논문 04-29-9C-13 한국통신학회논문지 '04-9 Vol.29 No.9C

# 움직임 추정오차의 예측을 이용한 고속 움직임 추정 방법

정회원 강 현 수*

# A fast motion estimation method using prediction of motion estimation error

Hyun-Soo Kang* *Regular Members*

요 약

본 논문은 고속 전역탐색법 중의 하나인 MSEA(multi-level successive elimination algorithm)를 개선한 방식으로서, MSEA의 단계에 따른 norm의 계산 결과를 이용하여 최종 단계의 SAD를 예측함으로써 더 이상의 단계를 수행할 필요가 없다고 판단되는 단계의 계산을 생략함으로써 계산량을 감소시키는 방법을 제안한다. 각 단계별 SAD의 예측을 위해 norm에 대한 이론적 분석이 이루어지며 실험을 통해 분석내용을 검증하고, 이를 바탕으로 새로운 알고리즘을 제안하고 실험을 통해 제안된 알고리즘의 성능을 평가한다.

**Key words :** motion estimation, MSEA, full-search algorithm, video coder

Abstract

This paper presents an enhanced MSEA(multi-level successive elimination algorithm) which is a fast algorithm of the full-search motion estimation. We predict the SAD at the final level using the values of norms at the preceding levels in MSEA and then decide on whether the processing at the following levels should be proceeded or not. We skip the computation at the following levels where the processing is not meaningful anymore. Consequently, we take computational gain. For the purpose of predicting the values of SAD at each level, we first show the theoretical analysis of the value of norm at each level, which is verified by experiments. Based on the analysis a new motion estimation method is proposed and its performance is evaluated.

## Ⅰ. 서 론

움직임 추정 (motion estimation: ME)은 비디오 시퀀스의 시간 방향 잔여성분 (redundancy)를 줄이는데 효과적이므로 H.261, H.263, MPEG-1, MPEG-2, MPEG-4 등의 비디오 압축 표준에서 뿐만 아니라 움직임 보상 부호화 기법이 적용된 비디오 부호화기에 널리 채용되고 있다. 그래서 좋은 복원 화질을 유지하면서, 좀 더 빠르고 정확한 움직임 벡터를 찾을 수 있는 방법이 요구된다. 전역탐색법 (full search algorithm)은 최적의 움직임 벡터를 찾는 반면 많은 계산량이 요구된다. 그래서 이러한 단점을 해결하기 위해, 많은 고속 알고리즘들이 제안되어 왔다. 예를 들어, 2-D logarithmic 탐색법, 3단

* 중앙대학교 첨단영상대학원 영상공학과 (hskang@cau.ac.kr)
논문번호 : 040034-0119, 접수일자 . 2004년 xx월 xx일
※이 논문은 2002학년도 중앙대학교 학술연구비 지원에 의한 것임

계 탐색법, conjugate direct 탐색법, cross 탐색법, 4단계 탐색법, diamond 탐색법 등이 있다 [1][2][3][4].

또한, 전역탐색법 자체에 대한 고속화 알고리즘 또한 활발 연구되었는데, PDE(partial difference elimination algorithm), SEA (successive elimination algorithm), MSEA (multi-level SEA) 등이 대표적이다. PDE는 MPEG의 reference s/w에서 구현되어 있는 방식으로서, SAD (sum of absolute difference)의 계산 과정 중 이전의 최소 SAD를 초과하는 경우 SAD의 계산을 더 이상 수행하지 않는 방식이다[5][6][7] SEA는 블록의 평균값으로부터 최적 벡터가 될 수 있는지의 여부를 판단하고 전체 블록에 대한 SAD의 계산이 불필요한 블록에 대한 계산을 수행하지 않음으로써 계산량을 감축하는 방법이다[8]. MSEA는 SEA의 다계층 접근방식으로서 SEA와 기본적인 개념은 동일하며 상대적은 많은 메모리를 요구하는 반면 계산량을 크게 줄이는 방법이다[9][10]. 여기서 언급된 PDE, SEA, MSEA는 모두 전역탐색법과 동일한 성능을 보이면서 계산량을 줄이는 고속 전역탐색법이라고 할 수 있다

본 논문은 MSEA의 단계에 따른 norm의 계산 결과를 이용하여 최종적인 SAD를 추정함으로써 더 이상의 단계를 수행할 필요가 없는 단계의 계산량을 생략함으로써 계산량을 감소시키는 방법을 제안한다 따라서, 제안된 방법의 성능은 SAD 추정 성능과 밀접한 관계가 있다. 정확하지 않은 추정 결과는 최적움직임 벡터를 찾지 못하는 결과를 초래하기 때문이다.

