시공간적 상관성을 이용한 움직임 벡터 예측 기반의 FASCO 블럭 정합 알고리즘

정회원 정 영 훈*, 김 재 호**

The FASCO BMA based on Motion Vector Prediction using Spatio-temporal Correlations

Young-hoon Jeong*, Jae-ho Kim** *Regular Members*

요 약

본 논문에서는 표준 비디오 부호화기를 위한 블럭 정합 방식을 제안하였다. 일반적으로 기존 방식에 사용되는 광역-협역 방식이 아닌 "슬라이스 경쟁"이라는 새로운 개념이 도입되었다. 기존의 SAD의 누적 방식인 순차 방식에서 확산 방식으로 변경함으로써 SAD증가 추이의 선형성이 확보되므로, 누적 초기에 움직임 벡터로서 가능성이 낮은 후보들을 미리 제거하여 불필요한 계산량을 줄이는 방식이다. 그리고 움직임 벡터 예측방식과 적응적 탐색 영역 개념을 도입하여 블럭 정합 방식을 효율적으로 지원하였다. 이 두 방식의 도입으로 약 13%의 계산량 감소가 발생하였으며, 최종적으로 기존의 고속 블럭 정합 방식들과 비교하면 39%~77%의 SAD 누적 횟수가 감소되었다. 그리고 다양한 테스트 영상에 대하여, 평균 MAD는 항상 낮으며, 전역 탐색 블럭 정합 방식에 가장 근접한 결과를 얻었다.

ABSTRACT

In this paper, a new block-matching algorithm for standard video encoder is presented. The slice competition method is proposed as a new scheme, as opposed to a coarse-to-fine approach. The order of calculating the SAD(Sum of Absolute Difference) to find the best matching block is changed from a raster order to a dispersed one. Based on this scheme, the increasing SAD curve during its calculation is more linear than that of other curves. Then, the candidates of low probability can be removed in the early stage of calculation. And new MV prediction technique with an adaptive search range scheme also assists the proposed block-matching algorithm. As a result, an average of 13% improvement in computational power is recorded by only the proposed MV prediction technique. Synthetically, the computational power is reduced by 39%~77% than that of the conventional BMAs. The average MAD is always low in various sequences. The results are also very close to the MAD of the full search block-matching algorithm.

Ⅰ. 서 론

디지털 비디오 신호는 높은 압축이 가능하도록 많은 중복성을 지니고 있다. 따라서 연속적인 프레임의 높은 시간적 중복성을 감소시키기 위하여, 두 프레임 간 블럭의 움직임 변위를 찾아내는 움직임 추정(motion estimation)은 비디오 압축에서 널리

* 삼성전자(주) Digital Media 총괄 중앙연구소 선임연구원(yhjung71@samsung.co.kr),
** 부산대학교 컴퓨터 및 정보통신연구소 (jhkim@pusan.ac.kr)
논문번호 : 010177-0710, 접수일자 : 2001년 7월 10일

사용되어지고 있다. 움직임 추정방식은 옵티컬 플로우(optical flow), 블럭-기반(block-based), 화소-회귀(pel-recursive), 그리고 베이지안(Bayesian method)[1]의 4가지로 나누어 질 수 있으며, 블럭 기반 방식은 블럭 당 한 개의 움직임 벡터가 필요하므로, 움직임 필드의 표현에 적은 계산량을 필요로 하므로, VLSI 구현에 용이한 점을 지니고 있다. 블럭 기반 방식 중, 블럭 정합 모델은 가장 하드웨어 복잡도가 낮으므로, 가장 실용적인 방식으로서, H.261, H.263 그리고 MPEG-1/2/4등의 비디오 압축의 국제 표준으로 사용되고 있다[2].

전역 탐색 방식은 탐색 영역 내 모든 후보를 검색하므로 가장 좋은 성능을 지니고 있지만, H.261이나 MPEG-1등의 비디오 부호화기 총 계산량의 60%~70%를 차지하므로 구현시 문제점을 내포하고 있다. 따라서 전역탐색 블럭 정합방식의 10배 이상 감소된 계산량만으로 비슷한 성능을 가지는 고속 블럭 정합 방식들이 제안되고 있으며, 본 논문에서는 기존 방식들과는 다른 접근 방식으로 효율적인 고속 블럭 정합 방식을 제안하였다.

Ⅱ. 블럭 정합 방식

블럭 정합 방식(Block Matching Algorithm: BMA)은 연속적인 두 프레임 사이의 시간적 상관성과 각 프레임 내의 블럭간의 공간적 상관성을 이용하며, 일반적으로 블럭 크기는 $M\times N$이다(그림. 1). 블럭 정합 방식의 기본적인 개념은 현재 블럭과 가장 유사한 블럭을 이전 프레임 내에서 찾는 것이다. 그리고 블럭 자체를 보내는 대신 예측 오차와 움직임 벡터(motion vector: MV)를 복호기로 보내는 방식이다. 현재 프레임은 동일한 크기의 블럭으로 나누어지며, 이 블럭은 정합의 기본 템플릿(template)이 된다. 탐색을 위한 계산량을 줄이기 위하여, 템플릿의 위치와 대응되는 이전 프레임 내의 블럭 주위의 탐색 영역을 w로 제한한다. 그리고 정합 척도를 이용하여 탐색 영역 내의 블럭 중 가장 유사한 블럭을 찾아낸다. 최종적으로 이 두 블럭 사이의 상대적인 변위가 움직임 벡터가 된다.

정합의 정확도를 평가하는 척도는 다양하지만, 일반적으로 MSE(mean square error), MAD(mean absolute difference), MPC(matching pel count)가 사용되어지며, 이외에도 CCF(cross correlation function)와 MiniMAX (minimized maximum error)가 있다. MAD, MPC, MiniMAX는 곱셈 대신 비교와 누적 계산이 필요하다. MAD는 MSE와 비슷한 성능을 가지고 있으면서 낮은 복잡도로 인해 널리 사용된다[3].

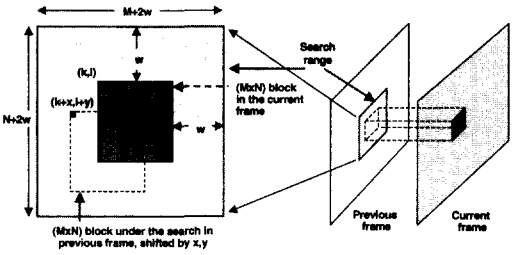


그림 1. 블럭 정합 방식의 탐색 영역과 기본 템플릿

현재 블럭과 이전 블럭의 위치가 각각 (k,l), $(k+x,l+y)$이면 MAD는

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

이다. $I_t(i,j)$와 $I_{t-1}(i,j)$는 현재 프레임과 이전 프레임 좌표 (i,j)의 화소의 밝기를 나타낸다. 그리고 (x,y)는 두 블럭간의 위치 변위이다. 또한 하드웨어 기반의 방식에서는 SAD 척도가 자주 언급된다.

$$SAD_{(k,l)}(x,y)=\sum_{i=1}^{M}\sum_{j=1}^{N}|I_t(i+k,j+l)-I_{t-1}(i+k+x,j+l+y)| \quad (2)$$

현 블럭 $I_t(i,j)$의 움직임 벡터 $mv(k,l)$는 수식 (3)으로 나타낼 수 있다.

$$mv(k,l)=\arg\min_{(x,y)} SAD_{(k,l)}(x,y) \quad (3)$$

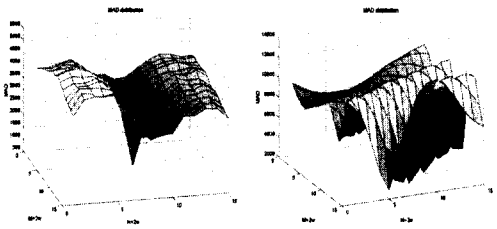
한 후보의 완전한 SAD는 $M\times N-1$번의 가산과 $M\times N$번의 감산이 요구된다. 만약 탐색 영역이 w로 제한되어진다면 총 후보 수는 $(2w+1)^2$이 된다. 예를 들어 블럭이 정방형의 16화소로 이루어지고 탐색 영역 w가 7이면, 전역탐색 블럭 정합 방식(Full Search BMA: FSBMA)을 위하여 약 576,000번의 가감산이 필요하다. 이처럼 전역탐색방식은 최상의 성능을 제공하지만 많은 계산량을 요구한다. 이러한 단점을 극복하기 위하여 많은 고속 블럭 정합 방식(Fast BMA: FBMA)이 지난 20년간 제안되었다.

