일반논문 (Regular Paper)
방송공학회논문지 제23권 제1호, 2018년 1월 (JBE Vol. 23, No. 1, January 2018)
https://doi.org/10.5909/JBE.2018.23.1.138
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

프레임 율 향상을 위한 분산 및 적응적 탐색영역을 이용한 움직임 추정 알고리듬

유 송 현[a)], 정 제 창[a)‡]

Motion Estimation Algorithm Using Variance and Adaptive Search Range for Frame Rate Up-Conversion

Songhyun Yu[a)] and Jechang Jeong[a)‡]

요 약

본 논문에서는 움직임 추정을 이용한 새로운 프레임 율 향상 변환 알고리듬을 제안한다. 제안 된 알고리듬은 더 정확한 움직임 벡터를 찾기 위해 움직임 추정 방법에서 오차의 분산을 추가적으로 이용한다. 그런 다음, 이웃 움직임 벡터들의 분산 및 현재 움직임 벡터와 이웃하는 평균 움직임 벡터 간의 분산을 사용하여 잘못 찾아진 움직임 벡터를 탐색한다. 탐색된 벡터들은 8 개의 이웃 움직임 벡터의 가중 합에 의해 수정된다. 또한, 보다 정확한 움직임 벡터를 찾고 동시에 계산 복잡도를 줄일 수 있는 적응적 탐색 영역 결정 알고리듬을 제안한다. 결과적으로, 제안하는 알고리듬은 기존의 알고리듬들에 비해 평균 최대 신호 대 잡음 비 (PSNR)와 구조적 유사도 (SSIM)을 각각 1.44dB 및 0.129까지 향상시켰다.

Abstract

In this paper, we propose a new motion estimation algorithm for frame rate up-conversion. The proposed algorithm uses the variance of errors in addition to SAD in motion estimation to find more accurate motion vectors. Then, it decides which motion vectors are wrong using the variance of neighbor motion vectors and the variance between current motion vector and neighbor's average motion vector. Next, incorrect motion vectors are corrected by weighted sum of eight neighbor motion vectors. Additionally, we propose adaptive search range algorithm, so we can find more accurate motion vectors and reduce computational complexity at the same time. As a result, proposed algorithm improves the average peak signal-to-noise ratio and structural similarity up to 1.44 dB and 0.129, respectively, compared with previous algorithms

Keyword : Frame rate up-conversion, motion estimation, adaptive search range

a) 한양대학교 전자컴퓨터통신공학과 (Department of Electronics and Computer Engineering, Hanyang University)
‡ Corresponding Author : 정제창(Jechang Jeong)
E-mail: jjeong@hanyang.ac.kr
Tel: +82-2-2220-0369
ORCID: http://orcid.org/0000-0002-3759-3116
※ 이 연구는 방위사업청 및 국방과학연구소의 재원에 의해 설립된 신호정보 특화연구센터 사업의 지원을 받아 수행되었음.
· Manuscript received November 13, 2017; Revised November 28, 2017; Accepted November 28, 2017.

Ⅰ. 서 론

프레임 율 향상 (FRUC: frame rate up-conversion)은 고품질의 영상을 보기 위해 프레임들 사이에 새 프레임을 생성하여 프레임 율을 올리기 위해서 영상 신호의 수신 측에서 사용되는 기술이다. 처음에는 원본 프레임을 단순히 복사하여 새 프레임을 생성하였다. 그러나 최근에 움직임 보상 FRUC (MC-FRUC: motion compensated FRUC) 알고리듬이 개발되었다[1]-[3],[9]-[11]. MC-FRUC는 움직 추정 (ME: motion estimation), 움직임 벡터 수정 (MVS: motion vector smoothing), 움직임 보상 프레임 보간 (MCI: motion- compensated interpolation)의 3 단계로 구성된다. ME는 각 블록의 움직임 벡터를 찾고, MVS는 부정확한 움직임 벡터를 판별 및 수정하고, MCI는 새로운 프레임을 생성한다.