## Ⅱ.MSEA의 소개

Li [8]의 방법은 MSEA의 1단계 SEA라고 말할 수 있다. MSEA에서는 첫번째 단계로서 $N\times N$ 블록에 대한 부등식으로부터 고려하지 않아도 되는 탐색점을 제외시키고, 두번째 단계로서 $N\times N$ 블록을 4개의 $N/2\times N/2$ 부블록으로 나누고 부블록의 화소값의 합에 대한 부등식으로부터 고려하지 않아도 되는 탐색점을 제외시킨다. 이러한 과정을 반복함으로써, 계산량을 크게 감소시키는 방법이 MSEA이다. MSEA의 가장 핵심되는 부등식을 이끌어내기 위하여 다음과 같은 비용함수를 정의한다.

$$AAD_k(x,y)=\sum_{i=1}^{2^k}\sum_{j=1}^{2^k}|f_k(i,j,t)-f_k(i+x,j+y,t-1)|,\quad k=0,1,2,\ldots,P \tag{1}$$

where

$$f_{k-1}(i,j,t)=f_k(si-1,2j-1,t)+f_k(2i-1,2j,t)+f_k(2i,2j-1,t)+f_k(2i,2j,t) \tag{2}$$

식(1)에서 $AAD_0(x,y)$은 현재 블록과 움직임 벡터 $(x,y)$가 가리키는 이전블록의 차이 블록의 절대값의 합임을 알 수 있다. Minkowski의 부등식 $|A-B|\geq\|A|-|B|$를 적용하면 다음과 같은 부등식을 얻을 수 있다.

$$AAD_P(x,y)\geq AAD_{P-1}(x,y)\geq\quad\geq AAD_0(x,y) \tag{3}$$

현재까지의 탐색위치 중 최적의 위치가 $(m,n)$이라고 하고 움직임 벡터 $(m,n)$의 추정 오차를 $SAD(m,n)$이라고 하자. 이 때, $SAD$를 $AAD$로 표현하면 $SAD(m,n)=AAD_P(m,n)$로 나타난다. 만약, 어떤 단계 k에 대해서$AAD_k(x,y)$가 $SAD(m,n)$보다 크다면, $(r,y)$는 절대로 최적 움직임 벡터가 될 수 없다. 역으로 말하면, 모든 k에 대해서 $AAD_k(x,y)$가 $SAD(m,n)$ 작다면, $(x,y)$가 최적의 움직임 벡터일 수 있음을 의미한다. 그래서, 어떤 k에 대해 $AAD_k(x,y)$가 $SAD(m,n)$보다 크다면 탐색점에서 제외함으로써 계산량을 줄일 수 있다. 따라서, MSEA는 $k=0$에 대해, $AAD_0(x,y)$가 $SAD(m,n)$보다 크다면 $(x,y)$를 탐색점에서 제외시키고, 그렇지 않은 경우 k=1에 대해 다시 $AAD_1(x,y)$와 $SAD(m,n)$를 비교하여 $(x,y)$를 탐색점에서 제외시킬지를 결정하게 된다. 이러한 과정이 $k>1$에 대해서도 반복적으로 수행된다. 이것은 식(3)의 관계를 이용한 것으로, $AAD_k(x,y)$와 $SAD(m,n)$는 다음의 관계를 가진다.

$$AAD_k(x,y)\leq SAD(m,n)\quad k=0,1,2,\ldots,P \tag{4}$$

모든 k에 대해 식(4)를 만족할 경우 $(x,y)$가 최적의 움직임 벡터일 수 있으므로 탐색점에서 제외시키지 않고, 모든 $k$에 대해 식(4)를 만족하므로

$SAD(m,n)$는 $SAD(x,y)$로 갱신되어 식(4)의 범위가 더욱 좁아지게 된다.

