

# 분할 블록 OBMC와 개선된 적응 가중 중간값 필터를 이용한 프레임률 증가 기법

(Frame Rate Up Conversion Method using Partition Block OBMC  
and Improved Adaptively Weighted Vector Median)

김 근 태<sup>1)</sup>, 고 윤 호<sup>2)\*</sup>  
(Kim Geun-Tae and Ko Yun-Ho)

**요 약** 본 논문에서는 영상의 프레임률을 증가시키기 위한 새로운 움직임 벡터 평활화 기법과 움직임 보상 기법에 대해 제안한다. 제안하는 움직임 벡터 평활화 방법은 적응적 가중 벡터 중간값 필터를 사용하는 기존 방법의 한계를 극복하기 위해 정확도를 반영하여 가중치를 개선함으로써 잘못된 움직임 평활화를 줄인다. 또한 기존 움직임 보상 기법인 OBMC로 보간된 영상의 화질을 향상시키기 위해 원래의 블록을 4등분하여 등분된 각 블록을 기준으로 움직임 벡터를 재추정하고 이를 통해 보간 프레임을 생성하는 방법을 제안한다. 모의실험 결과는 제안된 방법이 객관적 측면과 주관적 측면에서 기존 기법에 비해 우수한 화질을 제공할 수 있음을 보인다.

**핵심주제어** : 프레임률 증가 기법, 움직임 벡터 평활화, AWVM(적응적 가중 벡터 중간값), 움직임 보상, OBMC(겹쳐진 블록 움직임 보상)

**Abstract** This paper proposes a new motion vector smoothing and motion compensation method to increase the frame rate of videos. The proposed method reduces false motion vector smoothing by improving the weight with reflecting accuracy to overcome the limitation of the conventional motion vector smoothing using the adaptively weighted vector median. Also, to improve the interpolated image quality of the conventional OBMC(Overlapped Block Motion Compensation), a scheme that divides an original block into 4 sub-blocks and then generates the interpolated frame using the reestimated motion vector for each sub-block is proposed. The simulation results prove that the proposed method can provide much better objective and subjective image quality than the conventional method.

**Key Words** : Frame Rate Up Conversion, Motion Vector Smoothing, AWVM(Adaptively Weighted Vector Median), Motion Compensation, OBMC(Overlapped Block Motion Compensation)

---

\* Corresponding Author : koyh@cnu.ac.kr

+ 이 연구는 충남대학교 학술연구비에 의해 지원되었음.

Manuscript received January 18, 2019 / accepted February 26, 2019

1) 충남대학교 메카트로닉스공학과, 제1저자

2) 충남대학교 메카트로닉스공학과, 교신저자

## 1. 서 론

오늘날 영상처리 기술은 4차 산업혁명의 대두와 함께 산업 전반에 널리 활용되고 있다[1,2]. 이들 중 프레임률 증가 기법(Frame Rate Up

Conversion)은 원래의 영상렬에 새로운 프레임을 추가하여 단위 시간당 출력되는 프레임의 수를 증가시키는 기술이다. 프레임률 증가 기법은 서로 다른 프레임률을 가지는 영상간의 호환성을 보장하거나 동영상 압축 등에 사용될 수 있다. 한편 LCD(Liquid Crystal Display)의 경우 액정이 느린 응답 속도로 인하여 동영상 재생 시 모션 블러(Motion Blur)가 발생하는데 이러한 문제를 프레임률 증가 기법을 통해 해결할 수 있다. 이에 프레임률 증가 기법은 다수의 LCD 텔레비전 등에 실제 적용되고 있으며 이의 성능을 향상시키기 위한 연구가 지속적으로 진행되고 있다 [3-9].

초기 FRUC 알고리즘에서는 프레임을 단순히 반복하거나 양쪽 프레임들 간의 평균값을 사용하여 프레임을 증가시키는 방법이 존재했다. 이와 같은 방법들의 경우 복잡도가 낮은 만큼 그 구현이 간단하지만 생성된 프레임들은 물체의 경계에서 흐려짐이 발생하고 부자연스러운 객체의 움직임을 보여주기 때문에 화질을 보장할 수는 없다. 이러한 문제들로 인해 대부분의 FRUC 알고리즘의 경우, 영상 내 객체들의 움직임 정보를 사용하여 프레임을 구성한다. 이러한 방법을 ME/MC (Motion Estimation/Motion Compensation)를 이용한 프레임률 증가 기법이라 하는데 초기의 방법에 비해 복잡하지만 보다 좋은 화질을 가지는 프레임을 생성할 수 있다. 일반적으로 ME/MC를 기반으로 하는 프레임률 증가 기법은 크게 움직임 추정, 움직임 보정, 움직임 보상의 3단계로 구성이 된다.

먼저 움직임 추정 단계는 주어진 연속된 인접 프레임들 간의 정보를 통해 영상 내 객체들의 움직임 정보를 계산하는 과정이다. 초기 연구에서 움직임 추정 시 주로 사용된 방법은 BMA(Block Matching Algorithm)이다 [10-13]. BMA로 구한 블록들의 움직임 정보를 토대로 삽입될 보간 프레임을 만드는 과정에서 보간 프레임의 몇몇 블록의 경우 추정된 움직임 벡터의 움직임 궤적이 없거나 여럿 지나갈 가능성이 있다. 이로 인해 중첩(Overlap)과 홀(Hole)이 발생할 수 있는데 이러한 문제를 해결하기 위해 양방향 움직임 추정(Bilateral Motion Estimation, BME)이 제안되

었다 [14]. 해당 방법은 보간 프레임을 기준으로 양방향 움직임을 추정한다. 보간 프레임의 한 블록에 대응하는 블록을 이전 프레임과 현재 프레임에 크기는 같고 방향이 반대인 움직임 벡터를 이용하여 구하게 된다. 하지만 대부분의 움직임 추정 기법들은 블록 내 픽셀값들에 대한 차이를 기반으로 동작을 한다. 따라서 주기적인 패턴이나 픽셀값의 변화가 거의 없는 영상에 대해서는 최적의 움직임 정보를 추정하기 힘들기 때문에 해당 문제를 해결하기 위해 블록 내 화소 값들의 변화율을 기반으로 추정하는 기법이 제안되었다 [15].

