

서브샘플링을 이용한 새로운 고속 움직임 예측 알고리즘

김철중^o 채병조 오승준 정광수
광운대학교 전자공학부
(redlions, bjchae, sjoh)@media.kwangwoon.ac.kr
kchung@daisy.kwangwoon.ac.kr

New Fast Motion Estimation Search With Subsampling Method

Chul-Jung Kim^o Byung-Jo Chae Seoung-Jun Oh and Kwang-Su Chung
School of Electronics Engineering, Kwangwoon University

요약

동영상을 효율적으로 압축하기 위한 움직임벡터 예측에 관한 많은 연구가 진행되어 왔다. 가장 일반적인 FBMA(Full search-based Block Matching Algorithm)는 화질은 좋지만 계산량이 많기 때문에 실시간 인코딩을 요구하는 시스템에서 사용하는데 문제가 있다. 좋은 화질을 유지하면서 인코딩 속도를 해결하기 위한 많은 알고리즘들이 제안되어 왔지만 ASIC이나 소형 시스템에서 사용할 수 있는 방법이 계속 요구되고 있다. 본 논문에서는 계산량을 더욱 줄여 속도향상을 가져오면서 FBMA와 비슷한 SNR을 유지하는 방법인 NMS(New Fast Motion Estimation Search With Subsampling Method)를 제안하였다. NMS는 서브샘플링한 값을 이용하여 SAD값을 구하고 또한 새로운 Search를 제안하여 기존 방법들이 제공하는 주관적 화질이나 PSNR을 높게 유지하면서도 속도를 10~15% 정도 개선시킬 수 있다.

1. 서론

영상압축은 디지털 비디오를 전송하거나 저장할 때 중요한 역할을 한다. 화상회의와 같은 방대한 정보를 실시간 서비스하기 위해 비디오 데이터를 압축하고 복원하는 국제 표준이 제정되고 있다. 동영상 압축 표준방식에서 움직임 보상방식(Motion Compensation)은 비디오 신호를 압축하는데 적합한 기술이며 움직임을 보상하기 위해 움직임 벡터(Motion Vector)를 예측한다. 움직임 벡터를 예측하는 방법 중 블록정합(Block Matching)방법이 가장 보편적인 방법이다[1]. BMA는 다른 방법들보다 계산량이 매우 낮은 것이 장점이다. BMA방식은 탐색점 중에서 BDM(Block Distortion Mode)이 최소인 점을 움직임벡터로 결정한다. BDM을 결정하는 방법은 식(1)과 같다[2].

$$MAE(i, j) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)| \quad (1)$$

식(1)에서 $C(x+k, y+l)$ 과 $R(x+i+k, y+j+l)$ 은 각각 현재 프레임과 참조 프레임의 화소들을 가리킨다. $[-p, p]$ 를 탐색 영역이라 하면 i 와 j 의 범위는 $-p \leq i, j \leq p$ 이다. N, M 은 블록의 가로와 세로 크기이다. 따라서 탐색영역 중 $MAE(i, j)$ 가 최소가 되는 점의 블록을 가장 정합이 잘된 블록으로 결정하고 이 때의 (i, j) 를 움직임벡터로 정한다.

2. 움직임벡터 예측방법

FBMA의 단점인 속도문제를 개선하기 위하여 고속 움

직임벡터 예측 방법들이 제안되었다. TSS(Three Step Search)방법[1]과 NTS(New Three Step Search)방법[5]은 널리 알려진 고속 BMA 방법들이다. 이들은 FBMA 방법에 비해 계산량이 적어서 속도면에서 효과적이지만 시각적 화질이 떨어진다. 한편 최근의 나은 Nearest Neighbour방법[7]은 속도와 PSNR측면에서 모두 우수한 성능을 나타내고 있다. 이러한 고속 알고리즘들은 속도와 화질 사이에 타협점을 찾아 적절한 방법을 적용시키는 것이 매우 중요하기 때문에 최근에 PSNR과 시각적 측면을 고려한 알고리즘이 많이 제안되었다[3].

