

Quad-tree 기법을 이용한 움직임 추정과 프레임간 내삽

이 기 동^o, 최 종 수
 중앙대학교 전자공학과

Motion Estimation and Interframe Interpolation Using Quad-tree Method

Ki-Dong Lee, and Jong-Soo Choi

Department of Electronic Engineering, Chung Ang University

요 약

움직임 보상형 프레임간 내삽기법은 다른 동영상 압축기법과 함께 사용함으로써, 그 압축효과를 더욱 향상시킬 수 있는 장점을 가지고 있다. 그러나 블럭에 기반한 기존의 움직임 보상형 프레임간 내삽기법은 움직이는 물체의 경계에서 심한 블럭효과를 보인다. 본 논문에서는 Quad-tree 기법을 이용해 화소단위의 움직임을 추정하고, 이를 이용해 드러난 영역과 가리워지는 영역을 정확히 예측해 보상함으로써 움직이는 물체의 경계에서 나타나는 블럭효과를 제거했다. 그리고 제안된 방법에서는 순방향이나 역방향 움직임추정 시 송신측으로 부터 보내지는 움직임 벡터를 이용함으로써 움직임 추정을 위해 소요되는 계산량을 최소화 하였다.

1. 서론

움직임 보상형 프레임간 내삽이란 그림 1에서와 같이 이미 알고있는 두장의 프레임 R_1 , R_2 사이에 하나 이상의 프레임을 삽입하는 기법으로 동영상 압축이나 프레임율 변환 또는 느린화면을 재생하기 위한 목적으로 사용될 수 있다[1]. 동영상 압축의 경우, 송신측에서는 $N+1$ 개의 프레임중 하나의 프레임만을 보내고, 수신측에서 건너뛴 프레임을 내삽하기 때문에 기존의 압축기법과 함께 사용함으로써 압축효과를 더욱 향상시킬 수 있다[2].

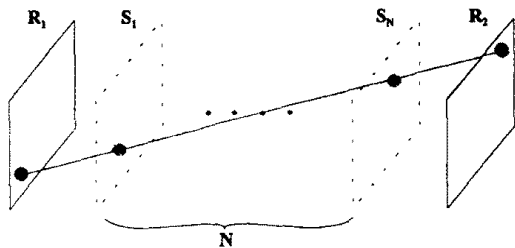


그림 1. 프레임간 내삽

그러나 블럭에 기반한 기존의 움직임 보상형 프레임간 내삽기법은 블럭 단위로 움직임을 추정하고, 동일 블럭내에 있는 화소들이 동일한 움직임을 갖는다는 가정 아래 건너뛴 프레임을 내삽한다. 그렇기 때문에 움직이는 물체

의 경계에서 나타나는 드러난 배경과 가리워지는 배경에 대한 보상이 어렵고 이로인해 심한 블럭효과가 나타난다 [3]. 이러한 블럭효과를 제거하기위해 Chi-Kong Wong 와 Oscar C. Au 는 세장의 프레임을 이용해 내삽될 프레임 내의 각 화소들을 움직이는 물체, 드러난 배경, 가리워진 배경, 정적인 배경등으로 분류하여 서로 다르게 보상해 주는 방법을 사용하기도 했다[4]. 그러나 이러한 방법으로는 각 화소들에 대한 정확한 분류가 어렵기 때문에 블럭효과가 크게 감소하지 않는다.

따라서 본 논문에서는 블럭효과를 제거하기 위해 송신측으로부터 보내진 블럭단위의 움직임 벡터에 Quad-tree 기법을 적용해 화소단위로 움직임을 추정하고, 이를 이용해 건너뛴 프레임을 내삽하는 방법을 제안한다. 송신측으로부터 받은 움직임벡터는 보통 역방향 움직임 추정에 의해 얻어진 것이다. 일반적으로 역방향으로 움직임을 추정한 경우, 가리워지는 배경에 대한 예측은 쉬운반면 드러나는 배경에 대한 예측은 어려워진다. 그렇기 때문에 가리워지는 배경과 드러난 배경을 모두 알아내기 위해서는, 수신측에서 별도의 순방향 움직임 추정을 해야하기 때문에 지나치게 많은 계산량의 증가를 가져온다. 따라서 본 논문에서는 순방향 움직임 추정시 요구되는 계산량을 감소시키기 위해, 송신측으로 부터 받은 역방향 움직임 벡터를 이용해 순방향 움직임 벡터를 추정하는 방법을 제안

한다.

