KWANGWOON UNIVERSITY

Image Restoration Part2

2018.10.2

Seoungjun Oh(sjoh@kw.ac.kr)
Wooju Lee (krosea@kw.ac.kr)

Multimedia LAB
VIA-Multimedia Center, Kwangwoon University

Contents

- Image Restoration
  - Spatial Filtering
    - 2-D Contra-harmonic Mean Filtering
    - 2-D alpha-trimmed mean filtering
- Example
- Assignment

Image Restoration

- Image Restoration
  - Operation of taking a corrupted/noisy image and estimating the clean original image

Original image

Noisy image
PSNR : 12.73 dB

Restoration

Estimated image
PSNR : 30.21 dB

Image Restoration

- Salt noise

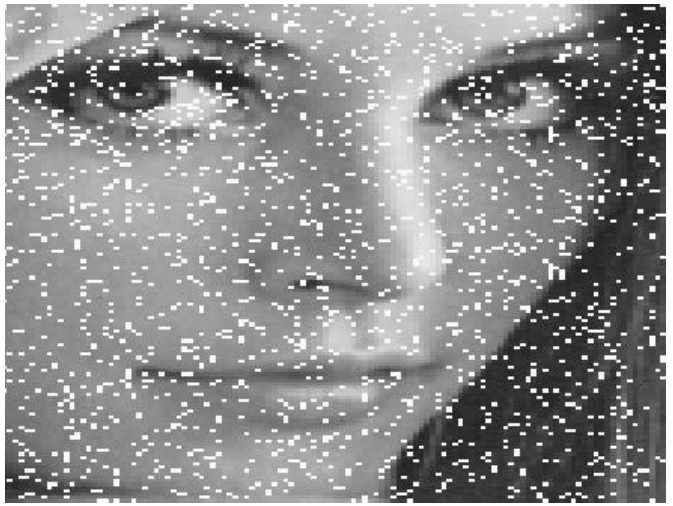

Image Restoration

- Pepper noise

Image Restoration

- Spatial filtering
  - Contra-harmonic mean filtering

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

where
- S_{xy} : set of coordinates in a subimage window of size $m \times n$ centered at point (x, y)
- Q : order of the filter
- $\hat{f}(x,y)$: restored image
- $g(x,y)$: corrupted image

Image Restoration

- Spatial filtering
  - Contra-harmonic mean filtering
    - 3x3 filtering
    - Q = 2

