

# 제한된 서브블록을 이용한 고속 움직임 추정 알고리즘

준회원 김 성 희\*, 정회원 오 정 수\*

## Fast Motion Estimation Algorithm Using Limited Sub-blocks

Seong-hee Kim\* Associate Member, Jeong-su Oh\* Regular Member

### 요 약

정합 블록의 모든 화소가 블록 정합에 동등하게 기여하지 않고, 정합 오차는 영상 복잡도의 영향을 크게 받는다. 이런 사실을 근거로 본 논문에서 영상의 복잡도에 의해 선택되는 일부 서브블록만을 이용한 고속 움직임 추정 알고리즘을 제안한다. 제안된 알고리즘은 정합 블록을 16개의 서브블록으로 나누고, 각 서브블록의 영상 복잡도를 계산하고, 일부 복잡한 서브블록만을 사용하여 부분 블록 정합을 수행하여 움직임 벡터를 검출한다. 시뮬레이션 결과는 제안된 알고리즘이 기존 탐색 알고리즘과 비교하여 무시할 수 있을 정도의 화질 저하를 야기하지만 상당한 계산량을 줄여주는 것을 보여준다.

Key Words : motion estimation, block matching, PDE, sub-block, image complexity

### ABSTRACT

Each pixel in a matching block does not equally contribute to block matching and the matching error is greatly affected by image complexity. On the basis of the facts, this paper proposes a fast motion estimation algorithm using some sub-blocks selected by the image complexity. The proposed algorithm divides a matching block into 16 sub-blocks, computes the image complexity in every sub-block, executes partial block matching using some sub-blocks with large complexity, and detects a motion vector. The simulation results show that the proposed algorithm brings about negligible image degradation, but can reduce a large amount of computation in comparison with conventional algorithms.

### I. 서 론

컴퓨터와 네트워크의 발전으로 동영상은 누구나 쉽게 접할 수 있는 정보 전달 매체가 되었다. 그러나 동영상의 방대한 데이터 용량 때문에 전송이나 저장 시 압축이 필수 불가결하다. 동영상 압축의 기본 원리는 영상 내의 공간적(spatial), 시간적(temporal), 통계적(statistical) 중복성을 제거하는 것이며, 이 중에서 가장 많은 압축 효과를 얻을 수 있는 것은 시간적 중복성을 제거하는 것이며, 이는 움직임 추정 및 보상(motion estimation/motion compensa-

tion: ME/MC)을 통해 수행된다. 움직임 추정을 할 때 현재 가장 많이 사용하는 방법은 블록 정합 알고리즘(block matching algorithm: BMA)이며, 이 방법은 MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264 같은 동영상 표준에서 사용된다<sup>1-7)</sup>.

블록 정합 알고리즘의 가장 일반적인 방법인 전역 탐색법(full search algorithm: FSA)은 단순하고 좋은 성능을 보이지만 많은 계산량을 요구하는 문제를 갖고 있다. 그래서 계산량을 줄이기 위해 많은 고속 알고리즘들이 연구되고 있다. 대표적인 고속 알고리즘에는 다소 성능 저하가 있지만 계산량을

※본 논문은 2005년도 부경대학교 기성회 학술연구비에 의하여 연구되었음(PK-2005-046).

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

논문번호 : KICS2005-07-296, 접수일자 : 2005년 7월 20일, 최종논문접수일자 : 2006년 1월 25일

많이 줄일 수 있는 4단계 탐색법(four step search: 4SS), Diamond Search(DS) 등이 있으며 이들보다 계산량의 감소는 적지만 성능 저하가 없는 PDE(partial difference elimination), FFSSG(fast full search with sorting by gradient), SEA(successive elimination algorithm), MSEA(multi-level SEA) 등이 있다<sup>8-14)</sup>.

