

논문 2009-46SP-3-3

적응적 탐색기반 움직임 추정을 사용한 프레임 율 변환 알고리즘

(Frame Rate Conversion Algorithm Using Adaptive Search-based
Motion Estimation)

김영덕*, 장준영*, 강문기**

(Young Duk Kim, Joonyoung Chang, and Moon Gi Kang)

요약

본 논문에서는 적응적 탐색기반 움직임 추정을 사용한 프레임 율 변환(FRC : Frame Rate Conversion) 알고리즘을 제안한다. 제안된 움직임 추정은 회귀탐색, 삼 단계탐색(3-SS : 3-Step Search), 그리고 단일예측탐색을 복합적으로 사용하며, 이 세 가지 탐색기법 중 블록 별 영역 특성에 가장 적합한 탐색 기법을 적용한다. 이러한 적응적 탐색방법을 적용함으로써 계산량의 증가를 억제하면서 움직임 추정의 정확도를 향상시킨다. 이를 위해 제안된 기법에서는 시간적 예측을 통해 영상전체를 블록 별 움직임 종류에 따라 3가지 영역으로 분할한다. 제안된 움직임 추정기법을 사용한 프레임 율 변환 알고리즘은 기존 알고리즘에 비해 주관적 및 객관적인 면에서 모두 뛰어난 결과를 보임을 실험을 통해 확인 할 수 있다.

Abstract

In this paper, we propose a frame rate conversion algorithm using adaptive search-based motion estimation (ME). The proposed ME method uses recursive search, 3-step search, and single predicted search as candidates for search strategy. The best method among the three candidates is adaptively selected on a block basis according to the predicted motion type. The adaptation of the search method improves the accuracy of the estimated motion vectors while curbing the increase of computational load. To support the proposed ME method, an entire image is divided into three regions with different motion types. Experimental results show that the proposed FRC method achieves better image quality than existing algorithms in both subjective and objective measures.

Keywords : Frame Rate Conversion (FRC), Adaptive Search, Motion Estimation (ME), Motion Compensation (MC)

I. 서론

텔레비전, 멀티미디어 콘텐츠, 컴퓨터, 그리고 캠코더

와 같은 개인 영상취득 시스템의 통합은 다른 프레임 율을 갖는 영상간의 호환성을 지원하는 프레임 율 변환 알고리즘(Frame Rate Conversion : FRC)을 필수적으로 요구하고 있다. 예를 들면, 24Hz의 프레임 율로 제작된 영화를 60Hz의 프레임 율로 동작하는 텔레비전에서 보기 위해서는 3:2 pull-down과 같은 FRC알고리즘이 필요하다. FRC알고리즘은 영상 포맷간의 호환성 지원 외에도 저 비트 율 동영상 압축 및 LCD상에서의 화질향상에 적용되고 있다. 동영상 압축에서는 부호화시 일부 프레임을 생략하여 데이터 전송량을 줄이며, 생략된 영상들은 복호화기에서 FRC기법을 사용해 보간된다. LCD를 사용하는 영상 재생기기들의 경우, 홀드 타

* 학생회원, ** 정회원, 연세대학교 전기전자공학과
TMS 정보 기술 사업단
(Institute of TMS Information Technology Yonsei University)

※ 본 연구는 한국과학재단의 연세대 생체인식연구센터(BERC)의 지원(R112002105070020(2009))과 지식경제부, 문화체육관광부 및 정보통신연구진흥원의 IT산업원천기술개발사업의 일환으로 수행하였음. [2008-F-031-01, 영상 및 비디오 콘텐츠를 위한 계산사진학 기술 개발]

접수일자:2008년8월13일, 수정완료일:2009년4월13일

입(hold-type) 소자특성에 기인하여 빠른 움직임 영역에서 움직임 블러(motion blur)가 발생된다. 이것은 FRC기법을 사용하여 프레임 율을 높임으로써 개선되고 있다.

프레임 반복이나 프레임 평균과 같은 간단한 FRC 알고리즘은 쉽게 구현될 수 있지만, 움직임 영역에서 부자연스러운 움직임(motion jerkiness) 혹은 움직임 블러를 발생시킨다. 이를 해결하기 위해 움직임 보상(MC : Motion Compensation) 기반 FRC 알고리즘(MC-FRC)^[1~11]들이 제안되었다. MC-FRC 알고리즘은 추정된 움직임 정보가 정확하다면 움직임 궤도상으로의 보간을 통해 고화질의 영상을 생성할 수 있다. 하지만, 움직임 추정의 정확성을 항상 보장하기는 어렵고, 이로 인하여 생성된 부정확한 움직임 벡터들은 블록킹 현상(block artifact)를 야기한다. 블록단위의 에러들은사람의 눈에 잘 띄는 특성이 있으므로, 블록킹 현상은 영상 보간에 있어서 해결되어야 할 중요한 문제이다. 따라서 프레임 율 변환 알고리즘에서는 참 움직임을 추적할 수 있는 정확한 움직임 추정 기법이 필수적이다. 이를 위해 화소단위 및 객체단위 움직임 추정방식이 제안되었지만, 하드웨어의 효율적 구현이 가능한 블록 정합 알고리즘(BMA : Block Matching Algorithm)^[1~12]이 가장 널리 사용되고 있다.

BMA방식의 움직임 추정을 통해 참 움직임 벡터를 생성하더라도, 블록 기반 움직임 모델의 한계로 인하여 블록킹 현상이 발생할 수 있다. 블록내의 모든 픽셀의 움직임이 하나의 움직임 벡터로 기술되므로, 특히 한 블록 내에 다른 움직임을 갖는 둘 이상의 객체가 포함되면 블록킹 현상이 발생할 수 있다. 이러한 블록킹 현상을 제거하고 고화질의 영상을 출력하기 위해 다양한 움직임 보상법이 제안되었다. 이들은 현재 블록에 대한 보상시 현재 움직임 벡터외에 주변 벡터들을 적용하여 여러 움직임 궤도상으로의 보간에 기반을 두고 있다. 널리 사용되고 있는 OBMC(Overlapped Block Motion Compensation)^[7~9]는 블록 경계부분에 대해 여러 움직임 보상의 평균값을 적용함으로써 블록킹 효과를 제거한다. WAMCI^[4](Weighted-Adaptive MC Interpolation)방식은 SAD(sum of absolute differences)의 역수를 사용하여 여러 움직임 보상 결과들의 비중적 평균을 출력한다.

