FURTHER IMPROVEMENTS OF CFA 3.0 BY COMBINING INPAINTING AND PANSHARPENING TECHNIQUES

Chiman Kwan and Jude Larkin

Applied Research, LLC, Rockville, Maryland, USA

ABSTRACT

Color Filter Array (CFA) has been widely used in digital cameras. There are many variants of CFAs in the literature. Recently, a new CFA known as CFA 3.0 was proposed by us and has been shown to yield reasonable performance as compared to some standard ones. In this paper, we investigate the use of inpainting algorithms to further improve the demosaicing performance of CFA 3.0. Six conventional and deep learning based inpainting algorithms were compared. Extensive experiments demonstrated that one algorithm improved over other approaches.

KEYWORDS

CFA 3.0, color filter array, demosaicing, inpainting, deep learning, pansharpening

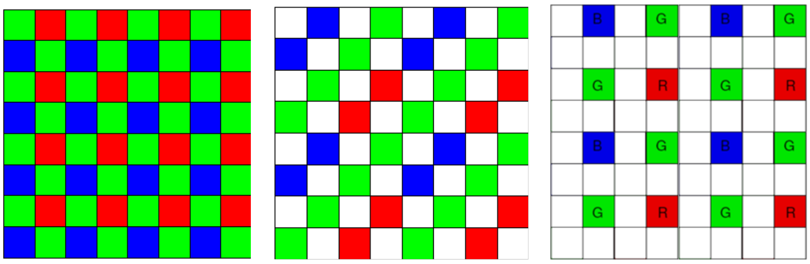
1. INTRODUCTION

Bayer pattern [1] was invented in the early 1980's and is still a very popular color filter array (CFA) for digital cameras. The Bayer pattern as shown in Figure 1(a) is also known as CFA 1.0 in the literature. Even for planetary explorations, NASA has adopted the Bayer pattern in the Mastcam imagers onboard the Mars rover Curiosity [2]-[5].

Aiming to improve the Bayer pattern in low lighting conditions, Kodak researchers [6,7] invented a red-green-blue-white (RGBW) CFA pattern, which is also known as CFA 2.0, which is shown in Figure 1(b). Half of the pixels in CFA 2.0 are white and the remaining pixels share the R, G, and B colors. Due to the presence of white pixels, the camera sensitivity is increased and hence the performance of CFA 2.0 in low lighting conditions should be better than CFA 1.0. Extensive experiments in [8] showed that CFA 2.0 is in indeed better than CFA 1.0 in low lighting conditions, where Poisson noise is dominant. Some additional studies were also carried out for CFA 2.0 [9].

In a recent paper by us [12], a new CFA pattern known as CFA 3.0 was proposed. In CFA 3.0 as shown in Figure 1(c), even more white pixels are introduced, hoping that the demosaicing performance will be further improved in low lighting conditions. Unfortunately, having more white pixels means that fewer color pixels will be present in the color filter array. Consequently, the overall performance of CFA 3.0 for low lighting images is slightly inferior to CFA 2.0 but still better than CFA 1.0 [10][11].

DOI: 10.5121/sipij.2020.11601



(a) (b) (c)

Figure 1. Three CFA patterns. (a) CFA 1.0; (b) CFA 2.0; (c) CFA 3.0.

In [12], we used an interpolation method known as local directional interpolation and nonlocal adaptive thresholding (LDI-NAT) [13] to create the luminance or panchromatic (pan) band. After that, the full resolution luminance band is then fused with the low resolution color image via pansharpening techniques to create the full resolution color image. The whole process is summarized in Figure 2. The luminance image is also termed the panchromatic image and we use them interchangeably in this paper.

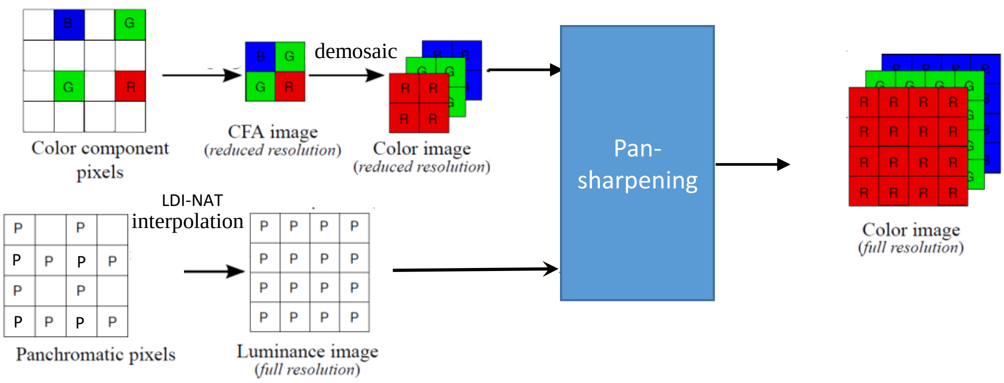


Figure 2. A pansharpening approach for CFA 3.0.

From Figure 2, it is natural to ask several research questions. First, are there any methods that can further enhance the performance of the luminance image? As we will see in the experiments, the demosaicing performance can be improved quite a lot if the ground truth panchromatic image is used. This means that if one can apply a high performing interpolation method to fill in the missing pixels in the panchromatic band, then the overall demosaicing performance will be increased. Second, if there does exist a good interpolation/inpainting algorithm, how much performance gain can we achieve?

In this paper, we will focus on answering the two aforementioned questions. In particular, we propose to investigate various inpainting methods to create the panchromatic band. In addition to the LDI-NAT method, we also applied five other methods, including conventional and deep learning algorithms. After the inpainting is done, we then apply various pansharpening algorithms to generate the final demosaiced images. We extensively evaluate the different combinations using the Kodak benchmark images.

There are three major contributions in this paper:

- We are the first ones to apply various inpainting methods to generate the pan band for demosaicing CFA 3.0.
- We are also the first team to investigate the combination of inpainting and pansharpening in demosaicing CFA 3.0.
- The combination of inpainting and pansharpening results are better than before, but there is still room for further improvement.

The rest of our paper is organized as follows. Section 2 summarizes the methods, data, and performance metrics. Section 3 presents all the experimental results. Finally, some concluding remarks and future directions will be given.

2. METHODS, DATA, AND PERFORMANCE METRICS

2.1. Architecture of Demosaicing CFA 3.0 with Inpainting and Pansharpening

In our earlier paper [12], we presented a standard approach to demosaicing CFA 3.0. For completeness, that architecture is depicted in Figure 3. The R, G, and B pixels in the CFA 3.0 are extracted to form a reduced resolution CFA image. A demosaicing algorithm (LDI-NAT) is used to demosaic it and generate a reduced resolution color image. Parallel to this activity, the white/panchromatic pixels in the CFA 3.0 are also interpolated to form the luminance image using the same LDI-NAT algorithm. The luminance image is then downsampled by two times via averaging and the reduced resolution luminance image is subtracted from the reduced resolution color image to generate the chrominance-luminance image. A simple upsampling via bicubic interpolation is then performed to generate the full resolution chrominance-luminance image. Finally, the luminance image and the chrominance-luminance image are added together to form the final demosaicing image. This simple architecture is very simple to understand and implement. Although there are many algorithms in the literature that could be used in the interpolation and demosaicing steps, we chose LDI-NAT in [12] simply because it has reasonable performance.

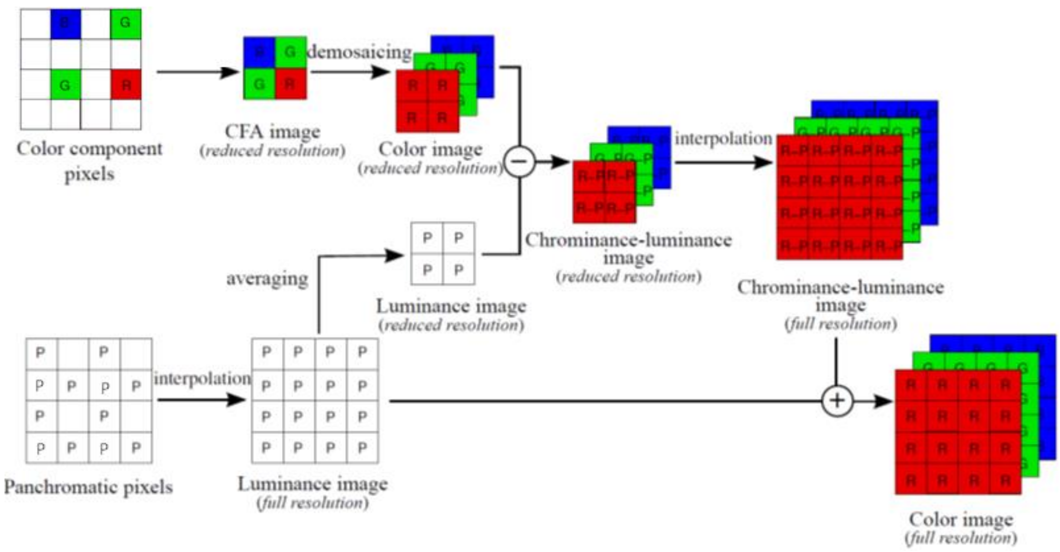


Figure 3. Standard demosaicing framework for CFA 3.0.

