

논문 2015-52-5-9

비정방형 블록을 이용한 보간 프레임의 정확도 향상 기법

(Accuracy Improvement of Frame Interpolation Algorithm using
Wedge-shaped Block Partitioning)

정재현*, 정호선*, 선우명훈**

(Jae Heon Jeong, Ho Sun Jung, and Myung Hoon Sunwoo[©])

요약

기존의 프레임율 증가 변환 알고리즘의 경우 움직임 추정을 위해 정방형의 블록을 사용하며, 임의의 형태를 갖는 물체의 경계를 정확하게 구분할 수 없어 정확한 움직임 벡터를 찾는 데 한계가 있었다. 이에 반해 제안하는 비정방형 블록 분할 알고리즘은 텍스처 정보를 이용하여 블록을 두 개의 비정방형 블록으로 분할하여 실제 물체의 경계를 찾는 데 적합하다. 또한 주변 블록의 분할 방법에 따라 선택적으로 움직임 벡터를 참조하는 알고리즘을 제안함으로써, 아웃라이어(Outlier)를 방지하고 실제 움직임 벡터를 찾는 데 탁월한 성능을 보인다. 실험 결과 제안하는 알고리즘은 기존 대비 평균 20-40%의 연산량으로 최대 2.03dB 높은 PSNR 성능을 보이는 것을 확인하였으며, SSIM 비교에서는 최대 0.0214의 향상을 확인하였다.

Abstract

This paper presents a novel frame rate up-conversion (FRUC) algorithm. Existing algorithms, in general, employ rectangular blocks for motion estimation and arbitrary shape of an actual object region cannot be precisely represented. On the other hand, the proposed wedge-shaped block partitioning algorithm partitions a rectangular block into two wedge-shaped blocks using the texture information, which makes better approximation for an actual object region. The wedge-shaped block partitioning algorithm as well as the adaptive motion vector prediction algorithm is used to reliably estimate the actual motion. Experimental results show that the proposed FRUC algorithm is superior to existing algorithms up to 1.988dB in PSNR and 0.0167 in SSIM comparisons.

Keywords: Frame rate up-conversion (FRUC), motion estimation (ME), block partitioning, motion vector smoothing, motion compensated frame interpolation (MCFI)

I. 서론

과거 프레임율 변환 기법은 미국의 NTSC(National

television systems committee) 및 유럽의 PAL(Phase alternation by line) 방식과 같이, 서로 다른 프레임율을 가지는 다양한 디스플레이 포맷 간의 변환을 위해 주로 사용되었다^[1]. 하지만 최근 UHD TV부터 스마트폰에 이르기까지 디스플레이 디바이스들의 보급이 급격히 증가하면서, 제한된 대역폭과 전력 소모 환경에서 영상의 품질을 향상시키기 위한 방법으로 프레임율 변환 알고리즘의 응용이 확대되고 있다^[2-3].

프레임율 증가 변환은 동영상 프레임 사이에 새로운 가상 프레임, 즉 보간 프레임을 삽입하여 영상의 연속성을 높이는 기술이다. 보간 프레임은 시간적으로 인접한 프레임들의 정보를 활용하여 생성되며, 정확한 보간

* 학생회원, ** 정회원, 아주대학교 전자공학부
(Department of Electrical and Computer
Engineering, Ajou University)

© Corresponding Author(E-mail: sunwoo@ajou.ac.kr)

※ 본 논문은 미래창조과학부 중견연구자지원사업
[2014R1A2A2A01002952, 미래형 실감미디어 지원을
위한 영상처리 전용 프로세서 연구]의 지원에 의하여
연구되었음.

Received ; January 14, 2015 Revised ; March 26, 2015

Accepted ; April 22, 2015

프레임의 생성을 통해 제한된 대역폭 및 디스플레이 환경으로 인해 발생하는 Motion Blur나 Motion Judder와 같은 영상왜곡을 줄일 수 있다^[4~5].

이러한 프레임 보간 과정은 크게 움직임 추정(Motion estimation)과 움직임 보상 보간(Motion compensated interpolation)의 두 단계로 이루어진다. 움직임 추정의 경우 연속적인 프레임들 사이에서 물체의 움직임 벡터를 추정하는 과정이며, 움직임 보상 과정은 움직임 벡터를 이용하여 새로운 프레임을 생성하는 과정이다.

