https://doi.org/10.2352/ISSN.2470-1173.2017.15.DPMI-088

Robust Defect Pixel Detection and Correction for Bayer Imaging Systems

Noha El-Yamany; Intel Corporation; Tampere, Finland

Abstract

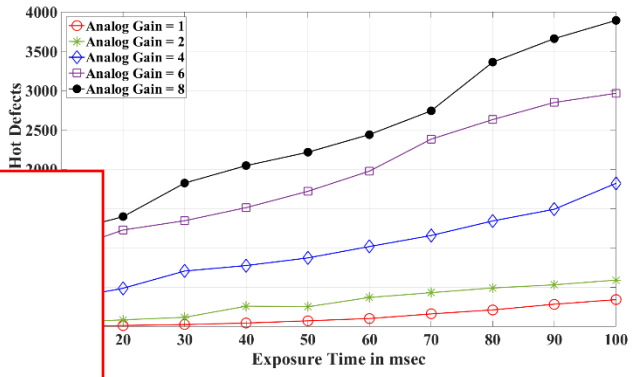
Defective pixels degrade the quality of the images produced by digital imagers. If those pixels are not corrected early in the image processing pipeline, demosaicing and filtering operations will cause them to spread and appear as colored clusters that are detrimental to image quality. This paper presents a robust defect pixel detection and correction solution for Bayer imaging systems. The detection mechanism is designed to robustly identify singlets and couplets of hot pixel, cold pixels or mixture of both types, and results in high defect detection rates. The correction mechanism is designed to be detail-preserving and robust to false positives, and results in high image quality. Both mechanisms are computationally cheap and easy to tune. Experimental results demonstrate the aforementioned merits as well as the solution outperformance of conventional correction methods.

Introduction

Digital imagers are widely used in everyday produ
smart phones, tablets, phablets, notebooks, cameras
wearables. Small form factor is the trend, and so the us
imaging systems having small sensors with small pix
inevitable. An important problem in the image processi
[1] of those systems is the detection and correction of th
pixels [2-12], which develop during and after the fabric
digital imager. The rates of those defects increase as th
decreases [2-3]. If defect pixels are not corrected early i
processing pipeline, demosaicing and filtering operations will cause them to spread and appear as colored clusters that degrade the image quality (IQ), as shown in figures (2), (3) and (6).

Defect pixel identification can be performed at the factory during the digital imager manufacturing. Dark-frame calibration is typically performed to identify the hot pixels-Permanently saturated ones, or those that respond more strongly to illumination. Light-field calibration is performed to identify the cold pixels-Permanently black ones, or those that are less sensitive to illumination. However, such calibrations would require complex and expensive systems at the factory. In addition, the rates and visibility of the hot defects, the dominant defect type [2-7], increase as the ISO/exposure time increases [2-5] (as in low-light and ultra-low-light conditions), as illustrated in figure (1) for a certain low-cost 8 megapixel (MP) sensor. Furthermore, defect pixels develop over the imager lifetime [2-7], and, therefore, the static defect maps found by calibrations will not faithfully represent the sensor defective pixels over time. Hence, an algorithm that constantly identifies the defect locations and updates the sensor defect map is essential for high IQ.

Defect pixel correction (DPC) is typically performed on the raw data coming from the sensor at an early stage in the image processing pipeline [1]. It is thus crucial that the correction mechanism does not introduce any artifacts that could later be amplified by the subsequent processing blocks, and hence degrade the IQ. Because the defect pixels are caused by a random process [4-6], they can appear as singlets, couplets, triplets or even clusters, of hot pixels, cold pixels or mixture of both types. There is generally more interest in the correction of singlets and couplets of defects, since imagers with triplets/clusters of defects are typically rejected in the production line. In addition, singlet/couplet correction would have little or no impact on the IQ, whereas triplet/cluster correction could affect the IQ negatively, especially at small features in the image. From the IQ tuning perspective, it is highly desirable that the DPC solution in the image processing pipeline would require small/no tuning effort. This guarantees consistent performance and IQ and faster turnaround time in an IQ tuning project.



t defect rates for an example low-cost 8MP sensor using dark-sis

Several solutions for defect pixel detection and correction have been published in the literature [7-12] or patented. In [7] a hot defect pixel tracing algorithm was presented. It determines the presence/absence of defect pixels by accumulating Bayesian statistics collected from a sequence of images taken over days. In [8] a hot defect pixel correction algorithm was proposed. It exploits a hot defect model calculated from dark-frame calibration and performs linear interpolation for defect correction. Bounding min-max filtering of different kernel sizes and median filtering were exploited for defect pixel correction in [9] and [10], respectively. In [11] a more complex, multi-step interpolation algorithm based on natural image properties was introduced for defect correction. In [12] a highly complex, sparsity-based iterative interpolation scheme was proposed for the correction of defects. However, to the current knowledge, there is no published work that explicitly addresses the problem of automatic defect pixel detection and correction in a unified framework, offering a robust and computationally-efficient solution for real-time and resource-constrained imaging systems.

