변형된 비용 함수를 이용한 움직임 추정 기법

°조한욱, 서정욱, 박재홍*, 정제창
한양대학교 전자통신공학과
전자통신연구원 전파 공학 연구부*
E-mail : jjeong@icsp2.hanyang.ac.kr

Motion Estimation Using Modified Cost Functions

°H. W. Cho, J. W. Suh, J. H. Park and J. Jeong
Department of Electronic Communications Engineering, Hanyang University
Radio Technology Section, ETRI*

동영상 압축 알고리즘에서 움직임 추정기법은 매우 중요한 역할을 담당하는 반면, 수행시간이나 하드웨어 구현에 어려움이 많아 이를 개선하기 위한 알고리즘들이 개발되어 왔다. 본 논문에서는 적절한 화소 분류를 통해 우수한 화질과 적은 계산량, 간단한 하드웨어 구조를 가지는 새로운 움직임 추정기법을 제안한다. 기존의 1-비트 화소 분류 방법에서 변형된 새로운 비용 함수를 이용한 2-비트, 3-비트 화소 분류 방법과 2차 비용함수를 이용한 화소 분류 방법을 제안한다. 또한 여러 고속 움직임 추정 알고리즘과도 쉽게 연결하여 사용할 수 있으며 우수한 성능을 나타내는 것을 모의 실험을 통해 보였다.

Ⅰ. 서론

최근의 디지틀 기술의 발달로 인해 멀티미디어 통신, 화상회의, HDTV, VOD등의 여러 가지 멀티미디어 서비스가 대중화됨에 따라 동영상 데이타의 효율적인 압축 기술의 개발이 중요한 문제로 대두되고 있다. 동영상 압축 기술 중에서도 움직임 추정(Motion Estimation)기법은 동영상 압축 알고리즘에서 매우 중요한 역할을 담당하는 반면, 계산량이 많아 수행시간이나 하드웨어 구현 시 어려움이 많으므로, 이를 개선하기 위한 많은 알고리즘들이 개발되어 왔다. 본 논문에서는 적절한 임계치를 이용한 화소 분류를 통해 우수한 화질과 적은 계산량, 간단한 하드웨어 구조를 가지는 효율적인 블록 정합 기준을 이용한 움직임 추정 기법을 제안한다.

효율적인 동영상 데이타의 압축은 시간적 중복성과 공간적 중복성을 제거함으로써 이루어진다. 이중에서 시간적 중복성을 제거하는 방법은 연속적인 영상 신호에서 현재 프레임의 화소들이 이전 프레임에 비해 어느 정도 움직였는지를 벡터로 나타낸 움직임 벡터를 추정하여 전체 영상의 전송대신 움직임 벡터를 전송함으로써 전송 데이타량을 줄이는 방법이다.

본 논문에서 제안하는 움직임 벡터 추정 방법은 기존의 전역 탐색 방법에서처럼 현재 프레임과 이전 프레임에서 대응되는 블럭간의 화소들의 차(difference)를 구한 후 기존의 움직임 벡터 결정 기준인 평균 제곱 오차(MSE)나 평균 절대 오차(MAE)를 쓰는 대신 적절한 임계치(Threshold)를 정해 화소들을 분류한 후 각 구간별로 대표값을 주어 그 대표값들의 합이 최소가 되는 블럭을 움직임 벡터로 추정하는 방법이다.

실험은 구간을 각각 두 구간, 네 구간, 여덟 구간으로 나누어 실험하였다. 그 결과는 계산량과 하드웨어 구현에 있어 기존의 전역 탐색 방법에 비해 많은 개선이 있었으며, 화질면에 있어서도 평균 절대 오차(MAE)를 이용한 방법보다 월등한 성능을 보였고, 평균 제곱 오차(MSE)를 이용한 방법에 거의 근접하는 성능을 보였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 일반적인 움직임 추정 기법과 블록 정합 기준에 대해서 설명하고, 3장에서는 제안하는 방법인 화소 분류를 통한 새로운 블록 정합 기준, 즉 1-비트, 2-비트, 3비트 화소 분류 방법, 그리고 2차 비용 함수를 이용한 화소 분류 방법에 대해서 설명하고, 이 새로운 블록 정합 기준과 다른 고속 알고리즘과의 결합에 대해서 설명한다. 4장에서는 위의 여러 가지 알고리즘들에 대한 다양한 실험 결과와 그 결과에 대한 고찰을 하였으며, 마지막으로 5장에서는 이상의 결과를 종합하여 결론을 지었다.

