

정규논문 (Regular Paper)

방송공학회논문지 제20권 제6호, 2015년 11월 (JBE Vol. 20, No. 6, November 2015)

<http://dx.doi.org/10.5909/JBE.2015.20.6.880>

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

동일한 움직임 벡터들의 방향과 크기를 고려한 프레임율 증가기법

박종근^{a)}, 정제창^{a)‡}

Frame Rate Up-Conversion Considering The Direction and Magnitude of Identical Motion Vectors

Jonggeun Park^{a)} and Jechang Jeong^{a)‡}

요 약

본 논문은 EBME(Extended Bilateral Motion Estimation) 알고리즘에서 동일한 움직임 벡터들의 방향과 크기를 고려한 알고리즘을 제안하였다. EBME는 BME(Bilateral Motion Estimation)보다 높은 연산량을 요구하기 때문에 프레임 내의 x, y방향 각각의 평균 움직임 벡터크기를 이용하여 동적프레임과 정적프레임을 판단하고, EBME 수행여부를 판단하여 연산량을 줄인다. 또한 동일한 움직임 벡터들의 방향과 크기를 비교하여 MVS(Motion Vector Smoothing)단계 수행여부를 판단함으로써 연산량을 줄인다. 제안하는 알고리즘을 적용한 실험 결과 기존의 EBME 알고리즘에 비해 수행시간은 단축되고 PSNR(Peak Signal to Noise Ratio)은 향상 되었다.

Abstract

In this paper, frame rate up conversion (FRUC) algorithm considering the direction and magnitude of identical motion vectors is proposed. extended bilateral motion estimation (EBME) has higher complexity than bilateral motion estimation (BME). By using average magnitude of motion vector with x and y direction respectively, dynamic frame and static frame are decided. We reduce complexity to decide EBME. also, After we compare the direction and magnitude of identical motion vectors, We reduce complexity to decide motion vector smoothing(MVS). Experimental results show that this proposed algorithm has fast computation and better peak signal to noise ratio(PSNR) results compared with EBME.

Keyword : Frame rate up conversion, Extended bilateral motion estimation, Motion estimation, Identical motion vectors, Motion vector smoothing

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

‡ Corresponding Author : 정제창(Jechang Jeong)

E-mail: jjeong@hanyang.ac.kr

Tel: +82-2-2220-4369

ORCID: <http://orcid.org/0000-0002-3759-3116>

※ 본 연구는 방위사업청 및 국방과학연구소에 의해 설립된 신호정보 특화연구센터의 지원을 받아 수행되었음.

※ 위 논문의 연구결과 중 일부는 “2015년도 한국방송공학회 하계학술대회”에서 발표한 바 있음.

· Manuscript received August 3, 2015; Revised September 24, 2015; Accepted October 19, 2015.

1. 서론

최근 초고화질 영상인 UHD(Ultra High Definition)영상이 활용도가 높아짐에 따라 영상데이터의 양이 증가하고 있지만, 전송하기 위한 대역폭은 여전히 제한되어 있다. 제한된 대역폭에서 초고화질 영상을 보내기위해 데이터 압축을 통해 데이터의 양을 줄이면 데이터 손실로 인하여 영상의 화질이 저하된다. 또한 프레임 속도를 증가시켜 LCD(Liquid Crystal Display)에서 화면 끌림 현상(Motion Blur Effect)을 해결 할 수 있지만, 제한된 대역폭에서 데이터의 양이 증가하게 된다^[1]. 따라서 디코더(decoder) 부분에 FRUC(Frame Rate Up-Conversion)를 이용하여 프레임수를 증가시키는 방법이 제안되었다^{[2][4]}.

초기의 FRUC 알고리즘은 이전프레임과 현재프레임의 평균값을 이용하여 보간프레임을 생성한다. 하지만 이 방법은 영상 내 물체의 움직임을 고려하지 않기 때문에 열악한 결과 이미지를 보인다. 이를 보완하기 위해 제안된 MC-FRUC(Motion-Compensated FRUC)는 움직임 예측(Motion Estimation) 과정에서 물체의 움직임을 찾아 움직임 벡터(Motion Vector)를 구하고 이를 이용하여 MCI(Motion-Compensated Interpolation) 과정에서 보간프레임을 생성한다.