그림 2는 제안한 방법을 블럭도로 나타낸 것이다. 수신측에서는 먼저 송신측으로부터 받은 블럭단위 역방향 움직임벡터를 이용해 블럭단위 순방향 움직임벡터를 구한다. 그리고 이들 블럭단위 움직임벡터에 Quad-tree 기법을 적용해 화소단위의 움직임을 추정하고, 이를 이용해 건너편 프레임에 내삽한다.

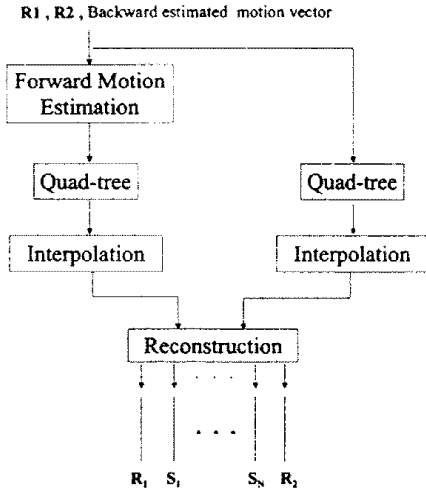


그림 2. 제안한 방법의 블럭도

2 장에서는 역방향 움직임벡터로부터 순방향 움직임 벡터를 얻어내는 방법과 Quad-tree 기법을 이용해 화소단위의 움직임을 추정하는 방법에 대해 알아본다. 3 장에서는 화소단위 움직임벡터를 이용해 건너편 프레임을 내삽하는 방법을 설명하고, 4 장에서는 모의실험 결과를 보인다. 그리고 마지막으로 6 장에서 결론을 맺는다.

2. 움직임 추정

2.1 순방향 움직임벡터 추정

건너편 프레임을 수신측에서 효과적으로 내삽하기 위해서는, 각 화소들을 움직이는 물체, 정적인 배경, 드러난 배경, 가리워진 배경 등으로 분류하고, 각각의 경우에 따라 다르게 처리해 주어야한다[5]. 그러나 일반적으로 동영상 부호화 기법에서 사용하고 있는 역방향으로 추정된 움직임 벡터만을 이용하게 될 경우, 가리워지는 배경에 대한 예측은 가능하지만 드러난 배경에 대한 예측이 어렵게 된다. 따라서 드러난 배경을 예측하기 위해서는 수신측에서 부가적으로 순방향 움직임을 추정해야하기 때문에 그만큼 많은 계산을 필요로하게 된다. 따라서 본 논문에서는 역방향 움직임 벡터를 이용해 적은량의 계산으로 순방향 움직임 벡터를 알아내는 방법을 제안한다.

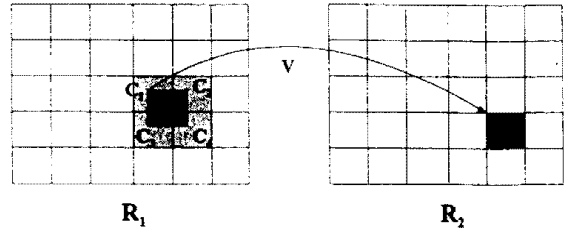


그림 3. 순방향 움직임 추정

그림 3에서 보는 바와 같이 먼저 R_1 과 R_2 프레임에 일정한 크기의 블럭으로 나눈다. R_2 프레임을 예측할 때, R_2 프레임 내의 B_2 블럭은 V 라는 역방향 움직임벡터를 이용해 R_1 프레임 내에 있는 B_1 영역을 참조하게 된다. 이때 B_1 영역이 R_1 프레임 내에서 걸치게 되는 블럭, 즉 C_1, C_2, C_3, C_4 블럭에 역방향 움직임벡터와 부호가 반대인 $-V$ 를 할당한다. R_2 프레임 내의 모든 블럭에 대해 이와 같이 적용하면 R_1 내의 각블럭에는 여러개의 움직임벡터들이 할당되게 되는데, 이들 후보벡터들 중에서 DFD (Displaced frame difference)가 가장작은 것을 해당블럭의 순방향 움직임 벡터로 선택한다.