움직임 보정 단계는 추정된 움직임 정보들을 수정한다. 대체적으로 잘못 추정된 움직임 벡터의 경우 주변 벡터와의 공간적 일관성이 부족하기 때문에 추정된 움직임 벡터 필드가 지역별로 일관적인 특성을 가지게 함으로써 부정확한 벡터들을 수정할 수 있다. 보통은 이 단계에서 움직임 벡터 필드 내에 존재하는 벡터들 중 크기와 방향이 상이한 이상치(Outlier)를 검출하고 이를 벡터의 각 성분 별로 분리하여 중간값 필터 처리를 하게 된다 [15-18]. 하지만 해당 방법은 다음과 같은 문제점이 있다. 먼저 움직임 추정을 통해 얻어진 벡터와 평활화를 통해 변경하고자 하는 벡터가 실제 이전 프레임과 중간 프레임의 정합을 어느 정도 다르게 보장하는지에 대한 고려가 안 되어 있다. 즉 단순히 벡터의 크기와 방향이 상이하다는 점만을 고려하여 이상치를 검출하고 평활화를 진행하면 해당 벡터에 대한 정합을 보장하는지 알 수 없다. 다음으로 벡터의 x성분과 y성분을 분리하여 중간값 필터를 처리함으로써 실제 주변 블록에서 발생하지 않은 새로운 벡터를 만들어 낼 수 있다. 끝으로 이러한 방법으로 일정 수준의 움직임을 얻기 위해서는 이를 여러 회 반복적으로 적용해야 한다.

이러한 문제를 해결하기 위해 후보 벡터와 다른 벡터들간의 거리를 누적하여 가장 작은 누적합을 가지는 후보 벡터를 중간값으로 선택하는 벡터 중간값(Vector Median) 필터를 사용하는 방법이 제안되었다 [19]. 또한 이를 확장하여 누적합 과정에서 가중치를 사용하는 적응적 가중 벡터 중간값(Adaptively Weighted Vector Median) 필터를 사용하는 방법이 제안되었다 [20]. 해당 기

법을 통해 평활화하고자 하는 움직임 벡터와 주변 움직임 벡터들 중 이상치 인 벡터의 영향을 줄이면서 벡터 중간값 평활화를 수행할 수 있다. 하지만 이러한 방법은 과도한 평활화로 인해 움직임 벡터가 잘못된 방향으로 수정될 가능성이 존재한다.

마지막으로 움직임 보상은 최종적으로 얻은 움직임 정보를 통해 삽입할 보간 영상을 만드는 단계이다. 블록 단위로 움직임 벡터를 찾고 블록 단위로 보간을 진행하기 때문에 주변 움직임 벡터와 상이할 시 블록 구획화 현상(Blocking Artifact)이 발생하게 된다[21]. 이는 블록 내 다른 움직임을 가지는 객체가 여럿 존재할 경우 잘못된 움직임 정보를 얻게 되고 추정된 한가지의 움직임 벡터로 모든 블록내 객체들의 움직임을 표현하면서 생긴다. 이를 완화시키기 위해 OBMC(Overlapped Block Motion Compensation)가 제안되었다[22]. 하지만 OBMC의 경우 해당 블록의 움직임 벡터뿐만 아니라 주변 이웃의 움직임 벡터를 일정한 가중치를 적용하여 함께 사용하므로 인접한 블록의 움직임 벡터가 상이한 경우 경계 부분이 흐려지는 문제가 발생한다. 이와 같은 문제를 줄이기 위해 주변 움직임 벡터들을 보간하고자 하는 블록에 대한 정합 정도를 가중치에 추가하여 OBMC를 진행하는 AOMBMC(Adaptive OBMC)가 제안되었다[23]. 하지만 두 움직임 보상 방법 모두 블록 단위로 보간 프레임을 생성하기 때문에 블록 내 움직이는 객체들을 하나의 움직임 정보로 표현해야 하는 한계로 인하여 보간 프레임에 잔상이 발생한다.

본 논문에서는 프레임을 증가를 위한 새로운 움직임 보정 방법과 움직임 보상 기법을 제안한다. 먼저 움직임 보정 방법의 경우 앞서 언급된 벡터의 각 성분을 분리하여 중간값 필터를 처리함으로써 발생하는 문제를 해결하기 위해 기존 적응적 가중치 벡터 중간값 필터(Adaptively Weighted Vector Median, AWVM)를 개선한 새로운 방법을 제안한다. 적응적 가중치 벡터 중간값 필터는 추정된 움직임 벡터 필드를 평활화하기 위해 정합 기준을 기반으로 하여 적응적으로 가중치를 달리 적용하여 벡터 중간값 필터를 수행하는 방법이다[20]. 본 논문에서는 기존 적응적

가중치 벡터 중간값 필터를 프레임율 증가를 위한 움직임 보정을 위해 적용하는 경우 과도한 벡터 평활화가 이루어질 수 있는 문제점을 지적하고 움직임 벡터의 정합 정도를 고려하는 새로운 적응적 가중치를 적용하는 방법을 제안한다.

움직임 보상 기법의 경우 기존 OBMC가 가지고 있던 잔상과 흐려지는 문제를 해결하기 위해 움직임 추정 시 사용되는 블록 크기로 움직임 보상을 진행하는 것이 아닌 기존 블록을 4등분한 크기의 블록단위로 움직임을 보상하는 방법을 제안한다. 제안된 방법을 위해서는 4등분된 각각의 블록들에 대한 움직임 벡터를 추정해야 한다. 이때 분할된 블록의 움직임 벡터의 경우 기존의 움직임 추정 방법과 같이 일정한 탐색 범위 내에서 모든 벡터들에 대한 정합 정도를 확인하는 전역 탐색(Full Search)이 아닌 등분된 블록을 포함하는 원래 크기의 블록에 대한 움직임 벡터와 주변 인접하는 블록의 움직임 벡터들을 후보 벡터로 설정한다. 그 후 후보 벡터들 중 가장 높은 정합 정도를 가지는 벡터를 선택하여 분할된 블록의 최종 움직임 벡터로 설정한다. 분할된 블록들에 대한 움직임 벡터를 모두 추정한 뒤 OBMC를 진행하여 하나의 블록 내에 움직임이 상이한 다수의 객체들이 존재할 가능성을 줄임으로써 잔상과 흐려지는 문제를 완화할 수 있게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 움직임 보정 방법과 움직임 보상 기법에 대해 소개한다. 3장에서는 제안하는 새로운 움직임 보정 방법과 움직임 보상 기법에 대해 설명한다. 4장에서는 기존 알고리즘과 제안하는 알고리즘의 성능 비교 평가에 대한 결과를 제시하고 5장에서 결론을 맺는다.

