

가변 스킵율 기반의 프레임간 보간기법

정회원 김 동 욱*, 최 연 성**

Interframe Interpolation Technique based on Variable Skip Rate

Dong-wook Kim*, Yeon-sung Choi** *Regular Members*

요 약

본 논문에서는 비디오 시퀀스의 가변 스킵율을 기반으로 한 새로운 비디오 보간기법을 제시한다. 제시된 기법에서 임의의 입력 프레임에 대한 스킵 여부는 움직임 복잡도 (motion complexity)의 정도에 의해 결정된다. 이 때 낮은 움직임 복잡도를 가질 때 프레임은 스킵되며 그렇지 않으면 부호화되고 전송된다. 이러한 움직임 복잡도를 결정하기 위해 프레임의 움직임 에지(MEF: Moving Edge in Frame) 이용한 새로운 기법이 도입된다. 수신단에서 스킵된 프레임에 대한 효과적인 보간을 위해서 움직임 필드 분할 기법이 적용되는데, 움직임 필드는 수리형태학적 필터링 및 영역별 움직임 벡터 합당 기법을 이용하여 분할된다. 구성된 움직임 필드는 각각의 움직임 특성을 바탕으로 영역별로 분류되며 이를 바탕으로 프레임간 보간을 행한다. 제시된 기법 및 기존의 기법에 대해 컴퓨터 모의 실험을 행한 결과, 동일한 전송 비트율을 가질 때, 제시된 기법은 기존의 방법에 비하여 향상된 보간 화질을 보이며, 화질의 변화를 줄일 수 있음을 확인할 수 있다.

ABSTRACT

A new video interpolation technique based on variable skip rate of video sequence is proposed in this paper. In the proposed technique, The determination whether a frame is skipped or not is done by the degree of motion complexity of the frame. If the motion complexity of a frame is low the frame is skipped, otherwise it is coded and transmitted. To determine the motion complexity of a frame a new technique using MEF (Moving Edge in Frame), the set of pixels considered as moving edges in a frame, is introduced. In the course of decoding and interpolating of receiver, the motion field is segmented. For the purpose of dividing vector field morphological filtering is applied. Morphological filtering also used to smooth the boundaries between the changed and unchanged region. In the simulation results, the proposed technique shows higher quality and lower fluctuation of picture quality than the conventional techniques on conditioning of the same bit rate.

I. 서 론

프레임간 보간 기법은 디스플레이 장치의 시간적 확장성에 대한 해결 방안이 될 수 있는데, 다양한 멀티미디어를 표현하는 단말기 시스템에서 프레임율을 높여야 할 필요가 있을 때 프레임 보간기법이

유용한 수단이 된다. 또한 프레임간 보간 기법은 기존의 동영상 압축 알고리즘과 결합할 경우 매우 유용한 압축 수단이 될 수 있는데, 입력 프레임 중 일부를 생략하고 적은 수의 프레임은 움직임 보상 및 DCT 등 기존의 동영상 부호화 기법으로 전송하고, 스킵된 프레임에 대해서 수신단에서 복원해 내코자 할 때 적용될 수 있다. 따라서, 프레임간 보간기법

* 전주대학교 전자메체공학과

** 군산대학교 정보통신공학과

논문번호: 99047-0204, 접수일자: 1999년 2월 4일

* 본 연구는 전주대학교 교내연구비로 수행되었습니다.

은 codec의 구현에 있어서 다소 복잡함이 따르지만 전송 비트율을 줄이기 위해서 유용한 수단이 될 수 있다.

초저비트율로 부호화된 영상을 전송하는 경우 문제점 중의 하나는 전송 능력의 한계로 인하여 시간 변화에 따른 화질의 차가 크며, 이는 대부분 움직임의 복잡도에 의존한다. 즉, 움직임이 적은 경우 움직임 벡터와 전송되는 예측 오차가 작기 때문에 적은 비트량으로도 양질의 영상을 표현할 수 있지만, 만일 움직임이 크거나 복잡한 경우 제한된 전송비트량으로 인하여 영상의 품질은 열화된다. 일반적으로 프레임간 보간 기법은 많은 경우 기준 프레임(reference frames) 간의 움직임 벡터를 바탕으로 스킵된 프레임을 복원한다. 움직임 보간을 위한 여러 가지 움직임 추정기법이 적용되었는데, Cafforio 등^[2]은 순방향 및 역방향 움직임 트래킹을 위해서 반복적(recursive) 알고리즘을 적용하였으며, Thoma 등^[4]은 제곱적 블록 매칭기법을 이용한 움직임 벡터 필드(motion vector field)의 추출기법을 제시하였다. 한편, 추정된 움직임 벡터를 바탕으로 움직임 보상형태의 영상 보간기법이 여러 가지 제안되었는데, Corbera 등^[5]은 순방향과 역방향에 대한 광류(optical flow)를 추정하고, 이를 바탕으로 프레임간 보간을 행하였는데, 특히 줌(zooming) 등의 카메라 동작으로 인한 광류 특성도 고려하였다. 그러나 이 경우 수신단에서 움직임 추정을 다시 해야 하는 번거로움이 뒤따르며, 또한 계산량이 증대된다. Wong 등^[7]은 블록 벡터를 이용하여 움직임 보간 기법을 제시하였는데, 블록단위로 동일한 벡터를 그대로 보간에 적용함으로써 성능이 떨어지며, 특히 움직임이 복잡한 경우 화질이 더욱 나빠질 수 있다. 이러한 블록 단위의 움직임 벡터를 적용한 예는 Kim 등^[8]이 제안한 기법에서도 발견할 수 있는데, 여기서는 다만 오정합된 움직임 벡터를 교정하기 위해 블록 단위의 움직임 벡터 스무딩 기법이 적용되었다. 그러나 이러한 화소 또는 블록 벡터 기반의 기법들이 갖는 문제점들을 살펴보면, 화소기반의 움직임 추정기법은 여러 가지 잡음의 영향 등으로 극부최소치 문제에 빠지기 쉽고, 블록 정합기법에 의한 움직임 추정은 물체의 형태를 고려하지 않기 때문에 오정합의 문제가 발생되기 쉽다.