기존의 고속 블럭 정합 방식은 탐색 위치 또는

후보의 수를 제한함으로써 계산량을 감소시킨다. 그리고 각 후보들의 배치와 각 스텝간 변위 차이를 제외하면 비슷한 광역-협역 접근(coarse-to-fine approach) 방식을 따른다. 따라서 전반적인 SAD의 분포가 단조 감소할 때 좋은 성능을 가진다. 그러나, SAD분포가 국부 최소값들을 가지면, 쉽게 빠져서 최적의 움직임 벡터를 찾지 못하는 단점을 지니고 있다[4].

두 종류의 SAD 오차 표면들이 그림 2에 있다. 단조 감소하는 SAD의 오차 표면이 그림 2-(a)에 나타나있다. 이러한 경우, 대부분의 고속 블럭 정합 방식들은 동일한 최적의 움직임 벡터를 찾을 수 있을 것이다. 이와 반대로 그림 2-(b)의 경우는 기존의 방식들은 쉽게 국부 최소값에 빠질 것이다.

널리 알려진 고속 블럭 정합 방식은 TSS(Three Step Search)[5], 2DLOGS(2-D LOGarithmic Search)[6], NTSS(New Three Step Search)[7], FSS (Four Step Search)[8], BBGDS(Block Based Gradient Descent Search)[9], OSA(Orthogonal Search Algorithm)[10], DS(Diamond search)[11], 그리고 HBMA(Hierarchical Block Matching Algorithm)[12] 등이 있다.



(a) 단일 급경사 계곡 (b) 복합 계곡

그림 2. 다양한 형태의 SAD 오차 표면

Ⅲ. 제안방식

제안하는 블럭 정합 방식을 효율적으로 지원하기 위하여 움직임 벡터(MV) 예측 방식에 대하여 기술할 것이다. 제안 방식은 주변 블럭의 움직임 벡터의 시간/공간적 상관도를 이용하여 예측하는 방식에 속한다. 다양한 예측 모델의 도입으로 몇 가지 추론을 이끌어 내었으며, 이 방식을 기반으로 중도-정지-제거(halfway-stop-reject) 개념을 가지는 고속 블럭 정합 방식인 FASCO BMA(FAst Slice COmpetition Block Matching Algorithm)을 제안하였다.

전체적인 흐름도는 그림 3에 있다. 제안 방식은 MV 예측 단계(MV prediction step), 후보 선택 단계 (candidate selection step), 후보 경쟁 단계 (candidate competition step)의 세 단계로 나누어지며, 각 단계는 블럭별로 처리된다. 먼저 간략하게 설명하면, MV 예측 단계 수행 후, 예측된 MV의 위치와 탐색 영역의 범위에 관련된 결과들은 다음 단계로 이전된다. 그리고 난 후, 미리 설정된 슬라이스 $Slice_{START}$에서 후보 선택 단계가 수행된다.

MV로서 가능성이 높은 후보들을 찾은 후, 후보 경쟁 단계에서 슬라이스를 증가시키면서, 각 후보들의 SAD값에 따라서 중도에 누적-중지 후 제거시킨다. 그리고 최종 슬라이스에 남아 있는 후보 중 가장 낮은 SAD값을 가지는 후보를 움직임 벡터로 설정한다. 상기 문장에서 제시된 "슬라이스(Slice)"란 용어는 다음절에서 자세하게 언급되어진다. 참고로 본 방식의 기본 개념의 일부는 [13]에 언급되어 있다. 참고논문[13]과 차별화 된 부분은 MV 예측부분과 후보 선택단계이며, 향상된 성능을 위하여 새롭게 도입이 된 단계이다.

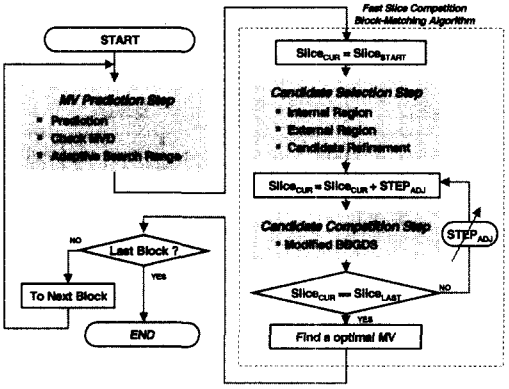


그림 3. 제안 방식의 흐름도

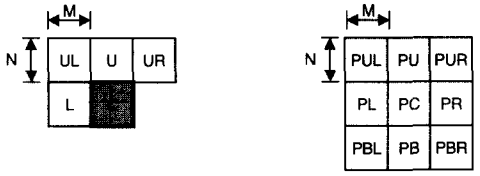
3.1. 시공간 상관성을 이용한 움직임 벡터 예측

현 블럭의 움직임 벡터는 시간 또는 공간축으로 주변 블럭 정보를 사용하여 예측할 수 있다[14-17]. 이러한 예측 MV는 블럭 정합 방식의 초기 MV로 설정되어진다. 기존 고속 블럭 정합 방식은 계산량 측면에서는 우수하지만 항상 최적의 움직임 벡터를 찾을 수가 없으며, 이는 압축률에 영향을 준다. 따라서, FSBMA와 달리, FBMA의 성능은 MV 추정 정확도에 따라 직접적인 영향을 받게 될 것이다. 만약 추정된 MV와 FSBMA에 의해 생성된 MV간의

유클리디안 거리, 즉 예측 오차(prediction error)가 적다면, 기존 FBMA는 적은 계산량을 요구할 것이며, FSBMA와 비슷하거나 동일한 MV를 선택할 것이다. 만약 그렇지 않다면, 국부 최소값에 빠지거나 최적의 MV를 찾기 위해 많은 후보의 선택과 계산이 필요할 것이다.

3.1.1 MV 예측 함수(MV prediction function)

먼저 움직임 벡터 필드의 시공간적 상관도를 분석하였다. 각각의 프레임은 16×16의 블럭으로 나누어져 있으며, 프레임 형식은 SIF(Standard Interchange Format)를 따른다. 테스트 영상으로 사용된 "Flower garden", "Football2", "Football3", "Mobile & Calendar", "Table tennis", "Tennis"는 총 197,340블럭이며, MV간의 변위를 분석하기 위하여 다양한 예측 모델들을 설정하였다. 그리고 단순한 척도(식 4)를 사용하여 예측 오차의 정확도를 평가하였다. 그림 4는 상관도 측정을 위해 사용된 관련 블럭의 표기법을 나타내고 있다. 그림 4-(a) 내의 4개 블럭은 현재 프레임에서 공간적 상관 블럭을 나타내며, 그림 (b)는 시간적으로 관련되는 블럭들을 표현하고 있다.