MSEA에서의 계산량 감축량은 $SAD(m,n)$가 초기에 얼마나 작은 값이 나오도록 탐색점을 설정하는가에 크게 의존한다. 그래서 가장 널리 사용되는 탐색 방법은 나선형 탐색법이다. 이는 움직임벡터 $(0,0)$ 또는 예측 움직임 벡터(MVP)를 중심으로 회전하면서 반경을 넓혀가는 방식으로 탐색하는 방법이다.

## Ⅲ. $AAD_k(x,y)$의 특성에 대한 고찰

### 1 이론적 특성

본 절에서는 단계에 따른 AAD의 평균과 분산을 확률적으로 얻고자 한다. 계산의 편의를 위해 $f(i,j,t)$와 $f(i+x,j+y,t-1)$의 차신호 $d(i,j)=f(i,j,t)-f(i+x,j+y,t-1)$가 i.i.d. (independent and identical density) 이고, $N(0,\sigma^2)$의 Gauss 분포를 가진다고 가정하자. 그리고, $(x,y)$가 최적 움직임 벡터인 경우, 차신호 $d(i,j)$는 최적 움직임 벡터를 이용한 시간 방향 예측기이므로 예측오차신호 간의 상관성은 무시할 수 있다.

이 때, 식(2)의 관계에 의해 $d_{k-1}(i,j)=f_{k-1}(i,j,t)-f_{k-1}(i+x,j+y,t-1)$는 $N(0,4\sigma_{d_k}^2)$ 의 Gauss 분포를 가진다. 여기서 $\sigma_{d_k}^2$는 $d_k(i,j)$의 분산이다. 즉, 다음과 같은 관계를 갖는다.

$$\sigma_{d_{k-1}}^2=4\sigma_{d_k}^2 \tag{5}$$

여기서 가장 하위 level $(K=P)$에서 $\sigma_{d_P}^2=\sigma^2$이므로, 결과적으로 다음과 같은 결과를 얻을 수 있다.

$$\sigma_{d_k}^2=4^{P-k}\sigma^2 \tag{6}$$

이 때, $X(i,j)=|d_k(i,j)|$라고 할 때, $X(i,j)$의 확률밀도함수는 다음과 같이 주어진다.

$$f_X(x)=\frac{2}{\sqrt{2\pi}\sigma_k}exp\left(-\frac{x^2}{2\sigma_k^2}\right)U(x) \tag{7}$$

여기서 $U(x)$는 단위계단함수(unit step function)이고 편의상 $\sigma_k^2=\sigma_{d_k}^2$로 두었다. $X(i,j)$의 분포가 주어졌으므로 평균은 다음과 같이 쉽게 얻을 수 있다.

$$\mu_X=\int_{-\infty}^{\infty}xf_X(x)dx=\sqrt{\frac{2}{\pi}}\sigma_k \tag{8}$$

식(8)을 이용함으로써 $AAD_k(x,y)$의 평균은 다음과 같이 주어진다.

$$\mu_{AAD_k}\equiv E[AAD_k(x,y)]=E\left[\sum_{j=1}^{2^k}\sum_{i=1}^{2^k}X(i,j)\right]$$
$$=\sum_{j=1}^{2^k}\sum_{i=1}^{2^k}E[X(i,j)]=2^{2k}\mu_X \tag{9}$$

여기서 식(6)을 이용하면 식(9)는 다음과 같이 표현된다.

$$\mu_{AAD_k}=\sqrt{\frac{2}{\pi}}2^{k+P}\sigma \tag{10}$$

이 결과는 $\mu_{AAD}$는 2의 지수 배에 비례함을 보여준다.

