

계층적 평활화 방법을 이용한 움직임 추정 알고리즘

정회원 김용태*, 임정은*, 손광훈*

Hierachically Regularized Motion Estimation Technique

Yongtae Kim*, Jungeun Lim*, Kwanghoon Sohn* *Regular Members*

요 약

본 논문에서는 보다 효율적이고 정확한 움직임 벡터를 추정하기 위하여 계층적 평활화 방법(hierachical regularization technique)을 이용한 움직임 추정 알고리즘을 제안한다. 계층적 평활화 기법을 이용하여 움직임 벡터들의 신뢰도를 증가시켰고, 주위 벡터와의 평활화를 통해 움직임 벡터들의 비트량을 감소시켰다. 또한 적은 후보 벡터를 이용하여 움직임 벡터를 예측하는 고속 움직임 추정 알고리즘을 적용하여 평활화 과정의 추가로 인해 생기는 많은 연산량을 감소시켰다. 실험 결과 제안한 계층적 평활화 방법을 이용한 고속 움직임 추정 알고리즘은 전 방향 탐색(full search) 알고리즘과 비교하여 비슷한 영상 화질에서 많은 연산량 감소를 얻을 수 있었으며 잘못된 벡터의 추정 및 확산을 줄일 수 있음을 확인하였다.

ABSTRACT

This paper proposes the hierachically regularized motion estimation technique for the efficient and accurate motion estimation. To use hierachical technique increases the reliability of motion vectors. And the regularization of neighbor vectors decreases bit rate of motion vectors. Also, using fast motion estimation algorithm with a few candidate vectors, the processing time added by regularization can be decreased. In the result of the experiment, the fast motion estimation with hierachical regularization technique achieves less computations and decreases estimation and distribution of false vectors.

I. 서론

오늘날 정보 산업 분야에 있어서 영상, 음성, 데이터 등에 관한 여러 가지 기반 기술과 집적회로 기술, 신호처리 기술, 통신 기술 그리고 다중화 기술이 급속하게 발전함에 따라 방대한 양의 정보를 동시에 다룰 수 있게 됨으로써, 독립적으로 이루어 지던 음성, 영상, 데이터가 하나의 매체로 처리 및 전송되는 멀티미디어의 시대가 도래하게 되었다. 이러한 멀티미디어를 통하여 전달되는 정보 중에서 인간의 시각을 통하여 의미를 전달하는 영상은 그 비중이 매우 크다. 따라서 영상 데이터를 효율적으로 부호화하기 위한 알고리즘이 요구된다.

동영상 부호화^[1-3]는 프레임간의 움직임 추정 및

보상 과정을 통하여 움직임 벡터를 추정한 후 복원 영상과 원 영상간의 차 영상을 변환 부호화하여 공간적인 상관도를 제거함으로써 이루어진다. 움직임 추정 및 보상 과정은 동영상에 존재하는 시간적 중복성(temporal redundancy)을 제거하여 높은 압축률을 얻는데 핵심적인 역할을 담당하고 있다. 그러나 움직임 추정 과정은 동영상을 부호화하는 과정에서 대부분의 연산량을 차지한다. 본 논문에서는 연산량을 줄이기 위해 움직임 벡터간의 상관도를 고려하여 적은 후보 벡터들을 이용하여 움직임 추정을 수행하는 고속 알고리즘을 사용한다. 적은 후보 벡터를 사용하여 연산량을 감소시켰고 이웃 움직임 벡터들간의 상관도를 고려하였기 때문에 보다 정확한 움직임 벡터를 추정한다. 또한 이러한 움직임 벡터

* 연세대학교 전기전자공학과 디지털 영상 매체 연구실(khshon@yonsei.ac.kr)
논문번호 : 010077-0426, 접수일자 : 2001년 4월 26일

들에 대하여 보다 신뢰도를 높이기 위하여 계층적 평활화 방법을 제안하였다.