## 2. 기존 움직임 보정과 움직임 보상 기법

### 2.1 움직임 보정 기법

프레임을 증가를 위한 기존 움직임 보정 기법은 미리 설정된 기준에 따라 움직임 벡터의 크기와 방향이 상이한 이상치를 검출하고 해당 벡터의 각 성분을 분리하여 중간값 필터 처리하는 것

이다[14]. 하지만 이처럼 각 성분을 분리하여 처리하는 경우 앞서 언급한 바와 같이 실제 존재하지 않는 벡터가 생성되는 등의 여러 문제가 발생할 수 있다. 이에 본 논문에서는 벡터 자체를 중간값 처리하는 방법에 기반하여 새로운 움직임 보정 기법을 제안하고자 한다. 이에 본 소절에서는 벡터 자체를 중간값 처리하기 위한 적응적 가중 벡터 중간값 필터(Adaptively Weighted Vector Median, AWVM)에 대하여 상세히 설명한다[20].

통상적으로 중간값 필터는 주변 신호의 값과 상이하게 다른 임펄스성 노이즈를 제거하는데 좋은 성능을 보이고 있다.  $N$ 개의 데이터 집합  $\{x, x_2, \dots, x_N\}$ 을 마스크 내부의 값들이라고 가정하고 해당 집합 내의 값들 중 중간값을  $x_M$ 이라 할 경우 식 (1)을 만족한다. 중간값 필터는 식 (1)과 같이 마스크 내의 값들 중 중간값을 찾고 선택하는 필터이다.

$$|x_M - x_i| \leq \sum_{j=1}^N |x_j - x_i| \quad j = 1, 2, \dots, N \quad (1)$$

이러한 중간값 필터에 가중치(Weight)를 추가하여 필터의 선택성을 제어하고 필터의 적응성과 범용성을 확장할 수 있다. 이러한 필터를 가중 중간값(Weighted Median, WM) 필터라 한다. 해당 필터는 마스크 내부의  $i$ 번째 데이터인  $x_i$ 에 이 데이터에 대한 가중치  $w_i$ 를 곱하여 중간값 처리한 것으로 정의할 수 있다. 가중치 집합  $W = \{w_1, w_2, \dots, w_N\}$ 에 대하여 집합  $X$ 의 중간값을  $x_{WM}$ 이라 할 경우 식 (2)가 성립한다.

$$\sum_{i=1}^N w_i |x_{WM} - x_i| \leq \sum_{i=1}^N w_i |x_j - x_i| \quad j = 1, 2, \dots, N \quad (2)$$

그리고 중간값 필터를 일반화하여 스칼라 데이터뿐만 아니라 벡터 데이터에도 적용할 수 있게 확장한 것이 바로 가중 벡터 중간값(Weighted Vector Median, WVM) 필터이다. 벡터 집합  $V = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N\}$ 와 가중집합  $W$ 가 주어졌을 때 WVM 필터는 식 (3)과 같이 정의 된다. 여기

서  $\vec{v}_{WVM}$ 은 집합  $V$ 내의 다른 벡터들과의 가중 합된 누적 거리를 최소화한다.

$$\sum_{i=1}^N w_i |\vec{v}_{WVM} - \vec{v}_i| \leq \sum_{i=1}^N w_i |\vec{v}_j - \vec{v}_i| \quad j = 1, 2, \dots, N \quad (3)$$

WVM 필터는 가중치가 고정되어 있는 반면 상황에 따라 그 가중치의 값이 변화하는 적응 가중 벡터 중간값(Adaptively Weighted Vector Median, AWVM) 필터가 소개되었다[20]. [20]에서는 추정된 움직임 벡터의 평활화를 위해 식 (4)와 같은 가중치가 제안되었다. 여기서  $\vec{v}_c$ 는 움직임 벡터 평활화를 위해 검사될 블록  $B$ 에 대한 움직임 벡터이고  $\vec{v}_i$ 는 중심 블록  $B$ 와 인접한 8개의 블록 중  $i$ 번째 블록에 대한 움직임 벡터를 의미한다. 블록 매칭 함수(Block Matching Function)  $DFD_B(\vec{v})$ 는 현재 프레임의 블록  $B$ 와 이전 프레임에서 대응하는 블록 간의 SSD(Sum of Squared Difference)값으로 식 (5)와 같이 정의된다. 식 (5)에서  $s$ 는 블록 내의 한 픽셀이고,  $\vec{v}$ 는 움직임 벡터 값이다.  $f_{n-1}$ 은 이전 프레임,  $f_{n+1}$ 은 현재 프레임을 의미한다. 식 (4)를 식 (3)에 적용하면 AWVM 필터는 식 (6)과 같이 유도 된다.

$$w_i = \frac{DFD_B(\vec{v}_c)}{DFD_B(\vec{v}_i)} \quad (4)$$

$$DFD_B(\vec{v}) = \sum_s [f_{n-1}(s - \vec{v}) - f_{n+1}(s)]^2 \quad (5)$$

$$\sum_{i=1}^N \frac{DFD_B(\vec{v}_c)}{DFD_B(\vec{v}_i)} |\vec{v}_{WVM} - \vec{v}_i| \leq \sum_{i=1}^N \frac{DFD_B(\vec{v}_c)}{DFD_B(\vec{v}_i)} |\vec{v}_j - \vec{v}_i| \quad (6)$$

(where  $j = 1, 2, \dots, N$ )

식 (4)에서와 같이 정의된 기존 AWVM의 가중치는 최대 1의 크기를 가진다. 이는 움직임 벡터 평활화를 수행할 블록  $B$ 에 대한 움직임 벡터가  $\vec{v}_c$ 이므로 자기 자신을 포함한 인접한 블록에서 추정된 움직임 벡터  $\vec{v}_i$ 를 블록  $B$ 에 적용하여 얻은  $DFD_B(\vec{v}_i)$ 가  $DFD_B(\vec{v}_c)$ 보다 항상 큰