기존의 알고리즘^[6~8]의 경우 현재 처리중인 블록을 단순히 정방형의 하위 블록으로 분할하여 움직임 추정을 수행한다. 이는 임의의 형태를 갖는 물체의 경계를 세밀하게 표현하기 어려우며, 화면 내 다수의 물체가 큰 움직임을 가질 경우 정확한 움직임 추정이 어렵다. 따라서 기존의 정방형 블록 분할 방법은 물체의 정확한 움직임을 찾아야 하는 프레임율 증가변환에는 적합하지 않다.

본 논문에서는 물체의 경계면 표시에 적합한 블록 분할 방법과, 적응적 움직임 벡터 참조 알고리즘을 제안한다. 제안하는 서브 샘플링 기반 비정방형 블록 분할 방법은 물체의 경계부근에서 정확하게 블록을 분할하여 임의의 경계를 갖는 서로 다른 물체를 분할하기에 적합하다. 또한 움직임 추정 시 주변 블록의 분할 결과에 따라 적응적으로 움직임 벡터를 참조함으로써, 움직임 벡터의 아웃라이어(Outlier)를 방지하고 실제 움직임을 찾기 위한 알고리즘을 제안하였다. 본 논문의 구성은 다음과 같다. 제 II 장에서는 제안한 프레임율 증가 변환 방법에 대해서 설명하며 제 III 장에서는 검증 결과에 대하여 서술하였다. 마지막으로 IV 장에서는 결론을 기술한다.

II. 제안하는 프레임율 증가 변환 알고리즘

제안하는 알고리즘은 시간적으로 인접한 두 개의 원본 프레임, 즉 F_t, F_{t-1} 프레임을 이용하여 보간 프레임 $F_{t-0.5}$ 를 생성한다. 우선 입력된 F_t, F_{t-1} 프레임은 서브 샘플링 되며 물체의 경계면 표시에 적합한 비정방형 블록으로 분할된다. 이후 주변 블록의 블록 분할 방법에 따라 움직임 벡터 정보를 적응적으로 참조하여 단방향 움직임 추정을 수행하며, 획득된 움직임 벡

터에 대하여 움직임을 보정을 수행한다. 이 후 움직임 보상 및 홀 및 중첩영역을 제거함으로써 최종 보간 프레임을 획득할 수 있다.

1. 서브 샘플링 기반 비정방형 블록 분할 알고리즘

제안하는 블록 분할 알고리즘은 서로 다른 물체 사이에 큰 텍스처의 차이가 존재한다는 특성을 이용한다. 물체의 텍스처는 블록 내부의 평균값, 블록 내부의 텍스처 균일도는 블록 내부의 분산값을 이용하여 수식화하였다. 블록 간 평균값의 차이가 가장 크고, 블록 내 분산값이 가장 작도록 현재 블록을 두 개의 비정방형 하위 블록으로 분할함으로써 물체의 경계부근에서 세밀한 블록 분할이 가능하다. 그림 1에서 제안하는 알고리즘을 도식화 하였다.

그림 1과 같이, 제안하는 비정방형 블록분할 알고리즘은 우선 처리중인 블록을 4:1 비율로 서브 샘플링 한다. 서브 샘플링 과정을 통해 평균값과 분산 값을 구할 픽셀의 개수는 1/4로 감소하며, 블록 분할에 필요한 연산량을 대폭 줄일 수 있다.

이후 식 (1)과 (2)를 이용하여 픽셀좌표 (x,y) 에 해당하는 픽셀을 다양한 ρ 와 θ 값에 따라 B_1 또는 B_2 의 두 개 블록으로 분할해본다. 이 때 ρ 와 θ 값의 범위는 각각 $[0,3], [0,2\pi]$ 로 제한하였으며, $\Delta\rho = 1, \Delta\theta = \pi/8$ 로 설정하였다.

$$f(x,y) = x \cos \theta + y \sin \theta - \rho \quad \begin{matrix} (-4 \leq x < 4) \\ (-4 \leq y < 4) \end{matrix} \quad (1)$$

$$B(x,y) \in \begin{cases} B_1, & (\text{if } f(x,y) < 0) \\ B_2, & \text{otherwise} \end{cases} \quad (2)$$