평활화 과정은 움직임 벡터의 신뢰도를 증가시키고 영상의 화질을 개선 할 수 있다. 즉, 한 객체 또는 배경내에서는 움직임 벡터들이 같은 값을 가지게 되므로 움직임 벡터의 신뢰도를 증가시키고 영상의 화질 또한 개선되어 블록화 현상(blocking artifact)을 줄일 수 있다. 또한 평활화 과정을 통해 얻은 움직임 벡터들은 서로 비슷한 값을 가지게 되므로 움직임 벡터의 정보량 또한 감소시킬 수 있다. 그러나 움직임 벡터 추정 시 잘못된 움직임 벡터가 선택될 경우 평활화 과정에서 잘못된 벡터가 확산되는 결과를 초래할 수 있다. 이런 문제를 해결하기 위해 본 논문에서는 계층적 평활화 방법을 제안한다. 이는 움직임 추정을 할 때 블록 단위를 큰 블록으로 할수록 움직임 벡터의 신뢰성이 증가한다는 점을 이용하여 큰 블록 단위로 움직임 추정을 하여 신뢰성이 있는 후보 벡터를 선택하고 큰 블록을 네 개의 작은 블록들로 나누어 평활화를 수행한다. 계층적 평활화 방법에서 경계 영역부분의 과평활화로 인해 경계가 보존되지 않는 경우를 없애기 위해 경계-보존 방향성 평활화를 적용하여 해결하였다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 전 방향 탐색 알고리즘과 움직임 벡터의 상관도를 이용한 움직임 추정 방법에 대하여 설명하였고, 제 3 장에서는 본 논문에서 제안한 움직임 벡터의 평활화 방법에 대하여 기술하였다. 제 4 장에서는 모의 실험 결과에 대하여 분석하였고 마지막으로 제 5 장에서는 결론을 맺었다.

II. 움직임 추정

움직임 추정 및 보상은 동영상에 존재하는 시간적인 중복성을 제거함으로써 높은 압축률을 얻는데 핵심적인 역할을 담당하고 있다. 동영상 압축 표준인 H.261, H.263, MPEG-2/4^[1-3]등에서는 블록 정합 알고리즘(BMA:block matching algorithm)을 이용하여 움직임 추정을 수행하고 있다^[4-5]. 블록 정합 알고리즘은 현재 프레임을 블록 단위로 나누어 각 블록이 이전 프레임에서 가장 잘 정합이 되는 곳을 찾아 그 위치의 차이, 즉 움직임 벡터만을 저장하는 방법으로 이를 통해서 영상을 적은 정보만을 가지고 부호화 할 수 있게 된다. 또한 수신단에서 영상의 화질을 더욱 좋게 하기 위하여 복원 영상과 원영상의 차 영상 정보를 DCT(discrete cosine trans-

form) 또는 웨이블릿 변환을 이용하여 부호화한다.

1. 전 방향 탐색 알고리즘

전 방향 탐색 알고리즘은 탐색 범위에 해당되는 모든 움직임 벡터에 대해서 정합 척도가 최적인 블록을 찾는다. 이때 사용되는 비용 함수로는 MAE(mean absolute error), MSE(mean square error) 등이 있다. 연산 시간의 단축을 위하여 보통 MAE를 많이 사용한다. 전 방향 탐색 알고리즘을 수행하기 위한 순서는 그림 1에 나타내었다. 모든 검색 범위 내의 후보 벡터에 대해 MAE나 MSE값을 구하고 가장 작은 값을 갖는 후보 벡터를 움직임 벡터로 선택한다. 그러나 모든 탐색 범위에 대해서 움직임 벡터를 추정하면 전체 부호화 과정의 70%~80% 정도의 연산량이 요구된다. 이를 극복하기 위하여 TSS(three step search), logarithm 알고리즘, PDE(partial distortion elimination), SEA(successive elimination algorithm), MRME(multi resolution motion estimation), 움직임 벡터의 상호 상관도를 이용한 움직임 추정 방법등 여러 가지 고속 알고리즘이 제안되었다^[6-10].

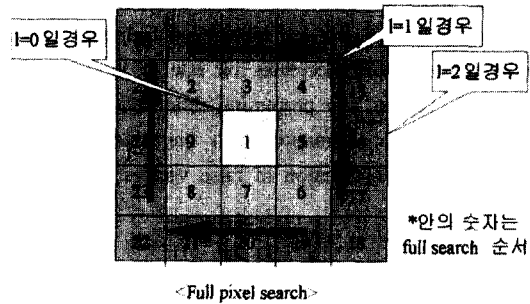


그림 1. 전 방향 탐색 순서

2. 움직임 벡터의 상관도를 이용한 움직임 추정 방법

움직임 벡터의 상관도를 이용하는 움직임 추정 방법은 탐색 범위 전체에 대하여 움직임 추정을 하지 않고 효율적으로 후보 움직임 벡터를 선택하여 연산 시간을 감소시키는 알고리즘이다^[6]. 미리 구한 주위의 벡터를 후보 벡터로 하여 현재 블록에 대하여 가장 적은 오류를 가지는 벡터를 선택한다. 그리고 벡터의 정확도를 높이기 위하여 가장 적은 오류를 나타내는 후보 벡터 주변으로 ± 1 만큼 재탐색을 하고 그 중 가장 적은 오류의 움직임 벡터를 선택한다. 그러나 이렇게 구한 움직임 벡터에 대한 오류

