

차영상의 윤곽정보를 이용한 가변 블록 정합 알고리즘에 관한 연구

正會員 權 炳 憲* 正會員 安 楨 吉** 正會員 黃 秉 元*

A Study on Variable Block Matching Algorithm Using Edge Information of Differential Image

Byong Hwon Kwon*, Jeong Kil Ahn**, Byong Won Hwang* *Regular Members*

要 約

본 논문에서는 동영상 압축에 핵심 기술인 이동정보량추정에 있어 새로운 가변블럭정합 알고리즘을 제안하였다.

이동정보량 추정에 있어서 고려되어야 할 실시간 처리와 정확한 운동벡터를 구하기 위해 연속된 두 프레임 영상을 이용하여 차영상을 구하고 이 차영상에서 나타난 윤곽정보를 2진화하여 윤곽의 포함정도에 따라 블록의 활동도를 정의하고 이 활동도에 따라 움직임이 많은 부분은 쿼드트리법을 적용하여 보다 정확한 운동벡터를 구하였다.

컴퓨터 모의실험으로 다른 고속블럭매칭 알고리즘과 PSNR과 계산량을 비교하여 제안된 알고리즘의 우수함을 입증하였다.

Abstract

This paper presents a new block matching algorithm in estimation of motion information that is important technology in compressing interframe.

In estimation of motion information, we find differential image using two successive frame image in order to find required real time processing and accurate moving vector, the binary image is made of edge information in this differential image and then define activity of block by inclusion level of edge and find more accurate moving vectors by applicating quadtree method to parts including many moving according to this activity.

The computer simulation results show that increased PSNR and reduced computation time.

* 한국항공대학교 항공전자공학과
Dept. of Avionics, Hankuk Aviation University

** (주) 금경계전 기술 연구소
Technical research Lab, Kum Kyung Co., LTD

論文番號 : 93186

接受日字 : 1993年 10月 5日

I. 서 론

화상전화 시스템, 화상회의, 디지털 HDTV 등에 응용을 목적으로 동영상 부호화 기법에 대한 연구가 활발해지면서, 동영상 부호화 방식의 핵심기술인 이동 정보량추정이 현재 많이 연구되고 있다. 보다 정확한 이동 정보 추정을 함으로 동영상에서 시간축상에 존재하는 상관성을 효과적으로 제거할 수 있다^(1,2).

지금까지 보다 정확한 이동정보 추정을 위해 화소 순환 방식⁽³⁾, 블럭정합방식, 위상 상관 방식 그리고 시공간 경사 방식등 많은 알고리즘이 제안, 연구 되어 왔다. 이들 방식들 중에서 이동정보에 대한 정확도, 효율성 및 실시간 처리등을 고려할때 블럭정합알고리즘방식이 가장 널리 이용된다^(4,10).

이러한 BMA기술은 많이 사용함에도 불구하고 단점을 갖고 있다. 그 단점으로는 블럭 대 블럭으로 운동벡터를 검출하므로 그 블럭내에서 움직임이 Uniform하다는 가정이 필요하다. 이러한 가정은 상대적으로 작은 블럭(8*8 또는 16*16)에서만 만족한다⁽²⁾. 그러나 블럭크기가 작아진다는 것은 블럭의 수가 늘어나고 전송해야 할 운동벡터가 늘어나는 결과가 되므로 전송부담이 생긴다^(5,16).

이런 문제점을 해결하기 위해 움직임이 일어나는 경계부분에 대해서 보다 섬세한 움직임 추정을 하는 블럭크기를 가변시키는 BMA연구가 있어왔으나, 블럭크기를 크게 하여 움직임 추정을 한후 예측오차가 크면 블럭을 나누는 방식에서는 중복된 계산으로 인해 효율적이지 못하고, 또한 블럭크기가 너무 다양한 경우에는 이동벡터량의 증가로 인해 부호화 시키기가 복잡하게 된다.

본 논문에서는 두 영상을 이용하여 얻어진 차영상에서의 윤곽정보를 이용하는데 이러한 윤곽은 현재 프레임이 이전 프레임으로부터 이동한 변위만큼의 윤곽이 드러나므로 이 윤곽을 움직임 정보로 이용하여 블럭의 활동도를 계산한다. 활동도에 의해 변화가 있는 부분과 없는 부분으로 분류하여 쿼드트리로 블럭크기를 가변화함으로써 BMA의 성능을 향상시키고자 한다.

II. 이동벡터의 고려 사항

이동 벡터는 이동보상형 예측 또는 보상형 내삽등에 응용(이용)될때 다음과 같은 측면을 고려하여 추

정하여야 한다^(6,11).

- 운동체의 실제 이동 정보의 추출
- SNR등의 예측 성능의 향상
- 전송될 이동 벡터의 데이터량 감소
- 추정된 이동정보의 다양한 활용

위와 같은 측면을 고려할 때 BMA를 적용하여 구해진 이동벡터는 운동체의 국지적인 특성만을 이용하여 인접 벡터간들간의 연관성을 고려하지 않았기 때문에 운동체의 실제 이동과 일치하지 않는 경우가 발생한다. 그러나 BMA이동벡터는 이미 고려된 운동체의 국지적인 이동(local motion) 특성을 유지하면서 운동체 전체의 이동(global motion)특성을 고려하여 인접한 이동 벡터간의 연관성을 높여줄 경우 운동체의 실제 이동(real motion)과 일치시킬 수 있다.

