

논문 2009-46SP-6-13

# 우세 움직임 벡터를 이용한 프레임 보간 기법

## (Frame Interpolation using Dominant MV)

최 승 현\*, 이 성 원\*\*

(Seung-Hyun Choi and Seong-Won Lee)

### 요 약

LCD(Liquid Crystal Display)의 발전은 이전에 사용하던 CRT를 대체하고 있다. 하지만 LCD와 같은 홀드 타입(Hold type)형 디스플레이의 구조적 특성으로 인해 동영상 재생 시 움직임 블러나 움직임 저더 등의 많은 문제점이 발생했다. 이를 해결하기 위해 화면 사이사이에 새로운 화면을 생성하여 화질을 개선하기 위한 화면 보간 기법을 사용하고 있다. 본 논문에서는 우세 움직임 벡터와 블록간 분산값의 차이를 이용한 화면 보간 기법을 제안한다. 제안하는 알고리즘은 먼저 생성되는 프레임 전후의 프레임에 대하여 블록매칭을 이용한 단방향 움직임 추정을 통한 블록 움직임 예측을 하게 된다. 그리고 블록 움직임 예측에서 찾지 못한 부분을 비교 블록간 분산값을 통해 픽셀 평균값과 영향이 큰 모션 벡터를 선택하여 적용시키는 픽셀 움직임 예측으로 새로운 프레임을 생성한다. 제안하는 알고리즘을 이용하여 여러 실험영상의 프레임 보간 결과를 비교하였으며, 기존 알고리즘인 양방향 화면 보간 방법에 비하여 약 3dB 정도의 PSNR 증가를 얻을 수 있었고 단방향 화면 보간 방법에 비해 계산량을 줄일 수 있었다.

### Abstract

The emerging display technology has been replaced the previous position of the CRT with the LCD. The nature of hold type display such as LCD, however, causes many problems such as motion blur and motion judder. To resolve the problems, we used frame interpolation technique which improves the image quality by inserting new interpolated frames between existing frames. In this paper, we propose a novel frame interpolation technique that uses dominant MV and variance different value in each block. At first, the proposed algorithm performs unidirectional motion estimation using blocking matching algorithm. The new frame is generated by pixel average using compared block variance or by pixel motion compensation using dominant motion vector, whether the motion estimation find the target area or not. Several experiments with the proposed algorithm shows that the proposed algorithm has better image quality than the existing bidirectional frame interpolation algorithm at the rate of about 3dB PSNR and has low complexity comparing to the unidirectional frame interpolation technique.

**Keywords :** frame interpolation, unidirectional motion estimation, bidirectional motion estimation, dominant motion vector, variance

### I. 서 론

기존에 사용하던 CRT기반의 디스플레이는 제조가 용이하였으나 부피가 크고 무거우며, 대형화가 힘들었다. 이를 극복하기 위해 PDP, LCD, OLED등 다양한 방

식의 디스플레이가 제안이 되었으며, 그 중에 현재 가장 많이 사용되고 대중화된 기술 중 하나가 LCD이다. LCD는 물리적인 구조상 현재 화면의 값이 다음 화면이 입력될 때까지 계속 유지되는 홀드 타입(Hold type) 디스플레이의 특성을 갖게 된다. 그러나 LCD와 같은 홀드 타입 디스플레이의 특성으로 인해 동영상을 재생할 때 모션 블러(Motion Blur), 모션 저더(Motion Judder) 같은 모션 아티팩트(Motion Artifact)가 나타나게 된다. 이러한 홀드 타입 디스플레이의 단점을 해결하기 위한 방법 중 하나로서 화면 보간 기법(Frame Interpolation)을 사용한다. 화면 보간 기법을 이용하여 일반적인 영

\* 학생회원, \*\* 정회원, 광운대학교 컴퓨터공학과  
(Kwangwoon University Computer Engineering)  
※ “본 연구는 서울시 산학협력사업의 지원과 교육과학기술부의 재원으로 한국연구재단의 지원(KRF-2006-331-D00420, 2009-0088064)과 2007년도 광운대학교 교내학술연구비 지원에 의해 연구되었음.”  
접수일자: 2009년7월3일, 수정완료일: 2009년10월26일

상의 프레임들 사이에 새로운 프레임을 구성, 삽입하여 홀드 타입 디스플레이의 단점을 극복할 수 있을 정도로 충분한 화면 재생율을 만들어 냄으로서 화질을 향상시킬 수 있다.

이러한 화면 보간 기법은 홀드 타입 디스플레이의 단점을 보완하는 것 이외에도, 영화를 TV에서 보기 위한 24Hz/30Hz FRC(Frame Rate Conversion), 미국향 콘텐츠와 유럽향 콘텐츠 간의 25Hz/30Hz FRC등에서 사용되며, 낮은 화면 재생율의 영상을 부드럽게 보기 위한 화면 재생율의 증가(15Hz에서 30Hz)에도 사용된다. 또한 최근 많은 연구가 진행되고 있는 분산 비디오 코딩(Distributed Video Coding)과 같은 영상 압축 기법의 복호화에서 화면의 재구성에 가장 중요한 요소로서 화면 보간 기법이 사용되고 있다.

초기의 화면 보간 기법은 새로운 프레임을 생성할 때 화면의 단순반복이나 전후 프레임의 선형적인 화면 보간 방법을 이용했다. 그러나 이런 방법들은 움직이는 물체에서 모션 아티팩트를 발생시키는 화질 열화가 발생할 수 있다. 따라 최근에선 움직임 추정을 이용한 보다 적극적인 화면 보간 방법들이 연구되고 있다<sup>[1-6]</sup>.