Ⅱ. 블록 정합 알고리즘

움직임 추정 및 보상 부호화에 있어서 가장 중요한 것은 이동 정보를 갖고 있는 움직임 벡터를 정확히 검출하는 것이며, 그 방법에 따라 블록 정합 알고리즘(Block Matching Algorithm)과 화소 순환 알고리즘(Pel Recursive Algorithm)으로 크게 나눌 수 있다. 블록 정합 알고리즘은 각 블록 단위로 움직임을 찾아 그 움직임 벡터를 그 블록의 모든 화소에 적용하는 방법이며, 화소 순환 알고리즘은 각 화소 단위의 움직임 벡터를 이전 화소들의 데이터를 사용하여 계산한 후

적용하는 방법이다. 화소 순환 알고리즘에 비해 블록 정합 알고리즘이 수행 시간이 적게 소요되며 하드웨어 구현이 용이하기 때문에 대부분의 움직임 보상 부호화에 널리 이용되고 있다.

블록 정합 알고리즘은 현재 프레임의 탐색 블록과 탐색영역내의 블록들과의 유사도를 화소단위로 이동해 가며 구한 후 유사도가 가장 높은 블록, 즉 정합 에러(matching error)가 가장 작은 블록의 위치를 구한다. 이렇게 구한 블록과 현재 탐색 블록과의 위치 차이가 바로 움직임 벡터이다. 정합 에러를 계산하는 함수들로 주로 사용하는 블록 정합 기준들에는 평균 제곱 오차(Mean Squared Error)나 평균 절대 오차(Mean Absolute Error)등이 있다.

$$MSE(i,j)=\frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n}[I(x,y,t)-I(x+i,y+j,t-\tau)]^2 \qquad (1)$$

$$MAE(i,j)=\frac{1}{mn}\sum_{x=1}^{m}\sum_{y=1}^{n}|I(x,y,t)-I(x+i,y+j,t-\tau)| \qquad (2)$$

$$CCF(i,j)=\frac{\sum_{x=1}^{m}\sum_{y=1}^{n}I(x,y,t)\cdot I(x+i,y+j,t-\tau)}{[\sum_{x=1}^{m}\sum_{y=1}^{n}I^2(x,y,t)]^2\cdot[\sum_{x=1}^{m}\sum_{y=1}^{n}I^2(x+i,y+j,t-\tau)]^2} \qquad (3)$$

이러한 블록 정합 기준들을 써서 전체 탐색 영역내에서 가능한 모든 움직임을 탐색하는 방식이 전역 탐색 방법이다. 모든 움직임을 탐색하기 때문에 블록 정합 알고리즘으로 얻을 수 있는 가장 정확한 움직임을 찾을 수 있다. 전역 탐색 방법의 한 블록당 탐색 횟수는 프레임간의 최대 움직임을 p 라고 했을 때 $(2p+1)^2$이 되며 한 프레임에서는 (영상의 블록의 개수) × $(2p+1)^2$ 이 되므로 엄청난 양의 계산량이 필요하다. 이처럼 전역 탐색 방법은 정확한 움직임 벡터를 찾을 수 있지만 많은 계산량 때문에 실시간 시스템 구현에 문제점이 많다. 따라서 계산량을 줄이면서 정확한 움직임 벡터를 찾기 위한 여러 가지 고속 알고리즘들의 연구가 활발히 진행되고 있다[1]-[6].

