

개선된 서브블록 정합 알고리즘

종신회원 오 정 수*

Improved Sub-block Matching Algorithm

Jeong-su Oh* *Lifelong Member*

요 약

본 논문은 블록 정합에 크게 기여하는 일부 서브블록들만 사용하는 서브블록 정합 알고리즘을 개선하기 위한 블록 정합 알고리즘을 제안한다. 하나의 서브블록 그룹을 사용하는 기존 알고리즘과 달리 제안된 알고리즘은 두 개의 서브블록 그룹을 사용하고 있다. 소 그룹을 이용한 정합은 적은 계산량으로 유사 블록 가능성이 큰 후보 블록들을 선택하고, 선택된 블록에서 대 그룹을 이용해 추가적인 정합은 잘못된 정합을 줄인다. 모의 실험 결과들은 제안된 알고리즘이 기존 알고리즘과 비교하여 같은 계산량에서 항상 높은 화질을 갖는 것을 보이고 있고, 적은 서브블록들을 이용한 정합에서 매우 뛰어난 성능을 갖는 것을 보여주고 있다.

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

ABSTRACT

This paper proposes a block matching algorithm to improve the sub-block matching algorithm that uses partial sub-blocks being a great contribution to the block matching. Unlike the conventional algorithm using the one sub-block group the proposed algorithm uses two sub-block groups. The matching using the small group selects a candidate block to be a good possibility of a similar block with a small computation cost and the additional matching using the large group in the selected block decreases a wrong matching. Simulation results show that the proposed algorithm always has good image quality at the same computation cost as compared to the conventional algorithm and it has an outstanding performance at the matching using a few sub-blocks.

I. 서 론

멀티미디어의 핵심 요소인 영상 데이터는 최근 우리 생활에서 중요한 정보로 자리를 잡고 있다. 그러나 동영상을 포함한 영상 데이터는 그 양이 방대하여 이를 실제 응용 분야에 적용하기 위해 영상 압축은 필수적이다. 영상 압축의 경우 영상 데이터가 갖는 중복성에 의존하여 데이터의 양을 줄이는 과정을 말한다. 영상 데이터의 중복성에는 공간적 중복성, 시간적 중복성, 통계적 중복성 크게 세가지로 구분한다. 시간적 중복성을 제거하는 것은 동영상 압축의 가장 핵심적인 원

리이고, 시간적 중복성을 제거하는 대표적 방법은 움직임 추정 (motion estimation)이며 이를 위해 흔히 사용되는 알고리즘은 블록 정합 알고리즘 (block matching algorithm: BMA)이다^[1]. 블록 정합 알고리즘의 원천적 방법인 전역 탐색 알고리즘 (full search algorithm: FSA)은 이전 프레임의 탐색 영역 전체를 최소 단위로 이동하면서 최적의 유사 블록을 찾는 것으로 정확하다는 장점이 있지만, 계산량이 많다는 단점을 갖고 있다. 그래서 전역 탐색 알고리즘의 계산량을 줄이기 위한 고속 알고리즘이 지속적으로 연구되고 있다. 대표적인 알고리즘으로 탐색 영역을 제한하

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

논문번호 : KICS2009-10-447, 접수일자 : 2009년 10월 7일, 최종 게재 논문통보일자 : 2010년 6월 28일

여 계산량을 감소시키는 4단 탐색법 (four step search: 4SS)^[2], 다이아몬드 탐색법 (diamond search: DS)^[3], 육각형 탐색 (hexagon-based search: HEXA)^[4] 등이 있고, 정합 과정에서 불가능한 유사 블록을 조기에 판별하여 계산량을 감소시키는 PDE(partial difference elimination)^[5], SEA(successive elimination algorithm)^[6], MSEA(multi-level SEA)^[7] 등이 있다. 앞의 알고리즘들에서는 계산량 감소가 크나 화질 저하가 발생하고 뒤의 알고리즘들에서는 화질 저하가 없으나 계산량 감소가 제한적이다. 그리고 최근에는 정합 블록 내 일부 화소가 정합에 크게 영향을 준다는 사실^[8,9]에 근거하여 정합에 기여도가 큰 서브블록 (sub-block: SB)들을 이용한 서브블록 정합 알고리즘 (sub-block matching algorithm)이 제안되었다^[8].