In this paper, we propose the architecture shown in Figure 4, which is essentially the same as the architecture shown in Figure 2 except the interpolation step. In Figure 4, we emphasize on the use of inpainting algorithms for generating the luminance image and the other parts are exactly the same as Figure 2. The difference between interpolation and inpainting is very subtle. Normally, interpolation is used to fill in missing pixels in images that have regular missing patterns. On the other hand, inpainting is referring to missing pixels with free form patterns. That is, the missing patterns can have arbitrary shapes. As can be seen from Figure 4, the pansharpening step is to utilize the high resolution luminance image to sharpen the reduced resolution color image and the final pansharpened image will be the demosaiced image.

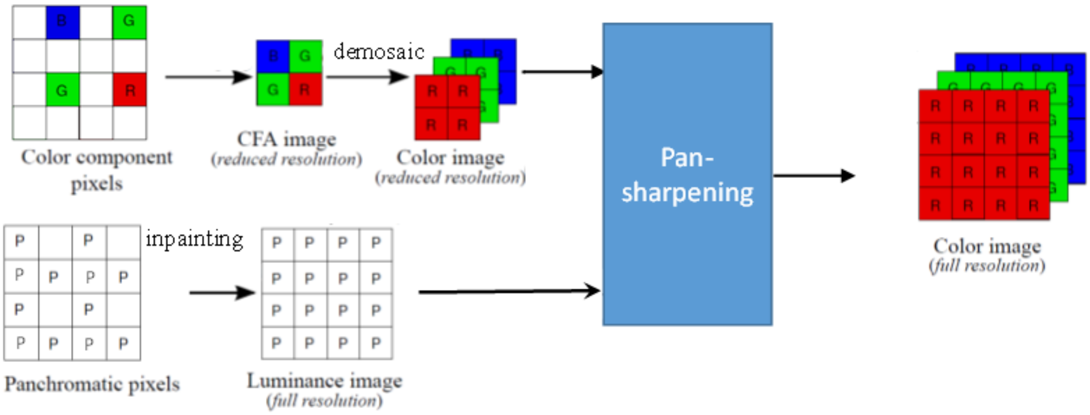


Figure 4. Proposed architecture for the combined inpainting and pansharpening demosaicing approach.

2.2. Inpainting Methods

In recent years, there are many new developments in inpainting. In this paper, we have evaluated the following six techniques:

- Linear Directional Interpolation and Nonlocal Adaptive Thresholding (LDI-NAT): This algorithm is a demosaicing algorithm. However, it can be used for both demosaicing as well as interpolation [13]. It has good performance in our earlier studies [8]. We used LDI-NAT in our earlier paper [12] and this will be the baseline for our inpainting investigations.
- Laplacian: This method [14] fills in each missing pixel using the Laplacian interpolation formula by finding the mean of the surrounding known values.
- Bilinear: This method simply uses the average of neighboring pixels to fill in the missing pixels. Bilinear and Laplacian have similar performance.
- Inpaint_nans: We denote this as "inpaint" in our later experiments. This method was developed by D'Errico[15]. This is a very simple method that only uses the neighboring pixels to estimate the missing pixels which will be referred as NaNs (not a number).
- FOE: The Field of Experts method (FOE) was developed by Roth [16]. This method uses pre-trained models that are used to filter out noise and obstructions in images.
- Generative Inpainting (GenIn)[17]: A new inpainting method, Generative Inpainting (GenIn), which is a deep learning-based method [17], was considered in our research. It was developed at the University of Illinois that aims to outperform typical deep learning methods that use convolutional neural network (CNN) models. GenIn builds on CNN and Generative Adversarial Networks (GAN) in an effort to encourage cohesion between created and existing pixels. GenIn ranked the first in one Github page (https://github.com/1900zyh/Awesome-Image-Inpainting), which contains many conventional and deep learning based algorithms. This is the reason we chose GenIn in this paper.

2.3. Pansharpening Methods

In the paper [18] written by us, we proposed a pansharpening approach to demosaicing CFA 2.0. This approach is illustrated in Figure 4. The missing pixels in the panchromatic band are interpolated. At the same time, the reduced resolution CFA is demosaiced. We then apply pansharpening to generate the full resolution color image. There are many pansharpening algorithms that can be used. Principal Component Analysis (PCA) [19], Smoothing Filter-based Intensity Modulation (SFIM) [20], Modulation Transfer Function Generalized Laplacian Pyramid (GLP) [21], MTF-GLP with High Pass Modulation (HPM) [22], Gram Schmidt (GS) [23], GS Adaptive (GSA) [24], Guided Filter PCA (GFPCA) [25], PRACS [26] and hybrid color mapping (HCM) [27]-[31] have been used in our experiments. The list is a representative, if not exhaustive, set of competitive pansharpening algorithms.

2.4. Data

We downloaded a benchmark data set (Kodak) from a website (http://r0k.us/graphics/kodak/) and selected 12 images, which are shown in Figure 5. It should be noted that this dataset is well-known and has been used by many authors in the demosaicing community such as [32]-[36]. These clean images will be used as reference images for objective performance metrics generation. Moreover, they will be used for generating noisy images that emulate low lighting conditions.

Figure 5. Twelve clean images from the Kodak dataset.

2.5. Metrics

Five performance metrics were used in our experiments to compare the different methods and CFAs. These metrics are well-known in the literature.

- Peak Signal-to-Noise Ratio (PSNR) [37]

Separate PSNRs in dBs are computed for each band. A combined PSNR is the average of the PSNRs of the individual bands. Higher PSNR values imply higher image quality.

- Structural SIMilarity (SSIM)
  In [38], SSIM was defined to measure the closeness between two images. An SSIM value of 1 means that the two images are the same.
- Human Visual System (HVS) metric
  Details of HVS metric in dB can be found in [39]. Higher values imply better results.
- HVSm (HVS with masking) [40]
  Similar to HVS, HVS incorporates the visual masking effects in computing the metrics. Higher values imply better results.
- CIELAB

We also used CIELAB [41] for assessing demosaicing and denoising performance in our experiments. Smaller values mean good results.

It should be noted that the HVS and HVSm have better correlation with human perceptions than the other three metrics [42][43].

3. EXPERIMENTAL RESULTS

In this section, we will first compare the performance of different inpainting algorithms on the generation of panchromatic bands. This step is critical for the overall performance of the demosaicing process. We will then focus on several case studies based on the performance of the inpainting results. In particular, we will generate the demosaicing results using the best inpainting method, the previous interpolation method of LDI-NAT in our earlier paper [12], and the ideal case of using the ground truth panchromatic band for inpainting.

3.1. Comparing Different Inpaintingmethods for Pan Band Generation

Here, we will focus on comparing the six different inpainting methods on each image from the KODAK dataset using the CFA3 pattern. For ease of exposition, we only used PSNR. The PSNR is calculated by comparing each inpainted result with the Ground Truth (Reference) pan image, which is generated by taking the average of the RGB bands in the original Kodak image. Table 1 summarizes the PSNR metrics of six inpainting algorithms using the 12 Kodak images. The missing pattern is the CFA 3.0 where 25% of the pixels in the panchromatic bands are missing. Figure 6 shows the averaged PSNR metrics of the five inpainting algorithms. We have the following observations:

- The method of LDI (LDI-NAT), which was used in our earlier paper [12], did not yield the best performance. It is 0.92 dB lower than the best performance algorithm (FOE).
- The deep learning method (GenIn) has a mediocre performance, which may be a little surprising because we had high expectation for it. We think that GenIn may be more

suitable for free-form missing clusters where big chunks of missing blocks with irregular shapes are present in images.

- FOE yielded the best performance, which is somewhat surprising because it was developed long time ago.

Table 1. Bilinear is the same as Laplacian. Inpainting results for 12 panchromatic images using five algorithms.

	Inpaint	FOE	Laplace	LDI	Bilinear	Generative
Img1	47.93	48.00	46.64	45.54	46.64	45.19
Img2	41.06	42.09	39.75	40.61	39.75	41.34
Img3	46.01	46.50	44.45	44.50	44.45	43.85
Img4	39.16	40.47	37.83	39.56	37.83	39.27
Img5	45.44	46.23	43.74	44.20	43.74	45.92
Img6	42.53	43.08	41.35	41.87	41.35	41.19
Img7	43.72	44.76	42.15	42.62	42.15	42.01
Img8	41.78	42.76	40.56	41.79	40.57	41.43
Img9	44.10	44.45	43.29	43.32	43.29	42.44
Img10	41.62	41.68	40.88	41.13	40.88	41.21
Img11	42.85	42.83	42.05	42.08	42.05	42.04
Img12	41.10	40.88	40.32	40.70	40.33	40.16
Average	43.11	43.25	41.92	42.33	41.92	42.17

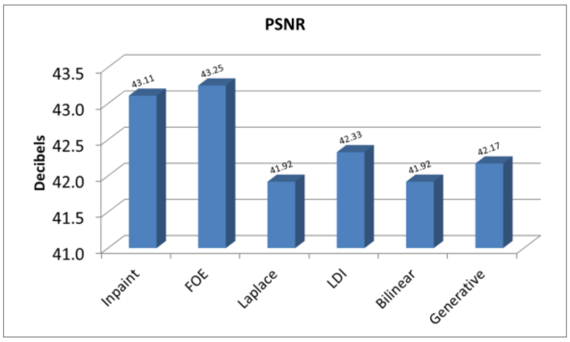


Figure 6. Averaged PSNR of five inpainting algorithms.

