electronics

Article

A Methodology to Analyze the Fault Tolerance of Demosaicking Methods against Memory Single Event Functional Interrupts (SEFIs)

Luis Alberto Aranda *, Alfonso Sánchez-Macián and Juan Antonio Maestro

ARIES Research Center, Universidad Antonio de Nebrija, 28040 Madrid, Spain; asanchep@nebrija.es (A.S.-M.); jmaestro@nebrija.es (J.A.M.)
* Correspondence: laranda@nebrija.es

Received: 24 August 2020; Accepted: 27 September 2020; Published: 2 October 2020

Abstract: Electronic circuits in harsh environments, such as space, are affected by soft errors produced by radiation. A single event functional interrupt (SEFI) can affect the behavior of a memory chip, with one or more rows, columns or even the whole device producing a wrong value when reading a set of stored bits. This problem may affect raw Bayer images stored in satellites and other spacecraft. In this paper, we present a methodology to analyze how different interpolation algorithms behave when they try to reconstruct the affected Bayer images into standard red, green and blue (RGB) images. This methodology can be used to compare and develop new fault-tolerant algorithms. The proposed methodology has been illustrated by studying a subset of interpolation algorithms. The results obtained from this example show that the interpolation algorithms that traditionally offer better results in a normal operation (in the absence of errors) are not always the best when SEFI errors are present in the Bayer images.

Keywords: color filter array (CFA) interpolation; demosaicking; fault tolerance; methodology; single event functional interrupt (SEFI)

1. Introduction

The use of electronic devices in satellites and other spacecraft is always challenging due to the harmful effects of space radiation [1]. In particular, memories are susceptible to soft errors that can cause the modification of the stored data [2]. Different studies (e.g., [3–5]) have concluded that single event functional interrupts (SEFIs) and single event upsets (SEUs) are the main consequences of this radiation in synchronous dynamic random access memories (SDRAMs). A single event upset consists of a single bit flip in one specific position of memory and can be prevented by single error correction (SEC) codes such as Hamming codes. SEFIs are produced when the control circuitry (e.g., decoding logic or control unit [6]) is struck by a heavy-ion that may produce the transition to an unknown state. The consequences may vary from reading corrupted data from a row to corrupting the whole device. According to the results from [3], SEFIs can be classified as:

- Row SEFIs: when part of a row, one row or several rows are affected.
- Column SEFIs: when part of a column, one or several columns are corrupted.
- Device SEFIs: the whole device is affected by the SEFI.

Three different patterns have been observed in space missions (e.g., Juno) when a SEFI occurs [7]:

- Single Row SEFI: only one row is affected.
- Band SEFI: several consecutive rows include corrupted data.

- Region SEFIs: more than one bank was affected.

SEFIs may be handled with complex configurations of error-correcting codes (e.g., [8–10]), but they may need the use of specific mitigation techniques based on software conditioning [6] (e.g., reloading the control registers or resetting the memory controller).

Images are a type of information that can be stored in memory for later access or as an intermediate step of image processing. Color filter array (CFA) demosaicking or interpolation is an important part of the image acquisition process in digital cameras. In this step, the raw pixel information captured by the charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS) image sensor is processed to provide the user a viewable image [11].

CFA demosaicking methods use an array of filters of different wavelength to capture the information necessary to create a color picture. A well-known filter array is the Bayer mosaic, which separates the wavelengths of the incident light into its red, green and blue (RGB) components following a specific pattern. Bayer patterns usually incorporate twice the number of green filters than red or blue filters to sample the green channel at a higher rate. This distribution of colors is chosen due to the fact that the human eye is more sensitive to green hue variations [12]. According to the mentioned distribution of color filters, the four most commonly used RGB Bayer patterns are illustrated in Figure 1.

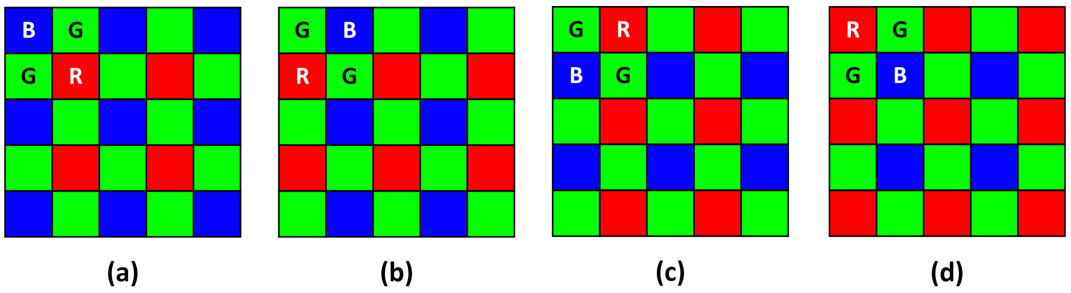

Figure 1. Bayer patterns: (**a**) BGGR, (**b**) GBRG, (**c**) GRBG, and (**d**) RGGB.

When the incident light passes through the previous patterns, it is filtered creating a raw Bayer image. A raw image is not ready to be displayed since each pixel contains the information of one of the three colors of the filter array. Consequently, the raw Bayer image has to be processed in order to obtain a full color image. This color reconstruction process is known as demosaicking or CFA interpolation and can be achieved by using several methods that will be described later in this paper.

In space applications, it is preferred to store raw Bayer images rather than full color images when the captured image is not immediately necessary (e.g., Earth observation). This approach is usually followed to save memory and to reduce the amount of information transmitted to Earth [13]. This is because the raw information of the Bayer image is tripled after the demosaicking process by creating a three-channel color image or, in other words, three matrices of the same size containing the information for the red, green, and blue color components.

