

Image Restoration

2018.09.27

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

Multimedia LAB

VIA-Multimedia Center, Kwangwoon University

Contents

❖ Image Restoration

- Additive Gaussian Noise
- Salt&Pepper Noise
- Spatial Filtering
 - 2-D Arithmetic Mean Filtering
 - 2-D Median Mean Filtering

❖ Example

❖ Assignment

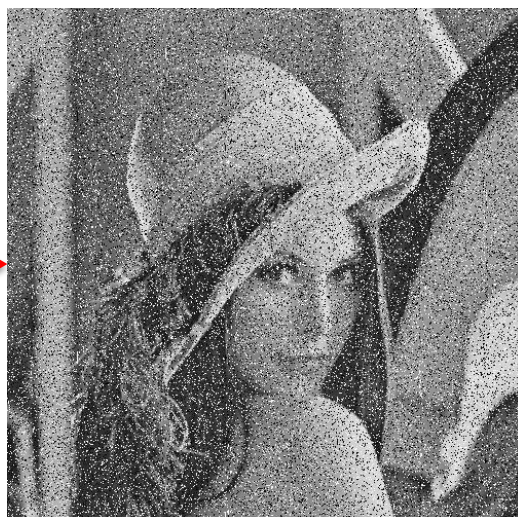
Image Restoration

❖ Image Restoration

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

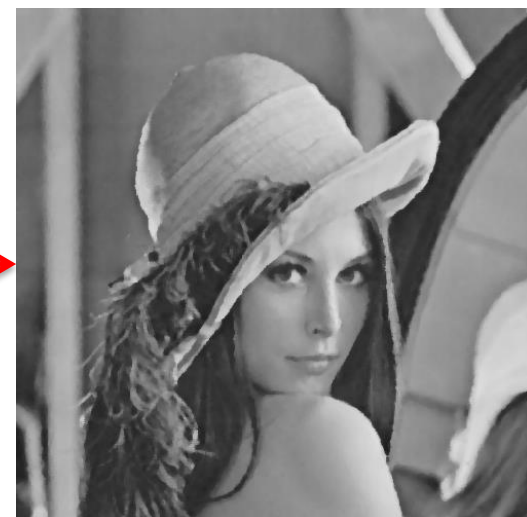


Original image



Noisy image
PSNR : 12.73 dB

Restoration



Estimated image
PSNR : 30.21 dB

Image Restoration

❖ Additive Gaussian noise



Image Restoration

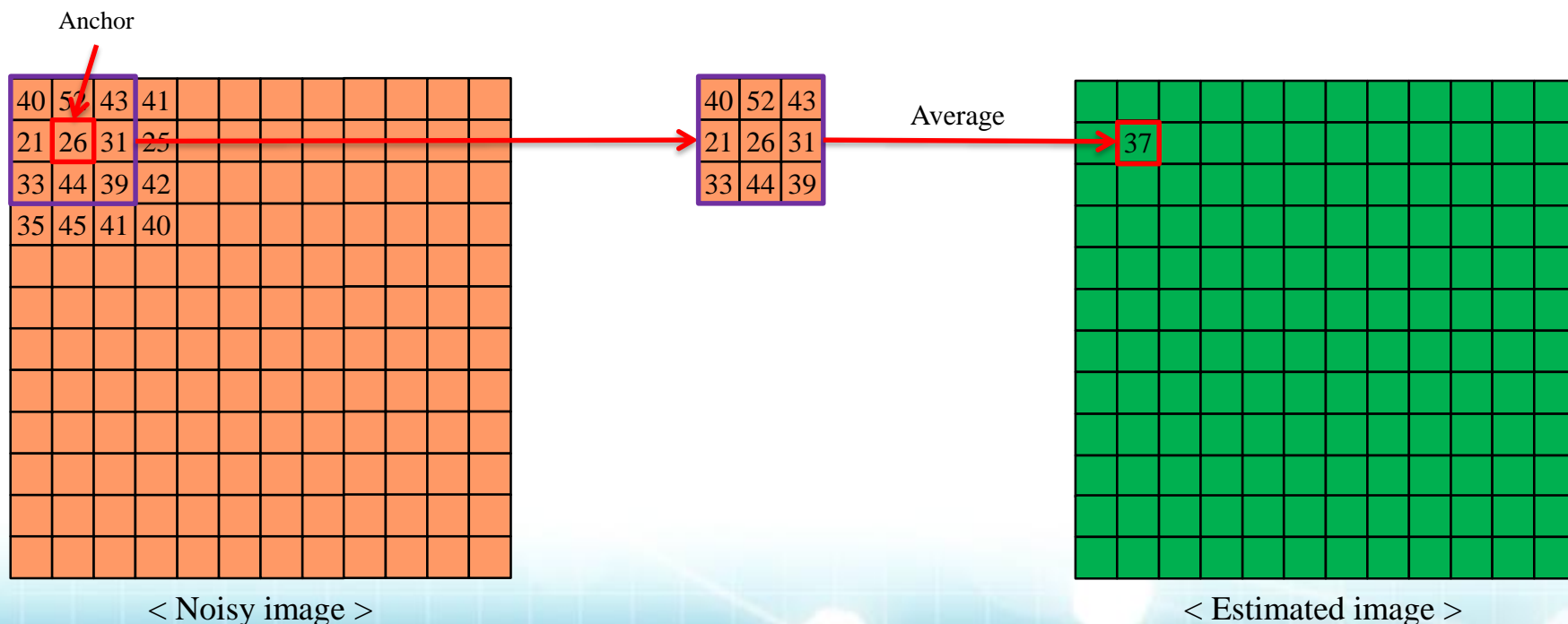
❖ Salt&Pepper noise



Image Restoration

❖ Spatial filtering

- 2-D arithmetic mean filtering
 - 3x3 filtering



- 2-D median mean filtering
 - 3x3 filtering



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

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

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

int main()
{
    BYTE **img_ori, **img_in_Gaus, **img_in_SP, **img_res;

    img_ori = MemAlloc_2D(WIDTH, HEIGHT);           // 2D input memory allocation
    img_in_Gaus = MemAlloc_2D(WIDTH, HEIGHT);
    img_in_SP = MemAlloc_2D(WIDTH, HEIGHT);
    FileRead("Lena(512x512).raw", img_ori, WIDTH, HEIGHT); // input image read
    FileRead("[Gaussian_Noise]Lena(512x512).raw", img_in_Gaus, WIDTH, HEIGHT);
    FileRead("[Salt&Pepper_Noise]Lena(512x512).raw", img_in_SP, WIDTH, HEIGHT);

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

    /////////////// arithmetic mean filtering (3x3, 5x5, 7x7, 9x9) on Gaussian noise ///////////////////
