

움직임 벡터 추정을 위한 탐색 영역 결정 방식

준회원 이 민 구*, 정회원 홍 민 철**

A Search Range Decision Algorithm For Motion Vector Estimation

Min-Gu Lee* Associate Member , Min-Cheol Hong** Regular Member

요 약

본 논문은 동영상 압축 방식에서 사용되는 움직임 벡터 추정의 탐색 영역을 적응적으로 결정하는 방식에 대해 제안한다. 일반적인 동영상 압축 방식에서 사용되는 움직임 벡터 예측 방식의 성능은 압축 효율을 결정하는 움직임 벡터 예측을 위한 전처리 과정의 역할을 하는 제안된 동적 탐색 영역 방식은 인접 블록의 움직임 벡터의 통계적 특성에 따라 효율적으로 탐색 영역을 결정하여 영상 화질의 저하 없이 평균 60(%) 이상의 계산량을 절감하게 된다. 제안된 방식의 성능은 실험을 통해서 확인할 수 있었다.

ABSTRACT

In this paper, we propose an adaptive search range decision algorithm for motion vector estimation in video coding. The performance of general motion estimation method in video coding mechanism is evaluated with respect to the motion vector accuracy and the complexity, which is trade-off. The proposed algorithm that plays as a role of pre-processing for motion vector estimation determines the motion search range by the local statistics of motion vector of neighboring blocks, resulting in more than 60(%) reduction of the computational cost without the loss of visual quality. Experimental results show the capability of the proposed algorithm.

1. 서 론

움직임 벡터 추정 방식은 시간적 중복성 (temporal redundancy)을 줄여 압축 효율을 개선시키는 특성으로 인해 MPEG-1, 2, 4 및 H.261, H.263에 사용되고 있으며[1], 대표적인 움직임 벡터 추정 방식으로는 블록 단위로 움직임 벡터 추정을 연산하는 블록 정합 방식(Block Matching Algorithm)으로서 이와 관련된 많은 논문들이 발표되었다[2-8]. 블록 정합 방식은 제한된 탐색 영역 내에서 움직임 벡터를 추정하는 형태에 따라, 전영역 탐색(Full Search) 방식 및 고속 움직임 벡터 추정 방식으로 분류할 수 있다. 전영역 탐색 방식은 입력 영상 내의 임의의 블록과 이전 압축 영상 내의 제한된 모든 탐색

영역 내에서 움직임 벡터를 추정함으로써 정확한 움직임 벡터를 추정하게 되고, 움직임 보상 영상과 입력 원 영상과의 오차 영상 값을 줄여 압축 효과 측면에서는 장점이 있으나 계산량이 복잡하다는 단점이 있다. 이러한 계산량의 문제점을 해결하고자 많은 고속 움직임 추정 방식들이 제안되었으며, 그 대표적인 방식들로는 TSS(Three Step Search)[2,3], FSS(Four Step Search)[4], 상기 방식의 변형된 고속 움직임 추정 방식[5-7] 및 변형된 모양의 움직임 탐색 기반의 움직임 추정 방식[8-9] 등이 있다. 상기 고속 움직임 추정 방식은 전영역 탐색 방식이 갖고 있는 계산량의 문제점을 크게 개선시켰으나 움직임 벡터 오류에 의해 압축 효율성이 크게 떨어지는 단점을 갖고 있다. 이와 같은 움직임 벡터의

* Net&TV(nkcm@netandtv.com) ** 숭실대학교 (mhong@e.ssu.ac.kr)

논문번호 : 020299-0709, 접수일자 : 2002년 7월 9일

* 본 연구는 정보통신부 기초기술 연구지원 사업 지원으로 수행되었음.

정확성 및 계산량의 절감의 상충되는 특성을 해결하기 위해 동적 탐색 영역을 이용한 고속 움직임 추정 방식들이 최근에 제안되었다[10,11]. 상기 방식들은 공통적으로 임의의 DFD(Displaced Frame Difference) 또는 DBD(Displaced Block Difference)를 임계치로 설정하여 탐색 영역을 결정하는 방식으로 기존의 고속 움직임 추정 방식과 비교하여 압축 효율성은 다소 개선되었으나 계산량 측면에서 상당한 단점을 갖고 있으므로 실시간 시스템에 적용하는데 어려움이 있다.