본 논문은 서브블록 정합 알고리즘에서 아주 적은 수의 서브블록을 사용하는 경우 화질이 떨어지는 것을 보완하기 위한 개선된 서브블록 정합 알고리즘을 제안한다. 제안된 알고리즘은 매크로블록 (macro-block)을 16개의 4×4 서브블록들로 나누고, 각 서브블록의 영상 복잡도를 계산하여 영상 복잡도에 따라 내림차순으로 정렬한다. 그리고 서브블록 정합 알고리즘을 위해 영상 복잡도가 높은 서브블록들로 2개의 그룹을 만든다. 한 그룹은 영상 복잡도가 높은 소수 서브블록들로 구성되고, 다른 한 그룹은 영상 복잡도가 높은 다수 서브블록들로 구성되고 소수 서브블록들을 포함한다. 기존 서브블록 정합 알고리즘은 한 그룹의 서브블록들을 이용해 서브블록 정합을 수행하지만 제안된 알고리즘은 소수 서브블록들을 이용한 서브블록 정합을 수행해 유사 블록 가능성을 판단하고 가능성이 큰 후보 블록에 대해 다수 서브블록들을 이용해 보다 정확한 서브블록 정합을 수행한다. 모의 실험 결과는 기존 서브블록 정합 알고리즘에서 소수의 서브블록들을 이용한 블록 정합에서 발생하는 화질 열화를 크게 개선시켜주는 것을 보여주고 있다.

II. 기존 서브블록 정합 알고리즘

본 장에서는 제안된 알고리즘과 비교될 기존 알고리즘으로 영상 복잡도에 의해 선택된 서브블록들을 이용한 서브블록 정합 알고리즘을 기술한다.

2.1 서브블록 정합 알고리즘

서브블록 정합 알고리즘은 블록 정합에서 블록 내의 모든 화소들이 동등하게 기여하는 것은 아니라는 사실에 근거해 정합에 크게 기여하는 서브블록의 화

소들만을 이용한 정합 알고리즘으로 다음과 같은 과정을 통해 수행된다.

1. 16×16 매크로블록을 서로 겹치지 않는 16 개의 4×4 서브블록들로 나눔
2. 16 개의 서브블록들의 영상 복잡도를 계산
3. 영상 복잡도 크기에 따라 내림차순으로 정렬
4. 상위 영상 복잡도를 갖는 지정된 수의 서브블록들을 선택
5. 선택된 서브블록들만을 이용해 서브블록 정합을 수행

서브블록 정합에서 정합에 크게 기여하는 화소는 서브블록 단위의 영상 복잡도를 이용해 결정한다. 이때 영상 복잡도는 식 (1)과 같이 서브블록 단위로 계산된다.

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

$$G(i,j) = \sum_{p=-1}^1 \sum_{q=-1}^1 |f(i,j) - f(i+p,j+q)|$$

여기서 n 은 서브블록의 크기이고, $IC(u,v)$ 는 (u,v) 에서 시작되는 서브블록의 영상 복잡도이고, $G(i,j)$ 는 (i,j) 에서 8 방향 기울기의 합이다. 본 논문에서는 움직임 추정 성능에 영향을 주지 않고 부가 계산량을 줄일 수 있도록 $G(i,j)$ 를 식 (2)와 같이 8 방향 기울기를 4 방향 기울기로 수정하였다.

$$G(i,j) = \sum_{p=-1}^1 \sum_{q=-1}^1 |f(i,j) - f(i+p,j+q)|, \quad (2)$$

$$|p+q|=1$$

2.2 서브블록 정합 알고리즘 성능 분석

서브블록 정합 알고리즘의 계산량은 전역 탐색 알고리즘과 비교하여 식 (3)과 같이 계산된다..

$$C = \frac{k}{16} \times 100 + 2.11[\%] \quad (3)$$

여기서 k 는 서브블록 정합 알고리즘에 사용된 서브블록 수이고, 상수 16과 2.11은 각각 전역 탐색 알고리즘에서 매크로블록의 서브블록 수와 영상 복잡도 계산을 위해 사용된 부가 계산량이다.

