DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-279

Development of an 8K UHDTV Demosaicing Processor Using Adaptive Interpolation Based on Local Edge Magnitude

Noriyuki Shirai and Yukihiro Nishida; NHK Science & Technology Research Laboratories; Tokyo, Japan

Abstract

Most single-chip cameras acquire color images using a color filter array called the Bayer filter. The raw output of these cameras, a Bayer pattern image, needs color demosaicing to generate a full-RGB image. We propose a novel color demosaicing algorithm suitable for real-time processing. This method uses an adaptive interpolation filter based on local directional anisotropy and local edge magnitude, and it shows good performance comparable to that of conventional demosaicing algorithms including state-of-the-art algorithms. Further, we develop a real-time demosaicing processor using the proposed algorithm for a practical 8K camera system.

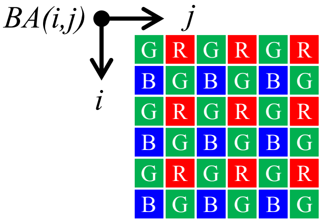
Introduction

8K Ultra-High Definition Television (8K) camera systems have been developed. Figure 1 shows 8K cameras [1-3] whose raw output is a Bayer pattern [4] image as shown in Figure 2. A Bayer pattern image has one of the three color components (R, G, and B) at each pixel and needs to be interpolated to a full-RGB image for subsequent post-production and broadcasting. This interpolating process is called “color demosaicing.” A high-performance demosaicing algorithm that can be implemented in a real-time processor is necessary for real-time applications. While there are a number of demosaicing algorithms, the state-of-the-art algorithms include iteration that is difficult to implement in a real-time processor.

(a)33M-pixel single-chip sensor[1] *(b)33M-pixel single-chip sensor[2]*

(c) Four-8M-pixel sensor[3]

Figure 1. *8K cameras that require demosaicing*



G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G

Figure 2. *Bayer pattern image*

Demosaicing method

The proposed demosaicing method uses a Bayer pattern image as input data and reconstructs a full-RGB image as follows (see Figure 3):

Step 1: Estimation of local directional anisotropy
Step 2: Estimation of local edge magnitude
Step 3: Interpolation of the green component
Step 4: Interpolation of the red and blue components.

We elaborate on the details of each of these steps below.

Step 1: Estimation of local directional anisotropy

We use the anisotropy-scale-mixture method [5] for the following interpolation calculation, as shown in Figure 4. The method focuses on local directional anisotropy with regard to perceived natural image statistics. First, we apply a set of simple linear filters to the input data to extract local directional features at multiple spatial scales. Horizontal features H_1 and H_2, and vertical features V_1 and V_2, are derived by the following formulas.

$$V_1(i,j) = |F_1 * BA(i,j)| \tag{1}$$

$$H_1(i,j) = |{F_1}^T * BA(i,j)| \tag{2}$$

$$F_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \tag{3}$$

$$V_2(i,j) = |F_2 * BA(i,j)| \tag{4}$$

$$H_2(i,j) = |{F_2}^T * BA(i,j)| \tag{5}$$

$$F_2 = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \tag{6}$$

where T is a transpose matrix, $BA(i,j)$ is the input data at the i-th column and j−th row, and F_1 and F_2 are simple linear filters. Next, we calculate an anisotropy parameter λ from the local directional features V_1, H_1, V_2, and H_2, as shown below.