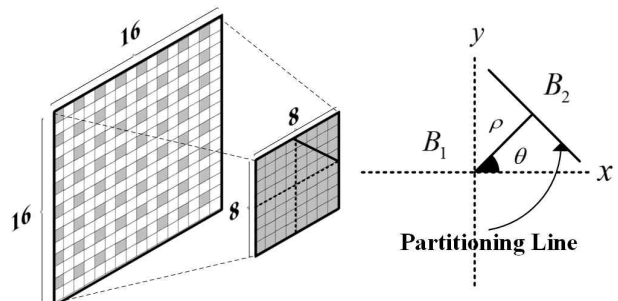


그림 1. 서브 샘플링 기반 비정방형 블록 분할 방법
Fig. 1. Subsampling-based wedge shaped block partitioning.

다양한 ρ 와 θ 값 조합 중, 식 (3), (4)와 같이 B_1 과 B_2 의 평균값 차이를 최대화하고, B_1 블록과 B_2 블록 내부의 분산 값의 합을 최소화 하는 ρ 와 θ 를 블록의 최종 분할 선의 파라미터로 결정함으로써, 텍스처 차이를 최대화 하는 비정방형 블록 분할을 수행할 수 있다.

$$(\rho, \theta) = \arg \max_{(\rho, \theta) \in (0 \leq \rho \leq 3, 0 \leq \theta \leq 2\pi)} |Mean(B_1) - Mean(B_2)| \quad (3)$$

$$(\rho, \theta) = \arg \min_{(\rho, \theta) \in (0 \leq \rho \leq 3, 0 \leq \theta \leq 2\pi)} |var(B_1)| + |var(B_2)| \quad (4)$$

제안하는 블록 분할 알고리즘은 비정방형 블록 분할 외에도 평탄한 텍스처 및 블록 내 텍스처가 보다 복잡한 경우를 표현하기 위해 16 x 16, 8 x 8 블록을 혼합하여 사용한다. 16 x 16 블록의 경우 평균값의 차이를 구할 수 없으므로 내부 분산값이 비정방형 블록들의 분산값의 합보다 작을 경우 사용되며, 8 x 8 블록의 경우 블록간 평균값 차이의 최댓값과 블록 내부의 분산값을 비정방형 블록 분할 방법과 비교하여 선택한다.

2. 주변 블록의 분할형태를 고려한 실제 움직임 추정 기법 및 움직임 벡터 보정

대부분의 프레임율 증가 알고리즘^[9~11]의 경우, 움직임 벡터의 아웃라이어를 방지하기 위해 주변 블록의 움직임 벡터를 활용하여 현재 블록의 움직임 벡터를 보정한다. 하지만 잘못된 주변 움직임 벡터를 참조할 경우, 오히려 보간된 프레임의 성능 열화가 발생할 수 있다. 그림 2는 주변 블록의 분할 결과와 현재 처리중인 블록과의 관계를 보여준다. B_2 블록과 B_4 블록의 경우 현재

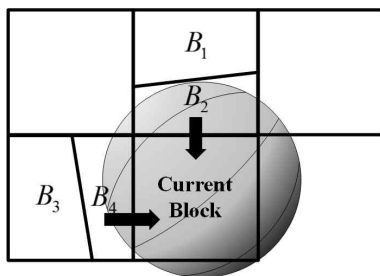


그림 2. 주변 블록의 분할 방법과 현재 처리 중인 블록과의 관계의 예

Fig. 2. Relationship between the current block and partitioning results of the adjacent blocks.

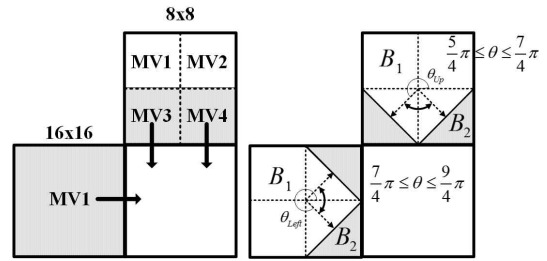


그림 3. 제안하는 주변 블록 분할 방법에 따른 적응적 움직임 벡터 참조의 예

Fig. 3. Example of neighboring blocks for the predicted MVs.