In this paper, a robust defect pixel detection and correction solution for Bayer [13] imaging systems is presented. The detection mechanism is designed to robustly identify singlets and couplets of hot pixels, cold pixels or mixture of both types, and results in high defect detection rates. The correction mechanism is designed to be detail-preserving and robust to false positives, and results in high IQ. Both mechanisms are computationally cheap and easy to tune.

In addition, the proposed solution does not require multi-image processing, nor pre-knowledge of the sensor defect parameters nor pre-calibration to identify the defect positions. The aforementioned merits make the proposed solution attractive for diverse real-time Bayer imaging systems.

The rest of the paper is organized as follows. First, the detection and correction mechanisms are described. The computational cost is then given for example design choices. Defect correction results are then demonstrated and discussed for images with simulated and real defects. Finally, conclusions are provided.

Algorithm Description

The Detection Mechanism

The proposed solution operates on the raw data coming from the Bayer sensor. Let $I(i,j)$ denote the pixel at the position (i,j) in the input raw Bayer image I, where $i = [1, 2, \ldots, H]$; $j = [1, 2, \ldots, W]$; H and W are the height and width of I, respectively. Let $O(i,j)$ denote the pixel at the position (i,j) after DPC. Normalized intensities within the range [0.0,1.0] are assumed for both I and O. Let $I_{avg}(i,j)$ denote the robust local average estimate of the same-color neighbors in an S×S window centered at (i,j). A pixel $I(i,j)$ is identified as a defect *if two conditions are met*.

Condition (A): The pixel is significantly different from its same-color neighbors in the S×S Bayer window centered at that pixel.

$$\textbf{\textit{For hot pixels}: } I(i,j) > (1+M_1)\times I_{avg}(i,j) \quad (1)$$

$$\textbf{\textit{For cold pixels}: } I(i,j) < (1-M_1)\times I_{avg}(i,j) \quad (2)$$

Condition (A) is rather intuitive, since a defect pixel responds abnormally to illumination, and thus becomes visibly different from its neighbors. M_1 is an algorithm parameter, which by design denotes the *detection strength*, and $M_1 \in (0.0,1.0)$. The smaller M_1 is, the stronger the detection is. I_{avg} should be *insensitive* to the presence of a hot/cold/mixed couplet in the same color channel, in order to enable the algorithm to robustly identify the defect, whether it is a singlet or belongs to a couplet in the same color channel. The α-trimmed mean [14] is one example of how to calculate I_{avg}, because the highest and lowest $\alpha/2$ values are discarded prior to computing the local average. For $\alpha = 2$ and S = 5, $I_{avg}(i,j)$ could then be calculated as follows

1. Sorting the eight same-color neighbors of $I(i,j)$, $\Omega(i,j)$, where

$$\Omega(i,j) = \{I(i-2,j-2), I(i-2,j), I(i-2,j+2), I(i,j-2), I(i,j+2), I(i+2,j-2), I(i+2,j), I(i+2,j+2)\} \quad (3)$$

2. Discarding the maximum and minimum values
3. Calculating $I_{avg}(i,j)$ by averaging the remaining six values

Condition (B): In the 3×3 Bayer window centered at the pixel, the local brightness difference at the pixel is significantly higher than the smallest local brightness difference for each color channel, for hot pixels. Conversely, for cold pixels, the local brightness difference at the pixel is significantly lower than the largest local brightness difference for each color channel.

For hot pixels:

$$d_{lb}(i,j) > M_2\times\min(d_{lb}(i,j-1), d_{lb}(i,j+1)) \quad (4)$$

$$d_{lb}(i,j) > M_2\times\min(d_{lb}(i-1,j), d_{lb}(i+1,j)) \quad (5)$$

$$d_{lb}(i,j) > M_2\times\min(d_{lb}(i-1,j-1), d_{lb}(i-1,j+1), d_{lb}(i+1,j-1), d_{lb}(i+1,j+1)) \quad (6)$$

For cold pixels:

$$d_{lb}(i,j) < M_2\times\max(d_{lb}(i,j-1), d_{lb}(i,j+1)) \quad (7)$$

$$d_{lb}(i,j) < M_2\times\max(d_{lb}(i-1,j), d_{lb}(i+1,j)) \quad (8)$$

$$d_{lb}(i,j) < M_2\times\max(d_{lb}(i-1,j-1), d_{lb}(i-1,j+1), d_{lb}(i+1,j-1), d_{lb}(i+1,j+1)) \quad (9)$$

All the conditions in (4)-(6) must hold for condition (B) to be true for hot pixels. Similarly, all the conditions in (7)-(9) must hold for condition (B) to be true for cold pixels. $d_{lb}(k,l)$ is the local brightness difference at (k,l), which is calculated as $d_{lb}(k,l) = I(k,l)-I_{avg}(k,l)$.

