

특집논문 (Special Paper)

방송공학회논문지 제18권 제3호, 2013년 5월 (JBE Vol. 18, No. 3, May 2013)

<http://dx.doi.org/10.5909/JBE.2013.18.3.342>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

## 블록의 성질과 프레임 움직임을 고려한 적응적 확장 블록을 사용하는 프레임을 증강 기법

박 대 준<sup>a)</sup>, 정 제 창<sup>a)†</sup>

### Adaptive Extended Bilateral Motion Estimation Considering Block Type and Frame Motion Activity

Daejun Park<sup>a)</sup> and Jechang Jeong<sup>a)†</sup>

#### 요 약

본 논문에서는 적응적인 확장 블록을 사용하는 프레임을 증강 기법인 AEBME (Adaptive Extended Bilateral Motion Estimation)을 제안하고자 한다. 기존의 EBME (Extended Bilateral Motion Estimation) 알고리즘은 동일한 구역에 두 번의 움직임 예측을 수행함으로써 높은 계산량이 요구되었다. 본 논문에서는 영상의 edge 정보를 활용한 블록 유형의 일치 유무를 고려하여 EBME 수행여부를 결정함으로써 움직임 예측 과정을 보다 빠르게 수행하도록 하였다. 움직임 벡터 평활화 과정이 적용되어 움직임 벡터 필드 내의 이상 벡터를 찾아 수정한다. 최종적으로 OBMC (Overlapped Block Motion Compensation)와 MCFI (Motion Compensated Frame Interpolation)이 프레임 움직임의 성질에 따라 적용되어 중간 프레임을 보간하게 된다. 실험 결과를 통해 제안하는 알고리즘이 기존의 알고리즘인 EBME에 비해 향상된 성능과 빠른 속도를 보임을 알 수 있다.

#### Abstract

In this paper, a novel frame rate up conversion (FRUC) algorithm using adaptive extended bilateral motion estimation (AEBME) is proposed. Conventionally, extended bilateral motion estimation (EBME) conducts dual motion estimation (ME) processes on the same region, therefore involves high complexity. However, in this proposed scheme, a novel block type matching procedure is suggested to accelerate the ME procedure. We calculate the edge information using sobel mask, and the calculated edge information is used in block type matching procedure. Based on the block type matching, decision will be made whether to use EBME. Motion vector smoothing (MVS) is adopted to detect outliers and correct outliers in the motion vector field. Finally, overlapped block motion compensation (OBMC) and motion compensated frame interpolation (MCFI) are adopted to interpolate the intermediate frame in which OBMC is employed adaptively based on frame motion activity. Experimental results show that this proposed algorithm has outstanding performance and fast computation comparing with EBME.

Keyword : Frame rate up conversion, motion estimation, overlapped block motion compensation, frame motion activity, block type matching

## 1. 서론

프레임율 증강 기법 (frame rate up conversion: FRUC)은 다양한 디스플레이 장치에서 프레임율을 증가시키기 위한 목적으로 사용되고 있다. LCD (liquid crystal display)는 큰 움직임이 있는 영상에서 모션 블러 현상 (motion blur effect)을 겪는다. 이는 LCD 같은 저장 방식의 디스플레이 (hold-type display)의 특성상 빛을 오랫동안 유지시키는 특성에 기인한다. 시청자는 이전 프레임의 잔상이 한동안 남아있는 LCD의 특성을 원치 않는다. 프레임율 증강 기법은 이러한 문제에 대처하는 가장 이상적인 기술이다. 프레임율을 두 배로 증가시킴으로 인해 모션 블러 현상을 극단적으로 줄여준다<sup>[1]</sup>. 프레임율 증강 기법은 제한된 대역폭 내의 환경에서도 유용하게 사용된다. 협대역폭 (narrow bandwidth)에서 부호기는 최대한 데이터 전송량을 줄여야 하며, 이는 전송하는 영상의 홀수 혹은 짝수 프레임만 전송함으로써 가능해진다. 복호기에서는 프레임율 증강 기법을 통해 사라진 중간 프레임을 재생성하게 된다.

