# 数字图像处理综合实验报告

# ——基于泛洪算法的不规则晶体标记软件

软件工程1703班 张兆琛 U201717017

目录

[1. 项目简介 2](#_Toc27952322)

[1.1 项目背景 2](#_Toc27952323)

[1.2 项目时间线 2](#_Toc27952324)

[1.3 项目成果 3](#_Toc27952325)

[2. 安装与使用说明 4](#_Toc27952326)

[2.1 安装 4](#_Toc27952327)

[2.2 功能说明 5](#_Toc27952328)

[2.3 使用示例 5](#_Toc27952329)

[3. 设计文档 8](#_Toc27952330)

[3.1 设计过程 8](#_Toc27952331)

[3.2 最终选择的解决方案 10](#_Toc27952332)

[3.3 使用的技术（关键基础知识点说明） 10](#_Toc27952333)

[4. 特色与创新 10](#_Toc27952334)

[4.1 泛洪算法 11](#_Toc27952335)

[4.2 凸包运算&多边形近似 11](#_Toc27952336)

[4.3 适于科研使用 12](#_Toc27952337)

[4.4 泛用性广 12](#_Toc27952338)

[5. 不足之处 12](#_Toc27952339)

[5.1 标记精度不足 12](#_Toc27952340)

[5.2 人机交互体验较差 13](#_Toc27952341)

[5.3 与成熟软件相比 13](#_Toc27952342)

[6. 总结 13](#_Toc27952343)

[7. 附录 14](#_Toc27952344)

[7.1 演示视频 14](#_Toc27952345)

[7.2 示例文件 14](#_Toc27952346)

[7.3 代码清单及可运行文件 14](#_Toc27952347)

## 1. 项目简介

### 1.1 项目背景

2019年第九届ACMPM亚太地区大学生数学建模竞赛(以下简称 “ACMPM竞赛”)是北京图象图形学学会主办，数学家（原校苑数模）承办的亚太地区大学生学科类竞赛，竞赛由亚太地区大学生数学建模竞赛组委会负责组织，是国内著名数学建模竞赛之一。

本次竞赛的时间为：2019年11月28日（星期四）上午6点至2019年12月2日（星期一）上午9点，在11月26日晚，熟识的两位其他学院的同学邀请我假如他们的队伍，负责**图像处理**以及实验数据计算方面的工作。我在大致了解了竞赛相关信息后，认为依靠我在**“数字图像处理”**这门科目上学习的内容可以完成此次比赛的数据处理工作，便与他们一同报名参加。