Because a defect pixel is visibly different from its neighbors, it is expected that the local brightness difference measured at the defect location will be considerably different from that at the other pixel locations in a very small neighborhood of the defect. Condition (B) is imposed, therefore, to differentiate between a defect and a small image feature/detail, thus enabling the algorithm to be robust against false positives. The use of the min operation in (4)-(6), for hot pixels, and the max operation in (7)-(9), for cold pixels, enables the algorithm to robustly identify the defect, whether it is a singlet or belongs to a hot/cold/mixed couplet in different color channels. M_2 is an algorithm parameter, which by design denotes *false positives control*, and $M_2 \in [1.0, U]$, where U is an upper bound. The larger M_2 is, the weaker the detection is and the less the false positives are.

The Correction Mechanism

Defect pixel correction is performed on the raw data at an early stage in the image processing pipeline. Hence, it is essential that defect correction does not introduce artifacts that could later be amplified by subsequent processing. Conventional correction based on linear interpolation of the defect pixel same-color neighbors [8] is not a robust approach, since one or more of the neighboring pixels might be defective too. Figure (2) depicts an example defect correction based on averaging of the same-color neighbors of the identified defects. As shown, the hot couplets were not corrected cleanly, and subsequent demosaicing and filtering in the pipeline caused the correction artifacts to spread and degrade the IQ.

Correction using the median of the same-color neighbors [10] of the defect pixel would be robust to the presence of couplets, but it could result in noticeable artifacts when the defects or false positives fall on the high-contrast edges or corners [12]. An example defect correction using the median of the same-color neighbors of the identified defects is depicted in figure (3). As shown, the mixed couplet was corrected cleanly. The false positives at the high-contrast edges, however, were corrected, leaving noticeable artifacts that were later amplified by subsequent processing in the pipeline, resulting in an unacceptable IQ.

In the proposed solution, the correction mechanism aims to replace the detected defect pixel with a robust, detail-preserving estimate. To achieve this goal, the responses of the identified defect to a set of directional filters are first calculated to find the image feature direction at the pixel location. A directional estimate along the feature direction is then used to correct the pixel. If the directional estimate does not result in a clean correction, for example because the detected defect pixel belongs to a couplet along the feature

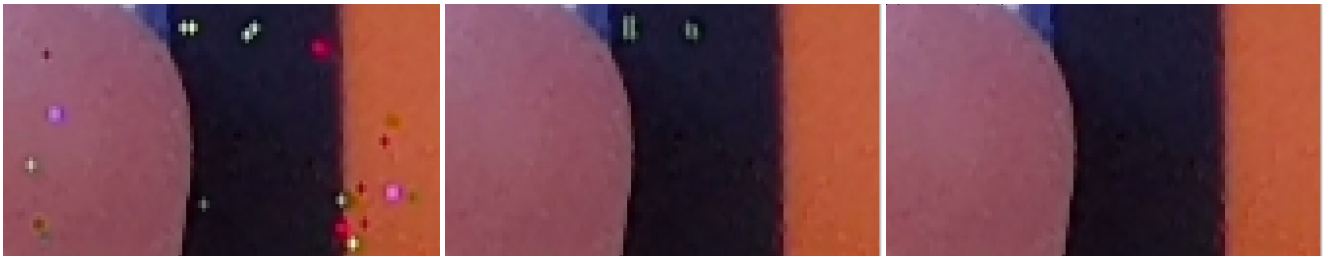


Figure 2. The result of different defect correction mechanisms for an image with simulated defects at (M_1, M_2) = (0.4, 10.0)
Left: No correction, middle: averaging-based correction, right: proposed correction (M_3 = 0.4)

Figure 3. The result of different defect correction mechanisms for an image with simulated defects at (M_1, M_2) = (0.4, 10.0)
Left: No correction, middle: median-based correction, right: proposed correction (M_3 = 0.4)

direction, a robust non-directional estimate is used instead. This strategy preserves the image details, and enables the algorithm to correct couplets properly and to be robust against false positives.

