

다양한 포맷변환을 지원하는 Transcoder의 개선된 Cubic Convolution Scaler

정희원 유영조*, 서주헌*, 종신회원 한종기*

Modified Cubic Convolution Scaler for Multiformat Conversion in a Transcoder

Young-Joe Yoo*, Ju-Heon Seo* *Regular Members*, Jong-Ki Han* *Lifelong Member*

요 약

본 논문에서는 임의의 배율로 디지털 영상을 확대와 축소하는 개선된 cubic convolution 보간법을 제안한다. 제안하는 공간해상도조정 방법은 압축된 HD 비트스트림이나 SD 비트 스트림을 transcoding시키는 과정에서 다양한 형식의 해상도를 조정하기위해 사용된다. Transcoder 등 다양한 응용분야에서 영상의 화질은 유지하면서 크기를 조정하는 것은 매우 중요한 기술이다. 해상도조정은 원본 디지털 데이터를 연속함수로 변환하는 단계와 새로운 샘플링간격에 맞게 재표본화하는 단계, 이렇게 두 단계로 구분된다. 우리는 원본영상과 해상도조정 된 영상사이의 관계를 고려하여 보간 필터의 kernel 개선에 초점을 맞췄다. 본 논문에서는 MPEG 표준에서 고려하는 다양한 영상 포맷들간 해상도를 변환시키기 위한 기술을 제안한다. 이 방법은 기존의 방법과 비교하였을 때, 정보 손실을 최소화하여 고화질의 해상도 변환 기능을 가능하게 한다.

Key Words : Transcoder, Cubic convolution scaler

ABSTRACT

We derive a modified version of cubic convolution interpolation for the enlargement or reduction of digital images by arbitrary scaling factors. The proposed scaling scheme is used to resize various format pictures in the transcoding system, which transforms the bitstream compressed at a bit rate, such as the HD bitstream, into another bit rate stream. In many applications such as the transcoder, the resolution conversion is very important for changing the image size while the scaled image maintains high quality. We focus on the modification of the scaler kernel according to the relation between formats of the original and the resized image. In the modification, various formats defined in MPEG standards are considered. We show experimental results that demonstrate the effectiveness of the proposed interpolation method.

I. 서론

Transcoding 기법은 대역폭이 다른 이종 네트워크 상에서 다양한 비디오 통신을 위한 중요한 기술이다. 예를 들어, 주문형 비디오 서비스시스템에서 서

버에 저장된 멀티미디어 정보는 채널 특성을 고려하지 않고 압축되어 있다. 이러한 특성은 미리 코딩된 비디오신호를 이질적인 네트워크를 통해 전송할 때 네트워크 연동에 문제를 초래한다. 이런 문제를 극복하는 효율적인 방법은 미리 코딩된 비트스트림

※ 본 연구는 서울시 산학연 협력사업(과제번호 10557)의 지원에 의하여 이루어진 것임.

* 세종대학교, 정보통신연구소, 정보통신공학과 (uline@nate.com, crazydreamer99@msn.com, hjk@sejong.ac.kr)

논문번호 : KICS2007-07-287, 접수일자 : 2007년 6월 29일, 최종논문접수일자 : 2007년 9월 7일

과 전송 채널사이의 연동기능을 제공하는 transcoder를 사용하는 것이다. Transcoder는 높은 비트율의 비트스트림을 받아 새로운 대역폭에 맞는 낮은 비트율의 비트스트림으로 변환시킨다.

Transcoding의 종류는 공간영역 transcoding과 주파수영역 transcoding으로 구분 지을 수 있다. 공간 영역의 처리 transcoder에서는 저장된 비트스트림을 화소신호로 복호화 한 후 화소영역에서 해상도 변환 처리가 수행된다. 그 해상도 변환된 신호를 목표 비트율에 맞게 재압축 한다^[1-3]. Shanableh와 Ghanbari [3]는 처리 속도를 높이면서 시간적 공간적 해상도를 변화시키는 transcoding 방법을 제안하였다. DCT 영역 transcoding는 화소신호로 복원함 없이 DCT 영역에서 직접적으로 transcoding 작업이 수행된다^[4-7]. [4]에서 Yim과 Isnardi는 DCT 영역에서의 해상도조정 비율이 2:1일때 효율적인 영상 크기 조정 방법을 설명하고 있다. [6]에서는 곱하기를 사용하지 않으면서 2:1비율의 해상도 축소과정을 수행하는 선형 처리 근사화에 대해 제안하였다. 이런 연구에서 알 수 있듯이 높은 화질을 유지하면서 해상도를 변환하는 것은 transcoding 방식에서 매우 중요하다. 기존방법들에서 제안된 transcoder는 정수비율의 공간 해상도 크기 조정을 하는 반면, 본 논문에서 제안하는 방식은 임의의 해상도 변환을 고려한다. 해상도조정 방법은 주어진 디지털 데이터를 연속신호 모델로 변환한 후 새로운 표본화 간격에 맞게 재표본화 한다[8]. 표본화 이론에 따르면 주어진 디지털신호와 sinc함수를 convolution시킴으로써 원래 아날로그 신호를 완벽하게 복원할 수 있다. 하

지만 sinc함수가 시간축으로 무한대에 걸쳐 존재하기 때문에, 이 함수를 물리적으로 실현하는 것은 어렵다. 현실적인 구현을 위해서 bilinear^[9], bicubic^[10], cubic spline^[11], B-spline^[8,12] 등 여러가지 다른 근사(近似)한 방법들이 제안 되었다. 더 좋은 화질의 해상도 조정된 영상을 얻기 위해 Park and Schowengerdt^[13]는 cubic convolution 보간법에 주파수 개념을 적용시켜왔고, Ramponi^[14]는 warped 거리 개념을 도입한 공간-변형 기법을 제안했다.

본 논문에서는 해상도와 비트율을 변환하는 transcoder를 설계한다. 이 때 임의의 비율로 크기조정이 가능한 cubic convolution기법을 사용한다. MPEG-2 표준으로 정의된 비디오코딩 알고리즘의 유연성은 다양한 형식과 영상의 크기를 지원하는 것으로 알려져 있다. 영상의 다양한 형식을 다루기 위해 본 논문에서는 해상도 변환 신호의 표본화 위치들간의 관계를 고려함으로써 개선된 cubic convolution scaler을 제안한다. 이 논문에서 제안하는 두 가지 개념은 다음과 같다. 첫째, 기존의 transcoder^[4,6]는 2분의 1 또는 4분의 1 크기로 영상의 해상도를 변환하는 반면, 제안하는 시스템에서는 1920×1080i에서 704×408i로 변환처럼 임의의 비율로 영상의 크기를 조정한다. 임의의 비율로 크기 변환하는 것은 기존의 방법보다 현실적이다. 두 번째로, 다양한 영상 포맷간 변환을 지원하기 위해 재표본화 위치의 조정을 제안한다. 예를 들어 순차주사 형태의 이미지를 순차주사로, 비월주사형태의 이미지를 비월주사로, 비월주사형태의 이미지를 순차주사로, 순차주사형태의 이미지를 비월주사로 변환하는

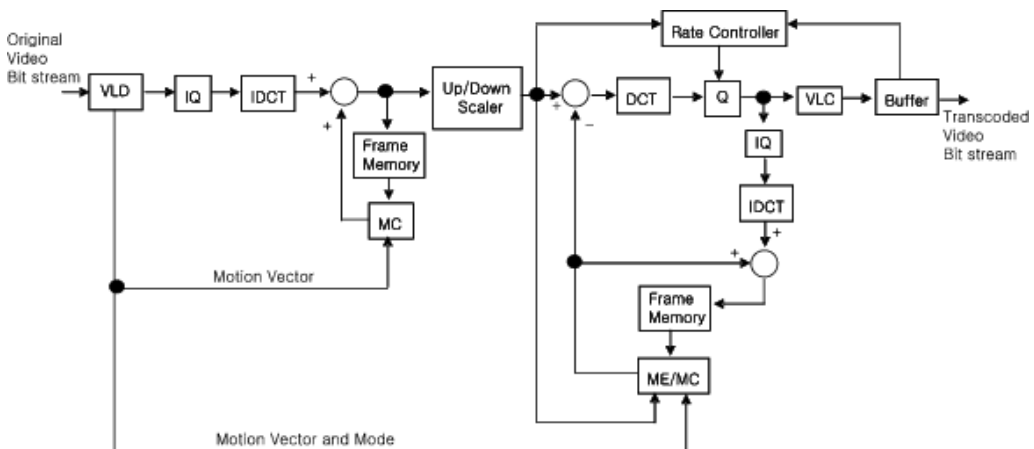


그림 1. 영상의 확대/축소가 가능한 transcoder 구조

경우에 대하여 표본화위치 보정기술을 제안하였다.