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-279

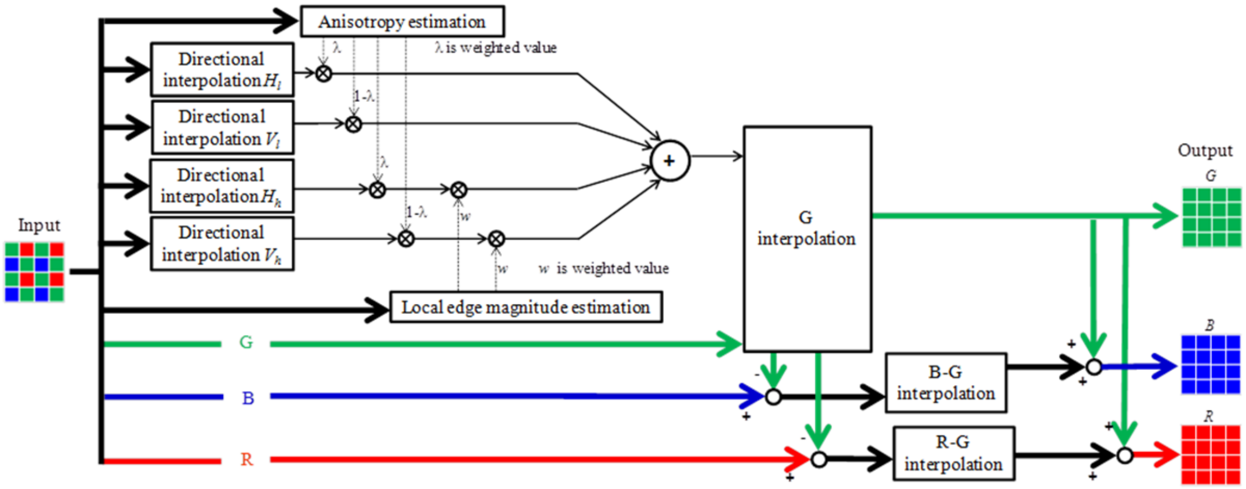


Figure 3. Demosaicing process of the proposed method

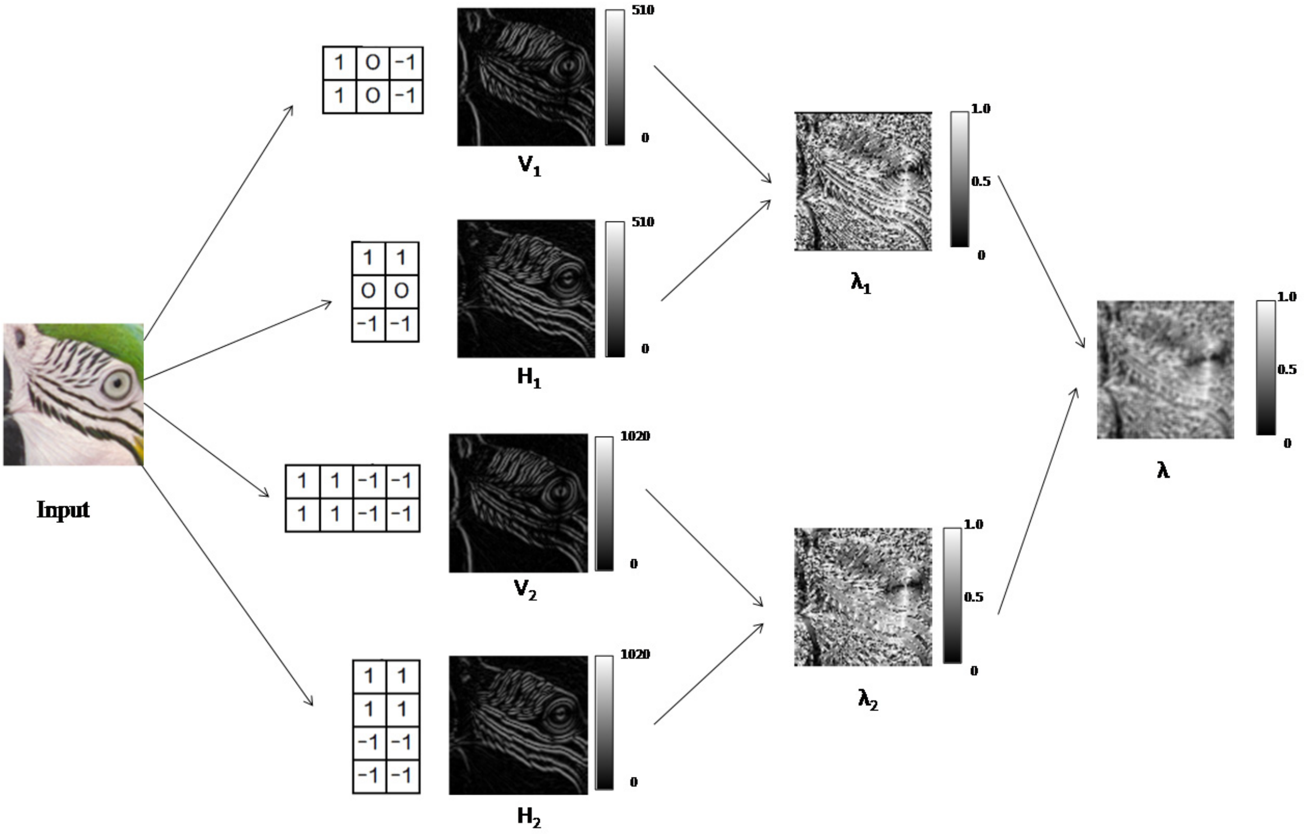


Figure 4. Calculation of λ

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-279

$$\lambda_1(i,j) = \frac{H_1(i,j)}{H_1(i,j)+V_1(i,j)+c} \tag{7}$$

$$\lambda_2(i,j) = \frac{H_2(i,j)}{H_2(i,j)+V_2(i,j)+c} \tag{8}$$

$$\lambda(i,j) = \{A * \lambda_1(i,j) + A * \lambda_2(i,j)\}/2 \tag{9}$$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}/9 \tag{10}$$

where c is a small constant value used to avoid division by zero. λ indicates a bias toward the horizontal component when $0.5 < \lambda \leq 1$ and a vertical bias when $0 \leq \lambda < 0.5$. Equation (9) calculates a spatial average for robust directional estimation.

Step 2: Estimation of local edge magnitude

We use a method based on the Kirsch operator [6] with modifications to estimate the local edge magnitude. We calculate local edge magnitude M_{dir} for eight directions using a set of simple 2D linear filters $F_{\mathrm{dir}}(\mathrm{dir{:}\,E, NE, N, NW, W, SW, S, SE})$ (Figure 5), and we calculate a weighting parameter w to be used for the subsequent interpolation, as shown below.

