특집논문-05-10-3-02

트랜스코더의 해상도 변환 모듈과 움직임 추정 모듈의 공동 최적화

한 종 기[a)‡], 곽 상 민[a)], 전 동 산[b)], 김 재 곤[b)]

Joint Optimization of the Motion Estimation Module and the Up/Down Scaler in Transcoders television

Jong-Ki Han[a)‡], Sang-Min Kwak[a)], and Dong-San Jun[b)], Jae-Gon Kim[b)]

요 약

해상도 변환모듈과 움직임 예측모듈은 트랜스코더를 이루는 중요한 모듈이다. 본 논문에서는 트랜스코더 시스템의 이 두 가지 모듈을 공동 최적화하는 기법을 제안한다. 제안하는 기법은 먼저 주어진 움직임 벡터에 대해 해상도 변환모듈을 최적화한 후, 최적화된 해상도 변환모듈에 대해 최적의 움직임 벡터를 결정한다. 기존 해상도 변환 기법들은 한 영상에 대해 변환함수를 최적화하여 사용한다. 본 논문에서는 해상도 변환 최적화를 위하여 적응적 3차 회선 변환기를 제안한다. 제안된 방법은 3차 회선 변환기의 인자값을 각 매크로블록 단위로 영상의 지역적 특성을 고려하여 적응적으로 조절한다. 움직임 예측모듈에서는 기존의 고속 트랜스코더 알고리듬에서 많이 연구된 움직임 벡터의 재사용 기법을 사용하였다. 입력 영상의 움직임 벡터를 재사용함으로써 연산량을 줄일 수 있고 이를 기본 움직임벡터로 사용해 작은 영역에서 재탐색해 움직임벡터를 결정할 경우 전역탐색 기법과 거의 동일한 화질의 영상을 얻을 수 있다. 해상도 변환모듈과 움직임 예측모듈의 공동 최적화를 통해서 트랜스코딩된 영상의 화질 열화를 최소화할 수 있는 알고리듬을 제안한다. 실험 결과 본 논문에서 제안하는 공동 최적화 기법이 기존에 연구되었던 다른 기법에 비해 화질의 열화가 적은 것을 알 수 있었고, 이를 통해 다른 기법과 비교해 해상도 변환으로 인한 정보의 손실이 가장 적음을 알 수 있다.

Abstract

A joint design scheme is proposed to optimize the up/down scaler and the motion vector estimation module in the transcoder system. The proposed scheme first optimizes the resolution scaler for a fixed motion vector, and then a new motion vector is estimated for the fixed scaler. These two steps are iteratively repeated until they reach a local optimum solution. In the optimization of the scaler, we derive an adaptive version of a cubic convolution interpolator to enlarge or reduce digital images by arbitrary scaling factors. The adaptation is performed at each macroblock of an image. In order to estimate the optimal motion vector, a temporary motion vector is composed from the given motion vectors. Then the motion vector is refined over a narrow search range. It is well-known that this refinement scheme provides the comparable performance compared to the full search method. Simulation results show that a jointly optimized system based on the proposed algorithms outperforms the conventional systems. We can also see that the algorithms exhibit significant improvement in the minimization of information loss compared with other techniques.

Keywords : Transcoder, interpolation, cubic convolution, motion vector estimation

a) 세종대학교 정보통신 공학과
Dept. of Information & Communications Engineering, Sejong University.

b) ETRI 방송 미디어 연구 그룹
Electronics and Telecommunications Research Institute Digital Broadcasting Research Division, Broadcasting Media Research Group

Ⅰ. 서 론

일반적으로 비디오 데이터는 서버측에서 일차로 부호화되어 저장된 후 수신측의 요청이 있을때 마다 전송된다. 그러나 이러한 데이터는 채널의 특성을 반영하지 않은 채 부호화된 것이므로 네트워크 자원을 효율적으로 이용할 수 없다. 그러므로 일차적으로 부호화된 비디오 데이터는 채널의 환경과 특성을 고려하여 재부호화되어야 한다. 트랜스코더는 부호화된 비디오 데이터를 복호한 후 채널환경이나 특성에 맞추어 재부호화함으로써 다양한 네트워크에 대해 적응적으로 전송될 수 있도록 한다[1][2]. 따라서 하나의 비디오 데이터를 다양한 채널을 통해 전송해야할 경우 트랜스코더는 필수적인 요소이다.

트랜스코더는 크게 DCT(Discrete Cosine Transform) 도메인 트랜스코더와 픽셀 도메인 트랜스코더로 나눌 수 있다. 픽셀 도메인 트랜스코더는 픽셀 단위까지 복호화하고, 목적한 화면 해상도와 비트량에 맞추어 재 부호화하는 것이다[1][3][4][5]. 이러한 트랜스코더는 화면 해상도를 축소시킬 경우에 깨끗한 화질을 얻을 수 있지만 해상도의 변환이 없는 경우 DCT 도메인 트랜스코더에 비해 연산량이 많은 단점이 있다. Shanableh와 Ghanbari[3]는 픽셀 도메인 트랜스코더에서 시간적/공간적 해상도 변환 시 연산량을 줄일 수 있는 기법에 대해 연구했다. DCT 도메인 트랜스코더는 입력된 비트스트림을 모두 복호하는 것이 아니라 DCT 도메인까지만 복호화해서 재 부호화하는 방식이다[6][7][8][9][10]. 이는 해상도 변환을 하지 않을 경우 픽셀 도메인 트랜스코더에 비해 빠른 연산이 가능하지만, 해상도 변환시에는 픽셀 도메인 트랜스코더에 비해 화질의 열화가 심하다. Yim과 Isnardi[6]은 DCT 도메인 상에서 화면 해상도를 1/2로 축소시키는 효율적인 방안을 연구하였다. Merhav[8]는 DCT 도메인 상에서 화면 해상도 변환 모듈을 선형 시스템으로 근사화하여 트랜스코더에서 사용하였다.

해상도 변환과 움직임 추정 모듈은 트랜스코더에서 매우 중요한 연구분야이다. 움직임 추정은 트랜스코딩 시스템에서 연산량이 가장 많은 모듈이고 화질에 직접적인 영향을 미치는 부분이다. 따라서 효율적인 움직임 탐색에 관한 기법은 많은 연구가 있었다[3][4][5]. 해상도 변환 모듈은 해상도를 축소함으로써 발생되는 정보손실을 최소화함으로써 트랜스코딩 시스템의 성능을 크게 개선시킬 수 있다. 하지만 이에 관한 연구는 1/2 내지 1/4와 같은 정수 비율 변환을 중심으로 한 연구가 대부분이다[6][8]. 본 논문에서는 정수 비율 변환이 아닌, 임의의 비율에 적합한 트랜스코딩 기법을 제안하고자 한다. 제안하는 트랜스코더 시스템의 해상도 변환기는 3차 회선 보간법(Cubic convolution interpolation)을 이용하고, 변환기의 인자값을 각 매크로블록마다 최적화시킴으로써 영상정보의 손실을 최소화시킬 수 있도록 하였다. 또한 일반적인 트랜스코더에서 해상도 변환은 대부분 개념적인 부분뿐만 아니라 실제 구현에 있어서도 모두 움직임 예측 모듈과 별개의 문제로 다루어져 왔다. 하지만 해상도 변환기의 인자값에 따라 최적의 움직임 벡터는 변하고, 움직임 벡터에 따라서 해상도 변환기의 최적의 인자값이 변하기 때문에 두 모듈은 종속적이다. 본 논문에서는 이와 같은 문제를 해결하기 위해 움직임 예측모듈과 해상도 변환모듈을 결합하여 최적화시키는 기법을 소개한다. 그동안 각각 별개의 문제로 생각되어 왔던 두 연구의 결합을 통하여 트랜스코딩으로 인해 발생되는 정보의 손실을 최소화하고자 한다.

본 논문은 다음과 같이 나누어져 있다. 2장에서는 일반적인 트랜스코딩 시스템과 문제점에 대해서 서술한다. 3장에서는 움직임 벡터가 고정된 경우 해상도 변환기의 최적화에 대해서 다룬다. 4장에서는 해상도 변환기의 인자값이 고정된 경우 최적의 움직임 예측에 대해서, 5장에서는 움직임 예측 모듈과 해상도 변환 모듈의 공동 최적화에 대해 다룬다. 본 논문에서 제안하는 알고리듬에 대한 실험 결과를 6장에서 나타내고, 7장에서는 본 논문에 대한 결론에 대해 서술한다.

