

프레임 보간을 위한 프레임 차이 기반의 적응형 확장 블록 움직임 추정

준희원 곽통일*, 정희원 조화현*, 윤종호*, 황보현*, 최명렬*

A Motion Estimation Using Adaptively Expanded Block based on Frame Difference for Frame Interpolation

Tong-ill Kwak* *Associate Member*,

Hwa-hyun Cho*, Jong-ho Yun*, Bo-hyun Hwang*, Myung-ryul Choi* *Regular Members*

요 약

본 논문에서는 프레임 보간을 수행하여 영상 끌림 현상(motion blur)을 줄이기 위해 적용되는 프레임 차이 기반의 적응형 확장 블록을 이용한 움직임 추정을 제안한다. 제안된 방법은 상관도 높은 주변 블록의 픽셀 정보를 이용하여 움직임 추정을 수행하기 때문에 정확한 움직임 벡터를 추출할 수 있고, 프레임 차이를 이용하여 연산할 블록과 연산하지 않을 블록을 구분함으로써 연산량을 줄일 수 있다. 제안된 방법은 시뮬레이션 결과를 통하여 기존의 방법들보다 적은 연산량으로 정확한 움직임 벡터를 찾는 것을 확인하였다.

Key Words : Motion estimation, Motion blur, Frame difference, Adaptively Expanded block

ABSTRACT

The hold-type display panel such as a liquid crystal displays(LCD) has problem of motion blur. The problem can be improved by a Frame Rate-up Conversion(FRC) using a frame interpolation. We propose a Motion Estimation(ME) by using adaptively expanded block based on frame difference for FRC. The proposed method is executed using an adaptively expanded block in order to get more accurate motion vector. By using frame difference, we can reduce complexity more significantly than conventional methods. We use quantitative analysis in order to evaluate experimental results. The results show that the proposed method has better performance and lower complexity than conventional methods.

1. 서 론

오늘날 디스플레이 시장에서 Liquid Crystal Display (LCD) 장치는 기술의 발전에 따라 많은 관심을 받고 있다. 그러나 LCD와 같이 hold-type의 디스플레이 장치에서는 동영상에서 화면 끌림 현상(motion blur)이 발생한다^[1-4]. 이 현상은 프레임 단위로 픽셀 값이 유지되기 때문에 생기는 것으로 동

영상에서 움직임 영역에 대하여 화질 열화를 일으킨다. 이와 같은 현상을 줄이기 위하여 기존의 프레임 사이에 새로운 프레임을 보간하여 삽입한다. 프레임 보간을 수행하면 프레임 당 픽셀 값을 유지하는 시간이 감소하기 때문에 화면 끌림 현상(motion blur)이 줄어들게 된다.

프레임 보간 방법으로는 프레임 반복과 연속되는 두 프레임의 평균 프레임을 사용하는 방법, 움직임

* 한양대학교 전자전기 제어계측공학과 SoC/ASIC 연구실(unitetong@asic.hanyang.ac.kr)

논문번호 : KICS2008-01-053, 접수일자 : 2008년 01월 29일, 최종논문접수일자 : 2008년 8월 6일

추정을 이용한 방법 등이 있다. 프레임 반복을 사용하는 방법과 평균 프레임을 사용하는 방법은 간단한 연산 방법으로 구현이 용이하지만 영상 끌림 현상을 이 방법들 중 영상 끌림 현상(motion blur)을 최대한 줄일 수 있는 방법은 움직임 추정을 이용하여 프레임을 보간하는 것이다⁵⁾. 프레임 보간에 적용되는 움직임 추정은 새로운 프레임을 생성하는 과정에서 잘못된 움직임 벡터를 사용하게 될 경우, 영상의 끊어짐 현상이 발생되므로 정확한 움직임 벡터를 찾는 것이 가장 중요하다.

