Adaptive pixel defect correction

Anthony A. Tanbakuchi[a,b], Arjen van der Sijde[c], Bart Dillen[b],
Albert Theuwissen[b,d], and Wim de Haan[e]

[a]DALSA Professional Imaging, Prof. Holstaan 4, Eindhoven, The Netherlands
[b]Rochester Institute of Technology, Imaging and Photographic Technology Dep., New York
[c]Philips Semiconductors Image Sensors, Prof. Holstaan 4, Eindhoven, The Netherlands
[d]Delft University of Technology, Mekelweg 4, Delft, The Netherlands
[e]Philips Components, Prof. Holstaan 4, Eindhoven, The Netherlands

ABSTRACT

Although the number of pixels in image sensors is increasing exponentially, production techniques have only been able to linearly reduce the probability that a pixel will be defective. The result is a rapidly increasing probability that a sensor will contain one or more defective pixels. Sensors with defects are often discarded after fabrication because they may not produce aesthetically pleasing images. To reduce the cost of image sensor production, defect correction algorithms are needed that allow the utilization of sensors with bad pixels. We present a relatively simple defect correction algorithm, requiring only a small 7 by 7 kernel of raw color filter array data that effectively corrects a wide variety of defect types. Our adaptive edge algorithm is high quality, uses few image lines, is adaptable to a variety of defect types, and independent of other on-board DSP algorithms. Results show that the algorithm produces substantially better results in high-frequency image regions compared to conventional one-dimensional correction methods.

Keywords: defect correction, CFA, interpolation, image sensor

1. INTRODUCTION

Defects in image sensors tend to be composed of single pixels, small pixel clusters, or may also include whole columns. In natural (pictorial) images, these types of image defects are usually unacceptable to the human observer. Therefore, sensors with defects must either be discarded or incorporate on-board defect removal to preserve the image quality. A processing scheme will be discussed in this paper.

We will introduce an adaptive defect correction algorithm that utilizes natural image properties to interpolate defective pixels. Section 2 briefly reviews relevant past work. In Section 3, we discuss useful natural image properties that are exploited in the algorithm. The algorithm and results are introduced in Sections 4 and 5. Finally, conclusions are presented in Section 6.

2. PREVIOUS WORK

Various methods for defect correction in single image sensor systems have been published or patented. Relatively simple methods use a horizontal register of data and average or substitute pixel values in defect locations. Pape[1] suggests that the previous operative pixel element in the readout buffer be substituted, while Fearnside[2] averages two neighboring known pixels in the buffer. These methods are computationally efficient but often produce visually poor results, particularly in high-frequency regions.

More advanced methods proposed utilize a two-dimensional region of good pixels surrounding the defective pixel(s). Komatsu and Saito[3] capture somewhat defocused images, interpolate defects in two-dimension and then refocus the image. However, defocused images have lost some image information. Furthermore, the authors

Further author information: (Send correspondence to primary author A.A.T.)
E-mail: A.A.T.: anthony@tanbakuchi.com, A.S.: arjen.van.der.sijde@philips.com, B.D.: bart.dillen@dalsa.com, A.T.: albert.theuwissen@dalsa.com, W.H.: wim.de.haan@philips.com

$$\begin{array}{ccccccc}
R & G & R & G & R & G & R \\
G & B & G & B & G & B & G \\
R & G & R & G & R & G & R \\
G & B & G & \mathbf{B} & G & B & G \\
R & G & R & G & R & G & R \\
G & B & G & B & G & B & G \\
R & G & R & G & R & G & R
\end{array}$$

Figure 1. Bayer color filter array pattern. Each letter represents one pixel with the corresponding red, green, or blue filter in place. The bold R represent a defective pixel in the image array.

describe a Landweber-type iterative method for defect correction for grey-scale image data. Smith[4] estimates defect values by averaging a two-dimensional region of surrounding known pixels including the defective pixel. However, he notes that small intensity defects are not sufficiently corrected with this method, and there is an apparent smearing effect. Finally, Rashkovskiy and Macy[5] interpolate the luminance image estimated by green Bayer pixels and luminance/chrominance differences in the Bayer pattern. However, the cubic B-spline filters used to interpolate the defects is computationally expensive to implement.

3. NATURAL IMAGE PROPERTIES

Development of an algorithm that is computationally efficient and visually pleasing requires exploiting the properties of natural images. By using these properties, we can achieve improved results, particularly in high-frequency image regions, without use of higher-order interpolation methods.

3.1. Edges

The human visual system uses complex spatial differencing mechanisms that emphasizes edges.[6] Irregularities in edges rapidly degrade perceived image quality. Because of the psycho physical nature of the human visual system, irregularities in objects — such as poor defect correction in edge regions — are easily detected because the observer expects that objects should be continuous and closed. Therefore, it is important that the method of defect correction preserves edges.

