Image Analysis Final Project

Theresa EL MURR

January 31, 2024

1 Introduction

In digital photography, a Color Filter Array (CFA) is a mosaic of tiny color filters placed over the pixel sensors of an image sensor to capture color information. The most common types of CFA patterns are the Bayer pattern and the Quad Bayer pattern. Each pixel under the CFA captures only one of the three primary colors (red, green, or blue), resulting in a grayscale image that contains incomplete color information—this is known as a raw image.

1.1 Problem Statement

The process of demosaicking involves reconstructing a full-color image from this raw data. The challenge lies in accurately interpolating the missing colors for each pixel to recover the complete RGB image without introducing artifacts, such as aliasing or false colors, which can significantly degrade image quality. In this project, we are tasked with implementing a demosaicking algorithm that utilizes only the raw grayscale image and the forward operator provided, modeling the effects of a CFA camera. The goal is to reconstruct four images from the open dataset of the National Gallery of Art, USA, for either the Bayer or Quad Bayer pattern.

1.2 Objective

The project aims to apply the different image analysis techniques learned over the semester in order to obtain a robust demosaicking algorithm capable of handling high-resolution images (1024x1024) and demonstrate effectiveness through quantitative metrics and visual assessment.

2 The Chosen Solution: Microsoft Linear Demosaicing with 5x5 Kernel Convolution

This method, proposed by H. Malvar, L. He, and R. Cutler in "High-quality linear interpolation for demosaicking of images Bayer-patterned color" , consists of linear interpolation for demosaicking Bayer-patterned color images. This method is particularly helpful because it doesn't require the separation of color planes since the filters are applied directly to the raw image.[Get11][MHC04][Pyx21]

2.1 The Theory Behind the Algorithm

The concept behind this method is that when interpolating missing pixels within each channel, using only the adjacent pixels within the same channel (Nearest Neighbors Demosaicing), the result might not be very accurate. To elaborate, it's necessary to consider the values of the neighboring pixels as well as values from the other available channels.

2.2 The Algorithm

The algorithm first extracts the Red, Blue, and Green channels from the raw image based on the Bayer CFA pattern. Then, it applies 2D convolution specific to certain pixel positions in the Bayer pattern. These filters are used to perform linear interpolation to estimate missing color values. The algorithm utilizes a set of predefined 2D filter coefficients which are shown in the figure below:

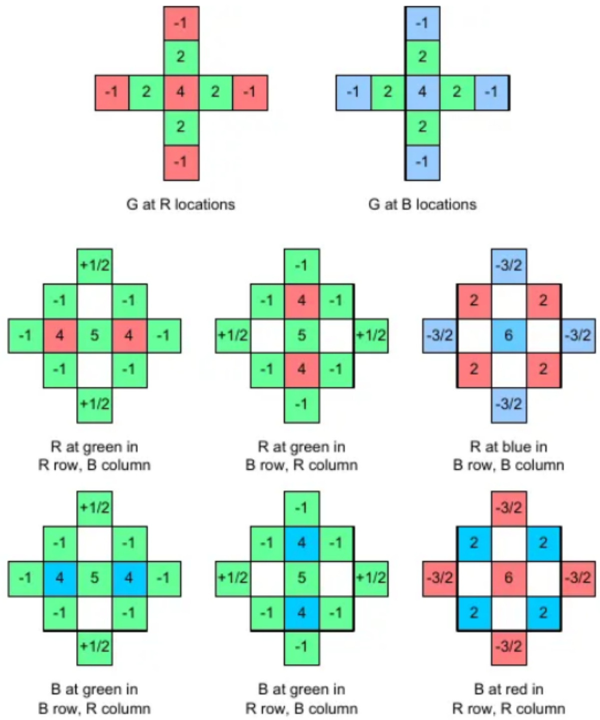


Figure 1: Filter Coefficients for High-Quality Linear Interpolation

These filters are then convolved with the raw image to estimate the missing values accurately, and the final result is the full-color image with improved quality compared to the naive implementation provided as an example.