다양한 프레임율 증강 기법들이 제안되어왔다. 물체의 움직임을 고려하지 않은 기법들로는 프레임 반복 기법, 프레임 평활화 방법이 있다. 이는 소프트웨어, 하드웨어적으로 구현이 쉬우나, 잔상 효과가 존재한다. 이후에 잔상 효과를 줄이기 위해 움직임 보상 기법이 사용된 보간 기법 (motion compensation frame rate up conversion: MC-FRUC)이 제안되어왔다<sup>[2][5][6]</sup>. 해당 기법들은 블록 정합 기법 (block matching algorithm: BMA)을 사용한 움직임 예측 기법과 움직임 보상 기법들로 구성된다.

MC-FRUC는 참 움직임을 고려한 접근 방법과 그렇지 않은 경우로 나뉘는데, 본 논문에서 사용하는 중복 격자 방식을 이용한 움직임 예측 기법 (extended bilateral motion

estimation: EBME)은 참 움직임을 고려하지 않는다<sup>[2]</sup>. EBME는 현재 프레임과 이전 프레임을 이용하여 두 번의 전역 탐색 (full search)을 수행하기 때문에 매우 느리며, 실시간 응용에 적합하지 않다. 반면에 서로 다른 중복된 격자를 통해 두 번의 움직임 예측을 수행하고, 추가적인 움직임 벡터 평활화 과정을 거침으로써 움직임 벡터의 정확도를 높여준다.

제안하는 AEBME (adaptive extended bilateral motion estimation) 알고리즘은 기존의 EBME 알고리즘을 edge 정보를 통한 블록 유형의 일치 유무에 따라 적응적으로 적용시킴으로써 움직임 예측 과정의 수행시간을 줄였다. 반복적인 움직임 벡터 평활화 과정이 적용되어 움직임 벡터 필드 내의 이상 벡터를 찾아 올바른 벡터로 수정하게 된다. 프레임 보간 과정에서 중첩 블록 움직임 보상 (overlapped block motion compensation: OBMC) 과정이 프레임 움직임의 성질에 따라 적용되어, 움직임 벡터의 부정확함으로 인해 발생하는 블록화 현상 (blocking artifact)을 줄여준다<sup>[4]</sup>. 최종적으로 움직임 보상 프레임율 증강 기법 (motion compensated frame interpolation: MCFI)을 통해 중간 프레임을 생성한다.

본 논문의 구성은 다음과 같다. II장에서는 EBME를 사용한 MC-FRUC 알고리즘에 대하여 간략히 소개를 하고, III장에서는 제안하는 알고리즘을 상세히 설명한다. 그리고 IV장에서는 실험결과와 실험환경을 제시하고, 마지막으로 V장에 결론을 맺는다.

## II. EBME를 사용한 MC-FRUC

### 1. 쌍방향 움직임 예측

그림 1은 쌍방향 움직임 예측 (bilateral motion estima

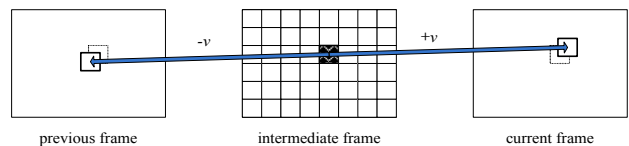


그림 1. 쌍방향 움직임 예측 설명  
 Fig. 1. Illustration of bilateral motion estimation

a) 한양대학교 전자컴퓨터통신공학과 (Department of Electronics and Computer Engineering, Hanyang University)

‡ Corresponding Author : 정제창(Jechang Jeong)

E-mail: jjeong@hanyang.ac.kr

Tel: +82-2-2220-4370

※ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음(NIPA-2013-H0301-13-1011)

· Manuscript received March 29, 2013 Revised May 23, 2013 Accepted May 23, 2013

tion)을 나타내며, 물체의 움직임이 중간 프레임의 시점으로 쌍방향으로 대칭적이라는 가정을 갖는다.

기존의 단방향 움직임 예측을 통한 BMA는 움직임 보상 과정에서 홀과 폐쇄 영역 (occluded region)으로 인해 성능 저하가 발생한다. 그러나 쌍방향 움직임 예측과 보상은 중간 프레임을 블록 단위로 나누어 이전 프레임과 현재 프레임의 블록들을 이용하여 탐색을 진행하기 때문에, 복원 후에 홀과 폐쇄 영역이 존재하지 않는다. 쌍방향의 블록들을 이용하여 (1)의 SBAD (the sum of bilateral absolute differences)를 계산하고, SBAD를 최소로 하는 움직임 벡터를 찾는다.

$$SBAD(v_x, v_y) = \sum_{v_x \in SR} \sum_{v_y \in SR} |f_{n-1}(x-v_x, y-v_y) - f_n(x+v_x, y+v_y)|$$

$$v = \arg \min_{(v_x, v_y) \in SR} SBAD(v_x, v_y)$$
(1)

여기서 (vx,vy)는 움직임 벡터 후보를 나타내며, fn-1과 fn은 이전, 현재 프레임을 나타낸다. v는 최종적인 움직임 벡터이다.

