International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 10, No. 6 (2017), pp. 119-130
http//dx.doi.org/10.14257/ijsip.2017.10.6.11

Image Resizing for Large Displays

Gwanggil Jeon

Department of Embedded Systems Engineering, Incheon National University
119 Academy-ro, Yeonsu-gu, Incheon 406-772, Korea
gjeon@inu.ac.kr

Abstract

The purpose of the image resizing is to rescale size of image to fit display device. To achieve this goal, we present a new image resizing technique. The proposed method consists of three steps, original image, (1) self-training by least squares step, (2) image upscaling step, (3) image unsharp masking step, and result image step. Three parameters p, k and h are computed to provide pleasant result images. Simulation results show that the proposed method outperforms benchmark methods in terms of PSNR, MSE, and SSIM metrics. In addition, visual performance comparison indicates that the proposed method is superior to the other methods.

Keywords*: image resizing, image zooming, display device, unsharp masking, least squares method*

1. Introduction

The signal scaling is an important and basic operation in image processing. The scaling can be enlarged and decreased, which is known as resizing process [1-2]. This resizing process is widely used in multimedia tools such as medical imaging and digital photography. The goal of interpolation method is to estimate the data with a continuous signal model and then restore this function on the grid proper to the wanted scaling [3].

Recently, the resizing method is applied in various fields. For instance, image fitting in the heterogeneous displays or screens can play an important role by rescaling technique [4-5]. In general, the resizing approaches are categorized into two classes: the enlarging process and the shrinking process [7]. In both processes, interpolation technique is important. There are several interpolation approaches for image resizing. One of well-known and simplest approaches is nearest neighbor interpolation method. In addition, bilinear interpolation method, bicubic interpolation method, and Lanczos interpolation are also widely used.

Although there are several interpolation approaches, the most important role of interpolation is to restore and preserve details in the restored image. The conventional interpolation methods cannot fully preserve the details, therefore we present a new interpolation method. There have been many applications such as deinterlacing [8-11], denoising [12-13], and demosaicking [14-15] that use interpolation methods.

In this paper, we proposed a new image resizing method. Rest of the paper is organized as follows. The proposed method is presented in Section 2. In Section 3, experimental results are described. Section 4 provides conclusion remarks.

2. Proposed Method

The image resizing is one of key techniques to manage low resolution images to high resolution images. The flowchart of the proposed method is shown in Figure 1. The proposed method consists of five blocks and three operations.

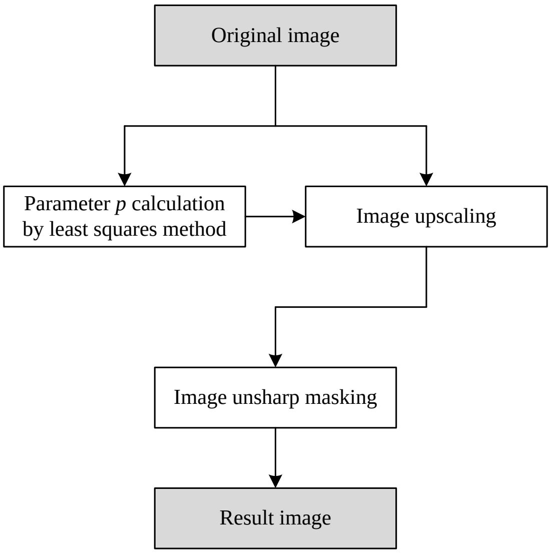


Figure 1. Flowchart of the Proposed Approach

Let us assume that the low resolution original image $A_{i,j}$ of size *height*×*width* straightly acquired from of size of 2*height*×2*width*, then $B_{2i,2j}$=$A_{i,\ j}$. Here, $B_{2i,2j}$ is result high resolution image. Now, the problem is how we can interpolate the interlacing lattice $B_{2i+1,2j+1}$ from the lattice $B_{2i,2j}$=$A_{i,\ j}$. In other words, calculating parameter *p* is one of goals of this work. Equation (1) shows the basic form of the image upsampling.

$$B_{2i+k,2j+l} = \sum_{k=0}^{1}\sum_{l=0}^{1} p_{2k+l} B_{2(i+k),2(j+l)} \tag{1}$$

If we assume that the given images can be considered as locally stationary, independent and identically distributed Gaussian model, then by standard least squares methods parameter *p* can be computed as

$$\vec{p} = C^{-1}\vec{c} \tag{2}$$

where C=[C_{kl}] and $\vec{c}$ =[c_k] are local covariances at the high resolution image. By applying Eq. (1) and Eq. (2), image upscaling is performed. This technique can be widely used such as in image deinterlacing, denoising and demosaicking, where one can magnify the size of grayscale or multichannel images.

