

일반논문-04-09-4-14

## 미디어 프로세서에 적합한 효율적인 움직임 탐색 알고리즘

노대영<sup>a)\*</sup>, 김성훈<sup>a)</sup>, 손채봉<sup>a)</sup>, 오승준<sup>a)</sup>, 안창범<sup>a)</sup>

### An Efficient Motion Search Algorithm for a Media Processor

Dae-Young Noh<sup>a)\*</sup>, Seong-Hoon Kim<sup>a)</sup>, Chae-Bong Sohn<sup>a)</sup>, Seoung-Jun Oh<sup>a)</sup> and Chang-Beom Ahn<sup>a)</sup>

#### 요 약

대부분의 비디오 부호화기에서 움직임 예측은 부호화된 영상의 화질을 결정하는 중요한 기술이다. 그러나 일반적으로 많이 사용되는 전역 탐색 방법은 많은 연산량이 필요하다는 단점이 있다. 이를 개선하기 위한 기존의 고속 움직임 예측 방법들은 연산량을 줄이기 위한 복잡한 프로그램 구조를 가지고 있어서 DSP와 같은 전용 프로세서에 구현하기가 부적합하다. 본 논문에서는 미디어 프로세서에 적합한 효율적인 움직임 탐색 방법을 제안한다. 주위 블록 간의 공간적 상관도를 이용해 예측 움직임 벡터를 구할 수 있으며, 또한 탐색 영역을 축소시킴으로써 전체 연산량을 줄일 수 있다. 제안된 방법의 효율을 확인하기 위하여 MPEG-4 SP@L3(Simple Profile at Level 3) 부호화기를 TM1300에서 구현하였다. 제안한 방법은 TM1300 프로세서 상에서 전역 탐색 방법에 근접한 화질을 유지하며 기존 고속 움직임 예측 방법보다 향상된 성능을 보였다.

#### Abstract

Motion Estimation is an essential module in video encoders based on international standards such as H.263 and MPEG. Many fast motion estimation algorithms have been proposed in order to reduce the computational complexity of a well-known full search algorithms(FS). However, these fast algorithms can not work efficiently in DSP processors recently developed for video processing. To solve for this, we propose an efficient motion estimation scheme optimized in the DSP processor like Philips TM1300. A motion vector predictor is pre-estimated and a small search range is chosen in the proposed scheme using strong motion vector correlation between a current macro block (MB) and its neighboring MB's to reduce computation time. An MPEG-4 SP@L3 (Simple Profile at Level 3) encoding system is implemented in Philips TM1300 to verify the effectiveness of the proposed method. In that processor, we can achieve better performance using our method than other conventional ones while keeping visual quality as good as that of the FS.

Keywords : BMA, media processor, Motion estimation, MPEG-4

## I. 서 론

디지털 전송기술이 급격히 발달하고 전송 매체가 다양화되면서 멀티미디어 기술의 필요성이 크게 부각되고 있다.

특히 개인용 휴대 단말기나 카메라와 같은 장치 독립형 시스템에서 실시간 동영상 부호화 및 전송에 대한 관심이 급격히 증가하였다. 이러한 환경에서 멀티미디어 정보의 효율적 전송과 저장을 위한 동영상 압축 기술은 핵심 기술로 자리를 잡았으며 이러한 기술에 대해서 국제 표준화 기구에서 표준들을 정하고 있다<sup>[1]</sup>. 가장 대표적인 표준화 활동인 ISO/IEC JTC1 SC29 WG11에서는 멀티미디어 통신 서비스를 고려하여 유동적으로 기존의 방식 및 새로운 기능

a) 광운대학교 VIA-Multimedia Center

VIA-Multimedia Center, Kwangwoon University

※ 본 연구는 2004년도 광운대학교 교내학술연구비 지원 및 한국 과학재단 목적기초 연구(R01-2002-000-00179-0) 과제로 수행되었음.

들을 모두 지원할 수 있는 부호화 도구를 표준화 하고 있다<sup>[2-4]</sup>. 그러나 표준화 된 비디오 부호화기를 구현하려면 높은 연산 능력이 필요하기 때문에 이를 지원하기 위한 미디어 프로세서들이 개발되고 있다. 최근에 발표되고 있는 고성능 DSP중에는 멀티미디어 응용 제품 개발에 공통되는 장치들을 집적화한 미디어 프로세서(Media Processor)가 있다<sup>[5]</sup>.

미디어 프로세서는 여러 가지 미디어를 실시간으로 처리하기 위하여 필요한 모듈을 갖추고 있으며 적합한 컴파일러를 제공하는 특수 목적용 DSP 이다. 이런 미디어 프로세서는 일반적인 범용 프로세서에 비해 낮은 파워를 사용하고 미디어 처리에 걸리는 시간을 줄이는 구조를 가지고 있다. 미디어 프로세서는 파워 소모를 줄이기 위하여 복잡한 제어 구조를 사용하지 않는다. 그리고 처리 과정에 걸리는 시간을 줄이기 위하여 VLIW 구조를 사용하며 캐시 메모리를 내장하여 외부 메모리와 데이터를 전송하는 시간을 줄이도록 되어 있다. 이런 미디어 프로세서는 동영상 부호화 시 움직임 예측 부분에 사용되는 SAD(Sum of Absolute Difference)를 고속으로 처리할 수 있는 방법을 제공하므로 동영상 실시간 비디오 부호화기 설계에 적합하다고 할 수 있다. 이런 미디어 프로세서의 종류에는 MAP, M-PIRE, D30V/MPEG, TM1300등이 있다. 본 논문에서는 이런 여러 가지의 미디어 프로세서 중 필립스사의 TM1300을 실험에 사용한다.

움직임 예측(Motion Estimation)은 부호화된 동영상의 화질과 비트율을 좌우하는 비디오 부호화기의 핵심 요소로 비디오 부호화기에서 가장 많은 연산을 필요로 하는 부분이다. 움직임 예측을 위해 일반적으로 사용되는 전역 탐색 방법(Full Search Algorithm : FS)은 블록 정합법(Block Matching Algorithm)을 기반으로 탐색 영역 내의 모든 매크로블록(Macroblock)을 검사하여 가장 작은 왜곡(distortion)을 가진 매크로블록을 찾는 방식이다<sup>[6]</sup>. 그러나 탐색 영역 내의 모든 탐색점(search point)을 검사하는 것은 많은 연산량을 필요로 한다. 이러한 이유로 다양한 방식의 고속 움직임 예측 방법들이 제안되었다. 잘 알려진 블록 정합법 기반의 고속 움직임 예측 방법으로 고정된 탐색 패턴을 사용하는 3단계 탐색 방법(Three Step Search : TSS)<sup>[7]</sup>, 다이아몬드 탐색 방법(Diamond Search : DS)<sup>[8]</sup>, 효과적인 3단계 탐색 방법(Efficient Three Step Search : ETSS)<sup>[9]</sup>과 이웃하는 블록들의 움직임 정보를 이용하는 인접 블록 참

조 탐색 방법(Nearest Neighbors Search : NNS)<sup>[10]</sup>, 적응형 예측 움직임 벡터 탐색 방법(Predictive Motion Vector Field Search Technique : PMVFAST)<sup>[11]</sup>, 그리고 영상 크기에 따른 계층적 구조를 활용하는 다중 영상 보간 부호화 방법(Interpolative Multiresolution Coding)<sup>[12]</sup>, 계층적 탐색 방법(Hierarchical searching algorithm)<sup>[13]</sup>등이 있다.

