数字图像处理实验三
Experiments 3

姚鸿勋 刘绍辉
shliu@hit.edu.cn

哈尔滨工业大学计算机科学与技术学院

2022秋

任务

- 熟悉图像直方图操作和空域滤波的思想，熟悉二值图像处理的思想
- 编程实现图像的空域滤波
  - 编程实现直方图均衡化和归一化操作
  - 编程实现给图像加高斯噪声和椒盐噪声
  - 编程实现中值滤波和均值滤波，思考其快速算法实现原理，对上述的含噪声图像进行处理
  - 尝试编程实现基于小波变换的图像去噪算法实现
  - 实验报告中体现不同高斯噪声和椒盐噪声的去噪效果比较，并采用PSNR和SSIM来进行评价
- 编程实现图像的某种模板算子操作
  - 例如，sobel边缘检测等算子，理解模板操作
- 理解DWT变换域去噪思想，实现基本的DWT图像去噪
- 编程实现基本的二值图像处理操作：阈值化处理和半色调处理

滤波器的频率响应

- 给定滤波器系数
  - 如何判断该滤波器的特性
- 滤波器的频率响应函数：$H(e^{j\omega}) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n}$

  滤波器的脉冲响应：$h(n)$
- 设脉冲响应为$h(n)=\{1,1,1\}$, 则其频率响应函数是

$$H(e^{j\omega}) = \sum_{n=0}^{2} h(n)e^{-j\omega n} = (1+2\cos\omega)e^{-j\omega}$$

- 图像处理中常用的滤波器 有什么特点？
  - 线性相位

$$\angle H(e^{j\omega}) = \beta - \alpha\omega$$

  - 脉冲响应满足什么条件才能满足线性相位呢
    - 对称
    - 反对称

模板操作

- 模板操作
  – 将模板在图像上漫游，将模板中心与待处理像素对齐
  – 将模板上的系数与对应的图像像素相乘
  – 将所有乘积相加
  – 将得到的值赋予模板中心所对应的像素

Ring现象

- 使用理想低通滤波器时，输出图像会变得比较模糊并出现振铃现象
  - 为什么？

各种模板算子

- Prewitt：水平 $\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$,竖直$\begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$
- Sobel：水平 $\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$,竖直$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$
- Roberts: 分为$45^o, 135^o$两种方向，为2×2模板，45^o: $\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$;
  135^o: $\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$
- Laplacian:$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix}$

基本的直方图操作

- 给定一幅图像，计算其直方图
- 如何快速计算某个区域的直方图
- 直方图操作具有哪些性质
- 直方图相交核有何应用？

中值滤波算法：Median Filtering Algorithms

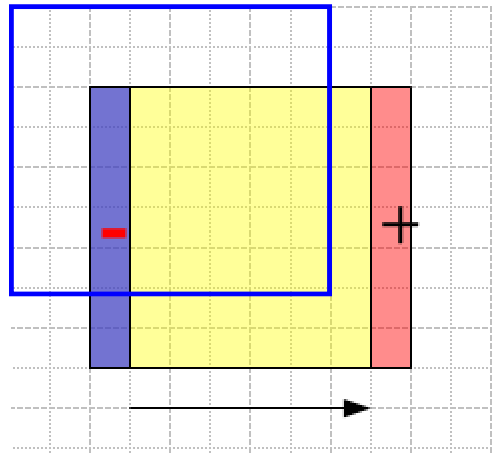
- Sorting
  - $O(r^2 log r)$, r is the kernel radius
  - With a bucket sort, $O(r^2)$
- The Classic algorithm[1]
  - Published in 1979, Thomas S. Huang
  - $O(r)$
- A Tree-based algorithm[2]
  - Published in 1993,Gil et al.
  - $O(log^2 r)$
- Sorted lists instead of histogram[3]
  - *Published in 1990, B. Chaudhuri*
  - $O(r^2)$
- Hierarchy of histogram[4]
  - Published in 2006, Weiss
  - $log(r)$, losting simplicity
- A Simplified Histogram[5]
  - Published in 2007, Simon Perreault et. Al.
  - $O(1)$
- Recent research
  - *CVPR2008*

