International Journal of Hybrid Information Technology
Vol. 9, No. 6 (2016), pp. 381-394
http://dx.doi.org/10.14257/ijhit.2016.9.6.34

Optimal Window Size for Gradient Information

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 scanning format conversion method has been developed rapidly. The traditional TV used interlaced scanning format to compensate motion artifacts. The converting process of interlaced signal into a progressive one is called scanning format conversion. Due to recently developed progressive devices, all interlaced scanned contents must be converted into progressive format. The directional information is widely used for scanning format conversion. In this paper, we assess and compare performances by varying widow size. The performances are evaluated in PSNR, MSE, and SSIM metrics.

Keywords: *Display, varying window size, directional information*

1. Introduction

The scanning format conversion (SFC) has been used at the start of TV broadcastings [1]. The SFC decreases the bandwidth (BW) by half, and it applies the spatial and temporal properties of human visual system: the human eyes are less sensitive to artifacts than to large area artifacts [2-4]. However, SFC has many steps and still being developed. For example, SFC is used in HDTV (high definition television), mobile phone internet and various formats of video [5]. Thus, there are needs for conversion between heterogeneous formats.

The interlaced signal is intended to be captured, stored, transmitted, and displayed in the same format [6]. Thus, both of interlaced and progressive scanning formats are in use for various purposes such as recording, transmitting and displaying motion pictures. Each frame has two fields (odd and even), which captured at different moments in time, interlaced signal frames can show motion artifacts (this effect is called as interlacing effects) [7-9]. Therefore, if an object moves quickly, unwanted artifacts are shown in a frame. These unwanted artifacts are comb-tooth artifacts at details (edges) of moving objects [10, 11]. To reduce the unwanted comb-tooth artifacts, we study optimal window size for edge direction based interpolation.

The SFC is an important approach for changing format. To display interlaced signal into progressive device, SFC process must be applied on interlaced signal before displaying it. The SFC increases the number of scanning lines in field, and therefore linear sampling rate conversion theory is enough to carry out the deinterlacing. The high upright components with motions and odd upright velocity cannot be completely restored, as for these velocities the issue becomes ill-posed. Therefore, it is impossible to solve the deinterlacing problem under all circumstances.

One of the most notable factors in analog TV is signal BW, which is represented in megahertz. The bigger the BW, the more cost and complicated the whole production and broadcasting chain. This approach is applied in cameras, storage systems, broadcast TV systems.

In this paper, we assess performance by varying window size. The paper is organized as follows. In Section 2, proposed approach is introduced. The proposed method is based on edge based line average method, where the window size is increased. Section 4

ISSN: 1738-9968 IJHIT

analyses the performance with regarding to the various sizes of window. Finally, our conclusions are provided in Section 5.

2. Proposed Method

The original frame is divided into two fields (interlacing). Then, the interlaced signal is vertically upsampled by deinterlacing method. These two processes are shown in Figure 1.

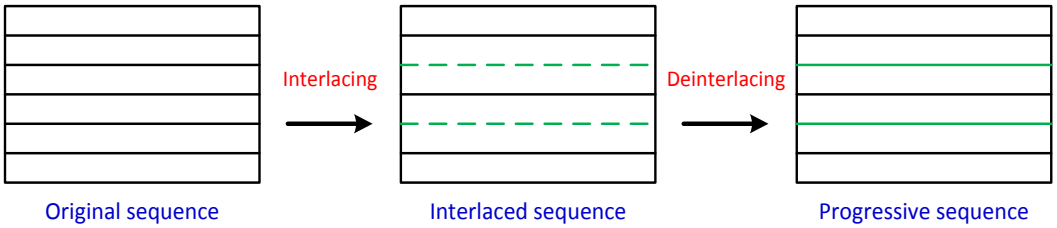

Figure 1. Basic Process of Interlacing and Deinterlacing

There are several filters that are based on linear interpolation. Table 1 shows various filters for linear interpolation. These filters are linear interpolation methods.

Table 1. Filters based on Linear Interpolation

Filters	Number of Taps
First order linear interpolator	2 taps
Catmull-Rom cubic interpolator	4 taps
Polyphase filter interpolator	6 taps

Equation (1) shows an example of 2-tap linear interpolation filter.

$$x_{LA}(i) = \frac{U(i) + L(i)}{2}, \tag{1}$$

where $U(i)$ and $L(i)$ are pixels located in upper and lower lines, and $x(i)$ is pixel to be interpolated. To restore a missing pixel, Eq (1) averages both pixels in previous line and subsequent line.

There are also non-linear interpolation methods. In general, non-linear interpolation filter produces sharp edges. Following equation is an example of one-dimensional non-linear interpolation filter. For example, edge-based line average (ELA) method utilizes directional correlation among adjacent pixels to linearly restore the missing pixels. Equation (2) show how directional correlation is calculated,

$$C(k) = \left| U(i+k) - L(i-k) \right|, \tag{2}$$

where k is displacement parameters and integer number. Now, the direction of the highest correlation is calculated as

$$\theta = \arg\min \{ C(k) \} \tag{3}$$

Finally, the missing pixel is computed as

$$x_{ELA}(i) = \frac{U(i+\theta) + L(i-\theta)}{2}, \tag{4}$$

Another non-linear method to restore a missing pixel is ELA based static texture region comparison method. Pixel difference $D(k)$ is calculated as

$$D(k) = \left| U(i-k) + L(i+k) \right|, \tag{5}$$

where k={-2,-1,0,1,2}. Now, P_{c1}, P_{c2}, and P_{c3} are calculated as

$$\begin{aligned} P_{c1} &= \frac{U(i-k) + L(i+k)}{2}, \\ P_{c2} &= \frac{U(i) + L(i)}{2}, \\ P_{c3} &= \frac{U(i+k) + L(i-k)}{2}, \end{aligned} \tag{6}$$

Now, we find the median value among P_{c1}, P_{c2}, and P_{c3} and using this value for restoration.

