

영상 비틀림을 이용한 움직임 추정 방법

임 동 근, 호 요 성
 광주과학기술원 정보통신공학과
 광주광역시 광산구 쌍암동 572번지
 dklim@god.kjist.ac.kr

Motion Estimation using Image Warping

Dong-Keun Lim and Yo-Sung Ho
 Dept. of Information and communications, K-JIST
 572 Ssangam-dong, Kwangsan-Gu, Kwangju, Korea

ABSTRACT

In low bit-rate video communication systems, insufficient bit allocation for coding produces blocking effects, having severe image degradations. In order to develop an effective motion compensation method for low bit-rate video coding, while reducing blocking effects, a more sophisticated transformation, called image warping, is generally attempted. In this paper, we propose an adaptive variable search range. The search range is determined by the PSNR of displaced frame difference. By eliminating unnecessary motion searches, we can reduce the computing time and increase the frame rate.

1. 서론

최근 64kbps 정도의 전송망을 이용한 초저속 영상 통신 방식의 구현에 많은 관심을 쏟고 있다. 기존의 H.261, H.263, MPEG 등에서 사용하고 있는 방법은 영상을 정규화된 사각형의 블록으로 쪼갠 후, 블록 정합 방법(BMA)을 이용한 움직임 추정과 움직임 보상 및 DCT 변환 방법을 함께 사용하는 것이다. 제한된 량의 정보만을 전송하는 초저속 영상 통신에서는, 이 방법을 사용하면 재생 화면의 블록 경계면이 불연속하게 되는 블록 현상(Blocking Effect)이 발생해서 화질의 저하를 가져온다.

이를 해결하기 위한 새로운 부호화 방법들이 제안되었다. 우선 영상을 삼각형의 블록으로 쪼갠 다음에, 영상 비틀림(Image Warping) 방법 또는 공간 변환(Spatial Transformation)이라고도 불리는 방법을 사용한다. 이 방법에서는 각 블록에 대하여 거친(Coarse) 움직임 추정(Motion Estimation)을 한 후, 블록간의 연결성(Connectivity)을 유지하기 위하여 반복적인 세밀한(Refined) 움직임 추정을 거치게 된다. 일반적으로 반복적인 세밀한 보정을 위해서는 Nakaya등이 제안한 Hexagonal Matching 방법[1][3]이 주로 사용되는데, 많

은 계산량이 필요하다.

영상 비틀림 방법을 사용할 때, 원하는 성능을 얻기 위한 반복의 횟수는 거친 움직임 추정의 정밀도에 많이 종속된다. 따라서 본 논문에서는 거친 움직임 추정의 정밀도를 높여서 반복의 횟수를 감소시키고 결국 계산량이 줄어드는 방법을 제안하였다. 즉, 세밀한 보정을 하기 전에 적용되는 거친 움직임 추정을 할 때, Mask의 크기와 탐색 영역의 크기가 어떠한 때 trade-off가 성립될 수 있음을 보인다. 그리고 능동적인 탐색 영역을 사용하였을 때, 계산량의 감소와 성능의 관계를 연구하였다.

2. 보간 함수의 적용

영상 비틀림에서는 영상을 충분히 잘 표현할 수 있는 적절한 위치의 꼭지점과 그것의 움직임 벡터를 찾는 방법이 중요하다. 또한 복호화할 때 그 움직임 벡터를 사용하여 삼각형 블록 내의 화소 값(Pixel)을 보상할 때 쓰이는 적절한 보간 함수(Interpolation Function)가 필요하다[2].