Huang's algorithm[1]

```
hist[0:255]: histogram array;
mdn: median value in a window;
ltmdn: number of pixels having gray levels less than mdn in a window;
leftcolumn[0: 2r]: the left-most column of the previous window;
rightcolumn[0:2r]: ther right-most column of the current window;
for(i = 1;i <= width; i++)/* i indicates picture line number*/
{
    for(j = 1; j <= height; j++)/* j indicates picture column number*/
    {
        /*initialize the kernel histogram hist[], find mdn and ltmdn for
        the first window */
        for( k = 0; k <= 2r; k++)
        {
          /*deleting the leftmost col in previous win*/
          gl = leftcolumn[k];
          hist[gl] = hist[gl] - 1;
          if(gl < mdn)
            ltmdn = ltmdn - 1;
          /*adding the rightmost col in current win*/
          gl = rightcolumn[k];
          hist[gl] = hist[gl] + 1;
          if(gl < mdn)
            ltmdn = ltmdn + 1;
        }
        if(ltmdn > th)
          /*the median in the curr win is smaller than the one
          in the previous window*/
          do{
            mdn = mdn - 1;
            ltmdn = ltmdn - hist[mdn];
          }while(ltmdn <= th)
        else
           while(ltmdn + hist[mdn] <= th)
           {
              ltmdn = ltmdn + hist[mdn];
              mdn = mdn + 1;
           }
        /* mdn is the desired median in all cases*/
        /* replace the target pixel value with mdn*/
    }
```

Huang's algorithm

- Figure



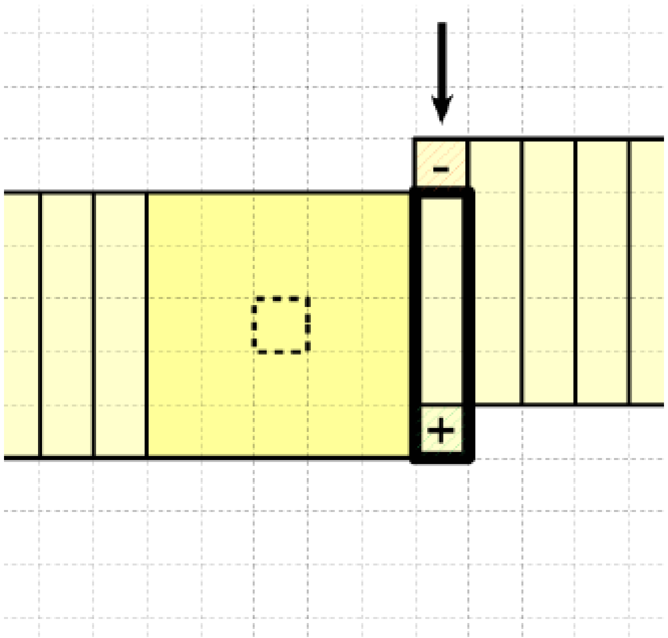
-

Algorithm 1 Huang's $O(n)$ median filtering algorithm.

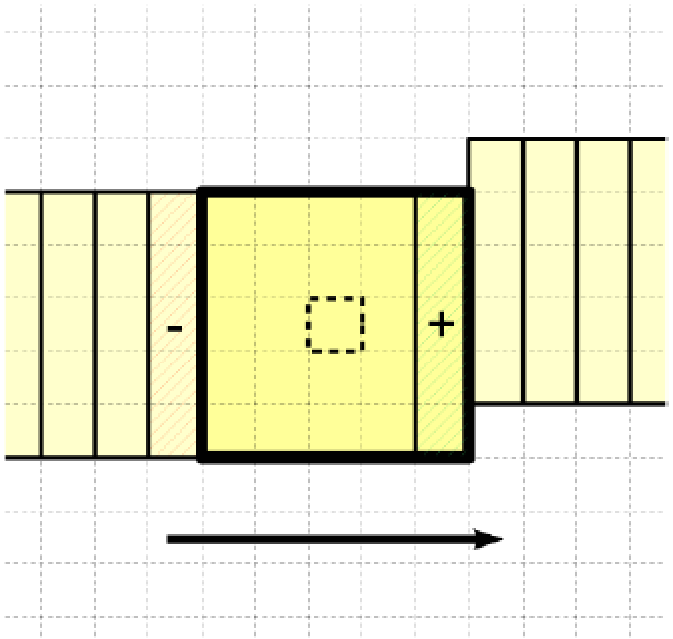
Input: Image X of size $m \times n$, kernel radius r
Output: Image Y of the same size as X
  Initialize kernel histogram H
  for $i = 1$ to m **do**
    for $j = 1$ to n **do**
      for $k = -r$ to r **do**
        Remove $X_{i+k,j-r-1}$ from H
        Add $X_{i+k,j+r}$ to H
      end for
      $Y_{i,j} \leftarrow \mathrm{median}(H)$
    end for
  end for