한편, 앞에서 열거된 기존의 비디오 보간 기법들의 공통점은 일정한 스킵율로 스킵되고, 수신단에서 일정한 보간율로 스킵된 프레임(skipped frame)을 보간하며, 움직임의 복잡도에 상관없이 일정한 보간

율을 갖는다.

일반적으로 프레임간 보간은 적절한 움직임 추정을 행하여 얻어진 움직임 벡터를 바탕으로 움직임 필드를 구성하고 기준 프레임들 사이에 일치되는 블록, 영역 또는 화소를 찾아 이를 바탕으로 보간 프레임을 구성한다. 이 때, 보간 프레임의 화질은 기준 프레임간의 움직임 정도에 의해 많이 좌우된다. 따라서, 움직임이 큰 경우 보간되는 프레임의 화질은 저하된다. 특히, 움직임이 비선형적일 경우 화질 저하 정도는 더욱 악화된다. 따라서, 보간되는 프레임 간격이 커질수록 움직임의 정도에 더 큰 영향을 받는 것이 일반적이다. 기존의 기법들은 스킵율이 높고 움직임이 비선형인 경우 오정합으로 인하여 보간 효율이 떨어지고 이로 인하여 복원된 영상간의 화질의 변화가 심할 뿐만 아니라 움직임 보상 효율에 따라 전송 비트량의 변화가 크다.

본 논문에서는 이러한 문제점을 해결하기 위하여 가변 간격의 프레임 보간 기법을 제시한다. 제안된 보간 기법은 기준 프레임 사이의 간격을 움직임의 복잡도에 따라 적용적으로 변화시킴으로써 프레임간 화질 변화를 적게 하고, 또한 블록단위의 추정된 움직임 벡터를 바탕으로 영역별 움직임을 고려하여 실제 움직임에 근접한 움직임 필드를 구성하고, 이를 프레임 보간에 적용한다. 본 논문의 구성은 제2장에서 움직임 보상 기반의 프레임간 보간을 위한 제안된 보간기법을 논의하고, 제3장에서 제시된 기법에 대한 모의실험 및 실험 결과에 대한 분석을 통해 제시된 기법의 성능을 평가하며 마지막으로 제4장에서 결론을 맺는다.

II. 가변간격의 프레임 보간기법

1. 움직임 보상기반의 프레임간 보간기법

프레임간 보간 기법은 동영상 압축에 효과적으로 적용될 수 있는 기법인데, 움직임을 고려하지 않는 선형 보간 방법은 효과적인 보간을 수행할 수 없다. 따라서 효과적인 보간을 위해서는 움직임 정보를 적절히 이용하여야 한다.

그림 1에서 기준 프레임 R_A , R_B 사이에 다수의 스킵된 프레임을 복원하고자 할 때 움직임을 가진 물체에 대해 움직임을 고려하여야 한다. 이 때, 가장 중요한 것은 정확한 움직임 필드와 구성이다. 물체의 움직임은 여러 가지 이유로 인하여 기술하기가 용이하지 않다. 즉, 움직임이 3차원적인 회전이

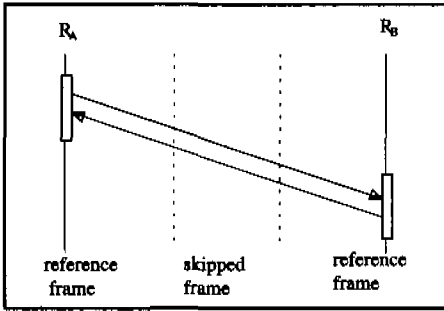


그림 1. 프레임간 보간

발생하거나 모양새가 변형될 때 정확한 움직임을 기술하기가 어렵다. 또한 움직임 보상에 적용되는 움직임 추정 기법이 일반적으로 블록 정합법에 바탕을 둔 평행 이동만을 고려하기 때문에 정확한 움직임의 추출은 쉽지 않다. 이 때 기준 프레임의 시간간격이 짧수록 정확한 움직임 필드의 구성이 어렵고, 따라서, 보간된 프레임의 화질은 더욱 열화된다.