값을 가지기 때문이다. 한편 식 (4)에서와 같이 정의된 가중치는 블록  $B$ 와 인접한 블록의 움직임 벡터들을 벡터 중간값 처리함에 있어 블록  $B$ 에 적용하기 부적합한  $v$ 를 외부자(Outlier)에 가깝게 처리할 수 있는 기능을 제공한다. 예를 들어 블록  $B$ 에 적용하기 부적합한  $\vec{v}_i$ 의 경우  $DFD(\vec{v}_i)$ 가 크게 나타나고 이는 식 (4)의 가중치를 낮게 만듦으로써 식 (6)의 조건을 만족시키는  $\vec{v}_{AWVM}$ 을 찾는 과정에서 해당 벡터의 영향을 상대적으로 감소시키게 된다. 하지만 이러한 과정은 외부자를 제외한 벡터들에 대해 단순히 벡터 중간값 필터를 적용하는 것과 같다. 따라서 평활화를 통해 얻어지는 움직임 벡터가 블록  $B$ 에 적용하기에 적정한 것인지를 적극적으로 반영하는데 한계가 있어 잘 추정된 움직임 벡터가 바르지 않은 방향으로 평활화가 될 가능성이 존재한다. 본 논문에서는 이러한 문제를 해결하기 위한 새로운 가중치 설정 방안을 제안하고자 한다.

## 2.2 움직임 보상 기법

기본적인 움직임 보상 기법은 해당 블록의 움직임 벡터만을 이용하여 보간된 영상을 생성한다. 이에 비해 OBMC(Overlapped Block Motion Compensation)는 주변 블록의 움직임 벡터를 일정한 윈도우 가중치만큼 적용하여 움직임을 보상하게 된다[22]. 구체적으로 블록을 4등분하여 각각의 블록에 인접하는 블록들의 움직임 벡터를 모두 사용한다. 따라서 왼쪽 상단 부분의 블록을 기준으로 보면 해당 블록을 보간하기 위해서는 왼쪽, 위, 왼쪽 위 대각선 방향에 위치하는 블록들의 움직임 벡터를 이용한다. 즉, 4등분 후 왼쪽 상단에 위치한 블록을 보간하기 위해서는 해당 블록의 움직임 벡터뿐만 아니라 왼쪽, 위, 왼쪽 위 대각선 방향에 위치하는 블록들의 움직임 벡터를 함께 이용한다. 구체적으로 움직임 보상 시 각 움직임 벡터에 대하여 윈도우 가중치를 적용함으로써 블록화 현상을 줄인다. 보통 사용되는 윈도우 가중치  $w_{p,q}[\ ]$ 는 코사인 형태를 가지며 이러한 OBMC의 과정은 식 (7)과 같이 표현될 수 있다.

$$f_n[s] = \frac{1}{2} \sum_{p=0}^1 \sum_{q=0}^1 w_{p,q}[s] \{ f_{n-1}[s - v_{i+p,j+q}] + f_{n+1}[s + v_{i+p,j+q}] \} \quad (7)$$

이러한 OBMC 기법은 블록간의 움직임 정보가 상이한 경우 보간된 영상의 블록 경계에서 허위의 경계가 나타나는 블록화 열화를 줄일 수 있다. 하지만 각 블록 별 움직임이 실제 움직임을 잘 반영하고 있음에도 인접한 블록 간 움직임이 상이한 경우 Fig. 1에서와 같은 잔상이 발생하게 된다. 이는 하나의 블록 내에 2개 이상의 움직임 특성이 다른 영역이 존재함에도 불구하고 블록 단위로 움직임 보상을 수행하는 데 그 근원적 이유가 있다. 이러한 잔상은 블록의 전체 영역에 작용하게 되므로 사용하는 블록의 크기를 줄인다면 잔상에 의한 영향을 일부 줄일 수 있다. 하지만 크기가 작은 블록을 사용하는 경우 움직임 추정 시 사용할 수 있는 블록 내의 질감 정보가 적어지므로 잘못된 움직임 추정의 가능성이 증가한다. 본 연구에서는 이러한 문제를 극복하기 위한 움직임 보상 방안을 제안하고자 한다.



Fig. 1 Afterimage in Interpolated Frame

## 3. 제안한 방법

### 3.1 개선된 움직임 벡터 보정 기법

앞서 언급한 바와 같이 기존 움직임 벡터 보정 기법은 미리 설정된 기준에 따라 움직임 벡터의 크기와 방향이 상이한 이상치를 검출하고 이상치에 대해서는 벡터의 각 성분을 분리하여 중간값 필터 처리한다. 이러한 방법은 움직임 벡터 보정

후 주변 블록에서 실제 존재하지 않는 움직임 벡터를 평활화 결과로 발생시킬 수 있을 뿐만 아니라 일정 수준 이상의 성능을 얻기 위해 해당 과정을 반복적으로 적용해야 하는 문제점을 가진다. 따라서 본 논문에서는 기존 적응 가중 벡터 중간값 필터를 개선한 적응적 가중 벡터 중간값 필터를 제안한다. 제안된 방법은 식 (8)에서와 같은 새로운 가중치를 사용하는 적응 가중 벡터 중간값 필터를 이용해 움직임 벡터 보정을 수행한다.

$$= \frac{SBAD_B(\vec{v}_j)}{BAD(\vec{v}_i)} \quad (8)$$

프레임을 증가 과정에서 얻은 움직임 벡터는 통상 양방향 움직임 추정을 통해 얻기 때문에 식 (5)에서와 같은  $FD(v)$ 를 사용해 블록 간의 매칭 정도를 확인할 수 없다. 따라서 식 (8)에서 제안된 가중치에서는 양방향 움직임 추정 시 주로 사용되는  $SBAD_B(\vec{v})$ 를 블록간의 매칭 정도를 평가하기 위한 기준으로 사용한다.  $SBAD_B(\vec{v})$ 는 보간 프레임의 한 블록에 대응하는 블록을 이전 프레임과 현재 프레임에 크기는 같고 방향은 반대인 움직임 벡터를 이용하여 구하게 되고 식 (9)와 같이 정의된다[14]. 식 (9)에서  $s$ 는 블록내의 한 픽셀이고,  $\vec{v}$ 는 움직임 벡터 값이다.  $f_{-1}$ 은 이전 프레임,  $f_{n+1}$ 은 현재 프레임을 의미한다.

$$SBAD_B(\vec{v}) = \frac{1}{s} \left| f_{n-1}(s-\vec{v}) - f_{n+1}(s+\vec{v}) \right| \quad (9)$$

