International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.3 (2014), pp.107-116
http://dx.doi.org/10.14257/ijmue.2014.9.3.11

Directionally Categorized Training Process for Hybrid Super-Resolution Algorithm

Gwanggil Jeon

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

Abstract

Nowadays, image upsampling methods have become common as large display market is growing. We present a new horizontal direction upsampling method for high resolution display using filter and bilinear filter. Two parameters δ and λ determine which reconstructed results should be used as the final value. Simulation results show that the proposed method achieves better results than the traditional methods from both of visual quality and objective performance.

Keywords: *image upsampling, image reconstruction, horizontal direction, edge direction*

1. Introduction

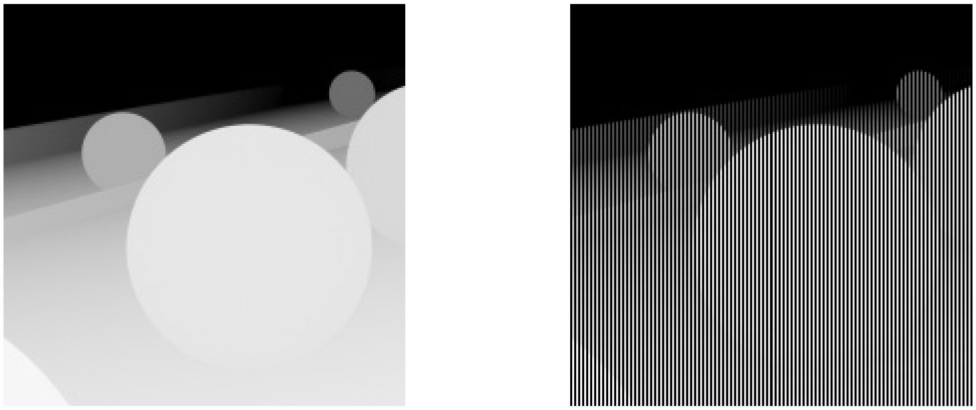
An image upsampling is adopted for image restoration to enhance the view of displayed images for viewers [1, 2]. The image upsampling can include image zooming or enlarging. Currently many display formats are used and each display format demands adjustable approach for resizing images by upsampling or downsampling [3, 4]. To meet this requirement, this paper studies the method how to appraise original pixel in missed pixels using adjacent pixels, and create bigger image from small one. Therefore few examples of resampling scenarios: *plasma display panel* to *liquid crystal display* or vice versa [5, 6].

There are two classes of resampling algorithms: one is traditional approach and the other is adaptive approach. In traditional approach, the algorithm is applied haphazardly in the whole image. Two of the representative methods are nearest neighbor method and bilinear method [7]. Both methods apply the same approach to the whole image, therefore it requires less computation. However at the same time, it causes blurred image and it is unsuitable for enlarging photographic images because it multiplies the subjective quality of artifacts resembling stairs. On the other hand, the adaptive method is intended to apply certain area and use different approach in the same image. Our proposed method is categorized in the second approach, i.e. adaptive method.

In this paper, we proposed a method for low resolution depth images to reconstruct the original sized images [8-11]. Figure 1 shows the original depth image and its horizontally how-resolution image. The remainder of the paper is organized as follows. The directionally categorized image reconstruction approach is introduced in Section 2. Section 3 provides the