Figure (4) depicts an example set of directional filters. The selection of the weights of those filters is motivated by line detection [14]. The responses of the identified defect pixel $I(i,j)$ to the filters can be calculated via 2D convolution.

$$R_h(i,j) = I(i,j) \otimes F_h \tag{10}$$

$$R_v(i,j) = I(i,j) \otimes F_v \tag{11}$$

$$R_{45}(i,j) = I(i,j) \otimes F_{45} \tag{12}$$

$$R_{135}(i,j) = I(i,j) \otimes F_{135} \tag{13}$$

R_h, R_v, R_{45} and R_{135} are the filter responses. The feature direction at the identified defect pixel $I(i,j)$ can be restricted to one of the directions indicated by the filters, or can be computed as a function of all directions. If only the directions in figure (4) are allowed, and if $R_{max}(i,j)$ denotes the maximum of the responses in (10)-(13), and $R_{min}(i,j)$ denotes the minimum of them, then the image feature direction at the identified defect pixel $I(i,j)$ is found as the one whose filter response is the maximum for hot pixels (or the minimum for cold pixels.) This procedure is described below.

For hot pixels:

If $R_h(i,j) = R_{max}(i,j) \rightarrow$ *Feature direction is horizontal*
If $R_v(i,j) = R_{max}(i,j) \rightarrow$ *Feature direction is vertical*
If $R_{45}(i,j) = R_{max}(i,j) \rightarrow$ *Feature direction is 45° diagonal*
If $R_{135}(i,j) = R_{max}(i,j) \rightarrow$ *Feature direction is 135° diagonal*

For cold pixels:

If $R_h(i,j) = R_{min}(i,j) \rightarrow$ *Feature direction is horizontal*
If $R_v(i,j) = R_{min}(i,j) \rightarrow$ *Feature direction is vertical*
If $R_{45}(i,j) = R_{min}(i,j) \rightarrow$ *Feature direction is 45° diagonal*
If $R_{135}(i,j) = R_{min}(i,j) \rightarrow$ *Feature direction is 135° diagonal*

-1	0	-1	0	-1
0	0	0	0	0
2	0	2	0	2
0	0	0	0	0
-1	0	-1	0	-1

F_h

-1	0	2	0	-1
0	0	0	0	0
-1	0	2	0	-1
0	0	0	0	0
-1	0	2	0	-1

F_v

-1	0	-1	0	2
0	0	0	0	0
-1	0	2	0	-1
0	0	0	0	0
2	0	-1	0	-1

F_{45}

2	0	-1	0	-1
0	0	0	0	0
-1	0	2	0	-1
0	0	0	0	0
-1	0	-1	0	2

F_{135}

Figure 4. Example set of directional filters

If the responses to two/more of the filters are the same as the maximum value for hot pixels (or the minimum value for cold pixels), the robust non-directional estimate is used in the correction.

The directional estimate, $I_D(i,j)$, along the image feature direction can be calculated as follows

$$\textbf{\textit{Horizontal}}: I_D(i,j) = I_h(i,j) = 0.5 \times (I(i,j-2) + I(i,j+2)) \tag{14}$$

$$\textbf{\textit{Vertical:}}\ I_D(i,j) = I_v(i,j) = 0.5 \times (I(i-2,j) + I(i+2,j)) \tag{15}$$

$$\textbf{\textit{45°:}}\ I_D(i,j) = I_{45}(i,j) = 0.5 \times (I(i-2,j+2) + I(i+2,j-2)) \tag{16}$$

$$\textbf{\textit{135°:}}\ I_D(i,j) = I_{135}(i,j) = 0.5 \times (I(i-2,j-2) + I(i+2,j+2)) \tag{17}$$

where $I_h(i,j)$, $I_v(i,j)$, $I_{45}(i,j)$ and $I_{135}(i,j)$ are directional averages. It is essential that the non-directional estimate, $I_{ND}(i,j)$, does not result in artifacts, which could later be amplified by subsequent processing in the pipeline. $I_{ND}(i,j)$ can be calculated as follows

$$\textbf{\textit{For hot pixels}}: I_{ND}(i,j) = I_{2max}(i,j) \tag{18}$$

For cold pixels: $I_{ND}(i,j) = I_{2min}(i,j)$ (19)

where $I_{2max}(i,j)$ and $I_{2min}(i,j)$ denote the second maximum and second minimum of Ω (i,j), respectively. The non-directional estimate in (18)-(19) results in a robust correction, especially if the pixel belongs to a hot/cold/mixed couplet along the identified direction.