In our work, we use adaptive deinterlacing with edge-based median filtering approach. Figure 2(a) shows the proposed method. Figure 2(b) shows an example of 3x3 window, where k is 1. Figure 2(c) shows window size corresponding to various k.

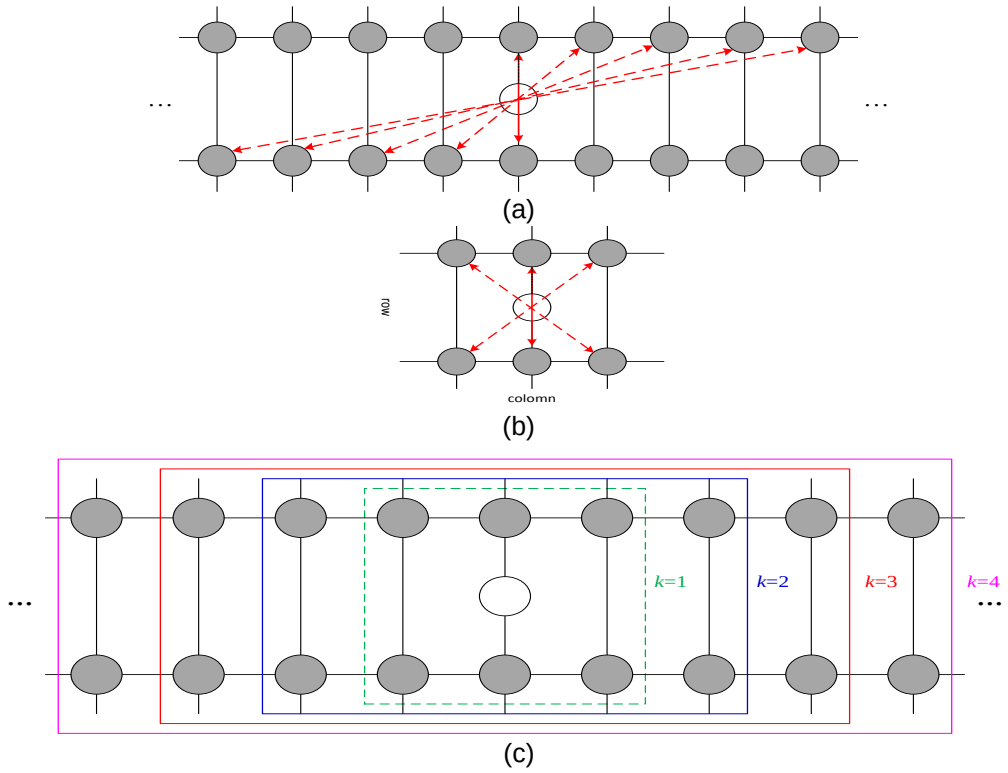

Figure 2. (a) Missing Pixel Calculation using Directional Interpolation. Pixels in Gray is Existing Pixels. Pixel in White is Missing Pixel, (b) An Example of 3x3 window, where *k* is 1, and (c) Window Size Corresponding to Varying *k*

Our proposed method enlarges filter size and compute edge directions. Equation (7) shows how the system restore the missing pixel:

$$P = \frac{U(i-k) + L(i+k)}{2}, \tag{7}$$

where k is integer number, ranging from $-N$ to N. The missing pixel is calculated as

$$x_{RES}(i) = \frac{U(i+k) + L(i-k)}{2}, \tag{8}$$

where k is the selected k from Eq. (7). We tested $|k|$ ranging from 0 to 5. Figure 3 shows General structure of interlacing and deinterlacing processes. It is noted that interlaced images has half vertical resolution of its original images. Figure 4 shows SSIM result images in still Football image. Figure 5 shows final deinterlaced images when k=0 and k=5.