ISSN: 1975-0080 IJMUE

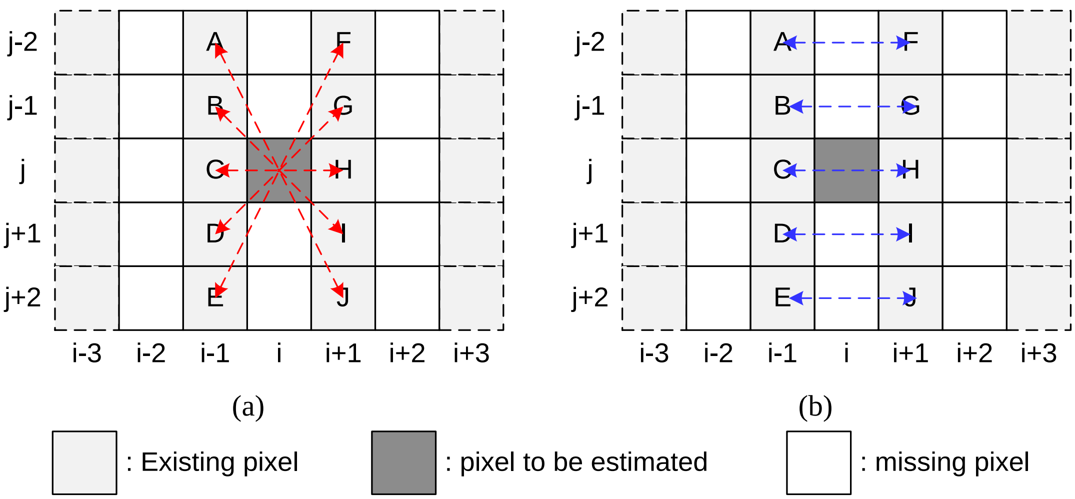
simulation results. The objective and visual performance comparison are conducted in this section. The paper is concluded with an overall discussion in Section 4.



(a) (b)

Figure 1. (a) Original depth map image (Image #1), (b) horizontally downsampled image

2. Directionally Categorized Image Reconstruction



(c)

Figure 2. (a) Estimation of pixel difference in five directions: δ_{60}, δ_{45}, δ_{0}, δ_{-45}, and δ_{-60}. (b) Estimation of pixel difference in five directions: λ_{j-2}, λ_{j-1}, λ_{j}, λ_{j+1}, and λ_{j+2}. (c) Pixel color explain: light gray is existing pixel, dark gray is pixel to be estimated, and white is missing pixel

Figure 2 shows the introduced directionally categorized upsampling method, where i and j stand for horizontal and vertical line numbers, respectively. Pixel x (i,j) is denoted as the center pixel intensity, which is located at (i, j) and requested to estimated in this work. We assume pixels A, B, C, D, E, F, G, H, I, and J are existing pixels at column number *i*-1 and *i*+1, as shown in Eq. (1).

$$\begin{aligned}
A &= x(i-1, j-2), & F &= x(i+1, j-2), \\
B &= x(i-1, j-1), & G &= x(i+1, j-1), \\
C &= x(i-1, j), & H &= x(i+1, j), \\
D &= x(i-1, j+1), & I &= x(i+1, j+1), \\
E &= x(i-1, j+2), & J &= x(i+1, j+2).
\end{aligned} \tag{1}$$

We evaluate two parameters, δ and λ. A parameter δ is calculated in different edge direction while a parameter λ is computed as a difference between column number *i*-1 and *i*+1. Equations (2) and (3) show parameters δ and λ.

$$\begin{aligned}
\delta_{60} &= \left| x(i+1, j-2) - x(i-1, j+2) \right|, \\
\delta_{45} &= \left| x(i+1, j-1) - x(i-1, j+1) \right|, \\
\delta_{0} &= \left| x(i+1, j) - x(i-1, j) \right|, \\
\delta_{-45} &= \left| x(i+1, j+1) - x(i-1, j-1) \right|, \\
\delta_{-60} &= \left| x(i+1, j+2) - x(i-1, j-2) \right|.
\end{aligned} \tag{2}$$

$$\begin{aligned}
\lambda_{j-2} &= \left| x(i-1, j-2) - x(i+1, j-2) \right|, \\
\lambda_{j-1} &= \left| x(i-1, j-1) - x(i+1, j-1) \right|, \\
\lambda_{j} &= \left| x(i-1, j) - x(i+1, j) \right|, \\
\lambda_{j+1} &= \left| x(i-1, j+1) - x(i+1, j+1) \right|, \\
\lambda_{j+2} &= \left| x(i-1, j+2) - x(i+1, j+2) \right|.
\end{aligned} \tag{3}$$

We assume $\delta_{\min}$ and $\lambda_{\min}$ are obtained as Eqs. (4) and (5).