### 2. 실험적 특성

본 절에서는 앞 절에서 분석한 이론적인 결과가 실험적인 결과와 일치하는지에 대해 조사하고자 한다. 그림1은 여러 가지 영상에 대해 측정한 $AAD_k(x,y)$의 평균을 보여주고 있다. 실험결과는 식(3)과 식(10)에서 유도한 바와 같이 단조 증가하는 형태를 띄며, 그 증가량은 Foreman영상을 제외하고는 2의 지수 승에 개략적으로 비례함을 보여준다. 좀 더 정확 살펴보면, 식(10)에서 유도한 2의 지수 승에 비례보다는 실험 결과가 2의 지수 승의 증가보다는 적음이 관찰된다. 특 , foreman 영상의 경우, 평균의 증가가 2의 지수 승이라고 보다는 선형적으로 증가함을 알 수 있다. 또한, 영상에 따라 그 증가량에 차이가 있음을 알 수 있다. 이것은 i.i.d의 가정에 잘 부합하는 영상일수록 이론적인 결과와 잘 일치하는 것으로 판단된다. 결론적으로, 이론적인 분석 결과를 따르지 않는 영상이 존재함을 알 수 있다. 따라서, 본 논문에서는 실험적 결과를 활용하여 이론적 분석 결과를 보충함으로써 새로운

알고리즘을 제안한다.

## Ⅳ. 제안된 알고리즘

본 장에서는 MSEA의 계산량을 감축하는 방법을 제안한다. 제안된 방법는 AAD의 평균에 대한 이론적, 실험적 결과에 근거하여 다음과 같이 이루어진다.

① $(x, y) = (0, 0)$에 대해 SAD를 계산하고 $SAD_{\min} = SAD(0.0)$, $n = 0$, $(x^*, y^*) = (0, 0)$으로 초기화한다. 여기서 $n$은 탐색점의 개수를 헤아리는 변수, $(x^*, y^*)$는 최적움직임 벡터이다.

② 탐색점을 정해진 규칙에 의해 할당한다. 즉, $(x, y) = f(n)$ 여기서 $f(n)$은 탐색전략에 따라 생성된 lookup table이고, 일반적으로 나선형 탐색법을 사용한다. 예를 들어, $n = 0$에 대해 $(0.0)$, $n = 1$에 대해 $(-1, 0)$, $n = 2$에 대해 $(-1.-1)$, …, 등으로 탐색점을 생성하는 함수이다.

③ $k = 0$으로 초기화 한다. 여기서 $k$는 해상도의 단계이다.

④ $AAD_k(x, y)$를 계산한다.

⑤ $k = 0$이면, $k \leftarrow k + 1$로 갱신하고, 과정4로 되돌아간다.

⑥ $AAD_k(x, y) \geq SAD_{\min}$이면 최적움직임 벡터가 될 수 없으므로 다음 탐색점에 대해 조사하기 위해 $n \leftarrow n + 1$로 하고 과정2로 되돌아간다

⑦ $AAD_k(x, y) < SAD_{\min}$이면 최적움직임 벡터가 될 수 있으므로, SAD의 추정을 수행한다. SAD의 추정치은 다음 식에 의해 이루어진다.

$$EST[SAD(x,y)] = \begin{cases} \dfrac{AAD_k - AAD_0}{k} \times L + AAD_0, & 0 < k < L \\ AAD_k, & k = L \end{cases}$$

여기서 $L$은 블록크기를 나타내는 값으로서 블록크기가 $D$일 때, $D = 2^L$이다.

⑧ $EST[SAD(x, y)] \geq SAD_{\min}$이면 최적움직임 벡터가 될 수 없다고 판단하여, 과정2로 간다.

⑨ $EST[SAD(x, y)] < SAD_{\min}$이면 최적움직임 벡터가 될 수 있다고 판단하여, 다음 단계에 대해 조사하기 위해, $k \leftarrow k + 1$로 하고 과정4로 되돌아간다. 만약, $k = L$이면, $(x, y)$를 최적 움직임 벡터 후보로 선정하여 $SAD_{\min} \leftarrow EST(SAD(x, y))$, $(x_*, y_*) \leftarrow (x, y)$로 저장(갱신)하고, $n$이 마지막 탐색점이 아니면 다음 탐색점에 대해 조사하기 위해 $n \leftarrow n + 1$로 하고 과정2로 되돌아간다. 만약 $n$이 마지막 탐색점이면 과정을 종료한다.

