

# 낮은 비트율 비디오 코딩을 위한 블럭 유사도를 이용한 동적 탐색 영역 크기를 갖는 블럭 정합 알고리즘

正會員 오 황 석\*, 박 근 한\*, 이 흥 규\*

## Block-Matching Algorithm based on Dynamic Adjustment of Search Window for Low Bit-Rate Video Coding

Hwang-Seok Oh\*, Gunhan Park\*, Heung-Kyu Lee\* *Regular Members*

### 요 약

본 논문에서는 전역 탐색 기법 블럭 정합 알고리즘(full search block-matching algorithm: FSBMA)의 계산량을 줄이고, 많은 빠른 탐색 알고리즘들이 가지는 지역적 최소값에 빠지는 문제점을 줄이기 위해 BMA에서 동적 탐색 영역을 조정하는 기법을 제안한다. 제안한 알고리즘은 3 단계를 거쳐 수행된다. 즉 (1) 블럭 유사도(block similarity)와 변이된 블럭의 차이(displaced block difference: DBD)를 이용한 새로운 탐색 원점 설정, (2) 블럭 유사도에 반비례하게 탐색 영역의 크기를 조정, 그리고 (3) 영상 특성 및 계산량을 고려하기 위한 임계값의 갱신 단계를 거친다. 본 논문은, 위 단계를 기반으로 하여, 고정 크기 블럭을 이용하는 동적 탐색 영역 조정 및 가변 크기 블럭을 이용하는 동적 탐색 영역 조정 기법을 제안한다. 제안된 기법은 FSBMA의 계산량을 줄이고, 빠른 탐색 기법 알고리즘들의 국부 최소치에 빠지는 문제점을 줄이기 위해 이들 기법에 쉽게 적용될 수 있다. 실험을 통하여 제안한 기법들이 MSE(mean square error) 성능과 계산량(탐색 블럭의 수) 측면에서 기존의 기법들과 비교하여 좋은 결과를 가짐을 보인다.

### ABSTRACT

A dynamic search window adjustment for block-matching algorithm(BMA) based on the block similarity is presented to reduce the computational complexity of full search BMA. The adjustment of the size of search window is performed in three steps:(1)Set a new search origin based on the block similarity and the displaced block difference(DBD), (2)Adjust the size of search window in inverse proportion to block similarity, and (3)Update the thresholds for accommodation to a given image sequence. Based on the procedures, the dynamic adjustment of search

\*한국과학기술원 전산학과  
論文番號:97437-1128  
接受日字:1997年 11月 28日

window with fixed size block(DASWF) and the dynamic adjustment of search window with variable size block (DASVV) are presented. The technique can be easily applied to full search BMA and several fast search algorithms to get more efficiency and to reduce a possibility of falling into a local-minimum. Experimental results show that the proposed technique has a good MSE performance and reduces the number of search points substantially.

## 1. 서 론

움직임 보상 부호화 기법은 비디오 데이터 압축에 매우 중요한 역할을 한다. MPEG-1, 2[1], ITU-T H.261[2], 그리고 H.263[3] 등과 같은 비디오 데이터 압축 기법들은 비디오 영상의 연속적인 프레임에 내재한 시간적 중복성을 줄이기 위하여 움직임 보상 부호화 기법을 채택하고 있다. 현재까지 움직임 추정을 위해 개발된 알고리즘은 블럭 단위 기법들로 블럭 정합 알고리즘(block-matching algorithm:BMA)이라고 불리며, 블럭 단위로 움직임 벡터(motion vector: MV)를 찾는다. BMA에서 MV를 찾기 위해서 먼저 현재 프레임을  $(N \times N)$  크기의 블럭으로 겹치지 않게 나누고 참조 프레임(reference frame)의  $(2w + N) \times (2w + N)$  크기의 탐색 영역(search area) 내에서 후보 블럭(candidate block)들과 비교를 한다. 여기서  $w$ 는 가로/세로 방향으로 MV의 최대 변위이다. 현재 블럭과 후보 블럭들과 비교 후 왜곡 함수 또는 정합 기준의 최소 오차를 갖는 후보 블럭과 현재 블럭과의 변위 차이를 MV라고 한다. 일반적으로 MV를 찾기 위해서 전역 탐색 BMA(full search BMA:FSBMA)가 많이 사용되고 있다. 전역 탐색에서 MV를 얻기 위해서  $(2w + 1)^2$ 의 후보 블럭들과 정합을 하여야 한다[4].

그러나 FSBMA에서 정합 기준,함수의 최소 왜곡을 갖는 블럭의 변위를 찾기 위한 많은 계산량 때문에 실시간 비디오 코덱을 개발하는데 많은 어려움이 있으며, 대부분 하드웨어로 구현을 하고 있다. FSBMA의 계산량을 줄이는 방법은 크게 3 가지로 분류할 수 있다. 첫째로, 블럭 내에서 정합에 참여하는 화소의 수를 줄이는 방법으로 이는 특정 패턴을 이용하거나 다운 샘플링을 하여 블럭의 크기를 줄여 하나의 블럭 내에서 계산량을 줄이는 방법이다. 이러한 연구는 [24] 등에서 찾아 볼 수 있다. 그러나 정합에 사용되는 화소들이 블럭의 특성을 잘 반영할 경우는 좋은 성능을 얻을 수 있지만 그렇지 못한 경우 심한 왜곡이 발

생한다. 두번째로, FSBMA의 계산량을 줄이기 위한 연구로, 현재까지 가장 많은 연구가 된 부분으로 탐색 영역 내에서 정합에 참여하는 후보 블럭들을 줄이는 방법으로 TSS(three step search)[5], NTSS(new three step search)[6], 2-D LOG(2-D logarithm search)[7], OTS(one at a time search algorithm)[8], 1-D FS(1-D full search)[9], CSA(cross search algorithm)[10], 그리고 PHODS(parallel hierarchical 1-D search)[11] 등과 같은 많은 빠른 탐색 기법들이 제안되었다. 현재도 FSBMA와 유사한 왜곡을 가지며, 계산량을 줄이기 위한 연구가 활발히 진행중에 있다. 그러나 빠른 탐색 기법들은 최소 왜곡을 갖는 곳으로부터 멀어질수록 왜곡이 증가한다는 가정하에서 개발되었기 때문에 지역적 최소값(local minimum)에 수렴하는 문제점이 있다. 지역적 최소값에 수렴하는 경우 움직임 보상 후 예측 오차가 상대적으로 커지기 때문에 오차를 부호화할 경우 많은 비트율을 사용하여야 한다. 또 다른 방법으로는 계산량을 줄이기 위해서 탐색 영역의 크기를 동적으로 조정하는 기법들이 제안되었다 [12, 13]. [12]에서는 TSS에서 DBD의 크기에 따라서 각 스텝마다 탐색 영역의 수렴 속도를 다르게 하여 TSS에서의 계산량을 줄였다. 그리고 [13]에 제안된 알고리즘은 블럭의 DBD 크기에 비례하여 탐색 영역의 크기를 결정 한다. 위 기법은 적응적 탐색 영역 조정(adaptive adjustment of search window:AASW)라고 불리며, 탐색 원점(search origin)을 결정하기 위하여 공간적으로 인접한 블럭들의 움직임 상관관계를 이용하며 블럭의 움직임 정도에 따라서 탐색 영역을 조정한다. AASW 기법은 크게 3 단계-(1)새로운 탐색 원점 설정, (2) 탐색 영역의 크기 결정, 그리고 (3) 프레임별로 움직임에 따른 블럭을 구분하기 위한 임계값 갱신-로 수행된다. 탐색 원점은 인과 관계에 있는 인접한 블럭들-왼쪽, 위쪽, 위오른쪽 방향의 블럭-의 움직임 벡터와 0 변위된 블럭을 이용하여 현재 블럭

의 MV를 예측한다. 그리고 위 후보 벡터들 중 최소의 DBD를 갖는 블록의 MV가 현재 블록의 움직임 벡터로 선택된다. 결정된 초기 예측 MV에 의하여 설정된 위치가 FSBMA를 위한 탐색 원점으로서 사용된다. 탐색 원점을 설정한 후, 탐색 영역의 크기를 새로운 탐색 원점에서의 DBD 크기를 고려하여 결정한다. 즉, DBD를 현재 블록의 움직임 정도를 가리키는 척도로 사용한다. AASW 기법에서는 DBD를 블록의 움직임 정도를 구분하기 위한 기준으로 사용하였다. 움직임 정도에 따라서 블록을 3 부류 - low-motion block, medium-motion block, high-motion block - 로 나누어 각각의 움직임 추정을 위한 탐색 영역의 크기를 결정한다. 각 블록의 종류에 따라서 움직임 영역의 최대 변위는 high-motion 블록을 위하여  $w$ , medium-motion 블록을 위하여  $w/2$ , 그리고 low-motion 블록을 위하여  $w/4$ 로 각각 설정하였다. AASW 기법의 탐색 원점, 탐색 영역의 크기, 그리고 임계값 갱신 등은 그림 1에 간략히 기술하였다.

