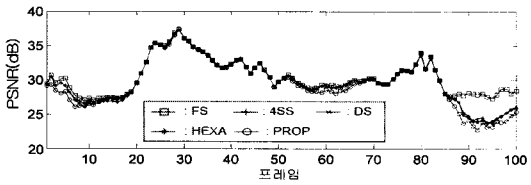


(a)

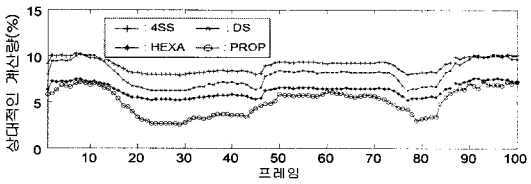


(b)

그림 6. 연속영상 'carphone' (a) 화질 (b) 계산량
Fig. 6. Image sequence 'carphone'
(a) picture quality (b) calculation volume



(a)



(b)

그림 7. 연속영상 'stefan' (a) 화질 (b) 계산량
Fig. 7. Image sequence 'stefan'
(a) picture quality (b) calculation volume

참고문헌

[1] T.Koga, K.Iinuma, A.Hirano, Y.Iigima and T.Ishiguro, Motion Compensated interframe coding for video conferencing, " Proc. Nat. Telecommun. Conf.. pp. G5.3.1-5.3.5, New Orleans, USA, Nov. 1981.

[2] L.M.Po and W.C.Ma, "A Novel Four-step Search Algorithm for Fast Block Motion Estimation," IEEE Trans. CSVT, vol. 6, pp. 313~317, June 1996.

[3] J.Y.Tham, S.Ranganath, M.Ranganath, and A.A.Kassim, "A Novel Unrestricted Center-based Diamond Search Algorithm for Block Motion Estimation," IEEE Trans. CSVT, vol. 8, pp. 313~317, A 1998.

[4] C.Zhu, X.Lin, and L.P.Chau, "Hexagon-based search pattern for Fast Block Motion Estimation," IEEE Trans. CSVT, vol. 12, pp. 349~355, May 2002.

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

[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] 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.

저자소개

오정수(Jeong-su Oh)

1992년 중앙대학교 대학원 전자공학석사
2001년 중앙대학교 첨단영상대학원 영상공학과 박사
2002년~현재 부경대학교 이미지시스템공학과
※관심분야 : 디지털영상처리, 영상검색