$$M_{\mathrm{dir}}(i,j) = |F_{\mathrm{dir}} * BA(i,j)| \tag{11}$$

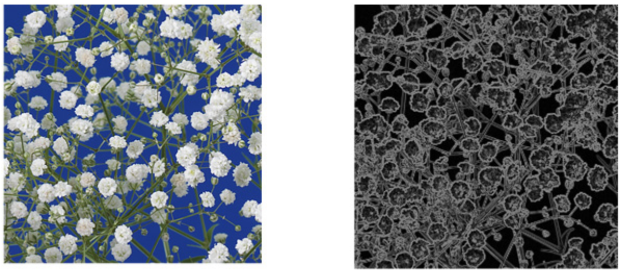
$$w(i,j) = \begin{cases} 1 & (M_{\max}(i,j) \geq \mathrm{thr}) \\ M_{\max}(i,j)/\mathrm{thr} & (M_{\max}(i,j) < thr) \end{cases} \tag{12}$$

where $M_{\max}$ is the maximum value of $M_{\mathrm{dir}}(\mathrm{dir{:}\,E, NE, N, NW, W, SW, S, SE})$ and *thr* is a constant. The optimal value of *thr* (approximately 16% of the maximum possible value of $M_{\max}$) has been experimentally derived using various test images. Figure 6 shows a calculation example of *w*.

$$F_{\mathrm{E}} \begin{bmatrix} -3 & -2 & 3 \\ -2 & 0 & 6 \\ -3 & -2 & 3 \end{bmatrix} \quad F_{\mathrm{NE}} \begin{bmatrix} -2 & 3 & 6 \\ -3 & 0 & 3 \\ -2 & -3 & -2 \end{bmatrix} \quad F_{\mathrm{N}} \begin{bmatrix} 3 & 6 & 3 \\ -2 & 0 & -2 \\ -3 & -2 & -3 \end{bmatrix} \quad F_{\mathrm{NW}} \begin{bmatrix} 6 & 3 & -2 \\ 3 & 0 & -3 \\ -2 & -3 & -2 \end{bmatrix}$$

$$F_{\mathrm{W}} \begin{bmatrix} 3 & -2 & -3 \\ 6 & 0 & -2 \\ 3 & -2 & -3 \end{bmatrix} \quad F_{\mathrm{SW}} \begin{bmatrix} -2 & -3 & -2 \\ 3 & 0 & -3 \\ 6 & 3 & -2 \end{bmatrix} \quad F_{\mathrm{S}} \begin{bmatrix} -3 & -2 & -3 \\ -2 & 0 & -2 \\ 3 & 6 & 3 \end{bmatrix} \quad F_{\mathrm{SE}} \begin{bmatrix} -2 & -3 & -2 \\ -3 & 0 & 3 \\ -2 & 3 & 6 \end{bmatrix}$$