미디어 프로세서에서 효율적인 움직임 탐색을 위해서는 1)SAD 가속기를 효과적으로 사용하고 2) 캐시(Cache)의 효율적인 사용으로 외부 메모리로의 데이터 전송을 줄이며 3)제어 명령(Control Operation) 사용의 축소 및 병렬처리의 효율 증가를 고려해야한다. 기존의 움직임 예측 방법에서 전역 탐색 방법은 너무 많은 데이터 전송으로 인해 미디어 프로세서에서 사용하기에 적합하지 않다. 고정된 탐색 패턴을 사용하여 움직임을 예측하는 방법은 미디어 프로세서의 내부 캐시와 외부 메모리와의 잦은 데이터 전송을 요구하며 이웃하는 블록들의 움직임 정보를 이용하는 고속 움직임 예측 방법은 복잡한 제어 명령을 사용함으로써 미디어 프로세서에 사용되기에 비효율적이다<sup>[14]</sup>. 그리고 계층적 구조를 활용하는 방법은 고정된 탐색 패턴을 사용하여 움직임을 예측하는 방법보다 계산량은 더 많다. 그렇지만 여러개의 매크로블록의 움직임을 동시에 탐색하도록 하기 위하여 병렬로 나열된 연산 장치(Processing Element)를 사용하도록 설계할 수 있는 VLSI에서 최적의 성능을 발휘한다<sup>[13]</sup>. 미디어 프로세서의 구조상 계층적 구조를 활용하는 방법으로 움직임을 예측하기에는 적합하지 않다<sup>[15]</sup>.

본 논문에서는 영상의 특성을 이용한 움직임 탐색을 효율적으로 하기 위하여 실험에 사용된 TM1300 미디어 프로세서에서 고려해야 할 사항을 정의하고, 각 고려 사항들에 따른 효율적인 움직임 탐색 방법을 제안한다.

## II. 미디어 프로세서 상에서 움직임 탐색을 위한 고려 사항

### 1. SAD 가속기의 사용

블록 정합에 대한 기준을 정하는 여러 측정 도구에는 여러 가지가 있지만 일반적으로 식(1)과 같이 비교적 간단한

$$SAD(i, j) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)| \quad (1)$$

SAD를 많이 사용한다.

$C(x+k, y+l)$  와  $R(x+i+k, y+j+l)$ 은 각각 현재 프레임과 참조 프레임에서의 블록 내부의 화소들을 가리킨다.  $x$ 와  $y$ 는 블록의 위치를 나타내며  $i$ 와  $j$ 의 범위는  $-p \leq i, j \leq p$  ( $[-p, p]$  : 탐색 영역)이다. 그리고  $N, M$ 은 블록의 크기를 나타내고  $k, l$ 은 블록 내에서 픽셀의 위치를 나타낸다. 탐색 영역 중  $SAD(i, j)$ 가 최소가 되는 탐색점의 블록을 정합이 가장 잘된 블록으로 정하고 이 때의  $(i, j)$ 를 움직임 벡터로 결정한다.

SAD 연산은 식(1)에서 보다시피 SUB, ADD, ABS의 3가지 산술 연산을 기초로 한다. 식(2)는 하나의 블록에 대한 움직임 탐색 시 산술 연산의 수를 나타낸다.

$$PN_{ob} = 3 \times (2p + 1)^2 \times MN \quad (2)$$

$(2p + 1)^2$ 은 하나의 탐색 영역 내에 있는 탐색점의 수를 나타내며,  $M$ 은 한 블록의 수평 화소의 수를  $N$ 은 한 블록의 수직 화소의 수를 나타낸다.

미디어 프로세서가 이전의 프로세서와 확연히 구분되는 점은 움직임 예측을 위한 SAD 가속기를 제공한다는 점이다. 실험에 사용된 Philips 사의 미디어 프로세서 TM1300의 경우 SAD 가속기는 동시에 8개의 화소에 대한 SAD를 구할 수 있다<sup>[16]</sup>. TM1300의 SAD 가속기를 사용하여 하나의 블록에 대한 전역 탐색 방법 시 산술 연산의 수를 보면 식(3)과 같다.

$$PN_{obSAD} = (2p + 1)^2 \times MN/8 \quad (3)$$

SAD 가속기를 제공하지 않는 하나의 RISC (Reduced Instruction Set Computer) 프로세서에서 하나의 블록에 대해 전역 탐색 방법을 적용한다면 식(2)와 같은 계산이 가능하다. 따라서 산술 연산 측면만 고려하면 SAD가속기를 사용하면 가속기를 사용하지 않은 프로세서에 비하여 블록 하나에 대한 움직임 탐색 속도가 24배 빠르다는 것을 알 수 있다. 그러나 실제로는 데이터의 읽기/쓰기와 데이터 정렬 등의 변수가 있어 더욱 빠른 속도를 낼 수 있다.

## 2. 데이터 캐시 메모리의 효율적인 사용

미디어 프로세서와 같은 DSP(Digital Signal Processor)는 대부분 고속의 명령 사이클을 가지고 있다. 그러나 하드

웨어 구성과 연관이 있는 외부 메모리에 대한 동작은 Register, 내부 메모리 혹은 캐시에서의 처리보다 추가적인 접근 사이클을 필요로 한다. 따라서 캐시의 사용 효율에 따라 많은 성능 차를 보인다.