처리중인 블록의 움직임과 밀접한 관계가 있지만, B_1 블록과 B_3 블록의 움직임 벡터가 현재 블록의 움직임 벡터에 잘못 참조될 경우 현재 처리중인 블록의 움직임 벡터에 아웃라이어를 발생시킨다. 제안하는 움직임 추정 알고리즘은 주변 블록의 분할 형태에 따라 움직임 벡터를 적응적으로 참조함으로써, 결정된 움직임 벡터의 아웃라이어를 최소화하는 효과가 있다.

제안하는 움직임 추정 방법은 주변 블록의 블록 분할 결과, 즉, ρ 와 θ 값에 따라 움직임 벡터의 참조여부를 결정한다. 그림 3은 제안하는 알고리즘을 보여준다.

$$MV_{Up_pred} = \begin{cases} MV(B_2), & \text{if } (\frac{5}{4}\pi \leq \theta_{Up} \leq \frac{7}{4}\pi) \\ MV(B_1), & \text{otherwise} \end{cases} \quad (5)$$

$$MV_{Left_pred} = \begin{cases} MV(B_2), & \text{if } (\frac{7}{4}\pi \leq \theta_{Left} \leq \frac{9}{4}\pi) \\ MV(B_1), & \text{otherwise} \end{cases} \quad (6)$$

그림 3과 같이, 상위 블록과 좌측 블록이 8 x 8 블록 또는 16 x 16 크기의 블록으로 분할되었을 경우, 현재 처리중인 블록에 인접한 움직임 벡터를 상위 예측 움직임 벡터, MV_{Up_pred} 및 좌측 예측 움직임 벡터, MV_{Left_pred} 로 선택한다. 만약 주변 블록이 비정방형 블록으로 분할되었을 경우, 식 (5), (6)과 같이 참조 블록의 분할 각도, θ 에 따라 적응적으로 움직임 벡터를 선택한다. 예측 움직임 벡터가 결정된 후에는 식 (7)에 따라 움직임 추정을 수행한다. 제안하는 알고리즘에서 움직임 추정은 정합 오차뿐만 아니라 예측 움직임 벡터와 현재 탐색중인 움직임 벡터와의 거리, d 도 함께 고려된다. 예측 움직임 벡터와 현재 탐색중인 움직임 벡

터와의 거리는 식 (8)로 정의된다. 기존의 프레임 추정 기법들이 움직임 추정을 수행함에 있어 정합 오차만을 고려하는 것과 달리, 제안하는 알고리즘은 정합 오차와 주변 움직임 벡터와의 거리를 함께 고려함으로써 움직임 벡터의 아웃라이어율을 감소시킬 수 있다.

$$MV = \arg \min \left\{ SAD + \sum_{i=1}^n d_i \right\} \quad (7)$$

$$d = |MV_{prediction} - MV|, \quad (8)$$

$$\{MV_{Up_pred}, MV_{Left_pred}\} \in MV_{prediction}$$

3. 실제 움직임 벡터 탐색을 위한 움직임 벡터 보정

비정방형 블록 분할과, 적응적 주변 움직임 벡터 참조 알고리즘을 통해 대부분의 움직임 벡터는 실제 움직임을 가리키게 된다. 하지만 여전히 잘못된 움직임을 가리키는 움직임 벡터가 남아있을 수 있으며 이러한 움직임 벡터들을 보정하기 위한 과정이 필요하다. 그림 4는 제안하는 움직임 벡터 보정 알고리즘을 보여준다. 제안하는 보정 알고리즘에서는 주변 움직임 벡터와의 거리값 차이를 고려하여 최종 움직임 벡터를 획득한다. 주변 움직임 벡터와의 거리는 식 (9)를 통해 구하며 현재 블록의 움직임 벡터와의 거리에는 4배의 가중치를 주었다. 주변 움직임 벡터와 거리차이가 가장 작은 움직임 벡터는 최종 움직임 벡터로 결정된다.

$$MV_{final} = \arg \min_{MV \in \{MV_1, \dots, MV_{12}\}} \sum_{i=1}^{12} |MV - MV_i| \quad (9)$$

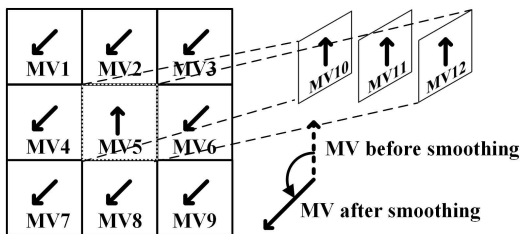


그림 4. 제안하는 움직임 보정 알고리즘
Fig. 4. Proposed MV smoothing algorithm.

