

論文93-30B-1-6

가산투영을 이용한 2단계 고속 블럭정합 알고리즘

(Two-Stage Fast Block Matching Algorithm Using Integral Projections)

金俊植*, 朴來弘*, 李炳旭**

(Joon-Seek Kim, Rae-Hong Park, and Byung-Uk Lee)

要約

본 논문에서는 동영상 전송시스템의 움직임 검출부에 대해 연구하였으며 움직임 검출시 과도한 계산량을 줄이기 위한 2단계 블럭정합 알고리즘을 제안하였다. 이 방법은 첫번째 단계로 가산투영을 사용한 1차원 왜곡 측정치를 사용하여 이동벡터 후보를 결정한 후 두번째 단계로 기존의 블럭정합 알고리즘의 2차원 왜곡 측정치를 사용하여 최종적인 이동벡터를 결정하는 방법이다. 이때 첫번째 단계의 왜곡 측정치의 계산시 1차원 계산을 수행하여 기존 방법 (16×16 블럭을 사용한 full search 방법)에 비해 4배 정도의 계산량 감축을 얻는 반면, 복원된 영상의 화질은 기존의 방법과 유사한 성능을 갖게 된다. 제안한 방법의 성능을 분석하기 위해 원영상과 잡음영상에 대해 실험하였으며, MPEG (Moving Picture Experts Group) SM3 (Simulation Model Three)와 결합한 경우에 대해서도 실험하였다. 실험 결과 제안한 방법이 기존의 방법에 비해 빠른 계산성능을 가지면서 기존의 방법과 유사한 화질의 영상을 얻을 수 있었다.

Abstract

In this paper, a two-stage block matching algorithm (BMA), which can reduce greatly the computational complexity of the conventional BMAs, is proposed, in which the one-dimensional distortion measure based on the integral projection is introduced to determine the candidate motion vectors and then among them a final motion vector is detected based on the conventional two-dimensional distortion measure. Due to the one-dimensional calculation of a distortion measure, the proposed algorithm can reduce the computational complexity of the conventional BMA (full search method with a 16×16 block) by a factor of 4, with its performance comparable to those of the conventional ones. Simulation results based on the original and noisy image sequences are shown. Also the simulation of the proposed method combined with the MPEG (Moving Picture Experts Group) SM3 (Simulation Model Three) is presented. Computer simulation shows that the proposed algorithm is fast with its performance comparable to those of the conventional ones.

*正會員, 西江大學校 電子工學科
(Dept. of Elec. Eng., Sogang Univ.)

**正會員, 大宇電子(株) 映像研究所
(Daewoo Electronics Co., Ltd., Video Research Center)

接受日字: 1992年 9月 26日

(※ 본 연구는 한국과학재단 목적기초 연구비 지원에 의해 이루어진 것임)

1. 서론

프레임간 부호화 (interframe coding) 방법은 시간상에서 연속적으로 변하는 영상신호를 압축하여 전송하는데 매우 효율적인 방법이다. 특히, 움직임 보상부호화 (MCC: Motion-Compensated Coding) 방법은 연속한 영상내에서 움직임을 검출/보상하여

부호화하는 방법으로 높은 압축률과 고화질의 영상을 얻을 수 있다.¹²

움직임 보상부호화 방법은 크게 두가지 부분으로 구성되어 있는데 하나는 영상내에 존재하는 움직임양을 검출/보상하는 움직임 검출부³와 움직임 예측에 의해 발생하는 예측오차를 부호화⁴하는 두가지 부분이다. 이때 예측오차 전송부의 성능은 그부분 나름대로의 방법에 의해 성능이 결정되지만 움직임 검출부의 성능에 의해서도 많은 영향을 받는다. 즉, 영상내에 존재하는 움직임을 정확히 검출하면 예측오차가 줄어들 수 있기 때문에 정확한 움직임 검출은 상당히 중요하다.

움직임 검출/보상방법은 크게 두가지로 나눌 수 있는데 첫째는 화소단위의 움직임을 검출하여 보상하는 화소순환 알고리즘 (PRA: Pel-Recursive Algorithm)¹⁵이며, 둘째는 블럭단위의 움직임을 검출을 통해 움직임 보상을 하는 블럭정합 알고리즘 (BMA: Block Matching Algorithm)^{16,11}이다. 전자는 비교적 정확한 움직임을 검출할 수 있지만 과도한 계산량으로 인해 하드웨어 구현에 많은 어려움이 있으며, 후자는 전자에 비해 화질은 조금 떨어지지만 계산량이 전자에 비해 월등히 적어 하드웨어 구현이 용이하여 영상전화기나 HDTV (High Definition Television) 등과 같은 많은 응용분야에서 사용되고 있다.

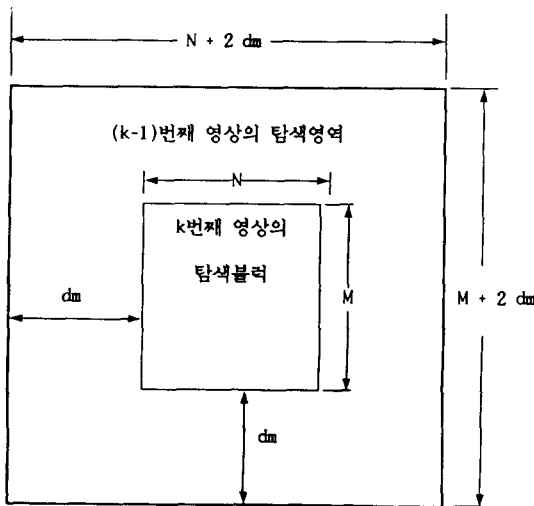


그림 1. 블럭정합의 탐색영역

Fig. 1. Search area of block matching algorithms.