잘못된 움직임 벡터는 보간 된 프레임의 화질 저하를 가져 오기 때문에 MC-FRUC의 성능은 ME 알고리듬에 크게 의존한다. 다양한 ME 알고리듬 중에서 일반적으로 두 가지 알고리듬, 즉 양방향 ME와 단방향 ME가 사용된다. 양방향 ME는 보간 된 프레임의 관점에서 움직임 벡터를 찾기 때문에 보간 프레임에 겹쳐진 영역이나 홀(hole)이 생기지 않는다는 장점이 있으나, 배경에 대칭적인 움직임이 둘 이상 있으면 잘못된 움직임 벡터를 찾을 수 있다는 단점이 있다. 이 문제를 해결하기 위해 여러 논문들이 제안되었다. Choi 등에 의해 제안된 알고리듬[4]은 정확한 움직임 벡터를 찾기 위해 움직임 추정 식에 side matching distortion (SMD)을 추가하였고, Kang 등에 의해 제안된 FRUC 알고리듬[5]은 움직임 추정을 할 때 오버 사이즈 블록을 사용하고 단방향 ME를 이 양방향 ME에 결합하여 정확도를 높였다. Kwon 등에 의해 제안된 알고리듬[11]은 영상의 사이즈를 조절하여 움직임 추정을 수행하였다. 이 논문들은 부정확 한 움직임 벡터를 줄이지만 잘못된 움직임 벡터는 여전히 남아있다. 단방향 ME는 원본 프레임의 관점에서 움직임 벡터를 추정하므로 보다 정확한 움직임 벡터를 찾을 수 있다. 그러나 단방향 ME도 오차의 절댓값의 합 (SAD: sum of absolute differences) 만을 움직임 판단 기준으로 사용하기 때문에 부정확 한 움직임 벡터를 찾을 가능성이 있따. SAD는 잔여 에너지를 의미하므로 최소 SAD를 가지는 위치는 실제 움직임과는 다를 수 있다. 일부 연구[6]-[8]는 MVS 단계에서 잘못된 움직임 벡터를 수정하려고 시도했지만 일부 잘못된 움직임 벡터는 여전히 수정하기가 어렵다.

이 문제를 해결하기 위해 오차의 분산을 사용하는 새로운 ME 알고리듬을 제안한다. 또한 복잡도를 줄이고 보다 정확한 움직임 벡터를 찾기 위해 적응적 탐색 영역 조정 알고리듬을 제안하고 잘못된 움직임 벡터를 보다 정확하게 검출하기 위한 새로운 MVS 알고리듬을 제안한다.

이 논문의 나머지는 다음과 같이 구성되어있다. 2 장에서는 제안 하는 알고리듬에 대해 자세히 설명한다. 3장에서는 기존의 알고리듬들과 비교 한 실험 결과를 보여준다. 마지막으로, 4장에서 본 논문을 결론짓는다.

Ⅱ. 제안하는 알고리듬

그림 1은 제안 하는 알고리듬의 전체 흐름도를 나타낸다. 제안하는 알고리듬은 단방향 움직임 추정, 움직임 벡터 수정, 홀(hole) 보간 과정을 포함한다. 제안 하는 알고리듬은 각 프레임 쌍 사이에서 순방향 및 역방향으로 SAD와 오차의 분산 값을 사용하여 움직임 벡터를 탐색하며, 각 움직임 벡터는 해당 블록의 분산 값에 의해 유효성이 검토된다. 다음으로, MVS 과정에서 부정확한 움직임 벡터는 이웃 움직임 벡터의 가중 평균을 사용하여 수정된다. 마지막으로, MCI에서 순방향 및 역방향 움직임 벡터를 사용하여 보간

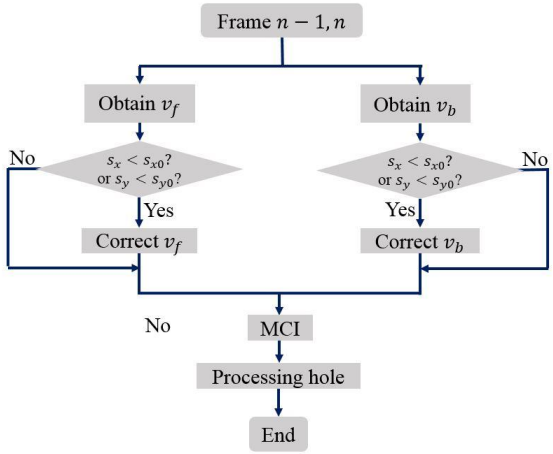


그림 1. 제안하는 알고리듬의 전체적인 흐름도
Fig. 1. Overall flowchart of proposed algorithm

프레임을 생성하고 홀(hole) 영역을 보간한다.

