

# 5-방향 탐색 알고리즘을 이용한 고속 움직임벡터 예측

## (A High Speed Motion Vector Estimation using 5-Directional Search Algorithm)

李根泳 \*

(Keun Young Lee)

### 요 약

본 논문은 동영상 부호화에 효과적인 고속 움직임추정 알고리즘, 5DS를 제안하였다. 이 알고리즘에서 움직임을 추정하기 위해서는 대부분의 움직임이 상하좌우 4방향으로 일어나기 때문에 먼저 이 방향들로 MAD를 구하여 블록정합을 한 다음, 이들 정합오차가 작은 쪽의 대각선 방향 탐색점 1개를 추가로 선정하고 블록정합을 시도하였다. 이렇게 함으로써 LOG 탐색 알고리즘이 갖지 못한 대각선 방향의 탐색이 가능하였다. 제안된 알고리즘은 기존의 대표적인 3단계탐색법 TSS와 LOG 탐색법과 비교하여 화질면에서는 대등하였으나 속도면에서 1.9배, 1.2배가 각각 빨라짐을 확인 하였다.

### Abstract

This paper presents a fast motion estimation algorithm, 5DS, useful for video coding. We first try block matching to 4 directions(N,E,W,S) to estimate motions in this algorithm, since most of motions in video are oriented to those directions, and then try one additional diagonal matching between the matching points having small MADs. It makes this algorithm possible for searching through a diagonal direction which is not adequate to logarithmic (LOG) search algorithm. This proposed algorithm has almost same PSNR but, 1.9, 1.2 times faster than classical block matching methods such as three steps search(TSS) and LOG search algorithms,

### I. 서 론

정보통신의 발달로 동영상 처리에 대한 요구가 증가하게 되었는데, 이에 따른 가장 큰 문제점은 데이터량의 증가이다. 동영상 데이터를 컴퓨터로 처리, 다양한 저장매체에 저장, 기존 통신망을 이용한 전송, 방송채널을 통해 송신하기 위한 H.263<sup>[1]</sup>, MPEG1<sup>[2]</sup> 및 MPEG2<sup>[3]</sup> 등의 압축규격이 제정되었다. 이 규격들은 시간에 대한 움직임 추정/보상과 공간을 이용하여 각각 시간상과 공간상에 존재하는 데이터의 중복을

제거하고 다양한 응용에 적합한 압축 비트열의 구조를 정의하고 있다.

시간적으로 연속된 동영상의 경우, 일반적으로 영상의 움직임은 작은 영역내에서 일어나고 그 크기가 크지 않다는 사실에 기초하여 움직임벡터를 추정하고 있다. 이 움직임벡터를 사용하여 연속된 프레임 사이의 상관성을 제거한다. 동영상에서 움직임예측하는 과정의 시간이 전체 부호화 시간의 약 30% 정도로 많은 시간이 소요되기 때문에, 빠르고 효과적인 움직임 추정 방법을 찾아 내는 것이 아주 중요하다. 이것을 위해 가장 많이 이용되는 방법이 연산이 간단하고 하드웨어 구현이 용이한 블록정합 알고리즘(BMA)이다.

블록정합 알고리즘 가운데에서 기본적인 전역탐색 알고리즘(Full Search, FS)은 주어진 탐색영역내의 모

\* 正會員, 成均館大學校 電子工學科

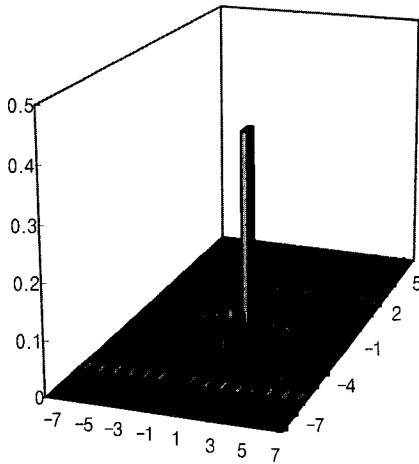
(Dept. of Elec. Eng., SungKyunKwan Univ.)