블럭정합 알고리즘은 영상을 일정크기의 블럭으로 나눈 후 각 블럭이 이전영상의 어떤 블럭과 가장 유

사한지를 탐색하여 각 블럭의 이동량을 결정하는 방법⁶으로 그림 1의 탐색영역내의 모든 블럭들에 대해 유사도를 계산하는 full search 방법과 실시간 처리를 좀 더 쉽게 하기 위해 탐색블럭의 수를 줄임으로써 계산량을 상당히 감축한 방법들이 제안되었다.^{7,11} 그러나 반도체 기술이 발전함에 따라 실시간 하드웨어 구현이 쉬워지고 좀 더 많은 계산을 빠르게 처리할 수 있게 됨에 따라 성능면에서 우수하면서 규칙적인 하드웨어 구조를 갖는 full search 방법이 선호되고 있으나 아직도 많은 계산량으로 인해 실시간 처리가 어렵다.

본 논문에서는 full search 방법과 결합이 가능하면서 이 방법보다 적은 계산량으로 이와 유사한 성능을 갖는 2단계 알고리즘에 관한 연구로, 이를 위해 가산투영 (integral projection)의 개념을 사용하여 2차원 정합을 1차원 정합으로 변환하여 계산상의 이득을 얻으면서 full search 방법에 근접한 복원성능을 갖는 알고리즘에 대해 연구하였다. 본 논문의 구성은 다음과 같다. II장에서 기존의 BMA에 대해 간략히 언급하였으며, III장에서는 가산투영의 수학적 배경에 대해 살펴보고, IV장에서 가산투영 기법과 결합된 블럭정합 알고리즘에 대해 설명하고 V장에서는 제안한 방법과 기존의 방법들과의 계산량 비교를 하였다. VI장에서는 컴퓨터 모의실험을 통하여 제안한 움직임 검출방법과 기존방법의 성능을 비교/분석하였으며, 시스템 측면에서의 성능을 분석하기 위해 MPEG (Moving Picture Experts Group) SM3 (Simulation Model Three)와 결합하여 실험하였으며, 마지막으로 VII장에서 결론을 맺었다.

II. 기존의 블럭정합 알고리즘

블럭정합 알고리즘은 시간적으로 서로 인접한 두장의 영상에서 각각의 영상을 일정한 크기의 블럭으로 나눈 후 해당 블럭의 움직임을 추정하는 알고리즘이다. 일반적으로 움직이는 물체는 회전운동 또는 크기의 확대나 축소가 있을 수 있으나 블럭정합 알고리즘은 초당 30장을 전송한다고 가정할 때 이전 영상에서의 물체는 현재 영상에서 물체의 변형이 많이 생기지 않고 단지 물체의 이동으로 근사화할 수 있다는 전제하에 사용되는 움직임 추정방식이다.

그림 1에서와 같이 블럭정합 알고리즘에서의 움직임 추정은 (k-1)번째 영상에서 $(M + 2dm) \times (N + 2dm)$ 의 탐색영역을 설정하고 k번째 영상에서의 $M \times N$ 크기의 블럭과 같은 크기의 (k-1)번째 영상에서의 블럭을 탐색영역을 벗어나지 않도록 하여 서로간의 왜곡 측정치를 계산하여 가장 적합한 블럭을 찾아

이때의 변위를 계산하여 해당 블록의 움직임 정도를 구한다. 이때 왜곡 측정치 평가함수로는 식 (1)에 정의되는 MAD (Mean Absolute Difference)를 사용한다.

$$MAD(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |S_k(m, n) - S_{k-1}(m+i, n+j)| \quad (1)$$

여기서 $S_k(\cdot)$ 와 $S_{k-1}(\cdot)$ 는 각각 k번째 영상과 (k-1)번째 영상을 뜻한다.

블럭정합 알고리즘에는 탐색영역내의 모든 점에 대해 움직임을 추정하는 full search방식이 가장 성능이 우수하나 계산량이 방대하여 실시간 처리에 많은 어려움이 있다. 이러한 문제점을 해결하기 위해 full search 방법에 비해 성능은 떨어지지만 탐색점 수를 줄여 계산시간에서 이득을 얻을 수 있는 방법들이 있다. 이 방법들 중에는 3단계로 탐색간격을 줄여 가면서 정합점을 찾는 3-step search 방식 (TSS)^[8] 과 탐색영역의 중심점에서 간격이 2인 마름모 형태의 꼭지점에 해당하는 점들을 포함하는 5개의 점에 대해 최적의 정합점을 찾고 다시 찾아진 점에 대해 같은 과정을 탐색영역의 경계에 도달하거나 최소점이 중심에 나타날 때까지 반복한 후 간격이 1인 주변의 9개 점에 대해 최종의 정합점을 찾는 direction of minimum distortion 방식^[7] 그리고 일정한 형태의 탐색점의 집합을 설정하고 이들에 대해 정합점을 찾는 menu vector 방식^[9] 이 있다. 또한 어느 한 축을 중심으로 왜곡 측정치가 감소하는 방향으로 탐색을 하다가 측정치가 그 방향에 대해 최소가 되면 탐색방향의 직각 방향으로 탐색과정을 반복하여 최적의 정합점을 찾는 one-at-a-time search 방식^[10] 과 3-step search방식과 유사하나 탐색점이 정사각형의 모서리에만 존재하는 cross search 방식^[12] , 그리고 지역통과 필터를 사용하여 계층적 영상을 만든 후 각 계층에 대해 움직임을 추정하는 계층적 블럭정합 알고리즘^[13] 등의 고속 알고리즘들이 있다. 이러한 고속 알고리즘들은 full search 방식에 비해 계산량 감축은 8배에서 10배정도이며 성능저하는 PSNR (Peak Signal to Noise Ratio)면에서 평균 2dB정도이다.

III. 가산투영의 수학적 배경^[14]

영상에 대한 가산투영은 임의의 방향을 따라 존재하는 화소들을 더한 값으로 정의되며, 수직 또는 수평선상의 화소들을 더한 양을 수직 또는 수평 가산투영이라 한다.