1. 움직임 추정

제안하는 알고리듬은 움직임을 순방향 및 역방향으로 각각 추정한다. 일반적으로 사용되는 추정 기준은 다음 식으로 주어진다.

$$SAD_f(B_{i,j},v_m)=\sum_{p\in B_{i,j}}|f_{N-1}(p)-f_N(p+v_m)| \quad (1)$$

$$v_f=\arg\min_{v_m\in SR}SAD_f(B_{i,j},v_m) \quad (2)$$

$$SAD_b(B_{i,j},v_m)=\sum_{p\in B_{i,j}}|f_N(p)-f_{N-1}(p+v_m)| \quad (3)$$

$$v_b=\arg\min_{v_m\in SR}SAD_b(B_{i,j},v_m) \quad (4)$$

여기서 SAD_f와 SAD_b는 각각 순방향과 역방향의 SAD 값을 나타내며 각 블록은 $B_{i,j}$, 각 화소의 위치는 p로 표시되었다. v_m은 움직임 벡터 후보를 의미하며 SR은 탐색 영역을 의미한다. 최종 움직임 벡터인 v_f와 v_b는 탐색 영역 안에서 최소 SAD값을 가지는 움직임 벡터 후보로 결정된다.

식 (1)—(4)에서 볼 수 있듯이, 일반적인 ME는 움직임 추정 시에 SAD만을 판단 기준으로 사용한다. 하지만 최소 SAD값이 물체의 실제 움직임을 항상 보장하지는 못한다. 다르게 말하면, 더 큰 SAD값을 갖는 움직임 벡터가 실제 움직임일 가능성이 있다. 이러한 문제를 해결하기 위하여 우리는 기존 SAD외에 오차의 분산 값이라는 새로운 기준을 움직임 추정에 사용한다. SAD는 평균적인 오차를 의미하므로, 이 값이 작더라도 오차의 절댓값의 분산이 크다면 보간 프레임의 화질 저하를 발생시킬 수 있으므로, 제안하는 알고리듬은 오차의 절댓값의 분산이 작은 위치가 실제 움직임일 가능성이 높다는 가정을 둔다.

그림 2는 제안하는 ME알고리듬의 흐름도를 자세히 보여준다. 변수 R은 오차의 절댓값의 편차를 나타내는 척도로, 다음의 식으로 주어진다.

$$R(e_{v_m})=\sum_{p\in e_{v_m}}|e_{v_m}(p)-E(e_{v_m})| \quad (5)$$

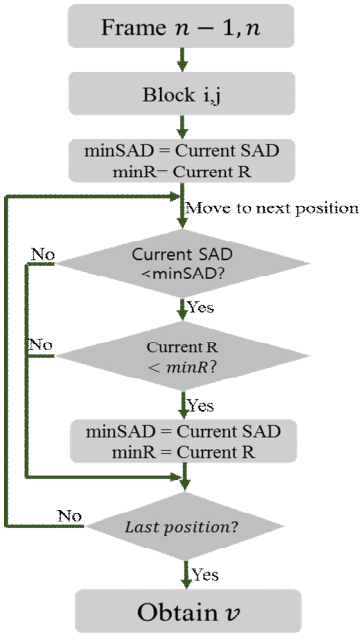


그림 2. 제안하는 ME 알고리듬 흐름도
Fig. 2. flowchart of proposed ME algorithm

여기서 e_{v_m}은 움직임 벡터 후보 v_m를 사용하여 얻은 오차 블록을 의미하고 $E(e_{v_m})$은 e_{v_m}의 평균값을 의미한다. 현재 움직임 벡터가 최종 움직임 벡터의 후보가 되기 위해서는 현재 SAD가 이 전의 최소 SAD인 minSAD보다 작을 뿐 아니라 현재 블록의 R값도 이 전까지의 최소 R값보다 작아야 한다. 제안하는 알고리듬은 ME를 순방향과 역방향으로 수행하여 결과적으로 움직임 벡터 v_f와 v_b를 얻는다.

2. 적응적 탐색 영역

탐색 영역이란 현재 블록의 움직임 벡터를 찾을 때 참조 영상에서 움직임 벡터의 후보가 될 수 있는 화소 영역의 크기를 의미한다. 탐색 영역의 결정은 매우 중요한데, 그 이유는 탐색 영역이 너무 작으면 실제 움직임이 포함되지 않을 수 있고, 탐색 영역이 너무 크면 연산량이 크게 증가하기 때문이다. 또한 한 영상 내에서도 정적인 부분과 동적인 부분이 동시에 존재할 수 있으며, 이러한 경우에 프레임 성질에 따른 적절한 탐색 영역 결정이 필요하다. 따라서 본 절에서는 적응적으로 탐색 영역을 결정하는 알고리듬을 제

안한다.