영상 비틀림은 식 (1)과 같은 변환을 적용한다.

$$\hat{f}_k(x, y) = I_{k-1}(x', y') \quad (1)$$

$$x' = f(x, y), \quad y' = g(x, y)$$

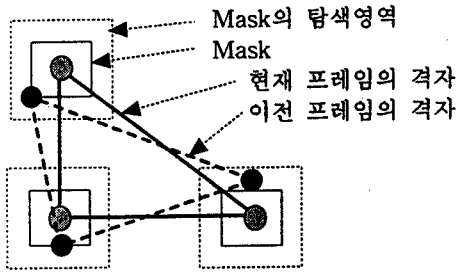
여기서 I_{k-1} 은 $(k-1)$ 번째 프레임 (이전 프레임)

\hat{f}_k 는 k 번째 프레임 (현재 프레임)

즉, 현재 프레임 (x, y) 위치의 예측 값은 이전 프레임 (x', y') 위치의 화소 값의 함수 형태로 나타내어지는데, 만약 실수 값을 가지게 되는 경우에는 쌍선형 보간(Bilinear Interpolation)을 사용하여 계산된다.

영상 비틀림에서 주로 사용하는 방법은 다음의 단계를 거친다. 먼저 영상을 삼각형으로 나눈 후 움직임 추정을 한다. 이를 위한 Mask의 모양은 <그림 1>과 같이 현재 프레임에 있는 삼각형의 각 꼭지점을 중심으로 구성된다. 이전 프레임의 탐색 영역에서 Mask를 이동시키면서 현재 프레임의 Mask와 이전 프레임의 탐색 영역의 Mask에 있는 화소들과의 오차 함수(Distortion Function)값을 구한다. 이 값이 최소가 되는 위치와 현

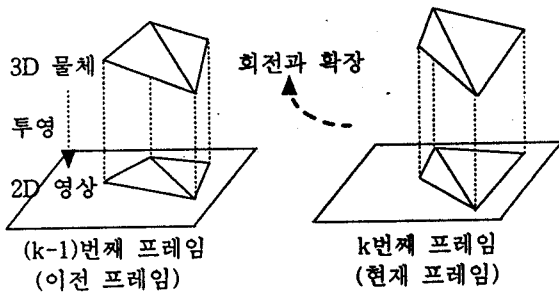
재 프레임의 꼭지점과의 변위를 움직임 벡터로 추정한다. <그림 1>에 보듯이 이전 프레임의 격자가 변형되는 역방향 예측법(Backward Prediction)을 사용하였다.



<그림 1> 적용된 Mask의 모양

다음에는 삼각형 내부의 값은 세 꼭지점의 움직임 벡터와 어파인 변환(Affine Transform)을 사용하여 움직임 벡터를 예측한 후 식 (1)과 같이 적용한다. <그림 2>를 보면 영상은 3차원 물체의 2차원 투영(projection)이다. 3차원 물체를 구성하는 삼각형 조각들의 윗쪽 방향으로의 회전과 확장 운동은 2차원 영상에서는 변형된 삼각형들로 투영된다. 여전히 삼각형의 형태를 띠며 격자 구조(structure topology)가 유지된다. 이것이 어파인 변환의 장점이다.

마지막으로 세밀한 보정을 반복적으로 적용한다.

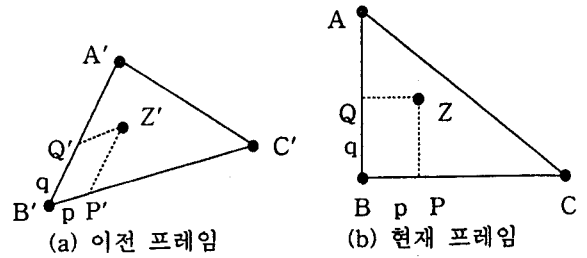


<그림 2> 3차원 삼각형조각의 2차원 투영

2-1. 기하학적인 변환(Geometric Transform)

기존의 정규화된 사각형 구조를 사용하던 BMA에서는 블럭의 평행 이동(displacement)에 대해서만 움직임 추정을 하는데 반하여 어파인 변환은 평행 이동, 회전(rotation), 확대 축소(zooming)를 모두 적용시키는 방법이다. 변환을 설명하기 위하여 주로 화소의 위치와 행렬과의 곱으로 움직임을 주로 표현하는 방법을 많이 사용하는데 어파인 변환은 식 (2)와 같이 정의된다.