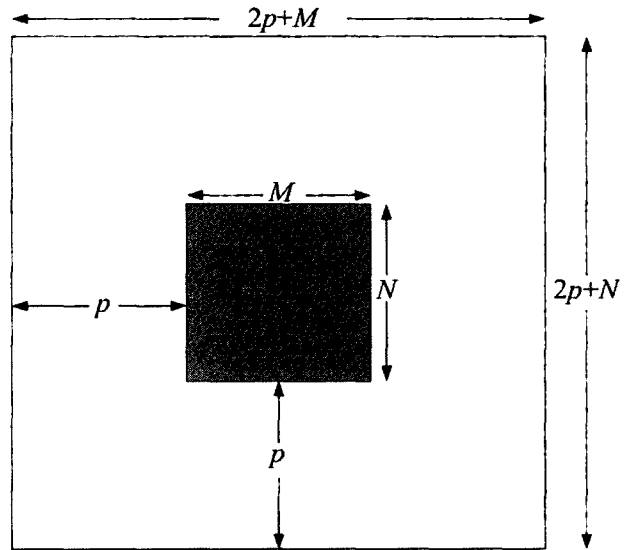


그림 1. 움직임 예측 방법의 탐색 영역  
Fig. 1. Search range for motion estimation

그림 1과 같은 탐색 영역에서 움직임 예측에 필요한 데이터량은  $(2p + M) \times (2p + N)$ 과 같다. 전역 탐색 방법을 사용하였을 경우  $p$ 를 15로 하면 블록 하나에서 움직임 예측에 필요한 데이터 량은 2,116개의 픽셀 값이 필요하게 된다. 그러나 미디어 프로세서에 동영상 압축을 위한 부호화기를 탑재시키기 위해서는 움직임 예측 모듈 뿐만이 아닌 다른 모듈들도 필요로 하기 때문에 움직임 예측 모듈에 할당되는 데이터 캐시 메모리의 사이즈가 작을수록 유리하다. 따라서 외부 메모리로의 접근을 줄이고 캐시 메모리의 효율을 높이기 위해서는 움직임 예측 방법의 탐색 영역을 줄이고 탐색 영역을 정형화 시킬 필요가 있다.

## 3. 데이터 흐름(Data Flow)의 정규화

통상 데이터 캐시는 한번의 메모리 접근 시 연속된 주소의 데이터를 전송 한다. 최근의 SDRAM과 같은 메모리 장치는 한번의 메모리 주소를 전송하는 것으로 연속된 주소의 여러 바이트의 데이터를 연속적으로 전송하는 모드를

지원한다. 이는 어드레싱(Addressing)에 따른 지연을 줄일 수 있어 전송 속도를 빠르게 한다. 이런 하드웨어 구조에 적절한 움직임 탐색 방법에는 연속된 주소의 데이터를 내부 캐시 메모리에 불러들여 사용하는 전역 탐색 방법과 같은 탐색 방법이 적절하다.

기존의 고속 움직임 예측 방법들은 탐색점을 줄이면서 영상의 화질을 유지하기 위해 복잡한 과정을 필요로 한다. 이러한 방법들은 이전 단계에서 행한 결과를 기초로 탐색점을 결정하는 방식으로 다음 데이터를 예측할 수 없다. 따라서 순차적인 데이터 전송에 의한 효율이 감소하고 데이터 캐시의 활용도도 떨어진다. 또한 데이터 처리의 병렬화가 힘들어 좋은 성능을 내기 힘들다. 그리고 비교/분기 등의 제어 명령의 증가로 파이프라인의 효율을 감소시킨다. 이런 문제점을 해결하기 위해서는 다소 탐색점이 늘더라도 비교/분기 제어 명령을 줄여서 움직임 탐색에 필요한 데이터를 일정한 영역으로 규정함으로써 데이터의 흐름을 정규화하여 캐시 메모리의 활용도를 높이도록 한다.

### Ⅲ. 제안된 움직임 벡터 예측 방법

2장에서 제시한 사항을 정리하면 미디어 프로세서에 적합한 움직임 탐색 방법은 순차적인 데이터 전송을 하고 정형화된 연산을 되풀이하여 병렬 처리 및 데이터 캐시의 효율을 높이고 제어 명령을 최소화하는 것이 필요하다는 것이다. 이와 같은 특징을 살리려면 전역 탐색 방법과 같은 블록 탐색 방법이 적합하다. 본 논문에서는 예측 움직임 벡터와 영상의 특성을 이용, 한정된 영역에서 블록 탐색을 하는 방법을 제안한다.

#### 1. 움직임 벡터 예측 및 탐색 영역 선택

영상의 공간적 유사도는 예측 움직임 벡터(Predicted Motion Vector : PMV)를 기준으로 보면 더욱 강해지는 현상을 실험을 통해 알 수 있다. 그림 2는 탐색 영역의 중심점을 기준으로 전역 탐색을 했을 때 움직임 벡터의 분포를

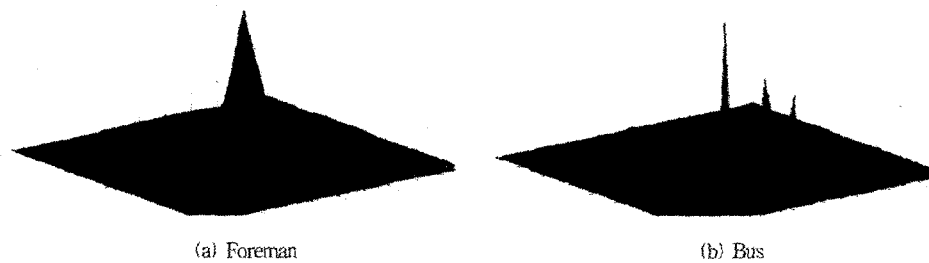


그림 2. FS에 의한 중심점 기준 움직임 벡터 분포  
Fig 2. Motion vector distribution using the FS algorithm versus (0,0) MV

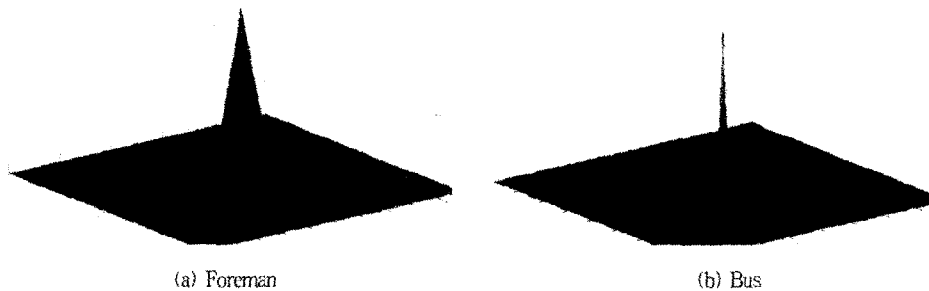


그림 3. 중간값 예측에 의한 PMV 기준 움직임 벡터 분포  
Fig 3. Motion vector distribution using the median predictor versus predicted MV

보여준다. Foreman 영상은 중심을 기준으로 그 주위에 움직임 벡터가 존재하는 반면 Bus 영상은 움직임 벡터가 좌우로 넓은 영역에 분포되어 있는 것을 볼 수 있다. 그림 3은 MPEG-4와 H.263과 같은 표준에 채용된 중간값 예측(Median Predictor)을 통해 얻은 예측값을 기준으로 할때 움직임 벡터의 분포이다. 중간값 예측은 현재 프레임의 공간적 유사도를 이용한 방법으로 현재 매크로블록의 왼쪽 매크로블록과 위쪽 매크로블록, 그리고 오른쪽 위쪽의 매크로블록의 움직임 벡터 중 중간값을 취하는 방법이다<sup>[17-19]</sup>. 두 영상 모두 예측 움직임 벡터를 기준으로 작은 영역 내에 대부분의 움직임 벡터가 있음을 알 수 있다.