Figure 5. *Filters used for estimation of local edge magnitude*



Input

w

Figure 6. *Calculation of w*

Step 3: Interpolation of the G component

We reconstruct the G component using local directional anisotropy and local edge magnitude estimated in Steps 1 and 2. First, we calculate a set of the G component (H_l, H_h, V_l, V_h) using horizontal low-pass and high-pass filters h_l and h_h, respectively, and vertical low-pass and high-pass filters v_l and v_h, respectively, as shown below.

$$H_l(i,j) = h_l * BA(i,j) \tag{13}$$

$$V_l(i,j) = v_l * BA(i,j) \tag{14}$$

$$H_h(i,j) = h_h * BA(i,j) \tag{15}$$

$$V_h(i,j) = v_h * BA(i,j) \tag{16}$$

$$h_l = [1/2 \quad 0 \quad 1/2] \tag{17}$$

$$v_l = {h_l}^T \tag{18}$$

$$h_h = [-1/4 \quad 0 \quad 2/4 \quad 0 \quad -1/4] \tag{19}$$

$$v_l = {h_h}^T \tag{20}$$

H_h and V_h are calculated from the R and B components of the input image. The interpolation of the G components using the R and B components assumes that the high-frequency component of the image is achromatic.

Next, the missing G components are interpolated as a weighted linear sum of horizontal and vertical components by equation (21). We use λ defined in equation (9) to control the balance of horizontal and vertical components, and *w* defined in equation (12) to control the contribution of high-frequency components in the interpolation.

$$G(i,j) = \begin{cases} \lambda(i,j) \cdot H_l(i,j) + \{1 - \lambda(i,j)\} \cdot V_l(i,j) \\ +\lambda(i,j) \cdot \mathrm{w}(i,j) \cdot H_h(i,j) + \\ +\{1 - \lambda(i,j)\} \cdot \mathrm{w}(i,j) \cdot V_h(i,j) \\ \qquad (\text{If } BA(i,j) \text{ is B or R component}) \\ BA(i,j) \\ \qquad (\text{If } BA(i,j) \text{ is G component}) \end{cases} \tag{21}$$

Step 4: Interpolation of the R and B components

Once the G component is reconstructed, the remaining R and B components should be interpolated. We used a method that was inspired by the work of Zhang and Wu [7]. To reconstruct the R component, we first compute the difference between the G and the R components at the location of the R component in the input data.

$$\Delta RG(i,j) = \begin{cases} BA(i,j) - \mathrm{G}(i,j) \\ \qquad (\text{If } BA(i,j) \text{ is R component}) \\ 0 \\ \qquad (\text{If } BA(i,j) \text{ is G or B component}) \end{cases} \tag{22}$$

We fill in the difference value at the location of the B component in the input data using its four diagonally closest difference values.

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-279

$$\Delta RG(i,j) = \begin{array}{l}\{\Delta RG(i-1,j-1) + \Delta RG(i-1,j+1)\\ + \Delta RG(i+1,j-1) + \Delta RG(i+1,j+1)\}/4\end{array} \tag{23}$$

Next, we interpolate the difference value at the G component in the input data using the four closest difference values including the ones filled in (23).

$$\Delta RG(i,j) = \begin{array}{l}\{\Delta RG(i-1,j) + \Delta RG(i+1,j)\\ + \Delta RG(i,j-1) + \Delta RG(i,j+1)\}/4\end{array} \tag{24}$$

In the final stage of this step, we recover the R value by adding the completed difference values to the G component at each spatial location.

$$R(i,j) = \Delta RG(i,j) + \mathrm{G}(i,j) \tag{25}$$

The B component is interpolated by substituting R and B in the above procedure.

Evaluation of the proposed method

We compared the performance of the proposed demosaicing method with other conventional algorithms reported in previous studies. In order to quantify these performances, we first prepared full-RGB images as truths (Figure 7), and generated Bayer pattern images from them that simulated the subsampling process with color filter array (CFA). Figures 8 and 9 compare the performance of the proposed method with other methods, including state-of-the-art algorithms. The figures show that our algorithm performs well compared with the state-of-the-art algorithms. In the case of achromatic color images such as image (g), although the performance of the proposed method is significantly better than the bilinear interpolation and the method in [8], there remains room for improvement.

