

多种滤波器的实现

1 实验结果



2 算术均值滤波器

计算 3×3 窗口内像素点灰度的平均值，作为窗口正中的像素点的灰度值：

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

代码如下：

```
for i=2:1:img_H-1
    for j=2:1:img_W-1
        Mean1(i,j)= (I_noise(i-1,j-1)+I_noise(i-1,j)+I_noise(i-1,j+1)+I_noise(i,j-1)+I_noise(i,j)+I_noise(i,j
            +1)+I_noise(i+1,j-1)+I_noise(i+1,j)+I_noise(i+1,j+1))/9;
    end;
end;
```

3 几何均值滤波器

计算 3×3 窗口内像素点灰度的乘积，再对所得值求 $1/n * n$ 次幂，将所得值作为窗口正中的像素点的灰度值：

$$f(x, y) = [\prod_{(s,t) \in S_{x,y}} g(x,y)]^{\frac{1}{mn}}$$

为避免像素点灰度为 0 时，乘积得 0 影响结果，先对图像矩阵整体加 1。

```
temp=I_noise+ones(img_H,img_W);
for i=2:1:img_H-1
    for j=2:1:img_W-1
        t=temp(i-1:i+1,j-1:j+1);
        s=prod(t(:));
        Mean2(i,j)=s.^(1/numel(t));
    end;
end;
```

4 谐波均值滤波器

表达式如下：

$$f(x, y) = \frac{mn}{\sum_{(s,t) \in S_{x,y}} \frac{1}{g(s,t)}}$$

```
for i=2:1:img_H-1
    for j=2:1:img_W-1
        t=I_noise(i-1:i+1,j-1:j+1);
        t=1./t;
        Mean3(i,j)=numel(t)/sum(t(:));
    end;
end;
```

5 逆谐波均值滤波器

表达式如下：

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

```
for i=2:1:img_H-1
    for j=2:1:img_W-1
        t=I_noise(i-1:i+1,j-1:j+1);
        Mean4(i,j)=sum(t(:).^4)/sum(t(:).^3);
    end;
end;
```

```
end;
```

6 中值滤波器

用像素相邻像素的灰度中值来代替该像素的值：

$$f(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s,t)\}$$

```
Mean5=medfilt2(I_noise,[3 3]);
```

7 最大值滤波器

用选取窗口中像素的灰度最大值来代替窗口中心像素的值：

$$f(x, y) = \underset{(s,t) \in S_{xy}}{\max} \{g(s,t)\}$$

```
for i=2:1:img_H-1
for j=2:1:img_W-1
    t=I_noise(i-1:i+1,j-1:j+1);
    line=t(:);
    Mean6(i,j)=max(line);
end;
end;
```

8 最小值滤波器

用选取窗口中像素的灰度最小值来代替窗口中心像素的值：

$$f(x, y) = \underset{(s,t) \in S_{xy}}{\min} \{g(s,t)\}$$

```
for i=2:1:img_H-1
for j=2:1:img_W-1
    t=I_noise(i-1:i+1,j-1:j+1);
    line=t(:);
    Mean7(i,j)=min(line);
end;
end;
```

9 中点滤波器

用选取窗口中像素的灰度最小值和最大值的中点值来代替窗口中心像素的值：

$$f(x, y) = \frac{1}{2} [\max_{(s,t) \in S_{xy}} \{g(s,t)\} + \min_{(s,t) \in S_{xy}} \{g(s,t)\}]$$

```
for i=2:1:img_H-1
for j=2:1:img_W-1
    t=I_noise(i-1:i+1,j-1:j+1);
    line=t(:);
    Mean8(i,j)= (max(line)+min(line))/2;
end;
end;
```

10 修正后的阿尔法均值滤波器

在窗口 S_{xy} 中去掉 $d/2$ 个灰度最大的像素点和 $d/2$ 个灰度最小的像素点，将剩余的点的平均值用来替代窗口中心的点的灰度值：

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

d 可以为 0 到 $mn - 1$ 之间的任意数。

```
d = 5;
for i=2:1:img_H-1
    for j=2:1:img_W-1
        t=I_noise(i-1:i+1,j-1:j+1);
        t=sort(t(:));
        min_num=ceil(d/2);
        max_num=floor(d/2);
        s=sum(t(min_num:9-max_num));
        Mean9(i,j)= s/(3*3-d);
    end;
end;
```

11 修正后的阿尔法均值滤波器

在窗口 S_{xy} 中去掉 $d/2$ 个灰度最大的像素点和 $d/2$ 个灰度最小的像素点，将剩余的点的平均值用来替代窗口中心的点的灰度值：

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

d 可以为 0 到 $mn - 1$ 之间的任意数。

```

d = 5;
for i=2:1:img_H-1
    for j=2:1:img_W-1
        t=I_noise(i-1:i+1,j-1:j+1);
        t=sort(t(:));
        min_num=ceil(d/2);
        max_num=floor(d/2);
        s=sum(t(min_num:9-max_num));
        Meang(i,j)= s/(3*3-d);
    end;
end;

```

12 自适应中值滤波器

自适应中值滤波器在两层上工作。在 A 层中，设 $A_1 = z_{med} - z_{min}$, $A_2 = z_{med} - z_{max}$ 。当 $A_1 > 0$ 且 $A_2 < 0$ 时，转到 B 层，否则增加窗口尺寸；窗口尺寸小于 S_{max} 时重复 A 层，否则输出 z_{xy} 。在 B 层中，设 $B_1 = z_{xy} - z_{min}$, $B_2 = z_{xy} - z_{max}$ 。当 $B_1 > 0$ 且 $B_2 < 0$ 时，转到 A 层，否则输出 z_{med} 。 d 可以为 0 到 $mn - 1$ 之间的任意数。

```

alreadyPro = false(size(I_noise)); %看是否完成进程
Smax=5; %最大窗口尺寸
for s = 3:2:Smax %起始窗口尺寸设为3
    %得到特定的灰度值
    zmin = ordfilt2(I_noise, 1, ones(s, s), 'symmetric');
    zmax = ordfilt2(I_noise, s * s, ones(s, s), 'symmetric');
    zmed = medfilt2(I_noise, [s s], 'symmetric');
    %进程B
    processB = (zmed > zmin) & (zmax > zmed) & ~alreadyPro;
    %进程A
    processA = (I_noise > zmin) & (zmax > I_noise);
    outZxy = processB & processA;
    outZmed = processB & ~processA;
    Mean10(outZxy) = I_noise(outZxy);
    Mean10(outZmed) = zmed(outZmed);
    alreadyPro = alreadyPro | processB;
    if all(alreadyPro(:))
        break;
    end
end
Mean10(~alreadyPro) = zmed(~alreadyPro);

```