

論文2001-38SP-11-11

슬라이스 경쟁 방식을 이용한 고속 블럭 정합 알고리즘

(A fast block-matching algorithm using the slice-competition method)

鄭暎勳*, 金在浩**

(Young Hoon Jeong and Jae Ho Kim)

요 약

본 논문에서는 표준화 영상부호화기에서 사용할 수 있는 새로운 블럭정합 방식을 제안하였다. 제안방식은 기존 방식처럼 광역-협역 탐색방식을 사용하는 방식이 아닌, 미리 선정된 각 후보에 대하여 SAD 누적추이를 사용하여 움직임 벡터를 찾는 방식이다. 확산누적배열(dispersed accumulation matrix)의 제안으로 SAD 추이곡선의 선형성(linearity)을 높여서 누적 초기에 가능성 낮은 후보를 제거하였다. 이를 토대로 고속 슬라이스 경쟁 블럭정합 방식을 제안하였다. 후보선출단계와 후보경쟁단계를 통하여 최종 움직임 벡터를 생성하였으며, 극부최소값에 빠질 확률을 줄이는 동시에 불필요한 연산을 감소시켰다. 기존의 고속 블럭정합 방법들에 비하여 연산량은 10%~70%가 줄었으며, 18%~35%의 실제 수행시간이 감소되었다. 평균 MAD는 다양한 테스트영상에 대하여 항상 우수하였으며, 전역탐색의 결과에 가장 가까운 결과를 나타내었다.

Abstract

In this paper, a new block-matching algorithm for standard video encoder is proposed. The algorithm finds a motion vector using the increasing SAD transition curve for each predefined candidates, not a coarse-to-fine approach as a conventional method. To remove low-probability candidates at the early stage of accumulation, a dispersed accumulation matrix is also proposed. This matrix guarantees high-linearity to the SAD transition curve. Therefore, base on this method, we present a new fast block-matching algorithm with the slice competition technique. The Candidate Selection Step and the Candidate Competition Step makes an out-performance model that considerably reduces computational power and not to be trapped into local minima. The computational power is reduced by 10%~70% than that of the conventional BMAs. Regarding computational time, an 18%~35% reduction was achieved by the proposed algorithm. Finally, the average MAD is always low in various bit-streams. The results were also very similar to the MAD of the full search block-matching algorithm.

I. 서론

* 正會員, (株)三星電子 디지털미디어 總括中央研究所
(Samsung electronics Co. Ltd.)

** 正會員, 釜山大學校 컴퓨터 및 情報通信研究所
(Research Institute of Computer, Information and
Communication, Pusan university)

接受日字:2001年1月17日, 수정완료일:2001年7月9日

입력영상의 연속적인 프레임사이의 높은 시간상 유사성으로 인하여 움직임 보상(motion compensation)은 비디오 압축분야에서 오랜 동안 널리 사용되어져 왔다. 움직임 보상은 시간상 인접한 두 프레임간에 존재하는 정보의 중복성을 억제하는 방법으로서 두 단계로 나누어진다. 먼저 움직임 추정(motion estimation)을 하여

두 프레임간의 움직임 정보를 생성하고, 현재 프레임과 움직임 예측 프레임(motion-predicted frame)사이의 차등정보를 부호화하는 오차 예측 코딩(prediction error coding)으로 나누어진다.

움직임 정보를 파악하는 다양한 방법 중, 블럭정합 방법(Block Matching Algorithm: BMA)은 블럭기반 움직임 예측과 보상(block-based motion estimation and compensation) 방법을 사용하므로 가장 낮은 복잡도를 가진다. 따라서 디지털 비디오 압축의 표준인 H261/263과 MPEG-1/2/4 등에서 채택되어 널리 사용되고 있다^[1-4].

블럭정합 방법 중 전역탐색 블럭정합 방식(full search BMA)은 탐색영역의 모든 위치를 검색하므로 가장 좋은 성능을 가지고 있지만, H261과 MPEG-1비디오 부호화기 전체 연산량의 60~70%를 차지하므로 하드웨어/소프트웨어 구현시 문제점을 안고 있다^[5].

움직임 추정의 정확도 평가를 위해 일반적으로 MSE(mean square error), MAD(mean absolute difference), MPC(matching pel count)의 3가지 척도(criteria)가 사용되며, 이외에 cross-correlation function (CCF)과 minimized maximum error (MiniMax)가 있다^[6]. CCF와 MSE는 곱셈(multiplication)과 덧셈(accumulation)연산이 필요하며, 나머지 척도들은 비교(comparison)와 덧셈(accumulation)연산이 필요하다. 따라서 하드웨어 복잡도를 고려하여 곱셈기보다는 비교기를 사용하는 척도가 유리할 것이다. MAD는 MSE와 비슷한 성능을 가지면서 낮은 복잡도로 인해 널리 사용되어진다. 본 논문에서는 MAD와 SAD(Sum of Absolute Difference)를 사용하였다.

II. 블럭 정합 방식

블럭정합 방식은 두 프레임사이의 시간적 상관성과 프레임 내 블럭간의 공간적 상관성을 동시에 이용한다. 일반적으로 블럭은 $M \times N$ 화소를 기준으로 정합이 이루어진다. 그리고 현재 프레임내의 블럭과 이전 프레임내의 동일한 크기의 블럭중 가장 유사한 블럭을 찾을 때, 검색시간과 연산량을 줄이기 위하여 현재 프레임 주변의 최대 탐색 영역 $(M(N) + 2w + 1)$ 내의 영역만 탐색하도록 제한을 둔다. 그리고, 정합 척도를 사용하여, 최대 탐색 영역내의 블럭 중 가장 유사한 블럭의 상대적인 위치를 움직임 벡터(motion vector)로 설정한다.