After image upsampling is performed, we apply unsharp masking (UM). The UM process is an image enhancement method. The upsampling process may include interpolation and estimation. However, as the original image size is half of result image, this procedure is ill-posed. In addition, the upsampled image is generally blurred as edge details are losing. Thus, the UM process enhances an image which is less blurry than the upsampled image. This process is performed as,

$$\begin{aligned} \overset{\boxplus}{B} &= B + \left\{ B - lpf\left(B\right) \right\} \cdot k \\ &= \left(1+k\right) B - lpf\left(B\right) \cdot k \end{aligned} \tag{3}$$

where *k* is the amount parameter which controls the UM effect. If *k* is bigger value, more UM effect is applied. The command lpf is low pass filter, and lpf(B) is result image after low pass filtering. We assume lpf filter is Eq. (3). Parameter $\overset{\boxplus}{B}$ is the UM effect applied result. The function imfilter filters the 2-dimensional image *x* with filter [1 1 1; 1 1 1; 1 1 1]/(*h*+8). In this paper, parameters *k* and *h* are set to 0.5 and 2. Figure 2 compares original blurred image and its corresponding UM process image.

Figure 2. (a) Original 'Mango' Image. (b) UM Processed 'Mango' Image

$$lpf\left(x\right)=imfilter\left(x,\frac{1}{h+8}\begin{bmatrix}1 & 1 & 1\\ 1 & h & 1\\ 1 & 1 & 1\end{bmatrix}\right) \tag{4}$$

3. Simulation Results

In this Section, to prove the quality of the presented approach, we tested experiments using MATLAB software with a processor of Intel(R) Core(TM) i5 CPU M460 @2.53GHZ. We used twenty LC dataset, between #1 to #20 images. Figure 3 shows twenty LC images. The size of test image is 720x540. For visual performance comparison, we used three benchmark methods, nearest neighbor (NN), bilinear (BI), and bicubic (BC) methods. We used three quality assessment tools: mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity (SSIM) metrics.

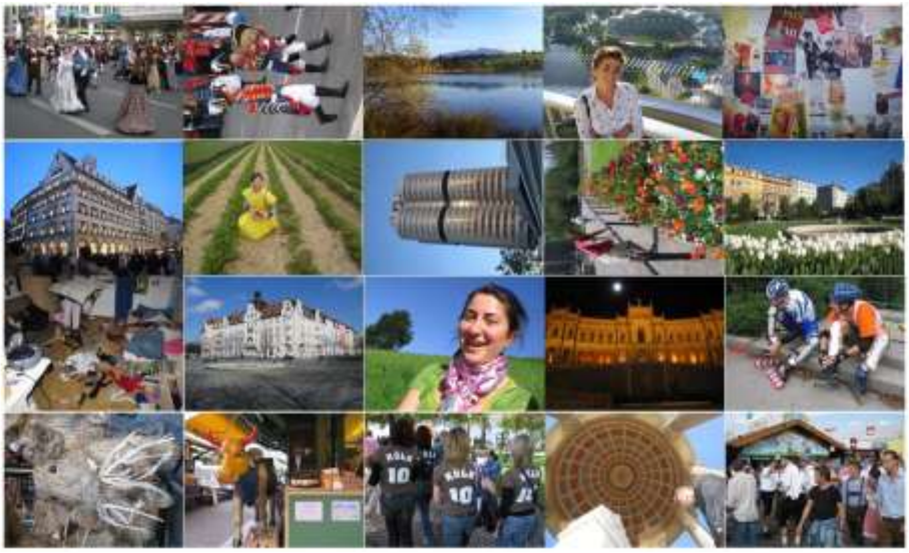


Figure 3. Test Image Sets: LC Dataset. #1-#20 Images are Used in the Simulation

To assess the performance of objective quality and visual performance, three metrics were used. The PSNR is computed as followed,

$$PSNR=10\times\log_{10}\left(\frac{MAX^{2}}{MSE}\right) \tag{5}$$

where MSE denotes the mean squared error and MAX stands for the maximum possible intensity of the given image. When a pixel is represented using *n* bits per sample, MAX is 2^n-1. In general, acceptable PSNR level is between 30 to 50 dB for 8 bits images. For the case of 16 bits images, 60 to 80 dB is desirable. The MSE is mean squared error, which is computed as,

$$MSE = \frac{1}{MN}\sum_{m=0}^{M}\sum_{n=0}^{N}\left[ori\left(m,n\right)-res\left(m,n\right)\right]^2 \qquad (6)$$

where ori and res stand for original and the result images.

The SSIM index is invented to assess the quality between the original and the restored images. The SSIM is used to estimate the similarity between two given images. As original image is used for assessment, SSIM is a full reference metric. The SSIM is intended to enhance the conventional quality assessment tools such as PSNR, MSE.