이 논문은 다음과 같이 구성되었다. II장은 transcoding과 다양한 이미지 포맷에 대하여 기술하고 있다. Transcoder에서 요구하는 크기 조정 방법은 보간법에 대한 기본 정보와 함께 III장에서 설명하고 있다. IV장에서는 개선된 cubic convolution 보간법을 제안한다. 제안한 알고리즘의 컴퓨터실험 결과를 V장에서 설명하고 있다. 결론은 VI장에서 서술된다.

II. Transcoder

2.1 Transcoder

디지털 저장 매체의 저장효율을 높이고 대역폭이 제한된 채널을 통한 전송품질을 높이기위해서 전송하려는 MPEG-2 비트스트림의 비트율 감소는 피할 수 없는 과정이다.^[2,3,15,16] 그림 1에서 transcoder는 양자화 스텝 크기와 공간과 시간상의 해상도등 부호화 변수를 조절함으로써 비트율을 감소시킨다. 그림에서 scaler는 확대/축소 기능을 수행하는 모듈이다. 입력 비디오 비트스트림은 가변길이 복호기(VLD)에 의해 복호화 되고, 그 후 역양자화기(IQ)에 의해 역양자화 된다. 역양자화 된 데이터는 inverse discrete cosine transform(IDCT)에 의해 화소 영역으로 복원된다. 복원된 데이터는 움직임 보상기(MC)에 의해 복원영상으로 변환된다. 복원된 영상은 scaler를 통해 다른 크기의 영상으로 해상도가 조정된다. 해상도크기가 조정된 영상은 다시 다른 비트율을 갖는 비트스트림으로 부호화된다. 그림 1에서 ME/MC, VLC, Q는 각각 움직임 예측/움직임 보상, 가변 길이 부호기, 양자화기를 나타낸다. 복호화단계에서 얻은 움직임 벡터와 모드에 대한 정보는 부호화단계에서 재사용할 수 있다. 수신된 움직임 벡터의 재사용 기법은 기존의 많은 transcoder 연구들에서 사용되었다^[3]. 이 논문에서는 복호화단계에서 복호된 움직임 벡터를 공간 해상도 감소 비율에 맞게 크기를 조절한다. 크기가 조절된 움직임 벡터주위를 가로 세로 ± 2 화소의 좁은 탐색 영역에 걸쳐 움직임 벡터 보정(refinement)를 수행한다. 우리는 MPEG-2 MP@HL과 MP@ML에서 SDTV 등 다양한 형식의 비디오 신호를 다루는 시스템을 고려한다. 이는 MPEG표준^[1]으로 정의된 다양한 영상 포맷들 사이의 변환을 지원해야함을 의미한다. 구체적으로 고려되는 형식은 1920x1080i, 1280x 720p, 704x480p, 704x480i, 640x480p를 포

함한다. 고려하는 가능한 형식들을 그림 2에서 나타내고 있다. i는 비월주사를, p는 순차주사를 의미한다. 이미지의 해상도가 수평 또는 수직 방향으로 정수가 아닌 비율로 변경되는 것을 유념하고 scaler를 설계하였다. 기존의 비월주사영상을 순차주사영상으로 변환하는 기술(IPC)은 오로지 수직적인 방향으로만 해상도를 두 배 증가시키는 것이다.^[18-20]

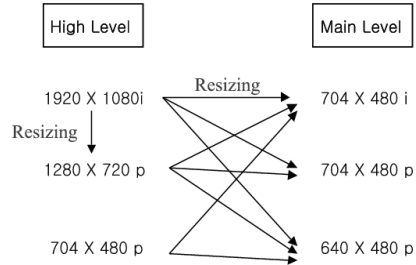


그림 2. MPEG-2표준의 다양한 영상 형식

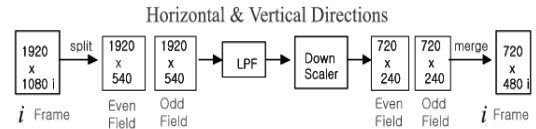


그림 3. 1920x1080i에서 720x480i로 변환 과정

본 논문에서는 가능한 모든 형식의 영상에 대해 처리가 가능한 개선된 해상도 크기 변환에 설계에 초점을 맞췄다. 그림 3, 4, 5는 각각 1920x1080i를 702x480i로, 1920x1080i를 1280x720p로, 그리고 1280x720p를 720x480i로 변환하는 과정을 보여주고 있다. 원본 영상이 비월주사일 때 splitter를 사용하여 even field와 odd field를 생성한다. 그리고 복원 영상이 비월주사일 경우는 merger를 사용하여 두 장의 field 영상을 가지고 하나의 프레임을 만들게 된다. 그림 4에서 1920x1080i 해상도를 1280x 720p 해상도로 변형할 때, 이 때 가로의 해상도는 감소하지만, 세로의 해상도는 증가하도록 변형한다. 그러므로 확대 변환과 축소 변환이 각각 분리되어 세로방향과 가로방향에 적용되어야 한다. 그림 4에서는 하나의 i 프레임이 두 개의 p 프레임으로 크기 조정되고, 반면 그림 5에서는 두 개의 p 프레임이 하나의 i frame으로 변환 된다.

III. 해상도조정

이번 절에서는 해상도조정 모듈의 설계에 대해

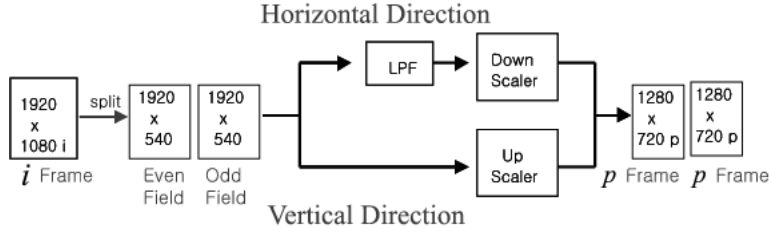


그림 4. 1920x1080i에서 1280x720p로 변환 과정

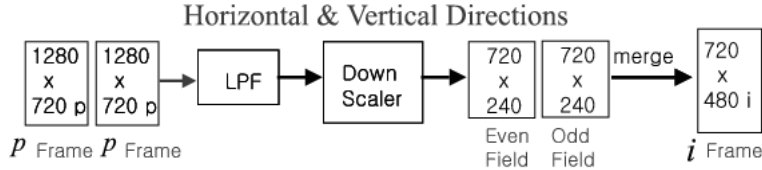


그림 5. 1280x720p에서 720x480i로 변환 과정

설명한다. 해상도변환의 과정은 불연속적인 원본 데이터를 연속하는 함수로 변형하고, 이 연속함수를 새로운 표본화 위치에서 재표본화한다. 주어진 원본 데이터 $f(x_k)$ 에 대해서 연속함수를 만드는 과정은 다음과 같이 표현될 수 있다.

$$\tilde{f}(x) = \sum_{k=-\infty}^{\infty} c_k \beta(x - x_k) \quad (1)$$

위 식에서 $\tilde{f}(x)$ 는 보간된 연속함수이고 $\beta(x)$ 는 보간 kernel이다. x 와 x_k 는 각각 연속적인 공간좌표와 불연속적인 공간좌표를 각각 나타낸다. 식 (1)에서 x_k 는 주어진 디지털신호의 샘플링위치이고 c_k 는 표본화된 데이터 $f(x_k)$ 에 의존적인 변수이다. 보간 kernel $\beta(x)$ 는 convolution과정을 통하여 불연속적인 데이터 $f(x_k)$ 를 연속적인 데이터 $\tilde{f}(x)$ 로 변환시킨다.

