

논문 2006-43SP-1-9

적응형 윈도우 크기 기반 NTSS (New Three-Step Search Algorithm) 알고리즘 (A New Adaptive Window Size-based Three Step Search Scheme)

유 종 훈*, 오 승 준**, 안 창 범***, 박 호 중***

(Jonghoon Yu, Seoung-Jun Oh, Chang-bum Ahn, and Ho-Chong Park)

요 약

NTSS 알고리즘 (New Three-Step Search Algorithm)은 대표적인 고속 블록 정합 알고리즘(Block Matching Algorithm: BMA)의 하나인 TSS 알고리즘 (Three-Step Search Algorithm)에 동영상이 갖는 중앙 편향적(Center-Biased) 특성을 반영한 방법이다. 그러나 NTSS는 움직임이 작은 동영상인 경우에는 TSS보다 개선된 성능을 보여주지만, 움직임이 큰 동영상에 대해서는 TSS와 큰 차이가 없으며, 탐색범위가 커질수록 오히려 성능이 떨어지는 단점이 있다. 본 논문에서는 움직임 벡터의 특성에 따라 적응적으로 탐색범위를 결정하여 탐색범위의 증가로 발생하는 NTSS의 단점을 보완함으로써 움직임이 큰 동영상에 대해서도 향상된 성능을 갖는 방법을 제안한다. 제안한 방법을 적용하였을 때 움직임이 작은 동영상에서는 기존 NTSS 방법과 동일한 화질을 유지하면서 움직임이 큰 동영상에서는 최대 0.5dB 이상 성능이 개선되었다.

Abstract

With considering center-biased characteristic, NTSS(New Three-Step Search Algorithm) can improve the performance of TSS(Three-Step Search Algorithm) which is one of the most popular fast block matching algorithms(BMA) to search a motion vector in a video sequence. Although NTSS has generally better quality than TSS for a small motion sequence, it is hard to say that NTSS can provide better quality than TSS for a large motion sequence. It even deteriorates the quality to increase a search window size using NTSS. In order to address this drawback, this paper aims to develop a new adaptive window size-based three step search scheme, called AWTSS, which can improve quality at various window sizes in both the small and the large motion video sequences. In this scheme, the search window size is dynamically changed to improve coding efficiency according to the characteristic of motion vectors. AWTSS can improve the video quality more than 0.5dB in case of large motion with keeping the same quality in case of small motion.

Keywords : motion, AWTSS, NTSS, BMA, window

I. 서 론

* 학생회원, 광운대학교 전자공학과, VIA-Multimedia 센터

(Depart of Electronic Engineering, Kwangwoon University, VIA-Multimedia Center)

** 정회원, 광운대학교, VIA-Multimedia 센터
(Kwangwoon University, VIA-Multimedia Center)

*** 정회원, 광운대학교, VIA-Multimedia 센터
(Kwangwoon University, VIA-Multimedia Center)

※ 본 논문은 한국전자통신연구원(ETRI)와 한국 소프트웨어 진흥원 IT-SoC 핵심설계인력양성 사업과제로 수행되었음.

접수일자: 2005년6월12일, 수정완료일: 2005년11월25일

디지털 전송기술의 발달과 전송 매체의 다양화가 지속되면서 디지털 멀티미디어의 활용은 다양한 분야에 적용되고 있다. 특히 개인용 휴대 전화기나 카메라와 같은 이동형 단말기에서 실시간 영상 부호화 및 전송에 대한 필요성이 급증하고 있으며, 디지털 방송 서비스를 제공하기 위하여 다양한 비디오 영상 콘텐츠가 필요하기 때문에 디지털 비디오 영상에 대한 실시간 부호화가 절실히 요구되어지고 있다. 영상 압축은 디지털 비디오를 저장하거나 전송함에 있어 가장 중요한 기술이다^[1]. 따라서 동영상 압축에 대한 연구가 활발히 진행되어 왔으며 관련 표준들도 활발히 제정되었다. 대표적인 압축 표준으로는

ISO(International Standardization Organization)의 MPEG(Motion Picture Expert Group) 표준과 ITU-T (International Telecommunications Union -Telecommunication Standardization Sector)의 H.26X 표준이 있다^{[3]~[8]}.

동영상 압축 방식에서는 비디오 신호의 시간상의 중복성을 제거하여 비디오 데이터를 압축하기 위하여 움직임 보상 (Motion Compensation: MC) 방식을 사용한다^[9]. MC를 사용하기 위해서는 움직임 벡터 (Motion Vector: MV)를 예측하여야 한다. MV를 찾는 움직임 예측 (Motion Estimation: ME) 과정과 MC 과정을 통하여 동영상 신호가 갖는 시간적 중복성을 제거시킴으로써 화질 저하를 최소화 하면서 낮은 비트율로 부호화할 수 있다. ME를 수행하는 여러 가지 방법이 제안되었으며 블록 정합 알고리즘 (Block Matching Algorithm: BMA)이 간단하면서도 효과적이기 때문에 현재 비디오 코딩 표준에서 널리 사용되고 있다^[10].

BMA 중에서도 전역 탐색 (Full-Search: FS) BMA는 모든 탐색 점을 검색하기 때문에 최적의 해를 제공하지만 계산량이 매우 많으므로 실시간 비디오 코딩에는 적합하지 않다. FS 방법이 갖는 계산량을 줄이면서도 FS를 통해 얻는 화질에 가까운 성능을 제공하는 고속 BMA들이 개발되었다. 탐색점 수를 줄여서 계산량을 감소하려는 BMA 기반 ME 방법으로 3 단계 탐색법 (Three Step Search: TSS)^[11], 4 단계 탐색법 (Four Step Search: FSS)^[12], 십자형 구조 탐색법 (Cross Search: CS)^[13], 교차 다이아몬드 구조 탐색법 (Cross-diamond Search: CDS)^[14] 등이 있으며, 참고문헌 [15]와 같은 저 비트율을 기반으로 하는 방법이 있다. 그 중에서도 TSS는 계산량과 효율적인 측면에서 가장 대표적인 방법이다. 그러나 TSS는 첫 번째 단계의 탐색점간에 거리가 멀어 움직임이 큰 영상에 대한 전역 최소값 (Global minimum)을 찾기에 적합하지만 움직임이 작은 영상에 대해서는 성능이 떨어지는 단점이 있다.

이러한 TSS의 단점을 보완하기 위해 중앙 편향적 (Center-Biased) 특성, 즉 실제 비디오 영상에서 대부분의 움직임 벡터는 중심에서 크게 벗어나지 않는다는 점을 고려하여 움직임이 작은 영상에서도 잘 동작할 수 있도록 개선한 방법인 NTSS (New Three-Step Search)가 제안되었다^[16]. 그러나 NTSS는 움직임이 큰 영상에 대해서는 성능이 개선되지 않고, 오히려 움직임이 큰 영상에 적합한 탐색을 위하여 탐색 범위를 크게

할 경우 화질이 손실되는 문제가 있다. NTSS의 단점을 보완하기 위하여 W. Jing과 L. Chau는 ETSS를 제안하였다^[17]. ETSS는 NTSS와 같이 중앙 편향적 특성을 고려하면서도 움직임이 큰 영상인 경우 NTSS보다 좋은 화질을 보인다. 그러나 움직임 적은 영상에서는 NTSS에 비해 화질이 떨어진다.