표 2. PMV를 기준으로 한정된 영역 내의 움직임 벡터의 분포 (크기 : CIF, 프레임 수 : 300, 단위 : %)

Table 2. PMV-based motion vector distribution for several video sequences (sequence size : CIF, frame number : 300, Unit : %)

분포 범위 (단위 : 픽셀)	Akiyo	Foreman	Mobile	Stefan
3×3	99.86	73.73	93.56	66.91
5×5	99.98	81.61	95.20	71.58
7×7	100	85.27	95.90	75.25
9×9	100	87.62	96.41	78.50
11×11	100	89.45	96.74	81.31
13×13	100	90.80	96.97	83.37
15×15	100	91.90	97.24	85.05
17×17	100	93.12	97.51	86.76
19×19	100	94.06	97.78	88.24

표 2에서는 각각의 실험 영상에서 전역탐색 방법으로 구한 움직임 벡터와 중간값 예측으로 나온 예측 움직임 벡터와의 차이값의 분포도를 나타냈다. 분포 범위는 두 움직임 벡터의 차이가 픽셀 단위로 어느정도의 영역만큼 떨어져 있는지를 나타낸다. Akiyo 영상의 경우 전역 탐색으로 구한 움직임 벡터를 기준으로 예측 움직임 벡터가 7×7영역, 즉 상하좌우 3픽셀의 영역 내에 위치하는 것을 알 수 있다. 움직임이 큰 영상인 Stefan 영상의 경우 전역 탐색에 의한 움직임 벡터의 상하좌우 9픽셀내의 영역에서도 예측 움직임 벡터값이 벗어나는 것을 알 수 있다. 중간값 예측에 의한 예측 움직임 벡터  $x_{MEDIAN}$ 과 전역 탐색에 의한 움직임

벡터  $x_{FS}$ 의 거리차를 분석하기 위해 식(4)와 같이 유클리디언 거리(Euclidean Distance) 방식으로 단순화하여 프레임별 평균을 그림 4에 나타내었다.

$$D_{mv} = \sqrt{(x_{MEDIAN} - x_{FS})^2 + (y_{MEDIAN} - y_{FS})^2} \tag{4}$$

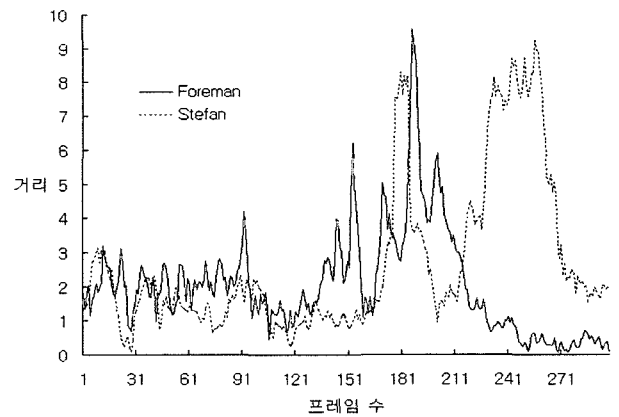


그림 4. FS와 중간값 예측에 의한 움직임 벡터 간의 프레임별 평균 거리 측정

Fig 4. Average distance comparison of frame-based MV using the FS versus median predictor

그림 4는 전역 탐색 방법과 중간값 예측에 의한 움직임 벡터의 차이값을 프레임 단위로 평균을 취한 것이다. Akiyo 영상과 Mobile 영상은 전 영역에 걸쳐 차이값이 1 미만이 되기 때문에 첨부하지 않고 Foreman과 Stefan 영상을 이용한 실험값만 그림으로 보였다. 그림 4가 보여주듯이 각 영상에서 움직임이 크고 빠른 부분에서 전역탐색 방법을 이용한 움직임 벡터와 중간값 예측에 의한 움직임 벡터간의 거리 차가 커지는 것을 알 수 있다. 그러나 움직임이 작고 느린 부분에서는 거리가 약 4 이내, 탐색 영역 상에서 9×9 영역 안에 움직임 벡터가 존재한다.

## 2. 블록 탐색

메모리 사용 효율을 높이기 위해 탐색은 저장된 순서를 따라 좌에서 우로 순차적으로 탐색한다. 그림 5는 탐색영역

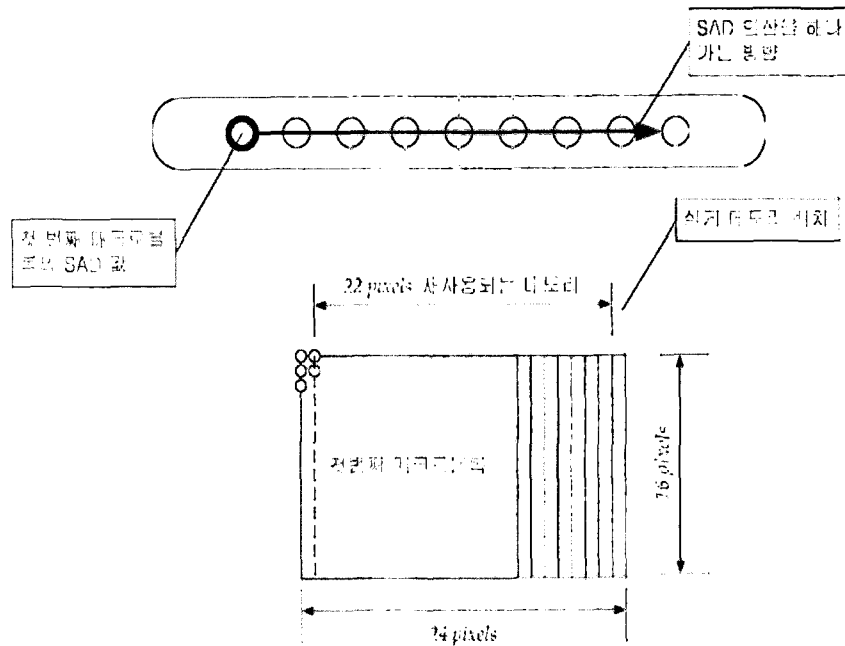


그림 5. 탐색 영역내의 한 라인의 메모리 배치  
Fig 5. Memory access for one-line in search range

내의 한 라인을 순차적으로 SAD 값을 구하며 비교할 때 메모리 맵을 나타낸 것이다. 가로 9 개의 연속된 매크로블록에 대해 메모리 맵을 보면 24x16 화소의 배열로 나타낼 수 있다. 첫번째 매크로블록의 SAD 값을 계산한 후 다음 매크로블록은 한 화소 이동한 블록이 된다. 즉 22x16 화소가 두번 이상 사용된다. 프로세서는 첫 번째 매크로블록의 첫 화소를 메모리로부터 읽어올 때 그 뒤의 데이터까지 한번에 데이터 캐시에 적재할 것이다. 따라서 한 라인 전체의 SAD 연산을 하나의 루프(Loop) 구문 내에서 할수 있으므로 가장 작은 SAD 값을 구하는 방법이 외부 메모리로부터의 데이터 전송을 줄일 수 있다. 라인 단위 탐색은 2.3에서 논의한 바를 충족한다. 정규화 된 같은 작업의 연속으로 병렬성을 높일 수 있고 제어 명령 사용으로 인한 오버헤드를 줄일 수 있다. 또한 고속 움직임 예측 방법에서 발생하는 이전 작업과 현재 작업 간의 인과 관계가 없으므로 캐시 미스 (Miss)와 잦은 외부 메모리 접근(Access) 모두에서 이점을 얻을 수 있다<sup>[21]</sup>.