$$\begin{aligned} x' &= a \cdot x + b \cdot y + c \\ y' &= d \cdot x + e \cdot y + f \\ T &= \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{bmatrix} \\ X &= [x \ y \ 1], \quad X' = [x' \ y' \ 1] \\ X' &= X \cdot T \end{aligned} \quad (2)$$



<그림 3> 영상 비틀림 전 후의 삼각형의 모양

따라서, 식 (2)의 변수 a 에서 f 까지를 구하기 위해서는 행렬 T 의 역행렬을 구해야 되는데 많은 시간이 걸리고 역행렬이 존재하지 않을 수도 있다.

2-2. 벡터 관계를 이용한 빠른 어파인 변환

식 (2)가 계산상의 단점을 가지므로 <그림 3>과 같은 관계를 일반적으로 사용한다.

여기서 $\overline{A'B'} \neq \overline{AB}$, $\overline{B'C'} \neq \overline{BC}$, $\overline{C'A'} \neq \overline{CA}$ 이고 $\overline{B'Q'} \neq \overline{BQ}$, $\overline{B'P'} \neq \overline{BP}$ 이지만

$$\begin{aligned} \frac{\overline{B'Q'}}{\overline{A'B'}} &= \frac{\overline{BQ}}{\overline{AB}} = q \leq 1, \\ \frac{\overline{B'P'}}{\overline{B'C'}} &= \frac{\overline{BP}}{\overline{BC}} = p \leq 1, \quad p+q \leq 1 \end{aligned} \quad (3)$$

의 관계가 성립한다. 이러한 성질을 이용하여 벡터 Z' 과 벡터 Z 의 관계식을 유추하면 식 (4)와 같다.

$$\begin{aligned} \vec{Z} &= \vec{B} + p\overline{BC} + q\overline{BA} = \vec{B} + p(\vec{C} - \vec{B}) + q(\vec{A} - \vec{B}) \\ \vec{Z} &= (1-p-q)\vec{B} + p\vec{C} + q\vec{A} \end{aligned} \quad (4)$$

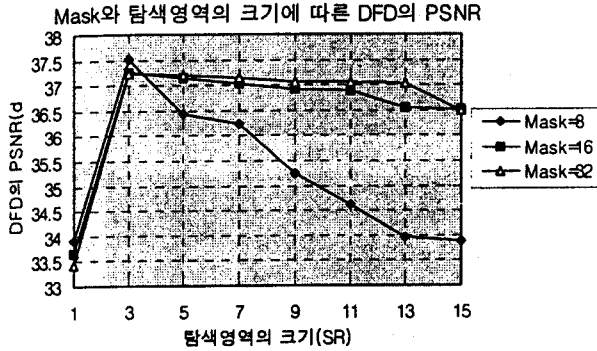
식 (2)를 적용하는 것과 식 (4)를 적용하는 것은 전적으로 동일하며 계산량은 식 (4)를 적용하는 것이 유리하다. 그런데 정규화되지 않은 도형에 대하여는 식 (4)와 같은 접근이 어려울 때도 있다.

3. Mask 크기와 탐색 영역 결정 실험

기존의 BMA에서는 Mask의 크기가 작고 탐색 영역이 클수록 좋은 결과를 가져온다. 그런데 탐색 영역은 어느 정도의 크기가 되면 성능은 거의 포화된다. 마찬가지로 영상 비틀림에 있어서 거친 움직임 추정도 비슷한 경향을 띄는데 BMA에서와 달리 최고 성능을 나타내는 영역보다 커지면 오히려 성능이 나빠지며, Mask의 크기도 작을 때가 더 유리할 때도 있고 클 때가 유리할 때도 있다. 이러한 경향은 움직임 추정을 하는데 사용하는 이전 프레임과 현재 프레임 간의 차이(DFD: Displaced Frame Difference)를 이용하여 구한 PSNR 값에 따라서 다른 특성을 가진다.

