# In-camera Processing Pipeline

Due on 2 June, 2014

*Digital Photography - Spring 2014*

Exercise 4

This exercise contains several important theoretical concepts you will hear in class: denoising, demosaicing and tone mapping. You are asked to implement different algorithms for each concept. **General hint:** Similar to previous exercises, always normalize the images before displaying them (divide the image by the maximum value).

## 1 Denoising

Due to various factors, every electronic sensor adds some noise to the image. For example, Im1 is contaminated with the "thermal noise", which is caused by the random movements of the electrons and depends on the sensor's temperature and the exposure time.

1. Try to denoise Im1 with a Guassian filter.
   **Hint:** You can use the "fspecial" function to produce a Gaussian filter and the "imfilter" function to filter the image.

2. Gaussian filters have two parameters that should be tuned: the filter size and sigma (or filter variance). Try different parameters. What are the effects of the two parameters?

3. In the class you have learned that convolution in the image domain equals multiplication in the frequency domain. Try to perform Gaussian filtering in the frequency domain. What are the effects of the two parameters of the Guassian function in this case?
   **Hint:** You can use the functions "fft2" and "ifft2" to convert between the image domain and the frequency domain.

4. Compare the filtered image in questions 1 and 3 in the same figure. You should encounter some artifacts in the filtered image using the frequency domain method. Implement a solution to overcome this problem.

5. You have implemented the median filter in Exercise 3. Is it possible to implement the median filter in the frequency domain? Why?

## 2 Demosaicing

Most image sensors capture a single value per pixel instead of an RGB image with three color channels (three values in each pixel). A Color Filter Array (CFA) placed on top of the sensor determines the color measured in every pixel (red, green, or blue). The most common pattern for CFAs is the so called Bayer pattern illustrated in Figure 1. It contains twice as many green pixels as red or blue. The simplest demosaicing algorithm uses a bilinear interpolation to estimate the

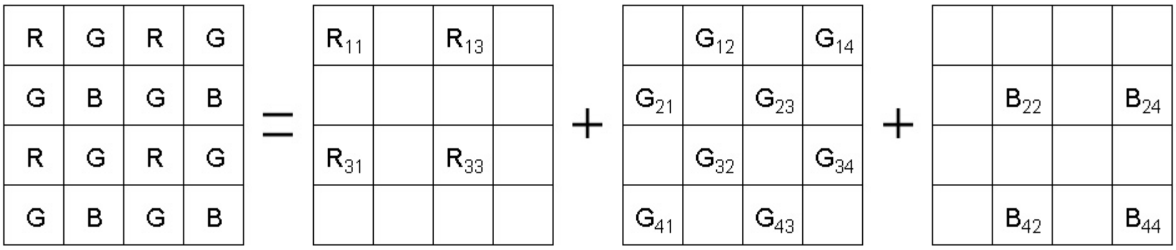


Figure 1: The Bayer color filter array and its subsampled channels.

two missing color components at each pixel. It uses the following filters:

$$F_{R,B} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad F_G = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{1}$$

for the red, blue, and green channels respectively. In this question you will demosaic Im2 to get the color image. Note that Im2 is only a one channel image with the pattern in the left side of Figure 1.

The demosaicing algorithm you will implement in this exercise is shown in Figure 2. The algorithm first separates the luminance and chrominance information of the image through a low-pass filter. Then demosaicing is performed on the chrominance image. The color pattern in the chrominance image is the same as in the Color Filter Array of Figure 1. We form the final result by adding the luminance information back into each channel of the chrominance image.

1. For Im2, compute the luminance image using a Gaussian low-pass filter with size 15*15, $\sigma = 3$. The chrominance image is obtained by subtracting the luminance image from the original mosaiced image.

