

계층적 방법을 이용한 움직임 벡터의 고속 평활화 알고리즘

김 용태*, 임 정은, 손 광훈*

* 연세 대학교 전기·전자 공학부

Fast Motion Vector Estimation using Hierarchical Regularization Technique

Yong Tae Kim, Jung Eun Lim, Kwang Hoon Sohn
Dept. of Electrical & Electronics Eng, Yonsei University

E-mail : khsohn@yonsei.ac.kr, wizard97@diml.yonsei.ac.kr

요 약

본 논문에서는 보다 효율적이고 정확한 움직임 벡터를 추정하기 위하여 계층적 평활화 방법(hierarchical 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.

I. 서 론

움직임 추정 및 보상 과정은 동영상에 존재하는 시간적 중복성(temporal redundancy)을 제거하여 높은 압축률을 얻는데 핵심적인 역할을 담당하고 있다.[1] 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 알

고리즘을 사용한다.[2] 적은 후보 벡터를 사용하여 연산량을 감소시켰고 이웃 움직임 벡터들간의 상관도를 고려하였기 때문에 보다 정확한 움직임 벡터를 추정한다. 또한 이러한 움직임 벡터들에 대하여 보다 신뢰도를 높이기 위하여 계층적 평활화 방법을 제안하였다.

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 움직임을 할 때 블록 단위를 큰 블록으로 할수록 움직임 벡터의 신뢰성이 증가한다는 점을 이용하여 큰 블록 단위로 움직임을 하여 신뢰성이 있는 후보 벡터를 선택하고 큰 블록을 네 개의 작은 블록들로 나누어 평활화를 수행한다.[3] 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 본 논문에서 제안한 움직임 벡터의 평활화 방법에 대하여 기술하였다. 제 3 장에서는 모의 실험 결과에 대하여 분석하였고 마지막으로 제 4 장에서는 결론을 맺었다.

II. 계층적 평활화 알고리즘

II-1. 고속 평활화 알고리즘

영상의 경계 영역을 보존하면서 평활화된 경계 정보를 갖도록 식(1)과 같은 비용 함수를 사용한다.[4] 식(1-1)의 f_1 은 보통의 MAE를 이용한 비용 함수이고 식(1-2)의 f_2 는 현재 후보 벡터와 주변 벡터(\overline{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 과평활화 되는 효과가 일어날 수 있기 때문에 적당한 값을 실험적으로 선택하여야한다. 마지막으로 식(2)에서처럼 비용 함수를 최소화하는 움직임 벡터 \overline{m}_x 를 정한다.

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

$$f_1 = \sum_{i,j \in block} |I_t(i,j) - I_{t-1}(i+m_i, j+m_m)| \quad (1-1)$$

$$f_2 = \sum_{k=0}^3 \frac{|\overline{N}_k - \overline{m}_k|}{b_k + b_c + 1}, \quad \overline{m}_k = (m_l, m_m) \quad (1-2)$$

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

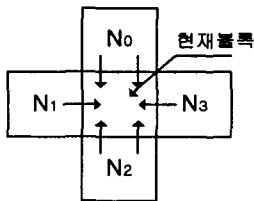


그림 1. 고속 평활화 방법

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 1에서와 같이 상·하·좌·우 벡터를 사용하였고 후보 벡터와 현재 블록의 벡터에 식(1)을 이용하여 가장 적은 오류의 벡터 V_0 를 선택한다.

III-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 못 하지만 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 2는 본 논문에서 제안한 알고리즘을 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.

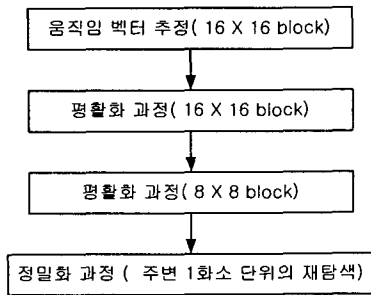


그림 2. 계층적 평활화 알고리즘

IV. 실험 결과

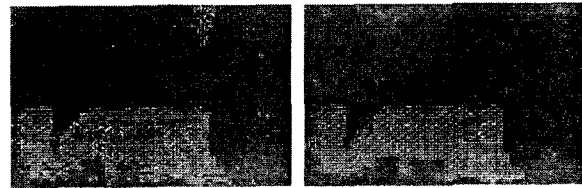
실험 영상은 그림 3의 'Train and tunnel'(688×560, 256 level) 영상과 'Garden(720×486, 256 level)'영상을 사용하였고 움직임 추정시 블록의 크기는 8×8을 기준으로 하였다. 비교 알고리즘으로는 평활화를 하지 않은 전 방향 탐색 알고리즘과 계층적 구조를 갖지 않는 일반적인 평활화 알고리즘을 사용하였다. 각 알고리즘에 대한 움직임 벡터 맵은 흰색은 '+' 방향, 검정색은 '-' 방향의 움직임 벡터를 의미하며 벡터의 크기는 선의 길이로 나타내었다. 움직임 벡터의 코딩은 MPEG-2의 VLC(variable length coding) 테이블을 사용하였다.