초기에 제안된 FRUC 기술은 블록 정합 알고리즘(Block Matching Algorithm)으로 단방향 움직임 예측(Unilateral Motion Estimation)을 이용하여 움직임을 추정하였다. 이는 현재프레임의 블록을 중심으로 이전프레임으로부터 움직임 벡터를 찾기 때문에 보간프레임 생성 시 홀(hole)과 중첩(overlap) 영역이 생기는 단점이 있다.

이런 문제를 해결하기 위해서 보간프레임의 각 블록을 기준으로 이전프레임과 현재프레임으로부터 대칭성을 이용하여 움직임 벡터를 찾는 BME(Bilateral Motion Estimation)가 제안되었다^[5]. EBME(Extended Bilateral Motion Estimation)에서는 이전프레임과 현재프레임을 이용하여 두 번의 full search를 사용한 후, 한 프레임에서 겹쳐지는 영역의 움직임 벡터를 비교하여 정확한 움직임벡터를 찾는다^[3]. 또한 MC-FRUC의 문제점을 보완하기 위해 향상된 신뢰도의 움직임 벡터를 구하기 위한 FRUC 알고리즘이 제안되었다^{[6][7]}.

제안하는 알고리즘에서는 EBME알고리즘은 한 프레임 내에서 BME과정을 거친 후, 가로축과 세로축 방향으로 반만큼 이동하여 새로운 블록들에 대해 BME과정을 거치며, MVS(Motion Vector Smoothing)단계에서 모든 이상 벡터가 수정될 때까지 median필터가 수행하기 때문에 높은 연산량을 요구한다. EBME의 연산량을 줄이기 위해 제안하는 알고리즘은 프레임 내의 x, y방향 각각의 평균 움직임 벡터크기를 이용하여 동적프레임과 정적프레임을 판단하고, EBME 수행여부를 결정하여 연산량을 줄인다. EBME 단계를 거친 최종적인 움직임 벡터들에서 주변의 동일한 방향과 크기를 비교하여 MVS단계 수행여부를 판단함으로써 연산량을 줄인다.

본 논문의 구성은 다음과 같다. II장에서는 기존 알고리즘을 설명하고, III장에서는 제안한 알고리즘에 대해 설명한다. 그리고 IV장에서는 실험결과를 제시하고, 마지막으로 V장에서는 결론을 맺는다.

II. 기존 알고리즘

1. BME(Bilateral Motion Estimation)

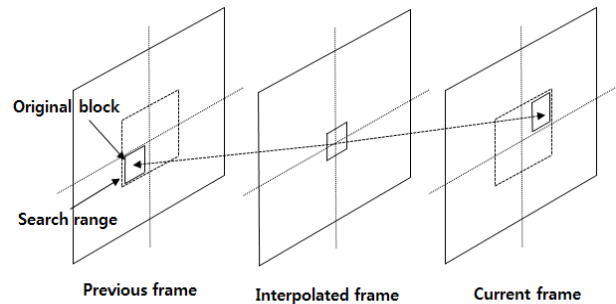


그림 1. 쌍방향 움직임 예측
 Fig. 1. Bilateral Motion Estimation

쌍방향 움직임 예측(Bilateral Motion Estimation)는 보간프레임(Interpolated frame)의 블록을 기준으로 이전프레임과 현재프레임으로부터 대칭성을 이용하여 움직임 벡터를 찾는다^[5]. 그림 1과 같이 쌍방향 움직임 예측은 식 (1), 식 (2)를 이용하여 SBAD(Sum of Bilateral Absolute Differen-

ces)값이 최소가 되는 위치로 정한다.