III. 가변 블럭크기 정합 알고리즘

1. 기존방식

이전 프레임과 현재 프레임 사이의 움직임을 검출하는 방법에는 PRA(Pel Recursive Algorithm)와 BMA(Block Matching Algorithm)가 있다.^(6, 11) PRA는 화소 단위로 움직임 값을 검출함으로써 BMA보다 정확하게 움직임을 검출해 낼 수 있는 반면에 계산량이 많고 복잡하다. 반면에 BMA는 블럭단위로 움직임 값을 검출해내기 때문에 PRA 보다 H/W 구성이 훨씬 간단하며 계산량도 많지 않아 실제 응용에 유리한 장점이 있다. 이러한 BMA 방법에는 탐색 속도를 향상시키기 위한 가변 블럭 정합 방법으로 DMD(Direction of Minimum Distortion), TSS(Three Step Search) 그리고 OTS(One at a Time Search) 등의 방법이 제안되었다.⁽⁷⁾

기존의 가변 블럭 방식은 움직임이 존재 유무에 따라 블럭을 가변시키는데 영상의 통계적 특성을 이용하여 활동도를 정의하여 사용하였다. 그러나 이방식은 처리 시간이 많이 소요되고 블럭과 블럭간의 상관성도에 따라 잘못된 활동도를 얻을 수 있다. 잘못된 활동도는 부정확한 운동벡터를 구하게 되므로 복원된 영상에서 화질열화를 가져온다. 본 연구에서는 기존의 방식중에서 가장 우수한 방식으로 입증된 TSS 방식과 그 성능을 비교, 분석하였다.

2. 차영상의 윤곽정보를 이용한 새로운 가변 블럭정합

2.1 블럭의 활동도

본 논문에서는 제안한 방식은 고정된 블록간의 평균이나 분산값을 이용하여 블록을 가변하는 것이 아니라, 연속하는 두 영상을 갖고 차영상을 구하여 나타나는 윤곽정보를 이용한다.

블록의 활동도는 실제 움직임이 있는 부분을 나타내주며 되므로 차영상에서 나타나는 윤곽부분을 이용하였다.

제안된 활동도는 차영상에서 나타난 윤곽부분을 2진화 하여 $N \times N$ 크기의 블록으로 자른 다음 윤곽부분이 포함된 정도로 정의한다. 이때 윤곽부분의 포함여부에 대한 검사는 입력이 연속하는 두 영상에서 얻은 2진화 영상이므로 블록안이 '1'의 갯수를 검사함으로 수행한다.

다음은 블록의 활동도 정의식을 나타내었다.

$$A(n) = \sum_{i=0}^M \sum_{j=0}^N B(i, j) : \text{Activity of } n\text{'th block} \quad (1)$$

n : Block number

M, N : Block size

$B(i, j)$: Binary image

위식에서 n 번째 블록의 활동도가 설정한 임계치보다 작을 경우는 다음블록에 대해 윤곽부분 포함 여부에 대한 검사를 수행하고 '1'의 갯수가 임계치 이상일 경우는 그 블록 안에서 윤곽 부분이 포함되어 있다고 가정하여 쿼드트리(Quadtree)를 적용하여 4개의 부분블록(Sub-block)으로 나눈다. 그러나 실제 실험을 통해서 보면 블록의 활동도는 하나의 임계치에 대해 움직임의 정도를 다 표현하기 어렵다. 다시말해 16*16블록의 활동도를 살펴보면 배경과 같은 부분의 블록에서는 활동도가 0인 경우도 있다. 즉 윤곽 부분이 없는 곳이 있다. 또 하나는 윤곽의 포함정도가 0이상이고 임계치 이하인 곳이 있다. 이렇게 블록의 활동도가 3가지 경우가 있는데 이 각 경우에 대해 움직임 추정을 달리해줌으로 화면내의 전체움직임과 국지적 움직임 모두를 고려해줄 수 있다.

이상의 알고리즘을 다시 정리하면 다음과 같다.

- (1) 현재 프레임과 이전 프레임으로부터 차영상을 구한다.
- (2) 차영상에서 나타난 윤곽정보에 국부8방향 윤곽 검출방법을 적용하여 2진화 영상을 얻는다.
- (3) 2진화 영상 $M \times N$ 크기의 블록으로 나눈다.
- (4) 블록의 '1'의 갯수를 검사한다.
- (5) 블록의 '1'의 갯수가 정해진 임계치보다 작고 0 이상이면 준 움직임 영역에 대한 움직임 검출

방법을 이용한다.

- (6) 블록의 '1'의 갯수가 정해진 임계치이상이면 움직임 영역 대한 운동벡터 검출방법을 이용한다.
- (7) 블록의 '1'갯수가 없을때는 움직임이 없는 영역으로 간주하여 다음 블록으로 넘어가서 과정(4)를 수행한다.