시간 t 에서 현재 프레임의 (k, l) 좌표에 위치하고 있는 블럭과 $t-1$ 시간에서 이전 프레임의 (x, y) 변위로 이동한 좌표 $(k+x, l+y)$ 의 블럭간의 SAD는

$$SAD_{(k,l)}(x,y) = \sum_{i=1}^M \sum_{j=1}^N | I_t(i+k, j+l) - I_{t-1}(i+k+x, j+l+y) | \quad (1)$$

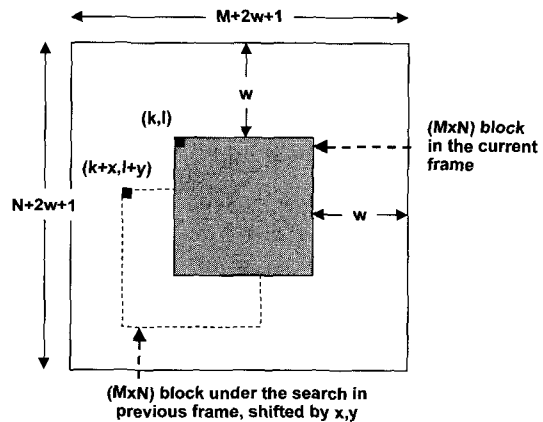


그림 1. 탐색영역내에서의 이전/현재 프레임
Fig. 1. The current and previous frames in a search range.

이다. M, N 은 블럭의 가로와 세로의 화소 수이며, $I_t(k, l)$ 은 시간 t 에서의 프레임 내의 좌표 (k, l) 의 밝기 값이다. 블럭 (k, l) 의 움직임 벡터 $mv(k, l)$ 는 식 (2)에 나타나있다.

$$mv(k, l) = \underset{(x,y)}{\arg \min} SAD_{(k,l)}(x,y) \quad (2)$$

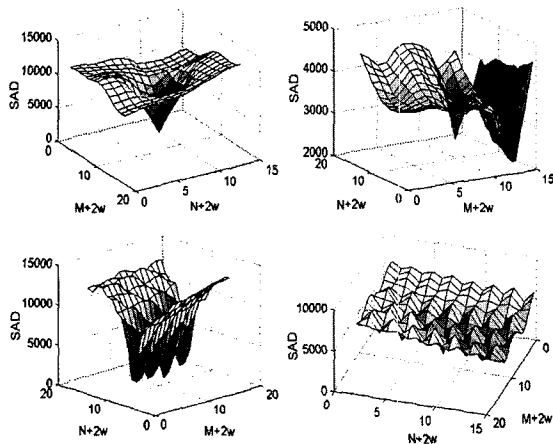
이전 프레임 내 임의의 SAD를 연산하기 위하여 $2MN$ 의 가산과 감산이 필요하다. 만약 연산량을 줄이기 위해 탐색영역을 w 로 제한한다면, 현재 프레임의 고정 블럭과 탐색 영역 내의 탐색해야 될 총 위치는 $(2w+1)^2$ 이 된다. 만약 블럭의 크기가 정방향 16의 크기이고, 탐색 영역 w 를 7이라고 가정한다면, 이론적으로 총 57600번의 가산과 감산이 필요하다. 이러한 전역탐색 블럭정합 방법은 움직임 벡터의 SAD값이 가장 낮으므로 SAD측면에서는 이상적인 탐색 방식이 된다. 그러나 이 방법은 많은 연산량이 소모되므로, 연산량을 감소시키는 동시에 전역탐색 블럭정합 방식의 성능에 가까운 고속 탐색 방식(fast (search) BMA)들이 제안이 되었다.

고속 탐색 방식들은 넓은 영역에 몇 개의 탐색점을

선정하고 추상적인 형태를 파악한 후, 탐색 영역을 좁혀 나가는 광역-협역 탐색(coarse-to-fine search) 방식을 따른다. 따라서 연산량은 급격히 줄어들지만, 탐색영역내의 SAD값이 균일하게 넓은 영역에 걸쳐서 단조감소하지 않으면 국부 최소값(local minimum)에 빠지기 쉬운 단점을 가진다. 그러므로 움직임 벡터의 SAD는 전역탐색 블럭정합 방식의 SAD보다 높아지게 된다.

그림 2는 여러 형태의 SAD (오차) 곡면의 3차원 그래프이다. 그림 2-(a)에서처럼 SAD 곡면 기울기가 단조감소인 경우, 대부분의 고속 탐색 정합 방식들은 정확한 움직임 벡터를 찾는다. 그러나 그림 2-(b),(c),(d)와 같은 경우는 국부 최소값에 빠지기 쉽다.

널리 알려진 고속탐색 블럭정합 방식들은 TSS (Three Step Search)^[7], 2DLOGS(2-D LOGarithmic Search)^[8], NTSS(New Three Step Search)^[9], FSS(Four Step Search)^[10], BBGDS(Block Based Gradient Descent Search)^[11], OSA(Orthogonal Search Algorithm), DS(Diamond Search)^[12], 그리고 HBMA(Hierarchical Block Matching Algorithm)등이 있다. 그리고 주변 블럭의 움직임 벡터나 이전 또는 다음 프레임의 움직임 벡터를 사용하여 현 블럭의 움직임 벡터를 예측을 하는 방식이 있다^[2]. 이러한 방식은 기본적인 방식에 부가적으로 사용될 수 있으므로 본 논문에서는 생략하였다.



(a) 단일 급경사 계곡 (b) 단일 계곡과 단일 협곡
 (a) single deep valley (b) single valley and canyon
 (c) 복합 계곡 (d) 평탄 영역내의 좁은 계곡
 (c) multiple valleys (d) narrow va- in flat region
 그림 2. SAD분포의 여러 형태

Fig. 2. Several types of SAD distribution.

III. 제안 방식

본 논문에서는 확산누적방식을 제안하여 각 후보들간의 SAD증가 추이를 선형적으로 증가시키면서 중도(halfway)에 높은 SAD값을 가지는 후보를 탈락시키고 최종적으로 낮은 SAD값을 가지는 후보의 추이를 지속적으로 관찰하여, 각 후보를 중도-정지-탈락(halfway-stop-reject)시키는 고속 블럭정합 방식을 제안하였다.

1. 확산누적방식

그림 3은 탐색영역내의 모든 후보들의 SAD추이를 나타낸 그림이다. 각각의 후보들은 식 (1)을 사용하여 MN 번의 SAD를 누적하였다(x 축). y 축 256에 위치한 225개의 값들이 최종 SAD값들이며, 그림 2에서 보여지는 곡면의 표면값들이 된다. 참고로, 기존의 블럭정합 방식들은 항상 최종 SAD값을 비교하여 가장 낮은 값을 찾는다. 그림 3에서 최적 움직임 벡터의 SAD값은 972가 되어야 한다. 그림 3에서 각 후보들의 추이는 급변하고, 서로 교차되면서 선형성을 확보하지 못하였다. 이러한 환경하에서는 누적 초기단계에서 부적절한 후보들을 제거할 정확한 기준을 찾기 어려울 것이다.