예를 들어, x축 방향의 1차원 가산투영 $g(y)$ 는 식 (2)와 같이 주어진다.

$$g(y) = \int f(x, y) dx \quad (2)$$

여기서 $f(x, y)$ 는 영상내의 (x, y) 위치의 화소값을 나타낸다. $f(x, y)$ 에 대한 2차원 Fourier 변환이 식 (3)과 같이 주어질 때

$$F(u, v) = \iint f(x, y) e^{-i2\pi(ux+vy)} dx dy \quad (3)$$

$u=0$ 인 $F(0, v)$ 은 식 (4)과 같은 형태가 된다.

$$\begin{aligned} F(0, v) &= \iint f(x, y) e^{-i2\pi vy} dx dy \\ &= \left[\int f(x, y) dx \right] e^{-i2\pi vy} dy \\ &= \int g(y) e^{-i2\pi vy} dy \end{aligned} \quad (4)$$

이때 $F(\cdot)$ 는 1차원 Fourier 변환 연산자이다. 식 (4)로부터 $F(0, v)$ 인 선분은 x 방향의 가산투영값을 1차원 Fourier 변환하여 얻을 수 있다. 다시 말해서 각 방향에 대한 가산투영의 Fourier 변환은 변환영역에서 방사선형태로 주어지므로, 여러 각도의 가산투영값을 1차원 Fourier 변환하면 $F(u, v)$ 에 대한 모든 값을 얻을 수 있다. 이렇게 얻어진 $F(u, v)$ 값은 $f(x, y)$ 의 임의의 방향에 대한 분포특성을 나타내므로 적절한 투영방향을 선택함으로써 $F(u, v)$ 를 $f(x, y)$ 에 대한 특징값으로 사용할 수 있다.

가산투영의 개념은 디지털 영상에도 적용이 가능하다. 영상내에 존재하는 부블럭의 수평 또는 수직 가산투영값은 그 블럭내에 존재하는 화소값들에 대한 정보를 포함하고 있으므로 그 부블럭에 대한 특징으로 사용될 수 있다. 이러한 개념을 사용하여 인접한 영상들에서 움직임을 검출하는 새로운 블럭정합 알고리즘을 IV장에서 제안하였다.

IV. 제안한 가산투영을 이용한 2단계 고속 블럭정합 알고리즘

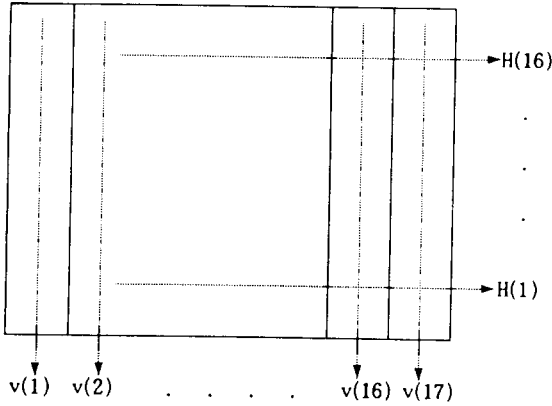
본 연구에서는 full search 방법의 탐색점 수를 줄여 계산량을 감축하는 방법을 택하지 않고 탐색점의 수는 원래의 값을 사용하고 블럭들의 유사성 검사의 계산량을 줄이는 방법을 택하였다. 유사도 판단 계산량을 줄이기 위해 두단계의 정합과정을 사용하였으며 첫번째 정합단계에서는 블럭간의 2차원 정합계산을 가산투영을 이용하여 1차원 정합으로 변환하여 정합 계산량을 감축하였다. 그러나 2차원 정합을 1차원 정합으로 변환할 때 발생하는 정보 손실로 인한 오정합을 보상하기 위해 두번째 정합과정에서 2차원 정합을 사용하여 부정확한 정합으로 인한 화질저하를

최소화하였다.

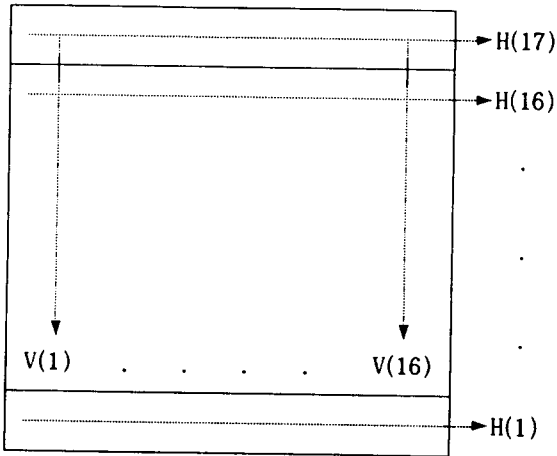
본 연구에서 가산투영은 계산량 감축을 위해 적용되었으며, 가산투영의 또다른 장점인 가산형 잡음에 강한 특성으로 인해 제안한 방법도 가산형 잡음에 강한 특성을 갖는다.^[15]

1. 탐색점들의 가산투영값 계산

본 절에서는 가산투영을 사용하여 영상의 2차원 정합을 1차원 정합으로 변환하는 과정에 대해 설명하겠다.



(a)



(b)

그림 2. 16×16블럭에 대한 가산투영

- (a) 수평방향 탐색
- (b) 수직방향 탐색

Fig. 2. Integral projection in a 16×16 subblock.
 (a) Search in the horizontal direction.
 (b) Search in the vertical direction.

영상에 대한 가산투영값은 식 (5)와 같이 임의의 방향에 존재하는 화소값들을 더하여 그 방향에 대한

가산투영값을 얻을 수 있다.

$$P_j = \sum_{i=1}^M I(i, j) \tag{5}$$

단, P_j 는 가산투영값을 나타내며, $I(i, j)$ 는 (i, j) 위치에서의 화소값을 나타낸다.