3.1 절의 내용을 근거로 탐색 영역의 크기를 9x9로 한다면 하나의 움직임 벡터를 구하는 데 있어 576개의 픽셀 값을 필요로 한다. 따라서 미디어 프로세서의 내부 캐시 메모리에 적재될 데이터의 크기를 기존 알고리즘에서

사용되는 2,116개의 픽셀 값보다 약 1/4로 줄일 수 있다. 한정된 9x9 탐색 영역의 탐색 대상 매크로블록의 수는 81개이다. 이는 33x33 영역에서 TSS의 탐색점 33개에 비해 2배 이상 많은 수이다. 따라서 보다 빠른 속도 향상을 위해 탐색 영역을 축소할 필요가 있다. 탐색 영역 내의 최소 SAD 값을 가지는 점이 하나며 최소점에서 멀어질수록 SAD 값이 선형적으로 커진다는 고속 움직임 예측 방법의 가정을 이용하면 단순한 이진 탐색(Binary Search)에 의해 탐색 블록 내의 탐색 영역을 반으로 줄일 수 있다. 탐색 블록 안에서 예측된 움직임 벡터가 속한 라인의 최소 SAD 값을 가지는 점과 그 바로 위의 라인 내의 최소 SAD 값을 가지는 점을 비교하여 SAD 값이 작은 쪽이 속한 영역만을 검색할 수 있다. 이 방법은 첫 번째 과정의 결과가 다음의 행동에 영향을 미치나 단순화가 가능하여 성능 저하가 거의 없이 연산량을 반으로 줄일 수 있는 장점이 있다.

### 3. 제안된 방법

본 논문에서 제안하는 방법을 단계별로 정리하면 다음과 같다. 제안한 방법에 대한 순서도를 그림 6에 도시하였다.

순서도에서는 탐색 영역을 일반적인 값으로 하여 P로 표기하였다.

- 단계 1) 움직임 벡터 예측을 통해 예측 움직임 벡터를 찾는다.
- 단계 2) 예측 움직임 벡터가 속한 수평 라인 L1의 픽셀을 모두 탐색하여 최소 SAD값을 가지는 위치의 SAD1을 찾는다.
- 단계 3) 단계2)에서 탐색한 수평 라인 L1의 윗 수평 라인 L2의 픽셀을 모두 탐색하여 최소 SAD 값을 가지는 위치의 SAD2를 찾는다.
- 단계 4) SAD1과 SAD2의 값을 비교하여 SAD1이 SAD2보다 작은 경우에는 탐색영역에서 L1아래에 있는 모든 영역을 탐색하고, 반대의 경우는 탐색영역에서 L2위에 있는 모든 영역을 탐색하여 최소 SAD값을 가지는 위치를 찾는다.

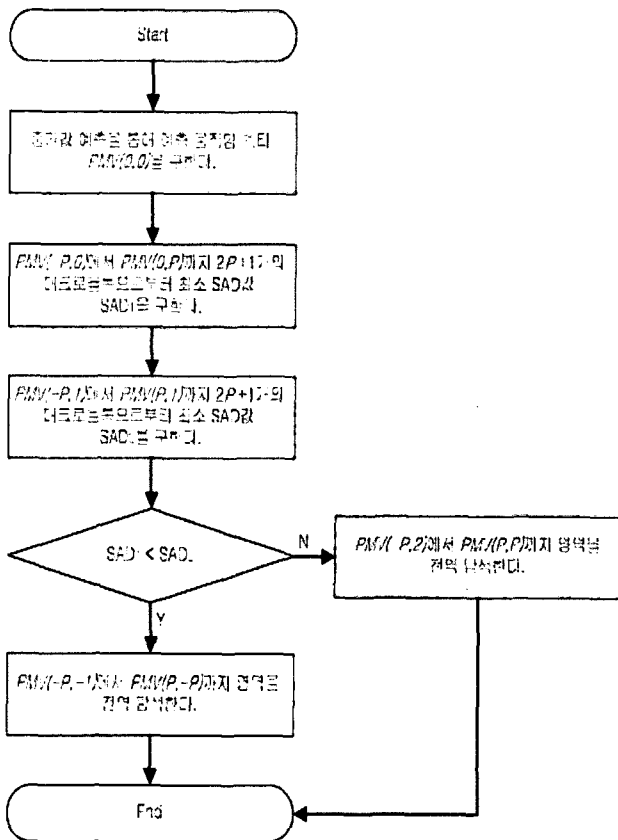


그림 6. 제안된 방법의 순서도  
Fig 6. Flow chart of proposed method

제안된 방법을 적용한 예를 그림 7에 보였다. 그림 7은 탐색영역이 x와 y 방향으로 각각 [-4,+4]이며, 초기에 예

측된 움직임 벡터가 (1)일 때, (1)이 속한 줄 L1 상에서 가장 작은 SAD 값을 가지는 벡터의 위치가 (2)이고, L2 상에서는 (3)이며, (3)의 위치에서의 SAD 값이 더 작아서 L2 라인의 윗 부분 탐색영역에서 가장 작은 SAD 값을 찾은 것이 (4)의 위치인 경우를 보인 것이다. L2 라인의 윗 부분을 모두 탐색했을 때 가장 작은 SAD값이 L2 라인에 존재한다면 움직임 벡터는 L2 라인에서 구한 위치가 된다.