2. Can you propose another method to obtain the chrominance information?

3. Demosaic the chrominance image with the given filters in Equation 1. Display the color image you computed.

   **Hint:** Apply the filters of Equation 1 to the corresponding color channels.

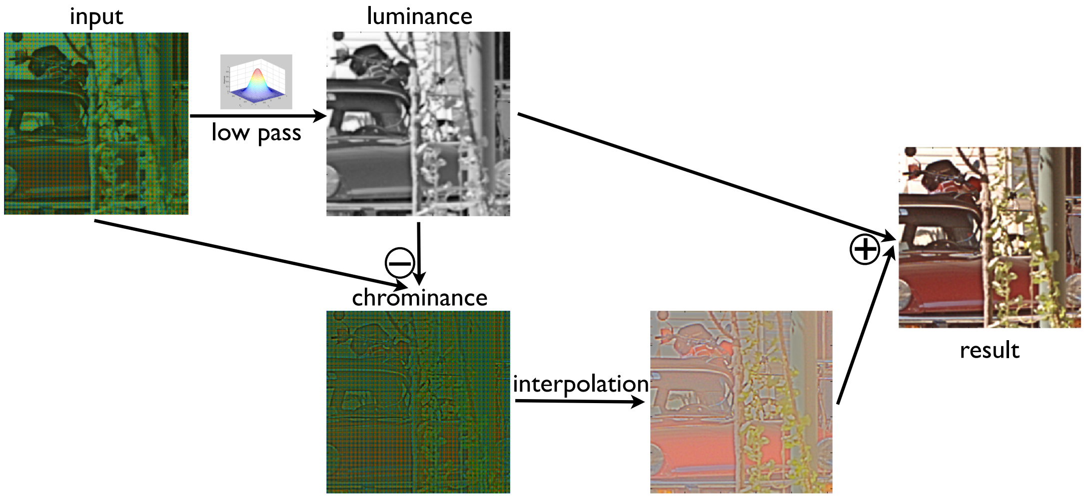


Figure 2: The pipeline of the demosaicing method

4. Add the luminance image back into each channel of the demosaiced chrominance image. display the result.

## 3 Tone Mapping

Tone mapping functions transform the brightness of the pixels so that the image looks visually pleasing and it fully utilizes the dynamic range of the target device. Two main classes of tone mapping algorithms, called global and local tone mapping, are usually used. In the following, you will implement a new demosaicing algorithm based on tone mapping. The low-pass filtering step in Figure 2 is replaced by a tone mapping step. The output of the tone mapping is also a luminance image.

1. Filter the input image with the following formula:

$$H(p) = I_{CFA}(p) * G_H + \frac{\overline{I_{CFA}}}{2} \tag{2}$$

   where $H(p)$ is the adaptation factor for pixel $p$, which will be used later. $I_{CFA}$ is the one channel input image (Im2), normalized between 0 and 1. $G_H$ is a two dimensional Gaussian filter:

$$G_H(x, y) = e^{-[(x^2+y^2)/2\sigma_H^2]} \tag{3}$$

   where $x \in [-3\sigma_H, 3\sigma_H]$ and $y \in [-3\sigma_H, 3\sigma_H]$. Use $\sigma_H = 3$. $\overline{I_{CFA}}$ represents the mean value of all the pixels in $I_{CFA}$. Display your result.

2. Perform a nonlinear tone mapping on the input image with the following equation:

$$I_{bip}(p) = (I_{CFA}(max) + H(p)) \frac{I_{CFA}(p)}{I_{CFA}(p) + H(p)} \tag{4}$$

Here $I_{CFA}(max)$ represents the maximum value in the input image. Display $I_{bip}$.

3. Perform steps 1 and 2 again using $I_{bip}$ as input image. Namely:

$$I_{ga}(p) = (I_{bip}(max) + A(p))\frac{I_{bip}(p)}{I_{bip}(p) + A(p)} \tag{5}$$

$$A(p) = I_{bip}(p) * G_A + \frac{\overline{I_{bip}}}{2} \tag{6}$$

$$G_A(x, y) = e^{-[(x^2+y^2)/2\sigma_A^2]} \tag{7}$$

Here again $x \in [-3\sigma_A, 3\sigma_A]$ and $y \in [-3\sigma_A, 3\sigma_A]$. We use $\sigma_A = 1.5$. Display your result $I_{ga}$.

4. $I_{ga}$ is the tone mapped image. Normalize $I_{ga}$ so that the minimum value is 0 and the maximum value is 1. Then apply the following low-pass filter on it to get the luminance image. Use the following filter:

$$F_{dem} = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{8}$$

$$L(p) = I_{ga} * F_{dem} \tag{9}$$

Here $L$ is your luminance image. Display it.

5. Use $L$ as your luminance image and apply the demosaicing method in question 2.

6. As a last step, apply the given function `post_processing` to your result. Display the final demosaiced image.

7. Compare the resulting demosaiced image with the one in question 2. Which one is better? Why do you think this is the case?

# 4 Remarks

The deadline for this exercise is **June 2nd, 2014, noon**. Please take a look at the following remarks before submitting your solution:

1. Put your code in one .m file. Name your file as **first_name_last_name_exe4.m**. Do not put " " (space) in your filename, use "_" instead.

2. Hint: Don't forget to put **clc; close all; clear all** commands at the beginning of your main code. This will clean the command window. Also all the previous figures will be closed and all the old variables will be deleted. This will help you avoid possible mistakes caused by previous variables.

3. Please put the correct titles on your plots and add **pause** command after displaying an image. This will help us to grade your submissions more quickly and accurately.
4. Please check your code before submitting, non-working codes will be graded **only** up to the erroneous line.