Super-resolution Image Processing Pipeline

Hassan K Najjar[1]

***Abstract*—this project describes the steps to process a Bayer raw sensor output image which is noisy, undersampled, and blurred. The final output is a de-noised, de-blurred, and upsampled version of the input image. Some of the in-between steps include lens shading correction, color balancing, demosaicing, color correction, etc.**

Keywords—raw image ; deblurring ; denoising ; image deconvolution ;

I. Introduction

Processing a raw image requires many steps to correct artifacts stemming from the optics (e.g., lens shading), imaging sensor (e.g., defective pixels, noise, Bayer sampling) among others before the image can be properly displayed or printed. In this project, an ISP pipeline was developed in MATLAB with post-processing that includes de-convolution and interpolation (super-resolution) and detail enhancement. Parameters for each can be adjusted depending on the input image, for instance, to accomplish stronger/weaker de-noise, sharpening, or defective pixel correction.

II. Algorithms

A. Pipeline sequence

The pipeline is split in 3 stages: 1) Bayer domain processing, 2) linear RGB processing, 3) post-processing in gamma-encoded domain. In the first stage, we begin by applying defective pixel correction (DPC) via Median filtering. Followed by lens shading correction (LSC), black level correction (BLC), white balancing (WB), and then denoising via adaptive Wiener filtering. In the second stage, the image is first demosaicked with local polynomial approximation combined with intersection of confidence rule (LPA-ICI) implementation, color correction, white balancing (again), and finally, gamma encoding. In the last stage, a de-convolution-interpolation algorithm is applied followed by Median sharpening.

B. Bayer domain processing

The raw image contains 4 channels of red, green1, green2 and blue. Median filtering is applied on each channel separately to remove 'salt-and-pepper' noise, also known as defective pixels. Denoise is applied on each channel separately as well. The denoising algorithm is an adaptive Wiener filtering. The noise is modeled as additive white noise:

$$y_c(i,j) = x_c(i,j) + n_c(i,j) \qquad (1)$$

Where $x_c(i,j)$ is the ideal channel image, $n_c(i,j)$ is the noise modeled as a Gaussian $N(0,\sigma_c)$. The noise variance, σ_c, is calculated as the average of all the local estimated variances in the channel image.

By adaptively filtering noise, edges and high frequency details are better preserved than linearly filtering an image.

Lens shading correction is applied on the image by use of an a posteriori algorithm which uses a white image, and a black image:

$$y^{lsc}(i,j) = const * \frac{y(i,j) - b(i,j)}{w(i,j) - b(i,j)} \qquad (2)$$

Where $y^{lsc}(i,j)$ is the lens shading corrected image, $y(i.j)$ is the uncorrected image, $b(i,j)$ is the black image, and $w(i,j)$ is the white image, and *const* is a scalar.

Black level is subtracted followed by white balancing using the shades-of-gray algorithm.

C. Linear rgb processing

The LPA-ICI algorithm is used for color filter array interpolation (CFAI) only, though the implementation can be used for simultaneous denoising and demosaicking [1]. Based on preliminary testing, it's best to separate the operations.

Color balancing is repeated in the rgb space, as it's been found to give better results. Followed by a posteriori color correction via a 3x3 matrix calibrated from the Macbeth ColorChecker chart. Finally, the color space gamma coding (γ = 1.8) is applied on the image for display.

D. Post-processing in non-linear domain

The final stage involves deblurring the image using the measured camera PSF, followed by an interpolation to achieve superresolution. See section blah blah for a comparison of low resolution image with high resolution image achieved using this method.

III. Pipeline Implementation

In this section, the algorithms' implementation will be shown with detailed discussions on each implementation.

A. Defective pixel correction

Defective pixels in CMOS sensors appear in mainly 4 patterns: 1) single pixel, 2) 2-sharing, 3) 2x2 sharing pixel, and 4) 4 sharing pixel. The Median kernel size has to be adjusted according to which of those are known to be output by the sensor. Defective pixels can be black or white, and for this reason, DPC must be applied prior to black level subtraction.

Figure 1a shows an input raw image with defective pixels which are then corrected in Figure 1b. Various kernel sizes have been tested, and for this special case, the sensor has

[1] *EE368: Digital Image Processing, Stanford University.*

vertical-sharing defective pixels, and so a horizontal kernel of size 1x3 was used and successfully corrected for that.