$$\lambda_{\min} = \min\left(\lambda_{j-2}, \lambda_{j-1}, \lambda_{j}, \lambda_{j+1}, \lambda_{j+2}\right), \tag{4}$$

$$\delta_{\min} = \min\left(\delta_{60}, \delta_{45}, \delta_{0}, \delta_{-45}, \delta_{-60}\right). \tag{5}$$

$$\begin{aligned}
&if\ \left\{\lambda_{\min}=\min\left(\lambda_{j-2},\lambda_{j-1},\lambda_{j},\lambda_{j+1},\lambda_{j+2}\right)\right\}\&\left\{\delta_{\min}=\min\left(\delta_{60},\delta_{45},\delta_{0},\delta_{-45},\delta_{-60}\right)\right\},\\
&\qquad then\ x_{prop}(i,j)=x_{filter}(i,j)\\
&otherwise,\\
&\qquad then\ x_{prop}(i,j)=\frac{x(i+1,j)+x(i-1,j)}{2}.
\end{aligned} \tag{5}$$

We denote the reconstructed results obtained by trained filter as $x_{filter}(i,j)$, which is obtained as Eq. (6),

$$x_{filter}(i,j)=imfilter\left(givenimage,h_{N\times N}\right) \tag{6}$$

where *imfilter* is Matlab command, *givenimage* is low-resolution input image, and *h* is obtained filter. Note that $h_{N\times N}$ is trained filter with $N\times N$ size, and $x_{filter}(i,j)$ is the filtered output image using given image and $h_{N\times N}$.

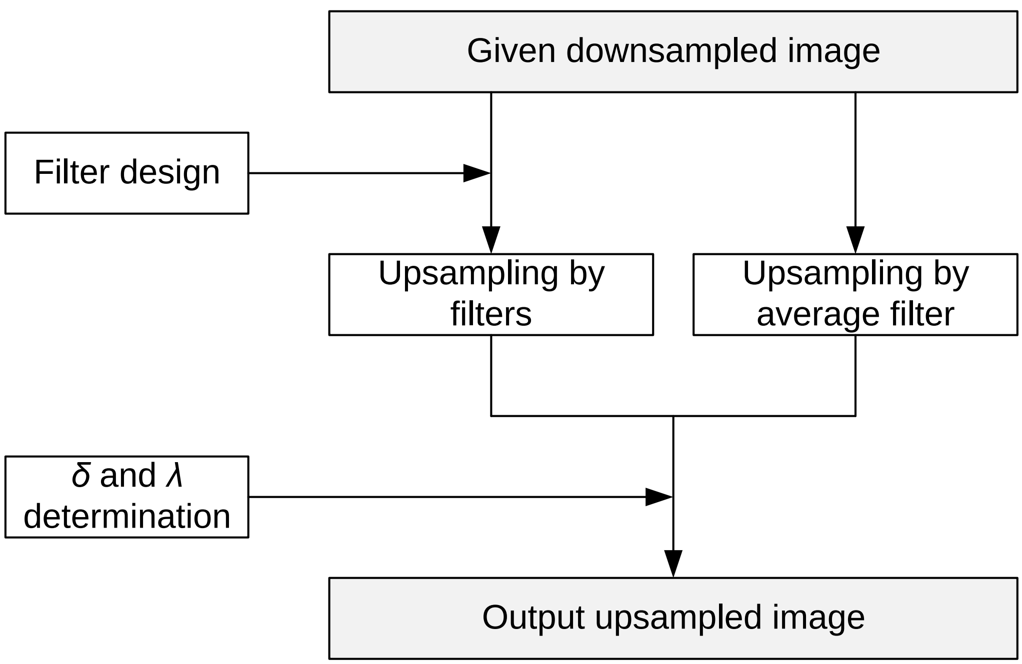
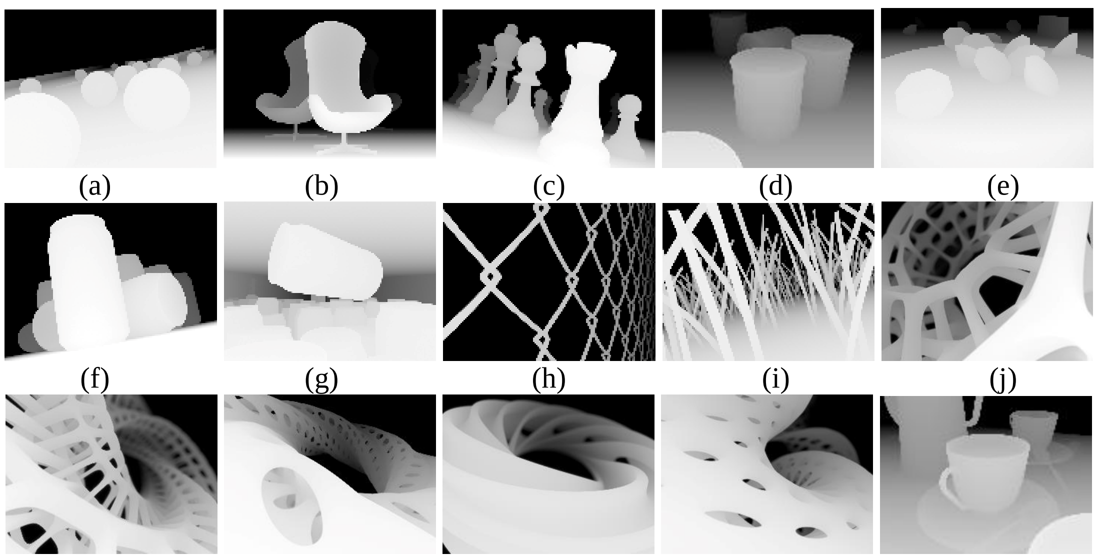