본 논문에서는 부호화기에 입력되는 동영상의 움직임 정도를 판단하여 그 특성에 적응적으로 탐색 범위를 적용토록 함으로써 움직임이 큰 영상에 대해서도 향상된 성능을 갖는 방법을 제안한다. 탐색 범위 적용에 관한 대표적인 논문으로 Lee와 그의 동료들이 제안한 방법이 있다^[18]. 이 방법은 고정된 초기 탐색 범위에서 탐색을 수행하여 단계 별로 탐색 범위의 크기가 축소되는 점에서 기존 방법과 같지만, 각 단계에서 얻은 최소값의 정확도를 판단하여 다음 단계에서는 축소율을 다르게 적용함으로써 속도를 개선시켰다. 그러나 본 논문에서 제안하는 탐색 범위의 크기 조절은 한 매크로 블록 (Macro Block: MB)에 대한 ME 과정을 수행하기 전에 이전 MB의 MV를 참조하여 탐색 범위를 결정하기 때문에 탐색 초기에 탐색 범위의 크기를 움직임 특성에 적합하도록 적용되어 성능이 개선된다.

본 논문은 다음과 같이 구성된다. II장에서 NTSS와 ETSS를 간략하게 설명하고, 각각의 단점을 보완 할 수 있는 방법을 제안한다. III장에서는 주어진 영상에 대한 움직임의 크기를 판단하는 방법을 설명하고, 판단된 결과에 따라 다양한 크기의 탐색 범위를 적용하는 방법을 제안한다. IV장에서는 제안된 방법을 통한 실험 결과를 보여주고, V장에서 결론을 맺는다.

II. 기존 방법의 특성

1. NTSS 와 ETSS의 비교

NTSS는 다음과 같은 두 가지 측면에서 TSS와 다르다. 첫째는 탐색의 첫 번째 단계의 중앙 편향적인 특성을 고려하는 것이며, 두 번째 차이점은 정적인 특성을 갖는 블록에 대해 중도 정지 기법 (Halfway-Stop Technique)을 적용하는 것이다. 저 비트율 응용 분야에서는 일반적으로 그림 1과 같이 15x15 크기, 즉 $w=7$ 의 크기의 탐색 범위를 사용한다. 본 논문에서 사용하는 탐색영역은 정방형이며, 탐색을 표시하는 매개 변수로 w 를 사용한다. 탐색 영역은 수평 및 수직 방향으로 각각 $-w$ 에서 $+w$ 이며 $[-w, +w]$ 로 표기한다.

NTSS의 첫 번째 단계에서는 중앙 편향적인 특성을

고려하여 그림 1과 같이 TSS에 탐색 범위 중심의 3x3 격자위에 8개의 탐색점을 추가로 검색한다. 중도 정지 기법을 적용하여 탐색 범위의 중심 탐색점에서 최소 BDM (Block Distortion Measure) 값을 가지면 탐색을 중지하고, 최소 BDM 값이 중심의 3x3 격자 상에서 중심 탐색점을 제외한 8개의 탐색점 중 하나인 경우에는 얻어진 최소값의 위치에 따라 3개 또는 5개의 추가적인 점들을 탐색하여 최소점을 MV로 정한다. 첫 번째 단계에서 최소 BDM 값이 중심의 9x9 격자위에 있는 8개의 탐색점 중 하나이면 TSS와 동일한 탐색 과정을 수행한다.

그림 1은 NTSS를 사용하여 중심점에서 움직임 벡터를 찾는 경우에 대한 두 가지 상이한 탐색 경로를 보여 준다. 최소값을 갖는 점의 위치가 중심점에서 수직이나 수평 방향을 따라 4개의 이웃하는 점들 중 하나일 때 즉, 그림 1에서 우측 수평방향에 있을 때 사각형으로 표시된 3개의 추가적인 탐색점을 갖게 되며, 대각선 방향을 따르는 4개의 점들 중 하나일 때 즉, 그림 1에서 좌측 대각선 상단에 있을 때 삼각형으로 표시된 5개의 추가적인 탐색점을 갖는다. NTSS는 정적인 블록에서 17개의 탐색점이 필요하고, 중심의 5x5 영역 내에서의 작은 MV에 대해 20개 또는 22개의 탐색점이 필요하다. 그리고 가장 많은 탐색점을 갖는 경우의 탐색점 수는 $25+8=33$ 이므로 TSS에서 요구되는 25개와 비교할 때 8개의 탐색점이 더 요구된다. 이와 같이 17~33개의 탐색점을 갖는 조건은 탐색을 수행하는 프레임에 포함된 블록들의 움직임 특성에 좌우된다.

ETSS는 NTSS와 마찬가지로 중앙 편향적인 특성을 고려하였지만, NTSS와의 차이점은 초기 탐색 단계에서 중심점에 이웃하는 탐색점의 수가 4이고, 이 중 하나가 최소 BDM 값을 갖는 경우에 DS (Diamond Search) 탐색 과정을 밟는다. 9x9 격자위에 8개의 탐색

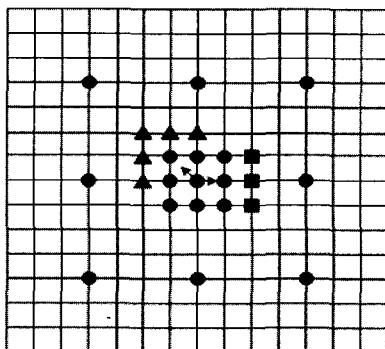


그림 1. NTSS를 사용하여 중심점에서 움직임 벡터를 찾는 경우에 대한 두 가지 상이한 탐색 경로
 Fig 1. Two different search paths for finding MV within 5x5 area in NTSSy

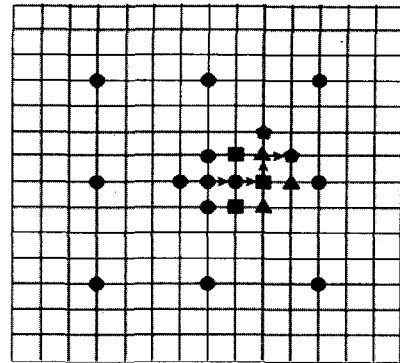


그림 2. ETSS를 사용하여 움직임 벡터를 찾는 과정에 대한 예
 Fig 2. An example of process finding MV using ETSS

점에서 최소 BDM 값을 갖는다면 NTSS와 동일하게 TSS를 수행한다.