다음은 위에서 서술한 제안된 알고리즘은 하나씩 설명하기로 한다. 우선 연속하는 두영상에서 차영상을 구하는 방법을 설명한다.

2.2 차영상의 윤곽정보에 대한 2진화

연속하는 영상의 차영상은 다음식으로 표현하면

$$DP(i, j) = ABS[IP(i, j) - IC(i, j)]$$

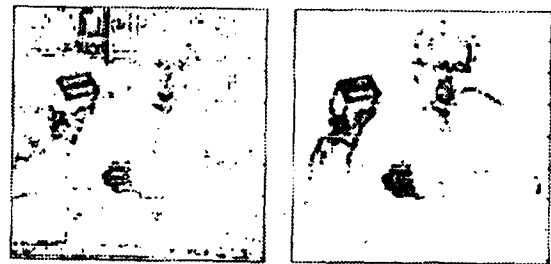
(2)

$IP(i, j)$: 이전 프레임의 화소값

$IC(i, j)$: 현재 프레임의 화소값

이렇게 얻어진 $DP(i, j)$ 화소값은 이전 프레임과 현재 프레임의 차영상을 바로 2진화 영상으로 나타내기 위해서는 임계치 결정이 어렵다. 그래서 차영상에서 나타난 부분에 국부8방향윤곽 검출방법을 사용하여 2진화 영상을 만들었다.

다음 그림 1는 차영상에서 얻은 윤곽정보를 2진화 영상을 만든 영상이다. 그림 1의 (a)는 Salesman101과 102프레임의 차영상 윤곽이고 그림(b)는 101과 103의 차영상 윤곽이다. 그림에서 보고 있듯이 실제로 움직임 부분을 나타내줄을 알 수 있다.



(a) 영상(101-102)

(b) 영상(101-103)

그림 1. 차영상에서 얻은 2진화 영상

Fig 1. Binary Image by Getting Differential Image

2.3 움직임 영역 분할

위에서 구한 LUT(Look up table)을 이용하여 활동도를 구하게 되는데 움직임의 유무판별을 하기 위해서는 임계치를 정하여야 하므로 임계치는 블록의 활동도를 모든 블록에 대해 평균으로 결정한다. 이렇게 임계치를 결정한 이유는 움직임이 많은 영역을 가장 잘 표현해줌으로서 실험을 통해서 확인하였기 때문이다.

$$\hat{A}(n) = \frac{1}{N} \sum_{i=0}^N A(n) \quad (3)$$

위에서 구한 임계치는 가변블록을 적용하는데 사용되고 실제 모든 블록내에서 블록의 활동도가 0인 부분과 0에서 임계치 사이인 부분이 존재하게 된다. 활동도가 임계치 이상인 블록은 움직임 영역으로 나타내고 활동도가 0과 임계치 사이에 값을 갖는 블록에 대해서는 준 움직임 영역으로 된다. 마지막으로 활동도가 0인 영역은 움직임이 없는 부분으로 나타내어 움직임 정도에 따라 적용적으로 움직임 검출을 수행한다. 다음은 움직임 영역분할을 나타낸 것이다.

- $A(n) > Th$: 움직임 영역
- $A(n) = 0$: 움직임이 없는 영역
- $0 < A(n) < Th$: 준 움직임 영역

만약 임계치를 몇개를 더 두어 움직임 영역분할을 한다면 움직임 정도에 따라 좀더 자세한 움직임 검출이 가능하지만 그러한 임계치 결정이 어렵고 너무 다양한 움직임 영역분할에 대한 운동 벡터 검출방법을 결정하기 어렵다.

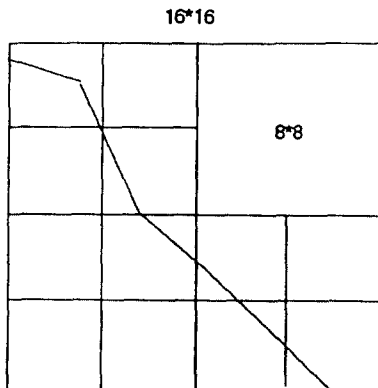


그림 2. 차영상의 윤곽에 대한 쿼드트리 분할

Fig 2. Quadtree Segmentation for Differential Image Edge

(노드 1: 16*16, 노드 2: 8*8)
(Node 1: 16*16, Node 2: 8*8)

그림 3은 실험영상에 대한 쿼드트리를 적용한 그림이다.

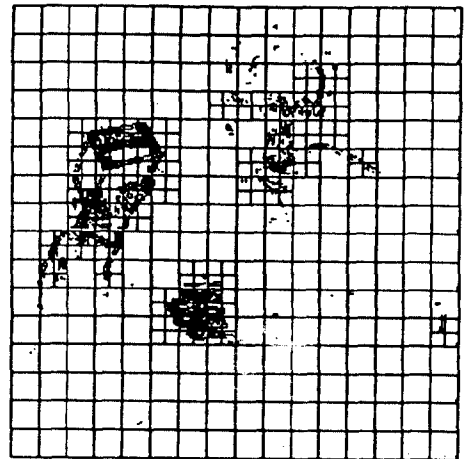
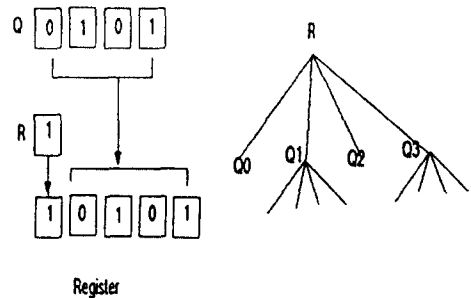