기존 AWVM에서 사용된 가중치는 식 (4)에서와 같이 분자에 평활화하고자 하는 블록  $B$ 에 대하여 추정된 움직임 벡터  $\vec{v}_c$ 에 대한 매칭 정보를 공통적으로 사용한다. 기존 기법에서는 모든 가중치가 1이하의 값을 가지도록하기 위하여 이러한 방법이 제안되었다. 하지만 AWVM을 수행하는데 있어 그 가중치가 1이하여야 하는 구속 조건은 반드시 필요한 것이 아니다. 한편 기존 기법은 설정된 가중치를 통해 이상치에 가까운 벡터의 영향을 가능한 줄이면서 벡터 중간값 필터를 처리할 수 있게 한다. 하지만 벡터 중간값

을 통해 얻은 움직임 벡터가 실제 블록  $B$ 에 얼마나 적절한 움직임 정보인지를 적극적으로 고려하지 않고 움직임 벡터 평활화를 수행하게 된다. 따라서 제대로 추정된 움직임 벡터가 지나치게 평활화되어 최선의 움직임 정보를 가질 수 없게 하는 문제점을 가진다. 제안된 방법에서는 식 (8)에서와 같이 가중치의 분자에 후보 벡터  $\vec{v}_j$ 에 대한 매칭 정도를 사용함으로써 AWVM 과정에서 후보 벡터의 적합도를 적극적으로 반영할 수 있게 된다. 최종적으로 제안하는 개선된 AWVM 필터는 식 (10)과 같이 표현할 수 있다.

$$\sum_{i=1}^N \frac{SBAD_B(\vec{v}_j)}{SBAD_B(\vec{v}_i)} |v_{wvm} - x_i| \leq \sum_{i=1}^N \frac{SBAD_B(\vec{v}_j)}{SBAD_B(\vec{v}_i)} |v_j - x_i| \quad (10)$$

(where  $j = 1, 2, \dots, N$ )

### 3.2 개선된 움직임 보상 기법

OBMC는 블록화 현상을 효과적으로 줄일 수 있지만 주변 움직임 벡터를 함께 사용하는 방법이므로 인접한 블록의 움직임 벡터가 상이한 경우 할로(Halo)라고도 일컬어지는 잔상이 발생할 수 있다. 이러한 잔상은 블록 전반에 걸쳐 작용을 하게 되므로 블록의 크기를 줄일 수 있다면 이러한 문제를 완화시킬 수 있다. 하지만 블록의 크기를 줄여 움직임 추정을 하는 경우 블록 내에 충분한 질감에 대한 특징 정보가 부족하여 움직임 추정의 정확도를 크게 저하시킬 수 있다. 따라서 본 논문에서 이러한 문제를 해결하기 위해 상대적으로 크기가 큰 블록을 기준으로 움직임을 추정하고 움직임 보간 직전에 블록을 분할하여 사용하는 분할 블록 OBMC(Partition Block OBMC, PBOBMC) 기법을 제안한다.

블록 분할 OBMC는 Fig. 2(a)에 도시된 기존 OBMC에서 사용하는 블록들을 Fig. 2(b)에서와 같이 4분할하여 각 분할된 블록들에 대하여 OBMC를 수행한다. 이를 위해서는 각 분할된 블록들에 대한 움직임 벡터 정보가 필요한데 이를 얻기 위하여 분할된 각 블록들에 대하여 새로운 전영역 탐색(Full Search)을 수행하지 않고 분할되기 전 자기 자신이 포함된 블록과 인접한 블록

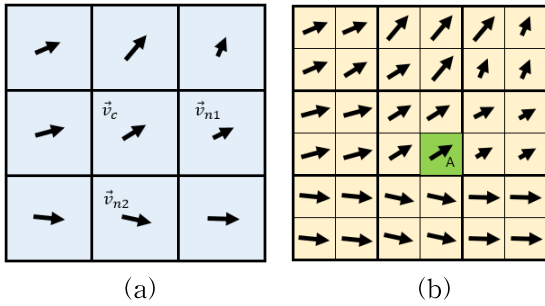


Fig. 2 Block Partition OBMC (a) Motion Vector Field before Partition (b) Motion Vector Field after Partition

들에서 얻은 움직임 벡터를 이용한다. 예를 들어 Fig. 2에서 'A'로 표시된 블록에 대한 움직임 벡터를 추정하는 구체적 과정은 다음과 같다. 분할된 'A' 블록의 실제 움직임은 분할되기 전 자기 자신이 포함된 블록의 움직임 또는 인접한 블록의 움직임과 큰 상관성을 가지게 된다. 따라서 분할되기 전 'A' 블록이 포함된 블록의 움직임 벡터  $v$  와 'A' 블록에 인접한 분할 전 블록의 움직임 벡터  $\vec{v}_{n1}$  과  $\vec{v}_{n2}$  를 후보 벡터로 이용하게 된다. 이들 3가지 후보 벡터를 분할된 'A' 블록에 적용하여 식 (9)의 값을 최소로 하는 벡터를 최종 움직임 벡터로 선정한다. 분할된 블록에 대한 이러한 움직임 추정 방법은 움직임 재탐색을 최소화하여 연산량을 저감시킬 수 있을 뿐만 아니라 질감 정보가 충분하지 않은 분할된 블록이 잘못된 움직임을 추정할 가능성을 원천적으로 차단한다. 모든 분할된 블록에 대하여 이와 같은 방식으로 움직임 벡터를 추정한 후 각 분할된 블록에 대하여 기존 OBMC와 동일한 방식으로 식 (7)에 따라 움직임을 보상한다.

#### 4. 실험 결과 및 고찰

##### 4.1 제안하는 움직임 보정 기법 평가

제안하는 움직임 보정기법에 대한 객관적인 성능평가를 위해 6개의 FullHD급 동영상을 선정하여 사용하였다. 원본 영상에서 n-2번째 프레임과

n번째 프레임을 사용하여 동일한 양방향 움직임 추정 기법으로 움직임 벡터 필드를 추정한 뒤, 각각 기존의 방법과 제안하는 방법으로 움직임 보정을 진행한다. 그 후 제안하는 움직임 보정 방법을 동일하게 적용하여 n-1번째에 대한 보간 영상들을 획득한다. 서로 움직임 보정 방법이 다른 보간 영상들은 원본 영상 n-1번째 프레임과의 PSNR(Peak Signal-to-Noise Ratio)을 이용해 비교하였다. 움직임 추정 시 사용한 블록의 크기는 32x24, 탐색 범위는 128x80이다.