종래의 Shannon 표본화 이론에 의하면, $f(x)$ 가 $(-\pi, +\pi)$ 로 대역폭이 제한되어 있을 때 아래와 같은 방법으로 이상적인 연속함수를 복원할 수 있다.

$$\tilde{f}(x) = \sum_{k=-\infty}^{\infty} f(x_k) \text{sinc}(x - x_k) \quad (2)$$

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad (3)$$

식 (1)과 연관지어 설명하면 c_k 와 $\beta(x)$ 는 각각 $f(x_k)$ 와 $\text{sinc}(x)$ 로 사용된다. 식 (2)의 이상적인 보

간 공식은 보간 kernel인 $\text{sinc}(x)$ 가 무한대시간영역에 존재하기 때문에 실제 계산이 불가능하다. 이 문제를 해결하는 방법은 구현 가능한 다항식을 이용해 근사적으로 복원하는 것이다. 여러 가지 실제적인 방법들은 새로운 샘플링 위치의 화소 값을 이웃한 화소의 밝기 값을 이용하여 예측하는 것이다.

$f(x_k)$ 를 주어진 해상도의 데이터라 하고, $\tilde{f}(x)$ 를 보간된 연속함수라고 하자. 이때 사용할 수 있는 샘플 중 가장 가까이 있는 데이터의 위치 좌표를 x_k 와 x_{k+1} 이라하고, 주어진 데이터의 표본화 간격은 1이라고 가정하면, x , x_k , x_{k+1} 는 다음과 같이 정의 된다.

$$s = x - x_k \quad (4)$$

$$1 - s = x_{k+1} - x \quad (5)$$

위 식에서 $0 \leq s \leq 1$ 이고 $x_k \leq x \leq x_{k+1}$ 이다. 보간된 연속함수 $\tilde{f}(x)$ 를 얻기 위해 사용되는 많은 방법중 간단한 bi-linear보간 알고리즘은 다음과 같다.

$$\tilde{f}(x) = (1 - s)f(x_k) + sf(x_{k+1}) \quad (6)$$

Bi-linear기법보다 복잡하지만 상당히 효율적인 다른 방법은 bicubic 방법이다^[11]. 이 방법의 문제점은 복원된 연속함수에 진폭 잔물결을 발생시키는 것이다. 이런 문제점은 보간의 끝부분의 경사를 강제로 고정시키는 cubic convolution 보간 함수^[10,22,23]를 사용함으로써 제거될 수 있다. 이 cubic

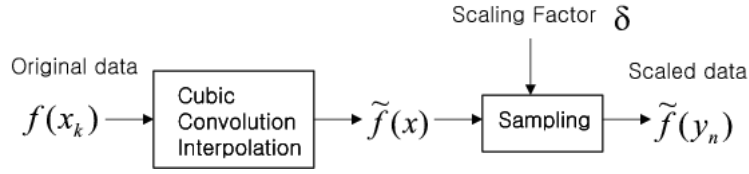


그림 6. 해상도 변환과정의 블록다이어그램

convolution 보간 기법의 kernel 함수는 다음과 같이 표현된다.

$$\beta(x) = \begin{cases} (\alpha + 2)|x|^3 - (\alpha + 3)|x|^2 + 1, & 0 \leq |x| \leq 1 \\ \alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha, & 1 \leq |x| \leq 2 \end{cases} \quad (7)$$

식 (7)에서 α 는 $\beta(x)$ 의 모양을 조절할 수 있는 조정인자(tuning parameter)이다. Rifman[22]과 Bernstein[23]은 $\text{sinc}(x)$ 함수처럼 $x=1$ 에서 -1의 기울기를 가지는 $\beta(x)$ 를 도출해내는 $\alpha=-1$ 로 정하였다. Keys[20]는 먹급수 확장의 관점에서 가능한 한 높은 차수로 원본 연속함수를 근사시키는 보간 함수를 유도하는 $\alpha=-1/2$ 로 사용했다. 식 (7)에서 (-2,2) 구간을 제외하고는 0이기에 때문에, 식 (1)에 식 (4)와 (7)를 대입하여 식은 다음과 같이 사용한다.

$$\begin{aligned} \tilde{f}(x) = & f(x_{k-1})(\alpha s^3 - 2\alpha s^2 + \alpha s) + \\ & f(x_k)[(\alpha + 2)s^3 - (3 + \alpha)s^2 + 1] + \\ & f(x_{k+1})[-(\alpha + 2)s^3 + (2\alpha + 3)s^2 - \alpha s] + \\ & f(x_{k+2})(-\alpha s^3 + \alpha s^2) \end{aligned} \quad (8)$$

위 식에서 $x_k \leq x \leq x_{k+1}$, $s = x - x_k$, $0 \leq s \leq 1$ 이고, 보간된 연속함수들은 $\tilde{f}(x) = f(x_k)$ 와 같이 주어진 디지털 신호와 일치한다. 이 식을 이용하면 보간 함수들은 어느 배율(정수, 유리수, 무리수)로도 보간될 수 있다. $\alpha=-1/2$ 를 식 (8)에 적용하면 다음과 같이 정리된다.

$$\begin{aligned} \tilde{f}(x) = & f(x_{k-1})(-s^3 + 2s^2 - s)/2 + \\ & f(x_k)(3s^3 - 5s^2 + 2)/2 + \\ & f(x_{k+1})(-3s^3 + 4s^2 + s)/2 + \\ & f(x_{k+2})(s^3 - s^2)/2 \end{aligned} \quad (9)$$

식 (7)에서 α 는 최적의 보간법을 얻기 위해 조율되도록 사용될 수 있다^[13]. Park과 Schowengerdt^[13]은 보간시킬 영상신호의 주파수 성분에 적응적으로 α 값을 설정해서 사용하였다.

Cubic convolution은 bilinear 보간보다 성능은 좋고, B-spline 보간^[10]보다는 간단하기 때문에 본 논문에서는 식 (9)의 cubic convolution 방법을 이용하는 알고리즘을 제안한다. 다음 절에서는 다양한 영상형식의 해상도를 조정하기 위해 식 (9)의 개선 방법을 제안한다.

IV. 개선된 Cubic Convolution 크기 조정

그림 6은 해상도 변환과정의 구조도를 보여준다. 여기서 $f(x_k)$ 는 주어진 원본 디지털 신호를 의미한다. 변환과정은 연속하는 함수 $\tilde{f}(x)$ 를 만들기 위한 보간과 새로운 간격 y_n 으로 연속함수를 재표본화하는 두 단계로 구성된다. 해상도 조정과정은 불연속 데이터 $f(x_k)$ 를 다른 해상도를 갖도록 크기 조정된 데이터 $\tilde{f}(y_n)$ 으로 변환시키게 된다. $\delta > 1$ 면, $f(x_k)$ 는 $\tilde{f}(y_n)$ 로 해상도가 확대 된다.

$$\delta = \frac{N}{M} \quad (10)$$

여기서 M 과 N 은 각각 원본영상과 크기 조정된 영상의 크기를 나타낸다.