This desire for an interpolation to render smooth edges has been the focus of much work in color reconstruction of the Bayer color filter array (CFA).[7] Laroche and Prescott[8] and Cok[9] describe methods for reconstructing the Bayer CFA by utilizing direction gradients to reduce the interpolation error. Cok remarks that by using edge information, "reconstruction errors are forced to occur in areas of the image composed of natural textures where the errors are not readily visible to the human observer."[9]

3.2. Color Correlation

In electronic systems, images are usually captured and displayed as red, green, and blue components. Finlayson, Hubel, and Hordley[10] describe how to use the correlation of the R-G-B signals to extract information for determining the scene white point. Color correlation can also be used to improve interpolation accuracy in defect correction.

The property of R-G-B correlation is readily demonstrated by computing the gradient in each color plane of a natural image. For natural images in general, the magnitudes of the gradients can be almost perfectly correlated within scene objects. The gradients at edge locations will also be correlated along the edge direction (the direction of minimum variance).

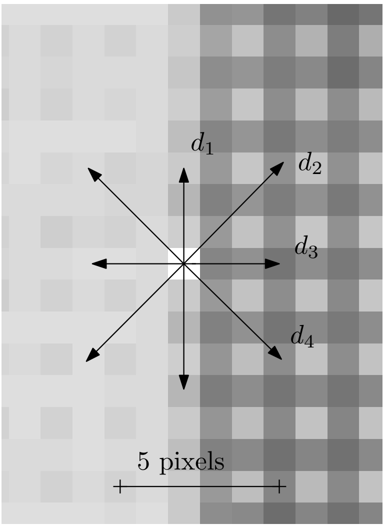


Figure 2. Direction vectors are created around the defect pixel (symbolized by white square) in a 7 × 7 kernel.

4. ADAPTIVE DEFECT CORRECTION ALGORITHM

If a pixel in an image is found to be defective, the observations in Section 3.2 states that special emphasis on preserving edge continuity. Since neighboring pixels provide the best information for interpolating a defective pixel, we would like to find the edge direction in this region and utilize the two neighboring pixels in this direction to estimate the missing pixel value. However, Figure 1 reveals that the neighboring pixels in a CFA sample are a different color than the defect location for all edge directions. (Although green pixels have a slightly different sampling pattern, we can safely treat them the same.) Therefore, the nearest known information of the same color as the defect is at least two pixel distant. This is not ideal, because the interpolation accuracy decreases rapidly with distance from the defect location.

The RGB colors of an image are locally correlated, neighboring pixels of differing color planes can be used for interpolation. By calculating the derivative in the edge direction of the non-defect color plane and applying this gradient to the nearest known pixel of the same color as the defect, we may utilize the neighboring pixel data for interpolation. Figure 3 presents an overview of the proposed algorithm laid out in the following paragraphs.

For a sampled input image $a[m, n]$ of M columns and N rows, $0 \leq m \leq M - 1, 0 \leq n \leq N - 1$, assume a defect at $a[m_0, n_0]$. Each image point is sequentially tested against a prerecorded defect map; when a defect location is found the correction algorithm is initiated at point $a[m_0, n_0]$.

The algorithm measures the pixel values in a 7 × 7.

The five steps of the algorithm are as follows,

1. The 7 × 7 kernel is broken up into 4 different edge direction vectors $d_i[n]$ where $i = \{1, 2, 3, 4\}$ and each vector with $N = 7$ elements and $n = \{-3, -2, \ldots, 3\}$. The four direction vectors i consist of the vertical, positive diagonal, horizontal, and negative diagonal directions shown in Figure 2 and defined respectively as:

$$d_1[n] = \{a[m_0 + 0, n_0 - 3], \ldots, a[m_0, n_0], \ldots, a[m_0 + 0, n_0 + 3]\} \tag{1}$$

$$d_2[n] = \{a[m_0 - 3, n_0 + 3], \ldots, a[m_0, n_0], \ldots, a[m_0 + 3, n_0 - 3]\} \tag{2}$$

$$d_3[n] = \{a[m_0 - 3, n_0 + 0], \ldots, a[m_0, n_0], \ldots, a[m_0 + 3, n_0 + 0]\} \tag{3}$$

$$d_4[n] = \{a[m_0 - 3, n_0 - 3], \ldots, a[m_0, n_0], \ldots, a[m_0 + 3, n_0 + 3]\} \tag{4}$$

   Where $d_i[n = 0]$ is always $a[m_0, n_0]$.

