Image Analysis project – Demosaicking

LIORET Nathan

I- Introduction

Most of RGB cameras use Color Filters Array (CFA) to filter the light before the sensors in each of the red, green and blue bands. This way we get the three color chanels of an area, distributed as shown in **Figure 1** for example (Bayer and Quad-Bayer patterns). In order to keep the same number of pixels, we must determine the values of the two other colors for each pixel (red and blue for green pixels, blue and green for red pixels and red and green for blue pixels). This is called demosaicking.

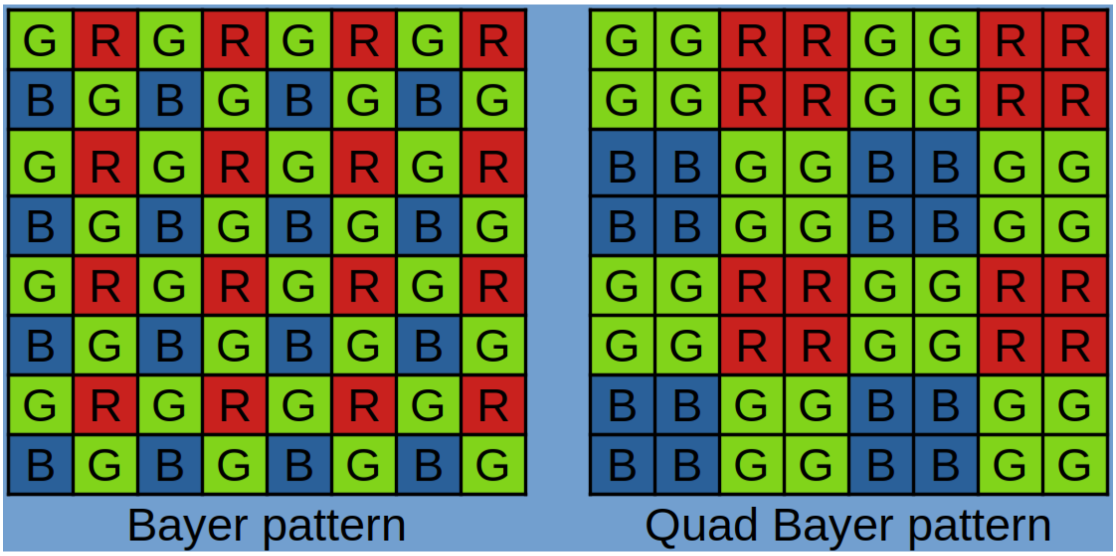

Figure 1 : Bayer and Quad-Bayer configurations *[1]*

Among the existing methods, many use interpolation to estimate the missing colors. We can cite the bilinear approach which is very common. In this project, we will see another, bit more complex, interpolation method called High-Quality Linear Interpolation (HQLI), which was first introduced for demosaicking Bayer patterns in *[2]*.

In order to test our model, we will compare the results obtained for each of the 4 images in **Figure 2** to the results obtained using the “naïve interpolation” given as a first approach to this project *[1]* :

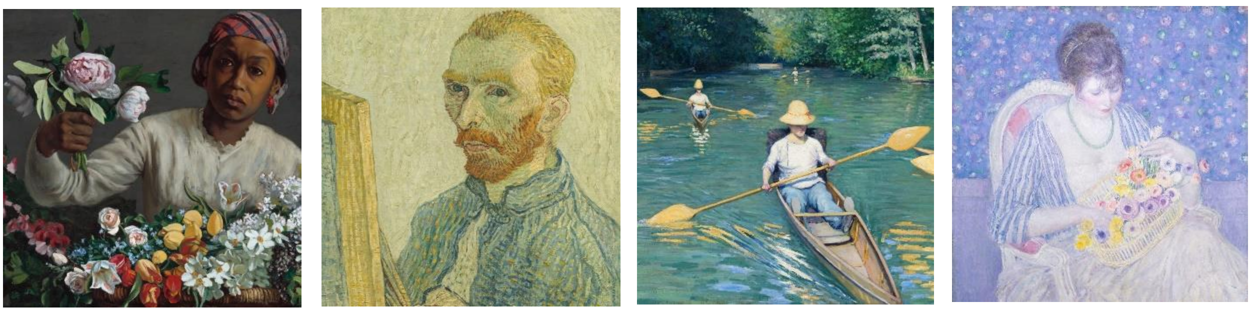

Figure 2 : Considered images for the testing of our proposed demosaicking solution *[3]*