2. 움직임 복잡도 분석에 의한 기준프레임 결정

2.1 MEF(moving edge in frame)의 구성

블록 정합법에 의한 움직임 보상은 평행 이동만을 고려하기 때문에 움직임이 서로 다른 영역들의 경계부분에 걸쳐진 블록에 있어서는 그 보상 효율이 크게 떨어진다. 따라서, 블록 단위의 움직임 벡터를 바탕으로 프레임간 보간을 적용할 경우 스킵율이 높을수록 영역별 경계부에서의 보간 효율은 떨어진다. 따라서, 프레임 단위로 움직임의 복잡도를 분석하여 스킵되는 프레임을 적절하게 결정하는 것이 보간 효율을 높이는 한가지 방법이 될 수 있다. 본 논문에서는 움직임 복잡도를 분석하고 이를 바탕으로 스킵되는 프레임을 결정함으로써 전송비트 대비 전송 효율을 높이고자 한다.

일반적으로 프레임간 차 (frame difference: FD)는 영역간 경계부분에서 움직임이 있을 경우 다른 부분에 비해 큰 차를 갖는다. 따라서, 프레임 차를 이용하면 움직임 복잡도를 분석할 수 있다. 본 논문에서는 움직임 영역외 경계부분을 움직임 에지(moving edge: ME)라 칭하고 프레임 전체에 대해 얻어진 움직임을 MEF(moving edge in frame)이라 부르기로 한다. ME는 일반적으로 프레임간 차를 구하여 차가 큰 경우에 발생하므로 문턱치를 높게 설정한다. 즉,

$$ME_k(x_1, x_2) = \begin{cases} 1, & \text{if } FD_k(x_1, x_2) > T_{ME} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

여기서 T_{ME} 는 ME를 얻기 위해 설정된 문턱치이며, $FD_k(x_1, x_2)$ 는 화소 (x_1, x_2) 위치에서 얻어진 프레임 k, l 간 차의 절대치이다. $ME_k(x_1, x_2)$ 는 (x_1, x_2) 에서 얻어진 프레임 차가 T_{ME} 보다 클 경우 움직임을 에지로 판정하고 1을 부여하며 그렇지 않으면 0을 갖는다. 프레임 전체 화소에 대해 얻어진 $ME_k(x_1, x_2)$ 를 이용하여 프레임 k, l 간의 MEF, 즉 $MEF_{k,l}$ 를 구성한다.

2.2 기준 프레임 결정

본 단계는 얻어진 MEF를 바탕으로 기준 프레임을 결정하는 단계이다. 그림 1에서 임의의 입력 시퀀스에 대해 프레임 R_A 와 R_B 는 기준 프레임으로서 전송되고 R_A 와 R_B 사이에 존재하는 일부 프레임에 대해서 전송하지 않고자 한다. 전송되는 R_A 와 R_B 를 결정하기 위해 먼저 입력 시퀀스가 $I_1, I_2, I_3, \dots, I_n$ 으로 입력될 때, 입력 시퀀스에 대해 맨 처음 프레임은 프레임내(intraframe) 부호화를 행하는 프레임으로서 반드시 부호화되고 전송되어야 하므로 I_1 을 기준프레임 R_A 로 놓는다. 다음으로 입력되는 프레임들에 대해서 스킵 여부를 결정하기 위한 과정은 다음과 같다.

- 단계 1: 기준 프레임 R_A 로 부터 바로 인접한 프레임들을 기준 프레임의 후보 프레임 R_C 로 설정한다.
- 단계 2: 기준 프레임 R_A 와 프레임 R_C 간의 $MEF_{A,C}$ 를 구하여 R_C 의 스킵프레임 여부를 다음과 같은 기준으로 결정한다.

$$\begin{matrix} \text{if } MEF_{A,C} < T_{MEF} & R_C \text{ 스킵됨} \\ \text{otherwise} & R_C \text{ 스킵되지 않음} \end{matrix}$$

여기서 T_{MEF} 는 스킵여부를 결정하기 위한 문턱치이다.

- 단계 3: 만일 R_C 가 스킵프레임이면 다음 입력 프레임을 R_C 로 재설정하고, 단계 2로 간다.
- 단계 4: 만일 단계 3에서 R_C 가 스킵 프레임이 아니면 이를 기준 프레임 R_B 로 결정하고 R_B 를 부호화하고 전송한다.
- 단계 5: 단계 4의 R_B 를 R_A 로 대체하고 단계 1로 간다.

기준프레임으로 결정된 영상들은 부호화되고 전

송된다. 이 때, 기준 프레임간의 간격에 대한 정보가 같이 전송되어 수신단에서 복호화되어 프레임간 보간시 이용된다.