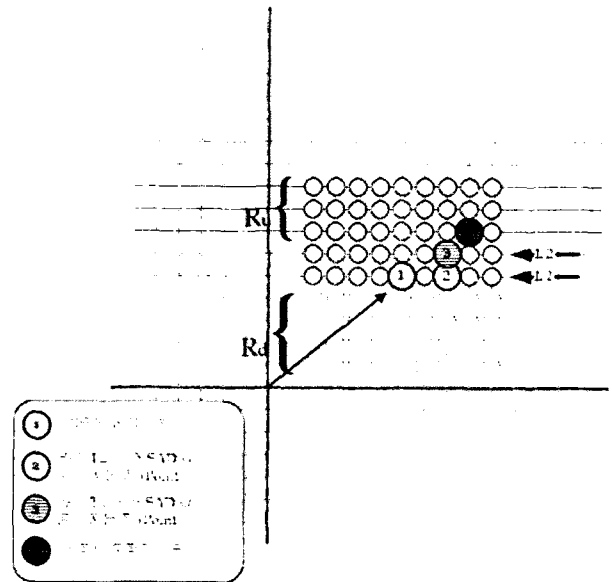


그림 7. 제안된 방법의 예  
Fig 7. Proposed method procedure

#### IV. 실험 및 고찰

본 논문에서 제안한 움직임 탐색 방법의 성능을 측정하기 위하여 제안한 방법, TSS 그리고 전역 탐색 방법인 MPEG-4 SP@L3 비디오 부호화기를 탑재한 Philips사의 미디어 프로세서 TM1300에 구현하여 실험하였다. 사용한 동영상 시퀀스는 움직임이 매우 적은 Akiyo영상, 움직임이 약간 있는 Foreman 영상 그리고 움직임이 큰 Stefan과 Mobile 영상을 사용하였다. 사용된 영상들은 352x288 크기의 CIF 영상이며 비트율을 300 kbps, 200 kbps로 고정하여 30fps로 부호화 하였다. 전역 탐색 방법과 TSS방법은 탐색 영역을 x와 y방향으로 각각 [-15,

15]로 설정하였다.

표 3과 4는 부호화 된 각 영상의 명암 성분에 대해 300 프레임의 평균 PSNR을 구한 것이다. 제안된 방법은 비트율이 300kbps인 경우 TSS보다 향상된 PSNR을 보였고 200kbps인 경우에는 TSS와 유사하였다.

표 3. 300kbps 비트율에서 각 동영상별 명암 성분에 대한 평균 PSNR (단위 dB)

Table 3. Average PSNR for several video sequences at 300kbps (Unit : dB)

	FS	TSS	제안된 방법
Akiyo	42.64	42.25	42.48
Foreman	31.84	31.32	31.45
Mobile	22.97	22.94	22.95
Stefan	25.32	25.03	25.30

표 4. 200kbps 비트율에서 각 동영상별 명암 성분에 대한 평균 PSNR (단위 dB)

Table 4. Average PSNR for several video sequences at 200kbps (Unit : dB)

	FS	TSS	제안된 방법
Akiyo	40.94	40.59	40.47
Foreman	29.70	29.83	29.50
Mobile	22.97	22.95	22.95
Stefan	24.98	24.84	24.99

그림 8~11은 300 kbps로 부호화된 실험 영상의 명암 성분에 대한 프레임별 평균 PSNR을 그래프로 나타낸 것이

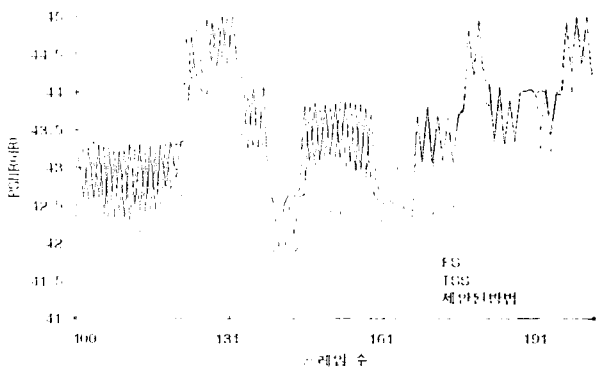


그림 8. Akiyo 영상의 프레임별 명암 성분의 PSNR (300kbps)  
Fig 8. Frame-based luminance PSNR for Akiyo sequence (300kbps)

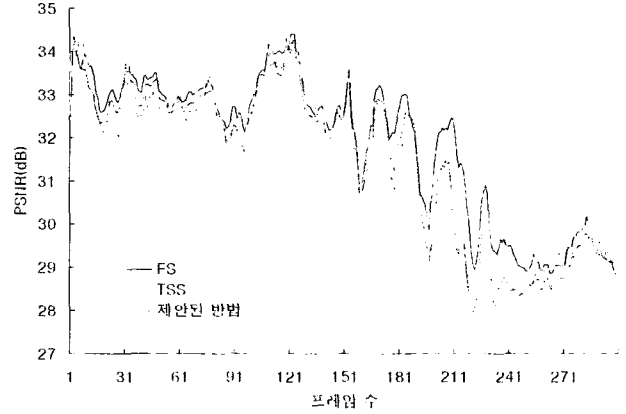


그림 9. Foreman 영상의 프레임별 명암 성분의 PSNR (300kbps)  
Fig 9. Frame-based luminance PSNR for Foreman sequence (300kbps)

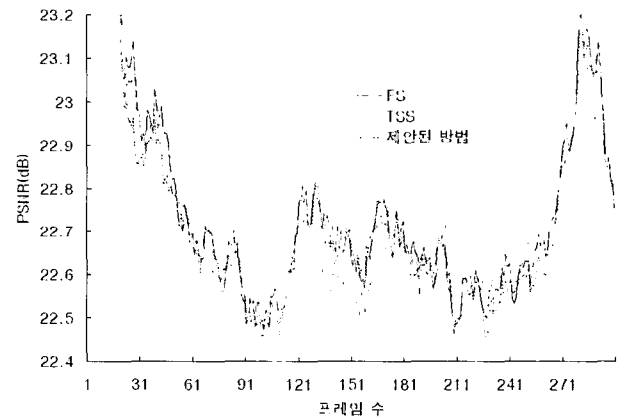


그림 10. Mobile 영상의 프레임별 명암 성분의 PSNR (300kbps)  
Fig 10. Frame-based luminance PSNR for Mobile sequence (300kbps)

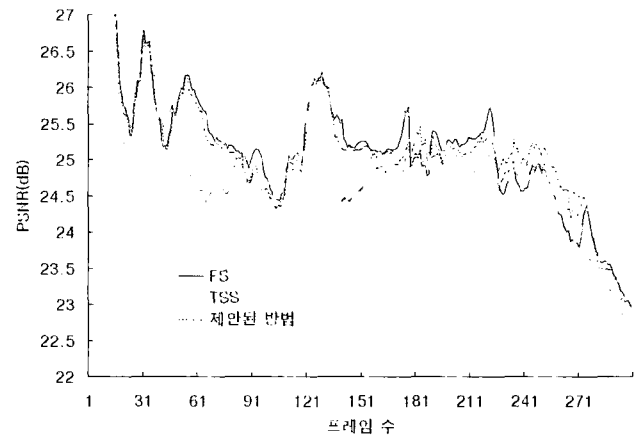


그림 11. Stefan 영상의 프레임별 명암 성분의 PSNR (300kbps)  
Fig 11. Frame-based luminance PSNR for Stefan sequence (300kbps)



다. 이 결과를 통하여 영상이 동일한 방향성을 가지고 부드럽게 움직일 때에 제안된 방법이 좋은 특성을 보인다는 것을 관찰할 수 있다.