그러나 Feng 등이 제안한 AASW 기법은 몇가지 문제점을 가지고 있다. 첫째로, AASW는 MV의 크기가 DBD의 크기와 비례한다고 가정하였다. 즉, 정합 오차가 가장 적은 블록이 탐색 원점으로부터 멀리 떨어질수록 DBD 값은 비례하여 커진다는 것이다. 그러나 실제 실험 비디오 영상들을 가지고 분석하였을 경우,

DBD 값의 크기와 MV의 크기와는 큰 상관 관계가 없음을 알 수 있다(그림 2 참조).

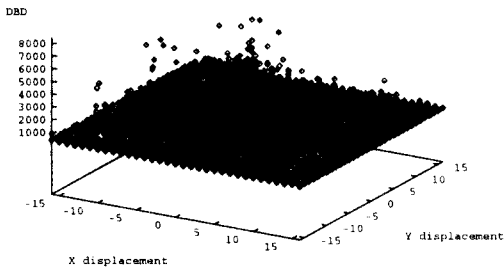
그림 2에서 화상 전하나 화상 회의와 같은 낮은 비트율을 갖는 응용 분야에 사용되는 비디오 영상은 움직임이 완만하고 부드러우며, 그리고 천천히 변한다. 그러므로, MV가 탐색 영역 내에 균일하게 분포하기 보다는 탐색 원점을 중심으로 분포하며, DBD 값의 크기와 MV의 크기는 큰 상관성이 없다는 것을 알 수 있다. 그래서, BMA에서 새로운 탐색 원점을 최소 DBD를 갖는 블록의 움직임 벡터로 설정할 경우 지역적인 최소값에 빠질 수 있다. 위에 언급한 저전송율 응용에 사용되는 비디오 영상들의 특성을 고려하면 BMA에서 계산량을 줄일 수 있다.

본 논문에서는 FSBMA의 계산량과 빠른 탐색 알고리즘의 극복 최소치에 수렴하는 문제점을 줄이기 위하여 동적 탐색 영역 조정(dynamic adjustment of search window: DASW)를 제안한다. DASW는 프레임 내에 존재하는 움직임 객체들의 움직임 구조를 이용하기 때문에 정합 블록 수를 줄일 수 있다. DASW 기법은 크게 (1) 새로운 탐색 원점 설정, (2) 탐색 영역의 크기 결정, 그리고 (3) 사용된 임계값의 동적인 갱신 과정을 거쳐서 수행된다. 본 논문에서는 프레임 내에 존재하는 움직임 객체를 최대한 이용하기 위하여 2가지 탐색 영역 조정 기법을 제안한다. 첫째, 움

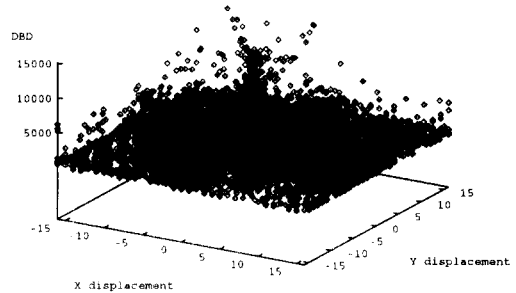
**알고리즘 : 적응적 탐색 영역 조정(Adaptive adjustment of search window:AASW)**

- 단계 1 : 첫 프레임에 대하여 각 블록의 MV를 전역 탐색을 통하여 찾는다.
- 단계 2 : 각 블록의 DBD 구하고, 이들을 내림 차순으로 정렬한다.
- 단계 3 : MV의 크기가  $w/2$ ,  $w/4$ 인 블록들의 수를 계산하고 이들 갯수를 각각  $N_a$ ,  $N_b$ 로 둔다.
- Step 4 : 임계값  $T_1$ 을  $N_a$  번째 큰 DBD로, 임계값  $T_2$ 를  $N_b$  번째 큰 DBD 값으로 설정한다.
- 단계 5 : 블록의 0 변위된 DBD와 인접한 블록의 MV를 이용한 DBD를 구하고, 이들 중 가장 작은 것을 현재 블록의 DBD로 하여, DBD의 값과 임계값을 비교하여 블록의 움직임 정도를 결정한다. 즉, 탐색 영역의 크기를 high-motion(if  $DBD > T_1$ ), medium-motion(if  $T_1 \geq DBD > T_2$ ), low-motion(if  $DBD \leq T_2$ )블록들을 위해 각각  $w$ ,  $w/2$ ,  $w/4$ 로 설정한다. 여기서  $w$ 는 초기 설정치이다.
- 단계 6 : 단계 5에서 결정된 탐색 영역 내에서 BMA를 수행한다.
- 단계 7 : 영상의 움직임의 변화에 따라 적응적으로 임계값  $T_1, T_2$ 를 갱신한다.

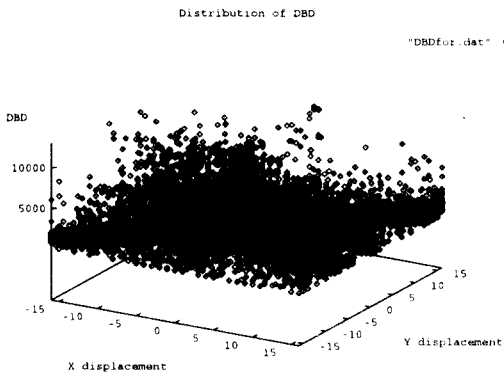
그림 1. Feng, et al.[13]에 의하여 제안된 적응적 탐색 영역 조정 기법



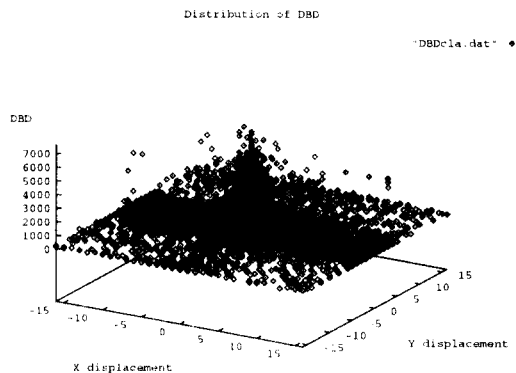
(a) Miss America



(b) Carphone



(c)Foreman



(d) Claire

그림 2. 각 실험 영상에서 예측 오차가 가장 작은 블럭에 대한 MV의 x, y 성분의 크기와 DBD 분포

직접 추정을 하고자 하는 블럭을 기존 BMA와 같이 겹치지 않게 고정 크기 블럭으로 나누어 인접한 블럭들의 유사도를 이용하여 탐색 영역의 크기를 결정하는 고정 크기 블럭을 사용하는 동적 탐색 영역 조정 기법(DASW with fixed block size: DASWF)를 제안한다. 그리고 고정 크기 블럭을 사용하 경우 서로 다른 움직임 객체가 동일한 블럭에 포함될 수 있으며, 이 경우 정확한 움직임을 추정할 수 없어 큰 예측 오차를 유발한다. 이러한 단점을 극복하고, 가능하면 각 블럭에 동일 움직임 객체를 포함하도록 하는 가변 크기

블럭을 이용하는 동적 탐색 영역 조정 기법(DASW with variable block size: DASWV)를 제안한다.

본 논문의 구성은 다음과 같다. 제2장에서 현재 블럭과 인접한 블럭들과의 유사도를 이용하여 탐색 영역의 크기를 동적으로 결정하는 DASW를 기술하며, 고정 크기 블럭을 갖는 DASW(DASWF), 가변 크기 블럭을 갖는 DASW(DASWV)를 설명한다. 제3장에서는 탐색 영역을 조정하는 이전 연구와 제안한 방법들의 성능 평가를 화상 진화나 화상 회의에 사용되는 비디오 영상들을 가지고 수행한 결과를 기술한다.