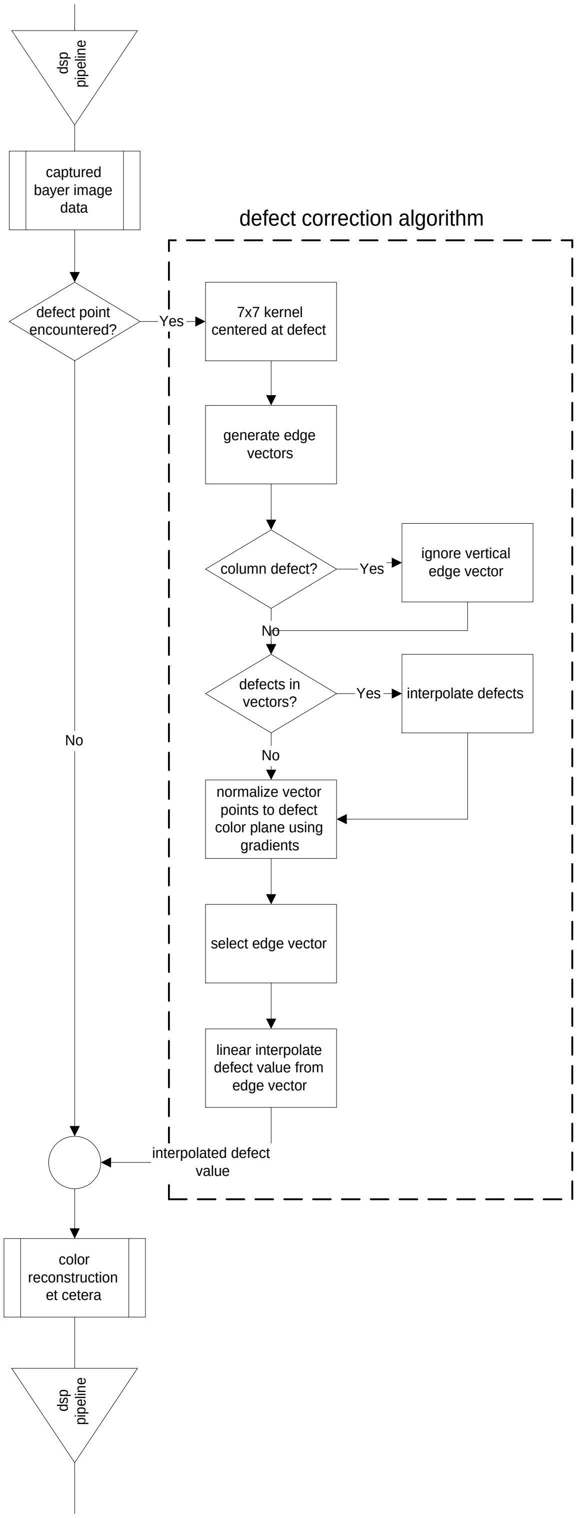
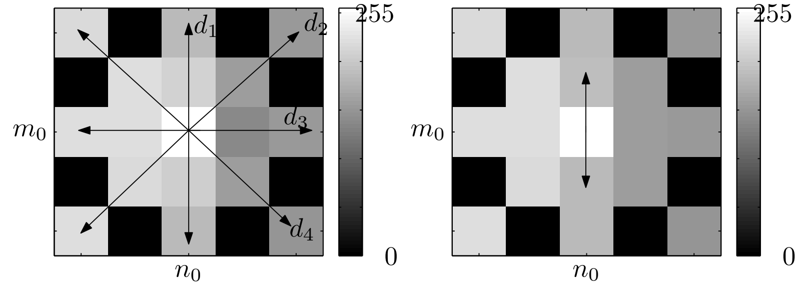

dsp pipeline
captured bayer image data
defect correction algorithm
defect point encountered?
Yes
7x7 kernel centered at defect
generate edge vectors
column defect?
Yes
ignore vertical edge vector
No
defects in vectors?
Yes
interpolate defects
No
No
normalize vector points to defect color plane using gradients
select edge vector
linear interpolate defect value from edge vector
interpolated defect value
color reconstruction et cetera
dsp pipeline

Figure 3. Flow diagram of algorithm.



(a) Non-normalized points.

(b) Normalized points with edge direction shown.

Figure 4. Algorithm processing steps. The pixels neighboring the defect in (a) are normalized to the color plane of the defect in (b). The edge direction is located and also shown in (b). The black pixels in (a) and (b) indicate the pixel is not involved in the calculations. Note that only a 5×5 region of the whole 7×7 kernel is visible.

2. The sensor defect map is checked to determine if $a[m_0, n_0]$ is part of a column defect. If true, the vector d_1 is thrown out, leaving $d_i[n]$ with $i = \{2, 3, 4\}$.

3. Next, each point n in all vectors i of $d_i[n]$ is checked for uncorrected defect points at $n = \{1, 2, 3\}$. Because the defects are sequentially corrected, we know that any defect in vector points $n < 0$ have already been corrected. If a defect exists in one of the vectors, the point is locally interpolated depending on its location specified as n_{defect}.

$$d_i[n_{defect}] = \begin{cases} d_i[n_{defect} - 2], & \text{if } n_{defect} = 3, \\ d_i[n_{defect} + 2], & \text{if } n_{defect} = 1, \\ d_i[-n_{defect}], & \text{otherwise.} \end{cases} \tag{5}$$

   The interpolation may appear awkward but it tries to ensure that a step function on one side of the vector is not averaged into the other side. This makes certain that the vector will not be falsely selected as the edge direction later in the algorithm.