기존 움직임 추정을 이용한 화면 보간 기법은 존재하는 화면 간의 움직임 추정에 의한 단방향 모션 벡터를 사용하는 방법과 생성될 화면에 대한 움직임 추정에 의한 양방향 모션 벡터를 사용하는 방법의 두 가지로 나눌 수 있다. 전자는 보간될 화면에 비어있는 영역이 생기므로 비어있는 영역을 채우기 위한 추가적인 연산이 필요하다. 한편 후자는 화면에 비어있는 영역은 발생하지 않으나, 움직임 추정이 부정확해져 이의 정확성을 높이기 위한 Refinement, 가변 블록 사이즈를 이용한 모션 벡터 찾기, 블록 아티팩트를 줄이기 위한 중첩 블록 움직임 보상 등의 기술들이 추가적으로 필요하게 된다.

본 논문에서는 화면 속 움직임 추정 효율을 높임과 동시에 화질 개선을 위하여, 블록 움직임 예측을 통해 새로운 화면을 생성하고, 생성된 화면에 부족한 데이터를 채우기 위해 픽셀 움직임 예측 알고리즘을 제안한다. 블록 움직임 예측은 기존 논문에서 사용된 단방향 모션 벡터를 사용하는 블록 매칭 알고리즘(Block Matching Algorithm, BMA)을 이용한다. 단방향 모션 벡터를 찾기 위한 BMA는 기타 다른 움직임 추정 알고리즘에 비해 계산량이 적고, 구현이 용이하며 MPEG 계열 및 H.261, H.263, H.264 등의 영상표준에서 많이 활용

되고 있고, 효율적인 구현방법에 대해서도 많은 연구가 이루어지고 있다<sup>[7]</sup>. 그리고 픽셀 움직임 예측 알고리즘은 비교 블록간 분산값의 차이를 이용해 비어있는 영역에 우세 움직임 벡터와 동일 위치에 있는 전후 프레임 픽셀값의 평균을 선택적으로 적용할 수 있는 방법을 새롭게 제안한다.

II장에서는 블록 움직임 예측을 이용한 기존의 화면 보간에 대해 소개하고, III장에서는 제안하는 알고리즘을 설명한다. IV장에서는 실험 내용 및 결과를 설명하고 V장에서 결론을 기술한다.

## II. 기존의 화면 보간 방법

화면 보간이란 영상이 존재하는 프레임의 정보를 이용하여 프레임 사이의 존재하지 않는 프레임을 재구성하게 된다. 이때 BMA 등의 기법을 이용한 움직임 정보를 기초로 하여 화면을 구성한다. 기존의 화면 보간 기법은 양방향 모션 벡터를 이용한 방법과 단방향 모션 벡터를 이용하는 방법이 있다.

### 1. 양방향 모션 벡터를 이용한 화면 보간

그림 2-1은 양방향 모션 벡터를 이용해 새로운 프레임을 생성하는 방법을 나타내고 있다. 이 방법은 먼저 새롭게 생성되는 프레임을 기준으로 블록을 나눈다. 나누어진 블록을 기준으로 이전 프레임과 이후 프레임에 검색 영역을 정하고 검색 범위 내에서 새롭게 생성되는 중간 프레임의 블록을 통과하면서 양쪽 블록간 에러 값이 가장 작은 모션 벡터를 찾게 된다. 이렇게 계산된 모션 벡터를 이용해 두 개의 연속된 프레임 N+1, N-1 사이에 새로운 프레임 N을 생성한다고 할 때 새로운 프레임에 해

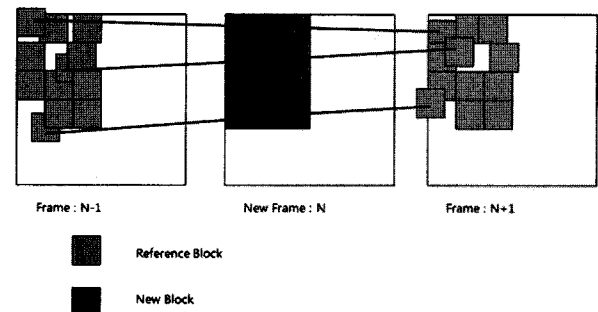


그림 2-1. 양방향 모션 벡터를 이용해 새로운 프레임 생성

Fig. 2-1. New frame creation using bi-directional motion vector.

당하는 블록을 재구성할 수 있게 된다<sup>[4~5, 7]</sup>.

이 방법은 생성되는 프레임에 비어있는 곳이 없도록 화면을 구성할 수 있다. 따라서 새롭게 생성되는 프레임에서 블록들의 겹침이나 비어있는 영역에 대한 처리가 필요 없게 된다.

새롭게 생성되는 프레임에 사용되는 수식은 아래와 같다<sup>[4~5, 7]</sup>.

$$F_N = \frac{F_{N-1}(X - \alpha \cdot MV) + F_{N+1}(X + (1 - \alpha) \cdot MV)}{2} \quad (1)$$

$MV$ 는  $F_{N-1}$ 과  $F_{N+1}$ 사이에서 계산된 모션 벡터 값이고 새롭게 생성되는  $F_N$ 의 위치에 따라  $\alpha$ 값에 가중치를 두게 된다.