그림 2는 한 블럭에 대한 수평 가산투영값과 수직 가산투영값을 나타내며 i 방향과 j 방향으로 인접한 탐색점들의 가산 투영값들을 구하는 방법에 대해 보이고 있다. 그림 2(a)는 블럭의 크기가 16×16일 경우 동일 수평선상의 오른쪽에 존재하는 탐색점의 가산투영값을 구하는 방법으로, 수평 가산투영값은 이전에 구해진 왼쪽 탐색점의 수평 가산투영값들 ($H(1), H(2), \dots, H(16)$) 각각에서, 왼쪽 블럭에는 존재하나 오른쪽 블럭에서는 빠지는 각 화소값을 빼주고 새로 첨가된 화소값을 더해줌으로써 새로운 블럭의 수평 가산투영값을 얻을 수 있다. 또한 수직 가산투영값의 경우 이전 블럭의 수직 가산투영값들 ($V(1), V(2), \dots, V(16)$)중 맨 왼쪽의 수직 가산투영값인 $V(1)$ 은 제거하고 새로운 수직 가산투영값인 $V(17)$ 을 계산하여 새로운 수직 가산투영값을 얻게 된다. 그림 2(b)는 동일 수직선상에 존재하는 인접 탐색점의 가산투영값을 구하는 방법으로 그림 2(a)와 유사한 과정을 거치게 된다.

2. 가산투영값을 사용한 정합과정

IV.1절에서는 가산투영값을 정합의 특징으로 사용하기 위해 2차원 블럭정보로부터 1차원 가산투영값을 구하는 과정에 대해 설명하였다. 이번 절에서는 구해진 가산투영값을 사용하여 최적의 블럭을 정합하는 과정에 대해 설명하는데 그 과정은 다음과 같다.

- 과정 1) 현재 영상의 $M \times N$ 블럭에 대해 수평 가산투영값 ($H_i(m), m = 1, 2, \dots, M$)과 수직 가산투영값 ($V_i(n), n = 1, 2, \dots, N$)을 계산한다.
- 과정 2) 이전 영상의 탐색점 $sp1$ (그림 3)에 대한 수평 가산투영값 ($H_{i-1.0.0}(m), m = 1, 2, \dots, M$)과 수직 가산투영값 ($V_{i-1.0.0}(n), n = 1, 2, \dots, N$)을 계산한다.
- 과정 3) 그림 3은 한 화소 떨어진 주변 탐색점에 대한 그림으로 과정 2에서 구해진 $sp1$ 의 수평 가산투영값과 수직 가산투영값을 사용하여 IV.1절에서 설명한 것과 같이 주변의 8개의 탐색점의 수평 가산투영값 ($H_{i-1.1.1}(m), m = 1, 2, \dots, M$)과 수직 가산투영값 ($V_{i-1.1.1}(n), n = 1, 2, \dots, N$)을 계산한다. 이와같

은 과정을 통해 탐색영역내의 모든 점에 대해 수직 및 수평 가산투영값을 계산한다.

과정 4) 1차원 가산투영값들을 사용하여 현재 영상의 블럭과 탐색블럭의 왜곡함수값 $D_{1d}(i, j)$ 을 계산한다.

$$D_{1d}(i, j) = \sum_{m=1}^M |H_i(m) - H_{i-1,j}(m)| + \sum_{n=1}^N |V_i(n) - V_{i-1,j}(n)| \quad (6)$$

과정 5) 위와 같은 과정을 통하여 주변의 모든 탐색점의 왜곡함수값을 구하여 가장 최소의 값을 갖는 정합점을 찾는다.

과정 6) 과정 5에 의해 얻어진 한개의 탐색점을 중심으로 그림 3과 같은 형태의 주변 8개의 탐색점들과 중심점에 대해 2차원 왜곡함수값 $D_{2d}(i, j)$ 을 계산한다.

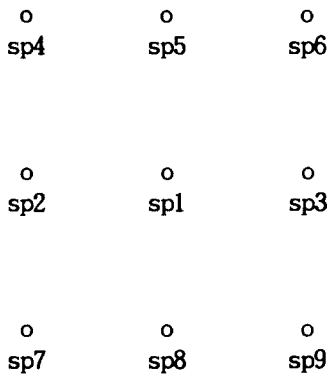


그림 3. 탐색점
Fig. 3. Search points.

$$D_{2d}(i, j) = \sum_{m=1}^M \sum_{n=1}^N |B_i(m, n) - B_{i-1,j}(m, n)| \quad (7)$$

단, $B_i(\dots)$ 는 현재영상내의 부블럭을 나타내며 $B_{i-1,j}(\dots)$ 는 이전영상의 (i, j) 위치에서의 부블럭을 나타낸다.

과정 7) 과정 6의 왜곡함수값중 최소의 값을 선택하여 최종 정합점을 결정한다.

V. 계산량 비교

이번 장에서는 기존의 full search 방법과 제안한 방법의 계산량 비교에 대해 알아 보겠다. 이를 위해 두 방법의 탐색영역의 범위는 ± 7 로 하고 사용된 부블럭의 크기는 16×16 으로 하였다. 왜곡함수를 MAD로 할 경우 다음과 같은 계산량이 필요하다.

Full search 방법의 1개의 탐색점에 대한 MAD를 계산하기 위해 필요한 덧셈, 뺄셈 그리고 절대값 계산량에 대해 각각 구해보면 255번의 덧셈 계산과 256번의 뺄셈과 절대값 계산이 필요하다. 따라서 ± 7 의 탐색 영역에는 225개의 탐색점이 존재하므로 57,375번의 덧셈과 57,600번의 뺄셈과 절대값 계산이 필요하다.

제안한 방법의 덧셈 계산량을 구해보면 현재 영상에서 부블럭의 수평 및 수직 가산투영값을 구하기 위해 $480(2 \times 16 \times 15)$ 번의 계산이 필요하며, 이전 영상에서 탐색영역의 중심점에 있는 탐색점에 대한 덧셈 계산도 480번이 필요하다. 그리고 구해진 가산투영값들을 사용하여 인접한 탐색점들의 가산투영값들을 구하기 위해 하나의 탐색점당 31번의 덧셈이 필요하므로 탐색영역 전체에 대해 $6,944(224 \times 31)$ 번의 덧셈을 수행한다. 또한 구해진 가산투영값들의 1차원 MAD 계산시 1개의 탐색점당 31번의 연산이 필요하므로 $6,975(225 \times 31)$ 번의 계산이 필요하며, 두번째 단계에서의 2차원 정합을 위한 MAD 계산시 2,295(255×9)가 필요하다.