본 논문에서는 블록 기반의 움직임 벡터 추정 시, 인접 블록의 움직임 벡터의 통계적 특성을 이용하여 적용적으로 움직임 탐색 영역을 결정하는 방식에 대해 제안한다. 일반적인 블록의 움직임 벡터는 인접 블록의 움직임 벡터와 매우 유사한 특성을 갖고 있으므로 인접 블록의 움직임 벡터 특성을 처리하고자 하는 움직임 벡터를 결정하기 위한 탐색 영역 설정에 효과적으로 이용할 수 있고, 이로 인해 계산량의 이득을 취할 수 있다. 제안된 방식은 인접 블록의 정보를 이용하여 국부 통계 특성을 결정하고, 이를 기반으로 움직임 벡터를 추정하고자 하는 블록의 탐색 영역을 적용적으로 결정하게 된다. 제안된 방식은 블록 정합 방식을 위한 전처리 과정 역할을 하는 것으로서, 압축 효율에 영향 없이 계산량을 감소시킬 수 있는 장점이 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 블록 정합 방식의 기본 개요에 대해서 기술하고, 3장에서는 제안된 동적 탐색 영역 방식에 대해 설명하며, 4장 및 5장에서 실험 결과 및 결론으로 본 논문을 맺는다.

II. 블록 정합 방식

동영상 압축 방식에서 사용되는 전영역 탐색 방식은 시간적 중복성을 이전 영상 내에 설정된 $U \times V$ 크기의 블록과 현재 영상내의 블록의 유사성을 특정 메트릭(metric)을 사용하여 비교 및 결정하게 된다. 현재 영상의 블록과 부호화된 이전 영상 사이의 유사성은 일반적으로 다음과 같이 정의되어 결정된다.

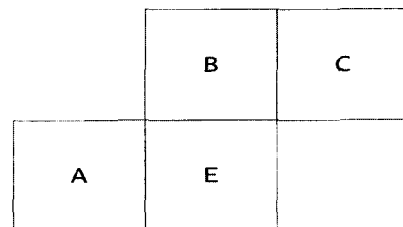
$$SAE(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\widehat{f}_{i-1}(i+jx, y) - f_i(i, j)|, \quad -w \leq x, y \leq w \quad (1)$$

식 (1)에서 \widehat{f}_{i-1} 및 f_i 는 부호화된 이전 압축 영상 및 현재 입력 영상을 나타내고, N 은 움직임

벡터 추정을 위한 임의의 블록의 크기를 나타낸다. 또한, i, j 는 2차원 영상의 수평 및 수직 방향의 위치를, w 는 고정된 최대 탐색 영역을, x, y 는 움직임 벡터의 수평 및 수직 방향의 탐색 지점을 나타낸다. 일반적으로 움직임 벡터 추정은 고정된 최대 탐색 영역 내에서 SAE 값이 가장 작은 x, y 값을 현재 매크로 블록의 수평 및 수직 방향의 움직임 벡터 값으로 결정한다.

대표적인 블록 정합 방식인 전영역 탐색 방식은 최대 탐색 영역 $|w|/2$ 거리에 있는 9개의 탐색 지점에서 SAE를 계산하고 두 번째 단계에서 이중 가장 작은 값을 갖는 위치를 중심으로 첫 단계가 결정된 탐색 지점 거리의 1/2 거리에 있는 8개의 지점에서 SAE를 계산하고 마지막 세 번째 단계에서 이중 가장 작은 값을 갖는 위치를 중심으로 두 번째 단계에서 결정된 탐색 지점 거리의 1/2 거리에 있는 8개의 지점에서 SAE를 계산하여 움직임 벡터를 추정한다. 이와 같이 일반적인 고속 움직임 벡터 추정 방식은 움직임 후보 벡터들을 움직임 벡터의 통계적인 특성에 따라 제한하여 계산량의 이득을 취하였다. 고속 움직임 벡터 추정 방식에서 사용하는 통계적인 특성은 국부 움직임 벡터에 적용하는데 한계가 있으며, 이로 인해 일정 탐색 영역내의 제한된 움직임 후보 벡터를 이용하여 정확한 움직임 벡터를 산출하는데 어려움이 있어 압축 효율의 저하를 초래하게 된다.

식 (1)에서 기술된 블록 정합 방식은 현재 영상의 임의의 블록과 가장 부합되는 움직임 벡터를 이전 부호화 영상의 직사각형 형태의 탐색 영역 내에서 추정하는 반면에, 화상회의 시스템 등의 응용 분야에서 인간의 특성을 고려하여 다이아몬드 형태의 움직임 탐색 영역을 정의할 수도 있다.



