

# 云南大学

## 本科实验报告

课程名称： 图像理解与计算机视觉

实验名称： 实验四. 形态学处理实验

学院（系）： 信息学院

专    业：

班    级：

姓    名：

学    号：

指导教师：

成    绩：

评    语：

*Steven*

## 一. 实验目的

通过编程实现能够使学生理解形态学图像处理的基本思想,掌握形态学处理的基本算法,包括腐蚀、膨胀、开运算、闭运算等。在此基础上,能够基于形态学基本运算实现几种形态学的应用处理。

## 二. 实验内容

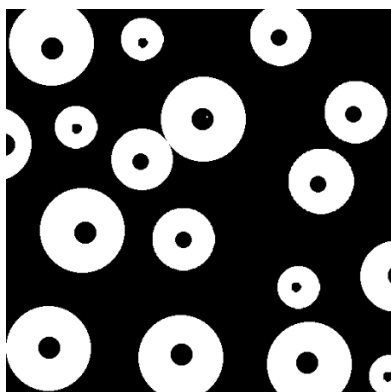
(1) 以下图像为给定的二值图像 $A_1$ , 自定义结构元 $B_1$ , 并实现结构元 $B_1$ 对图像 $A_1$ 的开运算, 接着对开运算结果进行闭运算;



(2) 以下图像为给定的二值图像 $A_2$ , 自定义结构元 $B_2$ , 并基于结构元 $B_2$ 实现图像 $A_2$ 中目标的边界提取操作;



(3) 以下图像为给定的二值图像 $A_3$ , 自定义结构元 $B_3$ 和起始点, 并基于起始点 $P$ 和结构元 $B_3$ 实现图像 $A_3$ 中目标的区域填充操作。



### 三. 实验环境

Matlab软件是图像处理领域广泛使用的仿真软件之一。本实验基于Matlab 2022版本完成。

### 四. 实验代码（详细注释，Times New Roman/宋体 五号字体 单倍行距）

#### 主函数

%% 使用: 在命令行内调用函数 Exp4(func), func 是不同功能的名字。要将文件和.m 文件放在同一路径下。

%% 每题的 demo 调用格式如下:

% 1. 先进行开运算后进行闭运算: Exp4("Ques1");

% 2. 边界提取: Exp4("Ques2");

% 3. 区域填充: Exp4("Ques3");

function Exp4(func)

if func == "Ques1"

img = imread("fingerprint.tif");

se = strel("diamond", 3); % 创建一个菱形结构元

img\_res = Ques1(img, se);

% 绘图

subplot(2, 2, 1); imshow(img); title("原图");

subplot(2, 2, 2); imshow(double(se.Neighborhood)); title("结构元, 尺寸: 7×7");

subplot(2, 2, 3); imshow(img\_res(:, :, 1)); title("开运算结果");

for i = 2:size(img\_res, 3)

subplot(2, 2, i + 2); imshow(img\_res(:, :, i)); title("第"+num2str(i - 1) + "闭运算结果");

end

elseif func == "Ques2"

img = imread("penny.tif");

se = strel("diamond", 3); % 创建一个菱形结构元

img\_res = Ques2(img, se);

<pre> subplot(2, 2, 1); imshow(img); title("原图"); subplot(2, 2, 2); imshow(img_res); title("边界提取"); subplot(2, 1, 2); imshow(double(se.Neighborhood)); title("结构元，尺寸：7×7");  elseif func == "Ques3"     img = im2double(imread("reflections.tif"));     se = strel("diamond", 3); % 创建一个菱形结构元     ptr = [59, 54];     new_img = Ques3(img, ptr, se);     subplot(2, 2, 1); imshow(img); title("原图");     subplot(2, 2, 2); imshow(new_img); title("区域填充");     subplot(2, 1, 2); imshow(double(se.Neighborhood)); title("结构元，尺寸：7×7"); end end </pre>
功能函数1：先进行开运算后进行闭运算
<pre> %% Ques1: 先进行开运算后进行闭运算 % Input/输入: %   img: 二维矩阵，表示灰度图像，二值图 %   se: 结构元 % Output/输出: %   img_new: 3 维灰度矩阵，共 6 层，分别代表开运算结果以及持续进行 6 次闭运算的结果 function img_new = Ques1(img, se)     img_new(:, :, 1) = imopen(img, se); % 开运算      while true         temp = imclose(img_new(:, :, size(img_new, 3)), se);         if temp == img_new(:, :, size(img_new, 3)) % 连续两次闭运算结果相同，则退出迭代             break;         else             img_new(:, :, size(img_new, 3) + 1) = temp;         end     end end end </pre>
功能函数2：用结构元对图像进行边界提取
<pre> %% Ques2: 用结构元对图像进行边界提取 % Input/输入: %   img: 2 维灰度矩阵，表示灰度图像，二值图 %   se: 结构元 % Output/输出: %   img_new: 2 维灰度矩阵，边界提取结果，二值图 function img_new = Ques2(img, se) </pre>