Ⅱ. 트랜스코더 시스템

한번 부호화된 비디오스트림은 미디어에 저장되거나 대역폭이 제한된 채널을 통해 전송될 경우 트랜스코더를 통하여 비트량을 감소시킬 필요가 있다[4][5][11][12]. 그림 1

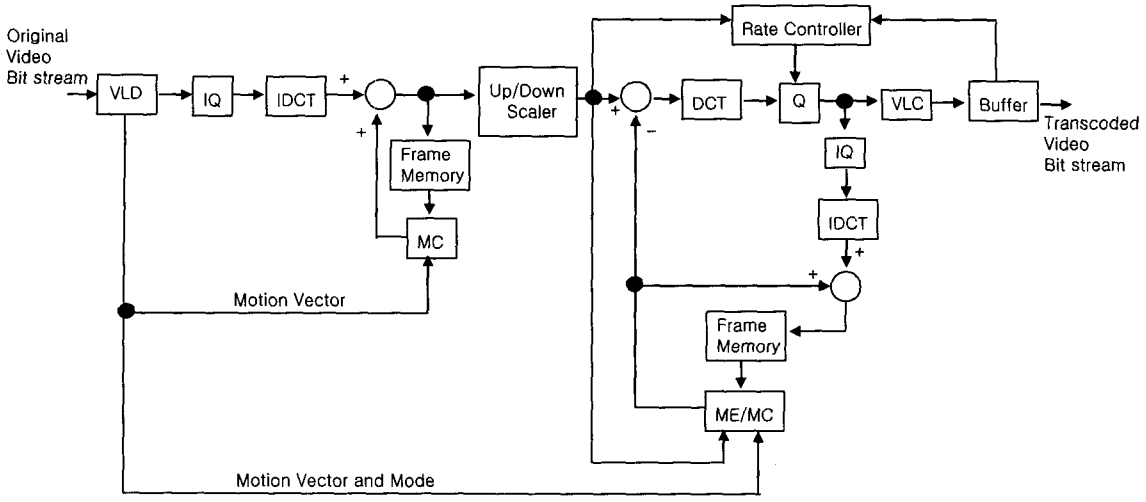


그림 1. 해상도 변환 트랜스코더 시스템
Fig. 1. The transcoder system with an up/down scaler

은 트랜스코더의 구조를 나타낸다. 트랜스코더는 양자화 간격을 조절하거나, 화면 해상도를 변환시켜 발생 비트량을 조절한다. 그림 1을 보면, 입력된 비디오스트림은 VLD (Variable Length Decode)를 통과해 역 양자화되어 DCT도메인까지 복호된다. 이러한 데이터는 역 DCT를 실행시키고 움직임을 보상해 픽셀 도메인의 데이터로 출력된다. 픽셀 도메인상에서는 화면 해상도 변환이 용이하므로 대부분의 화면 해상도 변환기는 이곳에 위치한다. 변환된 비디오 신호는 일반적인 비디오 부호기와 마찬가지로 움직임 예측을 하고 DCT변환을 거쳐 양자화한다. 양자화 파라미터는 채널의 상황에 따라 적응적으로 조절되며, 양자화된 데이터는 VLC(Variable Length Code)를 통과해 새로운 비트스트림을 출력한다. 트랜스코더는 이러한 과정을 거쳐 네트워크 환경에 따라 적응적으로 부호화가 가능하다. 여기서 연산량을 줄이기 위해, 대부분의 트랜스코더는 부호화 작업 중 움직임 예측모듈에서 입력 비트스트림의 움직임 데이터를 재사용한다[5]. 본 논문에서는 움직임 벡터를 재사용하기 위해 화면의 해상도 변환비율과 동일하게 움직임 벡터도 같은 비율의 크기로 변환시키고, 이와 같이 크기가 변환된 재사용 움직임 벡터를 중심으로 ± 2 영역에서 재탐색하였다.

1. 해상도 변환

디지털 영상의 해상도 변환은 트랜스코더 뿐만 아니라 여러 연구 분야에서 중요하게 다뤄지고 있다. 현재까지 많은 연구가 있었지만 비디오에 관련된 연구는 정수 비율 변환에 대해서 다룬 것이 대부분이다. 가로 세로방향의 정수 비율 해상도 변환[16]과, 영상을 세로방향으로 2배를 확대해 비월주사 영상을 순차주사 영상으로 변환(IPC`s: Interlaced to Progressive Conversion)하는 기법[13][14][15]등이 그것이다. 이러한 기법은 영상의 해상도 변환 비율이 한정적이므로 해상도 선택의 폭이 좁다. 이러한 문제를 해결하기 위해 본 논문은 다양한 임의 비율 해상도 변환을 위해 연산량이 비교적 작으며, 해상도 변환 시 정보손실이 적은 3차 회선 변환기[19][24]을 이용한다. 여기서 정보손실의 양은 원영상과 해상도 변환시킨 후 초기 해상도로 역변환시킨 재건된 영상과의 차이를 말한다. 3차 회선 변환기의 기존 연구들은 해상도 변환 인자값 a를 조절함으로써 최적화시킨다. 그러나 영상의 전역에 대해서 같은 해상도 변환 인자값을 적용하는 것은 영상의 지역적 특성을 반영할 수 없다. 따라서 본 논문은 영상의 매크로블록 단위로 인자값을 최적화시키는 기법을 제안한다. 그림 2에 나타나있는 것처럼 디지털 영상 $f(n,k)$는 해상도 변환기를 통해 $\tilde{f}(l,m)$으로 축소

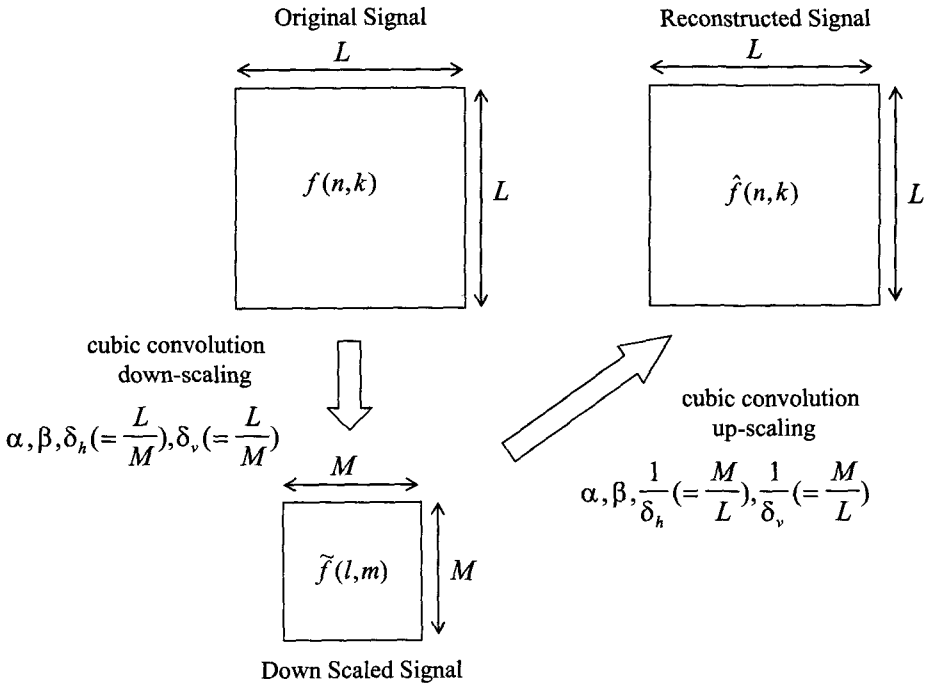


그림 2. 3차 회선 해상도 변환기를 통해 축소된 영상의 재건
Fig. 2. The reconstruction of the image down-scaled with a cubic convolution scaler

되고, 이는 다시 해상도 변환기를 통해 $\tilde{f}(n,k)$으로 확대된다. 원영상은 $f(n,k)$, 축소영상은 $\tilde{f}(l,m)$, 재건영상은 $\hat{f}(n,k)$으로 표시한다. $\tilde{f}(l,m)$과 $\tilde{f}_h(n,m)$을 수식으로 표현하면 다음과 같다.

$$\begin{aligned}\hat{f}(l,m) &= \tilde{f}_h(n^*-1,m)(\beta t^3-2\beta t^2+\beta t)\\ &+\tilde{f}_h(n^*,m)((\beta-2)t^3-(3+\beta)t^2+1)\\ &+\tilde{f}_h(n^*-1,m)(-(\beta+2)t^3+(2\beta+3)t^2-\beta t)\\ &+\tilde{f}_h(n^*+2,m)(-\beta t^3+\beta t^2)\end{aligned} \quad (1)$$

$$\begin{aligned}\hat{f}_h(n,m) &= f(n,k^*-1)(\alpha s^3-2\alpha s^2+\alpha s)\\ &+f(n,k^*)((\alpha+2)s^3-(3+\alpha)s^2+1)\\ &+f(n,k^*+1)(-(\alpha+2)s^3+(2\alpha+3)s^2-\alpha s)\\ &+f(n,k^*+2)(-\alpha s^3+\alpha s^2)\end{aligned} \quad (2)$$

여기서 $k^*=\lfloor m\times\delta_h\rfloor$, $s=m\times\delta_h-k^*$, $n^*=\lfloor l\times\delta_v\rfloor$, $t=l\times\delta_v-n^*$이다. $\lfloor A\rfloor$는 A보다 작은 최대의 정수를 말하며 δ_h와 δ_v는 각각 가로 세로방향의 변환비율이다. 재건영상은

$$\begin{aligned}\hat{f}(n,k) &= \hat{f}_v(n,m^*-1)(\alpha s^3-2\alpha s^2+\alpha s)\\ &+\hat{f}_v(n,m^*)((\alpha+2)s^3-(3+\alpha)s^2+1)\\ &+\hat{f}_v(n,m^*+1)(-(\alpha+2)s^3+(2\alpha+3)s^2-\alpha s)\\ &+\hat{f}_v(n,m^*+2)(-\alpha s^3+\alpha s^2)\end{aligned} \quad (3)$$

이며 $\hat{f}_v(n,m)$은

$$\begin{aligned}\hat{f}_v(n,m) &= \hat{f}(l^*-1,m)(\beta t^3-2\beta t^2+\beta t)\\ &+\hat{f}(l^*,m)((\beta+2)t^3-(3+\beta)t^2+1)\\ &+\hat{f}(l^*+1,m)(-(\beta+2)t^3+(2\beta+3)t^2-\beta t)\\ &+\hat{f}(l^*+2,m)(-\beta t^3+\beta t^2)\end{aligned} \quad (4)$$

와 같이 나타낼 수 있다. 여기서 $m^* = \lfloor k \times \frac{1}{\delta_h} \rfloor$, $s = k \times \frac{1}{\delta_h} - m^*$, $l^* = \lfloor n \times \frac{1}{\delta_v} \rfloor$, $t = n \times \frac{1}{\delta_v} - l^*$이다. 위 식에 대한 정리는 부록에 첨부되어 있다. 식(1), (2), (3), (4)에서 α와 β는 각각 가로와 세로방향의 해상도변환을 위해 학습될 인자 값들이다. 해상도 변환으로 인해 손실된 정보의 양은 축소 후 확대한 결과를 통해 아래와 같이 정의할 수 있다.