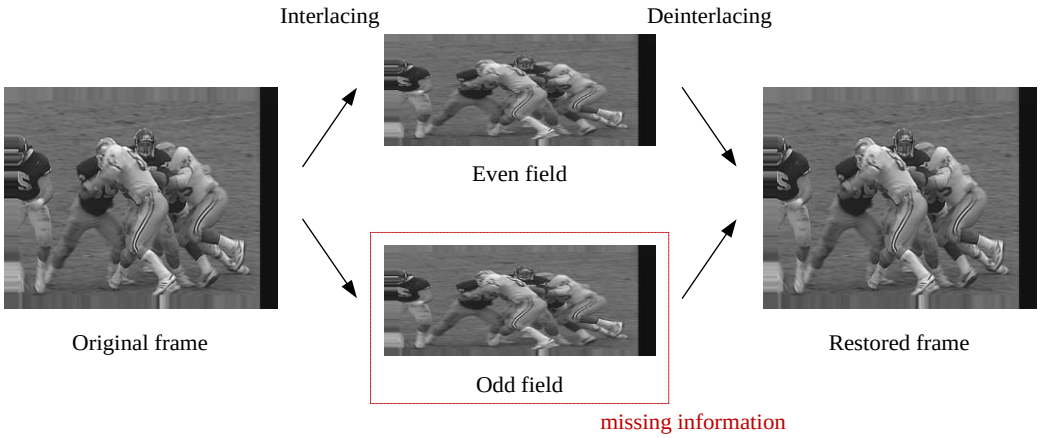

Figure 3. General Structure of Interlacing and Deinterlacing Processes

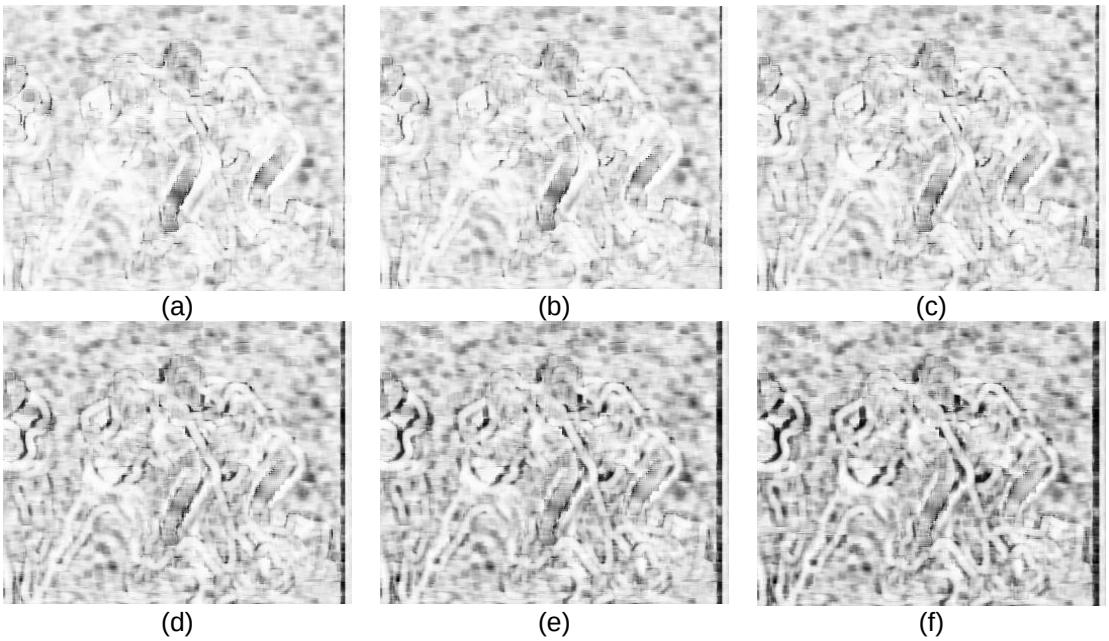

Figure 4. SSIM images in Football Sequence: *k*=0, *k*=1, *k*=2, *k*=3, *k*=4, and *k*=5

(a)

(b)

