

Dynamic Range Bad Pixel Correction within Edge-Preserving Colour Filter Array Demosaicking

Jim S. Jimmy Li

School of Computer Science, Engineering and Mathematics
Flinders University
GPO Box 2100, SA 5001, Australia
Email: jimmy.li@flinders.edu.au

Sharmil Randhawa

School of Computer Science, Engineering and Mathematics
Flinders University
GPO Box 2100, SA 5001, Australia
Email: sharmil.randhawa@flinders.edu.au

Abstract—Due to the increase in resolution of digital cameras which now contain millions of pixels, it is highly probable that an image sensor will contain a number of defective pixels due to errors in the fabrication process, and this will visually deteriorate the quality of the output image. Although those defective pixels would normally be mapped out in the manufacturing process, more bad pixels could appear over time. Existing bad pixel correction algorithms could mistake a good pixel as a bad pixel and details of an image could be removed as a result. In this paper, we present a dynamic range bad pixel detection method to reduce false positive error, so as to better preserve the fine details in an image. We also present a novel method using cubic interpolation to estimate a value for replacing a bad pixel with high accuracy. The final full colour image is then produced by our edge-preserving demosaicking method. It has been shown that our proposed method outperforms other existing techniques.

I. INTRODUCTION

All the sensor values produced by digital cameras should be non-defective and free of noise for the demosaicking process to produce an accurate and visually pleasing image. However, despite advances in the manufacturing process, digital cameras often contain a few defective pixels as a result of noise or fabrication errors [1]. A defective pixel is a pixel that responds inappropriately to exposed light from an image and thus produces an incorrect sensor value [2]. There are three main types of defective or bad pixels: hot, dead or noisy. A hot pixel produces a brighter than expected spot, while a dead pixel produces a darker than expected spot in the output image. A noisy pixel produces a sensor value which differs from neighbouring pixels by more than a certain amount when exposed to the same light conditions. Bad pixel correction (BPC) is the process of detecting and correcting defective pixels. Traditionally BPC and demosaicking have been performed in two separate stages, either in software or in a digital circuit implemented in hardware. Each stage adds to the complexity and expense of processing sensor values for final output on a display device.

In this paper, we propose a novel bad pixel correction (BPC) technique to filter out only bad pixels whilst leaving the rest of the image intact. The CFA image with bad pixels is firstly filtered to remove bad pixels before passing through a state-of-the-art demosaicking algorithm to produce an RGB image.

II. PROPOSED METHOD

In this section, we present a dynamic range bad pixel detection and correction method based on the characteristics of the adjacent neighbourhood pixels. The dynamic range bad pixel detection determines whether a test pixel is bad based on the adjacent pixels and the dynamic range bad pixel correction replaces the bad pixel with an appropriate value estimated from the neighbourhood pixels.

In this method, interpolation of the neighbourhood environment is used to give four estimates in the four cardinal directions of the test pixel in question. If the pixel under examination deviates from the maximum (or minimum) of the four estimates by a certain threshold, that pixel is deemed to be defective and replaced by the maximum (or minimum) of the four estimates accordingly. We use cubic interpolation to determine four accurate estimates for each pixel using the surrounding neighbourhood pixels.

A. Cubic Interpolation

In this section, we show how the four estimates for the pixel under examination extrapolated from neighbouring pixels along the four cardinal directions can be determined using cubic interpolation.

A third degree polynomial, $C(x)$, is constructed between each point for the given $n + 1$ points (x_0, y_0) to (x_n, y_n) :

$$C(x) = ax^3 + bx^2 + cx + d. \quad (1)$$

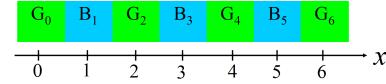


Fig. 1. 1-D Bayer pattern along the x-axis.

Fig. 1 shows the horizontal layout of pixels of a CFA image, with alternate green and blue pixels. For this illustration, let G_4 be the possible defect pixel. We use cubic interpolation to give an estimate for G_4 based on the green and blue pixels on the left.