서브블록 정합 알고리즘의 화질은 그림 1에 30[Hz] QCIF 영상 'carphone', 'foreman', 'stefan'의 각 100 프레임에 처리한 결과를 서브블록 수에 따라 상대적

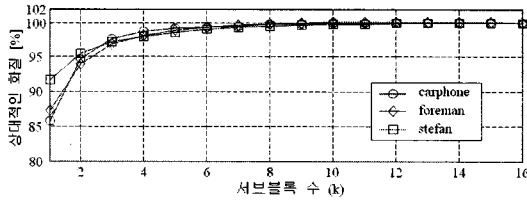


그림 1. 서브블록 수에 따른 상대적인 화질

인 화질로 보여주고 있다. 16 개의 서브블록을 사용하면 전역 탐색 알고리즘과 동일한 알고리즘이 된다. 6 개 이상의 서브블록을 사용하면 99[%] 이상을 보이고 12 개 이상의 서브블록을 사용하면 99.9[%] 이상을 보이고 있다. 한편 소수의 서브블록을 사용하는 경우 4 개의 서브블록까지는 비교적 완만한 화질 저하를 보이고 있으나 이보다 적은 수의 서브블록을 사용하면 85[%] 이상 유지하고 있지만 상대적으로 급격한 저하를 보이고 있다.

III. 제안된 서브블록 정합 알고리즘

II장을 통해 기존의 서브블록 알고리즘은 단지 1~2 개의 서브블록만을 사용해도 영상에 따라 전역 탐색 알고리즘의 85.7[%]~91.6[%]정도 화질을 유지시켜주고 있어 소수 서브블록을 이용한 서브블록 정합 알고리즘의 유효함을 보여주고 있고 아주 적은 수의 서브블록들을 사용하는 경우 상대적으로 크게 화질 저하되는 문제도 보여주고 있다. 또한 12개 이상의 서브블록을 사용할 경우 99.9[%] 이상으로 전역 탐색 알고리즘과 동등한 성능을 나타내는 것도 보여주고 있다. 이런 사실을 근거로 두 개의 서브블록 그룹을 이용한 개선된 서브블록 정합 알고리즘을 제안하고 있다. 즉, 소수 서브블록들을 이용한 정합에서 적은 계산량으로 유사 블록 가능성을 판단하고, 가능성 있는 후보 블록에 대해 다수 서브블록들을 이용한 정합에서 약간의 계산량 추가로 보다 정확한 정합을 수행해 화질 개선을 이룬다. 그림 2는 개선된 서브블록 정합 알고리즘의 매크로블록 단위의 정합 흐름도를 보여주고 있다.

서브블록 정렬 블록은 16×16 매크로블록을 서로 겹치지 않는 16 개의 4×4 서브블록들로 나누고, 서브블록들의 영상 복잡도를 이용해 내림차순으로 정렬하고, 서브블록 정합에 사용할 소수 서브블록 그룹 (N1) 과 다수 서브블록 그룹 (N2)를 결정한다. SB 블록 정합(N1) 블록은 현 탐색점에서 소수 서브블록들을 이용해 서브블록 정합을 수행한다. 소수 서브블록들의 정합 오차 (SAD_N1)가 소수 서브블록들의 최소 정합

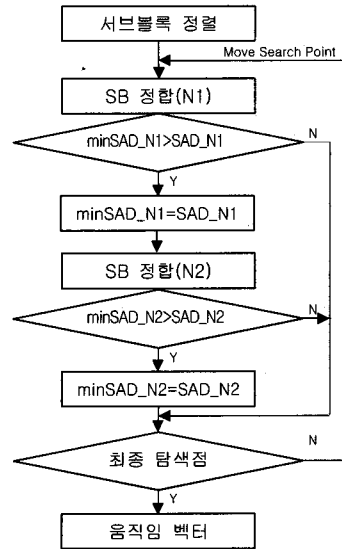


그림 2. 제안된 서브블록 정합의 흐름도

오차 (minSAD_N1)보다 크면 최종 탐색점인지 확인하고, SAD_N1가 minSAD_N1보다 작으면 minSAD_N1의 값을 SAD_N1로 대체하고, 블록 정합(N2) 블록에서 다수 서브블록들을 이용해 서브블록 정합을 수행한다. 다수 서브블록들의 정합 오차 (SAD_N2)가 다수 서브블록들의 최소 정합 오차 (minSAD_N2)보다 크면 최종 탐색점인지 확인하고, SAD_N2가 minSAD_N2보다 작으면 minSAD_N2의 값을 SAD_N2로 대체하고 최종탐색점인지 확인한다. 최종 탐색점 블록에서는 현재 탐색점이 최종 탐색점인지 확인하고 최종 탐색점이 아니면 다음 탐색점으로 이동하고, 최종 탐색점이면 minSAD_N2의 위치를 움직임 벡터로 결정하고 다음 매크로블록으로 이동한다. 제안된 알고리즘에서 영상 복잡도는 식 (2)를 식 (1)에 적용하여 계산되고, N1의 서브블록들은 N2의 서브블록들에 포함되고, N1의 서브블록 수는 실험에 따라 가변시키나 N2의 서브블록 수는 13로 고정하고 있다.