    printf("Gaussian noise PSNR : %.2f dB\n", GetPSNR(img_ori, img_in_Gaus, WIDTH, HEIGHT));

    ArithmeticMeanFilter(img_in_Gaus, img_res, 3, WIDTH, HEIGHT); FileWrite("[AMF_3x3_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
    printf("3x3 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

    ArithmeticMeanFilter(img_in_Gaus, img_res, 5, WIDTH, HEIGHT); FileWrite("[AMF_5x5_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
    printf("5x5 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

    ArithmeticMeanFilter(img_in_Gaus, img_res, 7, WIDTH, HEIGHT); FileWrite("[AMF_7x7_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
    printf("7x7 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

    ArithmeticMeanFilter(img_in_Gaus, img_res, 9, WIDTH, HEIGHT); FileWrite("[AMF_9x9_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
    printf("9x9 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));
}
```


Image Restoration

❖ Example

```
////////// median mean filtering (3x3, 5x5, 7x7, 9x9) on Gaussian noise ////////////
MedianMeanFilter(img_in_Gaus, img_res, 3, WIDTH, HEIGHT); FileWrite("[MM_3x3_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("3x3 median mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

MedianMeanFilter(img_in_Gaus, img_res, 5, WIDTH, HEIGHT); FileWrite("[MM_5x5_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("5x5 median mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

MedianMeanFilter(img_in_Gaus, img_res, 7, WIDTH, HEIGHT); FileWrite("[MM_7x7_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("7x7 median mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