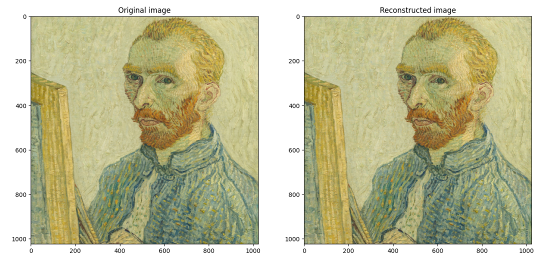
Here are the results obtained for the proposed algorithm:

Image	PSNR	SSIM
img_1	39.14	0.9801
img_2	34.52	0.9443
img_3	35.31	0.9515
img_4	33.57	0.9226

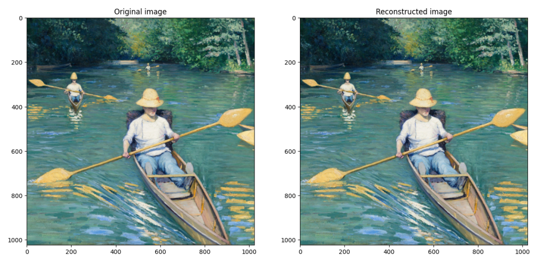
Table 1: PSNR and SSIM results of the Microsoft Linear Demosaicing algorithm on the different images.

And Here are the different outputs for each raw image by this method:

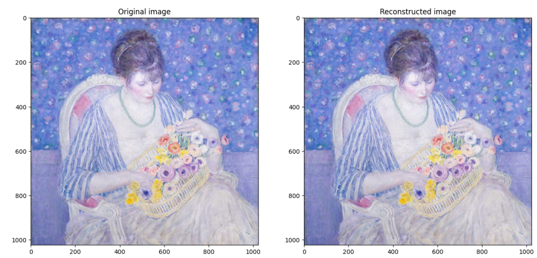
(a) Ouput of img1 with bayer



(b) Ouput of img2 with bayer



(c) Ouput of img3 with bayer



(d) Ouput of img4 with bayer

Figure 2: Output of the Malver-He-Cutler algorithm with Bayer CFA

2.3 Adaptation of the Algorithm to the Quad Bayer CFA

The previous algorithm showed excellent results, both visually and in terms of the PSNR and SSIM metrics, which appeared to be better than the Naive method proposed. However, the previous algorithm proposed the use of 5x5 filters and was applicable to the Bayer CFA. To be able to use this method on the Quad Bayer CFA, I realized that the Quad Bayer is similar to the Bayer, but each row and column is duplicated. So, I decided to perform this logic on the filters used for the convolution in the proposed method and duplicate each row and column to obtain 10x10 filters. Here's an example of how the filters were adapted to be applied when we have a Quad Bayer CFA:

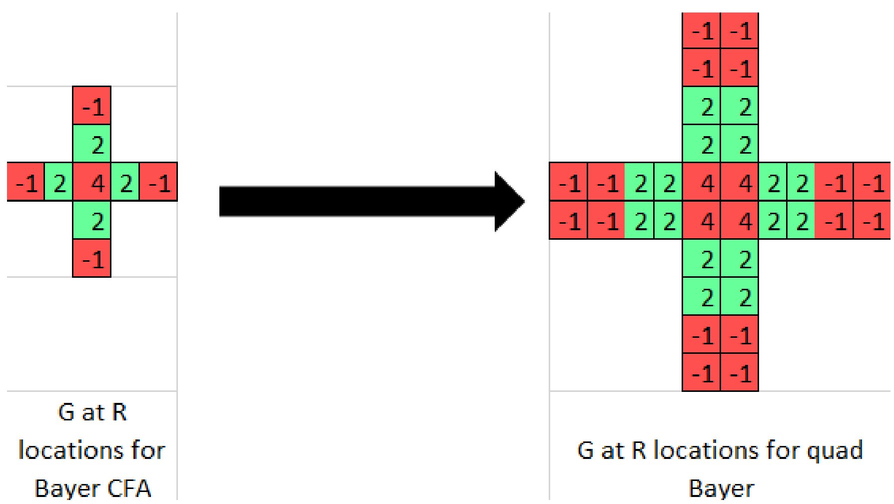


Figure 3: Example of Filter G at R locations transformation to be applied to the problem using Quad Bayer CFA

