ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 23-03-13
84
j.inst.Korean.electr.electron.eng.Vol.27,No.3,296~302,September 2023
https : //doi.org/10.7471/ikeee.2023.27.3.296

스케일러 하드웨어 설계를 위한 조합 보간 알고리즘의 연구

A Study of the Combinatorial Interpolation Algorithm for Scaler Hardware Design

한 시 연*, 강 봉 순*★

Si-Yeon Han*, Bong-Soon Kang*★

Abstract

As Multimedia industry has evolved, it has become possible to display resolutions in various formats. Therefore, the performance of a scaler algorithm that converts resolutions while maintaining high quality and its hardware implementation are important. Considering the hardware design of an image up/down scaler, this paper proposes a combinatorial scaler algorithm that uses modified bilinear interpolation in the vertical direction and bicubic interpolation in the horizontal direction to reduce the line memory burden. Through quantitative and qualitative evaluations, this paper compared the performance of the proposed algorithm with three other well-known algorithms, and also compared the hardware burden of its hardware implementation. This paper used a sinusoidal signal and eight typical images for performance evaluation.

요 약

멀티미디어 산업이 발전함에 따라 다양한 형식의 해상도를 표시할 수 있게 되었다. 따라서 고화질을 유지하며 해상도를 변환하는 스케일러 알고리즘의 성능과 이를 하드웨어로 구현하는 것은 중요하다고 할 수 있다. 본 논문에서는 이미지 확대/축소 스케일러의 하드웨어 설계를 고려하여 수직 방향으로는 수정된 양 선형 보간, 수평 방향으로는 양 3차 회선 보간을 사용하여 라인 메모리 부담을 줄인 조합 스케일러 알고리즘을 제안한다. 본 논문은 정량적 그리고 정성적 평가를 통해 제안하는 알고리즘의 성능을 널리 사용되는 다른 세 가지 알고리즘과 비교 평가하였고, 이를 하드웨어로 구현할 때에 필요한 하드웨어 부담을 비교하였다. 본 논문은 성능 평가를 위해 정현파 신호와 8개의 일반 이미지를 사용하였다.

Key words : scaler, bilinear, bicubic, combinatorial, line memory

Ⅰ. 서론

멀티미디어 산업의 발달로 TV, 스마트폰, 태블릿 PC 등 다양한 크기의 디스플레이 장치에서 Full HD, 4K Ultra HD 등 다양한 형식의 해상도를 표시할 수 있게 되었다. 이에 따라 고화질을 유지하면서 해상도를 변환(축소 또는 확대)할 수 있는 스케일러(scaler) 알고리즘의 성능이 중요해졌다.

스케일러 알고리즘의 대표적인 방법에는 nearest neighbor interpolation (NNI)[1], bilinear interpolation

* Dept. of Electronics Engineering, Dong-A University
★ Corresponding author
Email : bongsoon@dau.ac.kr, Tel : +25-51-200-7703
※ Acknowledgment
This paper was supported by research funds from Dong-A University.
Manuscript received Aug. 31, 2023; revised Sep. 22, 2023; accepted Sep. 24, 2023.

(BLI)[2], bicubic interpolation (BCI)[3]이 있다. NNI는 출력 해상도에 따라 새로 생성될 픽셀 위치에서 가장 가까운 픽셀의 정보를 복사하는 방법이다. 이 방법은 가장 간단한 스케일러 알고리즘이지만 대각선 방향으로 계단파 노이즈가 발생하기 때문에 성능이 가장 떨어진다고 할 수 있다. BLI는 보간할 위치와 인접한 두 픽셀 사이의 거리 차이에 해당되는 선형적인 가중치를 곱하여 새로운 보간 픽셀값을 계산한다. 이 방법은 두 픽셀 정보를 사용하므로 NNI 보다 알고리즘 성능이 더 우수하다고 할 수 있다. BCI는 수평(X축) 또는 수직(Y축) 방향의 연속된 픽셀값 4개로 구성된 3차 방정식을 이용하여 보간할 위치의 새로운 픽셀값을 계산하는 방법이다. 이 방법은 3차 방정식을 사용하기 때문에 알고리즘 성능이 가장 우수하지만, 연산량이 가장 많다는 단점이 있다. 이러한 방법들을 응용하여 스케일러 알고리즘 연구 및 하드웨어 구현에 관한 많은 연구가 진행되었다[4]-[6].