### 2. EBME (Extended Bilateral Motion Estimation)

쌍방향 움직임 예측의 계산량은 기존의 블록 정합 기법에 비해 1/4 수준이다. 이는 탐색 범위가 1/4로 감소함에

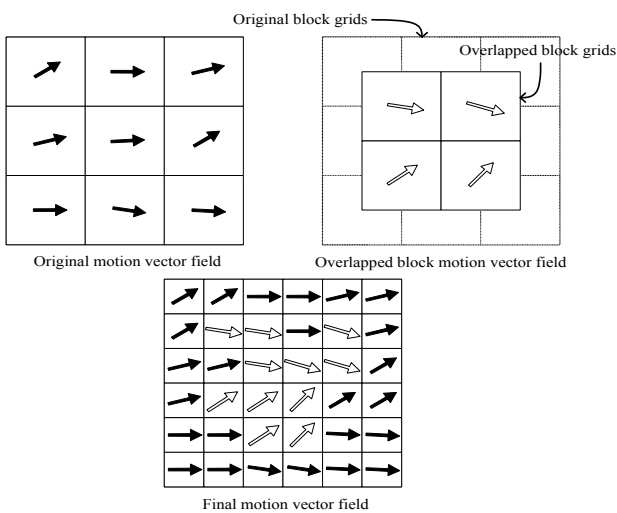


그림 2. EBME 설명  
Fig. 2. Illustration of the EBME

기인한다. 기존의 방법에 비해 계산량이 줄었을 뿐, 여전히 상당히 높은 계산량을 요구한다. 또한 물체의 움직임이 대칭적이지 않을 시에 참 움직임을 찾을 수가 없다. 따라서 EBME는 쌍방향 움직임 예측을 중복된 격자를 통해 추가적인 움직임 예측을 수행함으로써, 정확도를 높인다. 그림 2에 EBME가 수행되어 움직임 벡터 필드의 정확도가 높아지는 과정을 제시하였다. 원래의 격자를 통해 구한 SBAD와 중복된 격자를 통해 구한 SBAD를 비교하여, 더 작은 SBAD를 갖는 격자의 움직임 벡터를 사용함으로써 움직임 벡터 필드를 수정한다.

### 3. 반복적인 움직임 벡터 평활화

이상 벡터는 블록화 현상을 야기하며, 영상의 화질을 저하시킨다. 따라서 움직임 벡터 필드 내에서 이상 벡터를 찾아 제거하기 위해 움직임 벡터 평활화 과정이 도입되었다.

$$v_m = \frac{1}{9} \sum_{i=1}^9 v_i$$

$$D_i = abs(v_m - v_i)$$

$$D_n = \frac{1}{8} \sum_{i=2}^9 D_i$$
(2)

(2)에서, vm은 v1과 주변 이웃 블록들의 움직임 벡터들의 평균 움직임 벡터이다. Di는 평균 움직임 벡터 vm과 i 번째 블록의 움직임 벡터 vi와의 절대 차분치를 나타낸다. Dn은 주변 블록들의 Di 평균이다. 만약 D1이 Dn보다 클 경우에, v1은 이상 벡터로 분류된다.

한 프레임 내의 모든 이상 벡터를 찾아낸 후에 평활화가 수행된다. 움직임 벡터 평활화 과정은 그림 3을 따른다. 먼저 주변 8개의 움직임 벡터가 모두 이상 벡터가 아니고, 현재 블록의 움직임 벡터만이 이상 벡터인 경우를 찾아내 Median 필터를 취해 v1을 수정한다. 다음으로 현재 블록의 움직임 벡터가 이상 벡터일 때, 주변 8개의 움직임 벡터 중 한 개만 이상 벡터인 경우를 찾아내 Median 필터를 취해 v1을 수정한다. 현재 블록의 움직임 벡터가 이상 벡터일 때, 주변 움직임 벡터 중 이상 벡터가 2개, 3개인 경우를 거쳐, 주변 모두가 이상 벡터인 경우까지 평활화 과정을 반복한