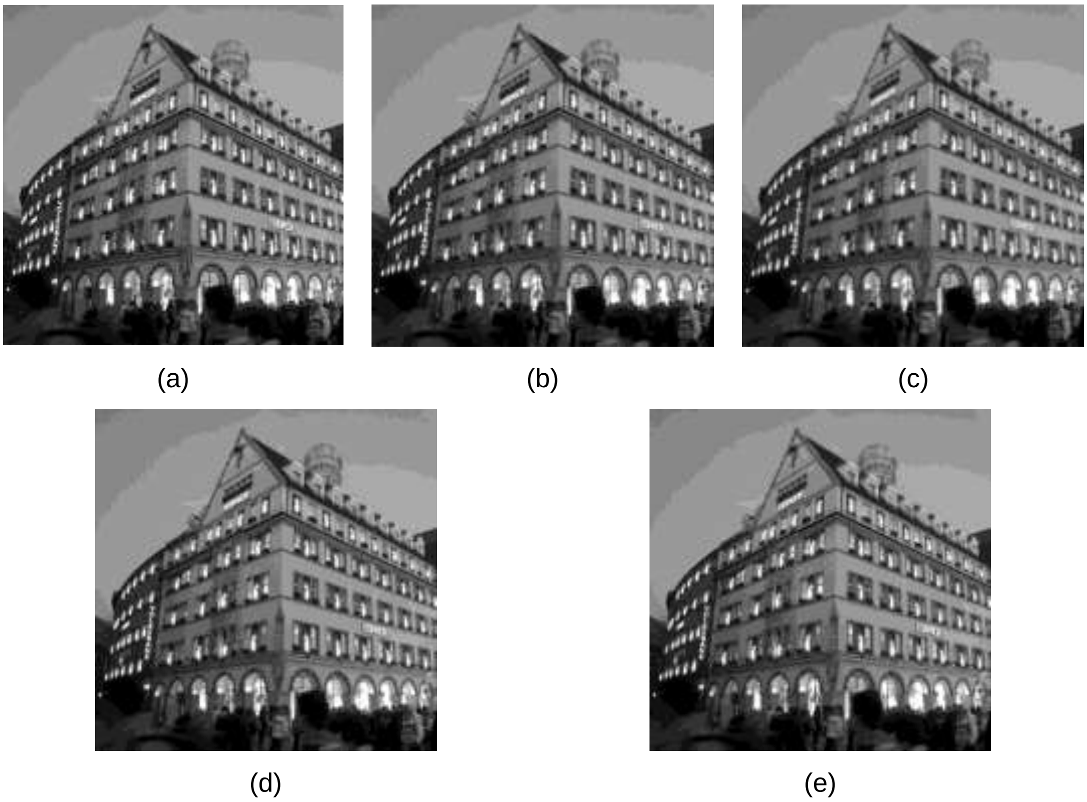
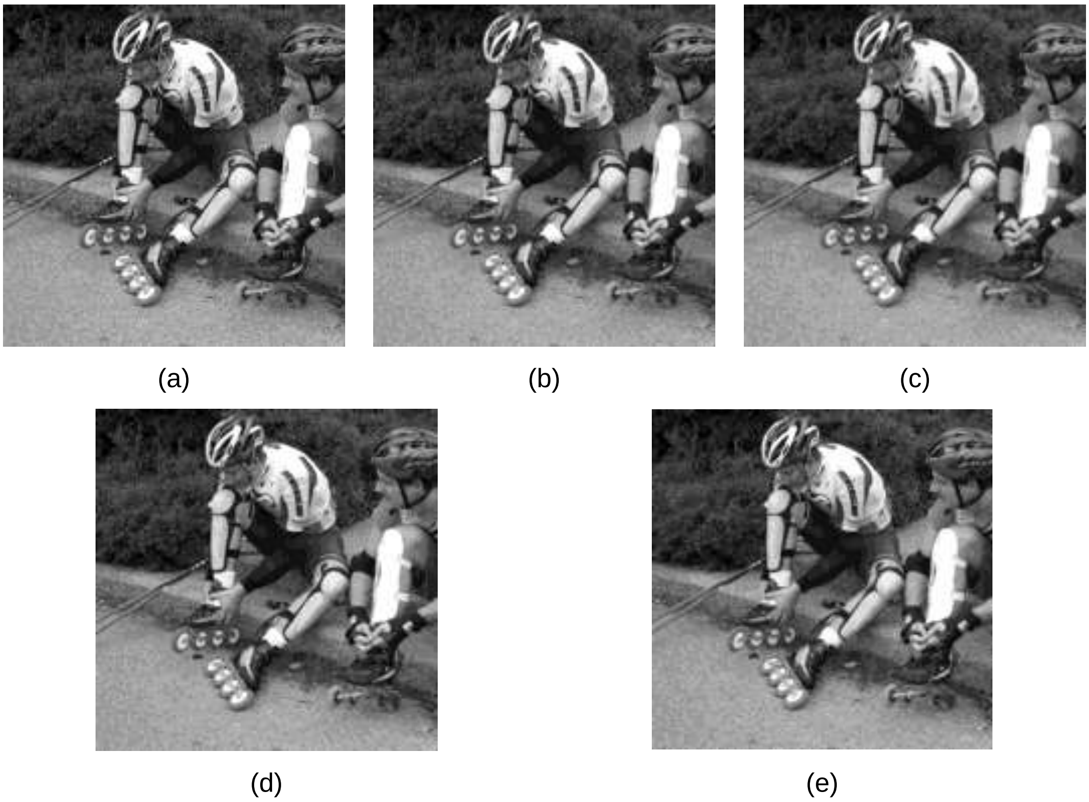