가 정해진 임계값보다 클 경우에는 4 : 1 서브샘플링 전 방향 탐색(subsampling full search)을 하여 움직임 벡터를 찾게 된다. 후보 벡터를 선택하는 방법은 공간적인 벡터의 상관성을 이용하는 방법과 시간축으로 확장하는 방법등이 있다.

먼저 공간적 상관도를 이용한 방법은 Causal한 방법(S1)과 non-Causal한 방법(S2)이 있는데 Causal한 방법은 다음과 같다. 그림 2에서 영상의 좌상단의 B(0, 0), B(1, 0), B(0, 1), B(1, 1)블록에 대해 4 : 1 서브샘플링 전 방향 탐색을 하여 움직임 벡터를 구한다. 이 블록들은 주변 블록이 없는 영상의 끝부분에 위치하여 후보 벡터로 사용할 수 있는 블록이 없으므로 전 방향 탐색을 하게 되는 것이다.

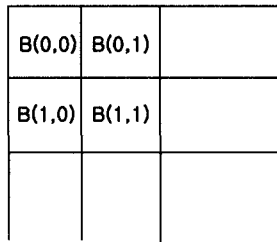


그림 2. 초기화

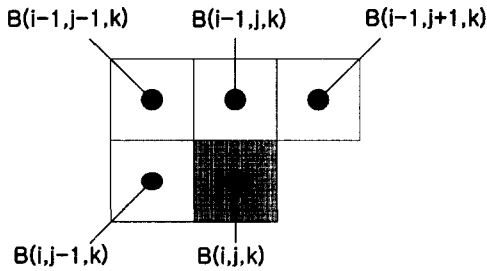


그림 3. 후보 벡터의 선택

예를 들어 그림 3에서 B(i, j, k)가 k번째 프레임에서 가로 i, 세로 j 번째 블록이라 할 때 B(i-1, j-1, k), B(i-1, j, k), B(i-1, j+1, k), B(i, j-1, k) 블록들의 움직임 벡터를 후보 벡터로 사용하여 현재 블록에 가장 작은 MAE값(MAD_0)을 내는 벡터, V_0 를 선택한다. 이 때 MAD_0 가 TH_1 값보다 작으면 V_0 를 그 블록의 벡터로 설정하고 다음 블록으로 이동하여 다시 그 블록의 주변 벡터들을 이용하여 V_0 를 구하게 된다. 여기서 보다 정확한 벡터를 찾기 위해서 V_0 를 주변으로 벡터를 찾는 정밀화 과정(refinement process)을 포함시킬 수 있다. 또한

MAD_0 값이 TH_1 보다 클 경우에는 V_0 를 주변으로 정밀화 과정을 반복 수행하고 종결 조건을 확인하는데 다음 세 가지 경우가 있을 수 있다. 먼저 정밀화 과정 후 가장 작은 MAD가 TH_1 보다 작은 경우 그 벡터를 V_0 로 선택하고 다음 블록으로 넘어간다. 두 번째로 정밀화 과정 후 가장 작은 MAD가 탐색 범위의 가운데에 빠질 경우 즉, 가장 작은 오류를 내는 벡터가 이전 정밀화 과정에서 나온 벡터와 동일할 때는 TH_2 (TH_1 보다 큰 값)와 비교하여 클 경우에는 4 : 1 서브 샘플링 전 방향 탐색을 수행하고 작을 경우 그 벡터를 V_0 로 선택하고 다음 블록으로 넘어간다. 마지막으로 정해진 반복 수행 횟수를 넘어갈 경우에는 4 : 1 서브 샘플링 전 방향 탐색하여 V_0 를 선택한다.