Figure 3. Block diagram of the proposed system

Finally, Figure 3 shows the flowchart of the proposed method. Recall that filter design process prevent real-time system, we conducted filter design process before the interpolation. In other words, using plenty number of dataset, the most appropriate filters are obtained [12]. There are two upsampling methods in this flowchart. One method is designed filter-based method and the other one is average filter-based method. Both results are quite similar to each other, thus final restored image should choose either results based on δ and λ determination. Based on the rules in Eq. (5), final results are determined.

4. Experimental Results

In this section, we compare the objective and subjective quality for the presented upsampling approach. We conducted an extensive simulation to test the performance of our method using an Intel(R) Core(TM) i3 CPU @2.4GHz. All methods were implemented in Matlab software, and were tested using our test depth map dataset. We ran our simulations on test depth map dataset, which are shown in Figure 4, and its original RGB images are shown in Figure 5. Note that the image size of Figure 4(a-i,o) are 720×540, and the others are 1440×1080.



(a) (b) (c) (d) (e)
(f) (g) (h) (i) (j)
(k) (l) (m) (n) (o)

Figure 4. Original test depth images: (a) billiards, (b) chairs, (c) chess, (d) cornbox, (e) diamonds, (f) diet_coke, (g) diet_pepsi, (h) fence, (i) hair, (j) math1, (k) math2, (l) math3, (m) math4, (n) math5, and (o) teacup.

Table 1 shows the PSNR result of different resampling methods for various sequences. The proposed method is compared with bilinear method and different sized filters including 3×3, 5×5, 7×7, 9×9, 11×11, and 13×13. From Table 1, we observed that our proposed method outperforms all other methods in all 15 images in terms of PSNR. The proposed method gives better PSNR results than the others by 6.359, 0.515, 0.512, 0.034, 0.036, 0.013, and 0.001 (dB). Table 2 shows the MSE comparison. As we can see, the proposed method gives the best performance in MSE by -2.554, -0.097, -0.096, -0.006, -0.007, -0.002, and -0.0004.