$$D_{Recon}(\alpha, \beta) = \sum_{n=0}^{L-1} \sum_{k=0}^{L-1} \{ f(n,k) - \hat{f}(n,k) \}^2 \quad (5)$$

$L \times L$은 원영상의 가로×세로 크기이다. 식(1), (2), (3), (4)의 α, β 는 학습을 통해 더욱 좋은 화질의 영상을 얻을 수 있다. 식(1)과 (2)의 해상도변환기를 최적화시키기 위해 Rifman[17]과 Bernstein[18]은 $sinc(x)$ 함수와 가장 비슷한 형을 기본 핵으로 사용하기 위해 $\alpha = -1$, $\beta = -1$으로 두었다. Keys[19]는 멱급수를 사용해, 해상도 변환기의 보간함수가 $\alpha = -\frac{1}{2}$, $\beta = -\frac{1}{2}$에 수렴할수록 원영상과 가까워지는 것을 보였다. Park와 Schowengerdt[20]는 영상의 주파수를 고려하여 α와 β를 최적화시켰다. 그러나 주파수는 영상 데이터의 지역적인 특성이 아닌, 전역적인 정보를 제공한다. 따라서 이를 극복하기 위해 우리는 α와 β를 매크로블록 단위로 최적화시키는 적응 보간 알고리듬을 제안하고자 한다. 본 논문에서 제안하는 트랜스코더 시스템은 매크로블록의 지역적 특성을 이용하여 인자값 α와 β를 최적화시킨 후, 이를 이용하여 영상을 축소함으로써 영상 데이터의 손실을 최소화한다.

2. 움직임 추정 모듈

해상도 변환의 결과 영상은 트랜스코더의 입력영상과 다른 새로운 영상이므로 움직임을 다시 예측하여야 한다. 그림 3에서 MB는 움직임을 예측할 현재 매크로블록이다. 여기서 매크로블록의 크기는 16×16이다. $f(n,k)$는 현재 MB에 해당되는 축소하기 전 원영상의 큰 블록이다. 각 매크로블록

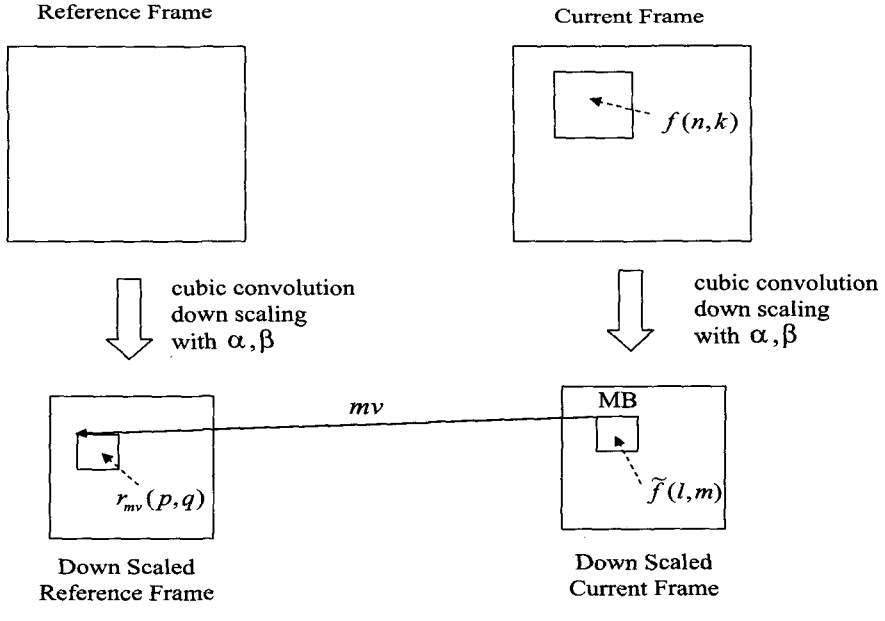


그림 3. 해상도 변환 트랜스코더의 움직임 예측 과정
Fig. 3. Motion estimation process in the transcoder using the down scaler

$$D_{Motion}(mv,\alpha,\beta) = \sum_{l=0}^{15}\sum_{m=0}^{15}\left(\tilde{f}(l,m)-r_{mv}(l,m)\right)^2 \quad (6)$$

$$= \sum_{l=0}^{15}\sum_{m=0}^{15}\left(\tilde{f}^2(l,m)-2r_{mv}(l,m)\tilde{f}(l,m)+r_{mv}^2(l,m)\right) \quad (7)$$

단위로 α, β를 훈련시키고, 이를 이용해 원영상을 3차 회선 변환기[19][24]로 축소시켰다. 축소된 영상에서 $\tilde{f}(l,m)$은 매크로블록의 픽셀값이다($0\le l\le 15$, $0\le m\le 15$). $r_{mv}(p,q)$는 현재영상의 매크로블록인 $\tilde{f}(l,m)$로부터 움직임 벡터 만큼 이동한 축소된 참조영상의 픽셀 데이터이다 ($0\le p\le 15$, $0\le q\le 15$).

움직임 예측 모듈의 최적화 정도는 현재 매크로블록인 $\tilde{f}(l,m)$과 참조 매크로블록인 $r_{mv}(p,q)$ 사이의 SSD(sum of squared difference)로 나타낼 수 있다.

식 (6)과 (7)에서 알 수 있듯이 D_{Motion}은 mv, α, β에 종속적이다.

3. 비용함수 최적화

트랜스코더 시스템에서 움직임 예측과 화면 해상도 변환을 통해 발생되는 정보손실의 양을 최소화 하기 위하여, 본 논문에서는 전체 왜곡값을 다음과 같이 정의한다.

$$D_T(mv,\alpha,\beta) = D_{Motion}(mv,\alpha,\beta) + \lambda \cdot D_{Recon}(\alpha,\beta) \quad (8)$$

식 (8)의 전체 왜곡값 D_T는 ME 과정에서 발생하는 왜곡 D_{Motion}와 해상도 변환 과정에서 발생하는 왜곡 D_{Recon}로 이루어지는 것을 의미한다. 즉, D_T를 최소화하기 위해서는, D_{Motion}과 D_{Recon}에 대한 mv, α, β의 최적화가 필요하다. α, β 값의 변화는 D_{Recon}뿐만 아니라 D_{Motion}에도 영향을 미치기 때문에 D_{Motion}과 D_{Recon}의 최소값은 동시에 찾아져야 한다. 예를 들어, 해상도 변환 모듈이 영상 정보의 특징을 효율적으로 고려하여 정보 손실이 거의 없게 해상도 축소를 수행했으나 움직임 추정모듈의 성능저하로 인해 큰 왜곡이 발생한다면, 트랜스코딩된 최종 영상의 화질에는 큰 열화가 발생한다. 다른 한편으로 해상도 변환 모듈이 잘못 설계되어 해상도 변환 과정에서 큰 열화가 발생한다면, 움직임 추정(ME)과정에서 아무리 정확한 움직임을 추정한다고 해도, 최종 트랜스코딩된 영상에는 큰 열화 효과를 남기게 된다. 이와 같이 트랜스코더의 구성 요소인 해상도 변환 모듈과 움직임 추정 모듈의 성능이 동시에 최적화되어야 트랜스코더 전체 성능을 향상 시킬 수 있다. 식 (8)에서 λ는 움직임 추정 모듈의 성능과 해상도 변환 모듈 성능 간의 상대적인 중요도를 표시하는 파라미터이다.

본 논문에서는 일차로 초기 움직임 벡터 mv에 대해서 3차 회선 변환기의 인자값 α, β를 조정하여 D_T를 최소화 시킨다. 그 후, 조정된 α, β의 값에 대하여 D_T를 최소화시키는 움직임 벡터 mv를 다시 찾는다. 이 과정의 반복을 통해 mv, α, β를 최적화시킴으로써, 두 모듈의 공동 최적화 가능하다.

Ⅲ. 3차 회선 해상도 변환기의 최적화

이 장에서는 주어진 움직임 벡터에 대해 3차 회선 변환기의 최적화를 통해 D_T값을 최소화시킨다. 움직임 벡터가 주어진 경우, D_T는 α, β만의 함수가 된다. 그러므로 3차 회선 변환기의 최적화를 통해 D_T를 최소화시킬 수 있다.