Ⅲ. 제안하는 새로운 블록 정합 기준

3.1 기존의 1-비트 화소 분류 방법

전역 탐색 방법에서 블록 정합 기준으로 주로 사용하는 평균 제곱 오차(MSE)나 평균 절대 오차(MAE)는 정확한 움직임 벡터를 찾을 수 있다는 장점에도 불구하고 많은 계산량 때문에 실시간 시스템 구현에 어려움이 있다. 따라서 기존의 블록 정합 기준보다 적은 계산량과 간단한 하드웨어 구현을 가능하게 해주는 새로운 블록 정합 기준이 1-비트 화소 분류 방법이다[1]. 이 방법에서는 기존의 전역 탐색 방법과 마찬가지의 방법으로 탐색을 행한 후 정합 에러의 계산시 새로운 블록 정합 기준을 사용함으로써 효율적인 움직임 추정을 행한다. 제안된 방법에서는 먼저 현재 탐색 블록과 이전 프레임의 화소들의 차이값을 정합 화소(matching pel)와 비정합 화소(mismatshing pel)로 나눈다.

$$T(x,y,i,j)=1,\ \text{if}\ |I(x,y,t)-I(x+i,y+j,t-\tau)|\le t$$
$$=0,\ otherwise \qquad (4)$$

즉, 적당한 임계치(threshold)를 정한 후 그 화소에서의 차이값이 임계치보다 크면 정합 화소로, 임계치보다 작으면 비정합 화소로 결정하는 것이다. 그 후 정합 화소들의 개수의 합을 구해 그 합이 가장 많은 블록의 위치를 움직임 벡터로 결정한다. T(x, y, i, j)는 화소 차이값에 대한 이진 비용 함수이며, 정합 화소인지 비정합 화소인지에 따라 각각 1과 0값을 가진다.

$$G(i,j)=\sum_x\sum_y T(x,y,i,j) \qquad (5)$$
$$(\ i=0,\ \pm1\ \cdot\cdot\ \pm p\ \ \&\ \ j=0,\ \pm1\ \cdot\cdot\ \pm p\ )$$

G(i, j)의 값은 현재 블록과 x, y 방향으로 각각 i, j 만큼 이동된 이전 프레임의 블록간의 정합에서 정합 화소들의 개수이다.

$$G_m(\ d_h,\ d_v)\ =\ \max\{G(i,j)\} \qquad (6)$$

따라서 i와 j를 변화시켜가며 탐색해서 가장 큰 값을 갖는 G(i,j)에서의 i, j값이 움직임 벡터의 좌표가 된다.

3.2 제안하는 방법 1 : 2-비트 화소 분류법

기존의 1-비트 화소 분류 방법은 간단한 하드웨어 구조가 가능하다는 장점이 있지만 화질 면에서 기대한 만큼의 성능을 보여 주지 못하고 있다. 즉 평균 제곱 오차뿐만 아니라 평균 절대 오차 방법에 비해서도 상당히 떨어지는 성능을 보이고 있다. 이에 본 논문에서는 기존의 방법에서 쓰인 0과 1로 나누는 이진 비용 함수에서 변형된 2-비트, 3-비트 비용 함수를 이용한 새로운 블록 정합 기준을 제안한다. 그림 10 에서 보는 것과 같이 현재 탐색 블록과 이전 프레임의 화소들의 차이값들을 세 개의 임계치를 정해 분류한후 그 구간에 속하는 화소들의 대표값으로 그 구간의 임계치를 부여한다. 각각의 임계치를 T_1, T_2, T_3라 하면 각 화소의 대표값은 다음과 같다.

$$T(x,y,i,j)=0,\ \text{if}\ \ 0\le|I(x,y,t)-I(x+i,y+j,t-\tau)|\le T_1$$
$$T(x,y,i,j)=T_1,\ \text{if}\ \ T_1\le|I(x,y,t)-I(x+i,y+j,t-\tau)|\le T_2$$
$$T(x,y,i,j)=T_2,\ \text{if}\ \ T_2\le|I(x,y,t)-I(x+i,y+j,t-\tau)|\le T_3$$
$$T(x,y,i,j)=T_3,\ \text{if}\ \ T_3\le|I(x,y,t)-I(x+i,y+j,t-\tau)| \qquad (7)$$

그 후 정합 화소들의 대표값들의 합을 구해 그 합이 가장 작은 블록의 위치를 움직임 벡터로 결정한다.