이렇게 양방향 모션 벡터를 이용한 화면 보간 방법의 경우 새롭게 생성되는 화면에 대한 정보를 모션 벡터값을 이용해 계산할 수 있어 추가적인 계산없이 곧바로 화면을 생성할 수 있는 장점이 있지만 새롭게 생성되는 프레임을 지나가는 모션 벡터 값이 정확하지 않아 화면 왜곡이 심해지는 경우가 많다. 따라서 움직임 예측은 양측 모션 벡터를 이용해 계산을 하고, 계산된 모션 벡터를 재조정하는 과정을 거친 후 움직임 보상을 위한 알고리즘을 추가하여 화질을 개선시킨다<sup>[2]</sup>. 하지만 화면을 구성하는 과정에서 블록킹 효과(Blocking effect)가 발생하게 된다.

## 2. 단방향 예측을 통한 화면 보간

단방향 예측을 통한 움직임 예측은 양방향 모션 벡터를 이용한 화면 보간과 마찬가지로 프레임 N+1에서 프레임 N-1에 대한 모션 벡터를 구하고 그 모션 벡터를 그대로 이용하여 새로운 프레임을 생성하는 방법이다. 그림 2-2는 단방향 예측에서의 BMA를 이용하여 구한 모션 벡터를 보여주고 있다. 그리고 그림 2-3은 단방향 예측을 통한 화면 보간에 대한 설명이다.

$$F_N(X + (1 - \alpha) \cdot MV) = \frac{F_{N-1}(X - \alpha \cdot MV) + F_{N+1}(X)}{2} \quad (2)$$

단방향 예측을 통한 움직임 예측은  $F_{N+1}$ 프레임의 블록이 기준이 되므로 위치를 고정시키고 생성되는  $F_N$  프레임 위치의 가중치  $\alpha$ 를  $F_{N-1}$ 프레임에 적용시키면 식 (2)와 같이  $(1 - \alpha)$ 의 위치에  $F_N$ 프레임이 생성된다.

이때 모든 블록이 같은 방향으로 모션 벡터를 가지고 있는 게 아니므로 그림 2-2에서처럼 다른 방향의 모션

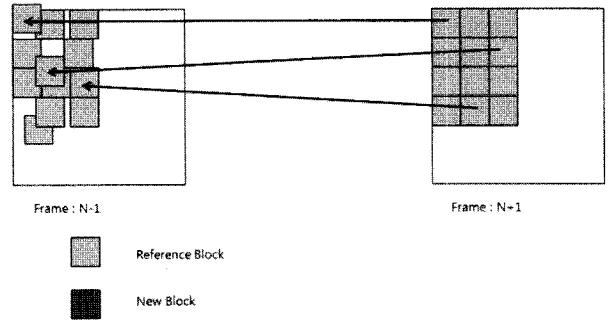


그림 2-2. 단방향 모션 벡터  
Fig. 2-2. Uni-directional motion vector.

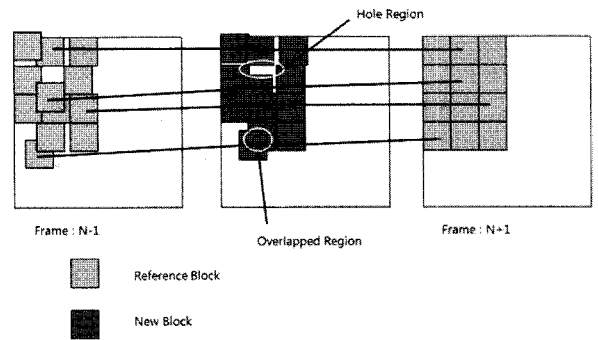


그림 2-3. 단방향 모션 벡터를 이용해 새로운 프레임 생성  
Fig. 2-3. New frame creation using uni-directional motion vector.

벡터가 나타나게 되고, 이를 이용해 새로운 프레임을 생성할 경우 그림 2-3처럼 모션 벡터가 적용되지 못한 비어있는 영역이나 블록이 서로 겹쳐 화소값이 중복되는 픽셀들이 생기는 영역이 존재하게 된다<sup>[10]</sup>. 따라서 이들에 대한 추가적인 처리가 필요하다.

### 가. 데이터가 중첩된 영역

현재 프레임에서 이전 프레임으로 모션 벡터를 계산할 경우 계산된 모션 벡터값이 겹치면서 새로운 프레임에서 블록들이 겹치는 영역이 발생하게 된다. 정확한 모션 벡터를 사용하기 위해 이 영역들은 겹치는 블록들 중 에러 값이 가장 작은 블록을 선택하여 해당 영역의 픽셀 값으로 사용한다<sup>[10]</sup>.

### 나. 데이터가 비어있는 영역

데이터가 비어있는 영역에 대한 모션 벡터 정보는 존재하지 않으므로 비어 있는 영역에 대한 모션 벡터를 주위에서 찾는다. 비어 있는 영역의 주변 픽셀을 따라 픽셀에 적용시킬 수 있는 모션 벡터를 찾고, 찾은 모션

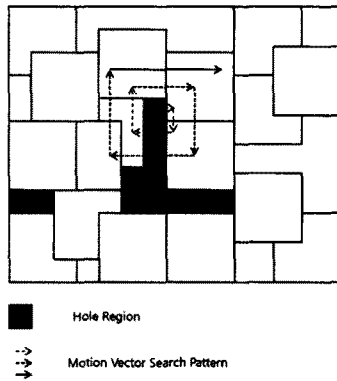


그림 2-4. 비어있는 영역에 대한 참조 모션 벡터 검색  
Fig. 2-4. Reference motion vector search on hole region.