4. Next, the vector points $d_i[-1]$ and $d_i[+1]$ (those nearest to the defect) are normalized to the color plane of the defect. This is done by calculating the derivative of the non-defect color plane on each side of $d_i[0]$ and applying it to the nearest sampled point from the same color plane as the defect (points $d_i[-2]$ and $d_i[+2]$). Note that the algorithm does not need to know the actual filter color of a given pixel — it is only aware that the Bayer pattern samples two colors in any given direction. (It is possible to account for the special sampling of green along the diagonals but it has little effect on the final result.)

   The procedure for each vector d_i is carried out as follows :

   (a) The directional derivatives ∂_i on each size of the defect $d_i[0]$ are calculated as,

$$\partial_i[-1] = \frac{d_i[-1] - d_i[-3]}{2} \tag{6}$$

$$\partial_i[+1] = \frac{d_i[+1] - d_i[+3]}{2}. \tag{7}$$

   (b) The directional derivative is used to correlate the nearest points to the defect color plane. Therefore, we can define new vectors that estimate the sample points with respect to the defect color plane. Let the new *normalized* vectors $\hat{d}_i$ be defined as,

$$\hat{d}_i[-1] = d_i[-2] + \partial_i[-1] \tag{8}$$

$$\hat{d}_i[1] = d_i[+2] + \partial_i[+1]. \tag{9}$$

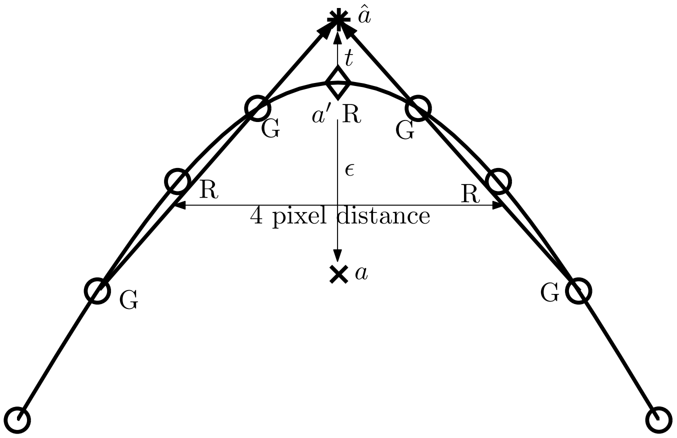


Figure 5. Reason for t aliasing tolerance in correction error test. If the limit for the distance from the defect pixel (a) to the defect correction algorithm's estimate ($\hat{a}$) is restricted to the maximum defect deviation ϵ, the estimate $\hat{a}$ would be rejected. However, the estimate $\hat{a}$ is nearer to the true scene value a'. Thus, the tolerance t must be added to ensure that defect points occurring at apex locations are not rejected when they are acceptable estimates.

(c) Using the normalized points found in the previous step, we may substitute $d_i[-1] = \hat{d}_i[-1]$ and $d_i[1] = \hat{d}_i[+1]$.

The process of normalizing the non-defect color plane points $d_i[-1]$ and $d_i[+1]$ is illustrated in Figure 4.

5. Next, we may determine an edge biased weight ξ_i that is based on the difference between $d_i[+1]$ and $d_i[-1]$. We can define a measure of a direction vector's alignment to the edge at $a[m_0, n_0]$ by,

$$\xi_i = \left(1 - \frac{(\delta_i)^k}{\sum_i (\delta_i)^k}\right) / (I - 1), \tag{10}$$

where

$$\delta_i = |d_i[-1] - d_i[+1]|. \tag{11}$$

The exponent k can be adjusted to modify the algorithm's sensitivity to the differences in $d_i[n]$, where sensitivity increases as k increases. The number of vectors in use is represented by I. The vector aligned closest to the edge direction (i.e. the minimum $|d_i[-1] - d_i[+1]|$) would have the maximum ξ_i weight.

6. Finally, the intensity value of the defect point $a[m_0, n_0]$ may be estimated with using the weight factors ξ_i to yield $\hat{a}[m_0, n_0]$,

$$\hat{a}[m_0, n_0] = \sum_i \xi_i \left(\frac{d_i[-1] + d_i[+1]}{2}\right). \tag{12}$$

4.1. Correction Error Test for Correlated Defects

Up to this point, no assumptions have been made about the correlation between a defect's value and the scene. The adaptive defect correction algorithm does not utilize the defect intensity value during the calculation of $\hat{a}[m_0, n_0]$ to avoid false influences as experienced by Smith.[4] However, if information about a defective pixel is at hand, it is possible to check the accuracy of the correction.

Sijde, Dillen, and Langen[11] address this situation with a 'levels test' to ensure that the defect interpolation method does not produce an estimate of the scene value that is wore than the defect value itself. Essentially, if the defect pixel is known to vary less than a certain amount from the actual scene value, it is possible to determine if the correction estimate is more or less satisfactory than the defect value itself.