그림 3. 실험영상에 대한 쿼드트리
Fig 3. Quadtree for Experiment Image

그림 3에서 볼 수 있는것과 같이 실제 윤곽이 있는 부분에 대한 쿼드트리 적용이 잘 되었음을 볼 수 있다. 만약 부분력의 수를 더 늘이면(4*4블럭, 2*2블럭) 블록의 모양이 너무 다양화되고 각 블록에 대한 움직임 검출과 코우딩이 복잡해 진다⁽⁸⁾.



(a) 그림(b)에 대한 QR지스터 (b) 쿼드트리 구조

그림 4. 쿼드트리에 대한 비트 할당

Fig 4. Bit Assignment for Quadtree Image

그림 4의 (a)에서 볼수 있듯이, 쿼드트리의 각 노드는 그 블록이 4개의 부분블록으로 나누어지면 그 노드는 1로 세트된다. 1로 세트된 노드는 4개의 가지를 갖고 각 가지는 다시 하나의 노드가 된다.

초기노드레벨은 1비트가 요구되고 그 노드치는 R 레지스터에 표시한다. 두번째 노드레벨은 4비트가 요구되고 그 노드치는 Q레지스터에 표시한다.

2.5 운동벡터 추정

움직임 벡터는 현재 프레임 블록안의 하나의 화소 $y_{i,j}$ 와 이전 프레임의 화소 $x_{i-\alpha, j-\beta}$ 간의 상관성을 비교하여 가장 큰 상관성을 갖는 (α, β) 를 구하고, 블록안의 모든 화소들에 대하여 같은 과정을 행하여 가장 많은 비율을 차지하는 (α, β) 를 그 블록에 대한 운동 벡터로 놓는다.

제안한 가변블록을 이용한 운동벡터 추정은 다음과 같다. 처음에 큰 블록크기를 갖고 움직임이 있는 모든블록들을 보정을 한다. 이것은 다시 블록의 활동도에 의해 보정된 블록과 보정되지 않은 블록으로 분류한다. 보정되지 않은 큰 블록은 쿼드트리를 이용하여 부분블록으로 나누고 부분블록에 대해서는 가변블록 정합 알고리즘을 적용한다. 이 알고리즘을 적용한 후 부분블록들을 다음과 같이 부분블록으로 분류한다.

1. 움직임 영역 블록
2. 준 움직임 영역 블록
3. 움직임이 없는 영역

위와 같은 과정을 그림 5에 보여준다.

본 논문에서는 초기블록을 16*16블록으로 선택을 하였고, 부분블록은 8*8로 하였다.

움직임 영역 분할에 따라 운동벡터 검출방법을 달리 하였다.

첫번째 움직임 영역에서는 R레지스터가 1이므로 1세대블록에서 삼단계 탐색법으로 운동벡터를 구하고 움직임 보정을 한다. 2세대블록에서 즉 4개의 부분블록에서 블록의 활동도를 계산하여 4개중 활동도가 가장 큰 블록에서는 Full search를 행하고 나머지 블록에 대해서는 삼단계 탐색법으로 운동벡터를 구한다.