그림 2는 중심점에 이웃하는 4개의 탐색점에서 최소 BDM 값을 갖는 경우에 대한 탐색 과정을 보여준다. 여기서 NTSS와 상이한 점은 처음에 설정한 탐색 영역 즉, 그림 2에서 중심의 5x5 격자 내에서 탐색을 멈추지 않고 현 탐색위치의 중심점이 최소값을 가지거나 프레임 경계를 만날 때까지 DS와 동일하게 탐색을 수행한다는 것이다. 그러나 이런 경우는 매우 드물기 때문에 다음에 설명하는 ETSS가 가지는 탐색점 수에 대한 분석에서는 이 경우가 발생하지 않는 것으로 가정한다.

ETSS는 정적인 블록인 경우에 13개 탐색점을 탐색하며, 움직임 작은 경우 즉, MV가 초기 중심점 근처에 있는 경우에 일반적으로 17~22개 탐색점을 탐색한다. 그리고 가장 많은 연산을 요구하는 경우는 초기 과정을 제외하고 TSS와 동일한 과정을 밟게 되는 경우로서 $25+4=29$ 개의 탐색점을 탐색하게 되어 NTSS보다 탐색점의 수가 4개 정도 적다. 따라서 일반적으로 ETSS는 NTSS에 비해 더 적은 연산량을 가진다. 5x5 격자내에서 탐색을 마치는 NTSS와 달리, 최소점을 찾을 때까지 탐색을 수행하기 때문에 움직임이 큰 영상이나 탐색 범위가 커질수록 약간의 화질 개선이 있지만 상대적으로 ETSS는 NTSS에 비해 중앙 편향적인 특성이 비교적 낮게 고려되었기 때문에 움직임이 작은 영상에서는 화질이 떨어진다.

2. 움직임 특성과 탐색 범위에 따른 NTSS의 성능 비교

TSS는 중심의 9x9 격자위에 8개의 탐색점이 중심점에서 탐색 범위 경계까지의 거리의 1/2지점에 위치한다. 그러나 NTSS의 중심에 이웃하는 8개의 추가된 탐색점

은 탐색 범위의 크기와 상관없이 고정되어있다. 그림 3과 같이 $w=7$ 인 경우에 첫 번째 단계에서 9×9 격자위에 8개의 탐색점간의 거리와 중심점으로부터 9×9 격자점까지의 거리는 4이다. 그러나 (b)와 같이 탐색 범위가 $w=14$ 로 증가한 경우 탐색점간의 거리는 7로 증가한다. 중심점에 이웃하는 3×3 격자위에 8개의 탐색점은 탐색 범위의 크기 변화와 상관없이 고정되어 있으며 탐색이 끝날 때 까지 5×5 내에서 탐색을 마친다. 따라서 중심점에 이웃하는 5×5 내의 탐색점과 9×9 의 탐색점간에 거리는 탐색 범위가 증가함에 따라 일정한 비율로 멀어지게 되며, 만약 전역 최소값이 그 사이에 위치한다면 정확한 탐색이 어려워진다. 그림 4에서 보여주듯이 Stefan 시퀀스와 같은 움직임이 큰 경우에 상기한 이유 때문에 탐색 범위가 클 경우 즉, $w=14$ 일때 $w=7$ 인 경우에 비해 성능이 떨어진다.

그림 4에서 $w=7$ 인 경우와 $w=14$ 인 경우의 PSNR 차이가 크게 네 부분에서 나타난다. 먼저 49~81번째 구간과 126~154번째 구간 프레임에서는 탐색 범위가 $w=7$ 일때 더 좋은 성능을 보여주며, 90~118번째 구간과 277~292번째 구간 프레임에서는 탐색 범위가 $w=14$ 일 때 상대적으로 좋은 성능을 보여준다. 그림 5는 $w=14$ 일 때 더 좋은 화질을 보여주는 90~118번째 구간 프레임 중에서 가장 차이가 큰 100번째 프레임을 보여준다. 이 프레임은 움직임 특성이 매우 큰 부분으로 탐색 범위가 큰 경우인 그림 5(b)가 상대적으로 고 화질을 제공한다. 특히 그림 5에서 직사각형 상자로 표시한 지역에서 테니스 선수, 그 뒷부분에 보이는 배경, 그리고 글자 지역 등에서 탐색 범위를 크게 사용한 경우에 화질이 월등하게 개선되었다.

그림 6은 $w=7$ 인 경우에 더 좋은 화질을 보이는 126

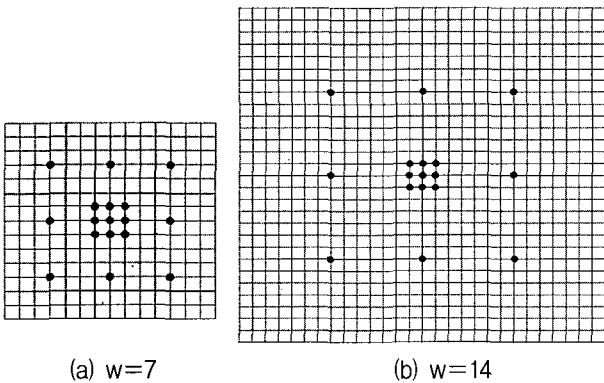


그림 3. $w=7, w=14$ 를 적용한 NTSS의 첫 번째 탐색 과정
 Fig 3. First searching process of NTSS applying $w=7, w=14g$

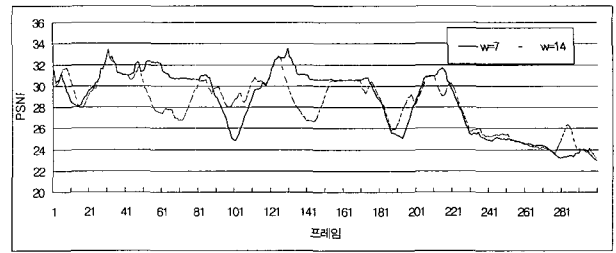


그림 4. CIF(352x288) 크기 Stefan 영상 300 프레임에 NTSS 를 두 가지 크기의 탐색 범위에 대하여 적용하여 얻은 PSNR 비교
 Fig 4. PSNR comparison by performing for Stefan (CIF(352x288),300 Frame) using two difference window sizes

~154번째 구간 프레임 중 가장 차이가 큰 143번째 프레임이다. 그림 6(a)와 (b) 중에서 탐색 범위가 작은 (a)인 경우가 더 좋은 화질을 보여준다. 그 이유는 해당 프레임의 움직임이 적기 때문이다. 다른 대표적인 실험 시퀀스에 적용하였을 때에도 유사한 결과를 얻었다. 이를 통하여 화질 측면에서 움직임이 큰 경우에는 탐색 범위가 클수록, 움직임이 작은 경우에는 탐색 범위가 작을수록 성능이 우수하다고 일반적으로 결론을 내릴 수 있다. 그러므로 움직임 특성에 적응적으로 탐색 범위의 크기를 조절하면 보다 향상된 화질을 얻을 수 있을 것이다. 이러한 실험 결과를 바탕으로 3장에서는 해당하는 프레임의 움직임 정도를 판단하여 탐색 범위를 적응적으로 변화시켜 화질을 향상시키는 새로운 방법을 제시한다.