벡터를 적용시켜 비어 있는 영역에 새로운 픽셀 데이터를 적용시켜 화면을 완성시킨다. 하지만 모션 벡터가 존재하지 않는 영역의 발생으로 인해 추가적인 연산을 피할 수 없고, 추가적인 연산에 대한 성능에 따라 알고리즘의 성능이 결정되는 단점이 있다.

그림 2-4는 기존의 비어있는 영역을 채우는 방법을 보여주고 있다<sup>[10]</sup>. 각 화살표는 모션 벡터를 검색하는 순서로 비어있는 영역의 픽셀로부터 출발하여 픽셀 주변의 모션 벡터를 검색하게 된다. 픽셀 바로 옆에 있는 모션 벡터를 먼저 검색하며 소용돌이 형태로 모션 벡터가 나타날 때까지 검색을 하게 된다.

모션 벡터  $MV_n$  계산은 주위 4개의 모션 벡터를 이용한다. 픽셀을 중심으로 한 바퀴 검색하는 걸 첫번째 경로, 검색된 다음 위치에서 한바퀴 검색하는 것을 두번째 경로로 지정하여 4개의 모션 벡터가 검색이 될 때까지 나선형으로 검색을 하게 된다. 나선형 검색을 통해 찾아진 모션 벡터는 아래 식 (3)과 같이 계산이 가능하다.

$$w^i = \frac{1}{\text{Number Of Passes}} \quad (3)$$

$$MV_n = \frac{\sum_{i=1}^4 w^i MV_n}{\sum_{i=1}^4 w^i}$$

식 (3)에서  $w^i$ 는 가중치로 검색 경로에 의해 검색된 모션벡터의 가중치를 결정한다. 가중치와 모션 벡터를 서로 곱한 다음 모두 더해 평균을 내면 비어있는 영역에 사용할 모션 벡터를 찾을 수 있다. 이는 이미지의 회전, 확대, 축소 등의 움직임이 있을 때 좀 더 강인한 모습을 보인다<sup>[10]</sup>.

### III. 제안 알고리즘

제안하는 알고리즘은 블록 움직임 예측을 이용해 구성된 새로운 프레임에 부족한 픽셀 데이터를 채워 넣는 픽셀 움직임 예측 알고리즘이다. 블록 움직임 예측에는 BMA를 사용한다. 그 후 계산된 모션 벡터를 이용해 단방향 모션 벡터를 이용한 화면 보간 기법으로 새로운 프레임을 생성하고 분산값을 이용해 픽셀 움직임 예측을 실시하여 필요한 픽셀 데이터를 모두 채운다.

#### 1. 블록 움직임 예측

제안 하는 알고리즘에서 사용하는 블록 움직임 예측은 단방향 모션 벡터 검색을 위한 블록 매칭 알고리즘이다. BMA의 효율적인 동작을 위한 여러 알고리즘이 있지만 본 논문에서는 BMA의 영향을 줄이기 위해 전역 탐색(Full Search)을 사용한다.

블록 매칭 알고리즘에서 사용하는 블록간 오류 값 계산은 SAD를 사용한다. 즉, 현재 프레임의 블록과 참조 프레임의 비교 블록간 SAD값이 가장 작은 블록 위치를 모션 벡터로 사용한다.

전후 프레임의 블록을 이용해 새로운 프레임을 생성한 뒤 블록이 겹치는 부분에 대한 처리를 한다.

#### 2. 픽셀 움직임 예측

먼저 전후 프레임의 블록들에 대해 분산값을 계산한다. 분산값은 프레임 N-1의 블록을 기준으로 프레임 N+1의 동일 위치 블록의 픽셀 분산값을 계산하고 분산값에 따라 적용시키는 픽셀 움직임 예측 알고리즘을 바꾼다. 분산값 임계치(Threshold)보다 높은 경우 픽셀 평균을 이용하고 낮을 경우 우세 움직임 벡터를 이용하여 픽셀값을 채운다.

##### 가. 비교 블록간 분산값 비교

비어있는 영역의 해당 블록 특성에 따라 서로 다른 알고리즘을 적용하여 보간된 영상의 화질을 향상시킬 수 있다. 제안된 알고리즘에서는 영상의 특성을 구분하기 위하여 블록의 분산값을 이용한다. 분산은 해당 데이터의 평균값에서 각 값들의 표준 편차의 제곱으로 계산을 하므로, 비교 블록별 분산값의 차이는 블록간 동일성을 확인해서 모션 벡터의 정확성을 구분할 수 있다. 비교 블록간 분산값의 차이가 작을수록 두 블록의 차이는 작다고 판단, 계산된 모션 벡터의 정확성을 보

장한다고 가정하고 이에 따라 비어 있는 영역의 해당 블록에 대해 우세 움직임 벡터를 이용하여 데이터를 채운다. 비교 블록간 분산값의 차이가 커질 경우 계산된 모션 벡터가 잘못된 모션 벡터라 판단을 하고 전후 프레임의 픽셀 평균값을 이용하여 비어있는 데이터를 채우게 된다. 두 알고리즘을 적용시키는 비교 블록간 분산값의 차이값 중 실험에 의하여 화질이 좋아지는 방향으로 임계치를 정한다.