接受日字: 1997年5月17日, 수정완료일: 1998年2月25日

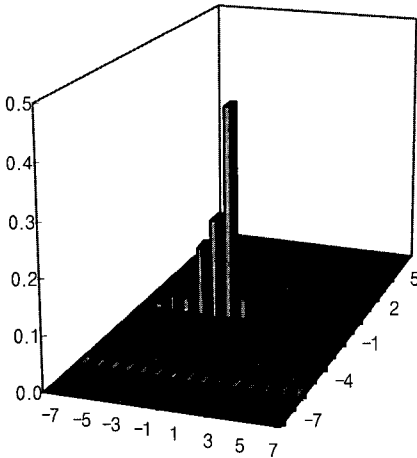
든 후보점을 조사하여 정합에러가 가장 작은 후보점을 이용하여 움직임 벡터를 결정하는 것으로 성능은 가장 우수하나 계산량이 많아 시간이 많이 소요된다. 수행 시간 단축이나 성능 개선을 목표로 3단계탐색(Three Step Search, TSS)<sup>[4] [5]</sup>, 4단계탐색(Four Step Search, FSS)<sup>[9]</sup>, 로그탐색(Logarithmic search, LOG), 평균 피라미드를 이용한 계층적 블록정합<sup>[6] [7]</sup>, 시간·공간적으로 인접한 블록의 움직임벡터를 이용한 방법<sup>[8]</sup> 등과 같이 여러가지 알고리즘들이 제안되고 있다.

알고리즘보다는 수행시간이 짧아지면서 그것에 근접하는 성능을 얻기위한 것들이다. TSS 알고리즘은 탐색점의 수가 적기 때문에 수행 속도는 빠르지만, 전역탐색 알고리즘에 비하여 성능이 크게 떨어진다. TSS 알고리즘은 H.261의 RM8과 MPEG1의 SM3에서 사용되었으나, 정합에러의 지역극소점에 수렴하는 단점이 있다. 그림 1에서 보는 바와같이 대부분의 움직임 방향이 중심에 분포하기 때문에, 중심 영역 부분을 블럭정합을 시도함으로써 성능 개선과 빠른 수행 성능을 얻을 수 있다. 탐색점의 수가 적기 때문에 전역탐색 알고리즘에 비해 성능은 떨어지지만, 실시간 동영상압축을 이용하는데, 큰 장점이 있다. LOG알고리즘이 중심영역을 시도한 방법이 되겠다.

본 논문에서는 실시간 동영상압축 응용에 이용하기 위하여 TSS 그리고 LOG 알고리즘보다 수행시간을 크게 단축시키면서, TSS보다 성능은 개선되고, 다른 알고리즘과 비슷한 성능을 가진 5DS(5-Directional Search)알고리즘을 제안하였다. 2장에서는 제안한 알고리즘을 설명하고, 3장에서는 기존의 다른 알고리즘들과 성능을 비교한 후, 4장에서 결론을 제시하였다.



(a) Table Tennis



(b) Flower garden

그림 1. 전역탐색 알고리즘의 움직임 벡터분포(60프레임)  
Fig. 1. Motion vectors distribution of full search algorithm(60 frames)

## II. 제안된 탐색 알고리즘

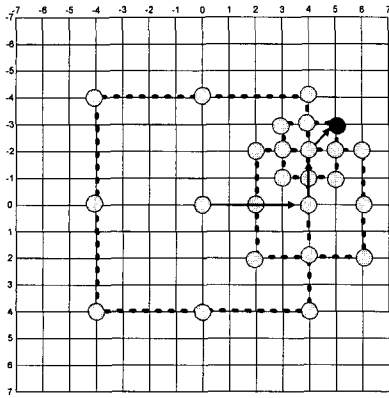
전역탐색 알고리즘은 탐색영역의 모든 후보점들을 대상으로 비교하고자하는 현재 블록과의 정합에러를 계산하여 최소의 에러를 가지는 후보점을 움직임벡터로 결정하는 방법이다. 이 방법은 탐색영역내의 모든 후보점에 대해 정합에러를 계산하므로 현재의 블록과 가장 유사한 블록을 찾아내므로 성능이 가장 우수하지만 탐색영역이 커지면 계산량이 급격하게 증가하는 단점이 있다.

이를 개선하기 위해 제안된 TSS 알고리즘(그림 2-a)은 탐색거리(4)에 위치한 8방향의 최초 후보점들을 선택하여 정합에러를 계산하고 이 중에서 최소의 정합에러를 가지는 후보점을 하나 선택한 다음, 탐색거리를 다시 반으로 줄여 탐색거리가 1이 될때까지 동일한 국부탐색 과정을 반복하여 최종 움직임벡터를 찾아낸다. TSS 알고리즘은 탐색영역내에서 하나의 전역 최소 에러점이 존재하고 이 점을 벗어나면 정합 에러가 단조 증가한다는 가정에 기초하고 있으나, 실제로는 정합에러의 단조증가 가정이 실제 동영상에 완전히 일치하지 않기 때문에, TSS 알고리즘은 지역 극소