일반적인 움직임 추정 방법은 블록 기반의 전역 탐색 움직임 추정이다. 이 방법은 이전 프레임의 탐색 영역내의 모든 위치에 대하여 블록 기반 움직임 추정을 하는 방식이다⁶⁻⁸⁾. 블록 기반의 전역 탐색 움직임 추정은 기본이 되는 방법이지만, 영상내의 노이즈나 빛의 변화, 카메라 줌인, 줌아웃, 카메라 패닝 등의 현상에 대하여 잘못된 움직임 추정을 수행할 수 있다⁹⁾.

정확한 움직임 추정을 수행하기 위해서 중첩 블록 기반의 움직임 추정이 제안되었다¹⁰⁾. 중첩 블록 기반의 움직임 추정 방법은 움직임 벡터를 찾게 될 블록의 크기보다 확장된 블록을 사용하여 움직임 추정을 수행하는 방법이다. 일반적으로 영상 내에서 이웃하는 픽셀들은 상관도가 높기 때문에 확장된 블록을 기반으로 움직임 추정을 수행하면 노이즈나 카메라 패닝 현상 등에 대하여 움직임 추정이 잘못 수행되는 것을 방지할 수 있다. 그러나 확장된 블록 기반 움직임 추정은 세밀한 부분이나 움직이는 물체와 배경사이의 블록에 대하여 잘못된 움직임 추정을 수행할 수 있다.

본 논문에서는 정확한 움직임 벡터를 추출하기 위하여 적응형 확장 블록을 사용하여 움직임 추정을 수행하였다. 적응형 확장 블록은 프레임 차이를 바탕으로 이웃한 블록의 상관도를 판단하여 높은 상관도를 가지는 블록을 사용하기 때문에 정확한 움직임 벡터를 추출하는 것이 가능하다. 또한 프레임 차이를 기반으로 움직임 추정을 수행할 블록과 수행하지 않을 블록을 결정함으로써 연산량을 줄였다. 제안한 알고리즘의 성능을 평가하기 위하여 기존의 알고리즘인 전역 탐색 움직임 추정 방법과 중첩 블록 기반의 움직임 추정 방법에 대하여 움직임 벡터의 정확도와 연산량을 비교하였다.

본 논문의 구성은 다음과 같다. II장에서 제안한 알고리즘에 대하여 설명하였다. III장에서는 기존의 방법들과 성능을 비교하여 제안한 알고리즘의 성능

을 검증하였다. 마지막으로 IV장에서 본 논문의 결론을 도출하였다.

II. 제안한 움직임 추정 알고리즘

본 논문에서는 정확한 움직임 추정을 수행하고 연산량을 줄일 수 있는 프레임 차이 기반의 적응형 확장 블록을 이용한 움직임 추정 방법을 제안하였다. 제안된 알고리즘은 연속된 프레임에서 구한 프레임 차이를 기반으로 움직임 추정 수행 블록을 결정하고 적응형 확장 블록을 사용하여 움직임 추정을 수행한다. 그림 1은 제안된 방법의 수행 과정을 나타낸다.

2.1 프레임 차이 기반의 움직임 추정

동영상에서 프레임내의 영역은 움직임이 없는 영역과 움직임이 있는 영역으로 분류할 수 있다. 영역을 구분하기 위하여 현재 프레임과 이전 프레임의 픽셀 값의 차이를 이용하여 프레임 차이를 생성한다. 프레임 차이에서 움직임이 없는 영역의 픽셀 값은 이전 프레임의 픽셀 값과 다르지 않기 때문에 ‘0’값을 갖는다. 반대로 움직임이 있는 영역의 픽셀은 이전 프레임의 픽셀 값과 차이가 있으므로 ‘1’ 이상의 값을 가지게 된다. 이를 바탕으로 프레임 차이에서 블록 단위의 픽셀 총합이 임계값 이상인 블록은 움직임이 발생한 블록으로 판단할 수 있고 그렇지 않은 경우는 움직임이 일어나지 않은 블록으로 판단할 수 있다. 임계값은 블록내의 노이즈가 있