본 논문에서는 적응적 탐색기반 움직임 추정을 사용한 프레임 율 변환(FRC) 알고리즘을 제안한다. 제안된

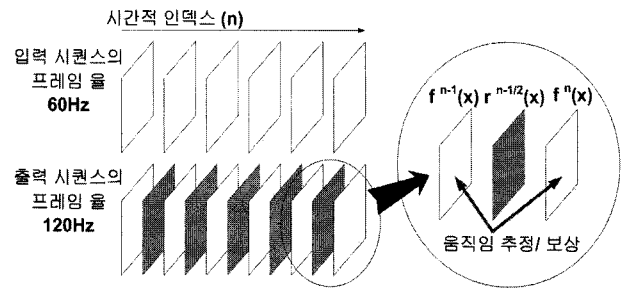


그림 1. 프레임 율 더블링 방식
Fig. 1. Frame rate doubling method.

움직임 추정기법은 블록의 움직임 특성에 가장 적합한 탐색방법을 적용함으로써, 계산 량의 증가를 억제하면서 움직임 추정의 정확도를 향상시킨다. 본 논문에서는 움직임 추정의 부정확성을 보상해 주는 기존의 WAMCI기법을 움직임 보상기법으로 사용한다. 이 움직임 보상기법은 SAD기반 움직임 벡터의 신뢰도를 고려하여 여러 개의 움직임 보상된 결과를 생성하고 이들을 결합시킨다.

본 논문의 구성은 다음과 같다. II장에서는 제안된 FRC알고리즘의 전체 구조를 설명하고, 이어서 제안된 적응적 탐색기반 움직임 추정기법을 자세히 설명하고 사용된 움직임 보상기법을 간략히 설명한다. III장에서는 다양한 영상에 대한 실험 결과 및 기존 알고리즘과의 비교를 제공한다. 마지막으로 IV장에서 논문의 결론을 제시한다.

II. 제안된 FRC 알고리즘

제안된 FRC알고리즘은 어떠한 프레임 율의 변환에도 적용될 수 있지만, 본 논문에서는 검증 및 설명의 편의성을 위해 입력 프레임 율을 두 배로 높이는 프레임 율 더블링 방식으로 제한하여 설명한다. 프레임 율 더블링 과정은 그림 1에 도시되어 있다. 시간적 색인과 공간적 색인을 각각 n 과 x 라 하면, 입력 시퀀스상에 존재하는 이전 프레임 $f^{n-1}(x)$ 과 현재 프레임 $f^n(x)$ 을 사용하여 $n-1/2$ 위치에 $r^{n-1/2}(x)$ 영상을 보간하는 작업이다. 따라서 움직임 추정 및 보상은 모두 $f^{n-1}(x)$ 과 $f^n(x)$ 을 사용하여 진행한다.

제안된 FRC알고리즘의 전체 구조는 그림 2에 도시하였다. 적응적 탐색기반 움직임 추정부에서는 시간적으로 예측된 움직임 종류에 따라서 가장 적합한 탐색방법을 선택하여 실행한다. 움직임 벡터들이 추정되면,

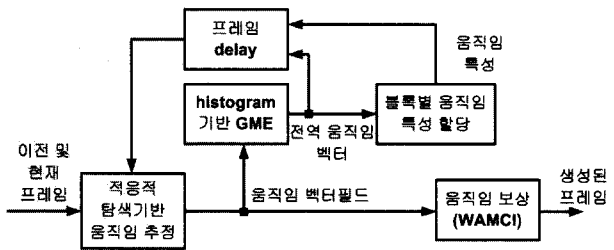


그림 2. 제안된 FRC 알고리즘의 블록도
Fig. 2. Block diagram of the proposed algorithm.

움직임 보상블록은 움직임 벡터의 신뢰도를 고려하여 여러 움직임 궤도상으로의 움직임 보상을 실행한다. 움직임 벡터들은 히스토그램 기반의 전역 움직임 추정에 사용된다. 이것은 가장 빈도수가 높은 움직임 벡터를 전역 움직임 벡터로 출력하여 블록 별 움직임 종류를 결정하는데 사용된다.

2 적응적 탐색기반 움직임 추정

가. 후보 탐색 알고리즘들

적응적 탐색기법은 움직임 추정의 정확도 향상 및 계산 량 감소를 위해 여러 탐색 방법들을 효율적으로 결합하여, 움직임 특성에 따라 블록별로 가장 적합한 탐색 방법을 사용한다. 본 논문에서는 회귀탐색, 예측기반 삼 단계탐색(3-SS with Predictions), 그리고 단일예측 탐색이 사용된다. 회귀탐색은 시공간적으로 예측된 벡터들과 이를 중심으로 하는 작은 크기의 갱신 벡터로 지정된 위치만을 탐색한다. 후보 움직임 벡터들을 시공간적 예측지점 중심으로 제한하여 탐색 위치를 줄이며, 움직임 벡터간의 유사도 및 결과의 정확도를 높일 수 있다. 하지만, 이러한 제한들로 인해 회귀탐색은 주변 블록들과 상당히 다른 움직임을 갖는 객체의 움직임을 추정할 때, 수렴 시간을 필요로 한다.

그림 3은 회귀탐색에서 수렴시간이 필요한 경우를 도시한다. 주어진 영상은 정지 배경과 빠르게 움직이는 자동차로 구성되어 있으며, 보간될 $r^{n-1/2}(x)$ 기준의 블록 그리드로 도시하였다. 도시된 프레임들 바로 이전에 장면전환이 발생했다면, 시간적 예측된 모든 후보 움직임 벡터들은 부정확하다. 또한, 자동차상의 블록 D를 처리하는 경우, 블록별 처리 순서가 왼쪽에서 오른쪽으로, 위에서 아래로 진행되므로, 공간적으로 예측된 후보 벡터들은 정지된 배경 영역에서 제공된다. 따라서 블록 D는 부정확한 움직임 벡터를 생성하게 되며, 이후의 블록들(E, F)도 마찬가지로 부정확한 움직임 벡터들

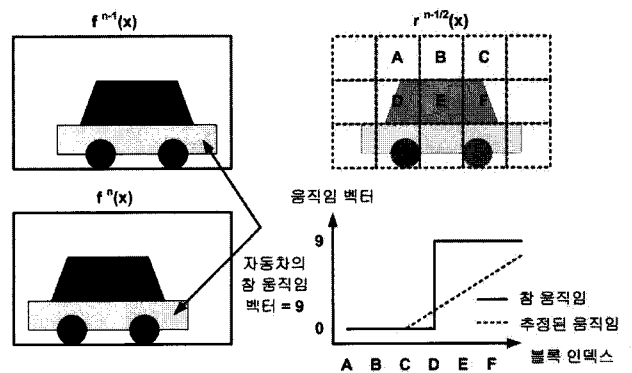


그림 3. 회귀탐색에서의 수렴 시간
Fig. 3. Convergence time in recursive search method

을 갖게 된다. 이것은 회귀탐색에서 수렴시간 때문에 발생하게 되며, 결과적으로 회귀탐색은 수렴시간동안에는 빠르게 움직이는 객체의 경계부분 혹은 객체 전체에 대해서 부정확한 움직임 벡터를 생성할 수 있다.