4. 움직임 보상을 이용한 보간 프레임 생성

움직임 보정이 완료된 후, 최종 획득한 움직임 벡터

를 이용하여 보간 프레임을 생성한다. 제안하는 프레임을 증가 변환 알고리즘에서는 일반적인 프레임을 변환 알고리즘과 같이 움직임 벡터의 절반에 해당하는 지점에 블록을 보상하는 방식으로 보간 프레임을 생성한다. 단방향 프레임 보간 알고리즘의 경우, 양방향 프레임 보간 방식과 달리 홀 및 중첩 영역이 발생하므로 이에 대한 처리가 필요하다. 중첩 영역의 경우, 픽셀 값의 평균을 이용하였으며, 홀의 경우는 IDW(Inverse-distance weighting spatial interpolation)^[12] 방식을 사용하여 생성해 주었다. IDW 방식은 낮은 연산량에 비해 높은 정확도를 보장하는 픽셀 보간 방식으로, 거리에 반비례하는 가중치를 사용하여 픽셀을 생성해 낸다.

III. 실험

제안한 프레임을 증가 변환 방법을 평가하기 위해 PSNR(Peak signal-to-noise ratio) 및 SSIM(Structural similarity) 을 이용하여 기존의 프레임을 증가 변환 알고리즘^[6-8]과 성능 비교를 수행하였다. PSNR의 경우 영상처리 분야에서 객관적인 평가를 위해 일반적으로 사용되며, SSIM 의 경우 인간의 시각적 특성을 반영한 평가 지표로써 프레임을 증가 변환 알고리즘에서 주로 사용되고 있다. 제안하는 알고리즘의 경우 16 x 16의 블록크기를 사용하였으며, 탐색 영역은 $[\pm 8, \pm 8]$ 로 설정하였다. VS-BMC^[6]의 경우 8 x 8 및 4 x 4의 매크로 블록과 일반적인 OBMC 방법을 이용하여 프레임 보간을 수행하였으며, 탐색 영역은 $[\pm 16, \pm 16]$ 를 사용하였다. Dual ME^[7]의 경우 8x8의 기본 블록 및 12x12 오버랩 블록과 $[\pm 16, \pm 16]$ 의 탐색영역을 사용하였고, PMVS^[8]의 경우 8x8의 매크로블록과 $[\pm 16, \pm 16]$ 의 탐색영역을 사용하였다. 테스트 영상은 일반적으로 사용되는 News, Football, Stefan, Coastguard, Foreman의 352 x 288 영상과, BasketballDrive, Vidyo1 와 같은 1920 x 1080, 1280 x 720의 고해상도 영상을 사용하였다. 성능 평가를 위해 최초 100장의 원본 영상에서 짝수 프레임을 50개를 삭제한 뒤, 홀수 프레임을 이용해 가상의 짝수 프레임을 만들고 원본 짝수 프레임과 화질 대조를 실시하는 방법으로 진행하였다. BasketballDrive의 경우 50장의 원본 영상에서 25개의 짝수 프레임을 생성하여 결과 비교에 사용하였다.

실험 결과, 대부분의 영상들에 대하여 제안하는 프레

표 1. 기존 프레임율 증가 알고리즘과의 PSNR 비교
Table 1. PSNR comparisons to different existing frame rate up algorithm.

Sequence (dB)	VS-BMC [6]	Dual ME [7]	PMVS [8]	Proposed
News	35.23	35.71	37.25	37.10
Football	21.27	22.27	22.17	22.33
Stefan	22.87	26.68	26.32	27.10
Coastguard	28.98	30.54	30.75	31.94
Foreman	32.62	32.89	32.71	31.93
Basketball Drive	25.40	27.31	26.02	27.41
Vidyo1	41.97	44.35	43.58	44.72
Average	29.76	31.39	31.26	31.79
$\Delta PSNR_{Avg}$	-2.03	-0.40	-0.53	0.00