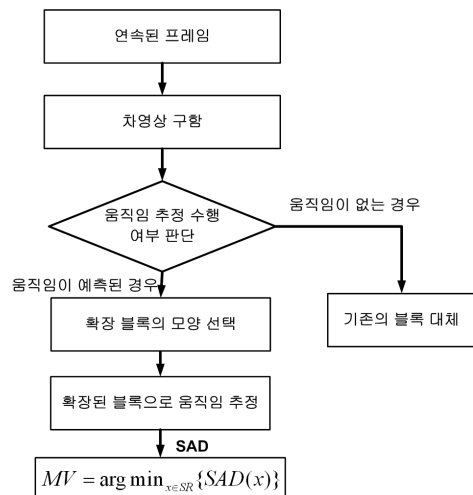


그림 1. 제안된 알고리즘의 순서도
Fig. 1. Flow chart of the proposed algorithm

을 경우와 각 픽셀들이 이전 프레임과 다음 프레임의 픽셀 값이 달라질 수 있음을 고려하여 실험을 통해 결정된다. 움직임 추정은 움직임이 있는 영역에 대해서 수행하는 것이므로, 움직임 여부를 예측하여 움직이는 영역에 대해서 움직임 추정을 수행하면 연산량을 줄일 수 있다. 그러나 카메라의 움직임이 있는 영상의 경우 대부분의 블록들이 임계값보다 큰 값을 가지기 때문에 연산량이 크게 줄지 않을 수도 있다. 이 과정에서 사용되는 프레임 차이는 다음 식(1)과 같이 구해진다.

$$FD(x,t)=f(x,t)-f(x,t-1) \quad (1)$$

여기서 $f(x,t)$ 는 현재 프레임의 픽셀 값을 의미하고, $FD(x,t)$ 는 프레임 차이에서의 픽셀 값을 의미한다. 식(2)는 프레임 차이에서 움직임이 발생한 블록을 판단하는 식이다.

$$FDB(i,t)=\sum FD(i+x,t)>Th \quad (2)$$

여기서 Th 는 임계값을 의미하고, $FDB(i,t)$ 는 프레임 차이에서의 블록 단위의 픽셀 총합을 의미한다.

프레임 차이를 통해 움직임이 일어났다고 판단되는 블록은 적응형 확장 블록 기반의 움직임 추정을 수행하고 움직임이 없다고 판단된 블록은 움직임 추정을 수행하지 않는다. 이로써 움직임 추정의 총 연산량을 줄일 수 있다. 그림 2는 프레임 차이를 기반으로 움직임 추정의 수행 여부를 판단하는 것을 보여준다. 그림 2의 (b)에서 흰색 영역은 움직임이 예측된 부분이고, 가장 어두운 블록들은 움직임이 있는 블록들로 판단되어 적응형 확장 블록을 이용한 움직임 추정을 수행하게 된다.

2.2 적응형 확장 블록 기반의 움직임 추정

기본적으로 움직임 추정을 수행할 때, 움직임을 갖는 물체는 관성을 가지고 있다는 것과 움직이는 물체는 연산되는 블록 크기보다 크다는 것을 가정한다. 이 가정을 근거로 이웃한 블록의 움직임은 현재 블록의 움직임과 상관도가 높다는 것을 알 수 있다. 그러나 움직이는 물체와 배경이 인접해 있는 블록에서는 이웃한 블록의 상관도가 낮기 때문에 움직임 추정을 수행할 때, 사용되어서는 안 된다. 본 논문에서는 프레임 차이를 기반으로 이웃한 블록이 상관도가 큰 블록인지 판단하고, 사용 가능한 블록일 경우, 움직임 추정에 사용할 블록으로 확장시킨다. 적응형 확장 블록은 SAD를 구할 때 사용되며