From (1), there are four unknown coefficients, namely a , b , c and d . Hence four equations are required to obtain a unique solution. They are as follows.

$$C(0) = G_0 = d. \quad (2)$$

$$C(2) = G_2 = 8a + 4b + 2c + d. \quad (3)$$

The estimates of the green values at positions $x = 1, 3$ and 5 are given by:

$$\hat{G}_1 = C(1) = a + b + c + d. \quad (4)$$

$$\hat{G}_3 = C(3) = 27a + 9b + 3c + d. \quad (5)$$

$$\hat{G}_5 = C(5) = 125a + 25b + 5c + d. \quad (6)$$

As four equations are needed to obtain a unique solution, two extra conditions are required. Using the hue assumption [3], [4] based on the correlation between colour planes, we propose the extra two equations be derived as follows.

Based on the assumption that the green and red/blue pixel values are well correlated with constant offsets along an edge [4]–[6]:

$$G_x - B_x = G_{x+1} - B_{x+1} = G_{x+2} - B_{x+2}. \quad (7)$$

From (7):

$$B_{x+2} - B_x = G_{x+2} - G_x. \quad (8)$$

By (8), this implies

$$B_3 - B_1 = \hat{G}_3 - \hat{G}_1. \quad (9)$$

Hence by (4) and (5)

$$B_3 - B_1 = 26a + 8b + 2c. \quad (10)$$

Similarly,

$$B_5 - B_3 = \hat{G}_5 - \hat{G}_3. \quad (11)$$

Hence by (5) and (6)

$$B_5 - B_3 = 98a + 16b + 2c. \quad (12)$$

To solve the four simultaneous equations, namely (2)–(3), (10), and (12), the matrix representation (13) is used.

$$MC = V, \quad (13)$$

where M , C and V are given by (14)–(16).

$$\text{Let } M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 8 & 4 & 2 & 1 \\ 26 & 8 & 2 & 0 \\ 98 & 16 & 2 & 0 \end{bmatrix} \quad (14)$$

$$C = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (15)$$

$$V = \begin{bmatrix} G_0 \\ G_2 \\ B_3 - B_1 \\ B_5 - B_3 \end{bmatrix} \quad (16)$$

$$\therefore C = M^{-1}V, \quad (17)$$

where M^{-1} is the inverse of the matrix M . This inverse exists and is shown in (18) and thus can be pre-evaluated and stored, making the algorithm computationally efficient.

$$M^{-1} = \frac{1}{72} \begin{bmatrix} -4 & 4 & -6 & 2 \\ 36 & -36 & 45 & -9 \\ -92 & 92 & -66 & 10 \\ 72 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

Solving (17) together with (18) gives the equations for the coefficients a , b , c and d as follows:

$$a = \frac{1}{72}[-4G_0 + 4G_2 - 6(B_3 - B_1) + 2(B_5 - B_3)]. \quad (19)$$

$$b = \frac{1}{72}[36G_0 - 36G_2 + 45(B_3 - B_1) - 9(B_5 - B_3)]. \quad (20)$$

$$c = \frac{1}{72}[-92G_0 + 24G_2 - 66(B_3 - B_1) + 10(B_5 - B_3)]. \quad (21)$$

$$d = G_0. \quad (22)$$

Once the coefficients a , b , c and d have been determined, the estimate of the green pixel based on the pixels on the left hand side is evaluated at $x = 4$:

$$\hat{G}_4^L = \frac{1}{3}[G_0 + 2G_2 - 3B_1 + 2B_3 + B_5]. \quad (23)$$

In general, for a 2D image, the estimate from the left-hand-side is given by:

$$\begin{aligned} \hat{G}^L(x, y) = & \frac{1}{3}[G(x-4, y) + 2G(x-2, y) - 3B(x-3, y) \\ & + 2B(x-1, y) + B(x+1, y)]. \end{aligned} \quad (24)$$

Similarly, the R, T and B estimates for the green pixel can be determined as follows.

$$\begin{aligned} \hat{G}^R(x, y) = & \frac{1}{3}[G(x+4, y) + 2G(x+2, y) - 3B(x+3, y) \\ & + 2B(x+1, y) + B(x-1, y)]. \end{aligned} \quad (25)$$

$$\begin{aligned} \hat{G}^T(x, y) = & \frac{1}{3}[G(x, y-4) + 2G(x, y-2) - 3R(x, y-3) \\ & + 2R(x, y-1) + R(x, y+1)]. \end{aligned} \quad (26)$$

$$\begin{aligned} \hat{G}^B(x, y) = & \frac{1}{3}[G(x, y+4) + 2G(x, y+2) - 3R(x, y+3) \\ & + 2R(x, y+1) + R(x, y-1)]. \end{aligned} \quad (27)$$