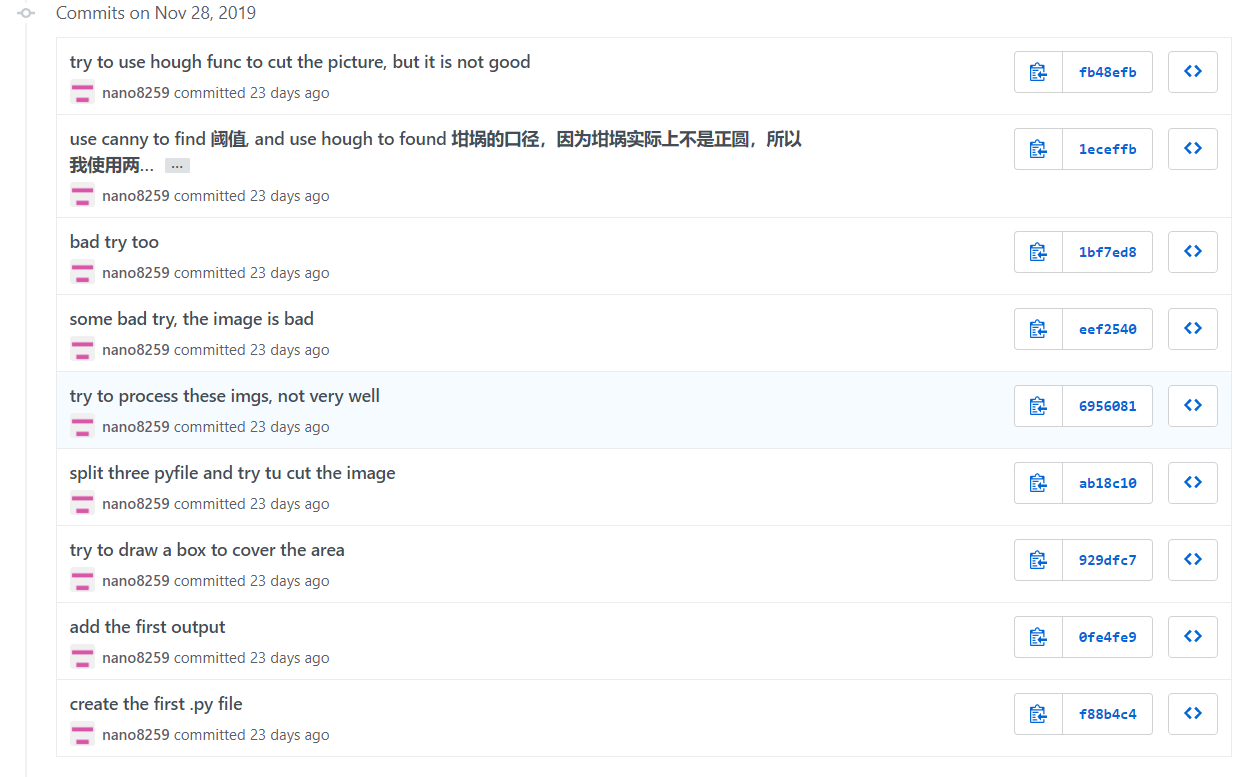
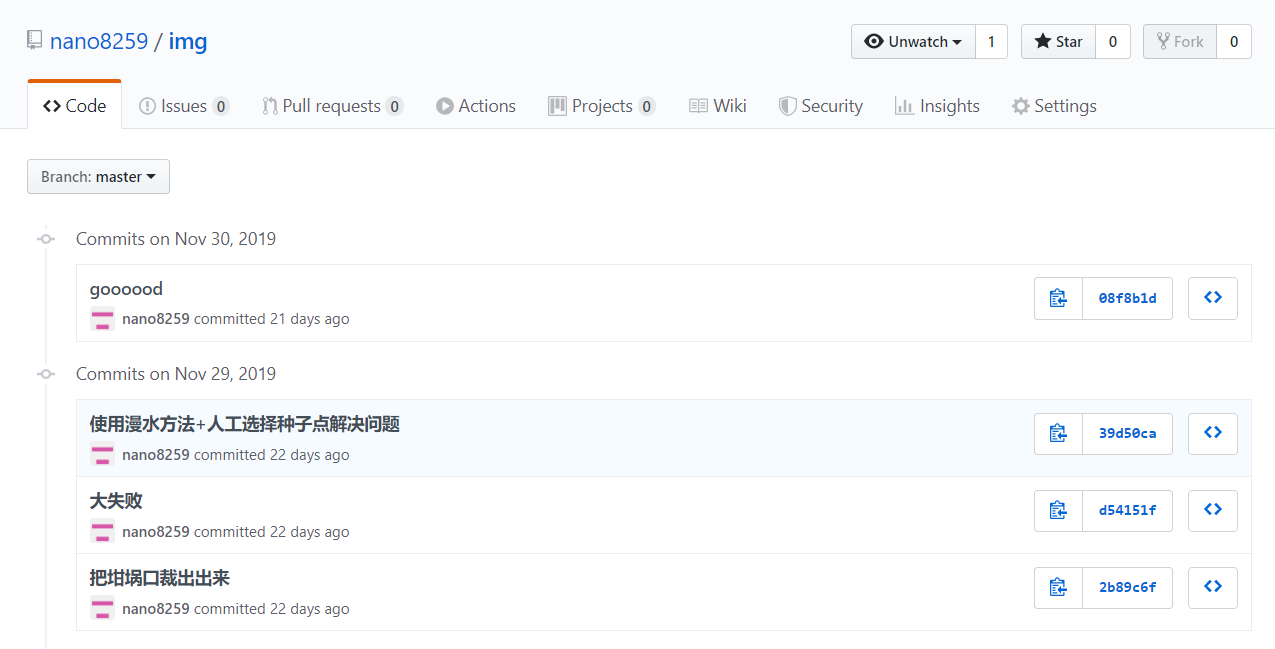
由于竞赛时间仅有短短的4天，所以本软件的开发还有许多不足之处，在竞赛结束后我也对软件进行了进一步的修改；在本报告中也会涉及一些关于竞赛数据计算的内容。