보간된 연속함수 $\tilde{f}(x)$ 는 식 (9)에 의해 계산된다. 해상도조정은 연속 함수 $\tilde{f}(x)$ 를 다음 식과 같이 재표본화하는 과정이다.

$$\tilde{f}(y_n) = \tilde{f}(x) \Big|_{x=y_n = n \cdot \frac{1}{\delta}}, \quad 0 \leq n < N \quad (11)$$

$x_k \leq y_n \leq x_{k+1}$ 이고 $s = y_n - x_k$ 이며 식 (11)은 다음과 같이 된다.

$$\tilde{f}(y_n) = \tilde{f}(x) \Big|_{s=y_n-x_k}, \quad 0 \leq n < N, \quad 0 \leq k < M \quad (12)$$

영상신호와 같은 다차원 데이터에 이 방법을 적용하는 것은 그림 6이 설명하고 있는 해상도조정 과정을 가로축과 세로축에 독립적으로 적용함으로써

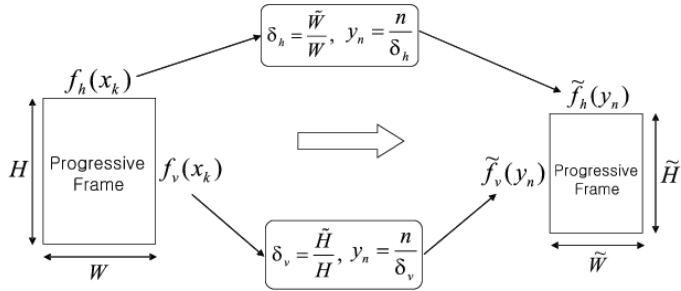


그림 7. 순차주사영상을 순차주사영상으로 해상도 변환하는 과정

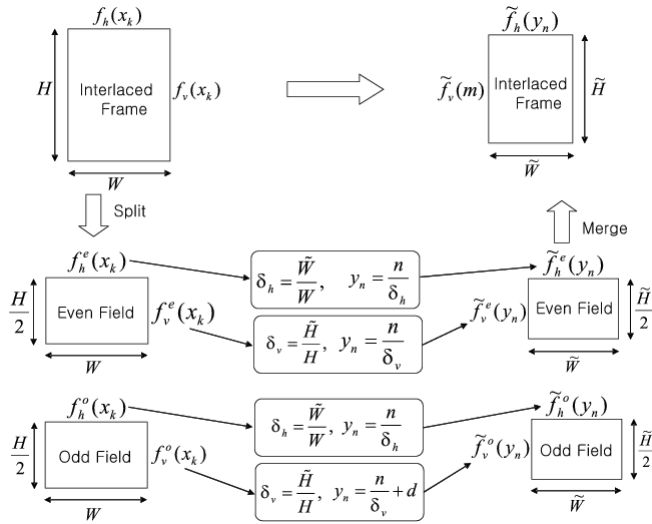


그림 8. 비월주사영상을 비월주사영상으로 해상도 변환하는 과정

수행된다. 다음 세부 절에서는 처리하는 영상의 형식에 따라 해상도조정방법을 설명한다.

4.1 순차주사 영상의 해상도조정

순차주사영상을 다른 해상도를 가지는 순차주사영상으로 변형하기 위해 scaler가 사용되는 과정이 그림 7에 설명되어 있다. 그림에서 $W, H, \tilde{W}, \tilde{H}$ 는 각각 원본영상의 폭과 높이, 변환된 영상의 폭과 높이를 나타낸다. 수평 방향 재표본화 과정에서는 배율 $\delta_h = \tilde{W}/W$ 이 사용된다.

$$\tilde{f}_h(y_n) = \tilde{f}(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h} = n \cdot \frac{W}{\tilde{W}}}, n = 0, 1, 2, \dots, \tilde{H}-1 \quad (13)$$

$$\begin{aligned} \tilde{f}_h(x) = & f_h(x_{k-1})(-s^3 + 2s^2 - s)/2 \\ & + f_h(x_k)(3s^3 - 5s^2 + 2)/2 \\ & + f_h(x_{k+1})(-3s^3 + 4s^2 + s)/2 \\ & + f_h(x_{k+2})(s^3 - s^2)/2 \end{aligned} \quad (14)$$

위 식에서 $f_h(x_k)$ 와 $\tilde{f}_h(y_n)$ 은 원본 데이터와 수평방향으로 크기 변환된 데이터이다. 그리고 y_n 은 재표본화 위치이다.

수직방향으로 크기변환 수행은 다음과 같이 수행된다.

$$\tilde{f}_v(y_n) = \tilde{f}_v(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_v} = n \cdot \frac{H}{\tilde{H}}}, n = 0, 1, 2, \dots, \tilde{H}-1 \quad (15)$$

$$\begin{aligned} \tilde{f}_v(x) = & f_v(x_{k-1})(-s^3 + 2s^2 - s)/2 \\ & + f_v(x_k)(3s^3 - 5s^2 + 2)/2 \\ & + f_v(x_{k+1})(-3s^3 + 4s^2 + s)/2 \\ & + f_v(x_{k+2})(s^3 - s^2)/2 \end{aligned} \quad (16)$$

위 식에서 $f_v(x_k)$, $\tilde{f}_v(y_n)$, y_n 은 원본 데이터와 수직방향으로 크기 변환된 데이터, 그리고 재표본화 위치이다.

4.2 비월주사 영상의 해상도조정

비월주사영상에 scaler를 적용할 때는 프레임 영상 $f(x_k)$ 을 even $f^e(x_k)$ 와 odd $f^o(x_k)$ field로 분리하고 나서 식 (13)과 (15)의 과정이 실행된다. 그림 8에서 $f_h^e(x_k)$, $f_v^e(x_k)$, $f_h^o(x_k)$, $f_v^o(x_k)$ 는 각각 even field와 odd field의 수평과 수직 데이터이다. field 데이터에 대해 수평 방향으로 재표본화의 수행은 다음과 같이 수행된다.

$$\tilde{f}_h^e(y_n) = \tilde{f}_h^e(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h}, n=0,1,2,\dots, \tilde{H}-1} \quad (17)$$

$$\tilde{f}_h^o(y_n) = \tilde{f}_h^o(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h}, n=0,1,2,\dots, \tilde{H}-1} \quad (18)$$

수직 방향 처리는 다음과 같이 정의된다.

$$\tilde{f}_v^e(y_n) = \tilde{f}_v^e(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_v}, n=0,1,2,\dots, \frac{\tilde{H}}{2}-1} \quad (19)$$

$$\tilde{f}_v^o(y_n) = \tilde{f}_v^o(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_v}, n=0,1,2,\dots, \frac{\tilde{H}}{2}-1} \quad (20)$$

각 field 신호를 크기 조정된 후, 그림 8에서처럼 합친다. 그리고 수직으로 합쳐진 데이터는 다음과 같이 된다.

$$\tilde{f}_v(m) = \begin{cases} \tilde{f}_v^e\left(\frac{n}{\delta_v}\right), n = \left\lfloor \frac{m}{2} \right\rfloor, \text{ if } m \text{ is even} \\ \tilde{f}_v^o\left(\frac{n}{\delta_v}\right), n = \left\lfloor \frac{m}{2} \right\rfloor, \text{ else if } m \text{ is odd} \end{cases} \quad (21)$$

식 (21)에서 $\tilde{f}_v^e(n/\delta_v)$ 에 대한 표본화 위치 $\{n/\delta_v, n=0,1,2,\dots, \tilde{H}/2-1\}$ 는 원본 프레임 데이터 $f_v(x_k)$ 의 표본화 위치 $\{2n/\delta_v, n=0,1,2,\dots, \tilde{H}/2-1\}$ 에 해당한다. 또한 $\tilde{f}_v^o(n/\delta_v)$ 에 대한 표본화 위치 $\{n/\delta_v, n=0,1,2,\dots, \tilde{H}/2-1\}$ 는 원본 프레임 데이터 $f_v(x_k)$ 에서 $\{2n/\delta_v, n=0,1,2,\dots, \tilde{H}/2-1\}$ 과 대응한다. 이것은 프레임 $f_v(x_k)$ 에서 재표본화 위치는 다음과 같음을 의미한다.

$$\left\{ \frac{0}{\delta_v}, 1, \frac{2}{\delta_v}, 1 + \frac{2}{\delta_v}, \frac{4}{\delta_v}, 1 + \frac{4}{\delta_v}, \dots, 1 + \frac{2}{\delta_v} \left(\frac{\tilde{H}}{2} - 1 \right) \right\}.$$