재건영상의 블록 $\hat{f}(n,k)$와 원영상의 블록 $f(n,k)$의 차를 최소화시키기 위해, 3차 회선 변환기를 반복적으로 수행시켜 최적화된 α^*와 β^*를 구한다. 최소의 D_T는 아래와 같이 표현할 수 있다.

$$\min_{\forall\alpha,\beta} D_T(\alpha,\beta,mv) = \min_{\forall\alpha,\beta}\{D_{Motion}+\lambda\cdot D_{Recon}\} \quad (9)$$

최대경사법[22][23]을 이용하여 최적화된 변환기의 인자값, α^*와 β^*는 다음과 같이 나타낼 수 있다.

$$(\alpha^*, \beta^*) = (\alpha + \Delta\alpha, \beta + \Delta\beta) \tag{10}$$

여기서

$$\Delta\alpha = -\eta \frac{\partial D_T}{\partial \alpha}(\alpha, \beta), \tag{11}$$

$$\Delta\beta = -\eta \frac{\partial D_T}{\partial \beta}(\alpha, \beta), \tag{12}$$

이다. 식(11)과 (12)에서 알 수 있듯이, 기울기 $-\frac{\partial D_T}{\partial \alpha}(\alpha, \beta)$와 $-\frac{\partial D_T}{\partial \beta}(\alpha, \beta)$를 따라 α와 β의 값은 각각 증가한다. (α, β)에서 시작하여 음의 기울기 $(-\frac{\partial D_T}{\partial \alpha}, -\frac{\partial D_T}{\partial \beta})$ 방향으로 D_T를 최소화시키는 (α^*, β^*)를 찾는다. η는 기울기 값 $-\frac{\partial D_T}{\partial \alpha}(\alpha, \beta)$와 $-\frac{\partial D_T}{\partial \beta}(\alpha, \beta)$에 곱해져서 α와 β의 증가량을 구하는 상수이다. 이 값은 보통 양의 작은 값$(0 < \eta < 1)$을 사용하는데, 이 η을 너무 작은 값을 사용하면 최적화 속도가 늦어지고, 너무 큰 값을 사용하면 최적화 과정이 converge하지 않게 된다. 여기서 $r_{mv}(l, m)$은 3차 회선 변환기의 인자값 α, β와 무관하기 때문에 식 (8)은 아래와 같은 편미분 방정식으로 표현할 수 있다.

$$\frac{\partial D_T}{\partial \alpha} = \frac{\partial \{D_{Motion}\}}{\partial \alpha} + \lambda \cdot \frac{\partial \{D_{Recon}\}}{\partial \alpha} \tag{13}$$

$$\frac{\partial D_T}{\partial \beta} = \frac{\partial \{D_{Motion}\}}{\partial \beta} + \lambda \cdot \frac{\partial \{D_{Recon}\}}{\partial \beta} \tag{14}$$

위 식에서 $M \times M$은 축소된 이미지 중 MB의 크기이고, $L \times L$은 원영상의 MB에 해당하는 블록의 크기이다. 식 (15)(16)(17)(18)의 $\frac{\partial \tilde{f}(l, m)}{\partial \alpha}$, $\frac{\partial \hat{f}(n, k)}{\partial \alpha}$, $\frac{\partial \tilde{f}(l, m)}{\partial \beta}$, $\frac{\partial \hat{f}(n, k)}{\partial \beta}$는 부록에 정리되어 있다. α, β값 모두 초기값은 $-\frac{1}{2}$이고 α, β값은 식 (10)과 같이 매회 갱신된다. 갱신된 인자값 α^*, β^*는 다음 시행의 해상도 변환에 사용된다. 이 과정은 각 매크로블록 별로 실행되고, 이러한 최적화 과정은 표 1에 정리되어 있다. 시작값 $\alpha^{(iter)}$은 본 알고리듬을 거쳐 음의 방향으로 $-\frac{\partial D_T}{\partial \alpha}(\alpha^{(iter)})$만큼 작은 $\alpha^{(iter+1)}$이 된다. 이 과정이 반복될수록 D_T의 값은 작아지게 되고, D_T가 충분히 작은 값을 가질 때까지 위 과정이

표 1. 3차 회선 해상도 변환기의 최적화(각 매크로블록 별로 α, β의 최적화 실행)
Table I. Optimization of the cubic convolution scaler

과정 1.	$\alpha = -1/2$, $\beta = -1/2$, $iter = 1$, $D_T^{(iter-1)} = \infty$으로 초기화
과정 2.	식(1)을 이용해 축소영상 블록 $\tilde{f}(n, k)$ 생성
과정 3.	식(3)을 이용해 재건영상 블록 $\hat{f}(l, m)$ 생성
과정 4.	식(11)과 식(12)를 이용해 $\Delta\alpha$, $\Delta\beta$계산
과정 5.	$(\alpha^*, \beta^*) = (\alpha + \Delta\alpha, \beta + \Delta\beta)$
과정 6.	$D_T^{(iter)}$ 계산
과정 7.	만약 $\{D_T^{(iter-1)} - D_T^{(iter)}\} / D_T^{(iter-1)} > \varepsilon$이면, $\alpha = \alpha^*$, $\beta = \beta^*$, $iter = iter + 1$, 과정 2로 이동
과정 8.	중지

$$\frac{\partial \{D_{motion}\}}{\partial \alpha} = \sum_{l=0}^{M-1} \sum_{m=0}^{M-1} \left\{ 2\tilde{f}(l, m) \frac{\partial \tilde{f}(l, m)}{\partial \alpha} - 2r_{mv}(l, m) \frac{\partial \tilde{f}(l, m)}{\partial \alpha} \right\} \tag{15}$$

$$\frac{\partial \{D_{recon}\}}{\partial \alpha} = \sum_{n=0}^{L-1} \sum_{k=0}^{L-1} 2(f(n, k) - \hat{f}(n, k)) \left(-\frac{\partial \hat{f}(n, k)}{\partial \alpha} \right) \tag{16}$$

$$\frac{\partial \{D_{motion}\}}{\partial \beta} = \sum_{l=0}^{M-1} \sum_{m=0}^{M-1} \left\{ 2\tilde{f}(l, m) \frac{\partial \tilde{f}(l, m)}{\partial \beta} - 2r_{mv}(l, m) \frac{\partial \tilde{f}(l, m)}{\partial \beta} \right\} \tag{17}$$

$$\frac{\partial \{D_{recon}\}}{\partial \beta} = \sum_{n=0}^{L-1} \sum_{k=0}^{L-1} 2(f(n, k) - \hat{f}(n, k)) \left(-\frac{\partial \hat{f}(n, k)}{\partial \beta} \right) \tag{18}$$

반복된다. 여기서 각 인자값은 단지 D_T는 의 기울기에 따라 갱신되므로, 이 과정을 통해 얻을 수 있는 값은 지역적으로 최소값을 갖는 값이다[23].

Ⅳ. 트랜스코더의 움직임 예측

이번 장은 주어진 해상도 변환기의 인자값 α^*, β^*에 대해서 D_T의 최적화시키는 움직임 예측에 대해 다룬다.

$$\min_{mv} D_T = \min_{mv} D_T \{D_{Motion} + \lambda \cdot D_{Recon}\} \quad (19)$$

여기서 D_{Recon}은 움직임 벡터 mv와 무관하므로 식 (19)는 아래와 같이 나타낼 수 있다.

식 (21)에서 $\tilde{f}(l,m)$은 움직임 벡터 mv와 관련이 없으므로 움직임 벡터 추정에 관련된 식은 아래와 같이 간략화 될 수 있다.

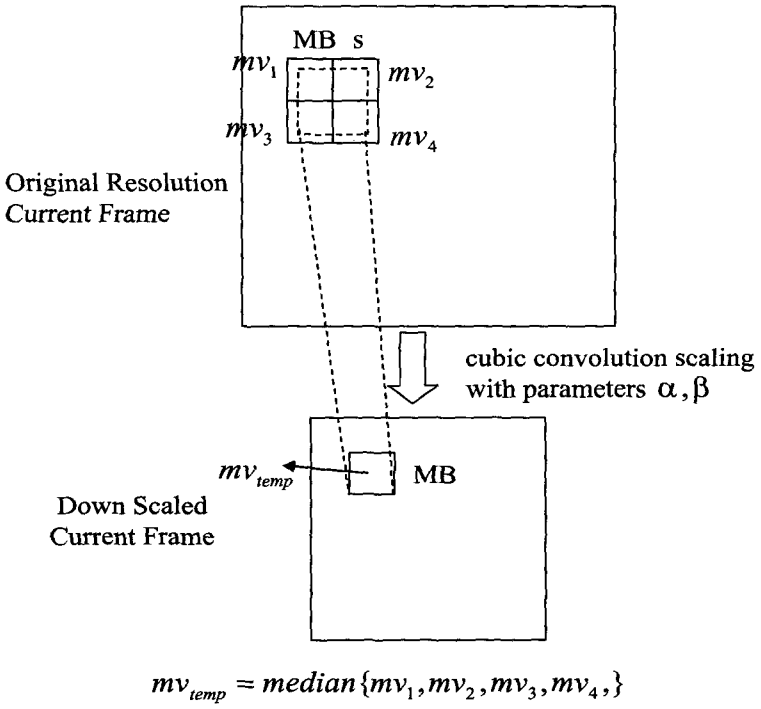
식(22)의 mv는 현재 매크로블록 $\tilde{f}(l,m)$와 참조 매크로블록 $r_{mv}(l,m)$간의 움직임 벡터이다.

움직임 추정은 연산량이 많은 부분이므로 이전에도 많은 연구가 이루어진 부분이다[1][2][3][4][5]. 여러 기존 알고리듬 중, 실질적으로 가장 효율성이 높은 것은 그림 4에 나온 움

$$\min_{mv} D_T = \min_{mv} D_{Motioin} \quad (20)$$

$$= \min_{mv} \{\tilde{f}^2(l,m) - 2r_{mv}(l,m)\tilde{f}(l,m) + r^2_{mv}(l,m)\} \quad (21)$$

$$\min_{mv} D_T = \min_{mv} \{-2r_{mv}(l,m)\tilde{f}(l,m) + r^2_{mv}(l,m)\} \quad (22)$$



$$mv_{temp} = median\{mv_1, mv_2, mv_3, mv_4,\}$$

그림 4. 트랜스코더의 움직임 예측모듈의 움직임 벡터 재사용

Fig. 4. An example to reuse the given motion vectors in motion estimation module of the transcoder

직임 벡터의 재사용이다. 그림 4와 같이 축소된 영상의 매크로블록은 축소되기 전 영상의 더욱 넓은 부분을 포함하고 있다. 따라서 현재 축소된 영상의 매크로블록에 해당되는 축소되기 이전 영상의 매크로블록들의 움직임 벡터들을 아래와 같이 이용한다.

$$mv_{temp} = median\{mv_1, mv_2, mv_3, mv_4\} \quad (23)$$

단 mv_{temp}는 최적화된 움직임 벡터가 아니라 최적의 움직임 벡터를 찾기 위한 기본 움직임 벡터이다. 최적의 움직임 벡터를 찾기 위해서는, 기본 움직임벡터 mv_{temp}를 중심으로 매우 작은 탐색영역만을 추정하는 과정이 추가되어야 한다. 이 같은 방식으로 계산량은 매우 크게 줄이면서, 전역탐색 기법과 거의 동등한 화질을 얻을 수 있다.