B. Dynamic Range Bad Pixel Detection and Correction

Once the four estimates are derived, the pixel under examination, namely G_4 , is compared with the maximum and minimum of the four estimates to determine whether the pixel is outside those limits from the neighbourhood estimate. G_4 is deemed to be defective if it is outside those limits. It is then re-mapped by (28) and (29), and the method is shown in Fig. 2.

$$\begin{aligned} \text{If } G(x, y) > \text{MAX}[\hat{G}^L(x, y), \hat{G}^R(x, y), \hat{G}^T(x, y), \hat{G}^B(x, y)] + Th, \\ G(x, y) = \text{MAX}[\hat{G}^L(x, y), \hat{G}^R(x, y), \hat{G}^T(x, y), \hat{G}^B(x, y)]. \end{aligned} \quad (28)$$

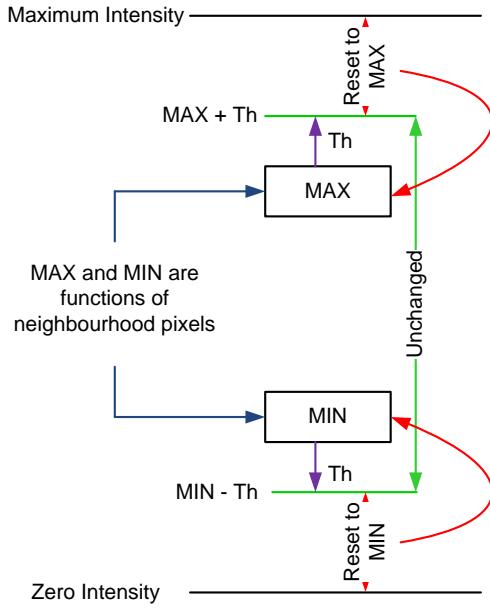


Fig. 2. Dynamic Range Bad Pixel Detection and Correction.

$$\text{If } G(x, y) < \text{MIN}[\hat{G}^L(x, y), \hat{G}^R(x, y), \hat{G}^T(x, y), \hat{G}^B(x, y)] - Th, \\ G(x, y) = \text{MIN}[\hat{G}^L(x, y), \hat{G}^R(x, y), \hat{G}^T(x, y), \hat{G}^B(x, y)]. \quad (29)$$

The flowchart of our proposed Bad Pixel Correction method using Cubic Interpolation (BPC-CI) algorithm is shown in Fig. 3.

III. RESULTS

To assess the performance of our proposed BPC algorithm, defective pixels in the form of random impulses of various magnitudes were added to the CFA data down-sampled from the twenty-four Kodak test images [7] in Fig. 4. The images represent a significant variety of homogeneous regions, colours, and textured areas. A bad pixel density of 0.5% was investigated, i.e. the CFA image data was corrupted by 0.5% bad pixels. In practice, it is unlikely for a digital camera to be corrupted by more than 0.5% bad pixels. However under extreme circumstances, the noise density of bad pixels could be higher especially in high temperatures and long exposure conditions.

For the proposed Bad Pixel Correction using Cubic Interpolation (BPC-CI) algorithm, the value of the threshold Th in (28) and (29) has influence on the true positive rate (TPR) and false positive rate (FPR). We tested with the full suite of images in Fig. 4, and we experimentally found that the mean and median of the threshold values giving the best CPSNR results was 0.1256 and 0.12 respectively. Hence a threshold value of $Th = 0.12$ was chosen to give a good CPSNR value for the images, as well as a good balance between TPR and FPR.

To evaluate our proposed BPC algorithm, it was compared with other state-of-the-art BPC algorithms. Our demosaicking