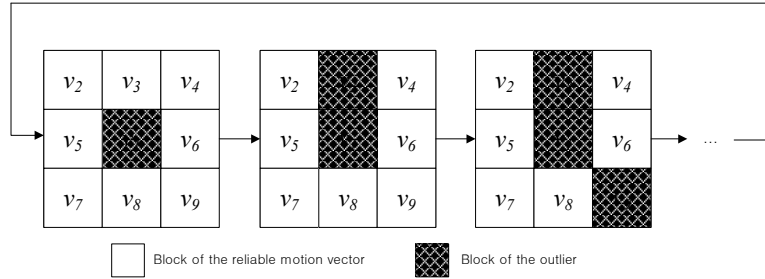


그림 3. 반복적인 움직임 벡터 평활화  
 Fig. 3. Recursive motion vector smoothing

다. 다시 처음으로 돌아가 모든 이상 벡터가 수정될 때까지 평활화 과정을 반복한다.

#### 4. 중첩 블록 움직임 보상 (OBMC)

OBMC는 블록화 현상을 극단적으로 줄여주는 방법으로, 대부분의 영상에서 화질 개선을 보여준다. 그림 4의 Bilinear 윈도우를 씌워 블록화 현상을 없애며, 윈도우 계수는 (3)을 통해 구한다.  $N \times N$ 은 블록의 크기를 나타내며,  $2N \times 2N$  크기의 윈도우를 만들어, 주변 블록들과 중첩하여 씌운다. 윈도우는 블록의 경계면을 부드럽게 만들어 블록화 현상을 없앤다.

$$w(u,v) = w_u \cdot w_v, w_u = \begin{cases} \frac{1}{N}(u + \frac{1}{2}) & \text{for } u = 0, \dots, N-1 \\ w_{(2N-1)-u} & \text{for } u = N, \dots, 2N-1 \end{cases} \quad (3)$$

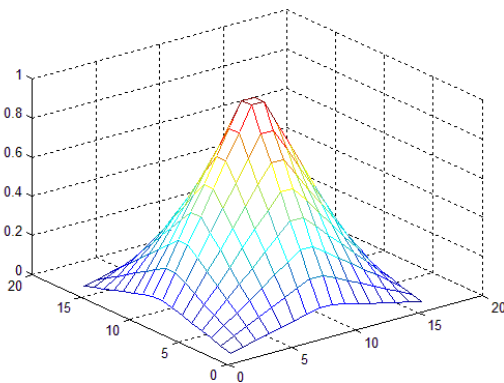


그림 4. Bilinear 윈도우  
 Fig. 4. Bilinear window

#### 5. 움직임 보상을 이용한 프레임 보간

(4)의 MCFI 과정을 통해 중간 프레임을 보간한다.  $(v_x, final, v_y, final)$ 은 반복적인 움직임 벡터 평활화 과정을 통해 결정된 최종 움직임 벡터이다.  $n-1, n$  프레임을 통해 중간 프레임인  $n-1/2$  프레임을 보간한다.

$$f_{n-\frac{1}{2}} = \frac{1}{2}f_{n-1}(x - v_{x,final}, y - v_{y,final}) + f_n(x + v_{x,final}, y + v_{y,final}) \quad (4)$$

### III. 제안하는 알고리즘

제안하는 AEBME 알고리즘의 흐름도는 그림 5에 제시한다. 블록 유형의 일치 유무에 따라 EBME를 수행여부를 결정하고, 프레임 움직임의 성질에 따라 OBMC를 적용여부를 결정한다. EBME에서 처음의 탐색이 정확하다고 판단이 될 경우, 추가적인 두 번째 탐색은 생략하여 계산량을 극단적으로 줄인다. OBMC가 대부분의 영상에 좋은 결과를 나타내나, 정적인 영상에 대해서는 평활화를 통한 화질 저하를 가져온다. 정적인 영상의 경우 대부분의 움직임 벡터가 영벡터  $(0,0)$ 이며, 블록화 현상이 거의 생기지 않으므로 OBMC의 적용이 필수적이지 않다. 반면에 동적인 영상의 경우 블록화 현상이 많이 일어나므로 OBMC 과정이 필수이다. 따라서 프레임 움직임의 성질에 따라 OBMC를 적용하여야 한다.