Figure 5. Two Results Images when (a) *k*=0 and (b) *k*=5

3. Performance Analysis

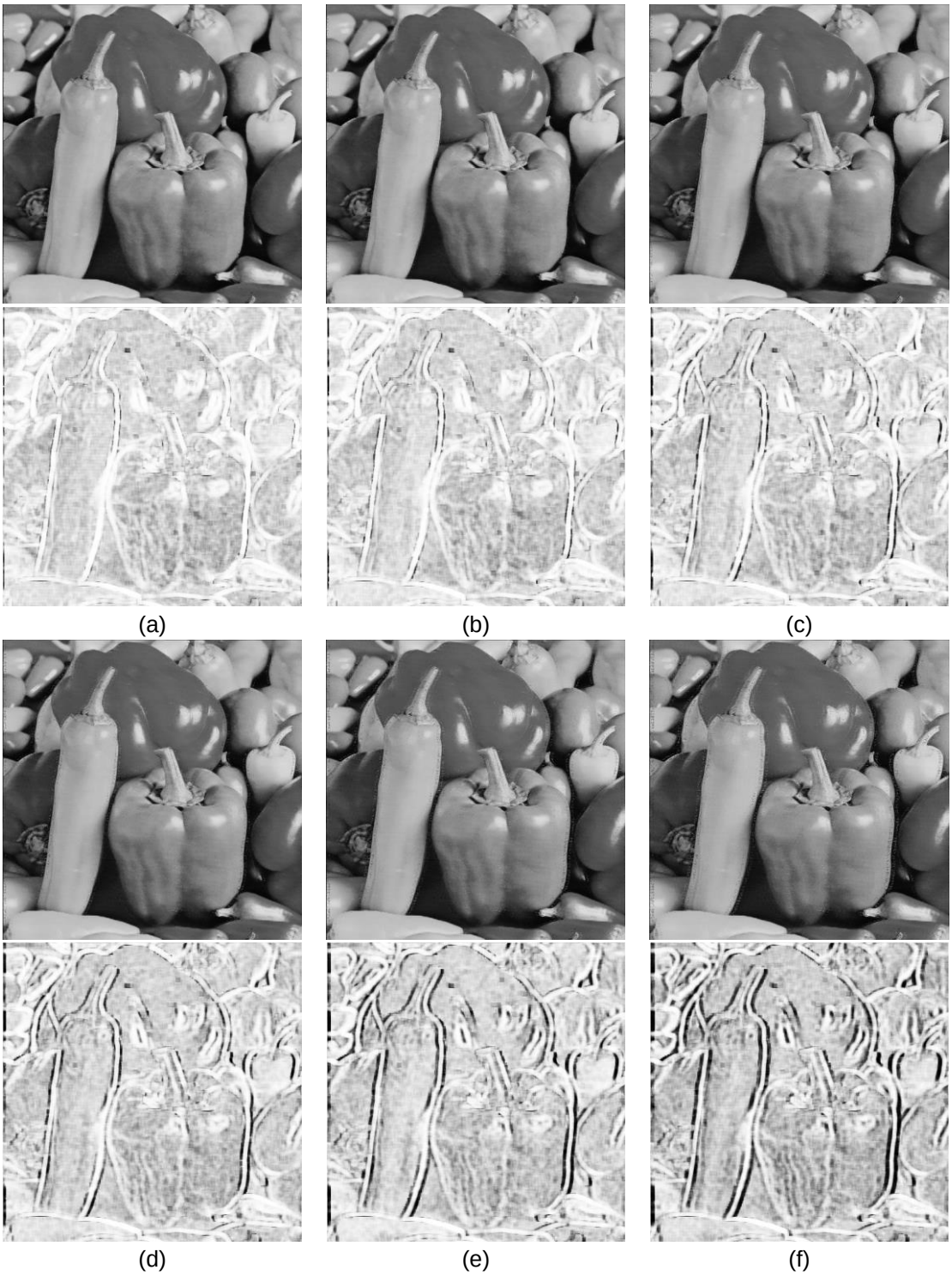

Figure 6. Visual Quality Comparison in Pepper Sequence with Difference Size of *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

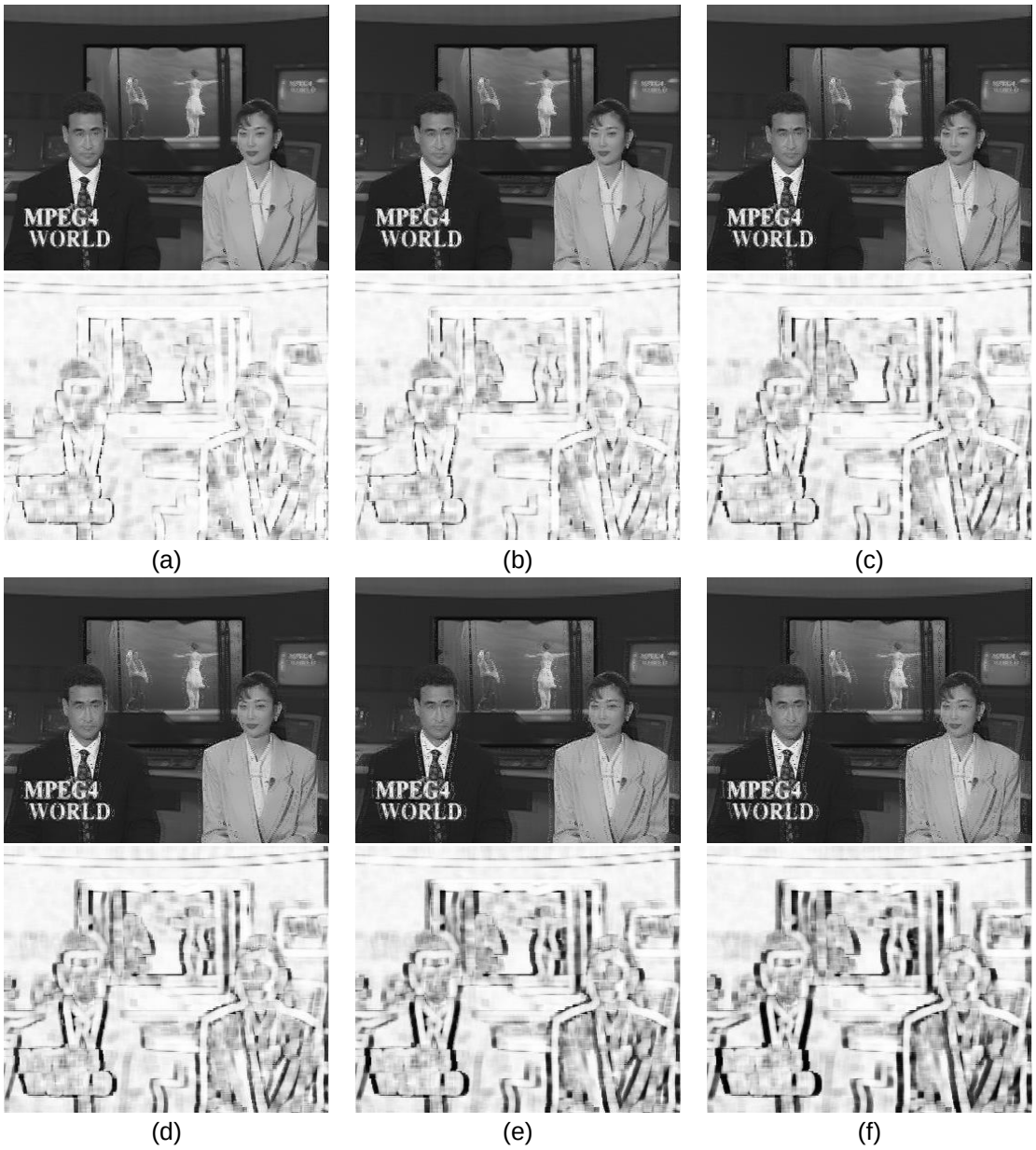

Figure 7. Visual Quality Comparison in News sequence with Difference Size of *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

In order to show significance of our window size assessment, we evaluate both of objective and subjective performances. We used 24 black and white images and their sizes were 1920×1080, 1280×720, 512×512, and 352×288. We used two objective performance metrics: PSNR, MSE, and SSIM metrics and the experiments were conducted on the Intel Core 2 Duo CPU E8500 @ 3.16 GHz. The SSIM is called as structural similarity index, which is a tool for assessing the restored image quality between two inputs. The SSIM is a full reference index. Therefore, the restored image assessment is based on an original image as a reference.