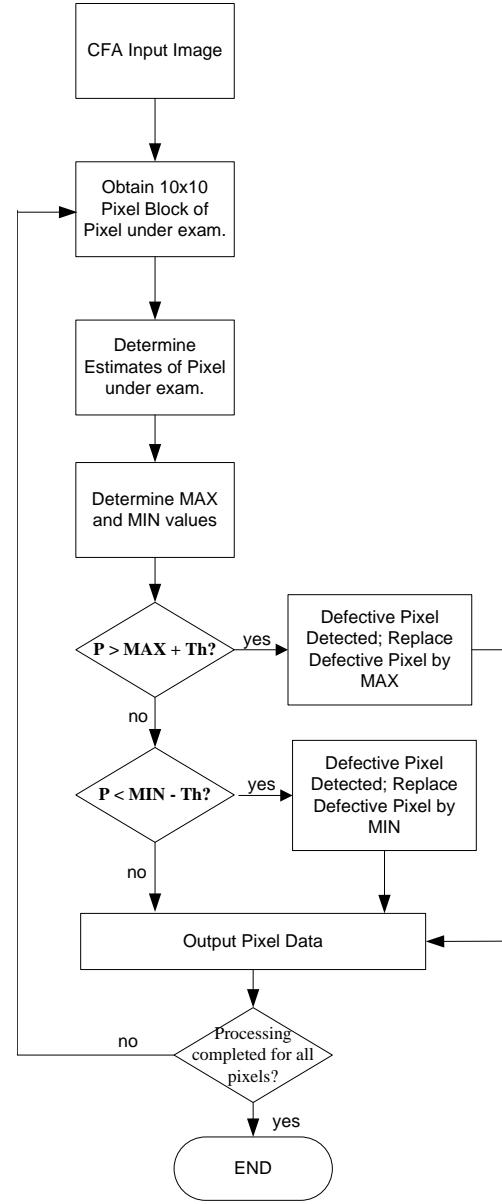


Fig. 3. Flowchart of the Proposed Bad Pixel Correction using Cubic Interpolation (BPC-CI) algorithm.

method as described in [8] was used to demosaic the corrected CFA images after bad pixel correction was carried out.

Table I gives the average and median true positive rates (TPR) and false positive rates (FPR) for the BPC algorithms, namely Bao [9], Pinto [10], Nagatsuma [11], Nishio [12], Kakarala [1], Li&Randhawa [13] and our proposed algorithm BPC-CI.

Tables II and III tabulate the CPSNR and NCD image quality values for the output demosaicked images from the various methods (compared with the original image), when the input CFA data image was corrupted with 0.5% bad pixel density, with the best results in bold. For comparison, we have also included the results from using only the demosaicking method as described in [8] on the corrupted CFA data image

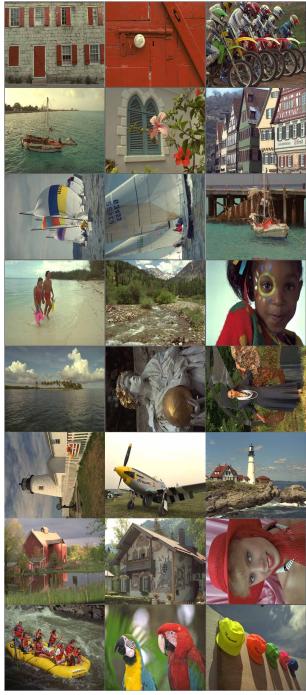


Fig. 4. Test Images 4(a)-(x) from left to right, top to bottom.

	True Positive Rate		False Positive Rate	
	Average	Median	Average	Median
Bao	75.24%	77.97%	0.28%	0.13%
Pinto	74.84%	77.55%	0.30%	0.14%
Nagatsuma	73.81%	76.60%	0.25%	0.15%
Nishio	73.22%	76.65%	1.16%	0.66%
Kakarala	88.26%	90.88%	5.79%	5.32%
Li&Randhawa	91.65%	94.07%	27.07%	26.86%
BPC-CI	73.86%	77.50%	0.13%	0.07%

TABLE I

AVERAGE AND MEDIAN TRUE POSITIVE AND FALSE POSITIVE RATES OF THE BPC METHODS FOR 0.5% BAD PIXEL DENSITY FOR THE TWENTY-FOUR IMAGES IN FIG. 4.

without any bad pixel correction. From these tables, it can be seen that for small bad pixel density values, the proposed BPC-CI method gave the best CPSNR and NCD image quality results.

Fig. 5 gives the visual assessment results for the various BPC methods for the cropped Parrots image (Fig. 4(w)) when its CFA image is corrupted with 0.5% bad pixels. Our proposed BPC-CI, Li&Randhawa's and Kakarala's methods all showed no visible bad pixels in their demosaicked output image, whilst there were still a few bad pixels remaining in the output images of the other BPC methods. However, it should be noted that the output image produced by Kakarala's method was blurred with colour artifacts. All in all, Tables II and III confirm that our proposed BPC-CI mostly gave the best image quality results.