3.2. Pansharpening Results using Different Inpaintingmethods

In this section, we will summarize three representative case studies. First, we will summarize the demosaicing results of an earlier approach in which applied LDI-NAT for interpolation the pan band. This will form the baseline for comparisons. Second, we will summarize the demosaicing results using the best inpainting method (FOE) based on results in Section 3.1. Third, we will summarize the ideal demosaicing results where the ground truth pan images are used in the demosaicing process.

3.2.1. LDI-NAT + Pansharpening

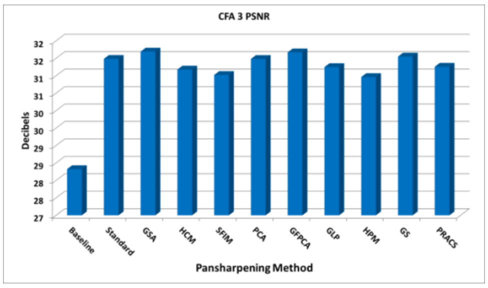
Here, the pan images were generated using LDI-NAT. After that, 11 pansharpening algorithms were applied to demosaic the 12 Kodak images. Table 2 summarizes all the performance metrics of those 12 images. For easier interpretation of those numbers in Table 2, Figure 7 shows the bar charts of the averaged performance metrics. In terms of PSNR, the best performing method is the GSA method. However, GFPCA achieved the highest performance in CIELAB, HVS, and HVSm. In particular, GFPCA is 3 dBs better than all the other methods in terms HVS and HVSmand this is remarkable. The "standard" method has the best score in terms of SSIM. However, the difference is very small between the "standard" method and others.

As mentioned earlier in Section 2, the PSNR and SSIM metrics do not necessarily match well with human perception. This turns out to be indeed the case. From those images in Figure 8, one can easily conclude that the GFPCA results have less artifacts and look closer to the ground truth images. The HVS and HVSm metrics corroborate the above subjective evaluations results.

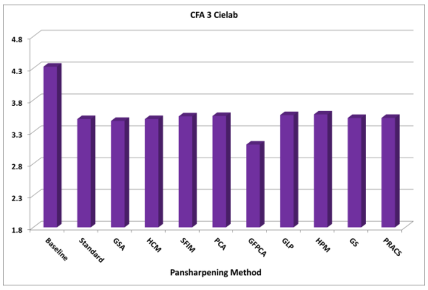
Table 2.Demosaicing results of Kodak images. LDI-NAT was used to generate the pan images.

Image		**Baseline**	**Standard**	**GSA**	**HCM**	**SFIM**	**PCA**	**GFPCA**	**GLP**	**HPM**	**GS**	**PRACS**	**Best Score**
Img1	PSNR	31.936	33.894	34.126	33.526	33.123	33.828	33.158	33.810	33.066	**34.130**	33.466	34.130
	Cielab	2.659	2.374	2.393	2.481	2.699	2.445	2.773	2.429	2.724	**2.351**	2.453	2.351
	SSIM	0.739	**0.859**	0.854	0.831	0.838	0.845	0.806	0.853	0.834	0.858	0.820	0.859
	HVS	28.233	27.610	28.438	28.309	28.453	**28.481**	27.947	28.353	28.468	28.403	28.407	28.481
	HVSm	29.767	29.031	29.881	29.830	29.975	29.934	28.834	29.822	**29.983**	29.836	29.868	29.983
Img2	PSNR	26.771	**30.585**	30.581	30.211	29.918	30.406	30.136	30.089	29.896	30.521	30.230	30.585
	Cielab	4.860	3.766	3.803	3.874	3.901	3.891	**3.219**	3.906	3.904	3.830	3.903	3.219
	SSIM	0.685	**0.868**	0.867	0.856	0.856	0.845	0.823	0.857	0.853	0.850	0.852	0.868
	HVS	23.976	24.331	24.486	24.304	24.264	24.726	**27.637**	24.398	24.228	24.403	24.445	27.637
	HVSm	25.515	25.629	25.803	25.665	25.666	26.076	**29.840**	25.721	25.613	25.703	25.762	29.840
Img3	PSNR	30.815	33.017	32.997	32.156	32.420	32.995	**34.055**	32.698	32.395	33.037	32.648	34.055
	Cielab	3.758	3.378	3.313	3.535	3.432	3.324	**2.949**	3.345	3.459	3.303	3.398	2.949
	SSIM	0.786	**0.888**	0.884	0.870	0.879	0.877	0.873	0.878	0.873	0.877	0.870	0.888
	HVS	27.087	27.099	27.266	27.081	27.211	27.403	**29.897**	27.192	27.218	27.366	27.221	29.897
	HVSm	28.861	28.734	28.928	28.894	28.976	29.065	**31.435**	28.870	28.973	29.023	28.900	31.435
Img4	PSNR	22.762	26.980	**27.496**	27.090	26.808	26.884	26.873	26.762	26.771	26.884	26.933	27.496
	Cielab	7.484	5.434	5.327	5.178	5.314	5.662	**4.841**	5.664	5.364	5.644	5.371	4.841
	SSIM	0.752	0.925	**0.925**	0.919	0.915	0.901	0.891	0.913	0.911	0.903	0.913	0.925
	HVS	20.315	20.370	21.077	21.064	21.160	20.986	**24.117**	21.095	21.203	20.867	20.918	24.117
	HVSm	21.997	21.682	22.476	22.526	22.656	22.377	**26.303**	22.547	22.706	22.240	22.337	26.303
Img5	PSNR	30.816	34.107	33.952	33.686	33.558	33.825	**34.541**	33.694	33.469	34.132	33.766	34.541
	Cielab	2.568	2.100	2.172	2.054	2.107	2.180	**1.914**	2.132	2.123	2.070	2.136	1.914
	SSIM	0.668	**0.868**	0.852	0.859	0.859	0.845	0.798	0.855	0.852	0.859	0.838	0.868
	HVS	27.733	27.824	28.155	28.083	28.083	28.344	**30.514**	28.154	28.083	28.132	28.146	30.514
	HVSm	29.444	29.335	29.707	29.676	29.772	29.912	**32.085**	29.750	29.775	29.662	29.688	32.085
Img6	PSNR	27.706	30.874	31.031	30.391	30.382	30.980	**31.381**	30.647	30.278	30.926	30.601	31.381
	Cielab	5.555	4.605	4.721	4.528	4.464	4.657	**3.797**	4.619	4.544	4.698	4.575	3.797
	SSIM	0.711	**0.896**	0.879	0.877	0.881	0.864	0.848	0.882	0.873	0.860	0.869	0.896
	HVS	24.678	24.823	25.114	24.877	25.031	25.034	**27.599**	25.159	25.047	25.097	24.967	27.599
	HVSm	26.353	26.293	26.606	26.470	26.603	26.520	**29.306**	26.665	26.611	26.591	26.495	29.306
Img7	PSNR	30.446	**34.517**	34.469	34.081	33.701	34.351	33.767	33.917	33.680	34.391	34.183	34.517
	Ciela	3.639	2.751	2.773	2.809	2.855	2.799	**2.501**	2.854	2.857	2.785	2.841	2.501