기존의 MSEA와의 차이점은 SAD를 추정하는 과정이 추가되고, 추정된 SAD로부터 최적 움직임 벡터 후보가 될 수 있는지를 판단하는 과정이 추가되었다는 점이다. 여기서 사용된 추정 방식은 과정7의 수식에서 나타난 바와 같이 단계별로 얻어진 AAD를 이용하여 선형적으로 추정하였다, 즉, $AAD_0$와 $AAD_k$ 사이를 선형적으로 연결하여 SAD를 계산하였다. 그림2는 $k = 2$, $L = 4$의 경우에 대한 예로서, $AAD_0$와 $AAD_2$에 선형 비례하도록 SAD를 추정하였다. 이는 $AAD_k$의 평균이 2의 배수에 비례하는 이론적 분석 결과에 비해 작은 값으로 추정되도록 함으로써, $EST[SAD]$가 $AAD_k$보다는 크도록 추정하되, 이론적인 결과보다는 작도록 추정함으로써, 최적 움직임 벡터이면서 최적 움직임 벡터 후보에서 배제되는 경우를 최소화하기 위함이다. 즉, 실험적인 결과에서 이론적인 결과를 따르지 않는 경우를 고려한 결과이다. 예를 들어 Foreman영상의 경우, 단계의 증가에 따라 $AAD_k$가 2의 배수에 비례하기 보다는 선형적으로 증가하는 특성을 가지고 있다. 이러한 추정 방식에서 주목할 점은, 기존의 MSEA의 k번째 단계에서 배제되지 않았던 탐색점이 배제됨으로써 계산량의 감축효과를 얻을 수 있다. 그러나, 잘못된 추정으로 인하여 최적움직임 벡터를 최적움직임벡터 후보군에서 제외시키는 오류를 범할 수 있다. 이러한 오류와 계산량 사이의 trade-off는 실험을 통해 보이도록 할 것이다.

제안된 방법은 기존의 MSEA에 과정7의 SAD추정을 위한 계산만을 추가함으로써 이루어질 수 있기 때문에 이로 인한 계산량의 증가는 거의 없으며 추정으로 인한 계산량 감축효과를 얻을 수 있음에 주목하자.

## V. 실험결과

본 장에서는 제안된 방법과 MSEA의 방법의 성능을 비교하기 위하여, 비디오 부호화기를 적용하지 않는 경우와 적용한 경우에 대해 각각 실험을 수행하였다. 부호화기를 적용하는 경우, 적용된 부호화기는 MPEG-4로서 MPEG-4 reference S/W인 MoMuSys에 제안된 방법과 MSEA를 구현하여 비교 실험하였다.

### 1. 비디오 부호화기를 적용하지 않은 경우

제안된 방법의 실험을 위해 16x16 블록단위의 움직임 추정, 16의 탐색영역, 그리고 실험영상은 다음과 같다

- Foreman QCIF, 30Hz, 100장 (frame no. 0-99)
- Coastguard QCIF, 30Hz, 100장 (frame no. 0-99)
- Container QCIF, 30Hz, 100장 (frame no. 0-99)
- Carphone QCIF, 30Hz, 100장 (frame no. 0-99)

표1은 원영상에 대한 움직임 추정 결과를 보여주고 있다. 표의 'PSNR'은 100장의 영상에 대한 평균 PSNR, 'Computational Complexity'은 전역탐색법의 계산량을 1로 하였을 때 각 방법에 대한 복잡도, 'Missing rate'는 최적 움직임 벡터를 찾지 못한 블록의 개수를 전체 블록의 개수로 나눈 값으로 최적 움직임 벡터를 찾지 못하고 놓치는 경우의 빈도를 나타낸다.

표에서 보는 바와 같이, Foreman, Coastguard, Container, Carphone 영상에서, 화질은 각각 0.21dB, 0.02dB, 0.00dB, 0.15dB의 저하가 있지만, 계산량에 있어서 기존의 MSEA의 66%, 50%, 80%, 56%에 불과하여 계산량 감축효과가 있음 보이고 있다.