(a) 현재 프레임 내의 인접 블럭들 (b) 이전 프레임내의 대응되는 블럭들

그림 4. 움직임 벡터 예측을 위한 지원 범위

C, U, B, L 그리고 R은 각각 Current, Upper, Bottom, Left, 그리고 Right를 의미한다. 그리고 접두어 P는 previous를 의미한다. 그림 4-(a)는 인과 모델(causal model)이며, 그림 4-(b)는 9개의 주변 블럭을 가지고 있는 비인과 모델(non-causal model)이다. 예측 함수는 평균 함수와 메디안 함수가 두 종류를 사용하였으며, 다양한 예측 모델 중 일곱 모델만 표 1에 표시하였다. MV_{px}와 MV_{py}를 예측 움직임 벡터의 각 좌표, MV_{ox}와 MV_{oy}를 FSBMA에 의한 최적 움직임 벡터의 좌표라고 가정을 한다면, 두 MV사이의 예측 오차는

$$PE=\sqrt{|MV_{px}-MV_{ox}|^{2}+|MV_{py}-MV_{oy}|^{2}} \quad (4)$$

로 나타내어진다.

각 모델에 대한 MV 예측 오차들은 표 1에 나타나 있다. 모델 1은 예측 과정을 수행하지 않는다. 즉 모든 블럭에 대하여 예측 MV가 항상 원점에 고정되어 있으며, 이 때 예측 오차는 1.446이였다.

표 1. 움직임 벡터 예측을 위한 다양한 예측 모델과 그 결과

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	N/A	N/A	N/A	**1.446**
2	U,UR,L	Average	Spatial	**0.833**
3	U,UR,L	Median	Spatial	**0.833**
.....	.....	.....	.....	.....
8	UL,U,UR L,PC	Average	Spatio-Temporal	**0.846**
9	UL,U,UR L,PC	Median	Spatio-Temporal	**0.775**
		.....	.....	.....
14	PUL,PU,PUR, PL,PC	Median	Temporal	**1.045**
15	PUL,PU,PUR, PL,PC,PR ,PLL,PL,PLR	Median	Temporal	**0.871**

유사한 방식으로 시간/공간/시공간적 상관성을 지니는 모델 2에서 15사이 모델들에 대하여 적용한 후, 예측 오차를 분석하였다. 표 1의 결과로부터 다음과 같은 통계적 특성을 얻을 수 있었다.

※ Function: 메디안 함수가 평균 함수보다 예측 오차가 적은 경향을 보인다.

※ Correlation: 시간적 상관 블럭만 고려한 경우가 가장 저조한 결과를 나타내고 있으며, 시-공간적 상관 블럭을 고려할 때 가장 좋은 성능을 가진다.

※ Number of the adopted blocks: 움직임이 적은 테스트 영상에서는 참조 블럭의 수가 많은 경우가 유리하며, 반대의 경우는 2 또는 3개의 블럭이 우수한 경향을 띄고 있다.

상기 추론으로부터, 최종적으로 모델 9를 선정하였으며, 예측 MV의 위치는 각 좌표축에 독립적인 2개의 수식으로 결정된다.

$$MV_{px}=Median(MV_{ULx}, MV_{Ux}, MV_{URx}, MV_{Lx}, MV_{PCx}) \quad (5\text{-}(a))$$

$$MV_{py} = Median(MV_{ULy}, MV_{Uy}, MV_{URy}, MV_{Ly}, MV_{PCy}) \quad (5\text{-(b)})$$

3.1.2 MV 변위 함수 (MV Displacement function)

그림 5-(a)와 그림 5-(b)에 전역탐색 블럭정합 방식을 이용하여, "Tennis"과 "Football2"의 움직임 벡터 필드의 예를 나타내었다. 첫번째 예의 움직임 벡터들은 높은 상관도를 지니고 있으며, 이러한 조건에서는 단순한 예측함수라도 성능이 우수할 것이다. 이와 반대로 후자(그림 5-(b))의 경우는 전자보다 낮은 상관도를 보이므로, 상대적으로 큰 예측오차로 인해 움직임 벡터들이 국부최소값에 빠지게 되고 적절하지 않은 움직임 벡터를 선출할 확률이 높아지게 된다. 결과적으로 화질에 영향을 주게 된다.

이러한 단점을 보완하기 위하여 MV 변위 (MV Displacement: MVD) 함수를 제안하였다.

각 축에 대한 MVD함수는 다음과 같다.

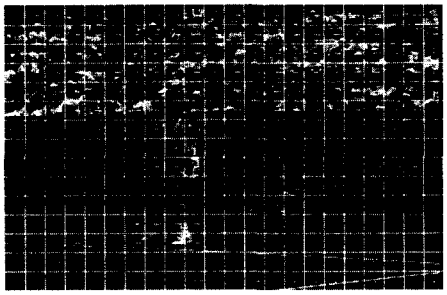
$$MVD_x = \frac{1}{N_{MV}} \sum_{n=0}^{N_{MV}-1} | MV_{nx} - MV_{px} | \quad (6\text{-(a)})$$

$$MVD_y = \frac{1}{N_{MV}} \sum_{n=0}^{N_{MV}-1} | MV_{ny} - MV_{py} | \quad (6\text{-(b)})$$

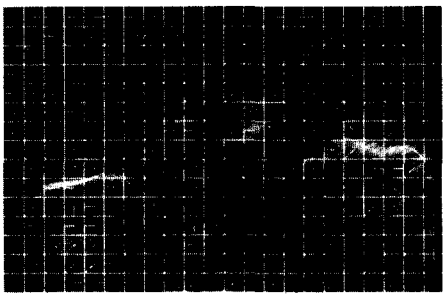
M_{MV}는 주변 블럭의 수이며, MV_{nx}와 MV_{ny}는 수식 (5)에 명시되어 있는 각 주변 블럭을 의미한다.

그림 6은 그림 3의 "MV prediction" 블럭 내의 전체 흐름도이다. 점선으로 둘러싸인 블럭은 양축에 대하여 독립적으로 진행되는 블럭이다. 예측 MV는 제안하는 블럭 정합 방식의 초기 MV가 된다. 전체적인 진행과정은 다음과 같다.

수식 5에 의해 각축에 대하여 예측된 움직임 벡터 (MV_{px}, MV_{py})들은 생성된다. 그리고 주변 MV에 대한 변위의 정도 (MVD_x, MVD_y)는 식 6에 의하여 결정된다. 만약 MVD가 1.0을 초과하면, 예측 MV와 적응적 탐색 영역(Adaptive Search Range: ASR)방식은 사용되지 않는다. 즉 예측 MV의 위치는 원점으로 고정이 되며, 적응적 탐색 영역은 최대 탐색영역이 된다. 이와 반대로 MVD가 1.0미만이면, 불필요한 계산량을 줄이기 위하여 적응적 탐색영역이 사용되어지며, 또한 예측 MV가 사용되어진다. 이 방식은 3.2.2.1절에 자세히 기술되어 있다. 상기 임계치 1.0은 실험에 의해 결정되었다. 결과적으로 최대 4개의 매개변수 (*Initial* MV_x , *Initial* MV_y, SR_x , SR_y)가 다음 단계에서 사용된다.