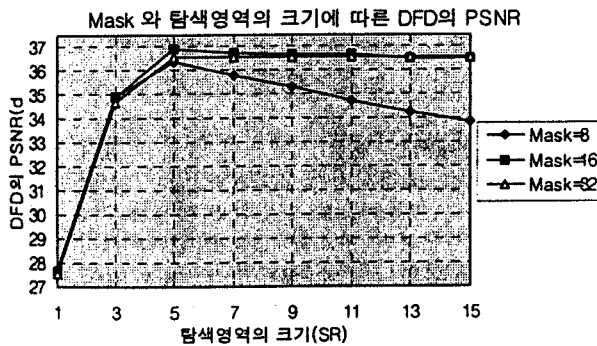
먼저 거친 움직임 추정을 할 때, 탐색 영역이 어느 정도에 도달하면 성능이 최고조에 이르렀다가 탐색 영역이 더 커지면 오히려 떨어짐을 보여주는 <그림 4>를

보자. 탐색 영역이 ±3정도일 때 충분한 성능을 나타낸다. 반면에 <그림 5>를 보면 탐색 영역이 더 커야 성능이 최고조에 이른다. 즉, DFD의 PSNR이 크면 탐색 영역이 작은 쪽이 좋고 반대의 경우는 탐색 영역이 큰 쪽이 좋다.



<그림 4> Miss America #1과 #4(DFD=31.230dB)의 Mask의 크기와 탐색 영역(SR)과의 관계

<그림 4>와 <그림 5>에서 보듯이 DFD에 따라 적절한 크기의 Mask를 사용하고 그에 따른 적합한 탐색 영역을 잡는 것이 중요하다.

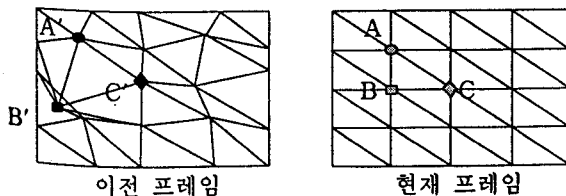


<그림 5> Miss America #70과 #73(DFD=25.593dB)의 Mask의 크기와 탐색 영역(SR)과의 관계

일반적으로 탐색 영역은 Mask 크기의 1/2이상보다 크지 않도록 하는 것이 적합하다. 그 이유는 어떤 영상의 경우는 Mask 크기의 1/2보다 큰 탐색 영역을 잡을 때, <그림 6>과 같이 격자 구조가 깨질 수 있다.

꼭지점 A와 C는 격자 구조를 깨뜨리지 않지만 꼭지점 B는 격자 구조를 깨뜨린다. 따라서, 이러한 경우가 발생하지 않도록 탐색 영역의 크기를 잘 잡아야 한다.

영상을 구성하는 인접한 삼각형들의 꼭지점들의 움직임 벡터가 Mask의 1/2크기보다 크더라도 비슷한 방향으로 함께 이동할 경우는 앞의 조건에 위배되더라도



<그림 6> 격자 구조의 깨어짐을 보여주는 예

더 좋은 성능을 보여줄 수도 있다. 이것은 영상마다 움직임의 형태가 제각기 다르기 나타나기 때문에 정확히 추정하기는 어렵다. 따라서, 본 논문에서는 Mask의 크기보다 탐색 영역이 큰 경우에 대해서는 고려하지 않았다.

