Computer Vision

Undergraduate Course

Chapter 8. Image Restoration (Practice)

Dongbo Min

Department of Computer Science and Engineering

Chungnam National University, Korea

Practice Lecture (1/2)

- **Perform Salt-and-Pepper Noise Removal**
  - Generate a noisy image with Salt-and-Pepper Noise (Chapter. 8.2.1)
  - Using the median filtering’s code provided in textbook (Chapter 8.3.2)
  - Using the outlier rejection method’s code provided in textbook (Chapter 8.3.4)

- **Perform Gaussian Noise Removal**
  - Generate a noisy image with AWGN (Additive White Gaussian Noise). (Chapter 8.2.2)
  - Using the simple average filtering’s code provided in textbook (Chapter 8.4.2)
  - Using the adaptive filtering’s code provided in textbook (Chapter 8.4.3)

Practice Lecture (2/2)

- **Perform Inverse Filtering**
  - Generate a blurry image using the code provided in textbook (Chapter 8.6)
  - Perform inverse filtering using the code provided in textbook (Chapter 8.6)
  - Generate a motion-blurry image using the code provided in textbook (Chapter 8.6.1)
  - Perform inverse filtering using the code provided in textbook (Chapter 8.6.1)

- **Perform Wiener Filtering**
  - Generate a blurry image using the code provided in textbook (Chapter 8.7)
  - Perform inverse filtering using the code provided in textbook (Chapter 8.7)

Principles for homework submission

- **MATLAB homework**
  - Submit all source codes (m file) for each (sub-) problem
  - If the codes do NOT work, then there will be a penalty.
  - The report for MATLAB homework should include the intermediate process, reason, and final results.

- **Report homework**
  - The report should include the intermediate process, reason, and final results.
  - The report homework should be done **by hand, NOT using any computer software**.

Example of Source Code

- For each problem, the source code should consist of **two functions**, as below.
  - In the 'homwork_main.m', the results should **appear** or be **saved** as below.

homework_main.m

```
in1 = imread('cameraman.tif');

out1 = function_example(in1);

imshow(out1); % or use imwrite(out1, 'output.png');
```

function_example.m

```
% Please make sure that there is a return variable to save an output.
% In the example below, 'y' is the return variable.
function y = function_example( im )

% Implement your code here.

end
```

숙제 제출 원칙

- **매트랩 숙제**
  - 각 세부문제 별로 모든 소스 코드를 제출
  - 만약 코드가 작동하지 않을 경우, 감점
  - 매트랩 숙제에 대한 보고서는 중간 결과, 이유, 최종 결과 등을 모두 포함하여 자세히 서술할 것

- **문제풀이 숙제**
  - 보고서는 중간 결과, 이유, 최종 결과 등을 모두 포함하여 자세히 서술할 것
  - 문제풀이 숙제는 반드시 손으로 해서 낼 것 (컴퓨터 SW를 사용하지 말 것!)

Practice Homework (1/2)

1. (MATLAB) 3x3와 5x5 median filter를 직접 구현하여 수행하여라, MATLAB 내장함수를 사용하지 마시오. (`ordfilt2` or `medfilt2`)
   (정렬(sorting)이 포함된 모든 함수는 직접 구현해야한다.)
   - 'twin.tif' 이미지를 grayscale image로 변환 (`'rgb2gray'`)
   - 위의 이미지에 MATLAB function `imnoise('input image', 'salt & pepper', D)`를 사용하여 salt-and-pepper noise를 더하여라.

   D는 noise density 즉, 얼마나 많은 픽셀이 salt and pepper noise에 의해 변하게 되는지를 의미한다.

   - 그리고 median filter를 적용하라.
   - 직접 구현한 함수와 내장함수를 비교하여라.

Practice Homework (1/2)

1. (MATLAB) Implement 3x3 and 5x5 median filter **by yourself**, NOT using MATLAB functions (`ordfilt2` or `medfilt2`)

   (All the functions including 'sorting' should be implemented **by yourself**.)

   – The image 'twin.tif' should be converted into grayscale image ('`rgb2gray`')

   – Add the salt-and-pepper noise to the grayscale image 'twin.tif' using the MATLAB function `imnoise('input image', 'salt & pepper', D)`

   D means the noise density indicating how many pixels are corrupted by salt and pepper noise.

   – Then, apply the median filter (you implement above) to the noisy 'twin.tif' image.

Practice Homework (2/2)

2. (MATLAB)Implement adaptive filtering in Chapter 8.4.3 **by yourself**.
   1. ‘`twin.tif`’ (grayscale image)에 Gaussian noise를 더하여 노이즈 이미지를 만들어라.

      `imnoise(t, 'gaussian', 0, 0.005)`
   2. 7×7 filtering mask를 사용하여 각 픽셀(x, y)의 $m(x, y)$과 ${\sigma_f}^2(x, y)$를 구하라. The filtering mask is an uniform average filter.
   3. 전체 이미지에 대하여 모든 ${\sigma_f}^2(x, y)$값의 평균을 가지고 n을 구하라.
   4. Perform the following equation about adaptive filtering

$$m_2(x, y) = m(x, y) + \frac{\max(0, {\sigma_f}^2 - n)}{\max({\sigma_f}^2, n)}(I_G(x, y) - m(x, y))$$

(1) $m(x, y)$ 와 $m_2(x, y)$를 비교하여라.

(2) 결과 $m_2(x, y)$ 와 MATLAB function `wiener2(input, [7,7])`의 결과를 비교하여라. It should be identical.

$m(x, y)$ and ${\sigma_f}^2$: the mean and variance of the mask at a pixel (x, y)
$I_G(x, y)$: an original input intensity at a pixel (x, y)
n: the variance of noise over the entire image

Practice Homework (2/2)

2. (MATLAB)Implement adaptive filtering in Chapter 8.4.3 **by yourself**.
   1. Generate a noisy image by adding the Gaussian noise to '`twin.tif`' (grayscale image)

      ```
      imnoise(t, 'gaussian', 0, 0.005)
      ```

   2. Obtain $m(x,y)$ and ${\sigma_f}^2(x,y)$ for each pixel (x,y) when 7×7 filtering mask is used. The filtering mask is an uniform average filter.
   3. Obtain n by taking the mean of all values of ${\sigma_f}^2(x,y)$ over the entire image.
   4. Perform the following equation about adaptive filtering

$$m_2(x,y) = m(x,y) + \frac{\max(0, {\sigma_f}^2 - n)}{\max({\sigma_f}^2, n)}\left(I_G(x,y) - m(x,y)\right)$$

(1) Compare two results $m(x,y)$ and $m_2(x,y)$.

(2) Compare your result $m_2(x,y)$ and the result by the MATLAB function `wiener2(input, [7,7])`. It should be identical.

$m(x,y)$ and ${\sigma_f}^2$: the mean and variance of the mask at a pixel (x,y)
$I_G(x,y)$: an original input intensity at a pixel (x,y)
n: the variance of noise over the entire image