프레임 단위로 탐색 영역을 결정하며 이 전 프레임에서의 움직임 벡터의 평균값을 사용한다. 이는 다음 식으로 주어진다.

$$SR_n = \begin{cases} \pm 16 & \text{if } v_{fx}+v_{fy}+v_{bx}+v_{by} \le 4\times m \\ \pm 32 & otherwise \end{cases} \quad (6)$$

여기서 $v_{fx}, v_{fy}, v_{bx}, v_{by}$는 각각 이 전 프레임에서의 순방향의 가로방향, 세로방향, 역방향의 가로방향, 세로방향의 움직임 벡터의 평균값을 의미한다. 현재 프레임에서의 탐색 영역은 식 (4)에서 SR_n으로 표현되며 이는 이 전 프레임의 움직임 벡터들로부터 결정된다. 단위는 화소이다. 임계값을 의미하는 m은 실험적으로 5로 결정하였다.

3. 움직임 벡터 수정

본 절에서는 ME 단계에서 얻은 움직임 벡터들 중에서 순방향 및 역방향 움직임 벡터들의 분산 값을 이용하여 부정확한 움직임 벡터들을 판별하는 알고리듬을 제안한다. 우선 주변 블록들은 같은 물체에 속하고, 비슷한 움직임을 가질 것이라는 가정을 둔다. 첫 번째로, 주변 8개 블록의 움직임 벡터들의 분산 값을 각 방향마다 구하고 이를 s_x, s_y라 한다. 그리고 현재 블록의 움직임 벡터와 이웃하는 8개의 움직임 벡터들의 평균값 간의 분산 값을 구한다. 이를 s_{x0}, s_{y0}라 한다. 만약 s_{x0}가 s_x보다 크거나 s_{y0}가 s_y보다 크면 현재 움직임 벡터를 부정확한 움직임 벡터로 판단한다.

한 프레임의 모든 움직임 벡터들의 판단이 끝난 뒤 부정확한 움직임 벡터는 가중 합[7]을 사용하여 수정되며, 이는 그림 3에서 볼 수 있다. 수정된 움직임 벡터 v_c는 다음의 식으로 주어진다.

$$v_c = \sum_{i=1}^{8} w_i \times v_i \quad (7)$$

여기서 v_i는 현재 블록과 인접한 8개 블록의 움직임 벡터를 의미하며 w_i는 이에 상응하는 가중치 값으로 그림 3에 나타나 있다.

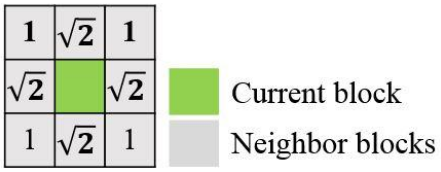


그림 3. MVS에 사용되는 주변 블록들의 가중치
Fig. 3. weights of neighbor blocks for MVS

4. 움직임 보상 프레임 보간

수정된 움직임 벡터 필드 v_{fc}, v_{bc}를 사용하여 각각의 보간 프레임 f_f, f_b가 생성된다. 이 두 보간 프레임 들을 비교하면 겹쳐지는 영역과 홀(hole)이 발생할 수 있다. 두 개의 보간 프레임을 하나의 보간 프레임으로 합치는 과정에서 겹쳐지는 부분은 두 프레임의 평균 화소 값으로 처리한다. 합치는 과정은 다음과 같은 식으로 주어진다.

$$= \begin{cases} f_c(x,y) & \\ f_f(x,y) & \text{if } only\, f_f(x,y) \ne Null \\ f_b(x,y) & \text{if } only\, f_b(x,y) \ne Null \\ \dfrac{f_f(x,y)+f_b(x,y)}{2} & \text{if } f_f(x,y), f_b(x,y) \ne Null \\ 0(hole) & otherwise \end{cases} \quad (8)$$

현재 위치에서 순방향 보간 프레임의 화소가 홀이 아니고 역방향 보간 프레임의 화소가 홀이면 현재 픽셀은 순방향 보간 프레임의 화소로 채워지고, 그 반대의 경우 현재 화소는 역방향 보간 프레임의 화소로 채워진다.