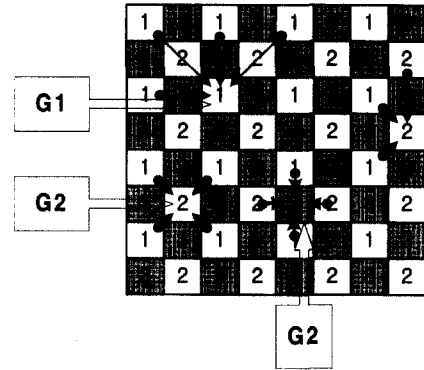


그림 4. 각 그룹에서의 블록 선택

위의 방법은 Causal한 벡터만 가져오므로 주변 블록의 벡터에 대해 상관성을 100% 사용하였다고 볼 수 없다. 이런 점을 해결하기 위해 non-Causal한 방법은 그림 4에서와 같이 세 개의 그룹으로 블록들을 나누어 움직임 추정을 하게 되는데 G1(Group 1)의 블록은 Causal한 방법을 이용하여 움직임 벡터를 구하고 G2는 G1에서 구한 벡터를 후보 벡터로 사용하여 벡터를 구한다. 이 때 G2내의 경계 블록들은 후보 벡터들이 부족해지는데 이를 피하기 위해 그림 4에서처럼 상단에 있는 G2의 블록을 후보 벡터에 포함시킨다. G3의 경우 G2의 벡터들을 후보 벡터로 하여 움직임 추정을 하게 되는데 G3는 다른 G1과 G2와 비교하여 바로 인접한 블록의 벡터를 후보 벡터로 이용하기 때문에 매우 정확한 움직임 추정을 하므로 정밀화 과정을 필요로 하지 않는다.

III. 계층적 평활화 알고리즘

1. 고속 평활화 알고리즘

움직임 추정을 할 때 현재 블록의 벡터는 주변 벡터들과의 상관도를 고려하지 않고 현재 블록과 탐색 범위 내에 있는 추정 블록간의 오류만을 고려하기 때문에 움직임 벡터 맵을 살펴보면 한 객체에 대해 불구하고 벡터들이 균일하게 분포하지 않는다. 이는 움직임 벡터를 주변 벡터와의 상관도를 고려하지 않고 단지 MSE나 MAE를 최소로 하는 블록을 탐색하는 데 기인한다. 따라서 객체와 상관없이 비용 함수만을 이용하여 움직임 벡터를 추정하는 것은 신뢰도가 감소하게 된다. 이를 해결하기 위하여 주변 벡터와의 상관도를 고려하여 벡터의 신뢰도를 증가시키는 평활화(regularization)과정이 필요하다^[11-12].

영상의 경계 영역을 보존하면서 평활화된 경계 정보를 갖도록 식(1)과 같은 비용 함수를 사용한다. 식(1-1)의 f_1 은 보통의 MAE를 이용한 비용 함수이고 식(1-2)의 f_2 는 현재 후보 벡터와 주변 벡터(\bar{N}_k)와의 차이값을 나타내는 파라미터이다. 주변 네 블록의 움직임 벡터와 현재 후보 벡터와의 차를 구하여 그 절대값의 합을 비용 함수로 사용한다. 단순히 벡터의 차만을 이용할 경우 경계 영역에서 평활화로 인해 경계가 깨지는 현상이 발생할 수 있는데 이를 방지하고자 경계 영역에서는 b_k, b_c 등의 Sobel 값을 이용하여 평활화를 수행하는 f_2 의 효과를 줄인다. 여기서 b_k 는 각 경계면에서의 Sobel 평균값이고 b_c 는 현재 블록의 전체 Sobel 평균값이다. 식(1)에서 λ 는 라그랑제 계수로서 평활화 정도를 나타내는 값으로 작으면 평활화의 효과를 낼 수 없고 계수가 크게 되면 평활화가 많이 되지만 과평활화되는 효과가 일어날 수 있기 때문에 적당한 값을 실험적으로 선택하여야한다. 마지막으로 식(2)에서처럼 비용 함수를 최소화하는 움직임 벡터 \bar{m}_x 를 정한다.