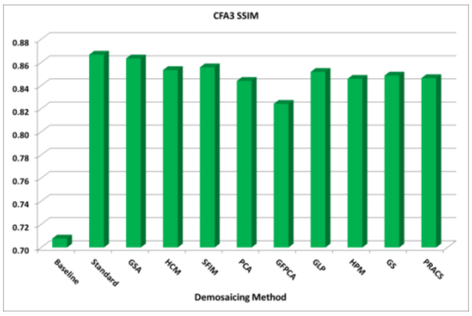
	b												
	SSIM	0.731	**0.904**	0.903	0.896	0.894	0.897	0.853	0.894	0.891	0.897	0.892	0.904
	HVS	27.968	28.395	28.409	28.323	28.199	28.497	**32.017**	28.321	28.159	28.411	28.415	32.017
	HVSm	29.538	29.687	29.696	29.661	29.579	29.800	**34.461**	29.628	29.517	29.706	29.727	34.461
Img8	PSNR	26.939	30.748	**31.078**	30.419	30.253	30.568	30.319	30.390	30.123	30.670	30.439	31.078
	Cielab	4.697	3.707	3.566	3.769	3.704	3.758	**3.200**	3.746	3.734	3.707	3.812	3.200
	SSIM	0.733	**0.900**	0.899	0.885	0.890	0.883	0.860	0.891	0.886	0.888	0.877	0.900
	HVS	24.460	24.087	25.013	24.854	24.969	25.039	**28.845**	25.004	24.996	24.866	24.883	28.845
	HVSm	26.141	25.461	26.453	26.378	26.496	26.473	**30.948**	26.476	26.519	26.277	26.335	30.948
Img9	PSNR	29.775	32.268	32.682	32.117	31.742	32.676	**33.783**	32.316	31.669	32.668	32.318	33.783
	Cielab	3.062	2.705	2.592	2.601	2.911	2.561	**2.202**	2.669	2.974	2.566	2.595	2.202
	SSIM	0.508	0.634	**0.637**	0.623	0.623	0.582	0.615	0.577	0.564	0.582	0.616	0.637
	HVS	26.329	26.028	26.753	26.632	26.808	26.748	**30.150**	26.823	26.820	26.779	26.621	30.150
	HVSm	27.955	27.482	28.234	28.181	28.362	28.228	**31.987**	28.331	28.365	28.264	28.115	31.987
Img10	PSNR	27.054	30.354	30.547	29.970	29.885	30.350	**31.177**	30.014	29.822	30.309	30.118	31.177
	Cielab	4.808	3.975	3.930	3.927	3.915	3.991	**3.223**	4.075	3.940	3.959	3.936	3.223
	SSIM	0.687	0.867	**0.867**	0.856	0.857	0.832	0.802	0.858	0.853	0.855	0.848	0.867
	HVS	24.184	24.135	24.517	24.440	24.459	24.450	**28.393**	24.508	24.441	24.515	24.458	28.393
	HVSm	25.796	25.521	25.928	25.963	25.969	25.867	**30.357**	25.931	25.936	25.935	25.910	30.357
Img11	PSNR	29.027	32.011	**32.234**	31.703	31.707	32.121	31.682	31.835	31.655	32.143	31.687	32.234
	Cielab	4.282	3.556	3.529	3.654	3.606	3.545	**3.412**	3.628	3.627	3.543	3.605	3.412
	SSIM	0.722	0.882	**0.883**	0.866	0.875	0.875	0.840	0.875	0.871	0.876	0.862	0.883
	HVS	26.763	26.320	27.143	27.134	27.175	27.080	**28.744**	27.177	27.215	27.085	27.089	28.744
	HVSm	28.417	27.778	28.626	28.708	28.724	28.548	**30.402**	28.693	28.762	28.552	28.586	30.402
Img12	PSNR	25.845	28.451	29.115	28.779	28.769	28.796	**29.171**	28.807	28.733	28.782	28.712	29.171
	Cielab	4.525	3.669	3.558	3.610	3.621	3.786	**3.176**	3.707	3.649	3.783	3.620	3.176
	SSIM	0.770	0.909	**0.910**	0.903	0.902	0.880	0.883	0.891	0.889	0.880	0.902	0.910
	HVS	24.168	23.290	24.590	24.674	24.658	24.384	**27.561**	24.584	24.652	24.385	24.521	27.561
	HVSm	25.807	24.728	26.100	26.219	26.227	25.842	**29.625**	26.115	26.211	25.843	26.026	29.625
Average	PSNR	28.324	31.484	**31.692**	31.178	31.022	31.482	31.670	31.248	30.963	31.550	31.258	31.692
	Cielab	4.325	3.502	3.473	3.502	3.544	3.550	**3.101**	3.564	3.575	3.520	3.520	3.101
	SSIM	0.708	**0.867**	0.863	0.853	0.856	0.844	0.824	0.852	0.846	0.849	0.846	0.867
	HVS	25.491	25.359	25.913	25.815	25.872	25.931	**28.618**	25.897	25.878	25.859	25.841	28.618
	HVSm	27.132	26.780	27.370	27.347	27.417	27.387	**30.465**	27.379	27.414	27.303	27.312	30.465



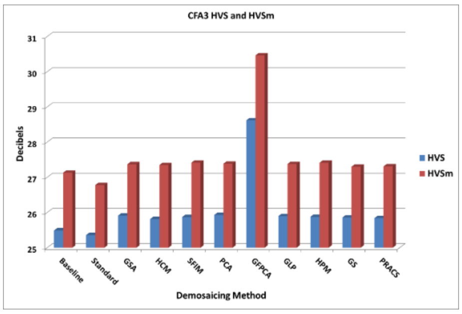
(a) PSNR



(b) CIELAB



(c) SSIM



(d) HVS and HVSm

Figure 7. Averaged performance metrics of all the demosaicing results for the LDI-DAT case.

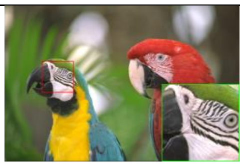
Bird image				
Light-house image				
	(a) GT	(b) GSA	(c) GFPCA	(d) Standard

Figure 8. The ground truth image and three selected demosaiced images using the LDI-NAT inpainting method for pan band.

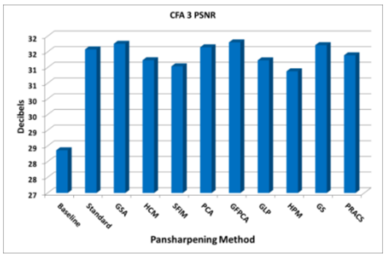
3.2.2. FOE + Pansharpening

In Section 3.1, we observed that the FOE algorithm yielded the best inpainting performance. Here, we show the demosaicing results of the 12 Kodak images by a combination of FOE and various pansharpening algorithms. Table 3 summarizes all the performance metrics of those 12 images. Figure 7 shows the bar charts of the averaged performance metrics. It can be seen that GFPCA achieved the highest performance in PSNR, CIELAB, HVS, and HVSm. Similar to the LDI-NAT case, GFPCA is 3 dBs better than all the other methods in terms HVS and HVSm. From those images in Figure 10, one can easily conclude that the GFPCA results have less artifacts and look closer to the ground truth images.

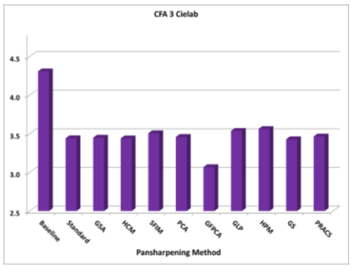
Table 3.Demosaicing results of Kodak images. FOE was used to generate the pan images.