### 1.2 项目时间线

|  |  |
| --- | --- |
| 11月26日 | 支教社团中两位同学（分别来自物理、数学学院）邀请我与他们组队，一同参加ACMPM竞赛。 |
| 11月28日6:00 | 竞赛赛题公布，开始构思图像处理所需要的技术内容。 |
| 11月28日14:00-18:00 | 对图像进行初步处理，进一步观察图像后发现并不能依靠简单的方法解决。 |
| 11月28日21:00-24:00 | 使用霍夫算法捕获坩埚口，对图像进行归一化裁剪 |
| 11月29日8:00-12:00 | 通过查询资料，确定使用泛洪算法解决问题，初步尝试泛洪算法后，确定需要人工选择种子点辅助标记。 |
| 11月29日14:00-19:00 | 编写使用泛洪算法以及人工选择种子点的软件，并且人工标注出晶体轮廓。 |
| 11月30日 | 编写程序计算完成竞赛需要的其他数据。 |

本项目github仓库地址：<https://github.com/nano8259/img>

本项目git commit记录展示：

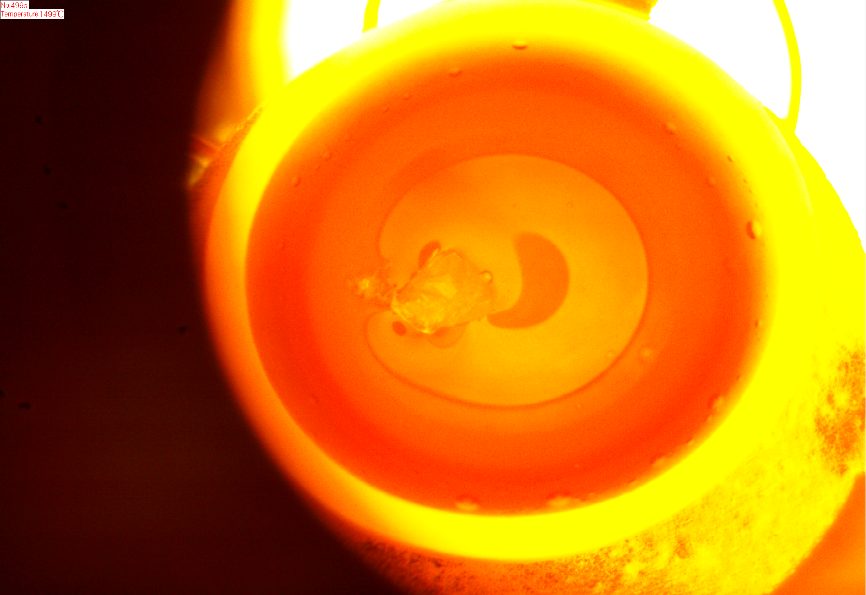


### 1.3 项目成果

在本项目中，先后实现了：识别图像中的圆形并且依此放缩并裁剪图像、使用泛洪算法+人工选择种子点的方法对不规则多边形的晶体进行精确标注。

下面展示本项目的标记结果：本项目的数据集中共有114张图片，在此仅展示其中的一张图片，其余结果图片会在项目文件压缩包中展示。

通过展示的实例可以看出，经过本项目可以分别标记两个不同大小的较模糊不规则晶体。



原始图像

图像中较大晶体的标注 图像中较小晶体的标注

## 2. 安装与使用说明

### 2.1 安装

本项目运行于python3.7.4环境下，需要主要依赖的包版本为：numpy1.17.2、opencv-python4.1.1.26。另外也提供可以在windows环境下运行的.exe可执行文件，无需安装，双击打开即可运行程序。

本软件使用pyinstaller进行从.py文件到.exe文件的打包工作，因为pyinstaller本身的局限性，使用.exe文件打开本软件会比较卡顿，建议通过运行.py文件使用此软件。

### 2.2 功能说明

本软件的主要界面为一图形界面，主要功能如下：

1. 选择文件：在打开本软件后，软件会自动提示用户选择图片，目前支持三种图片格式：jpg/png/bmp，选择了要标记的图片后进入本软件的图形界面。
2. 标记图像区域：标记图像区域为默认工作模式，在此模式下点击鼠标左键，即可以点击的像素点，以当前选择的正负差值进行泛洪运算，进行标记。

在这一工作模式下按’m’键即可切换到“④橡皮擦”工作模式

1. “洪峰”调节：使用键盘上的‘0’‘1’‘2’‘3’ 四个按键对“洪峰”进行调节，此次的洪峰为一形象说法，即泛洪运算的差值参数，详见4.1 泛洪算法。
2. 橡皮擦：在此模式下点击鼠标左键，即可擦去鼠标左键周围，以“洪峰”大小为界限的标记区域。