3. 영역 기반의 프레임간 보간

3.1 움직임 영역 추출

부호화되어 수신된 영상정보는 복호화되고 재구성된 프레임을 얻는다. 본 논문에서는 블록 단위의 평행 이동에 대한 정보인 움직임 벡터를 프레임 보간에 적용하기 때문에, 블록단위의 움직임 벡터 중 일부는 교정될 필요성이 있다. 그러한 벡터로는 블록정합 과정에서 발생할 수 있는 실제 움직임과 다른 오벡터 그리고 서로 다른 움직임을 갖는 영역들의 경계에 걸쳐있는 블록의 벡터 등이다. 이러한 문제점을 해결하기 위해 본 논문에서는 블록 단위의 움직임 벡터를 영역별 움직임 벡터로 변환한다. 영역 단위의 구성은 블록 기반에 비하여 영역별로 움직임 벡터를 할당할 수 있는 장점을 갖는다. 따라서, 본 논문에서는 영역 단위의 움직임 벡터를 얻기 위하여 움직임 분할 기법을 도입한다.

이를 위한 첫 단계로 영상을 움직임 영역과 정지 영역으로 분류한다. 일반적으로 프레임 보간 기법이 적용되는 영상은 움직임의 범위 및 움직임이 크지 않은 영상을 대상으로 하는 것이 일반적이다. 카메라 흔들림 등 여러 가지 요인으로 인한 흔들림의 영향을 제거하기 위해 배경을 제외한 움직임 영역을 추출한다. 배경영역은 카메라가 정지되어 있는 경우 전 후 프레임간 공간적 변화가 없는 영역이며, 이 영역으로부터 움직임 영역을 얻기 위해 프레임간 차를 이용한다. 즉,

$$C_{k,l}(x_1, x_2) = \begin{cases} 1, & \text{if } FD_{k,l}(x_1, x_2) > T_{MR} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

여기서, $FD_{k,l}(x_1, x_2)$ 는 프레임 k, l 간의 (x_1, x_2) 에서의 화소 값 차의 절댓치를 의미하며, $C_{k,l}$ 는 문턱치 적용 후 얻어지는 값으로 움직임 영역에 속하는 화소는 1, 배경에 속하는 화소는 0을 갖는다. T_{MR} 은 움직임 영역을 구하기 위해 적용되는 문턱치인데, 식 (2)에서 적용된 T_{MR} 보다는 적은 값을 사용한다. 왜냐하면, 움직임 에지는 움직임 영역 중 큰 프레임 차를 가지는 경우이기 때문이다.

3.2 수리형태학적 필터링을 이용한 영역 스무딩

본 단계는 전 단계에서 얻어진 결과를 바탕으로

배경과 움직임 영역의 경계를 스무딩 (smoothing)시키는 단계이다. 이를 위해 본 논문에서는 수리형태학적 필터링 (morphological filtering) ^[6]을 적용한다. 수리형태학의 수학적 분석은 소위 Minkowski 대수로부터 유도할 수 있다. 2 차원 정수 공간 R^2 에서 집합 A, B 이 있다고 할 때, A 와 B 의 Minkowski 덧셈 $A[+]B$ 과 뺄셈 $A[-]B$ 는 각각

$$A[+]B = \cup_{x \in B}(A+x)$$

와

$$A[-]B = \cap_{x \in B}(A+x)$$

로 정의되며 ^[6], 이를 바탕으로 팽창 (dilation)과 침식 (erosion)은 각각 다음과 같이 정의된다.

$$dilation(A, B) = A[+]B$$

$$erosion(A, B) = A[-](-B)$$

여기서, $(-B)$ 는 집합 B 의 180° 회전한 집합 (reflection)을 의미한다. 주어진 영상을 A , 그리고 구성소 (structuring element)를 S 라고 할 때, A 의 고립점을 제거하고 경계는 매끄럽게 처리하기 위해 개방 (opening)과 폐쇄 (closing)을 순서대로 행하며, 이 때, 개방 $open(A, S)$ 과 폐쇄 $close(A, S)$ 는 아래와 같이 정의된다.

$$open(A, S) = dilation(erosion(A, S), S) = (A[-](-S))[+S] \quad (3)$$

$$close(A, S) = erosion(dilation(A, -S), -S) = (A[+](-S))[-S] \quad (4)$$

여기서, 개방은 침식 과정인 $erosion(A, S)$ 을 행한 후 팽창을 행하고, 폐쇄는 팽창인 $dilation(A, -S)$ 을 행한 후 침식 과정을 행하여 얻는다. 이 때, M 은 움직임 영역이며, S 는 이 때 적용된 구성소이다. 본 과정은 화소 단위의 움직임 영역에 대해 필터링을 통하여 보간시키거나 제거시키는 과정이며, 특히 배경과 움직임 영역간의 경계부분 매끄럽게 만든다.

4. 움직임 벡터 필드의 구성

본 단계에서는 영역별로 움직임 벡터를 할당하기 위하여 영역 분할 기법을 도입한다. 영역 분할은 블록 분할에 비하여 객체별 움직임을 기술하기가 용이하다. 영역별 움직임 벡터 할당을 위한 벡터의 추정은 계산량을 고려하여 전송되어 온 블록 단위의 벡터를 이용하여 행한다. 먼저, 블록 단위의 벡터를

이용하여 영역별 움직임 벡터를 할당하고, 다음으로 영역별 경계부에 대해 교정을 실시하는데, 세부적인 과정은 다음과 같다.