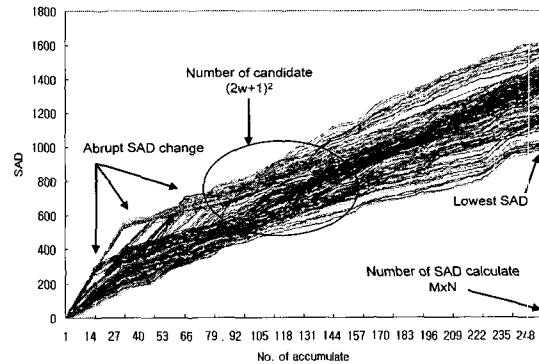


그림 3. 탐색영역 내 모든 후보들의 SAD 추이
 Fig. 3. SAD transition of all candidates.

SAD 추이의 선형성이 확보되지 못하는 경우는 두 프레임사이의 동일 위치의 화소값들의 차이의 증감이 불규칙적으로 발생하는 경우이며, 급격한 움직임으로 인한 선형성의 감소를 방지하기 위하여, SAD의 누적시 왼쪽 상단에서 오른쪽 하단으로의 일반적인 순차누적 방식(sequential accumulation method)이 아닌 확산누적방식(dispersed accumulation method)을 제안하였다.

확산누적방식은 이진화(halftoning)에서 널리 사용되는 베이어디더 행렬(Bayer dither matrix)을 사용한다^[13]. 이 행렬은 정방형태로 구성이 되어 있으며, 행렬 내의 연속되어지는 값들은 멀리 배치되어 있다(표 1). 이 값들은 누적순서를 정할 때 사용되어진다.

표 1. 베이어 디더 행렬을 이용한 누적 순서
Table 1. The accumulation order of Bayer dither matrix.

i	$x_{mask}(i), y_{mask}(i)$	i	$x_{mask}(i), y_{mask}(i)$	i	$x_{mask}(i), y_{mask}(i)$	i	$x_{mask}(i), y_{mask}(i)$
1	0, 0	5	1, 1	9	2, 1	13	1, 2
2	3, 3	6	4, 4	10	4, 3	14	3, 4
3	3, 1	7	4, 2	11	4, 1	15	3, 2
4	1, 3	8	2, 4	12	2, 3	16	1, 4

확산누적방식의 MAD는 식 (6)으로 표현된다. s 는 현 슬라이스(slice)의 위치이며, S 는 총 슬라이스의 수이다. M_{mask}, N_{mask} 는 베이어 디더행렬의 가로, 세로 크기가 된다. 본 제안방식에서 슬라이스는 각 후보간 SAD값을 동일한 누적 횟수에서 비교하기 위해 도입하였으며, 두 프레임사이에서 각각 대응되는 M 개의 화소쌍의 차이가 SAD에 누적되었다면 슬라이스는 1이 증가된다.

$$SAD_{(k, \delta)}(x, y, s) = \sum_{s_{cur}=1}^s |I_t(i_{mask} + k, j_{mask} + \delta) - I_{t-1}(i_{mask} + k + x, j_{mask} + \delta + y)| \quad (3)$$

식(3)에서 x 축 상대좌표 i_{mask} , y 축 상대좌표 j_{mask} 는 다음의 식으로 정의되어진다.

$$i_{mask} = \frac{M}{M_{mask}} \sum_{d_{cur}=1}^S x_{mask}(d_{cur}) + x_{mask}(s_{cur}) \quad (4)$$

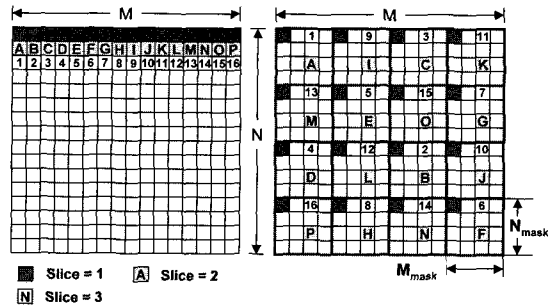
$$j_{mask} = \frac{N}{N_{mask}} \sum_{d_{cur}=1}^S y_{mask}(d_{cur}) + y_{mask}(s_{cur})$$

$$S = M_{mask} \times N_{mask} \quad (s = 1, 2, \dots, S) \quad (5)$$

따라서, $x_{mask}(s_{cur}), y_{mask}(s_{cur})$ 은 4×4 마스크 내의 화소 위치, $x_{mask}(d_{cur}), y_{mask}(d_{cur})$ 은 $M \times N$ 내 4×4 마스크 위치이며, 최종 MAD $MAD_{(k, \delta)}(x, y)$ 는 다음과 같이 정의되어진다.

$$MAD_{(k, \delta)}(x, y) = \frac{1}{MN} SAD_{(k, \delta)}(x, y, S) \quad (6)$$

그림 4는 슬라이스 3, 즉 두 프레임간 $3 \times M$ 의 화소를 비교하여 SAD를 누적시켰을 경우, 두 방식의 화소 참조위치의 차이를 보여주고 있다. 슬라이스 1과 관련되는 화소들은 회색(■), 슬라이스 2는 알파벳(A), 그리고 슬라이스 3에 관련되는 화소들은 숫자(N)로 표시하였다. 순차누적방식에서는 현재 프레임 내 임의의 위치로 물체가 이동을 하였을 경우, 급격한 SAD의 변화가 생긴다(그림 3 참조). 그러나 제안하는 방식은 2차원 공간상 균일하게 확산된 샘플링방식이므로, 이전 방식의 급격한 SAD변화들은 이중 베이어디더의 순서에 따라 누적추이곡선의 전반에 일정간격으로 분할되어져 기존 방법보다 높은 선형성을 확보할 수 있다. 부연하면, 이중 베이어디더에 의해 정해진 순서는 이론상 2차원 공간상에서 가장 넓게 분포되어지는 패턴이므로, 가우시안 분포를 지니는 일반 영상신호나 급격한 변화도 추이곡선에 골고루 분할되므로 기존 방법보다 높은 선형성을 유지할 수 있다.