$$SBAD(mv_x, mv_y) = \sum_{x \in S_x} \sum_{y \in S_y} |f_{n-1}(x - mv_x, y - mv_y) - f_n(x + mv_x, y + mv_y)| \quad (1)$$

$$v = \arg \min_{(mv_x, mv_y) \in S_{x,y}} SBAD(mv_x, mv_y) \quad (2)$$

식(1), 식(2)에서 (mv_x, mv_y) 는 움직임 벡터 후보를 나타내며, f_{n-1} 은 이전 프레임, f_n 은 현재 프레임을 나타낸다. 단방향 예측은 보간프레임 생성 과정에서 홀과 중첩영역이 생기는 문제가 발생하지만, 쌍방향 움직임 예측은 보간 프레임을 기준으로 블록단위로 순차적으로 움직임 벡터를 구하기 때문에 단방향 예측의 문제인 홀과 중첩영역이 발생하지 않는다.

2. EBME(Extended Bilateral Motion Estimation)

확장 쌍방향 움직임 예측(Extended Bilateral Motion Estimation)은 쌍방향 움직임 예측을 한번 수행한 후 다음 블록의 그리드(grid)에서 가로축과 세로축 방향으로 반만큼 이동하여 새로운 블록을 형성하고, 다시 쌍방향 움직임 예측을 수행하여 중첩된 영역에 더 정확한 움직임 벡터를 사용한다. 그림 2와 같이 쌍방향 움직임 예측을 한번 수행하여 Original 움직임 벡터들을 구한 뒤, Original block grid에서 가로축과 세로축 방향으로 반만큼 이동하여 새로운 블록인 Overlapped block를 형성하고 쌍방향 움직임 예측을 수행하여 Overlapped 움직임 벡터들을 구한다. 다음으로 Original

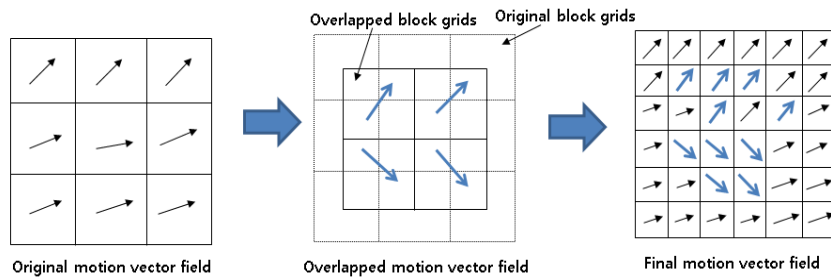


그림 2. EBME에서 움직임 벡터 선택 과정

Fig. 2. Example of the motion vector field obtained by the EBME process

block과 Overlapped block의 겹처지는 영역에 대하여 움직임 벡터를 비교한다. 각각의 블록은 16×16 크기 단위로 구한 움직임 벡터를 네 개의 8×8 단위 블록으로 나누어 겹처지는 부분에 대하여 SBAD를 비교하여 최소가 되는 움직임 벡터로 수정한다.

3. MVS(Motion Vector Smoothing)

최종적으로 구해진 움직임 벡터에는 잘못 판단된 움직임 벡터들이 존재 한다. 따라서 식 (3)을 이용하여 $D_o > D_n$ 인 경우 이상 벡터로 판단하며, 그림 3과 같이 이상 벡터인 v_0 는 정확한 움직임 벡터(Correct motion vector)들이 median 필터를 거쳐 수정된다^[3]. 이상 벡터인 v_0 가 한 번의 수행을 거쳐도 남아있으면, 그림 3과 같이 다시 똑같은 과정을 반복하여 모든 이상 벡터가 수정될 때까지 반복한다. 각각 움

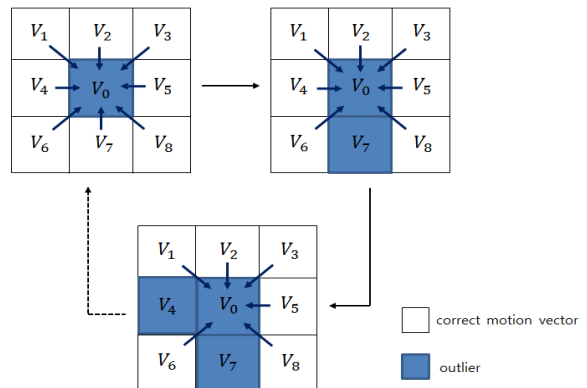


그림 3. MVS 과정

Fig. 3. MVS process

직업 벡터에 **flag**를 표시하여 **flag=1**일 때는 이상 벡터, **flag=0**일 때는 정확한 움직임 벡터로 판단한다. 식 (3)에서 v_m 은 평균 움직임 벡터이며, v_0 는 이상벡터를 나타낸다.