제안한 방법의 뺄셈연산은 탐색영역에서 인접한 탐색점들의 가산투영값을 구할 때 하나의 탐색점당 16번이 필요하므로 탐색영역 전체에 대해 $3,584(224 \times 16)$ 번이 소요된다. 그리고 첫번째 단계에서의 1차원 MAD 계산시 $7,200(2 \times 16 \times 225)$ 번의 뺄셈연산이 추가되며, 두번째 단계에서의 2차원 MAD 계산시 $2,304(256 \times 9)$ 번의 뺄셈연산이 필요하다. 마지막으로 절대값 계산은 1차원 MAD 계산시 $7,200(2 \times 16 \times 225)$ 번이 필요하며, 2차원 정합시 $2,304(256 \times 9)$ 번의 연산이 필요하다.

표 1. 계산량 비교(16×16 블럭)
Table 1. Comparison of computational requirements(16×16 block).

	Addition	Subtraction	Absolute conversion
Full search	57,375	57,600	57,600
Proposed method	17,174	13,088	9,504

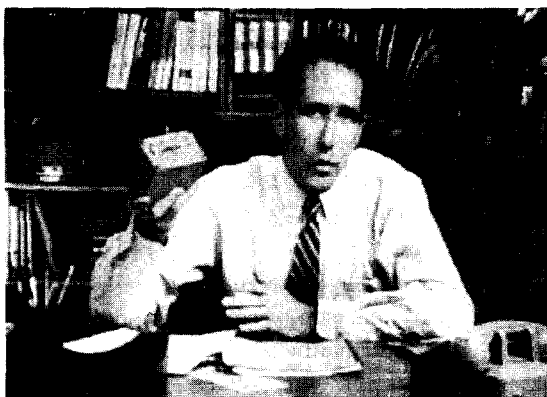
위의 계산결과들을 요약하여 표 1에 나타내었다. 표 1에서 덧셈, 뺄셈 그리고 절대치 연산의 수행속도를 모두 같다고 가정할 경우 제안한 방법은 기존의 방법에 비해 평균적으로 1/4정도의 계산량을 가짐을 알 수 있다.

VI. 실험결과 및 토의

본 절에서는 제안한 움직임 검출방법의 성능분석을 위해 컴퓨터 모의실험 결과를 분석하였다. 실험에 사용한 영상은 SIF (Simulation Input Format)의 "Miss America" 영상 150장과 "Susie" 영상 150장 그리고 "Salesman" 영상 200장으로 luminance 영상의 해상도는 352×240이고 chrominance 영상의 해상도는 176×120이며 8 bits로 양자화된 영상으로 그림 4에 luminance 영상만을 보였다. 이들 영상중 제안한 방법의 움직임 검출성능만을 분석하기 위해 luminance 영상만을 사용하여 실험하였다.



(a)



(b)

그림 4. 원 영상

(a) "Miss America" 영상(첫번째 프레임)

(b) "Salesman" 영상(첫번째 프레임)

Fig. 4. Original images.

(a) "Miss America" sequence(1st fram).

(b) "Salesman" sequence(1st fram).

먼저 제안한 방법의 움직임 검출성능만을 분석하기 위한 실험결과에 대해 알아보겠다. 비교분석을 위한 기존의 방법으로는 full search 방법과 참고문헌 [14]

의 가산투영값만을 사용하여 움직임 검출한 방법 및 본 논문에서 제안한 방법인 가산투영을 사용하여 정합한 후 주변의 탐색점들에 대해 최종적으로 2차원 정합을 하여 움직임을 검출하는 2단계 정합방법에 대해 실험하였다. 이때 탐색영역의 범위는 물체의 움직임이 크지않은 head-and-shoulder 영상으로 간주하여 ±7로 하였으며 부블럭의 크기는 CCITT의 H. 261^[18] 과 MPEG^[6,17] 에서 규정하고 있는 8×8과 16×16인 경우에 대해 각각 실험하였다. 또한 블럭정합 알고리즘에서 두 블럭의 유사도를 측정하는 왜곡함수로는 MAD를 사용하였으며 세종류의 실험영상에 대한 결과의 성능분석 측정치로 식 (8)의 PSNR (Peak Signal to Noise Ratio)를 사용하였다.

$$PSNR = -10 \log \frac{(1/KL) \sum_{i,j}^K \sum_{i,j}^L |O(i,j) - R(i,j)|^2}{255^2} \quad (8)$$

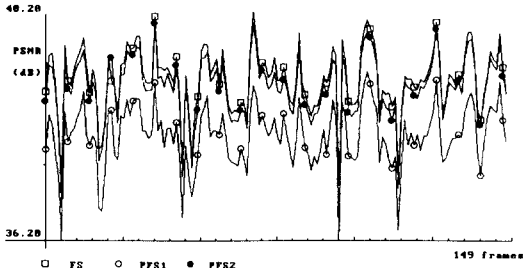
여기서 O()는 원영상의 화소값을 나타내고 R()는 복원된 영상의 화소값이며, K과 L은 영상의 가로와 세로의 크기이다.

표 2. 움직임 검출성능에 대한 평균 PSNR(원영상)
Table 2. Average PSNR for the results of motion detection(original sequences).

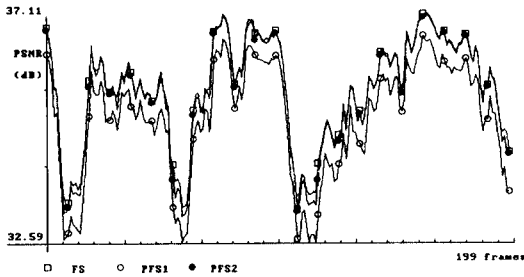
	(단위: dB)					
	Miss America		Salesman		Susie	
	8 × 8	16 × 16	8 × 8	16 × 16	8 × 8	16 × 16
FS	39.86	39.09	36.50	35.60	36.64	35.53
PFS1	38.58	38.07	36.01	35.01	35.80	34.74
PFS2	39.51	38.93	36.39	35.52	36.49	35.47