## II. BMA를 위한 동적 탐색 영역 크기 조정

비디오 영상에는, 특히 영상 전화기나 영상 회의용 비디오 영상에서는 움직임이 완만하고 매우 느리게 변화한다. 만약 프레임 내에서 인접한 블록들이 동일한 움직임 객체를 포함한다면 MV들간의 상호 상관성은 매우 크게 된다. 프레임 내에서 움직임 객체는 여러 블록에 걸쳐 있을 수 있기 때문에 이러한 블록들간의 상호 상관성을 이용하면 탐색 영역의 크기를 줄일 수 있다. 또한 그림 2에 보는 바와 같이 저전송 매체용 화상 전화나 화상 회의 시스템용 비디오 영상들에는 많은 블록들의 MV가 탐색 원점 부근에 분포함을 볼 수 있다.

위에 언급한 저전송용 응용에 사용되는 비디오 영상들의 특성을 고려하여 본 논문에서는 동적 탐색 영역 조정(dynamic adjustment of search window:DASW) 기법을 제안한다. DASW는 프레임 내에 존재하는 움직임 객체들의 움직임 구조를 이용하기 때문에 정한 블록 수를 줄일 수 있다. DASW 기법은 크게 (1) 새로운 탐색 원점 설정, (2) 탐색 영역의 크기 결정, 그리

고 (3) 사용된 임계값의 동적인 갱신 과정을 거쳐서 수행된다. 본 논문에서는 프레임 내에 존재하는 움직임 객체를 최대한 이용하기 위하여 2가지 탐색 영역 조정 기법을 제안한다. 첫째, 움직임 추정을 하고자 하는 블록을 기존 BMA와 같이 겹치지 않게 고정 크기 블록으로 나누어 인접한 블록들의 유사도를 이용하여 탐색 영역의 크기를 결정하는 고정 크기 블록을 사용하는 동적 탐색 영역 조정 기법(DASW with fixed block size:DASWF)를 제안한다. 그리고 고정 크기 블록을 사용한 경우 서로 다른 움직임 객체가 동일한 블록에 포함될 수 있으며, 이 경우 정확한 움직임을 추정할 수 없어 큰 예측 오차를 유발한다. 이러한 단점을 극복하고, 가능하면 각 블록에 동일 움직임 객체를 포함하도록 하는 가변 크기 블록을 이용하는 동적 탐색 영역 조정 기법(DASW with variable block size:DASWV)를 제안한다.

### 2.1 고정 크기 블록을 사용하는 동적 탐색 영역 조정 기법 : DASWF

DASWF 기법은 블록 유사도(block similarity)와 인

#### 알고리즘 : 고정 크기 블록을 이용하는 동적 탐색 영역 조정:DASWF

단계 1 : 현재의 프레임의 영역 분할 알고리즘을 이용하여 동일한 특성을 지니는 부분으로 분할한다.

단계 2 : 주어진 블록에 대하여, 현재 블록과 인과 관계에 있는 4개의 블록들(위-왼쪽, 위, 위-오른쪽, 왼쪽 블록)과의 블록 유사도를 계산한다. 블록 유사도는 세그멘테이션 정보를 이용하여, 블록의 화소에 대하여 두 블록에서 같은 영역에 속하는 화소 수의 비로 계산된다.

단계 3 : 현재 블록과 참조 프레임에서 0 변위된 블록과의 DBD( $DBD_{(0,0)}$ )와 가장 큰 블록 유사도를 갖는 인접한 블록의 MV 만큼 변위된 블록과의 DBD( $DBD_{adj}$ )를 구한다.

단계 4 : 새로운 탐색 원점을 단계 3의 두 DBD 중 작은 값을 갖는 블록의 변위로 설정한다.

단계 5 : 탐색 영역의 크기를 다음에 따라 결정한다.

단계 5.1 : 새로운 탐색 원점이 인접한 블록의 MV 만큼 변위된 위치인 경우( $DBD_{(0,0)} > DBD_{adj}$ ), 탐색 영역의 크기는 블록 유사도에 반비례하게 결정된다.

단계 5.2 : 새로운 탐색 원점이 0 변위된 위치인 경우( $DBD_{(0,0)} \leq DBD_{adj}$ ), 탐색 영역의 크기는 (block similarity)  $\times$  Max{x,y| MV component of the adjacent block } + w  $\times$  (1 - (block similarity)) 로 결정되며, 여기서 w 초기 최대 변위이다.

단계 6 : 결정된 탐색 영역의 크기 내에서 BMA를 수행한다.

단계 7 : 계산량과 영상의 특성을 고려하여 임계값  $T_{seg}$ 를 갱신한다.

그림 3. 인접한 블록과의 유사도와 DBD를 이용하여 동적으로 탐색 영역 크기를 결정하는 고정 크기 블록을 갖는 DASWF 알고리즘

접한 블럭의 MV를 이용한 DBD를 기반으로 탐색 영역의 크기를 결정한다. 그림 3에 간략히 알고리즘을 기술하였다. 연속한 비디오 프레임에서 각 블럭의 움직임 변위는 움직임 객체와 매우 밀접한 관계가 있기 때문에, 만약 인접한 블럭들이 같은 객체의 영역에 포함된다면 그들은 유사한 MV와 DBD를 가진다. DASWF에서는 움직임 객체들의 구조를 이용하여, 탐색 영역의 크기를 동적으로 조정한다. 사용된 인접한 블럭의 유사도는 입력 영상을 동일한 영역으로 분할한 세그멘테이션 정보를 이용하여 결정한다. 다음의 각 절은 DASWF 기법을 구체적으로 설명한다.

**새로운 원점 설정:** DASWF에서는 탐색 영역의 크기를 조정하기 위해서 현재 블럭과 인접한 블럭들 사이의 블럭 유사도를 고려한다. 비디오 영상에는 인접한 블럭들이 같은 움직임 객체에 포함된다면 각 블럭들의 움직임 구조가 매우 비슷하다. 그러므로, 현재 블럭의 초기 MV를 현재 블럭과 가장 밀접하게 관련된 블럭의 MV로 설정하고, 이곳으로부터 작은 움직임 변위 내에서 MV를 세부적으로 찾을 수 있다.

인접한 블럭간의 블럭 유사도를 결정하기 위해서 DASWF에서는 입력 프레임 전체를 동일한 특성을 갖는 영역끼리 나누는 세그멘테이션 정보를 이용한다. 일반적으로 영역 분할은 프레임 내의 밝기, 텍스처 등과 같은 특성을 고려하여 프레임을 비슷한 특성을 지니는 영역으로 나누는 것으로 서로 다른 영역은 몇 가지 서로 다른 특성을 지니게 된다. 이러한 특성들은 서로 다른 영역으로 나누는 기준이 되며, 영역을 분할할 경우 동일한 특성을 지니는 영역인가를 검사하는 데 사용된다. 많은 영역 분할 알고리즘들이 연구 개발되어 왔으며, [19]에 잘 요약되어 있다.

DASWF에서는 계산량을 고려하여 간단한 영역 분할 알고리즘(region splitting algorithm)을 사용한다. 영역 분할 알고리즘은 탑-다운(top-down) 접근법으로 영상의 전체 프레임이 하나의 동일한 특성을 지니는 영역으로 가정하고 시작한다. 영역 분할 절차에서 주어진 영역의 전체가 동일한 특성을 가지지 않을 경우 4개의 부영상(subimage)로 나누고 각각의 부영상에 대하여 위와 같은 절차를 반복해서 한 영역 내의 동질성이 같을 때

멈춘다. 이 과정에서 동질성을 갖는가에 대한 검사는 영역에 포함된 화소들의 차이가 주어진 임계값  $T_{seg}$ 와 비교하여 임계값보다 큰 경우 영역을 분할하고, 그렇지 않으면 멈춘다.