4. 영상 시퀀스의 실험 결과 및 고찰

본 논문에서는 여러 가지 블럭 및 Mask 크기로 나누기에 용이한 352x288 크기를 가진 CIF(Common Intermediate Format)의 Miss America (Missa)영상을 3 프레임 간격(#1, #4, #7, ...)으로 뽑아서 10Hz의 프레임률을 가지는 영상 30 프레임을 가지고 실험하였다. 성능 개선을 위해 능동적으로 Mask의 크기를 가변시킬 수도 있지만, 성능 비교를 위하여 16x16 크기로 고정시켰다. 주로 실험한 내용은 거친 움직임 추정을 할 때, 탐색 영역을 적절하게 잡아 쓸모없이 성능을 떨어뜨리고 세밀한 보정을 할 때도 계산량을 증가시키는 요인을 제거하였다.

적절한 탐색 영역을 찾기 위하여 고려한 실험 영상은 <표 1>과 같다. 그 이유는 Missa 영상 시퀀스 중에는 DFD의 PSNR이 아주 작은 경우에 대한 것이 별로 없기 때문이다. <그림 7>은 <표 1>의 내용을 그래프로 그린 것인데, PSNR이 낮은 영상에서의 탐색 영역의 크기는 급격하게 증가함을 볼 수 있다.

성능 검사를 위해 사용한 PSNR은 MxN크기의 영상에 대해서 다음 식 (5)와 같이 적용하였다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ dB} \quad (5)$$

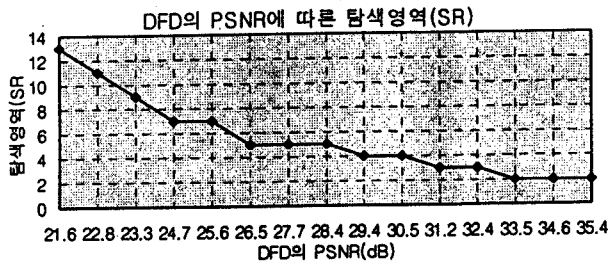
$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I_k(x, y) - I_{k-1}(x, y))^2$$

실험에 사용한 영상은 움직임이 많지 않기 때문에 영상 통신에서 적용될 수 있는데, Missa 영상같은 경우 PSNR이 32dB정도 이상만 되더라도 인간 시각이 거슬림을 느끼지 못할 정도의 성능을 보장할 수 있다.

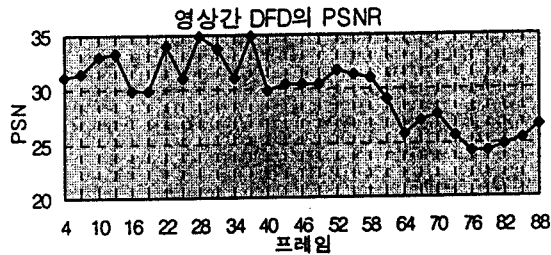
<그림 9>는 <그림 8>과 같은 영상 시퀀스에 <그림 10>의 능동적인 탐색 영역을 적용해서 얻은 성능과 탐색 영역을 ±7로 고정시켰을 경우를 서로 비교하고 있다.

<표 1> DFD의 PSNR에 따른 탐색영역(SR)

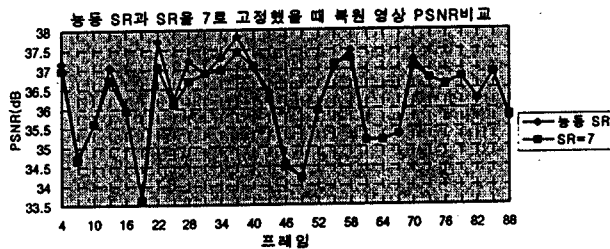
사용한 영상	Missa #76, #82	Missa #70, #75	Missa #82, #87	Missa #78, #81	Missa #70, #73
DFD PSNR	21.59594	22.81126	23.33157	24.66922	25.59304
SR	13	11	9	7	7
복원된 PSNR	35.62206	35.27937	35.71101	36.63558	36.92213
사용한 영상	Missa #0, #7	Missa #1, #9	Missa #1, #4	Missa #0, #2	Missa #1, #2
DFD PSNR	27.69488	29.36081	31.23030	33.53621	35.38383
SR	5	4	3	2	2
복원된 PSNR	33.54558	34.67771	37.24159	37.88290	38.42940