在这一工作模式下按’m’键即可切换到“⑤仿射变换”工作模式

1. 仿射变换：在这一工作模式下顺时针选择图像上的四个点，即可将选择的四边形区域仿射变换到600\*600的正方形区域，作为新的工作区域。

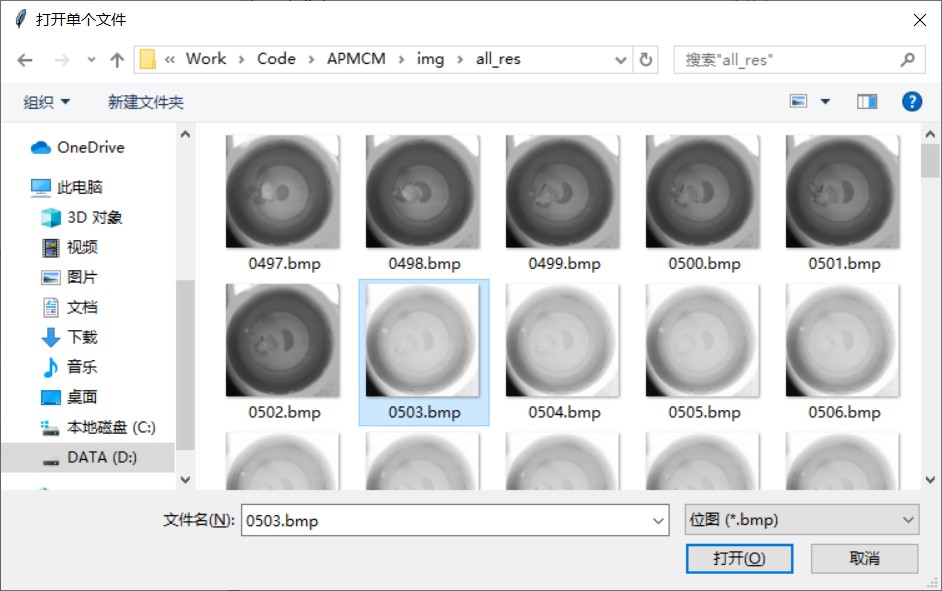
在这一工作模式下按’m’键即可切换到“②标记图像区域”工作模式

1. 使用多边形拟合标记区域：当选中了满意的标记区域后，单击鼠标右键即可调用凸包算法，使用多边形拟合标记区域
2. 直方图均衡化：点击‘h’键，即可对读入的图像进行直方图均衡化，提高标记精度。
3. 高斯模糊：点击‘b’键，即可对读入的图像进行高斯模糊，提高标记精度。
4. 重置图像：点击‘r’键，即可重置目前的图像为读入的原始图像，重新开始标记。
5. 保存标记信息及图像：当拟合出了自己想要的区域后，即可点击‘w’键保存标记的数据。
6. 选择下一图片：在任意情况下点击‘n’键，即可调出文件读取界面，读取下一个想要处理的文件。

### 2.3 使用示例

下面以一次标记为例，对程序的使用做一些简要的介绍：

1. 双击打开程序后，首先需要选择要进行标记的文件（若选择打开彩色图片，将会自动转换为灰度图像）：



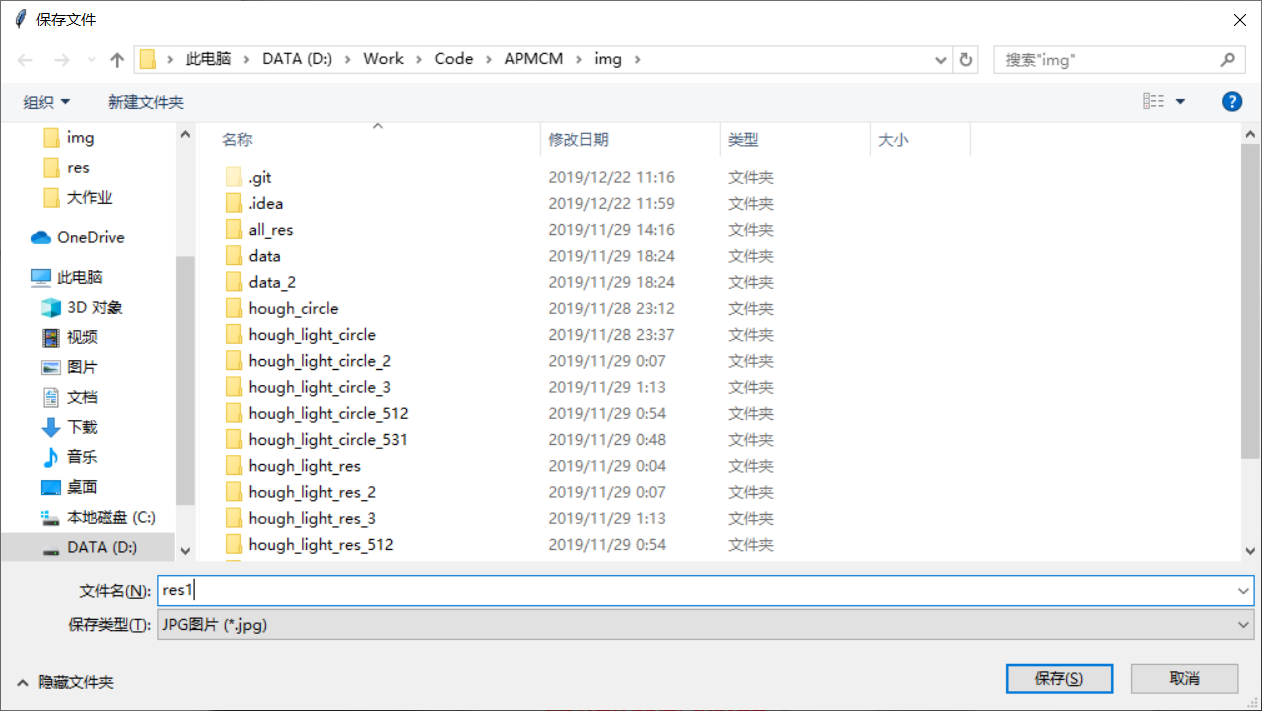
1. 在打开的图像上单击左键后，程序会以点击的像素点，以当前选择的正负差值（默认为+-1，差值可以调节，详见2.2节的内容）进行泛洪运算，进行标记。