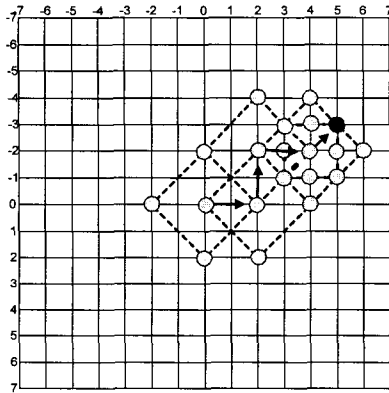
지금까지 제안된 대부분의 알고리즘들은 전역탐색

점에 수렴하는 경우가 있어 전역탐색에 비해 성능이 떨어진다.

선 방향으로 탐색하는데에는 탐색 횟수가 증가되는 단점이 있다.



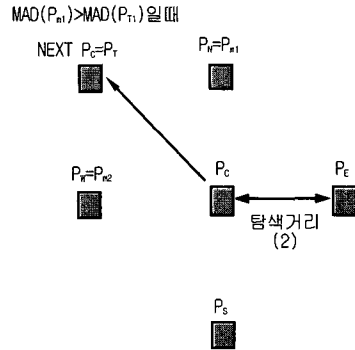
(a)



(b)

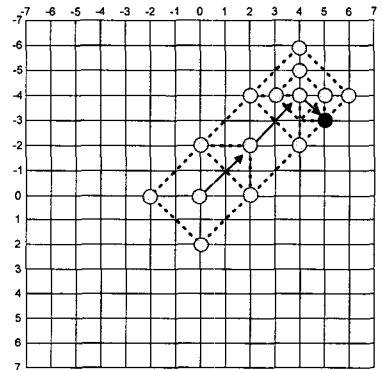
그림 2. 탐색 알고리즘, TSS, LOG (a) TSS 알고리즘 탐색과정 (b) LOG 알고리즘 탐색과정  
Fig. 2. Search algorithms, TSS, LOG (a) Searching procedure of TSS algorithm (b) Searching procedure of LOG algorithm

LOG탐색 알고리즘(그림 2-b)은 움직임 벡터가 중심을 기준으로 대부분 분포한다는 기준으로 탐색거리를 2단계로 시작하는 탐색법을 보인다. 표 1에서 보면, 움직임 방향이 대각방향보다 상·하·좌·우 방향이 분포가 크므로, LOG탐색 알고리즘처럼 중앙점에서 시작하여 탐색거리를 2로하여 상·하·좌·우, 4방향으로 탐색을 반복하다가 마지막 탐색단계에서는 탐색거리 1로 하고 8방향 탐색을 하고 마치는 경우이다. 이 방법은 다른 블록정합 알고리즘에 비하여 단순하며, 단지 정합 시간 단축을 목적으로 고안된 알고리즘이다. 직각 방향인 4방향으로 탐색을 진행하기 때문에 대각



$P_C$  : 중앙탐색점  
 $P_N, P_E, P_S, P_W, \dots, P_{N1}, P_{N2}, P_{E1}, P_{E2}, \dots$  : 8방향 탐색점  
 $P_T$  : 추가탐색점  
 $P_{N1}, P_{N2}$  : 수평·수직방향의 최소 MAD 탐색점

(a)



(b)

그림 3. 제안 알고리즘, 5DS (a) 5DS 알고리즘 탐색 방법 (b) 5DS 알고리즘의 탐색과정  
Fig. 3. Proposed algorithm, 5DS (a) Searching method of 5DS algorithm (b) Searching steps of 5DS algorithm

표 1. 움직임벡터 분포  
Table 1. Motion vector distribution

방향	Table Tennis	Flower Garden
상-하	0.1169	0.0222
좌-우	0.1293	0.8118
우상-좌하	0.0645	0.0042
좌상-우하	0.0567	0.0052
1×1영역	0.3670	0.0428
3×3영역	0.5459	0.4733
5×5영역	0.7413	0.6756