Therefore, Bayer images stored in memory can be corrupted due to the radiation effects previously commented, affecting the quality of the demosaicked RGB color image. As an example, Figure 2 shows the result of performing CFA interpolation in a Bayer image affected by a band SEFI.

As explained before, SEFI errors may affect control registers, corrupting the information retrieved from one or several rows of memory (row and band SEFIs). This produces the corruption of a set of image pixels belonging to adjacent image rows (and stored in the corresponding rows of memory). Due to the diverse approaches used by different algorithms, corruption of image rows may have different effects on the final quality of the image. Thus, in those environments where SEFI errors are expected, the decision about the demosaicking algorithm to be used might change. This paper presents a methodology to compare these type of algorithms in the presence of SEFI errors. The methodology is applied to a subset of algorithms showing how the errors affect the peak signal-to-noise ratio (PSNR) and how algorithms degrade at a different rate. This methodology can be extrapolated to other algorithms. The conclusion of this study provides insight into the development of new fault-tolerant interpolation methods or real-time intelligent systems.

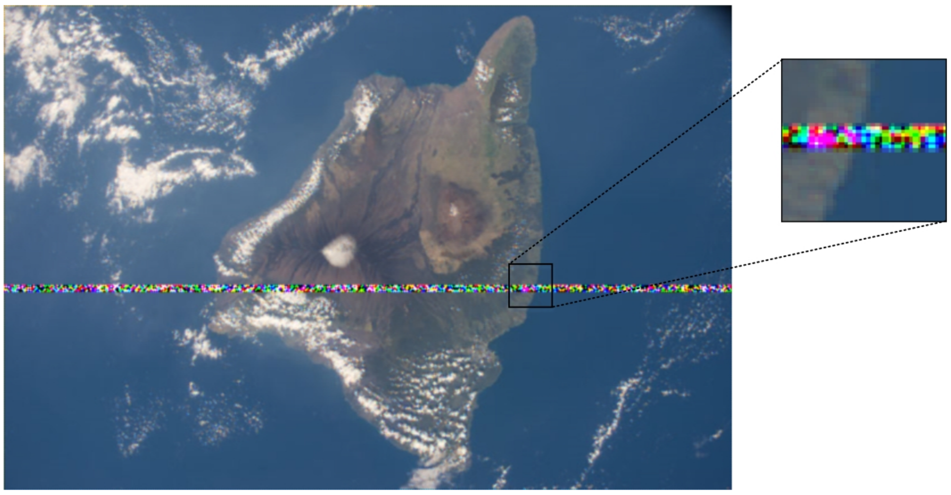

Figure 2. Detailed view of a corrupted demosaicked image when the stored Bayer image is affected by a band SEFI.

The rest of the paper is organized as follows. Section 2 introduces the motivation behind our work and other related work already performed in this area. In Section 3, the methodology is explained. The first part of the section describes the experimental set-up and, in the second part, the results obtained for the example subset of algorithms are presented and discussed. Finally, the conclusions are summarized in Section 4.

2. Motivation and Related Work

Simplicity, reduced latency, and limited power consumption are factors that have made static random access memories (SRAMs) widely used in space applications. However, looking at density, SDRAMs outperform them. Due to this [4], and their resistance to a high total ionizing dose [14], SDRAMs have also been considered in several missions. As an example, the European Space Agency (ESA) is proposing double data rate type three (DDR3) chips for the Jupiter Icy Moons Explorer mission [15]. However, experiments radiating DDR3 commercial off-the-shelf memory chips show that they are affected by SEUs, with an important sensitivity to SEFIs [6] so these memories may have malfunctions during a space mission. When these electronic devices are used to store Bayer images, this means that the quality of the reconstructed image may be degraded by the SEU or the SEFI. Therefore, fault-tolerant demosaicking algorithms have to be developed to deal with these radiation-induced errors present in space missions. Another alternative to fault-tolerant demosaicking algorithms can be the implementation of real-time intelligent systems that reconfigure the hardware or change the software implementation of the algorithm on-the-fly by a more robust algorithm to adapt it to the radiation dose received or the current position of the satellite in the orbit [16]. In order to facilitate the creation of these algorithms or systems, we have developed a methodology and an experimental set-up that will help with the decision process providing insight regarding the behavior of different demosaicking algorithms in the presence of SEFIs.

Over the years, several CFA demosaicking have been proposed [17–19] but, usually, the main focus of these works is on improving the quality of the reconstructed image, which is usually measured in terms of PSNR. There are also other works related to the design of new CFA patterns. These patterns are usually presented as an alternative to the mentioned Bayer patterns, and are carefully designed to minimize the reconstruction artifacts introduced by the demosaicking method applied [20,21]. In this case, the performance achieved by the algorithm is again measured in terms of PSNR. As an alternative to the PSNR quality metric, the authors in [22] propose a new methodology for evaluating demosaicking algorithms in industrial applications. They introduce two error measures based on edge accuracy. These measures, which are tailored to industrial applications, facilitate the interpolation algorithm decision process.

From the previous paragraph, it can be drawn that related works are focused on developing new patterns, algorithms, or methodologies that mainly improve the state of the art in terms of image quality. Some studies perform a quality evaluation of several demosaicking algorithms and others that assess the effect of Gaussian noise on the demosaicking process [23,24] but, to the authors' best knowledge, there are no methodologies to characterize both patterns and algorithms in the presence of soft errors aiming to develop fault-tolerant algorithms or intelligent systems.