(a) 순차누적 (b) 확산누적
(a) sequential accumulation (b) dispersed acc-

그림 4. 두 방식에서의 참조 위치의 예 (~Slice3).

Fig. 4. The example of referenced coordinate in two methods (~Slice3).

그림 3과 동일한 블럭에 대하여 제안하는 방식을 사용한 결과를 그림 5에 나타내었다. 두 그림에서 각 후보들의 최종 SAD값들은 동일함을 알 수 있다. 추이곡선들은 초기 0에서부터 최종 SAD까지 급격한 변화 없이 선형적으로 증가하였다.

본 논문에서는 M, N 은 16, M_{mask}, N_{mask} 는 4를 사용하였으므로 식 (5)에 의하여 총 16개의 슬라이스를 가진다. 블럭의 크기가 2^n 이라면 곱셈기를 사용하지 않고 이동연산만으로 참조위치가 정해진다.

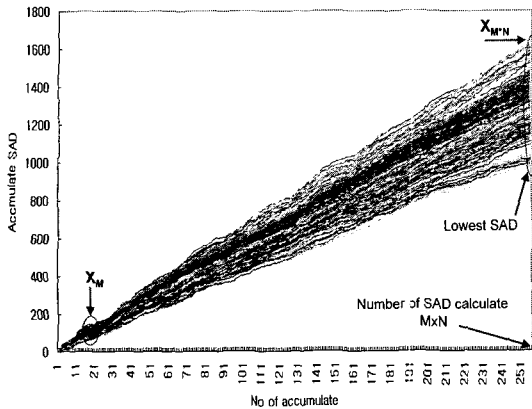


그림 5. 제안하는 방식을 적용후의 SAD추이
Fig. 5. The SAD transition after proposed method.

확산누적방식의 타당성을 검증하기 위하여 각 슬라이스별로 SAD의 추이와 SAD_{MIN} (후보들 중 SAD의 최소값)의 변화율과 각 슬라이스사이의 이동편위, 상관도등을 분석하였다.

표 2는 초기 슬라이스 Slice1과 최종 슬라이스 Slice16에서 SAD_{MIN} 값을 가지는 두 움직임 벡터 사이의 유클리디안 거리를 나타낸 것이다.(식(7))

$$Dist = \sqrt{(x_{Slice16} - x_{Slice1})^2 + (y_{Slice16} - y_{Slice1})^2} \quad (7)$$

표 2. 초기/최종 슬라이스 움직임벡터의 거리
Table 2. The distance of motion vector btw. the first and last slice.

- FG: Flower Garden- FB2: Football2
- MB: Mobile- TT: Table tennis

Dist	percentage (%)				Dist	percentage (%)			
	FG	FB2	MB	TT		FG	FB2	MB	TT
0	71.8	38.3	68.9	68.1	11	0.59	1.75	0.53	0.74
1	5.56	6.08	8.82	4.65	12	0.71	2.03	0.48	0.73
2	3.13	4.61	4.55	3.37	13	0.51	1.73	0.39	0.63
3	2.12	3.31	2.92	2.32	14	0.60	1.61	0.37	0.74
4	1.90	3.25	2.27	2.00	15	0.49	1.44	0.26	0.63
5	1.84	2.85	2.14	2.05	16	0.31	1.20	0.22	0.39
6	1.41	2.70	1.53	1.83	17	0.33	0.98	0.17	0.38
7	1.23	2.40	1.43	1.85	18	0.23	0.89	0.19	0.32
8	1.18	2.28	1.09	1.42	19	0.26	0.82	0.08	0.32
9	0.93	1.92	0.88	1.08	20	0.23	0.79	0.14	0.28
10	0.69	1.74	0.72	0.93	-	-	-	-	-

총 131,340블럭이 사용되었으며, 각 거리에 대한 비율을 산출하였다. 평균적으로 62%의 블럭에서 초기 슬라이스의 움직임 벡터의 위치가 최종 슬라이스와 동일하다는 결과를 보여준다. 부연하면, 그림 5의 X_M 지점

(전역탐색 SAD 연산량의 1/16)에서의 최소값을 가지는 후보가 X_{M*N} 지점에서도 최소값을 가질 확률이 평균 62%정도 되는 것을 의미한다. 그리고 상관도등을 분석한 결과 X_M 이상(Slice2이상)의 지점에서는 표 2의 확률보다 월등히 높아짐을 파악할 수 있었다. 예를 들면, Slice2와 Slice16간은 78.3%이며, Slice8과 Slice16간은 90.9%, Slice14와 Slice16간은 95.2%의 블럭에서 SAD_{MIN} 을 가지는 후보와 최종 슬라이스(Slice16)의 후보의 위치와 동일하였다.

2. 고속 슬라이스 경쟁 블럭정합 방식

3.1절에 언급되어진 확산누적방식을 기반으로 고속 블럭정합 방식을 제안한다.

슬라이스 구조(그림 6-(a))와 Slice1, Slice8, Slice16의 SAD 분포(그림 6-(b)) 그리고 SAD 추이 곡선과 연관성(그림 6-(c))을 나타내었다. 하단부 (c)에서 점선으로 표시되어있는 세 부분의 후보들은 (b)의 Slice1, Slice8, Slice16의 SAD 표면값들과 관련이 있다. 그림 6-(b)에서 Slice8과 Slice16의 SAD분포가 거의 동일하다. 이는 SAD의 크기에 따른 후보의 순서가 추이곡선상의 대응되는 두 위치에서 거의 동일한 순서가 된다는 것을 의미한다. 그림 6-(b),(c)에서는 전역탐색 블럭정합 방식처럼 각 슬라이스의 모든 후보들의 SAD값을 나타내었다.

먼저 이해를 돕기 위해 제안 방식에 대하여 간단히 설명하면, 제안하는 방식은 크게 나누어서 후보선출단계와 후보경쟁단계로 나누어진다(그림 7).