IV. 실험 결과

본 장에서는 제안된 알고리즘의 성능 평가를 위해 기존 서브블록 정합 알고리즘과 서브블록 수에 따라 화질과 계산량을 비교하고 있다. 모의 실험은 30[Hz] QCIF 영상 'carphone', 'foreman', 'stefan', 'table tennis'의 각 100 프레임을 대상으로 하였고, 정합을 위한 매크로블록의 크기는 16×16이고, 탐색 영역은 정합 블록을 중심으로 ±7이다.

표 1은 제안된 알고리즘이 크게 영향을 주는 1~6 개의 서브블록 수에서 기존 알고리즘과 제안된 알고리즘의 화질 (PSNR)과 계산량 (CCost)을 비교하고 있다. 화질은 PSNR로 표현하였고, 계산량은 매크로블록당 정합과 복잡도 계산을 위해 사용된 +와 -의 연산 수로 표현하였다. 그리고 화질과 계산량의 증가량인 $\Delta PSNR$, $\Delta CCost$ 는 식 (4)와 식 (5)로 계산된다.

$$\Delta PSNR = \left(\frac{PSNR_p(n) - PSNR_C(n)}{PSNR(16)} \right) \times 100\% \quad (4)$$

$$\Delta CCost = \left(\frac{CCost_p(n) - CCost_C(n)}{CCost(16)} \right) \times 100\% \quad (5)$$

여기서 첨자 P와 C는 각각 제안된 알고리즘과 기존 알고리즘을 의미하고, n은 서브블록 수로 16은 전역 탐색 알고리즘과 동등한 결과이다. 모든 영상과 모든 서브블록 수에서 제안된 알고리즘은 계산량 증가보다 화질 증가가 크고, 서브블록 수가 적을수록 화질 증가는 더욱 더 크게 발생한다. 기존 알고리즘에 대한 제안된 알고리즘의 계산량 증가는 전역 탐색 알고리즘기준으로 최대 3.40[%]가 증가하지만 화질은 최대 13.53[%]가 증가하고 있다. 이 결과들은 제안된 알고리즘이 계산량을 다소 추가하여 기존 알고리즘의 화

표 1. 기존 서브블록 정합 알고리즘과 비교

실험 영상	SB 수	conventional		proposed		증가량 (%)		R_PSNR
		PSNR	Ccost	PSNR	Ccost	$\Delta PSNR$	$\Delta CCost$	
carphone	1	29.15	7902	33.75	9484	13.53	1.64	99.25
	2	32.23	13808	33.87	14894	4.85	1.13	99.60
	3	33.20	19714	33.94	20578	2.19	0.90	99.81
	4	33.56	25620	33.97	26331	1.19	0.74	99.88
	5	33.75	31525	34.00	32125	0.72	0.62	99.96
	6	33.80	37431	34.00	37945	0.59	0.53	99.97
	16	34.01	96489					
foreman	1	28.39	7902	32.16	9570	11.60	1.73	98.81
	2	30.58	13808	32.36	15002	5.46	1.24	99.41
	3	31.56	19714	32.44	20890	2.73	1.01	99.67
	4	31.95	25620	32.46	26436	1.57	0.85	99.73
	5	32.20	31525	32.50	32210	0.92	0.71	99.84
	6	32.35	37431	32.54	38000	0.58	0.59	99.98
	16	32.55	96489					
stefan	1	23.23	7902	25.07	10510	7.26	2.70	98.91
	2	24.22	13808	25.19	15638	3.84	1.90	99.42
	3	24.62	19714	25.25	21123	2.49	1.46	99.66
	4	24.83	25620	25.28	26748	1.75	1.17	99.75
	5	24.97	31525	25.29	32451	1.27	0.96	99.79
	6	25.08	37431	25.30	38194	0.89	0.79	99.86
	16	25.34	96489					
table	1	24.77	7902	26.84	11186	7.57	3.40	98.35
	2	25.73	13808	27.02	16100	4.72	2.38	99.00
	3	26.16	19714	27.07	21470	3.36	1.82	99.20
	4	26.49	25620	27.13	27002	2.36	1.43	99.41
	5	26.67	31525	27.16	32612	1.80	1.13	99.54
	6	26.77	37431	27.18	38297	1.52	0.90	99.61
	16	27.29	96489					