To illustrate the methodology explained in the next section, we have selected eight well-known demosaicking methods: Neighbor, Bilinear, Smooth Hue, Gradient-based, Median-based, Hamilton–Adams, directional linear minimum mean square-error estimation (LMMSE), and a method based on color difference interpolation (CDI). These interpolation algorithms have been chosen to cover a range of algorithms from the simplest one (e.g., Neighbor), to more advanced algorithms (e.g., CDI-based), and even those used in commercial digital cameras (e.g., Gradient-based). There are, indeed, more advanced algorithms that can also be tested, but this subset was chosen since they are described exhaustively in the literature [25–28] and their source code is easily accessible. A brief explanation of the selected algorithms is presented below for the sake of completeness.

- Neighbor [25]: this method is based on the nearest-neighbor interpolation. The remaining RGB pixel values in the color demosaicked image are obtained by using the nearest neighbor's values in the raw Bayer image. This method creates blocky artifacts and it is only used when the application requires a high-speed implementation.
- Bilinear [25]: in this method, the remaining pixel values of the RGB demosaicked image are estimated by performing a convolution with a bilinear interpolation matrix. There are other linear interpolation methods such as bicubic or biquadratic, however bilinear is the most common.
- Smooth Hue [25]: in order to mitigate chrominance distortions created by the previous method, hue values are interpolated rather than chrominance values. Typically, the green channel is bilinearly interpolated and then used in the hue interpolation of the red and blue channels.
- Gradient-based [25]: the edges in the demosaicked RGB image obtained after applying the previously described methods are less defined. The gradient-based method adaptively interpolates the green channel by using the information provided by the vertical and horizontal gradients. If, for example, the horizontal gradient is smaller than the vertical gradient then the interpolation is performed only in the horizontal direction. The red and blue channels are interpolated normally by using a bilinear convolution since the human eye is more sensitive to green variations.
- Median-based [25]: this method calculates missing color values by processing the result of a Bilinear method. Median filtering is applied to differences between color planes (red-green, blue-green). Results are then combined with the original information to calculate the final pixel values.
- Hamilton–Adams [26]: this algorithm is divided into two steps. First, the green channel is interpolated by adding the smallest of the second-order gradients of the blue and red channels in the vertical and horizontal directions to the average of the green samples as correction terms. Then, the red and blue channels are similarly interpolated by using the second-order gradients of the green channel.
- LMMSE [27]: in this case, the missing green channel samples are adaptively estimated in both horizontal and vertical directions using the LMMSE estimation. These directional estimates are fused to obtain a full resolution green channel. Finally, the remaining two color channels are obtained by interpolating and then subtracting them from the green channel.

- CDI-based [28]: this algorithm is based on hard-coded heuristics for interpolation. The authors propose a scheme that exploits the correlation between different color channels and neighborhood pixels. The algorithm is a linear combination of low pass filter (LPF) and color difference interpolation (CDI).

Theseeight interpolation algorithms are used together with the traditional Bayer patterns shown in Figure 1 to illustrate the methodology described in the following section.

3. Fault Tolerance Methodology

In order to analyze the behavior of different demosaicking algorithms in the presence of SEFIs affecting the stored Bayer image, a fault-injection-based methodology has been proposed. This approach allows the user to characterize and compare algorithms in a relatively short amount of time by using a set of Bayer images and MATLAB scripts. The experimental methodology is described in detail in the subsection below together with the set-up required. In Section 3.2 some example results are presented. Finally, some remarks about the experimental methodology proposed in this paper are discussed in Section 3.3.

3.1. Experimental Set-Up

As mentioned before, the experimental set-up consisted of several synthetic Bayer images and MATLAB scripts obtained from [25]. In particular, 40 images were selected:

- 10 images belonging to the standard Kodak color image database [29] which include various scenes
- 10 space-related images of 640 × 480 pixels each
- 20 space-related images of 1920 × 1280 pixels each.

Every space-related image has been randomly selected from the NASA image and video library [30] avoiding the repetition of similar images to obtain a representative subset of space images (see Figure 3).

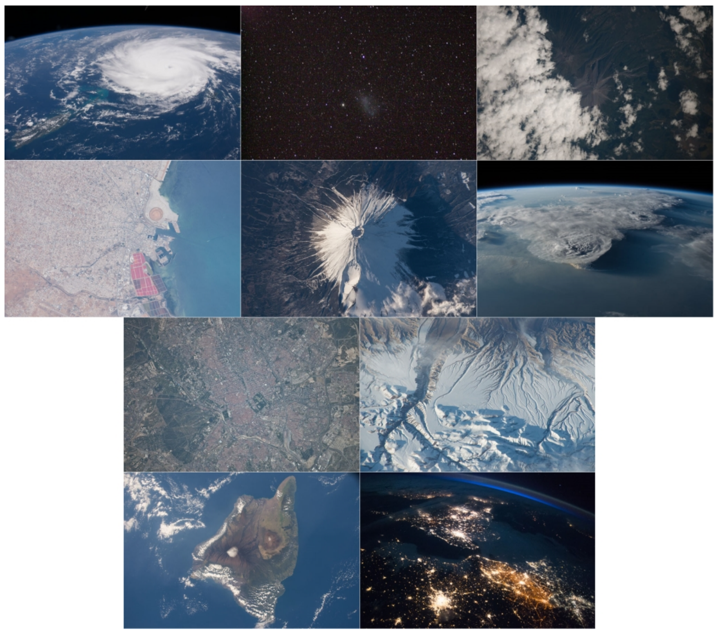

Figure 3. Sample subset of the space-related test images selected [30] (shown in red, green and blue (RGB) format).