4.1 영역별 움직임 벡터 할당

영역별 움직임 벡터 할당을 위하여 먼저 수신되어 복호화된 영상에 대해 영역분할을 한다. 이 때, 영역 분할은 위에서 얻어진 움직임 영역에 대해서만 행한다. 영역 분할된 영상에 대해 영역에 포함되거나 걸쳐져 있는 블록들의 움직임 벡터들을 이용하여 최소 DFD (displaced frame difference)를 가지는 벡터를 그 영역의 움직임 벡터로 정한다. 즉, 임의의 영역 R에 포함된 블록 움직임 벡터들이

$$\{v_{R_1}, v_{R_2}, \dots, v_{R_n}\} \quad (8)$$

로 주어질 때, 이 벡터들을 이용하여 DFD를 구하고 최소 DFD를 갖는 벡터를 영역 R에 대한 움직임 벡터 v_R 로 한다. 즉,

$$v_R = \min[DFD(v_{R_1}, v_{R_2}, \dots, v_{R_n})] \quad (9)$$

여기서 $DFD(v_{R_i})$ 는 기준 프레임 R_A, R_B 에 대해서

$$DFD(v_{R_i}) = \sum_{x_1, x_2 \in R} [R_A(x_1, x_2) - R_B(x_1, x_2)]^2 \quad (10)$$

이다.

4.2 화소별 움직임 벡터 할당

영역별로 얻어진 움직임 벡터를 바탕으로 화소별 움직임 벡터를 정한다. 즉, 임의의 영역에 대해 얻어진 움직임 벡터를 그 영역에 포함된 화소에 동일한 벡터를 할당한다.

5. 보간 알고리즘

일반적으로 스킵된 프레임은 재구성하기 위해서는 기준 프레임간에 구해진 양방향 움직임 벡터 필드를 바탕으로 이루어진다. 즉, IPPP...IPPP.. 구조에서 수신되는 벡터를 순방향 움직임 벡터라고 하고, 그 반대 방향을 역방향 움직임 벡터라고 하자. 이와 같이 양방향의 움직임 벡터 필드를 구하기 위해서는 수신단에서 역방향 움직임 벡터를 구해야 한다. 본 논문에서는 계산상의 복잡도를 고려하여, 순방향 움직임 벡터만을 이용하여 보간을 시도한다.

그림 2에서, 기준 프레임이 R_A, R_B 이고, 보간

되는 프레임을 I_1, I_2 로 두자. 부호화 과정에서 R_B 프레임의 각 블록들은 R_A 로부터 움직임 보상된다. 한편, R_B 프레임의 각 블록 또는 화소는 R_A 프레임의 그것과 반드시 일대일 대응 관계를 이루는 것은 아니다. 즉, R_A 의 화소 또는 블록은 R_B 의 움직임 보상에 반복해서 이용되기도 하고 또는 전혀 보상에 이용되지 않는 경우도 발생될 수 있다. 예를 들어, R_A 에서 P는 반복해서 보상에 이용된 부분에 해당되며, Q와 S는 일대일 대응관계를 이루며 한번만 보상에 이용된 부분, 그리고 R은 보상에 전혀 이용되지 않는 부분에 해당된다. 한편, 프레임 R_B 에 있어서 움직임 영역에 해당되는 f, g, h와 배경에 해당되는 i, j는 모두 프레임 R_A 의 대응되는 각 화소와 일대일 대응관계를 이루고 있으며, 이에 반해 a, c, d와 b, e는 각각 동일한 화소 P_1, P_2 와 중복해서 대응되어 움직임 필드가 형성되어 있으며 일대일 대응관계를 갖지 못한다. 이러한 점을 고려하여 R_A 의 R_B 의 대응관계에서 R_A 의 각 화소는 다음과 같이 분류된다.

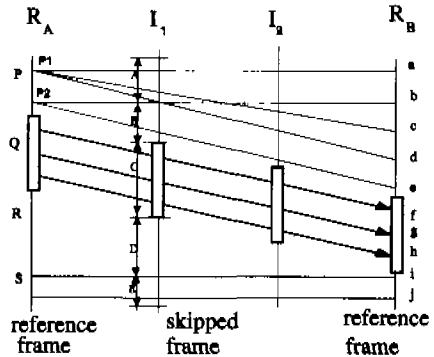


그림 2. 움직임 특성별 화소의 분류

첫째, R_A 와 R_B 간에 일대일 대응 관계를 갖는 화소

둘째, R_A 의 한 화소에 대해 R_B 의 여러 개의 화소가 대응되는 관계를 갖는 화소

셋째, R_B 의 어떠한 화소와도 대응관계를 갖지 않는 화소

위의 분류에서 첫번째의 경우는 배경 또는 움직이는 물체에 해당되는 부분이며, 세번째의 경우는 드러난 배경의 경우에 해당되며, 두번째의 경우는 중복 대응문제로서 교정되어야 할 대상이다.