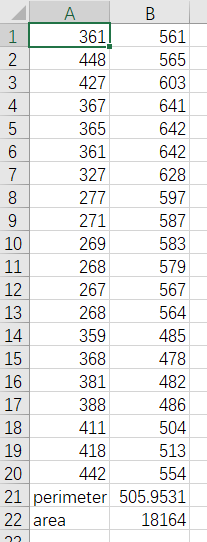
1. 当选中了合适的范围后，单击鼠标右键，程序将使用凸包算法构建一个多边形拟合选中的位置。



1. 程序会以绿色标记出选中区域的边界，以黄色标记出选中区域的形心，若对此结果满意，则可按下’w’键保存，可以选择三种文件格式：jpg/png/bmp，若不输入文件格式，则默认为bmp格式。



1. 保存后会在保存的目录下发现两个文件“文件名.bmp/jpg/png”以及“文件名.csv”，分别是带有标记的原图以及关于标记的多边形的信息：形心坐标、各定点坐标、周长与面积（坐标以图形左上角为原点），如下图所示：

## 3. 设计文档

### 3.1 设计过程

图像处理，首先对数据进行观察，发现114张图片中有的图片是彩色的，有的图片是黑白的，这两种图片模式不同代表的语义并不清楚。最后的3张图片是完全没有晶体的，尝试使用数字图像处理中的代数运算，看看是否能够直接通过最后一张图来计算出晶体的位置：

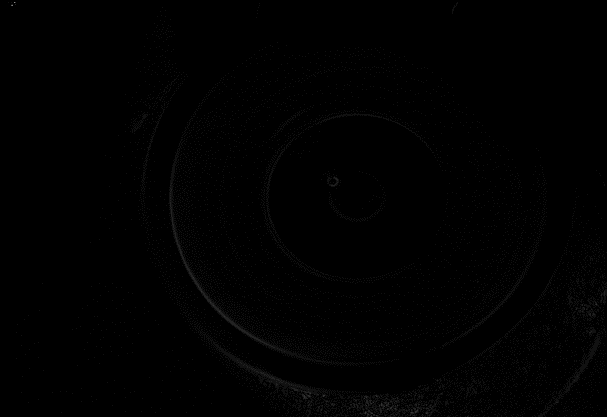
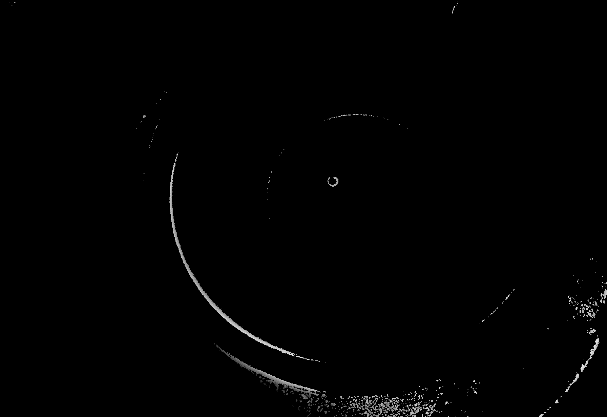
核心代码如下：