본 논문에서는 스케일러 알고리즘을 하드웨어로 구현할 때 복잡도가 매우 높은 라인 메모리 사용을 최소화하면서 BLI 보다도 성능이 우수한 방법을 제안하고자 한다. 앞에서 설명한 것처럼 BCI 알고리즘은 성능이 가장 우수하지만, 수직 방향으로 3차 방정식을 적용하려면 3개의 라인 메모리(line memory)가 필요하다. 따라서 RGB 3개 채널을 모두 구현하려면 총 9개의 라인 메모리가 필요하다. 본 논문에서는 라인 메모리 수를 줄이기 위해 수직 방향은 BLI, 수평 방향은 BCI을 적용한 combinatorial interpolation(CI) 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 스케일러 알고리즘과 제안하는 CI 알고리즘의 원리와 수식을 소개한다. 3장에서는 이들 알고리즘에 대한 정량적, 정성적 평가와 더불어 선행 연구와의 비교를 진행한다. 마지막으로 4장에서는 본 논문의 결론을 서술한다.

II. 본론

1. Modified Bilinear Interpolation

일반적인 BLI는 그림 1과 같이 X, Y축 방향으로 각각 인접한 픽셀 2개를 이용한다[2]. 입력 영상에서 현재 픽셀 위치 (x_0, y_0)와 이에 해당하는 픽셀값을 $P(x_0, y_0)$라 가정하자. 그럼, 각 축의 현재 픽셀 위치와 인접한 픽셀 위치를 x_1과 y_1으로 표시할 수 있다.

그림 1에서 보간 하려는 위치를 (x,y)라 가정하면 픽셀값 $P(x,y)$는 수식 1~3과 같이 Y축 2번과 X축 1번의 총 3번 선형 보간으로 구할 수 있다.

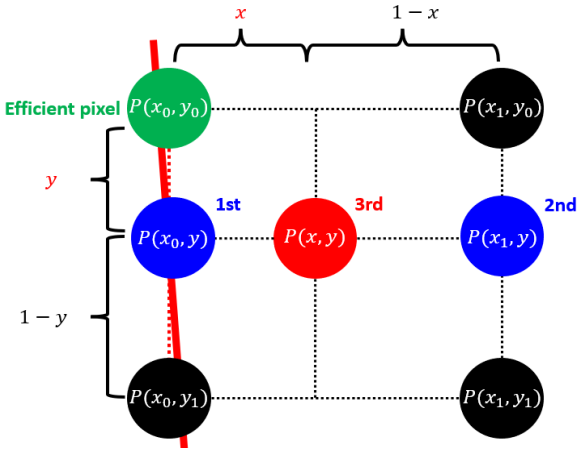

Fig. 1. Bilinear interpolation method.
그림 1. 양 선형 보간 방법

$$P(x_0,y) = P(x_0,y_0)\times(1-y)+P(x_0,y_1)\times y \quad (1)$$

$$P(x_1,y) = P(x_1,y_0)\times(1-y)+P(x_1,y_1)\times y \quad (2)$$

$$P(x,y) = P(x_0,y)\times(1-x)+P(x_1,y)\times x \quad (3)$$

수식 1에서 $P(x_0,y_1)$은 현재 픽셀 값 $P(x_0,y_0)$의 Y축으로 인접한 픽셀값이다. 이 두 픽셀값에 거리 비례 법칙을 적용하면, 첫 번째 Y축 보간 픽셀값 $P(x_0,y)$를 구할 수 있다. 이의 과정을 오른쪽으로 한 픽셀 이동하여 적용하면, 두 번째 Y축 보간 픽셀값 $P(x_1,y)$를 수식 2와 같이 구할 수 있다. 마지막으로 수식 1~2에서 계산한 두 개의 Y축 보간 픽셀에 비례 법칙을 적용하면, 보간하려는 픽셀값 $P(x,y)$를 수식 3과 같이 구할 수 있다.