3.3 제안하는 방법 2 : 3-비트 화소 분류법

3-비트 화소 분류 방법은 2-비트 화소 분류 방법과 마찬가지로 현재 탐색 블록과 이전 프레임의 화소들의 차이값들을 일곱 개의 임계치를 정해 분류한후 그 구간에 속하는 화소들의 대표값으로 그 구간의 임계치를

부여한 후 정합 화소들의 대표값들의 합을 구해 그 합이 가장 작은 블록의 위치를 움직임 벡터로 결정한다. 3-비트 화소 분류 방법은 2-비트 화소 분류 방법에 비해 하드웨어 구조의 복잡도는 다소 커지지만 화질면에서 좀더 나은 성능을 보여주고 있다. 그러나 더 이상의 구간으로 화소를 분류하는 것은 복잡도의 증가에 비해 화질의 향상이 그다지 효율적이지 못함을 실험을 통해 알 수 있었다.

3.4 제안하는 방법 3 : 2차 함수로 모델링한 비용 함수를 이용한 화소 분류 방법

제안하는 방법 1과 제안하는 방법 2는 평균 절대 오차(MAE)에서 사용하는 화소들의 차이값의 절대값을 사용한 방법인데 비해 제안하는 방법 3에서는 평균 제곱 오차(MSE)에서 사용하는 화소들의 차이값의 제곱을 이용해서 화소들을 분류하였다. 이 경우 계산량의 증가는 있지만 평균 제곱 오차에 근사화하는 비용 함수를 사용하는 만큼 화질에서, 특히 PSNR 측면에서는 월등한 성능을 보이고 있다.

3.5 임계치 결정

본 논문에서 제안하는 화소 분류 방법에 있어서 적절한 임계치를 결정하여 화소들을 잘 분류하는 것은 매우 중요한 요소이다. 이것은 곧 임계치의 결정이 복호된 화질에 미치는 영향이 매우 크다는 것을 의미하는 것이다. 본 논문에서는 임계치를 결정하는 방법으로 최적 양자화 레벨 결정 방법을 사용하였다. 임계치를 주어 화소들에 대표값을 부여하는 과정이 양자화 알고리즘과 유사하기 때문이다. 최적의 임계치를 찾기 위해 로이드-맥스(Lloyd-Max) 최적 양자화 알고리즘을 사용하였다.

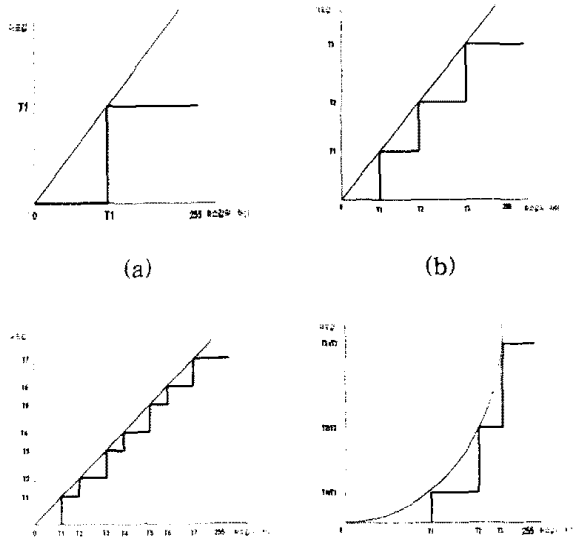

그림 1. 제안하는 새로운 블록 정합 기준 (a) 1-비트 화소 분류 (b) 2 -비트 화소 분류 (c) 3-비트 화소 분류 (d) 2차 함수를 이용한 화소 분류