Simon's Algorithm

- Advantages
  - Information is retained between columns
  - Each pixel: 2r+1 histograms to be processed
- Improved
  - Information is retained between rows



Firstly, generating the histograms of columns



Secondly, generating the histogram of kernel

Distributivity of Histogram

- For disjoint regions *A* and *B*:

$$H(A \cup B) = H(A) + H(B)$$

Note: this operation is $O(1)$

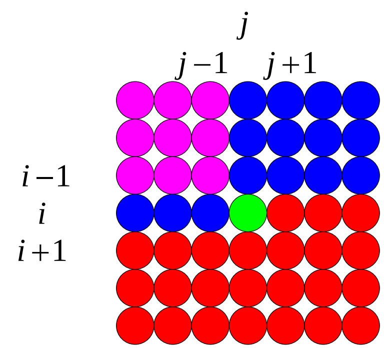
- Simon's algorithm

Algorithm 2 The proposed $O(1)$ median filtering algorithm.

Input: Image X of size $m \times n$, kernel radius r
Output: Image Y of the same size
  Initialize kernel histogram H and column histograms $h_{1...n}$
  for $i = 1$ to m **do**
    for $j = 1$ to n **do**
      Remove $X_{i-r-1,j+r}$ from h_{j+r}
      Add $X_{i+r,j+r}$ to h_{j+r}
      $H \leftarrow H + h_{j+r} - h_{j-r-1}$
      $Y_{i,j} \leftarrow \text{median}(H)$
    end for
  end for

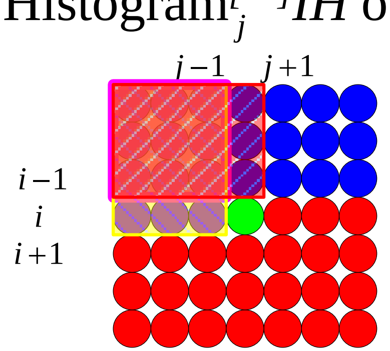
Computing the Histogram effectively

- Given an arbitrary rectangle area in an image *I*
  - How to compute the Histogram for this area?
- Possible solutions
  - A cumulative image[*] *CI* of a image *I*



$$CI(i,j) = CI(i,j-1) + CI(i-1,j) - CI(i-1,j-1)$$

  - An integral Histogram[**] *IH* of a image *I*



$$IH(i,j,b) = IH(i,j-1,b) + IH(i-1,j,b)$$
$$- IH(i-1,j-1,b) + Q(I(i,j))$$

*P.Viola and M. Jones, "Robust real-time face detection". Proceedings of ICCV 2001, Vol.II: 747, 2001

**Mikhail Sizintsev, Konstantinos G. Derpanis and Andrew Hogue. Histogram-based Search: A Comparative Study. CVPR 2008

Reference

1.Thomas S. Huang, George J. Yang and Gregory Y. Tang. A Fast Two-Dimensional Median Filtering Algorithm. IEEE Trans. On Acoustics, Speech and Signal Processing. Vol. 27, NO.1 Feb. 1979, 13-18

2. J. Gil and M.Werman, “Computing 2-D min, median, and max filters,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 5, pp. 504–507,May 1993.

3.B. Chaudhuri, “An efficient algorithm for running window pel gray level ranking 2-D images,” Pattern Recognit. Lett., vol. 11,no. 2, pp.77–80, 1990.