2.2 Quad-tree 기법을 이용한 화소단위 움직임 추정

블럭에 기반한 기존의 움직임 보상형 프레임간 내삽기법에서는 블럭내의 모든 화소들이 동일한 움직임을 갖는 것으로 간주하고 이를 보상해주기 때문에 블럭효과가 심하게 나타난다[3]. 이는 동일블럭 안에 하나 이상의 서로 다른 움직임이 존재할 수 있음에도 불구하고 이를 고려하지 않았기 때문에 발생하는 문제이다.

따라서 본 절에서는 송신측으로부터 받은 블럭단위의 역방향 움직임 벡터와 앞 절에서 얻은 블럭단위의 순방향 움직임 벡터에 Quad-tree 기법을 적용해 화소단위의 움직임을 추정하는 기법을 제안하고자 한다.

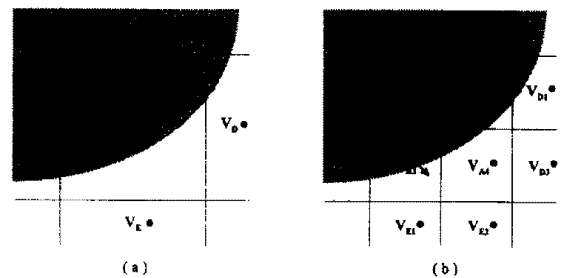


그림 4. Quad-tree 방법

그림 4에서와 같이 동일 블럭 안에 하나 이상의 서로 다른 움직임이 존재할 수 있다. 만약 움직이는 물체가 2차원 평면운동만 한다고 가정한다면, 동일 블럭내의 움직임들은 인접블럭의 움직임으로부터 추정할 수 있다. 따

라서 하나의 블럭 A를 4개의 하위블럭 A_1, A_2, A_3, A_4 로 나누면, 이들 하위블럭의 움직임은 이들을 포함하고 있는 상위블럭 A와 4근방에 위치한 인접상위블럭 B, C, D, E의 움직임 V_A, V_B, V_C, V_D, V_E 로부터 추정해 낼 수 있다. 다시 A_1, A_2, A_3, A_4 등의 블럭을 상위 블럭으로 하여 이와 같은 과정을 반복하면 블럭단위의 움직임으로부터 화소단위의 움직임을 추정해 낼 수 있다. 일반적인 블럭정합 방법에서는 블럭의 크기가 작아질수록 오정합을 하게될 확률이 높아지지만, 제안한 방법에서는 그 탐색 대상을 4근방에 위치한 인접상위블럭의 움직임 벡터로 제한하기 때문에 그러한 문제는 크게 발생하지 않는다.

3. 프레임간 내삽

지금까지 우리는 순방향과 역방향의 화소단위 움직임 벡터를 구했다. 이를 이용해 건너편 프레임을 내삽할 수 있는데 그림 5에서 처럼 역방향 움직임 벡터를 이용해 건너편 영상을 내삽하면, 가리워지는 배경은 잘 표현되는 반면 드러난 배경은 오정합에 의해 엉뚱한 값으로 채워진다. 마찬가지로 순방향 움직임 벡터를 이용해 건너편 프레임을 재구성하면, 드러난 배경이 잘 나타나고 가리워지는 배경은 오정합에 의해 엉뚱한 값으로 채워진다.