As mentioned earlier, the algorithm checks if the directional estimate would result in a clean correction; i.e., it is an estimate that is not visibly different from its same-color neighbors. If it does not result in a clean correction, the robust non-directional estimate is used instead. This procedure is described below.

For hot pixels:

$$\begin{aligned} &\text{If} \quad I_D(i,j) > (1+M_3)\times I_{avg}(i,j), \text{ then } O(i,j) = I_{ND}(i,j) \\ &\text{Else} \quad O(i,j) = I_D(i,j) \end{aligned} \tag{20}$$

For cold pixels:

$$\begin{aligned} &\text{If} \quad I_D(i,j) < (1-M_3)\times I_{avg}(i,j), \text{ then } O(i,j) = I_{ND}(i,j) \\ &\text{Else} \quad O(i,j) = I_D(i,j) \end{aligned} \tag{21}$$

M_3 is an algorithm parameter, which by design balances the selection between the directional and non-directional corrections, and $M_3 \in (0.0,1.0)$. Defect correction is performed sequentially, in a raster-scan order. Hence, the corrected pixel is used as input for the next detection. If a pixel $I(i,j)$ is not identified as a defect, then no correction is performed on that pixel; i.e. $O(i,j) = I(i,j)$.

Computational Cost

The computational cost of the proposed solution varies with the choices made about S, the robust local average estimate, the directional filters, and the directional and non-directional estimates. However, the cost is generally small, due to the algorithm simplicity. To give an idea about the low complexity of the proposed solution, the cost will be presented here given some assumptions about the detection and correction mechanisms.

For the detection, let us assume that S = 5 and that α-trimmed mean with $\alpha = 2$ is used for calculating I_{avg}. For the correction, let us assume that the filter set in figure (4) is used and that only four directions are allowed. Also, let the directional and non-directional estimates be defined as in (14)-(19). Examining the coefficients of the filters in figure (4), comparing the responses in (10)-(13) is in fact equivalent to comparing the directional averages, $I_h(i,j)$, $I_v(i,j)$, $I_{45}(i,j)$ and $I_{135}(i,j)$. This simple optimization reduces the complexity of finding the image feature direction.

Given the above assumptions and optimization, the maximum per-pixel cost for defect detection and correction is depicted in table (1). In the cost estimate presented, it is assumed that a min or a max operation is implemented using 1 CMP and 2 MUXs. Also, one-time per-image calculations are not counted.

Table 1: The per-pixel cost for example design choices

Operation	Detection	Correction	Total
ADD	5	4	9
SUB	1	0	1
MUL	4	1	5
CMP	34	8	42
MUX	50	6	56
SHIFT	1	4	5
AND	6	0	6

The Tunable Parameters and Tuning Strategy

The algorithm has three tunable parameters. By design, M_1 controls the detection strength, M_2 controls the amount of false positives, and M_3 balances the selection between the directional and non-directional corrections. The IQ tuner would need to select the M_1 and M_2 values that result in the best tradeoff between defect correction and false positives. M_3 should be tuned to be the same as (or close to) the value selected for M_1. Tuning M_3 is especially important in the presence of couplets, as the directional estimate might be biased toward the couplet direction. It is also possible to reduce the tunable parameters to only two, by setting $M_3 = M_1$.

At high ISO/exposure time settings, as in low-light and ultra-low-light conditions, the visibility/rate of the defects increase, and thus a small M_1 value and a big M_2 value should be used. Conversely, at low ISO/exposure time settings, as in bright scenes, the visibility/rate of the defects decrease, and thus a high M_1 value and a small M_2 value should be selected. M_3 should be tuned in the same manner as M_1. Linear interpolation of the three tunable parameters is performed to calculate their values at the capture conditions between the high and low ISO/exposure time settings. For a given sensor, it is also possible that the IQ tuner finds a *universal parameter set* that achieves the best IQ and balance between defect correction and false positives across different capture settings.

Experimental Results

For quantitative analysis and comparison purposes, the detection and correction results for a set of 10 test images with simulated defects are first presented and discussed. The images cover different scene contents and capture conditions, and each has a resolution of 3122(H)×4208(W); 13MP. Two of the images are partially depicted in figures (2) and (3). Using a defect pixel simulation model, hot and cold pixels were randomly generated and injected in each raw image. The simulated defects range from mild to strong ones, as depicted in figures (2) and (3). The percentage of the injected defects is approximately 0.1% of the image size, to increase the likelihood of creating hot/cold/mixed couplets. The design choices mentioned previously were used in generating the results presented.