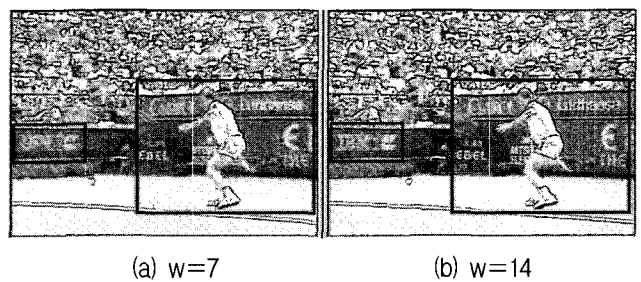


그림 5. CIF 크기 Stefan 영상의 100번째 프레임
 Fig 5. 00th frame in Stefan sequence of CIF size

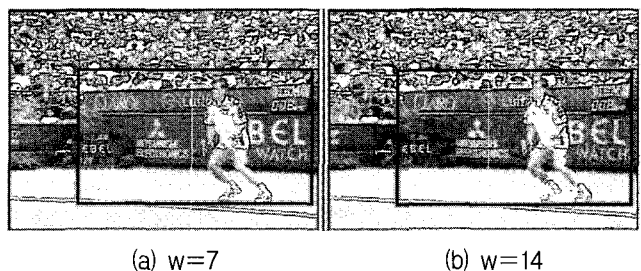


그림 6. CIF 크기 Stefan 영상의 143번째 프레임
 Fig 6. 143th frame in Stefan sequence of CIF size

III. 시퀀스 움직임 특성에 적응하는 탐색 범위를 가지는 새로운 움직임 탐색 방법

제안하는 방법에서 움직임 탐색은 다른 정합법과 마찬가지로 MB 단위로 수행된다. 그리고 탐색을 시작하기 전에 바로 전 프레임, 또는 두 프레임 전의 MV까지 고려하여 움직임의 크기를 판단한 후, 그 특성에 부합되는 탐색 범위에서 탐색을 수행한다. 고려되는 MV는 현재 프레임의 MB와 프레임 상에 같은 위치에 MV가 선택되며, 그 근거는 영상의 Linear Motion 특성에 기인한다. 입력 시퀀스가 가지는 움직임 정도를 판단하는 방법은 다음과 같이 크게 두 가지로 나누어 질 수 있다.

1. 이전 프레임의 움직임 벡터 크기를 참조하는 방법

표 1은 Stefan 영상에서 탐색 범위의 크기 $w=14$ 인 경우가 $w=7$ 인 경우보다 화질이 좋은 프레임 중에서 차이가 큰 100번째 프레임에 대한 MV 값, SAD 값, 탐색 방법 등을 보여준다. MB의 (i,j) 는 이 프레임에서 비교하고자 하는 MB의 위치를 나타내며, 7개의 MV는 다른 MV들에 비해 탐색 범위의 크기에 따라 상대적으로 SAD (Sum of Absolute Differences) 값의 차이가 크기 때문에 선택되었다. 표 1은 $w=7$ 인 경우 MV들이 정확한 전역 최소값을 찾지 못하고 탐색 범위 경계 부분에 걸리면서 $w=14$ 인 경우보다 큰 오류를 갖는 것을 보여준다. 움직임이 커서 탐색 영역을 벗어나는 경우를 위하여 움직임이 큰 MB에 대해서는 큰 탐색 영역을 사용하는 것이 요구되므로, 이전 프레임에서 탐색 범위의 경계선상에 위치한 움직임 벡터를 갖는 MB에 대해 현재 프레임에서는 탐색 범위를 식(1)에 따라 갱신한다.

$$d_c = d_p \times 2, \text{ if } |v_i| = d_p \text{ or } |v_j| = d_p \quad (1)$$

식 (1)에서 (v_i, v_j) 는 이전 프레임의 움직임 벡터이

표 1. Stefan영상 100번째 프레임의 두개의 탐색 범위에 크기에 따른 MV 값, SAD 값의 탐색 방법
Table 1. MV and SAD, and Searching algorithm according to the two search window sizes

MB의 (i,j)	w=7			w=14		
	SAD	MV	탐색방법	SAD	MV	탐색방법
(10,8)	3502	(-7,0)	TSS	1485	(-8,0)	TSS
(11,8)	5455	(2,0)	NTSS	2450	(-8,0)	TSS
(12,8)	4399	(-1,-2)	NTSS	2518	(-9,0)	TSS
(2,11)	4118	(5,6)	TSS	2651	(-8,0)	TSS
(3,11)	3868	(-7,0)	TSS	2760	(-8,0)	TSS
(4,11)	3927	(-7,-5)	TSS	2034	(-8,0)	TSS
(11,11)	4524	(-2,0)	NTSS	2274	(-8,0)	TSS

며, dp 와 dc 는 각각 이전 프레임과 현재 프레임의 MB의 탐색 영역 크기이다. 만약 이전 프레임의 참조 MB에서 탐색한 최소점이 탐색 범위 경계선에 위치할 경우, 움직임 특성이 크다고 판단하여 탐색하려는 MB의 탐색 범위를 수직, 수평 방향으로 각각 2배씩 확장한다.

그림 7은 식(1)과 같이 MV가 탐색 범위 경계선 상에 위치한 경우에 탐색 범위를 2배로 적용한 결과가 중심에서 더 가까운 위치에 최소점이 위치할 경우에 대한 결과보다 개선된 화질과 탐색점 수를 갖는지를 실험한 결과이다. 이 실험에서 CIF 크기의 4개의 영상 시퀀스 Akiyo, Coastguard, Stefan, Foreman 등을 각각 300프레임을 사용하여 얻은 결과의 평균을 보여준다. MV의 위치가 7인 경우, 즉 경계선상인 $dp=7$ 에서 PSNR과 연산량 측면에서 모두 가장 우수하며, MV가 중심으로부터의 거리가 가까운 지점에 위치할수록 PSNR이 현저하게 떨어지며 MB당 평균 탐색점도 증가하게 된다. 그 이유는 탐색 범위를 2배로 적용하는 빈도가 높아지므로 더 많은 탐색점을 요구하게 되며, 움직임 특성이 적절히 고려되지 않고 탐색 범위가 증가하기 때문에 PSNR 또한 떨어지는 것을 볼 수 있다.