DASWF 기법에서는 세그멘테이션 정보와 함께 인접한 블럭들의 MV와 이들의 DBD를 이용하여 새로운 탐색 원점을 설정한다. 먼저, 새로운 탐색 원점은 가로/세로로 0 만큼 변위된 위치에서의 DBD값과 인접한 블럭중 현재 블럭과 블럭 유사도가 가장 큰 블럭의 MV 만큼 변위된 위치에서의 DBD값을 비교하여 작은 DBD 값을 가지는 위치를 새로운 탐색 원점으로 한다. 현재 블럭과의 비교되는 인접한 블럭들은 스캐닝 순서에서 인과 관계에 있는 블럭들로 위-왼쪽, 위, 위-오른쪽, 왼쪽 블럭들이다. 영상의 경계에 현재 블럭이 위치할 경우 비교 가능한 인접한 블럭만을 고려한다. DASWF에 사용된 블럭 유사도는 전체 블럭의 화소 수에 대한 두 블럭의 화소들이 세그멘테이션 정보에서 같은 영역에 포함되는 화소 수의 비로 계산된다. 즉, 블럭 유사도는 0에서 1 사이의 범위를 가지며, 블럭 유사도가 1이면 두 블럭은 동일한 영역에 포함되며 같은 움직임 객체나 같은 배경에 포함된다. 이 경우, 인과 관계에 있는 인접한 블럭의 MV를 이용하여 새로운 탐색 원점을 설정하고 이로부터 작은 범위 내의 탐색 영역에서 세부적으로 현 블럭의 MV를 찾으면 많은 계산량을 줄일 수 있다. 블럭 유사도가 0에 가까운 값일 수록 인접한 블럭들의 정보를 이용할 수 없으므로 탐색 영역은 큰 범위를 가져야 한다.

새로운 원점 설정 과정에서 현재 블럭과 참조 프레임에서 0 변위 블럭과의 DBD와 인접한 블럭의 MV 만큼 변위된 DBD 중 작은 쪽이 새로운 탐색 원점이 된다. 즉,  $DBD_{(0,0)}$ 를 0 변위된 위치의 DBD라 하고,  $DBD_{adj}$ 를 가장 큰 블럭 유사도를 갖는 인접한 블럭의 MV 만큼 변위된 위치에서 DBD라고 하면, 만약,  $DBD_{(0,0)}$ 가  $DBD_{adj}$ 보다 작다면 0 변위된 위치가 새로운 탐색 원점으로 설정되며, 그렇지 않으면 인접한 블럭의 MV가 가리키는 곳이 원점이 되어 이곳으로부터 BMA를 수행하게 된다.

제안한 DASWF는 움직임 객체의 움직임 방향을 이용하기 때문에 여러가지 빠른 탐색 알고리즘들이 갖는 국부 최소치에 빠지는 문제점을 줄일 수 있다. 만약

인접한 블럭들이 현재 블럭과 동일한 객체를 포함하고 있다면, 인접한 블럭들 중에서 그 객체를 가장 많이 포함하는 블럭의 MV를 이용하여 탐색 원점을 설정하고, 그곳에서 탐색 영역의 크기를 결정한다. 이와 같이 인접한 블럭의 유사성을 이용하며, 탐색 영역은 크게 줄일 수 있다. 즉, 움직임 방향의 반대 쪽에 해당되는 탐색 영역의 후보들은 정합을 할 필요가 없다.

**탐색 영역의 크기 조정**: 새로운 탐색 원점을 설정한 후, 블럭 유사도와 인접한 블럭의 MV를 이용하여 각 블럭의 탐색 영역의 크기를 결정한다. 탐색 영역의 크기는 그림 3의 단계 5에서 결정된다. MV의 최대 변위는 단계 4에서 새로운 탐색 원점이 설정되는 방법에 따라 다르게 조정된다.

새로운 탐색 원점이 인접한 블럭의 MV가 가리키는 위치로 설정된다면 ( $DBD_{adj} < DBD_{(0,0)}$ 인 경우), MV의 최대 변위는 단지 블럭 유사도만을 고려하여 결정된다. DASWF에서 새로운 탐색 원점이 어느 정도 정확하게 설정되었는가를 블럭 유사도를 이용하여 식 (1)과 같이 각 블럭의 예측 정도를 구분한다.

The block is =

$$\begin{cases} \text{poor-predicted} & \text{if block similarity} \leq T_{low} \\ \text{medium-predicted} & \text{if } T_{low} < \text{block similarity} \leq T_{high} \\ \text{well-predicted} & \text{if block similarity} > T_{high} \end{cases} \quad (1)$$

여기서  $T_{low}$ ,  $T_{high}$ 는 블럭을 구분하기 위한 임계값이다. 각 블럭의 최대 변위를 poor-predicted 블럭을 위해서  $w$ , medium-predicted 블럭을 위해서  $w/2$ , 그리고 well-predicted 블럭을 위해서  $w/4$ 로 설정한다.  $w$ 는 초기 변위이다. Well-predicted 블럭에 대하여, 현재 블럭은 인접한 블럭과 매우 유사한 특성을 지니며, 동일한 움직임 객체 내에 포함될 확률이 높으며, 그들의 움직임 변위 또한 비슷하다. 그래서 가장 잘 정합이 이루어지는 후보 블럭은 새로운 탐색 원점 부근에 존재한다. 이러한 사실로부터 탐색 영역의 크기를 줄여서 계산량을 감소 시킬 수 있다. Poor-predicted 블럭에 대하여서는 반대의 결과를 얻을 수 있다.

두번째로, 만약 새로운 탐색 원점이 0 변위로 설정된 경우 ( $DBD_{(0,0)} \leq DBD_{arj}$ 인 경우) 탐색 영역의 최대 변위는 식 (2)와 같이 조정된다.

$$\begin{aligned} \text{displacement} = & (\text{block similarity}) \\ & \times \text{Max}\{x, y | \text{MV component of the adjacent block}\} \\ & + w \times (1 - \text{block similarity}) \end{aligned} \quad (2)$$

이 경우, MV는 0와  $w$  사이에 존재하며,  $(x, y)$ 는 현재 블럭과 가장 큰 블럭 유사도를 갖는 인접한 블럭 MV의 x축과 y축의 변위이다.  $DBD_{(0,0)} \leq DBD_{arj}$ 은 가장 정합이 잘 이루어지는 블럭이 초기 탐색 원점 부근에 존재함을 의미한다.

블럭 유사도는 인접한 두 블럭들이 같은 움직임 구조를 가지는가를 보여주며, 각 블럭의 변위 또한 블럭 유사도에 비례함을 보여 준다. 각 블럭에 대하여 탐색 영역의 크기가 결정되면, 기존의 BMA를 수행한다. 제안한 기법은 TSS, NTSS, 2D-LOG 등과 같은 빠른 탐색 기법에도 쉽게 적용될 수 있다.

**임계값의 갱신**: DASWF 알고리즘에는 탐색 영역 크기 조정을 위해서 블럭을 분류하기 위한 2개의 임계값  $T_{low}$ ,  $T_{high}$ 와 입력 프레임들 동일한 특성을 지니는 부분으로 분할하는데 필요한 임계값  $T_{seg}$ 가 사용된다. 3개의 임계값 모두 제안한 알고리즘의 성능과 계산량에 밀접한 관계가 있다. 임계값  $T_{low}$ 는 블럭의 새로운 탐색 원점이 poor-predicted 또는 medium-predicted 되었는가를 결정하는데 사용되며, 임계값  $T_{high}$ 는 블럭의 새로운 탐색 원점이 medium-predicted 또는 well-predicted 되었는가를 분류하는데 사용된다. DASWF에서는  $T_{low}$ 와  $T_{high}$  각각 30%와 70%으로 고정시켜서 사용한다. 이들 임계값은 여러가지 실험 영상들을 대상으로 하여 프레임 내에서 인접한 두 블럭들의 MV의 차이와 블럭 유사도 사이의 관계를 구하여 이로부터 새로운 탐색 원점이 어느 정도로 정확하게 예측되는가를 고려하여 설정된다.