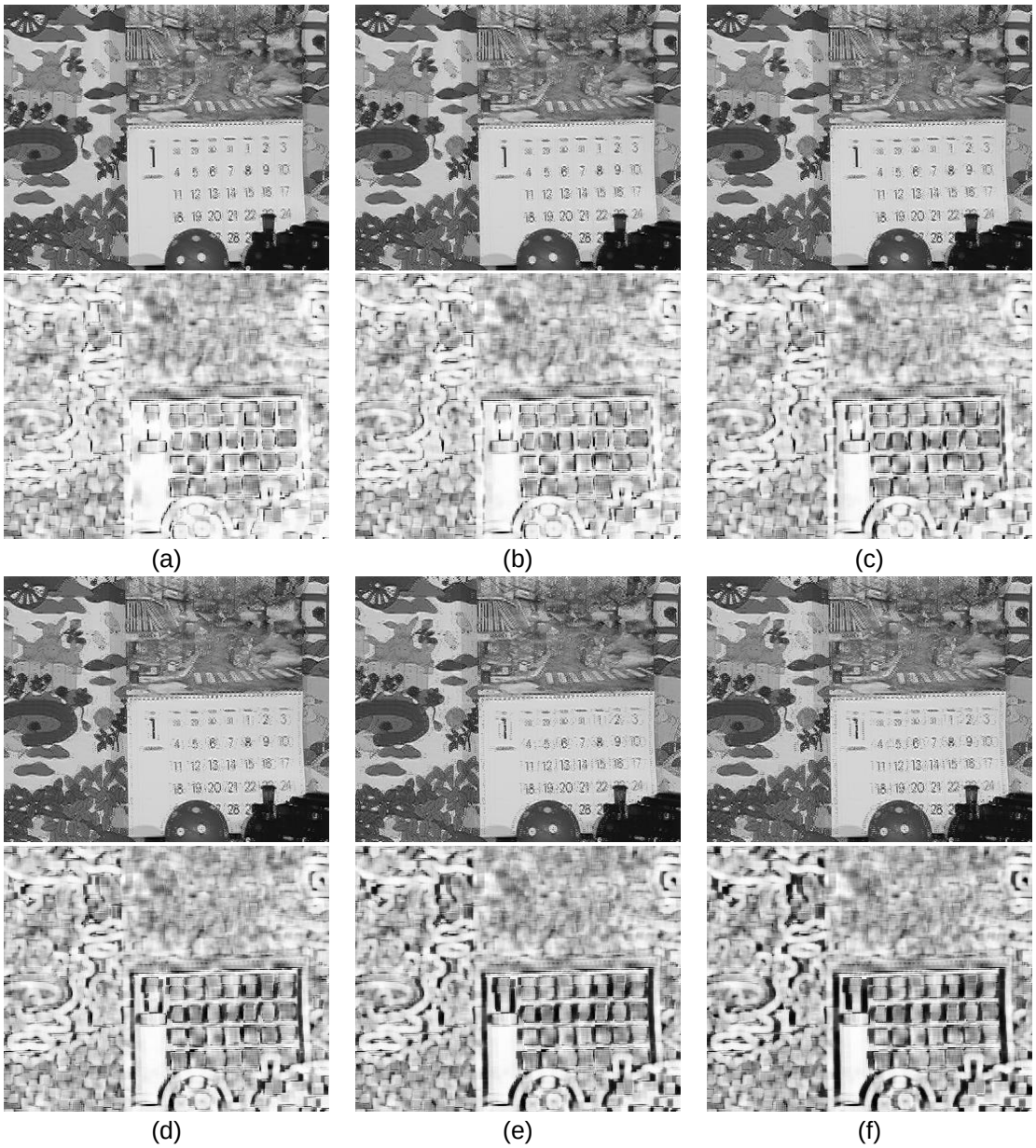

Figure 8. Visual Quality Comparison in Toy Sequence with Difference Size of *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

In this simulation, we compared six window sizes (*ws*): *k*=0, *k*=1, *k*=2, *k*=3, *k*=4, and *k*=5. Figures 6-9 shows the results of size *ws* on Pepper, News, Toys, and Milkdrop. It can be found that bigger *ws* parameters do not always give the best performance. As *ws* increases wrong directional information can be used, and unwanted artifacts can be used. Objective performance can be found in Tables 2, 3, and 4. Table 2 shows PSNR results for six *k* values. It can be found that the PSNR decreases as window size increases.