(a) 높은 상관도의 MV(Tennis 62~63 프레임)



(b) 낮은 상관도의 MV(Football2 144~145프레임)

그림 5. 움직임 벡터 필드의 두 가지 예

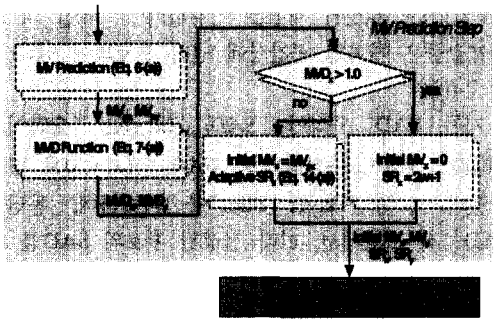


그림 6. 제안하는 MV예측방식의 흐름도

3.2. 고속 슬라이스 경쟁 블럭 정합 방식 (Fast Slice Competition Block Matching Algorithm)

고속 슬라이스 경쟁 블럭 정합 방식은 확산 누적 방식에 근거를 두고 있다. 따라서 블럭 정합 방식을 언급하기 전에 확산 누적 방식에 대하여 기술한다.

3.2.1. 확산 누적 방식 (Dispersed accumulation method)

움직임 벡터로서 상대적으로 낮은 확률을 가지는

후보들을 제거하기 위하여, 각 후보의 SAD 증가 상태를 지속적으로 관찰해야 한다. 그리고 난 후, 상대적으로 큰 SAD값을 가지는 후보들에 한하여, 누적을 중지하고 조심스럽게 제거되어져야 한다. 최종 SAD값을 정확하게 예측하기 위하여 SAD 증가 곡선은 선형화 되는 것이 가장 좋을 것이다. 만약 그렇지 않는다면, 예측 오차로 인해 부적절한 후보가 움직임 벡터로 선정이 될 것이다. 이러한 문제를 해결하기 위하여 확산 누적 방식을 제안하였으며, 왼쪽 상단에서 오른쪽 하단으로 누적시키는 기존 방식과 다름을 쉽게 알 수 있을 것이다(그림 7-(a), (b)).

3.2.1.1 기본 개념

대부분의 블럭 정합 방식은 식 (2)에 의한 SAD를 서로 비교함으로써 움직임 벡터를 선택한다. 이러한 방식은 약간의 불합리한 문제점을 가지고 있다. 즉 전역탐색보다 월등하게 계산량이 적은 고속 탐색 블럭정합 방식일지라도 후보의 가능성에 관계없이 고정적인 $2\times M\times N$의 계산량이 필요하다. 이러한 문제를 해결하기 위하여 다음과 같은 방식을 제안하였다.

확산 누적 방식은 디지털 간색(halftoning)에서 사용되어지며, 정방형의 형태를 지니고 있는 베이어 디더(Bayer dither) 배열을 이용한다[18]. 배열 내 각 임계값들은 이차원 공간상에 널리 분포되어 있다(그림 7-(c)).

기존 SAD 수식인 $SAD_{(j,k)}(x,y)$는 "슬라이스(slice)" 개념이 추가됨으로 $SAD_{(x,y)}(x,y,s)$로 변경되었다. s는 "Bayer Slice"를 의미하며, 동일한 SAD누적 조건에서, 미리 선정된 후보의 SAD를 비교하기 위하여 제안되었다. 본 논문에서는 표기의 편리함을 위하여 "Bayer slice" 대신 "Slice"란 용어를 사용하였다. 그리고 슬라이스는 1에서 $M_{mask}\times N_{mask}$까지 증가될 수 있다.

슬라이스 s에서, 현재 프레임 상의 (x,y)에 위치하고 있는 블럭과 이전 프레임의 $(x+k,y+l)$에 있는 블럭과의 SAD는

$$SAD_{(k,l,s)}(x,y)=\sum_{s_{curr}=1}^{s}|I_t(i_{mask}+k,j_{mask}+l)-I_{t-1}(i_{mask}+k+x,j_{mask}+l+y)| \quad (7)$$

로 쓰여진다. 여기서 s_{curr}은 현재 슬라이스의 인덱스이며, $x_{mask}(\cdots)$와 $y_{mask}(\cdots)$는 확산 누적 배열의 좌표이다.

상대적인 가로/세로 좌표, i_{mask}와 j_{mask}는

$$i_{mask}=\frac{M}{M_{mask}}\sum_{d_{curr}=1}^{s}x_{mask}(d_{curr})+x_{mask}(s_{curr}) \quad (8\text{-(a)})$$

$$j_{mask}=\frac{N}{N_{mask}}\sum_{d_{curr}=1}^{s}y_{mask}(d_{curr})+y_{mask}(s_{curr}) \quad (8\text{-(b)})$$

이다. s_{curr}은 국부 마스크내의 누적 화소의 위치와 관련이 있으며, d_{curr}은 $M\times N$블럭 내 마스크의 위치와 연관되어 있다. 그리고 최종 슬라이스는 식 (9)에 의하여 결정된다.

$$S=M_{mask}\times N_{mask}\quad(s_{curr}=1,2,\cdots S) \quad (9)$$

최종적으로 (k,l)에 위치한 블럭의 SAD $SAD_{(k,l)}(x,y)$는

$$SAD_{(k,l)}(x,y)=SAD_{(k,l)}(x,y,S) \quad (10)$$

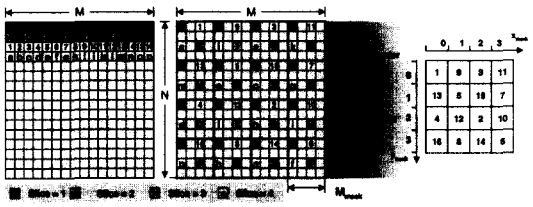
로 정해진다.

만약 M개의 화소가 SAD에 누적이 될 경우, 현재 슬라이스는 1만큼 증가되어 *Slice 1*이 되며, 개의 화소가 누적되었을 경우 *Slice N*이 된다. 이해를 돕기 위하여 기존방식과 제안방식의 참조 위치를 비교하였다. 현재 슬라이스 S_{curr}은 4이며, 그림 7-(a)는 순차적 누적방식의 순서를 보여준다. 만약 네 번째 라인의 인덱스들이 'a'에서 'p'로 표기되어 있다면, 확산 누적 방식에서는 그림 7-(b)과 같이 2차원 배열상에서 균일하게 표본화가 된다. 결국 확산 누적 배열은 크기가 4인 정방형 배열을 이중으로 사용하여 $M\times N$배열을 생성한 것이다.

블럭 크기 $M\times N$은 16×16 이며, 탐색 영역 w가 7일 경우, 탐색 영역 내 모든 후보들에 대한 SAD 추이 곡선을 그림 8에 나타내었다. 각 곡선은 한 후보의 SAD 증가 추이를 나타낸 것이며, 식 (2)에 의한 누적 횟수는 256 ($M\times N$), 그리고 총 곡선의 수는 225 ($=(2w+1)^2$)가 된다. 각 곡선의 가장 높은 SAD값은 최종 SAD를 나타내며, 그림 2의 예처럼, SAD 오차 표면을 형성한다는 것을 쉽게 추측할 수 있다. 그림 8-(a)와 그림 8-(b)는 동일한 블럭에 대한 두 방식의 차이를 보여주고 있다. 최종 SAD값은 서로 동일하지만, 추이 곡선의 선형성 차이를 쉽게 발견할 수 있을 것이다.