$$f = f_1 + \lambda f_2 \quad (1)$$

$$f_2 = \sum_{k=0}^3 \frac{|\bar{N}_k - \bar{m}_k|}{b_k + b_c + 1}, \quad \bar{m}_k = (m_l, m_m) \quad (1-1)$$

$$f_1 = \sum_{i,j \in \text{Block}} |I_t(i, j) - I_{t-1}(i + m_l, j + m_m)| \quad (1-2)$$

$$\bar{m}_x = \arg(\min_{m_k} f) \quad (2)$$

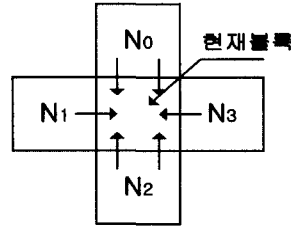


그림 5. 고속 평활화 방법

보통 평활화하는 방법은 반복 과정을 통해 수렴할 때까지 수행하지만 이는 너무 많은 연산량을 요구하게 된다. 본 논문에서는 많은 연산량을 감소시키기 위하여 고속 알고리즘을 사용하였다. 고속 알고리즘은 벡터들 간의 상호 상관도를 이용하여 수행하였다. 후보 벡터로 그림 5에서와 같이 상·하·좌·우 벡터를 사용하였고 후보 벡터와 현재 블록의 벡터에 식(1)을 이용하여 가장 적은 오류 값을 나타내는 벡터 V_0 를 선택한다. 또한 움직임 벡터 추정에 신뢰도를 증가시키기 위하여 V_0 를 기준으로 하여 ± 1 화소 주변으로 재탐색을 수행하였다. 이 고속 평활화 알고리즘을 이용하여 평활화 효과를 얻을 수 있었고 연산량을 감소시킬 수 있었다.

2. 계층적 평활화 알고리즘

제안된 고속 평활화 알고리즘은 현재 블록에서 상·하·좌·우 네 블록의 벡터만을 후보벡터로 하여 평활화를 하므로 초기에 정확한 움직임 벡터를 찾지 못할 경우 잘못 찾은 벡터로 인하여 주변 블록들의 움직임 벡터들도 신뢰도가 감소할 수 있다. 따라서 초기에 정확한 움직임 벡터를 추정하지 못한 경우, 평활화 방법을 이용하면 잘못된 움직임 벡터의 확산을 초래할 수 있다. 이런 문제를 해결하기 위하여 본 논문에서는 계층적 평활화 방법을 제안하였다. 움직임 추정에서 블록의 크기가 커질수록 객체의 세밀한 움직임은 표현을 못하지만 블록 전체의 움직임에 대해서는 신뢰성을 높일 수 있으므로 큰 블록 단위로 움직임 추정을 하면 작은 블록 단위로 움직임 추정을 했을 때 자주 발생하는 잘못된 움직임 벡터의 추정을 감소시킬 수 있다. 이런 성질을 이용하여 본 논문에서는 초기에는 큰 블록에서 움직임 추정과 평활화를 한 후 큰 블록을 작은 네 개의 블록들로 나누어 다시 평활화와 정밀화 과정을 하게 된다. 그림 6은 본 논문에서 제안한 알

고리들을 나타낸다. 먼저 16×16 블록 단위로 움직임 벡터의 추정을 하게 되는데 이때 주변 벡터와의 상관성을 이용한 고속 움직임 추정 알고리즘을 이용한다. 이렇게 구한 벡터들은 평활화 알고리즘을 이용하여 평활화하게 된다. 여기서 평활화된 움직임 벡터들은 16×16 블록 크기로 수행하였기 때문에 세밀한 움직임들을 포함하지 못하여 물체의 경계들이 많이 깨지거나 블록화 현상이 발생하기도 한다. 이런 점을 보완하기 위해 8×8 블록 단위로 평활화 알고리즘을 수행한다. 정밀화 과정은 고속 움직임 추정을 사용함으로써 발생하는 후보 벡터 부족으로 움직임 추정과 평활화 과정으로 나온 벡터가 원래의 벡터와 미세한 차이가 생길 수 있는 현상을 줄인다. 평활화 과정을 통해 나온 벡터 주위 ±1 화소 재탐색하여 MAE값이 가장 작은 오류를 내는 벡터를 정밀화된 움직임 벡터로 구하게 된다. 이러한 정밀화 과정은 평활화 과정을 통하여 PSNR이 낮아지는 현상을 감소시킬 수 있다.