본 논문은 블록 정합에서 정합 블록의 일부 화소가 정합에 크게 영향을 준다는 사실에 근거하여 PDE 기반의 변형된 고속 알고리즘을 제안한다. 정합 블록에서 예지와 같이 불연속적인 밝기 변화가 큰 영역은 블록을 대표할 수 있고, 또한 이들은 큰 정합 오차를 발생시켜 PDE 알고리즘에서 유사 블록 가능성을 보다 빨리 판단할 수 있게 하고, 이로 인해 정합 계산량을 줄여준다. 제안된 알고리즘은 밝기 변화가 큰 영역을 표현하기 위해 영상 복잡도(image complexity: IC)를 정의하고, 알고리즘의 구현과 부가정보를 고려하여 16×16 정합 블록을 16개의 4×4 서브블록으로 나누고, 각 서브블록의 영상 복잡도를 계산하여 복잡도가 높은 일부 서브블록만으로 부분 블록 정합을 수행한다. 시뮬레이션 결과는 선택된 정합 서브블록의 수에 따라 다소 차이가 있지만 제안된 알고리즘이 기존 알고리즘과 비교해 미세하게 화질을 떨어뜨리지만 계산량을 크게 감소시키는 것을 보여준다.

## II. 기존 알고리즘

본 장에서는 제안된 알고리즘과 비교 평가될 기존의 움직임 추정 알고리즘인 전역 탐색법과 PDE를 기술한다.

### 2.1 전역 탐색법

전역 탐색법은 현재 프레임의 정합 블록과 이전 프레임의 탐색 영역 내의 모든 후보 블록 중에서 가장 유사한 블록을 찾아 움직임 벡터(motion vector: MV)를 얻는 알고리즘으로 단순하나 방대한 계산량을 요구한다. 블록간의 유사성은 식 (1)로 표현되는 절대 오차의 합(sum of absolute difference: SAD)을 이용하여 평가한다.

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

여기서  $f_i$ 와  $f_{i-1}$ 는 각각 현재와 이전 프레임이고,  $N$ 은 정합 블록의 크기이고,  $(i, j)$ 와  $(x, y)$ 는 각각 정합 블록과 탐색 영역 내의 후보 블록의 위치이다.

SAD가 최소가 되는 후보 블록의  $(x, y)$ 가 움직임 벡터가 된다.

### 2.2 PDE

PDE는 블록 정합 오차를 계산할 때, 후보 블록의 완전한 블록 정합 오차를 구하지 않고 일정 간격마다 그때까지의 부분 블록 정합 오차와 최소 블록 정합 오차를 비교한다. 부분 블록 정합 오차가 같거나 더 크다면, 나머지 부분 블록 정합 오차의 결과에 상관 없이 해당 블록은 유사 블록이 될 수 없으므로 해당 블록의 블록 정합 오차 계산을 그만두고 다음 후보 블록으로 이동한다. 이때 부분 블록 정합 오차는 식 (2)를 이용한다.

$$pSAD^k(x, y) = \sum_{i=0}^k \sum_{j=0}^{N-1} |f_i(i, j) - f_{i-1}(i+x, j+y)|, \quad k = 0, 1, 2, \dots, N-1 \quad (2)$$

여기서  $k$ 는 보통 정합 블록의 한 줄(16개의 화소)을 단위로 한다. PDE는 탐색 영역 내의 모든 후보 블록에 대해서 탐색을 하기 때문에 화질은 전역 탐색법과 같으면서도 계산량을 많이 줄일 수 있다.