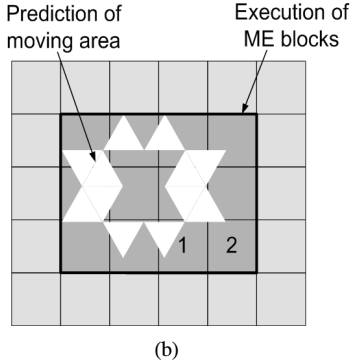
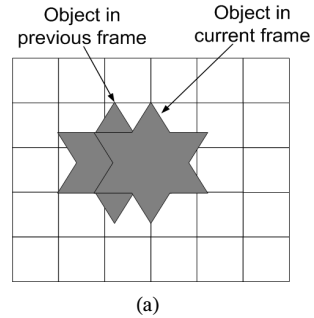


그림 2. 프레임 차이 기반의 움직임 추정 : (a) 연속된 프레임에서의 물체 움직임, (b) 프레임 차이에서 블록의 구분
Fig. 2. Motion estimation based on frame difference : (a) Moving objects in sequence frame, (b) Segmentation of block in frame difference

확장 가능한 블록을 판단하는 과정은 다음과 같다.

- 1) 현재 연산할 블록의 상,하,좌,우 블록에 대하여 프레임 차이에서 블록의 픽셀 총합과 임계값과 비교한다.
- 2) 프레임 차이에서 이웃한 블록의 픽셀 총합이 임계치보다 높은 블록은 연산하려는 블록과 상관도가 큰 블록으로 판단할 수 있으므로 움직임 추정에 사용할 블록으로 확장시킨다.
- 3) 블록의 모양이 결정된 적응형 확장 블록을 이용하여 움직임 추정을 수행하고, 움직임 벡터를 추출한다.

예를 들어 그림 2에서 1번 블록을 적응형 확장 블록을 사용하여 움직임 추정을 할 경우, 좌측 블록과 우측 블록, 상단 블록에 대해서 움직임이 예측된 블록으로 판단하게 되어 적응형 확장 블록으로 결정하게 된다. 결정된 블록을 이용하여 SAD를 구하고 최소인 위치를 가리키는 움직임 벡터를 추출한다. 그림 3은 그림 2의 1번과 2번 블록에 대한 적응형 확장 블록의 예시를 보여준다. 그림에서 블록의 중

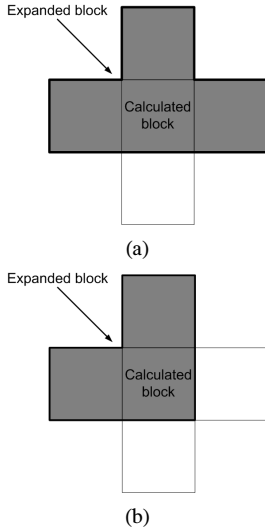


그림 3. 적응형 확장 블록: (a) 그림2의 블록1의 경우, (b) 그림2의 블록2의 경우
 Fig. 3. Adaptively expanded block : (a) Block1 case in Fig.2, (b) Block2 case in Fig.2

심은 현재 연산될 블록을 의미하고, 검은색 블록은 프레임 차이에서 상관도가 높다고 판단된 블록으로 움직임 추정에 사용될 블록을 의미한다.

III. 시뮬레이션

제안한 알고리즘에 대한 성능 평가는 연속된 영상에 대하여 전역탐색법, 확장 블록 기반의 움직임 추정, 마지막으로 본 논문에서 제안한 적응형 확장 블록 기반의 움직임 추정을 적용하여 얻은 결과를 비교하여 검증하였다. 정확한 움직임 추정의 판단은 각 알고리즘으로 추출된 움직임 벡터를 이용하여 만든 영상을 비교함으로써 검증하였고, 연산량에 대한 검증은 제안된 방법에서 연산되지 않는 블록의 개수와 전체 블록의 개수를 비교함으로써 수행하였다.