위의 과정을 살펴보면, 보간 계산을 위해서는 현재 유효한 픽셀 위치와 보간 하려는 위치의 총 2가지 값이 필요한 것을 볼 수 있다. 2가지 값을 계산하려면 X, Y축 방향으로 확대 및 축소비율에 따른 증가값인 $xinc(nx)$, $yinc(ny)$가 필요한데 수식 4~5와 같이 표시할 수 있다. 각 증가 값의 초깃값 $xinc(0)$, $yinc(0)$는 0이다.

$$xinc(nx) = \frac{1}{x\ scale\ ratio} + xinc(nx-1) \quad (4)$$

$$yinc(ny) = \frac{1}{y\ scale\ ratio} + yinc(ny-1) \quad (5)$$

여기에서 $x\,scale\,ratio$와 $y\,scale\,ratio$는 독립적인 확대 및 축소비율이며, 대칭 또는 비대칭으로 설정 할 수 있다. $nx = 0, 1, \cdots, floor\{x\,scale\,ratio \times xsize\}$이고 $ny = 0, 1, \cdots, floor\{y\,scale\,ratio \times ysize\}$로 표시할 수 있다. 여기서 $floor(\cdot)$ 함수는 소수점을 제거하는 연산을 수행하고 $xsize$와 $ysize$는 입력 영상의 크기이기에 nx와 ny는 최종 확대 및 축소비율에 따른 결과 영상의 크기에 해당하

는 값이다. 수식 4~5에서 계산된 X, Y축 방향으로의 증가 값은 정수부와 소수부로 구성되어 있다. 그럼, 정수부의 값은 보간 계산에 사용할 유효한 픽셀 위치 (x_{nx}, y_{nx})가 되고 소수부는 보간하려는 위치 (x, y)가 되는데, 수식 6~7과 같이 구할 수 있다.

$$x_{nx} = floor\{xinc(nx)\},\ x = xinc(nx) - x_{nx} \quad (6)$$

$$y_{ny} = floor\{yinc(ny)\},\ y = yinc(ny) - y_{ny} \quad (7)$$

하지만 앞에서 가정한 수식 4~5의 초깃값 0을 수식 6~7에 적용하여 확대 기능을 수행하면, 마지막 픽셀 BLI 처리에서 불연속이 발생할 수 있다. 이를 개선하기 위하여 수식 8~9의 초깃값을 MATLAB 2023a에서 다양한 보간 방법을 이용해 이미지의 크기 변환 기능을 제공하는 내부 함수인 imresize()[7]에 사용된 것과 같이 사용한다. 이 초깃값을 사용하면 기존의 불연속 문제를 개선할 수 있었는데, 이 논문에서는 이를 modified bilinear interpolation (mBLI)이라고 부르겠다.

$$xinc(0) = \frac{1}{x\,scale\,ratio} + 0.5 \times (1 - \frac{1}{x\,scale\,ratio}) \quad (8)$$

$$yinc(0) = \frac{1}{y\,scale\,ratio} + 0.5 \times (1 - \frac{1}{y\,scale\,ratio}) \quad (9)$$

2. Bicubic Interpolation

BCI는 그림 2와 같이 X, Y축으로 각각 인접한 픽셀 4개를 이용한다[3]. BCI에서는 수식 4~5의 초깃값 0을 사용한다. 원 영상의 유효한 현재 픽셀 위치가 (x_0, y_0)라 가정하자. 각 축의 위치에 -1, +1, +2를 더해 준 값을 사용하면, 그림 2와 같이, X축의 경우 $P(x_{-1}, y_0)$, $P(x_0, y_0)$, $P(x_1, y_0)$, $P(x_2, y_0)$의 4개 픽셀값으로 표시할 수 있다.