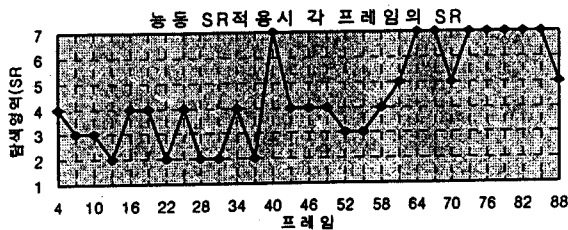
<그림 7> 실험 결과인 능동적인 탐색 영역



<그림 8> 실험 영상의 DFD의 PSNR



<그림 9> 능동 탐색 영역과의 성능 비교

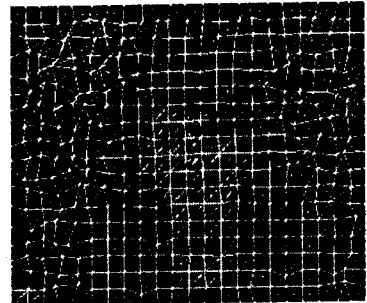


<그림 10> 제안된 방법을 사용할 때 탐색 영역

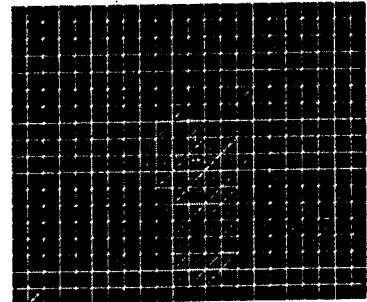
이제 <그림 10>의 탐색 영역에 따른 성능을 계산하자. 먼저 프레임당 오차함수의 계산량을 보면, Mask의 크기를 BS, 탐색 영역을 SR이라 할 때, $BS^2 \cdot (2SR+1)^2 \cdot (352/BS-1) \cdot (288/BS-1) + 352 \cdot 288$ 이고 각 프레임당 전송되어야 할 bit수는 "헤더+(352/BS-1)·(288/BS-1)·2·할당bit"이다. 이 결과를 정리하면 <표 2>와 같다. 계산량과 전송 데이터량의 관점에서 볼 때 모두 유리하다. 삼각형의 격자 구조를 이용한 영상 비틀림을 적용하였으므로, 이전과 현재 프레임의 격자 구조는 <그림 11>과 같이 변형된다.

< 표2> 계산량과 전송 bit수의 비교

	능동 SR적용	SR=7로 고정
오차함수 계산량	296,582,400	596,332,800
전송 bit수	12,879	14,300



(a) 이전 프레임의 격자구조의 모양



(b) 현재 프레임의 격자구조의 모양

<그림 11> 역방향 예측에서의 격자구조의 모양

5. 결론

본 논문에서는 영상 시퀀스의 움직임 탐색 영역을 동일한 크기로 고정시키지 않고 영상간의 차이(DFD)의 PSNR에 따라 능동적으로 가변하는 방법을 제안하였다. 이 방법을 사용하면 부호화 성능을 떨어뜨리지 않으면서 움직임 추정 시간을 절약할 수 있다. 또한 압축된 데이터량도 줄일 수 있기 때문에 지금까지 사용하던 방법에 이러한 점을 고려한다면 초저속 전송로를 통한 영상 통신에서 더 높은 프레임률을 얻을 수 있다.

REFERENCES

- [1] Y. Nakaya, H. Harashima, "Motion Estimation Based on Spatial Transformations," IEEE Trans. on Ckt. and Sys., vol. 4, pp. 339-356, June 1994.
- [2] G. Wolberg, Digital Image Warping, IEEE Computer Society Press, 1990.
- [3] Y. Nakaya, H. Harashima, "An Iterative Motion Estimation Method using Triangular Patches for Motion Compensation," SPIE '91, vol. 1605, pp. 546-557, 1991.