나. 픽셀 평균

이는 움직이지 않는 화면에 대한 처리로서 움직이는 물체가 영역 밖으로 사라져 배경이 다시 나타났을 때 사용될 수 있다. 또한 움직임이 너무 커 제대로 모션 벡터를 찾지 못하거나 화면 안으로 새로운 물체가 나타나 이전에 없었던 모션이 새로 생기는 경우 움직임에 대해 화면의 밝기 변화를 줄일 수 있어 사람의 눈에 좀더 자연스러운 영상을 보여줄 수 있게 된다. 새로 생성되는 프레임의 비어있는 영역과 동일한 위치의 전후 프레임 픽셀 평균값을 이용해 비어있는 영역의 데이터를 삽입한다.

$$(f2_m, f2_n) = \frac{(f3_m, f3_n) + (f1_m, f1_n)}{2} \tag{4}$$

f1, f2, f3는 각각 프레임 N-1, N, N+1을 나타내고, (m,n)은 프레임에서의 위치정보를 나타낸다. 새로운 프레임 N에서 픽셀 정보가 부족한 위치를 (f2\_m, f2\_n)이라 할 때, 해당 위치의 픽셀값은 식(4)과 같이 프레임 N의 위치 (f2\_m, f2\_n)을 기준으로 전후 프레임 N-1과 N+1의 동일 위치에 있는 픽셀 값인 (f1\_m, f1\_n)과 (f3\_m, f3\_n)의 평

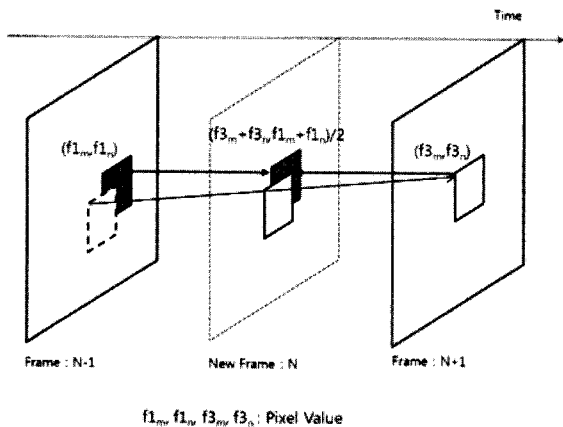


그림 3-1. 전후 프레임의 픽셀 평균  
Fig. 3-1. Pixel average of before and after frame.

균을 사용한다. 이는 그림 3-1에도 나와 있는데 프레임 N에서 검은색으로 채워진 블록과 흰색으로 채워진 블록 둘이 겹치는 부분이 있는데 겹치는 부분을 제외한 검은색 블록에 알고리즘을 적용시키게 된다.

다. 우세 움직임 벡터(Dominant MV)

새로 생성되는 프레임을 블록으로 나누었을 때 각 블록에 가장 큰 영향을 미친 모션 벡터를 계산 할 수 있다. 분산이 적은 경우 움직임이 있는 픽셀의 주변 픽셀도 비슷하게 움직일 가능성이 크므로 해당 블록에 대해 가장 큰 영향을 미친 모션 벡터와 비슷한 방향으로 움직임이 있었다고 할 수 있다.

$$\begin{aligned} (f2_m, f2_n) &= \frac{(f3_m, f3_n) + (f1_m, f1_n)}{2} \\ (f3_m, f3_n) &= (f3_{m+\frac{MV}{2}}, f3_{n+\frac{MV}{2}}) \\ (f1_m, f1_n) &= (f1_{m-\frac{MV}{2}}, f1_{n-\frac{MV}{2}}) \end{aligned} \tag{5}$$

현재 프레임 N의 블록 중 참조 프레임 N-1에서 찾은 모션 벡터를 MV라고 하고, 모션 벡터를 찾기 위한 f1, f3 위치를 각각 m', n'과 m'', n''이라 할 때, 값이 비어있는 곳을 계산하는 것은 식 (5)과 같다. 그림 3-2에서 새로운 프레임 N의 블록을 통과하는 모션 벡터는 여러 개가 있으므로 그 중 블록에 영향이 큰 모션 벡터를 하나 선택하여 그와 동일한 모션 벡터 MV를 적용하여 비어있는 공간에 채울 수 있는 픽셀 정보를 가져와 평균을 내서 값을 채우게 된다.

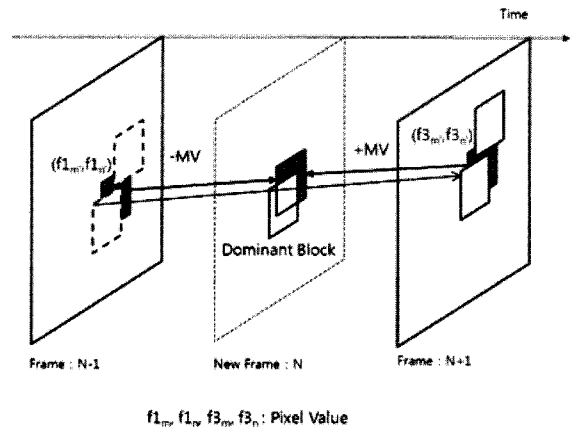


그림 3-2. 블록에 가장 영향력이 큰 모션 벡터 (우세 움직임 벡터, MV)  
Fig. 3-2. Most influential in the block motion vector (Dominant vector).