Figure 7. *Test images used in this study*

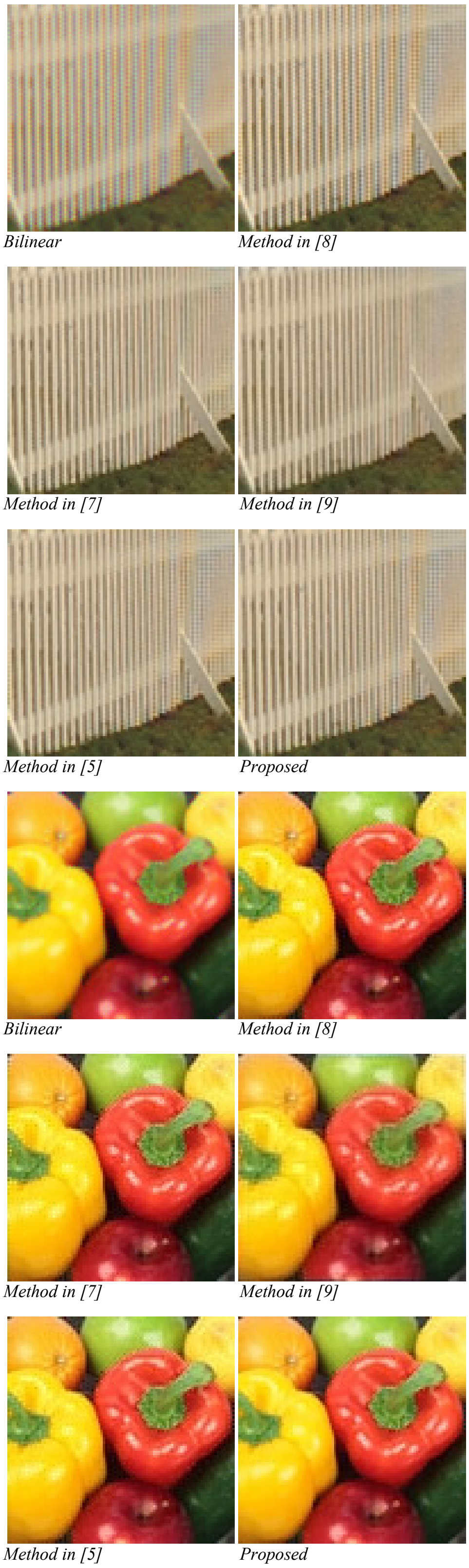

Bilinear *Method in [8]*
Method in [7] *Method in [9]*
Method in [5] *Proposed*
Bilinear *Method in [8]*
Method in [7] *Method in [9]*
Method in [5] *Proposed*

Figure 8. *Demosaicing results for the test images*

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-279

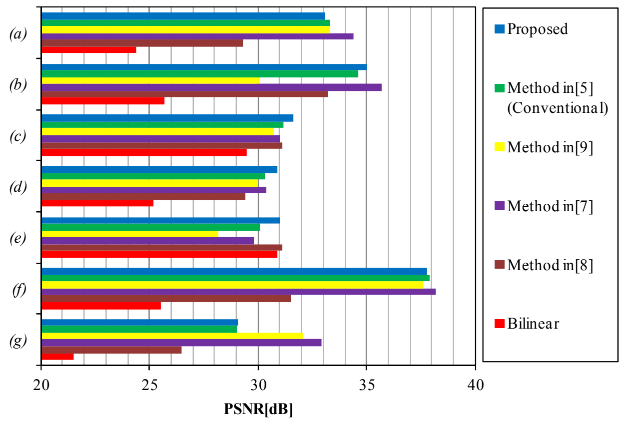


Figure 9. *PSNR results of the proposed method and the other methods*

Development of an 8K demosaicing processor

Overview

We develop an 8K demosaicing processor that implements the proposed method as shown in Figure 10. Table 1 summarizes its specifications. A Bayer pattern video captured by a single-chip 8K camera is used as the input for the processor (see Figure 1 (b)). The interface between the camera output and the processor input is HD-SDI (1.5G or 3G-SDI). The output of the processor is full RGB 8K video and the interface is U-SDI [10, 11, 12].