이러한 LOG탐색 알고리즘의 단점을 개선하기 위하여 대각선방향으로 탐색할 수 있는 변형된 알고리즘, 5DS를 제시한다(그림 3). 제안된 방법은 표 1에 나타난 바와 같이 움직임벡터가 중심부분과 대각선방향(NE,NW,SE,SW)보다 상·하·좌·우(N,E,W,S)방향으로 많이 분포한다는 통계적 특성을 이용하여 수평, 수직방향 탐색을 우선하며, 중앙점에 대부분 분포하고 있기 때문에, 탐색거리는 2로하고 탐색 반복회수에 제한을 두지 않는다. 먼저 상·하·좌·우방향 4개의 후보점을 선택하여 정합에러를 계산하고 이들 블록의 상·하·좌·우방향을 4 탐색점들 가운데 가장 작은 정합에러를 갖는 탐색점과 인접방향을 두 탐색점 가운데 작은 MAD값을 갖는 다른 하나의 탐색점을 선택한 다음, 이 두 탐색점 사이(대각선방향)로 움직임벡터를 탐색할 대각방향에 있는 하나의 정합후보점을 추가로 선택하여 정합에러를 계산한다. 지금까지 선택된 5개 탐색점들의 정합에러 가운데 최소가 되는 후보점을 다음 탐색의 중심점으로 지정한다. 탐색과정은 현재 탐색의 중심점이 최소 정합에러점이 되거나, 또는 탐색영역 가장자리에 다다를 때까지 정합을 반복한다. 마지막 탐색단계에서는 탐색거리를 1로 하였다.

제안된 알고리즘, 5DS를 요약하면 다음과 같다.

- 단계 1. 탐색거리를 2로 한다.  
탐색영역의 중앙점(P<sub>C</sub>)과 상하, 좌우의 4 방향 탐색점 P<sub>N</sub>, P<sub>E</sub>, P<sub>S</sub>, P<sub>W</sub>에 대한 MAD를 계산한다.
- 단계 2. 최소 MAD값을 가지는 탐색점(P<sub>m1</sub>)을 결정한다.
- 단계 3. P<sub>m1</sub>과 인접방향을의 두 탐색점 가운데 작은 MAD값의 탐색점(P<sub>m2</sub>)을 선택하고, P<sub>m1</sub>, P<sub>m2</sub> 사이의 사선 방향으로 새로운 탐색점(P<sub>T</sub>)을 설정하여 MAD를 계산한다.
- 단계 4. 만약, MAD(P<sub>C</sub>) ≤ MAD(P<sub>m1</sub>) and MAD(P<sub>C</sub>) ≤ MAD(P<sub>T</sub>)이면 단계 6으로 간다. 그렇지 않으면,  
MAD(P<sub>m1</sub>) ≤ MAD(P<sub>T</sub>)이면 P<sub>C</sub> = P<sub>m1</sub>으로 놓고,  
MAD(P<sub>m1</sub>) > MAD(P<sub>T</sub>)이면 P<sub>C</sub> = P<sub>T</sub>으로 놓는다.
- 단계 5. 만약, 탐색점 P<sub>C</sub>가 탐색영역의 변두리 점이

면 이때의 P<sub>C</sub>를 최종 움직임벡터 후보로 정하고 끝낸다.

그렇지 않으면, P<sub>C</sub>를 새로운 중앙점으로 하고, 다음단계의 4방향 MAD를 구한다.

이때 전 단계 탐색에서 사용하였던 것의 MAD는 중복하여 구할 필요가 없다. 단계 2로 간다.

단계 6. 탐색거리를 1로 하여 단계 1 ~ 4를 수행한 다음, 이때의 P<sub>C</sub>를 최종 움직임벡터 후보로 정하고 끝낸다.

이 알고리즘은 4개의 방향 탐색점들의 정합에러를 이용하여 4방향외에 MAD가 낮은 쪽의 대각선 방향으로 추가로 하나의 탐색점을 선택, 모두 5개의 탐색점을 사용함으로써 움직임 벡터가 존재 가능한 대각선 방향을 예측하고, 그 방향으로 추가적인 탐색을 수행할 수 있도록 함으로써 정합 후보점수를 줄여 수행시간을 단축하는 것이다.

### III. 시뮬레이션 결과

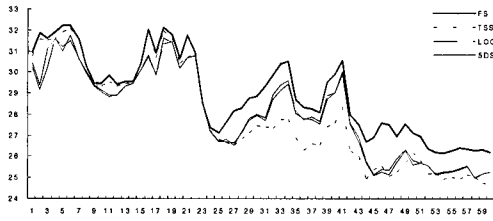
제안된 방법의 성능을 평가하기 위하여 FS, TSS, LOG 알고리즘들과 비교하였다. 실험 영상으로(그림 4) 352×240 SIF 포맷 영상 “Table Tennis”, “Flower Garden” 을 각각 60프레임 이용하였다. 영상의 전체적인 특징을 살펴보면, “Table Tennis” 는 카메라 Zooming과 화면전환이 나타나며 움직임 부분이 한정되어 있고, “Flower Garden” 은 움직임은 단일방향으로 존재하고 고주파 부분을 많이 포함하고 있다. 움직임 벡터를 추정하기 위해 16×16 크기의 블록에 대해 반화소 탐색을 수행하였다. 성능비교를 위한 평가함수로는 PSNR(Peak Signal to Noise Ratio)을 이용하였고 정합에러 측정함수로는 MAD(Mean Absolute Difference)를 이용하였다.