이러한 중복 대응 문제를 해결하기 위해서 본 논문에서는 DFD를 이용한다. 예를 들어 R_A 의 P_1 은 R_B 의 a, c, d에 중복해서 움직임 필드가 형성되어 있다. 이 경우 a, c, d의 각각의 경우에 대해 P_1 간의 DFD 들을 구하고 그 중에서 최소 값을 갖는 것을 화소를 선택하고 이것을 실제 이동 위치로 간주한다. 그 외의 경우는 잘못된 벡터로 간주한다. DFD가 가장 적은 경우가 a 이라면, a만이 P_1 에 대응되는 위치로 판정하고 P_1 과 c, d간의 대응관계를 제거한다. 결과적으로 R_B 의 c, d, e는 움직임 필드가 제거되고 R_A 와 어떠한 대응관계도 갖지 못한다. 이러한 과정을 통해 재구성된 움직임 필드에서 R_A 와 R_B 의 각 화소는 일대일 대응관계를 갖는 경우와 대응관계를 갖지 못하는 경우의 두 가지로 정리된다.

이러한 과정을 통해 재구성된 움직임 필드를 바탕으로 보간을 행한다. 움직임에 따라 발생하는 화소의 성격은 크게 다음과 같이 분류할 수 있다. 즉, 정지된 배경영역(static background region), 움직이는 영역(moving part), 가려진 영역(covered region), 드러난 영역(uncovered region) 등이다. 각 영역에 대한 보간 방법은 아래와 같다.

1) 배경영역 : 배경영역은 기준 프레임간에 움직임 벡터의 크기가 영인 화소로 구성된 영역을 의미한다. 이 경우에는 대응되는 화소간의 평균값을 얻어 보간 된다. 보간 되는 프레임 I_1 에서 E에 해당되는 경우이다.

2) 움직임 영역 : 움직임 부분은 벡터의 크기가 영이 아니고, 기준 프레임간에 일대일 대응되는 경우이며 보간 프레임 I_1 의 C에 해당되는 경우이다. 기준 프레임들 각각 R_A , R_B 일 때, N을 기준 프레임간 간격, k를 R_A 로부터 간격이라 할 때, 보간되는 프레임 I_k 의 움직임 화소 (x_k, y_k) 에 대한 보간값은 해당되는 기준 프레임의 화소 값의 거리 가중된 평균으로 보간된다.

3) 가려진 영역 : R_A 에서 R_B 로 물체가 이동한 후 이동되는 물체로 인하여 배경이 가려지는 경우인데, 보간 프레임 I_1 에서 D에 해당되는 경우이다. 이러한 영역은 R_A 로부터 보간된다.

4) 드러난 배경 : 드러난 배경의 경우 R_B 의 일부분이 R_A 에 존재하지 않을 때 발생된다. 보간 프

레이프 I_1 에서 B에 해당되는 경우이다. 이러한 영역은 R_B 로부터 보간된다.

Ⅲ. 컴퓨터 모의실험 및 검토

제시된 알고리즘의 성능 평가를 위해서 컴퓨터 모의실험을 행하였다. 대상 영상으로는 352×288 크기의 'salesman' 비디오 시퀀스 200 프레임과 256×256 크기의 'Trevor' 비디오 시퀀스 100 프레임 사용하였으며, 둘 다 30 fps (frames per sec) 율을 갖는다. 제시된 기법의 성능을 평가하기 위해 보간 기법을 적용하지 않는 IPPP...IPPP.. 구조를 갖는 비보간 기법(non-interpolation)과 스킵 프레임수를 고정시킨 기법(MCI-CI: motion compensated interpolation -constant interval) [7]과 비교하였다. MCI-CI 기법은 고정 스킵율을 가지며 기준 프레임 사이에 2 프레임씩 스킵하도록 설정하고 초당 10프레임의 전송율을 가진다.

일반적으로 여러 가지 기법들에 대한 성능 평가는 화질과 비트율 모두를 고려해야만 그 정확한 평가를 할 수 있다. 본 논문에서는 성능 평가를 위해서 비트율을 고정시키고 화질을 평가하는 방식을 택했다. 즉, 세 가지 기법의 객관적인 성능 평가가 될 수 있도록 하기 위해서 세 가지 방법 모두 동일한 전송 비트율/초를 가지도록 하였다. 이를 위해 모든 기법에 대해 I-프레임과 P-프레임의 할당 비트량을 각각 동일하게 할당하였다. 비교되는 각 기법별 I-프레임 및 P-프레임의 구성 및 비트율 할당 관계를 살펴보면, 첫째, I-프레임의 부호화에 있어서 비보간 기법의 경우 초당 30프레임의 전송율을 가지며 본 모의 실험에서는 초당 2 개의 I-프레임을 갖도록 설정하였으며, 다른 두 가지 방법의 경우 초당 약 10프레임의 전송율을 가지며 따라서, 초당 한 개의 I-프레임을 갖도록 했다. 따라서 비보간 기법은 15프레임마다 프레임 내(I-프레임) 부호화를 하였으며 다른 두가지 기법에 대해서는 10 프레임마다 프레임 내(I-프레임) 부호화를 하였다. 이러한 가정을 바탕으로 각 기법에 대한 I-프레임 할당 비트율을 동일하게 하기 비보간 기법은 다른 두 기법에 비해 1/2의 비트를 할당했다. 둘째, 초당 전송되는 프레임수 중에서 P-프레임의 수는 비보간 기법은 30프레임 중 28프레임, 나머지 두가지 기법의 경우 약 10프레임 중 약 9 프레임의 P-프레임을 갖는다. 따라서, P-프레임 전송에 소요되는 비트량을 각 기

법별로 일치시키기 위해 비보간 기법의 할당 비트율은 다른 두 기법의 약 1/3을 할당하였다.