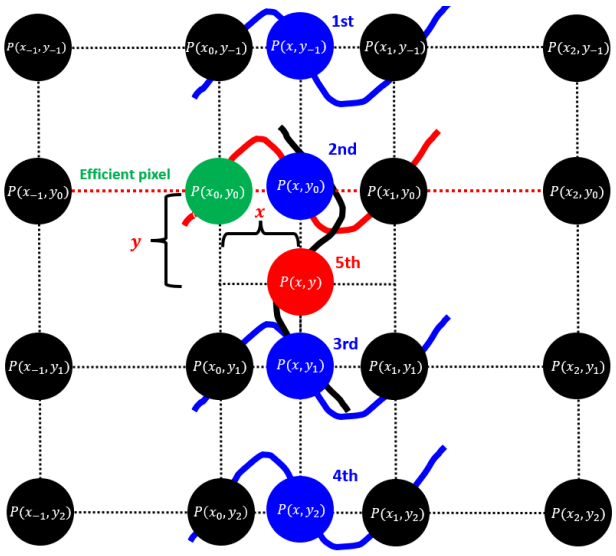

Fig. 2. Bicubic interpolation method.
그림 2. 양 3차 회선 보간 방법

이 픽셀값들을 사용하여 X축의 보간 하려는 위치 (x, y_0)의 픽셀값 $P(x, y_0)$를 구하기 위해 수식 10의 3차 방정식과 이의 1차 미분한 수식 11을 사용한다. a, b, c, d는 각 차수의 계수값으로, 양 3차 회선 보간을 위해 계산되어야 한다. $x_0 = 0$이라 가정하면 인접한 위치는 $x_1 = 1$이 되고, 수식 10~11에 각각 대입하면 $P(x_0, y_0) = d$, $P(x_1, y_0) = a + b + c + d$, $P'(x_0, y_0) = c$, $P'(x_1, y_0) = 3a + b + c$와 같은 새로운 수식을 구할 수 있다. 이를 a, b, c, d의 4개 계수에 대해 다시 정리하여 수식 12를 구한다.

$$P(x, y_0) = ax^3 + bx^2 + cx + d \quad (10)$$

$$P'(x, y_0) = 3ax^2 + bx + c \quad (11)$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(x_0, y_0) \\ P(x_1, y_0) \\ P'(x_0, y_0) \\ P'(x_1, y_0) \end{bmatrix} \quad (12)$$

수식 12에서 $P(x_0, y_0)$와 $P(x_1, y_0)$는 입력 이미지에서 각 위치에 해당하는 픽셀값이다. 수식 12에 있는 1차 미분값은 이전 픽셀값과 다음 픽셀값 사이의 기울기로 수식 13~14와 같이 계산할 수 있다. 이를 수식 12에 대입한 후, 계수 a, b, c, d를 다시 정리하면 수식 15를 구할 수 있다. 수식 15에서 볼 수 있듯이, $P(x_{-1}, y_0)$, $P(x_0, y_0)$, $P(x_1, y_0)$, $P(x_2, y_0)$의 픽셀값들은 입력 이미지에서 X축으로 인접한 4개 픽셀값이다. 따라서 X축의 3차 방정식의 계수를 구할 수 있고, 이를 수식 10에 적용하면 $P(x, y_0)$를 구할 수 있다. 위의 과정을 Y축에 그대로 적용하면, 그림 2에서 볼 수 있듯이, BCI를 사용한 최종 결괏값 $P(x, y)$를 구할 수 있다.

$$P'(x_0, y_0) = \frac{P(x_1, y_0) - P(x_{-1}, y_0)}{2} \quad (13)$$

$$P'(x_1, y_0) = \frac{P(x_2, y_0) - P(x_0, y_0)}{2} \quad (14)$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} -0.5 & 1.75 & -1.75 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(x_{-1}, y_0) \\ P(x_0, y_0) \\ P(x_1, y_0) \\ P(x_2, y_0) \end{bmatrix} \quad (15)$$