Both Kodak and space image sets were sub-sampled to generate synthetic Bayer images with different filter patterns. In particular, four of the most widely used Bayer patterns (already shown in

Figure 1) were chosen for the experiments presented in the next subsection: BGGR, GBRG, GRBG, and RGGB [31]. However, any other Bayer pattern could be used.

Row and band SEFIs were generated by simulating the effect on an image of the corruption of different number of rows of a memory chip. To understand the effect of a row SEFI on a stored image, a DDR3 chip configuration similar to the one included in the ZC702 Xilinx FPGA board [32] memory chip (MT41J256M8HX) was considered. This chip had eight banks, each one including eight segments with 1024 columns and 4096 rows. Therefore, in each row there were 1024 bytes, which would be equivalent to 1024 color elements (raw pixels). With images of 640 pixels per row, an approximation could be made to consider that a row of a memory chip affected by a SEFI corresponded to 1.5 rows of a raw image. For images of 1920 pixels per row, one SEFI-affected memory chip row corresponded approximately to half a row of an image.

In the experiments presented in the next subsection, 480 SEFIs were simulated by corrupting multiples of half a row of each image by injecting random bits in each pixel of the randomly selected rows. The interpolation algorithms were tested and then the error was corrected before continuing with the campaign. These steps of the methodology are illustrated in Figure 4 for a better understanding.

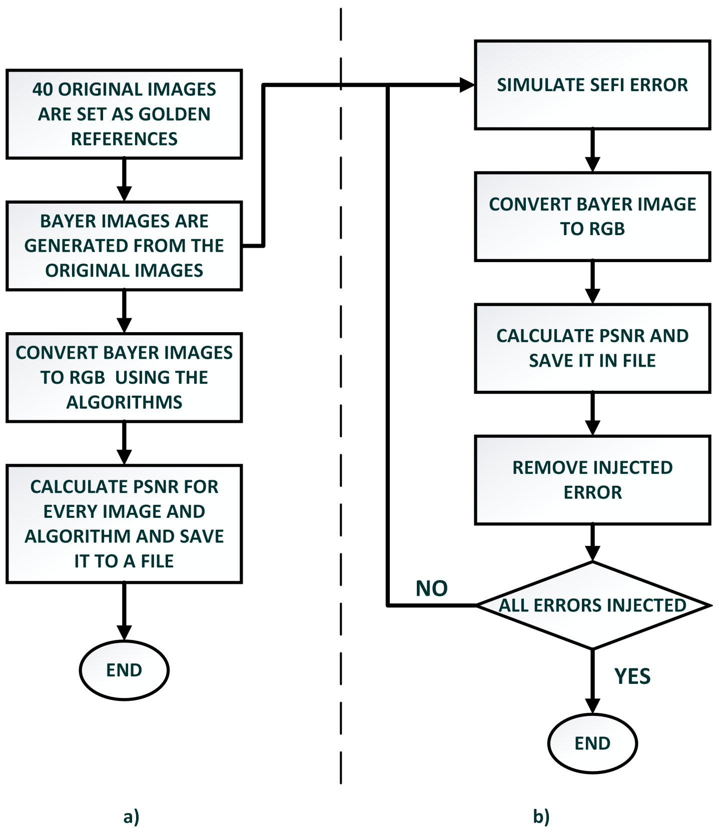
40 ORIGINAL IMAGES ARE SET AS GOLDEN REFERENCES
BAYER IMAGES ARE GENERATED FROM THE ORIGINAL IMAGES
CONVERT BAYER IMAGES TO RGB USING THE ALGORITHMS
CALCULATE PSNR FOR EVERY IMAGE AND ALGORITHM AND SAVE IT TO A FILE
END
SIMULATE SEFI ERROR
CONVERT BAYER IMAGE TO RGB
CALCULATE PSNR AND SAVE IT IN FILE
REMOVE INJECTED ERROR
ALL ERRORS INJECTED
NO
YES
END
a)
b)

Figure 4. Fault injection methodology flow chart. (**a**) Without errors. (**b**) With errors.

First, each of the 40 original images were set as “golden” references. They were used to compare the performance of the interpolation algorithms by calculating the difference in terms of PSNR between the interpolated image and the “golden” image. Then, two experiments were be performed. The first one determined the behavior of each interpolation algorithm in the absence of errors, i.e., it measured the actual quality of the algorithms compared to the “golden” image, in a standard error-free scenario (see Section 3.2.1). The second experiment analyzed the behavior of each algorithm in the presence of errors (see Section 3.2.2). With these experiments, algorithms that behaved well under standard circumstances but that offered worse results in a specific environment with errors could be detected. The steps of the second experiment of the methodology could be performed in a loop as follows.

1. A SEFI was injected by simulating the corruption of random rows (a multiple of half a row) of the original Bayer image.
2. The corrupted image was converted to RGB by using the interpolation algorithms under test (in our case the eight interpolation algorithms explained in Section 2).
3. These resulting corrupted RGB images were compared to the initial golden RGB image and the PSNR is calculated.
4. The injected error was then removed. Notice that, to make the process easier, the corruption rows could be performed by accumulation, i.e., keeping the error and adding an additional corrupted row (or half a row) on top of it. The error was only removed when the maximum selected number of error rows was reached (e.g., six rows in the experiments).
5. The loop started again until the specified number of errors was injected.