Figure 4. Visual Performance Comparison on #6 LC Dataset. (a) Original Image, (b) Result of Nearest Neighbor Method, (c) Result of Bilinear Method, (d) Result of Bicubic Method, and (e) Result of Proposed Method

The restored images are displayed in Figs. 4-6. In all figures, images (a) indicate original cropped images with 256x256 size. In the same manner, images (b), (c), (d), and (e) are the results by nearest neighbor method, bilinear method, bicubic method, and the proposed method. All figures show that the proposed method generates subjectively pleasant results when compared to the benchmark methods. For example, the staircase artifices are shown in nearest neighbor result, blurring effect are shown in bilinear and bicubic, while the proposed method well preserved details. This can be found in window and door areas in Figure 4.



(a) (b) (c)

(d) (e)

Figure 5. Visual Performance Comparison on #15 LC Dataset. (a) Original Image, (b) Result of Nearest Neighbor Method, (c) Result of Bilinear Method, (d) Result of Bicubic Method, and (e) Result of Proposed Method

Figure 5 shows 'bikemen' image. The result image by the proposed method well restore characters on the jersey, helmet, and roller skate. Figure 6 shows roof images. Figures 6(b), 6(c), and 6(d) do not well preserve roof details, while the proposed method shows the details.

Figures 7 and 8 show SSIM results where #1 and #3 images were adopted for comparison. Four images (a) to (d) are results of nearest neighbor, bilinear, bicubic, and the proposed methods, respectively. In SSIM metric, the result close to '1' indicates better image while the result close to '0' indicates worse image. As one can see, the proposed method provides the best SSIM result throughout the given image. In Figure 8, SSIM value on edge area near the mountain is high. This implies that the proposed method outperforms the other methods.