Ⅳ. 실험 및 고찰

실험에 사용한 영상은 Football영상과 Flowergarden 영상, 그리고 Tabletennis영상이다. Football영상은 움직임이 크고 많이 발생하며, Flowergarden영상은 복잡한 텍스춰(texture)가 많은 영상이고, Tabletennis영상은 움직임이 비교적 적은 영상이다. 화면의 크기는 모두 352 × 240 크기의 SIF(Source Input Format)화면이며, 1프레임부터 40프레임까지의 화면을 대상으로 실험하였다. 탐색하는 블록의 크기는 16 × 16 매크로블록이며, 탐색 영역의 범위는 x, y 방면 모두 -16 ~ 15 화소 단위로 하였다. 그리고 여러 가지 알고리즘들의 객관적인 비교를 위해 화면 가장자리에서의 처리는 하지 않았다. 화질의 측정은 PSNR(Peek Signal to Noise Ratio)을 이용한 객관적 화질 평가와 눈으로 직접 화면을 관찰하는 주관적 화질 평가를 통해 하였다. 실험결과를 보면 2-비트 화소 분류 방법은 평균 절대 오차(MAE)를 이용한 전역 탐색 방법 보다 대체로 비슷하거나 평균적으로 높은 성능을 보였으며, 3-비트 화소 분류 방법은 이 보다 약간의 화질 향상을 가져왔다. 이에 비해 제안하는 방법 3, 즉 2차 비용함수를 이용한 화소 분류방법은 평균 절대 오차를 이용한 방법보다는 월등한 화질을 보였으며, 거의 평균 제곱 오차(MSE)를 이용한 전역 탐색 방법에 근접하는 성능을 보였다. 또 제안한 방법과 여러 고속 알고리즘 중 3단계 탐색 방법(Three-Step Search)[3]과 화소 부표본화 방법(Pixel Subsampling)[2]을 결합한 모의 실험을 행하였다. 이 결과에서는 움직임이 적은 Table tennis 영상이 다른 두 영상에 비해 좀 더 나은 성능을 보임을 알 수 있다. 복원된 영상을 통한 주관적 화질 평가에 있어서도 화소 분류를 이용한 탐색 방법이 MAE를 이용한 방법보다 좀 더 자연스러운 화면을 보이고 있음을 알 수 있다.

Ⅴ. 결론

본 논문에서는 움직임 추정 부호화 방법 중 블록 정합 알고리즘에 있어 블록 정합 에러를 구해 움직임 벡터를 구하는 새로운 블록 정합 기준을 제안하였다. 기존의 1-비트 화소 분류 방법에서 변형된 새로운 비용함수를 이용한 2-비트, 3-비트 화소 분류 방법과, 화소 분류시 사용되는 임계치를 결정하는 방법을 제안하였다. 또한 평균 제곱 오차에 근사화하는 모델링(modeling)인 2차 비용함수를 이용한 화소 분류 방법도 제안하였다. 이 새로운 블록 정합 기준은 기존의 블록 정합 기준인 평균 제곱 오차나 평균 절대 오차에 비해 각 화소들이 움직임 벡터를 결정하는데 중요한 역할을 함으로써 좋은 결과를 가져오게 하며, 간단한 하드웨어 구조와 계산량의 감소로 실시간 시스템의 구현을 용이하게 해주는 장점이 있다. 또한 기존의 다른 고속 움직임 추정 알고리즘과의 결합에서도 우수한 성능을 보여 여러 가지 다른 응용 분야에서도 널리 쓰일 수 있을 것이다. 또한 화소 분류에서 쓰이는 임계치는 장면 전환 검출에도 그대로 쓰일 수 있는 장점이 있다.향후과제로는 2차 비용함수를 이용한 화소 분류 방

법에서 좀더 적절한 임계치를 결정하는 방법을 찾고, 제안하는 방법의 부화소 단위 움직임 추정 기법에의 응용 등이 있겠다.

VI. 참고 문헌

[1] H.Gharavi and Mike Mills, "Blockmatching Motion Algorithms - New Results," IEEE Transactions on Circuits and Systems, vol. 37, no. 5, pp. 649-651, MAY 1990.

[2] Bede Liu and Andre Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," IEEE Transactions on Circuits and Systems for Video Technology, vol. 3, no. 2, pp. 148-157, April 1993.