이러한 수렴시간을 줄이기 위해서는 Full Search(FS)와 같이 움직임의 변화에 즉각적으로 반응할 수 있는 탐색 방법이 필요하다. 계산 량의 문제로 본 논문에서는 FS대신 3-SS를 사용하였다. 그림 3의 블록 D의 경우처럼, 빠른 움직임 객체의 한 블록에 대해 3-SS방식으로 정확한 움직임을 추정할 수 있다면, 이 후 처리되는 주변의 블록들에 대해서도 정확한 움직임을 추정할 수 있다. 회귀탐색의 시공간적 예측을 통해 3-SS로 구한 움직임 벡터를 주변 블록이 참조함으로써 전달된다.

하지만, 3-SS방식은 움직임 벡터의 정밀도를 변경시켜서 탐색시간을 줄이는 방식이므로, 국부 극소점에 빠져 부정확한 움직임 벡터를 생성할 수 있다. 이를 해결하기 위해 제안된 방식에서는 4개의 공간적 예측된 벡터와 2개의 시간적 예측된 벡터를 탐색영역에 포함시킨 예측기반 3-SS방식(3-SS with Predictions)을 사용하였다. 즉 예측에 의해 선정된 후보 움직임 벡터들은

$$CPMV = \left\{ \begin{matrix} v_{i,j-1}^n, v_{i-1,j-1}^n, v_{i-1,j}^n, v_{i-1,j+1}^n, \\ v_{i+2,j-2}^{n-1}, v_{i+2,j+2}^{n-1}, gv^{n-1} \end{matrix} \right\} \quad (1)$$

이다. 여기서 (i,j)와 n은 공간적 블록 색인 및 시간적 색인을 의미하고, v는 움직임 벡터를 의미하고, gv^{n-1} 은 이전 프레임의 전역움직임 벡터를 의미한다. CPMV는 기존 3-SS방법의 첫 번째 단계에서 접근된다. 즉 예측기반 3-SS는 아래와 같이 SP_{1st}^a 을 첫 번째 단계에서 탐색하며 나머지 두 단계는 기존의 3-SS와 동일하게 동작한다.

$$SP_{1st}^a = \left\{ (c,d) \mid \begin{matrix} (c,d) = (4p,4q), \\ (c,d) \in SR \end{matrix} \right\} \cup CPMV \quad (2)$$

여기서 p와 q는 정수를 의미하며, SR은 주어진 탐색 영역(Search Range)을 의미한다. 또한, 효율적인 하드웨어 구현을 위해 회귀탐색도 예측기반 3-SS의 변형으로 구현하였다. 구현된 회귀탐색은 첫 번째 탐색단계에서 아래의 SP_{1st}^b 로 정의된 탐색지점에 대해서 탐색을 진행하며 역시 나머지 두 단계는 기존의 3-SS와 동일하게 동작한다.

$$SP_{1st}^b = \{(0,0)\} \cup CPMV \quad (3)$$

제안된 방식에서는 예측기반 3-SS와 회귀탐색 외에도, 계산량을 줄이기 위해 단일예측탐색을 사용한다. 이 방법은 시간적 예측된 움직임 벡터 $v_{i,j}^{n-1}$ 를 그대로 현재 블록의 움직임 벡터로 사용하는 방법이다. 따라서 연속된 프레임에서 동일한 움직임을 가지는 블록들에 대해서는 효율적인 움직임 추정방식이 된다. 하지만 장면전환, 비선형 움직임, 혹은 빠른 움직임 객체들이 존재하는 경우에는 잘못된 움직임 벡터를 생성할 수 있다. 따라서 예측된 움직임 벡터의 신뢰도를 검사하여, 위와 같은 비정상적인 상황이 있는지를 검사한다. 만약, 예측된 움직임 벡터의 신뢰도가 낮으면 예측기반 3-SS를 실행하여 움직임 벡터를 다시 추정한다. 신뢰도 추정에 필요한 계산량에도 불구하고, 정속의 전역 움직임을 갖는 영상들에 대해서는 단일예측탐색으로 계산량을 줄이는 효과가 있다.

나. 정합 코스트 함수 (matching cost function)

SAD만을 매칭 코스트로 사용하는 방식은 계산량이 적다는 장점이 있지만, 영상에 반복적 패턴이나 유사한 무늬가 존재하는 경우 최소 SAD를 갖는 변위가 항상 참 움직임과 일치하지는 않는다. 또한 빠른 움직임을 추적하기 위해 큰 탐색영역을 사용하는 경우, 추정된 움직임 벡터의 정확도는 떨어지는 경향이 있다. 따라서 제안된 예측기반 3-SS와 회귀탐색에서는 SAD와 함께 주변 움직임 벡터와의 유사도를 사용하여 정확도를 높였다. 즉, 현재 블록의 움직임 벡터 $v_{i,j}^n$ 는

$$v_{i,j}^n = \arg \min_v \left\{ S(v) \times \left(1 + \frac{1}{\lambda} \sum |v - v_{neighbor}^n| \right) \right\} \quad (4)$$

로 구해진다. 여기서 S(v)는 SAD를 나타내고,

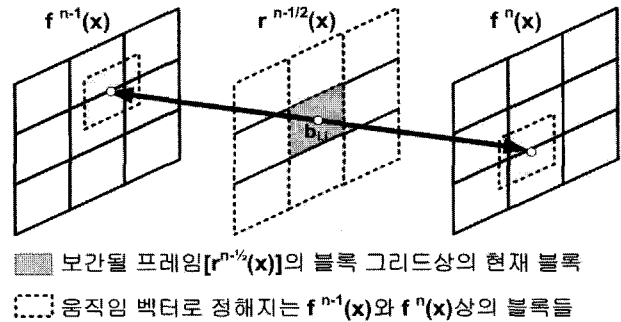


그림 4. 보간될 프레임 중심의 양방향 SAD사용
Fig. 4. Bi-directional SAD represented on the block grid of the frame to be interpolated.