임율 증가 변환 알고리즘이 기존의 알고리즘에 비해 높은 PSNR을 갖는 것을 확인할 수 있었다. 표 1에서 볼 수 있듯이, 제안하는 알고리즘과 기존의 알고리즘의 평균 PSNR 차이, 즉 $\Delta PSNR_{Avg}$ 비교에서, 제안하는 알고리즘이 VS-BMC^[6], Dual ME^[7], PMVS^[8]에 비해 각각 평균 2.03dB, 0.40dB, 0.53dB 만큼 높은 PSNR을 가지는 것을 확인하였다. 특히 Football과 Stefan과 같이 움직임이 많은 영상에서는 기존 알고리즘보다 최대 1.06dB에서 4.23dB까지 높은 PSNR을 보여, 움직임이 많은 영상을 표현하는데 탁월한 성능을 보이는 것을 확인하였다. 또한 1920 x 1080 영상과 같이 고해상도 영상에 대해서도 기존 알고리즘에 비해 높은 성능을 보였다. 이는 물체 주변에서의 정확한 비정방형 블록 분할이 기존의 정방형 블록 분할에 비해 실제 움직임 벡터를 찾는 데 효과적이라는 것을 보여준다.

표 2. 기존 프레임율 증가 알고리즘과의 SSIM 비교
Table 2. SSIM comparisons to different existing frame rate up algorithm.

Sequence (dB)	VS-BMC [6]	Dual ME [7]	PMVS [8]	Proposed
News	0.9884	0.9890	0.9898	0.9903
Football	0.7839	0.8172	0.8033	0.8163
Stefan	0.8635	0.9365	0.8965	0.9334
Coastguard	0.9222	0.9404	0.9476	0.9517
Foreman	0.9709	0.9732	0.9734	0.9710
Basketball Drive	0.9193	0.9325	0.9178	0.9339
vidyo1	0.9961	0.9967	0.9971	0.9974
Average	0.9206	0.9408	0.9322	0.9420
$\Delta SSIM_{Avg}$	-0.0214	-0.0012	-0.0098	0.0000

표 3. 알고리즘별 SAD 연산량 비교
Table 3. Number of SAD calculation comparisons to different existing frame rate up algorithm.

	Number of SAD calculation
VS-BMC [6]	$[B^2 \times (2S+1)^2 + 4B \times (2S+1)^2] \times N$ 696,960
Dual ME [7]	$2[(1.5B)^2 \times (2S+1)^2 - [(1.5B)^4 - 24(1.5B)^2 + 144]]$ 936,432
PMVS [8]	$2[B^2 \times (2S+1)^2 + 4B^2]$ 559,616
Proposed	$2[B^2 \times (2S+1)^2 + (\frac{B}{2})^2 \times (3/\Delta\rho) \times (2\pi/\Delta\theta)]$ 563,712

표 2는 기존 알고리즘과 제안하는 알고리즘의 SSIM 성능 비교를 보여주는 표이다. 대부분의 영상에 대해서, 제안하는 프레임율 증가 변환 알고리즘이 기존 알고리즘에 비해 높은 SSIM을 보이며 최대 0.0214의 평균 SSIM 향상을 확인할 수 있다.

표 3은 기존 알고리즘과 제안하는 알고리즘의 블록 당 SAD 연산량을 비교한 표이다. 블록의 크기인 B 를 16, 탐색범위인 S 는 16, VS-BMC^[6]의 반복횟수 N 을 2회로 설정하였으며 제안하는 알고리즘의 변수는 $\Delta\rho = 1$, $\Delta\theta = \pi/8$ 로 설정하였다.

움직임 추정의 복잡도를 증가시켜 보간 프레임의 화질을 향상시키는 기존 알고리즘들과는 달리, 제안하는 알고리즘은 블록 분할과정에서 약간의 연산량 추가만으로 움직임 추정 과정의 복잡도를 유지, 보간된 영상의 화질을 향상시킬 수 있었다. 이러한 방법으로 제안하는 알고리즘은 기존의 알고리즘에 비해 평균 SAD 연산량을 최대 40%까지 감소시킨다.

IV. 결론

본 논문에서는 프레임율 증가 알고리즘에서 물체의 실제 움직임 벡터를 찾기 위한 새로운 방법들을 제안하였다. 첫 번째로 물체의 경계선 표현에 적합한 서브 샘플링 기반 비정방형 블록 분할 알고리즘을 사용하였는데, 이는 기존 정방형 블록 분할 알고리즘보다 물체의 경계에서 정확한 블록 분할 성능을 보인다. 두 번째로 주변 블록의 분할 결과에 따라 참조 움직임 벡터를 적응적으로 변경하는 알고리즘을 제안하였다. 이는 비정방형 블록 분할 방법과 함께 적용되어 물체의 아웃라이

어를 최소화하고 실제 움직임을 찾는 데 탁월한 성능을 보인다. 실험 결과 기존의 방법들에 비해 20~40%의 SAD 연산량으로 평균 PSNR에 있어 최대 2.03dB, 평균 SSIM에 있어서 최대 0.0214만큼 높은 것을 확인할 수 있다.