표 2는 세종류의 실험영상에 대한 평균 PSNR로 FS는 기존의 full search 방법이고 PFS1은 full search 방법에 가산투영을 적용하여 1차원 정합만을 사용하는 방법이며, PFS2은 본 논문에서 제안한 2단계 정합방법의 결과이다. 표에서 보듯이 제안한 방법은 블럭의 크기에 관계없이 평균 PSNR면에서 기존의 방법과 거의 차이를 보이지 않고 있으며, 특히 사용 블럭의 크기가 커짐에 따라 성능개선 효과가 커짐 (표 2에서 "Miss America" 영상의 경우 8×8 블럭에 대해 full search방법과 제안한 방법의 평균 PSNR 차이는 0.35dB인 반면 16×16 블럭의 경우 0.16dB이며, "Salesman" 영상에 대해서 8×8 블럭은 0.11dB, 16×16 블럭은 0.08dB이고 "Susie" 영상에 대해서 8×8 블럭은 0.15dB, 16×16 블럭은 0.06dB이다.)을 알 수 있다. 이것은 가산

투영을 사용함으로써 인해 원영상에 존재하는 잡음성분 (균일하지 못한 조명으로 인한 빛의 변화나 카메라에 의한 잡음성분)의 효과가 줄어든 것으로 판단된다. 그림 5는 16×16 블럭을 사용한 경우의 실험영상 각각에 대한 PSNR 그래프이다. 이 그래프에서 보듯이 제안한 방법은 기존의 방법과 거의 성능차를 보이지 않는다.



(a)



(b)

그림 5. PSNR 그래프(원영상)
(a) "Miss America" 영상
(b) "Salesman" 영상

Fig. 5. PSNR graph(Original sequences).
(a) "Miss America" sequence.
(b) "Salesman" sequence.

두번째 실험으로 잡음영상에 대한 성능분석을 위해 i.i.d. (independent identically distributed)의 평균이 0이고, 표준편차가 각각 5와 10인 Gaussian 잡음을 원영상에 첨가한 잡음영상으로부터 움직임을 검출한 후 구해진 이동벡터를 사용해 원영상으로부터 결과영상을 복원하는 실험을 하였다. 먼저 표준편차 5인 잡음을 영상에 첨가한 경우 각각의 블럭 크기에 대한 평균 PSNR이 표 3에 나타나 있으며, 잡음영상에 대해서도 기존의 방법과 유사한 성능을 가짐을 알 수 있다. 또한 표 4는 표준편차가 10인 잡음을 첨가한 경우로 이때에도 기존의 방법과 유사한 성능을 갖

게됨을 알 수 있다. 표 3과 4에서 "Miss America" 영상의 경우 잡음이 없는 경우와 비교해 볼 때 잡음이 점점 강해질수록 기존의 방법과 성능차가 좀 더 줄어든 것을 볼 수 있는데 이것은 "Miss America" 영상들의 평균 화소값이 "Salesman" 영상에 비해 작기 때문에 잡음에 의한 영향이 크게 나타나며, 가산투영이 합형태의 연산으로 가산형 Gaussian 형태의 잡음에 강한 특성을 보이기 때문이다. 또한 "Salesman" 영상에 잡음이 첨가된 경우 움직임 검출시의 사용 블럭의 크기가 커짐에 따라 검출성능이 향상된 것을 볼 수 있는데 이것 또한 가산투영이 잡음에 강한 특성에 의한 결과이다. 그림 6와 7은 16×16 블럭을 사용한 경우의 잡음영상에 대한 실험결과 PSNR 그래프로 표에서 얻은 결과와 똑같은 추세임을 알 수 있다.

표 3. 움직임 검출성능에 대한 평균 PSNR (잡음영상, $\sigma=5$)

Table 3. Average PSNR for the results of motion detection(noisy sequences, $\sigma=5$).

	Miss America		Salesman		Susie	
	8 × 8	16 × 16	8 × 8	16 × 16	8 × 8	16 × 16
FS	37.97	38.29	35.60	35.45	35.52	35.22
PFS1	37.40	37.50	34.72	34.61	34.12	33.93
PFS2	37.82	38.15	35.32	35.35	35.03	35.04

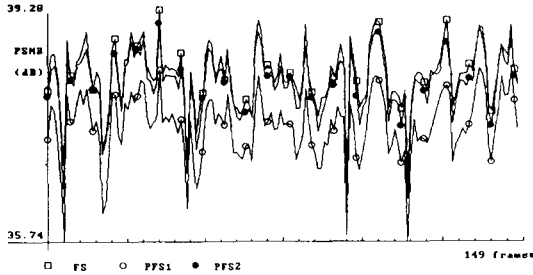
표 4. 움직임 검출성능에 대한 평균 PSNR (잡음영상, $\sigma=10$)

Table 4. Average PSNR for the results of motion detection(noisy sequences, $\sigma=10$).

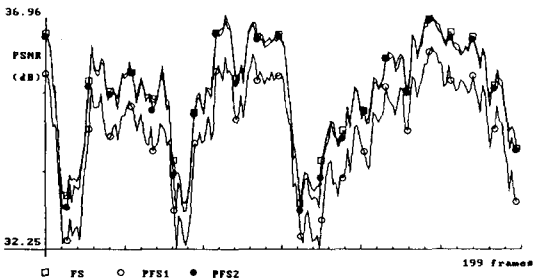
	Miss America		Salesman		Susie	
	8 × 8	16 × 16	8 × 8	16 × 16	8 × 8	16 × 16
FS	36.10	36.93	32.96	34.56	33.21	34.19
PFS1	36.01	36.44	31.97	33.07	32.06	32.34
PFS2	36.11	36.82	32.52	34.23	32.75	33.63

본 논문에서 제안한 방법을 3-step search 방법^[6]에 적용한 결과에 대해 알아 보면 다음과 같다. 실험에 사용한 블럭의 크기는 8×8이고 탐색영역의 크기는 ±7이다. 표 5는 이에 대한 실험결과의 평균 PSNR로 TSS는 기존의 3-step search 방법이고, PTSS1은 기존의 3-step search 방법에 가산투영을 적용하여 1차원 정합만으로 움직임을 검출하는 방법이고^[15] PTSS2는 본 논문에서 제안한 2단계 정합 방법을 사용한 움직임 검출방법이다. Full search

방법의 경우와 마찬가지로 3-step search 방법에서도 제안한 방법이 기존의 방법과 유사한 성능을 가짐을 알 수 있다.