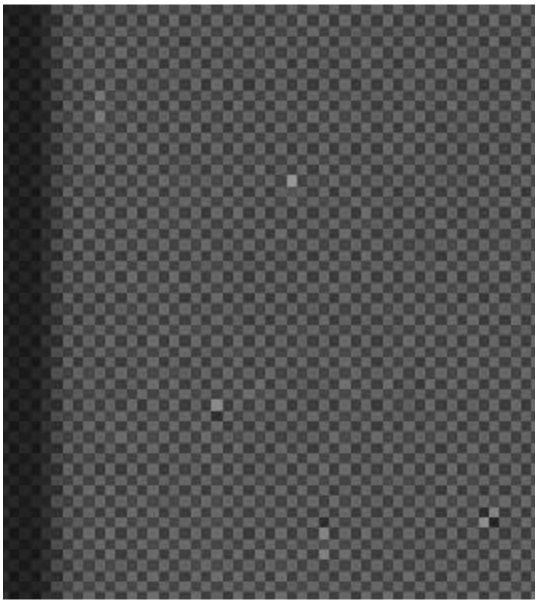

Figure 1a: Input raw image with black and white DP

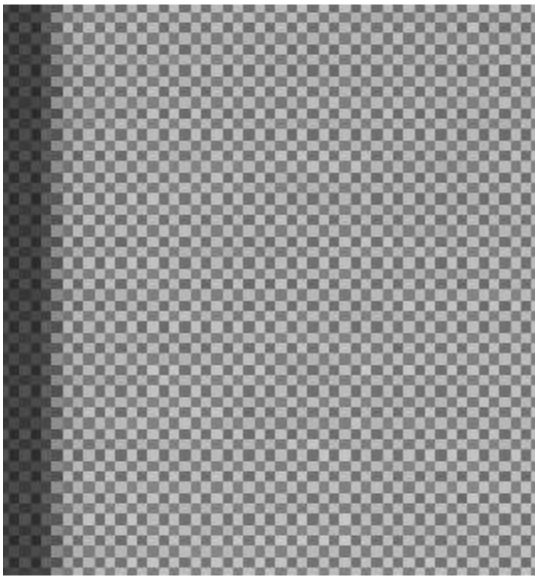

Figure 1b: Median filtered image 1x3 kernel

B. Bayer denoise

Using the LPA-ICI implementation, the denoise problem can be treated as an estimation problem jointly with demosaicking, however, this didn't give good results (see Appendix). For this reason, an adaptive filter was used in lieu of the joint method.

Figure 2 shows a comparison of the image before and after denoising for a single Bayer channel. Each channel is denoised with a small kernel—[3 3] for greens and [2 2] for red and blue—and then the channels are synthesized back together. Since green correlates with luminance more so than blue *and* red, and hence, has a higher correlation with visual noise, the kernel is chosen to be bigger.

Figure 2a: Denoised GreenR channel

Figure 2b: GreenR channel prior to denoise

C. CFAI (demosaicking)

Figure 3 shows the advantage of LPA-ICI for interpolation of Bayer images. The algorithm has the advantage of preserving thin lines and boundaries, by taking advantage of intersection of confidence rule (ICI). The ICI adds a constraint on the LPA for each pixel *p(i,j)* by defining the scale (window size) of LPA [2].

Figure 3a: gradient corrected linear interpolation

Figure 3b: LPA-ICI algorithm interpolation

D. *Super-resolution*

We begin with a low resolution image *LR(i,j)* and we'd like to obtain a high resolution image *HR(i,j)* from this single image. *LR(i,j)* is a blurred version of *I(i,j)*. Mathematically, we can say:

$$LR(i,j) = I(i,j) * h(i,j) + \varepsilon \qquad (3)$$

Where *h(i,j)* is the point spread function (PSF) of the camera, usually resembling a *sinc* function, ε is noise associated with (mostly) demosaicking.

The goal is to recover the un-blurred image *I(i,j)*. For simplicity, the noise, ε, is assumed to be filtered in the Bayer domain (Section II.B). Therefore, if the *h(i,j)* is known, the inverse function $h^{-1}(i,j)$ can be used to recover *I(i,j)*.

$$I(i,j) = h^{-1}(i,j) * LR(i,j) \qquad (4)$$

Now to obtain the image *HR(i,j)* we can apply any one of the methods in [3] [4]. As an initial test of concept, bicubic interpolation is used to get the higher resolution image *HR*. Figure 4 shows a comparison of *LR* and *HR* images.

Figure 4a: Post-gamma encoded image

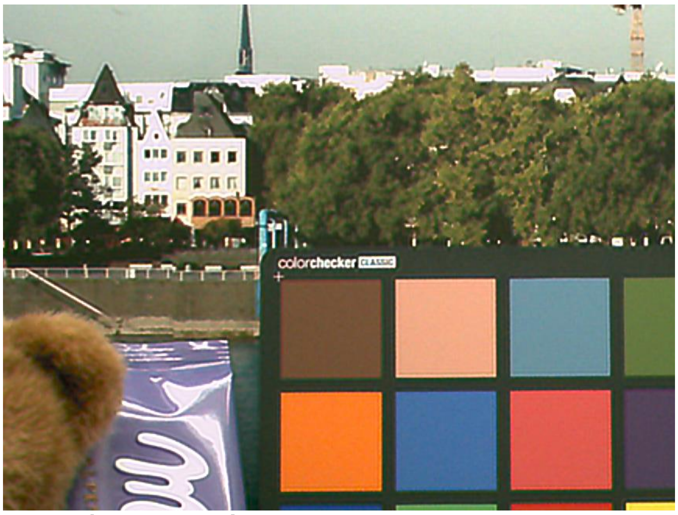

Figure 4b: Super-resolution image

IV. Discussion

In this section, some of the artifacts and problems associated with this implementation are going to be analyzed and plausible solutions are going to be discussed.

Halos around bright edges (seen in Figure 4a/4b), seem to be a product of the spatial denoising [5], and can be mitigated by use of better adaptive filtering.

The assumption made in Section II.D is that the *PSF* is the same for all channels and throughout the imaging plane, is in most cases, untrue and causes many artifacts especially on 80% image heights and above (close to the edge and corner of the image plane).

Figure 5a: Super-resolution image (edge)

Figure 5b: Post-gamma encoded image

REFERENCES

[1] K. Hirakawa and T. W. Parks, "Joint Demosaicing and Denoising," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 15, no. 8, 2006.

[2] D. K. V. Paliy, R. Bilcu, S. Alenius and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," Nokia Research Center, Tempere, 2007.

[3] D. Glasner, "Super-Resolution from a Single Image," in *The Weizmann Institute of Science*, Israel.

[4] H. He and W.-C. Siu, "Single Image Super-Resolution using Gaussian Process Regression," The Honk Kong Polytechnic University. [Online].

[5] T. Wilson and C. Sheppard, "The halo effect of image processing by spatial frequency filtering," *Optik - International journal for Light and Electron Optics.*