$v_{neighbor}^n$ 는 주변 블록들의 움직임 벡터를 의미하고, λ 는 전체 코스트상의 비율을 조정하는 계수 값이다. 여기서 SAD값은

$$S(v) = \sum_{x \in b_{i,j}} |f^{n-1}(x+v) - f^n(x-v)| \quad (5)$$

처럼 생성될 $r^{n-1/2}(x)$ 의 현재 블록 $b_{i,j}$ 기준으로 대칭적으로 구하여 정확도를 높이도록 구성하였다. 그림 4에는 보간될 프레임 기준의 블록 그리드를 사용하여 양방향 SAD를 적용하는 방식에 대해 도시하였다.

다. 최적화된 탐색방법 결정

제안된 움직임 추정에 사용될 세 가지 탐색방법에 대하여, 전체 영상을 세 가지 움직임 영역, 즉 전역(G), 지역(L), 경계(B) 움직임 영역들로 구분한다. 이 결과는 1 프레임 지연을 거쳐 다음 프레임의 움직임 추정의 종류를 결정하게 된다. 전역움직임 영역은 장면에서 동일한 움직임을 갖는 가장 큰 영역을 의미하여, 보통 정속의 선형움직임을 가지므로 단일예측탐색에 적합하다. 경계움직임 영역은 전역움직임 영역과 비전역움직임 영역의 경계부분으로 정의되어 움직임 벡터의 불연속점이 포함된 영역이다. 따라서 수렴시간을 요구하지 않는 예측기반 3-SS방식이 적합하다. 지역움직임 영역은 이들을 제외한 모든 블록들을 포함한다. 이 영역에서는 회귀탐색이 적용되어 주변 움직임 벡터들과의 유사성이 높은 정확한 움직임 벡터가 생성되며, 예측기반 3-SS이 생성한 움직임 벡터를 참조함으로써 수렴시간 문제를 해결한다.

설명된 움직임 종류별 영역구분은 아래의 식과 같이 먼저 신뢰도가 높고 전역움직임 벡터에 유사한 움직임 벡터를 갖는 블록들은 전역움직임 영역으로 할당되고

나머지 블록들은 지역움직임 영역으로 구분한다.

$$\begin{cases} \text{if } |v_{i,j}^n - gv^n| < th_g \ \&\& \ S(v_{i,j}^n) < TH_a \\ \text{else} \end{cases} \quad \begin{matrix} b_{i,j} \in G \\ b_{i,j} \in L \end{matrix} \quad (6)$$

여기서 gv^n 는 현재 프레임에서 구해진 전역움직임 벡터를 의미하며, th_g 는 전역움직임 벡터와의 유사도를 측정하는 상수 문턱 값을 의미한다. TH_a 는 정수단위 움직임 추정에서 허용될 수 있는 최대 SAD를 의미하는 적응적 문턱 값이다. 보통 고주파 성분이 많은 영역에서는 SAD가 큰 값을 가지며, 저주파로 구성된 평평한 영역에서는 SAD가 작은 값을 가지는 경향이 있다. 이처럼 SAD와 블록 내에 포함된 고주파 정보를 고려하여 허용되는 최대 SAD를 아래와 같이 정의하였다.

$$TH_a = \min \left\{ th_a, \sum_{x \in b_{i,j}} |f^n(x - v_{i,j}^n) - f^n(x - v_{i,j}^n + 1)| \right\} \quad (7)$$

여기서 1 는 $(1,1)^T$ 을 의미하는 벡터이며, th_a 는 상수 문턱 값을 의미한다. th_a 는 잘못된 움직임 벡터 사용시 고주파 성분이 많은 영역을 사용할 경우 부정확한 움직임 벡터를 정확하다고 오판하는 단점을 보완한다. 이와 같이 전역 및 지역움직임 영역으로 초기 구분을 수행한 다음, 영역의 단절된 부분을 처리하기 위해 3×3 블록단위의 마스크를 사용하여 클로징(closing) 처리를 진행한다. 단절된 섬 형태로 전역움직임 영역이 존재하면 경계움직임 영역이 많아지므로 계산량이 많아지는 단점이 있어 이를 방지하기 위함이다. 마지막으로 전역 및 지역움직임 영역의 경계블록들에 대해 경계움직임 영역으로 지정한다.

3 히스토그램기반 전역움직임 추정

블록단위 움직임 벡터들을 사용하여 가장 빈도수가 높은 벡터를 전역움직임 벡터로 정의하게 된다. 움직임 벡터들의 히스토그램 $p_{mv}(i, j)$ 은 아래와 같이 정의된다.

$$p_{mv}(i, j) = \text{number}(v_x = i, v_y = j) \quad (8)$$

여기서 v_x 와 v_y 는 각각 움직임 벡터의 수평과 수직 성분을 의미하며, number 는 해당 움직임 벡터의 개수를 의미한다. 블록단위 움직임 벡터들은 보통 ± 1 정도의 오차를 가질 수 있으므로, 최대 빈도수를 갖는 벡터를 구할 때, 아래와 같이 3×3 윈도우 내의 합의 최대

값을 갖는 벡터를 구하여 안정성을 높인다.

$$gv^n = \arg \max_{(i, j)} \sum_{k=-1}^1 \sum_{l=-1}^1 p_{mv}(i+k, j+l) \quad (9)$$

이와 같이 구해진 전역 움직임 벡터를 사용하여 II장에서 설명된 움직임 영역결정이 이루어지며 이 결과는 1 프레임 지연을 거쳐 다음 프레임의 움직임 추정의 종류를 결정하게 된다.

4 움직임 보상기법

움직임 벡터의 생성이 끝나면 여러 움직임 궤도를 고려한 WAMCI^[4]방식을 사용하여 움직임 보상을 수행한다. 이 기법은 움직임 보상시 현재 블록의 움직임 벡터와 주변 블록의 움직임 벡터를 모두 사용하여 각각의 움직임 보상 결과를 비중적 평균으로 결합하는 방식이다. 즉, 아래와 같이 움직임 벡터의 신뢰도를 고려하여 여러 움직임 궤도상의 움직임 보상 결과를 결합한다.

$$r^{n-1/2}(x) = \frac{\sum_{(k,l)} \left\{ \frac{1}{S(v_{i+k,j+l}^n)} \times g_{k,l}^{n-1/2}(x) \right\}}{\sum_{(k,l)} \frac{1}{S(v_{i+k,j+l}^n)}} \quad (10)$$

여기서 k, l 은 현재 및 주변 움직임 벡터를 표시하는 색인이며, $g_{k,l}^{n-1/2}(x)$ 은 각각의 움직임 벡터를 사용하여 단일 궤도상으로 움직임 보상을 수행한 아래와 같은 결과를 의미한다.

$$g_{k,l}^{n-1/2}(x) = \left(\frac{f^{n-1}(x + v_{i+k,j+l}^n) + f^n(x - v_{i+k,j+l}^n)}{2} \right) \quad (11)$$

식(10)에서는 단일 궤도 움직임 보상 결과들끼리 SAD, 즉 $S(v_{i+k,j+l}^n)$ 의 역수로 결정되는 비중 값을 사용하여 여러 움직임 보상결과들을 결합하여 최종 프레임 생성한다. 이와 같이 움직임 벡터의 신뢰도를 사용하여 여러 $g_{k,l}^{n-1/2}(x)$ 들을 결합함으로써, 현재 움직임 벡터가 부정확하더라도 주변 움직임 벡터들 중에 정확한 움직임 벡터가 존재하면, 정확한 결과를 얻을 수 있다.