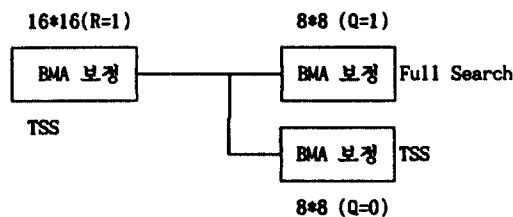


그림 6. 움직임 영역에서의 운동보상
Fig 6. Motion Compensation for Moving Region

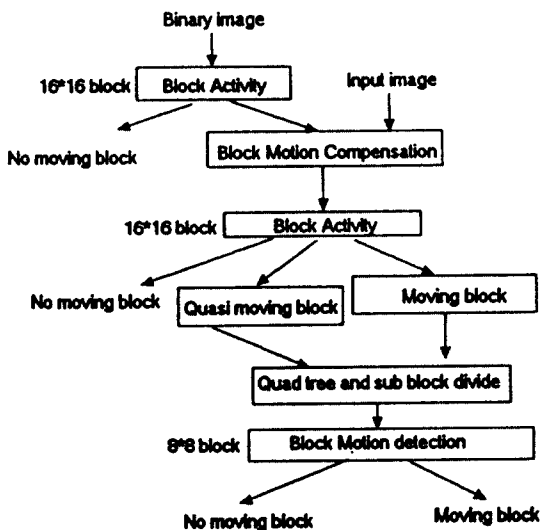


그림 5. 블록의 종류
Fig 5. Classification of Block

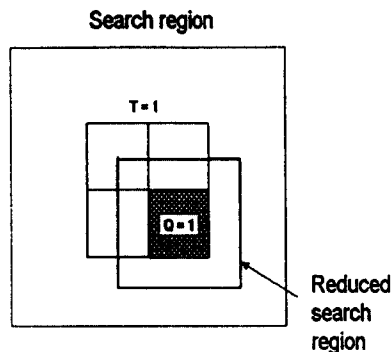


그림 7. 준 움직임영에서의 운동보상
Fig 7. Motion Compensation for Quasi Moving Region

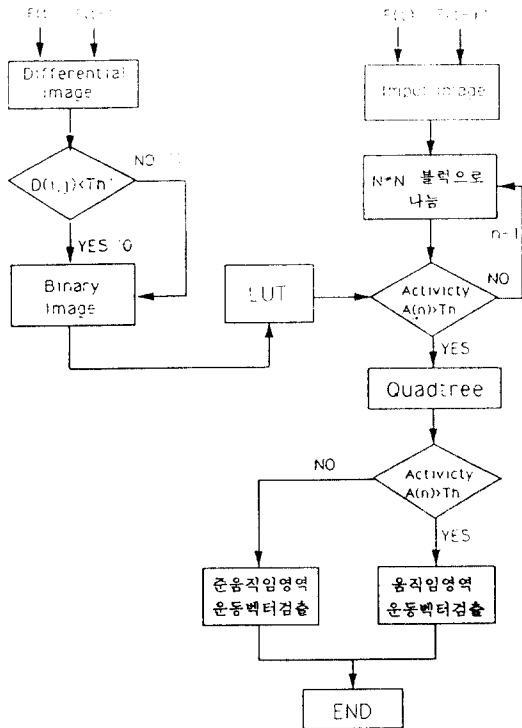


그림 8. 제안된 알고리즘 순서도
Fig. 8. Flowchart of proposed algorithm

이렇게 함으로 국지적 이동 특성을 고려하면서 움직임물체의 전체적 움직임 특성을 나타내줌으로 정확한 운동벡터 검출이 가능하다.

두번째 준 움직임 영역에서는 R레지스티가 1이므로 1세대 블록을 4개의 부분블록으로 나누어 이들 4개의 부분블록에서 활동도를 구하여 가장 큰 활동도를 갖는 부분블록에서 움직임을 추정함으로 1세대 블록에서의 탐색영역보다 줄어든 탐색영역에서 움직임을 검출할 수 있다.

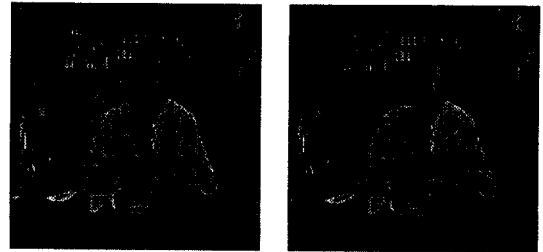
IV. 모의 실험 결과 및 분석

본 실험은 486PC에서 256개소를 갖고 256*256크기의 'Salesman'영상을 30프레임 이용하였고 색차성분은 휘도 성분과 상관성이 강하다는 점을 이용하여 휘도신호만 이용하였다.

평가방법으로 주어진 연속된 두 프레임간의 운동 벡터를 구하는데 걸리는 소요시간과 계산량 그리고

복원된 영상의 화질을 비교하였다. 화질의 정도를 알아보기 위한 객관적 평가기준으로 PSNR(Peak signal to noise ratio)를 이용하였다¹⁰⁾.

다음은 본 실험에 사용한 원영상이다.



(a) 원영상 101 (b) 원영상 105

그림 9. 원영상
Fig. 9. Original image

1. 계산량 비교

기존의 full search방법과 제안된 방법의 계산량 비교에 대해 알아 보았다. 이를 위해 두 방법의 탐색 영역의 범위는 ±8로 하고 사용된 두 블록의 크기는 16*16으로 하였다. 왜곡함수는 MAE로 할 경우, 다음과 같은 계산량이 필요하다.

Full search방법의 1개의 탐색점에 대한 MAE를 계산하기 위해 필요한 덧셈, 뺄셈 그리고 절대값 계산량에 대해 각각 구해보면 255번의 덧셈 계산과 256번의 뺄셈과 절대값 계산이 필요하다. 따라서 ±7의 탐색영역에는 255개의 탐색점이 존재하므로 65,025번의 덧셈과 65,280번의 뺄셈과 절대값 계산이 필요하다. 제안한 방법의 계산량은 블록의 크기를 가변하기 때문에 움직임영역에 따라 가변되는 블록의 수가 달라지므로 평균치로 구한다. 우선 초기블록을 16*16으로 할때 움직임이 없는 영역에서는 TSS방법으로 움직임을 추정하므로 탐색점은 다음과 같다. $(1 + 8 * \log_2 W = 25)$ 곱셈, 덧셈의 수(탐색점*255 = 6375), 뺄셈과 절대값 계산은(탐색점*6400)이 필요하다.