Let N_d and N_i denote the number of the defects injected in the image, and the number of the defects correctly identified by the algorithm, respectively. The detection rate (DR) is then defined as N_i/N_d. Let N_t and N_f denote the total number of pixels in the image, and the number of the false positives, respectively. The false positive rate (FPR) is then defined as N_f/N_t. Let E_c and E_f denote the average absolute correction error, and average absolute false positive error, respectively, which are defined as follows

$$E_c = (1/N_i)\,\Sigma|I_o(k,l) - I_c(k,l)| \;\forall\; (k,l) \in \Psi \tag{22}$$

$$E_f = (1/N_f)\,\Sigma|I_o(k,l) - I_c(k,l)| \;\forall\; (k,l) \in \aleph \tag{23}$$

I_o is the original (ground-truth) raw image without the injected defects, and I_c is the result of DPC of the image with the injected defects. Ψ and $\aleph$ denote the collections of the positions of the N_i correctly identified defect pixels, and the N_f false positives, respectively. Table (2) depicts the DR and FPR values for the test images at different algorithm parameter values. The receiver operating characteristic (ROC) curves, averaged over the 10 test images, are depicted in figure (5). Tables (3) and (4) depict the E_c and E_f values, respectively, for the proposed solution for two M_3 values, 1.0 and 0.4, at $(M_1, M_2) = (0.4, 10.0)$, and for the averaging-based and median-based corrections at $M_3 = 0.4$.

Table 2: The detection and false positive rates for example algorithm parameter *values* (M_1, M_2, M_3)

Image Index	(0.8, 1.0, 1.0)		(0.6, 1.0, 1.0)		(0.4, 1.0, 1.0)		(0.4, 5.0, 1.0)		(0.4, 10.0, 1.0)		(0.4, 15.0, 1.0)	
	DR%	FPR%	DR%	FPR%	DR%	FPR%	DR%	FPR%	DR%	FPR%	DR%	FPR%
1	92.9306	0.0036	96.0345	0.0110	98.4876	0.0615	98.1359	0.0097	97.4853	0.0047	96.5796	0.0034
2	89.5727	0.0049	94.4963	0.0092	98.7299	0.0502	98.6985	0.0037	98.5809	0.0019	98.2360	0.0015
3	99.2108	0.0024	99.5768	0.0084	99.7712	0.0431	99.7712	0.0159	99.2909	0.0118	97.9641	0.0105
4	96.3141	0.0122	98.4598	0.0295	99.3412	0.1125	99.1720	0.0250	98.4420	0.0102	97.4359	0.0072
5	96.7236	0.0121	98.4152	0.0362	99.4124	0.2006	99.0474	0.0439	97.9256	0.0165	96.6257	0.0111
6	94.4199	0.0019	97.4727	0.0070	99.4495	0.0361	99.3160	0.0031	99.0742	0.0020	98.5153	0.0017
7	82.6592	0.0144	89.6824	0.1003	95.8603	0.7006	94.6358	0.0191	93.1980	0.0080	91.9577	0.0060
8	95.7929	0.0001	98.2227	0.0054	99.6136	0.1314	99.0899	0.0041	96.6601	0.0013	94.2474	0.0011
9	91.0690	0.0015	94.9382	0.0080	97.8552	0.0612	97.5549	0.0055	96.8342	0.0027	95.6932	0.0021
10	94.3148	0.0014	97.3210	0.0107	99.0363	0.0895	98.7003	0.0250	97.1176	0.0145	94.3678	0.0117

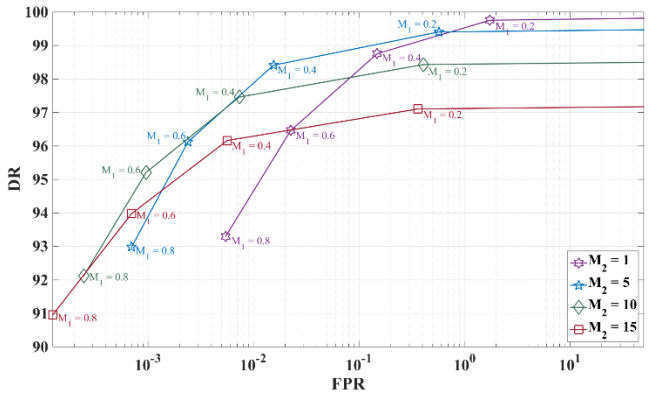


Figure 5. The average ROC curves for the proposed solution

Table 3: The average absolute correction error (E_c) for the proposed, averaging-based and median-based corrections