기존의 일반적인 탐색점 수를 줄이는 방법으로는 화질이나 PSNR을 유지하면서 속도를 크게 개선하기 힘들다. 따라서 본 논문은 최적의 탐색점을 찾기 위하여 요구되는 계산량을 효과적으로 줄이기 위해 서브샘플링 방법으로 BDM을 구하고 이에 부합되는 탐색 알고리즘을 설계하여 화질을 유지하면서도 계산 속도를 향상시키는 방법을 제안한다.

3. 서브샘플링을 이용한 움직임벡터 예측

서브샘플링한 화소 개수에 따른 전체프레임의 평균 BDM 값을 계산하면 50개 이상의 화소를 취한 평균 BDM과 256개 화소 모두를 사용한 값과 거의 유사하다[4][8]. 따라서 본 논문은 참조논문[4][8]에서 제시된 패턴에서 0에서 47번째까지의 화소들을 48개 화소들을 취하여 BDM을 구한다. 서브샘플링하여 구한 BDM과 256개를 이용한 원래의 BDM은 공간적 유사성을 최대한 이용

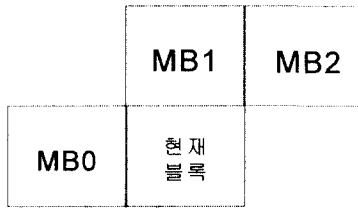


그림 1. 참조할 주위블록

한 것이므로 각각의 값을 비교했을 때 큰 차이를 보이지는 않지만 작은 차이를 보일 수 있다. 그 차이를 보완하기 위하여 4장에서와 같이 주위블록의 움직임벡터를 이용하는 방법을 사용할 수 있다.

4. 주변 움직임벡터를 이용한 예측

영상의 공간적 유사성 및 연속성을 이용하는 방법이다. 이미 구해진 주위 매크로블록들의 움직임벡터를 이용하여 현재 매크로블록의 움직임벡터를 구한다.

그림 1은 참조한 주위의 매크로블록을 나타내고 있다. MB0, MB1, MB2의 각각의 수평, 수직방향 움직임벡터들 각각의 중간값을 취하여 현재 매크로블록의 움직임벡터 예측에 사용하게 된다. 만약 주위의 매크로블록들의 움직임벡터가 존재하지 않는다면 다음 식(2)와 같은 방법으로 대체한다.

```

if(MB0 = INTRA & MB1 = INTER)
    MV.x = MB1.x, MV.y = MB1.y
else if(MB0 = INTER & MB1 = INTRA)
    MV.x = MB0.x, MV.y = MB0.y
else if (MB0 = INTER & MB1 = INTER & MB2 = INTRA)
    MV.x = (MB0.x+MB1.x)/2, MV.y = (MB0.y+MB1.y)/2
-- (2)
    
```

여기서 MV.x, MV.y는 수평수직방향의 움직임벡터 예측값이고 INTRA, INTER는 인코딩된 모드를 나타낸다.

5. NMS(New Fast Motion Estimation Search with SubSampling) 방법

```

Initial Step:
A = {(i,j) | i,j = d/2, -d/2, 0}
B(i,j) = {(i-1,j-1),(i-1,j+1),(i,j),(i+1,j-1),(i+1,j+1),(i-2,j-2),
          (i-2,j),(i-2,j+2),(i,j-2),(i,j+2),(i+2,j-2),(i+2,j),(i+1,j-2)}
C(i,j) = { (i-1,j),(i+1,j),(i,j-1),(i,j+1) }
M = 0, S = ∅, MinSad = 0, n=0, -d ≤ (i,j) ≤ d (d=7)

Step 1:
For all (i,j) ∈ A, find (i,j) such that SADsub is minimum.
MV = (i,j); MinSad = SADsub(i,j);
B(i,j) = B(MV);
Go to the Step 2

Step 2:
For all (i,j) ∈ B(i,j), find (i,j) such that SADsub is minimum.
    
```