질을 크게 개선시키고 있음을 보여주는 것이다. 한편 전역 탐색 알고리즘에 대한 상대적인 화질 (R_PSNR)에서 단 1 개의 서브블록을 사용하는 경우 영상에 따라 전역 탐색 알고리즘 대비 85.7[%]~91.64[%]의 화질인 기존 알고리즘을 제안된 알고리즘은 98.35[%]~99.25[%]의 화질로 개선시키고 있고, 2 개의 서브블

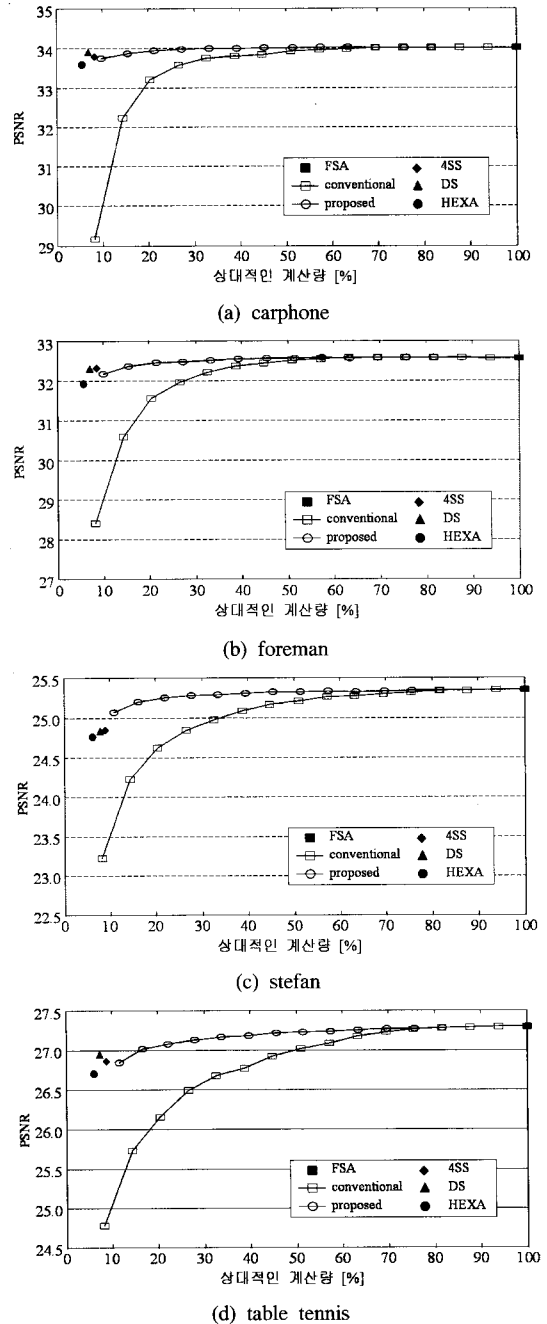


그림 3. 제안된 알고리즘의 성능

록을 사용하는 경우 모든 영상에서 전역 탐색 알고리즘 대비 99[%] 이상의 화질을 보이고 있다.

표 2는 손실이 있는 기존 고속 알고리즘 4SS, DS, HEXA와 1개의 서브블록을 사용한 제안된 알고리즘 P(1)을 비교하고 있다. 기존 알고리즘과 제안된 알고리즘은 계산량 표현을 위해 각각 검색된 후보블록의 수와 연산 수를 사용하여 직접적인 비교가 어려워 FSA에 대한 상대적인 비율로 보여주고 있다. 제안된 알고리즘은 유사 화질에 대해 계산량이 1-2%정도 많은데 그 원인이 식 (3)에서 지정한 부가계산에 의해 발생하는 것이고 고화질 요구에 의해 다수의 서브블록들을 사용시 그 영향은 줄어든다.