Image Index	(0.4,10.0,1.0)	(0.4,10.0,0.4)		
	Proposed	Proposed	Averaging	Median
1	0.3793	0.3796	0.3852	0.3854
2	0.3689	0.3691	0.3741	0.3741
3	0.4258	0.4259	0.4319	0.4321
4	0.3675	0.3677	0.3729	0.3731
5	0.3676	0.3678	0.3747	0.3749
6	0.3644	0.3647	0.3690	0.3691
7	0.3924	0.3928	0.4009	0.4014
8	0.3651	0.3654	0.3744	0.3749
9	0.3822	0.3823	0.3885	0.3888
10	0.3767	0.3769	0.3873	0.3875

Table 4: The average absolute false positive error (E_f) for the proposed, averaging-based and median-based corrections

Image Index	(0.4,10.0,1.0)	(0.4,10.0,0.4)		
	Proposed	Proposed	Averaging	Median
1	0.0498	0.0547	0.1145	0.1125
2	0.0197	0.0268	0.0828	0.0596
3	0.0329	0.0334	0.0518	0.0423
4	0.0592	0.0645	0.1087	0.1215
5	0.0480	0.0511	0.0891	0.0971
6	0.0256	0.0306	0.0966	0.0658
7	0.0224	0.0255	0.1011	0.1071
8	0.0186	0.0214	0.1025	0.0472
9	0.0454	0.0494	0.1076	0.0976
10	0.0417	0.0423	0.0672	0.0639

From the presented experimental results, a few observations are worth mentioning. Generally, the detection rate is high and the false positive rate is low. Most of the defects that were missed by the detection mechanism were of the mild type. The DR for image #7 was the lowest among the test images. That image contained high-frequency textured regions in which the defects were trapped. Since the detection mechanism strives not to detect very small features as defects, those trapped defects were not picked up.

As designed, decreasing M_1 results in stronger detection, and hence higher DR and higher FPR, since more pixels (both defects and false positives) are picked up to be corrected. At the same M_1 value, and as designed, increasing M_2 provides more protection against false positives, and thus generally results in smaller DR and smaller FPR. As expected, decreasing M_3 has a small impact on the E_c and E_f values, since it only affects the pixels at which a decision to use the non-directional correction was made. The proposed solution results in less E_c and E_f values compared to the averaging-based and median-based corrections, due to its robust, detail-preserving correction strategy.

The predictable behavior associated with increasing/decreasing the parameters M_1, M_2 and M_3 is one important feature of the proposed solution–Linear interpolation of the parameters, to derive their values between the high and low ISO/exposure time settings, would yield consistent DPC performance and IQ, as well as fast turnaround time in an IQ tuning project.

The defect correction results for the proposed solution for two of the test images at (M_1, M_2, M_3) = (0.4, 10.0, 0.4) were depicted in figures (2) and (3). This parameter set was selected via IQ tuning so as to achieve the best tradeoff between defect correction and detail preservation for the sensor used in capturing the test images. From the results depicted in figures (2) and (3), it is worth noting how the proposed detection mechanism was capable of picking up the hot/cold singlets and the hot/cold/mixed couplets. In addition, the proposed correction mechanism replaced the picked-up defect pixels by a clean, detail-preserving estimate, resulting in higher IQ compared to the averaging-based and median-based corrections.

Figure (6) depicts example correction results for the proposed solution at M_3 = 0.4, and the averaging-based and median-based corrections for an image with real defects captured by the same 13MP sensor mentioned previously. The image was taken under 20 lux, unity analog gain and 690 msec exposure time, to increase the likelihood of defect visibility. The same algorithm design choices mentioned perviously were used in generating the results. The defect pixel locations were identified by the proposed detection mechanism with (M_1, M_2) = (0.4, 10.0). From the depicted results in figure (6), it is observed that the proposed solution picked up and corrected the mild hot defect located on the left side of the "+" sign in the image. In addition, the solution resulted in better detail preservation and better robustness to false positives, in comparison to the averaging-based and median-based corrections, which can be seen by examining the "C" and "O" letters.

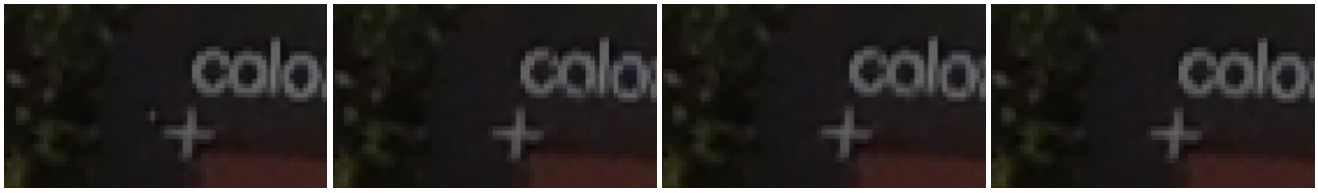


Figure 6. The result of different defect correction mechanisms for an image with real defects at (M_1, M_2) = (0.4, 10.0)
Left to right: No correction, averaging-based correction, median-based correction and proposed correction (M_3 = 0.4)