These fault injection campaigns were repeated for each Bayer image until all the PSNR values were obtained. As a practical example of the methodology, the results obtained for the eight interpolation algorithms explained in Section 2 are presented next.

3.2. *Experimental Results*

3.2.1. Quality of the Algorithms in the Absence of Errors

The averaged PSNR values in decibels (dB) of the 40 interpolated images in the absence of errors when compared to the corresponding "golden" image are presented in Figure 5.

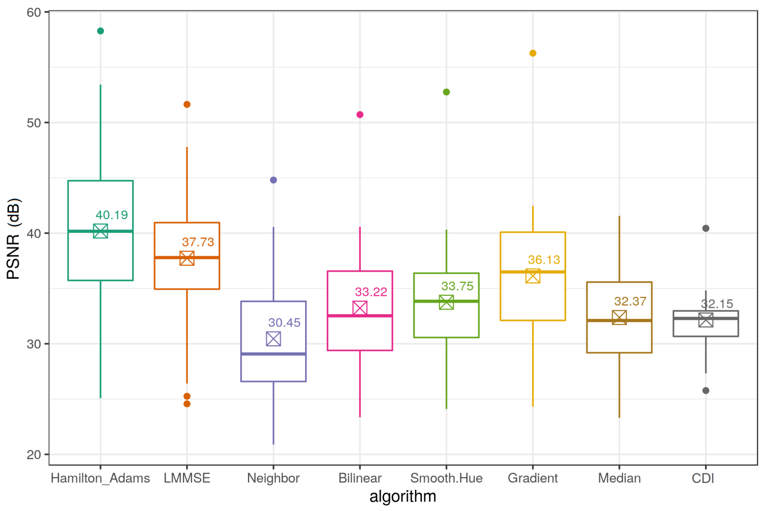

Figure 5. Averaged peak signal-to-noise ratio (PSNR) for each algorithm in the absence of errors.

These results showed that the Hamilton–Adams algorithm was the best one in terms of PSNR, followed by the LMMSE, Gradient, Smooth Hue, Bilinear, Median-based, and CDI-based. The worse PSNR was obtained with the Neighbor algorithm. The Hamilton–Adams and the LMMSE algorithms outperformed the rest of the algorithms because they were developed implementing more advanced strategies to deal with demosaicking artifacts such as noise-caused color artifacts in the demosaicking process [27]. This implies that, in a standard scenario without errors (or where errors were negligible), the Hamilton–Adams algorithm would be the best choice if the final image quality was to be maximized. However, as will be discussed in the following subsections, the best algorithm in the absence of errors may not be the best option in a harsh environment where errors are likely events.

3.2.2. Quality of the Algorithms in the Presence of Errors

The results of the fault injection campaign are presented in Figure 6. This figure shows the behavior of the PSNR in dB for each algorithm as a function of the number of image rows affected by a SEFI.

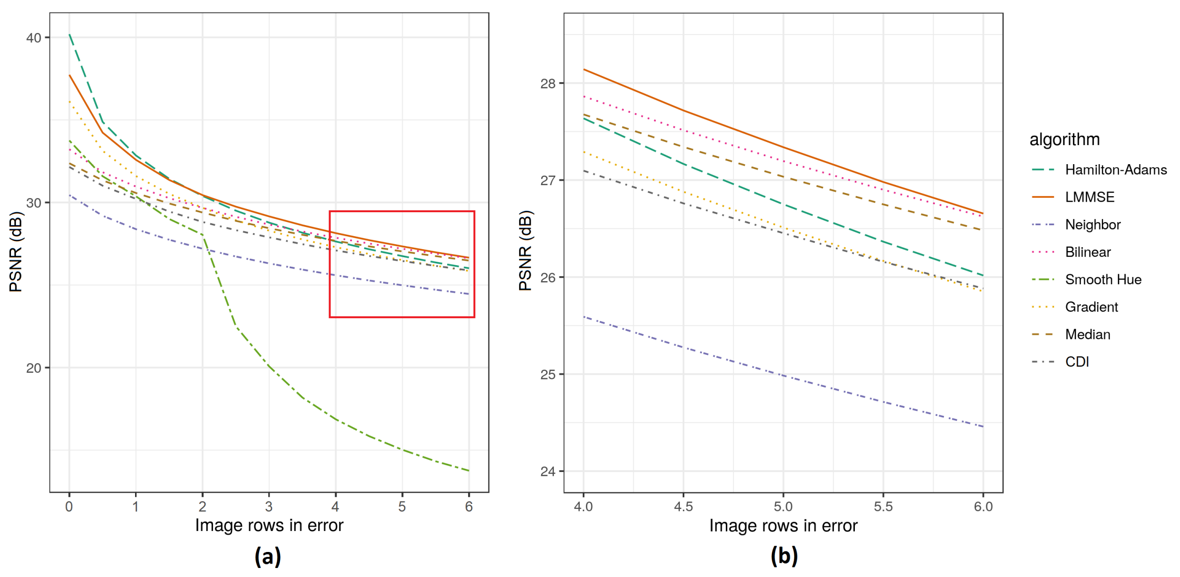

Figure 6. (**a**) Full view and (**b**) detailed view of the error behavior as a function of the number of image rows affected by a single event functional interrupt (SEFI).