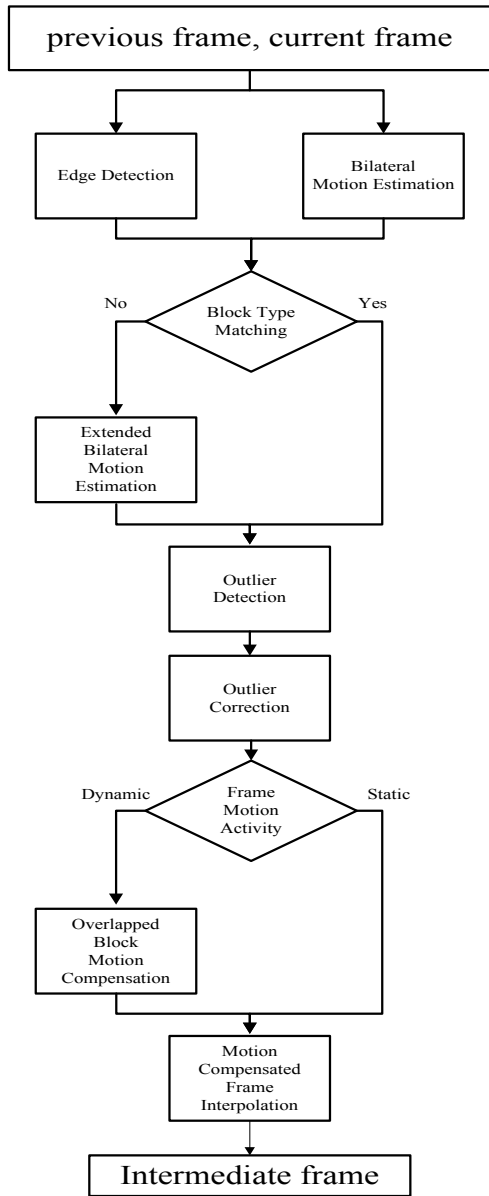


그림 5. 제안하는 AEBME 알고리즘의 흐름도  
 Fig. 5. Flowchart of the proposed AEBME scheme

1. 블록 유형의 일치

참 벡터가 아닌 경우에 이전 참조 프레임과 현재 참조 프레임에서 서로 다른 물체를 인식하게 된다. 참 벡터를 찾기 위해서 두 물체가 같은 물체인지 확인하는 과정을 거친

다. 최소 SBAD를 갖는 지점을 통해 얻은 움직임 벡터가 반드시 참 벡터는 아니다. 따라서 블록 유형 일치 유무를 통해 각 블록들의 움직임 벡터가 올바른지 판단한다.

우선 쌍방향 움직임 예측을 통해 움직임 벡터와 참조 블록들을 찾아낸다. 양측의 참조 블록들에 대해 다음과 같은 과정을 수행한다. 소벨 마스크를 통해 얻은 edge 정보를 통해, 참조 블록의 평균 edge를 구한다. 블록 내의 픽셀들의 edge 값이 평균 edge보다 크면 edge 픽셀로, 작으면 flat 픽셀로 분류한다. 블록 내의 edge 픽셀 비율이 T1 보다 크면 edge 블록으로, 작으면 flat 블록으로 분류한다.

참조 블록들의 블록 유형이 일치하지 않을 경우에는 절대 참 벡터가 될 수 없으므로 추가적인 예측이 필요하다고 고려되어 EBME를 수행하여 움직임 벡터의 정확도를 높인다.

2. 프레임 움직임의 성질 확인

정적인 영상은 움직임벡터의 대부분이 영벡터라는 특성을 통해, 프레임 평균 움직임 벡터가 0에 가까울 것이라는 결론을 얻었다. 따라서 프레임 평균 움직임 벡터의 크기에 따라 OBMC 적용 여부를 결정한다. 프레임 평균 움직임 벡터가 T2보다 큰 경우에는 동적인 프레임이라는 판단을 하여 OBMC를 적용한다. 반대의 경우에는 정적인 프레임이라는 판단을 하여 OBMC 과정을 건너뛴다. 최종적으로 MCFI를 통해 중간 프레임을 생성한다.

$$|v_i| = |v_{x,i}| + |v_{y,i}|$$

$$v_{avg} = \frac{1}{M} \sum_{i \in frame} |v_i| \tag{5}$$

(5)는 프레임 평균 움직임 벡터를 구하는 과정으로, M은 프레임 내의 블록의 총 개수를 의미한다. 프레임 내의 모든 블록에 대해 움직임 벡터의 절대치를 평균한다.