III. 실험

제안된 프레임 율 변환 알고리즘은 정지배경, 이동움직임, 카메라 줌, 비선형 움직임 등 다양한 형태의 움직임을 포함한 영상 시퀀스들을 사용하여 검증하였다. 원

표 1. 실험에 사용된 알고리즘 정리

Table 1. Summary of algorithms used in the experiments.

	움직임 추정	MV 처리	움직임 보상	블록크기
BiChoi	2단계 계층탐색	후처리	OBMC	16x16
BMAHa	다운샘플링+ OBME+FS+상 관도 사용	median	OBMC	16x16
3DRS	3-DRS	no	WAMCI	16x8
FSOBMC	FS	no	OBMC	16x16
WAMCI	FS	median	WAMCI	16x16
PRO	제안된 방식	no	WAMCI	16x8

래 비일주사된 필드 시퀀스들은 순차주사화 알고리즘^[13]을 적용하여 프레임 시퀀스를 생성한 후 제안된 프레임 윌 변환기법을 적용하였다. occlusion을 포함한 복잡한 움직임 영상에 대한 검토를 위해 실제 영상들과 인위적으로 생성한 기차 영상을 합성하여 사용하였다. 합성된 영상("trsyn" 접미어 사용)은 다른 속도(느린 속도에서부터 빠른 속도)로 움직이는 여러 개의 기차들로 구성하였다. 실험 결과는 주관적 화질평가 및 객관적 PSNR비교를 통해 검증하였다. 테스트 시퀀스들은 먼저 시간적으로 2배 다운샘플링(down sampling)한 후, 제거된 프레임을 움직임 추정 및 보상을 통해 생성하여 원래 프레임과 비교하여 PSNR값을 구하였다.

검증을 위해 사용한 기존 프레임 윌 변환 알고리즘들은 BiChoi^[1], BMAHa^[6], 3DRS^[12], FSOBMC, WAMCI^[4]로서 모두 보간된 프레임의 블록 그리드를 중심으로 양방향 SAD를 사용하여 구현하였다. 실험에 사용한 탐색 영역의 크기는 수평과 수직 방향으로 ± 16 을 공통적으로 사용하였고, 표 1에는 적용된 움직임 추정/보상기법들에 대해 간단히 정리하였다.

그림 5는 제안된 적응적 탐색기반 움직임 추정기법을 적용하기 위한 영역 분할의 결과를 보여준다. "football" 시퀀스의 31번째 영상 생성 시 얻어진 움직임

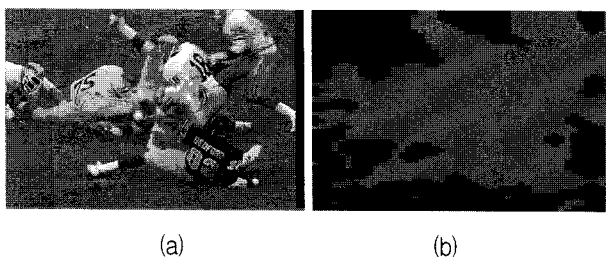


그림 5. 영역분할 (a) "football"영상 (b) (a)의 영역분할
Fig. 5. Region segmentation (a) The 31th picture of "football" sequence (b) Segmented region.

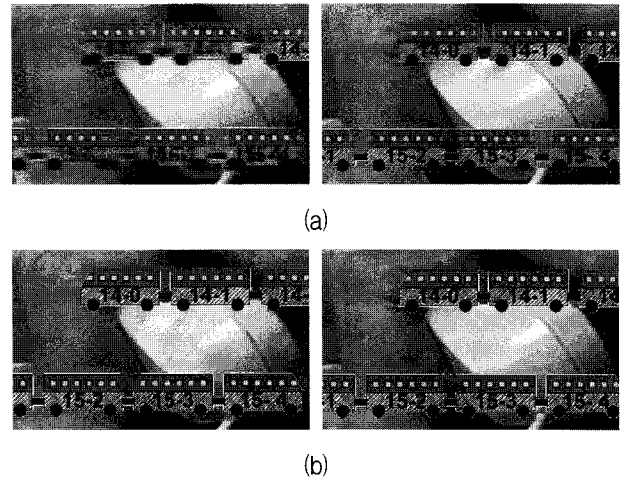


그림 6. susie_trsyn 시퀀스의 처음 2프레임에 대해 생성된 프레임 결과 (a) 3DRS방식 (b) 제안된 방법
Fig. 6. Partially magnified results for the first two pictures of "susie_trsyn" (a) 3DRS method (b) proposed method.

벡터 등의 정보를 사용하여 얻은 결과로서 다음 프레임인 33번째 영상을 생성하기 위한 움직임 추정에 사용된다. 검은 색 영역이 경계움직임 영역(B), 밝은 회색 영역이 지역움직임 영역(L), 그리고 어두운 회색 부분이 전역움직임 영역(G)을 각각 의미한다. 각각의 영역들이 잘 지정되었음을 알 수 있다.

그림 6은 회귀탐색의 단점인 수렴시간을 보여주는 실험결과로서, (a)와 (b)는 각각 3DRS방식과 제안된 PRO방식으로 움직임 추정 및 보상을 수행한 결과를 부분적으로 확대한 영상이다. "susie_trsyn"시퀀스의 처음 2프레임을 생성한 결과들로서 (a)에 표시된 3DRS방식에서는 빠르게 움직이는 기차를 추적하기 위해 처음 두 프레임에서 잘못된 움직임 벡터를 생성하여 문제가 발생함을 알 수 있다. 이에 반해 제안된 알고리즘에서는 정확한 움직임을 통해 올바른 프레임을 생성함을 알 수 있다.