※후보 선출 단계

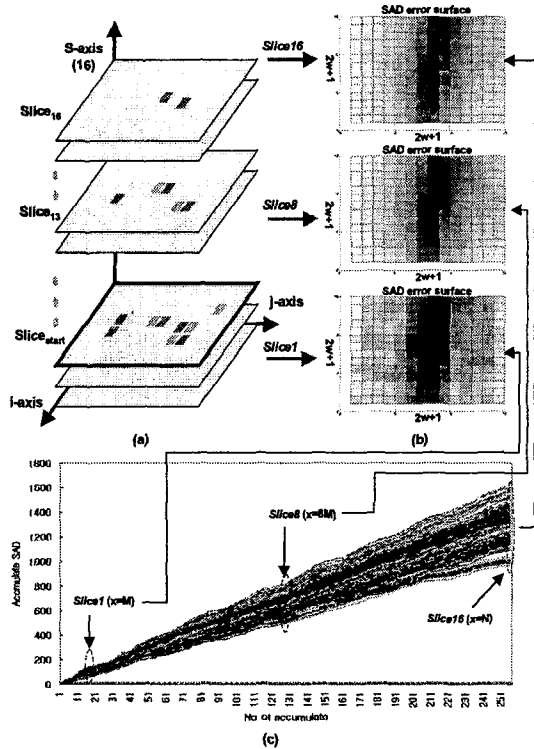
(candidate selection step: CCS)

: 계산량이 적게 필요한 슬라이스($Slice_{START}$)에서 적응적 초기후보모델과 3단계의 후보 선출 단계를 수행하여, 탐색영역 내 전반적인 SAD의 분포를 파악하고 가능성 있는 후보들을 선출한다.

※후보 경쟁 단계

(candidate competition step: CCS)

: 슬라이스가 증가될수록 최상단의 슬라이스와 유사성(선형성)이 높으므로, $Slice_{START}$ 에서 선출된 후보 주위를 검색하여 가능성 낮은 후보들을 제거시킨다. 슬라이스를 증가시키며 동일한 과정을 반복한다. 그리고 마



(a) 기본 구조 (a) basic structure
 (b) 슬라이스 1,-8,-16의 SAD분포 (b) SAD distribution of Slice1,-8,-16
 (c) SAD 추이곡선에서의 슬라이스 1,-8,-16 (c) Slice1,-8,-16 in SAD transition curves

그림 6. 기본구조와 슬라이스간의 상관성.
 Fig. 6. Basic structure and correlation between slices.

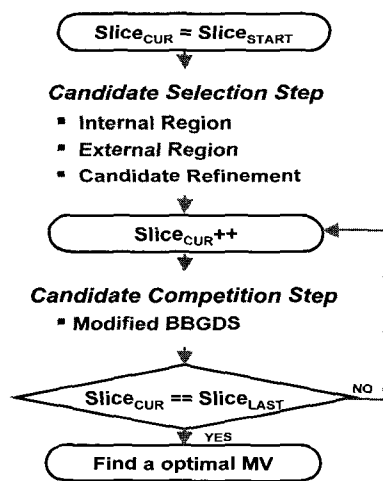


그림 7. 제안방식의 전체 흐름도.
 Fig. 7. The flowchart of the proposed algorithm.

지막 슬라이스에 남아있는 후보 중 가장 낮은 SAD값의 후보가 움직임 벡터가 된다.

상기 두 단계는 슬라이스간의 높은 상관성을 가지는 SAD분포를 토대로 진행되어진다.

1) 후보선출단계(CCS)

첫 번째 단계의 목표는 가능성 높은 후보들을 선출이다. 먼저 초기 후보모델과 항상 수반되는 중도-정지-제거(Halfway-Stop-Reject)방식을 언급한 후 이 방식들을 사용하는 본 단계에 대한 설명으로 전개된다.

(1) 초기후보모델(initial candidate model)

본 논문에서는 국부최소값에 빠지지 않기 위하여 많은 후보를 균등간격으로 고정 배치하여, 전반적인 분포를 파악한 후, 적절한 후보들을 추출하고 그 주변만 검색하는 방식을 사용하였다. 기본적으로 슬라이스내의 총 후보 $(2w+1)^2$ 중 3×3 블록당 한 후보를 할당하였다. 그리고 움직임 벡터의 60~70%이상이 집중되어있는 중심부 $(-1 \leq x \leq +1, -1 \leq y \leq +1)$ 에 후보 8개를 추가하여 두 그룹(기본그룹, 확장그룹)으로 나누어져 있는 초기후보모델(initial candidate model)을 정하였다. 기본그룹은 후보0에서 후보 20까지, 확장그룹은 후보21에서 32까지이다.

표 3. 기본 그룹의 추가 후보

Table 3. The additional candidates of basic group.

index	additional candidate	index	additional candidate
13	21,22,23	17	21,22,23,30,31,32
14	24,25,26	18	24,25,26,27,28,29
15	27,28,29	19	21,22,23,27,28,29
16	30,31,32	20	24,25,26,30,31,32

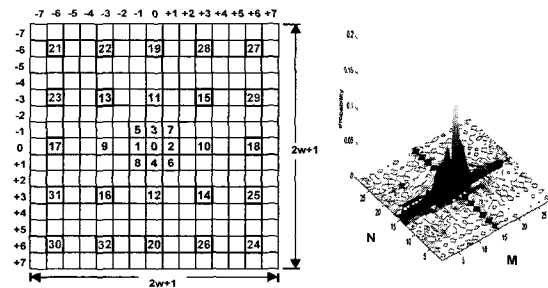


그림 8. 초기후보모델과 움직임 벡터의 분포 분포 $(M, N=16, w=7)$

Fig. 8. Initial candidate model and MV distribution. $(M, N=16, w=7)$.

표준 테스트영상들을 사용하여 움직임 벡터의 분포를 그림 8에 나타내었다. 움직임 벡터는 대부분 중심부에 있으며, 카메라 이동에 의해 대각선보다는 축상에 분포할 확률이 높다. 따라서 기본그룹의 형태를 마름모 형태로 후보들을 배치하였다. 기본그룹에서 후보 선별 과정이 종결되고, 만약 선별된 후보가 후보13~20 사이에 존재한다면 표 3의 각 후보들을 추가시킨다. 이 방식은 불필요한 계산을 줄이기 위함이다. 움직임 벡터가 대부분 중심부에 존재하므로 기본그룹을 수행하고 난 뒤, 후보의 분포를 파악하여 적응적으로 주변의 후보들을 추가하는 방식을 채택하였다. 이 방식은 33개의 후보에 대해 일괄적으로 수행하는 방식보다 평균 MAD를 0.03% 증가시켰지만, 연산량은 20.66% 감소하였다.