V. 해상도 변환기와 움직임 벡터의 공동 최적화

3장과 4장에서는 움직임 벡터와 해상도 변환기 각각의 최적화에 대해 정리했다. 이 장에서는 두 모듈의 공동 최적화를 통해 식(8)에서 정리된 전체 왜곡값 D_T를 최소화시킨다. 식(8)에서 D_{Motion}은 움직임 벡터 mv와 3차 회선 변환기의 인자값 α, β의 함수이고, D_{Recon}은 단지 α, β의 함수이다. 따라서 해상도 변환기의 최적화에서는 D_T가 최소값을 갖는 mv로 고정시킨 후 $\{D_{Motion} + \lambda D_{Recon}\}$가 최소값을 갖는 인자값 α^*, β^*을 구한다. 이와 반대로 움직임 벡터의 최적화는 D_T가 최소값을 갖는 α^*, β^*로 고정시키고 D_{Motion}이 최소가 되는 움직임 벡터를 구한다. 이 두 알고리듬의 교대 반복으로 이루어진 공동 최적화 알고리듬은 표2에 요약되어 있다. 표 2의 과정 2에서 새로운 인자값 (α^*, β^*)을 구하기 위해 움직임 벡터는 고정시킨다. 과정 3에서는 과정 2로부터 얻어진 새로운 해상도 변환기를 사용해 움직임을 예측한다. 움직임 벡터 mv가 고정된 동안 과정 2에 따라 인자값 α와 β를 갱신하더라도 전체 왜곡값 D_T는 증가하지 않는다. 과정 3에서도 α, β가 고정된 동안 움직임 벡터 mv를 갱신하더라도 전체 왜곡값 D_T는 증가하지 않는다. 이처럼 공동 최적화 알고리듬을 반복하는 도중 증가함수는 포함되어 있지 않으므로 해상도 변환기와 움직임 예측 모듈은 공동 최적화를 통해 수렴할 수 있다. 표2의 공동 최적화 알고리듬의 성능은 mv, α, β의 초기 조건에 영향을 받으며, mv와 α, β값이 같은 시각에 최적화 되지 않고 교대로 최적화되는 알고리듬으로써 트랜스코더 시스템을 local optimum 상태로 수렴시킨다.

표 2. 움직임 예측기와 해상도 변환기의 공동 최적화 알고리듬 (전체 최적화 알고리듬)
Table 2. Joint optimization algorithm of the scaler and the ME for each MB

과정 1.	$iter=1$, $\alpha=-1/2$, $\beta=-1/2$, $D_T^{(0)}=\infty$로 초기화
과정 2.	표 1의 기법을 이용해 해상도 변환기에 대한 최적화
과정 3.	식(22)의 움직임 벡터 mv계산
과정 4.	$D_T^{(iter)}$계산
과정 5.	만약 $D_T^{(iter-1)} - D_T^{(iter)} / D_T(iter-1) > \varepsilon$ 이면 $iter=iter+1$, 과정 2로 이동
과정 6.	중지

VI. 실험 및 결과

본 논문에서 제안한 알고리듬의 실험을 위해 "football" 영상과 "mobile and calender"영상을 사용하였다. 비디오 코덱은 MPEG-2 MP@ML으로 코딩하고 GOP 구조는 $N=12$, $M=3$으로 하였다. 입력 비디오시퀀스는 $720\times480i$ 해상도의 영상을 비트율 $R_o(=8Mbps)$로 코딩된 것이다. 트랜스코더를 통해 해상도 변환된 영상은 해상도 $432\times288i$, 비트율 $R_T(=3Mbps)$로 코딩하였다. 축소된 영상의 화질에 대한 평가 기준으로는 해상도 $720\times480i$

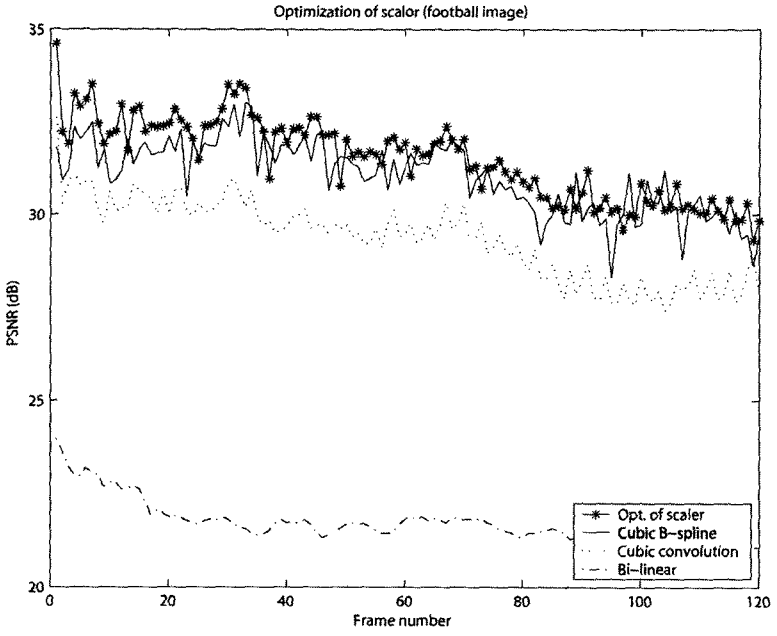


그림 5. 해상도 변환 트랜스코더의 PSNR (본 논문에서 제안하는 해상도 변환기법과 양선형, 일반적인 3차 회선, 3차 B-spline 보간법을 사용한 해상도 변환기의 PSNR 비교, 움직임 예측은 일반적인 기법 사용)

Fig. 5. The PSNR comparison of the images resulted from the transcoding systems using the up/down scaler. The test image is "football." The scalers are the optimized scaler, the cubic B-spline, the conventional cubic convolution scaler, and bi-linear interpolator. The motion vectors are estimated by the conventional ME scheme.

의 원영상과 축소시킨 후 다시 확대시킨 재건된 영상간의 PSNR을 사용하였다.

1. 해상도 변환기 최적화 결과

본 논문에서 제안한 알고리듬 중 해상도 변환기와 기존에 연구된 해상도 변환기를 비교하였다. 기존의 해상도 변환 알고리듬은 양선형[24]와 B-spline[25], 3차 회선[19] 변환기을 사용하였다. 그림 5는 원영상과 재건 영상간의 PSNR을 각 알고리듬별로 표시한 것이다. 기존의 해상도 변환 알고리듬은 입력된 데이터에 대해 고려하지 않은 반면, 본 논문에서 제안한 해상도 변환 알고리듬은 영상의 지역적 특성을 반영하고 있으므로 대체적으로 좋은 화질을 얻을 수 있었다. 이는 기존 알고리듬과 비교했을 때 본 논문에서 제안한 해상도 변환을 통한 정보의 손실이 가장 적음을 말한다.

2. 움직임 예측 결과

이번 절은 4장에서 다룬 움직임 예측 기법에 관한 실험 결과에 대해 다룬다. 그림 6과 그림 7을 통해 움직임 벡터의 재사용이 얼마나 효율적인 알고리듬인지에 대해 알 수 있다. 그림 6을 보면 움직임 벡터를 그대로 다시 사용한 경우에는 전역탐색에 비해 대략 1~4dB 의 PSNR이 떨어지지만, 재사용된 움직임 벡터를 중심으로 작은 영역에 대해서 움직임 벡터를 재탐색한 결과는 전역탐색과 거의 유사한 PSNR을 나타내고 있다. 그림 7을 보면 전역탐색 기법의 경우 계산량이 매우 크고, 움직임 벡터 재사용한 경우 한 프레임당 매우 짧은 연산시간을 보여주고 있다. 재사용 움직임 벡터를 중심으로 재탐색한 경우, 재사용 기법에 비해 약간 더 긴 연산시간을 필요로 하지만 전역탐색과 비교해 매우 작은 연산량만을 필요로 하는 것을 알 수 있다.

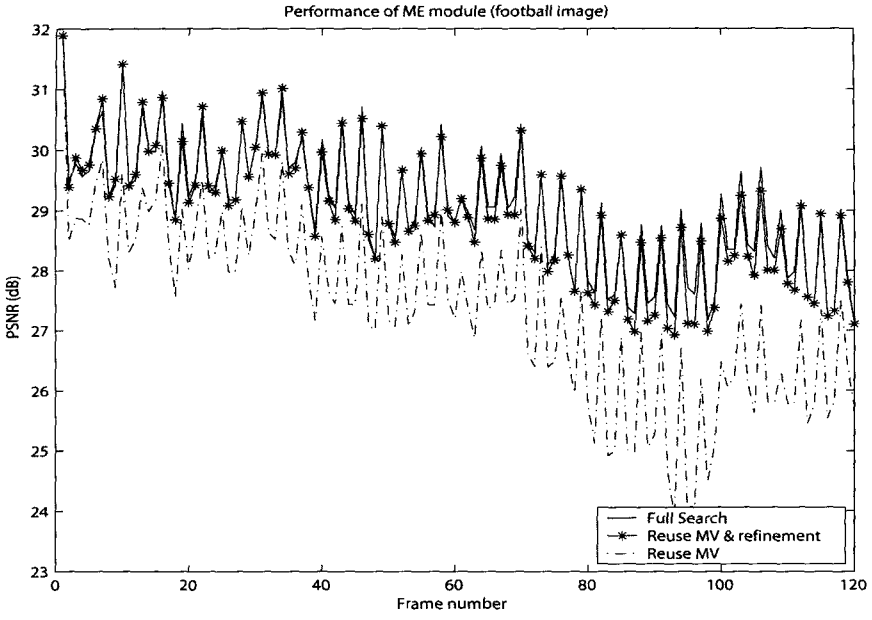


그림 6. 해상도 변환 트랜스코더의 움직임 예측 기법에 따른 PSNR (일반적인 3차 회선 해상도 변환기 사용)
Fig. 6. The PSNR comparison of the images resulted from the transcoding systems using the up/down scaler. The ME schemes are "Full search ME," "Reuse and refinement ME," and "Reused ME." The scalers are the conventional cubic convolution scalers.