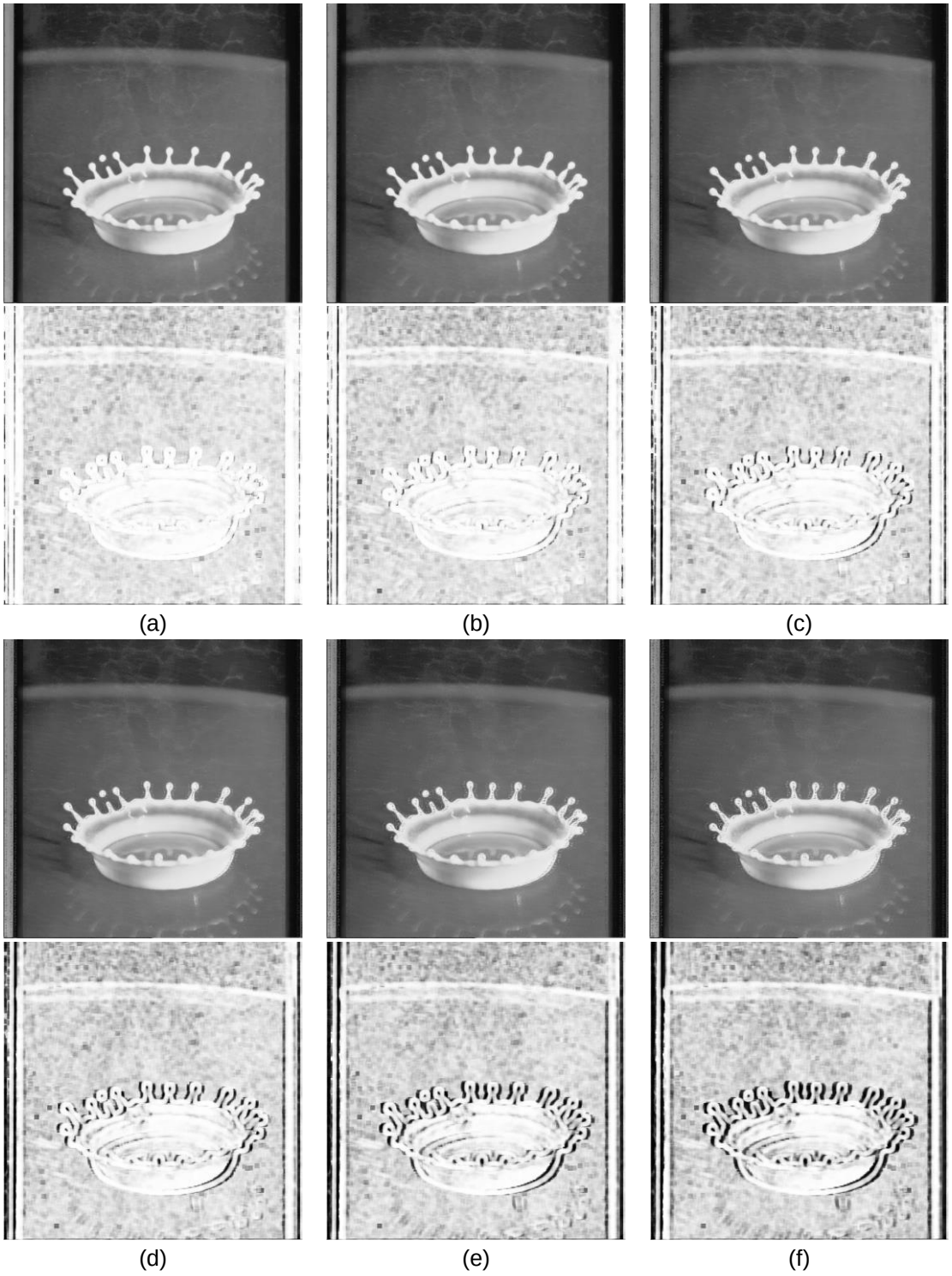

Figure 9. Visual Quality Comparison in Milkdrop Sequence with Difference Size of *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

Tables 3 and 4 show MSE and SSIM results for six *k* values. Similarly, performances became worse as window size increases. Results in Tables 2-4 are depicted in Figures 10-12, where six *ws* scenario results are displayed.

Table 2. PSNR Performance Comparison for Various *k*

Image number	*k*=0	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5
1	30.605	30.106	29.311	28.486	27.657	26.990
2	32.855	32.289	31.593	30.772	29.954	29.266
3	22.890	22.817	22.710	22.586	22.466	22.343
4	26.274	26.212	25.891	25.482	25.106	24.782
5	22.284	21.885	21.428	21.028	20.633	20.331
6	34.355	33.945	33.061	31.896	30.679	29.550
7	29.093	28.688	28.151	27.525	27.003	26.546
8	24.617	24.447	24.116	23.807	23.555	23.346
9	28.942	28.676	28.417	28.142	27.854	27.552
10	24.838	24.521	24.055	23.612	23.296	23.094
11	28.937	28.704	28.412	27.987	27.410	26.901
12	34.849	33.745	32.472	31.104	29.974	29.222
13	31.709	31.025	30.127	29.056	28.244	27.529
14	41.727	40.655	39.079	37.347	35.835	34.518
15	32.017	31.240	30.295	29.356	28.471	27.683
16	26.615	26.268	25.831	25.357	24.836	24.375
17	37.611	35.877	33.838	32.101	30.723	29.605
18	21.466	21.388	21.240	21.008	20.716	20.425
19	27.413	26.907	26.317	25.632	25.180	24.788
20	32.756	32.333	31.754	31.132	30.516	29.944
21	32.860	32.228	31.248	30.153	29.128	28.073
22	35.079	34.374	33.515	32.805	32.146	31.560
23	32.130	31.522	30.526	29.421	28.343	27.376
24	36.602	35.830	34.822	33.776	32.787	31.838
Avg.	30.355	29.820	29.092	28.315	27.605	26.985

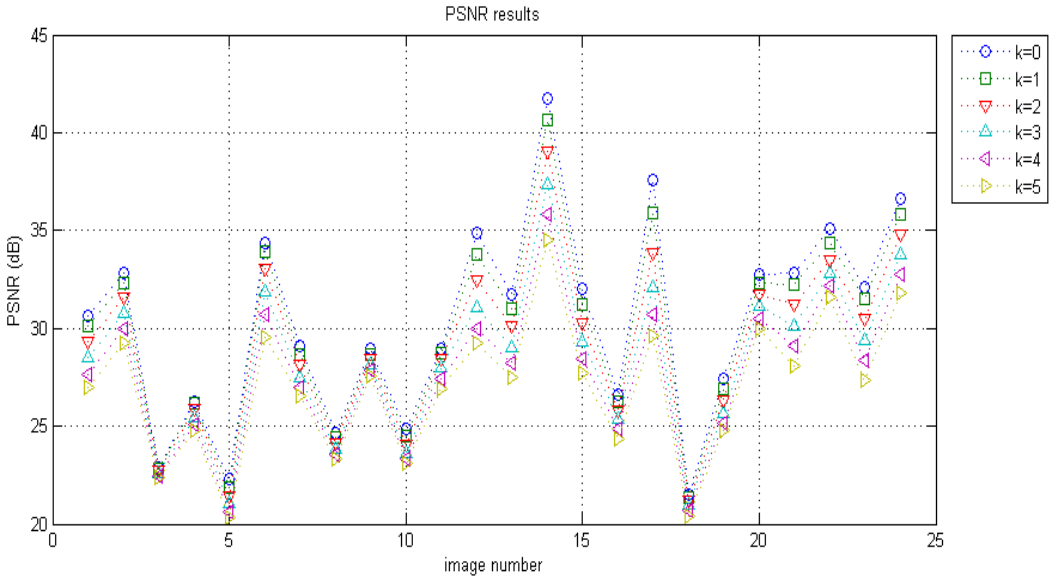

Figure 10. PSNR Results Comparison for six *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