IV. 실험 결과

실험은 CIF(352x288) 영상들의 51개 홀수 프레임이 이

용하여 진행하였고, 알고리즘 수행을 통해 50개의 짝수 프레임이 생성되었다. 원본 짝수 프레임과 알고리즘 수행을 통해 생성된 짝수 프레임 사이의 PSNR (peak signal to noise ratio)을 계산하여 객관적 화질 비교의 척도로 삼았다. 비교 알고리즘으로 EBME를 사용하였다. EBME에 비해 객관적 화질이 얼마나 좋은지, 얼마나 빨라졌는지를 측정하고, 블록 유형 일치 판단 알고리즘이 실험 영상마다 어떤 결과를 보이는지 제시하였다.

실험을 위한 설정 환경은 다음과 같다. 기본 블록 크기를 16x16으로 하였고, 탐색 범위를 -8~+8로 하였다. 쌍방향 움직임 예측 후 진행되는 EBME와 반복적인 움직임 벡터 평활화 과정에서는 8x8 크기의 블록을 사용하였고, 따라서 OBMC에서 사용되는 bilinear 윈도우 크기 N은 16으로 설정하였다. 블록 내 edge 픽셀 비율 T1은 0.6으로, 프레임 움직임 성질 확인 척도인 T2는 0.5로 선택하였다. 실험 영상으로 Akiyo, Coastguard, Container, Flower, Football, Foreman, Mobile, Stefan, Table 그리고 Tempete를 사용하였다.

객관적 화질 평가를 위해 주로 PSNR을 척도로 사용한다. 표 1에 제시한 결과에 따르면, AEBME 알고리즘의 화질은 EBME와 비슷하다. 미약한 감소는 블록 유형 일치 여부에 따른 EBME 과정의 생략에 기인하며, PSNR 향상은 프레임 움직임을 고려하여 정적인 영상에 OBMC를 적용하지 않은데에 기인한다.

표 2는 각 영상별로 평균적으로 몇 번의 EBME가 추가적으로 수행되었는지를 나타낸다. 즉, 블록 유형 불일치가 영상별로 얼마나 일어나는지를 측정하였다. 정적인 영상에서는 대부분이 일치하는데 반해, 동적인 영상에서는 불일치가 정적인 영상에 비해서는 많았으나, 상당 부분에서 블록 유형이 일치함을 보여준다. 따라서 추가적인 EBME 과정이 필요치 않은 부분이 많으며, 이를 통해 움직임 예측 계산량이 극적으로 줄어들음을 알 수 있다.

총 수행 시간을 표 3에 제시하였다. 추가적으로 여러 과정이 추가되었음에도 불구하고 AEBME는 EBME에 비해 2배에 가깝게 속도가 빨라졌음을 알 수 있다. 이는 추가적인 EBME 과정이 블록 유형 일치 확인 과정을 통해 대부분이 생략되었기 때문이다.

표 1. PSNR 비교  
 Table 1. PSNR comparison

Sequence	EBME(dB)	AEBME(dB)	Difference
Akiyo	46.119	46.917	+0.798
Coastguard	30.401	30.382	-0.019
Container	44.073	44.675	+0.602
Flower	30.915	30.616	-0.299
Football	23.23	23.185	-0.045
Foreman	33.911	33.838	-0.073
Mobile	27.154	27.145	-0.009
Stefan	25.064	25.029	-0.035
Table	30.326	30.295	-0.031
Tempete	29.457	29.401	-0.056
Average	32.065	32.148	+0.083

표 2. 블록 유형 불일치 횟수  
 Table 2. The number of block type mismatch blocks

Sequence	EBME	AEBME	Gain(times)
Akiyo	357	4.469	79.884
Coastguard	357	19.245	18.550
Container	357	2.612	136.677
Flower	357	3.674	97.170
Football	357	24.061	14.837
Foreman	357	5.102	69.973
Mobile	357	8.265	43.194
Stefan	357	27.163	13.143
Table	357	10.735	33.256
Tempete	357	10.633	33.575
Average	357	11.596	30.786