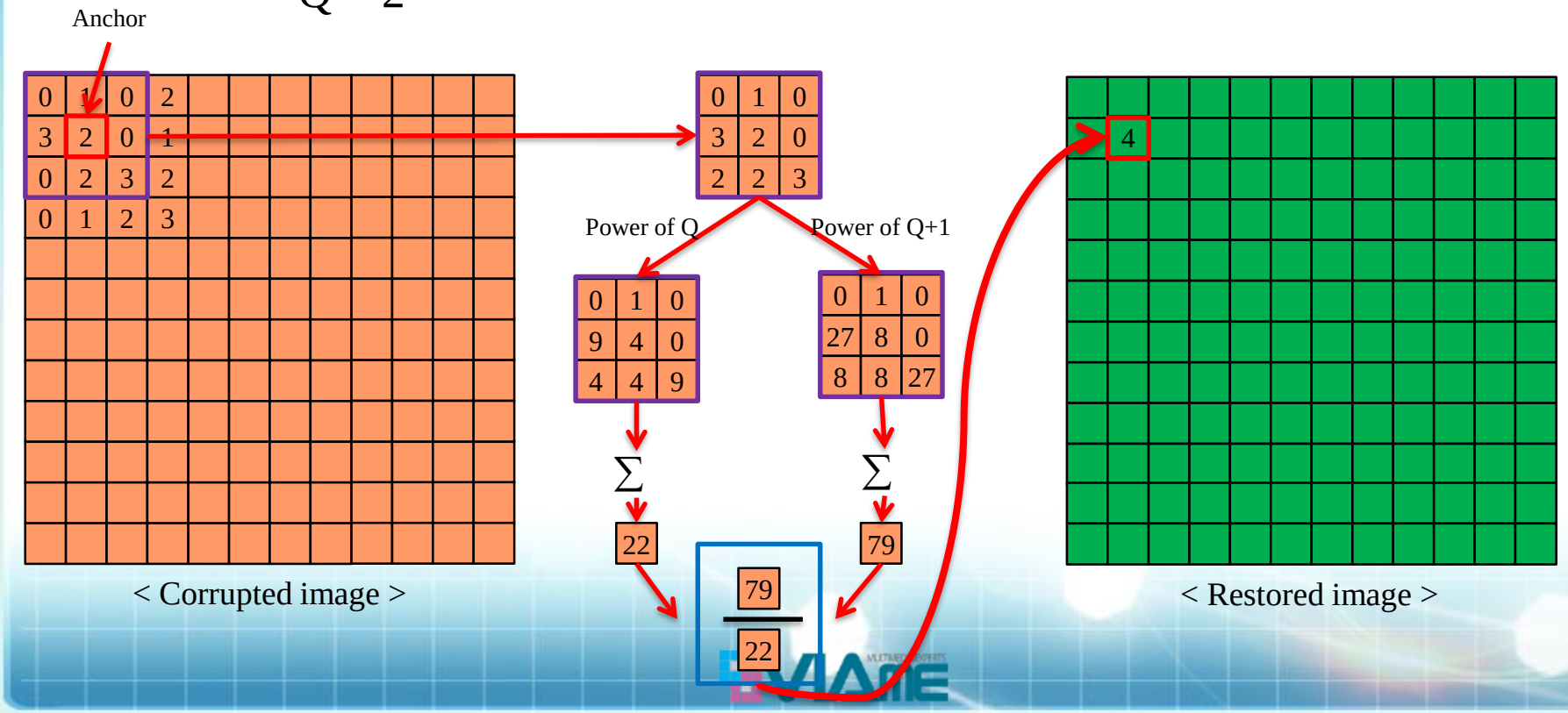

Image Restoration

- Gaussian and salt&pepper noise

Original image

Corrupted by Gaussian noise
PSNR : 19.29 dB

Additionally corrupted by salt&pepper noise
PSNR : 14.87 dB

Image Restoration

- Spatial filtering
  - 2-D alpha-trimmed mean filtering
    - Delete the d/2 lowest and the d/2 highest gray-level values

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

where
- S_{xy} : set of coordinates in a subimage window of size $m \times n$ centered at point (x, y)
- $g_r(x,y)$: remaining $mn-d$ pixels of $g(s,t)$ in the neighborhood S_{xy}
  d/2 lowest and d/2 hightest values are deleted
- $\hat{f}(x,y)$: restored image

Image Restoration

- Spatial filtering
  - 2-D alpha-trimmed mean filtering
    - 3x3 filtering($m = 3$, $n = 3$)
    - $d = 4$