표 5~7은 제안된 움직임 탐색 방법과 TSS 방법, 그리고

표 5. 각 동영상별 부호화 속도 (단위 : 프레임/초)  
Table 5. Execution speed for several video sequences (Unit : frame/second)

	FS	TSS	제안된 방법
Akiyo	0.4	5.3	8.1
Foreman	0.4	4.8	7.7
Mobile	0.4	3.9	5.1
Stefan	0.4	4.3	6.1

표 6. 각 동영상별 움직임 예측 모듈의 연산량(단위 : 사이클)  
Table 6. Execution cycles of ME module for several video sequences (Unit : cycles)

	FS	TSS	제안된 방법
Akiyo	14,020,149	1,973,372	595,653
Foreman	14,203,424	1,981,505	591,522
Mobile	14,020,149	1,996,899	675,507
Stefan	14,110,648	1,984,151	704,365

표 7. MPEG-4 부호화기 내에서 움직임 예측 모듈이 차지하는 복잡도(단위 : %)  
Table 7. Complexity evaluations of ME module on MPEG-4 encoder (Unit : %)

	FS	TSS	제안된 방법
Akiyo	28.53	27.28	13.89
Foreman	28.48	25.28	12.71
Mobile	28.53	20.25	9.61
Stefan	27.79	22.59	12.04

전역 탐색 방법의 속도와 복잡도를 비교한 것이다. 이 실험은 동영상 부호화기의 움직임 예측 모듈에 미디어 프로세서의 내부 데이터 캐시 메모리를 제한적으로 할당하여 실험한 것으로써 할당된 크기는 2 킬로바이트(KB) 밑으로 제한하였다. 표 5는 MPEG-4 부호화기를 사용하여 300프레임의 시퀀스들의 전체 부호화 속도를 측정하였다. 제안된 방법이 TSS 방법보다도 계산속도가 약 2배정도 빨라졌음을 알 수 있다. 표 6은 미디어 프로세서에서 각 시퀀스들을 300프레임만큼 부호화하는데 소요되는 움직임 예측 모듈의 동작 사이클 수를 나타낸 것이다. 제안된 방법이 TSS 방법에 비해 동작하는 사이클 수가 약 1/3 크기로 줄어든 것을 알 수 있다. 그리고 표 7은 MPEG-4 전체 부호화기와 움직임 예측 모듈이 차지하는 복잡도의 비교치를 측정하여 정리한 것이다. 표 7을 통해 복잡도가 1/2 이하로 줄어든 것을 볼 수 있다.

그림 12는 300kbps로 부호화된 Stefan 영상의 화질을 보여준다. TSS 방법보다 제안된 방법을 적용하였을 때 화면상에서 문자 영역의 왜곡이나 블록화 현상이 훨씬 감소하였고, 전역 탐색 방법과 유사한 화질을 보여주었다.

## V. 결론

본 논문에서는 예측 움직임 벡터와 영상의 특성을 고려하여 탐색할 영역을 설정하고 그 영역 내에서 미디어 프로세서에 적합한 움직임 벡터를 구하는 방법을 제안하였다. 기존 고속 움직임 예측 방법들은 적은 수의 탐색 점을 가짐에도 불구하고 많은 비교 및 분기의 사용과 비효율적인 메모리 사용으로 미디어 프로세서에 사용하기에는 효율이 떨어진다. 반면, 제안한 방법에서는 기존 고속 움직임 예측 방법 보다 다소 많은 탐색점을 가졌으나 탐색 영역을 정형화시키고 데이터의 흐름을 순차적으로 정규화시켜 미디어 프로세서 내부의 캐시 메모리를 효율적으로 사용할 수 있었다. 더불어 비교 분기 및 제어 명령의 사용을 줄여 외부 메모리의 호출을 작게 할 수 있었다.

제안된 방법은 움직임이 작고 부드러운 영상에서 전역 탐색 방법과 거의 비슷한 화질을 유지하면서 기존의 고속 움직임 예측 방법보다 빠른 속도를 보였다. 실제 TM1300 미디어 프로세서를 이용한 실험을 통하여 움직임 예측 모

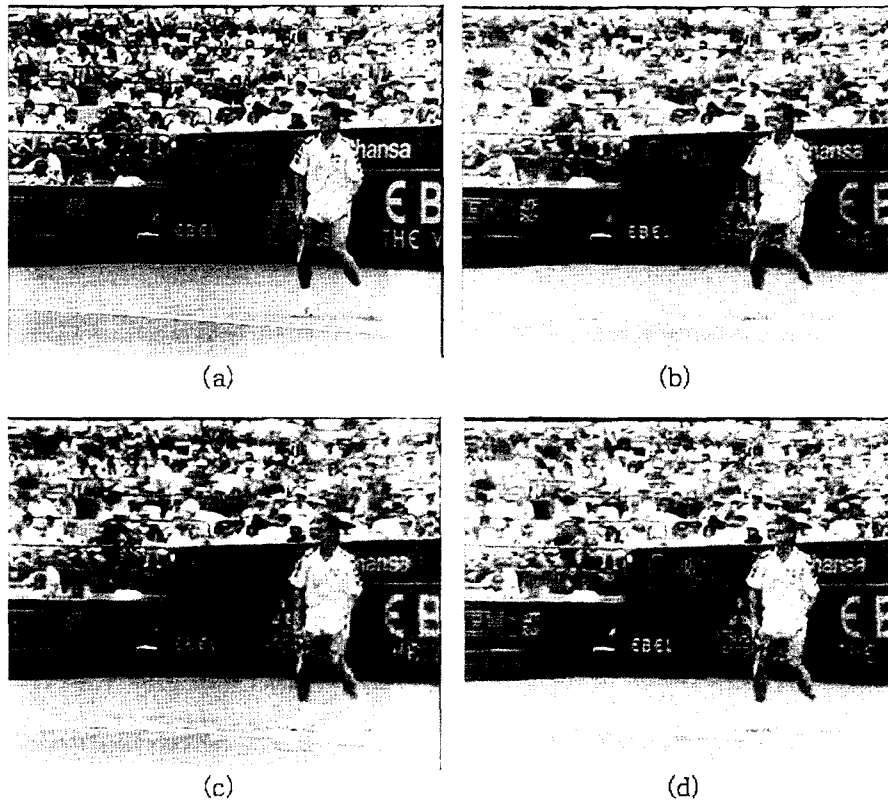


그림 12. 300kbps로 부호화된 Stefan 영상의 시각적 화질 비교 (a) 원영상 (b) FS (c) TSS (d) 제안된 방법  
 Fig 12. Comparison of subjective quality for Stefan sequence at 300kbps (a) Original frame (b) FS (c) TSS (d) Proposed method

들의 속도가 3배 이상 향상됨을 알 수 있었다. 따라서 제안된 방법은 미디어 프로세서를 사용하여 동영상을 실시간 부호화하는 각종 서비스 및 시스템에 사용될 수 있다. 특히 크기와 전력 소모량 등의 자원에 제한이 많은 모바일 환경에서 동영상 비디오 실시간 부호화기에 사용되는 미디어 프로세서들의 효율을 높임으로써 상대적으로 사용자에게 보다 나은 양질의 서비스를 공급할 수 있다.