Image		**Baseline**	**Standard**	**GSA**	**HCM**	**SFIM**	**PCA**	**GFPCA**	**GLP**	**HPM**	**GS**	**PRACS**	**Best Score**
Img1	PSNR	32.004	33.988	34.241	33.621	33.150	33.990	33.320	33.897	33.090	**34.295**	33.591	34.295
	Cielab	2.651	2.347	2.387	2.460	2.700	2.412	2.756	2.408	2.728	**2.321**	2.434	2.321
	SSIM	0.741	0.858	0.851	0.830	0.833	0.847	0.807	0.849	0.829	**0.858**	0.821	0.858
	HVS	28.147	27.283	28.239	28.064	28.206	**28.291**	28.034	28.096	28.183	28.220	28.264	28.291
	HVSm	29.690	28.717	29.694	29.606	29.741	**29.756**	28.878	29.589	29.720	29.665	29.735	29.756
Img2	PSNR	26.836	**30.908**	30.855	30.446	30.112	30.747	30.393	30.119	29.876	30.867	30.569	30.908
	Cielab	4.834	3.636	3.693	3.745	3.791	3.751	**3.145**	3.848	3.845	3.688	3.777	3.145
	SSIM	0.690	0.879	**0.879**	0.867	0.865	0.858	0.831	0.863	0.860	0.862	0.865	0.879
	HVS	23.926	24.215	24.322	24.125	24.023	24.577	**27.851**	24.110	23.877	24.253	24.344	27.851
	HVSm	25.455	25.458	25.613	25.459	25.398	25.886	**29.896**	25.434	25.270	25.519	25.615	29.896
Img3	PSNR	30.836	33.074	32.982	32.147	32.424	33.112	**34.350**	32.723	32.312	33.153	32.656	34.350
	Cielab	3.748	3.328	3.316	3.492	3.434	3.243	**2.901**	3.292	3.469	3.223	3.376	2.901
	SSIM	0.788	**0.887**	0.880	0.869	0.877	0.881	0.875	0.877	0.871	0.881	0.869	0.887
	HVS	26.942	26.720	26.877	26.635	26.789	27.048	**30.071**	26.794	26.752	27.015	26.941	30.071
	HVSm	28.725	28.377	28.564	28.479	28.577	28.731	**31.491**	28.505	28.541	28.694	28.639	31.491
Img4	PSNR	22.785	27.014	**27.530**	27.124	26.806	26.938	26.852	26.696	26.677	26.939	27.069	27.530
	Cielab	7.453	5.324	5.249	5.054	5.216	5.552	**4.810**	5.638	5.315	5.534	5.246	4.810
	SSIM	0.754	**0.930**	0.929	0.923	0.919	0.907	0.894	0.916	0.914	0.908	0.918	0.930
	HVS	20.284	20.307	21.038	20.982	21.080	20.941	**24.156**	21.027	21.078	20.823	20.890	24.156
	HVSm	21.962	21.621	22.424	22.444	22.565	22.312	**26.236**	22.474	22.578	22.177	22.287	26.236
Img5	PSNR	30.856	34.218	33.821	33.785	33.588	33.980	**34.732**	33.675	33.420	34.305	33.835	34.732
	Cielab	2.559	2.075	2.240	2.027	2.105	2.136	**1.892**	2.134	2.134	2.025	2.142	1.892
	SSIM	0.670	**0.863**	0.840	0.853	0.850	0.844	0.798	0.846	0.843	0.857	0.832	0.863
	HVS	27.696	27.709	28.050	28.017	27.964	28.309	**30.640**	28.068	27.924	28.097	28.107	30.640
	HVSm	29.403	29.199	29.585	29.601	29.652	29.864	**32.091**	29.660	29.627	29.613	29.629	32.091
Img6	PSNR	27.731	30.919	31.046	30.401	30.348	31.096	**31.422**	30.597	30.138	31.031	30.702	31.422
	Cielab	5.532	4.540	4.814	4.458	4.441	4.505	**3.772**	4.588	4.562	4.551	4.531	3.772
	SSIM	0.716	**0.903**	0.880	0.884	0.885	0.878	0.853	0.885	0.874	0.874	0.875	0.903
	HVS	24.581	24.559	24.929	24.608	24.786	24.868	**27.678**	24.942	24.755	24.922	24.830	27.678
	HVSm	26.256	26.028	26.410	26.225	26.368	26.343	**29.303**	26.449	26.337	26.405	26.348	29.303
Img7	PSNR	30.510	**34.925**	34.820	34.376	33.902	34.788	34.014	33.962	33.648	34.827	34.604	34.925
	Cielab	3.618	2.651	2.683	2.705	2.770	2.686	**2.454**	2.815	2.814	2.669	2.738	2.454
	SSIM	0.736	**0.912**	0.910	0.904	0.901	0.907	0.859	0.899	0.896	0.908	0.901	0.912
	HVS	27.928	28.319	28.313	28.148	27.992	28.420	**32.321**	28.111	27.822	28.333	28.377	32.321
	HVSm	29.493	29.546	29.542	29.457	29.333	29.658	**34.585**	29.388	29.172	29.565	29.622	34.585
Img8	PSNR	26.974	30.834	**31.219**	30.493	30.309	30.797	30.476	30.402	30.104	30.900	30.598	31.219
	Cielab	4.685	3.669	3.517	3.725	3.669	3.681	**3.179**	3.728	3.714	3.626	3.761	3.179
	SSIM	0.735	**0.900**	0.900	0.886	0.889	0.887	0.860	0.890	0.884	0.892	0.879	0.900
	HVS	24.405	23.909	24.904	24.704	24.818	24.954	**28.973**	24.850	24.809	24.787	24.798	28.973
	HVSm	26.086	25.295	26.357	26.259	26.365	26.400	**30.940**	26.353	26.370	26.211	26.246	30.940
Img9	PSNR	29.800	32.236	32.67	32.07	31.57	32.73	**33.914**	32.25	31.44	32.72	32.338	33.91

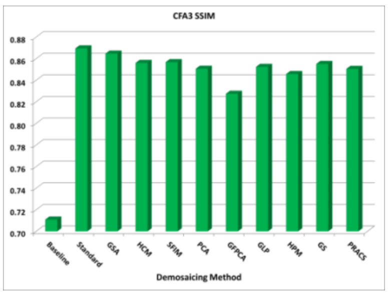
				2	8	5	5		5	7	8		4
	Cielab	3.052	2.674	2.587	2.570	2.927	2.506	**2.174**	2.648	2.998	2.506	2.576	2.174
	SSIM	0.511	0.634	**0.637**	0.623	0.622	0.587	0.615	0.578	0.563	0.588	0.621	0.637
	HVS	26.255	25.799	26.586	26.410	26.604	26.597	**30.276**	26.652	26.585	26.629	26.500	30.276
	HVSm	27.881	27.266	28.062	27.978	28.161	28.073	**31.999**	28.167	28.141	28.110	27.988	31.999
Img10	PSNR	27.087	30.381	30.564	29.944	29.839	30.474	**31.298**	29.928	29.663	30.427	30.217	31.298
	Cielab	4.788	3.932	3.912	3.872	3.883	3.899	**3.190**	4.071	3.936	3.874	3.879	3.190
	SSIM	0.690	**0.867**	0.867	0.855	0.855	0.838	0.806	0.855	0.849	0.858	0.849	0.867
	HVS	24.051	23.775	24.238	24.036	24.066	24.201	**28.526**	24.175	23.976	24.258	24.241	28.526
	HVSm	25.663	25.177	25.643	25.603	25.599	25.609	**30.328**	25.609	25.514	25.668	25.688	30.328
Img11	PSNR	29.048	32.024	**32.245**	31.679	31.699	32.222	31.708	31.779	31.557	32.244	31.757	32.245
	Cielab	4.268	3.513	3.500	3.617	3.577	3.476	**3.394**	3.603	3.611	3.473	3.560	3.394
	SSIM	0.726	0.886	**0.887**	0.870	0.878	0.883	0.843	0.877	0.873	0.884	0.867	0.887
	HVS	26.669	26.089	26.907	26.817	26.905	26.875	**28.814**	26.884	26.887	26.882	26.905	28.814
	HVSm	28.329	27.577	28.414	28.436	28.486	28.368	**30.422**	28.435	28.477	28.374	28.416	30.422
Img12	PSNR	25.869	28.419	29.106	28.715	28.703	28.869	**29.150**	28.732	28.620	28.855	28.744	29.150
	Cielab	4.506	3.609	3.494	3.555	3.569	3.662	**3.149**	3.669	3.613	3.660	3.558	3.149
	SSIM	0.773	0.915	**0.917**	0.908	0.907	0.891	0.888	0.895	0.893	0.892	0.908	0.917
	HVS	24.036	22.977	24.301	24.326	24.342	24.145	**27.476**	24.244	24.285	24.150	24.266	27.476
	HVSm	25.658	24.416	25.776	25.848	25.878	25.579	**29.381**	25.750	25.817	25.584	25.732	29.381
Average	PSNR	28.361	31.578	31.758	31.234	31.038	31.646	**31.802**	31.230	30.879	31.714	31.390	31.802
	Cielab	4.308	3.441	3.449	3.440	3.507	3.459	**3.068**	3.537	3.561	3.429	3.465	3.068
	SSIM	0.711	**0.870**	0.865	0.856	0.857	0.851	0.827	0.852	0.846	0.855	0.851	0.870
	HVS	25.410	25.138	25.725	25.573	25.631	25.769	**28.735**	25.663	25.578	25.697	25.705	28.735
	HVSm	27.050	26.556	27.174	27.116	27.177	27.215	**30.463**	27.151	27.130	27.132	27.162	30.463



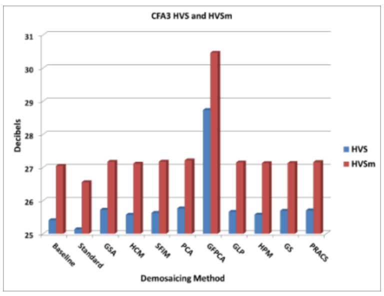
(a) PSNR



(b) CIELAB



(c) SSIM



(d) HVS and HVSm

Figure 9. Averaged performance metrics of all the demosaicing results for the FOE inpainting case.

Figure 10. The ground truth image and three selected demosaiced images using the FOE inpainting method for pan band.

3.2.3. REF + Pansharpening

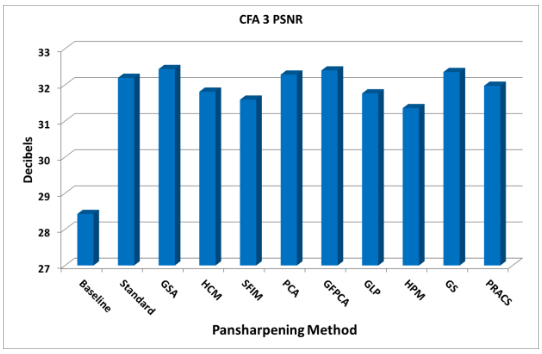
It will be important to show the best achievable demosaicing performance of those 12 Kodak images using CFA 3.0. The gap between this ideal case and what we have so far will indicate the room for further improvement. To generate the ideal demosaicing results, we used the ground truth pan images, which are generated by taking the average of the RGB bands in the original clean Kodak images. Table 4 summarizes the performance metrics for all the demosaicing results. Figure 11 plots the averaged metrics. The GFPCA has the best metrics in CIELAB, HVS, and HVSm. GSA has the best performance in terms of PSNR and the Standard method achieved the best in SSIM. From Figure 12, it is quite obvious that GFPCA has the least artifacts.

Table 4.Demosaicing results of Kodak images. Ground truth pan images were used in the pansharpening process.