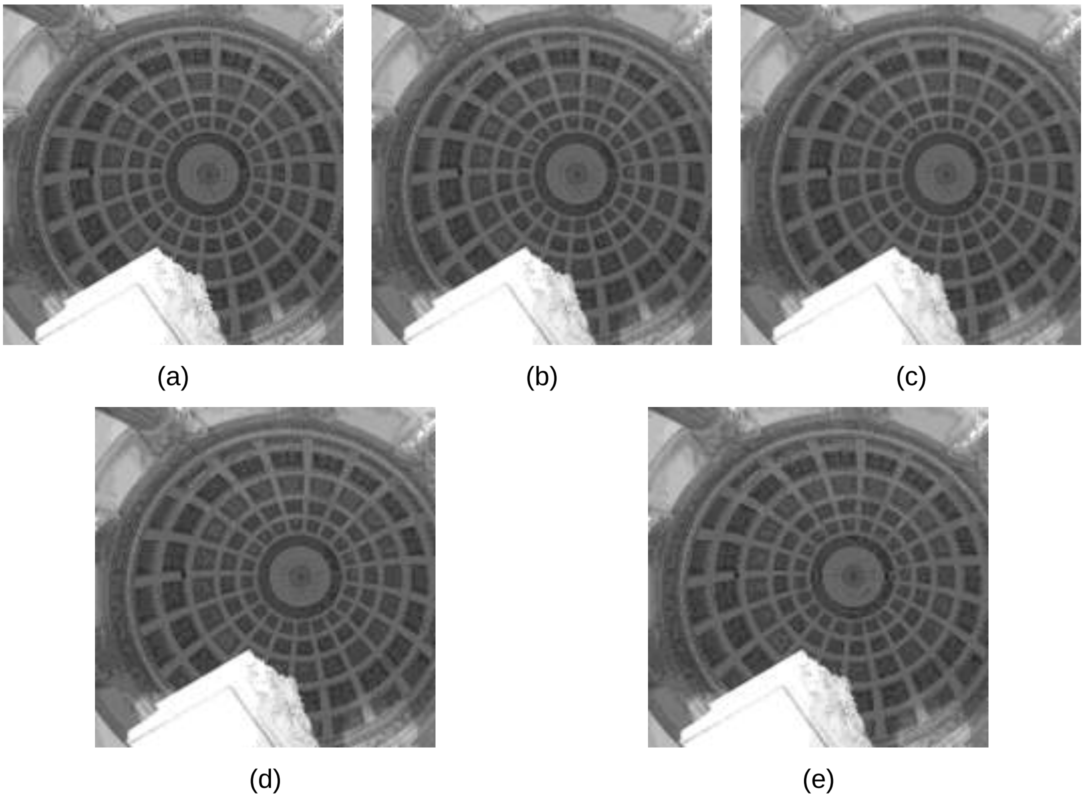
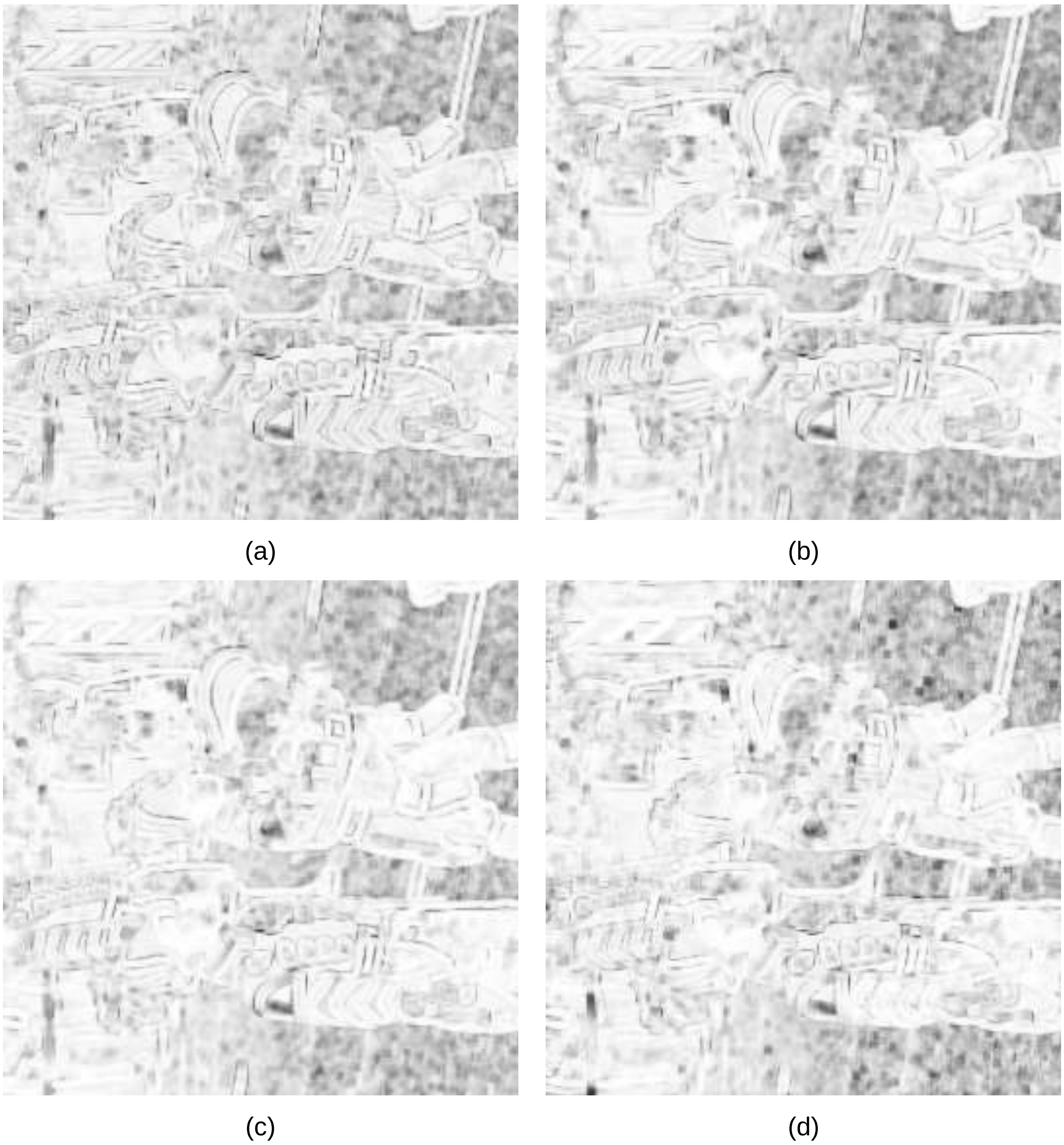


Figure 6. Visual Performance Comparison on #19 LC Dataset. (a) Original Image, (b) Result of Nearest Neighbor Method, (c) Result of Bilinear Method, (d) Result of Bicubic Method, and (e) Result of Proposed Method

Objective performance comparisons are provided in Tables 1-3, using PSNR, MSE, and SSIM metrics. In terms of PSNR metric, the proposed method outperforms other methods with the amount of 0.776, 0.375, and -0.537 dB. The PSNR metric tells the bicubic method provides better performance than the prosed method, but the visual performance comparison informs the proposed method yields better result. Similarly, we used MSE and SSIM metrics for performance assessment. The proposed method outperforms other methods with the amount of -19.392, -8.506, and 18.756 for MSE metric, 0.010, 0.024, and -0.011 for SSIM metric.

Figure 9 shows visual performance comparison in zoomed area. As described in the previous section, the nearest neighbor method is one of simplest methods which replaces every pixel with a number of pixels of the same intensity. Therefore, the restored image shows unwanted jaggedness but all pixels are original. In the result of nearest neighbor method, staircase artifacts are clearly shown. The bilinear method and the bicubic method provide blurred image. However, the proposed method generate pleasant result image that preserving details. Both methods introduce a successive change into the result even where the original pixel has discrete changes. Therefore, both methods lessens contrast which is undesirable for linear pattern images. Although bicubic method is known as superior to the bilinear method, but the difference is negligible and the complexity is higher.