Table 1 Performance Evaluation for the Proposed AWVM Filter

PSNR (dB)	Algorithm		
	proposed method	conventional method #1	conventional method #2
00070	30.660	29.055	28.850
Fighter	37.522	36.666	35.775
Jockey	29.183	27.846	27.392
Pocari	29.483	28.762	27.545
ReadySteadyGo	29.549	27.450	26.269
YachtRide	32.629	30.965	28.135
Average	31.504	30.124	28.994

Table 1은 각 실험 영상에 대하여 각기 다른 움직임 보정 방법에 따른 평균 PSNR을 보여주고 있다. proposed method는 제안하는 개선된 AWVM 기법을 통해 평활화를 진행한 결과이고, conventional method #1은 기존의 AWVM 기법을 통한 결과이고 마지막으로 conventional method #2는 벡터의 성분 별로 각각 중간값 필터를 취하는 기법[15,16]을 통한 결과이다. 모든 영상에 대하여 제안하는 개선된 AWVM이 성능을 향상시키고 있으며 평균적으로 기존의 AWVM 기법에 비해 약 1.4dB의 성능 개선 효과가 있음을 알 수 있다. 이는 제안하는 움직임 보정 방법을 이용했을 때, 움직임 벡터 필드의 분포가 매칭 정도를 반영하여 보다 정확하게 수정되어 이후 움직임 보상단계에서 화질이 개선되었기 때문이다.

Fig. 3은 YachtRide 영상의 보간된 프레임들 중 일부에 대한 화질 평가 비교 결과를 보여주고 있다. 제안한 개선된 적응적 가중 벡터 중간값이 적용된 보간 프레임이 전체적으로 기존의 기법에 비해 높은 성능을 보여주고 있다. 이는 제안한 기법이 기존 기법에 비해 평활화하고자 하는 블록에 대한 움직임 벡터의 정합 정도를 고려하는데 기인한다.

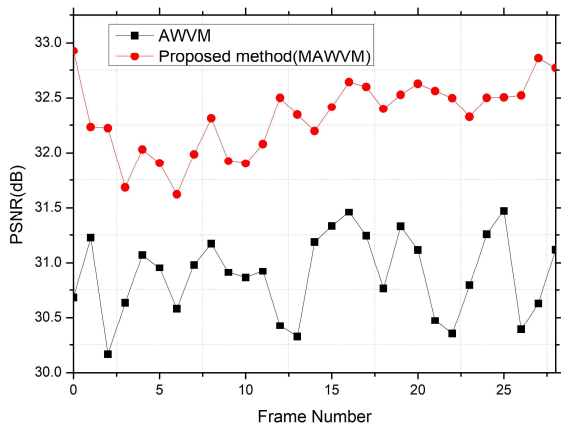


Fig. 3 Image Quality Evaluation for Each Frame of YachtRide

Fig. 4는 제안한 움직임 보정 기법의 보간 영상을 움직임 벡터 필드와 함께 다른 보정 기법과 비교하여 보여주고 있다. Fig. 4에서 각 점들은 보간 영상에서 각 블록의 중심 위치를 나타내고 붉은 선은 추정된 움직임 벡터의 크기와 방향을 표시한다. Fig. 4(a)는 움직임 보정을 하지 않고 추정된 움직임 벡터 정보를 그대로 이용해 보간된 영상이고 Fig. 4(b)는 [15-18]에서 소개된 기존 움직임 보정 기법을 통해 얻은 보간 영상이며 Fig. 4(c)는 [20]에서 소개된 AWVM 필터를 움직임 보정에 적용한 영상이다. 끝으로 Fig. 4(d)는 제안하는 움직임 보정 기법을 통해 얻은 보간된 영상이다. 추정된 움직임 벡터 필드에 존재하는 일부 이상치로 인해 Fig. 4(a)에서와 같이 움직임 보정 과정을 거치지 않은 보간 영상의 경우 화질이 크게 저하됨을 볼 수 있다. 또한 기존 방법들의 경우 움직임 보정을 통한 개선 정확도가 낮아 보간된 영상에 잔상이 존재하는 등의 화질 저하가 발생하는 영역이 발생한다. 하지만 제안

하는 방법의 경우 보정되는 움직임 벡터가 실제 해당 블록에 적절한지를 반영하는 방법이므로 움직임 벡터가 정확하게 보정되어 가장 우수한 주관적 화질을 보여주고 있다.

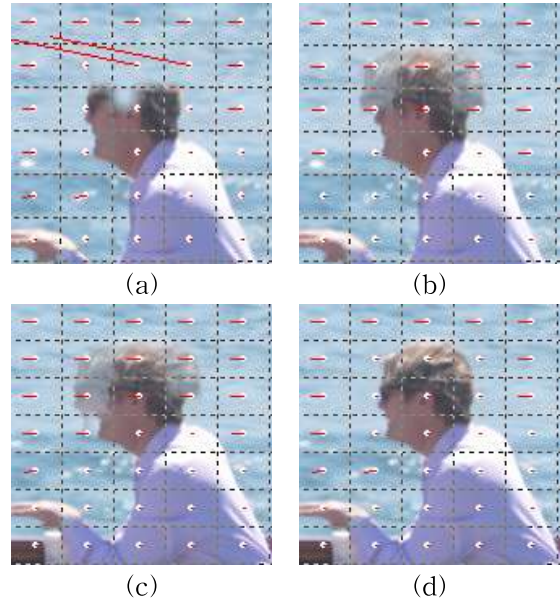


Fig. 4 Motion Compensated Image with Motion Vector Field (a) without Motion Vector Refinement (Original Vector Field) (b) Conventional Motion Vector Refinement[15] (c) Conventional AWVM Filter[20] (d) Proposed AWVM filter

#### 4.2 제안하는 움직임 보상 기법 평가

제안하는 블록 분할 OBMC 움직임 보상 방법에 대한 객관적 성능 평가를 위해 4.1에서와 동일한 6개의 FullHD급 동영상을 사용하여 기존 OBMC의 성능과 비교하였다. 4.1에서와 같이 원본 영상의 n-2번째 프레임과 n번째 프레임을 이용하여 n-1번째 프레임에 대한 보간 영상을 생성한다. 그리고 생성된 n-1번째 프레임과 원본 영상의 n-1번째 프레임간의 PSNR을 구하여 움직임 보상 방법에 대한 객관적 성능을 비교 평가한다. 이때 움직임 보상 방법을 제외한 움직임 추정과 움직임 보정 과정은 동일한 방법을 사용한다. 구체적으로 양방향 움직임 추정을 기반으



로 하여 초기 움직임 벡터 필드를 획득하고 본문에서 제안한 벡터 중간값 필터를 이용하여 움직임을 보정한다.