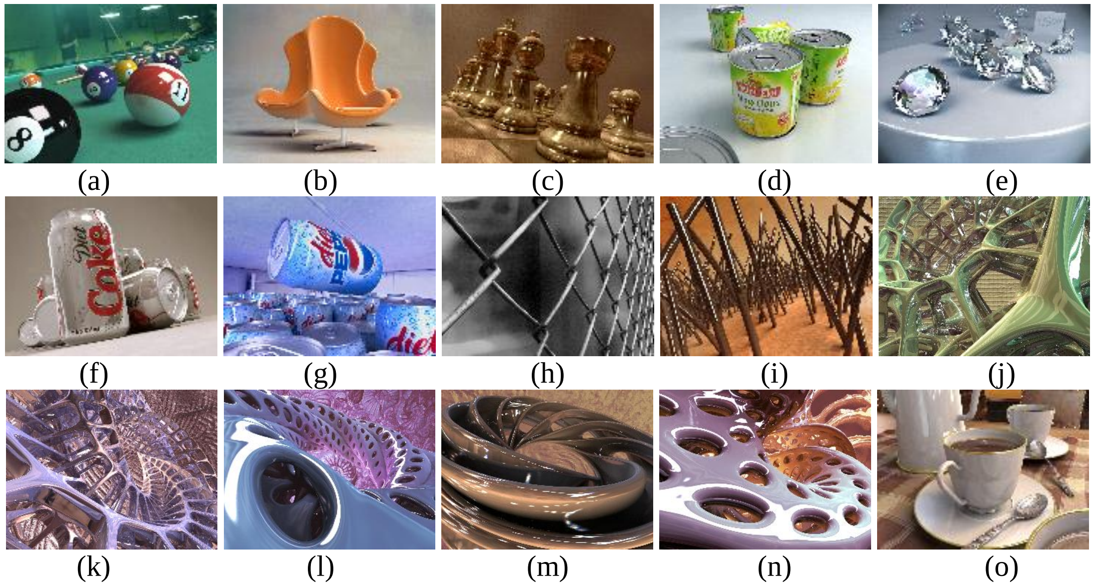


Figure 5. Original test optical images: (a) billiards, (b) chairs, (c) chess, (d) cornbox, (e) diamonds, (f) diet_coke, (g) diet_pepsi, (h) fence, (i) hair, (j) math1, (k) math2, (l) math3, (m) math4, (n) math5, and (o) teacup

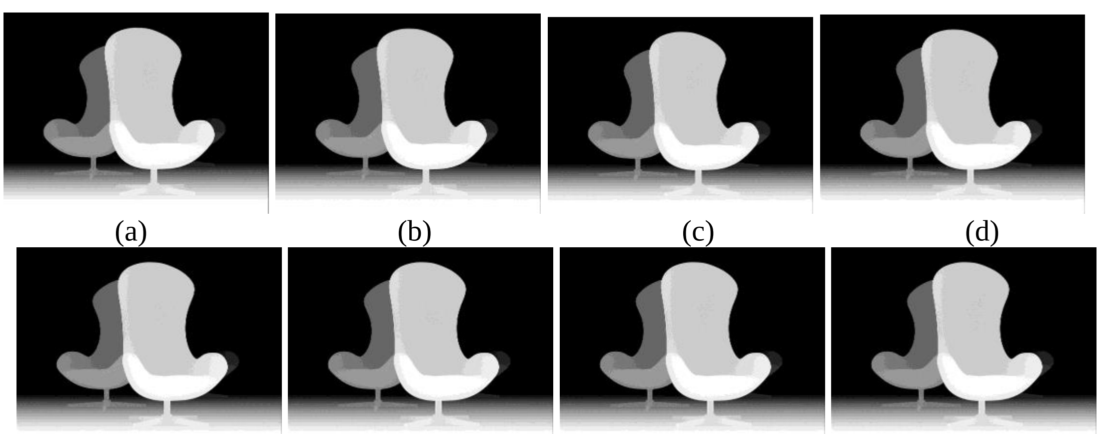
Table 1. PSNR comparison for filters with different size on fifteen depth images