(2) 중도-정지-제거 방법

(Halfway-Stop-Reject method: HSR)

그림 9는 후보선출단계의 각 단계에 포함되어 있는 중도-정지-제거 블럭의 자세한 흐름도이다.

미리 정해진 위치의 후보들에 대하여 식 (3)을 적용하여 SAD를 증가시키면서, SAD_{CUR} 가 기존의 절대임계치 (Th_{ABS} : ABSolute Threshold) 이상이면, 누적 중지 후, 현 후보를 탈락시킨다. 그렇지 않으면, SAD_{MIN} 과 비교하여 SAD_{MIN} 의 갱신 유무를 판단한다.

$$Th_{ABS} = P_{ABS} \times SAD_{MIN} \quad (8)$$

정해진 후보들에 대하여 상기의 과정이 종결되면, 탈락되지 않고 남은 후보 중 최대값인 SAD_{MAX} 와 상대임계치 (Th_{REL} : RELative Threshold)를 구한다. 그리고 상대임계치 이상의 후보들을 제거시킨다(그림 9). 확산 누적방식의 높은 선형성은 각 후보들의 SAD 증가추이를 일정하게 하므로 고정적인 임계값으로 가능성 낮은 후보를 제거하는 방식은 적절할 것이다.

$$Th_{REL} = P_{REL} \times (SAD_{MAX} + SAD_{MIN}) \quad (9)$$

(3) 후보 선출(candidate selection)

현재 슬라이스 $Slice_{CUR}$ 는 $Slice_{START}$ 로 고정되어 있다. 앞에서 언급한 초기후보모델과 HSR방법을 사용하여 후보를 선출한다.

1단계 : 내부영역 후보선출

(그림 10: internal region)

초기후보 모델에서 언급된 기본 모델의 21개 후보에 대하여 순차적으로 HSR방식을 사용하여, 낮은 SAD값을 가지는 후보들을 선출한다. 가능성 낮은 후보의 빠른 제거를 위하여, SAD_{MIN} 값을 가지는 후보에 대하여 후보정제과정을 수행한다.

2단계 : 외부영역 후보선출

(그림 10: external region)

남아있는 후보가 후보13~20에 포함 되어있으면 표 3을 참조하여 주변의 후보를 추가시킨다. 추가한 후 동일한 과정을 반복하여 남아있는 후보들을 갱신한다. 만약 그림 2-(a)처럼 낮은 SAD값을 가지는 후보들이 중심에 분포할 경우 2단계는 수행되지 않는다.

3단계 : 후보 정제 과정

(그림 10: candidate refinement)

만약 후보가 경계 영역에 존재하지 않거나 두 번째 단계가 종결되면 후보 정제(refinement)과정을 거친다. 이 과정은 남아있는 각 후보의 주위 ($-1 \leq x(y) \leq +1$)를 검색하여 보다 낮은 SAD값을 가진 후보들을 찾는 과정이다. 즉, 후보선출과정의 1단계와 2단계는 탐색영역내의 전반적인 SAD의 분포를 파악하는 것이며, 정제 과정은 움직임 벡터로서 가능성이 높은 후보 주위를 검색하여 주변에 더 낮은 SAD값을 가지는 후보의 유무를 조사하는 단계이다. 만약 중심점보다 더 낮은 SAD가 주변에 존재하면 후보를 갱신한다.

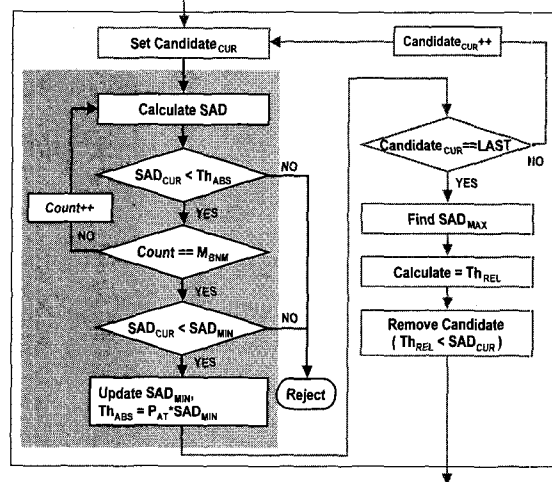


그림 9. 중도-정지-제거 방식의 블럭도

Fig. 9. The block diagram of halfway-stop-reject method.

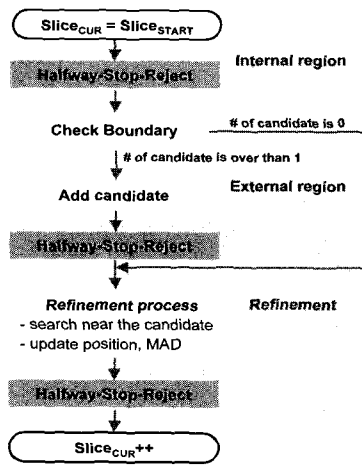


그림 10. 후보 선출 단계의 블록도
Fig. 10. The block diagram of candidate selection step.

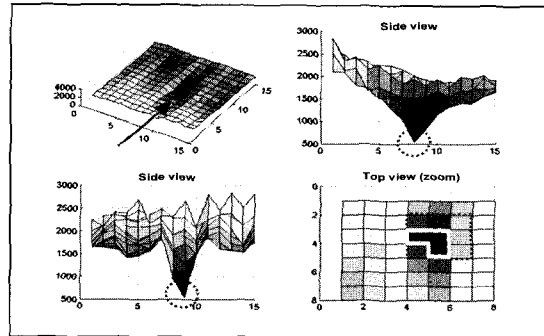
2) 후보 경쟁 단계(CCS)

후보선출단계가 $Slice_{START}$ 에서 끝나면 후보경쟁단계가 최종슬라이스인 $Slice_{LAST}$ 까지 진행된다. 동일한 과정이 각 슬라이스 별로 수행된다. 슬라이스 상관도가 높으므로, 슬라이스 축(그림 6-(a)의 s -axis)상을 기준으로, 선출된 후보 근처만 탐색하기 위해 초기후보모델을 BBGDS(block based gradient descent search)방식과 동일한 3×3 블록과 이동방식을 채택하였다. HSR 방식을 사용하여 가능성 낮은 후보들을 제거시킨다.