The logic behind such a test for estimated value is based on the human visual system and the Nyquist Theorem. At times, the defect value may be acceptable to the observer because the human visual system is adaptive with respect to scene reflectances and frequency. For example, if an observer looks at a 18% gray reflection card with a small 25% reflecting square in the middle, the card will look dark and the patch will look rather light — the square will be readily noticeable because it occurs inside a low frequency region (the gray card is actually just a DC reflectance). But, if the 25% reflectance square is placed in the middle of another card with reflectance that is rapidly varying from 2–95% reflectance, the patch will now appear a dark gray; it will *not* be as readily apparent as in the 18% gray card. In fact, if the spatial frequency is moderately high, a very small square may not be visible at all). We may relate the 25% reflectance square to a defective pixel — a defect in a low-frequency region is readily apparent, while one in a high-frequency region is less so or may not even be visible.

The defect correction algorithm estimates the intensity value of the defective pixel by averaging neighboring information. By the Nyquist Theorem, the highest nonaliased frequency sampled in red, green and blue by the Bayer CFA is equal to one half the maximum frequency in a given direction. Since the sample frequency of blue and red is every other pixel, the highest nonaliased frequency (the Nyquist frequency) would be 1/4 cycle per pixel. Once a pixel becomes defective the Nyquist frequency is further decreased in this region to 1/8 cycle per pixel. Therefore, in the adaptive defect correction algorithm, it is possible that the data kernel sampled an image region with an unaliased frequency of 1/4 cycle per pixel. Because the algorithm calculates gradients at a two-pixel space, this region would yield a gradient of 0 due to aliasing — we cannot correlate the neighboring data. This can result in an estimate for the defect value off a large amount. However, if the defect pixel is known only to deviate by 10%, the actual defective pixel value may have less error than the estimated defect value. If we can quantify the typical magnitude of a pixels' defect then we may use this to decide whether the estimated defect value or the defect value itself is nearer to the true value of the original scene.

When image frequencies in the defect region exceed the Nyquist frequency, aliasing may cause the estimate to have a far greater error than the defect value itself. An observer will tend to be unaware of small magnitude defects in high-frequency regions compared to low-frequency regions. Therefore, it is preferable to allow the defect value to pass unchanged if it is known that its error is less than the estimated error, which shall only occur in high-frequency regions.

We may determine this *correction error test* as follows: let the defective pixel value in the image $a[m_0, n_0]$ be denoted as a, the true, but unknown, scene value for the defect denoted as a', and the dynamic range of the image given to be C (where $C = 255$ for an 8-bit image). Then, the adaptive defect correction algorithm produces an estimate for the defect represented by $\hat{a}$.

In a factory, setting the defect pixel value a can be sampled N times over a range of illuminance levels to determine the deviation of a from a'. We may define the maximum fractional defect deviation tolerance ϵ from the samples a_i as:

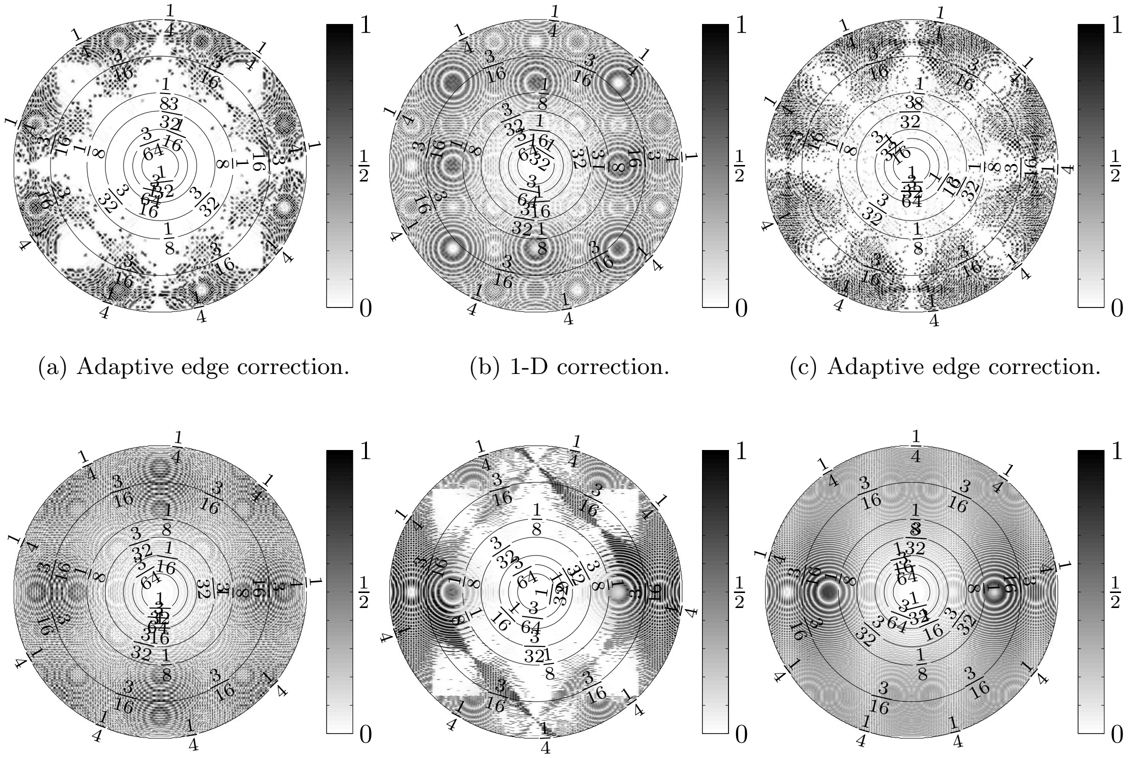
$$\epsilon = \frac{\max|a_i - a'|}{C}. \tag{13}$$