Conclusions

In this paper, a robust defect pixel detection and correction solution was presented. The algorithm enjoys several features, which recommend it for diverse, real-time imaging systems that exploit Bayer sensors in the camera pipeline. The detection mechanism robustly identifies singlets and couplets of hot pixels, cold pixels or mixture of both types, and results in high detection rates. The correction mechanism is detail-preserving and robust to false positives, and results in high IQ. Both mechanisms are computationally cheap and do not require any pre-calibration or pre-knowledge of the sensor defect positions or defect model parameters. The algorithm parameters are designed to have a predictable behavior. That makes them easy to tune, and guarantees consistent performance and IQ, and fast turnaround time in an IQ tuning project. Experimental results for images with simulated and real defects demonstrated the aforementioned merits of the solution, as well as its outperformance of conventional averaging-based and median-based corrections. The proposed solution will be improved to achieve close-to-perfect defect detection, especially in high-frequency textured regions, and to also achieve higher correction accuracy.

Acknowledgements

The author would like to thank Gokturk Cinserin for his help with the optimization of the proposed solution. Also thanks go to Gilad Michael for providing the tool used to generate and inject the simulated defects.

References

[1] R. Ramanath, W. E. Snyder, Y. Yoo and M. S. Drew, "Color image processing pipeline: a general survey of digital still cameras," *IEEE Signal Processing Magazine*, pp. 34-43, January 2005.

[2] G. H. Chapman, R. Thomas, I. Koren and Z. Koren, "Relating digital imager defect rates to pixel size, sensor area and ISO," IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, pp. 164-169, 2012.

[3] G. H. Chapman, J. Leung, R. Thomas, Z. Koren and I. Koren, "Tradeoffs in imager design with respect to pixel defect rates," IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, pp. 231-239, 2010.

[4] J. Leung, G. H. Chapman, Y. H. Choi, R. Thomas, Z. Koren and I. Koren, "Analyzing the impact of ISO on digital imager defects with an automated defect trace algorithm," Proc. SPIE 7536, Sensors, Cameras, and Systems for Industrial/Scientific Applications XI, 75360F, 2010.

[5] J. Leung, G. H. Chapman, I. Koren and Z. Koren, "Characterization of gain enhanced in-field defects in digital imagers," IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 155–163, 2009.

[6] J. Leung, J. Dudas, G.H. Chapman, I. Koren and Z. Koren, "Quantitative analysis of in-field defects in image sensor arrays," in IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 526 –534, 2007.

[7] J. Leung, G. H. Chapman, I. Koren and Z. Koren, "Automatic detection of in-field defect growth in image sensors," IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, pp. 305–313, 2008.

[8] G. H. Chapman, R. Thomas, I. Koren and Z. Koren, "Improved image accuracy in hot pixel degraded digital cameras," IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, pp. 172-177, 2013.

[9] E. Chang, "Kernel-size selection for defect pixel identification and correction," Proc. SPIE 6502, Digital Photography III, 65020J, 2007.

[10] S. Wang, S. Yao, O. Faurie, and Z. Shi, "Adaptive defect correction and noise suppression module in the CIS image processing system," in Proc. SPIE Int. Symposium on Photoelectronic Detection and Imaging, vol. 7384, p. 73842V, 2009.

[11] A. Tanbakuchi, A. van der Sijde, B. Dillen, A. Theuwissen and W. de Haan, "Adaptive pixel defect correction," in Proc. SPIE Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications IV, vol. 5017, pp. 360–370, 2003.

[12] M. Schoberl, J. Seiler, B. Kasper, S. Foessel and A. Kaup, "Sparsity-based defect pixel compensation for arbitrary camera raw images," IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1257-1260, 2011.

[13] B. E. Bayer, "Color imaging array," July 20, 1976, US Patent 3,971,065.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Third Edition, 2007.

Author Biography

Noha El-Yamany received her B.Sc. and M.Sc. degrees in Electrical Engineering from Alexandria University in Alexandria, Egypt, in 1999 and 2002, respectively, and her Ph.D. degree in Electrical Engineering from Southern Methodist University in Dallas, Texas, USA, in 2010. Since 2012 she has been working at Intel Corporation in Tampere, Finland. Her research interests include camera pipeline algorithms, computational photography and real-time imaging architecture.