IV. 실험 결과

본 장에서는 계층적 평활화 방법을 이용한 고속 움직임 추정에 대한 부호화기의 모의 실험 및 결과를 고찰한다. 모의 실험은 Pentium-II PC에서 Visual C++ 6.0을 이용하여 수행하였다. 실험 영상은 그림 7의 ‘Train and tunnel’(688×560, 256 level) 영상과 ‘Garden(720×486, 256 level)’ 영상을 사용하였고 움직임추정 시 블록의 크기는 8×8을 기준으로 하였다. 비교 알고리즘으로는 평활화를 하지 않은 전 방향 탐색 알고리즘과 계층적 구조를

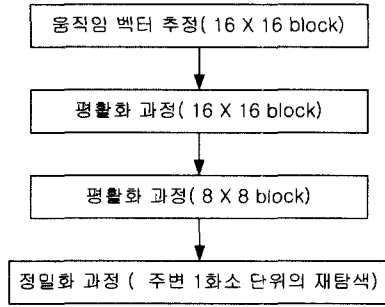


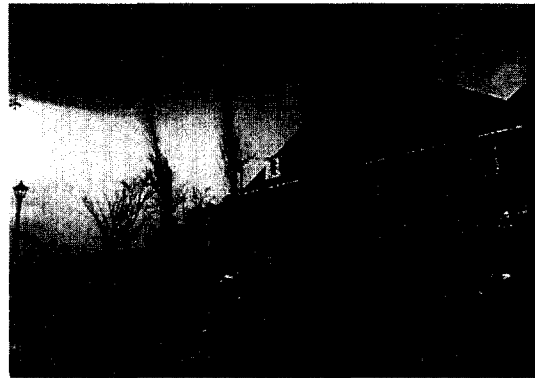
그림 6. 계층적 평활화 알고리즘

갖지 않는 일반적인 평활화 알고리즘을 사용하였다. 각 알고리즘에 대한 움직임 벡터 맵은 흰색은 ‘+’ 방향, 검정색은 ‘-’ 방향의 움직임 벡터를 의미하며 벡터의 크기는 선의 길이로 나타내었다. 움직임 벡터의 코딩은 MPEG-2의 VLC(variable length coding) 테이블을 사용하였다.

‘Train and tunnel’ 영상은 다른 부분에는 움직임이 거의 없고 기차 부분에서 움직임이 크게 나타난다. 따라서 그림 8의 움직임 벡터 맵에서 움직임이 거의 없는 부분은 벡터들이 없고 움직임이 큰 기차 부분에서 벡터들이 크게 나타난다. 그림 8(a)는 전 방향 탐색 알고리즘의 벡터맵으로 MAE값만을 가지고 벡터를 찾을 경우 실제 벡터와는 상관없는 벡터가 추정되는 경우가 많다. 한 객체내의 움직임 벡터는 대부분 비슷한 값을 가지고 있지만 8(a)의 벡터 맵에서 기차에 해당되는 블록들의 벡터들은 불규칙하게 분포됨을 볼 수 있다. 8(b)에서 일반 평활화 알고리즘을 통해 많은 벡터들이 균일화됨을 확인할 수 있었다. 8(a)에서 우상단의 터널부분에 해당되는



(a) Train and tunnel: frame #3

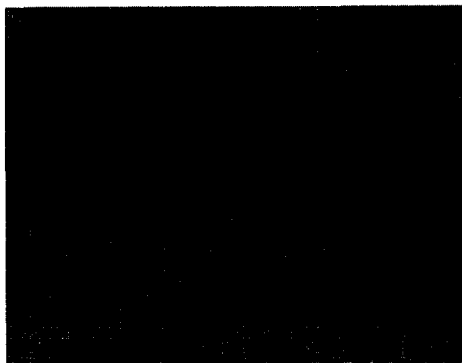


(b) Flower and Garden: frame #5

그림 7. 실험영상

부분은 화소들의 값이 균일하여 MAE만을 가지고 추정할 경우 잘못된 움직임 벡터들이 추정되지만 8(b)에서는 일반 평활화 알고리즘을 통해 잘못된 움직임 추정을 피할 수 있다. 하지만 8(b)에서 대부분의 벡터들이 균일하게 되지만 움직임 추정단계에서 잘못 찾은 벡터가 평활화과정에서 확산되어 몇몇 부분에서 기차의 움직임과는 정반대의 움직임 벡터가 추정되기도 한다. 8(c)에서는 계층적 방법을 사용하여 이런 문제를 해결하였다. 8(b)에서 나타나는 특정 부분에서의 문제점이 8(c)의 벡터맵에서는 대부분 사라지는 것을 볼 수 있다. 이는 보다 큰 블록 크기로 움직임 추정 단계와 평활화 단계를 추가하여 신뢰성이 높은 움직임을 후보벡터로 하여 평활화를 수행하기 때문이다.