이 재표본화 위치들은 균등한 간격을 유지하고 있지 않다. 그러므로 수직방향의 데이터를 크기 조정할 때, 표본화 위치는 이런 불규칙성을 보정하기 위해 다음과 같이 수정되어야 한다.

$$\tilde{f}_v^e(y_n) = \tilde{f}_v^e(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_v}, n=0,1,2,\dots, \frac{\tilde{H}}{2}-1} \quad (22)$$

$$\tilde{f}_v^o(y_n) = \tilde{f}_v^o(x) \Big|_{x=y_n=\left(n \cdot \frac{1}{\delta_v} + d\right), n=0,1,2,\dots, \frac{\tilde{H}}{2}-1} \quad (23)$$

위 식에서 $(n \cdot 1/\delta_v + d)$ 는 수정된 표본화 위치를 의미한다. 원본 프레임의 수직 축 관점에서 볼 때, 식 (22)와 (23)의 표본화 위치는 $\{(2n \cdot 1/\delta_v)$ 와 $(1+2n \cdot 1/\delta_v + 2 \cdot d), n=0,1,2,\dots, \tilde{H}/2-1\}$ 이 된다. 표본화 위치 $\{(2n \cdot 1/\delta_v)$ 와 $(1+2n \cdot 1/\delta_v + 2d)\}$ 들은 $\{0, 1/\delta_v, 2/\delta_v, 3/\delta_v, 4/\delta_v, \dots, (\tilde{H}-1)/\delta_v\}$ 과 같이 프레임영상의 수직방향으로 균등 간격을 유지하도록 설정되어야 하기 때문에 d 는 다음과 같이 된다.

$$d = \frac{1}{2} \left(\frac{1}{\delta_v} - 1 \right) \quad (24)$$

그러므로 odd field의 수직방향에서 표본화간격은 식 (24)의 d 처럼 수정되어야 한다. 합쳐지는 과정에서 수평방향 보정은 필요 없기 때문에 수평 데이터는 다음과 같다.

$$\tilde{f}_h(y_n) = \begin{cases} \tilde{f}_h^e(y_n) \text{ is in the even line of the merged frame} \\ \tilde{f}_h^o(y_n) \text{ is in the odd line of the merged frame} \end{cases} \quad (25)$$

4.3 하나의 비월주사영상을 두 개의 순차주사영상으로 해상도조정

한 장의 비월주사영상을 두 개의 순차주사영상으로 크기조정 하기위해 scaler가 사용되는 과정을 그림 9에 표현하였다. 여기서는 병합과정이 사용되지 않는다. even field와 odd field에 적용되는 크기조정은 다음과 같다.

$$\tilde{f}_h^e(y_n) = \tilde{f}_h^e(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h}, n=0,1,2,\dots, \tilde{H}-1} \quad (26)$$

$$\tilde{f}_v^e(y_n) = \tilde{f}_v^e(x) \Big|_{x=y_n=n \cdot \frac{1}{2\delta_v}, n=0,1,2,\dots, \tilde{H}-1} \quad (27)$$

$$\tilde{f}_h^o(y_n) = \tilde{f}_h^o(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h}, n=0,1,2,\dots, \tilde{H}-1} \quad (28)$$

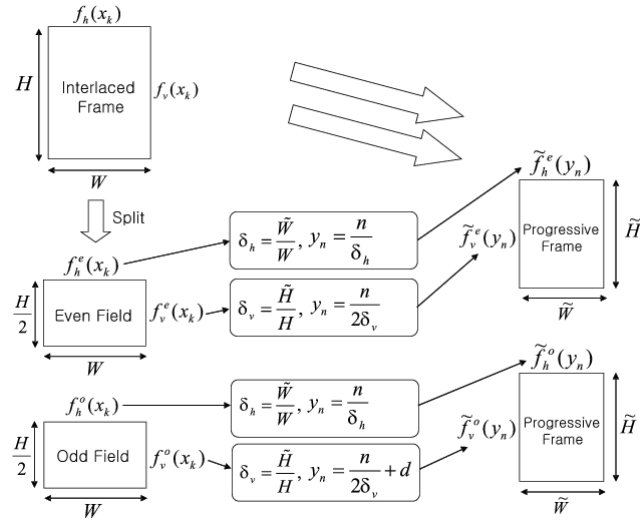


그림 9. 한 장의 비월주사영상을 두 장의 순차주사영상으로 해상도 변환하는 과정

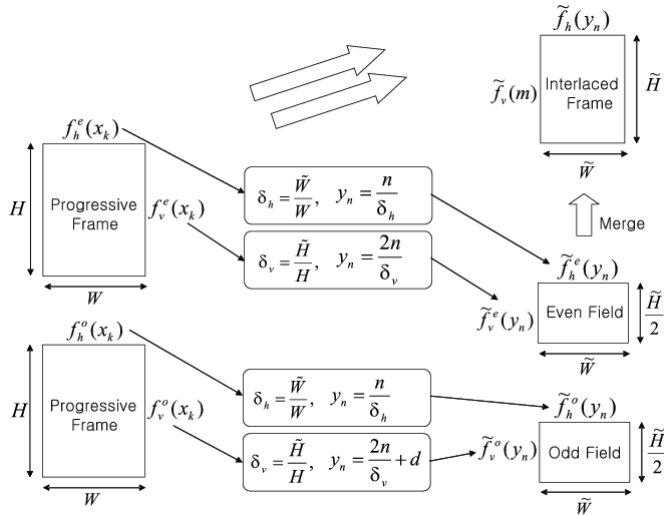


그림 10. 두 장의 순차주사영상을 한 장의 비월주사영상으로 해상도 변환하는 과정

$$\tilde{f}_v^o(y_n) = \tilde{f}_v^e(x_k) \Big|_{x=y_n = \left(n \cdot \frac{1}{2\delta_v} + d\right)}, n=0,1,2,\dots,\tilde{H}-1 \quad (29)$$

다음과 같이 정해진다.

$$d = -\frac{1}{2} \quad (30)$$

식 (27)과 (29)의 각각의 field 데이터에서 표본화 위치는 원본 프레임 $f_v(x_k)$ 의 수직 축 관점에서 $\{(n \cdot 1/\delta_v), n=0,1,2,\dots,\tilde{H}-1\}$ 과 $\{(1+n \cdot 1/\delta_v + 2 \cdot d), n=0,1,2,\dots,\tilde{H}-1\}$ 에 대응된다. 이런 과정에서는 병합처리가 사용되지 않기 때문에, $\{(n \cdot 1/\delta_v)\}$ 와 $\{(1+n \cdot 1/\delta_v + 2 \cdot d)\}$ 는 $\{0, 1/\delta_v, 2/\delta_v, 3/\delta_v, 4/\delta_v, 5/\delta_v, \dots, (\tilde{H}-1)/\delta_v\}$ 가 되어야만 한다. 그러므로 d 는

4.4 두 개의 순차주사영상을 하나의 비월주사영상으로 해상도조정

그림 10에서처럼 두 개의 순차주사 프레임을 하나의 비월주사 프레임으로 크기 조정하는 과정에서는 splitter가 필요없다. 이런 경우 신호 처리과정은 다음 식과 같이 요약된다.

$$\tilde{f}_h^c(y_n) = \tilde{f}_h^c(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h}, n=0,1,2,\dots, \tilde{W}-1} \quad (31)$$

미지 크기 조정과정의 전체 복잡도는 다음과 같다.

$$\tilde{W} \times \tilde{H} \times \{P+A\} \quad (36)$$

$$\tilde{f}_v^c(y_n) = \tilde{f}_v^c(x) \Big|_{x=y_n=2n \cdot \frac{1}{\delta_v}, n=0,1,2,\dots, \frac{\tilde{H}}{2}-1} \quad (32)$$