(a)



(b)

그림 6. PSNR 그래프(잡음영상, $\sigma=5$)

- (a) "Miss America" 영상
- (b) "Salesman" 영상

Fig. 6. PSNR graph(Noisy sequences $\sigma=5$).

- (a) "Miss America" sequence.
- (b) "Salesman" sequence.

표 5. 움직임 검출성능에 대한 평균 PSNR (8×8 블럭을 사용한 3-step search 방법)

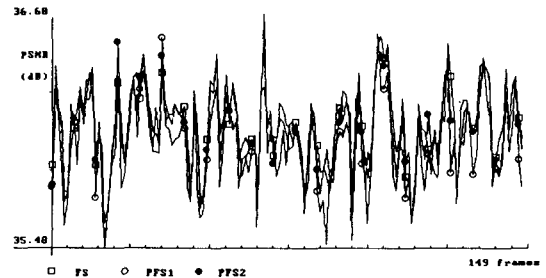
Table 5. Average PSNR for the results of motion detection (3-step search method motion using an 8×8 block).

(단위: dB)

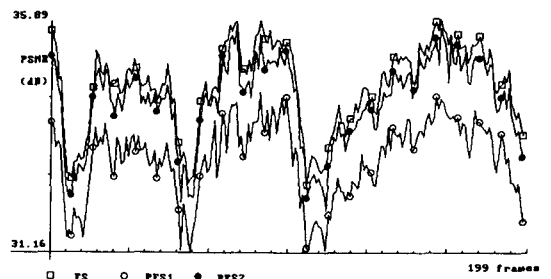
	Miss America	Salesman	Susie
TSS	39.14	36.14	35.19
PTSS1	38.11	35.72	34.54
PTSS2	39.02	36.11	35.18

마지막으로 제안한 방법의 시스템 전체적인 측면에서의 성능분석을 위해 MPEG SM3과 본 논문에서

제안한 방법과 결합한 실험의 결과에 대해 살펴보겠다. MPEG SM3의 실험에서 하나의 GOF (Group of Frame)는 12장의 영상으로 구성되어 있으며, 한장의 intraframe과 3장의 predicted frame 그리고 8장의 interpolated frame의 특성을 갖도록 하였다. 또한 움직임 추정시 half-pel 해상도까지 움직임을 추정하였으며 고경 전송채널의 전송율은 1.15 Mbps이고 전송 버퍼의 용량은 120 kbits로 하고 초기의 버퍼상태는 30 kbits로 가정하였다. [16-18] 표 6에 실험결과와 평균 PSNR을 나타내었다. 표에서 보듯이 세종류의 영상에 대해서 제안한 방법은 기존의 방법과 거의 유사한 성능을 얻었다. 이로써 제안한 움직임 검출방법은 시스템의 전체적인 측면에서도 기존의 방법과 유사한 성능을 갖고 있음을 알 수 있다. 그림 8은 각 실험영상의 luminance 영상에 대한 PSNR 그래프로 MPEG는 MPEG SM3을 나타내며 PMPEG는 제안한 방법과 결합된 MPEG SM3을 나타낸다. 그림에서 보듯이 기존의 방법과 거의 유사한 성능을 나타내고 있다.



(a)



(b)

그림 7. PSNR 그래프(잡음영상, $\sigma=10$)

- (a) "Miss America" 영상
- (b) "Salesman" 영상

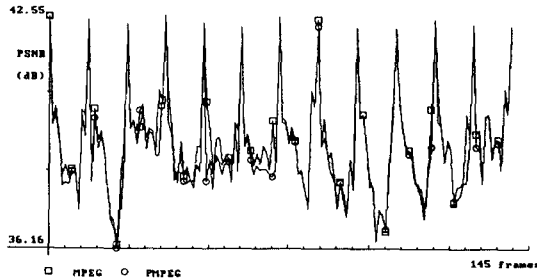
Fig. 7. PSNR graph(Noisy sequences $\sigma=10$).

- (a) "Miss America" sequence.
- (b) "Salesman" sequence.

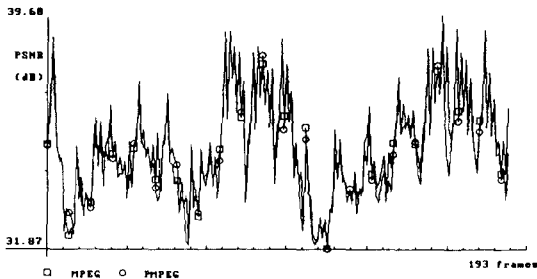
표 6. MPEG SM3에 대한 평균 PSNR
Table 6. Average PSNR for the results of MPEG SM3.

(단위: dB)

	Miss America			Salesman			Susie		
	Y	I	Q	Y	I	Q	Y	I	Q
MPEG	39.07	35.69	37.12	35.36	43.11	44.70	34.82	43.93	43.08
PMPEG	38.91	35.39	37.99	35.27	42.91	44.47	34.74	43.77	42.91



(a)



(b)

그림 8. PSNR 그래프(MPEG SM3)

- (a) "Miss America" 영상
- (b) "Salesman" 영상

Fig. 8. PSNR graph(MPEG SM3).

- (a) "Miss America" sequence.
- (b) "Salesman" sequence.

Ⅶ. 결론

움직임 검출기법은 동영상 전송시스템에서 상당히 중요한 역할을 하는 부분으로 움직임을 얼마나 정확히 검출하느냐에 따라 시스템의 성능이 좌우될 수 있는 부분이며, 지금까지 많은 연구가 진행되었으며 고 성능의 움직임 검출기법들이 제안되었다. 그러나 여러가지 제안된 기법들중에서 하드웨어 구현이 상대적으로 쉽고 계산량도 적당하며, 움직임 검출성능도 비

교적 우수한 방법인 3-step search 방법이 블럭정합 알고리즘으로 가장 많이 사용되고 있다. 그러나 고화질의 영상을 얻기 위해서는 3-step search 방법보다는 탐색영역내의 모든 점을 정합하는 full search 방법이 적합하나 계산량이 방대하여 실시간 처리에 어려움이 많다. 움직임 검출성능은 필요한 계산량에 따라 변하기 때문에 좋은 성능을 갖는 알고리즘일수록 많은 계산량이 필요하여 실시간 처리가 어려운 문제로 남아 있다.