<pre> img_erode = imerode(img, se); % 腐蚀操作 img_new = img - img_erode; % 原图减去腐蚀结果 end </pre>
<p>功能函数3：用结构元对图像的目标区域进行填充</p> <pre> %% Ques3: 用结构元对图像的目标区域进行填充 % Input/输入: %   img: 2 维灰度矩阵，表示灰度图像，二值图 %   ptr: 1×2 向量，表示目标区域的起始点 %   se: 结构元 % Output/输出: %   img_new: 2 维灰度矩阵，区域填充结果，二值图 function img_new = Ques3(img, ptr, se)     img_last = zeros(size(img)); img_last(ptr(1), ptr(2)) = 1; % 初始化     % 定义交运算和并运算的函数     and = @(x, y) double(x + y == 2); % ∩:全 1 为 1；全 1 的话和为 2，用这个函数即可表示交运算     or = @(x, y) double(x + y &gt;= 1); % ∪:有 1 则 1；用这个函数即可表示并运算     i = 1; % 迭代次数      while 1         img_new = and(imdilate(img_last, se), ((-img) + 1)); % 先膨胀，再∩运算         if img_last == img_new % 若前后两次的膨胀结果相同，则迭代结束             break;         else             img_last = img_new;         end         i = i + 1; % 计数     end      fprintf('迭代次数: %d\n', i);     img_new = or(img_new, img); % img_new 此时只是膨胀区域为 1，为保持其他区域不变，要做并运算 end </pre>

