

数字图像处理作业 7

邹永浩 2019211168

1. 修正的阿尔法均值滤波器

该滤波器与中值滤波器类似，其对一定区域内的像素进行排序，去除最大和最小区域的值，再对中间部分取均值作为结果。优点是保留了中值滤波器的效果，而且在边缘的处理上略好于中值滤波器。

其代码实现为(AlphaTrimmedMeanFilter.java):

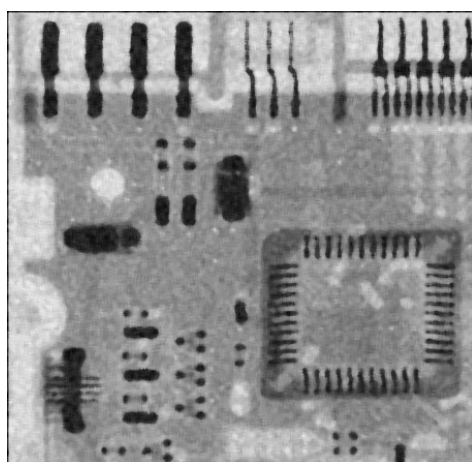
```
// 获取区域内像素值
for (int a = left; a <= right; a++) {
    for (int b = top; b <= down; b++) {
        if (a >= 0 && a < srcImg.getWidth()
            && b >= 0 && b < srcImg.getHeight()) {
            result[count++] =
                srcImg.getRaster().getSample(a, b, 0);
        }
    }
}

// 排序
Arrays.sort(result);
int mean = 0;

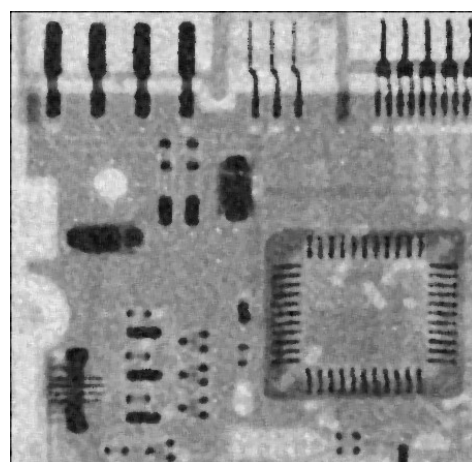
// 去除最小和最大区域

// alpha 相当于教材中的 d/2
for (int a = alpha; a < count - alpha; a++) {
    mean += result[a];
}
return mean / (count - 2 * alpha);
```