PDE는 초기에 좋은 예측을 하여 최소 블록 정합 오차가 작게 되면 정합 중 정합 오차 계산을 그만두는 후보 블록이 많아질 것이다. 그리고 최적의 움직임 벡터는 대부분 탐색 영역의 중심에 있다. 이런 성질을 이용한 나선형 PDE(spiral PDE)는 탐색 영역 내에서 후보 블록들의 순차적 이동 대신에 중심에서 밖으로 나가는 나선형 이동을 하는 PDE로 단 순하지만 매우 효율적인 방법이다<sup>10)</sup>. 또한 PDE는 블록 정합 오차를 계산할 때 블록 내의 화소 정합 순서에 따라서도 성능이 달라질 수 있다. 블록 정합 오차에 크게 기여하는 화소를 먼저 계산한다면, 부분 블록 정합 오차의 경계에 빨리 도달하게 되어 부분 블록 정합 오차 계산을 더 빨리 그만두게 된다. 그래서 부분 블록 정합 초기에 큰 정합 오차가 발생하도록 블록 내 화소 정합의 순서를 재정렬하는 PDE도 연구되었다<sup>11,12)</sup>. 일반적으로 영상이 복잡할수록 정합 오차가 크게 발생하므로, 이들은 영상의 복잡도에 따라 부분 블록의 정합 순서를 재정렬하여 부분 블록 정합 초기에 큰 정합 오차가 발생하도록 하고 있다.

## III. 제안된 알고리즘

본 장에서는 정합 특성에 근거하여 블록 정합에

기여가 큰 화소들만을 이용해 부분 블록 정합을 수행하는 제안된 알고리즘을 기술하고 있다.

### 3.1 정합 특성

정합이란 대상물 사이의 유사성을 평가하는 것이고, 블록 정합처럼 작은 영역의 정합에서 밝기 변화의 불연속을 나타내는 예지는 시각적인 평가와 정량적인 평가 둘 다에서 중요한 요소이다. 즉 예지는 시각적으로 민감하여 영상 인식이나 비교에서 다른 영역보다 우선시 되어지고, 정량적 평가에서도 예지 영역이 다른 영역 보다 더 크게 나타난다. 따라서 예지를 포함하는 복잡한 영역은 블록 정합 알고리즘에서 다른 영역보다 기여도가 더 크다고 할 수 있다<sup>[6]</sup>. 이런 정합 특성은 그림 1의 누적 정합 오차에서 확인할 수 있다. 복잡한 영역의 영향을 보기 위해, 누적 정합 오차는 정합 블록을 4x4 크기의 16개의 서브블록들로 나누어 서브블록의 복잡한 영역을 먼저 계산하였다. 그림에서 최소 정합 오차를 갖는 후보 블록은 서브블록의 수에 관계없이 최소 혹은 그에 근사하는 부분 정합 오차를 갖고 있어 일부 서브블록만으로도 적절한 유사 블록을 찾을 수 있음을 의미한다. 그래프의 전반부에서 큰 기울기를 갖는 것은 정합 오차가 큰 것이므로 복잡한 영역이 큰 정합 오차를 발생시키고 있어 블록 정합에 있어 크게 기여할 수 있음을 의미한다. 그래프의 후반부에서 발생하는 큰 기울기는 후보 블록 내의 복잡한 영역에 의한 것이다.

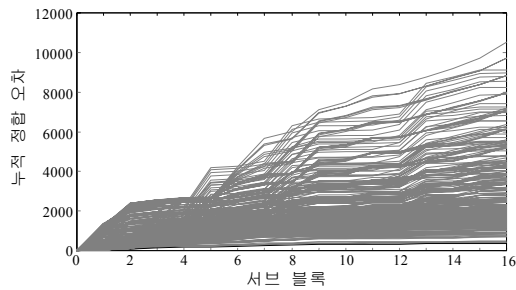


그림 1. 서브블록 단위의 누적 정합 오차

### 3.2 영상 복잡도

제안된 알고리즘은 정합 블록에서 기여가 큰 화소들만을 사용하여 정합을 수행하므로 정합을 위한 화소를 선택하는 것은 중요한 일이고, 이를 위해 본 논문에서 영상 복잡도를 정의하였다. 여기서 영상 복잡도는 정합 기여도가 큰 화소, 즉 정합 특성에 의해 예지를 포함하는 복잡한 영역의 화소를 찾는