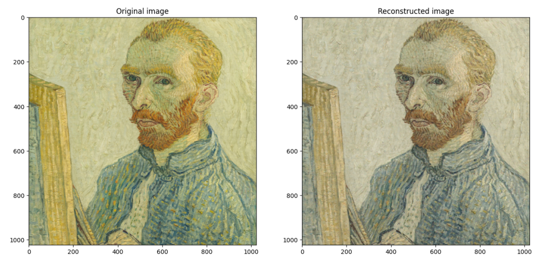
After changing the different filters, we applied the same logic as explained previously. The entire conversion was based on a change of the filters. The results obtained were good, and the algorithm was performant, but the results obtained using the naive implementations were better, which means that not only was the naive method more efficient, but it also required fewer resources (space and computational time). In this table, we can look at the results obtained while performing the Malvar-He-Cutler algorithm on a raw image with a Quad Bayer CFA:

Image	PSNR	SSIM
img_1	27.69	0.8392
img_2	22.41	0.5732
img_3	24.08	0.6390
img_4	22.93	0.6135

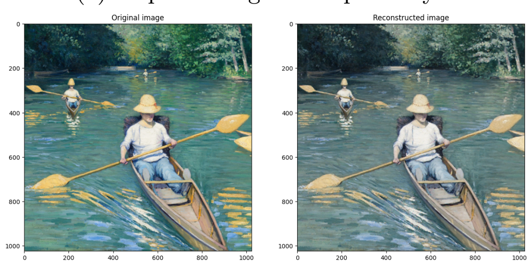
Table 2: PSNR and SSIM results of the Microsoft Linear Demosaicing algorithm on the different raw images with a Quad Bayer CFA.

And here are the images obtained by this method:

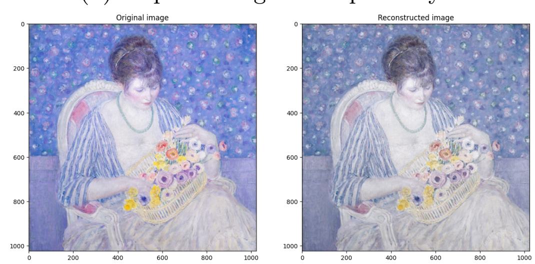
(a) Ouput of img1 with quad bayer



(b) Ouput of img2 with quad bayer



(c) Ouput of img3 with quad bayer



(d) Ouput of img4 with quad bayer

Figure 4: Output of the Malver-He-Cutler algorithm with Quad Bayer CFA

3 Conclusion

In conclusion, the Microsoft Linear Demosaicking with 5x5 kernel convolution algorithm, also known as the Malvar-He-Cutler algorithm, is a well-performing algorithm and is highly efficient in demosaicking raw images with a Bayer CFA. Therefore, it is recommended for demosaicking problems. However, the adaptation I provided performed well when we had a Quad Bayer CFA, but it required 10x10 filters and therefore more storage and computational time, and it was less performant than the naive implementation. To address this issue, we could try the Quad to Bayer conversion before applying the Malvar-He-Cutler method or simply use the naive implementation provided as an example since it performed better.

References

[Get11] Pascal Getreuer. Malvar-He-Cutler Linear Image Demosaicking. *Image Processing On Line*, 1:83–89, 2011. https://doi.org/10.5201/ipol.2011.g_mhcd.

[MHC04] H.S. Malvar, Li-wei He, and R. Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[Pyx21] Pyxalis. Pyx imageviewer user guide. https://pyxalis.com/wp-content/uploads/2021/12/PYX-ImageViewer-User_Guide.pdf, 2021.