$$\begin{aligned} v_m &= \frac{1}{9} \sum_{i=0}^8 v_i \\ D_o &= \text{abs}(v_m - v_0) \\ D_n &= \frac{1}{8} \sum_{i=1}^8 \text{abs}(v_m - v_i) \end{aligned} \quad (3)$$

III. 제안하는 알고리즘

제안하는 알고리즘의 흐름도는 그림 4이며, 프레임 내의

x, y방향 각각의 평균 움직임 벡터크기와 임계치 T_1 을 비교하여 동적프레임과 정적프레임을 판단하고, EBME 수행 여부를 결정하여 연산량을 줄였다. 또한 움직임 벡터의 방향과 크기를 비교하여 동일한 움직임 벡터를 판단하고, MVS단계 수행 여부를 결정한다. 동적 프레임에서는 움직임이 많은 벡터들이 포함되어 있어 블록화현상이 발생하기 때문에 OBMC(Overlapped Block Motion Compensation)^[3]를 이용하여 블록화현상을 줄였다.

1. 프레임 내의 움직임 정도 판별

프레임 내의 움직임 정도에 따라 동적프레임과 정적프레임을 분류하기위해서 프레임 내의 x, y방향 각각의 평균 움

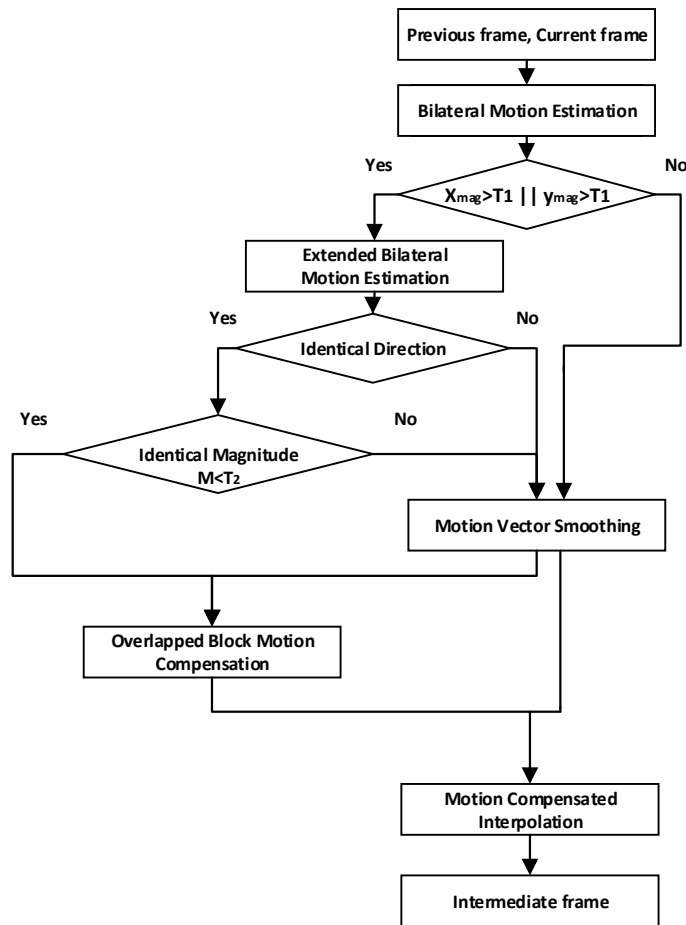


그림 4. 제안하는 알고리즘의 흐름도
 Fig. 4. Flowchart of the proposed scheme

직입 벡터크기를 기준으로 임계치 T_1 과 비교 하여 동적프레임과 정적프레임을 분류한다.