특히, Coastguard와 Container 영상의 경우, 제안된 방법이 최적 움직임 벡터를 제외시키는 경우가 거의 발생하지 않아 같은 화질을 유지하면서 계산량의 감축 효과를 얻을 수 있음을 보여주고 있다.

한편, Foreman과 Carphone영상의 경우 각각 missing rate가 약10%, 8% 정도로서 최적 움직임 벡터를 제외시키는 경우가 Coastguard와 Container 영상의 경우보다 다소 높아 약간의 화질 저하를 가져왔다. 앞 장의 AAD의 실험 결과에서 보았듯이 Foreman과 Carphone 영상의 경우는 이론적으로 분석한 AAD의 특성과 실험적으로 구한 AAD 의 특성이 다름에 따라, 추정에 따른 missing rate의 증가가 화질 저하의 원인으로 분석된다. 그러나, 최적 움직임 벡터를 제외하였다고 하더라도 AAD에 기초하여 부최적 움직임 벡터를 찾게 되므로 심각한 화질 저하를 초래하지는 않음을 알 수 있다.

### 2. MPEG-4 부호화기에 적용한 경우

MPEG-4 부호화기(MoMuSys)에 MSEA와 제안된 방법을 적용하였다. 이전 절의 실험 환경과 동일하게 16x16 블록단위의 움직임 추정, 16의 탐색영역, 목표 비트량은 64kbps, 그리고 실험영상은 다음과 같다.

- Foreman QCIF, 15Hz, 50장 (frame no. 0-98)
- Coastguard QCIF, 15Hz, 50장 (frame no. 0-98)
- Container QCIF, 15Hz, 50장 (frame no. 0-98)
- Carphone QCIF, 15Hz, 50장 (frame no. 0-98)

표2는 움직임 추정 결과를 보여주고 있다. 표2에서 볼 수 있듯이, 목표 비트율을 64kbps로 정하였으나, 실제 발생비트량은 완전히 같지 않으나 거의 비슷한 비트량이 발생되었다고 할 수 있다. 이 때 PSNR과 CPX를 비교해보면, PSNR에 있어서 제안된 방법과 기존의 MSEA가 거의 비슷한 값을 가지는 반면, 제안된 방법이 계산량에 있어서 이득이 있음을 관찰할 수 있다. Coastguard 의 경우 기존의 MSEA 계산량의 약 42%와 약 37%에 해당함으로써 많은 계산량 이득을 얻을 수 있음을 알 수 있다.

## VI. 결 론

본 논문은 MSEA의 단계에 따른 AAD의 계산결과를 이용하여 최종적인 SAD를 추정함으로써 더 이상의 단계를 수행할 필요가 없다고 판단되는 경우, 그 단계의 계산을 생략함으로써 계산량을 감소시키는 방법을 제안하였다. SAD의 추정을 위해 AAD에 대한 이론적 분석이 이루어졌으며, 실험을 통해 분석내용의 정확성에 대해 조사였다. 그리고, 실험적 특성과 이론적 특성을 바탕으로 SAD를 추정하는 알고리즘을 제안하였으며, 4가지의 동영상에 대해 실험하여 제안된 방식의 우수성을 입증하였다. 실험은 부호화기를 적용하지 않은 경우와 적용한 경우에 대해서 각각 수행되었으며, 각 경우에 대해 모두 제안된 방식이 계산량에 있어서 이득이 있음