그림 4는 *Alfred James Pacino* 영상에 대한 시뮬레이션 결과이다. 영상의 크기는 640×352 이고 기본 블록의 크기는 8×8 , 탐색 영역은 16×16 로 정하여 연속된 30 프레임에 대하여 시뮬레이션을 수행하였다. 시뮬레이션 결과에서 사람의 움직임의 경계 부분과 사람의 머리카락 부분에 대해서 기존의 방법들은 잘못된 움직임 벡터를 찾는 반면에 제안된 알고리즘은 정확한 움직임 벡터를 찾는 것을 알 수 있다. 또한 움직임이 없는 영역에 대하여 프레임 차이 기반의 움직임 추정을 수행할 때 기존의 방법들보다 연산량이 감소되는 것을 실험 결과를 통해 알 수 있다.



(a)



(b)



(c)



(d)



(e)

그림 4. *Alfred James Pacino* 영상에 대한 시뮬레이션 : (a) 현재 프레임, (b) 프레임 차이, (c) 전역 탐색 움직임 추정, (d) 중첩 블록 기반 움직임 추정, (e) 제안한 알고리즘
 Fig. 4. Simulation of *Alfred James Pacino* : (a) Current frame, (b) Frame difference, (c) Full search motion estimation, (d) Overlapped block motion estimation, (e) The proposed algorithm

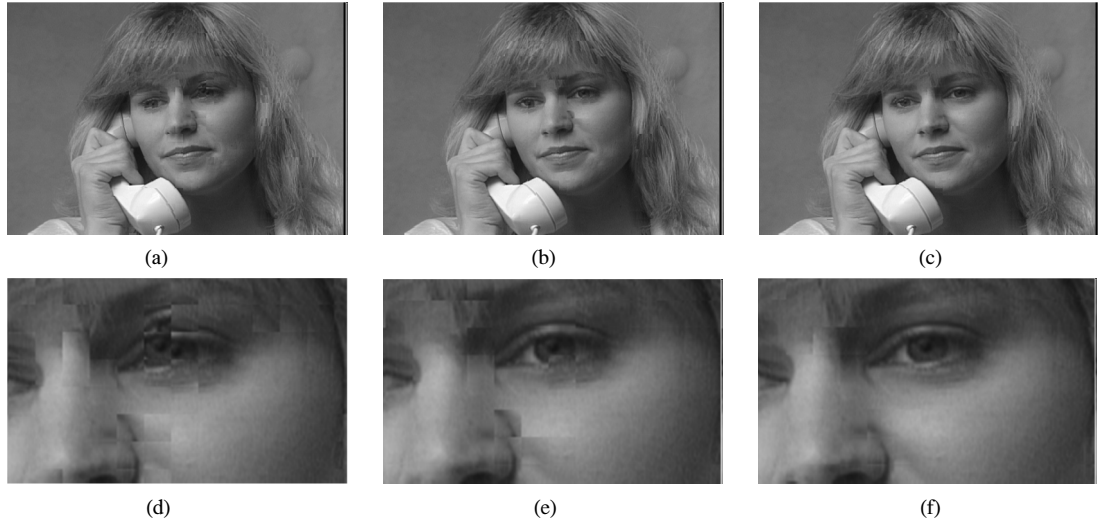


그림 5. *Susie* 영상에 대한 시뮬레이션 : (a) 전역탐색 움직임 추정, (b) 중첩 블록 기반 움직임 추정, (c) 제안한 알고리즘의 움직임 추정, (d) a의 확대 영상, (e) b의 확대 영상, (f) c의 확대 영상
 Fig. 5. Simulation of *Susie* : (a) Full search motion estimation, (b) Overlapped block motion estimation, (c) The proposed algorithm, (d) Expand of a, (e) Expand of b, (f) Expand of c

그림 5는 *Susie* 영상의 시뮬레이션 결과를 보여준다. 영상의 크기는 704×480이고, 연산 블록의 크기는 16×16, 탐색영역의 크기는 32×32로 정하여 연속된 30 프레임에 대하여 시뮬레이션을 수행하였다. 기존의 FSME와 OBME 알고리즘에서 눈 주변과 코 주변에서 잘못된 움직임 벡터를 찾아 부드럽지 못한 결과를 나타낸다. 그러나 제안된 알고리즘의 시뮬레이션 결과를 보면 정확한 움직임을 찾아내는 것을 알 수 있다.