IV. 실험 결과

제안하는 알고리즘의 성능을 검증하기 위해 양방향 모션 벡터를 사용한 알고리즘, 기존 단방향 모션 벡터를 사용한 알고리즘과 제안한 알고리즘을 비교하였다. 실험은 352×288(CIF), 704×576(4CIF), 1280×720(HD), 1920×1080(Full HD) 크기의 영상을 이용해 영상의 100 프레임까지 짝수번째 프레임에 알고리즘을 적용시켜 새로운 영상을 만들었다. 알고리즘을 적용해 만들어진 영상들은 해당 영상의 원본 영상과 비교하여 PSNR을 계산하였다. 실험에 사용한 블록 매칭 알고리즘의 블록 크기는 8×8이고 탐색 범위는 +16픽셀에서 -16픽셀까지다. 블록크기를 8×8로 만든 이유는 블록 비교시 잔여 에너지 감소와 모션 벡터 추정의 정확성을 타협한 결과이기 때문이다<sup>[6]</sup>. 실험에 사용하는 영상은 Coastguard (CIF), City(4CIF), Stockholm(HD), Blue\_sky(Full HD)이다.

그림 4-1은 단방향 모션 벡터를 이용해 계산된 값을 사용하여 생성된 프레임이다. 화면 중간중간에 보이는 검은 부분이 모션 벡터를 찾지 못한 부분들이다. 이 부분을 채우는 알고리즘을 구분하여 실험을 진행 하였다.

또한 두 블록 간 분산 값 차의 절대 값은 실험에 의해 계산되었다. 실험에 사용된 영상은 Blue\_sky이고 각각 분산 값 차의 절대 값이 10, 20, 30, 40, 50, 60 값을 가지고 테스트한 결과가 표 4-1이다. 분산에 따른 모션 벡터의 영향력을 비교하기 위해 비어있는 영역이 포함된 블록에 영향이 큰 모션 벡터를 Dominant MV로 표시하였다. 모션 벡터는 단방향 움직임 추정으로 계산된 값을 사용했다.

알고리즘의 실험은 모션 벡터를 이용해 임시 중간 프

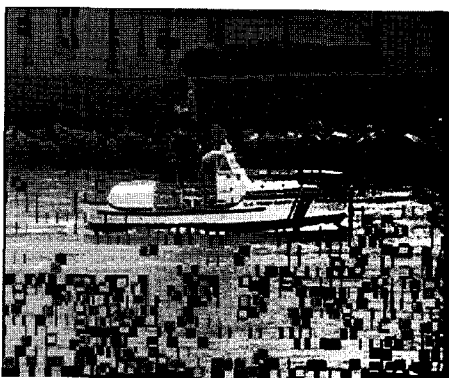


그림 4-1. 실험 중간 영상(Coastguard)  
Fig. 4-1. Experiment Mid-course image.

표 4-1. 분산값 차의 절대값에 따른 PSNR 변화

Table 4-1. PSNR changes according to the absolute value of variance differences.

분산값	10	20	30	40
Coastguard	40.6276	40.6241	40.6226	40.6211
City	40.3792	40.3872	40.3914	40.3940
Stockholm	40.9982	41.0019	41.0043	41.0057
Blue_sky	40.5340	40.5386	40.5416	40.5436
분산값	50	60	Dominant MV	
Coastguard	40.6208	40.6197	40.6154	
City	40.3955	40.3968	40.4014	
Stockholm	41.0067	41.0073	41.0099	
Blue_sky	40.5456	40.5473	40.5598	

레이미를 생성시킬 때 단방향 모션 벡터를 UD-BMA로 표시한다. 새롭게 생성되는 프레임을 기준으로 양측의 모션 벡터를 구하는 방법을 BI-BMA로 표시한다.

비어 있는 곳에 대해 적용시키는 픽셀 움직임 예측 알고리즘은 동일 위치의 픽셀값 평균을 이용하는 방법을 PixelAvg, 영향이 큰 벡터값을 이용한 방법을 Dominant MV라 표시하고 표에 정리하였다. 비어있는 영역에 대해 새롭게 생성되는 프레임을 기준으로 양방향의 모션 벡터를 구하는 방법을 Bilateral로 표시한다. 모션 벡터값으로 인해 이미지 경계를 넘어서는 값이 존재할 경우가 생기는데 이런 경우 모션 벡터를 사용하지 않고 PixelAvg를 통해 평균값으로 대체하였다. 이런 부분은 이미지 경계 부분에서 나타나게 된다.

표 4-2에서 UD-BMA는 비어있는 영역처리를 해야 하므로 비어있는 영역 처리 알고리즘을 더하기 기호로 묶어 놓았다. 표 4-2에서 보는 것처럼 BI-BMA는 UD-BMA에 전역 움직임 추정 알고리즘을 더한 것에 비해 PSNR이 낮게 측정됐다. 또한 표 4-2에서 PixelAvg과 Dominant MV를 비교해 보면, 해상도에 따라 다른 값을 보여주고 있다. CIF급 영상에서는 PixelAvg가 Dominant MV보다 PSNR이 같거나 조금 더 높은 수치를 보여주고 있다. 하지만 4CIF급 이상에서는 PixelAvg가 Dominant MV보다 PSNR이 낮아진다.