이는 기존의 방법보다 제안한 방식이 $\tilde{W} \times \tilde{H}$ 만큼의 더하기 연산을 추가로 요구하는 것을 의미한다. 더하기 연산(A)는 곱하기 연산(P)보다 매우 짧은 CPU time을 소비하기 때문에 증가된 복잡도는 별로 크지 않다.

$$\tilde{f}_h^o(y_n) = \tilde{f}_h^o(x) \Big|_{x=y_n=n \cdot \frac{1}{\delta_h}, n=0,1,2,\dots, \tilde{W}-1} \quad (33)$$

V. 실험 결과

$$\tilde{f}_v^o(y_n) = \tilde{f}_v^o(x) \Big|_{x=y_n=(2n \cdot \frac{1}{\delta_v} + d), n=0,1,2,\dots, \frac{\tilde{H}}{2}-1} \quad (34)$$

제안한 알고리즘의 성능을 평가하기 위하여 실제 영상을 가지고 시뮬레이션을 수행하였다. 실험에 사용한 영상은 mobile and calendar, flower garden과 susie이다. 첫 번째 시뮬레이션은 객관적 화질측정 실험이다. 제안한 알고리즘의 객관적인 평가를 위해 연속적인 해상도변경에 따른 정보 손실의 양을 확인하였다. 정보 손실을 측정하기 위해 Peak 신호대 잡음률(PSNR), 신호대 잡음률(SNR), 평균 제곱 오차(MSE) 등을 고려할 수 있다. 이것들 중 PSNR과 MSE는 기존의 여러 논문에서 사용되고 있다^[14,24,25]. 따라서 본 논문에서는 정보 손실의 측정을 위해 PSNR과 MSE를 사용하였다. 실험영상은 저대역·통과필터(LPF)를 통과한 후 기존의 방식과 제안한 방식을 각각 사용하여 0.5배 크기로 줄여준다. 그 다음 원래 영상의 크기로 돌아가기 위해 $\delta=2$ 로 설정하여 행방향과 열방향으로 각각의 보간법을 이용한다. 표 2에서는 mobile and calendar, flower garden 과 susie 영상을 대상으로 축소후 확대된 영상의 원본영상대비 PSNR과 MSE를 측정하였다. 표에서 보는 바와 같이, 제안한 방식을 이용한 영상의 PSNR이 기존의 방식을 이용한 영상의 PSNR보다 훨씬 높다. 또한 이 표에서는 실험영상을 비정수배 크기인 490×350 로 감소시키는 다른 실험의 결과를 보여주고 있다. 이 경우에서 역시 제안한 방식이 기

식 (32)와 (34)의 표본화 위치 $\{(2n \cdot 1/\delta_v) \text{와 } (2n \cdot 1/\delta_v + d), n=0,1,2,\dots, \tilde{H}/2-1\}$ 는 원본 순차주사 프레임의 수직 축의 관점에서 표본화 위치 $\{0, 1/\delta_v, 2/\delta_v, 3/\delta_v, 4/\delta_v, 5/\delta_v, \dots, (\tilde{H}-1)/\delta_v\}$ 를 구성해야 한다. 그러므로 d 는 다음과 같이 되어야 한다.

$$d = \frac{1}{\delta_v} \quad (35)$$

4.5 알고리즘의 계산 복잡도

기존의 방법과 제안 한 방법의 계산적 부담이 되는 요인은 식 (11)과 (23)에 표현되어있는 표본화 위치 y_n 의 결정, 식 (12)에 있는 s의 계산, 식 (9)에 있는 필터 kernel의 생성과 FIR 필터링의 수행이 있다. 값 $\tilde{f}(y_n)$ 을 얻기 위해서 한번은 y_n 의 계산 단계에서 식 (11)과 같이 기존의 방식으로 수행을 하고, 다른 한번은 식 (23)의 제안한 방식으로 수행한다. y_n 을 얻는 과정을 제외하고, s를 계산하거나, filter kernel을 만들거나, filtering을 수행하는 등의 다른 단계의 복잡도는 제안한 방식과 기존의 방식에서 같다. 계산 복잡도는 표 1에 요약되어 있다. 크기 조정된 영상의 크기를 $\tilde{W} \times \tilde{H}$ 라 가정한다면, 이

표 1. 식 (9), (11), (12), (22), (23)을 이용하여 샘플을 얻기 위해 필요한 계산 복잡도. P는 곱하기 연산의 횟수, A는 더하기 연산의 횟수를 나타낸다.

단계	식 (9), (11), (12) 기존의 방식	식 (9), (12), (22), (23) 제안한 방식
식 (11)과 (23)에 있는 y_n 의 계산	P=1	P=1, A=1
식 (12)에 있는 s의 계산	A=1	A=1
식 (9)에 있는 필터 kernel 만들기	P=3, A=2	P=3, A=2
식 (9)에 있는 FIR 필터링	P=4, A=3	P=4, A=3
전체 복잡도	P=8, A=6	P=8, A=7

표 2. 제안하는 알고리즘의 정보 손실 관점에서 성능 비교 PSNR의 증가는 정보 손실의 감소를 의미한다.

Test image	Scheme	704x480i ⇒ 352x240i ⇒ 740x480i	704x480i ⇒ 490x350i ⇒ 704x480i
Mobile and calendar	Conventional scheme	PSNR=19.77dB	PSNR=22.16dB
		MSE=684.19	MSE=394.69
	Proposed scheme	PSNR=20.65dB	PSNR=23.61dB
		MSE=559.40	MSE=282.89
Flower garden	Conventional scheme	PSNR=21.02dB	PSNR=23.82dB
		MSE=513.55	MSE=269.73
	Proposed scheme	PSNR=21.69dB	PSNR=24.67dB
		MSE=440.04	MSE=221.71
Susie	Conventional scheme	PSNR=34.19dB	PSNR=37.24dB
		MSE=24.76	MSE=12.28
	Proposed scheme	PSNR=35.30dB	PSNR=38.35dB
		MSE=19.19	MSE=9.50

존의 방식보다 성능이 우수한 것을 관찰할 수 있다. 이는 기존의 방식은 재샘플링 위치 관계를 고려하지 않은 반면, 제안한 보간법은 크기조정된 영상과 원본 영상의 샘플링 위치 사이 관계를 고려하였기 때문이다. 이런 결과들로부터 제안한 방식이 영상 해상도를 변환할 때 매우 효율적이라는 것을 확인할 수 있다.

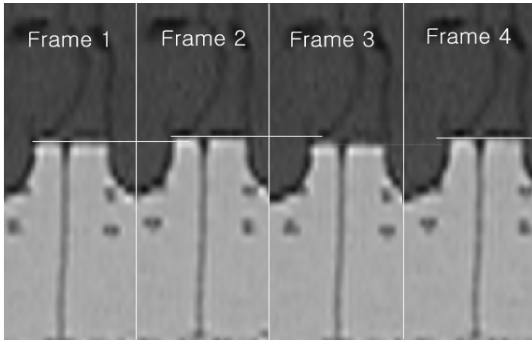


그림 11. 기존의 방식을 이용하여 해상도를 조정된 연속하는 네 장의 이미지. 해상도 변환영상들의 수직적위치가 일정하지 않다.

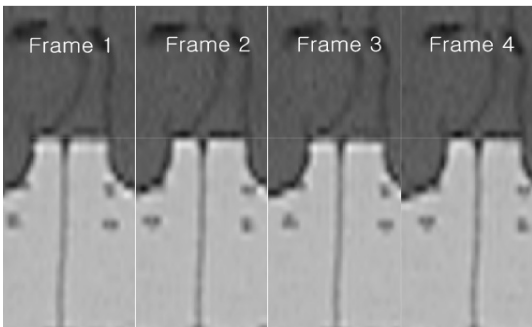
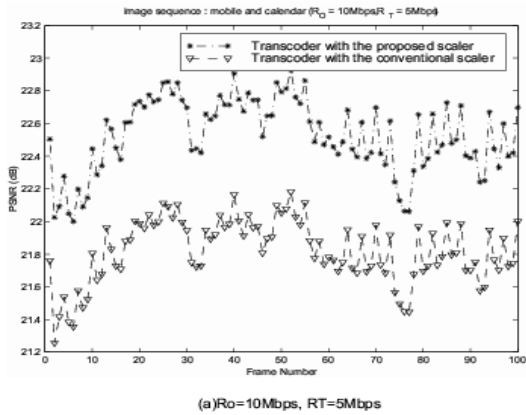


그림 12. 제안한 방식을 이용하여 해상도 조정된 연속하는 네 장의 이미지. 해상도 변환영상들의 수직적위치가 일정하다.

