



北京交通大学

图像处理与机器学习 基础实验

实验四 均值/高斯 低通滤波







实验四 均值/高斯 低通滤波



实验数据

-- 源图像：被噪声污染的图像

-- 格式：*.raw

-  Pulse_Noised_Image.raw
-  Gaussian_Noised_Image.raw
-  Multiplicative_Noised_Image.raw
-  Additive_Noised_Image.raw





实验四 均值/高斯 低通滤波

无标题 - MFCApplication1

文件(F) 编辑(E) 视图(V) 帮助(H) 打开bmp图像 直方图绘制 直方图均衡 均值滤波 高斯滤波 中值滤波 sobel算子边缘提取 形态学处理





实验四 均值/高斯 低通滤波

资源管理器

MFCApplication1View.h

MFCApplication1View.cpp

Application1

CMFCApplic

//均值滤波

int MeanFilterFlag;

BYTE* meanFilterImg; // 存储均值滤波后的图像

void meanFilter(BYTE*, int, int, BYTE*); // 均值滤波器

int convolution(int* operatr, BYTE* block); //卷积函数

//高斯滤波

int gaussianFlag;

BYTE* gaussianImg; // 存储高斯滤波后的图像

void gaussian(BYTE*, int, int, BYTE*); // 高斯滤波器



空间域滤波

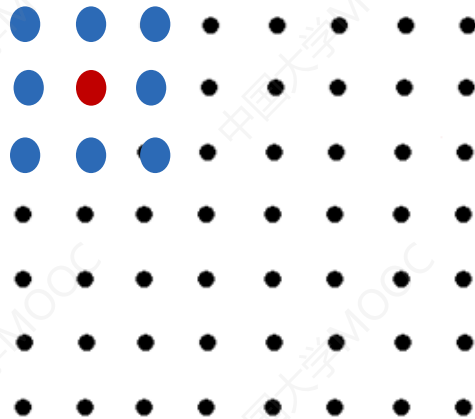
$$y(j,i) = \sum_m \sum_n h(m,n) x(j+m, i+n)$$

均值滤波

$$h(m,n) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \frac{1}{9}$$

高斯滤波

$$h(m,n) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \times \frac{1}{16}$$





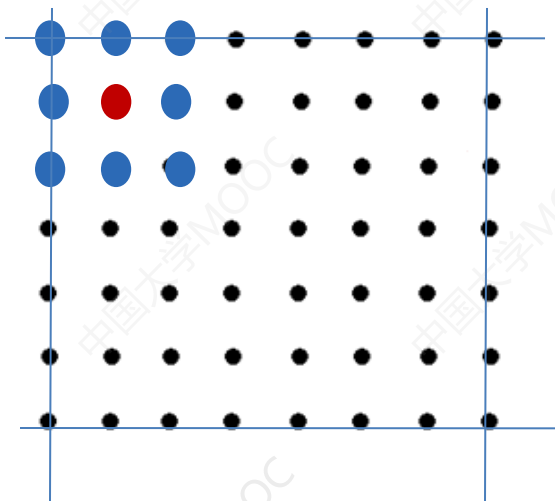
实验四 均值/高斯 低通滤波

```
void CMFCApplication1View::meanFilter(BYTE* image, int width, int heighth, BYTE* outImg)
```

```
{  
    int smth[9];  
    int i, j, m, n;  
    BYTE block[9];  
  
    int value;  
    for (i = 0; i < 9; i++)  
        smth[i] = 1;  
  
    for (i = 0; i < heighth; i++)  
    {  
        for (j = 0; j < width; j++)  
        {  
            if (i == 0 || j == 0 || i == height - 1 || j == width - 1)  
                outImg[i * width + j] = 0;  
            else  
            {  
                for (m = -1; m < 2; m++)  
                    for (n = -1; n < 2; n++)  
                        block[(m + 1) * 3 + n + 1] = image[(i + m) * width + j + n];  
  
                value = convolution(smth, block);  
                outImg[i * width + j] = BYTE(value / 9.);  
            }  
        }  
    }  
}
```

1	1	1
1	1	1
1	1	1

 $\times \frac{1}{9}$





实验四 均值/高斯 低通滤波

```
int CMFCApplication1View::convolution(int* operatr, BYTE* block)
{
    int value;
    int i, j;
    value = 0;
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
            value += operatr[i * 3 + j] * block[i * 3 + j];

    return value;
}
```

```
void CMFCApplication1View::gaussian(BYTE* image, int width, int height, BYTE* outImg)
```

```
{
    int smth[9];
    int i, j, m, n;
    BYTE block[9];

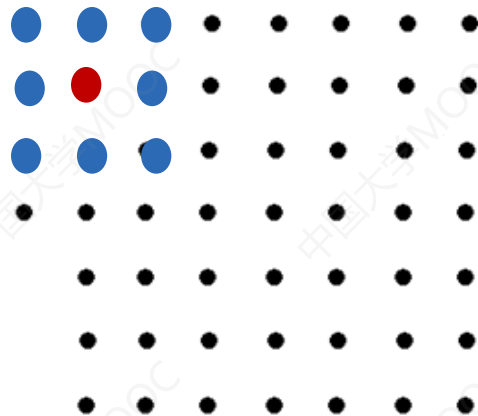
    int value;

    smth[0] = 1; smth[4] = 4;
    smth[1] = 2; smth[5] = 2;
    smth[2] = 1; smth[6] = 1;
    smth[3] = 2; smth[7] = 2;
    smth[8] = 1;

    for (i = 0; i < height; i++)
    {
        for (j = 0; j < width; j++)
        {
            if (i == 0 || j == 0 || i == height - 1 || j == width - 1)
                outImg[i * width + j] = 0;
            else
            {
                for (m = -1; m < 2; m++)
                    for (n = -1; n < 2; n++)
                        block[(m + 1) * 3 + n + 1] = image[(i + m) * width + j + n];

                value = convolution(smth, block);
                outImg[i * width + j] = BYTE(value / 16. );
            }
        }
    }
}
```

1	2	1
2	4	2
1	2	1

 $\times \frac{1}{16}$ 



实验四 均值/高斯 低通滤波





谢 谢

本课程所引用的一些素材为主讲老师多年的教学积累，来源于多种媒体及同事和同行的交流，难以一一注明出处，特此说明并表示感谢！