결과의 정량적인 평가는 각 결과 영상에 적용한 PSNR(Peak to Signal Noise Ratio) 값과 연산이 수행되는 블록의 개수를 비교함으로써 수행되었다. PSNR 값은 입력된 영상의 픽셀과 연산 후의 모든 픽셀들에 대한 차이의 합을 의미한다. 그러나 PSNR은 영상 전체에 내포되어 있는 오차에 대한 결과이므로 PSNR의 결과가 절대적인 수치는 아니다. PSNR 값이 높을수록 원본 영상에 가까운 결과로 판단할 수 있지만, 일반적으로 30dB 이상의 결과에 대하여 신뢰성 있는 영상으로 판단한다. PSNR 식은 다음 식(3)과 같다.

$$PSNR = \frac{255^2 * FS}{\sum_x |f_n(x) - f_i(x)|^2} \quad (3)$$

표 1에 각 영상의 30프레임을 적용하여 시뮬레이션 결과에 대한 PSNR과 Complexity를 나타내었다.

연산된 블록 수를 Complexity라고 정의하고 FSME 방법의 연산량을 100%로 가정하였을 때, 각 영상의 연산된 블록 비를 Complexity ratio로 정하였다. Complexity ratio의 결과를 보면 *Susie* 영상의 경우 약 25%로 75%의 연산량 감소가 있음을 알 수 있었고, Alfred James Pacino의 경우 Complexity ratio가 약 48%로 52%의 연산량 감소가 있음을 확인할 수 있다.

IV. 결 론

프레임 보간을 위한 움직임 추정을 수행함에 있어 가장 중요한 관점은 정확한 움직임 벡터를 찾는 것이다. 또한 많은 연산량이 필요한 과정이므로 실시간 처리를 위하여 적은 연산량으로 움직임 추정이 수행되어야 한다. 본 논문에서 제안한 알고리즘은 정확한 움직임 벡터를 찾기 위하여 적응형 확장 블록을 기반으로 움직임 추정을 수행하고, 동시에 프레임 차이를 이용함으로써 연산량을 감소시킨다. 실험 결과를 통하여 제안된 알고리즘이 기존의 전역 탐색 움직임 추정과 중첩 블록 기반 움직임 추정의 방법보다 시각적인 측면에서 더욱 우수한 성능을 나타냄을 확인하였고, 적은 연산량으로 움직임 추정을 수행함을 보였다.

표 1. 시뮬레이션 결과에 대한 PSNR 과 complexity 비교

Sequence name	Average			Complexity ratio (%)
	PSNR (dB)			
	FSME	OBME	Proposed algorithm	
<i>Susie</i>	29.370	28.343	28.508	25.93
<i>flower garden</i>	22.512	21.209	21.524	11.65
<i>Ice</i>	25.889	25.416	26.263	49.67
<i>Alfred James Pacino</i>	39.333	36.822	37.586	48.47

향후 움직임 추정을 통해 구한 움직임 벡터를 사용할 때, 효과적으로 적용하는 방안에 대한 추가적인 연구가 필요하다. 또한 연산 과정이 복잡하기 때문에 효율적인 설계 방안을 검토하여 실시간 처리를 위한 하드웨어 구현이 연구되어야 할 것이다.