기준이다. 또한 알고리즘 구현의 편의와 부가 정보를 고려하여 영상 복잡도는 정합 블록을 4x4의 16개의 서브블록으로 나누어 서브블록 단위로 계산된다. 영상 복잡도는 식 (3)에 의해 계산된다.

$$IC(u, v) = \sum_{l=0}^{n-1} \sum_{m=0}^{n-1} G(u+l, v+m),$$

$$G(i, j) = \sum_{p=-l}^1 \sum_{q=-1}^1 |f_i(i, j) - f_i(i+p, j+q)| \quad (3)$$

여기서  $n$ 은 서브블록의 크기이고,  $G(i, j)$ 는  $(i, j)$ 에서 8 방향 기울기의 합이고,  $IC(u, v)$ 는  $(u, v)$ 에서 시작되는 서브블록의 영상 복잡도이다.

### 3.3 서브블록을 이용한 고속 움직임 추정

본 논문은 정합 특성을 근거로 정합 기여도가 큰 일부 서브블록을 이용한 고속 움직임 추정 알고리즘을 제안하고 있다. 그림 2는 제안된 알고리즘의 흐름도를 보여주고 있다.

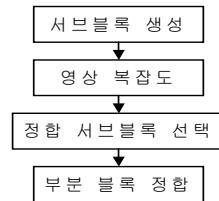


그림 2. 제안된 알고리즘의 흐름도.

“서브블록 생성” 과정에서는 현재의 16x16 정합 블록을 서로 겹치지 않는 16개의 4x4 서브블록으로 나누고, “영상 복잡도” 과정에서 식 (3)을 이용해 각 서브블록의 영상 복잡도를 계산한다. “정합 서브블록 선택” 과정에서는 영상 복잡도가 높은 서브블록을 정합 서브블록으로 선택하고, “부분 블록 정합” 과정에서 선택된 서브블록에서 복잡도가 높은 서브블록부터 식 (4)와 같이 부분 정합을 수행하고 최종적으로 움직임 벡터를 예측한다.

$$pSAD^k(x, y) = \sum_{l=1}^k \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} |f_l(u, v) - f_{l-1}(u+x, v+y)|,$$

$$k = 1, 2, \dots, K \quad (4)$$

여기서  $K$ 는 정합을 위해 선택된 서브블록의 수이고,  $(x, y)$ 와  $(u, v)$ 는 각각 정합 블록의 위치와  $K$ 에 의해 결정되는 서브블록의 위치이다.  $K$ 는 영상의 화질과 정합 계산량을 고려하여 사전에 선택되어야 한다.

#### IV. 시뮬레이션 결과 및 고찰

본 장에서는 제안된 알고리즘의 성능 평가를 위해, QCIF(144×176) 비디오 연속 영상 ‘foreman’, ‘stefan’, ‘carphone’에서 시뮬레이션이 수행되었다. 이때 탐색 영역은 정합 블록을 중심으로 ±7이고, 정합 블록 크기는 16×16이고, 탐색은 나선형 탐색 방법을 사용한다.

##### 4.1 서브블록을 이용한 정합

제안된 알고리즘의 일부 서브블록만을 이용한 정합 능력을 확인하기 위해 그림 3과 4에 정합을 위해 선택되는 서브블록의 수에 따라 전역 탐색법과 같은 움직임 벡터를 찾는 정합 블록의 수와 움직임 벡터에 의해 재구성된 영상의 화질을 최대 신호에 대한 잡음비(PSNR)로 나타내고 있다. 그림 3은 단 1개의 서브블록을 사용해도 영상에 따라 40%~70% 정도의 정합 블록이 전역 탐색법과 일치하는 움직임 벡터를 검출하며, 모든 서브블록을 사용해야 전역 탐색법과 일치하는 움직임 벡터를 검출하는 경우는 5개의 정합 블록도 안 되는 것을 보여 주고 있다. 이는 모든 서브블록을 필요로 하는 소수의 정합 블록을 위해서 모든 정합 블록을 16개의 서브블록을 사용해 블록 정합을 수행하는 것은 계산량의 낭비이고, 제한된 서브블록만으로도 블록 정합을 할 수 있음을 알려주고 있다. 그림 4는 일부 서브블록을 사용하여 움직임 벡터를 검출하지 못해도 비교적 유사한 움직임 벡터를 찾기 때문에 그에 따른 화질의 영향은 적어 사람의 눈으로 인식하기 어려운 정도임을 보여주고 있다. 실제 서브블록 1개만을 사용하는 경우 전체 서브블록을 사용하는 경우와 비교하면 영상에 따라 85~90%의 화질을 유지하고 있고, 9개 이상의 서브블록을 사용하면 화질 저하를 감지하기 어렵다.