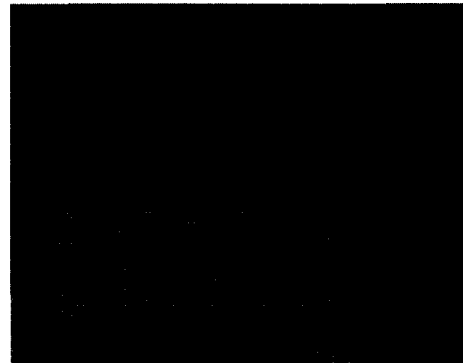
표 1에서 보는 바와 같이 제안된 알고리즘이 전 방향 탐색 알고리즘에 비하여 PSNR이 약간 떨어짐을 볼 수 있는데 이는 움직임 벡터 추정시 비용 함수에 MAE외에 벡터들의 차값이 포함되어 최소의 MSE를 내는 움직임을 벡터를 선택하지 않아 발생하는 현상이다. 보통 평활화 과정이 진행될수록 PSNR은 떨어지게 되는 trade off관계를 보이고 있다. 부호화된 비트 수 면에서 제안된 방식이 전 방향 탐색에 비해 약 70%로 감소하는 것을 볼 수 있는데 이는 현재 움직임을 이전 블록의 움직임 벡터와 차를 구하여 코딩하기 때문에 평활화가 잘된 벡터맵일수록 움직임 벡터간의 차값이 작아져서 적은 비트 수로 코딩된다. 또한 고속 알고리즘을 사용하여 연산 시간도 'Train and tunnel' 영상에 대해 실험한 결과 일반 평활화 알고리즘에서는 24.100sec가 소요되나 제안 알고리즘에서는 7.885sec가 소요되어 약 30%로 감소하는 것을 확인하였다. 그림 9와 그림 10은 움직임 보상이며 복원한 영상이다. 영상 화질 면에서 거의 비슷한 결과가 나옴을 볼 수 있다.



(a) 전방향 탐색 알고리즘의 벡터 맵



(b) 일반 평활화 알고리즘의 벡터 맵



(c) 제안 알고리즘의 벡터 맵

그림 8. 움직임 벡터 맵(Train and tunnel: frame #3)

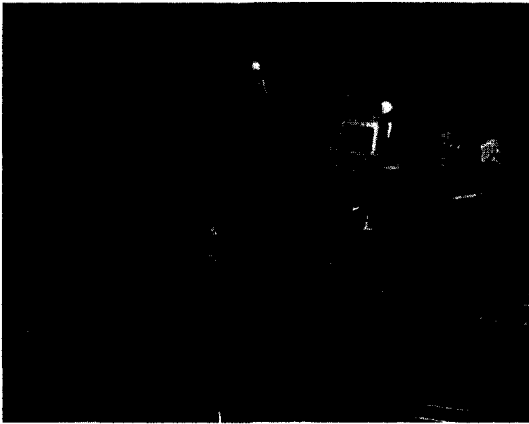
표 1. 각 알고리즘의 성능 평가

(a) Flower and Garden: frame #5

	전방향 탐색 알고리즘	일반 평활화 알고리즘	제안 알고리즘
PSNR [dB]	30.94	30.16	30.02
Coded Bits [bits]	33,262	27,065	24,444

(b) Train and Tunnel: frame #3

	전방향 탐색 알고리즘	일반 평활화 알고리즘	제안 알고리즘
PSNR [dB]	34.82	33.31	33.35
Coded Bits [bits]	17,964	14,847	13,401



(a) 전방향 탐색 알고리즘의 복원 영상



(a) 전 방향 알고리즘의 복원 영상



(b) 제안된 알고리즘의 복원 영상

그림 9. 복원한 영상 (Train and tunnel: frame #3)



(b) 제안된 알고리즘의 복원 영상

그림 10. 복원한 영상 (Flower and Garden: frame #5)

V. 결론

본 논문에서는 보다 효율적이고 정확한 움직임 벡터를 추정하기 위하여 계층적 평활화 방법을 이용한 움직임 추정 알고리즘을 제안하였다. 제안 방식을 통하여 잘못된 벡터의 추정을 줄여 벡터의 신뢰도를 보다 증가시켰고 움직임 벡터의 코딩 효율도 향상시켰다. 또한 영상의 블록화 현상을 감소시켜 주관적인 화질을 개선하였다. 특히 배경에서 화소값들이 일정하여 MAE만을 비용함수로 하여 판단할 경우 무질서한 벡터들이 나오게 되는데 평활화를 통해 이러한 문제를 해결할 수 있었다. 고속 움직임 추정 알고리즘을 사용하여 연산 시간을 최소화하였고 경계-보존 방향성 평활화 방법을 이용하여 경계면에서 과평활화되는 현상을 감소시켰다.