1. # 第二个图减第一个图，代数运算
2. sub\_img = cv2.absdiff(img2, img1)
3. # 使用阈值进行二值化处理
4. ret, mask\_front = cv2.threshold(sub\_img, 15, 255, cv2.THRESH\_BINARY)
5. # 掩膜操作
6. res = cv2.bitwise\_and(img2, img2, mask=mask\_front)

实验结果如下：

0599.bmp，图中仅有小粒晶体 0610.bmp图中晶体完全消失

两图相减后取绝对值 进行二值化后

实验后发现，在拍摄过程中相机存在微小移动，导致无法通过简单的代数运算找出晶体的位置。

* 如此经过多次实验及观察后，发现图像具有以下几个问题：
* 在拍摄过程中摄像机有晃动；
* 拍摄的相机并不与坩埚口的平面垂直，导致坩埚口的形状并不是正圆而是椭圆；
* 图像中并不完全是坩埚，还有许多其他无关内容；
* 图像不够清晰，有噪音；
* 晶体的颜色、边界与背景区别不大等。

这些问题都对目标晶体的提取以及追踪产生了一定的困难，需要逐个解决。

首先解决的问题是对于坩埚口的定位以及图像的裁剪。我们首先对图像的尺寸进行变换，将坩埚口变成正圆型，随后使用了“霍夫算法”对正圆型进行特征提取，特征提取完成后以圆形的外接正方形进行裁剪（图像的边缘与坩埚口相切）。这样一次性就解决了三个问题：

* 消除镜头的晃动的影响；
* 消除图像中无关内容对晶体图像识别提取的影响；
* 便于通过坩埚口的直径测量晶体的物理尺度。

接下来进行晶体的识别，首先尝试了二值化的方法，但是效果不是很好，最终我选择了floodFill算法进行识别。

### 3.2 最终选择的解决方案

最终我选择了floodFill算法进行识别。floodFill算法的基本原理是选取种子点，然后将种子点周围颜色相近的点纳入识别的范围，这样依次循环直到找不到颜色相近的点。在使用floodFill算法标记出来晶体的点后，使用凸包算法画出晶体的轮廓并且将晶体的轮廓视作一个凸多边形，存储多边形的形心作为晶体的质心，存储多边形的各个顶点作为多边形的轮廓信息，进行后续的面积、速度、加速度等信息的处理。

### 3.3 使用的技术（关键基础知识点说明）

1. **代数运算**：在本软件中大量使用了代数运算，如在图像上将拟合的多边形绘制出来，绘制出仿射变换的四边形区域等等。
2. **二值化方法**：在本软件的凸包拟合过程中使用了二值化方法，因为cv2的库中提供的泛洪算法是直接将泛洪区域绘制在图像上，所以需要使用二值化方法将泛洪区域提取出来做进一步计算。
3. **掩膜运算**：在橡皮擦功能中，本来使用的是掩膜运算（围绕点击点构建一张掩膜，将备份图像的对应区域通过掩膜运算覆盖到工作图像上来）；但是在后期代码重构过程中将掩膜运算进行了优化，直接使用python自带的list赋值功能实现。
4. **仿射变换**：本软件提供仿射变换功能，在仿射变换模式下，顺时针选取4个点即可将选取的区域仿射变换到工作区域，便于进一步的处理。
5. **直方图均衡化**：本软件提供直方图均衡化功能，当要标记的目标的灰度级和背景较为接近的时候，使用泛洪算法标记区域常常会溢出到背景中；在对原始图像进行直方图均衡化后，原本较为接近的灰度级之间的差距可能会加大，有助于提高泛洪算法的精度，防止语出。
6. **高斯模糊**：本软件提供高斯模糊功能，当原始图像中噪声较多的时候，可以使用高斯模糊来降噪，方便后续标记。
7. **多边形近似**：详见4.2凸包运算&多边形近似
8. **凸包算法**：详见4.2凸包运算&多边形近似
9. **泛洪算法**：详见4.1泛洪算法

## 4. 特色与创新

本软件使用泛洪算法，专精于不规则形状的晶体标注，功能多样且实用，与目前市面上流行的标注软件有很多不同之处。

### 4.1 泛洪算法

泛洪算法（Flood fill算法）也称为漫水填充，是从一个区域中提取若干个连通的点与其他相邻区域区分开（或分别染成不同颜色）的经典算法。因为其思路类似洪水从一个区域扩散到所有能到达的区域而得名。

