EE569 Digital Image Processing

**HOMEWORK #1**

**HOMEWORK – Introduction to Digital Image Processing**

**Issued: 13/01/2020 Due: 27/01/2020**

# Problem 1: Image Demosaicing and Histogram Manipulation (50%)

1. **Bilnear Demosaicing (10%)**

**Motivation**

The aim is to practice demosaicing method that can transform gray-scaled image to colorful RGB image. The bilinear transformation method is the simplest way to transform the gray-scaled image into the color space. The color value of three channels of the pixel depends on the processed image and its neighboring pixels. The value varies when the global position of the pixel changes. Hence, if-else argument should be designed to compute pixel value with different position in the image.

**Approach and procedure**

Before starting computing, memory is allocated to store the initial image and extended image and output image. To prevent different manipulation on the pixels on the edge, the original image is extended with mirror rule. Mirror rule means that the pixel values on the extended edge should be color channel is judged based on the position of pixel and compute the pixel value for different color channel. For different color, there is slightly difference. The computation method for green color is most complex. Before direct computation, whether the row position of pixel should be decided to ensure that neighbor position of other color pixel is right. The type of pixel value is character. Character can be not computed directly.

**Results**

The result is amazing. The obtained image is more colorful. No impulse pixels are obtained.

**Discussion**

The computation of the pixels depends on the position of the pixel. The averaging process is safe. There is overflow for the final computed pixel. The maximum pixel value is 255. The addition for C++ is 2 byte operator space. The final result for each pixel will be within 0-255 because the total summation will be divided by 2 or 4, so there is no overflow.

**Answers**

1. In Figure 1.1.1 and Figure 1.1.2, you are shown with the obtained dog picture and the original dog image. Compared with the original image, the obtained image is more colorful. The grass is much greener. More shapes can be viewed from scanning. 图片包含 小狗, 草, 动物, 哺乳动物