Figure 10*. 8K UHDTV demosaicing processor*

Performance

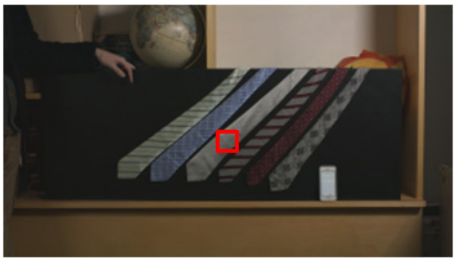
Figure 11 shows an example of the processor's output images in comparison with linear interpolation. The quality of the output is similar to the simulated results reported in the previous section. For example, color artifacts on a stripe texture including high-frequency components close to the Nyquist frequency are reduced effectively.

Figure 13 shows the results of the performance evaluation carried out for the 8K test images shown in Figure 12. The processor performed better than linear interpolation except for images (h) and (i), which have less high-frequency components textures and relatively much noise.

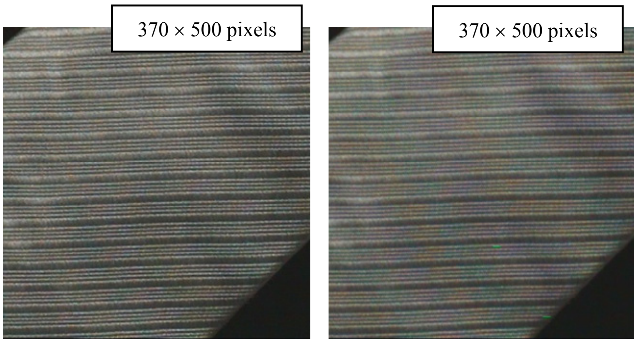
In addition to good video quality, the processing delay time of the processor is less than one frame (1.001/60 s), which is suitable for use in real-time application including live program production.

Table 1. Specifications of the 8K demosaicing processor

Input	Video format	G1, G2, B, R (Bayer pattern)
	Number of pixels	3840 × 2160
	Frame frequency	60, 60/1.001 Hz
	Bit depths	10 bit
	interface	1.5G-SDI × 16 or 3G-SDI × 8
Output	Video format	RGB 4:4:4
	Number of pixels	7680 × 4320
	Frame frequency	60, 60/1.001 Hz
	Bit depths	12 bit
	interface	U-SDI [10, 11, 12]
Delay		< 1.001/60 s
Dimensions (W × H × D)		430 × 44 × 351 mm
Weight		4.5 kg
Power consumption		Approx. 90 W

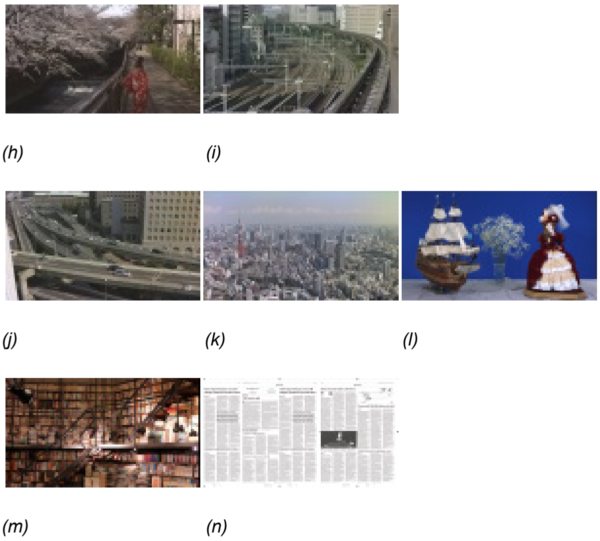


Output (full RGB video)



Processor *Bilinear interpolation*

Figure 11*. Example of improvement by the processor*


(h) (i)

(j) (k) (l)

(m) (n)