비디오 영상의 분할을 위해서 사용되는 임계값  $T_{seg}$ 는 계산량(정합을 하는 후보 블럭의 수)을 고려하여 동적으로 변한다. 만약 계산량이 예상한 블럭 정합 수보다 클 경우 임계값을 증가된 계산량에 비례하여 식 (3)과 같이 변화시킨다. 계산량이 예측 값보다 클 경우  $T_{seg}$ 를 증가시켜 프레임들 동일한 영역으로 영역 분할 기법을 사용하여 분할할 경우 동일한 영역의 범위를 크게 함으로 탐색 영역 조정에서 탐색 영역의 범위를 줄일 수 있어 블럭 정합 수를 줄일 수 있다.

$$T_{seg}^n = T_{seg}^{n-1} + \bar{M} \times \frac{(\text{Number of matching blocks}) - (\text{Number of the expected matching blocks})}{(\text{Number of the expected matching blocks})} \quad (3)$$

여기서  $\bar{M}$ 는 주어진 프레임의 평균이다. 임계값  $T_{seg}$ 의 초기값은  $\bar{M}$ 의 반으로 설정하였다. 위의 임계값 갱신 방법을 이용하여 대략적으로 원하는 계산량 내에서 MV를 구할 수 있다.

2.2 가변 크기 블럭을 이용한 동적 탐색 영역 조정 : DASWV

DASWV 기법은 DASWF 기법에서 한 블럭에 여러 개의 움직임 객체가 일부분 포함되는 경우 움직임을 정확히 예측을 할 수 없기 때문에 이러한 문제점을 극복하기 위해서 제안한다. DASWV에서 탐색 영역의 크기를 결정하는 방법은 DASWF에서와 같이 블럭 유사도와 인접한 블럭들의 상관 관계를 이용한다. DASWV 알고리즘은 그림 4에 기술한 바와 같이 크게 가변 블럭으로 나누기, 새로운 탐색 원점 설정 및 탐색 영역의 크기 조정, 그리고 임계값 갱신의 과정을 거친다.

가변 크기 블럭으로 나누기 : DASWV에서는 주어진 프레임을 가변 크기 블럭으로 나눈다. 주어진 프레임을 가변 크기 블럭으로 나누기 위해서 DASWF에서와 같이 세그멘테이션 정보를 이용한다. 블럭을 나누기 전에 먼저 영역 분할 알고리즘을 사용하여 동질성을 갖는 영역으로 분할 한 후 이를 기반으로 가변 크기 블럭으로 나눈다. 본 논문에서 사용된 가변 크기 블럭은 최대  $32 \times 32$ 에서 최소  $8 \times 8$ 이며, 탑-다운 방식으로 나누어 분할 정보를 쿼드-트리(quad-tree)로 표현한다. 주어진 블럭을 작은 4개의 부분블럭으로 나눌 것인지는 블럭의 동질성(homogeneity) 값이 주어진 임계값  $T_{homo}$ 과 비교하여 결정한다. 즉 블럭의 동질성 값이 임계값  $T_{homo}$ 보다 큰 경우 블럭을 나누지 않으며, 작은 경우 블럭이 여러 영역을 포함할 수 있기 때문에 4개의 부분블럭으로 나눈다. 그리고 각 블럭에 동질성 값을 할당한다. 동질성 값은 블럭의 총 화소 수에 대하여 세그멘테이션 정보에서 동일한 영역에 포함되는 화소 수의 비이다. 가변 크기 블럭의 정보를 표현하기 위해서 쿼드-트리를 이용하며, 쿼드-트리의 각 노드에서 분할이 일어나면 '1'을, 더 이상 분할이 일어나지 않는 경우 '0'를 할당한다.

알고리즘 : 가변 크기 블럭을 사용하는 동적 탐색 영역 조정 : DASWV

단계 1 : 현재 프레임을 영역 분할 알고리즘을 사용하여 세그멘테이션을 수행한다.

단계 2 : 세그멘테이션 정보를 이용하여 현재 프레임을 가변 크기 블럭으로 나눈다. 주어진 블럭이 동질성 임계값  $T_{homo}$  보다 큰 경우 분할을 멈추고 그렇지 않은 경우, 주어진 블럭을 4개의 부영상으로 나누어 각 블럭에 대하여 동질성을 검사한 후 위 과정을 반복한다. 여기서, 블럭의 크기는 최대  $32 \times 32$ 에서 최소  $8 \times 8$ 으로 한다. 동질성 값은 주어진 블럭의 총 화소에 대하여 세그멘테이션 정보에서 같은 영역에 포함되는 화소의 비를 의미한다. 나누어진 각 블럭에 동질성 값을 부여한다.

단계 3 각 블럭을 동질성 값의 내림 차순으로 정렬한다.

단계 4: 각 블럭에 대하여,  
현재 블럭의 모든 인접한 블럭에 대한 움직임 추정이 이루어 지지 않았으며, 그 블럭의 탐색 원점을 중심으로 초기에 설정된 탐색 영역의 크기  $w$ 를 가지고 BMA를 수행한다. 인접한 블럭이 이미 움직임 추정이 수행되었다면, 현 블럭과 인접한 블럭의 블럭 유사도를 비교하여 가장 유사한 인접한 블럭의 MV 만큼 변위된 위치의 DBD와 0 변위된 위치에서의 DBD를 계산하여 새로운 원점을 설정하고 탐색 영역의 크기는 DASWF에서와 같이 수행한다. 탐색 원점과 영역 크기가 결정된 후 BMA를 수행한다.

단계 5: 임계값  $T_{seg}$ 를 갱신한다.

그림 4. 가변 크기 블럭을 이용하는 DASW 알고리즘



**새로운 탐색 원점 설정 및 탐색 영역의 크기 결정 :** 전체 프레임을 가변 크기 블록으로 나눈 후, 각 블록에 부여된 동질성 값을 내림 차순으로 정렬한다. 먼저 동질성 값이 가장 큰 블록부터 움직임 추정을 시작한다. 첫 블록에 대하여 그 블록의 탐색 원점을 중심으로 초기에 설정된 탐색 영역의 크기  $w$ 로 탐색을 수행한다. 그 이후의 블록에 대하여서는 인접한 모든 블록(방향 제한 없음)에 대하여 이미 움직임 추정을 수행한 블록들이 존재하면 그 블록들과 현재 블록과의 블록 유사도와 MV를 고려하여 DASWF와 같은 방법으로 새로운 탐색 원점 및 탐색 영역의 크기가 설정되며, 움직임 추정이 끝난 인접한 블록이 없을 경우 첫 블록과 같은 방법으로 움직임 추정을 수행한다.

DASWV에서는 움직임 객체를 포함하는 대표적인 블록들(동질성 값이 큰 블록들)을 먼저 움직임 추정을 한 후 이와 인접한 블록들에 대하여 먼저 찾아진 블록들의 MV와 DBD를 그리고 블록 유사도를 이용하여 탐색 영역의 범위를 줄일 수 있다.

**임계값 갱신 :** DASWV에 사용된 임계값은 DASWF에 사용된 임계값과 함께 가변 크기 블록으로 나눌 때 사용된 블록의 동질성 임계값  $T_{\text{homo}}$ 이다. 임계값  $T_{\text{homo}}$ 은 가변 크기로 나눌 때 프레임당 블록의 수와 밀접한 관계가 있다. 임계값이 클 수록 블록의 수는 많아지며, 각 블록은 같은 영역을 포함하게 된다. 반면에 임계값이 작아질 수록 하나의 블록에 여러개의 움직임 필드를 포함할 수 있다. 본 논문에서는  $T_{\text{homo}}$ 을 70%로 설정하였다.

### III. 실험 결과 및 분석

본 실험에서는 동적 탐색 영역 크기를 조정하는 DASWF와 DASWV의 성능과 계산 복잡도를 분석하기 위하여 CIF(common immediate format:(352×288)), 30frames/s의 3개의 비디오 영상 - *Miss America*, *Carphone*, *Foreman* - 과 QCIF(quarter CIF:(176×144)), 30frames/s의 3개 비디오 영상 - *Miss America*, *Carphone*, *Susie* - 을 사용한다. 움직임 추정 및 보상은 영상의 밝기 성분(Y 성분)에만 적용하며, 결과는 밝기 성분에 대하여 각 기법들을 비교한다. 실험에 사용된 영상은 ITU-T의 H.263 비디오 압축 기법을 분석하기

위해 사용된 비디오 영상으로서, 주로 한 사람의 화자가 영상의 가운데 있으며, 움직임은 *Miss America*와 같은 매우 적은 것에서부터 *Carphone*, *Foreman*과 같은 어느 정도 움직임이 존재하는 화상 전화기 및 화상 회의용 영상이다. 움직임 추정에 사용된 정합 기준은 MAE(mean absolute error)이며, 고정 크기 블록을 이용한 움직임 추정의 경우 16×16의 블록 크기에 최대 MV의 범위 16을 사용하여 정합을 한다. 가변 크기 블록의 경우 최대 32×32에서 8×8 크기의 블록을 움직임 추정 단위로 사용한다. 탐색 영역 크기를 결정하는 기법들의 성능을 비교하기 위하여 MSE(mean square error) 성능과 계산량을 비교하기 위하여 각 블록당 평균 블록 정합 수(number of search points: NSP)를 사용한다. 각 실험 영상을 대상으로 하여 4가지 탐색 조정 기법-초기에 설정하는 방법(conv.), [13]에 제안된 AASW 방법, 그리고 본 논문에서 제안한 2가지 방법 DASWF와 DASWV-을 FS와 TSS, 그리고 NTSS에 적용하였을 경우 각 기법들의 MSE 성능과 NSP 계산량을 비교한다.