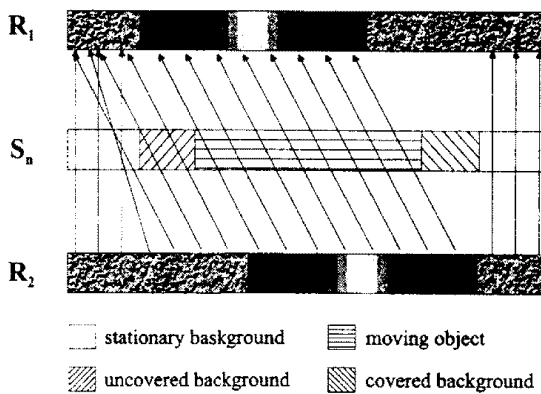


그림 5. 역방향 움직임벡터를 이용한 내삽

그림 6의 (a)와 (b)는 합성영상을 이용해 실험한 결과이다. (a)는 역방향 움직임벡터에 의해 얻어진 영상이고, (b)는 순방향 움직임벡터에 의해 얻어진 영상이다. 따라서 이 두장의 영상과 R_1, R_2 프레임을 이용하면 건너편 프레임을 효과적으로 내삽할 수 있다. 먼저 움직이는 물체와 정적인 배경부분을 (a), (b) 두영상의 평균값으로 채우면 그림 6의 (c)와 같이 된다. 가리워지는 배경부분은 R_1 의 같은위치에 있는 화소들의 밝기값으로 채우고 드러난 배경부분은 R_2 의 같은 위치에 있는 화소들의 밝기 값으로 채우면 그림 6의 S와 같이 건너편 프레임을 내삽할 수 있다. 이는 J.Ribas-Corbera 와 J.Sklansky 가 사용한 방법[6]

으로 배경이 정지해 있다고 가정할 때, 드러난 배경과 가리워지는 배경을 잘 보상한다. 이러한 과정을 거친 후에도 채워지지 않은 부분은 중간값필터를 이용해 채운다.

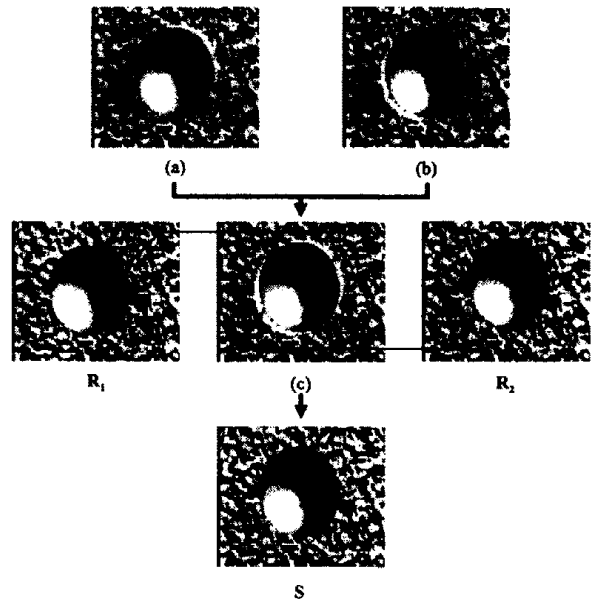


그림 6. 제안한 방법을 이용한 내삽과정

4. 모의 실험 결과

본 절에서는 352 X 288 크기의 "Salesman"영상을 이용해 모의실험을 하였다. 비교대상으로는 Chi-Kong Wong 와 Oscar C. Au 에 의해 제안된 MCTI(motion compensated temporal interpolation)방법과 MMCTI(modified motion compensated temporal interpolation)방법을 사용하였다. MCTI는 블럭에 기반하여 움직임을 보상하는 방법이고, MMCTI는 MCTI에서 생기는 블럭효과를 감소 시키기위해 세장의 프레임을 이용해 내삽하고자 하는 프레임의 화소들을 움직이는 물체, 정적인 배경, 드러난 배경, 가리워진 배경 등으로 분류하여 처리하는 방법이다.

그림 7은 Salesman 영상의 9번째와 11번째 프레임을 이용해 10번째 프레임을 재구성한 것이다. 손의 움직임이 많은 영상으로 제안한 방법에 의해 내삽된 영상이 MCTI나 MMCTI 방법의해 내삽된 영상에 비해 오른손과 왼손부근에서 블럭효과가 덜 나타나는 것을 볼 수 있다. 그림 8은 $N = 1$ 일때, 즉 한장의 프레임을 내삽했을 때, 내삽한 프레임의 PSNR를 그래프로 나타낸 것이다. 프레임간 움직임이 많은 구간, 즉 드러나는 영역과 가리워지는 영역이 많은 10번과 60번 프레임 부근에서 제안한 방법의 성능이 우수하게 나타나는 것을 볼 수 있다.