##### 4.2 기존 알고리즘과의 비교

제안된 알고리즘의 성능 평가를 기존의 알고리즘인 전역 탐색법, 나선형 PDE(SPDE), 블록 내 정합 순서를 재정렬하는 PDE(APDE)와 화질과 정합을 위한 계산량을 비교한다. APDE는 블록 내 정합 순서를 결정하는 방법상 차이는 있지만, 제안된 알고리즘에서는 서브블록 16개 모두를 선택한 경우를 APDE로 간주한다. 실험에 앞서 정합을 위한 서브블록의 수를 결정하기 위해 서브블록의 수에 따른 화질을 비교하였다. 그 결과 전역 탐색법 화질의

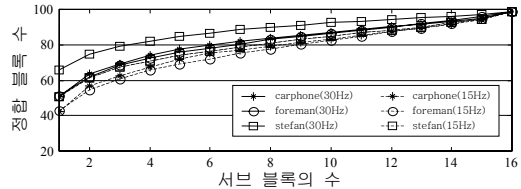


그림 3. 서브블록 수에 따른 전역 탐색법의 MV와 일치하는 정합 블록의 수

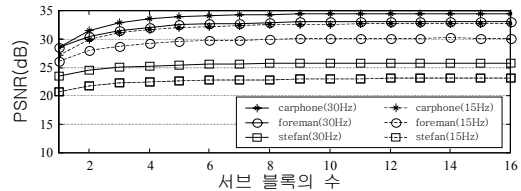
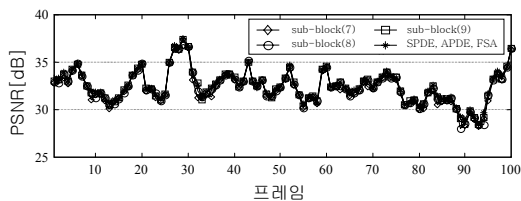


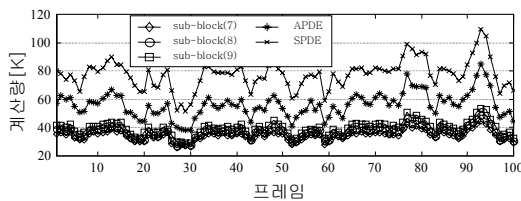
그림 4. 서브블록 수에 따른 화질

99.5% 이상을 유지하기 위해 30Hz 영상에서는 7개 이상의 서브블록, 15Hz 영상에서는 9개 이상의 서브블록이 요구된다. 그래서 기존 알고리즘과의 비교 실험에서 30Hz 영상에서는 7, 8, 9개의 서브블록이, 15Hz 영상에서는 9, 10, 11개의 서브블록이 사용되었다.

그림 5와 6은 30Hz의 연속 영상 ‘foreman’과 ‘stefan’에 대해서 프레임에 따른 화질과 계산량을 나타내고 있다. 화질은 PSNR로 비교하고 계산량은 한 프레임의 움직임 추정을 위해 계산되는 부분 블록의 개수이다. 부가적인 계산은 상대적으로 작아 고려하지 않았다. 서브블록 수에 따라 다소 차이는 있지만 전체적으로 기존 알고리즘과 비교해 화질 저하는 거의 눈에 띄지 않으나 계산량은 상당히 줄어들고 있다.