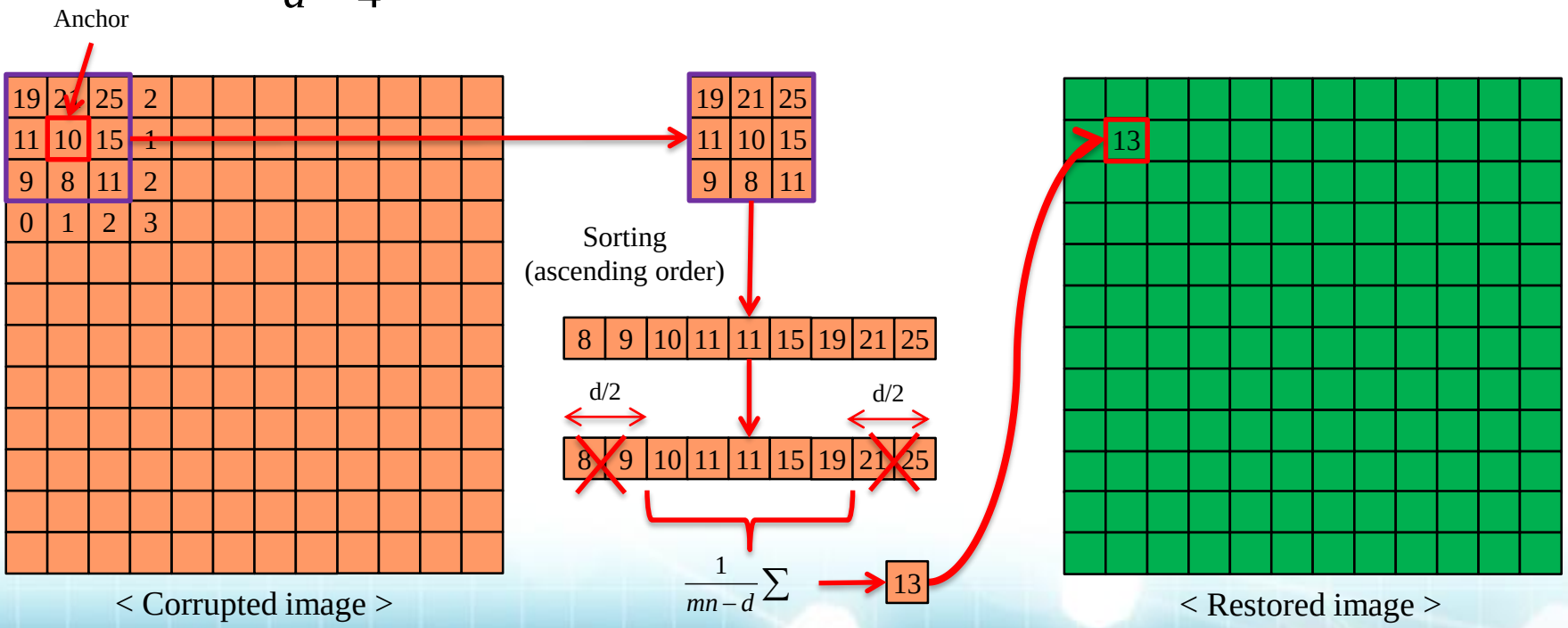

Image Restoration

❖ Example

```
#include <stdio.h>
#include <stdlib.h>             //  header file
#include <math.h>

#define WIDTH 512               //  image size
#define HEIGHT 512
typedef unsigned char BYTE;

unsigned char** MemAlloc_2D(int width, int height);                 // 2D memory allocation
void MemFree_2D(unsigned char** arr, int height);                   // 2D memory free
void FileRead(char* filename, unsigned char** img_in, int width, int height);   //  read data from a file
void FileWrite(char* filename, unsigned char** img_out, int width, int height); //  write data to a file

float GetPSNR(unsigned char** img_ori, unsigned char** img_dist, int width, int height);        //  PSNR calculation
void ContraHarmonicMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, float Q, int width, int height);

int main()
{
    BYTE **img_ori, **img_in_S, **img_in_P, **img_res;
    float Q;
    char filename_out[100];

    img_ori = MemAlloc_2D(WIDTH, HEIGHT);                           //  2D input memory allocation
    img_in_S = MemAlloc_2D(WIDTH, HEIGHT);
    img_in_P = MemAlloc_2D(WIDTH, HEIGHT);
    FileRead("Lena(512x512).raw", img_ori, WIDTH, HEIGHT);                  //  input image read
    FileRead("[Salt_Noise]Lena(512x512).raw", img_in_S, WIDTH, HEIGHT);
    FileRead("[Pepper_Noise]Lena(512x512).raw", img_in_P, WIDTH, HEIGHT);

    img_res = MemAlloc_2D(WIDTH, HEIGHT);                           //  2D output memory allocation

    ////////////// Contra-harmonic mean filtering on salt noise ////////////////////////////////////
    printf("Salt noise PSNR : %.2f dB₩n₩n", GetPSNR(img_ori, img_in_S, WIDTH, HEIGHT));

    for(Q = -2 ; Q <= 2 ; Q++){
        ContraHarmonicMeanFilter(img_in_S, img_res, 3, Q, WIDTH, HEIGHT);       // contra-harmonic mean filtering
        printf("3x3 contra-harmonic mean filter(Q = %.1f) PSNR : %.2f dB₩n", Q, GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
        sprintf(filename_out, "[Salt_CHMF_3x3_%.1f]Lena(512x512).raw", Q);
        FileWrite(filename_out, img_res, WIDTH, HEIGHT);                        // output file write
    }
    printf("₩n₩n");

    ////////////// Contra-harmonic mean filtering on pepper noise ////////////////////////////////////
    printf("Pepper noise PSNR : %.2f dB₩n₩n", GetPSNR(img_ori, img_in_P, WIDTH, HEIGHT));

    for(Q = -2 ; Q <= 2 ; Q++){
        ContraHarmonicMeanFilter(img_in_P, img_res, 3, Q, WIDTH, HEIGHT);       // contra-harmonic mean filtering
        printf("3x3 contra-harmonic mean filter(Q = %.1f) PSNR : %.2f dB₩n", Q, GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
        sprintf(filename_out, "[Pepper_CHMF_3x3_%.1f]Lena(512x512).raw", Q);
        FileWrite(filename_out, img_res, WIDTH, HEIGHT);                        // output file write
    }
    printf("₩n₩n");

    MemFree_2D(img_ori, HEIGHT);
    MemFree_2D(img_in_S, HEIGHT);           // 2D memory free
    MemFree_2D(img_in_P, HEIGHT);
    MemFree_2D(img_res, HEIGHT);

    return 0;
}
```