Table 2 는 각 실험 영상에 대하여 각기 다른 움직임 보상 기법에 따른 평균 PSNR을 보여주고 있다. 전반적으로 제안하는 움직임 보상을 사용하는 경우의 성능이 우수함을 알 수 있고 평균적으로 약 0.3dB의 성능 향상이 있었다. 이는 제안하는 방법이 보다 작은 크기의 분할된 블록을 사용하기 때문에 블록단위로 보간을 진행시 블록 전반에 걸쳐 영향을 미치는 잔상의 영향이 줄어들었기 때문이다.

Table 2 Performance Evaluation for the Proposed Block Partition OBMC

PSNR (dB)	Algorithm	
	proposed method (Block Partition OBMC)	conventional method (OBMC)
00070	30.660	30.490
Fighter	37.522	37.413
Jockey	29.183	29.119
Pocari	29.483	29.242
ReadySteadyGo	29.549	28.844
YachtRide	32.629	32.186
Average	31.504	31.216

Fig. 5는 제안하는 분할 블록 OBMC와 기존 기법인 OBMC로 생성한 보간 프레임들 중 일부 프레임에 대한 화질 평가 결과를 비교하여 보여주고 있다. 전반적으로 모든 프레임에서 제안하는 분할블록 OBMC가 높은 성능을 보임을 알 수 있다. 이는 제안된 방법이 보간 프레임 생성시 보다 작은 블록을 기준으로 정확도 높은 움직임 벡터를 사용하여 보간을 진행했기 때문이다.

Fig. 6은 제안하는 움직임 보상 기법을 통해 보간된 영상의 주관적 화질을 기존 방법과 비교하여 보여주고 있다. Fig. 6(a)는 기존의 움직임 보상 기법인 OBMC를 통해 보간한 영상이고 Fig. 6(b)는 제안하는 움직임 보상 기법을 통해 보간된 영상이다. Fig. 6(b)에서 빨간색으로 표시

된 원의 내부 영역은 기존 방법에 비해 주관적 화질이 크게 개선되는 부분을 표시한 것이다. 제안하는 방법의 경우 보다 작은 블록 영역들에 대하여 정확한 움직임 벡터를 이용하여 움직임 보상이 이루어지기 때문에 기존의 방법에 비해 보간된 영상들에 잔상이 사라져 주관적 화질이 크게 개선되는 것을 확인할 수 있다.

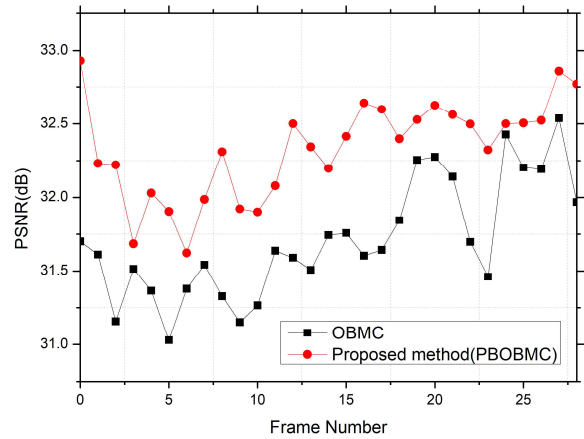


Fig. 5 Image Quality Evaluation for Each Frame of YachtRide

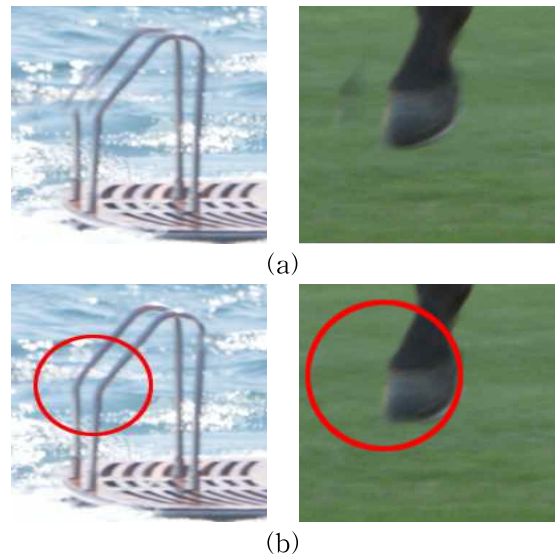


Fig. 6 Subjective Evaluation for the Proposed Block Partition OBMC (a) Conventional OBMC Method (b) Proposed Block Partition OBMC Method

## 5. 결 론

본 논문에서는 프레임률 증가를 위한 움직임 보정 방법과 움직임 보상 방법을 제안하였다. 제안된 움직임 보정 방법인 개선된 적응적 가중 벡터 중간값 필터는 기존 적응적 가중 벡터 중간값 필터가 가지는 과도한 평활화를 막기 위해 움직임 벡터의 정합 정도를 고려하는 새로운 가중치를 이용한다. 이러한 가중치는 과도한 평활화를 방지하여 보다 정확한 움직임 보정을 가능하게 한다. 다음으로 제안된 움직임 보상 방법인 분할 블록 OBMC는 기존 OBMC가 가지는 잔상 현상을 완화하기 위해 움직임 보상 단계에서 분할된 블록을 사용하는 방식이다. 분할된 블록들에 대한 움직임 벡터를 효율적으로 재추정하기 위해 전역 탐색을 대신해 주변 움직임 벡터들의 정합 정도를 계산하여 높은 정합 정도를 가지는 벡터를 선택하여 움직임 보상에 사용할 벡터를 추정한다. 모의 실험을 통해 개선된 적응적 가중 벡터 중간값을 이용한 움직임 보정과 분할 블록 OBMC를 통한 보간 영상 생성 방법의 성능을 기존 기법들과 비교하여 제안된 방법들의 성능이 객관적 화질과 주관적 화질 측면 모두에서 우수함을 입증하였다.

