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%)**
2. **Motivation**

The aim is to practice demosaicing method that can transform gray-scaled image to colorful RGB image.

1. **Approach and procedure**

The gray-scaled *dog.raw* should be extended to prevent boundary pixels from different operation. Before starting computing, memory is allocated to store the initial image, extended image and output image to prevent mistakes for frequently changing the initial image. 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.

1. **Results**

There are some impulse points in the pixel value.

1. **Discussion**

The computation of the pixels depends on the position of the pixel. The neighbor of pixel for different color is changed by whether the number is odd or even.

1. **Answers**

In Figure 1, you are shown the original color version of *dog.raw*. Your task is to experiment with the concept that multiple images of the same object which have been distorted by noise with the same characteristics may be used to restore the original.

1. **Malvar-He-Cutler (MHC) Demosaicing (20%)**
2. **Motivations**
3. **Approach and procedure**
4. Judge the primary color of pixel.
5. Compute difference between value of original color and neighbor color.
6. Compute estimated value for other color of the pixel.
7. 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

1. **Results**

The difference appears on the

A dog standing on grass

Description automatically generatedA brown and white dog standing on top of a grass covered field

Description automatically generated

Figure 1.1.1

Figure 1.1.2

1. **Discussion**
2. **Answers**

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

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

**1. Motivation**

**2. 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

**3. Results**

**4. 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.

**5. Answers**

# Problem 2: Image Denoising (50%)

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

This is the simplest method among the problems to denoise image. Pixels are averaged by the neighboring pixels with the same weight.

1. **Approach and procedure**
2. **Results**
3. **Discussion**
4. **Answers**
5. **Bilateral Filtering (10%)**
6. **Motivation**

Bilateral filter compute weight based on Gaussian Probability distribution. It measures the pixels and neighboring pixels with probability distribution model.

1. **Approach and procedure**
2. define the hyper-parameter to estimate the models
3. assign the each neighboring pixel weight value.

**3. Results**

**4. Discussion**

**5. Answers**

1. **Non-Local Means(NLM) Filtering(10%)**
2. **Block matching and 3-D (BM3D) transform filter (10%)**
3. **Mixed noises in color image (10%)**

Appendix A

Environment configuration

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

Uses