Table 3. MSE Performance Comparison for Various *k*

Image number	*k*=0	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5
1	56.57	63.45	76.20	92.14	111.52	130.03
2	33.70	38.39	45.06	54.43	65.72	77.00
3	334.27	339.94	348.37	358.53	368.55	379.09
4	153.36	155.57	167.48	184.03	200.67	216.23
5	384.33	421.32	468.05	513.19	562.08	602.51
6	23.85	26.22	32.14	42.03	55.61	72.13
7	80.13	87.95	99.54	114.97	129.65	144.04
8	224.58	233.56	252.07	270.66	286.82	300.96
9	82.95	88.21	93.62	99.74	106.57	114.26
10	213.44	229.59	255.60	283.03	304.43	318.92
11	83.06	87.63	93.73	103.37	118.05	132.75
12	21.29	27.46	36.80	50.43	65.41	77.78
13	43.87	51.35	63.14	80.82	97.42	114.86
14	4.37	5.59	8.04	11.98	16.96	22.98
15	40.87	48.87	60.76	75.41	92.46	110.87
16	141.77	153.57	169.83	189.39	213.54	237.43
17	11.27	16.80	26.87	40.08	55.06	71.22
18	463.96	472.37	488.77	515.55	551.46	589.69
19	117.97	132.56	151.84	177.77	197.26	215.89
20	34.47	38.00	43.42	50.11	57.74	65.88
21	33.66	38.93	48.79	62.78	79.48	101.34
22	20.19	23.75	28.95	34.08	39.67	45.40
23	39.82	45.81	57.61	74.30	95.24	118.98
24	14.22	16.99	21.42	27.25	34.23	42.59
Avg.	110.75	118.49	130.75	146.09	162.73	179.28

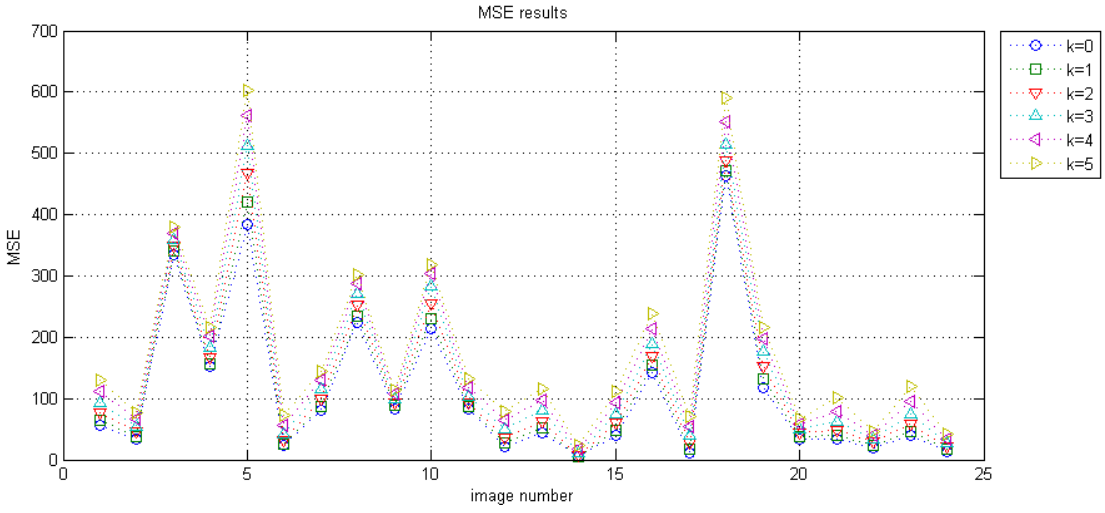

Figure 11. MSE Results Comparison for Six *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

Table 4. SSIM Performance Comparison for Various *k*

Image number	*k*=0	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5
1	0.9075	0.8938	0.8745	0.8536	0.8327	0.8149
2	0.9378	0.9244	0.9006	0.8725	0.8449	0.8190
3	0.7372	0.7282	0.7173	0.7039	0.6916	0.6783
4	0.8549	0.8434	0.8221	0.7973	0.7689	0.7434
5	0.8254	0.8003	0.7718	0.7422	0.7134	0.6898
6	0.9595	0.9540	0.9410	0.9210	0.8977	0.8733
7	0.8512	0.8374	0.8207	0.7990	0.7794	0.7614
8	0.7950	0.7846	0.7661	0.7458	0.7280	0.7117
9	0.8336	0.8183	0.8033	0.7872	0.7689	0.7507
10	0.8440	0.8323	0.8153	0.7979	0.7855	0.7762
11	0.8794	0.8662	0.8446	0.8177	0.7884	0.7613
12	0.9154	0.8895	0.8589	0.8217	0.7881	0.7685
13	0.8463	0.8342	0.8217	0.8042	0.7916	0.7799
14	0.9512	0.9403	0.9217	0.8935	0.8646	0.8333
15	0.8929	0.8772	0.8529	0.8247	0.7945	0.7641
16	0.8168	0.8015	0.7807	0.7570	0.7316	0.7082
17	0.8927	0.8748	0.8503	0.8233	0.7980	0.7766
18	0.7813	0.7704	0.7486	0.7232	0.6992	0.6788
19	0.9079	0.8920	0.8664	0.8356	0.8089	0.7861
20	0.8732	0.8582	0.8383	0.8151	0.7932	0.7710
21	0.8581	0.8535	0.8354	0.8081	0.7919	0.7467
22	0.9307	0.9142	0.8917	0.8679	0.8429	0.8191
23	0.9008	0.8874	0.8618	0.8371	0.8078	0.7861
24	0.9414	0.9305	0.9128	0.8916	0.8663	0.8380
Avg.	0.8723	0.8586	0.8383	0.8142	0.7907	0.7682