5. 홀 (hole) 보간

합쳐진 보간 프레임에는 여전히 홀이 남아있기 때문에 이 홀들을 채워야한다. 영상의 특성 상 홀과 가장 가까운 위치에 있는 화소가 홀과 같은 물체에 속할 가능성이 가장 크기 때문에 홀과 인접한 8개의 화소들을 홀 보간에 사용한다. 8개의 화소 중에 홀인 화소는 제외하고 나머지 화소들의 움직임 벡터들의 평균값이 현재 처리하고 있는 홀의 움

직임 벡터가 된다. 순방향과 역방향에서 각각 따로 홀의 움직임 벡터를 계산하며, 가로 방향과 세로 방향도 각각 따로 계산한다. 이렇게 계산된 움직임 벡터를 사용하여 보간된 홀의 화소 값은 다음과 같은 식으로 주어진다.

$$f_{hole}(x,y) = \left\{ \begin{matrix} f_{N-1}(x,y)+f_N(x+v_{fxa},y+v_{fya}) \\ +f_N(x,y)+f_{N-1}(x+v_{bxa},y+v_{bya}) \end{matrix} \right\}/4 \quad (9)$$

여기서 $f_{fxa},f_{fya},f_{bxa},f_{bya}$는 각각 순방향 프레임에서 가로방향, 세로방향, 역방향 프레임에서 가로방향, 세로방향의 홀 주위 8개 화소들의 움직임 벡터들의 평균값을 의미한다.

Ⅲ. 실험 결과

보간된 프레임의 화질 비교를 위해서 객관적인 기준으로 각 알고리듬마다 평균 최대 신호 대 잡음 비 (PSNR: peak signal-to-noise ratio)와 구조적 유사도 (SSIM: structural similarity), 그리고 계산 시간을 측정하였다. 실험을 위한 시퀀스로는 CIF 표준 테스트 시퀀스들과 HEVC 표준 테스트 시퀀스 중 고해상도 영상 3개를 사용하였다. 블록 크기는 8×8 화소를 사용하였고 탐색 영역은 Ⅱ-2절에서 설명하였듯이 적응적으로 프레임마다 ±16, ±32화소를 사용하였다. 비교 알고리듬으로는 dual motion estimation

표 1. PSNR값 비교
Table 1. comparison of PSNR values

Resolution	Test sequence (# of frames)	DME[5]		DS-ME[7]		Proposed	
		PSNR	Δ	PSNR	Δ	PSNR	Δ
352×288	*Foreman (300)*	31.58	0.77	30.37	-0.43	30.81	.
	Mobile (300)	25.00	0.56	24.18	-0.25	24.43	.
	Crew (300)	29.21	-0.65	28.90	-0.96	29.86	.
	Akiyo (300)	45.61	1.25	40.69	-3.66	44.36	.
	Mother & Daughter (300)	42.31	1.44	39.41	-1.45	40.86	.
	Soccer (300)	24.27	-1.44	24.55	-1.16	25.71	.
	News (90)	35.78	-1.40	35.65	-1.53	37.18	.
	Football (90)	23.57	-0.57	23.20	-0.95	24.15	.
2560×1600	*Traffic (150)*	37.34	-0.34	37.03	-0.65	37.68	.
	PeopleOnStreet (150)	26.69	-1.34	27.88	-0.15	28.03	.
1920×1080	*BQTerrace (600)*	31.75	-0.22	31.22	-0.75	31.97	.
Average [dB]		32.10	-0.18	31.19	-1.09	32.28	.