(a) (b)

(c) (d)

Figure 7. SSIM Map Images on #1 LC Dataset. (a) The Nearest Neighbor Method, (b) The Bilinear Method, (c) The Bicubic Method, and (d) The Proposed Method

4. Conclusions

In this paper, we proposed an image resizing method. The goal of the proposed method is to rescale image size to fit the given display device. The proposed method consists of three steps, S1: least squares method based self-training, S2: image upscaling, and S3: image unsharp masking. Parameter p is calculated in S1 and parameters k and h are calculated in S3. Experimental result show that the presented approach provides satisfactory results in terms of PSNR, MSE, and SSIM metrics. Moreover, subjective performance comparison tells that the proposed method is superior to the other methods.

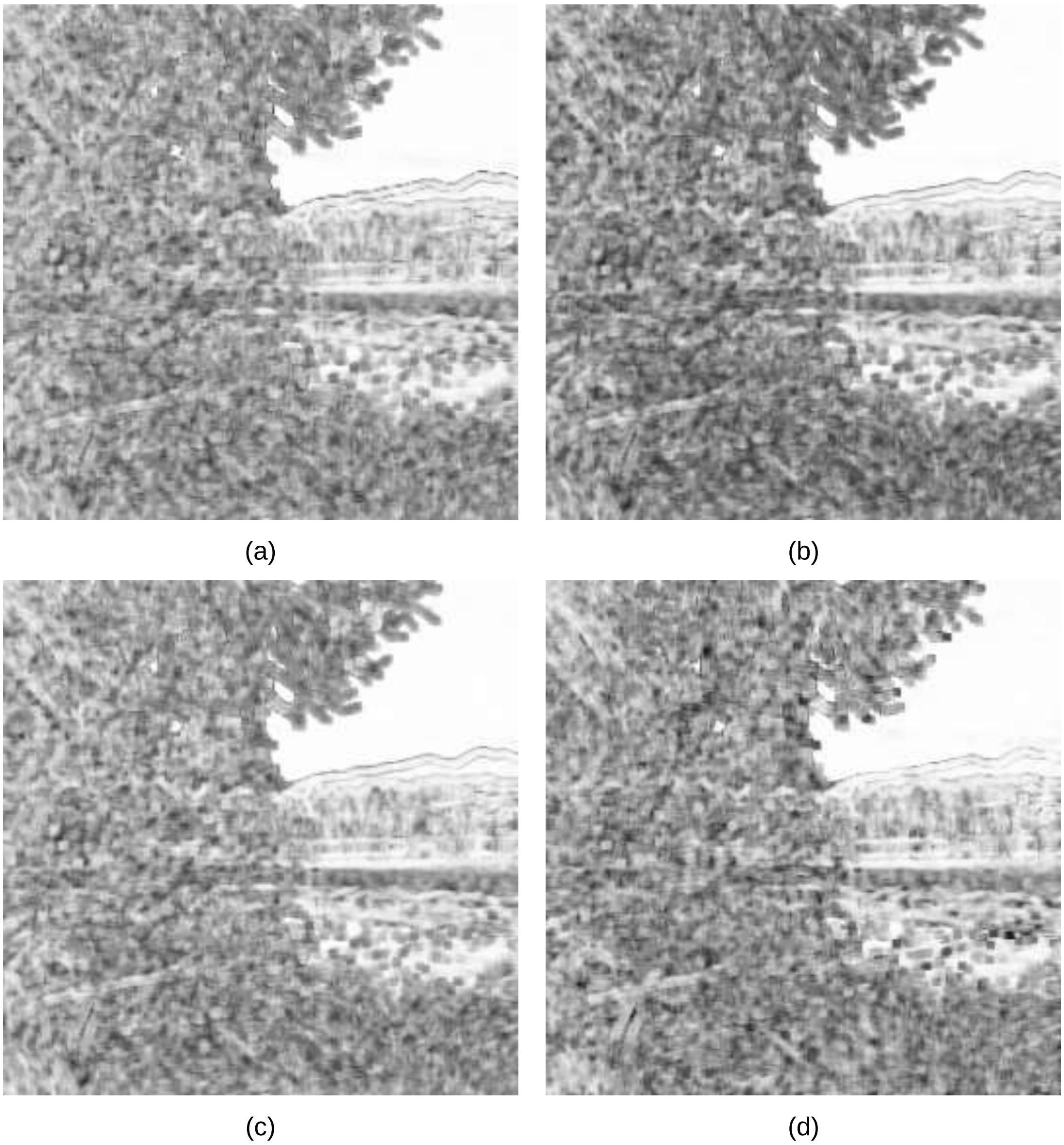


Figure 8. SSIM Map Images on #3 LC Dataset. (a) The Nearest Neighbor Method, (b) The Bilinear Method, (c) The Bicubic Method, and (d) The Proposed Method

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) Grant funded by the Korean Government (2015R1D1A1A01058171). This paper is a revised and expanded version of a paper entitled "Zooming Method for Higher Resolution Displays" presented at SIP2016.