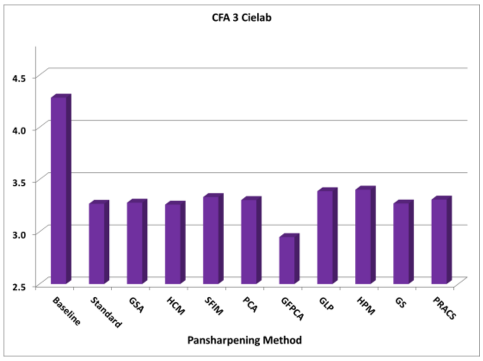
Image		**Baseline**	**Standard**	**GSA**	**HCM**	**SFIM**	**PCA**	**GFPCA**	**GLP**	**HPM**	**GS**	**PRACS**	**Best Score**
Img1	PSNR	32.056	34.252	34.539	33.878	33.385	34.273	33.601	34.119	33.218	**34.597**	33.816	34.597
	Cielab	2.635	2.261	2.311	2.382	2.635	2.333	2.688	2.340	2.673	**2.236**	2.376	2.236
	SSIM	0.746	**0.896**	0.888	0.866	0.873	0.885	0.837	0.891	0.869	0.896	0.848	0.896
	HVS	28.179	27.292	28.313	28.130	28.260	**28.363**	28.112	28.145	28.232	28.291	28.332	28.363
	HVSm	29.715	28.701	29.730	29.642	29.762	**29.789**	28.899	29.610	29.743	29.698	29.771	29.789
Img2	PSNR	26.887	**31.613**	31.551	31.053	30.669	31.432	30.835	30.624	30.339	31.561	31.213	31.613
	Cielab	4.810	3.436	3.499	3.548	3.608	3.568	**3.040**	3.691	3.684	3.501	3.594	3.040
	SSIM	0.695	0.906	**0.906**	0.893	0.892	0.881	0.851	0.890	0.886	0.886	0.890	0.906
	HVS	23.982	24.346	24.464	24.269	24.146	24.723	**28.156**	24.225	23.989	24.387	24.496	28.156
	HVSm	25.505	25.515	25.679	25.541	25.458	25.947	**30.070**	25.486	25.331	25.578	25.692	30.070
Img3	PSNR	30.887	33.356	33.255	32.333	32.621	33.407	**34.886**	32.951	32.476	33.451	32.908	34.886
	Cielab	3.734	3.248	3.236	3.412	3.341	3.161	**2.819**	3.217	3.384	3.141	3.308	2.819
	SSIM	0.792	**0.920**	0.913	0.900	0.910	0.911	0.899	0.910	0.904	0.911	0.898	0.920
	HVS	26.958	26.684	26.880	26.622	26.746	27.058	**30.303**	26.773	26.697	27.025	26.971	30.303
	HVSm	28.736	28.311	28.539	28.449	28.506	28.711	**31.578**	28.459	28.466	28.673	28.641	31.578
Img4	PSNR	22.850	27.850	**28.547**	28.011	27.609	27.718	27.470	27.378	27.368	27.709	27.906	28.547

	Cielab	7.419	5.037	4.972	4.729	4.929	5.327	**4.629**	5.412	5.052	5.306	4.978	4.629
	SSIM	0.758	**0.953**	0.952	0.946	0.942	0.927	0.913	0.939	0.936	0.929	0.940	0.953
	HVS	20.336	20.427	21.253	21.218	21.306	21.133	**24.531**	21.231	21.307	21.011	21.081	24.531
	HVSm	22.007	21.670	22.554	22.606	22.707	22.418	**26.453**	22.603	22.733	22.281	22.401	26.453
Img5	PSNR	30.918	34.782	34.337	34.285	34.072	34.514	**35.281**	34.126	33.833	34.874	34.328	35.281
	Cielab	2.539	1.955	2.146	1.903	1.990	2.033	**1.817**	2.034	2.031	1.917	2.049	1.817
	SSIM	0.677	**0.928**	0.903	0.917	0.917	0.907	0.831	0.914	0.910	0.920	0.890	0.928
	HVS	27.739	27.778	28.163	28.144	28.071	28.428	**30.870**	28.171	28.021	28.213	28.219	30.870
	HVSm	29.436	29.212	29.632	29.667	29.698	29.909	**32.173**	29.705	29.671	29.660	29.679	32.173
Img6	PSNR	27.796	31.616	31.775	30.997	30.937	31.816	**32.177**	31.213	30.646	31.740	31.315	32.177
	Cielab	5.508	4.331	4.611	4.230	4.197	4.348	**3.619**	4.398	4.339	4.385	4.341	3.619
	SSIM	0.720	**0.934**	0.909	0.913	0.916	0.902	0.874	0.916	0.904	0.898	0.902	0.934
	HVS	24.627	24.620	25.068	24.739	24.903	25.002	**28.061**	25.070	24.868	25.057	24.969	28.061
	HVSm	26.295	26.029	26.484	26.311	26.422	26.413	**29.485**	26.512	26.395	26.474	26.433	29.485
Img7	PSNR	30.561	**35.654**	35.516	34.954	34.410	35.479	34.405	34.453	34.069	35.511	35.273	35.654
	Cielab	3.598	2.496	2.535	2.554	2.633	2.540	**2.374**	2.696	2.694	2.524	2.599	2.374
	SSIM	0.741	**0.942**	0.940	0.933	0.930	0.936	0.882	0.928	0.924	0.936	0.929	0.942
	HVS	27.968	28.420	28.420	28.245	28.069	28.528	**32.628**	28.195	27.882	28.440	28.500	32.628
	HVSm	29.520	29.564	29.566	29.488	29.345	29.680	**34.727**	29.402	29.176	29.588	29.663	34.727
Img8	PSNR	27.013	31.282	**31.717**	30.880	30.677	31.221	30.804	30.762	30.419	31.331	31.015	31.717
	Cielab	4.658	3.468	3.310	3.536	3.476	3.503	**3.051**	3.546	3.534	3.449	3.596	3.051
	SSIM	0.740	0.937	**0.937**	0.921	0.926	0.923	0.890	0.927	0.921	0.927	0.913	0.937
	HVS	24.453	23.969	25.015	24.811	24.916	25.060	**29.237**	24.955	24.907	24.889	24.909	29.237
	HVSm	26.129	25.317	26.420	26.326	26.416	26.455	**31.046**	26.415	26.425	26.264	26.311	31.046
Img9	PSNR	29.869	32.768	33.338	32.636	32.078	33.414	**34.743**	32.823	31.912	33.407	32.908	34.743
	Cielab	3.037	2.555	2.462	2.437	2.811	2.371	**2.078**	2.539	2.892	2.371	2.457	2.078
	SSIM	0.512	0.665	**0.668**	0.653	0.653	0.618	0.636	0.609	0.594	0.619	0.651	0.668
	HVS	26.315	25.823	26.770	26.607	26.787	26.781	**30.702**	26.833	26.768	26.814	26.668	30.702
	HVSm	27.938	27.232	28.181	28.121	28.277	28.192	**32.204**	28.285	28.265	28.229	28.097	32.204
Img10	PSNR	27.156	31.142	31.373	30.600	30.491	31.212	**32.149**	30.538	30.209	31.161	30.910	32.149
	Cielab	4.755	3.702	3.688	3.630	3.639	3.703	**3.025**	3.875	3.717	3.665	3.671	3.025
	SSIM	0.696	0.916	**0.916**	0.903	0.905	0.881	0.829	0.905	0.899	0.904	0.894	0.916
	HVS	24.115	23.899	24.405	24.192	24.209	24.365	**29.045**	24.327	24.112	24.424	24.404	29.045
	HVSm	25.720	25.235	25.738	25.703	25.677	25.704	**30.552**	25.693	25.594	25.763	25.790	30.552
Img11	PSNR	29.136	32.760	**33.043**	32.336	32.370	32.998	32.289	32.466	32.181	33.026	32.385	33.043
	Cielab	4.238	3.313	3.300	3.430	3.378	3.278	**3.265**	3.424	3.427	3.276	3.385	3.265
	SSIM	0.731	0.919	**0.919**	0.902	0.911	0.914	0.868	0.910	0.906	0.916	0.898	0.919
	HVS	26.758	26.210	27.108	26.996	27.083	27.074	**29.120**	27.067	27.059	27.080	27.104	29.120
	HVSm	28.408	27.623	28.524	28.536	28.574	28.474	**30.554**	28.530	28.564	28.479	28.531	30.554
Img12	PSNR	25.977	29.260	**30.255**	29.758	29.756	29.915	30.136	29.711	29.595	29.894	29.716	30.255
	Cielab	4.472	3.408	3.272	3.327	3.348	3.475	**2.983**	3.486	3.409	3.473	3.344	2.983
	SSIM	0.778	0.940	**0.942**	0.933	0.932	0.914	0.910	0.920	0.918	0.915	0.932	0.942
	HVS	24.186	23.100	24.702	24.750	24.761	24.520	**28.263**	24.630	24.720	24.523	24.646	28.263
	HVS	25.815	24.465	26.07	26.18	26.19	25.85	**29.911**	26.04	26.16	25.85	26.020	29.911