표 2. SSIM값 비교
Table 2. comparison of SSIM values

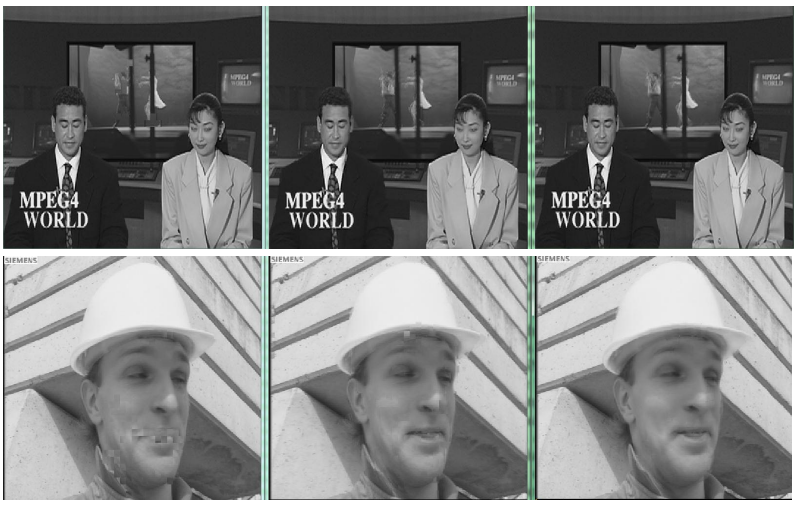
Resolution	Test sequence (# of frames)	DME[5]		DS-ME[7]		Proposed	
		SSIM	Δ	SSIM	Δ	SSIM	Δ
352×288	*Foreman (300)*	0.992	0.002	0.980	-0.010	0.990	.
	Mobile (300)	0.986	0.009	0.965	-0.012	0.977	.
	Crew (300)	0.959	-0.004	0.939	-0.024	0.963	.
	Akiyo (300)	0.999	0.001	0.999	0.000	0.999	.
	Mother & Daughter (300)	0.998	0.000	0.997	-0.001	0.998	.
	Soccer (300)	0.957	0.004	0.897	-0.056	0.953	.
	News (90)	0.997	-0.001	0.996	-0.002	0.998	.
	Football (90)	0.842	-0.017	0.730	-0.129	0.859	.
2560×1600	*Traffic (150)*	0.997	0.000	0.997	0.000	0.997	.
	PeopleOnStreet (150)	0.980	-0.005	0.984	-0.001	0.985	.
1920×1080	*BQTerrace (600)*	0.993	-0.001	0.994	0.000	0.994	.
Average		0.973	-0.001	0.953	-0.021	0.974	.

표 3. 계산시간 비교 (1프레임 생성하는데 걸린 시간 (초))
Table 3. comparison of computation time (Time to generate 1 frame (seconds))

Resolution	Test sequence (# of frames)	DME[5]		DS-ME[7]		Proposed	
		TIME	Δ	TIME	Δ	TIME	Δ
352×288	*Foreman (300)*	0.24	+0.06	0.09	-0.09	0.18	.
	Mobile (300)	0.27	+0.17	0.07	-0.03	0.10	.
	Crew (300)	0.31	+0.13	0.10	-0.08	0.18	.
	Akiyo (300)	0.09	+0.04	0.05	-0.01	0.05	.
	Mother & Daughter (300)	0.19	+0.07	0.09	-0.03	0.12	.
	Soccer (300)	0.33	+0.00	0.10	-0.23	0.33	.
	News (90)	0.11	+0.05	0.05	-0.01	0.07	.
	Football (90)	0.40	-0.12	0.12	-0.40	0.52	.
Average		0.24	+0.05	0.08	-0.11	0.19	.
2560×1600	*Traffic (150)*	9.04	+3.17	4.33	-1.53	5.87	.
	PeopleOnStreet (150)	15.15	+4.65	6.52	-3.98	10.50	.
1920×1080	*BQTerrace (600)*	9.81	+4.87	4.13	-0.81	4.94	.
Average		11.33	+4.23	4.99	-2.11	7.10	.

(DME)[5], direction-select motion estimation (DS-ME)[7]을 사용하였다. 실험 결과를 나타낸 표 1-3에서 Δ값은 기존의 결과들과 제안하는 결과와의 차이값을 의미한다.

표 1은 기존 알고리듬들과 제안하는 알고리듬의 시퀀스별 평균 PSNR값을 나타낸다. 평균적으로 PSNR 값은 DME에 비해 0.18 dB, DS-ME에 비해 1.09 dB 증가하였다. mother & daughter 시퀀스의 경우 DME에 비해 1.44 dB이 증가하였다. 표 2는 각 알고리듬들을 사용했을 때의 결과 영상의 평균 SSIM값을 나타낸 것이다. 평균 SSIM 값은 제안하는 알고리듬에서 DME에 비해 0.001, DS-ME에 비해 0.021 증가하였다. 표 3은 결과 영상 1프레임을 생성하는데에 소요되는 평균시간을 초 단위로 나타낸 것이다. 실험 영상의 해상도에 따라 소요시간이 크게 달라지기 때문에, 표 3에서는 CIF 시퀀스와 고해상도 시퀀스들의 평균을 각각 표시하였다. CIF 시퀀스에서 소요 시간은 제안하는 알고리듬에서 DME에 비해 평균적으로 0.05초 감소하였으며



(a) (b) (c)

그림 4. CIF해상도 시퀀스의 결과 보간 프레임 (a) DME[5], (b) DS-ME[7], (c) 제안하는 알고리듬
Fig. 4. results images of CIF sequences (a) DME[5], (b) DS-ME[7], (c) proposed