We could simply set the correction error test to reject $\hat{a}$ when $|\hat{a} - a| > \epsilon$. However, this relation is flawed, $\hat{a}$ would be rejected when its error is actually less than the error of a at some frequencies. A correct relation must also account for aliasing due to the defective pixel which is treated as missing data in the correction algorithm.

In regions where a single pixel defect occurs, the Nyquist frequency (f_N) is reduced to 1/8 cycle per pixel. Figure 5 shows the situation for a single defective red pixel. The red pixels are now sampled every 4 pixels. Since the adaptive edge algorithm uses gradients to estimate the defect pixel value, the maximum possible unaliased gradient would be 1/16. When the defect pixel a occurs at the apex of the signal (as shown in Figure 5) the defect correction algorithm will overestimate its value — but its overestimate can never exceed $1/16C$. Although, in theory, the overestimate would never exceed the $f_N C$, in practice we must also recognize that no system has a MTF equal to one for all frequencies. Therefore, the formal definition of the aliasing tolerance t may be defined as:

$$t = f_N \times \text{MTF}(f_N). \tag{14}$$

In words, the added tolerance t is a function of the Nyquist frequency in the region and the MTF at that frequency.



(a) Adaptive edge correction.

(b) 1-D correction.

(c) Adaptive edge correction.

(d) 1-D correction.

(e) Adaptive edge correction.

(f) 1-D correction.

Figure 6. Error images for single pixel defects (a) & (b), 2×2 pixel cluster defects (c) & (d), and column defect correction (e) & (f). Contour lines indicate the input zone plate frequency in cycles/pixel. Shading level indicates the magnitude of error in corrected defect.

Finally, the formal definition of the correction test may be defined to choose the new value of the defect location $a[m_0, n_0]$ from the old defect value a or the estimate $\hat{a}$ via:

$$a[m_0, n_0] = \begin{cases} \hat{a}, & \text{if } |\hat{a} - a| \leq \epsilon C + tC, \\ a, & \text{otherwise.} \end{cases} \tag{15}$$