The first comment on these results is that the Smooth Hue algorithm was the most affected by SEFIs, in particular when two or more adjacent image rows were corrupted, with an important negative slope. With just one image row affected it already behaved worse than the Neighbor and Median-based algorithms in terms of PSNR. This phenomenon can be explained due to the green channel dependency of the algorithm. As mentioned in the Introduction, standard Bayer patterns contained twice the number of green pixels than red or blue pixels (see Figure 1). Therefore, the green channel values were more affected by SEFIs, so an algorithm that used the green values to interpolate the red and blue channels would obtain poorer outcomes in the presence of SEFIs than an algorithm that independently interpolated each of the three color channels. Something similar happened to the Hamilton–Adams algorithm. It started being the best one when two or fewer image rows were affected, but LMMSE, Bilinear and Median-based algorithms outperformed it when more than four image rows were corrupted. If the system was configured so it could detect a band SEFI, an adaptive solution could be implemented that made use of the Hamilton–Adams algorithm or the LMMSE algorithm depending on the number of rows affected.

In order to have more information to select the best algorithm or algorithms for the desired application, the average execution time and/or the resulting demosaicked images could be obtained to complement the previous quality analysis. As an example, the average execution time for the subset of demosaicking algorithms chosen is summarized in Table 1.

Table 1. Relative execution time for the eight demosaicking algorithms (average).

Hamilton-Adams	LMMSE	Neighbor	Bilinear	Smooth Hue	Gradient	Median	CDI
2.04x	154.75x	1.13x	1.00x	1.15x	3.21x	1.93x	127.38x

It can be observed in the table that the Bilinear was the fastest algorithm of the subset followed by the Neighbor, Smooth Hue, Median, Hamilton-Adams, Gradient, CDI, and LMMSE, which was the algorithm that required more time to generate the RGB demosaicked image.

As mentioned before, a visual comparison of the eight demosaicking algorithms may also be used to provide more insight into the decision process. An example of this is presented in Figure 7.

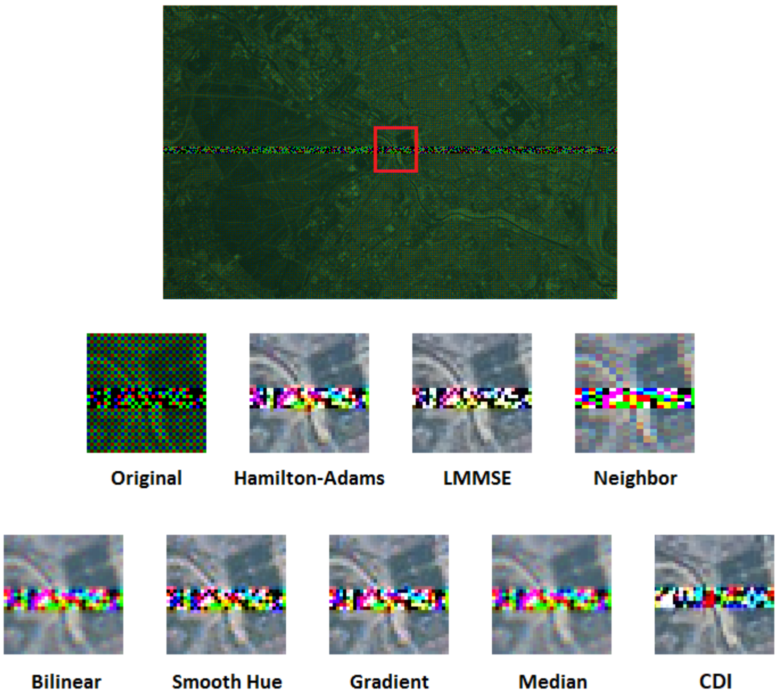

Figure 7. Visual comparison example of the interpolation algorithms in the presence of SEFIs.

Figure 7 illustrates the behavior of each interpolation algorithm when the original Bayer image has six rows corrupted due to a SEFI. Several conclusions can be drawn from this comparison. The first fact that can be noticed is that none of the algorithms were able to manage the color band induced by the SEFI in a proper way. However, the final image was still viewable since all the demosaicking methods studied performed local operations in the Bayer image. In addition to this, it can be seen that some algorithms such as the Hamilton–Adams, the Smooth Hue or the Gradient, propagated the effect of the SEFI color band to adjacent rows of the image, thus degrading the quality of the final image even more. Conversely, this effect was more constrained in the LMMSE and the Neighbor algorithms. The former had a higher PSNR than the latter because the Neighbor algorithm created block artifacts not only in the SEFI horizontal band but also in the rest of the image even when no SEFI was present. Lastly, it can be observed that the Median and the Bilinear algorithms smoothed the upper and the lower rows of the color band. This behavior could explain the higher PSNR of these methods.

This qualitative analysis can be further extended with a quantitative analysis to support the discussion written in the previous paragraph. In order to do that, the methodology flow chart presented in Figure 4 can still be followed but modifying the way of computing the PSNR value of each image. If the pixels directly affected by the fault injection were excluded from the PSNR computation, then, the propagation effects of the SEFI color band to the adjacent rows can be measured. The results from this experiment are summarized in Table 2.

Table 2. Averaged PSNR values (in dB) in the absence of errors vs excluding the pixels directly affected by the SEFI. Degradation values show the difference between these two scenarios for each algorithm.

	H-A	LMMSE	Neighbor	Bilinear	Smooth Hue	Gradient	Median	CDI
In the absence of errors	40.19	37.73	30.45	33.22	33.75	36.13	32.37	32.15
Excluding affected rows	37.44	36.78	29.63	32.45	32.38	34.75	31.85	31.35
Degradation	2.75	0.95	0.82	0.77	1.37	1.38	0.52	0.80

It can be observed in the table that the Hamilton–Adams (named H-A in the table), the Gradient, and the Smooth Hue algorithms had (on average) greater degradation values compared to the rest of the algorithms. On the other hand, the Median algorithm was the best in this aspect due to its inherent median-filtering properties. The values obtained from this experiment, together with the visual comparison, support the statement from the previous discussion. Therefore, both visual and numerical experiments can be used in the methodology to facilitate the analysis of the SEFI effects in adjacent rows.