그림 7과 8은 "football_trsyn"시퀀스의 13번째 영상과 "foreman"시퀀스의 143번째 영상에 대해 프레임 윌 변환 기법을 적용한 결과를 보여준다. 그림 7은 빠른 움직임을 갖는 기차를 포함하고 있고 영상에서 선수들의 움직임은 비선형적인 복잡한 움직임을 보이고 있다. 그림8의 경우는 foreman이 카메라를 향해 얼굴을 접근시키는 영상이다. BiChoi방식에서는 상위 계층에서 부정확한 움직임 벡터를 생성하여 하위 계층에 영향을 주어 결국 일부 블록들에서 눈에 띄는 오류를 생성하였다. BMAHa방식에서는 OBME를 통해 상대적으로 큰

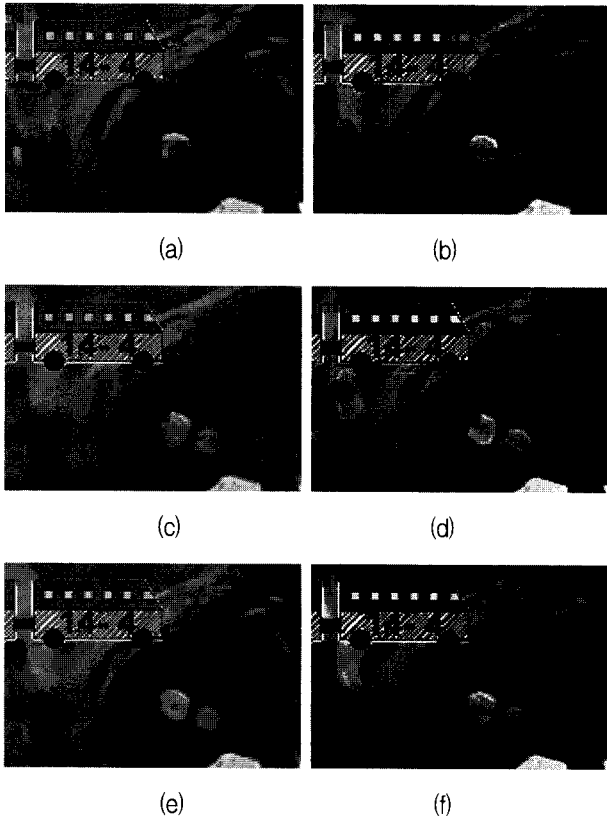


그림 7. “football_trsyn”의 결과 (a) BiChoi (b) BMAHa (c) 3DRS (d) FSOBMC (e) WAMCI (f) PRO
 Fig. 7. Partially magnified results of the “football_trsyn” (a) BiChoi (b) BMAHa (c) 3DRS (d) FSOBMC (e) WAMCI (f) PRO.

블록에 대해 동일 움직임을 갖는다는 가정을 사용함으로써 움직임 경계 상에서 잘못된 움직임 벡터를 생성하는 문제를 보였다. FS를 사용하는 FSOBMC와 WAMCI방식의 경우, 탐색 영역내의 모든 영역에 대해 매칭을 실행함으로써 SAD를 기준으로 국부 극소점에 빠지지 않지만, 영상내의 잡음, 조도의 변화, 그리고 움직임 물체로 인한 occlusion등으로 인해 최소의 SAD를 갖는 변위가 참 움직임 궤도와 일치하지 않기 때문에 프레임 율 변환기법에서는 잘못된 영상보간을 하는 문제를 발생시킨다. 3DRS기법의 경우 인접 블록들과의 유사성에 기반하여 탐색 지점을 제한하고 있으므로 상당히 안정적인 결과를 출력하였으나, 전이시간을 필요로 하는 문제가 있다. 그림 8의 경우 기존 알고리즘들은 모자 윗부분, 입 주위부분, 모자 챙 부분들에서 시각적 왜곡을 보이고 있다. 제안된 기법은 움직임 경계상에서는 전이시간이 필요 없는 3-SS방식을 적용하고, 이를 참조하여 회귀탐색을 실행함으로써 가장 자연스럽게 시각적 왜곡이 적은 프레임을 생성하였다.

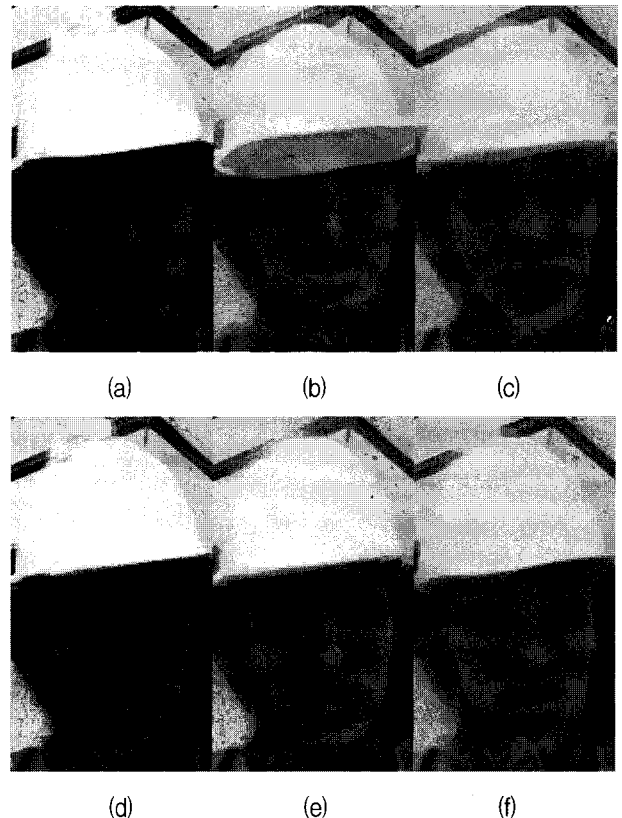
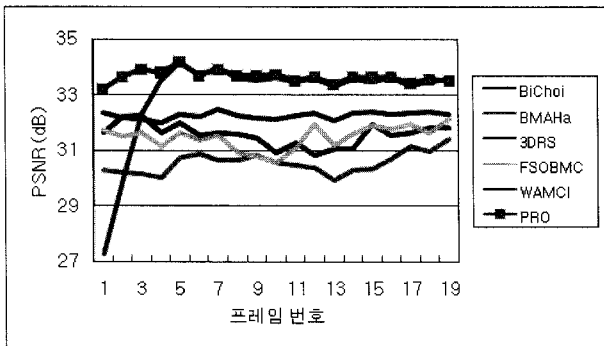


그림 8. “foreman”의 결과 (a) BiChoi (b) BMAHa (c) 3DRS (d) FSOBMC (e) WAMCI (f) PRO
 Fig. 8. Partially magnified results of the “foreman” (a) BiChoi (b) BMAHa (c) 3DRS (d) FSOBMC (e) WAMCI (f) PRO.