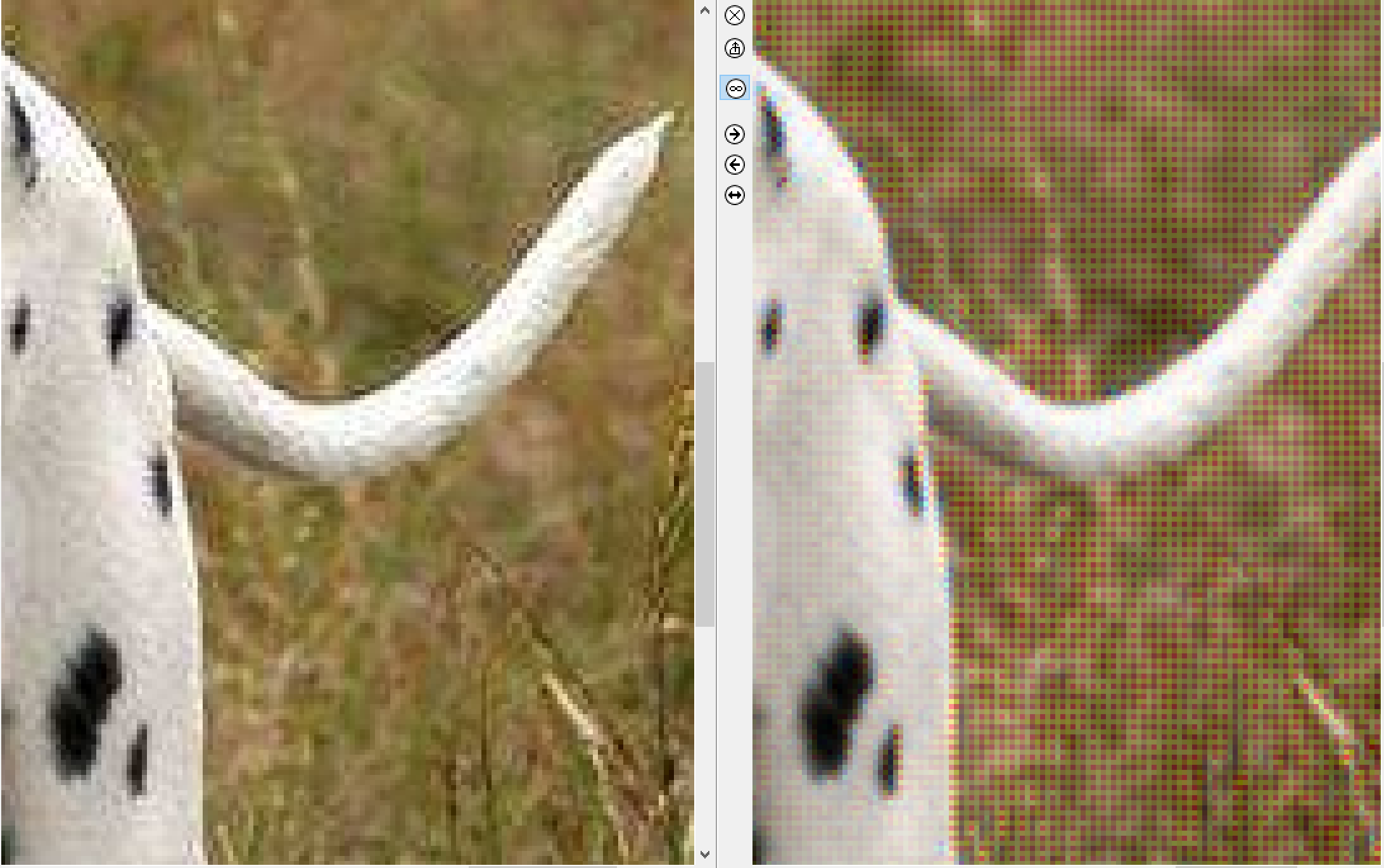
   描述已自动生成图片包含 草, 小狗, 户外, 田野

   描述已自动生成

**Figure 1.1.2 original image**

**Figure 1.1.1 obtained image**

1. Problem may be caused by the local average process for computation. The bilinear method ignores the global hue for the image, the color seems to be distributed equally from the obtained image.



**Figure 1.1.3 colorful artifact compared with the original image**

1. **Malvar-He-Cutler (MHC) Demosaicing (20%)**

**Motivations**

The MHC method to demosaic image is to add small computation compliment when we compute color value of three channels for the image.

**Approach and procedure**

1. Judge the primary color of pixel.
2. Compute difference between value of original color and neighbor color.
3. Compute estimated value for other color of the pixel.
4. Loop the above three steps for all pixels and get the image.

The computation procedure for difference value between original color and neighbor color depends on the color category of the pixel, so the specific function is chosen to compute the delta value for the pixel. There may be overflow for the unsigned char data type, so a compliment function needs to revise the computation result for each pixel

**Results**

**Discussion**

Overflow happens when the increment compliment estimation by the neighboring pixels is computed. Hence, the negative value may appear. The possible result can be negative for the pixel value. This cause the overflow for the unsigned char type because the unsigned char value can not be negative. The uncorrelated value causes several peak pixel with pure red, blue that is not correlated to the neighboring pixel value. The solution to eliminate this noise is to combine two methods above. When the negative value is got, average method can be chosen.

**Answers**

**Figure 2**: Averaged pepper.raw of 10 (left) and 100 (right) noisy images

**c). Non-Local Means (NLM) Filtering (10%)**

**Motivation**

**Approach and procedure**

* 1. compute the number of pixels corresponding to dedicated one gray-scaled value
  2. draw the histogram to describe the distribution with pixels numbers and gray-scaled value
  3. design the transfer function corresponding to the histogram
  4. apply transfer function to transfer current image pixels to expected image pixels
  5. write image result to the file

**Results**

**Discussion**

The first try to get RGB image does not consider the boundary problem, so as the figure shows that the boundary of the image is lacked.

**Answers**

# Problem 2: Image Denoising (50%)

1. **Baisc denoising methods (10%)**

**Motivation**

This is the simplest method among the problems to denoise image. Each pixel are obtained by averaging the neighboring pixels with the same weight. This weighs neighboring pixels equally by default. This assumption is defective when there are some discontinuous changes in the image.

**Approach and procedure**

**Results**

**Discussion**

**Answers**

1. **Bilateral Filtering (10%)**

**Motivation**

Bilateral filter compute weight based on Gaussian Probability distribution. It measures the pixels and neighboring pixels with Gaussian probability distribution model. When the distance between measured pixel and target pixel is large, the weight for the measured pixel will be small. The assumption is that the neighboring pixels has large influence on the target pixel value. This is also defective if there are different shapes or segments in the image.

**Approach and procedure**

More detailed steps are as follows:

1. define the hyper-parameter to estimate the models
2. assign the each neighboring pixel weight value.

**Results**

**Discussion**

**Answers**

1. **Non-Local Means(NLM) Filtering(10%)**

**Motivation**

The basic idea of NLM algorithm is to “build a pointwise estimate of the image where each pixel is obtained as a weighted average of pixels centered at regions that are similar to the region centered at the estimated pixel” (Kostadin, 2007). The size of window for the processing pixel to compute the distance from the neighboring pixel and the window to compute the gaussian coefficients is different. The implementation for the algorithm is difficult. The result is not obviously improved.

**Approach and procedure**

Each pixel is computed by weighted averaging neighboring pixels by Euclidean distance metric with the patch.

**Results**

**Discussion**

1. **Answers**
2. **Block matching and 3-D transform filter (10%)**

**Motivation**

**Approach and procedure**

**Results**

**Discussion**

**Answers**

**(e). Mixed noises in color image (10%)**

**Motivation**

**Approach and procedure**

**Results**

**Discussion**

**Answers**

# Appendix A

Environment configuration

1. Compilation environment
2. Coding IDE
3. Github code cloud store

Uses