'Train and tunnel' 영상은 다른 부분에는 움직임이 거의 없고 기차 부분에서 움직임이 크게 나타난다. 따라서 그림 4의 움직임 벡터 맵에서 움직임이 거의 없는 부분은 벡터들이 없고 움직임이 큰 기차 부분에서 벡터들이 크게 나타난다. 그림 4(a)에서 MAE값만을 가지고 벡터를 찾을 경우 실제 벡터와는 상관없는 벡터가 추정되는 경우가 많다. 한 객체내의 움직임 벡터는 비슷한 값을 갖지만 그림 4(a)에서 기차에 해당되는 블록들의 벡터들은 불규칙하게 분포됨을 볼 수 있다. 그림 4(b)에서 일반 평활화 알고리즘을 통해 많은 벡터들이 균일화됨을 확인할 수 있었다. 그림 4(a)에서 우상단의 터널 부분에 해당되는 부분은 화소들의 값이 균일하여 MAE만을 가지고 추정할 경우 잘못된 움직임 벡터들이 추정되지만 그림 4(b)에서는 일반 평활화 알고리즘을 통해 잘못된 움직임 추정을 피할 수 있다. 하지만 그림 4(b)에서 대부분의 벡터들이 균일하게 되지만 움직임 추정 단계에서 잘못 찾은 벡터가 평활화과정에서 확산되어 몇몇 부분에서 기차의 움직임과는 정반대의 움직임 벡터가 추정되기도 한다. 그림 4(c)에서는 계층적 방법을 사용하여 이런 문제를 해결하였다. 그림 4(b)에서 나타

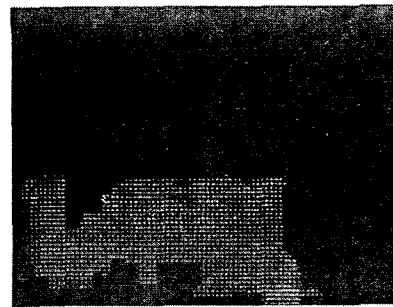


(a) Train and tunnel: frame #3 (b) Flower and Garden: frame #5

그림 3. 실험영상



(a) 전방향 알고리즘의 벡터 맵 (b) 일반 평활화 알고리즘의 벡터 맵



(c) 제안 알고리즘의 벡터 맵

그림 4. 움직임 벡터 맵(Train and tunnel: frame #3)

나는 특정 부분에서의 문제점이 그림 4(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 MAE 외에 벡터들의 차값이 포함되어 최소의 MSE를 내는 움직임을 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 감소하

는 것을 볼 수 있는데 이는 현재 움직임 벡터를 이전 블록의 움직임 벡터와 차를 구하여 코딩하기 때문에 평활화가 잘된 벡터 맵일수록 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 5와 그림 6은 움직임 보상하여 복원한 영상이다. 주관적인 영상 화질 면에서 거의 비슷한 결과가 나올 수 있다.

표 1. 각 알고리즘의 성능 평가
(a) Flower and Garden: frame #5

	전방향 탐색	일반 평활화	제안 알고리즘
PSNR(dB)	30.94	30.16	30.02
Bits	33,262	27,065	24,444

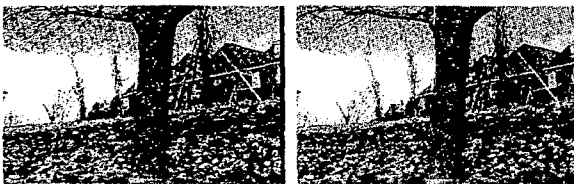
(b) Train and Tunnel: frame #3

	전방향 탐색	일반 평활화	제안 알고리즘
PSNR(dB)	34.82	33.31	33.35
Bits	17,964	14,847	13,401



(a) 전방향 알고리즘의 복원 영상 (b) 제안된 알고리즘의 복원 영상

그림 5. 복원한 영상 (Train and tunnel : frame #3)



(a) 전방향 알고리즘의 복원 영상 (b) 제안된 알고리즘의 복원 영상

그림 6. 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.

감사의 글: 본 논문은 산업자원부에서 시행한 차세대 신기술 개발사업에서 수행한 연구결과입니다.

참고 문헌

- [1] J. C. Tsai, C. H. Hsieh "Block-matching motion estimation using correlation search algorithm," Signal Processing: Image Comm. 13, pp.119-133, 1998.
- [2] 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.
- [3] E. Salari, W. Li, "A fast quadtree motion segmentation for image sequence coding," Signal Processing: Image Communication, vol. 14, pp. 811-816, 1999.
- [4] 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.