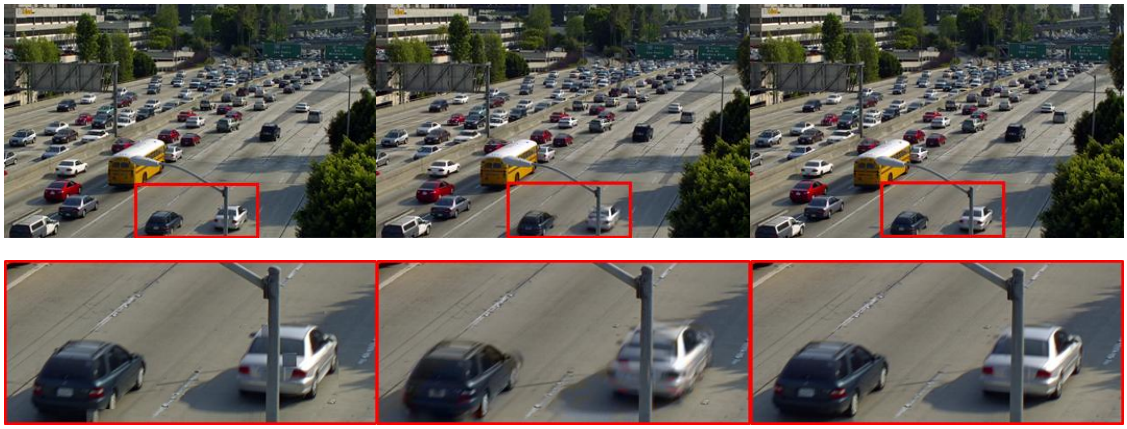


그림 5. Traffic 시퀀스의 결과 보간 프레임 (a) DME[5], (b) DS-ME[7], (c) 제안하는 알고리듬
Fig. 5. results images of Traffic (a) DME[5], (b) DS-ME[7], (c) proposed

DS-ME에 비해 0.11초 증가하였다. 고해상도 영상들에서는 DME에 비해 4.23초 감소하였으나, DS-ME에 비해 2.11초 감소하였다. 결론적으로 객관적으로 결과를 분석하였을 때 DME에 비해 큰 화질 향상은 없었지만 계산량이 크게 감소하였고, DS-ME에 비해서는 계산량은 증가하였지만 PSNR과 SSIM이 크게 향상되었다. 그림 4는 CIF시퀀스들 중 News와 Foreman 시퀀스에서 기존 알고리듬들과 제안하는 알고리듬의 결과 보간 프레임을 보여준다. 비록 DME에 비해 객관적 수치는 크게 향상되지 않았지만, 주관적 화질을 크게 향상되었음을 알 수 있다. DS-ME에 비해서도 주관적으로 선명한 보간 프레임을 얻을 수 있었다. 그림 5는 고해상도 시퀀스들 중 Traffic 시퀀스에서의 결과 영상을 보여준다. 기존 알고리듬들에서는 자동차와 가로등이 겹치는 부분에서 보간을 잘 하지 못했지만 제안하는 알고리듬에서는 비교적 선명하게 보간하였다.

일반적으로 SAD만 사용했을 때는 움직임 추정 결과 탐색 영역 안에서 SAD가 최소가 되는 위치가 움직임 벡터로 선택이 된다. SAD가 작을수록 실제 움직임과 일치할 확률이 높지만, 최소의 SAD가 항상 물체의 실제 움직임을 나타내는 것은 아니다. 제안하는 알고리듬에서는 SAD외에 오차의 분산 값을 또 다른 움직임 판단 기준으로 제시하여 두 개의 값을 이용해 이중으로 움직임 벡터 후보들의 적정성을 판단을 하였다. 결과적으로 더 정확한 움직임을 찾을 수 있었고 PSNR과 SSIM을 사용했을 때 객관적으로 기존 알고리듬들에 비해 향상 된 결과를 얻었다. 실제로 주관적으로 보았을 때도 선명한 보간 영상을 얻을 수 있었다. 또한 제안하는 알고리듬은 움직임이 적은 프레임들에서 적응적으로 작은 탐색영역을 선택하기 때문에 두 번의 움직임 추정을 고정된 탐색영역에서 사용하는 DME와 비교했을 때 계산시간 측면에서 이득이 있었다. 그러나 DS-ME는 고정된 탐색영역을 사용함에도 상대적으로 작은 탐색 영역 (±16 화소)을 사용하기 때문에 제안하는 알고리듬에 비해 계산시간이 적게 소요되었다.