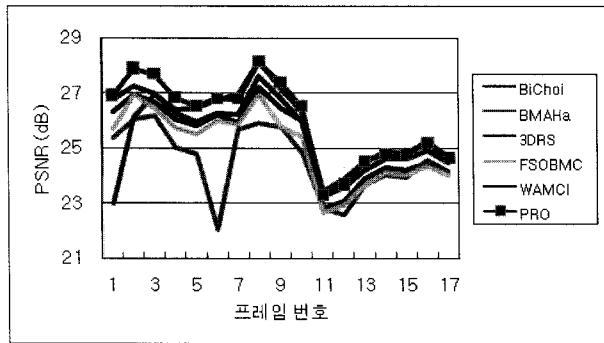
그림 9는 합성 시퀀스(“trsyn”접미어)들에 대해 PSNR을 측정된 결과들을 보여준다. 그림에서 볼 수 있듯이 제안된 방식이 기존의 다른 프레임 율 변환 알고리즘에 비해 정확히 프레임을 생성함을 알 수 있다. 3DRS방식의 경우 수렴시간 이후에는 상당히 높은 PSNR결과를 보이지만, 시퀀스의 초기 몇 프레임동안은 수렴시간으로 인해 상당히 낮은 PSNR결과를 보였다. 결론적으로, 제안된 알고리즘은 수렴시간문제를 해결하여 정확한 움직임 벡터를 생성함을 알 수 있다.

표 2에는 실험에 사용된 영상 시퀀스에 대한 평균 PSNR값을 표시하였다. 시퀀스 별 최상의 PSNR값은 굵은 숫자를 사용하여 표기하였으며, 표에 도시된 것처럼, 제안된 방식이 대부분의 영상에서 좋은 결과를 보임을 확인할 수 있다.

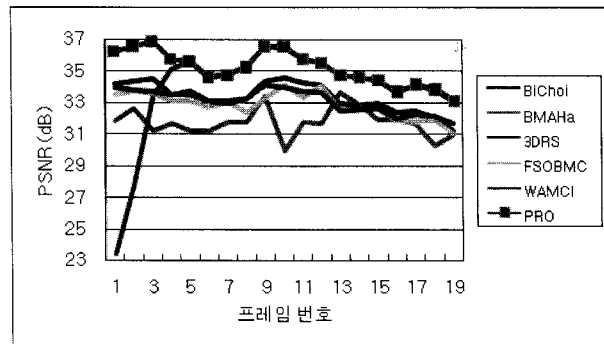
표 3에는 수평과 수직 방향으로 ± 16 의 탐색영역을 사용하였을 때 움직임 추정 알고리즘들의 블록매칭에 필요한 계산량을 표시하였다. BiChoi기법에서 2레벨 계층구조를 생성하는 과정, BMAHa에서 전역 움직임 추



(a)



(b)



(c)

그림 9. PSNR 결과비교 (a) "mobile_trsyn" (b) "pingpong_trsyn" (c) "susie_trsyn"
Fig. 9. PSNR comparisons (a) "mobile_trsyn" (b) "pingpong_trsyn" (c) "susie_trsyn".

정을 수행하는 과정, PRO방식에서 블록별 움직임 영역을 선별하는 과정 등에 대한 계산량은 제외하였다. 표 3의 결과를 보면 1회 탐색당 소요되는 계산량은 거의 유사한데 비해 블록당 탐색횟수의 차이가 있음을 알 수 있다. 제안된 기법의 경우 블록의 움직임 특성에 따라서 탐색횟수가 변경되므로 실험에서 구해진 데이터를 기반으로 평균값으로 사용한다. 여기서 BG는 전역 움직임 영역으로 지정되었으나 시간적 예측의 실패로 간주되어 예측기반 3-SS기법이 적용된 영역을 의미한다. 실험에서 얻어진 통계치는 각각 4영역에 대해서

표 2. 시퀀스 별 평균 PSNR결과

Table 2. Average PSNR results of each test sequence.

시퀀스	BiChoi	BMAHa	3DRS	FS-OBMC	WA-MCI	PRO
mobile_tr	31.55	30.54	33.01	31.47	32.25	33.64
football_tr	26.23	24.96	26.90	25.52	26.54	27.30
susie_tr	33.35	31.72	33.83	32.79	33.09	35.19
coastguard	33.82	32.32	34.04	33.71	33.25	33.87
hallmonitor	36.18	35.64	36.33	35.98	35.81	36.36
missa	38.07	38.09	38.30	38.03	38.26	38.34
mitc	32.98	18.82	36.73	30.70	36.61	36.78
salesman	36.12	35.92	36.05	36.01	36.05	36.06
garden	29.69	28.70	29.75	29.41	29.55	29.91

표 3. 블록 당 움직임 추정의 계산량

Table 3. Computational cost for motion estimation per a block.

ME종류	탐색 횟수	1회 탐색당 계산량			
		ADD	ABS	MUL	DIV
BiChoi	33*33/4+25	H*V*2-1	H*V	0	0
BMAHa	33*33	H*V*2+3	H*V+2	4	1
3DRS	24	H*V*2-1	H*V	0	0
FSOBMC	33*33	H*V*2-1	H*V	0	0
WAMCI	33*33	H*V*2-1	H*V	0	0
PRO	mean{1*G+24*L+105*B+106*BG}	H*V*2+9	H*V+4	1	0

*H와 V는 각각 블록의 너비와 높이를 의미함.

표 4. 움직임 추정 및 전체 알고리즘의 계산시간

Table 4. Computational time for motion estimation and total FRC methods

	Type	mobiletr	footballtr	hallmonitor	mitc	garden
BiChoi	ME	4.30	3.84	0.88	9.52	0.89
	Total	5.24	4.64	1.19	11.83	1.19
BMAHa	ME	14.19	12.25	3.03	32.55	3.06
	Total	14.78	12.73	3.16	33.94	3.17
3DRS	ME	0.33	0.25	0.06	0.72	0.06
	Total	1.00	0.83	0.19	2.23	0.20
FSOBMC	ME	13.94	12.00	2.97	32.05	2.98
	Total	14.50	12.49	3.08	33.34	3.09
WAMCI	ME	14.39	12.23	3.03	32.52	3.02
	Total	15.03	12.81	3.16	34.14	3.17
PRO	ME	0.45	0.34	0.09	0.91	0.09
	Total	1.40	1.04	0.27	2.42	0.25

G(38%), L(45%), B(15%), BG(2%)로서 평균적으로 블록당 28.56번의 탐색횟수를 기록하였다. 표 4에는 6가지 실험 영상들에 대해 1프레임 당 움직임 추정 및 전체 프레임 을 변환 알고리즘의 계산시간을 측정된 결과를 표시하였다. Pentium-4 3.00GHz CPU와 2GB RAM을 장착한 컴퓨터에서 측정된 결과로서, 3DRS기법이 가장

빠른 연산시간을 보였다. 제안된 방식은 움직임 추정에서 더 많은 탐색횟수를 처리하며, 블록별 움직임 영역 설정 등의 보조적 계산시간이 요구되어 3DRS기법에 비해 많은 시간을 소모하였지만 기존의 FS기반 기법들에 비해서는 훨씬 적은 시간을 소모하였다.