그림 3은 제안된 알고리즘의 성능을 시각적으로 잘 보여주기 위해 계산량에 따른 화질을 보여 주고 있다. 계산량은 전역 탐색 알고리즘의 계산량을 기준으로 백분율로 표시되었다. 기존 알고리즘은 모든 서브블록 수에 대한 결과를 보여주고 있으나 제안된 알고리즘은 다수 서브블록의 수인 13 서브블록까지만 결과를 보여주고 있다. 제안된 알고리즘은 주어진 계산량에 대해 항상 높은 PSNR을 보이고 있고, 계산량이 적을수록 PSNR 개선은 더 높여주고 있다. 전역 탐색 알고리즘의 10[%] 계산량을 사용하는 경우 제안된 알고리즘은 PSNR을 최대 4[db] 이상 개선시키고 있다. 한편 그림 3에서는 손실 고속 알고리즘과도 비교되고 있다. 계산량에 따른 화질을 볼 때 복잡한 'stefan' 영상에서 다른 알고리즘보다 우수한 반면 다른 영상에서는 동등한 화질에서 미세한 계산량이 더 요구되고 있다.

표 2. 기존 손실 고속 알고리즘과 비교

실험영상	carphone		foreman		stefan		table	
	PSNR	cost [%]	PSNR	cost [%]	PSNR	cost [%]	PSNR	cost [%]
FSA	34.01	100	32.56	100	25.34	100	27.29	100
4SS	33.82	8.4	32.33	8.7	24.85	9.0	26.73	8.9
DS	33.93	7.0	32.32	7.1	24.84	8.0	26.89	7.5
HEXA	33.60	5.9	31.92	5.7	24.76	6.3	26.42	6.1
P(1)	33.75	9.8	32.16	9.9	25.07	10.3	26.84	11.6

V. 결 론

본 논문은 블록 정합에 크게 기여하는 서브블록들을 이용한 고속 움직임 추정 알고리즘인 서브블록 정합 알고리즘을 개선하는 서브블록 정합 알고리즘을 제안하였다. 기존 서브블록 정합 알고리즘은 소수 서브블록들을 사용하는 경우 계산량은 서브블록 수에 비례하게 크게 감소하나 화질은 상대적으로 더 크게

감소하는 문제를 갖는다. 제안된 알고리즘은 두 개의 서브블록 그룹을 사용하여 소수 서브블록들로 계산량 감소를 유지시키고, 필요에 따라 다수 서브블록으로 화질을 개선시키고 있다. 모의 실험 결과에서 제안된 알고리즘은 1 개의 서브블록만으로 전역 탐색 알고리즘의 98.35[%]~99.25[%] 화질을 유지시키고 또한 계산량에 대한 화질이 기존 알고리즘보다 항상 우수하고 계산량이 적을수록 그 우수성 높아지는 것을 보여 주고 있다. 한편 기존 손실 고속 알고리즘과 비교 시 기존 알고리즘은 고정된 성능이지만 제안된 알고리즘은 영상에 따라 다소 차이가 있지만 기존 알고리즘과 동등한 성능을 보이면서 필요 시 서브블록 수를 증가시켜 화질을 개선할 수 있는 장점이 있다.

참 고 문 헌

- [1] F. Dufaux and F. moscheni, "Motion Estimation Technique for Digital TV : A Review and a New Contribution," Proceedings of the IEEE, Vol.83, pp.858~876, Jun. 1995.
- [2] L. M. Po, and W. C. Ma, "A novel four-step search algorithm for block motion estimation," IEEE Trans. CSVT, Vol.6, No.3, pp.313~317, 1996.
- [3] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation, IEEE Trans, Image Process, Vol.9, No.2, pp.287, 2000.
- [4] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," IEEE Trans. CSVT, Vol.12, No.5, pp.349~355, 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. Image Processing, 8, pp. 435~438, 1999.
- [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, 9, pp.501-504, 2000.

- [8] 김성희, 오정수, “제한된 서브블록을 이용한 고속 움직임추정 알고리즘”, 한국통신학회 논문지, 31권, 3C호, pp.258~263, 2006.
- [9] J.N.Kim and et al., “Adaptive matching scan algorithm based on gradient magnitude for fast full search in motion estimation,” IEEE Trans. Consumer Electronics, Vol.45, pp.762~772, 1999.

오 정 수 (Jeong-su Oh)

중신회원



1990년 2월 중앙대학교 전자공학과

1992년 8월 중앙대학교 대학원 전자공학 석사

2001년 8월 중앙대학교 첨단영상대학원 영상공학과 박사

2002년~현재 부경대학교 이
미지시스템공학과

<관심분야> 디지털 영상 처리, 적외선 신호처리