표 3. 수행 시간  
 Table 3. Computation times

Sequence	EBME(sec)	AEBME(sec)	Gain(times)
Akiyo	8.078	4.263	1.895
Coastguard	8.043	4.507	1.785
Container	8.67	4.572	1.896
Flower	8.647	4.678	1.848
Football	8.384	4.775	1.756
Foreman	7.931	4.321	1.835
Mobile	9.218	5.04	1.829
Stefan	8.121	4.664	1.741
Table	8.941	4.931	1.813
Tempete	8.325	4.54	1.834
Average	8.436	4.629	1.823

## V. 결론

본 논문에서는 블록 유형 일치 여부와 프레임 움직임 성질을 고려한 효과적인 프레임율 증강 기법인 AEBME 알고리즘을 제안하였다. 블록 유형 일치 여부는 edge 정보를 활용하여 추가적인 탐색을 줄여줌으로써 불필요한 계산을 상당량 감소시켰다. 반복적인 움직임 벡터 평활화를 통해 이상 벡터를 찾아, 수정함으로써 움직임 벡터의 정확도를 높였고, 프레임 움직임 성질을 고려하여 OBMC를 선택적으로 사용함으로써 정적인 영상에서의 화질 저하를 줄였다. 최종적으로 MCFI를 통해 중간 프레임을 복원하였다.

AEBME 알고리즘은 기존의 알고리즘에 비해 객관적 화질을 향상 혹은 유지시켰고, 블록 유형 일치 여부를 판단함으로써 수행시간은 향상시켰다. 실험 결과를 통해 제안하는 AEBME 알고리즘이 기존의 EBME 알고리즘에 비해 상당히 성능을 향상시켰음을 알 수 있다.

## 참고 문헌

- [1] S. H. Chan, T. X. Wu and T. Q. Nguyen, "Comparison of two frame conversion schemes for reducing LCD motion blurs," IEEE Signal Processing Letters, vol. 17, no.9, pp. 782-786, Sept. 2010.
- [2] S. J. Kang, K. R. Cho and Y. H. Kim, "Motion compensated frame rate up-conversion using extended bilateral motion estimation," IEEE Trans. Consumer Electron., vol. 53, no. 4, pp. 1759-1767, Nov. 2007.
- [3] S. J. Kang, D. G. Yoo and S. K. Lee, "Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion," IEEE Trans. Consumer Electron., vol. 54, no. 4, pp. 1830-1838, Nov. 2008.
- [4] M. T. Orchard and C. J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach," IEEE Trans. Image Processing, vol. 9, no. 5, pp.1509-1521, Sept. 1994.
- [5] B. D. Choi, J. W. Han, C. S. Kim and S. J. Ko, "Motion-Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion compensation," IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no. 4, pp. 407-416, Apr. 2007.
- [6] D. Wang, A. Vincent, P. Blanchfield and R. Klepko, "Motion-Compensated Frame Rate Up-Conversion-Part II: New Algorithms for Frame Interpolation," IEEE Trans. Broadcasting, vol. 56, no.2, pp. 142-149, Jun. 2007.
- [7] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd Edition, Prentice Hall, New Jersey, 2010.

## 저 자 소 개

### 박 대 준



- 2011년 2월 : 한양대학교 전자통신컴퓨터공학부 졸업 (학사)
- 2011년 3월 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 석박사통합과정
- 주관심분야 : 비디오 압축, 영상처리

### 정 제 창



- 1980년 2월 : 서울대학교 전기공학과 졸업
- 1982년 2월 : KAIST 전기전자공학과 석사
- 1990년 : 미국 미시간대학 전기공학과 공학박사
- 1980년 ~ 1986년 : KBS 기술연구소 연구원 (디지털 및 뉴미디어 연구)
- 1990년 ~ 1991년 : 미국 미시간대학 전기공학과 연구교수 (영상 및 신호처리 연구)
- 1991년 ~ 1995년 : 삼성전자 멀티미디어 연구소 (MPEG, HDTV, 멀티미디어 연구)
- 1995년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 교수 (영상통신 및 신호처리 연구실)
- 1990년 12월 : 정보통신부장관상 수상
- 1998년 11월 : 과학기술자상 수상
- 2007년 : IEEE Chester Sall Award 수상
- 2008년 : ETRI Journal Paper Award 수상
- 2011년 5월 : 제 46회 발명의 날 녹조근정훈장 수훈
- 주관심분야 : 영상처리, 영상압축, 3DTV