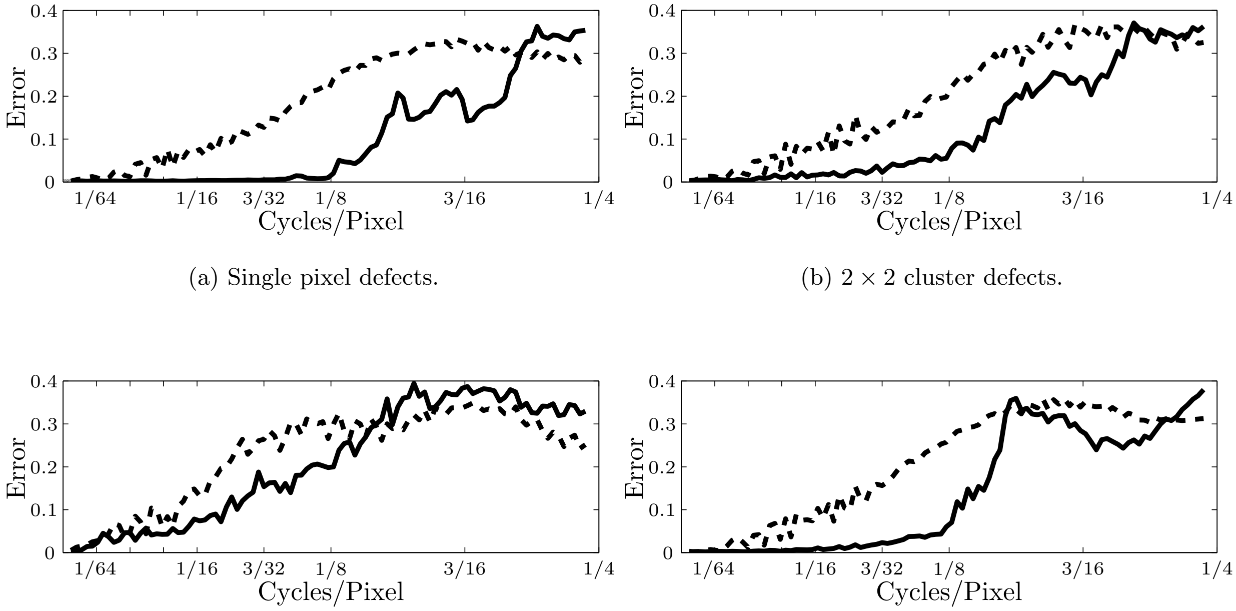
5. EXPERIMENTAL RESULTS

To obtain quantitative information related to the error in defect correction, an experiment was developed where the error could be measured as a function of input frequency, angle, and defect type.

The selected input target image for measuring correction accuracy was a grey-scale zone plate containing frequencies from 0 to $^1/_4$ cycle per pixel. The types of defects tested were single pixels, 2×2 pixel clusters, 3×3 pixel clusters, and single columns. The defects were implanted in the target at 100% magnitude (simulating the worst-case situation when defective pixel sites have no correlation to the scene value) at a high density and processed by the adaptive edge algorithm.

For comparison, the same tests were conducted with using a simple one-dimensional (1-D) defect correction algorithm. This algorithm simply averages the two closest known good pixels of the same color plane in the horizontal register.

The results of both defect correction methods can be seen in the error images shown in Figure 6. Figure 7 provides a nice graphically compares the two algorithms.



(a) Single pixel defects.

(b) 2 × 2 cluster defects.

(c) 3 × 3 cluster defects.

(d) Column defects.

Figure 7. Comparison of algorithm defect mean correction error as a function of frequency. Solid lines indicate adaptive edge correction, dashed lines indicate 1-D correction. Error is expressed as a fractional value ranging from 0 to 1. Without defect correction the defect error would plot at 1 for all frequencies.

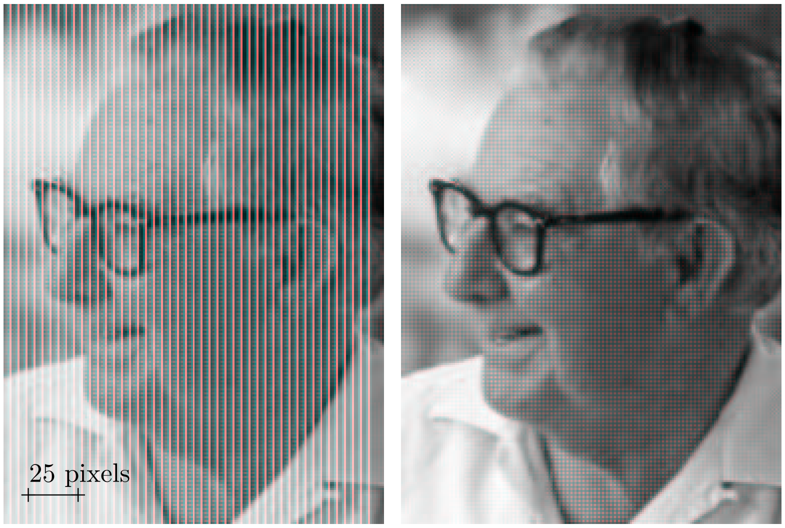
In the error images, the adaptive edge algorithm has a much smaller visible error than the one-dimensional algorithm for low frequencies. Furthermore, the adaptive edge algorithm has high accuracy in high frequencies when an edge direction is closely aligned to the direction of frequency propagation in the target.

The adaptive algorithm is extremely successful at low to moderate frequencies, while at high frequencies the results tend to be either good or poor. The one-dimensional algorithm tends to produce mediocre but consistent results. Clearly, there is a trade off for correlating the pixels' color planes; our estimates become erratic where frequencies exceed the Nyquist rate for the defect color plane.

To reduce the complexity in comparing the two algorithms, Figure 7 plots the mean error against frequency (the mean error for all angles has been computed at each frequency). The graphs show that the adaptive algorithm outperforms the one-dimensional method for all tested types of defects. Table 1 illustrates the gain in accuracy by listing the maximum correctable frequency with a mean error $\leq 10\%$. For the tested defect types, the adaptive algorithm correct frequencies 1.88x higher than the one dimensional algorithm.

To visually illustrate the effectiveness of the adaptive edge algorithm with natural images, single-column defects were implanted into a sample image at a four column interval. Column defects were chosen because they render the vertical edge direction vector useless — making them one of the most challenging types of defects to correct . Figure 8(a) shows a sample Bayer image that contains defects in every fourth column. The image was processed by the algorithm and the results can be seen in Figure 8(b). The results show the algorithm is very accurate; even when the edge direction is vertical the correction appears good.