표 1과 표 2는 QCIF 영상에 대한 각 탐색 영역 조정 기법들이 FS, TSS, NTSS에 적용되었을 때와 CIF 영상에 대한 각 탐색 영역 조정 기법들이 FS, TSS, NTSS에 적용되었을 때 화소당 평균 MSE, 블록당 평균 탐색 점의 수, 프레임당 평균 블록 수의 결과를 보여준다. QCIF 영상, CIF 영상을 대상으로 각 기법들이 FS에 적용되었을 경우 제안한 DASWF와 DASWV는 FS-conv.에 비하여 계산량을 약 50% 정도 줄일 수 있었으며, 화소당 MSE 값은 유사하다. 고정 크기 블록을 사용하는 경우 즉, DASWF는 FS-conv.와 유사하지만 가변 크기 블록을 사용하는 DASWV는 경우에 따라서 FS-conv.보다 우수한 MSE 성능을 보이기도 한다. FS-AASW에 비교하여서는 제안한 2가지 방법이 계산량에서는 비슷한 성능을 보이지만 MSE 측면에서는 우수함을 보인다. 가변 크기 블록을 사용하는 DASWV는 가변 크기 블록의 구조를 위한 부가 정보를 포함해야 되므로 부가적인 비트를 필요로 한다. 그러나 고정 크기 블록을 사용하는 기법들에 비교하여 프레임당 평균 블록 수를 비교하면, 적은 블록으로 표현되기 때문에 블록들의 구조 정보를 위한 부가 비트는 고정 크기 블록들의 MV를 표현하기 위해서 사용된 비트들에 의하여 상쇄된다. 즉 DASWV

에서의 비트수(MV 정보와 블록 구조 표현을 위한 정보)는 conv., AASW, DASWF의 MV 표현을 위한 비트 수와 유사하다.

TSS에 각 기법들이 적용되었을 경우 실험 결과로부터 제안한 기법이 계산량에서는 conv.나 AASW 보다는 약 30% 줄일 수 있으며, MSE 값은 영상에 따라 조금씩 다르지만 타 기법에 비하여 매우 낮은 값을 보인다. DASWF에 비하여 DASWV는 더욱 정확한 움직임 구조를 이용하기 때문에 MSE 성능에서 우수함을 볼 수 있다.

NTSS에 대하여 각 알고리즘의 성능 향상은 TSS만큼 커지는 않다. NTSS 알고리즘은 저 전송율 응용 분야에 사용되는 비디오 영상의 MV들은 탐색 원점 부근에 많이 분포한다는 사실을 이용하여 개발되었기 때문에 TSS 기법에 비하여 우수한 성능을 보인다. 표에서 제안한 기법이 NTSS에 적용되었을 경우 계산량은 비슷하지만 약간의 MSE 성능이 우수함을 볼 수 있다.

여러가지 비디오 영상에 대한 결과로부터 비디오 영상내에 움직임 영역이 많을 경우 제안한 기법의 성

능이 더욱 우수하게 나타남을 알 수 있다. 즉 *Miss America* 영상에서의 성능 차이보다 *Carphone*에서의 성능 차이가 큼을 알 수 있다.

그러나 제안한 기법들은 동적으로 탐색 영역의 크기를 결정하기 위하여 세그멘테이션 알고리즘을 이용하므로 추가적인 계산량이 필요하다. 이 부분은 움직임 추정 전처리로 행해지며, 필요한 연산 수는 블록당  $N^2 \log N$ 로 탐색 범위  $w$ 가  $N$ 과 비슷할 경우 전역 탐색에 필요한 연산에 비하여 미약하기 때문에 표 1과 표 2의 결과에 포함시키지 않았다.

그림 5와 그림 6에는 QCIF의 *Susie*와 *Carphone*을, CIF의 *Miss America*와 *Carphone* 영상을 대상으로 프레임 변화에 따른 각 프레임의 화소별 평균 MSE 값을 도시하였다. 각 그래프로부터 제안한 기법이 MSE 성능 측면에서 우수함을 알 수 있다.

실험 결과로부터 제안한 기법이 기존의 기법에 비하여 많은 계산량을 줄일 수 있기 때문에 화상 전화나 화상 회의 시스템과 같이 낮은 복잡도를 가져야 하는 응용분야에 사용될 수 있음을 알 수 있다.

표 1. QCIF 실험 영상에 대하여 각 탐색 영역 조정 기법들을 움직임 추정 기법에 적용하였을 경우 화소당 평균 MSE, 블록당 평균 탐색점 수(NSP), 그리고 프레임당 평균 블록 수(NB) 비교

(가)FS에 적용하였을 경우

BMA method	Miss America			Susie			Carphone		
	MSE	NSP	NB	MSE	NSP	NB	MSE	NSP	NB
FSBNA-conv.	6.96	886	99	44.62	886	99	66.10	886	99
FSBMA-AASW	7.29	415	99	48.15	483	99	70.06	496	99
FSBMA-DASWF	7.02	376	99	44.76	462	99	67.24	498	99
FSBMA-DASWV	6.98	387	73	44.01	431	78	66.50	457	85

(나)TSS에 적용하였을 경우

BMA method	Miss America			Susie			Carphone		
	MSE	NSP	NB	MSE	NSP	NB	MSE	NSP	NB
TSS-conv.	16.69	28	99	65.29	29	99	92.54	29	99
TSS-AASW	13.17	32	99	58.52	32	99	87.61	32	99
TSS-DASWF	11.22	18	99	56.25	19	99	85.19	20	99
TSS-DASWV	10.91	21	74	54.38	24	79	82.67	23	84

(다) NTSS에 적용하였을 경우

BMA method	Miss America			Susie			Carphone		
	MSE	NSP	NB	MSE	NSP	NB	MSE	NSP	NB
NTSS-conv.	11.42	17	99	57.76	19	99	76.76	18	99
NTSS-AASW	11.56	21	99	58.22	21	99	76.73	20	99
NTSS-DASWF	9.42	17	99	53.87	18	99	71.92	18	99
NTSS-DASWV	9.13	18	73	51.85	19	78	69.41	21	84

표 2. CIF 실험 영상에 대하여 각 탐색 영역 조정 기법들을 움직임 추정 기법에 적용하였을 경우 화소당 평균 MSE, 블럭당 평균 탐색점 수(NSP), 그리고 프레임당 평균 블럭 수(NB) 비교

(가) FS에 적용하였을 경우

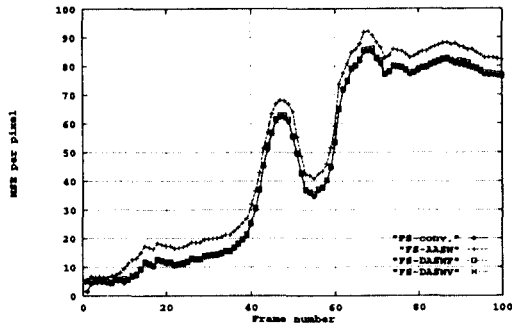
BMA method	Miss America			Foreman			Carphone		
	MSE	NSP	NB	MSE	NSP	NB	MSE	NSP	NB
FSBNA-conv.	7.18	984	396	38.58	984	396	27.97	984	396
FSBMA-AASW	7.78	432	396	42.86	536	396	31.59	243	396
FSBMA-DASWF	7.49	353	396	39.30	587	396	28.84	275	396
FSBMA-DASWV	7.01	372	357	36.71	547	386	25.41	286	378