움직임 영역에서는 초기블록에서는 움직임이 없는 영역과 같이 TSS방법으로 움직임을 추정하므로 같은 계산수가 나오고 다시 4개의 부분블록으로 나누어져서 그중 활동도가 가장 큰영역에서 full search방법을 사용하고 나머지 영역에서는 TSS방법으로 움직

임을 추정하므로 3개의 부블럭수에 탐색점수를 곱하고 덧셈, 뺄셈 절대값수를 곱하면 움직임이 있는 영역 전체의 계산수를 산출할 수 있다.

마지막으로 준 움직임 영역에서는 초기 블록을 4개의 부블럭으로 나누어 그중 가장 큰 활동도를 갖는 블록에 대해 탐색영역을 하여 TSS방법으로 움직임을 추정한다. 탐색영역이 반으로 줄기 때문에 탐색점수는 $(1+8 \cdot \log_2 W, W=4)$ 이다.

여기서 나타난 계산량은 30 프레임에 대해 계산한 평균 계산량 비교이다.

표 1은 움직임 영역별로 평균 블록수를 나타내었다. 표 2는 평균 블록수에 필요한 덧셈, 뺄셈, 절대치 계산량을 비교한 것이다.

표 2에서 알 수 있는 바와 같이 제안된 방법은 full search에 비해 계산량이 훨씬 줄어들었다. 하지만 TSS 방법에 비해서는 계산량이 2배정도가 더 많다. 이것은 계산량에 비해 PSNR측면에서 월등한 성능 향상이 있으므로 계산량 측면에서는 Trade off 관계가 있다.

표 1. 움직임 영역에 대한 평균 블록수(*평균치)

Table 1. Average block number for moving region segmentation

영역	블럭수	블럭수
움직임	영역	56.7*
준 움직임	영역	70.2*
움직임이 없는	영역	127.5*

표 2. 계산량 비교

Table 2. Comparison of computational requirement

방법	연산	Addition	Subtraction	Absolute conversion
	Full search		14,630,625	14,688,000
TSS		1,434,375	1,440,000	1,440,000
Proposed method		2,894,658	2,906,880	2,906,880

2. 운동벡터 검출 소요시간의 비교

기존의 full search 방법에서 정합오차 측정의 척도인 평균자승오차, 절대오차, 상호상관성 함수와 제안방식이 평균자승오차 함수, 각 방법을 사용하여 구한 운동벡터 검출 소요시간이 나타나 있다. 제안방식에서는 임계치에 따른 소요시간도 비교하였다.

그러나 제안방식은 우선 주어진 연속적인 프레임의 차영상을 구하고 이차영상의 윤곽을 검출하여 2진화 영상을 만드는 것이 필요하다. 따라서 실질적인 운동벡터 검출소요시간을 고려되어야 하며 실험에 의하면 하나의 화상윤곽을 검출하여 2진화 화상을 만드는 데 드는 소요 시간이 약 1초~2초가 소요된다. 표 4는 이 소요시간을 고려한 기존방식들에 대한 제안방식들이 소요시간의 비율 나타낸다.

표 3에서 볼 수 있듯이 각 방법의 소요시간을 비교해 보면 제안방식을 사용한 경우는 자승평균오차나 절대오차, 상호상관함수를 사용한 경우에 비해 각각 약 15.8배와 9.2배의 속도향상이 있음을 알 수 있다.

표 3. 운동벡터 검출 소요 시간 비교

Table 3. Comparison of times for motion vector detection (단위 초)

방법	소요시간	운동벡터 검출	소요 시간
		Sales 영상	
full search	MSE		79
	MAE		48
	NCCF		94
제안방법	MSE		5

표 4. 운동벡터 검출 소요 시간 비

Table 4. The ratio of the time for detecting motion vector

	MSE	MAE	NCCF
제안방식	1/11.2	1/6.8	1/13.4

3. 복원된 영상의 화질

움직임 검출성능만을 분석하기 위한 실험결과에 대해 알아보겠다. 비교분석을 위한 기존의 방법으로는 고정블럭 Full search 방법과 고속블럭정합 방법인 TSS 방법을 이용하였다. 이때 탐색영역의 범위는 ± 7 로 하였으며 부블럭의 크기는 CCITT의 H.261⁽¹²⁾과 MPEG^(13, 14)에서 규정하고 있는 8×8 과 16×16 인 경우에 대해 각각 실험하였다. 또한 블럭정합알고리즘에서 두 블럭의 유사도를 측정하는 왜곡함수로는 MSE를 사용하였으며 실험영상에 대한 결과의 성능분석 측정치로 식 (4)의 PSNR(Peak Signal to Noise Ratio)를 사용하였다.

$$PSNR = 10 \log_{10} \frac{(255)^2}{MSE} \quad [dB] \quad (4)$$

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |F(i, j) - F'(i, j)|^2 \quad (5)$$

표 5는 제안된 방법과 전 영역탐색법 그리고 3단계 탐색법을 갖고 비교한 평균 PSNR이다.

표 5. 움직임 검출성능에 대한 평균 PSNR

Table 5. Average PSNR for the results of motion detection (단위 : dB)

방법	실험영상	Saesman image
Full Search		28.28
TSS (Three step search)		25.33
제안된 방법		27.92

표 5에서 볼수 있듯이 제안된 방법은 Full search 방법과 비교해 평균 PSNR차이는 0.36dB이고 TSS 방법에 비해 2.59dB이다. 즉 Full search 방법에 필적할 만한 성능을 나타냄을 알 수 있다.