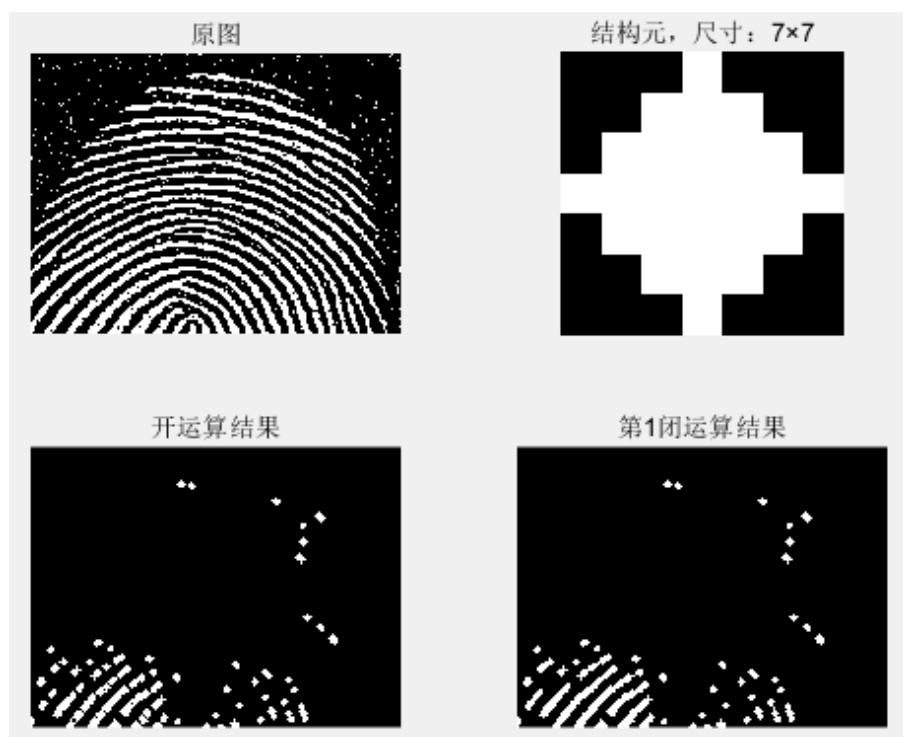


Exp4.m

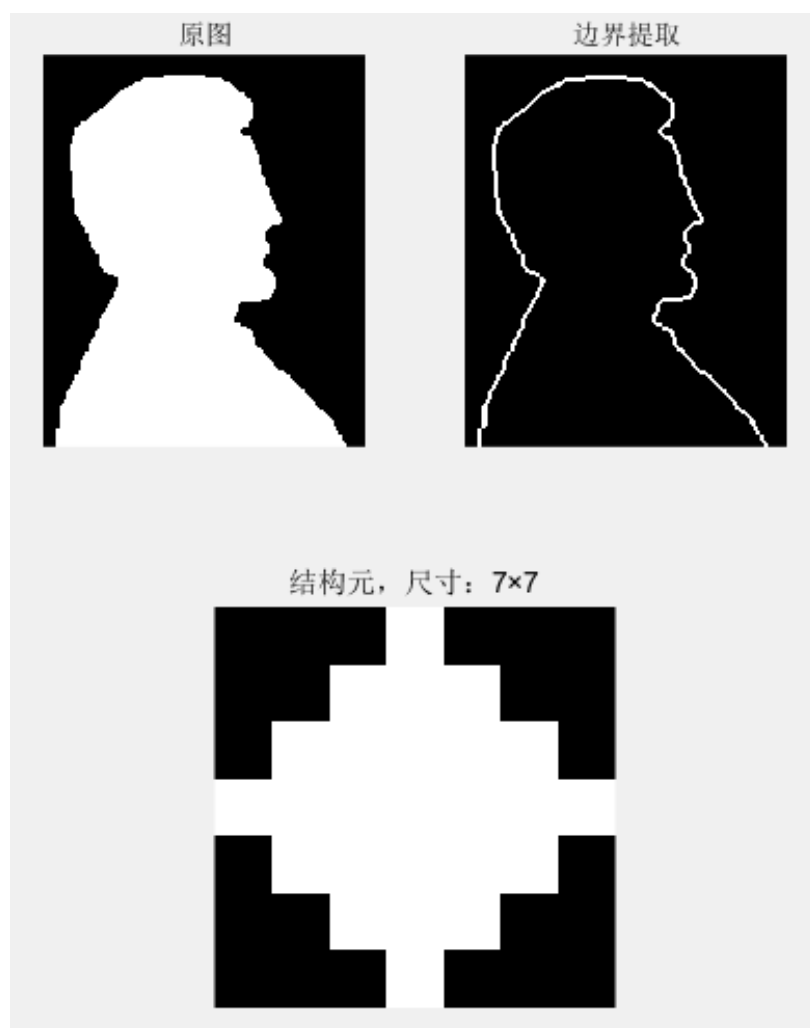
附件（.m文件）：

## 五. 实验结果

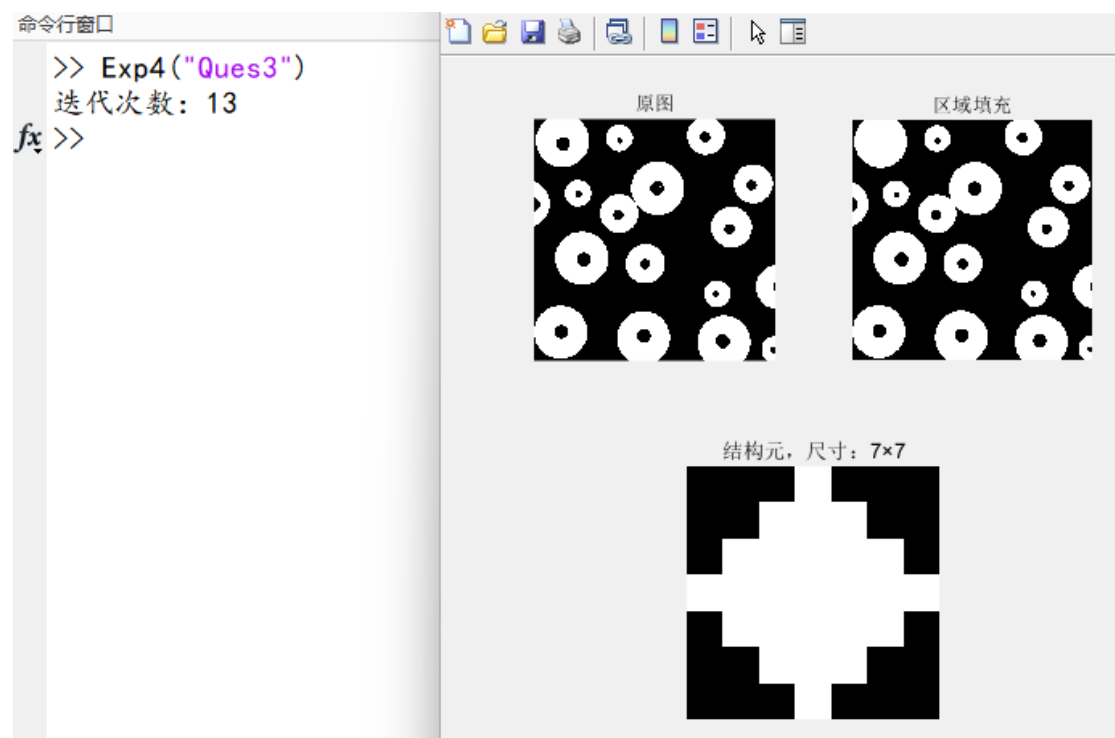
（1）二值图像 $A_I$ 、结构元 $B_I$ 、对应的开运算结果、接续进行闭运算的结果



(2) 二值图像 $A_2$ 、结构元 $B_2$ 、及对应的边界提取结果



(3) 二值图像 $A_3$ 、结构元 $B_3$ 、起始点 $P$ 的坐标及对应的区域填充结果



## 六. 结果分析及体会

本次实验比较顺利，主要是这次实验内容并不涉及到频率域只有空间域，更加直观，也更容易理解。