(a) PSNR



(b) 계산량

그림 5. 30Hz 연속 영상, ‘foreman’.

V. 결론

본 논문은 정합 블록에서 일부 화소가 정합 오차에 크게 기여하고, 이들이 움직임 벡터 결정에 중요한 역할을 한다는 사실에 근거해 정합 블록 내의 일부 서브블록만을 이용한 고속 움직임 추정 알고리즘을 제안하였다. 정합을 위한 화소들은 영상이 복잡한 영역에 분포되어 서브블록 단위로 복잡도를 계산해 복잡도의 크기에 따라 정해진 수만큼의 서브블록이 선택되어 부분 블록 정합이 수행되었다. 제안된 알고리즘의 성능 평가를 위해 일부 서브블록을 이용한 정합 능력을 평가하고, 제안된 알고리즘을 30Hz와 15Hz 연속 영상에 적용하여 기존 알고리즘과 화질과 계산량에 대해 비교 평가하였다. 시뮬레이션 결과 많은 정합 블록이 적은 서브블록으로도 정확한 움직임 벡터를 검출할 수 있는 것을 보여주었고, 기존 알고리즘과 비교에서 화질은 기존 알고리즘과 비슷하게 유지하면서 계산량은 눈에 띄게 감소시키는 것을 보여주었다. 또한 제안된 알고리즘은 다른 고속 움직임 추정 기법과 결합한다면 더 많은 계산량의 감소를 얻을 수 있으며, 정합에 사용되는 서브블록의 수를 조절함으로써 화질과 계산량을 조절할 수도 있다.

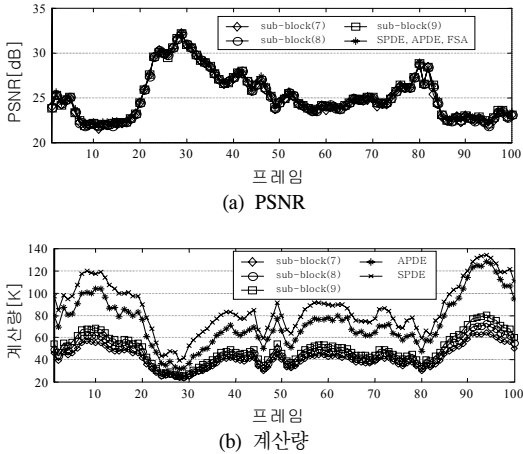


그림 6. 30Hz 연속 영상, "stefan"

표 1. 30Hz 100 프레임의 연속 영상

sequence	car phone		foreman		stefan	
방법	PSNR(dB)	계산량	PSNR(dB)	계산량	PSNR(dB)	계산량
Proposed (서브블록 7)	99.33%	12.10%	99.93%	11.73%	99.24%	14.38%
Proposed (서브블록 8)	99.59%	13.12%	99.60%	12.67%	99.46%	15.79%
Proposed (서브블록 9)	99.78%	14.09%	99.63%	13.59%	99.51%	17.15%
APDE	34.01	60118	32.55	56084	25.34	73608
APDE (서브블록 8)	100%	20.56%	100%	19.18%	100%	25.18%
SPDE	34.01	77266	32.55	77526	25.34	88481
SPDE (서브블록 8)	100%	26.43%	100%	26.52%	100%	29.57%
FSA	34.01	292336	32.55	292336	25.34	292336
FSA (서브블록 8)	100%	100%	100%	100%	100%	100%