MedianMeanFilter(img_in_Gaus, img_res, 9, WIDTH, HEIGHT); FileWrite("[MM_9x9_GN]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("9x9 median mean filter PSNR : %.2f dB\n\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

////////// arithmetic mean filtering (3x3, 5x5, 7x7, 9x9) on salt&pepper noise ////////////
printf("Salt&Pepper noise PSNR : %.2f dB\n\n", GetPSNR(img_ori, img_in_SP, WIDTH, HEIGHT));

ArithmeticMeanFilter(img_in_SP, img_res, 3, HEIGHT, WIDTH); FileWrite("[AMF_3x3_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("3x3 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

ArithmeticMeanFilter(img_in_SP, img_res, 5, HEIGHT, WIDTH); FileWrite("[AMF_5x5_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("5x5 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

ArithmeticMeanFilter(img_in_SP, img_res, 7, HEIGHT, WIDTH); FileWrite("[AMF_7x7_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("7x7 arithmetic mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

ArithmeticMeanFilter(img_in_SP, img_res, 9, HEIGHT, WIDTH); FileWrite("[AMF_9x9_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("9x9 arithmetic mean filter PSNR : %.2f dB\n\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

////////// median mean filtering (3x3, 5x5, 7x7, 9x9) on salt&pepper noise ////////////
MedianMeanFilter(img_in_SP, img_res, 3, WIDTH, HEIGHT); FileWrite("[MM_3x3_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("3x3 median mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

MedianMeanFilter(img_in_SP, img_res, 5, WIDTH, HEIGHT); FileWrite("[MM_5x5_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("5x5 median mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

MedianMeanFilter(img_in_SP, img_res, 7, WIDTH, HEIGHT); FileWrite("[MM_7x7_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("7x7 median mean filter PSNR : %.2f dB\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

MedianMeanFilter(img_in_SP, img_res, 9, WIDTH, HEIGHT); FileWrite("[MM_9x9_SP]Lena(512x512).raw", img_res, WIDTH, HEIGHT);
printf("9x9 median mean filter PSNR : %.2f dB\n\n", GetPSNR(img_ori, img_res, WIDTH, HEIGHT));

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

return 0;
}
```

Image Restoration

❖ Example

```
void ArithmeticMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, int width, int height)
{
    int padding = mask_size/2;
    float temp;
    int h,w,i,j;

    BYTE** img_in_padding = MemAlloc_2D(width + 2*padding, height + 2*padding);    // 2D memory allocation

    for(i = 0 ; i < height ; i++){
        for(j = 0 ; j < width ; j++){
            img_in_padding[i+padding][j+padding] = img_in[i][j];
        }
    }

    ?

    MemFree_2D(img_in_padding, height + 2*padding);
}
```

```
void MedianMeanFilter(BYTE** img_in, BYTE** img_out, int mask_size, int width, int height)
{
    int padding = mask_size/2;
    int idx;
    int h,w,i,j,n;

    float temp;
    BYTE** img_in_padding;

    float* data = (float*)malloc(sizeof(float)*mask_size*mask_size);

    img_in_padding = MemAlloc_2D(width + 2*padding, height + 2*padding);

    for(i = 0 ; i < height ; i++){
        for(j = 0 ; j < width ; j++){
            img_in_padding[i+padding][j+padding] = img_in[i][j];
        }
    }

    ?

    free(data);
    MemFree_2D(img_in_padding, height + 2*padding);    // memory free
}
```

Assignment

Assignment

❖ Example code completion

● Function implementation

- void ArithmeticMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, int width, int height)
 - 2-D arithmetic mean filtering
 - img_in : noisy image
 - img_out : estimated image to be written
 - mask_size : filter mask size
 - width : image width
 - height : image height

Assignment

❖ Example code completion

● Function implementation

- `void MedianMeanFilter(unsigned char** img_in, unsigned char** img_out, int mask_size, int width, int height)`
 - 2-D median mean filtering
 - `img_in` : noisy image
 - `img_out` : estimated image to be written
 - `mask_size` : filter mask size
 - `width` : image width
 - `height` : image height