使用 5x5 的滤波器，d 取 14 与中值滤波(d 取 24 就相当于中值滤波)进行比较结果如下：

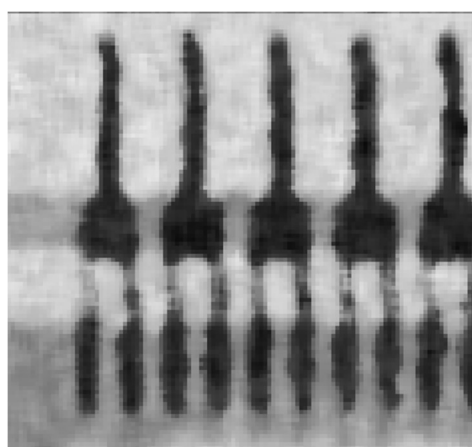


阿尔法滤波器

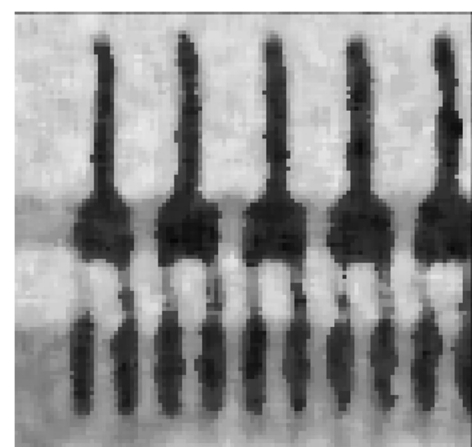


均值滤波器

可以看到两种方法结果类似，但是如果把局部放大，可以看出阿尔法均值滤波的平滑度更好一些。



阿尔法滤波器



均值滤波器

2. 自适应中值滤波器

自适应中值滤波器要更加复杂一些，但在处理更大概率的脉冲噪声时效果优于普通中值滤波。其想法为：

- 1) 如果当前像素与中值都不是极值时，输出当前像素
- 2) 如果当前像素是极值，输出中值

3) 否则增加窗口尺寸，如果已经是最大窗口，就输出中值

代码如下：

```
private static int filterA(int i, int j, BufferedImage srcImg, int
filterSize) {
    // . . .
    Arrays.sort(result);
    int z_med = result[count / 2];
    int a_1 = z_med - z_min;
    int a_2 = z_med - z_max;

    // 如果中值不是极值，进入过程B
    if (a_1 > 0 && a_2 < 0) {
        int z_xy = srcImg.getRaster().getSample(i, j, 0);
        return filterB(z_xy, z_min, z_max, z_med);
    }

    // 否则增大窗口

    if (filterSize <= MAX_FILTER_SIZE) {
        return filterA(i, j, srcImg, filterSize + 2);
    }

    // 如果超过最大窗口直接输出中值

    return z_med;
}

private static int filterB(int z_xy, int z_min, int z_max, int
z_med) {
    int b_1 = z_xy - z_min;
    int b_2 = z_xy - z_max;

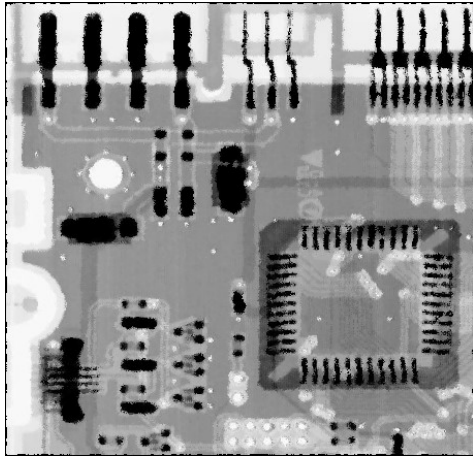
    // 如果当前像素不是极值，输出当前像素

    if (b_1 > 0 && b_2 < 0) {
        return z_xy;
    }

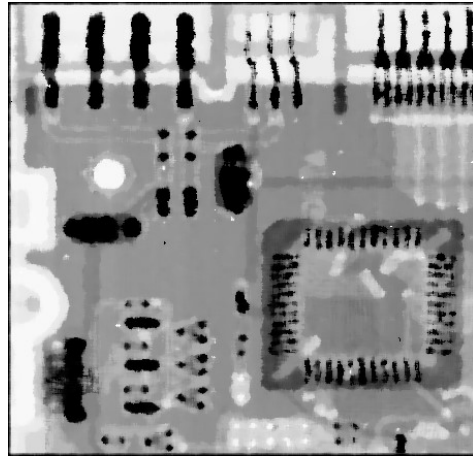
    // 否则输出中值

    return z_med;
}
```

使用 7x7 的自适应中值和普通中值滤波器的结果如下：

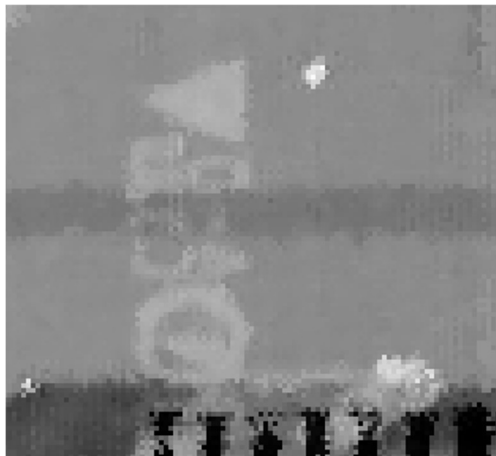


自适应中值滤波器



中值滤波器

可以看到，普通中值滤波器会出现扭曲失真的现象，而自适应中值滤波器则细节和边缘都处理的非常好，尤其是在一些边界不是特别明显的地方



自适应中值滤波器



中值滤波器

自适应中值滤波器的细节更加清晰，特征明显。

代码见：

https://github.com/zouyonghao/Image_alpha_trimed_and_median_filter