IV. CONCLUSION

In this paper we presented a novel Bad Pixel Correction using Cubic Interpolation (BPC-CI) method, which consists

of a dynamic range bad pixel detection and correction based on the characteristics of adjacent neighbourhood pixels. It has been shown that our proposed BPC-CI has a very low average FPR (0.13%). For small bad pixel densities (0.5%), it has been shown that our BPC-CI method is able to remove bad pixels effectively with minimal colour artifacts while producing images with superior CPSNR and NCD image quality values.

REFERENCES

- [1] R. Kakarala, "Digital image system and method for combining demosaicing and bad pixel correction," *US Patent 7,015,961 B2*, 2006.
- [2] J. H. Stanback, X. Zhang, R. Kakarala, and B. Y. P. Ying, "System and method for detecting and correcting defective pixels in a digital image sensor," *US Patent 7460688*, 2008.
- [3] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, pp. 1521–1527, 2001.
- [4] W. Lu and Y.-P. Tan, "Color filter array demosaicking: New method and performance measures," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1194–1210, 2003.
- [5] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," *US Patent 4,642,678*, 1987.
- [6] J. E. Adams Jr., "Interactions between color plane interpolation and other image processing functions in electronic photography," *Proc. SPIE*, vol. 2416, pp. 144–151, 1995.
- [7] Kodak, "Kodak photo cd pcd0992," <http://r0k.us/graphics/kodak/>, 1991.
- [8] J. S. J. Li and S. Randhawa, "Color filter array demosaicking using high order interpolation techniques with a weighted median filter for sharp color edge preservation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1946–1957, 2009.
- [9] Y. Bao, "Bad pixel correction while preserving features," *US Patent 7064768*, 2006.
- [10] V. Pinto and D. Shaposhnik, "Dynamic identification and correction of defective pixels," *US Patent 7388609*, 2008.
- [11] H. Nagatsuma, Y. Masui, T. Nakamura, and K. Nishiwaki, "Defective pixel detector, imaging device, and defective pixel detection method," *US Patent 2009/0016638 A1*, 2009.
- [12] S. Nishio, H. Daiku, and A. Kokubo, "Circuit and method for correction of defect pixel," *US Patent 7,263,215 B2*, 2007.
- [13] J. S. J. Li and S. Randhawa, "Adaptive order-statistics multi-shell filtering for bad pixel correction within cfa demosaicking," *Proceedings Of The IEEE International Region 10 Conference (IEEE TENCON'09)*, 2009.

Image	Bao + [8]	Pinto + [8]	Nagatsuma + [8]	Nishio + [8]	Proposed BPC-CI	Li&Randhawa	Kakarala	Demosaic using [8]
Fig. 4(a)	33.96	32.62	33.43	30.38	34.50	30.16	24.68	30.32
Fig. 4(b)	37.44	37.69	38.01	35.39	38.93	35.37	31.61	30.36
Fig. 4(c)	34.05	31.73	32.61	25.70	34.85	31.70	24.91	29.80
Fig. 4(d)	35.53	33.76	34.67	31.79	36.40	32.31	26.16	31.44
Fig. 4(e)	37.66	39.38	39.49	34.43	39.85	38.18	32.17	31.20
Fig. 4(f)	32.48	30.30	30.85	27.51	32.84	27.13	21.99	30.17
Fig. 4(g)	37.95	38.49	38.71	35.52	39.69	37.22	30.97	32.11
Fig. 4(h)	37.79	36.84	37.68	33.17	39.91	36.67	30.91	31.55
Fig. 4(i)	35.71	34.38	35.23	30.76	36.50	32.86	27.66	30.35
Fig. 4(j)	38.79	40.08	40.11	37.24	40.57	37.40	32.32	33.62
Fig. 4(k)	30.72	26.96	27.81	24.10	30.30	27.89	22.20	28.94
Fig. 4(l)	36.77	36.90	37.14	34.52	37.80	36.85	31.86	30.89
Fig. 4(m)	37.92	39.00	39.34	37.04	40.08	35.13	29.96	30.91
Fig. 4(n)	36.93	35.76	36.23	32.37	38.33	36.24	30.57	30.13
Fig. 4(o)	33.77	31.73	32.42	28.54	34.00	31.99	26.37	29.29
Fig. 4(p)	36.35	36.71	36.99	33.33	37.59	29.83	26.32	31.03
Fig. 4(q)	37.95	37.20	37.69	32.93	38.65	36.06	30.53	33.08
Fig. 4(r)	35.10	33.62	34.34	30.36	35.99	32.80	26.86	30.81
Fig. 4(s)	35.55	35.30	35.59	32.90	36.56	33.48	28.76	30.72
Fig. 4(t)	32.10	29.11	29.62	26.78	32.23	29.53	24.97	29.41
Fig. 4(u)	37.22	38.13	38.38	34.85	39.03	37.77	32.13	30.76
Fig. 4(v)	34.50	33.38	33.86	29.96	35.28	32.71	27.54	30.03
Fig. 4(w)	38.81	39.90	40.57	34.74	40.84	40.75	33.18	31.27
Fig. 4(x)	38.40	40.14	40.34	37.42	40.35	39.50	33.29	30.89