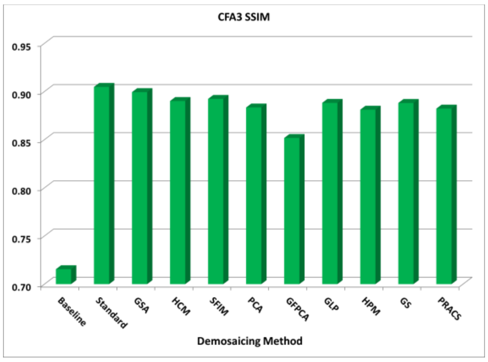
	m			4	4	5	2		7	7	6		
Average	PSNR	28.426	32.195	**32.437**	31.810	31.590	32.283	32.398	31.764	31.355	32.355	31.974	32.437
	Cielab	4.283	3.268	3.278	3.260	3.332	3.303	**2.949**	3.388	3.403	3.270	3.308	2.949
	SSIM	0.715	**0.905**	0.899	0.890	0.892	0.883	0.852	0.888	0.881	0.888	0.882	0.905
	HVS	25.468	25.214	25.880	25.727	25.771	25.920	**29.086**	25.802	25.713	25.846	25.858	29.086
	HVSm	27.102	26.573	27.260	27.215	27.253	27.295	**30.638**	27.229	27.211	27.212	27.253	30.638



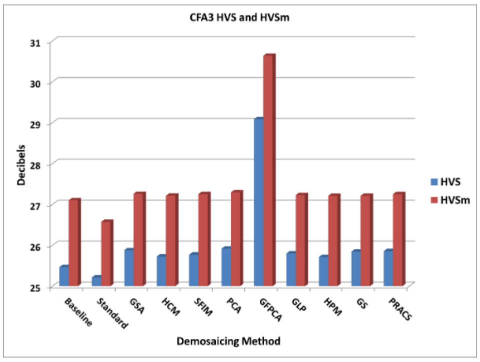
(a) PSNR



(b) CIELAB



(c) SSIM



(d) HVS and HVSm

Figure 11. Averaged performance metrics of all the demosaicing results for the ground truth pan case.

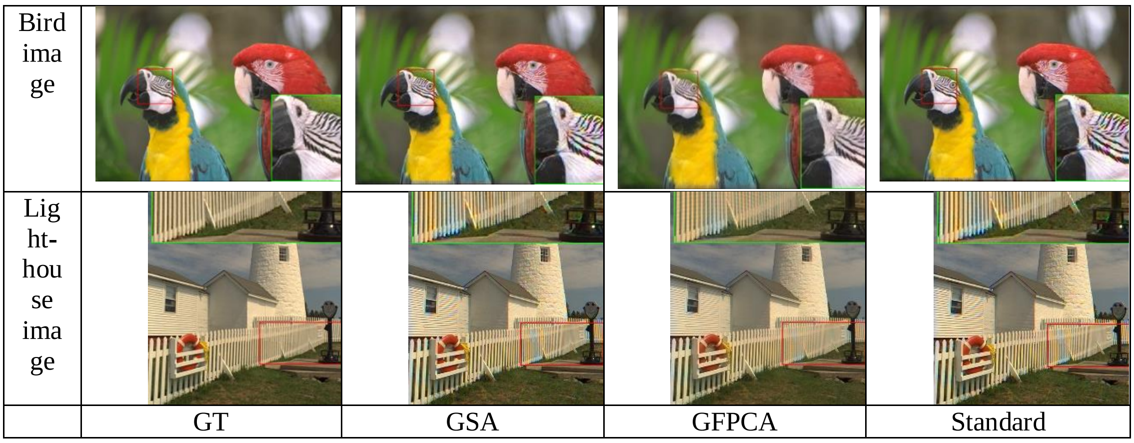


Figure 12. The ground truth image and three selected demosaiced images using the ground truth pan case.

3.3. Discussions and Comparisons

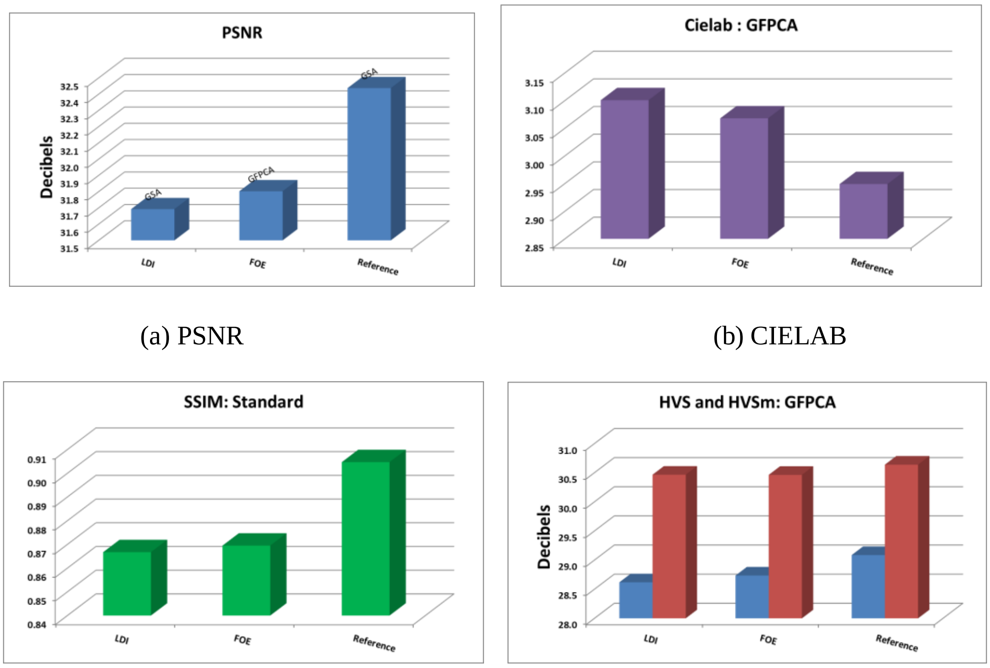
We summarize the key results in Tables 2 to 4 and put them into Table 5 and Figure 13. We have the following observations:

- In terms of PSNR, the combination of FOE and GFPCA improved over the combination of LDI-NAT and GSA by 0.11 dB. There is still a somewhat big gap of 0.635 dB between FOE/GFPCA and GT/GSA.
- In terms of CIELAB, the difference between the FOE/GFPCA and LDI-NAT/GSA is 0.033 whereas the difference between GT/GFPCA and FOE/GFPCA is 0.119. Relatively speaking, there is still a gap for further improvement.
- In terms of SSIM, the difference between FOE/Standard and LDI-NAT/Standard is 0.003 and the difference between GT/Standard and FOE/Standard is 0.035.
- In terms of HVS, the difference between FOE/GFPCA and LDI-NAT/GFPCA is 0.117 dB and the difference between GT/GFPCA and FOE/GFPCA is 0.351 dB.
- In terms of HVSm, the difference between FOE/GFPCA and LDI-NAT/GFPCA is 0.002 dB and the difference between GT/GFPCA and FOE/GFPCA is 0.175 dB.

The above observations also answer the two questions raised in Section 1. First, after some extensive experiments, it was found that there do exist better algorithms (FOE and inpaint-nans) than the LDI-NAT method. Second, we also quantify the performance gain of the better algorithms. In short, it appears that the best inpainting algorithm (FOE) closes the gap between the FOE and the ideal case. However, even the demosaicing results with GT pan may still have room for improvement, which will be a future topic to pursue.

Table 5.Demosaicing results

	LDI-NAT for Pan	**FOE for Pan**	**Reference (GT) Pan**
Metrics	**Metric/ Best PS**	**Metric/ Best PS**	**Metric/Best PS**
PSNR (dB)	31.692/GSA	31.802/GFPCA	32.437/GSA
CIELAB	3.101/GFPCA	3.068/GFPCA	2.949/GFPCA
SSIM	0.867/Standard	0.870/Standard	0.905/Standard
HVS (dB)	28.618/GFPCA	28.735/GFPCA	29.086/GFPCA
HVSm (dB)	30.465/GFPCA	30.463/GFPCA	30.638/GFPCA



(a) PSNR (b) CIELAB

(c) SSIM (d) HVS and HVSm

Figure 13. Comparison of the demosaicing results of using different combinations of inpainting and pansharpening algorithms.

4. CONCLUSIONS

In this paper, we focus on further improving the demosaicing performance of CFA 3.0. Our idea is to see if newer inpainting algorithms can help improve the overall demosaicing performance. Six conventional and deep learning based methods were compared and the FOE method yielded slight better performance than others. One key observation is that there is still room for improvement because, when we used the ground truth pan band, the overall demosaicing performance is much better than what we have right now. Hence, one future direction is to seek better inpainting methods. Another direction is to develop an end-to-end deep learning approach to demosaicing CFA 3.0.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGEMENTS

This work was partially supported by NASA Jet Propulsion Laboratory under contract # 80NSSC17C0035. The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of NASA or the U.S. Government.