4.B. Weiss. Fast Median and Bilateral Filtering. ACM Trans. Graph., Vol.25, No.3, 2006, 519-526

5.Simon Perreault and Patrick Hebert. Median Filtering in Constant Time. IEEE Trans. On Image Processing. Vol. 16, No. 9, Sep 2007, 2389-2394

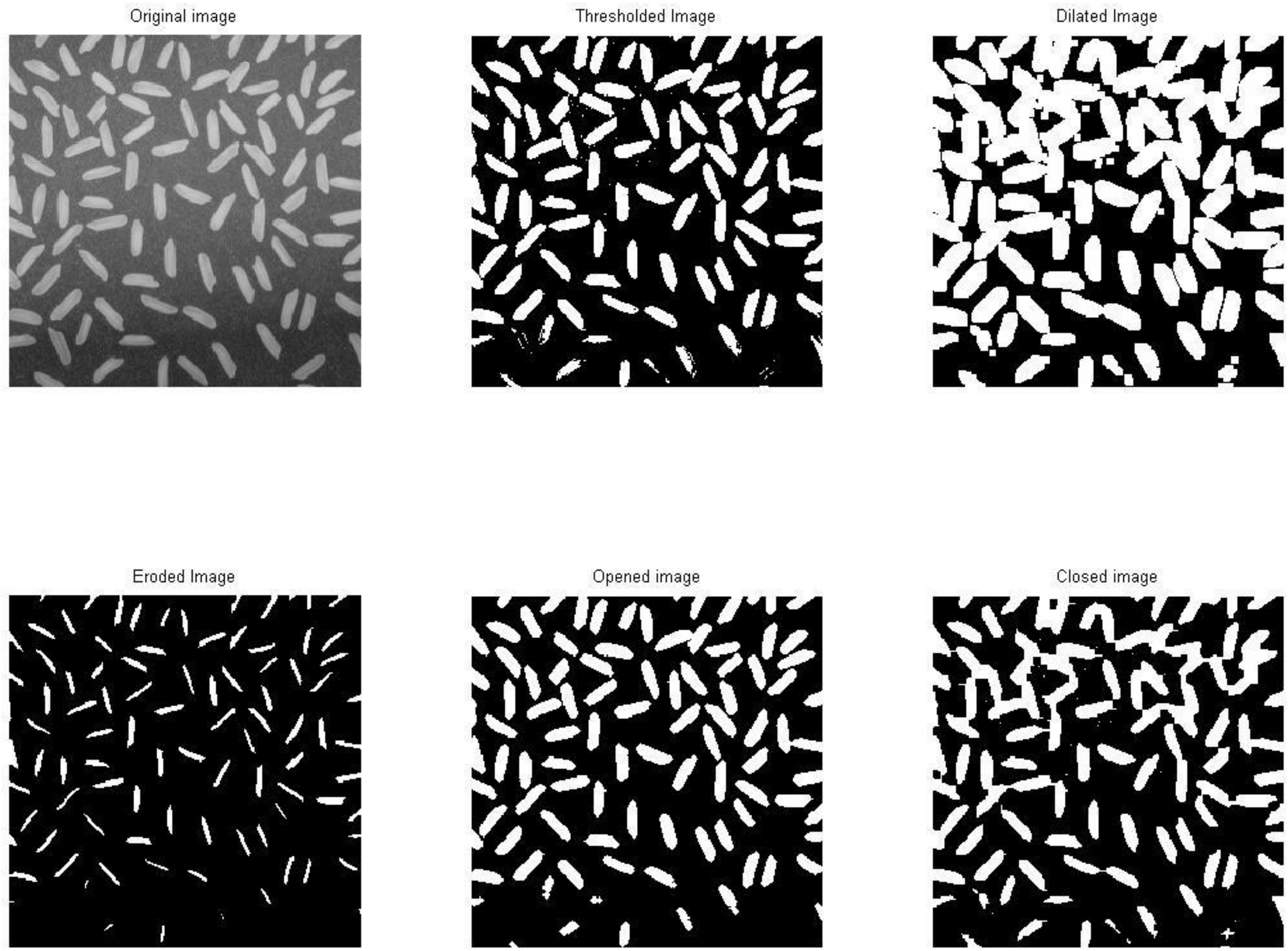
6.Mikhail Sizintsev, Konstantinos G. Derpanis and Andrew Hogue. Histogram-based Search: A Comparative Study. CVPR 2008