II- High-Quality Linear Interpolation

This method was first introduced in *[2]* to deal with Bayer patterns. It generalises the bilinear approach by considering 5x5 linears filters, each filter corresponding to a specific configuration of color adjancement as shown in **Figure 3** (coefficients are to be divided by 8 in order to stay in the same range of values in the image) :

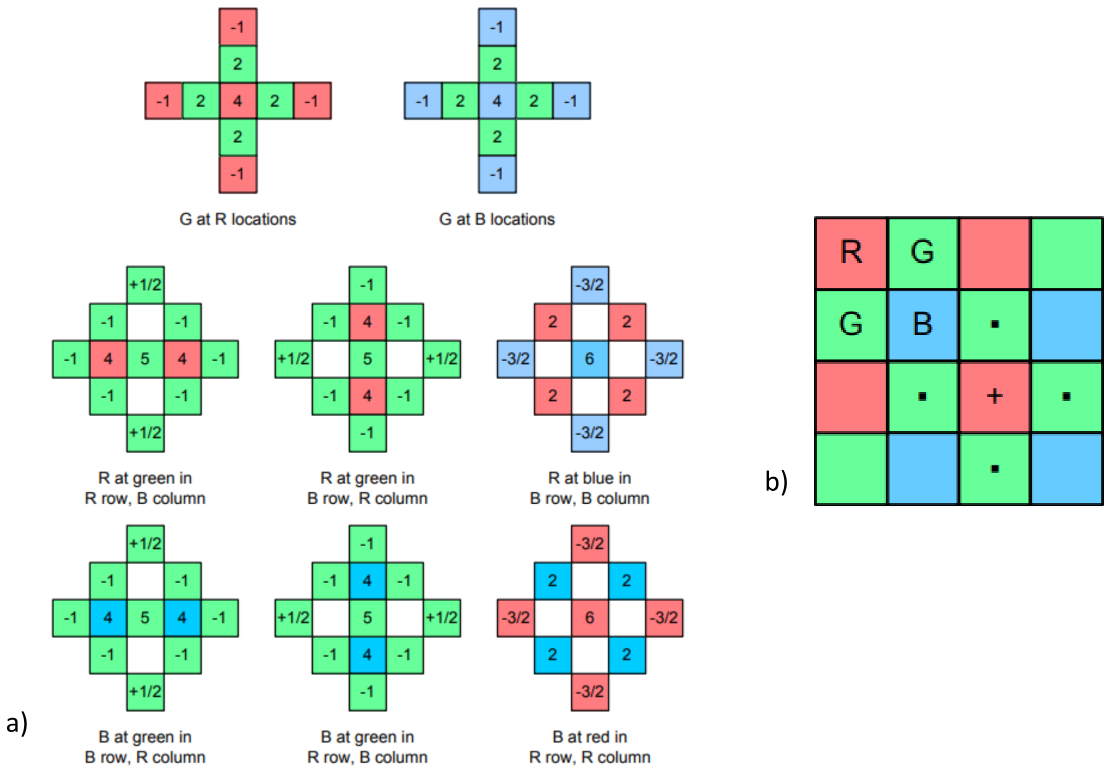

Figure 3 : a) HQLI filters corresponding to each pixel configuration (R : red, G : green, B : blue) for a typical Bayer pattern b)

To generalize the HQLI method to Quad-Bayer paterns, we can simply consider Quad-Bayer as a Bayer patern with bigger “pixels” (made of 4 pixels). This way, the considered filters will have size 10x10 instead of 5x5, with the same disposition of coefficients but divided by 32 instead of 8 to stay in the same range of values than the original image. Considering that when using this approximation, 4 coincident pixels of the reconstructed image will have the same value, we loose information so we can expect less convincing results than for the classical Bayer patern (see section III-Results).