	bilinear	3×3	5×5	7×7	9×9	11×11	13×13	Prop.
billiards	42.917	48.761	48.764	49.242	49.240	49.275	49.276	49.263
chairs	33.456	37.328	37.333	37.509	37.495	37.479	37.472	37.475
chess	31.940	38.576	38.619	39.977	39.981	40.106	40.104	40.118
cornbox	40.358	44.665	44.699	44.872	44.884	44.883	44.899	44.925
diamonds	41.496	46.525	46.572	46.846	46.858	46.928	46.957	46.995
diet_coke	33.646	38.433	38.531	39.027	39.091	39.228	39.378	39.488
diet_pepsi	41.281	46.206	46.268	46.716	46.700	46.721	46.710	46.712
fence	26.085	31.855	32.281	33.581	33.628	33.713	33.768	33.810
hair	23.714	29.559	29.567	30.481	30.526	30.617	30.645	30.691
math1	42.178	50.000	50.174	50.923	50.935	51.007	51.027	51.049
math2	41.003	48.443	48.768	49.376	49.384	49.446	49.463	49.478
math3	42.239	50.904	51.093	51.790	51.808	51.851	51.851	51.859
math4	50.812	58.285	58.435	58.330	58.330	58.335	58.332	58.322
math5	47.639	56.763	56.859	57.617	57.585	57.498	57.496	57.495
teacup	37.788	42.203	42.214	42.717	42.759	42.818	42.844	42.879
Average	42.917	48.761	48.764	49.242	49.240	49.275	49.276	49.263

Table 2. MSE comparison for filters with different size on fifteen depth images

	bilinear	3×3	5×5	7×7	9×9	11×11	13×13	Prop.
billiards	3.322	0.865	0.864	0.774	0.775	0.768	0.768	0.770
chairs	29.341	12.031	12.018	11.539	11.576	11.619	11.637	11.630
chess	41.598	9.026	8.937	6.537	6.530	6.346	6.349	6.328
cornbox	5.988	2.221	2.204	2.118	2.112	2.112	2.105	2.092
diamonds	4.608	1.447	1.432	1.344	1.340	1.319	1.310	1.299
diet_coke	28.087	9.328	9.119	8.135	8.016	7.768	7.503	7.316
diet_pepsi	4.842	1.558	1.536	1.385	1.390	1.384	1.387	1.386
fence	160.163	42.421	38.456	28.508	28.205	27.658	27.309	27.047
hair	276.483	71.979	71.840	58.206	57.614	56.409	56.055	55.464
math1	3.938	0.650	0.625	0.526	0.524	0.516	0.513	0.511
math2	5.161	0.931	0.863	0.751	0.749	0.739	0.736	0.733
math3	3.884	0.528	0.506	0.431	0.429	0.425	0.425	0.424
math4	0.539	0.097	0.093	0.096	0.096	0.095	0.095	0.096
math5	1.120	0.137	0.134	0.113	0.113	0.116	0.116	0.116
teacup	10.821	3.916	3.905	3.478	3.445	3.399	3.378	3.351
Average	3.322	0.865	0.864	0.774	0.775	0.768	0.768	0.770

The visual results for test depth map dataset also show that the proposed method is superior to the other methods. The proposed approach is more efficient in terms of visual quality. The proposed method does not give stairs-resembling artifacts, providing satisfactory visual quality. It is also found that the proposed method gives more effective visual quality with smoother edges. Figure 6 and 7 shows the results images for chairs and diet_coke images.



(a) (b) (c) (d)
(e) (f) (g) (h)

Figure 6. Reconstructed chairs images: (a) bilinear method, trained filters with different size (b) 3×3, (c) 5×5, (d) 7×7, (e) 9×9, (f) 11×11, (g) 13×13, and (h) proposed method