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [S_{t-1}(m, n) - S_t(m, n)]^2}$$

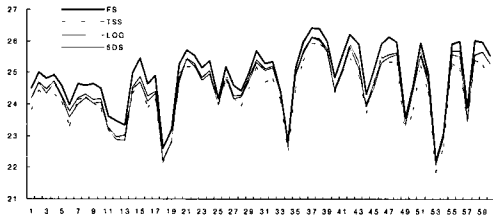
$$MAD(i, j) = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y |S_{t-1}(x+i, y+j) - S_t(i, j)|$$

여기에서 M×N은 영상의 크기이고 S<sub>t</sub>(i,j)는 t번째 영상(i,j)에서의 화소값이고 X×Y는 움직임 추정 영상의 탐색영역 크기를 나타낸다. 움직임 추정방법의

성능을 판단하기 위하여 차영상은 고려하지 않고 움직임 보상된 영상만 평가하였다.



(a) Table Tennis



(b) Flower Garden

그림 4. 테스트 영상 시퀀스에 대한 PSNR(dB)분포. Fig. 4. PSNR(dB) distribution of test image sequences.

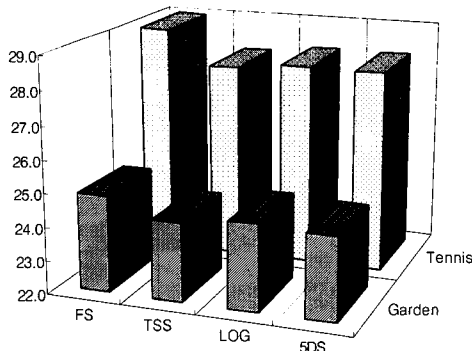
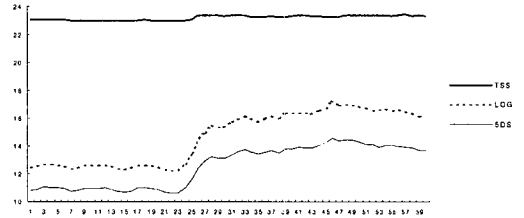


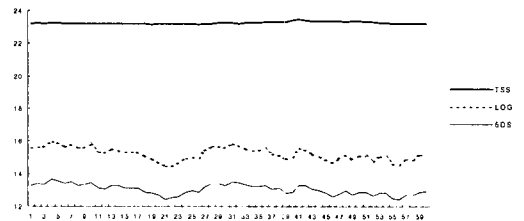
그림 5. 테스트 영상 시퀀스에 대한 평균 PSNR(dB) 분포 Fig. 5. Averaged PSNR distribution of test image sequences

그림 4는 탐색 알고리즘들의 각 프레임에 대한 PSNR의 변화를, 그림 5는 영상시퀀스 전체에 대한 PSNR의 평균을 나타내었다. 그림 6은 탐색 알고리즘의 프레임에 대한 탐색점 수의 변화를, 그림 7은 영상시퀀스 전체에 대한 평균 탐색점 수를 나타내었다. 제안한 5DS탐색방법은 각 영상시퀀스에 대하여 TSS와 LOG탐색방법과 비교하였을 때 평균 PSNR에서는 약간 나아지거나 비슷하면서도 탐색속도(탐색

점의 수)에서는 각각 1.9배, 1.2배 향상됨을 알 수 있다. 이것은 TSS에 대해서는 8방향탐색 대신에 5방향 탐색으로, LOG 탐색에 대하여는 대각선 탐색이 가능하도록 함으로써 탐색단계를 줄임으로써 탐색속도가 개선된 것으로 생각된다.



(a) Table Tennis



(b) Flower Garden

그림 6. 테스트 영상 시퀀스에 대한 탐색점수 분포 Fig. 6. Searching points distribution of test image sequences.