The correction error test further limits the maximum error in the estimated defective pixel intensity to the pre-measured maximum defective pixel intensity deviation. The effectiveness of the test is illustrated by comparing the correction errors for single-pixel defects at a variety of maximum defect intensity deviations, as shown in Figure 9.



(a) Bayer image with defects.

(b) Corrected Bayer image.

Figure 8. Effect of adaptive edge algorithm on sample image with column defects every fourth pixel.

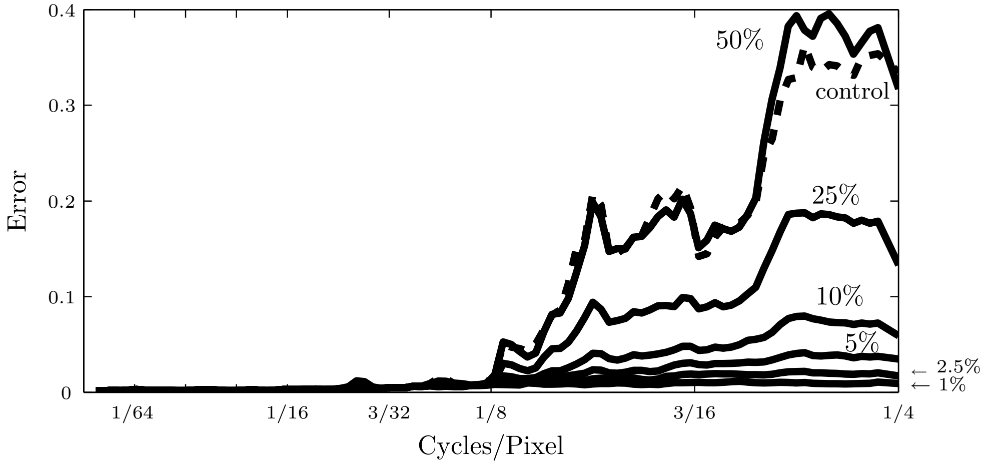


Figure 9. Effect of correction error test on varying defect magnitudes corrected by the adaptive defect correction algorithm. Defect magnitudes are expressed in percent of the image's dynamic range. The control line represents the mean error without the correction error test. Results are representative of single pixel defects. Error is expressed as a fractional value ranging from 0 to 1.

Table 1. Average maximum frequency that each algorithm can correct 100% defects up to with mean error ≤ 10%.

	MAX FREQUENCY (CYCLES/PIXEL)		
DEFECT TYPE	ADAPTIVE EDGE	1-D	IMPROVEMENT[a]
single pixel	0.140	0.075	1.87x
2×2 cluster	0.130	0.070	1.86x
3×3 cluster	0.070	0.042	1.67x
single column	0.130	0.063	2.01x
double column	0.094	0.047	2.00x

[a] Increase in frequency the adaptive edge algorithm can handle compared to the one dimensional algorithm.

6. CONCLUSION

A novel approach for correcting defects by using natural image properties has been presented here. The algorithm is unique because it is designed to utilize raw Bayer data — allowing it to be performed before the color reconstruction algorithm without other modification. The accurate correction results shown in Figure 8 give evidence of the substantial improvements in defect value estimation compared to simple one-dimensional algorithms. If the algorithm is compared to the one dimensional correction proposed by Sijde, Dillen, and Langen,[11] it is clear that the edge vectorization and color correlation are more complicated. However, the increase in complexity provides a substantially better estimate of the defect pixel value.

ACKNOWLEDGMENTS

This work was conducted as part of a work/education experience through Philips Image Sensors Semiconductors/DALSA Professional Imaging of Eindhoven, The Netherlands and the Imaging and Photographic Technology Department of the Rochester Institute of Technology, Rochester, New York.

REFERENCES

1. D. D. Pape, "Defect correction for CCD and CID imagers." EU Patent 0-458-030-A1, 1991.
2. W. T. Fearnside, "Defect correction in solid state imaging." US Patent 4,535,359, 1985.
3. T. Komatsu and T. Saito, "A high-resolution image aquisition method with defect-pixel recovery for solid-state image sensors," *IEEE* , pp. 1053–1056, 2001.
4. S. G. Smith, "Defect correction in electronic imaging system." EU Patent 1-003-332-A2, 2000.
5. O. Rashkovskiy and W. Macy, "Method of determining missing color values for pixels in a color filter array." US Patent 6,181,376 B1, 2001.
6. B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Massachusetts, 1995.
7. B. E. Bayer, "Color image array." US Patent 3,971,065, 1976.
8. C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients." US Patent 5,373,322, 1994.
9. D. R. Cok, "Signal processing method and apparatus for sampled image signals." US Patent 4,630,307, 1986.
10. G. D. Finlayson, P. M. Hubel, and S. Hordley, "Color by correlation," *IS&T* **Fifth Color Imaging Conference: Color Science**, pp. 6–11, 1997.
11. A. van der Sijde, B. G. Dillen, and R. Langen, "Image sensor signal defect correction." US Patent Application 0012476 A2, Jan. 2002.