참 고 문 헌

- [1] A. A. S. Sluyterman, et al., "What is Needed in LCD Panels to Achieve CRT-like Motion Portrayal?", *Journal of the SID*, Vol.14, No.8, pp.681-686, 2006.
- [2] T. S. Kim, et al., "Response Time Compensation for Black Frame Insertion", *SID 06 DIGEST*, pp.1793-1796, 2006.
- [3] N. Mishima, et al., "Novel Frame Interpolation Method for Hold-Type Display", *2004 ICIP*, pp.1473-1476, 2004.
- [4] E. B. Bellers, et al., "Motion Compensated Frame Rate Conversion for Motion Blur Reduction", *SID 07 DIGEST*, pp.1454-1457, 2007.
- [5] S. J. Kang, et al., "Performance Comparison of Motion Estimation Methods for Frame Rate Up-Conversion", *IDW'06 DES*, pp.1633-1636, 2006.
- [6] K. M. Yang, et al., "A Family of VLSI Designs for the Motion Compensation Block-Matching Algorithm", *IEEE Trans. on Circuit and Systems*, Vol.36, No.10, pp.1317-1325, 1989.
- [7] P. Kuhn, "Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation", *Kluwer Academic Pub*, pp.17-60, 1999.
- [8] I. E. G. Richardson, "Video Codec Design," Wiley, pp.93-107, 2002.
- [9] S. H. Lee, et al., "Motion Vector Correction Based on the Pattern-Like Image Analysis", *IEEE Trans. on Consumer Electronics*, Vol.49, No.3, pp.479-484, 2003.

- [10] T. H. Ha, et al., "Motion Compensated Frame Rate Conversion by Overlapped Block-based Motion Estimation algorithm", *IEEE International Symposium on Consumer Electronics*, pp.345-350, 2004.

곽 통 일 (Tong-ill Kwak)

준회원



2007년 2월 한양대학교 전자컴퓨터공학과 졸업
 2008년 7월 한양대학교 전자컴퓨터공학과 석사과정 재학
 <관심분야> 디지털 영상처리, 디지털 영상 압축, 하드웨어 설계

조 화 현 (Hwa-hyun Cho)

정회원



1999년 2월 한양대학교 전자컴퓨터공학과 졸업
 2001년 2월 한양대학교 전자컴퓨터공학과 석사 졸업
 2006년 2월 한양대학교 전자컴퓨터공학과 박사 졸업
 <관심분야> 디지털 영상처리, 화질 향상 기법, 디지털 영상 압축, 하드웨어 설계

윤 종 호 (Jong-ho Yun)

정회원



2001년 2월 한양대학교 전자컴퓨터공학과 졸업
 2003년 2월 한양대학교 전자컴퓨터공학과 석사 졸업
 2008년 7월 한양대학교 전자컴퓨터공학과 박사과정 재학
 <관심분야> 디지털 영상처리, 화질 향상 기법, 디지털 영상 압축, 하드웨어 설계

황 보 현 (Bo-hyun Hwang)

정회원



2004년 2월 한양대학교 전자컴퓨터공학과 졸업
2006년 2월 한양대학교 전자전기 제어계측공학과 석사 졸업
2008년 7월 한양대학교 전자컴퓨터공학과 박사과정 재학
<관심분야> 디지털 영상처리, 화질 향상 기법, 디지털 영상 압축, 하드웨어 설계

최 명 렬 (Myung-ruyl Choi)

정회원



1983년 한양대학교 전자공학과 학사 졸업
1991년 미시간 주립대학교 컴퓨터공학과 박사 졸업
1991년 7월 한양대학교 전자컴퓨터공학과 박사과정 재학
1991년 3월~10월 생산기술 연구원 전자정보실용화센터 조교수
1991년 11월~1992년 8월 생산기술연구원 산하 전자부품종합기술연구원 선임연구원
1992년~현재 한양대학교 전자컴퓨터공학부 교수
<관심분야> VLSI 설계, 영상처리, RFID, ITS