III- Results

In this section we take a closer look at the results obtained using HQLI for Bayer and Quad-Bayer configurations. We compare those results to the ones obtained using naïve interpolation in **Table 1**.

Image 1				
	Naïve Bayer	*HQLI Bayer*	*Naïve Quad*	*HQLI Quad*
PSNR	34.63	38.80	30.98	31.56
SSIM	0.9502	0.9795	0.9108	0.9182
Comput. time	0.25 s	16.4 s	77.6 s	4.79 s
Image 2				
	Naïve Bayer	*HQLI Bayer*	*Naïve Quad*	*HQLI Quad*
PSNR	30.31	34.00	26.96	28.85
SSIM	0.8430	0.9430	0.7577	0.8069
Comput. Time	0.25 s	17.2 s	80.1 s	5.30 s
Image 3				
	Naïve Bayer	*HQLI Bayer*	*Naïve Quad*	*HQLI Quad*
PSNR	31.98	35.07	28.61	29.36
SSIM	0.8941	0.9509	0.8280	0.8447
Comput. Time	0.19 s	18.8 s	65.5 s	5.40 s
Image 4				
	Naïve Bayer	*HQLI Bayer*	*Naïve Quad*	*HQLI Quad*
PSNR	29.88	33.05	26.65	29.03
SSIM	0.8145	0.9213	0.7230	0.8011
Comput. Time	0.27 s	28.9 s	87.3 s	4.80 s
Mean				
	Naïve Bayer	*HQLI Bayer*	*Naïve Quad*	*HQLI Quad*
PSNR	31.70	35.23	28.30	29.70
SSIM	0.8755	0.9487	0.8049	0.8427
Comput. Time	0.24	20.3	77.6	5.07

Table 1 : Comparison of the results for Naïve interpolation and HQLI for the 4 available images and both Bayer and Quad-Bayer configurations, along with the mean for each method.

Here we clearly see the improvement in Bayer results when using HQLI, with a 11.1% improvement in terms of PSNR and a 8.36% improvement in terms of SSIM meaning the obtained results is better both when looking at the colors (PSNR) and at the geometric similarity (SSIM). However, when we look at the computation time, HQLI takes a lot more time than Naïve interpolation to compute (x85), which might be a limitation in some situations.

Now, if we look at the Quad Bayer results, we can see a small prediction improvements between Naïve interpolation and HQLI but nothing really significant (+5% for PSNR and +4.6% for SSIM) ; but the real improvement is made on the computation time which is 15 times better for HQLI, quite the contrary than for Bayer !

Finally, and as expected for HQLI, the Quad Bayer results are far worser than Bayer ones as the method was not designed for Quad Bayer and we simply consider a coarser scale (two times smaller) to fall again in a Bayer configuration.

IV- Conclusion

We implemented another interpolation method, different from bilinear interpolation and initially developed for Bayer configuration. We adapted it to Quad Bayer configuration with a relative success even though it does not give as good results as the Bayer version. Many other, more complex, methods exists, especially those based on machines learning and more recently, deep learning such as PIPNet [4].

To make further improvements, we should turn to those more complex (and recent) methods. For Quad Bayer, we must look for specifically designed methods as passing from a classical Bayer to a Quad Bayer by simply downscaling the image (as it is done here in a way) will always give coarser results.

V- Bibliography

[1] Image analysis course SICOM 3A – Project, the presentation page (README) of this project : https://gricad-gitlab.univ-grenoble-alpes.fr/mullemat/sicom_image_analysis_project

[2] High-Quality Linear Interpolation for demosaicing of Bayer-patterned color images, Henrique S. Malvar, Li-wei He, and Ross Cutler, Microsoft Research : https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Demosaicing_ICASSP04.pdf

[3] Dataset from the National Gallery of Art, USA : https://github.com/NationalGalleryOfArt/opendata

[4] Beyond Joint Demosaicking and Denoising: An Image Processing Pipeline for a Pixel-bin Image Sensor, S M A Sharif, Rizman Ali Naqvi, Mithun Biswas, Rigel-IT, Bangladesh, Sejong University, South Korea : https://arxiv.org/pdf/2104.09398.pdf