(나) TSS에 적용하였을 경우

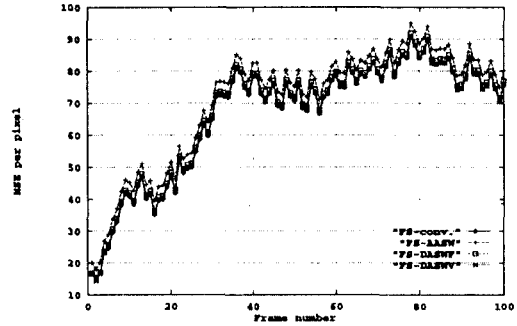
BMA method	Miss America			Foreman			Carphone		
	MSE	NSP	NB	MSE	NSP	NB	MSE	NSP	NB
TSS-conv.	8.39	31	396	55.06	31	396	37.18	31	396
TSS-AASW	8.61	25	396	57.13	21	396	38.19	23	396
TSS-DASWF	7.65	19	396	45.99	25	396	35.63	21	396
TSS-DASWV	7.31	20	364	42.13	23	392	33.73	22	381

(다) NTSS에 적용하였을 경우

BMA method	Miss America			Foreman			Carphone		
	MSE	NSP	NB	MSE	NSP	NB	MSE	NSP	NB
NTSS-conv.	7.54	22	396	54.79	21	396	33.69	22	396
NTSS-AASW	7.50	25	396	52.22	23	396	35.11	23	396
NTSS-DASWF	7.67	22	396	46.73	22	396	33.15	22	396
NTSS-DASWV	7.19	21	365	43.62	25	389	31.24	24	384

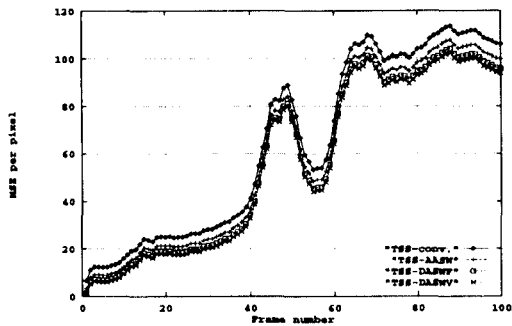


(a) Susie

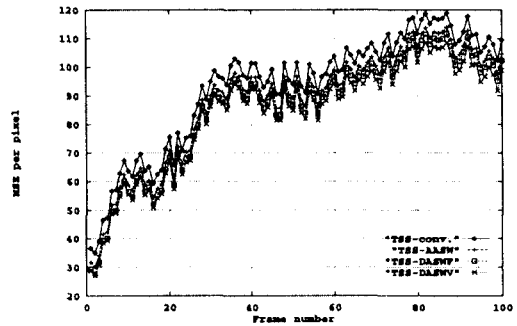


(b) Carphone

(가) FS에 적용하였을 경우

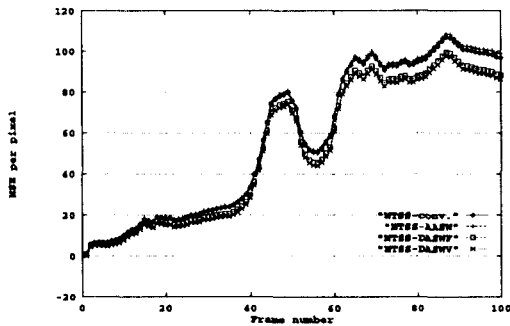


(a) Susie

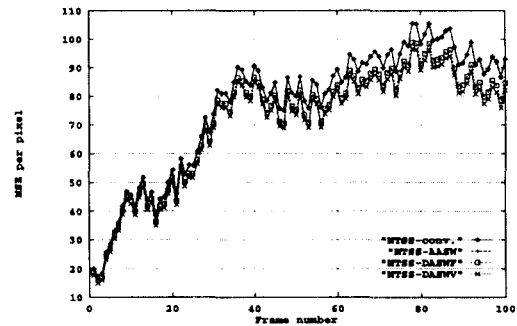


(b) Carphone

(나) TSS에 적용하였을 경우



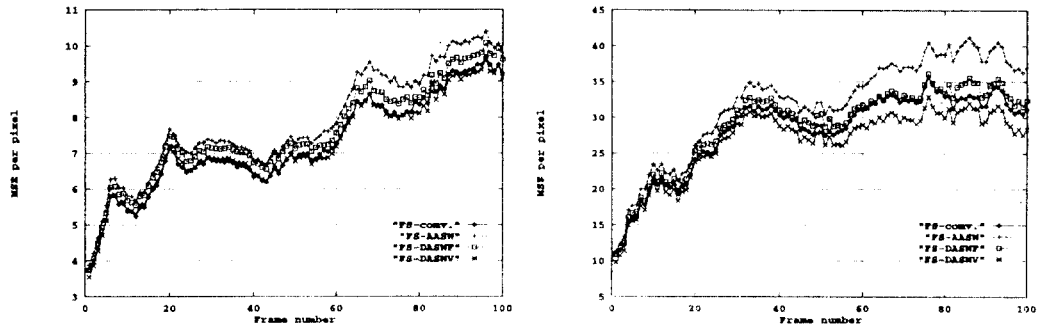
(a) Susie



(b) Carphone

(다) NTSS에 적용하였을 경우

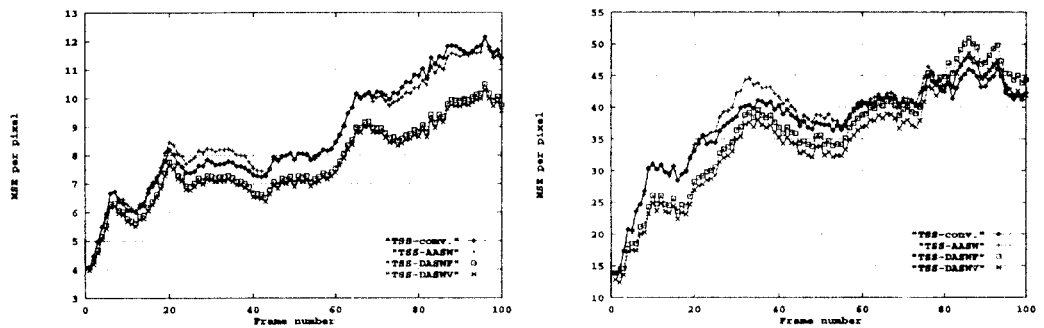
그림 5. QCIF 영상 *Susie*와 *Carphone* 영상에 대하여 각 기법을 적용하였을 경우 프레임 변화에 따른 각 화소당 MSE 성능



(a) Miss America

(b) Carphone

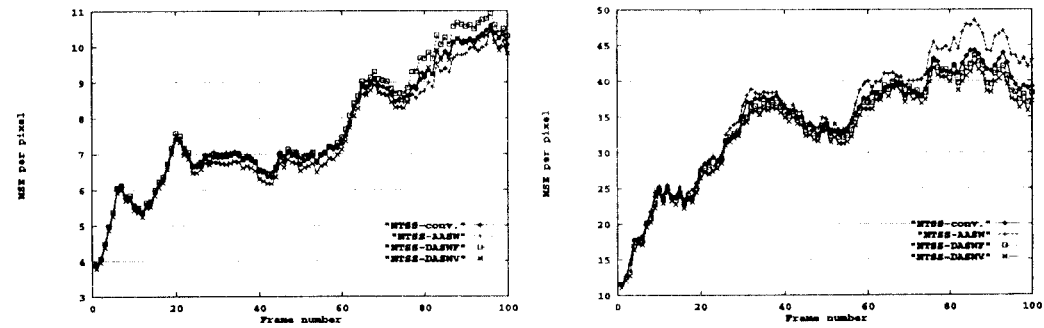
(가) FS에 적용하였을 경우



(a) Miss America

(b) Carphone

(나) TSS에 적용하였을 경우



(a) Miss America

(b) Carphone

(다) NTSS에 적용하였을 경우

그림 6. CIF 영상 *Miss America*와 *Carphone* 영상에 대하여 각 기법을 적용하였을 경우 프레임 변화에 따른 각 화소당 MSE 성능