2. 바로 전 프레임과 두 프레임 전의 움직임 벡터 비교에 의한 방법

1장에서 제시한 방법만으로 움직임이 큰 특성을 갖는다고 단정하기에 부족하다. 바로 전 프레임의 i, j 의 위치가 탐색 범위의 경계선상에 있지 않더라도 두 프레임 전에서 구한 값과 차이가 크다면 움직임이 큰 특성을 갖는다고 할 수 있다. 움직임의 크기를 판단하기 위한 두 번째 방법으로써, 바로 전 프레임과 두 프레임 전의 MV간의 거리를 고려하여 거리의 차가 크다면 움직임이 큰 특성을 갖는다고 판단한다. 두 벡터가 가질 수 있는 차의 범위는 0~14이다. 차 값이 13 이상이면 두

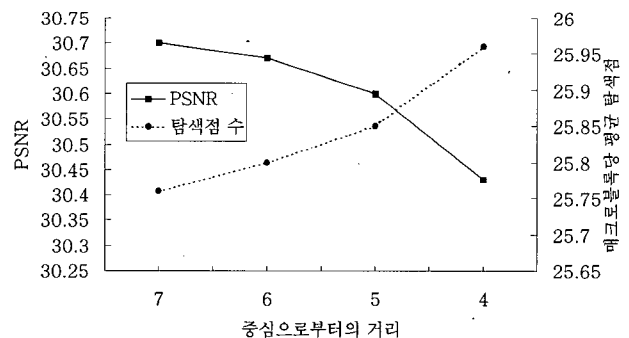


그림 7. i, j 의 범위에 대한 평균 PSNR과 매크로 블록 당 평균 탐색점 수

Fig 7. Average PSNR for range of i, j and Average searching points per a MB

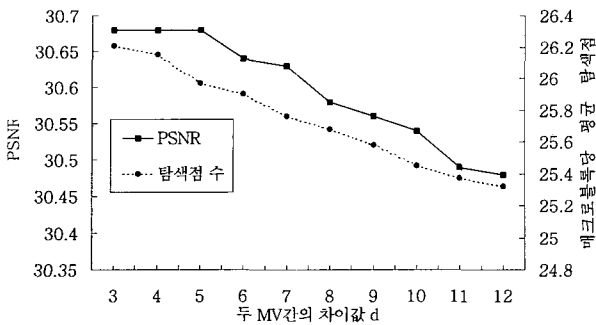


그림 8. 바로 전 MV와 두 프레임 전 MV의 거리 d에 따른 탐색 범위의 크기 적용과 결과
Figure 8. Applying window size according to distance d of MVs in previous and two frame old

개의 MV 중 적어도 하나는 탐색 범위 경계선 상에 위치하는 경우이므로 즉, 1장의 방법에서 탐색 범위를 확장하는 경우에 해당하므로 2장의 방법에 적용되는 경우에서 제외한다.

탐색 범위의 크기를 조절하기 위한 적당한 두 벡터의 차 d를 정하기 위하여 PSNR과 MB당 평균 탐색점 수를 비교한다. 그림 8은 탐색 범위를 증가시키기 위한 두 MV의 차이 값을 각각 적용하였을 경우 나타나는 결과를 보여준다. 두 MV의 차이가 5일 때 그림과 같이 블록당 평균 탐색점에 대해 PSNR이 가장 우수한 결과를 나타내며, 거리차를 12로 적용한 결과는 5와 비교하여 탐색점은 0.6개 정도 적게 요구되나 PSNR은 0.2이상 떨어지는 것을 볼 수 있다. 따라서 두 MV 간의 차이값에 따른 PSNR과 평균 탐색점의 관계를 반영하여 식(2)와 같은 규칙으로 적절한 탐색 범위를 적용함으로써 NTSS가 가지는 큰 움직임 특성에 대한 단점을 보완토록 하였다.

$$d_c = d_p \times 2, \text{ if } |v_i - v_i'| \geq 5 \text{ or } |v_i - v_j'| \geq 5 \quad (2)$$

식(2)에서 (v_i, v_j') 는 바로 전 프레임의 MV, 그리고 (v_i', v_j') 는 두 프레임 전의 MV를 나타낸다. 여기서 탐색 범위를 결정하고자 하는 MB는 두 가지 조건을 모두 만족하더라도 탐색 범위 증가는 2배로 고정되며, 현재 프레임에서 MB들은 탐색 범위의 크기를 결정하기 전에 $w=7$ 로 초기화 된다.

VI. 실험 과정 및 결과

4장에서는 제안한 두 가지 방법을 각각 적용한 경우와 두 방법을 조합하여 적용한 경우에 대한 성능을 기존 방법과 비교한다. 성능 실험을 위해 비디오 부호화기로 H.263+를 사용하였다^[19]. 실험 영상 시퀀스로 가

표 2. Stefan 시퀀스에 대한 제안된 방법과 NTSS의 성능 비교

Table 2. Comparative result through suggested methods

방법	성능	PSNR	MB당 평균 탐색점 수
NTSS(w=7)		28.64	25.17
NTSS(w=14)		28.33	27.03
방법 (1)		28.80	25.73
방법 (2)		28.83	25.97
조합한 방법		28.92	26.24

장 보편적으로 사용되는 Akiyo, Coastguard, Stefan 등을 사용하였다. 각 시퀀스의 크기는 352×288의 CIF 크기이며, 각각 300 프레임을 사용하였다. GOP는 첫 번째 프레임만을 인트라(intra)로 부호화 하였고, 나머지는 모두 인터(inter)로 부호화하였다. 비트율은 비트율 제어 방법을 적용하여 각 영상의 움직임 특성을 고려하여 저 비트율(48~1,024 kbps)과 고 비트율(96~2,048 kbps)에 대해 각각 부호화 하였으며, 비트 율을 명시하지 않은 실험의 경우, 기본적으로 저 비트율을 적용한다.

표 2는 Stefan 시퀀스에 $w=7$, $w=14$ 인 경우의 NTSS 방법을 적용한 결과와 본 고에서 제안한 두 가지 방법, 그리고 이 두 가지 방법을 조합한 방법을 적용하였을 때 얻은 결과를 보여준다. 각각의 방법을 별도로 적용 하였을 때 기존의 NTSS보다 PSNR이 0.16~0.50dB 정도 향상되었으며, 조합한 방법을 적용하였을 경우 최대 0.59dB 까지 향상되었다.

표 3에서는 다른 움직임 특성을 가진 네 개의 영상의 결과를 보여준다. 작은 움직임을 갖는 대표적인 영상 Akiyo는 기존의 NTSS와 제안된 방법이 같은 비트율 상에서 동일한 PSNR을 나타낸다. 그리고 Stefan 과 같은 큰 움직임 특성을 가진 영상뿐만 아니라 Foreman, Coastguard 와 같은 영상에서도 같은 비트율 상에서 화질이 향상되었다.

표 4는 각 영상에 대해 제안된 방법을 대표적인

표 3. $w=7$ 과 $w=14$ 를 적용한 네 개의 실험 영상 시퀀스에 대한 PSNR(dB)과 비트율(kbps)

Table 3. PSNR(dB) and Bitrate(kbps) for four sequences applying $w=7$ and $w=14$

영상종류	방법	목표 비트율 (kbps)	비교대상	NTSS		제안방법
				w=7	w=14	
Akiyo		48	PSNR	35.02	35.01	35.02
		96	PSNR	37.86	37.86	37.86
Foreman		256	PSNR	30.72	30.81	30.94
		512	PSNR	33.22	33.34	33.43
Coastguard		256	PSNR	28.04	27.95	28.07
		512	PSNR	30.40	30.26	30.43
Stefan		1024	PSNR	28.64	28.33	28.92
		2048	PSNR	32.51	32.15	32.74

BMA(Block Matching Algorithm)들과 비교한 것이다. 제안된 방법을 적용한 결과는 TSS에 비해 상당히 개선되었으며 FS에 가까운 성능을 보인다. 특히 Stefan 영상의 결과에서는 큰 움직임 특성에 상대적으로 좋은 성능을 갖는 TSS에 비해 0.3dB에 가까이 개선되었다.