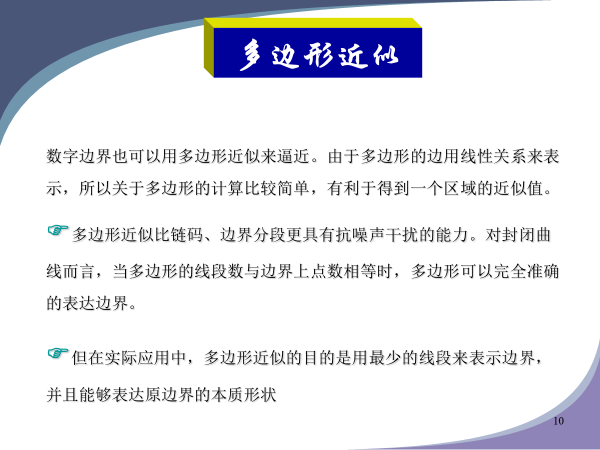
泛洪算法通常有两种工作模式：四联通模式与八连通模式，下面以灰度图、四联通为例讲解泛洪算法（从直观的角度理解，因为个人水平有限，所以不使用数学公式讲解）：

首先需要选取一个点作为种子点，然后检查周围的四个点（若为八连通模式则检查周围八个点），判断周围的点的灰度值是否处于[目前选择点的灰度值-向下差值，目前选择的灰度值+向上差值]的范围内，若处于此范围，则将这个点纳入泛洪算法的漫水区域，对这个点周围的四个点进行判断操作，依次这样操作下去，直到漫水范围不再变大为之。

在本程序中，泛洪算法上下差值仅有0,1,2,3,5五个档位。

### 4.2 凸包运算&多边形近似

使用泛洪算法标记出来的区域是一些没有规律的点的集合？那么怎么将这个标记的区域集合使用计算机可以描述的方式记录下来？我想到了薛老师曾经讲解的**第十二讲：图像的表示与描述**，最终我选择了使用多边形近似的方法，关于多边形近似的定义与优点从薛老师的PPT中摘抄如下：



确定了使用多边形近似的方法后，第二个问题是：该使用什么样的实现方法来实现多边形近似的效果呢？我在程序中选择了同样是薛老师在**第十二讲**中讲解的凸包算法：用不严谨的话来讲，给定二维平面上的点集，凸包就是将最外层的点连接起来构成的凸多边形，它能包含点集中所有的点。

最终，结合了凸包运算和多边形近似的方法，我完成了将泛洪算法标记的区域转换为计算机可以描述的多边形信息。

### 4.3 适于科研使用

本软件专精于科研方面的晶体标记，可以直接输出便于程序处理的多边形信息以及现成的周长与面积信息。

### 4.4 泛用性广

本软件并不仅能处理多边形晶体，对于人脸等常见物体的标记工作还是可以手到擒来的。以下为对于人脸的演示。

## 5. 不足之处

因为时间不足，个人技术实力有限的原因，此软件尚存在一些缺陷。

### 5.1 标记精度不足

因为本软件使用的多边形拟合算法为凸包算法，所以拟合出的多边形只能是凸多边形，但是在实际应用中，晶体并不一定是凸多边形，如下图所示，显然晶体的右下部分应该是凹陷的，但是因为凸包算法的局限性，拟合出来的多边形只能是凸多边形。