IV. 결 론

본 논문에서는 적응적 탐색기반 움직임 추정을 사용한 프레임 율 변환 알고리즘을 제안하였다. 제안된 움직임 추정 기법은 블록별 움직임 종류에 따라 적응적으로 회귀탐색, 삼 단계탐색, 그리고 단일예측탐색을 적용함으로써, 계산 량의 증가를 억제하면서도 추정된 움직임 벡터의 정확도를 향상시켰다. 특히 빠르게 움직이는 물체들과 복잡한 이동성 움직임을 갖는 영상들에서 기존의 방법들에 비해 현저한 정확도 향상을 얻을 수 있었다. 제안된 알고리즘들은 다양한 움직임을 갖는 실제 영상 및 합성된 영상 시퀀스들에 적용하여 검증하였으며, 기존의 다른 프레임 율 변환 기법에 비해 뛰어난 주관적 화질 및 PSNR결과를 보임을 확인하였다.

참 고 문 헌

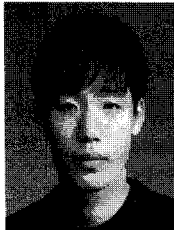
- [1] B. Choi, S. Lee, S. Ko, "New frame rate up-conversion using bi-directional motion estimation," *IEEE Trans. Consum. Electron.*, Vol. 46, No. 3, pp. 603-609, Aug. 2000.
- [2] B. Choi, J. Han, C. Kim, and S. Ko, "Frame Rate Up-Conversion Using Perspective Transform," *IEEE Trans. Consum. Electron.*, Vol. 52, No. 3, pp. 975-982, Aug. 2006.
- [3] K. Hilman, H. Park, Y. Kim, "Using Motion-Compensated Frame-Rate Conversion for the Correction of 3:2 Pulldown Artifacts in Video Sequences," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 10, No. 6, pp. 869-877, Sep. 2000.
- [4] S. Lee, O. Kwon, and R. Park, "Weighted-Adaptive Motion-Compensated Frame Rate Up-Conversion," *IEEE Trans. Consum. Electron.*, Vol. 49, No. 3, pp. 485-492, Aug. 2003.
- [5] B. Jeon, G. Lee, S. Lee, and R. Park "Coarse-to-Fine Frame Interpolation for Frame Rate Up-Conversion Using Pyramid Structure," *IEEE Trans. Consum. Electron.*, Vol. 49, No. 3, pp. 499-508, Aug. 2003.
- [6] T. Ha, S. Lee, and J. Kim, "Motion Compensated Frame Interpolation by new Block-based Motion Estimation Algorithm," *IEEE Trans. Consum. Electron.*, Vol. 50, No. 2, pp. 752-759, May. 2004.
- [7] J. Zhai, K. Yu, J. Li, and S. Li, "A Low Complexity Motion Compensated Frame Interpolation Method," *Proc. IEEE International Symposium on Circuits and Systems*, Vol. 5, pp. 4927-4930, May 2005.
- [8] S. Fujiwara and A. Taguchi, "Motion-Compensated Frame Rate Up-Conversion Based on Block Matching Algorithm with Multi-size Blocks," *Proc. Int. Symp. Intelligent Signal Processing and Communication Systems*, pp. 353-356, Dec. 2005.
- [9] J. Wang, N. Patel, and W. Grosky, "A Fast Block-based Motion Compensation Video Frame Interpolation Approach," *Proc. Asilomar Conf. Signals, Systems, Computers*, Vol. 2, pp. 1740-1743, Nov. 2004.
- [10] P. Csillag and L. Boroczky, "Frame-rate conversion based on acceleration and motion-based segmentation," *Proc. SPIE Vol. 2952*, pp. 438-448, 1996.
- [11] V. Lin and S. G. Chen, "Efficient algorithm for frame rate conversion," *Proc. SPIE Vol. 2308*, pp. 694-702, 1994.
- [12] G. de Haan, P. W.A.C. Biezen, H. Huijgen, and O. A. Ojo, "True-Motion Estimation with 3-D Recursive Search Block Matching," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 3, No. 5, pp. 368-379, Oct. 1993.
- [13] 김영덕, 장준영, 강문기, "관심영역을 고려한 적분 프로젝션기반 움직임 추정을 사용한 순차주사화 알고리즘", 대한전자공학회, 제45권 SP편 제3호, pp20-29, 2008년 5월

저 자 소 개



김 영 덕(학생회원)
 1996년 연세대학교 전자공학과
 학사 졸업.
 1998년 연세대학교 전자공학과
 석사 졸업.
 1998년~삼성전자 연구원.
 2005년~연세대학교 전기전자
 공학과 박사과정.

<주관심분야 : 순차주사화, 프레임 을 변환, 움직임 추정 및 보상, 영상 보간>



장 준 영(학생회원)
 2004년 연세대학교 전기전자
 공학과 학사 졸업
 2006년 연세대학교 전기전자
 공학과 석사 졸업
 2006년~연세대학교 전기전자
 공학과 박사과정

<주관심분야 : 잡음제거, 순차주사화, 움직임 추정 및 보상>



강 문 기(정회원)-교신저자
 1986년 서울대학교 전자공학과
 학사 졸업.
 1988년 서울대학교 전자공학과
 석사 졸업.
 1994년 Northwestern University
 박사 졸업.

1994년~1997년 Assistant Professor.

(University of Minnesota, Duluth)

1997년~1999년 연세대학교 전기전자공학과
 조교수.

1999년~2004년 연세대학교 전기전자공학과
 부교수.

2004년~연세대학교 전기전자공학과 정교수.

<주관심분야 : 영상복원, 초해상도 영상복원, 비선형 필터링, 비디오 분석 및 처리 부화소단위의 움직임 추정>