부호화 방법으로는 세 가지 기법 공히 동일한 부호화 기법을 적용하였는데, 시작 프레임과 매 초당 시작하는 첫 프레임에 대해서는 프레임내 부호화를 행하고, 나머지 프레임들에 대해서는 기존 프레임들에 대해 프레임간 움직임 추정을 통한 움직임 예측 및 예측 오차 부호화 기법을 적용하였다. 또한, 움직임 벡터의 비트율 계산 및 DCT 계수에 대한 양자화와 비트율 계산은 MPEG-2의 VLC코드 및 양자화 테이블을 적용하였다. 고정 비트의 할당 방법으로는 프레임당 목표 비트 (target bits)를 설정하고 프레임 단위의 발생 비트량과 목표 비트량을 비교하여 목표비트량에 가장 근접하는 비트량을 갖도록 양자화 파라미터를 조절하여 비트량을 할당하였다. 다만, 이러한 방식은 고정비트량 하의 여러 가지 기법간의 화질에 대한 성능 평가를 위한 것이다.

그림 3은 실험에 사용된 'Salesman' 영상이다. 그림 4는 프레임간 움직임 복잡도 분석을 위해 얻어진 MEF인데, 배경영역 등에서 발생하는 고립점 등은 제거된다. 그림 5는 프레임간 보간을 위하여 고려되는 움직임 영역을 나타낸 것인데, 정제선 내부가 움직임 영역이다. 이 때 사용되는 문턱치는 FD와 절대치 값이 5이상이면 움직임 영역으로, 아니면 배경영역으로 결정하였으며, 움직임 에지의 경우 문턱치를 20으로 설정하였다.

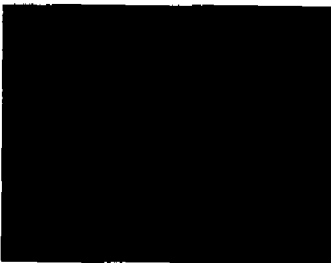


그림 3. 'salesman' 영상



그림 4. 프레임 움직임 에지(MEF)

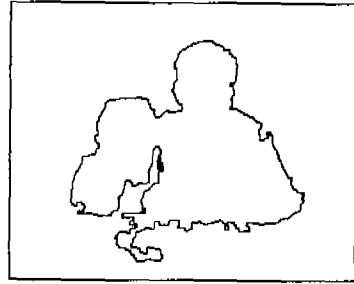


그림 5. 움직임 영역

이러한 값들은 여러가지 영상에 대해 반복적으로 테스트를 한 후 얻어진 것인데, 결정 방법은 문턱치 적용으로 얻어지는 결과를 시각적으로 움직임 영역 또는 움직임 에지로 판단되어 얻어지는 결과와 비교하여 가장 유사한 경우에 해당되는 값들이다.

그림 6과 그림 7은 각각 'Salesman' 및 'Trevor' 비디오 시퀀스에 대한 실험 결과인데, 각각의 그림에서 맨 아랫쪽 그래프는 스킵되는 프레임 수를 나타내며 위쪽 3 개의 그래프는 제시된 가변 간격기법을 적용하여 보간된 결과(Proposed)와 기존의 고정 스킵 간격의 결과(MCI-CI) 및 비보간 기법의 실험 결과를 각각 제시한 것이다. 제안된 기법은 부호화단에서 최소한 스킵되는 프레임 수를 1 프레임 이상으로 하고 최고 3 프레임까지 스킵하였다. 실험 결과에서 제안된 기법은 비보간 기법 및 고정 스킵 간격을 갖는 MCI-CI 기법에 비해 화질의 변화가 크게 줄어들음을 알 수 있다. 화질 평가를 위하여 전송 비트율을 'Salesman' 비디오 시퀀스의 경우 150 Kbit/s, 'Trevor' 비디오 시퀀스에 대해서 100 Kbit/s로 실험을 하였는데, I-프레임이 매우 높은 비트량을 차지하기 때문에, I-프레임에 대해 적절한 압축기법을 사용할 경우 비트율을 크게 감소시킬 수 있다. 이러한 전송 비트율을 바탕으로 'Salesman' 시퀀스에 대해서 PSNR에 있어서 비보간 기법은 30.4 dB, MCI-CI 기법은 32.4 dB, 제안된 기법의 경우 32.9 dB로서 평균 화질에 있어서 MCI-CI 기법에 비해 약 0.5 dB의 성능향상을 가져왔다. 'Trevor' 영상에 대해서 비보간 기법은 31.2 dB, MCI-CI 기법은 31.7 dB, 제안된 기법의 경우 32.4 dB로서 평균 화질에 있어서 MCI-CI 기법에 비해 약 0.7 dB의 성능향상을 가져왔다. 스킵 프레임이 차지하는 비율에 있어서, 제안된 기법의 경우 전송 프레임수는 'Salesman' 영상에 대해서 200 프레임에 대해 143 프레임이 스킵되었으며 스킵율은 약 0.71며, 'Trevor' 영상에 대해서 100 프레임에 대해