표 2. 15Hz 50 프레임의 연속 영상

sequence	carphone		foreman		stefan	
방법	PSNR(dB)	계산량	PSNR(dB)	계산량	PSNR(dB)	계산량
Proposed (서브블록 9)	99.65%	15.55%	99.55%	16.44%	99.15%	23.81%
Proposed (서브블록 10)	99.74%	16.65%	99.78%	17.65%	99.34%	25.89%
Proposed (서브블록 11)	99.84%	17.98%	99.89%	18.84%	99.50%	27.89%
APDE	32.16	67585	29.7	71273	22.7	107938
APDE (서브블록 16)	100%	23.12%	100%	24.38%	100%	36.92%
SPDE	32.16	85910	29.7	93654	22.7	121059
SPDE (서브블록 16)	100%	29.39%	100%	32.04%	100%	41.41%
FS	32.16	292336	29.7	292336	22.7	292336
FS (서브블록 16)	100%	100%	100%	100%	100%	100%

표 1과 2는 30Hz와 15Hz의 세 연속 영상에서 평균적인 화질과 계산량을 비교하고 있다. 영상에 따라 다소 차이가 있지만 화질은 기존 알고리즘과 비교해 적은 서브블록에서 1% 미만, 많은 서브블록에서는 0.5% 미만의 저하가 생기지만 눈으로 느끼기 어렵고, 계산량은 전역 탐색법의 10~15%에 불과하고 나선형 PDE나 APDE에 비해서도 효율적이라는 것을 알 수 있다. 표 2의 15Hz 영상의 경우 30Hz 영상 보다 움직임이 크므로 서브블록을 더 많이 사용하였다.

참고 문헌

- [1] Frederic Dufaux and Fabrice moscheni, "Motion Estimation Technique for Digital TV : A Review and a New Contribution," *Proceedings of the IEEE*, 83, pp.858~876, 1995.
- [2] MPEG-1 Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mb/s, ISO/IEC 11 172, 1993.
- [3] MPEG-2 Generic Coding of Moving Pictures and Associated Audio Information, ISO/IEC 13 818, 1996.
- [4] MPEG-4-Information Technology-Coding of Audio-Visual Objects-Part 2: Visual, ISO/IEC 14 496-2, 2000.
- [5] "Video Codec for Audiovisual Services at p\_64 kbits," *International Telecommunications Union, ITU-T Recommendation H.261*, 1993.
- [6] "Video Coding for Low Bitrate Communi-

ation,” *International Telecommunications Union, ITU-T Recommendation H.263*, 1998.

[7] “Joint Final Committee Draft of Joint Video Specification(itu-trec. h.264 - iso/iec 14 496-10 avc),” Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, Doc. JVT-D157, 2002.

[8] L. M. Po and W. C. Ma, “A novel four-step search algorithm for fast block motion estimation,” *IEEE Trans. Circuits systems Vodeo Technology*, 6, pp. 313-317, June 1996.

[9] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” *IEEE Trans. Image Processing*, 9(2), pp. 287-290, 2000.

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

[11] J. N. Kim, S. C. Byun, Y. H. Kim, and B. H. Ahn, “Fast full search motion estimation algorithm using early detection of impossible candidate vectors,” *IEEE Trans. Signal Process.*, 50(9), pp. 2355 - 2365, 2002.

[12] B. Montrucchio and D. Quaglia, “New Sorting-Based Lossless Motion Estimation Algorithm and a Partial Distortion Elimination Performance Analysis,” *IEEE Trans. Circuits and Systems for Video Technology*, 15(2), pp 210-220, 2005.

[13] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” *IEEE Trans. Image Processing*, 8, pp. 435~438, 1999.

[14] X. Q. Gao, C. J. Duanmu, and C. R. Zou, “A multilevel successive elimination algorithm for block matching motion estimation,” *IEEE Trans. Image Processing*, 9, pp.501~504, 2000.

[15] 김성희, 오정수, “서브 블록을 이용한 블록 정합 알고리즘,” *한국해양정보통신학회 2005년도 춘계종합학술대회*, 9(1), pp.655~658, 2005.

김 성 희 (Seong-hee Kim)

준회원



2004년 8월 부경대학교 화상정보공학부 졸업  
2004년 8월~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