본 연구에서는 움직임 검출시 과다한 계산량을 줄이면서 성능의 열화는 최소가 되게 하는 새로운 2단계 블럭정합 알고리즘을 제안하였으며, 이 방법은 기존의 여러가지 방법과 쉽게 결합이 가능하면서 가산투영을 사용한 1차원 정합과정과 2차원 정합과정의 2단계 정합을 사용하여 16×16 블럭을 사용한 full search 방법에 비해 4배 정도의 계산량 감축을 얻었으며, 여러가지 실험을 통해 제안한 방법의 성능을 평가해 본 결과 제안한 방법은 기존의 full search 방법에 의해 복원된 영상들과 유사한 화질의 영상을 복원할 수 있었다. 또한 가산형 잡음영상에 대해서도 기존의 방법과 유사한 성능을 보이며 잡음이 강할수록 그리고 움직임 추정시 사용블럭의 크기가 커질수록 성능개선 효과가 큼을 알 수 있었다. 그리고 MPEG SM3과 결합하여 실험해 본 결과 시스템 전체적인 측면에서도 기존의 방법과 유사한 성능을 지니면서 계산량을 상당히 감축시켜 고화질의 영상복원시 실시간 처리의 가능성을 높였다.

하드웨어 측면에서도 가산투영 방법은 병렬처리를 이용한 파이프라인 아키텍처를 사용하여 쉽게 구현이 가능하므로 [19, 20] 이에 대한 연구가 좀 더 절실히 요구되며 앞으로 이러한 병렬처리가 가능하게 되면 하드웨어 구현시 더 큰 계산시간의 감축을 얻을 수 있을 것이다.

參考文獻

- [1] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349-389, Mar. 1981.
- [2] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366-406, Mar. 1980.
- [3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984, pp. 252-338.
- [4] H. G. Musmann *et al.*, "Advances in

- picture coding." *Proc. IEEE*, vol. 73, pp. 523-548, Apr. 1985.
- [5] A. N. Netravali and J. D. Robbins. "Motion-compensated television coding: Part I," *Bell Syst. Tech. J.*, vol. 58, pp. 631-670, Mar. 1979.
- [6] F. Giorda and A. Racciu. "Bandwidth reduction of video signal via shift vector transmission." *IEEE Trans. Commun.*, vol. COM-26, pp. 1002-1004, Mar. 1977.
- [7] J. R. Jain and A. K. Jain. "Displacement measurement and its application in interframe image coding." *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1808, Dec. 1981.
- [8] T. Koga et al.. "Motion-compensated interframe coding for video conferencing." in *Proc. Nat. Telecom. Conf.*, Nov. 29-Dec. 3, 1981, pp. G 5.3.1-G 5.3.5.
- [9] Y. Ninomiya and Y. Ohtsuka. "A motion compensated interframe coding scheme for television pictures." *IEEE Trans. Commun.*, vol. COM-30, pp. 201-211, Jan. 1982.
- [10] R. Srinivasan and K. R. Rao. "Predictive coding based on efficient motion estimation." *IEEE Trans. Commun.*, vol. COM-33, pp. 888-896, Aug. 1985.
- [11] S. Kappagantula and K. R. Rao. "Motion compensated interframe image prediction." *IEEE Trans. Commun.*, vol. COM-33, pp. 1011-1014, Sept. 1985.
- [12] M. Ghanbari. "The cross-search algorithm for motion estimation." *IEEE Trans. Commun.*, vol. COM-38, pp. 950-953, July 1990.
- [13] M. Bierling. "Displacement estimation by hierarchical block matching." in *Proc. SPIE Conf. Visual Communications and Image Processing '88*, vol. 1001, pp. 942-951, Cambridge, Mass., Nov. 1988.
- [14] J.-S. Kim and R.-H. Park. "A fast feature-based block matching algorithm using integral projections." *IEEE J. Select. Areas Commun.*, vol. SAC-10, pp. 968-971, June 1992.
- [15] J.-S. Kim and R.-H. Park. "Feature-based block matching algorithm using integral projections." *IEE Electron. Lett.*, vol. 25, pp. 29-30, Jan. 1989.
- [16] ISO/IEC JTC1/SC2/WG11. "MPEG video simulation model three." MPEG90/041, July 1990.
- [17] ISO CD 11172-2 rev 1. "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s." Nov. 1991.
- [18] Specialists Group on Coding for Visual Telephony. "Proposal on extending the scope of rec. H.261 to lower bit rates." CCITT SGXV Doc. #446, 1988.
- [19] J. L. C. Sanz and E. B. Hinkle. "Computing projections of digital images in image processing pipeline architectures." *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, pp. 198-207, Feb. 1987.
- [20] T. Komarek and P. Pirsch. "Array architectures for block matching algorithm." *IEEE Trans. Circuits, Syst.*, vol. CAS-36, pp. 1301-1308, Oct. 1989.

著者紹介

金 俊 植 (正會員) 第29卷 B編 第4號 參照
현재 서강대학교 전자공학과 박사
과정 재학중

朴 來 弘 (正會員) 第23卷 第6號 參照
현재 서강대학교 전자공학과
부교수



李 炳 旭 (正會員) 1957年 1月
14日生. 1979年 2月 서울대학교
전자공학과. (학사) 1981年 8月
한국과학 기술원 전기 및 전자공
학과. (석사) 1991年 6月
Stanford Univ. Electrical
Engineering (Ph.D.) 1981年
8月~1985年 8月 대우전자 중앙연구소 대리. 1990
年 7月~현재 대우전자 영상연구소 책임연구원