$$MV_{mag} = \frac{1}{N} \sum_{i=1}^N |MV_i| \quad (4)$$

$$MV_{mag}(x_{mag}, y_{mag}) = \begin{cases} DF, & \text{if } x_{mag} > T_1 \text{ or } y_{mag} > T_1 \\ SF, & \text{otherwise.} \end{cases} \quad (5)$$

식 (4), 식 (5)에서 동적프레임인 경우 움직임 벡터의 변화가 많아 BME 보다 정확한 움직임 벡터를 찾는 EBME를 이용하며, 정적프레임인 경우 움직임 벡터의 변화가 적어 EBME 보다 연산량이 낮은 BME를 이용한다. 식 (4)에서 N 은 프레임 내의 블록의 총 개수이고, 식(5)에서 MV_{mag} 은 한 프레임 내의 평균 움직임 벡터크기를 나타내며, x_{mag} 과 y_{mag} 은 프레임 내의 x, y 방향 각각의 평균 움직임 벡터크기이다. DF는 동적프레임, SF는 정적프레임을 나타낸다.

2. 동일한 움직임 벡터들의 방향과 크기 판단

프레임 내의 동일한 움직임 벡터들이 그림 5와 같이 존재할 경우, 3×3마스크에 포함된 움직임 벡터들의 방향과 크기를 판단하고, MVS단계를 수행하지 않음으로써 연산량을 줄인다.

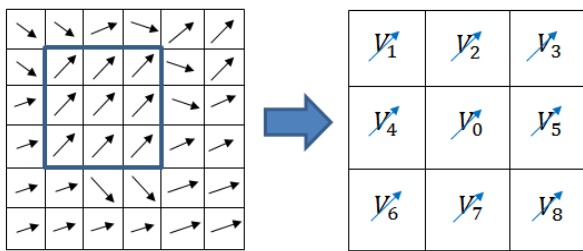


그림 5. 동일한 움직임 벡터일 경우
Fig. 5. Example of the identical motion vectors

동일한 움직임 벡터들의 방향과 크기 판단 과정에서 먼저, 움직임 벡터들의 방향을 판단하게 된다. 3×3마스크에 포함된 움직임 벡터들이 그림 6과 같이 (+, +), (+, -), (-, -), (-, +) 방향 중 하나의 방향에 모두 포함될 경우, 동일한 움직임 벡터크기로 판단을 하게 된다.

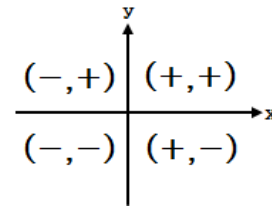


그림 6. 움직임 벡터 방향
Fig. 6. Direction of the motion vectors

다음으로 동일한 움직임 벡터크기를 판단하게 되는데 식 (6)을 통하여 크기를 구하며, 식 (7)에서 M 이 임계치 T_2 보다 작으면 3×3마스크에 포함된 움직임 벡터들 모두 $flag=0$ 으로 판단하여 MVS단계를 수행하지 않는다. MV_m 은 움직임 벡터들의 크기 차이의 합을 나타내며, x_m, y_m 는 x, y방향 각각의 움직임 벡터들의 크기차이의 합이다.

$$MV_m = \sum_{i=1}^8 |V_i - V_0| \quad (6)$$

$$MV_m = (x_m, y_m)$$

$$M = \sqrt{(x_m)^2 + (y_m)^2}$$

$$flag = \begin{cases} 0, & \text{if } M < T_2 \\ skip, & \text{otherwise.} \end{cases} \quad (7)$$

IV. 실험 결과

실험에는 CIF(352×288) 크기의 9개 영상(Bus, Coastguard, Container, Flower, Football, Foreman, Mobile, Stefan, Tempete)을 사용하였다. 원본 홀수프레임을 그림 7과 같은 방법으로 전체 짝수 보상프레임을 생성하였고, 원본 짝수 프레임과 비교하여 PSNR(Peak Signal to Noise Ratio)를 구하였다. 블록화현상을 줄여주기 위해서 OBMC를 사용하였으며, 현재프레임과 이전프레임의 확장된 블록의 겹쳐지는 영역에 따른 가중치를 이용한 MCI (Motion Compensated Interpolation)^[3]를 수행하였다.