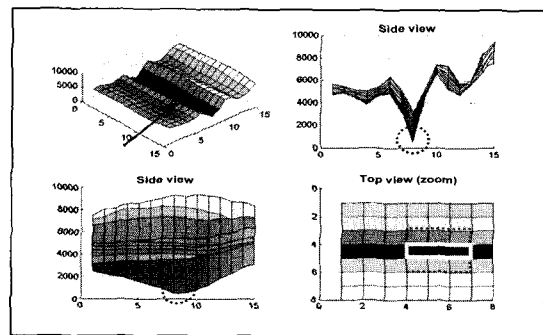
그러나 기존 BBGDS방식은 항상 현재 위치를 중심으로 주위 여덟 후보의 SAD를 구하지만, 제안방식은 제거된 후보는 상단의 슬라이스에서 후보가 될 수가 없는 차이가 있다. 그림 11은 단일급경사 계곡과 단일 협곡에서의 SAD분포와 후보경쟁단계에서의 변형된 BBGDS의 형태를 보여주고 있다.

그림 11-(a)의 오른쪽 하단의 그림을 제외한 나머지는 한 슬라이스 탐색영역 내 모든 SAD분포를 나타낸 입체도, 측면도이다. 점선의 원들은 현 슬라이스에서 가장 낮은 값이며, 주위값과 큰 차이가 있다. 이러한 값들의 차이는 최종 슬라이스에서 유지가 될 것이며, 움직임 벡터 위치는 변경될 가능성이 낮다. 그러므로 주변 여덟 후보를 고려하는 BBGDS방식(오른쪽 하단의 점선)은 비효율적이다. 부연하면 본 논문에서는 현 슬라이스의 움직임 벡터 주변의 3×3 블록내 후보들의 SAD를 계산하면서 Th_{ABS} 보다 높아지게 되면 후보를

탈락시켜 이후의 슬라이스에서 반영되지 못하도록 하였다. 본 예제에서는 실선으로 탈락되지 않는 후보를 표시하였다.



(a) 변형된 BBGDS 모델.(단일 급경사 계곡)
(a) Modified BBGDS model.(single deep valley)



(b) 변형된 BBGDS 모델.(단일 협곡)
(b) Modified BBGDS model.(single canon)

그림 11. 후보경쟁단계의 제안 모델
Fig. 11. The proposed model in CCS.

그림 11-(b)는 움직임 벡터가 x 축 방향으로 이동할 확률이 높은 경우이다. 따라서 다음 슬라이스의 후보는 블록의 중심에 있는 후보와 동일한 축상의 3개의 후보가 된다.

IV. 실험 결과

본 논문에서는 특색이 다르고 널리 사용되는 여섯 종류의 테스트 영상을 사용하였으며, 널리 알려진 고속 탐색 방식(TSS, NTSS, BBGDS, FSS, 2DLOG, DS)과 비교하였다. 실험에 사용된 매개변수 중 $Slice_{START}$ 는 3이며, P_{ABS} 와 P_{REL} 은 1.5와 0.5로 고정하였다.

한 블록을 처리하기 위해 사용되는 식 (3)의 사용빈도는 약 500~12,000이었다. 풀이 깊은 단일 계곡모양

의 블럭은 약 600번 정도 사용이 되었고, 평탄영역일 경우는 10,000번 이상 사용되는 블럭도 있었다. 기존의 블럭정합 방식과 제안하는 슬라이스 경쟁방식의 평균 MAD와 SAD수식 사용 횟수를 표 4에서 비교하였다. 평균 MAD는 기존 고속 블럭탐색 방식보다 결과가 우수하고 전역 탐색방식에 근접하는 결과를 보여주고 있으며, 다양한 입력영상에 대하여, SAD의 횟수는 기존 방식의 평균보다 10%~70% 줄어들었다.

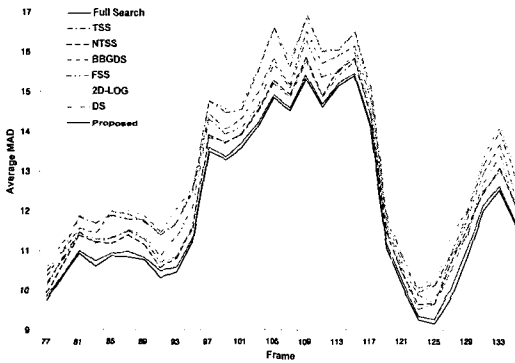


그림 12. 평균 MAD의 비교 (Football2(73~148 frame))
Fig. 12. The comparison of the average MAD. (Football2(73~148 frame)).

제안 방식을 기존 블럭정합 방법들의 평균 연산량으로 조절한 후, 움직임 동작이 많은 "Football2" 테스트

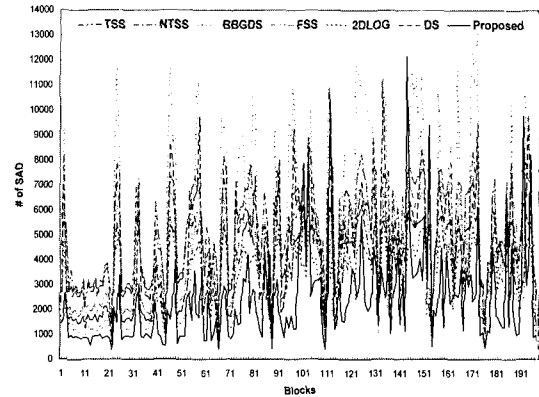


그림 13. SAD 사용빈도 예.(Football2(81 frame))
Fig. 13. The example of the frequency of SAD usage. (Football2(81 frame)).

영상 중 가장 움직임이 큰 블럭들에 대하여, MAD를 비교하였다(그림 12). 제안하는 방식은 기존의 방식에 비해 전역탐색의 결과에 가장 근접함을 알 수 있다. 그리고 81번째 프레임에 대하여, 각 블럭정합방식에 따른 계산량을 비교하였다. 제안하는 방식은 평균적으로 가장 적은 계산량을 필요로 하며, 다양한 SAD 분포 특성에 따라 계산량이 변화되어진다(그림 13).