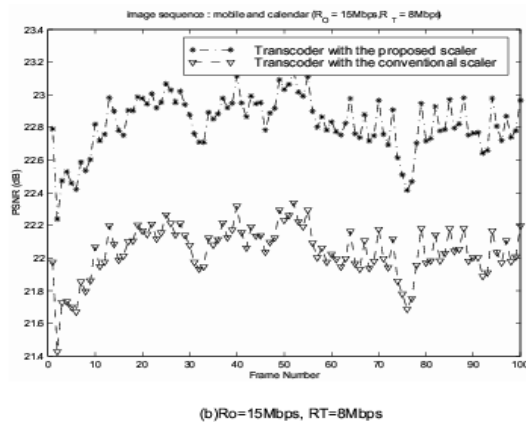
제안한 방식의 주관적인 평가를 보여주기 위한 추가 실험 결과를 그림 11과 12에 나타냈다. 이 실험에서 해상도 확대 비율은 5.7로 하였고, 결과 영상들의 포맷은 순차주사방식이다. 그림은 mobile and calendar 영상에서 울타리와 말의 코 부분을 5.7배 확대한 결과를 나타낸다. 결과 영상의 비교를 위해 네 장의 연속되는 영상을 하나의 영상으로 합쳐서 표시했다. 그림에서 차이를 강조하기 위해서 흰색 수평선을 추가로 표시하였다. 기존 방법으로 크기조정된 결과 영상의 수평선은 프레임간 수직적으로 불규칙적인 반면, 제안한 방식은 수직적으로 변동없는 영상을 만들었다.

Transcoding의 측면에서 제안한 방식의 유용성을 확인하기 위해 그림 1에 설명된 방식으로 실험하였다. 시뮬레이션에 사용된 codec은 MPEG-2 MP@ML이다. 원본 영상은 해상도는 740×480i이며, 비트율 R_0 (=15Mbps 또는 10Mbps)로 부호화 되었다. Transcoder에서 주어진 비트스트림은 복호화되고 나서 480×336i로 공간 해상도를 조정한다. 그리고 크기조정된 영상을 비트율을 R_1 (=10Mbps 또는 5Mbps)로 transcoding 한다. 이 실험은 그림 3에서 설명된 i frame을 다른 i frame으로 변환하는 과정이다. PSNR에 의해 화질을 평가하기 위해 원래 영상 크기인 704×480i로 해상도를 복원한 후 화질을 측정했다. 그림 13과 14는 각각 mobile and calendar와 flower garden의 시뮬레이션 결과를 나타낸다. 그림 13과 14는 제안한 방식이 기존의 방식보다 평균적으로 0.7과 0.8dB 높은 PSNR을 나타냄을 보여준다.

그림 4와 5에 설명되어 있는 방식으로 transcoder의 성능을 평가하기 위해, 우리는 이 두 방식(비월 주사영상을 순차주사영상으로 변환하는 그림 4의



(a)Ro=10Mbps, RT=5Mbps

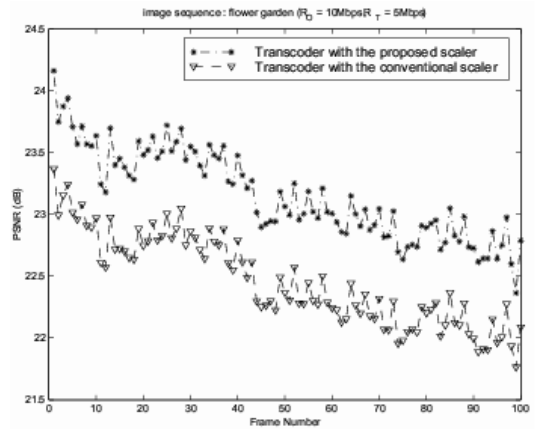


(b)Ro=15Mbps, RT=8Mbps

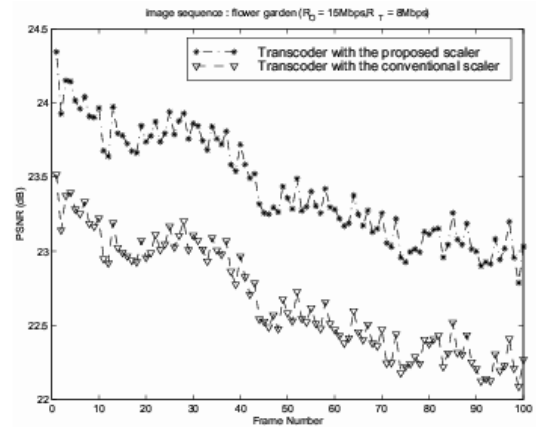
그림 13. Transcoder에서 기존의 해상도 변환 방식과 제안한 방식을 이용한 결과 영상의 PSNR비교 테스트 영상은 mobile ad calendar이고, 변환 기법은 cubic convolution을 사용하였다

과정과 순차주사영상을 비월주사영상으로 변환하는 그림 5의 과정을 연속적으로 결합한다. 원본 영상의 해상도는 $740 \times 480i$ 이며, 비트율을 $R_0=15Mbps$ 로 압축되어있다. 부호화된 비트스트림을 비트율을 $R_1=10Mbps$ 이고 $480 \times 360p$ 크기인 신호로 transcoding 한다. 그 transcoding된 영상($480 \times 360p$)을 비트율 $R_2=5Mbps$ 이고 $704 \times 480i$ 인 영상으로 다시 transcoding한다. 이 실험결과를 그림 15에 나타냈다. 제안한 방식을 사용한 transcoder의 성능은 기존의 방식과 비교하였을 때 평균적으로 1.2dB 높은 PSNR을 나타낸다.

제안한 샘플링 위치 보정 방식이 cubic convolution scaler뿐만 아니라 B-spline 기법에도 적용될 수 있음을 보이기 위해 B-spline기법을 이용한 transcoder의 성능을 그림 16에 나타냈다. 다른 시뮬레이션 조건들은 그림 13과 같게 설정되고 공간 해상도



(a)Ro=10Mbps, RT=5Mbps



(b)Ro=15Mbps, RT=8Mbps

그림 14. Transcoder에서 기존의 해상도 변환방식과 제안한 방식을 이용한 결과영상의 PSNR 비교 테스트 영상은 flower garden이고, 변환기법은 cubic convolution기법을 사용하였다

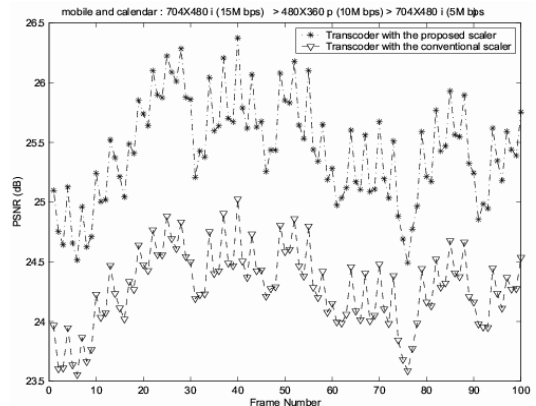


그림 15. 복합적인 transcoder에서 기존의 해상도 변환방식과 제안한 방식을 이용한 결과 영상의 PSNR 비교 테스트 영상은 mobile and calendar이고, 변환 기법은 cubic convolution을 사용하였다