REFERENCES

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

[2] J. F. Bell III, et al., "The Mars Science Laboratory Curiosity Rover Mast Camera (Mastcam) Instruments: Pre-Flight and In-Flight Calibration, Validation, and Data Archiving," AGU Journal Earth and Space Science, 2017.
[3] M. Dao, C. Kwan, B. Ayhan, and J. F. Bell, "Enhancing Mastcam Images for Mars Rover Mission," 14th International Symposium on Neural Networks, pp. 197-206, 2017.
[4] C. Kwan, B. Budavari, M. Dao, B. Ayhan, and J. F. Bell, "Pansharpening of Mastcam images," IEEE International Geoscience and Remote Sensing Symposium, pp. 5117-5120, Fort Worth, Texas, 2017.
[5] B. Ayhan, M. Dao, C. Kwan, H. Chen, J. F. Bell, and R. Kidd, "A Novel Utilization of Image Registration Techniques to Process Mastcam Images in Mars Rover with Applications to Image Fusion, Pixel Clustering, and Anomaly Detection," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(10), pp. 4553-4564, 2017.
[6] J. Hamilton and J. Compton, Processing color and panchromatic pixels. U.S. Patent 20070024879A1, 2007.
[7] T. Kijima, H. Nakamura, J. T. Compton, J. F.; Hamilton, and T. E. DeWeese, Image sensor with improved light sensitivity. U.S. Patent 0 268 533, Nov., 2007.
[8] C. Kwan and J. Larkin, "Demosaicing of Bayer and CFA 2.0 Patterns for Low Lighting Images," Electronics, 8, 1444, 2019.
[9] C. Kwan and J. Larkin, "Comparison of Denoising Algorithms in Demosacing Low Lighting Images Using CFA 2.0,"Signal & Image Processing: An International Journal (SIPIJ), vol. 11, no. 5, October 29, 2020.
[10] C. Kwan and J. Larkin, "Demosaicing for Mastcam Images Using A New Color Filter Array," Signal & Image Processing: An International Journal (SIPIJ), Vol. 11, No. 3, May 31, 2020.
[11] C. Kwan, J. Larkin, and B. Budavari, "Demosaicing of Real Low Lighting Images Using CFA 3.0," Signal & Image Processing: An International Journal (SIPIJ), vol. 11, no. 4, August 2020.
[12] C. Kwan, J. Larkin, and B. Ayhan, "Demosaicing of CFA 3.0 with Application to Low Lighting Images," Sensors, 20(12), 3423, June 22, 2020.
[13] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," J. Electron. Imaging, 20, 2011.
[14] D. Doshkov, P. Ndjiki-Nya, H. Lakshman, M. Köppel, and T. Wiegand, "Towards efficient intra prediction based on image inpainting methods," 28th Picture Coding Symposium. IEEE, 2010.
[15] Inpaint_nans, https://www.mathworks.com/matlabcentral/fileexchange/4551-inpaint_nans, accessed on October 29, 2020.
[16] S. Roth and M. J. Black, "Fields of Experts," Int J Comput Vis., 82: 205, 2009.
[17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Generative Image Inpainting with Contextual Attention," arXiv:1801.07892 [cs.CV]. 2018.
[18] C. Kwan, B. Chou, L. M. Kwan, and B. Budavari, "Debayering RGBW Color Filter Arrays: A Pansharpening Approach," IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference, pp. 94-100, New York City, 2017.
[19] G. Vivone, et al.,"A Critical Comparison Among Pansharpening Algorithms," IEEE Trans. Geoscience and Remote Sensing, 53(5), 2015.
[20] J. G. Liu, "Smoothing filter based intensity modulation: A spectral preserve image fusion technique for improving spatial details," Int. J. Remote Sens., 21, 18, 2000.
[21] B. Aiazzi, et al.,"MTF-tailored multiscale fusion of high-resolution MS and pan imagery," Photogramm. Eng. Remote Sens., 72(5), pp. 591–596, 2006.
[22] G. Vivone, et al., "Contrast and error-based fusion schemes for multispectral image pansharpening," IEEE Trans. Geosci. Remote Sensing Lett., 11(5), pp. 930–934, 2014.
[23] C. Laben and B. Brower, Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening. U.S. Patent 6 011 875, Jan. 4, 2000.
[24] B. Aiazzi, et al., "Improving component substitution pansharpening through multivariate regression of MS+pan data," IEEE Trans. Geosci. Remote Sensing, 45(10), pp. 3230–3239, 2007.
[25] W. Liao, et al., "Processing of multiresolution thermal hyperspectral and digital color data: Outcome of the 2014 IEEE GRSS data fusion contest," IEEE J. Select. Top. Appl. Earth Observ. Remote Sensing, 8, 6, 2015.
[26] J. Choi, et al., "A new adaptive component-substitution based satellite image fusion by using partial replacement," IEEE Trans. Geosci. Remote Sens., 49, 1, 2011.

[27] J. Zhou, C. Kwan, and B. Budavari, "Hyperspectral image super-resolution: A hybrid color mapping approach," Journal of Applied Remote Sensing, 10, 3, article 035024, 2016.
[28] C. Kwan, J. H. Choi, S. Chan, J. Zhou, and B. Budavai, "Resolution Enhancement for Hyperspectral Images: A Super-Resolution and Fusion Approach," IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 6180 – 6184, New Orleans, 2017.
[29] C. Kwan, B. Budavari, and G. Feng, "A Hybrid Color Mapping Approach to Fusing MODIS and Landsat Images for Forward Prediction," Remote Sensing, 10(4), 520, 2017.
[30] C. Kwan, B. Budavari, A. Bovik, and G. Marchisio, "Blind Quality Assessment of Fused WorldView-3 Images by Using the Combinations of Pansharpening and Hypersharpening Paradigms," IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 10, pp. 1835-1839, Oct. 2017.
[31] C. Kwan, B. Ayhan, and B. Budavari, "Fusion of THEMIS and TES for Accurate Mars Surface Characterization," IEEE International Geoscience and Remote Sensing Symposium, pp. 3381-3384, Fort Worth, Texa, 2017.
[32] L. Zhang and X. Wu, "Colordemosaicking via directional linear minimum mean square-error estimation," IEEE Trans. Image Processing, 14, 2167–2178, 2005.
[33] W. Lu and Y. P. Tan, "Color filter array demosaicking: New method and performance measures," IEEE Trans. on Image Processing, 12, 1194–1210, 2003.
[34] E. Dubois, "Frequency-domain methods for demosaicking of Bayer-sampled color images," IEEE Signal Proc. Letters, 12, 847–850, 2005.
[35] B. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," IEEE Transactions on Image Processing, 11, 997–1013, 2002.
[36] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," IEEE Trans. on Image Processing, 13, 1263-1274, 2004.
[37] C. Kwan, X. Zhu, F. Gao, B. Chou, D. Perez, J. Li, Y. Shen, and K. Koperski, "AssessmentofSpatiotemporalFusionAlgorithmsforPlanetandWorldviewImages," Sensors,18,1051, 2018.
[38] SSIM.Availableonline:https://en.wikipedia.org/wiki/Structural_similarity.(Accessedon26April2019).
[39] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli, "New full quality metrics based on HVS," Second International Workshop on Video Processing and Quality Metrics, Scottsdale, AZ, USA, 22–24 January 2006.
[40] N. Ponomarenko,F.Silvestri,K.Egiazarian,M. Carli, J. Astola, and V. Lukin, "Onbetween-coefficientcontrastmaskingofDCTbasisfunctions,"Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-07,Scottsdale,AZ,USA,25–26January2007.
[41] X. Zhang and B. A. Wandell, "A spatial extension of cielab for digital color image reproduction," SID Journal, 1997.
[42] C. Kwan, J. Larkin, B. Budavari, B. Chou, E. Shang, and T. D. Tran, "A comparison of compression codecs for maritime and sonar images in bandwidth constrained applications," Computers, 8(2), 32, April 28, 2019.
[43] C. Kwan, J. Larkin, B. Budavari, E. Shang, and T. Tran, "Perceptually Lossless Compression with Error Concealment for Periscope and Sonar Videos," Signal & Image Processing: An International Journal (SIPIJ), vol. 10(02); pages 01-14, April 30, 2019.

AUTHORS

Chiman Kwan received his Ph.D. degree in electrical engineering from the University of Texas at Arlington in 1993. He has written one book, four book chapters, 15 patents, 70 invention disclosures, 380 technical papers in journals and conferences, and 550 technical reports. Over the past 25 years, he has been the PI/Program Manager of over 120 diverse projects with total funding exceeding 36 million dollars. He is also the founder and Chief Technology Officer of Signal Processing, Inc. and Applied Research LLC. He received numerous awards from IEEE, NASA, and some other agencies and has given several keynote speeches in several international conferences.

Jude Larkin received his B.S. in Computer Science from Franciscan University of Steubenville in 2015. He is a software engineer at ARLLC. He has been involved in diverse projects, including mission planning for UAVs, image fusion, image demosaicing, and remote sensing.