표 4-2에서 주변 픽셀 모션 벡터를 이용한 알고리즘을 Neighbor MV라 하고 본 논문에서 제안한 분산값

표 4-2. 알고리즘에 따른 실험 영상들의 50frame PSNR평균값  
Table 4-2. 50frame PSNR average of experiment sequences according to several algorithm.

실험 영상	BI-BMA <sup>[2]</sup>	UD-BMA +Neighbor MV <sup>[10]</sup>	UD-BMA + Bilateral
Coastguard	37.1387	40.6829	40.2691
City	36.8381	40.3852	40.0046
Stockholm	37.4449	40.9847	40.8193
Blue_sky	35.6932	40.4990	40.0741
실험 영상	UD-BMA + Dominant MV	UD-BMA + PixelAvg	UD-BMA + Variance
Coastguard	40.6154	40.6567	40.6226
City	40.4014	40.3274	40.3914
Stockholm	41.0099	40.9687	41.0043
Blue_sky	40.5598	40.4038	40.5416

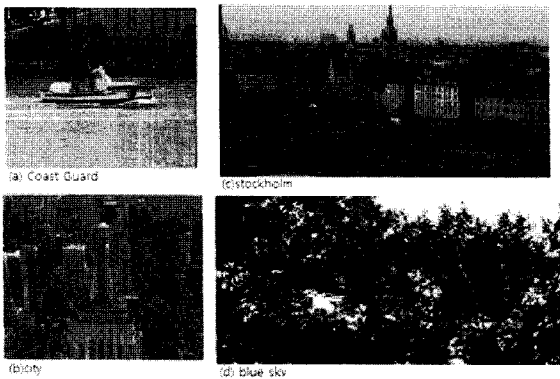


그림 4-3. 제안한 알고리즘의 결과 화면  
Fig. 4-3. Result image of proposed algorithm.

차이를 이용한 알고리즘을 Variance라 표시하였다. Variance의 경우 Dominant MV에 가까운 PSNR값을 보여주었다. 모든 해상도에서 0.01dB이하의 차이를 보였다. 비어있는 영역 주변 모션 벡터를 이용한 Neighbor MV의 경우 저해상도에서 PSNR수치는 가장 높게 나타났다. 그 차이는 PixelAvg, Dominant MV와 0.03dB~0.07dB정도였다. 하지만 고해상도이미지의 경우 Dominant MV, Variance이 미세하게 높은 수치를 나타내고 있으므로 거의 차이가 없다고 할 수 있다.

양방향 모션 벡터를 계산할 때 새롭게 생성되는 프레임의 블록 위치가 모션 벡터를 계산하는 기준이 되므로 모션 벡터값이 부정확해 질수 있지만 단방향 모션 벡터의 경우 화면상의 정보를 토대로 새로운 프레임에 사용할 모션 벡터를 찾게 되므로 단방향 예측을 통한 화면

보간 방법이 양방향 모션 벡터를 이용한 화면 보간 방법에 비해 좀 더 정확한 움직임 예측을 하고 있다고 말할 수 있다.

블록간 분산 값의 차를 이용하여 우세 움직임 벡터와 픽셀 평균을 선택적으로 적용하는 방법은 기존의 단방향 예측을 이용한 화면 보간 기법에서 사용하였던 Neighbor MV에 비해 연산에 대한 비용이 적다. Neighbor MV를 사용할 경우 비어 있는 영역에서 소용돌이 형태로 검색하여 주변에서 찾을 수 있는 모션 벡터 4개에 대해 거리에 따라 가중치를 부여하고, 가중치에 따른 계산값을 사용하므로 필요한 모션 벡터를 찾을 때까지 연산이 계속 된다. 또한 검색영역이 넓어 질수록 검색에 필요한 연산비용도 크게 늘어난다. 따라서 우세 움직임 벡터 하나와 분산값을 사용하는데 필요한 연산비용에 비해 Neighbor MV를 찾기 위한 연산 비용이 커지게 된다. 이런 특징은 모션 벡터가 큰 고해상도 영상에서 잘 나타난다. 그리고 블록간 분산값의 차를 이용하는 방법은 우세 움직임 벡터만을 사용한 때와 비슷한 성능을 보여주었다. 이는 모션 벡터를 기준에 의해 필요한 만큼만 사용함으로써 화질에 영향을 주고, 블록에 영향력이 큰 모션 벡터를 계산하는데 드는 비용을 절감할 수 있다. 블록들의 분산을 구하는 연산이 추가되므로 상황에 따라 추가적인 연산 비용이 더 들어갈 수 있다.

그리고 저해상도에서 평균값이 PSNR에 영향을 미치지 않지만 고해상도일수록 모션 벡터의 영향이 커진다고 할 수 있다. 저해상도와 고해상도에서 물체의 움직임이 물리적으로 동일할 때, 저해상도에서는 모션 벡터가 작게 나타나지만 고해상도에서는 저해상도에 비해 모션 벡터가 크게 나타나므로 우세 움직임 벡터를 사용하는 것이 더 좋은 화질을 보여주게 된다.

### V. 결 론

본 논문에서 제안하는 화면 보간 방법은 기존의 단방향 모션 벡터를 이용한 화면 보간 방법을 이용해 지역 움직임 추정을 통한 새로운 프레임을 생성하고, 비어 있는 영역에 대해 비교 블록간 분산값의 차이를 이용해 적용시키는 알고리즘을 달리하여 새로운 화면을 합성하는 방법을 제안하였다.