Table 1. PSNR Performance Comparison on Four Different Methods. NN: Nearest Neighbor, BI: Bilinear, BC: Bicubic, and PM: Proposed Method

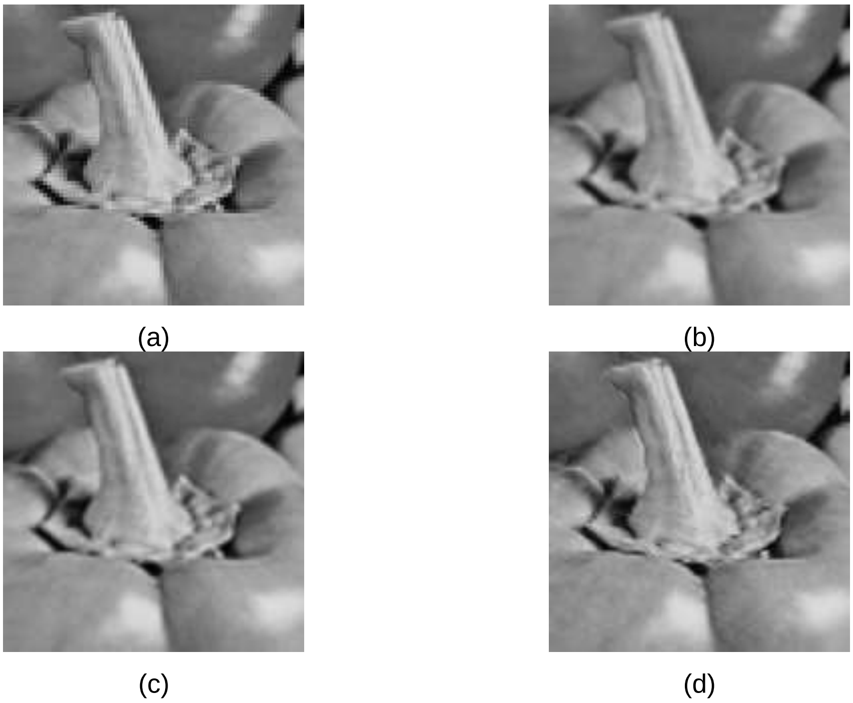
	NN	BI	BC	PM
1	25.358	25.887	26.848	26.367
2	27.447	28.541	29.824	29.454
3	22.191	22.031	22.547	21.794
4	28.218	29.079	30.475	30.384
5	27.621	28.069	29.039	28.559
6	26.046	26.788	27.852	27.488
7	25.167	25.042	25.644	24.741
8	29.015	29.296	30.027	29.243
9	23.718	23.855	24.624	23.941
10	23.850	23.977	24.765	24.130
11	26.864	27.346	28.365	27.924
12	24.339	24.343	25.024	24.300
13	30.225	31.180	32.464	32.263
14	30.645	30.909	31.666	30.712
15	25.553	25.976	26.862	26.400
16	24.779	25.075	26.067	25.724
17	26.256	26.702	27.663	27.329
18	25.731	26.288	27.338	26.800
19	29.623	29.794	30.440	29.969
20	26.515	27.005	27.890	27.170
Avg.	26.458	26.859	27.771	27.234

Table 2. MSE Performance Comparison on Four Different Methods. NN: Nearest Neighbor, BI: Bilinear, BC: Bicubic, and PM: Proposed Method

	NN	BI	BC	PM
1	189.361	167.655	134.373	150.114
2	117.054	90.992	67.717	73.740
3	392.597	407.382	361.738	430.214
4	98.008	80.379	58.293	59.529
5	112.457	101.439	81.124	90.612
6	161.621	136.241	106.635	115.960
7	197.890	203.644	177.277	218.254
8	81.583	76.466	64.628	77.416
9	276.244	267.655	224.216	262.390
10	267.966	260.223	217.055	251.253
11	133.870	119.817	94.754	104.868
12	239.422	239.216	204.471	241.601
13	61.741	49.554	36.873	38.615
14	56.048	52.749	44.311	55.195
15	181.051	164.232	133.932	148.951
16	216.347	202.084	160.825	174.062
17	153.983	138.951	111.375	120.278
18	173.791	152.853	120.026	135.871
19	70.921	68.191	58.756	65.489
20	145.070	129.588	105.690	124.774
Avg.	166.351	155.466	128.203	146.959

Table 3. SSIM Performance Comparison on Four Different Methods. NN: Nearest Neighbor, BI: Bilinear, BC: Bicubic, and PM: Proposed Method

	NN	BI	BC	PM
1	0.81328	0.80253	0.83749	0.82687
2	0.86628	0.86787	0.89527	0.88492
3	0.68644	0.63468	0.68952	0.66722
4	0.88879	0.89465	0.92041	0.91575
5	0.84512	0.83719	0.87115	0.86231
6	0.89377	0.89662	0.91869	0.91706
7	0.72937	0.68636	0.74292	0.71098
8	0.88860	0.87900	0.89797	0.89110
9	0.76651	0.73621	0.78632	0.76454
10	0.83143	0.81376	0.84774	0.84068
11	0.84717	0.83944	0.87055	0.86140
12	0.82357	0.80589	0.83860	0.82775
13	0.90698	0.90828	0.93036	0.92599
14	0.87562	0.86451	0.88898	0.87573
15	0.75921	0.72654	0.77699	0.75721
16	0.81114	0.79198	0.84205	0.83848
17	0.85708	0.85011	0.87924	0.87483
18	0.86097	0.85650	0.88846	0.88310
19	0.79474	0.77678	0.81416	0.80825
20	0.86105	0.85911	0.88581	0.87634
Avg.	0.83036	0.81640	0.85113	0.84052