A : 좌측 블록 B : 상위 블록
C : 상위 우측 블록 E : 움직임 추정 블록

그림 1. 인접한 블록들의 위치

III. 제안된 방식

일반적인 영상의 움직임 벡터 특성은 움직임이 미비한 경우가 대부분이며 움직임이 큰 경우는 인접된 영역과 유사한 정도의 움직임을 갖는 경우가 대부분인 특성을 나타낸다. 이러한 인접되어 있는 블록 사이의 움직임 벡터의 상관관계를 고려하지 않는 기존의 움직임 추정 방식은 고정된 움직임 탐색 영역 내에서 움직임 벡터를 결정하게 되므로 불필요한 위치를 탐색하여 계산량의 증대를 초래하는 결과를 낳게 된다.

본 논문에서는 블록 단위의 움직임 추정을 수행 중인 블록과 인접 블록들과의 국부 움직임 벡터 통계적 특성을 결정하고자 하는 블록을 E라고 할 때, 이미 움직임 추정을 수행한 블록 중 블록 E와 상관관계가 가장 높은 블록은 A, B, 및 C 블록이며, 각 블록은 수평 및 수직 방향의 움직임 벡터를 이미 결정한 상태이며, 각 블록의 움직임 벡터 값을 $(MV_A_x, MV_A_y), (MV_B_x, MV_B_y), (MV_C_x, MV_C_y)$ 라 가정한다. 또한 식 (1)에서 정의한 바와 같이 최대 탐색 영역을 $|w|$ 라 가정한다. 제안된 동적 탐색 영역 결정 방식은 다음과 같은 4개의 스텝으로 구성되어있다.

첫 번째 과정은 인접 블록 수평 및 수직 방향의 국부 특성을 확인하기 위해 최대 움직임 벡터 영역 값을 식 (2)와 같이 결정한다.

$$\max_{mv_i} = \max(|MV_A_i|, |MV_B_i|, |MV_C_i|) \text{ for } i = x, y, \quad (2)$$

식 (2)에서 $\max(\cdot)$ 는 최대 값을 결정하는 함수이며, $|\cdot|$ 는 절대 값을 의미한다. 식 (2)와 같이 결정된 국부 최대 값을 움직임 벡터 추정을 위한 탐색 영역으로 사용하는 경우, 인접 블록과의 상관관계가 매우 높은 영역에서는 움직임 벡터 예측을 위한 탐색 영역으로 움직임 벡터 오차 값이 작게 되나, 움직임 벡터들 사이의 상관관계가 적은 영역에서는 탐색 영역의 제약성으로 인한 움직임 벡터 오류가 커지게 되어 압축 효율을 저하시키는 결과를 초래하게 된다. 더욱이, 식 (2)와 같이 정의된 결과를 움직임 탐색 영역으로 결정하여 움직임 벡터 오

류가 큰 경우, 인접된 블록의 탐색 영역 결정 과정에 부정적인 영향을 미칠 수 있으므로 부가적인 제약조건이 탐색 영역을 결정하기 위해 정의되어야 한다. 상기와 같은 오류를 최소화시키기 위해, 본 논문에서는 동적 탐색 영역을 결정하기 위한 두 번째 과정으로 인접 블록 움직임 벡터의 통계적 특성을 다음과 같이 정의하여 국부 탐색 영역을 결정하기 위해 사용하였다.

$$\alpha_i = \frac{(|MV_A_i| + |MV_B_i| + |MV_C_i|)}{\text{for } i = x, y} \quad (3)$$

식 (3)의 움직임 벡터 절대 값의 합을 나타내는 국부 특성은 움직임 탐색 영역을 블록 단위로 결정하기 위해 적은 계산량을 요구하며, α_i 값이 작은 경우는 인접한 블록의 모든 움직임 벡터 상관관계가 매우 큼을 의미한다. 이는 그림 2의 블록 E의 움직임 벡터가 '0'에 가까운 벡터를 가진 확률이 매우 높음을 의미하며 움직임 탐색 영역을 작게 설정하여 계산량을 줄일 수 있도록 정의되었다. 반면에 α_i 값이 큰 경우, 상대적으로 한 개 이상의 인접 블록의 움직임 벡터가 큰 값을 갖게 되어 블록간의 상관관계가 낮은 경우이다. 이와 같은 α_i 의 특성을 이용하면, 인접 블록의 움직임 벡터와 유사성이 큰 블록의 탐색 영역은 작게 설정하여도 움직임 벡터 오류가 발생할 확률이 미비하며, 인접 블록과의 상관관계가 작은 특성을 나타내는 경우 탐색 영역을 크게 설정하여 움직임 벡터의 오류를 작게 할 수 있다. 이와 같은 이유로 α_i 에 따라 움직임 탐색 영역을 다음과 같이 결정한다.