TABLE II
CPSNR (dB) IMAGE QUALITY RESULTS FOR 0.5% BAD PIXEL DENSITY.

Image	Bao + [8]	Pinto + [8]	Nagatsuma + [8]	Nishio + [8]	Proposed BPC-CI	Li&Randhawa	Kakarala	Demosaic using [8]
Fig. 4(a)	34.56	36.37	35.35	40.26	34.17	40.07	95.03	35.58
Fig. 4(b)	26.04	25.47	25.38	27.16	25.24	29.49	48.78	28.09
Fig. 4(c)	44.56	47.33	46.19	70.59	43.93	50.39	115.56	45.92
Fig. 4(d)	21.09	22.31	21.57	24.15	20.63	24.34	61.16	21.82
Fig. 4(e)	19.29	18.69	18.66	22.44	18.60	20.07	41.32	20.56
Fig. 4(f)	32.68	34.70	33.94	39.53	32.15	41.20	98.35	33.29
Fig. 4(g)	12.04	11.80	11.77	12.92	11.66	12.88	27.18	12.93
Fig. 4(h)	11.86	11.94	11.73	13.31	11.36	12.83	25.40	12.82
Fig. 4(i)	30.44	31.23	30.51	36.26	29.78	34.18	76.30	32.02
Fig. 4(j)	10.85	10.53	10.53	11.28	10.48	11.71	23.96	11.48
Fig. 4(k)	54.60	61.35	59.19	72.92	54.92	58.63	124.51	55.10
Fig. 4(l)	27.69	27.10	27.04	29.04	26.97	28.66	45.92	29.39
Fig. 4(m)	18.33	18.04	17.90	19.12	17.64	21.25	51.53	19.64
Fig. 4(n)	34.35	34.05	33.79	38.17	33.31	35.00	66.79	36.65
Fig. 4(o)	50.32	52.08	51.07	61.44	49.68	52.95	105.16	52.17
Fig. 4(p)	25.14	25.03	24.90	27.84	24.57	30.14	69.16	26.24
Fig. 4(q)	18.48	18.49	18.47	21.06	18.22	20.01	35.79	19.10
Fig. 4(r)	25.63	26.95	26.17	32.63	25.32	28.08	62.11	26.56
Fig. 4(s)	28.77	28.83	28.62	31.16	28.21	31.09	56.92	29.91
Fig. 4(t)	33.24	35.41	34.73	40.82	33.00	36.97	72.74	34.28
Fig. 4(u)	27.27	26.76	26.67	29.38	26.51	28.69	50.18	28.96
Fig. 4(v)	36.33	36.89	36.46	43.42	35.74	39.67	82.05	37.71
Fig. 4(w)	15.15	14.53	14.47	16.20	14.54	15.12	24.45	16.54
Fig. 4(x)	15.00	14.34	14.29	15.58	14.36	15.48	30.78	16.54

TABLE III
NCD($\times 10^{-3}$) IMAGE QUALITY RESULTS FOR 0.5% BAD PIXEL DENSITY.



Fig. 5. (a) the original cropped Parrots image, (b) CFA Data Image corrupted with 0.5% Bad Pixels, and the demosaicked output images using (c) [8] with no bad pixel correction, (d) Bao-WM, (e) Pinto-WM, (f) Nagatsuma-WM, (g) Nishio-WM, (h) Kakarala, (i) Li&Randhawa and (j) proposed BPC-CI.