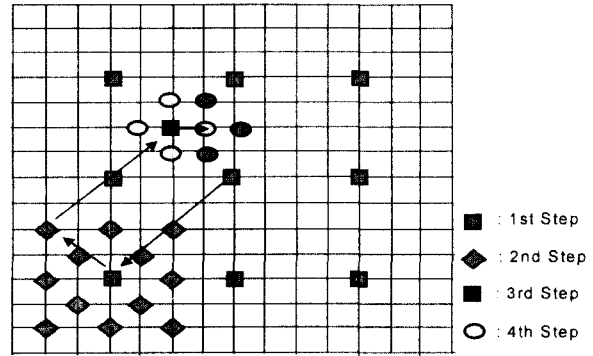


그림 2. NMS Search방법

```

If SADsub(i,j) < MinSad
    MinSad = SADsub(i,j); and MV = (i,j);
Go to the Step 3
Step 3:
Find previous 3 neighbour MV's median value (i,j)
Get SADsub of median value (i,j)
If SADsub(i,j) < MinSad
    MinSad = SADsub(i,j); and MV = (i,j);
C(i,j) = C(MV);
Go to the Step 4
Step 4:
For all (i,j) ∈ C(i,j), find (i,j) such that SADsub is minimum.
If SADsub(i,j) < MinSad
    MinSad = SADsub(i,j); and MV = (i,j);
C(i,j) = C(MV);
if(n=3) Go to the Final Step
else repeat Step 4 and n=n+1;
else
    Go to the Final Step
Final Step:
MV is Motion Vector (i,j)
    
```

4. 실험 결과

H.263+ 코덱에 움직임예측 모듈을 기존 방법과 NMS 방법으로 대체하여 각 방법들의 성능을 비교한다. 사용한 영상은 H.263과 MPEG에서 성능 검사를 위하여 가장 보편적으로 여러 테스트 영상을 사용하였다. 사용된 영상으로는 아키오(Akiyo), 세일즈맨(Salesman), 수지(Suzie), 홀(hall), 포어맨(foreman)등이다. 영상포맷은 QCIF이고 목표로 하는 비트율을 56 Kbps로 하여 150프레임을 부호화하였다. 주로 실시간 인코딩에 맞는 속도의 향상을 중심으로 했기 때문에 타겟비트율을 56Kbps로 했고 주요 비교영상도 헤드앤숄더(head and shoulder)영상이 아키오와 세일즈맨으로 하였다. NMS는 256개 화소 중에 48개를 샘플링을 하여 계산하였으므로 계산량이 이론적으로 FBMA에 비해 5%이하, TSS에 비해 50%이하이다. 실험 결과를 표1과 표2에 정

표 1. 아키오 영상 (QCIF, 150frame/56Kbps)

구 분	Full Search	TSS	NEAR	TSWS	NMS
부호화된 프레임	48	48	48	48	48
총 부호화 시간	1.47	0.27	0.29	0.27	0.26
PSNR(Y)	40.48	40.34	40.27	40.49	40.50

표 2. 세일즈맨 영상 (QCIF, 150frame/56Kbps)

구 분	Full Search	TSS	NEAR	TSWS	NMS
부호화된 프레임	48	48	48	48	48
총 부호화 시간	1.9	0.27	0.27	0.29	0.26
PSNR(Y)	35.03	34.94	34.98	35.07	35.05

리하였다. 이 표에서 알 수 있듯이 NMS가 가장 빠르며, 전체적인 속도 향상은 10%~15%이다.

PSNR측면에서도 NMS방식은 FS방식에 비해 크게 떨어지지 않음을 알 수 있다. TSS 방식에 비해 0.1~0.2dB 정도 개선되었고 FBMA과 비교해도 떨어지지 않음을 알 수 있다. 세일즈맨 영상을 네 가지 방식으로 각각 부호화한 후 다시 복호화한 결과를 그림 3에 보였다. 서브샘플링하여 NMS 영상의 시각적 화질이 기존의 알고리즘에 비해 떨어지지 않음을 알 수 있다. 세일즈맨 영상에서 프레임별로 PSNR 값을 비교한 것을 그림 4에 보였다.