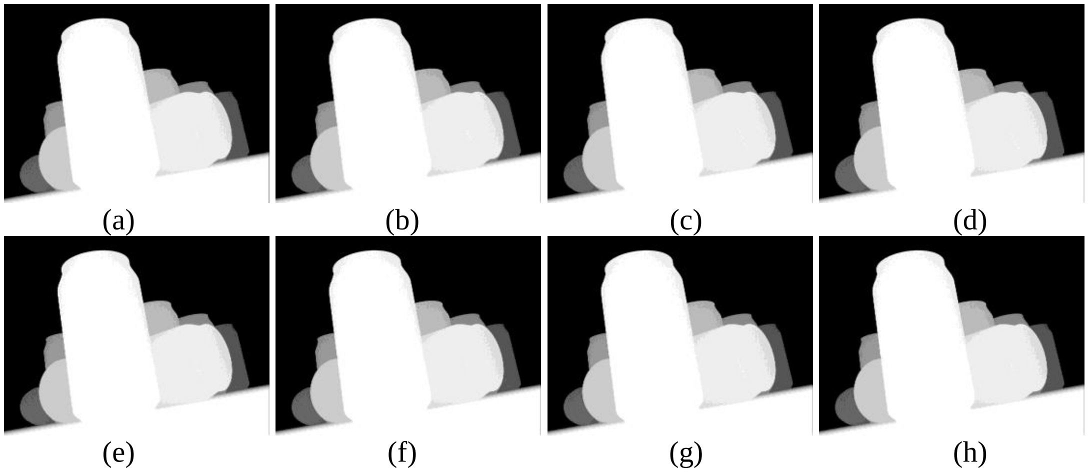


Figure 7. Reconstructed diet_coke images: (a) bilinear method, trained filters with different size (b) 3×3, (c) 5×5, (d) 7×7, (e) 9×9, (f) 11×11, (g) 13×13, and (h) proposed method

Figures 8 and 9 show reconstructed images with different filter size and the proposed method. As we can see, the proposed method gives the best visual performance with less outlier artifact. To clearly see the visual artifact, Figures 8 and 9 show the image difference between original images and the reconstructed one. As we can see, the results obtained by bilinear method are the poorest. As filter size increases, reconstructed images are better.

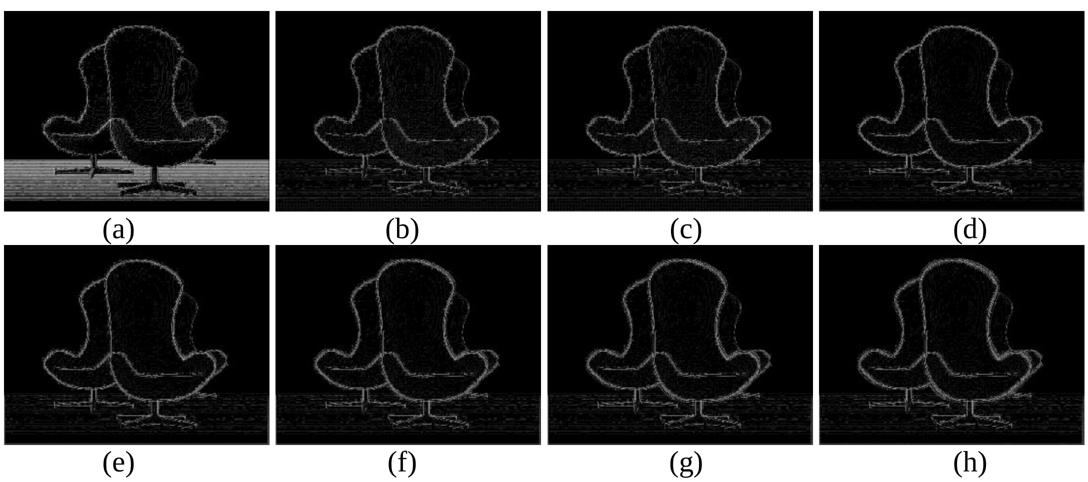
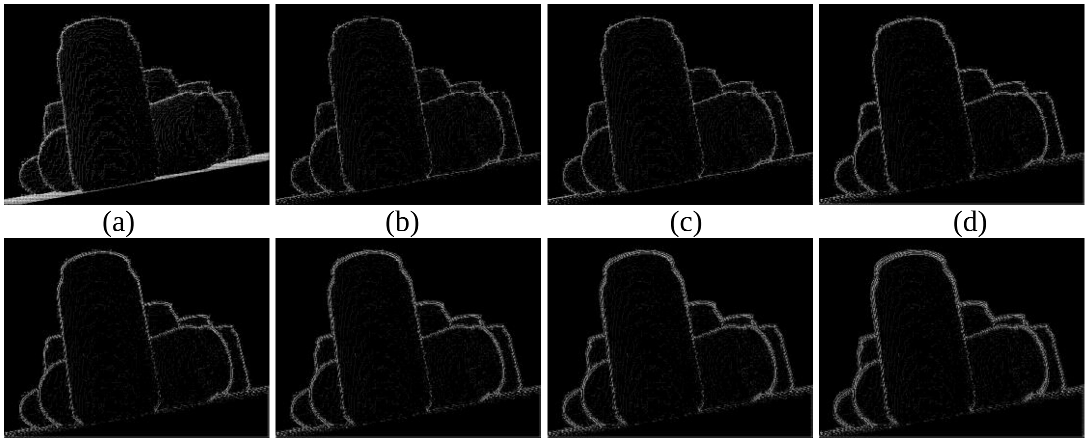