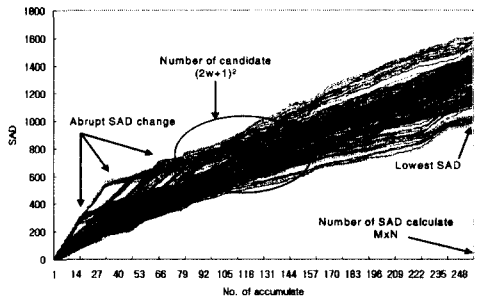
만약 현재 블럭의 상단에 새로운 물체가 나타난

다면, 그림 8-(a) 누적 초기에 급격한 SAD 변화가 발생한다. 이처럼 급격 또는 약간의 SAD변화는 선형성에 영향을 주게 되므로 낮은 가능성의 후보를 제거할 정확한 기준을 마련하기 어려워진다. 그러나 제안하는 확산 누적 방식은 급격한 SAD 변화를 SAD 추이 곡선에 등간격으로 분할함으로써 기존 방식보다 높은 선형성을 확보하게 된다.

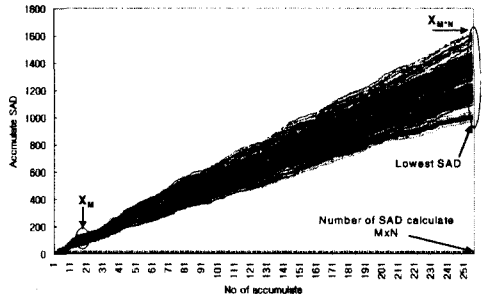


(a) 순차누적방식 (b) 확산 - (c) 베이어 디더 배열

그림 7. 두 방식의 참조 위치 비교



(a) 순차 누적 방식

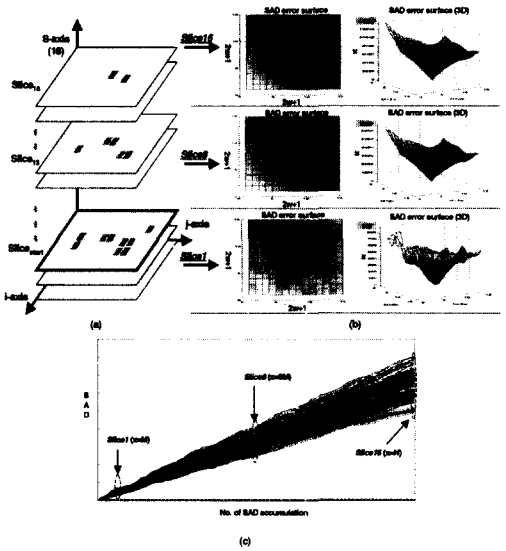


(b) 확산 누적 방식

그림 8. 블럭내 모든 후보의 SAD 추이 곡선

3.2.1.2 확산 누적 방식을 이용한 슬라이스 구조

전체적인 개념을 파악하기 위하여, 제안하는 슬라이스 구조와 SAD 오차 표면의 입체도와 평면도 그리고 SAD 추이 곡선을 그림 9에 나타내었다. 다시 언급하면, 기존 블럭 정합 방식은 최종 SAD 오차 표면, 즉 *Slice 16*(그림 9-(a))상에서만 수행되어졌다. 그러나 제안 방식은 슬라이스 축인 S-축을 도입하여, SAD 누적 초기에 미리 선정된 후보의 가능성을 조사하여 불필요한 계산을 줄이는 방식이며, 누적 곡선의 선형성을 높여 후보제거를 위한 타당성 있는 기준을 마련하였다. 그림 9에서는 *Slice1*, *Slice8* 그리고 *Slice16*의 SAD (오차) 표면을 보여주고 있으며, SAD 추이 곡선과 오차 표면과의 연관성도 세 그룹의 점선으로 그림 9-(c)에 표시되어 있다.



(a) 슬라이스 구조
(b) Slice 1,-8,-16에서의 SAD 표면
(c) SAD 추이 곡선

그림 9. 슬라이스 구조, 일부 슬라이스의 SAD 분포

Slice a 의 표면은 식 11에 의해 생성된다.

$$SAD\ Surface_{Slice\ a} = \sum_{k=-w}^{w} \sum_{l=-w}^{w} SAD_{(k,l,a)}(x,y) \quad (11)$$

각 SAD 표면의 검은 셀들은 낮은 SAD값을 가지는 후보를 의미한다. 본 예제에서 *Slice 8*과 *Slice 16*의 오차 표면은 비슷한 분포를 지니고 있으며, 이는 곧 추이 곡선이 높은 선형성을 가지고 있다는 것을 의미한다.

3.2.1.3 SAD 추이 곡선의 선형성 검증

SAD 추이 곡선의 선형성은 MV 선택에 영향을

미치므로 선형성의 정도는 검증되어야 한다. SAD의 분포, 각 슬라이스 사이의 SAD 상관도, SAD_{MIN}(각 슬라이스의 최소 SAD)의 증가 추이, $MV_{Slice\ n}$(Slice n에서 SAD_{MIN}을 가지는 후보) 변화 등의 항목들이 분석되어졌으며[13], 총 131,340블럭이 사용되었다.

본 논문에서는 $MV_{Slice\ n}$의 변화에 대해 간단하게 언급하였다. $MV_{Slice\ 1}$의 경우, 69%의 블럭이 *Slice 16*에서의 MV, 즉 최적의 MV(= $MV_{Slice\ 16}$)와 동일한 위치에 있었다. 그리고 81.3%의 블럭에서 $MV_{Slice\ 3}$과 $MV_{Slice\ 16}$의 위치가 동일하였으며, $MV_{Slice\ 8}$과 $MV_{Slice\ 14}$경우는 각각 90.9%와 95.2%의 블럭에서 최적 MV와 동일한 위치에 있음을 분석하였다. 그리고 $MV_{Slice\ 16}$과 동일한 위치에 대한 분석이 아닌 근접 위치내의 분석일 경우, 확률은 급격하게 증가하였다.

3.2.2 FASCO 블럭 정합 방식의 커널

확산 누적 방식에 근거를 두고 있는 FASCO 블럭 정합 방식은 후보 선택 단계(Candidate Selection Step: CSS)와 후보 경쟁 단계(Candidate Competition Step: CCS)로 나누어진다.

3.2.2.1 후보 선택 단계(CSS)

본 단계의 목표는 MV로서 높은 확률을 가지는 후보들을 $Slice_{START}$에서 선택하는 것이다.

$Slice_{START}$는 미리 정해진 슬라이스이며, 일반적으로 2 또는 3이 된다. 본 절에서는 초기 후보 모델(initial candidate model)과 중도-정지-제거(halfway-stop-reject) 방식이 언급되어져 있으며, 그 이후 본 단계에 대한 설명이 되어있다.

■ 초기 후보 모델(Initial candidate model)

식 7에 의해 낮은 슬라이스(*Slice 2* 또는 *Slice 3*)에서는 적은 계산량이 요구되므로, 기존 방식보다 많은 후보를 할당할 수가 있다. 이는 국부 최소값에 빠지지 않게 하기 위함이다. 본 모델에서는 후보들을 균일하게 3:1 표본화 간격으로 배치하였으며, 60%~70%이상의 MV가 집중되어 있는 중심부 ($-1 \le x \le +1, -1 \le y \le +1$)에는 8개의 후보들을 집중 배치하였다. 모델의 중심부는 항상 추정된 움직임 벡터 (MV_{px}, MV_{py})로 이동 탐색 영역은 $2\times(2w)+1$이 된다(그림 10). 다양한 실험 후 후보들을 배치를 하였다. 제안하는 모델은 크게 나누어서 기본그룹(basic group)과 확장 그룹(extended group)으로 구성되어 있다. 기본 그룹은 Candidate 0에서 Candidate 20의 후보들이며, 나머지 후보들은 확장 그룹에 속한다. 89개의 후보 중 실제적으로 탐색 영역 내의 33 또는 32개의 후보가 사용된다. 자세한 후보 선출 과정은 아래에 언급되어 있다.