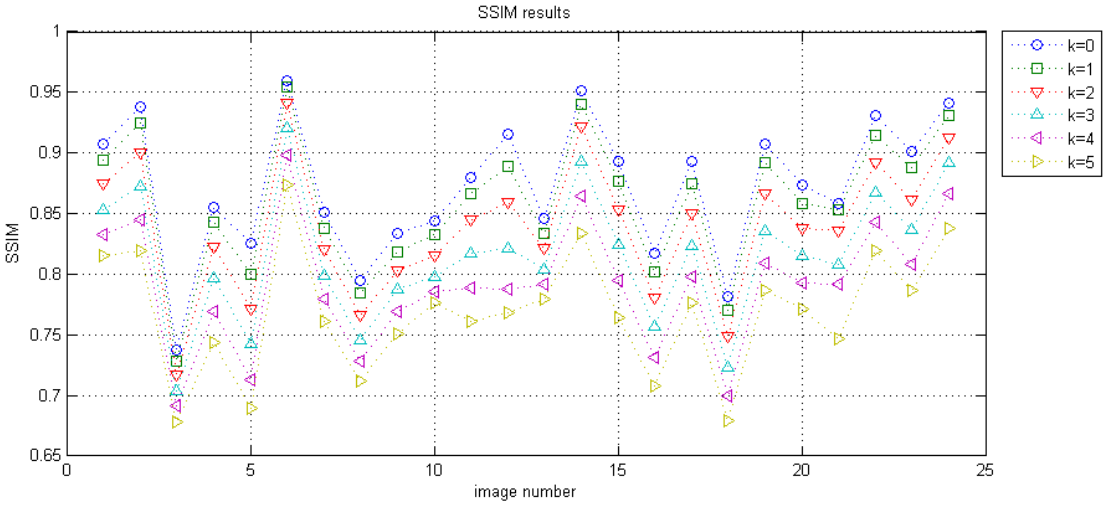

Figure 12. SSIM Results Comparison for Six *ws*: (a) *k*=0, (b) *k*=1, (c) *k*=2, (d) *k*=3, (e) *k*=4, and (f) *k*=5

4. Conclusions

In this paper, we study scanning format conversion method that uses directional information. The traditional TV systems use interlaced scanning format, while the recently developed devices use progressive scanning format. Therefore all contents must be transformed into progressive format before being displayed. The directional information is widely used for scanning format conversion. In this paper, we varied window size and assessed and compared objective and visual performances.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) Grant funded by the Korean Government(MSIP)(2014025627).

This paper is a revised and expanded version of a paper entitled "Window Size Arrangement using Directional Information" presented at CIA2016.

References

[1] T. Doyle, "Interlaced to sequential conversion for EDTV applications", Proceedings 2nd Int. Workshop Signal Processing of HDTV, **(1998)**, pp. 412-430.
[2] K. Jack, "Video Demystified - A Handbook for the Digital Engineer", 4th ed., Elsevier, Jordan Hill, Oxford, **(2005)**.
[3] E. B. Bellars and G. De Haan, "Deinterlacing: A Key Technology for Scan Rate Conversion", Elsevier, Amsterdam, **(2000)**.
[4] G. De Haan, "Television display processing: past & future", Proceedings of IEEE ICCE'07, Las Vegas, **(2007)**, pp. 1-2.
[5] K. Kang, G. Jeon and J. Jeong, "A single field interlaced to progressive format conversion using edge map in the image block", Proceedings of IASTED SIP 2009, Hawaii, USA, **(2009)**, pp. 80-85.
[6] P.-Y. Chen and Y.-H. Lai, "A low-complexity interpolation method for deinterlacing", IEICE Trans. Inf. Syst. E90-D(2), **(2007)**, pp. 606-608.
[7] F. Michaud, C. T. Le Dinh, and G. Lachiver, "Fuzzy detection of edge direction for video line doubling", IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 3, pp. 539-542, **(1997)**.
[8] Y.-L. Chang, C.-Y. Chen, S.-F. Lin and L.-G. Chen, "Motion compensated de-interlacing with adaptive global motion estimation and compensation", IEEE Conference on Image Processing, **(2003)** September.
[9] H. Y. Lee, J. W. Park, T. M. Bae, S. U. Choi and Y. H. Ha, "Adaptive scan rate up-conversion system based on human visual characteristics", IEEE Transactions on Consumer Electronics, vol. 46, **(2000)** November.
[10] G. Jeon, "Contrast Intensification in NTSC YIQ", IJCA, vol. 6, no. 4, **(2013)** August, pp. 157-166.
[11] G. Jeon, "Measuring and Comparison of Edge Detectors in Color Spaces", IJCA, vol. 6, no. 5, **(2013)** October, pp. 21-30.

Authors

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.