3. Combinatorial Interpolation

그림 1에서 볼 수 있듯이, mBLI은 X축, Y축 방향으로 인접한 픽셀 2개를 사용한다. X축으로는 하드웨어 구현에서 매우 간단한 플립플롭 소자를 사용하면 되고, Y축

으로는 1개의 라인 메모리가 필요하다. 반면 BCI에서도 X축으로는 플립플롭을 사용하면 되지만, Y축으로는 3개의 라인 메모리가 필요하여 하드웨어 크기가 매우 커지는 부담이 있다. 이를 개선하기 위하여, 본 논문에서는 Y축에는 1개의 라인 메모리가 필요한 mBLI를 적용하고, X축에는 간단한 플립플롭을 사용하는 BCI를 조합한 CI를 그림 3과 같이 제안하여 사용하고자 한다. 따라서 CI의 성능은 mBLI보다도 우수하고 BCI 보다는 다소 떨어진다. 하지만 RGB 3개 채널의 BCI 구현에서 부담이 매우 큰 라인 메모리 수를 총 9개에서 3개로 줄여서 하드웨어가 상대적으로 간단한 알고리즘이라고 할 수 있다. 따라서 CI는 Y축으로 라인 메모리를 줄일 수 있는 mBLI 수식 1을 사용한다. 그리고 X축으로 BCI의 수식 15를 사용하면 원하는 위치의 픽셀값 $P(x,y)$을 구할 수 있다.

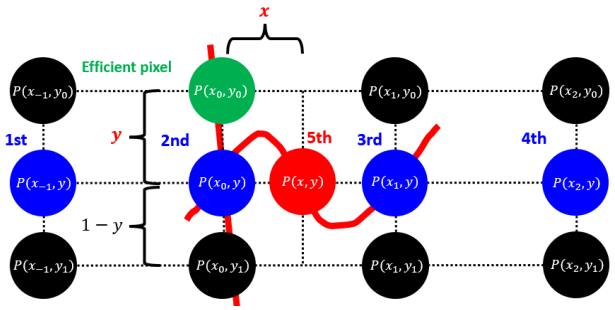

Fig. 3. Proposed combinatorial interpolation algorithm.
그림 3. 제안하는 조합 보간 알고리즘

Ⅲ. 실험 및 결과

그림 4는 제안하는 CI 알고리즘의 블록도이다. 제안하는 CI 알고리즘에서는 RGB에 총 3개의 라인 메모리를 사용하여 하드웨어 복잡도를 줄인 것을 확인할 수 있다.

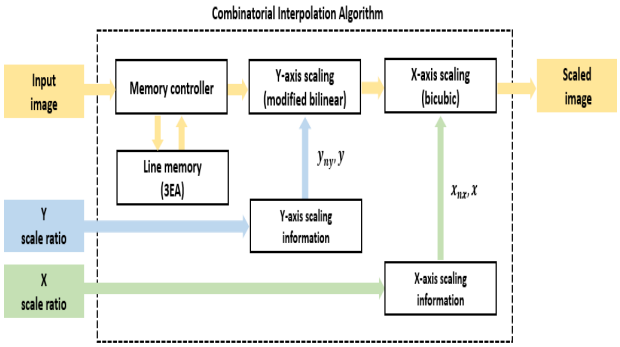

Fig. 4. Block diagram of Proposed combinatorial interpolation algorithm.
그림 4. 제안하는 조합 보간 알고리즘의 블록도

1. 정량적 평가

알고리즘의 정량적 평가를 위해 정현파 신호의 Signal-to-Noise Ratio(SNR)[8]와 그림 5의 256×256 영상 8장에 대해 Peak Signal-to-Noise Ratio (PSNR)[9]를 측정했다. 측정을 위해 사용된 확대 및 축소비율은 모두 1.3배를 사용했다.