REFERENCES

- [1] K. Hilman, H.-W. Park, and Y.-M. Kim, "Using motion compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Trans. on CSVT*, Vol. 10, No. 6, pp. 869-877, Sep. 2000.
- [2] Q. Huynh-thu and M. Ghanbari, "Temporal aspect of perceived quality in mobile video broadcasting," *IEEE Trans. Broadcasting*, vol. 54, no. 3, pp. 641 - 651, Sep. 2008. 6
- [3] S.-C. Han and J. W. Woods, "Frame-rate up-conversion using transmitted motion and segmentation fields for very low bit-rate video coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 1997, pp. 747 - 750.
- [4] J. Someya, N. Okuda and H. Sugiura, "The Suppression of Noise on a Dithering Image in LCD Overdrive," *IEEE Trans. Consumer Electronics*, vol. 52, no. 4, pp. 1325-1332, Nov. 2006.
- [5] Y.-H. Ko, J.-H. Kim, S.-W. Lee, and H.-S. Kang, "Dual Block Truncation Coding for Overdriving of Full HD LCD Driver," *IEEE Trans. Consumer Electronics*, vol. 58, no. 1, pp.1-7, Feb. 2012
- [6] 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 Syst. Video Technol.*, vol. 17, no. 4, pp. 407-416, Apr. 2007.
- [7] S.-J. Kang, S.-J. Yoo and Y.-H. Kim, "Dual motion estimation for frame rate up-conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1909-1914, Dec. 2010.
- [8] U.-S. Kim and M.-H. Sunwoo, "New frame rate up-conversion algorithms with low computational complexity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 384-393, Aug. 2014.
- [9] S.-J. Kang, D.-G. Yoo, S.-K. Lee, and Y. H. Kim, "Design and Implementation of Median Filter based Adaptive Motion Vector Smoothing for Motion Compensated Frame Rate Up-Conversion," *International Symposium on Consumer Electronics*, May. 25~28, 2009.
- [10] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *Proc. Eur. Signal Process. Conf., Smolenice, Slovak Republic*, Jul. 2005.
- [11] J. Guo and J. Kim, "Adaptive motion vector smoothing for improving side information in distributed video coding," *J. Inf. Process. Syst.*, vol. 7, no. 1, Mar. 2011.
- [12] G. Y. Lu and D. W. Wong, "An adaptive inverse-distance weighting spatial interpolation technique," *Comput. Geosci.*, vol. 34, no. 9, pp. 1044.1055, Sep. 2008.

저 자 소 개



정 재 현(학생회원)
2013년 아주대학교 전자공학
학사 졸업.
2015년 현재 아주대학교 전자공학
석사 재학.

<주관심분야 : 멀티미디어 코덱, 멀티미디어 신호
처리, 멀티미디어용 ASIP설계>



정 호 선(학생회원)
2012년 아주대학교 전자공학
학사 졸업.
2014년 아주대학교 전자공학 석사
졸업.

<주관심분야 : 멀티미디어 코덱, 멀티미디어 신호
처리, 멀티미디어용 ASIP설계>



선우 명 훈(평생회원)
1980년 서강대학교 전자공학
학사 졸업.
1982년 한국과학기술원 전기전자
공학 석사 졸업.
1982년~1985년 한국전자통신
연구소(ETRI)연구원

1985년~1990년 Univ. of Texas at Austin
전자공학 박사 졸업
1990년~1992년 Motorola, DSP Chip Operation
(미국)
1992년~현재 아주대학교 전자공학과 교수
2011년~현재 IEEE CASS Board of Governor
2011년~현재 IEEE Fellow Member
2012년~2014년 대한전자공학회 반도체
소사이어티 회장
2014년~현재 대한전자공학회 반도체
소사이어티 자문위원
<주관심분야 : VLSI 및 SoC Architecture, 통신
및 멀티미디어 ASIP 및 SoC 설계>