图像二值化：两种方式

- 阈值处理二值化

- Half-tone处理二值化

图像二值化之：膨胀、腐蚀、开、闭运算



图像二值化：半色调

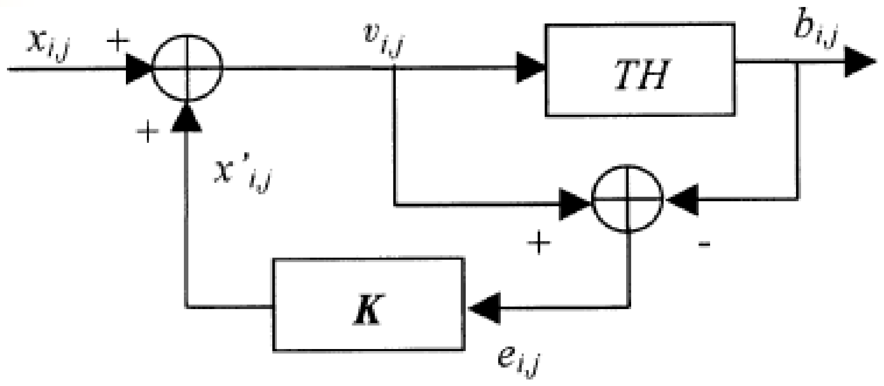
- 彩色图像的half-tone

逆Half-tone

- 逆half-tone

Half-tone



$$K = \frac{1}{48} \times \begin{bmatrix} & & \Delta & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$

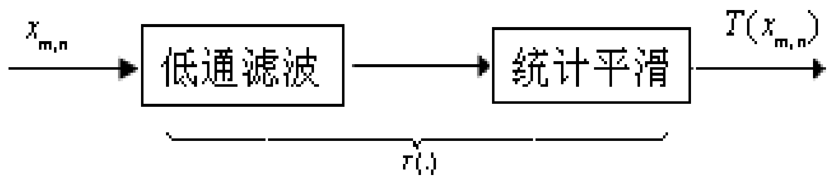
$$v_{i,j} = x_{i,j} + x'_{i,j}$$

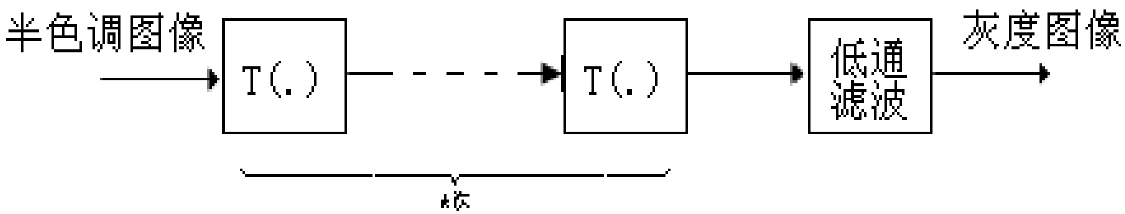
$$x'_{i,j} = \sum \sum e_{i-m,j-n} \times k_{m,n}$$

$$e_{i,j} = v_{i,j} - b_{i,j}$$

$$b_{i,j} = \begin{cases} 0, v_{i,j} < TH \\ 1, v_{i,j} \geq TH \end{cases}$$

Inverse half-tone





Reference

- V. Monga, N. Damera-Venkata, and B. L. Evans, "Design of Tone Dependent Color Error Diffusion Halftoning Systems", *IEEE Transactions on Image Processing*, vol. 16, no. 1, Jan. 2007, pp. 198-211.
- *N. Damera-Venkata, B. L. Evans, and V. Monga, "Color Error Diffusion Halftoning", IEEE Signal Processing*

  Magazine, vol. 20, no. 4, pp. 51-58, Jul. 2003, invited paper