#### IV. 결 론

본 논문에서는 FSBMA의 많은 계산량을 줄이며, 빠른 탐색 기법 알고리즘들의 지역적인 최소치에 빠지는 문제점을 해결하기 위하여 BMA를 위한 탐색 영역을 블록의 특성에 따라서 동적으로 조정하는 두 가지 기법을 제안하였다. DASWF는 비디오 영상의 각 프레임에 고정 크기 블록으로 나누고 각 블록의 스캐닝 순서에 따라서 움직임 추정을 수행한다. 이때 인접한 블록들과 블록 유사도를 계산하여 현재 블록의 초기 MV를 예측 및 탐색 영역의 크기를 결정한다. DASWV는 각 프레임 내에 존재하는 움직임 영역을 더욱 정확히 추정하기 위하여 쿼드 트리를 이용하여 가변 크기 블록으로 나누어 각 블록에 대하여 움직임을 추정한다. DASWV 알고리즘에서도 블록의 유사도를 이용하여 움직임 추정시 탐색 영역의 크기를 동적으로 결정한다. 제안한 동적 탐색 영역 조정 기법들은 전역 탐색 기법뿐 아니라 기존의 빠른 탐색 기법에도 쉽게 적용될 수 있다. 실험을 통하여 제안한 DASWF와 DASWV가 탐색 영역의 크기를 초기로 설정한 경우와 AASW 기법에 비교하여 MSE 성능과 탐색 계산량 측면에서 우수함을 보였다. 또한 실험을 통하여 제안한 기법들이 계산량을 줄일 수 있으며, 높은 화질을 유지하므로 화상 전화기나 화상 회의의 시스템과 같이 복잡도가 낮은 응용 분야에 사용될 수 있음을 보였다.

#### 참 고 문 헌

1. D. L. Gall, "MPEG: A Video Compression Standard for Multimedia Application," *Communication of the ACM*, 34(4), 46-58, (Apr. 1991).
2. M. Liou, "Overview of the P×64 kbit/s Video Coding Standard," *Communication of the ACM*, 34(4), 59-63, (Apr. 1991).
3. R. Schaphorst and Rapporteur, "Draft Recommendation H.263(Video coding for low bitrate communication)," *ITU-T SG15 WP15/1*, (Jul. 1995).
4. Y. Baek, H. -S. Oh, and H. -K Lee, "Block-matching criterion for efficient VLSI implementation," *Electronic Letters*, 32(13), 1184-1185, (Jun. 1996).
5. J. R. Jain, and A. K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Trans. On Communications*, Com-29 (12), 1799-1808, (Dec. 1981).
6. R. Li, B. Zeng, and M. L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans. On Circuits and Systems for Video Technology*, 4(4), 438-441, (Aug. 1994).
7. T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing," *Proc. NTC81*, G5.3.1-G5.3.5, (Dec. 1981).
8. R. Srinivasan and K. R. Rao, "Predictive Coding Based On Efficient Motion Estimation," *IEEE Trans. On Communications*, 33(8), 888-896, (Aug. 1985).
9. M. -J. Chen, L. -G. Chen and T. -D. Chiueh, "One-Dimensional Full Search Motion Estimation Algorithm for Video Coding," *IEEE Trans. On Circuits and Systems for Video Technology*, 4(5), 504-509, (Oct. 1994).
10. Ghanbari, "The Cross-Search Algorithm for Motion Estimation," *IEEE Trans. On Communications*, 38(7), 950-953, (Jul. 1990).
11. L. -G. Chen, W. -T. Chen, Y. -S. Jehng, and T. -D. Chiueh, "An Efficient Parallel Motion Estimation Algorithm for Digital Image Processing," *IEEE Trans. On Circuits and Systems for Video Technology*, 1(4), 378-385, (Dec. 1991).
12. L. -W. Lee, J. -F. Wang, J. -Y. Lee, and J. -D. Shie, "Dynamic Search-Window Adjustment and Interlaced Search for Block-Matching Algorithm," *IEEE Trans. On Circuits and Systems for Video Technology*, 3(1), 85-87, (Feb. 1993).
13. Feng, K. -T. Lo, H. Mehrpour and A.E. Karbo-wiak, "Adaptive Block Matching Motion Estimation Algorithm for Video Coding," *Electronics letters*, 32(18), 1542-1543, (Aug. 1995).
14. S. Kappagantula and K. R. Rao, "Motion Compensated Interframe Image Prediction," *IEEE Trans. On Communications*, Com-33(9), 1011-1015,

(Sep. 1985).

15. W. Tawbi, F. Horn, E. Horlait, and J.-B. Stefani, "Video Compression Standards and Quality of Service," *The Computer Journal*, **36**(1), 43-54, (Jan. 1993).
16. B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Trans. On Circuits and Systems for Video Technology*, **3**(2), 148-157, (Apr. 1993).
17. V. Seferidsi and M. Ghanbari, "General Approach to Block-Matching Motion Estimation," *Optical Engineering*, **32**(7), 1464-1474, (Jul. 1993).
18. ITU-T Recommendation H.261, "Video Codec For Audiovisual Services at P×64 kbit/s," (1993).
19. I. Pitas, "Digital Image Processing Algorithm" Prentice Hall International Ltd., 254-297, (1993).
20. B. -M. Wang, J. -C. Yen, and S. Chang, "Zero Waiting-Cycle Hierarchical Block Matching Algorithm and its Array Architectures," *IEEE Trans. On Circuits and Systems for Video Technology*, **4** (1), 18-28, (Feb. 1994).
21. E. Chan, A. A. Rodringuez, R. Gandhi, and S. Panchanathan, "Experiments On block-matching techniques for video coding," *ACM Multimedia Systems*, **2**(5), 228-241, (1994).
22. W. Li and E. Salari, "Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. Circuits and Systems for Video Technology*, **4**(1), 105-107, (Jan. 1995).
23. M. -J. Chen, L. -G. Chen, T. -D. Chiueh, and Y. -P. Lee, "A New Block-Matching Criterion for Motion Estimation and Its Implementation," *IEEE Trans. Circuits and Systems for Video Technology*, **5**(3), 231- 236, (Jun. 1995).
24. Y. -L. Chan and W. -C. Siu, "New Adaptive Pixel Decimation for Block Motion Vector Estimation," *IEEE Trans. On Circuits and Systems for Video Technology*, **6**(1), 113-118, (Feb. 1996).
25. L. -M. Po and W. -C. Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Trans. On Circuits and Systems for*

*Video Technology*, **6**(3), 313-317, (Jun. 1996).

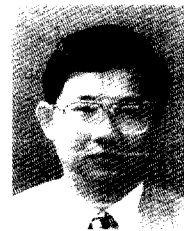
26. H. Jozawa, K. Kamikura, A. Sagta, H. Kotera, H. Watanabe, "Tow-Stage Motion Compensation Using Adaptive Global MC and Local Affine MC," *IEEE Trans. On Circuits and Systems for Video Technology*, **7**(1), 75-85, (Feb. 1997).



오 황 석(Hwang-Seok Oh) 정회원  
 1992년 2월 : 경북대학교 자연과학  
 대학 전자계산학과  
 졸업(학사)  
 1994년 2월 : 한국과학기술원 전산  
 학과 졸업(공학석사)  
 1994년 3월~현재 : 한국과학기술원  
 전산학과 박사과정

※주관심분야: video coding, parallel processing for  
 video coding, multimedia system

박 근 한(Gunhan Park) 정회원  
 1994년 2월 : 한국과학기술대학 전산학과 졸업(학사)  
 1996년 2월 : 한국과학기술원 전산학과 졸업(공학석사)  
 1996년 3월~현재 : 한국과학기술원 전산학과 박사 과정  
 ※주관심분야: multimedia system, content based re-  
 trieval system, video data processing



이 흥 규(Heung-Kyu Lee) 정회원  
 1978년 2월 : 서울대학교 전자공학  
 과 졸업(학사)  
 1981년 2월 : 한국과학원 전산학과  
 졸업(이학석사)  
 1984년 8월 : 한국과학기술원 전  
 산학과 졸업(공학박  
 사)

1985년 3월~1986년 8월 : U. of Michigan, U.S.A.(Rese-  
 arch Scientist)  
 1986년 9월~현재 : 한국과학기술원 전산학과 교수(부  
 교수)  
 1991년~1995년 12월 : 공업진흥청 JTC1/SC29(전문위원)  
 1994년 6월~1996년 12월 : 한국통신기술협회 ITU-T SC-9  
 VOD위원회(위원장)

※주관심분야: real-time processing, fault-tolerant sys-  
 tem, multimedia service system real-  
 time operating system