가. SNR 수치 측정

SNR은 신호 대 잡음 비율로 수식 16과 같고 수치가 높을수록 성능이 좋음을 의미한다[8]. 정량적 평가를 위해 사용한 정현파 신호 $y(n)$은 수식 17을 이용해서 생성하면 된다.

$$SNR = 10\log_{10}\left(\frac{signal\,power}{noise\,power}\right) \tag{16}$$

$$y(n) = a \times \sin\left(\frac{2\pi fn}{f_s}\right) \tag{17}$$

$n = 1, 2, \cdots, cycle \times sample$, a는 진폭, f는 주파수, f_s는 샘플링 주파수, $cycle$은 주기, $sample$은 한 주기당 샘플 수로 fs/f이다. 수식 17의 각 파라미터 값를 $a = 2^7$, $f = 100KHz$, $fs = 100MHz$, $cycle = 1$로 설정하여 총 1,000개의 샘플 수를 가지는 정현파를 생성하여 비교하였다.

Table 1. Comparison of SNR for down scaling and up scaling of a sinusoidal function by a factor of 1.3.
표 1. 정현파의 1.3배 축소, 확대 SNR 비교[dB]

Method \ Scaling	Down Scaling	Up Scaling
NNI	47.43	47.41
mBLI	118.35	118.26
BCI	163.46	163.46

표 1과 같이 1차원 정현파 신호 성능을 비교하면 BCI의 성능이 가장 우수하다는 것을 알 수 있다. 하지만, X축이 아닌, Y축에 BCI를 적용하여 3채널의 RGB를 구현하려면 총 9개의 라인 메모리가 필요하므로 하드웨어 복잡도가 증가한다. 따라서 Y축에 mBLI을 적용한 CI를 사용할 수 있는데, 따라서 성능은 BCI보다 떨어지지만, 하드웨어 복잡도가 작은 장점이 있다.

나. PSNR 수치 비교

PSNR은 화질의 손실 정보를 평가할 때 사용하는 수치로, 수식 18과 같이 Mean Square Error(MSE)를 이용

Fig. 5. Experimental images : (a) Diagonal, (b) Lena, (c) CZP, (d) Camera, (e) Baboon, (f) Goldhill, (g) Peppers, (h) Airplane.

그림 5. 실험 영상 : (a) Diagonal, (b) Lena, (c) CZP, (d) Camera, (e) Baboon, (f) Goldhill, (g) Peppers, (h) Airplane

해 계산할 수 있다[9]. MSE는 입력 영상과 출력 영상 간의 차를 제곱한 평균으로, 수치가 낮을수록 성능이 좋음을 의미한다. PSNR은 수치가 높을수록 성능이 좋음을 의미한다.

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \tag{18}$$

그림 5는 PSNR 정량 평가를 위해 사용한 총 8개의 이미지를 보여준다. 그림 8(c)의 Circular Zone Plate (CZP)[10] 영상은 아래 수식 19를 이용해 생성하였다.

$$\begin{aligned} X(n+1) &= -\frac{t}{2} + n \quad (n = 0, 1, 2, \cdots, t-1) \\ Y &= X' \\ CZP &= ceil\left(127.5 \times \cos\left\{\frac{\pi}{t}(X^2 + Y^2)\right) - 0.5\right\} + 128 \end{aligned} \tag{19}$$

t는 생성될 CZP 영상의 크기로 256이다. X는 $-t/2$부터 1씩 증가하고 Y는 X의 전치 행렬이다. $ceil(\cdot)$ 함수는 소수점 올림 연산을 수행한다.

표 2~3은 각각 1.3배 확대 후 축소 및 축소 후 확대의 성능 결과를 보여준다. PSNR의 수치는 모두 BCI, CI, mBLI, NNI 순으로 높다는 것을 볼 수 있다.

Table 2. Comparison of PSNR for up scaling and then down scaling by a factor of 1.3.