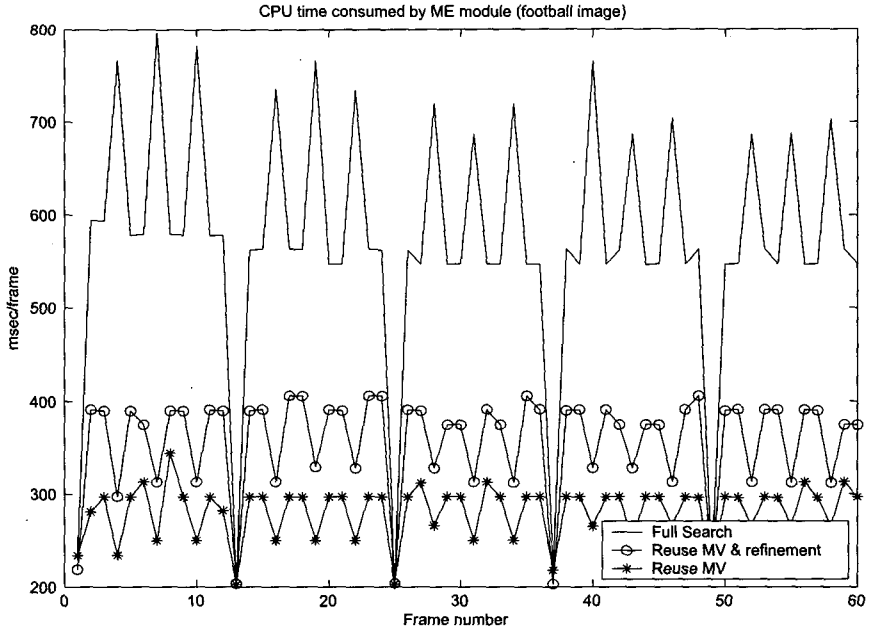


그림 7. 움직임 예측기법에 따른 각 프레임별 부호화 시간
Fig. 7. The CPU time's consumed by "Full Search," "Reuse MV & refinement," and "Reuse MV."

3. 공동 최적화 알고리듬 결과

그림 8과 9는 표 2에서 정리한 공동 최적화 알고리듬의 결과와 기존 시스템과의 비교결과를 보여주고 있다. 그림 8은 해상도 변환 최적화 알고리듬과 움직임 최적화 알고리듬, 공동 최적화 알고리듬을 비교하고 있다. "Opt. of scal-

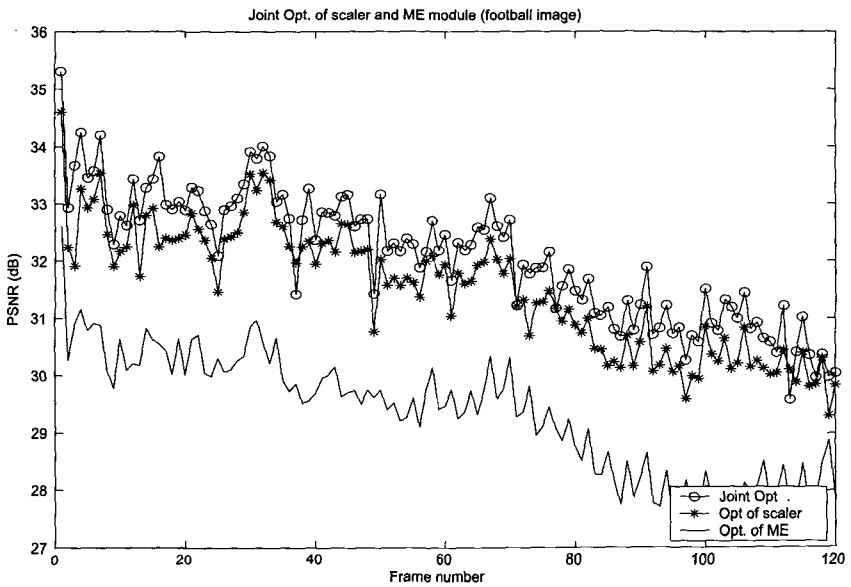


그림 8. 본 논문에서 제안한 움직임 예측기법과 해상도 변환기법, 공동 최적화 알고리듬의 PSNR 비교(Football 영상)
Fig. 8. The PSNR comparison of the images (football image) resulted from the transcoding systems, "Joint optimized transcoder," "Transcoder optimized in the scaler only," and "Transcoder optimized in the ME module only."

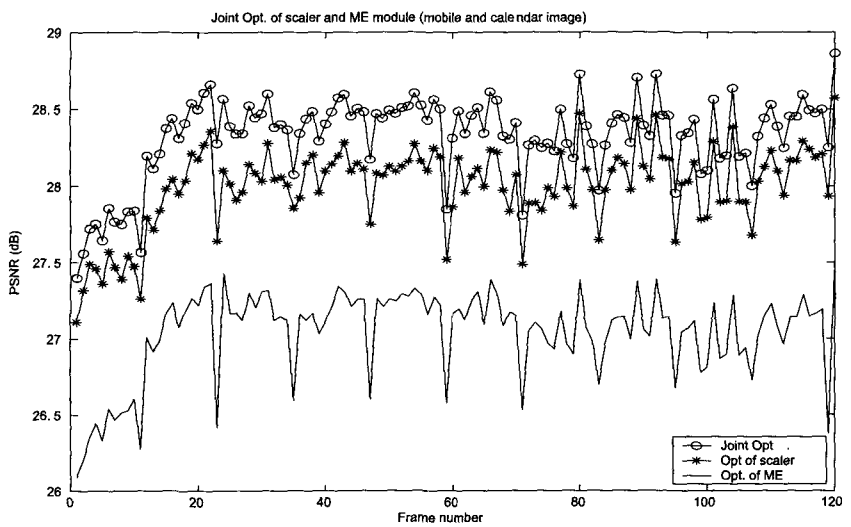


그림 9. 본 논문에서 제안한 움직임 예측기법과 해상도 변환기법, 공동 최적화 알고리듬의 PSNR 비교 (Mobile and calender 영상)
Fig. 9. The PSNR comparison of the images (mobile and calendar image) resulted from the transcoding systems, "Joint optimized transcoder," "Transcoder optimized in the scaler only," and "Transcoder optimized in the ME module only."

er" 는 표 1에서 정리한 해상도 변환기에 대한 결과이고, "Opt. of ME" 는 본 논문의 움직임 벡터 최적화 알고리듬의 결과이다. 단 움직임 벡터의 최적화는 $\alpha=-1/2$, $\beta=-1/2$로 3차 회선 변환기로 해상도를 축소시킨 다음 본 논문에서 제안하는 움직임 예측 알고리듬을 사용하였다. "Joint Opt."는 표 2에 정리한 공동 최적화이다. 실험을 위해 "football"영상과 "mobile and calender" 영상을 사용하였다. 실험 결과 공동 최적화 알고리듬의 화질이 가장 깨끗한 것을 알 수 있다.

Ⅶ. 결론

본 논문은 해상도 변환기와 움직임 예측기의 공동 최적화에 대해 다루었다. 제안하는 알고리듬은 트랜스코더의 움직임 예측과 해상도 변환을 통해 발생하는 손실을 양쪽 모두에서 축소시킨다.

첫째. 3차 회선 변환기의 최적화된 커널을 각 매크로블록 단위로 예측하였다. 해상도 변환의 성능이 기본 커널의 최적화와 상관도가 높은 것을 증명하였고, 몇 가지 기존 알고리듬과 비교해 본 논문에서 제안하는 해상도 변환 알고리듬이 가장 정보의 손실이 적은 것을 알 수 있었다. 둘째로 움직임 예측모듈과 해상도 변환기의 공동최적화에 대해 다루었다. 시뮬레이션 결과 기존 알고리듬과 움직임 예측 모듈만의 최적화에 비해 더욱 좋은 화질을 보여주었다.