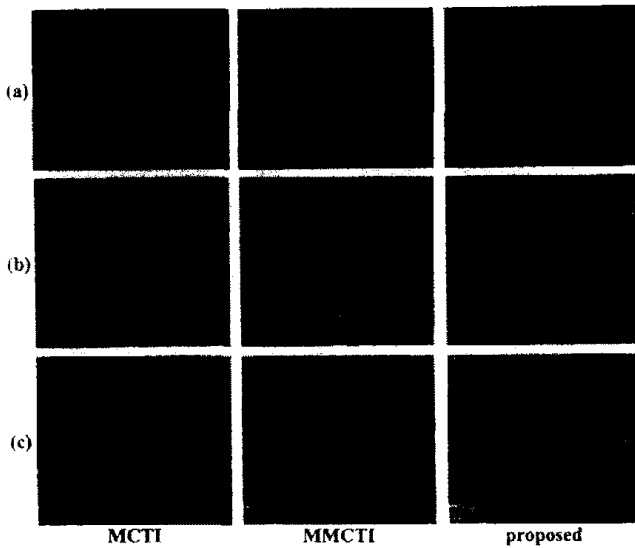


그림 7. 내삽영상(Salesman 의 10 번 프레임)

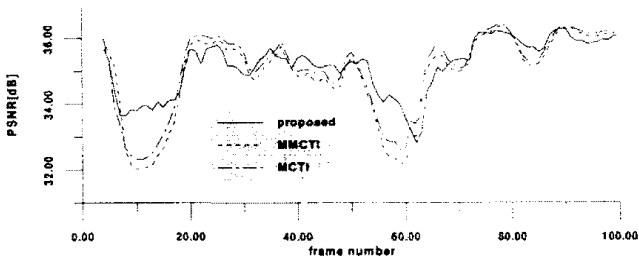


그림 8. PSNR 비교 (Salesman, N = 1)

5. 결론

본 논문에서는, 기존의 블럭기반 움직임 보상형 프레임 간 내삽기법에서 나타나게 되는 블럭효과를 감소시키기 위해 Quad-tree 기법을 이용해 화소단위의 움직임벡터를 구했으며, 드러나는 배경과 가리워지는 배경을 보다 정확히 예측하기 위해 역방향과 순방향움직임 벡터를 모두 사용하였다. 모의 실험을 통해 제안한 방법으로 내삽한 영상이 기존의 방법으로 내삽한 영상에 비해 블럭효과가 줄어든것을 볼 수 있었다.

- [1] Chung-Lin Huang and Teh-Tzong Choa, "Motion-compensated interpolation for scan rate up-conversion," OPTICAL ENGINEERING, vol. 35, no. 1, pp. 166-176, Jan. 1996.
- [2] Ciro Cafforio, Fabio Rocca, and Stefano Tubaro, "Motion Compensated Image Interpolation," IEEE Trans. Commun., vol. 38, no. 2, pp. 215-222, Feb. 1990.
- [3] Chi-Kong Wong and Oscar C. Au, "Fast Motion Compensated Temporal Interpolation for Video," Proc. SPIE Visual Comm. and Image Proc., vol. 2501, pp. 1108-1118, 1995.
- [4] Chi-Kong Wong and Oscar C. Au "Modified Motion Compensated Temporal Frame Interpolation for Very Low Bit Rate Video," Proc. IEEE ICASSP, vol. 6, pp. 2327-2330, 1996.
- [5] Peter Csillag and Lilla Boroczky, "Estimation of Accelerated Motion for Motion-compensated Frame Interpolation," Proc. SPIE Visual Comm. and Image Proc., vol. 2727, pp. 604-614, 1996.
- [6] J.Ribas-Corbera and J.Sklansky, "INterframe Interpolation of Cinematic Sequences," J. Visual Communication and Image Representation, vol. 4, no. 4, pp. 392-406, Dec. 1993.