실험조건은 기본 블록 크기를 16×16 으로 하였고, 탐색 범위는 ± 8 로 하였으며, $T_1=0.6$, $T_2=0.3$ 로 설정하였으며, Visual Studio 2012로 구현하였다.

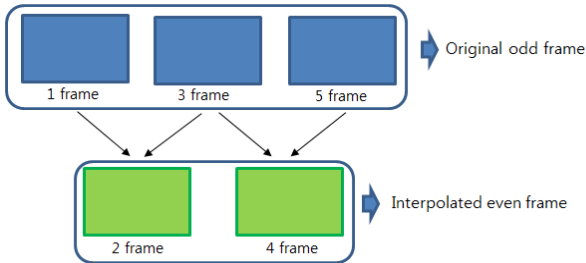


그림 7. 짝수 보상프레임 생성 과정
 Fig. 7. Creation process of the interpolated even frame

EBME 알고리즘과 제안된 알고리즘의 주관적 화질에서는 그림 8과 같이 별 다른 차이점이 없다. 그림 9의 EBME

알고리즘에서 OBMC는 기본 블록 크기의 2배인 윈도우 사이즈를 주변 블록들과 중첩시켜 블록화 현상을 줄여주기 때문에 정확한 주변 블록들을 참고 하지 않으면, 영상의 화질이 저하 된다. 제안한 알고리즘에서는 정적프레임으로 판단하여 OBMC를 생략함으로써 EBME 알고리즘 보다 탈력 숫자들이 명확하게 보임을 알 수 있다.

표 1에서 Mobile영상에서 다른 영상보다 PSNR항상 높아 EBME 알고리즘 보다 제안한 알고리즘이 평균 0.011dB PSNR이 증가함을 볼 수 있고, 표 2에서 제안하는 알고리즘의 수행속도가 전체적으로 EBME 알고리즘 보다 시간이 단축되었다. Container에서는 정적프레임이 많이 포함되어 수행시간이 많이 감소하였다. 카메라의 Global Motion을 고려한다면, 움직임 벡터의 크기가 같더라도 일정한 값을 가질 수 있어 EBME 적용 하지 않아도 되어 연산량을 줄일 수 있을 것으로 예상된다.



그림 8. Container영상에 대한 주관적 화질 비교
 Fig. 8. Subjective result compared with previous algorithm on the Container image

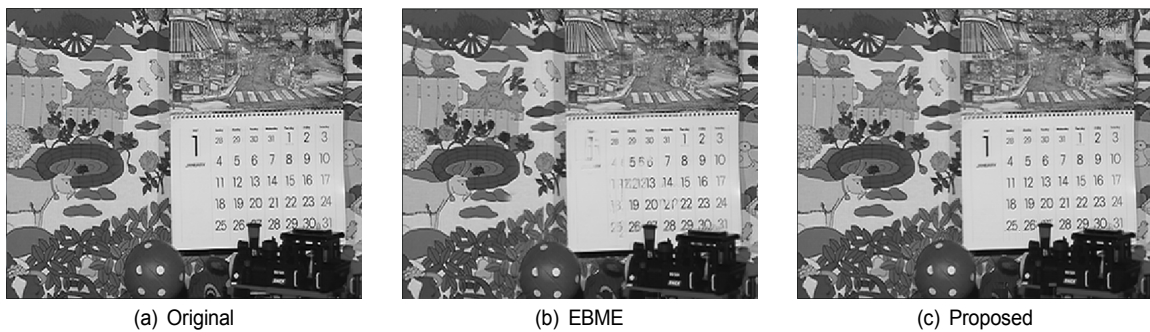


그림 9. Mobile영상에 대한 주관적 화질 비교
 Fig. 9. Subjective result compared with previous algorithm on the Mobile image