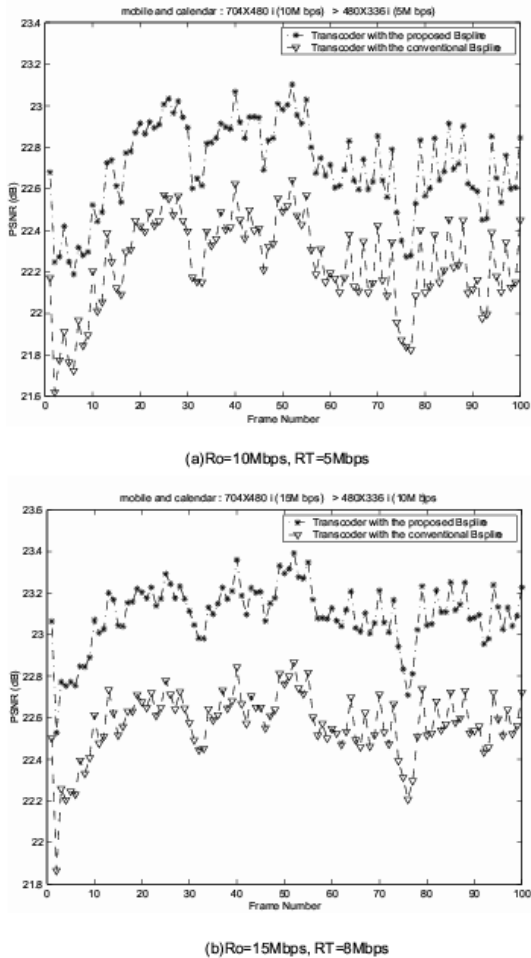


그림 16. Transcoder에서 기존의 해상도 변환방식과 제안한 방식을 이용한 결과 영상의 PSNR비교 테스트 영상은 mobile and calendar이고, 변환 기법은 B-spline을 사용하였다

조정을 위해 B-spline기법이 사용된다. 시뮬레이션 결과는 제안한 방식이 기존의 방식보다 좋은 화질을 갖는 비트스트림을 발생시키는 것을 나타낸다. 이는 비트율을 변화시키는 동시에 공간 해상도를 변화시키는 transcoding 방식에서 제안하는 방식이 유용하다는 것을 의미한다.

VI. 결론

본 논문에서는 MPEG 표준에서 고려되는 다양한 영상포맷의 크기를 조정하기 위해 개선된 샘플링 위치 보정 기법을 제안 하였다. 제안한 방식은 복원된 영상을 고화질로 유지하기위해 transcoding에서

사용될 수 있다. 실험 결과 제안한 방식이 기존의 방식보다 성능이 뛰어남을 보여준다. 결과를 통해서 알 수 있듯이, 제안한 알고리즘은 임의의 크기조정이 가능한 방식이다.

참 고 문 헌

- [1] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," IEEE Trans. Multimedia 2, 101-110 (June 2000).
- [2] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion vector resampling for compress video downscaling," IEEE Trans. Circuits Syst. Video Technol. 9, 929-936 (Sep. 1999).
- [3] J. Youn, M. T. Sun, and C. W. Lin, "Motion vector refinement for high performance transcoding," IEEE Trans. Multimedia 1, 30-40 (Mar. 1995).
- [4] C. Yim and M. A. Isnardi, "An efficient method for DCT-domain image resizing with mixed field/frame-mode macroblocks," IEEE Trans. Circuits Syst. Video Technol. 9, 696 - 700 (Aug. 1999).
- [5] J. Song and B. L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," IEEE Trans. Circuits Syst. Video Technol. 10, 767 - 775 (Aug. 2000).
- [6] N. Merhav, "Multiplication-free approximation algorithms for compressed-domain linear operations on images," IEEE Trans. Image Process. 8, 247 - 254 (Feb. 1999).
- [7] S. Liu and A. C. Bovik, "Local bandwidth constrained fast inverse motion compensation for DCT-domain video transcoding," IEEE Trans. Circuits Syst. Video Technol. 12, 309 - 319 (May 2002).
- [8] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," IEEE Trans. Image Process. 4, 247 - 258 (Mar. 1995).
- [9] W. K. Pratt, Digital Image Processing, John Wiley and Sons, New York (1991).

- [10] R. G. Keys, "Cubic convolution interpolation for digital image rocessing," IEEE Trans. Acoust., Speech, Signal Process. 29, 1153 - 1160 (Dec. 1981).
- [11] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," IEEE Trans. Acoust., Speech, Signal Process. 26, 508 - 517 (1978).
- [12] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," IEEE Trans. Pattern Anal. Mach. Intell. 13, 277 - 285 (Mar. 1991).
- [13] S. K. Park and R. A. Schowengerdt, "Image reconstruction by arametric cubic convolution," Comput. Vis. Graph. Image Process. 23, 258 - 272 (Sep. 1983).
- [14] G. Ramponi, "Warped distance for space-variant linear image nterpolation," IEEE Trans. Image Process. 8, 629 - 639 (May 1999).
- [15] H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG ompressed bitstream scaling," IEEE Trans. Circuits Syst. Video Technol. 6, 191 - 199 (Apr. 1999).
- [16] G. D. L. Reyes, A. R. Reibman, S. F. Chang, and J. C. I. Chuang, "Error resilient transcoding for video over wireless channels," IEEE J. elected Area Commun. 18, 1063 - 1074 (June 2000).
- [17] ISO/IEC and ITU-T, "Information technology-Generic coding of moving pictures and associated audio: Video," ISO/IEC/JTC1/SC29/WG11, N0801, Apr. 1995.
- [18] H. Hwang, "Interlaced to progressive scan conversion for HD-MAC application," IEEE Trans. Consum. Electron. 38(3), 151 - 156 (Aug. 1992).
- [19] M. Karlsson, P. Pohjala, H. Rantanen, and S. Kalli, "Evaluation of scanning rate up conversion algorithms; subjective testing of interlaced to progressive conversion," IEEE Trans. Consum. Electron. 38(3), 162 - 167 (Aug. 1992).
- [20] 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. Consum. Electron. 40(2), 119 - 129 (May 1994).
- [21] R. Li, N. K. Chung, K. T. Mo, D. M. Fisher, and V. Wong, "A lexible display module for DVD and set-up box applications," IEEE Trans.Consum. Electron. 43(3), 496 - 503 (Aug. 1997).
- [22] S. S. Rifman, "Digital rectification of ERTS multispectral imagery," Proc. Symp. Significant Results Obtained from ERTS-1 (NASA SP-327) I (Sec. B), 1131 - 1142 (1973).
- [23] R. Bernstein, "Digital image processing of Earth observation sensor data," IBM J. Res. Dev. 20, 40 - 57 (1976).
- [24] M. Karczewicz and M. Gabbouj, "Robust B-spline image modeling with application to image processing," EEE Trans. Image Process. 7, 912 - 917 (June 1998).
- [25] H. J. Kim and C. C. Li, "Lossless and lossy image compression using biorthogonal wavelet transforms with multiplierless operations," IEEE Trans. Circuits Syst. II: Analog digital signal Process 45, 1113 - 1118 (Au g. 1998).

유 영 조 (Young-Joe Yoo)

정회원



2002년 3월 세종대학교 정보통신공학과 입학
 현재 세종대학교 정보통신 공학과 공학사 재학
 <관심분야> H.264, Transcoder

서 주 헌 (Ju-Heon Seo)

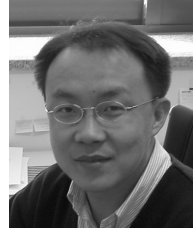
정회원



2006년 2월 세종대학교 정보통신공학과 공학사
현재 세종대학교 정보통신공학과 공학석사 재학
<관심분야> H.264, Transcoder

한 중 기 (Jong-Ki Han)

중신회원



1992년 2월 KAIST 공학 학사
1994년 2월 KAIST 공학 석사
1999년 2월 KAIST 공학 박사
1999년 3월~2001년 8월 삼성전자DM 연구소 책임 연구원
2001년 9월~현재 세종대학교 정보통신공학과 부교수

<관심분야> H.264, Transcoder, Scalable Video Codec