Image Restoration

❖ Example

```
void ContraHarmonicMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, float Q, int width, int height)
{

}
```

?

Image Restoration

KWANGWOON UNIVERSITY

❖ Example

```
#include <stdio.h>              //  header file
#include <stdlib.h>
#include <math.h>

#define WIDTH 512               //  image size
#define HEIGHT 512
typedef unsigned char BYTE;

unsigned char** MemAlloc_2D(int width, int height);                                                 // 2D memory allocation
void MemFree_2D(unsigned char** arr, int height);                                                   // 2D memory free
void FileRead(char* filename, unsigned char** img_in, int width, int height);                       //  read data from a file
void FileWrite(char* filename, unsigned char** img_out, int width, int height);                     //  write data to a file

float GetPSNR(unsigned char** img_ori, unsigned char** img_dist, int width, int height);            //  PSNR calculation

void ArithmeticMeanFilter
(unsigned char** img_in, unsigned char** img_out, int mask_size, int width, int height);            // arithmetic mean filter

void MedianMeanFilter
(unsigned char** img_in, unsigned char** img_out, int mask_size, int width, int height);            // median mean filter

void ContraHarmonicMeanFilter
(unsigned char** img_in, unsigned char** img_out, int mask_size, float Q, int width, int height);   // contra-harmonic mean filter

void AlphaTrimmedMeanFilter
(unsigned char** img_in, unsigned char** img_out, int mask_size, int d, int width, int height);     // alpha-trimmed mean filter

int main( )
{
    BYTE **img_ori, **img_noise, **img_res;
    float Q;
    int d;
    char filename_out[100];
    img_ori = MemAlloc_2D(WIDTH, HEIGHT);                                   //  2D input memory allocation
    img_noise = MemAlloc_2D(WIDTH, HEIGHT);
    FileRead("Lena(512x512).raw", img_ori, WIDTH, HEIGHT);                  //  input image read
    FileRead("[Noise]Lena(512x512).raw", img_noise, WIDTH, HEIGHT);

    img_res = MemAlloc_2D(WIDTH, HEIGHT);                                   //  2D output memory allocation

    printf("PSNR : %.2f dB₩n₩n", GetPSNR(img_ori, img_noise, WIDTH, HEIGHT));

    ArithmeticMeanFilter(img_noise, img_res, 5, WIDTH, HEIGHT);             //  5x5 arithmetic mean filtering
    printf("5x5 arithmetic mean filter PSNR : %.2f dB₩n₩n₩n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
    sprintf(filename_out, "[AMF_5x5]Lena(512x512).raw");
    FileWrite(filename_out, img_res, WIDTH, HEIGHT);
```

Image Restoration

❖ Example

```
    MedianMeanFilter(img_noise, img_res, 5, WIDTH, HEIGHT);                    //  5x5 median mean filtering
    printf("5x5 median mean filter PSNR : %.2f dB₩n₩n₩n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
    sprintf(filename_out, "[MMF_5x5]Lena(512x512).raw");
    FileWrite(filename_out, img_res, WIDTH, HEIGHT);

    for(Q = -2 ; Q <= 2 ; Q++){                                                //  5x5 contra-harmonic mean filtering
        ContraHarmonicMeanFilter(img_noise, img_res, 5, Q, WIDTH, HEIGHT);
        printf("5x5 contra-harmonic mean filter(Q = %.1f) PSNR : %.2f dB₩n", Q, GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
        sprintf(filename_out, "[CHMF_5x5_%.1f]Lena(512x512).raw", Q);
        FileWrite(filename_out, img_res, WIDTH, HEIGHT);                       // output file write
    }
    printf("₩n₩n");

    for(d = 0 ; d <= 24 ; d += 2){                                             //  5x5 alpha-trimmed mean filtering
        AlphaTrimmedMeanFilter(img_noise, img_res, 5, d, WIDTH, HEIGHT);
        printf("5x5 alpha-trimmed mean filter(d = %d) PSNR : %.2f dB₩n", d, GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
        sprintf(filename_out, "[ATMF_5x5_%d]Lena(512x512).raw", d);
        FileWrite(filename_out, img_res, WIDTH, HEIGHT);
    }

    MemFree_2D(img_ori, HEIGHT);            // 2D memory free
    MemFree_2D(img_noise, HEIGHT);
    MemFree_2D(img_res, HEIGHT);

    return 0;
}
```