그림 10은 실험영상에 대한 PSNR 그래프이다.

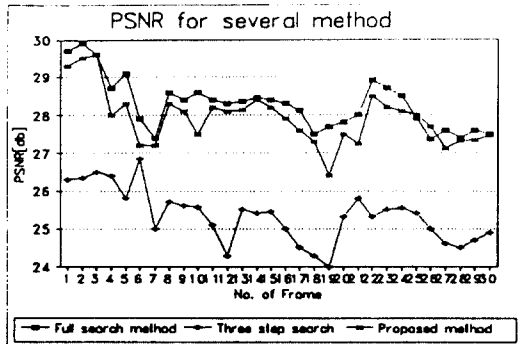


그림 10. 원영상에 대한 PSNR
Fig 10. PSNR for Original Image

두번째 실험으로 잡음영상에 대한 성능분석을 위해 평균이 0이고 표준편차가 각각 5와 10인 Gaussina 잡음을 원영상에 첨가한 잡음영상으로부터 움직임을 검출한 후 구해진 이동벡터를 사용해 원영상으로부터 결과 영상을 복원하는 실험을 하였다. 먼저 표준 편차가 5인 잡음을 영상에 부가한 경우 각 방법에 대한 평균 PSNR이 표 6에 나타나 있으며, 잡음 영상에 대해서도 기존의 방법과 유사한 성능을 가짐을 알 수 있다. 또한 표 7는 표준편차가 10인 잡음을 원영상에 부가한 경우로 이때에도 기존의 방법과 유사한 성능을 갖게됨을 알 수 있다. 표 6과 7에서 알수 있듯이 제안된 방법은 잡음 영상에 강한 특성을 나타냄을 알 수 있다. 이것은 잡음이 가해진 영상을 차영상에서 윤곽정보를 얻기위해 윤곽검출을 수행함으로 잡음에 대한 영향을 이부분에서 제거할 수 있다.

표 6. 움직임 검출성능에 대한 평균 PSNR(잡음영상 $\sigma=5$)

Table 6. Average PSNR for the results of motion detection(noisy sequences $\sigma=5$) (단위 : dB)

방법	실험영상	Salesman image
Full Search		27.38
TSS (Three step search)		24.47
제안된 방법		26.95

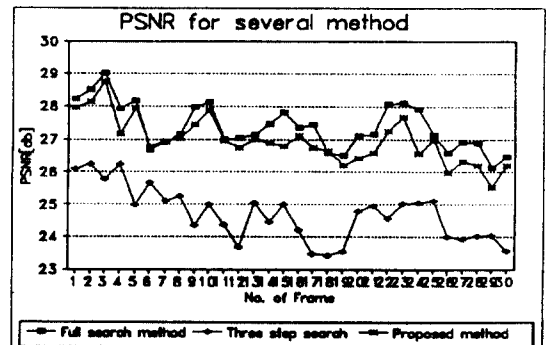


그림 11. PSNR(잡음영상 $\sigma=5$)
Fig 11. PSNR(Noise image $\sigma=5$)

표 7. 움직임 검출성능에 대한 평균 PSNR(잡음영상 $\sigma = 10$)
 Table 7. Average PSNR for the results of motion detection(noisy sequences $\sigma = 10$) (단위 : dB)

방법	실험영상 Salesman image
Full Search	26.63
TSS (Three step search)	24.47
제안된 방법	26.95

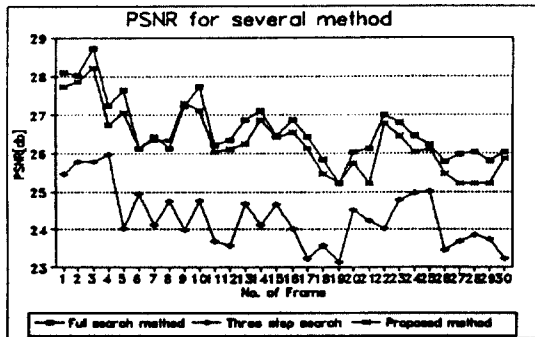


그림 12. PSNR(잡음영상 $\sigma = 10$)
 Fig 12. PSNR(Noise Image $\sigma = 10$)

V. 결 론

움직임 검출기법은 동영상 전송시스템에서 상당히 중요한 역할을 하는 부분으로 움직임을 얼마나 정확히 검출하느냐에 따라 시스템의 성능이 좌우될 수 있는 부분이며, 지금까지 많은 연구가 진행되었으며 고성능의 움직임 검출기법들이 제안되었다. 그러나 여러가지 제안된 기법들중에서 하드웨어 구현이 상대적으로 쉽고 계산량도 적당하며, 움직임 검출성능도 비교적 우수한 방법인 3단계 탐색방법이 많이 사용되기도 하지만, 고화질의 영상을 얻기 위해서는 3단계 탐색방법보다는 탐색영역내의 모든 점을 정합하는 Full search방법이 적합하나 계산량이 방대하여 실시간 처리에 어려움이 많다. 움직임 검출성능을 갖는 알고리즘일수록 많은 계산량이 필요하여 실시간 처리가 어려운 문제로 남아 있다.