동일한 환경에서 동일한 조건으로 최적화를 시킨 후, 각 블럭정합 방법들의 평균 수행시간을 조사하여 표 5에 나타내었다. 전역탐색방법보다 5.2배, 기존의 고속 블럭정합 방식에 비해 18%~35% 수행 시간이 단축이 되었다.

표 4. 각 BMA의 성능비교 (평균MAD와 SAD횟수)

Table 4. The performance comparison of each BMAs (Average MAD and # of SAD)

		FS	TSS	NTSS	BB-GDS	FSS	2D-LOG	DS	Proposed
Mobile	MAD	9.457	9.698	9.464	9.465	9.593	9.570	9.531	9.506
	# of SAD	28278	2970	3071	1930	2489	1852	2200	735
Flower garden	MAD	8.846	9.731	8.946	9.052	9.393	9.375	9.076	8.919
	# of SAD	29030	3530	3566	2741	3209	2281	2886	1083
Football2	MAD	8.7603	9.089	9.020	9.330	9.205	9.515	9.440	8.870
	# of SAD	32335	3936	4897	4991	4169	3085	4200	2878
Table tennis	MAD	5.180	6.971	5.486	5.482	5.436	5.480	5.332	5.374
	# of SAD	28979	3317	3211	2225	2718	2010	2428	1194
Tennis	MAD	10.055	11.515	12.395	11.514	11.928	12.033	11.948	11.160
	# of SAD	31616	3512	3948	4313	3630	2665	3435	2181
Football3	MAD	7.353	8.350	8.360	8.432	8.391	8.407	8.406	7.499
	# of SAD	17951	1519	1546	701	1124	862	879	998

표 5. 각 BMA의 소요시간(1 프레임)
Table 5. The time required of each BMAs(1 frame).

	FS	TSS	NTSS	BB-GDS	FSS	2D-LOG	DS	pro-posed
time(sec)	0.972	0.225	0.239	0.210	0.209	0.220	0.208	0.176
relative time	5.522	1.278	1.357	1.193	1.187	1.250	1.181	1.000

그리고 제안하는 방식은 후보선출단계에서 연산량의 25%~40%가 소요되었으며, 나머지는 후보 경쟁단계에서 사용되었다. 후보 선출단계 1, 2단계 이후, 남아있는 후보의 수는 평균 1.3~3.9개이며, 후보경쟁단계에서는 평균 1.0~1.8개의 후보들이 남아있었다.

V. 결론

본 논문에서는 표준 부호화기에서 사용하는 블럭정합 방법 중 연산량의 단점을 보완한 고속 블럭정합 방식을 제안하였다.

각각의 후보의 SAD증가 추이의 선형성을 높이기 위하여, 이중 베이어디더(Bayer dither) 행렬을 사용하는 확산누적방식(dispersed accumulation method)을 제안하였으며, 이로써 대부분의 블럭에서 가능성이 적은 후보를 제거하기 위한 판단기준을 마련하였으며, 타당성을 검증하였다.

이를 토대로 후보선출단계(candidate election step)와 후보경쟁단계(candidate competition step)로 구성되어 있는 고속 슬라이스 경쟁 블럭정합방식을 제안하였다. 이 방식은 SAD증가의 추이를 판단하여, 각 후보의 최종 SAD값을 예측하는 방식이다. 널리 알려진 기존의 고속 블럭정합 방법들에 비해 계산량은 10%~70%정도 감소하였으며, 실제 수행시간에서는 18%~35% 감소되었다. 그리고 화질에 영향을 미치는 평균 MAD는 다양한 테스트 영상에 대하여 항상 우수하였으며, 전역탐색에 가장 가까운 결과를 보였다. 현재는 슬라이스 간 상관도가 높은 점을 고려하여 하드웨어 기반의 단순화된 블럭정합 방식에 대해 연구하고 있다.

참고 문헌

[1] International Telecommunication Union, "Video codec for audiovisual services at p×64kbits,"

ITU-T Recommendation H.261, Mar. 1993.

[2] International Telecommunication Union, "Video coding for low bitrate communication," ITU-T draft H.263, July 1995.

[3] ISO/IEC JTC1/SC29/WG11, "ISO/IEC CD 11172 : Information technology," MPEG-1 Committee draft, Dec. 1991.

[4] ISO/IEC JTC1/SC29/WG11, "ISO/IEC CD 13818 : Information technology," MPEG-2 Committee draft, Dec. 1991.

[5] M. Ghanbari, *Video coding an introduction to standard codecs*, The institution of Electrical Engineers, 1999.

[6] D. Lim and Y. Ho, "A fast block matching motion estimation algorithm based on statistical properties of object displacement", *IEEE Region 10 Int. Conf. on Global Connectivity in Energy, Computer, Communication and Control*, vol. 1, pp. 138-141, -- 1998.

[7] T. Koga and K. Iinuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. NTC81*, pp. 531-535, Nov. 1981.

[8] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799-1806, Dec. 1981.

[9] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 4, pp. 438-442, Aug. 1994.

[10] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 6, pp. 313-317, June 1996.

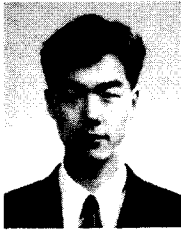
[11] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, pp. 419-422, Aug. 1996.

[12] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion

estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 287-290, Feb. 2000.

- [13] B. E. Bayer, "An optimum method for two level rendition of continuous-tone pictures," in *Proc. IEEE Int. Conf. Commun.*, Conference Record, pp. (26-11)-(26-15), 1973.

저 자 소 개

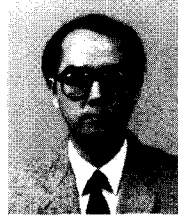


鄭映勳(正會員)

1971年 10月 4日生.1994年 2月 동의대학교 전자공학과 공학사. 1996年 2月 부산대학교 전자공학과 공학석사. 2001年 8月 부산대학교 전자공학과 공학박사. 주관심 분야는

Image processing, VLSI design,

Image (video) compression system 등임



金在浩(正會員)

1980년 부산대학교 전기기계공학과 공학사. 1982년 한국과학기술원 산업전자공학과 공학석사. 1990년 한국과학기술원 전기 및 전자공학과 공학박사. 현재 부산대학교 전자공

학과 부교수. 주관심 분야는 VLSI design, Parallel processing, Image processing, Color image processing, Image communication 등임.