$$local_range_i = \begin{cases} \frac{|w|+2}{4}, & \text{for } \alpha_i > 2 \\ \frac{|w|+4}{8}, & \text{otherwise} \end{cases} \quad (4)$$

식 (4)에 의해 결정된 움직임 탐색 영역은 식 (2)의 국부 최대 움직임 벡터 값과 비교하여, 동적 탐색 영역을 다음과 같이 결정한다.

$$dynamic_search_i = \max(2 \times \max_{mv_i}, local_range_i) \text{ for } i = x, y \quad (5)$$

식 (5)에서 \max_mv_i 에 2를 곱해준 이유는 움직임 탐색 영역에 의한 움직임 벡터 추정 오류를 줄이기 위한 과정으로서, 실험적으로 설정되었다.

제안된 방식의 마지막 과정은 최대 탐색 영역의 제약 조건을 부여한 것으로서, 식 (4)에 의해 결정된 수평 및 수직 방향의 탐색 영역은 최대 탐색 영역 $|w|$ 범위를 벗어나는 것을 방지하기 위한 것으로서 다음과 같이 정의된다.

$$\begin{aligned} new_search_i = \\ \min(dynamic_search_i, w) \quad \text{for } i = x, y \end{aligned} \quad (6)$$

식 (6)에서 $\min(.)$ 은 최소값을 결정하는 함수이다.

상기와 같은 방식은 세 개의 인접한 블록들의 움직임 벡터를 이용하여 현재 움직임 벡터를 찾고자 블록의 탐색 영역을 결정하는데 영상의 경계 영역에서는 사용 가능한 인접 블록이 1개 이상 존재하지 않을 수 있다. 상기와 같은 경우, 사용 가능한 인접 블록의 수는 현재 블록 'E'의 탐색 영역을 결정하는 신뢰도에 해당하므로, 인접 블록들 중 2개 이상이 영상 영역 밖에 위치한 경우, 해당 인접 블록의 움직임 벡터의 수평 및 수직 방향 성분은 각각 '0'으로 고려한다. $A(n)$ 을 사용 가능한 인접 블록의 수라 가정하면, 다음과 같다.

$$search_range_i = \begin{cases} new_search_i, & \text{for } A(n) > 1 \\ w, & \text{otherwise} \end{cases} \quad (7)$$

IV. 실험 결과

제안된 방식을 H.263+ TMN8(Test Model Number 8)에 적용하여 각기 다른 해상도, 비트율 및 frame rate에 대해 실험하였고, 고정적인 탐색 영역을 갖는 전영역 탐색 방식과 비교하였다. 본 논문에서는 QCIF 영상 30 frames/sec인 300 frames 의 동영상에 대해, 최대 탐색 영역 $|w|=15$ 로 움직임 벡터 추정하는 동영상 부호화 과정을 통해 10 frames/sec로 압축한 Foreman과 News에 대해 기술하기로 한다. 제안된 알고리즘의 성능을 측정하기 위해 PSNR(Peak-Signal to Noise Ratio)이 사용되었으며 제안된 알고리즘의 계산량을 측정하기 위해 실험한

동영상의 각 블록 당 평균 탐색 지점 수를 이용하였다. PSNR 및 계산량 비교를 위해 사용된 수식은 다음과 같다.

$$PSNR = 10 \log \frac{MN \times 255^2}{\|f - \tilde{f}\|^2}, \quad (8)$$

$$Relative\ Operation(\%) = \frac{\text{motion vector encoding time of proposed}}{\text{motion vector encoding time of TMN8}} \quad (9)$$