부 록

1. $\frac{\partial \tilde{f}(l,m)}{\partial \alpha}$, $\frac{\partial \tilde{f}(l,m)}{\partial \beta}$, $\frac{\partial \hat{f}(n,k)}{\partial \alpha}$, $\frac{\partial \hat{f}(n,k)}{\partial \beta}$의 정리

식(15), (16), (17), (18)의 $\frac{\partial \tilde{f}(l,m)}{\partial \alpha}$, $\frac{\partial \tilde{f}(l,m)}{\partial \beta}$, $\frac{\partial \hat{f}(n,k)}{\partial \alpha}$, $\frac{\partial \hat{f}(n,k)}{\partial \beta}$에 대한 정리이다. 그림 10은 그림 2의 3차 회선 해상도 변환기를 사용해 영상의 해상도를 변환시키는 과정을 좀 더 자세하게 나타내고 있다. 우선 $f(n,k)$는 가로방향으로 $\tilde{f}_h(n,m)$으로 축소

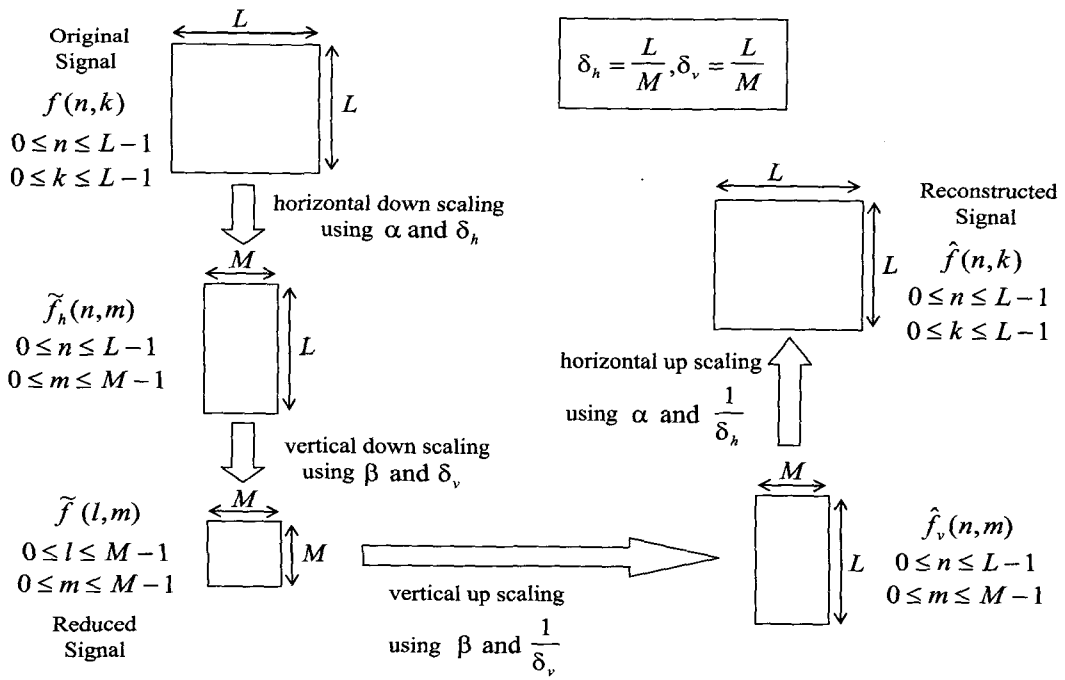


그림 10. 3차 회선 해상도 변환기를 사용한 영상 축소와 영상 재건과정

Fig. 10. The detailed description of the resolution reduction and reconstruction using the cubic convolution scaler.

된다. 이를 수식으로 나타내면

$$\begin{aligned}\tilde{f}_h(n,m) &= f(n,k^*-1)(\alpha s^3-2\alpha s^2+\alpha s)\\ &+ f(n,k^*)((\alpha+2)s^3-(3+\alpha)s^2+1)\\ &+ f(n,k^*+1)(-(\alpha+2)s^3+(2\alpha+3)s^2-\alpha s)\\ &+ f(n,k^*+2)(-\alpha s^3+\alpha s^2)\end{aligned} \quad (A1)$$

와 같고 여기서 $k^*=\lfloor l\times\delta_v\rfloor$, $s=m\times\delta_h-k^*$이다. δ_h는 가로방향의 영상 축소비율이고 이는 L/M과 같다. $\lfloor A\rfloor$는 A보다 작은 가장 큰 정수이다. 가로방향으로 축소된 영상신호 $\tilde{f}_n(n,m)$는 다시 세로방향으로 축소된다.

$$\begin{aligned}\tilde{f}(l,m) &= \tilde{f}_h(n^*-1,m)(\beta t^3-2\beta t^2+\beta t)\\ &+ \tilde{f}_h(n^*,m)((\beta+2)t^3-(3+\beta)t^2+1)\\ &+ \tilde{f}_h(n^*+1,m)(-(\beta+2)t^3+(2\beta+3)t^2-\beta t)\\ &+ \tilde{f}_h(n^*+2,m)(-\beta t^3+\beta t^2)\end{aligned} \quad (A2)$$

여기서 $n^*=\lfloor l\times\delta_v\rfloor$이고, $t=l\times\delta_v-n^*$이다. δ_v는 가로방향의 축소비율이고 이는 L/M과 같다. 축소된 영상은 재건영상을 만들기 위해 다시 세로 방향으로 확대한다.

$$\begin{aligned}\hat{f}_v(n,m) &= \tilde{f}(l^*-1,m)(\beta t^3-2\beta t^2+\beta t)\\ &+ \tilde{f}(l^*,m)((\beta+2)t^3-(3+\beta)t^2+1)\\ &+ \tilde{f}(l^*+1,m)(-(\beta+2)t^3+(2\beta+3)t^2-\beta t)\\ &+ \tilde{f}(l^*+2,m)(-\beta t^3+\beta t^2)\end{aligned} \quad (A3)$$

여기서 $l^*=\lfloor n\times\frac{1}{\delta_v}\rfloor$이고, $t=n\times\frac{1}{\delta_v}-l^*$이다. 세로방향으로 확대된 영상신호를 가로방향으로 확대해 재건연상 $\hat{f}(n,k)$을 만든다.

$$\begin{aligned}\hat{f}(n,k) &= \hat{f}_v(n,m^*-1)(\alpha s^3-2\alpha s^2+\alpha s)\\ &+ \hat{f}_v(n,m^*)((\alpha+2)s^3-(3+\alpha)s^2+1)\\ &+ \hat{f}_v(n,m^*+1)(-(\alpha+2)s^3+(2\alpha+3)s^2-\alpha s)\\ &+ \hat{f}_v(n,m^*+2)(-\alpha s^3+\alpha s^2)\end{aligned} \quad (A4)$$

여기서 $m^*=\lfloor k\times\frac{1}{\delta_h}\rfloor$이고 $s=k\times\frac{1}{\delta_h}-m^*$이다.

$\frac{\partial\tilde{f}(l,m)}{\partial\alpha}$을 구하기 위해 식(A2)를 α에 대해 편미분하면

$$\begin{aligned}\frac{\partial\tilde{f}}{\partial\alpha}(l,m) &= \frac{\partial\tilde{f}_h}{\partial\alpha}(n^*-1,m)(\beta t^3-2\beta t^2+\beta t)\\ &+ \frac{\partial\tilde{f}_h}{\partial\alpha}(n^*,m)((\beta+2)t^3-(3+\beta)t^2+1)\\ &+ \frac{\partial\tilde{f}_h}{\partial\alpha}(n^*+1,m)(-(\beta+2)t^3+(2\beta+3)t^2-\beta t)\\ &+ \frac{\partial\tilde{f}_h}{\partial\alpha}(n^*+2,m)(-\beta t^3+\beta t^2)\end{aligned} \quad (A5)$$

이고 $n^*=\lfloor l\times\delta_v\rfloor$, $t=l\times\delta_v-n^*$,

$$\begin{aligned}\frac{\partial\tilde{f}_h}{\partial\alpha}(n^*,m) &= f(n,k^*-1)(s^3-2s^2+s)\\ &+ f(n,k^*)(s^3-s^2)\\ &+ f(n,k^*+1)(-s^3+2s^2-s)\\ &+ f(n,k^*+2)(-s^3+s^2)\end{aligned} \quad (A6)$$

이다. 여기서 $k^*=\lfloor m\times\delta_h\rfloor$, $s=m\times\delta_h-k^*$이다. 식(A6)은 식(A1)의 α에 대한 편미분으로 구할 수 있다. 식(A2)의 β에 대한 편미분은

$$\begin{aligned}\frac{\partial\tilde{f}}{\partial\beta}(l,m) &= \tilde{f}_h(n^*-1,m)(t^3-2t^2+t)\\ &+ \tilde{f}_h(n^*,m)(t^3-t^2)\\ &+ \tilde{f}_h(n^*+1,m)(-t^3+2t^2-t)\\ &+ \tilde{f}_h(n^*+2,m)(-t^3+t^2)\end{aligned} \quad (A7)$$

이고 여기서 $n^*=\lfloor l\times\delta_v\rfloor$, $t=l\times\delta_v-n^*$이다.
식(A4)에 대해 α와 β에 대해 각각 편미분을 하면

$$\begin{aligned}\frac{\partial\tilde{f}}{\partial\alpha} &= \frac{\partial\tilde{f}_v}{\partial\alpha}(n,m^*-1)(\alpha s^3-2\alpha s^2+\alpha s)\\ &+ \frac{\partial\tilde{f}_v}{\partial\alpha}(n,m^*)((\alpha+2)s^3-(3+\alpha)s^2+1)\\ &+ \frac{\partial\tilde{f}_v}{\partial\alpha}(n,m^*+1)(-(\alpha+2)s^3+(2\alpha+3)s^2-\alpha s)\\ &+ \frac{\partial\tilde{f}_v}{\partial\alpha}(n,m^*+2)(-\alpha s^3+\alpha s^2)\\ &+ \tilde{f}_v(n,m^*-1)(s^3-2s^2+s)\\ &+ \tilde{f}_v(n,m^*)(s^3-s^2)\\ &+ \tilde{f}_v(n,m^*+1)(-s^3+2s^{2-s})\\ &+ \tilde{f}_v(n,m^*+2)(-s^3+s^2)\end{aligned} \quad (A8)$$

$$\begin{aligned}\frac{\partial\tilde{f}}{\partial\beta} &= \frac{\partial\tilde{f}_v}{\partial\beta}(n,m^*-1)(\alpha s^3-2\alpha s^2+\alpha s)\\ &+ \frac{\partial\tilde{f}_v}{\partial\beta}(n,m^*)((\alpha+2)s^3-(3+\alpha)s^2+1)\\ &+ \frac{\partial\tilde{f}_v}{\partial\beta}(n,m^*+1)(-(\alpha+2)s^3+(2\alpha+3)s^2-\alpha s)\\ &+ \frac{\partial\tilde{f}_v}{\partial\beta}(n,m^*+2)(-\alpha s^3+\alpha s^2)\end{aligned} \quad (A9)$$