Image Restoration

❖ Example

```
void AlphaTrimmedMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, int d, int width, int height)
{









}
```

?

KWANGWOON UNIVERSITY

Assignment

VIAME MULTIMEDIA EXPERTS

Assignment

- Example code completion
  - Function implementation
    - void ContraHarmonicMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, float Q, int width, int height)
      - 2-D contra-harmonic mean filtering
        - img_in : corrupted image
        - img_out : restored image to be written
        - mask_size : filter mask size
        - Q : the order of the filter
        - width : image width
        - height : image height
      - Apply boundary padding before filtering

Assignment

- Example code completion

```
Salt noise PSNR : 15.54 dB

3x3 contra-harmonic mean filter(Q = -2.0) PSNR : 29.63 dB
3x3 contra-harmonic mean filter(Q = -1.0) PSNR : 27.99 dB
3x3 contra-harmonic mean filter(Q = 0.0) PSNR : 22.14 dB
3x3 contra-harmonic mean filter(Q = 1.0) PSNR : 15.67 dB
3x3 contra-harmonic mean filter(Q = 2.0) PSNR : 11.66 dB


Pepper noise PSNR : 15.95 dB

3x3 contra-harmonic mean filter(Q = -2.0) PSNR : 8.16 dB
3x3 contra-harmonic mean filter(Q = -1.0) PSNR : 9.66 dB
3x3 contra-harmonic mean filter(Q = 0.0) PSNR : 22.50 dB
3x3 contra-harmonic mean filter(Q = 1.0) PSNR : 32.23 dB
3x3 contra-harmonic mean filter(Q = 2.0) PSNR : 30.31 dB
```

Assignment

- Example code completion
  - Function implementation
    - void AlphaTrimmedMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, int d, int width, int height)
      - 2-D alpha-trimmed mean filtering
        - img_in : corrupted image
        - img_out : restored image to be written
        - mask_size : filter mask size
        - d : the number of the lowest and highest pixel values to be deleted
        - width : image width
        - height : image height
      - Apply boundary padding before filtering

Assignment

- Example code completion

```
PSNR : 14.87 dB

5x5 arithmetic mean filter PSNR : 23.97 dB


5x5 median mean filter PSNR : 26.44 dB


5x5 contra-harmonic mean filter(Q = -2.0) PSNR : 13.77 dB
5x5 contra-harmonic mean filter(Q = -1.0) PSNR : 17.54 dB
5x5 contra-harmonic mean filter(Q = 0.0) PSNR : 23.97 dB
5x5 contra-harmonic mean filter(Q = 1.0) PSNR : 18.57 dB
5x5 contra-harmonic mean filter(Q = 2.0) PSNR : 14.63 dB


5x5 alpha-trimmed mean filter(d = 0) PSNR : 23.97 dB
5x5 alpha-trimmed mean filter(d = 2) PSNR : 24.41 dB
5x5 alpha-trimmed mean filter(d = 4) PSNR : 24.93 dB
5x5 alpha-trimmed mean filter(d = 6) PSNR : 25.46 dB
5x5 alpha-trimmed mean filter(d = 8) PSNR : 25.93 dB
5x5 alpha-trimmed mean filter(d = 10) PSNR : 26.28 dB
5x5 alpha-trimmed mean filter(d = 12) PSNR : 26.51 dB
5x5 alpha-trimmed mean filter(d = 14) PSNR : 26.63 dB
5x5 alpha-trimmed mean filter(d = 16) PSNR : 26.69 dB
5x5 alpha-trimmed mean filter(d = 18) PSNR : 26.70 dB
5x5 alpha-trimmed mean filter(d = 20) PSNR : 26.67 dB
5x5 alpha-trimmed mean filter(d = 22) PSNR : 26.59 dB
5x5 alpha-trimmed mean filter(d = 24) PSNR : 26.44 dB
```