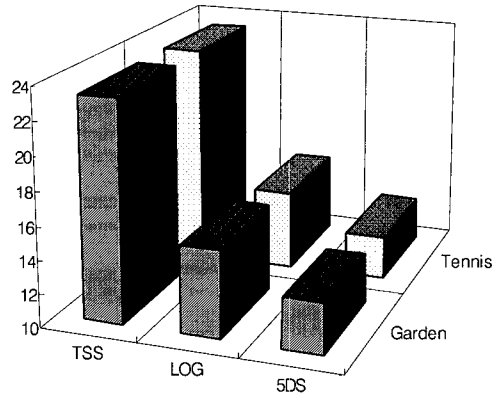


그림 7. 테스트 영상 시퀀스에 대한 평균 탐색점수 분포 Fig. 7. Average searching point number distribution of test image sequences.

#### IV. 결론

본 논문에서는 정합의 속도개선에 중점을 두어 TSS, LOG 알고리즘보다 탐색점수를 크게 줄이면서

성능도 비슷한 탐색 알고리즘, 5DS 알고리즘을 제안하였다. 제안한 5DS 알고리즘은 일반적인 영상시퀀스에서 대부분의 움직임벡터들이 탐색영역 중심점을 기준으로 사선 방향보다는 상·하·좌·우 방향에 많이 분포한다는 통계적 사실을 기반으로 하여 상·하·좌·우 방향으로 4개의 후보점을 선택하여 검사를 수행하고, 이미 구하여진 상·하·좌·우 방향 후보점의 정합에러를 이용하여 1개의 탐색점을 추가로 선택하여, 대각선 방향의 움직임을 탐색하였다. 이 방법을 이용하면 대각선방향으로 탐색할 수 있도록 5개의 탐색점들을 시용함으로써 탐색점의 수를 크게 줄이면서도 우수한 성능을 나타냄을 확인하였다.

### 참 고 문 헌

- [1] H.261 ITU-T Recommendation H.261, "Video codec for audiovisual services at p×64 kbit/s" Mar.1993.
- [2] MPEG-1 ISO/IEC 11172-2, "Information Technology-Coding of Moving Picture and Associated Audio for Digital Storage Mediaa at up to about 1.5mbit/s : Part 2 Video," Aug. 1993.
- [3] MPEG-2 ISO/IEC 13818-2, "Information Technology-Generic coding of Moving Picture and Associated Audio Information : Video," Darft, Mar. 1994.
- [4] T.Kog A,K.Iinuma,A.Hirano,"Motion Compensated Inter Frame Coding for Video Conference," Proc. NTC, 1981, pp. G5.3.1-5.3.5.
- [5] M. Ghanbari, "The corss-search algorithm for motion estimation," IEEE Trans. Commun., vol. 38, 1990, pp. 950-953.
- [6] K.M.Nam,R.H.Park, "A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid," IEEE Trans. Circuits and systems for video tech., vol. 5, No. 4, 1995, pp. 344-351.
- [7] K.W.Chun and J.B.Ra,"Fast block-matching algorithm by successive refinement of matching criterion," in Proc. SPIE Conf. Visual Commun. Image Process., vol. 1818, 1992, pp. 552-560.
- [8] Junavit chalidabhongse, "Fast Motion Vector Estimation for Video Coding Based on Multerresolution-Spatio-Temporal Correlations," SPIE, vol. 2727, pp. 645-656.
- [9] Lai-Man Po,Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," IEEE Trans. circuits and systems for video tech., vol. 6, No. 3. 1996, pp. 313-317.

### 저 자 소 개



李 根 泳(正會員)

1947년 12월 30일 출생. 1966년 3월 ~ 1973년 2월 全南大學校 電氣工學 工學士. 1973년 3월 ~ 1975년 2월 漢陽大學校 電氣工學 工學碩士. 1975년 3월 ~ 1978년 8월 漢陽大學校 電氣工學 工學博士. 1975년 3월 ~ 1977년 2월 漢陽大學校 講師. 1977년 3월 ~ 1981년 8월 光雲大學校 傳任講師, 助教授. 1979년 3월 ~ 1980년 2월 덴마크 工大 客員教授. 1987년 9월 ~ 1988년 8월 러프버러大(英) 客員教授. 1981년 9월 ~ 現在 成均館大學校 助教授, 副教授, 教授. 1995년 8월 ~ 1997년 7월 附設 情報通信技術研究所 所長. 1996년 3월 ~ 1998년 1월 附設 情報通信大學院 院長. 1998년 3월 현재 전기전자컴퓨터 공학부장