식 (8)에서 $\|\cdot\|$ 은 Euclidean norm이고 f 와 \tilde{f} 는 각각 원 영상과 추정된 움직임 벡터로 재구성한 영상이다. 그림 2 및 4는 'Foreman' 및 'News' 영상의 비트율에 따른 PSNR 비교를 한 것으로, TMN8에서 사용된 고정 탐색 영역을 사용한 전영역 탐색 방식과 제안된 동적 탐색 영역 결정 방식은 효과적으로 각 블록 단위의 움직임 탐색 영역을 결정함으로써, 움직임 벡터 오류 및 움직임 보상된 영상과 원 영상의 차이인 예측 영상 잉여(prediction residual)에 큰 차이가 없음을 의미한다. 반면에 본 논문에서 제안된 방식은 고정된 탐색 영역을 사용하는 방식과 비교하여 계산량 측면에 큰 장점이 있음을 그림 3 및 그림 5에서 확인할 수 있다. 'Foreman' 영상은 비트율에 따라 지속적으로 평균 65(%)의 계산량이 절감되었으며, 'News' 영상은 평균 90(%)의 계산량이 감소되었음을 확인할 수 있었다. 계산량의 감소 정도는 영상의 움직임 정도에 따라 다르게 나타나며, 위 실험의 경우 'Foreman' 영상이 'News' 영상보다 상대적으로 움직임 정도가 크므로 계산량의 감소는 적은 것으로 해석할 수 있다.

본 논문에서 제안된 동적 탐색 영역 결정 방식은 움직임 벡터 예측을 위한 전처리 과정의 일종으로 화질 및 압축 효율에 손실 없이 계산량을 줄일 수 있는 효율적인 방식이라 할 수 있다.

V. 결론

본 논문에서는 동영상 압축 방식의 고속 움직임 벡터 추정을 위해 움직임 탐색 영역을 적응적으로 제어하는 방식을 제안하였다. 제안된 방식은 블록 단위의 탐색 영역 결정을 위해 인접 블록을 이용한 국부 통계 특성을 정의 및 이용하였으며 평균 60(%) 이상 계산량을 줄이면서 유사한 성능을 유지

되는 것을 실험을 통해 검증할 수 있었다. 제안된 방식은 움직임 벡터 추정을 위한 전처리 과정으로서, 기존의 전영역 탐색 또는 고속 움직임 추정 방식의 성능 저하 없이 계산량을 줄일 수 있는 효과적인 방식이라 할 수 있겠다. 더불어, 계산량 이득의 정도는 영상 간 또는 블록간의 움직임 벡터 상관관계가 높을수록 커짐을 확인할 수 있었다.

현재 움직임의 정도에 따라 더 민감하게 반응하는 탐색 영역에 대해 연구 중에 있다. 탐색 영역을 결정하게 하는 인접한 블록의 움직임 벡터 크기 및 움직임의 방향을 고려한 방식을 이용하여 화질 저하 없이 계산량이 더욱 감소된 결과를 기대할 수 있을 것으로 사료된다.