표 2. 1.3배 확대 후 축소 PSNR 비교[dB]

Method / Image	NNI	mBLI	BCI	CI
Diagonal	23.07	35.70	43.96	39.45
Lena	21.81	32.65	40.36	36.01
CZP	6.03	14.89	21.08	16.98
Camera	19.92	32.65	40.36	36.01
Baboon	19.35	30.20	37.08	31.96
Goldhill	23.37	36.72	44.50	39.28
Peppers	23.33	37.68	47.00	40.06
Airplane	28.53	43.04	52.85	45.29
Average	20.67	33.39	41.21	35.93

Table 3. Comparison of PSNR for down scaling and then up scaling by a factor of 1.3.

표 3. 1.3배 축소 후 확대 PSNR 비교[dB]

Method / Image	NNI	mBLI	BCI	CI
Diagonal	22.12	33.31	36.36	34.78
Lena	20.69	31.80	34.82	33.62
CZP	6.02	11.47	12.33	11.90
Camera	18.88	28.66	30.98	29.98
Baboon	18.89	26.47	27.89	27.09
Goldhill	22.43	32.62	34.91	33.71
Peppers	22.70	34.36	36.89	35.23
Airplane	27.20	39.93	43.39	40.24
Average	19.87	29.70	32.20	30.82

다. 선행 논문과 수치 비교

본 논문에서 제안한 CI 알고리즘 성능을 평가하기 위하여, 하드웨어 구현을 보여준 선행 연구와 비교하였다. 이 논문에서는 하드웨어 자원량을 줄이기 위해 BLI와 BCI를 결합한 이중 선형-3차 회선 보간 알고리즘을 제안했다[11]. 선행 논문에서는 PSNR 측정을 위해 Lena와 Camera 입력 영상을 0.5배로 축소 후 확대한 결과를 사용하였다. 이와 동일한 조건으로 제안한 조합 보간 알고리즘 성능을 비교하였다. 표 4에서 볼 수 있듯이, 본 논문에서 제안하는 CI 방식의 PSNR 수치가 더 좋은 것을 볼 수 있다.

Table 4. Comparison of PSNR for down scaling and then up scaling to that in the previous paper.
표 4. 선행 논문과 축소 후 확대 PSNR 비교[dB]

Method / Image	Moon *et al.*[11]	CI
Lena	27.71	29.38
Camera	25.10	25.82

2. 정성적 평가

알고리즘을 정성적으로 평가하기 위해, 그림 5의 Diagonal 이미지와 Lena 이미지를 1.3배 확대 후 축소한 결과를 비교하였다. 그림 6에서 볼 수 있듯이, NNI는 대각선 경계 부근인 고주파 영역에 계단 모양과 같은 노이즈가 눈에 띄게 나타난 것을 볼 수 있다. 하지만 그 외 mBLI, BCI, CI 간의 차이는 크지 않은 것을 알 수 있다.

Ⅳ. 결론

본 논문에서는 Y축에는 mBLI를, X축에는 BCI를 적용하는 CI 알고리즘을 제안하였다. 정량적 평가에서는 BCI, CI, mBLI, NNI 순으로 좋은 성능을 보여주었다. 반면, 정성적 평가 결과 NNI의 성능이 눈에 띄게 나빴지만, 다른 세 가지 방식 간의 차이는 크지 않은 것으로 나타났다. 따라서 본 논문에서 제안한 CI 알고리즘은 BCI에 비해 정량적 평가 성능이 조금 떨어지지만, 사용되는 라인 메모리 수를 줄여 하드웨어 설계 부담을 줄일 수 있음을 보여주었다.

References

[1] R. Crane, *A Simplified Approach to Image Processing*, Prentice Hall, 1997.

[2] R. C. Gonzalez, R. E. Woods and S. L. Eddins, *Digital Image Processing using MATLAB*, Prentice Hall, 2004.

[3] Paulinternet.nl, "Cubic interpolation," https://www.paulinternet.nl/?page=bicubic

[4] Y. H. Jun, J. H. Yun, J. S. Park and M. R. Choi, "Implementation of a Modified Cubic Convolution Scaler for Low Computational Complexity," *Journal of Korea Multimedia Society*, vol.10, no.7, pp.838-845, 2007.