## References

- [1] Lee, T.B. and Kang, H.S., "Material Estimation Method Using Dual-Energy X-Ray Image for Cargo Inspection System," *Journal of the Korea Industrial Information Systems Research*, Vol. 23, No. 1, pp. 1-12, 2018.
- [2] Oh, K.J., Han, D.H., and Kwon, S.K., "Character Floating Hologram using Detection of User's Height and Motion by Depth Image," *Journal of the Korea Industrial Information Systems Research*, Vol. 23, No. 4, pp. 33-40, 2018.
- [3] Lee, Y.H. and Huang, M.R., "Algorithm and Architecture Design of a Hardware-Efficient Frame Rate Upconversion Engine," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 26, No. 11, pp. 2553-2566, 2018.
- [4] Song, W., Heo, P.G., Choi, G., Oh, S.R., and Park, H.W., "Motion Compensated Frame Interpolation of Occlusion and Motion Ambiguity Regions Using Color-Plus-Depth Information," *IEEE International Conference on Image Processing*, pp. 1478-1482, 2018.
- [5] Lee, K., "Adaptive Frame Rate Up-Conversion Algorithms using Block Complexity Information," *Journal of Korea Multimedia Society*, Vol. 21, No. 8, pp. 813-820, 2018.
- [6] Huang, Y.L., Chen, F.C., and Chien, S.Y., "Algorithm and Architecture Design of Multirate Frame Rate Up-conversion for Ultra-HD LCD Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 27, No. 2, pp. 2739-2752, 2017.
- [7] Thang, N.V., and Lee, H.J., "A Semi-Global Motion Estimation of a Repetition Pattern Region for Frame Interpolation," *IEEE International Conference on Image Processing*, pp. 2563-2566, 2017.
- [8] Someya, J., Okuda, N. and Sugiura, H., "The Suppression of Noise on a Dithering Image in LCD Overdrive," *IEEE Transactions on Consumer Electronics*, Vol. 52, No. 4, pp. 1325-1332, 2006.
- [9] Chan, S.H., Wu, T.X., and Nguyen, T.Q., "Comparison of Two Frame Rate Conversion Schemes for Reducing LCD Motion Blurs," *IEEE Signal Processing Letters*, Vol. 17, No. 9, pp. 783-786, 2010.
- [10] Lu, J., and Liou, M.L., "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation," *IEEE*

- Transactions on Circuits And Systems For Video Technology, Vol 7, No. 2, pp. 429-433, 1997.
- [11] Kwon, H., "Multi-Level Motion Estimation Algorithm using Motion Information in Blocks," Journal of Korea Multimedia Society, Vol. 6, No. 2, pp. 259-266, 2003.
- [12] Koo, S.J., and Kim, K.W., Kim, J.H., and Hwang, C.S., "Motion Estimation for Transcoding Using Intermediate Data on the Compressed Video," Proceedings of the Korea Society for Industrial Systems Conference, pp. 288-299, 2001.
- [13] Kim, K.K. and Park, K.N., "An Adaptive Block Matching Motion Estimation Method using Optical Flow," Journal of the Korea Industrial Information Systems Research, Vol. 13, No. 1, pp. 57-67, 2008.
- [14] Akin, A., Cetin, M., Ozcan, Z., Erbagci, B., and Hamzaoglu, I., "An Adaptive Bilateral Motion Estimation Algorithm and its Hardware Architecture," IEEE Transactions on Consumer Electronics, Vol. 58, No. 2, pp. 712-720, 2012.
- [15] Jung, Y.H., Kim, J.H., and Ko, Y.H., "Frame Rate Up Conversion Method using Bilateral Motion Estimation Based on Texture Activity and Neighboring Motion Information," Journal of Korea Multimedia Society, Vol. 17, No. 7, pp. 797-805, 2014.
- [16] Kang, S.J., Cho, K.R., and Kim, Y.H., "Motion Compensated Frame Rate Up-Conversion Using Extended Bilateral Motion Estimation," IEEE Transactions on Consumer Electronics, Vol. 53, No. 4, pp. 1759-1767, 2007.
- [17] Park, J.G., Bae, C.Y., Lee, K.J., and Jeong, J.C., "Frame Rate Up-Conversion Considering the Direction and Magnitude of Motion Vectors," Journal of Broadcast Engineering, Vol. 20, No. 6, pp. 328-331, 2015.
- [18] Park, D.J. and Jeong, J.C., "Adaptive Extended Bilateral Motion Estimation Considering Block Type and Frame Motion Activity," Journal of Broadcast Engineering, Vol. 18, No. 3, pp. 342-348, 2013.
- [19] Astola, J., Haavisto, P., and Neubo, Y., "Vector median filters," Proceedings of the IEEE, Vol. 78, No. 4, pp. 678-689, 1990.
- [20] Alparone, L., Barni, M., Bartilini, F., and Cappellini, V., "Adaptively Weighted Vector-Median Filters for Motion-Field Smoothing," IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, Vol. 4, pp. 2267-2270, 1996.
- [21] Park, K.N., "An Effective Postprocessing Algorithm for Block Encoded Images Using Adaptive Filtering and Interpolation," Journal of the Korea Industrial Information Systems Research, Vol. 12, No. 1, pp. 39-45, 2007.
- [22] Choi, B.D., Han, J.W., Kim, C.S., and Ko, S.J., "Motion Compensated Frame Interpolation Using Bilateral Motion Estimation and Adaptive Overlapped Block Motion Compensation," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 4, pp. 407-416, 2007.
- [23] Lee, S.H., Kwon, O.J., and Park, R.H., "Weighted-Adaptive Motion-Compensated Frame Rate Up-Conversion," IEEE Transactions on Consumer Electronics, Vol. 49, No. 3, pp. 485-492, 2003.



김근태 (Kim Geun-Tae)

- 정회원
- 충남대학교 메카트로닉스공학과 공학사
- 충남대학교 메카트로닉스공학과 석사과정
- 관심분야 : 영상신호처리, 컴퓨터비전, 로봇비전



고윤호 (Ko Yun-Ho)

- 정회원
- 부산대학교 전자공학학과 공학사
- KAIST 전기및전자공학과 공학 석사
- KAIST 전기및전자공학과 공학 박사
- 하이닉스반도체 선임연구원
- 충남대학교 메카트로닉스공학과 교수
- 관심분야 : 영상신호처리, 컴퓨터비전, 영상인식, 영상압축