그림 9는 두 가지 다른 탐색 범위에 대한 결과를 비교한 그림이다. 제안된 방법의 결과는 탐색 범위 $w=14$ 에서 감소하는 PSNR을 크게 보완하였으며 탐색 범위 $w=7$ 에서 감소하는 PSNR 또한 상당히 개선된 것을 알 수 있다. 특히 49~81번째와 126~154번째 프레임 내에서 $w=14$ 를 적용한 결과가 상당히 낮은 성능을 보이는데 제안된 방법을 적용한 결과는 $w=7$ 에서의 결과와 동일한 값을 유지하여 전체적인 PSNR을 향상시키는데 큰 영향을 미쳤다.

그림 10은 위에서 설명했던 100번째와 143번째 프레임에 대해 제안된 방법을 적용한 결과이다. 100번째 프레임을 보면 (c)는 (b)의 화질보다 다소 떨어지나 (a)보다 상당히 개선된 화질을 보여준다. 특히 블록으로 표시된 부분의 배경도 카메라에 의한 움직임 때문에 적용한 방법에 따라 화질의 차이를 보이며, 글자 지역은 사람의 눈이 인식하기가 쉬워 화질의 개선여부를 더 명확하게 보여준다. 그리고 143번째 프레임에서 (f)는 $w=7$ 을 유지하여 (d)와 유사한 화질을 보이며 (e)보다 개선된 것을 볼 수 있다.

이처럼 다양한 크기의 탐색 범위의 적용은 개선된 화질을 보여주지만 연산량의 증가 또한 반드시 수반된다. 효율적인 탐색 범위를 적용시킴으로써 연산량의 증가는 화질의 개선으로 보상 받을 수 있으나, 만약 비효율적인 탐색 범위가 적용된다면 연산량은 상당히 증가할 것이며 화질의 개선은 의미가 없어질 것이다. 따라서 MB당 평균 탐색점의 수는 연산량의 크기를 판단하기

표 4. 기존 방법에서 $w=7$ 을 사용한 경우와 제안한 방법을 사용한 경우에 대한 영상 시퀀스의 평균 PSNR(dB)과 비트율(kbps)

Table 4. Average PSNR(dB/Frame) and Bitrate (kbps) of the sequences for using $w=7$ and using proposed method

영상종류	방법	목표 비트율 (kbps)	비교대상	FS	TSS	NTSS	제안 방법
Akiyo	48	PSNR		35.06	34.80	35.02	35.02
	96	PSNR		37.89	37.83	37.86	37.86
Foreman	256	PSNR		31.32	30.60	30.72	30.94
	512	PSNR		33.38	33.14	33.22	33.30
Coastguard	256	PSNR		28.11	27.94	28.04	28.07
	512	PSNR		30.48	30.40	30.40	30.43
Stefan	1024	PSNR		28.96	28.63	28.64	28.92
	2048	PSNR		32.79	32.51	32.51	32.74

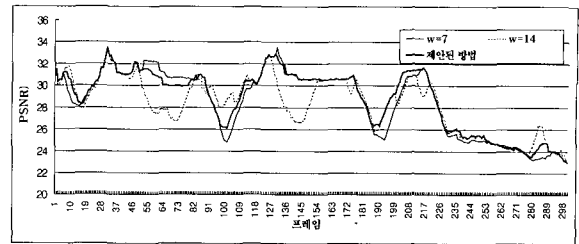
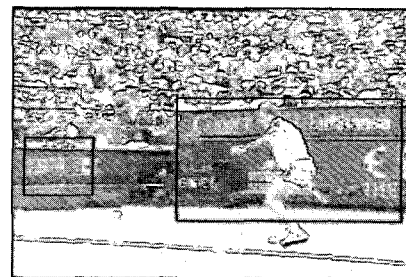


그림 9. $w=7$ 과 $w=14$ 를 적용한 NTSS의 두 가지 결과와 제안된 방법 비교

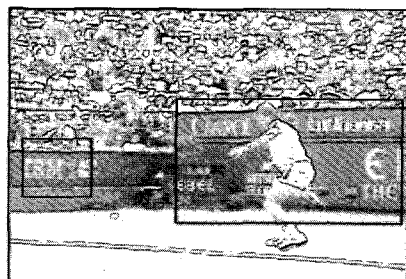
Figure 9. Comparison of Two results of NTSS applying $w=7$, $w=14$, and proposed method

위한 가장 중요한 기준중 하나이다.

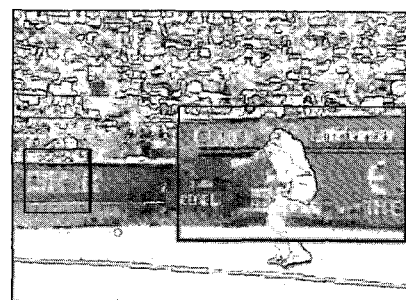
다음 표 5는 각 실험영상에서의 블록 당 평균 탐색점수를 보여준다. (a)는 대표적인 움직임이 작은 (Stationary) 특성을 가진 Akiyo 영상이다. 실험 결과에서 제안된 방법은 NTSS와 동일한 수치를 나타내는데 이것은 제안된 방법이 움직임이 작은 특성을 갖는지를 정확히 판단하여 불필요한 탐색 범위의 크기 증가를 방지하였음을 보여준다. (b)와 (c)는 각각 Foreman과 Coastguard에 대한 결과로써 크고 작은 움직임 특성을 모두 갖는 대표적인 영상이다. 이 결과에서 제안된 방



(a)



(b)



(c)

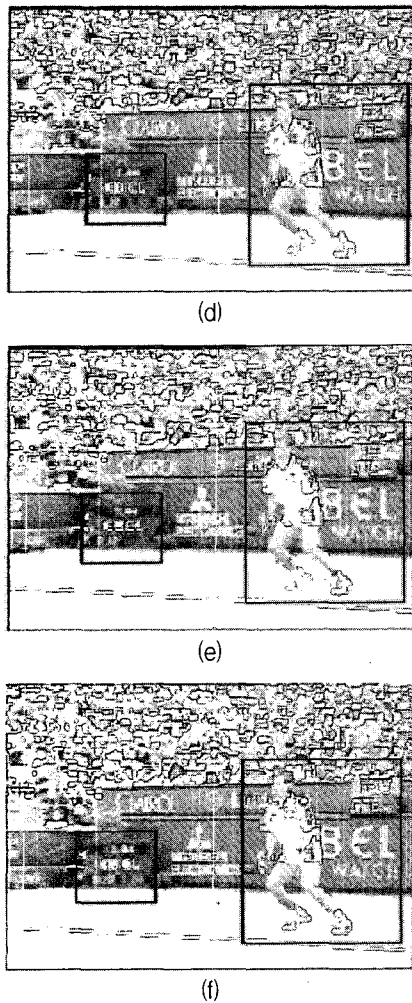


그림 10. 100, 143 번째 프레임들의 화질 비교. (a),(b),(c)는 100번째 프레임에서의 각각 $w=7$, $w=14$, 제안된 방법을 적용한 프레임이며, (d),(e),(f)는 143번째 프레임으로 100번째와 같은 순서로 적용된 그림이다.