Ⅳ. 결 론

본 논문에서 FRUC에서 보간 프레임의 품질을 향상시키기 위한 새로운 움직임 추정 알고리듬을 제안하였다. 제안하는 알고리듬은 움직임 추정에서 단방향 움직임 추정의 단점을 극복하기 위해 SAD뿐만 아니라 오차의 분산값을 또 다른 기준으로 활용하였고 이 분산 값을 이용하여 부정확한 움직임 벡터를 판단하였다. 그리고 FRUC에서 중요한 파라미터인 탐색 영역을 프레임 단위로 적응적으로 조정하고 화소 단위 기반 홀(hole)을 보간 하였다. 결과적으로 DME[5]에 비해 PSNR과 SSIM은 조금 향상되었으나 계산량은 크게 감소하였고, DS-ME[7]에 비해 계산량은 증가하였으나 객관적 화질은 PSNR에서 최대 3.66 dB, SSIM에서

최대 0.129만큼 증가하였다.

참 고 문 헌 (References)

[1] C. Cafforio, F. Rocca, and S. Tubaro, "Motion Compensated Image Interpolation," *IEEE Trans. Communication*, Vol.38, No.2, pp.215－222, February 1990.

[2] S. Lee, Y. Shin, S. Yang, H. Moon, and R. Park,"Adaptive Motion-Compensated Interpolation For Frame Rate Up-Conversion," *IEEE Trans. Consumer Electronics*, Vol.48, No.3, pp.444－450, August 2002.

[3] S. Lee, O. Kwon, and R. Park, "Weighted-Adaptive Motion Compensated Frame Rate Up-Conversion," *IEEE Trans. Consumer Electronics*, Vol.49, No.3, pp.485－492, August 2003.

[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 Trans. Circuits Syst. Video Technol.*, Vol.17, No.4, pp.407－416, April 2007.

[5] S. Kang, S. Yoo, and Y. Kim, "Dual Motion Estimation for Frame Rate Up-Conversion," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.20, No.12, pp.1909－1914, December 2010.

[6] C. Wang, L. Zhang, Y. He, and Y. Tan, "Frame Rate Up-Conversion Using Trilateral Filtering," *IEEE Trans. Circuits Syst. Video Technol., Vol.*20, No.6, pp.886－893, March 2010.

[7] D. Yoo, S. Kang, and Y. Kim, "Direction-Select Motion Estimation for Motion-Compensated Frame Rate Up-Conversion," *J. Display Technol.*, Vol.9, No.10, pp.840－850, October 2013.

[8] U. Kim, and M. Sunwoo, "New Frame Rate Up-Conversion Algorithms with Low Computational Complexity," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.24, No.3, pp.384－393, March 2014.

[9] N. Kim, D. Jun, J. Lee, and Y. Lee, "Blocking Artifacts Detection in Frequency Domain for Frame Rate Up-Conversion," *Journal of Broadcast Engineering*, Vol.21, No.4, July 2016.

[10] J. Park and J. Jeong, "Frame Rate Up-Conversion Considering the Direction and Magnitude of Identical Motion Vectors," *Journal of Broadcast Engineering*, Vol.20, No.6, November 2015.

[11] H. Kwon and C. Lee, "Efficient Motion Compensated Interpolation Technique using Image Resizing," *Journal of Broadcast Engineering*, Vol.18, No.4, July 2013.

저 자 소 개

유 송 현

- 2015년 2월 : 한양대학교 융합전자공학부 학사
- 2015년 3월 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 석박사
- ORCID : https://orcid.org/0000-0003-0725-9202
- 주관심분야 : 영상압축, 영상처리, 프레임 율 향상, Super Resolution

정 제 창

- 1980년 2월 : 서울대학교 전자공학과 학사
- 1982년 2월 : KAIST 전기전자공학과 석사
- 1990년 : 미국 미시간대학 전기공학과 박사
- 1980년 ~ 1986년 : KBS 기술연구소 연구원 (디지털 및 뉴미디어)
- 1990년 ~ 1991년 : 미국 미시간대학 전기공학과 연구교수 (영상 및 신호처리)
- 1991년 ~ 1995년 : 삼성전자 멀티미디어 연구소 (MPEG, HDTV, 멀티미디어)
- 1995년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 교수 (영상통신 및 신호처리)
- ORCID : http://orcid.org/0000-0002-3759-3116
- 주관심분야 : 영상압축, 영상처리