여기서 $m^*=\lfloor k\times\frac{1}{\delta_h}\rfloor$, $s=k\times\frac{1}{\delta_h}-m^*$이다. 식 (A8)과 (A9)의 $\frac{\partial\tilde{f}_v}{\partial\alpha}(n,m)$와 $\frac{\partial\tilde{f}_v}{\partial\beta}(n,m)$에 대한 값은 식(A3)의 α와 β에

대한 편미분으로 아래와 같이 구할 수 있다.

$$\begin{aligned}\frac{\partial \tilde{f}_v}{\partial \alpha}(n,m) &= \frac{\partial \tilde{f}}{\partial \alpha}(l^*-1,m)(\beta s^3-2\beta s^2+\beta s)\\ &+ \frac{\partial \tilde{f}}{\partial \alpha}(l^*,m)((\beta+2)s^3-(3+\beta)s^2+1)\\ &+ \frac{\partial \tilde{f}}{\partial \alpha}(l^*+1,m)(-(\beta+2)s^3+(2\beta+3)s^2-\beta s)\\ &+ \frac{\partial \tilde{f}}{\partial \alpha}(l^*+2,m)(-\beta s^3+\beta s^2)\end{aligned} \quad (A10)$$

$$\begin{aligned}\frac{\partial \tilde{f}_v}{\partial \beta}(n,m) &= \frac{\partial \tilde{f}}{\partial \beta}(l^*-1,m)(\beta s^3-2\beta s^2+\beta s)\\ &+ \frac{\partial \tilde{f}}{\partial \beta}(l^*,m)((\beta+2)s^3-(3+\beta)s^2+1)\\ &+ \frac{\partial \tilde{f}}{\partial \beta}(l^*+1,m)(-(\beta+2)s^3+(2\beta+3)s^2-\beta s)\\ &+ \frac{\partial \tilde{f}}{\partial \beta}(l^*+2,m)(-\beta s^3+\beta s^2)\\ &+ \tilde{f}(l^*-1,m)(s^3-2s^2+s)\\ &+ \tilde{f}(l^*,m)(s^3-s^2)\\ &+ \tilde{f}(l^*+1,m)(-s^3+2s^2-s)\\ &+ \tilde{f}(l^*+2,m)(-s^3+s^2)\end{aligned} \quad (A11)$$

여기서 $l^* = \lfloor n\times\frac{1}{\delta_v} \rfloor$, $t = n\times\frac{1}{\delta_v} - l^*$이다.

참 고 문 헌

[1] J. Xin, M.-T.Sun, B.-S. Choi, and K.-W. Chun, "An HDTV-to-SDTV spatial transcoder," IEEE Trans. on Circuits and Systems for Video Tech., vol. 12, pp. 998-1008, Nov. 2002

[2] M.-J. Chen, M.-C Chu, and C.-W. Pan, "Efficient motion-estimation algorithm for reduced frame-rate video transcoder," IEEE Trans. on Circuits and Systems for Video Tech.,vol 12, pp. 269-275, April 2002.

[3] T. Shanabl도 and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolution and different encoding formats," IEEE Trans. on Multimedia, vol. 2, pp.101-110, June 2000.

[4] Bo Shen, I.K. Sethi, and B. Vasudev, "Adaptive motion vector resampling for compress video downscaling," IEEE Trans. on Circ. And Syst. for Video Tech., vol. 9, pp.929-936, Sept. 1999.

[5] J. Youn, M.-T. Sun, and Chia-Wen Lin, "Motion vector refinement for high performance transcoding," IEEE Trans. on Multimedia, vol. 1, pp.30-40, Mar. 1999.

[6] C.Yim and M.A. Isnardi, "An efficient method for DCT-domain image resizing with mixed field/frame-mode macroblocks," IEEE Trans. on Circuits and Systems for Video Tech., vol. 9, pp. 696-700, Aug. 1999

[7] J.Song and B-L Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, pp.767-775, Aug. 2000.

[8] N. Merhav, "Multiplication-free approximation algorithm for compressed-domain linear operations on images," IEEE Trans. on Image Processing, vol.8, pp. 247-254, Feb. 1999.

[9] S. Liu and A. C. Bovik, "Local bandwidth constrained fast inverse motion compensation for DCT-domain video transcoding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 12, pp. 309-319, May 2002.

[10] K.-T. Fung, Y.-L. Chan, and W.-C. Siu, "New architecture for dynamic frame-skipping transcoder," IEEE Trans. on Image Processing, vol. 11, pp.886-900, Aug. 2002.

[11] H. Sun, W.Kwok, and J.W. Zdepski, "Architecture for MPEG compressed bitstream scaling," IEEE Trans. on Circ. And Syst. for Video Techno., vol. 6, pp. 191-199, April 1999.

[12] C.D.L. Reyes, A.R. Reibman, S.-F.Chang, and J.C.-I. Chuang, "Error resilience transcoding for video over wireless channels," IEEE Journal of Selected Area in Commun.., vol. 18, pp. 1063-1074, June 2000.

[13] H. Hwang, "Interlaced to progressive scan conversion for HD-MAC application," IEEE Trans. on Consumer Electronics, vol. 38, no. 3, pp. 151-156, Aug. 1992.

[14] M. Karlsson, P. Pohjala, H. Rantanen, and S. Kalli, "Evaluation of scanning rate up conversion algorithm; subjective testing of interlaced to progressive conversion," IEEE Trans. on Consumer Electonics, vol. 38, no. 3, pp. 162-167, Aug. 1992.

[15] M. H. Lee, J. H. Kim, J. S. Lee, K. K. Ryu, and D. I. Song, "A new algorithm for interlaced to progressive scan conversion based on directional corrections and its IC design," IEEE Trans. on Consumer Electronics, vol. 40, no. 2, pp. 119-129, May 1994.

[16] R. Li, N. K. Chung, K. T. Mo, D. M. Fisher, and V. Wong, "A flexible display module for DVD and set-up box applications," IEEE Trans. on Consumer Electornics, vol. 43, no. 3, pp. 496-503, Aug. 1997.

[17] S. S. Rifman. "Digital Rectification of ERTS Multispectral Imagery," Proc. Symposium Significant Results Obtained from ERTS-1 (NASA SP-327), I, Sec. B, pp. 1131-1142, 1973.

[18] R. Berstein, "Digital Image Processing of Earth observation sensor data," IBM J. Res. Devel., vol. 20, pp. 40-57, 1976.

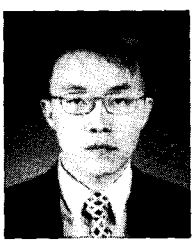
[19] R. G. Keys, "Cubic convolution interpolation for digital image processing," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 29, pp. 1153-1160, Dec. 1981.

[20] S. K. Park and R. A. Schowengerdt, "Image reconstruction by parametric cubic convolution," Computer Vision, Graphics, and Image Processing, vol. 23, pp. 258-272, Sept. 1983.

[21] K. W. Simon, "Digital image reconstruction and resampling of landsat imagery," Symposium on Machine Processing of Remotely Sensed Data, Purdue University, pp. 3A-1 -3A-11, June 1975.

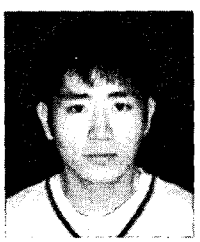
[22] D. G. Luenberger, Optimization by Vector Space Methods, New York, Wiley, 1969.
[23] D. G. Luenberger, Linear and nonlinear programming, Addison-Wesley, 1984.
[24] W. K. Pratt, Digital image processing, John Wiley & Sons, INC., 1991.
[25] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 13, pp. 277-285, March 1991.

저 자 소 개



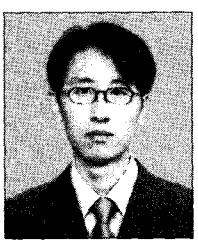
한 종 기

· 1992년 한국과학기술원(KAIST), 전기 및 전자공학과, 공학사
· 1994년 한국과학기술원(KAIST), 전기 및 전자공학과, 공학 석사
· 1999년 한국과학기술원(KAIST), 전기 및 전자공학과, 공학 박사
· 1999년 ~ 2001년 삼성전자 디지털 미디어 연구소, 책임연구원
· 2001년 ~ 현재 세종대학교 정보통신공학과 조교수
· 주관심분야 : 비디오 신호처리 및 압축, Transcoding, Scalable Video Coding



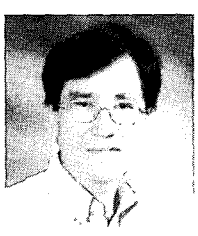
곽 상 민

· 2005년 세종대학교, 정보통신공학과, 공학사
· 2005년 ~ 현재, 세종대학교, 정보통신공학과, 석사과정
· 주관심분야 : 비디오 신호처리 및 압축, Transcoding, Scalable Video Coding



전 동 산

· 2002년 2월 : 부산대학교 전자공학과(공학사)
· 2004년 2월 : KAIST 전기 및 전자공학과(공학석사)
· 2004년 5월 ~ 현재 : ETRI 디지털방송연구단 방송미디어연구그룹 연구원
· 주관심분야 : 영상통신, 이미지 프로세싱, 패턴인식, Scalable Video Coding, TV-Anytime/MPEG-7/MPEG-21



김 재 곤

· 1990년 2월 : 경북대학교 전자공학과(공학사)
· 2005년 2월 : KAIST 전기 및 전자공학과(공학박사)
· 2001년 9월 ~ 2002년 11월 : 뉴욕 콜롬비아대학교 연구원
· 1992년 9월 ~ 현재 : ETRI 디지털방송연구단 방송미디어연구그룹 선임연구원/ 방통융합미디어연구팀장
· 주관심분야 : 영상통신, 비디오신호처리, 디지털방송, 멀티미디어 프래임워크, TV-Anytime/ MPEG-7/MPEG-21