실험 결과 기존의 양방향 모션 벡터를 이용한 화면 보간 방법에 비해 전반적으로 제안하는 화면 보간 방법

이 우수한 PSNR값을 보였다. 특히 기존 단방향 예측을 통한 화면 보간 알고리즘에 비해 제안하는 알고리즘이 저 해상도에서는 비슷한 성능을 보였지만 고해상도로 갈수록 우수한 성능을 보였다. 또한 기존의 단방향 모션 벡터를 이용한 화면 보간 방법보다 PSNR값이 더 높은 것을 확인할 수 있었다.

제안하는 알고리즘에서 블록 매칭 알고리즘의 경우 연산량을 줄이면서 성능을 유지하는 다른 탐색 알고리즘을 적용시켜 연산량을 줄일 수 있고, 모션 벡터의 경우 영상 표준에 사용되는 모션 벡터를 그대로 이용할 수 있게 되므로 모션 벡터를 찾기 위한 비용을 절감하거나, 화질개선 알고리즘을 추가를 위한 비용으로 사용할 수 있게 되므로 연산의 복잡도 또는 화질에 대해 기존 알고리즘에 비해 장점을 가질 수 있게 된다.

향후 제안된 알고리즘에 대해 영상 표준의 모션 벡터를 이용한 실험과, 화질 개선 알고리즘을 적용시킨 테스트를 진행하여 제안 알고리즘의 화질 개선을 목표로 실험을 진행할 예정이다. 또한 제안된 알고리즘의 하드웨어 구현을 위한 알고리즘 최적화 및 간소화를 진행하여 하드웨어 구현시 알고리즘의 성능을 검증할 예정이다.

## 참 고 문 헌

- [1] Hilman, K. H. Park, Y. Kim, "Using motion-compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences", *IEEE transactions on Circuit and system for Video Technology*, vol10, issue6, pp.869-877, Sept. 2000.
- [2] B. Choi, S. Lee, "New frame rate up-conversion using bi-directional motion estimation", *IEEE Transactions on Consumer Electronics*, vol.46, Issue.3, pp603-609, Aug. 2000.
- [3] T. Ha, S. Lee and J. Kim, "Motion Compensated Frame Interpolation by new Block-based Motion Estimation Algorithm," *IEEE Transactions on Consumer Electronics*, vol.50, Issue.2, pp.752-759, May, 2004.
- [4] B. Choi, J. Han, C. Kim, and S. Ko, "Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," *IEEE Transactions on Circuit and system for Video Technology*, vol.17, issue.4, pp.407-416, Apr. 2007.
- [5] Y. Yang, Y. Tung and J. Wu, "Quality Enhancement of Frame Rate Up-Converted Video by Adaptive Frame Skip and Reliable Motion Extraction" *IEEE transactions on Circuit and system for Video Technology*, vol.17, issue.12, pp.1700-1713, Jan. 2007.
- [6] Z. Gan, L. Qi and X. Zhu, "Motion compensated frame interpolation based on H.264 decoder," *Electronics Letters*, vol.43, Issue.2, pp.96-98, Jan. 2007.
- [7] 남현우, "개선된 크로스-육각 패턴을 이용한 고속 블록 정합 움직임 추정 알고리즘", *한국컴퓨터정보학회 논문지*, vol.11, no.4, pp77-85, 2007.09
- [8] A. Huang, T. Nguyen, "A multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation", *IEEE transactions on image processing*, vol.17, issue.5, pp694-708, may. 2008.
- [9] S. Lee, O. Kwon, R. Park, "Weighted-adaptive Motion-Compensated Frame Rate Up-Conversion", *IEEE Transactions on Consumer Electronics*, vol.49, Issue.3, pp.485-492, Aug. 2003.
- [10] B. Jeon, G. Lee, S. Lee, R. Park, "Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure", *IEEE Transactions on Consumer Electronics*, Vol.49, Issue.3, pp499-508, Aug. 2003.
- [11] 하태현, 이성주, 김성식, 성준호, 김재석, "중첩 블록 기반 움직임 추정에 의한 중간 영상 합성기법", *한국방송공학회논문지*, 제 9권, 제 1호, pp54-63, mar. 2004.
- [12] N. Mishima, G. Itoh, "Novel Frame Interpolation Method For Hold-type Displays", *ICIP04 International Conference on image processing*, Vol.3, pp1473-1476, Singapore, Oct.2004
- [13] D. Lim, Y. Ho, "A Fast Block Matching Motion Estimation Algorithm Using Optimal Search Patterns", *Proceeding of SPIE*, Vol.4310, pp.767-775, Jul. 2001.



저 자 소 개



최 승 현(학생회원)  
 2006년 광운대학교 컴퓨터공학과  
 학사 졸업.  
 2008년 광운대학교 컴퓨터공학과  
 석사 졸업.  
 2009년 광운대학교 컴퓨터공학과  
 박사 재학중.

<주관심분야 : 미디어 프로세서 및 SOC설계, 영  
 상신호처리 >



이 성 원(정회원)  
 1988년 서울대학교 제어계측  
 공학과 졸업(공학사).  
 1990년 서울대학교 제어계측  
 공학과 석사졸업  
 2003년 University of Southern  
 California 전기공학과  
 박사졸업

2009년 현재 광운대학교 전자정보공과대학  
 컴퓨터공학과 교수.

<주관심분야 : 미디어프로세서 및 SOC설계, 영상  
 신호처리, Power-Aware Computing>