Figure 12. *8K test images used in this study*

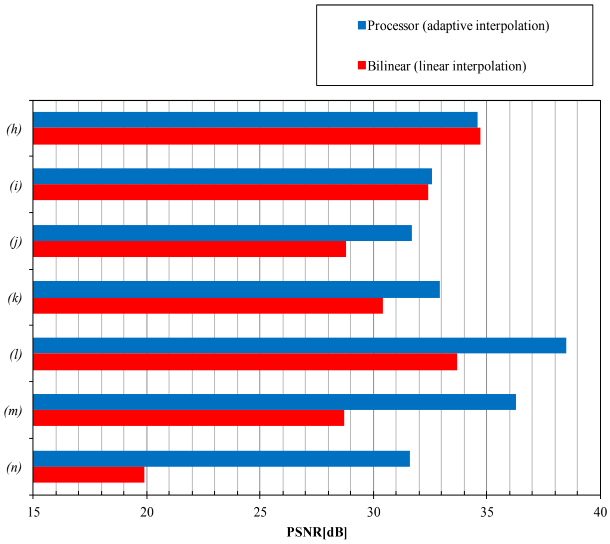


Figure 13. *PSNR results of the proposed method and bilinear interpolation*

Conclusion

We developed a demosaicing processor that converts Bayer pattern 8K images into full-resolution 8K images with high quality and in real time. Our demosaicing algorithm uses an adaptive interpolation filter based on local directional anisotropy and local edge magnitude, and it shows good performance comparable to conventional demosaicing algorithms including state-of-the-art algorithms. We will continue studies to improve the performance for monochromatic images. Further, we will develop a processor running at the frame frequency of 120 Hz to support a 120 Hz camera system.

References

[1] H. Shimamoto, T, Yasue. K. Kitamura, T. Watabe, N. Egami, S. Kawahito, T. Kosugi, T. Watanabe, and T. Tsukamoto, “A Compact 120 Frames/sec UHDTV2 Camera with 35mm PL Mount Lens,” SMPTE Motion Imaging Journal, vol. 123, no. 4, pp. 21-28, 2014.

[2] R. Funatsu, T. Yamashita, K. Mitani, and Y. Nojiri, “Single-Chip Color Imaging for UHDTV Camera with a 33M-Pixel CMOS Image Sensor,” in: R. Widenhorn, V. Nguyen (Eds.), Proceedings of SPIE, 7875, 2011. (787502:1–11).

[3] H. Shimamoto, T. Yamashita, N. Koga, K. Mitani, M. Sugawara, F. Okano, M. Matsuoka, J. Shimura, I. Yamamoto, T. Tsukamoto, and S. Yahagi, “An 8K × 4K Ultrahigh-Definition Color Video Camera with 8M-Pixel CMOS Imager,” SMPTE Motion Imaging Journal, vol. 114, no. 7-8, pp. 260-268, 2005.

[4] B. E. Bayer, “Color Imaging Array,” U. S. Patent, 3971065, 1976.

[5] S. Tajima, R. Funatsu, Y. Nishida, “Chromatic Interpolation based on Anisotropy-Scale-Mixture Statistics,” Signal Processing, vol. 97, pp. 262-268, 2014.

[6] R. A. Kirsch, “Computer Determination of the Constituent Structure of Biological Images,” Computers and Biomedical Research, vol. 4, no. 3, pp. 315-328, 1971.

[7] L. Zhang, and X. Wu, “Color Demosaicking via Directional Linear Minimum Mean Square-Error Estimation,” IEEE Trans. Img. Processing, vol. 14, no. 12, pp. 2167-2178, 2005.

[8] H. S. Malvar, L. He, and R. Cutler, “High Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images,” Proceedings of IICASPP III, vol. 3, pp. 485-488, 2004.

[9] G. Jeon, and E. Dubois. “Demosaicking of Noisy Bayer-Sampled Color Images with Least-Squares Luma-Chroma Demultiplexing and Noise Level Estimation,” IEEE Trans. on Img. Processing, vol. 22, no. 1, pp. 146-156, 2013.

[10] Recommendation ITU-R BT.2077-1, “Real-Time Serial Digital Interfaces for UHDTV Signals,” 2015.

[11] SMPTE ST 2036-4, “Ultra High Definition Television–Multi-link 10 Gb/s Signal/Date Interface Using 12-Bit Width Container,” 2015.

[12] ARIB STD-B58, “Interface for UHDTV Production Systems,” 2014.

Author Biography

Noriyuki Shirai received his BE and ME degrees in Electrical Engineering from Tokyo University of Science in 2002 and 2004, respectively. He joined Japan Broadcasting Corporation (NHK) in 2004. Since 2013, he has been with NHK Science and Technology Research Laboratories. Currently, his work focuses on the development of 8K UHDTV systems.