표 1. PSNR 비교
Table 1. PSNR Comparison

Sequence	EBME(dB)	Proposed(dB)	Difference
Bus	25.979	25.981	0.002
Coastguard	32.683	32.671	-0.012
Container	39.074	39.360	0.286
Flower	30.125	30.126	0.001
Football	23.575	23.496	-0.079
Foreman	36.552	36.425	-0.127
Mobile	27.290	27.464	0.174
Stefan	21.763	21.757	-0.006
Tempete	29.115	28.972	-0.143
Average	29.573	29.584	0.011

표 2. 수행시간
Table 2. Computation Times

Sequence	EBME(sec)	Proposed(sec)	Relative Rate(%)
Bus	36.217	36.164	0.15
Coastguard	70.365	69.917	0.64
Container	73.086	38.866	46.82
Flower	61.349	61.134	0.35
Football	35.679	34.16	4.26
Foreman	68.566	58.905	14.09
Mobile	76.619	68.377	10.76
Stefan	69.230	66.932	3.32
Tempete	62.133	45.449	26.85
Average	61.472	53.323	13.26

V. 결론

본 논문에서 EBME알고리즘의 연산량을 줄이기 위해 프

레이프 내의 x, y방향의 각각의 평균 움직임 벡터 크기를 이용하여 동적프레임과 정적프레임을 판단한다. 동적프레임인 경우, 움직임 벡터 변화가 많아 EBME를 사용하여 정확한 움직임 벡터를 찾고, 정적프레임인 경우, 움직임 벡터 변화가 적기 때문에 연산량이 적은 BME를 이용하였다. 동적프레임과 정적프레임을 판단 후, 프레임 내의 여러 개의 동일한 움직임 벡터들이 존재한다면, 여러 개의 동일한 움직임 벡터들의 방향과 크기를 비교하여 MVS단계를 통과시켜 연산량을 줄였다.

실험 결과 EBME 알고리즘 보다 평균 0.011dB의 PSNR 향상을 가져왔으며, 상대 속도 개선이 EBME 알고리즘 보다 평균 13.26%로 개선되었다.

참고 문헌 (References)

- [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. H. Lee, O. Kwon, and R. H. Park, "Weighted-adaptive motion compensated frame rate up-conversion," *IEEE Trans. Consumer Electronics*, vol. CE-49, no. 3, pp. 485-492, Aug. 2003.
- [3] S. J. Kang, K. R. Cho, and Y. H. Kim, "Motion Compensated Frame Rate Up-Conversion Using Extended Bilateral Motion Estimation," *IEEE Trans. Consumer Electronics*, vol. 53, no. 4, pp. 1759-1767, Nov. 2007.
- [4] D. J. Park, and J. C. Jeong, "Adaptive Extended Bilateral Motion Estimation Considering Block Type and Frame Motion Activity," *JBE*, vol. 18, no. 3, May 2013.
- [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] S. J. Kang, S. Yoo, and Y. H. Kim, "Dual Motion Estimation for Frame Rate Up-Conversion," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1909-1914, Dec. 2010.
- [7] U. S. Kim and M. H. Sunwoo, "New Frame Rate Up-Conversion Algorithms with Low Computational Complexity," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 384-393, Mar. 2014.

저 자 소 개



박 종 근

- 2014년 2월 : 숭실대학교 정보통신전자공학부 졸업
- 2014년 3월 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 석사과정
- ORCID : <http://orcid.org/0000-0002-2475-3555>
- 주관심분야 : 영상처리, 영상압축



정 제 창

- 1980년 2월 : 서울대학교 전자공학과 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)
- 1998년 11월 : 과학기술자상 수상
- 1990년 12월 : 정보통신부장관상 수상
- 2007년 : IEEE Chester Sall Award 수상
- 2008년 : ETRI Journal Paper Award 수상
- 2011년 5월 : 제46회 발명의 날 녹조근정훈장 수훈
- ORCID : <http://orcid.org/0000-0002-3759-3116>
- 주관심분야 : 영상처리, 영상압축, 3DTV