69 프레임이 스킵되어 스킵율은 0.69이다. 이에 비해 기준 프레임 사이에 2프레임씩 스킵한 MCI-CI 기법은 스킵율이 약 0.67로서 제안된 기법이 더 높은 스킵율에도 불구하고 더 높은 화질을 보인다. 'Salesman' 영상은 배경이 매우 복잡한 영상이기 때문에 비보간 기법은 전송 프레임수는 다른 두가지 기법에 비해 3배이며, I-프레임은 2배, 따라서 다른 2가지 기법에 비해 I-프레임에 할당되는 비트량은 그 절반이기 때문에 화질이 매우 열화되었다. 또한 MCI-CI 기법의 경우 움직임이 복잡한 프레임에서 그 성능이 매우 떨어지는데, 예를 들어 'Salesman' 비디오 시퀀스의 5 - 16번 프레임사이에서 화질의 변화가 크다. 이에 비해 제안된 기법은 스킵율을 적절히 가변시킴으로써 양호한 보간 화질을 얻을 수 있다. 한편, 그림 6과 7에서 기존의 기법이 제안된 기법에 비해 더 좋은 화질을 보이는 경우가 발생되는데, 이것은 다음과 같은 이유로 발생된다. 즉, 제안된 기법은 스킵되는 프레임수가 1 - 3 프레임으로 가변적이다. 반면에 기존의 기법의 경우 스킵되는 프레임수가 2프레임으로 고정적이다. 따라서, 제안된 기법과 기존의 고정적 기법에서 기준 프레임 및 보간되는 프레임의 위치가 서로 일치하지 않으며, 따라서, 기존의 기법의 경우 기준 프레임으로 처리되는 프레임이 제안된 기법에서는 스킵되어 보간되기 때문에 그러한 경우 기존의 고정 기반의 기법에 대한 결과가 더 좋은 화질을 보일 수 있다. 제안된 기법은 최대 스킵 프레임 수를 3프레임보다 높게 스킵하는 경우 화질 변화를 더욱 줄이면서 전송에 소요되는 비트량을 줄일 수 있음을 예상할 수 있는데, 다만 스킵 간격이 커지면 지연이 커지기 때문에 최대 3 - 4 프레임이 적절한 것으로 사료된다.

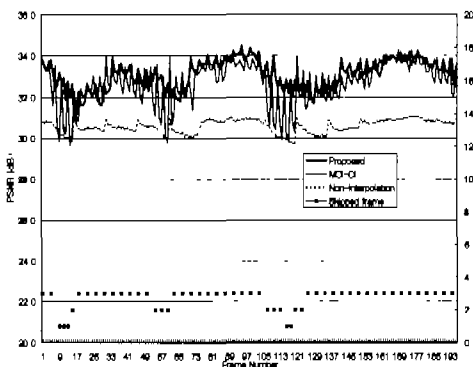


그림 7. 'Salesman' 비디오 시퀀스에 대한 화질 비교

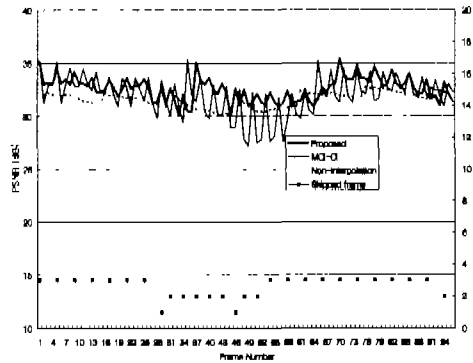


그림 8. 'Trevor' 비디오 시퀀스에 대한 화질 비교

IV. 결론

본 논문에서는 가변 간격의 프레임간 보간 기법을 제시하였다. 제시된 기법에서 스킵 프레임은 프레임의 움직임 복잡도를 바탕으로 결정되며 움직임 복잡도를 평가하기 위해 움직임 에지물 이용하였다. 제안된 기법은 보간기법을 적용하지 않는 기법 및 고정 스킵율을 갖는 기법들에 비해 개선된 화질을 얻을 수 있으며, 또한 시간 축 대비 화질의 변화를 줄일 수 있었다.

참고 문헌

- [1] H. G. Musemann, P. Pirsch, and H. J. Grallert, "Advances in picture coding," Proc. IEEE 73, pp. 523-548, 1985.
- [2] C. Cafforio and F. Rocca, and S. Tubaro, "Motion compensated image interpolation," IEEE Trans. Commun. COM-38, pp. 215-222. 1990.
- [3] V. Seferidis and M. Ghanbari, "General approach to block matching motion estimation," Opt. Eng., vol. 32, n1o. 7, pp. 1464-1474, July 1993.
- [4] R. Thoma and M. Biering, "Motion compensating interpolation considering covered and uncovered background," Signal Processing: Image Communication I, pp. 191-212, 1989.
- [5] J. Ribas-Corbera and Sklansky, "Interframe interpolation of cinematic sequences," Journal