Fig 10. Quality comparison of 100th and 143th frames. (a),(b),(c) are frames applying $w=7$, $w=14$, proposed method in 100th frame respectively, (d),(e),(f) are 143th frame applying with same as 100th frame.

법은 NTSS와 비교하여 거의 비슷한 수의 탐색점을 갖는다. 마지막으로 (d)는 움직임이 큰 특성을 가진 Stefan 영상으로 블록 당 탐색점은 NTSS와 비교해서 $w=7$ 에서 약간의 탐색점이 더 요구되나 $w=14$ 에서는 오히려 적게 요구됨을 알 수 있다.

다음 그림 11은 각 영상의 탐색 패턴에 대한 분포도를 나타낸 것이다. 그림에서 Center로 표시된 것은 탐색점이 중심점에 위치한 경우, 즉 움직임이 없는 경우이며 Three-Step으로 표시된 것은 TSS의 탐색 과정을 따를 경우, 그리고 NTSS는 중심에 이웃하는 5×5 내에서 탐색을 수행한 경우이다. 이 결과는 움직임에 크기에 따라

특성에 맞는 탐색 과정을 얼마나 정확하게 수행하는가를 보여준다. (a)는 움직임이 매우 작은 특성을 가지므로 제안된 방법을 적용한 결과는 $w=7$ 의 크기를 유지하여 연산량의 낭비를 줄인다. 나머지 영상들에서는 $w=14$ 를 적용한 결과가 $w=7$ 에 비해 Three-Step 패턴의 분포가 낮으며 반대로 NTSS 패턴의 분포는 높다. 이것은 움직임 특성이 작은 부분에서 탐색 범위의 크기가 증가할 경우, TSS의 탐색 과정을 수행해야 하는데 NTSS에 탐색 과정을 따르면서 오류를 범하는 것을 보여준다. 그러나 제안된 방법을 적용한 결과는 전체적으로 $w=7$ 에 가까운 분포를 보이면서 어느 정도가 $w=14$ 의 패턴을 따르는 것을 보이는데 이는 움직임이 매우 커지는 부분에서만 탐색 범위를 증가하여 화질을 개선하고 연산량을 크게 증가시키지 않았다는 것을 나타낸다.

표 5. 각 실험 영상에서 블록 당 평균 탐색점 수
Table 5. Average searching points per a MB for each sequence

(a) Akiyo

BMA	W=7		w=14	
	SearchPT	SpeedUP	SearchPT	SpeedUP
FS	225.00	1.00	961.00	1.00
TSS	25.00	9.00	33.00	29.12
NTSS	22.60	9.96	23.59	40.74
제안방법	23.47	9.59	23.47	40.95

(b) Foreman

BMA	W=7		w=14	
	SearchPT	SpeedUP	SearchPT	SpeedUP
FS	225.00	1.00	961.00	1.00
TSS	25.00	9.00	33.00	29.12
NTSS	17.20	13.08	17.20	55.87
제안방법	17.20	13.08	17.20	55.87

(c) Coastguard

BMA	W=7		w=14	
	SearchPT	SpeedUP	SearchPT	SpeedUP
FS	225.00	1.00	961.00	1.00
TSS	25.00	9.00	33.00	29.12
NTSS	22.23	10.12	21.58	44.53
제안방법	22.29	10.09	22.29	43.11

(d) Stefan

BMA	W=7		w=14	
	SearchPT	SpeedUP	SearchPT	SpeedUP
FS	225.00	1.00	961.00	1.00
TSS	25.00	9.00	33.00	29.12
NTSS	25.17	8.94	27.03	35.55
제안방법	26.24	8.91	26.24	36.62

V. 결 론

본 논문에서는 기존의 NTSS가 움직임이 큰 영상과

탐색 범위의 크기가 증가한 경우 TSS에 비해 화질의 개선이 없거나 오히려 감소하는 점을 보완하기 위하여 적응형 윈도우 기반의 NTSS를 제안하였다. 이 알고리

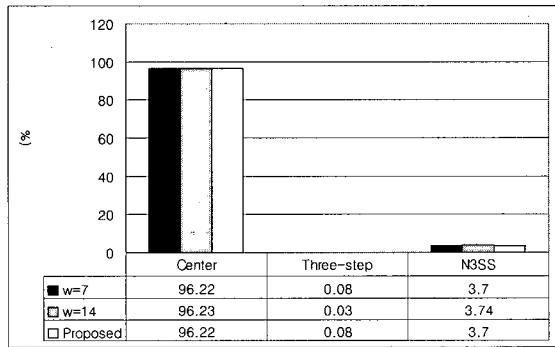
즘은 움직임 특성에 따라 적절한 탐색 범위를 적용함으로써 움직임이 큰 특성을 가진 영상에서 기존의 NTSS는 물론, TSS에 비해서도 개선된 화질을 보여준다.

NTSS와 비교하였을 때, 움직임이 작은 특성을 갖는 영상에 대해서 PSNR 측면에서 유사한 결과를 보이면서 움직임이 큰 특성을 가진 영상에서는 0.27~0.58dB 증가하였다. 또한 움직임 특성을 정확히 판단하고 효율적인 탐색 범위를 적용하여 요구되는 매크로 블록 당 평균 탐색점 수는 최대 1.07 정도로 연산량이 크게 증가하지 않았다. 따라서 제안된 방법은 움직임 특성에 좌우되지 않는 화질을 제공할 수 있는 움직임 예측이다

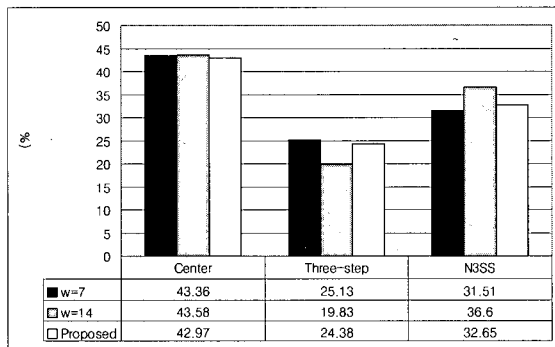
따라서 본 논문에서 제시하는 방법은 NTSS의 특성을 유지하여 저 비트율 기반의 영상과 TV, 고화질 TV와 같은 매우 큰 탐색 범위를 사용하는 분야에도 효과적으로 적용 시킬 수 있다.