Figure 8. Difference chairs images between original and reconstructed images: (a) bilinear method, trained filters with different size (b) 3×3, (c) 5×5, (d) 7×7, (e) 9×9, (f) 11×11, (g) 13×13, and (h) proposed method



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 9. Difference diet_coke images between original and reconstructed images: (a) bilinear method, trained filters with different size (b) 3×3, (c) 5×5, (d) 7×7, (e) 9×9, (f) 11×11, (g) 13×13, and (h) proposed method

5. Conclusions

An image upsampling method has been introduced in this paper. The missing pixels are restored with two ways: filter based approach and bilinear approach. By means of two parameters δ and λ, one may choose interpolation approach. Experimental results show that the introduced approach provides much better performance than the other traditional methods.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning(2013R1A1A1010797).

References

[1] K. Jack, "Video Demystified A Handbook for the Digital Engineer", Elsevier, (**2005**).

[2] C. J. Veenman and M. J. T. Reinder, "The Nearest Subclass Classifier: A Compromise between the Nearest Mean and Nearest Neighbor Classifier", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 9, (**2005**) September, pp. 1417-1429.

[3] E. B. Bellers and G. de Haan, "Advanced de-interlacing techniques," in Proc. ProRisc/IEEE Workshop on Circuits, Systems and Signal Processing, Mierlo, The Netherlands, (**1996**) November, pp. 7-17.

[4] Y. -C. Fan, H. -S. Lin, H. -W. Tsao and C. -C. Kuo, "Intelligent intra-field interpolation for motion compensated deinterlacing," in Proc. ITRE 2005, vol. 3, **(2005)** June, pp. 200-203.

[5] H. K. Hong, H. H. Shin and I. J. Chung, "In-plane switching technology for liquid crystal display television", Journal of Display Technology, vol. 3, no. 4, (**2007**), pp. 361-370.

[6] J. -J. Lyu, J. Sohn, H. Y. Kim and S. H. Lee, "Recent trends on patterned vertical alignment (PVA) and fringe-field switching (FFS) liquid crystal displays for liquid crystal television applications", Journal of Display Technology, vol. 3, no. 4, (**2007**), pp. 404-412.

[7] S. -Y. Kim and Y. -S. Ho, "Fast edge-preserving depth image upsampler," IEEE Trans. Consumer Electron., vol. 58, no. 3, **(2012)** August, pp. 971-977.

[8] 3D-AVC Test Model 5, document JCT3V-C1003, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, **(2013)** January.

[9] 3D-HEVC Test Model 3, document JCT3V-C1005, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, **(2013)** January.

[10] L. P. J. Vosters, C. Varekamp and G. de Haan, "Evaluation of efficient high quality depth upsampling methods for 3DTV", in Proc. SPIE 8650, Three-Dimensional Image Processing and Applications, California, USA, **(2013)** March, pp. 865005.

[11] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in Proc. SPIE 5291, Stereoscopic Displays and Virtual Reality Systems XI, California, USA, **(2004)** May, pp. 93.

[12] http://www.dofpro.com/cgigallery.htm.

Author

Gwanggil Jeon

He 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 assistant 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.