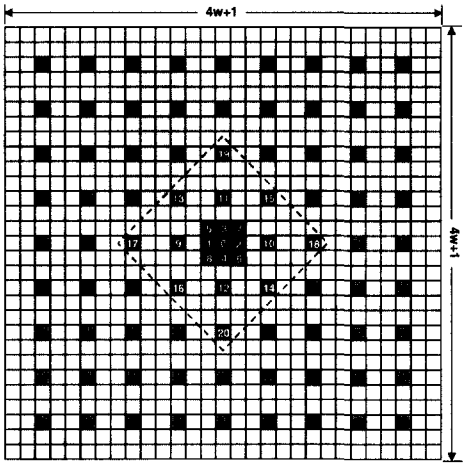


그림 10. CSS에서의 초기 후보 모델

표 2. 확장영역의 부가 후보

Index	Additional candidates
13	21,22,23,33,34,35,46,52,57,58,59,70,72,80,82
14	24,25,26,36,37,38,50,56,60,61,62,76,78,86,88
15	27,28,29,39,40,41,49,53,63,64,65,75,77,81,83
16	30,31,32,42,43,44,47,55,66,67,68,71,73,85,87
17	21,22,23,30,31,32,45,46,47,69,70,71
18	24,25,26,27,28,29,48,49,50,74,75,76
19	21,22,23,27,28,29,51,52,53,79,80,81
20	24,25,26,30,31,32,54,55,56,84,85,86

■ 중도-정지-제거방식(Halfway-Stop-Reject(HSR) method)[13]

그림 12의 흐름도는 그림 11의 중도-정지-제거 방식의 자세한 블럭도이다. 이 방식은 CCS에서 항상 사용되며, 낮은 가능성의 후보를 제거하기 위해 도입되었다. 다음은 HSR 방식의 간략화 된 내용이다.

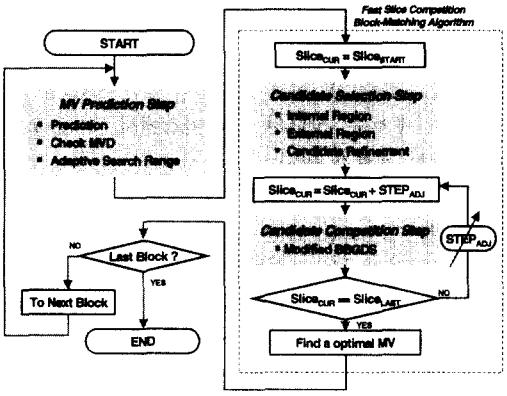


그림 12. HSR방식의 흐름도

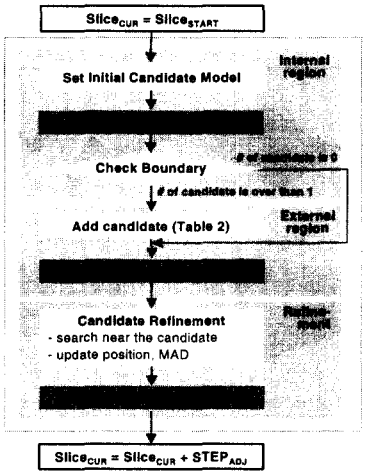


그림 13. CCS의 흐름도

미리 선정된 후보에 대하여 식 7을 이용하여 SAD를 누적시킨다. 최대 누적 범위는 $Slice_{START}$로 제한된다. 만약 누적과정에서 현재 후보의 SAD 값이 절대 임계값을 초과할 경우 현 후보는 제거되어진다. 만약 그렇지 않다면, SAD_{MIN}과 비교 후, SAD_{MIN}과 Th_{ABS}의 갱신 유무를 결정한다(그림 12).

$$Th_{ABS} = P_{ABS} \times SAD_{MIN} \quad (12)$$

만약 상기 과정이 종결되면 남아있는 후보에 대하여, 식 13의 Th_{REL}를 사용하여 상대적으로 높은 SAD를 가지는 후보를 제거한다.

$$Th_{REL} = P_{REL} \times (SAD_{MAX} + SAD_{MIN}) \quad (13)$$

앞서 언급처럼, 확산 누적 방식의 선형성을 보장은 대부분 후보들의 SAD를 동등한 비율로 증가시켰으며, 이 때 불필요한 계산을 줄이기 위하여 Th_{ABS}와 Th_{REL}를 도입하였다. 즉, 이 방식은 고정 임계값이 아닌 남은 후보들 사이의 SAD값의 분포에 따라 적응적으로 임계값이 결정되는 방식이다. 예를 들면, SAD분포가 깊은 계곡 형태를 가지는 블럭에서는 많은 후보들이 미리 제거가 되어지며, 반대의 경우는 국부 최소값에 빠지지 않기 위하여, 비슷한 SAD값을 가지는 후보들이 많이 남아있는 경향을 보인다.

■ CSS 수행 과정

현재 슬라이스 $Slice_{CUR}$는 $Slice_{START}$에 고정되어 있으며, 초기 후보 모델과 HSR방식을 사용하여 가능성 높은 후보들을 선택하는 단계이다. 본 과정은 3단계로 구성되어 있다.

1 단계 : 내부 영역에서 후보 선택

초기 후보 모델의 21개 후보에 대하여 HSR방식을 적용하여, SAD의 분포를 파악하고 가능성 있는 후보들을 추출해낸다. 기존 논문[13]과 동일한 방식이다.

2 단계 : 외부 영역에서 후보 선택

2단계에서는 선택적 후보 추가 방식과 탐색 영역을 제한하는 두 가지 방법이 적용된다.

다음은 두 방법의 자세한 설명이다.

-방법 1 : 후보의 선택적 추가

앞 단계에의 초기 후보 모델에서 마름모 형태의 경계에 걸쳐지는 후보 즉 *Candidate13 ~20*이 제거되지 않고 남아있다면, 표 2에 해당되는 후보들이 추가된다. 그리고 난 후 HSR방식이 동일하게 적용된다. 많은 MV의 분포가 중심에 위치하고 있으므로, 중심의 분포에 따라 선택적으로 적용된다. 이러한 적응적 방식은 불필요한 계산을 줄이기 위함이며, 결과적으로 MAD측면에서는 0.04%의 미세한 상승을 야기시켰지만, 약 20%의 계산량 감소가 발생하였다.

- 방법 2 : 적응적 탐색 영역(Adaptive Search Range: ASR)

2.1절에 언급된 MV 예측 과정에서 주변 블럭의 MV간의 상관도를 평가하는 MVD함수를 제안하였다. 만약 결과가 0에 가깝다면, 현 블럭의 MV는 주변 MV와 유사성이 높아지며, MV 예측과정이 먼저 수행되었으므로 MV 예측 오차는 더욱 감소하였을 것이다. 이러한 조건에서는 전 탐색영역을 검색하는 것은 불필요할 것이다. 따라서 MVD의 결과에 따라 탐색영역의 범위가 제한되는 방식을 제안하였다.