(a) (b)

(c) (d)

Figure 9. Visual Performance Comparison in Zoomed Area (a) Nearest Neighbor Method, (b) Bilinear Method, (c) Bicubic Method, and (d) The Proposed Method

References

[1] R. C. Gonzalez, R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Addision-Wesley, Reading MA, **(1992)**.

[2] R. C. Tam and A. Fournier, "Image interpolation using unions of spheres," The Visual Computer, vol. 14, **(1998)**, pp. 401-414.

[3] E. Maeland, "On the comparison of interpolation methods", IEEE Trans. Med. Image, vol. 7, no. 3, **(1988)**, pp. 213-217.

[4] J. A Parker, R. V Kenyon and D. E Troxel, "Comparison of interpolating methods for image resampling", IEEE Trans. Med. Image, vol. 2, no. 1, **(1983)**, pp. 31-39.

[5] R. Adipranata, E. Cherry, G. Ballangan and R. P. Ongkodjojo, "Fast Method for Multiple Human Face Segmentation in Color Image", International Journal of Advanced Science and Technology, **(2009)**, pp. 19-32.

[6] D. Bhattacharyya, A. Roy, P. Roy and T.-h. Kim, "Receiver Compatible Data Hiding in Color Image", International Journal of Advanced Science and Technology, **(2009)**, pp. 15-24.

[7] B. Bhanu, J. Peng, T. Huang, and B. Draper, "Introduction to the special issue on learning in computer vision and pattern recognition", IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 35, no. 3, **(2005)**, pp. 391-396.

[8] J. Wang, G. Jeon, and J. Jeong, "Efficient adaptive de-Interlacing algorithm with awareness of closeness and similarity", SPIE Optical Engineering, vol. 51, no. 1, **(2012)**, pp. 017003.

[9] X. Chen, G. Jeon, and J. Jeong, "A filter switching interpolation method for deinterlacing," SPIE Optical Engineering, vol. 51, no. 10, **(2012)**, pp. 107402.

[10] S. J. Park, G. Jeon, and J. Jeong, "Covariance-based adaptive deinterlacing method using edge map," in Proc. IEEE IPTA2010, Paris, France, **(2010)**, pp. 166-171.

[11] S. J. Park, G. Jeon, and J. Jeong, "Deinterlacing algorithm using edge direction from analysis of the DCT coefficient distribution", IEEE Trans. Consumer Electronics, vol. 55, no. 3, **(2009)**, pp. 1674-1681.

[12] G. Jeon, S. J. Park, Y. Fang, M. Anisetti, V. Bellandi, E. Damiani, and J. Jeong, "Specification of efficient block matching scheme for motion estimation in video compression", SPIE Optical Engineering, vol. 48, no. 12, **(2009)**, pp. 127005.

[13] J. Wu, Z. Xu, G. Jeon, X. Zhang, and L. Jiao, "Arithmetic coding for image compression with adaptive weight-context classification", Signal Processing: Image Communication, vol. 28, no. 7, **(2013)**, pp. 727-735.

[14] X. Chen, G. Jeon, and J. Jeong, "Voting-based directional interpolation method and its application to still color image demosaicking", IEEE Trans. Circuits and Systems for Video Technology, vol. 24, no. 2, **(2014)**, pp. 255-262.

[15] X. Chen, G. Jeon, J. Jeong, and L. He, "Multidirectional weighted interpolation and refinement method for Bayer pattern CFA demosaicking", IEEE Trans. Circuits and Systems for Video Technology, vol. 25, no. 8, **(2015)**, pp. 1271-1282.

Author

Gwanggil Jeon, received the BS, MS, and PhD (summa cum laude) degrees in Department of Electronics and Computer Engineering from Hanyang University, Seoul, Korea, in 2003, 2005, and 2008, respectively. From 2008 to 2009, he was with the Department of Electronics and Computer Engineering, Hanyang University, from 2009 to 2011, he was with the School of Information Technology and Engineering (SITE), University of Ottawa, as a postdoctoral fellow, and from 2011 to 2012, he was with the Graduate School of Science & Technology, Niigata University, as an assistant professor. He is currently an associate professor with the Department of Embedded Systems Engineering, Incheon National University, Incheon, Korea. His research interests fall under the umbrella of image processing, particularly image compression, motion estimation, demosaicking, and image enhancement as well as computational intelligence such as fuzzy and rough sets theories. He was the recipient of the IEEE Chester Sall Award in 2007 and the 2008 ETRI Journal Paper Award.