참 고 문 헌

[1] A. Murat Tekalp, Digital Video Processing, Prentice Hall, 1995.  
 [2] ISO/IEC, JTC1/SC29/WG11, Generic Coding of Audio-Visual Objects - Part 2 : Visual, IS 14496-2, Dec. 1998.  
 [3] ISO/IEC, JTC1/SC29/WG11, MPEG-4 Video Verification Model Version 17.0, N3515, Jul. 2000.  
 [4] ISO/IEC, JTC1/SC29/WG11, Generic coding of Audio -Visual Objects - Part : 5 Simulation Software, IS 14496-5, Dec. 1998.  
 [5] A. Dasu and S. Panchanathan, "A Survey of Media Processing Approaches," IEEE Trans. Circuits Syst. Video Technol, Vol. 12, No. 8, pp. 633-645, Aug. 2002.  
 [6] B. Furht, J. Greenberg and R. Westwater, Motion Estimation Algorithms for Video Compression, Florida Atlantic Univ, 1997.  
 [7] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in Proc. NTC81, New Orleans, LA, pp. C9.6.1.-9.6.5, Nov. 1981.  
 [8] S. Zhu and K.-Kuang, Ma, "A New diamond search algorithm for fast block-matching motion estimation," IEEE Trans. on Image Processing, Vol. 9, No 2, pp. 287-290, Feb 2000.  
 [9] X. Jing and L. P. Chau, "An Efficient Three-Step Search Algorithm for Block Motion Estimation," IEEE Trans. on Multimedia, Vol. 6, No. 3, pp. 435-438, Jun. 2004.  
 [10] M. Gallant, G. Côté, and F. Kossentini, "An Efficient Computation-Constrained Block-Based Motion Estimation Algorithm for Low Bie Rate Video Coding," IEEE Trans. on Image Processing, Vol. 8, No. 12, pp. 1816-1823, Dec. 1999.  
 [11] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) -

- Enhancing Block Based Motion Estimation," in proceedings of Visual Communications and Image Processing 2001 (VCIP-2001)
- [12] K.M. Uz, M Vetterli, and D. LeGall, "Interpolative Multiresolution Coding of Advanced Television with Compatible Subchannels," IEEE Trans. Circuits Syst. Video Technol., Vol. 1, No. 1, pp. 86-99, Mar. 1997.
- [13] G. Fujita, T. Onoye, and I. Shirakawa, "A VLSI Architecture for Motion Estimation Core Dedicated to H,263 Video Coding," IEEE Trans. Electron., Vol. E81-C, No. 5, pp. 702-707, May, 1998.
- [14] J. A. E. G. Richardson, Video Codec Design, WILEY, 2002.
- [15] Y. Nie and K. K. Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," IEEE Trans. on Image Processing, Vol. 11, No. 12, pp. 1442-1449, Dec. 2002.
- [16] Philips, TM1300 Preliminary Data Book, Aug. 1999.
- [17] Y. W. Huang, S. Y. Ma, C. F. Shen, and L. G. Chen, "Predictive Line Search : An Efficient Motion Estimation Algorithm for MPEG-4 Encoding Systems on Multimedia Processors," IEEE Trans. Circuits Syst. Video Technol. Vol. 13, No. 1, pp. 111-117, Jan, 2003.
- [18] 장호연, 명진수, 오승준, 정광수 "시공간 특성을 이용한 고속 움직임 벡터 예측 방법," 제15회 영상처리 및 이해에 관한 워크샵, pp.581-586, 2003.
- [19] 이성호, 명진수, 오승준, 정광수, "움직임 특성을 이용한 새로운 고속 움직임 예측 방법," 신호처리 합동 학술대회 2002, Vol. 15, No. 1, pp.295
- [20] Philips Semiconductors, Book2 - Cookbook Part D : Optimizing TriMedia Application, Oct. 1999.

---

## 저 자 소 개



### 노 대 영

- 2001년 2월 : 광운대학교 전자공학부 졸업 (공학사)
- 2003년 8월 : 광운대학교 전자공학과 대학원 졸업(공학석사)
- 2003년 9월~현재 : 광운대학교 대학원 전자공학과 박사과정
- 주관심분야 : Motion Estimation, 영상 처리 시스템



### 김 성 훈

- 2001년 2월 : 광운대학교 전자공학부 졸업 (공학사)
- 2001년 3월~2003년 8월 : 광운대학교 대학원 전자공학과 졸업(공학석사)
- 2003년 9월~현재 : 광운대학교 대학원 전자공학과 박사과정
- 주관심분야 : Real Time System, DMB



### 손 채 봉

- 1993년 2월 : 광운대학교 전자공학과 졸업 (공학사)
- 1995년 2월 : 광운대학교 전자공학과 졸업 (공학석사)
- 2000년 2월 : 광운대학교 전자공학과 (공학박사수료)
- 2000년 3월~현재 : 한양여자대학교 인터넷정보과 전임강사
- 2001년 3월~현재 : (주) 손랩 대표이사
- 한국방송공학회 종신회원, MPEG포럼 회원
- 주관심분야 : DMB시스템, DTV시스템, 디지털 신호처리

저자 소개



오 승 준

- 1980년 2월 : 서울대학교 전자공학과 졸업(공학사)
- 1982년 2월 : 서울대학교 전자공학과 대학원 졸업(공학석사)
- 1988년 5월 : 미국 Syracuse University 전기/컴퓨터공학과 졸업(공학박사)
- 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 컴퓨터공학부 겸임교수
- 1992년 9월~현재 : 광운대학교 전자공학부 교수 (멀티미디어연구실)
- 2004년 9월~현재 : 광운대학교 산학협력단 VIA-Multimedia Center 센터장
- 2002년 3월~현재 : SC29-Korea 의장 및 MPEG Forum 부의장
- 주관심분야 : 비디오 데이터 처리, 비디오 코덱 시스템, 비디오 및 영상압축, 멀티미디어시스템



안 창 범

- 1981년 2월 : 서울대학교 전자공학과 졸업(학사)
- 1983년 8월 : 한국과학기술원 전기 및 전자공학과 졸업(석사)
- 1986년 8월 : 한국과학기술원 전기 및 전자공학과 졸업(박사)
- 1986년 9월~1991년 4월 : 미국 University of California, Irvine 연구조교수
- 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, 영상시스템