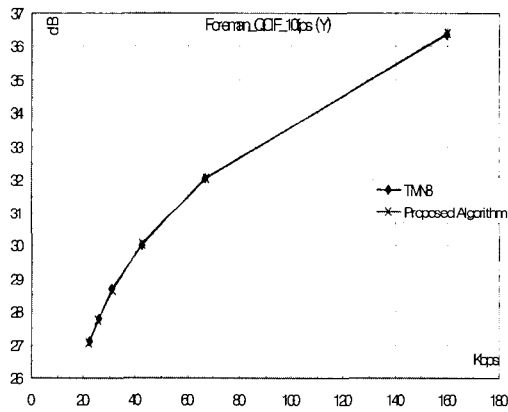


그림 2. QCIF Foreman 동영상의 비트율에 따른 PSNR 비교

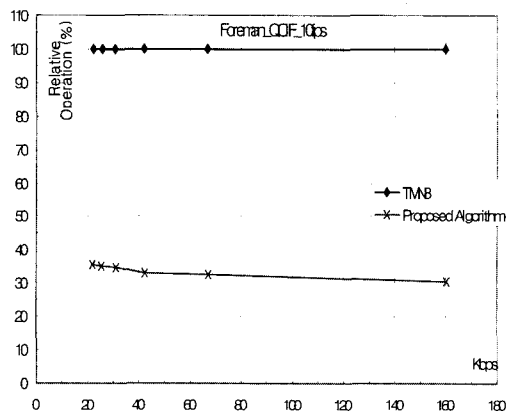


그림 3. QCIF Foreman 동영상의 비트율에 따른 계산량 비교

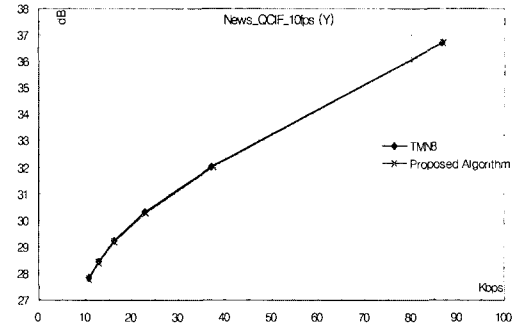


그림 4. QCIF News 동영상의 비트율에 따른 PSNR 비교

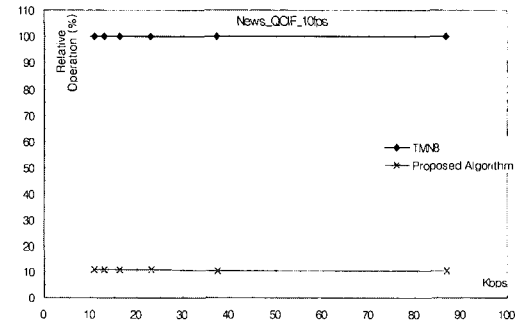


그림 5. QCIF News 동영상의 비트율에 따른 계산량 비교

참고문헌

- [1] V. Bhaskaran and K. Konstantinides, Image and Video Compression Standards, Kluwer Academic Publishers, 1997.
- [2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing." Proc. NTC81, pp.G5.3.1-5.3.5, Dec. 1981.
- [3] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits and Systems for Video T

ech., vol.4, no.4, pp.438-442, Aug. 1994.

[4] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech.*, vol.6, no.3, pp.313-317, June. 1996.

[5] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits and Systems for Video Tech.*, vol.3, no.2, pp.148-157, Apr. 1993.

[6] Y. -L. Chan and W. -C. Siu, "New Adaptive Pixel Decimation for Block Motion Vector Estimation," *IEEE Trans. Circuits and Systems for Video Tech.*, vol.6, no.1, pp.113-118, Feb. 1996.

[7] J. Feng, K. T. Lo, H. Mehrpour and A. E. Kartowiak, "Adaptive Block Matching Algorithm," *Electron. Lett.*, vol.32, pp.1542-1543, Aug. 1995.

[8] J. Y. Tham, S. Ranganath, M. Ranganth, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Tech.*, vol.8, no.4, pp.369-377. Aug. 1998.

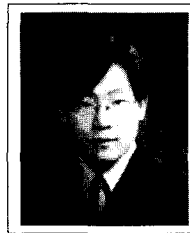
[9] V. Christopoulos and J. Cornelis, "A Center-Biased Adaptive Search Algorithm for Block Motion Estimation." *IEEE Trans. Circuits and Systems for Video Tech.*, vol.10, no.3, pp.423-426, Apr.2000.

[10] L. W. Lee, J. F. Wang, J. Y. Lee and J. D. Shie, "Dynamic search-window adjustment and interlaced search for block-matching algorithm," *IEEE Trans. Circuits and Systems for Video Tech.*, vol.3, no.1, pp.85-87, Feb. 1993.

[11] H. S. Oh and H. K. Lee, "Adaptive adjustment of the search window for block matching algorithm with variable block size," *IEEE Trans. Consumer Electron.*, vol.44, pp.659-666, Aug. 1998.

이 민 구 (Min-Gu Lee)

준회원



2000년 8월 : 숭실대학교 정보통신공학과 졸업
2002년 8월 : 숭실대학교 정보통신공학과 공학석사

<관심분야> 영상 압축

홍 민 철 (Min-Cheol Hong)

정회원



1988년 2월 : 연세대학교 전자공학과 (공학사)
1990년 8월 : 연세대학교 전자공학과 (공학석사)
1997년 9월 : Northwestern University 전기 및 컴퓨터공학과 (공학박사)

1990년 7월-1991년 8월: (주)LG 정보통신 (연구원)
1997년 9월-1998년 8월: Northwestern University (Research Fellow)
1998년 8월-2000년 2월: (주)LG 전자 (선임연구원)
2000년 3월-현재: 숭실대학교 정보통신전자공학부 조교수

<관심분야> 영상복원, 비선형 영상처리 및 필터링, 차세대 동영상 압축 방식, Blind image deconvolution