6. 결론

본 논문에서는 디터링 패턴을 이용한 서브샘플링 방식과 주위의 움직임벡터를 이용한 탐색 방법인 NMS를 제안하였다. NMS 방법은 기존의 TSS 방식과 Nearest Neighbour 방식보다 속도가 10~15% 정도 향상되었다. PSNR 값을 기존 방법과 비교하였을 때 0.1dB~0.2dB 정도 높거나 거의 같은 수준을 유지하였다. 시각적 화질도 FMBA 방법을 통하여 얻은 결과와 유사하기 때문에 기존의 TSS 방식과 Nearest Neighbour 방식에 비해 실제적으로 유용한 방법이라고 할 수 있다. 그러므로 본 논문에서 제안한 NMS 방법을 ASIC화하여 향후 이동 멀티미디어 서비스에서 보편적으로 사용될 이동형 단말기의 비디오 코덱으로 활용할 수 있도록 하기 위한 연구가 진행되고 있다.

참고문헌

[1] B. Firth, J. Greeberg, and R. Westwater, *Motion Estimation Algorithms for Video Compression*, Florida Atlantic Univ.
 [2] A.M. Tekalp, *Digital Video Processing*, Upper Saddle River, New Jersey: Prentice Hall, 1995.
 [3] M. Bierling, "Displacement Estimation by Hierarchical Block Matching," *Proc. SPIE Visual Commun. Image Processing '88*, vol 1001, pp.942-951, 1988.
 [4] Fang-Hsuan Cheng and San-Nan Sun, "New Fast and Efficient Two-step Search Algorithm for Block Motion Estimation," *IEEE Trans. on Circuits Syst. Video Tech.*, vol

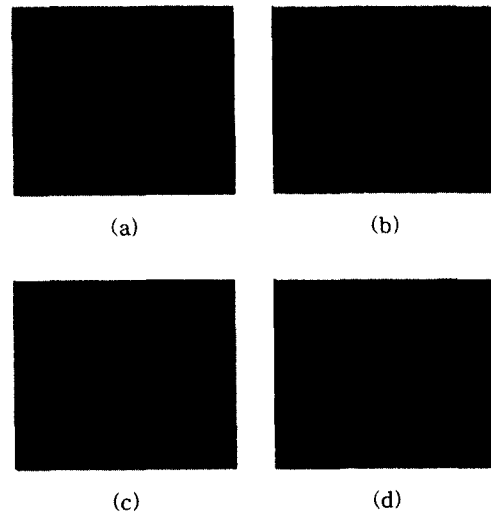


그림 3. QCIF 56kbps 세일즈맨 영상: (a) FBMA, (b) TSS, (c) NEAR, (d) NMS

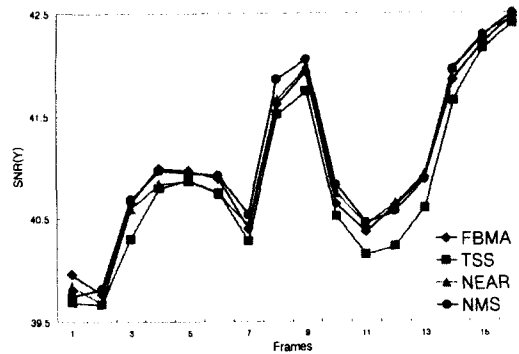


그림 4. Salesman(qcif) 프레임별 PSNR(Y)비교

9, no 7, pp.977-983, Oct. 1999.
 [5] R.Li, B.Zeng, and M. L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation," *IEEE. Trans. on Circuits Syst. Video Tech.*, vol 4, no 5 pp.438-442, Oct. 1994.
 [6] M. Bierling and R. Thoma, "Motion Compensation Field Interpolation Using a Hierarchically Structured Displacement Estimator", *Signal Process.*, vol 4, no 4, pp.387-405, 1986.
 [7] Michael G, Guy C. and Faouzi K., "An Efficient Computation-Constrained Block-based Motion Estimation Algorithm for Low Bit Rate Video", *IEEE. Trans. on Image Processing*, vol 8, no 12 pp.1816-1822, DEC. 1999.
 [8] 김철중, 채병조, 오승준, 정광수, "서브샘플링을 이용한 수정된 Two-Step 고속 움직임 예측 알고리즘", 한국정보과학회 춘계종합학술발표회, 2001, 4, pp. 459-462