을 보였다. 결과적으로 제안된 방식은 기존의 MSEA와 비슷한 화질을 유지하면서 계산량에 있어서 20%~50%의 감축을 가져왔다

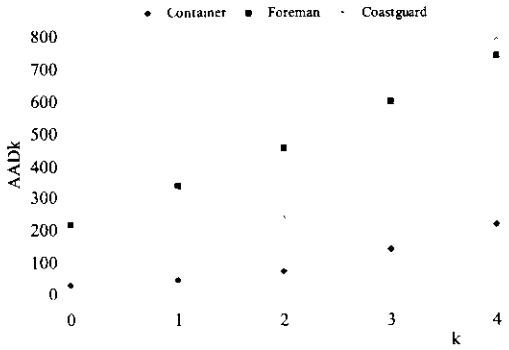



그림1. 단계에 따른 AAD

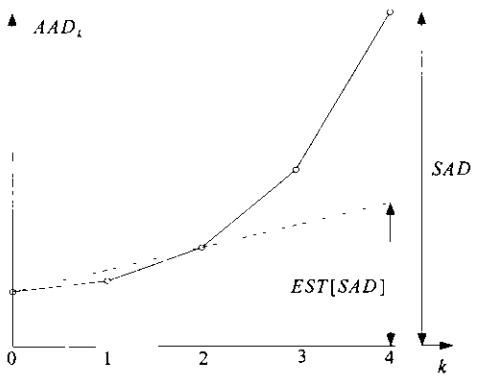



그림2 SAD의 추정 방법 (L=4, k=2의 경우의 예)

표1 원영상에 적용한 MSEA와 제안된 방법의 성능 비교

| Method | Image | PSNR | Computational Complexity | Missing Rate |
|---|---|---|---|---|
| MSEA | Foreman QCIF | 32 68 | 0.0140 | 0.000 |
| Proposed | | 32 47 | 0.0092 | 0.102 |
| MSEA | Coastguard QCIF | 32.22 | 0.0344 | 0.000 |
| Proposed | | 32.20 | 0 0172 | 0.011 |
| MSEA | Container QCIF | 43 67 | 0.0750 | 0 000 |
| Proposed | | 43.67 | 0 0598 | 0.003 |
| MSEA | Carphone QCIF | 34.11 | 0 0252 | 0.000 |
| Proposed | | 33.96 | 0 0143 | 0.083 |

표2 MPEG-4 부호화기에 적용한 MSEA와 제안된 방식의 성능 비교

| Image | MSEA | | | Proposed Method | | |
|---|---|---|---|---|---|---|
| | PSNR | Total bits | CPX | PSNR | Total Bits | CPX |
| Foreman | 31.75 | 209424 | 0 0283 | 31.71 | 209928 | 0 0153 |
| Coastguard | 29.07 | 208632 | 0.1604 | 29.16 | 209808 | 0 0668 |
| Container | 35.25 | 189840 | 0.0440 | 35.36 | 191496 | 0.0404 |
| Carphone | 34 14 | 206096 | 0 0207 | 34 04 | 206584 | 0 0203 |

* CPX : computational complexity

## 참 고 문 헌

[1] F.Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," Proc. IEEE, vol, 83, pp. 858-879, June 1995.

[2] 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.

[3] L.K.Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding,"IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 419-423, Aug. 1996.

[4] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation,"in Proc. Int. Conf. Inform., Comm., Signal Processing, Singapore, Sept. 9-12, 1997, pp. 292-296.

[5] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," Proc. SPIE, vol. 2419, pp. 100-118, 1995

[6] "ITU-T recommendation H.263 software implementation," Digital Video Coding Group, Telenor R&D, 1995.

[7] J. N. Kim and T S. Choi, "A fast full-search motion-estimation algorithm using representative pixels and adaptive matching

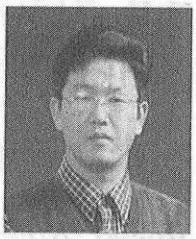
scan,"IEEE Trans. on CSVT, vol. 10, no. 7, pp. 1040-1048, Oct. 2000.

[8] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. on Image Processing, vol. 4, no. 1, pp. 105-107, Jan. 1995

[9] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation,"IEEE Trans. on Image Processing, vol. 9, no. 3, pp. 501-504, March 2000.

[10] J. Y. Lu, K. S. Wu and J. C. Lin, "Fast full search in motion estimation by hierarchical use of Minkowski's inequality," Pattern Recognition, vol. 31, no. 7, pp. 945-952, pp. 945-952, 1998.

강 현 수(Kang Hyun-Soo) 정회원



1999년 2월 : 한국과학기술원 전기및전자공학과 (공학박사)
1995년 5월~2001년 4월 : 하이닉스반도체 (주) 선임연구원
2001년 5월~2002년 2월 : 한국전자통신연구원 선임연구원
2002년 3월~현재 : 중앙대학교 첨단영상대학원 조교수
<관심분야> 영상처리, 부호화, 컨텐츠보호기술 등.