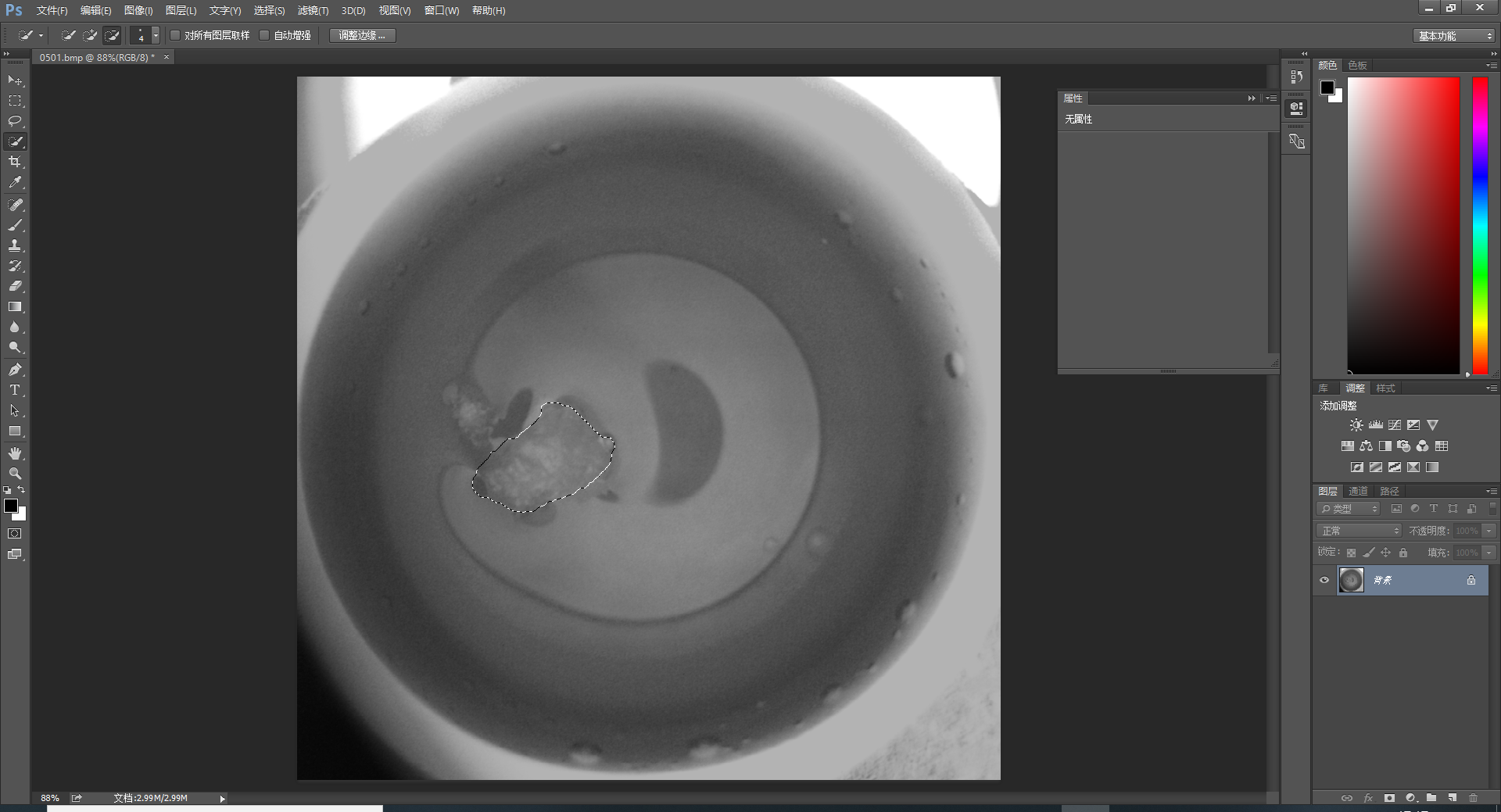
 

### 5.2 人机交互体验较差

由于软件功能较为复杂，所以我没有将有限的时间花在人机交互的体验上。本软件的交互仅仅使用键盘与鼠标，且很多功能（如选择洪峰）进行后并没有给用户相应的提示，在人机交互上还需要很大的提升。

### 5.3 与成熟软件相比

使用专业制图软件Adobe Photoshop中的“快速选择工具”来对晶体进行标记，发现PS的选择工具对于多边形的的选择明显更加的贴合，但本软件与PS相比的优点是可以直接输出便于程序处理的多边形信息以及现成的周长与面积信息。



## 6. 总结

通过本次实验，我实践了我在数字图像处理课程中学习到的大量知识点，对于知识的掌握程度进一步提升，同时也认识到了自己在这一门课程上知识体系的不足之处。

在本软件的实现过程中，我认识到了：当真正面临到实际问题时，才会发现数字图像处理的内容是博大精深的，很多问题是需要依靠多种知识综合使用才能得出想要的结果的，比如这一次的软件的设计过程中，最开始我想要用最简单的代数运算+二值化方法来进行处理，效果很差，最后使用多种方法综合使用才完成任务。这也督促着我继续努力，在薛老师讲解的知识的基础上继续学习更多的图像处理方法，深入弄懂每种处理方法背后的数学原理，不断提高自己的知识水平。

感谢薛老师认真负责的教学为我本次软件编写打下了坚实的基础。同时也感谢薛老师布置了此次综合作业，给了我将软件编写完成的动力，丰富了实践经验，本次综合实验我收获良多。

## 7. 附录

### 7.1 演示视频

因为本软件使用较为复杂，所以我为此软件录制了一个演示视频，在演示视频中我讲解了这个软件的使用方法。演示视频已经放在压缩包中。

### 7.2 示例文件

在“示例图片”中是使用本软件标记的一些图像。

### 7.3 代码清单及可运行文件

在Code和Build目录下分别是本软件的代码与可执行文件。