[3] Y. Q. Shi and X. Xia, "A Thresholding Multiresolution Block Matching Algorithm," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 2, pp. 437-440, April 1997.

[4] Jianhua Lu and Ming L. Liou, "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation," IEEE Transactions on Circuits and Systems for video tecnology, vol. 7, no. 2, pp. 429-433, April 1997.

[5] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COM-29, no. 12, pp. 1799-1808, Dec. 1981.

[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., Nov. 29-Dec. 3, 1981, pp. G5.3.1-5.3.5.

[7] Frederic Dufaux and Fabrice Moscheni, "Motion Estimation Techniques for Digital TV : A Riview and a New Contribution," Proceedings of the IEEE, vol. 83, no. 6, pp. 858-876, June 1995.

Football 영상(1~40 프레임)	평균 PSNR(dB)
MSE	23.0565
MAE	22.8512
1-비트 화소 분류	22.3305
2-비트 화소 분류	22.8923
3-비트 화소 분류	22.9345
3단계 탐색 방법	22.4293
3단계 + 2-비트 화소 분류	22.3597
화소 부표본화 방법	22.8337
화소 부표본화 + 2-비트 화소 분류	22.8347

표 1. Football 영상의 평균 PSNR

Flowergarden 영상(1~40 프레임)	평균 PSNR(dB)
MSE	23.7794
MAE	23.6593
1-비트 화소 분류	23.3527
2-비트 화소 분류	23.6695
3-비트 화소 분류	23.7022
3단계 탐색 방법	22.2720
3단계 + 2-비트 화소 분류	22.0251
화소 부표본화 방법	23.6287
화소 부표본화 + 2-비트 화소 분류	23.6050

표 2. Flowergarden 영상의 평균 PSNR

Tabletennis 영상(1~40 프레임)	평균 PSNR(dB)
MSE	29.8459
MAE	29.6776
1-비트 화소 분류	29.2133
2-비트 화소 분류	29.6991
3-비트 화소 분류	29.7480
3단계 탐색 방법	28.6873
3단계 + 2-비트 화소 분류	28.6494
화소 부표본화 방법	29.6634
화소 부표본화 + 2-비트 화소 분류	29.5937

표 3. Tabletennis 영상의 평균 PSNR

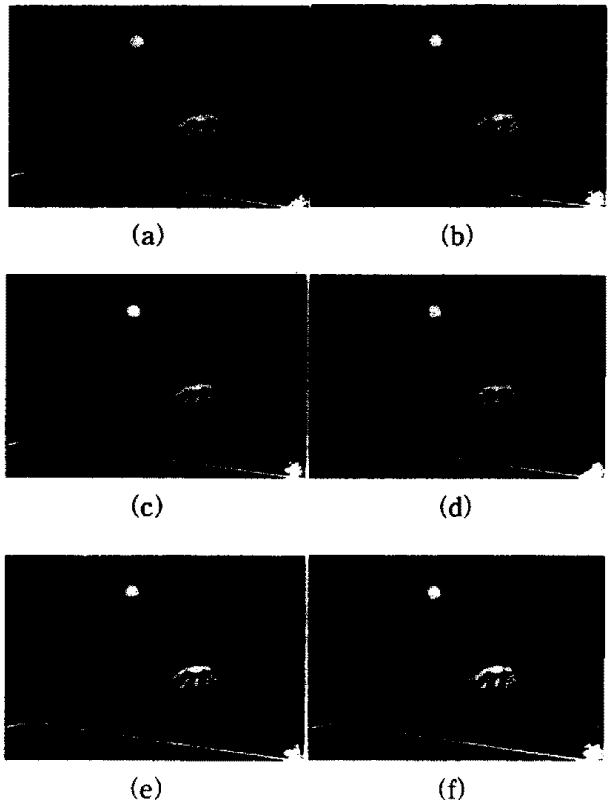

그림 2. Tabletennis 영상의 2번째 프레임 (a) 원 영상 (b) MSE (c) MAE (d) 1-비트 화소 분류 방법 (e) 2-비트 화소 분류 방법 (f) 3-비트 화소 분류 방법