[5] J. Y. Han and S. W. Lee, "Low-power VLSI Architecture Design for Image Scaler and Coefficients Optimization," *The Institute of Electronics Engineers of Korea-Semiconductor and Devices*, Vol.47, No.6, pp.22-34, 2010.

[6] C. C. Lin, M. H. Sheu, H. K. Chiang, C. Liaw, Z. C. Wu and W. K. Tsai, "An Efficient Architecture

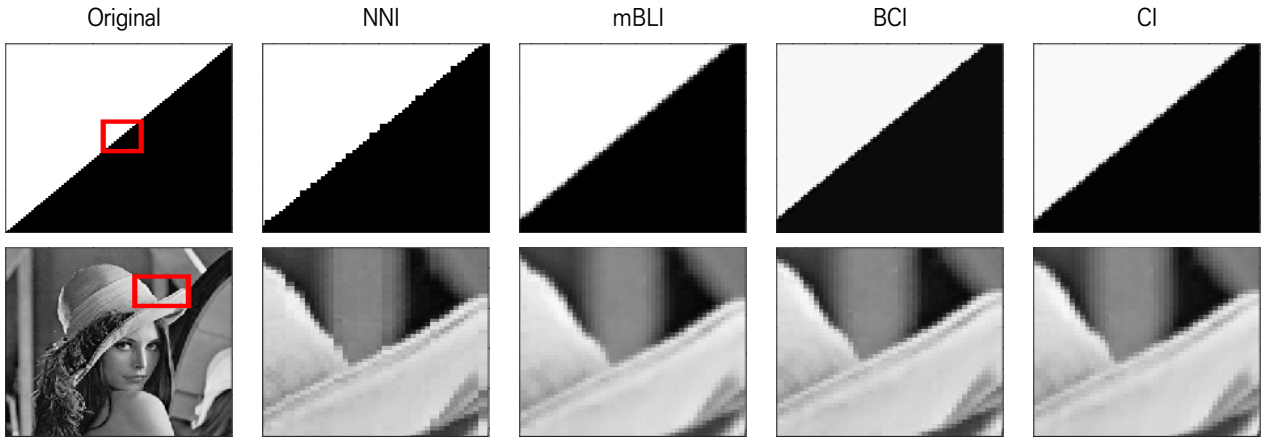

Fig. 6. Results of up scaling and then down scaling by a factor of 1.3.
그림 6. 1.3배 확대 후 축소 결과

of Extended Linear Interpolation for Image Processing," *Journal of Information Science and Engineering 26*, vol.26, no.2, pp.631-648, 2010.
[7] MathWorks, "imresize"https://kr.mathworks.com/help/matlab/ref/imresize.html
[8] S. Haykin, M. Moher, *Introduction to Analog and Digital Communications*, Wiley, 2007.
[9] Wikipedia, "Peak signal-to-noise ratio," https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
[10] J. H. Park, W. W. Jang, S. J. Lee, B. D. Kwak and B. S. Kang, "Hardware Implementation of 2D-CZP Pattern for System verification," *2007 Autumn Annual Conference of IEIE,* vol.30, no.2, pp.597-598, 2007.
[11] H. M. Moon and S. B. Pan, "VLSI Architecture of Digital Image Scaler Combining Linear Interpolation and Cubic Convolution Interpolation," *Journal of The Institute of Electronics and Information Engineers*, vol.51, no.3, pp.112-118, 2014. DOI: 10.5573/ieie.2014.51.3.112

BIOGRAPHY

Si-Yeon Han (Member)

2023 : BS degree in Electronics Engineering, Dong-A University.
2023~Present : MS degree course in Electronics Engineering, Dong-A University.

Bongsoon Kang (Member)

1985 : BS degree in Electronics Engineering, Yonsei University.
1987 : MS degree in Electrical Engineering, University of Pennsylvania.
1990 : PhD degree in Electrical Engineering, Drexel University.
1989~1999 : Senior Staff Researcher, Samsung Electronics.
1999~Present : Prof. of Dept. of Electronic Engineering, Dong-A University.