3.3. About the Experimental Methodology

It is worth mentioning that the presented experimental methodology can be extended by performing additional experiments such as testing other SEFI patterns (e.g., columns) or analyzing the PSNR values in an incremental way instead of the absolute values of each algorithm. In this manner, more information about the intrinsic behavior of the algorithms can be obtained. Concerning the algorithms, it should also be remarked that the methodology can be applied to any interpolation algorithm. In this paper, a few demosaicking algorithms have been characterized to highlight the fact that better-quality algorithms may behave worse in the presence of errors. However, more advanced state-of-the-art interpolation algorithms can also be compared following the same methodology.

For the actual error injection, SEFIs have been tested since these errors are harmful for the image quality. In this paper, two simplifications have been performed in order to make the experimental method more focused:

- SEUs or any other soft error effects have not been simulated. Although SEUs are quite common errors, and therefore they represent the standard failure model when testing many space applications, they are less representative in image demosaicking. This is because a regular number of SEUs will not heavily modify the PSNR of the final images, being negligible in most cases.
- Error accumulation has not been considered. Once an error has been injected and tested, it is removed before continuing with the fault injection campaign. The isolated error model is the most usual approach in literature, since soft errors are usually scarce events, and therefore it is not common for them to accumulate.

Therefore, the experimental methodology followed in this paper may have some shortcomings in situations where errors arrive at a much higher rate than usual, with majority presence of other kinds of soft errors such as SEUs. Nevertheless, as explained in this subsection, this should be an infrequent case.

4. Conclusions

In this paper, a methodology to analyze the behavior of demosaicking algorithms under the effects of SEFI errors produced by radiation is presented. The methodology can be implemented in any computer since it does not require additional equipment and provides enough insight to draw several conclusions from the algorithms under test. As a practical example of the methodology, we have put into perspective eight well-known interpolation algorithms. Particular conclusions such as that the Smooth Hue algorithms suffers an important quality decrement after two rows in error, becoming the worst algorithm in PSNR by a large margin can be drawn. As well as more general conclusions such as that better-quality algorithms (e.g., Hamilton–Adams) keep their leadership in terms of PSNR in the absence of errors or when the SEFI affects two or fewer rows of an image, but other classic algorithms (e.g., Bilinear) produce better results in terms of PSNR when four or more image rows are corrupted. These results, for example, could be used to create an adaptive system to decide which algorithm to apply depending on the presence of a SEFI in the memory chip.