실험 결과 전 방향 탐색 알고리즘과 비교하여 PSNR은 약간 감소하지만 평활화를 통하여 움직임 벡터의 균일화를 통해 약 70%정도 움직임 벡터의 코딩 효율을 높일 수 있었다. 연산 시간측면에서도 기존 방식의 30%정도로 시간을 감소시키는 것을 확인하였다.

참고 문헌

- [1] S. Okubo, K. McCann, A. Lippman, "MPEG-2 requirements, profiles and performance verification - Framework for developing a generic video coding standard," Signal Processing: Image Comm., vol. 7, pp. 201-209, Sep. 1995.

[2] MPEG-4 Video Verification Model Version 8.0, July 1997.

[3] A. Puri, R.V. Kollaritis, B.G. Haskell, "Basic of stereoscopic video, new compression results with MPEG-2 and a proposed for MPEG-4," Signal Processing: Image Comm., vol 10, pp. 201-234, 1997.

[4] J. C. Tsai, C. H. Hsieh "Block-matching motion estimation using correlation search algorithm," Signal Processing: Image Comm. 13, pp.119-133, 1998.

[5] Puri, R. V. Kollaritis, and B. G. Haskell, "Stereoscopic Video Compression Using Temporal Scalability," Proceedings SPIE Visual Communications and Image Processing, May 1995.

[6] J. Chalidabhongse and C.-C Jay Juo, "Fast Motion Vector Estimation Using Multiresolution-Spatio-Temporal Correlations," IEEE Trans. on Circuits Syst. Video Tech., vol. 7, no. 3, pp. 477-488, Jun. 1997.

[7] H. S. Wang and M. Mersereau, "Fast Algorithm for the Estimation of Motion Vectors," IEEE Trans. on Image Processing, vol. 8, no. 3, pp. 435-438, Mar. 1999.

[8] J. N. Kim and T. S. Choi, "A Fast Three-Step Search Algorithm with Minimum Checking Points Using Unimodal Error Surface Assumption," IEEE Trans. on Consumer Electronics, vol. 44, no. 3, pp. 638-648, Aug. 1998.

[9] E. Salari, W. Li, "A fast quadtree motion segmentation for image sequence coding," Signal Processing: Image Communication, vol. 14, pp. 811-816, 1999.

[10] S. Sethurama, "Stereoscopic Image Sequence Compression Using Multiresolution and Quadtree Decomposition Based Disparity and Motion Adaptive Segmentation," Ph.D. Thesis, Carnegie Mellon Univ., Pittsburgh PA, 1996.

[11] M. H. Kim and K. H. Sohn, "Edge-preserving directional regularization technique for disparity estimation of stereoscopic images," IEEE Trans. on Consumer Electronics, vol. 45, no. 3, Aug. 1999.

[12] J.Y. Chang, R.F. Chang, "Edge-based motion compensated classified DCT with quadtree for image sequence coding," Signal Processing: Image Communication, vol. 11, pp. 187-197, 1998.

김 용 태(Yongtae Kim)



2000년 2월 : 연세대학교
기계전자공학부(학사)
2000년 3월~현재 : 연세대학교
전기전자공학과
석사과정

<주관심 분야> 영상 및 비디오 압축, 움직임 및 변이 추정, 3차원 동영상 압축기법

임 정 은(Jungeun Lim)



1999년 2월 : 경원대학교
전자공학과, 공학사
2001년 2월 : 연세대학교
전기전자공학과 석사
2001년 3월~현재 : 연세대학교
전기전자공학과
박사과정

<주관심 분야> 3차원 영상 부호화/복호화, 영상 압축, 영상 통신

손 광 훈(Kwanghoon Sohn)

한국 통신학회 논문지 제 25권 7A호 참조