참고 문헌

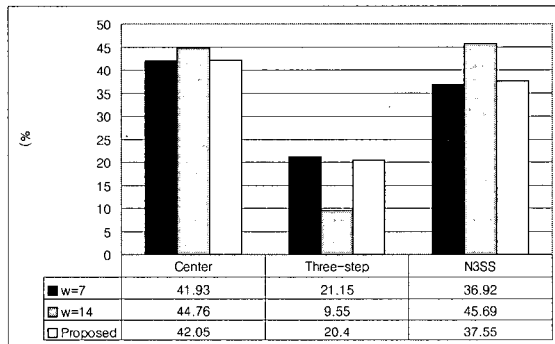
- [1] Iain E. G. Richardson, *Video Codec Design* JOHN WILEY & SONS, LTD. 2002, pp.93-125
- [2] A.Murat Tekalp, *Digital Video Processing*, Prentice Hall. p.101-106. 1995.
- [3] ISO/IEC, JTC1/SC29/WG11, Generic coding of moving pictures and associated audio, Part 2 Video, IS 11172-2, March. 1995
- [4] ISO/IEC, JTC1/SC29/WG11, Generic coding of moving pictures and associated audio, Part 2: Video, IS 13818-2, March.1995
- [5] ISO/IEC, JTC1/SC29/WG11, Generic coding of moving pictures and associated audio, Part 2: Video, IS 14496-2, Nov. 1998.
- [6] ITU-T, Video Coding for Low Bitrate Communication, Draft Recommendation H.263, July,1995.
- [7] ITU-T, Video Coding for Low Bitrate Communication, Draft Recommendation H.263+, July,1995.
- [8] ITU-T, Joint Final Committee Draft of Joint Video Specification, Draft Recommendation H.264, July, 2002
- [9] Jerry D. Gibson, Toby Berger, Tom Lookabaugh, Dave Lindbergh, Richard L. Baker *Digital Compression for Multimedia* Morgan Kaufmann Publishers, Inc. San Francisco, California. 1998, pp.376-385
- [10] Vasudev Bhaskaran, Konstantinos Konstantinides, *Image And Video Compression Standards* Kluwer Academic Publishers, 1997, pp.120-125
- [11] T. Koga, K. Iinuma, a Hirano, Y. Iijima, T. Ishiguro, "Motion-compensated interframe coding



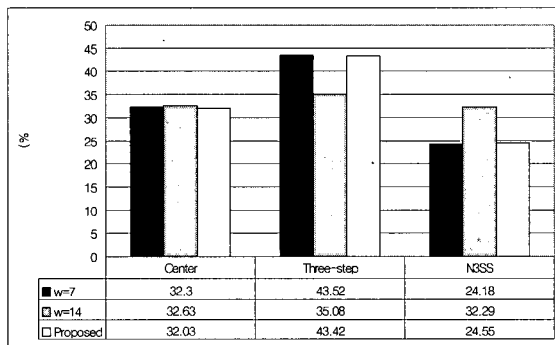
(a) Akiyo



(b) Foreman



(c) Coastguard



(d) Stefan

그림 11. 각각의 실험 영상에 대한 탐색 방식의 분포도
figure 11. Distribution of searching pattern for each test sequences

for video conferencing," in *Proc. Nat. Telecommun. Conf.*, NewOrleans, LA, Nov. 29-Dec. 3 1981, P.G5.3.1-G5.3.5

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

[13] M.Ghanbari, "The cross-search algorithm for motion estimation", *IEEE Trans. Commun*, Vol.7, pp.950-953, July 1990

[14] C. H. Cheung and L. M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1168-1177, Dec. 2002

[15] J. H. Lee, K. W. Lim, B. c. Song, and J. B. Ra, "A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp 1289-1301, Dec. 2001.

[16] R. Li, B.Zeng, and M.L.Liou, "A New three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst.VideoTechnol.*, vol.4, pp. 438-442, Aug. 1994

[17] Wuan Jing and Lap-Pui Chau, Senior Member "An Efficient Three-Step Search Algorithm for Block Motion Estimation" *IEEE Trans. on Multimedia*, Vol. 6, NO. 3, JUNE 2004

[18] Liang-Wei Lee, Jhing-Fa Wang, Jau-Yien Lee, and Jung-Dar Shie "The Dynamic Search-Window Algorithm" *IEEE Trans. on Circuit and System, Video Tech.* Volume 3. pp. 85-87. February 1993

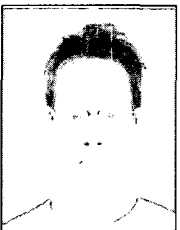
[19] ITU-T, Video Coding for Low Bitrate Communication, Draft Recommendation H.263+, July.1998.

저 자 소 개



오 승 준 (정회원)
 1980년 서울대학교 전자공학과 학사
 1982년 서울대학교 전자공학과 대학원 석사
 1988년 5월 미국 Syracuse University 전기/컴퓨터공학과 박사

1982년 3월~1992년 8월 한국전자통신연구원 근무 (멀티미디어연구실 실장)
 1986년 7월~1986년 8월 NSF Supercomputer Center 초청 학생연구원
 1987년 5월~1988년 5월 Northeast Parallel Architecture Center 학생연구원
 1992년 3월~1992년 8월 충남대학교 컴퓨터공학과 겸임교수
 1992년 9월~현재 광운대학교 전자공학부 교수 (VIA-Multimedia Center 멀티미디어연구실)
 2002년 3월~현재 SC29-Korea 의장 및 MPEG Forum 부의장
 <주관심분야 : 비디오 데이터 처리, 비디오 코덱 시스템, 비디오 및 영상압축, 멀티미디어시스템>



유 종 훈 (학생회원)
 1999년 8월 광운대학교 전자공학과 학사
 2003년 9월~현재 광운대학교 대학원 전자공학과 석사과정
 <주관심분야 : Real Time System, DMB>



안 창 범 (정회원)
 1981년 서울대학교 전자공학과 학사
 1983년 8월 한국과학기술원 전기 및 전자공학과 석사
 1986년 8월 한국과학기술원 전기 및 전자공학과 박사

1986년 9월~1991년 4월 미국 University of California, Irvine 연구교수
 1991년 4월~1992년 8월 생산기술연구원 전자정보센터 부교수
 1992년 9월~현재 광운대학교 전기공학과 교수 (신호처리연구실)
 2001년 9월~2002년 8월 미국 University of California, Irvine 객원연구원
 2004년 9월~현재 광운대학교 산학협력단 VIA-Multimedia Center
 1989년 Sylvia Sorkin Greenfield Award 수상
 1997년 IR52 장영실상 수상
 <주관심분야 : 다차원 신호처리, 영상 압축, 영상 시스템>



박 호 중 (정회원)
 1986년 서울대학교 전자공학과 학사
 1987년 12월 University of Wisconsin-Madison, 전기/컴퓨터공학과 석사
 1993년 5월 University of Wisconsin-Madison, 전기/컴퓨터공학과 박사

1993년 9월~1997년 8월 삼성전자 선임연구원
 1997년 9월~현재 광운대학교 전자공학과 부교수
 <주관심분야 : 음성/오디오 신호처리, 멀티미디어 시스템>