Author Contributions: Conceptualization, L.A.A. and J.A.M.; Data curation, L.A.A. and A.S.-M.; Formal analysis, A.S.-M. and J.A.M.; Methodology, A.S.-M. and J.A.M.; Software, L.A.A. and A.S.-M.; Supervision, J.A.M.; Writing—original draft, L.A.A.; Writing—review & editing, A.S.-M. and J.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Siegle, F.; Vladimirova, T.; Ilstad, J.; Emam, O. Mitigation of radiation effects in SRAM-based FPGAs for space applications. *ACM Comput. Surv. (CSUR)* **2015**, *47*, 37–72. [CrossRef]
2. Argyrides, C.; Reviriego, P.; Pradhan, D.K.; Maestro, J.A. Matrix-based codes for adjacent error correction. *IEEE Trans. Nucl. Sci.* **2010**, *57*, 2106–2111. [CrossRef]
3. Bak, G.; Lee, S.; Lee, H.; Park, K.; Baeg, S.; Wen, S.; Wong, R.; Slayman, C. Logic soft error study with 800-MHz DDR3 SDRAMs in 3x nm using proton and neutron beams. In Proceedings of the 2015 IEEE International Reliability Physics Symposium, Monterey, CA, USA, 19–23 April 2015; pp. SE.3.1–SE.3.5.
4. Guertin, S.M.; Partterson, J.D.; Nguyen, D.N. Dynamic SDRAM SEFI detection and recovery test results. In Proceedings of the 2004 IEEE Radiation Effects Data Workshop, Atlanta, GA, USA, 22 July 2004; pp. 62–67.
5. Bougerol, A.; Miller, F.; Guibbaud, N.; Gaillard, R.; Moliere, F.; Buard, N. Use of laser to explain heavy ion induced SEFIs in SDRAMs. *IEEE Trans. Nucl. Sci.* **2010**, *57*, 272–278. [CrossRef]
6. Herrmann, M. Radiation effects in DRAMs. In *Ionizing Radiation Effects in Electronics*; Bagatin, M., Gerardin, S., Eds.; CRC: Boca Raton, FL, USA, 2016; pp. 77–95, ISBN 9781498722605.
7. Guertin, S.M.; Allen, G.R.; Sheldon, D.J. Programmatic impact of SDRAM SEFI. In Proceedings of the 2012 IEEE Radiation Effects Data Workshop, Tucson, AZ, USA, 16–20 July 2012; pp. 1–8.
8. Longden, L.; Thibodeau, C.; Hillman, R.; Layton, P.; Dowd, M. *Designing a Single Board Computer for Space Using the Most Advanced Processor and Mitigation Technologies*; White Paper-Case Study; Maxwell Technol. Inc.: San Diego, CA, USA, 2002.
9. Pontarelli, S.; Cardarilli, G.C.; Re, M.; Salsano, A. Error correction codes for SEU and SEFI tolerant memory systems. In Proceedings of the 2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Chicago, IL, USA, 7–9 October 2009; pp. 425–430.
10. Sánchez-Macián, A.; Reviriego, P.; Maestro, J.A. Combined SEU and SEFI protection for memories using orthogonal latin square codes. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2016**, *63*, 1933–1943. [CrossRef]
11. Gunturk, B.K.; Glotzbach, J.; Altunbasak, Y.; Schafer, R.W.; Mersereau, R.M. Demosaicking: Color filter array interpolation. *IEEE Signal Process. Mag.* **2005**, *22*, 44–54. [CrossRef]
12. Ramanath, R.; Snyder, W.E.; Bilbro, G.L.; Sander, W.A. Demosaicking methods for Bayer color arrays. *J. Electron. Imaging* **2002**, *11*, 306–316. [CrossRef]
13. Triana, J.S.; Bautista, S.; González, F.A.D. Identification of design considerations for small satellite remote sensing systems in low earth orbit. *J. Aerosp. Technol. Manag.* **2015**, *7*, 121–134. [CrossRef]
14. Herrmann, M.; Grurmann, K.; Gliem, F.; Schmidt, H.; Ferlet-Cavrois, V. In-situ TID test of 4-Gbit DDR3 SDRAM devices. In Proceedings of the 2013 IEEE Radiation Effects Data Workshop, San Francisco, CA, USA, 8–12 July 2013; pp. 1–7.
15. Schmidt, H.; Hermann, M.; Grürmann, K.; Gliem, F.; Ferlet-Cavrois, V. Radiation hard memory. Radiation testing of candidate memory devices for Laplace mission. In Proceedings of the 2015 CNES/ESA Radiation Effects Final Presentation Days, Toulouse, France, 6–9 March 2015; pp. 1–112.
16. Sabogal, S.; George, A.; Wilson, C. Reconfigurable Framework for Environmentally Adaptive Resilience in Hybrid Space Systems. *ACM Trans. Reconfigurable Technol. Syst. (TRETS)* **2020**, *13*, 1–32. [CrossRef]
17. Chen, X.; He, L.; Jeon, G.; Jeong, J. Multidirectional weighted interpolation and refinement method for Bayer pattern CFA demosaicking. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1271–1282. [CrossRef]
18. Wang, Y.Q. A multilayer neural network for image demosaicking. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1852–1856.
19. Heinze, T.; von Lowis, M.; Polze, A. Joint multi-frame demosaicing and super-resolution with artificial neural networks. In Proceedings of the IEEE 19th International Conference on Systems, Signals and Image Processing (IWSSIP), Vienna, Austria, 11–13 April 2012; pp. 540–543.

20. Henz, B.; Gastal, E.S.; Oliveira, M.M. Deep joint design of color filter arrays and demosaicing. *Comput. Graph. Forum* **2018**, *37*, 389–399. [CrossRef]
21. Wang, J.; Wu, J.; Wu, Z.; Jeon, G. Filter-based Bayer pattern CFA demosaicking. *Circuits Syst. Signal Process.* **2017**, *36*, 2917–2940. [CrossRef]
22. Breier, M.; Haas, C.; Li, W.; Merhof, D. Color filter arrays revisited-evaluation of Bayer pattern interpolation for industrial applications. In Proceedings of the IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, France, 19–21 July 2016; pp. 52–57.
23. Hernández, D.C.M.; González, F.A.D.; Correa, J.S.T.; Cabrera, P.R.P. Quality evaluation of chromatic interpolation algorithms for image acquisition system. *J. Aerosp. Technol. Manag.* **2016**, *8*, 339–351. [CrossRef]
24. Tan, H.; Zeng, X.; Lai, S.; Liu, Y.; Zhang, M. Joint demosaicing and denoising of noisy Bayer images with ADMM. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2951–2955.
25. Swirski, L. CFA interpolation detection. In *Topics in Security: Forensic Signal Analysis*; University of Cambridge: Cambridge, UK, 2009.
26. Hamilton, J.F., Jr.; Adams, J.E. Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera. U.S. Patent 5,629,734, 29 July 1997.
27. Zhang, L.; Lukac, R.; Wu, X.; Zhang, D. PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras. *IEEE Trans. Image Process.* **2009**, *18*, 797–812. [CrossRef] [PubMed]
28. Jaiswal, S.P.; Au, O.C.; Jakhetiya, V.; Yuan, Y.; Yang, H. Exploitation of inter-color correlation for color image demosaicking. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1812–1816.
29. Franzen, R. *Kodak Lossless True Color Image Suite*. 2013. Available online: http://r0k.us/graphics/kodak/ (accessed on 15 February 2019).
30. Grubbs, R. *NASA Image and Video Library*. 2018. Available online: https://images.nasa.gov/ (accessed on 22 May 2019).
31. Trifan, A.; Neves, A.J. A survey on lossless compression of Bayer color filter array images. *Int. J. Comput. Electr. Autom. Control Inform. Eng* **2016**, *10*, 729–734.
32. Xilinx, Inc. *ZC702 Evaluation Board for the Zynq-7000 XC7Z020 All Programmable SoC, User Guide, UG850 (v1.6.1)*. 2018. Available online: https://www.xilinx.com/support/documentation/boards_and_kits/zc702_zvik/ug850-zc702-eval-bd.pdf (accessed on 9 February 2019).