다음의 두 다항식은 simulated annealing방식을 사용하여 얻었다. 여기서 SR은 각 축에 대한 탐색영역이며, d는 식 6의 MVD를 의미한다.

$$SR_x = \lfloor\ 0.027d_x^3 - 0.293d_x^2 + 1.059d_x + 0.955\ \rfloor$$
$$SR_y = \lfloor\ 0.027d_y^3 - 0.293d_y^2 + 1.059d_y + 0.955\ \rfloor \quad (14)$$

각 방식의 성능은 4장에 언급되어 있다.

3 단계: 후보 정제

만약 마름모 형태의 경계에 남아있는 후보가 없거나 두 번째 단계가 끝나게 되면, 후보 정제 과정이 수행된다. 이 단계의 목표는 남아있는 각 후보의 주위 3×3 후보들에 대하여, HSR방식으로 더 낮은 SAD를 가지는 후보를 찾는 것이다.

2.2.2. 후보 경쟁 단계(CCS)[13]

CSS에서는 초기 후보 모델과 3단계를 사용하여 높은 가능성의 후보들을 선출하였다. 본 단계는 $Slice_{START}$ 이후의 슬라이스부터 최종 슬라이스까지 진행이 되어지며, CSS에서 남아있는 후보들을 기반으로 슬라이스를 증가시켜가며, 최종 MV를 선택하게 된다.

CCS의 초기 후보 모델과 그 과정들이 설명된다.

■ 초기 후보 모델

CCS이후, 남아있는 후보들은 초기 후보 모델과 HSR방식의 사용으로 제거, 대치 또는 유지되어진다. 제안하는 초기 후보 모델은 BBGDS[9]와 동일한 3×3 모델과 탐색 방식을 사용하며, 그림 9에서처럼 슬라이스 축(S-axis)을 따라 적용된다. BBGDS의 모델과 상이한 사항은 현재 슬라이스에서 모델 모양과 중심 위치가 이전 슬라이스의 SAD분포에 따라 다르게 상속을 받는 점이다.

기본 규칙은 다음과 같다.

1) 만약 *Slice n*에서 한 후보의 SAD가 Th_{ABS} 이상이면, 그 후보는 탈락되며, *Slice n+1* 이상의 모든 슬라이스상의 동일한 위치에 있는 후보도 탈락된다.
2) 3×3 블럭 내 후보 중 가장 낮은 SAD를 가지는 후보는 상단의 슬라이스에서 3×3 모델의 중심 후보가 된다.

다음의 두 예제는 초기 후보 모델과 개념을 나타낸 그림이다. 그림 13은 동일한 SAD 표면을 서로 다른 관찰각도에서 표현한 것이다. 슬라이스가 증가하더라도 확산 누적 방식으로 인하여 비슷한 SAD 표면을 유지하므로, 다음의 예제와 같은 경우에서는 항상 9개의 후보(점선으로 표시된 후보)는 불합리하다. 따라서 상단의 슬라이스에서는 3개의 후보(실선으로 표시된 후보)만 남아있게 된다. HSR방식을 사용하여 남아있는 후보들에 대해 동일한 과정이 적용이 되며, 최종 슬라이스까지 반복된다.

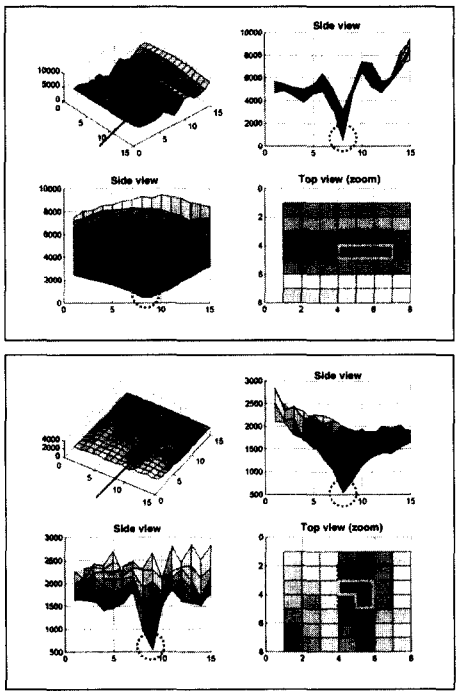


그림 13. CCS에서 변형된 BBGDS의 예

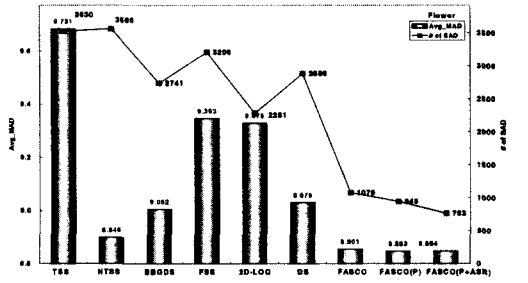
Ⅳ. 실험 결과

결과 비교의 공평성을 위하여, 정적이거나 카메라의 회전과 이동이 많거나 활동성이 높은 오브젝트들이 포함되어 있는 다양한 특성을 가지는 6종류의 테스트 영상을 이용하였다. 그리고 제안 방식의 결

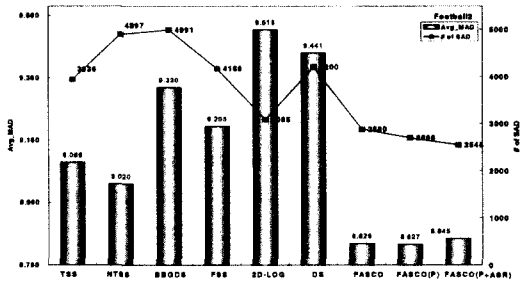
과를 검증하기 위하여 잘 알려진 고속 블럭 정합 방식(TSS, NTSS, BBGDS, FSS, 2DLOG, DS)들과 비교하였다. 본 실험에서 사용된 3가지의 매개변수들 중, $Slice_{START}$ 는 3, P_{ABS} 와 P_{REL} 는 1.5와 0.5로 고정시켰다.

한 블럭에 대한 SAD사용 빈도(식 7의 사용 횟수)는 500~12000에서 변화하며, 깊은 단일 계곡 형태의 SAD분포의 블럭에서는 약 600번, 평탄 영역인 경우에는 10000정도의 빈도를 보였다.

널리 알려진 FSBMA와 제안 방식을 사용하여 SAD의 사용 횟수와 평균 SAD에 대하여 표 3에 결과를 분석하였다. 테스트 영상의 종류에 관계없이 평균 MAD는 전역 탐색 방식에 가장 근접하였다. 여기서 "FASCO"는 MV 예측과 적응적 탐색영역(ASR)을 사용하지 않은 제안방식이며, "FASCO(P)"의 P는 MV 예측, "P+ASR"은 앞서 언급된 두 방식을 의미한다. 시각적인 비교를 위해, "Flower garden"과 "Football2"에 대한 결과를 그림 14에 나타내었다. 막대는 MAD와 관련 있으며, 검은 점은 SAD의 빈도수를 나타낸다.



(a) Flower Garden



(b) Football2

그림 14. 시각적 비교를 위한 두 그래프

SAD사용 측면에서 비교하면, 제안 방식은 기존 방식의 평균 MAD보다 낮게 유지하면서 9%~77% 감소되었으며, "P+ASR" 방식은 39%~77%의 SAD 감소가 발생하였다. 그리고 두 가지 옵션을 사용한 "P+ASR"방식은 MAD를 유지하면서, 계산량을 약 13%정도 감소시켰다.