본 연구에서는 움직임 검출시 과도한 계산량을 줄

이면서 성능의 열화는 최소가 되게 하는 새로운 가변 블럭정합알고리즘을 제안되었으며, 이 방법은 기존의 여러가지 방법과 쉽게 결합이 가능하면서 여러가지 실험을 통해 제안한 방법의 성능을 평가해 본 결과, 제안한 방법은 기존의 고정블럭 Full search방법에 의해 복원된 영상들과 유사한 화질의 영상을 복원할 수 있다.

참 고 문 헌

- Jain J.R., Jain, A.K., "Displacement Measurement and Its Application in Interframe Coding," IEEE Trans., COM-29, (12), 1981, pp. 1799-1808.
- Chan M.H., "Variable Size Block Matching Motion Compensation with Application to Video Coding," IEE Proceedings, Vol. 137, Pt. I, No.4, Aug. 1990, pp.205-212.
- Netraveli A.N., Robins J.D., "Motion Compensated Television Coding-Part I," Bell Syst. Tech. J., Vol.51, pp.631-670.
- Puri A., Hang H.M., Schiling D.L., "Interframe Coding with Variable Block Size Motion Compensation," in Proceedings, GLOBECOM '87, The Japan, Nov. 1987, pp.2.7.1-2.7.5.
- Chong Uk, LEE, "Intraframe Compression of HDTV Images based on Adaptive Block Size Discrete Cosine Transform," 92국내외 한국과학기술자 학술회의 춘계 Workshop, 전자·정보통신분과, 1992.5. pp.3-22.
- Nusman H.M., Pirsh P., Gravoert H.J., "Advances in Picture Coding," IEEE Trans. Commun. Vol. 29, Dect. 1981, pp.1736-1740.
- Vaisey D.J., Gersho A., "Variable Block-size Image Coding," ICASSP., 1987, pp. 1051-1054.
- 박규태, 이현주, "이동보상 예측오차의 subband 부호화에 관한 연구," 한국통신학회 신호처리 합동학술대회, 1989.9
- Jayant N.S., Digital Coding of Waveforms, Prentice-Hall, Inc. 1984.
- Majid Rabbani, Paul W.Jones, Digital Image Compression Techniques, SPIE., 1991.
- Bierling M., "Displacement Estimation by Hierarchical Blockmatching," Proc. SPIE, Vol.

- 1001, 1988, pp.942-950.
12. Specialists Group on Coding for Visual Telephony, "Proposal on Extending the Scope of rec. H.261 to lower bit rates," CCITT SG XV Doc. #446, 1988.
 13. ISO/IEC JTC1/SC2/WG11, "MPEG Video Simulation Model Three," MPEG90/041, July 1990.
 14. ISO CD 11172-2 rev 1, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbits/s," Nov. 1991.
 15. Perman W.A., "Variable Block Rate and Blockwise Adaption in Cosine Transform Image Coding," ICASSP., 1988, pp.773-776.
 16. Cordell P.J., Clarke R.J., "Low Bit-Rate Image Sequence Coding Using Spatial Decomposition," IEE Proceedings I, Vol.139, No.6, Dec. 1992.
 17. J. Biemond, "A Pel Recursive Wiener-based Dusokacenebt Estimation Algorithm," Signal Processing 18, pp. 399-412, 1987.



權炳憲(Byong Hwon Kwon) 정회원
1964년 8월 28일생
1987년 2월 : 한국항공대학 항공전자공학과 졸업(공학사)
1989년 2월 : 한국항공대학 대학원 항공전자공학과(공학석사)
1991년 3월 ~ 현재 : 한국항공대학 대학원 항공전자공학과(박사과정)

1989년 1월 ~ 현재 : (주)금성사 영상미디어 연구소 제직
※ 주관심분야 : 동영상 신호처리, ASIC 설계, Video signal processing



安 楨 吉(Jeong Kil Ahn) 정회원
1968년 11월 15일
1991년 2월 : 한국항공대학 항공전자공학과 졸업(공학사)
1993년 8월 : 한국항공대학 대학원 항공전자공학과(공학석사)
1993년 4월 ~ 현재 : (주)금성계진 기술연구소 제직

※ 주관심분야 : 동영상 신호처리, SCADA system.



黃 秉 元(Byong Won Whang) 정회원
1947년 2월 2일생
1972년 2월 : 한국항공대학 항공전자공학과 졸업(공학사)
1981년 3월 : 일본동경대학 대학원 전기공학과(공학석사)
1984년 3월 : 일본동경대학 대학원 전기공학과(공학박사)

1973년 3월 ~ 1984년 4월 : 여수 수산전문대학 전임강사
1984년 5월 ~ 1985년 3월 : 국방과학연구소 선임연구원
1985년 4월 ~ 현재 : 한국항공대학교 항공전자공학과 교수
1993년 9월 ~ 현재 : 일본동경대학 전기공학과 교환교수
※ 주관심분야 : 영상처리 및 인식, 동영상 신호처리