표 4. 성능비교(기존방식의 평균과 다양한 옵션의 제안방식)

	MAD	- 0.053	- 0.361	- 0.437	- 0.891	- 0.730	- 0.730	- 0.47
	# of SAD (%)	69.65	64.45	31.64	9.64	39.11	39.11	44.92
	MAD	- 0.061	- 0.369	- 0.440	- 0.884	- 0.775	- 0.775	- 0.49
	# of SAD (%)	70.76	66.86	36.00	11.45	43.11	43.11	48.44
	MAD	- 0.071	- 0.368	- 0.422	- 0.627	- 0.771	- 0.771	- 0.45
	# of SAD (%)	77.17	74.86	39.59	40.80	48.85	67.90	58.20

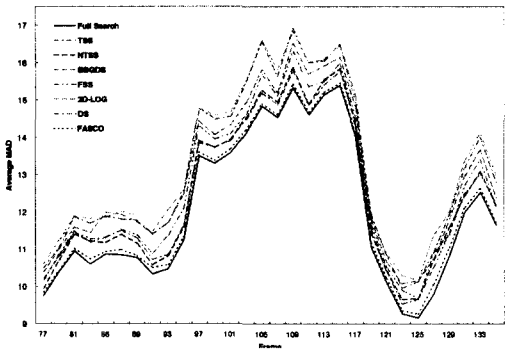


그림 15. 프레임 축에 대한 평균 MAD (Football2 (frame77~135))

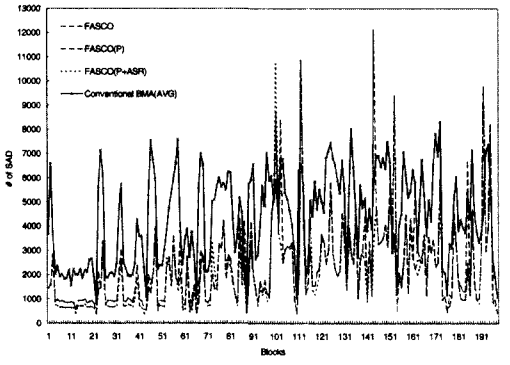


그림 16. 프레임 내의 SAD 사용빈도의 예 (Football2, frame 81, 198 blocks)

그림 15는 프레임 축에 대한 평균 MAD의 변화들을 비교한 그림이다. 움직임이 많은 "Football2" 테스트 영상에서 77~135프레임의 평균 MAD를 나타내었다. 결과적으로 전역탐색 블럭정합 방식에 가

장 근접한 결과를 보였다. 그리고 각 블럭에 대하여 계산량의 변화를 그림 16에 나타내었다. 총 198블럭이 사용되었으며, 실선은 기존 방식들의 평균 SAD사용 빈도이고 나머지 점선들은 제안 방식의 결과들이다. 대부분의 블럭에서 계산량이 낮음을 보여주고 있다.

V. 결 론

기존의 광역-협역 방식이 아닌 중도-정지-제거 개념을 포함하고 있는 슬라이스 경쟁 방식을 제안하였다. SAD 누적 곡선의 선형성을 보장하는 확산 누적 방식의 사용으로 움직임 벡터로서 가능성이 낮은 후보들을 누적 초기에 제거를 하여 불필요한 계산량을 감소시켰다.

이를 기반으로 후보 경쟁 단계(Candidate Selection Step)와 후보 선출 단계(Candidate Competition Step)의 두 단계로 구성되어 있는 고속 슬라이스 경쟁 블럭 정합 방식(Fast Slice Competition Block Matching Algorithm(FASCO BMA))을 제안하였다.

그리고 주변 블럭의 움직임 벡터를 이용하여 현 블럭의 움직임 벡터를 추정하기 위한 방법도 제안되었다. 다양한 예측 함수를 제안하여 성능을 평가하고, 주위의 움직임 벡터의 상관도를 측정하여, 예측된 움직임 벡터의 신뢰도를 평가하는 방식을 도입하였다. 움직임 벡터 예측과정이 없는 블럭 정합 방식과 동일한 화질을 유지하면서 약 13%의 계산량을 감소시켰다. 그리고 기존 고속 블럭 정합 방식에 비하여 제안방식은 39%~77%의 SAD 누적의 감소가 발생하였으며, 평균 MAD도 낮은 상태를 유지하였다.

현재 연구중인 과제는 계산량 비교를 위하여 일반 논문에서 사용하고 있는 SAD 누적 횟수뿐 아니라 다양한 연산 횟수를 비교하기 위한 연구를 하고 있으며, 몇 단계로 구성되어있는 본 방식의 복잡도를 감소시켜, 하드웨어 기반의 효율적인 방식에 중점을 두고 있다. 메모리 접근 횟수와 계산량의 감소, 리소스 공유, 메모리에서 레지스터로의 효율적 전송, 파이프라인 구조 등 다양한 기법이 적용된다.

참 고 문 헌

[1] A. murat tekalp, *Digital video processing*, Prentics Hall, 1995.

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

[3] D. Lim and Y. Ho, "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.

[4] J. Feng, K.-T Lo, H. Mehrpour, and E. Karbowiak, "Adaptive block-matching algorithm for video compression," *in Proc. IEE Vision, Image and Signal Proc.*, vol. 145, Issue 3, pp. 173 -178, June 1998.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. COM-38, pp. 950-953, July 1990.

[11] 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.

[12] M. Bierling, "Displacement estimation by hierarchical block-matching," *in Proc. SPIE Conf. on Visual Commun. and Image Proc.*, vol. 1001, pp. 942-951, Jan. 1988.

[13] Y. H. Jeong and J. H. Kim, "A fast block-matching algorithm using the slice-competition method," *Journal of the institute of electronics engineers of korea*, Nov. 2001. (계제예정)

[14] J. Chalidabhongse and J. Kuo, "Fast motion vector estimation using multiresolution-spatio -temporal correlations," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 7, no. 3, pp. 477-488, June 1997.

[15] I. Ismaeil, A. Docef, F. Kossentini, and R. Ward, "Efficient motion estimation using spatial and temporal motion vector prediction," *Int. Conf. on Image Proc.*, vol. 1, pp.70-74, 1999.

[16] M. Mattavelli and G. Zoi, "Vector-tracing algorithms for motion estimation in large search windows," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 10, pp. 1426 -1437, Dec. 2000.

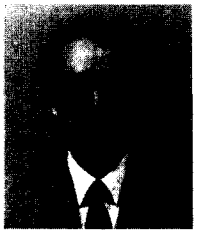
[17] C.-H. Hsieh, P.C. Lu, J.-S. Shyn, and E.-H. Lu, "Motion estimation algorithm using interblock correlation," *Electronics Letters*, vol. 26, Issue 5, pp. 276 -277, Mar. 1990

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

정 영 훈(Young-hoon Jeong) 정회원

1994년 2월 : 동의대학교 전자공학과 학사
1996년 2월 : 부산대학교 전자공학과 석사
2001년 8월 : 부산대학교 전자공학과 박사
2001년 8월~현재 : 삼성전자 digital media총괄 중앙연구소 선임연구원
<주관심 분야> system on Chip, ASIC/system 설계, Video coding, Image compression 등.

김 재 호(Jae-ho Kim) 정회원



1980년 2월 : 부산대학교 전기 기계공학과학사
1982년 2월 : KAIST 산업전자공학과 석사
1990년 2월 : KAIST 전기 및 전자공학과 박사
현재 : 부산대학교 전자공학과 부교수.
<주관심 분야> VLSI design, Parallel processing, Image processing, Color image processing, Image communication 등.