



# 哈爾濱工業大學

HARBIN INSTITUTE OF TECHNOLOGY

面向微系统的图像测试技术课程报告

## 圆孔图像提取与识别

卢 XX, 张 XX

\*\*\*\*\* , \*\*\*\*\*

自动化测试与控制系

指导教师 金鹏

2018 年 6 月 10 日

## 1 引言

MEMS 技术是近年来随着微电子技术的不断成熟而发展起来的一门新兴的交叉学科。其涵盖领域包括光学、机械、电子、计算机等诸多门类。随着 MEMS 技术发展及应用的扩展,对其微细结构的性能及结构的测试要求也日益迫切。由于其结构尺度属微纳米量级,因此传统意义的测试技术,几乎无能为力。本文针对 MEMS 装配领域中的目标物识别,通过图象处理处理手段,利用计算机技术及 CCD 图像采集,进行微型圆孔的自动图像识别。

## 2 设计要求

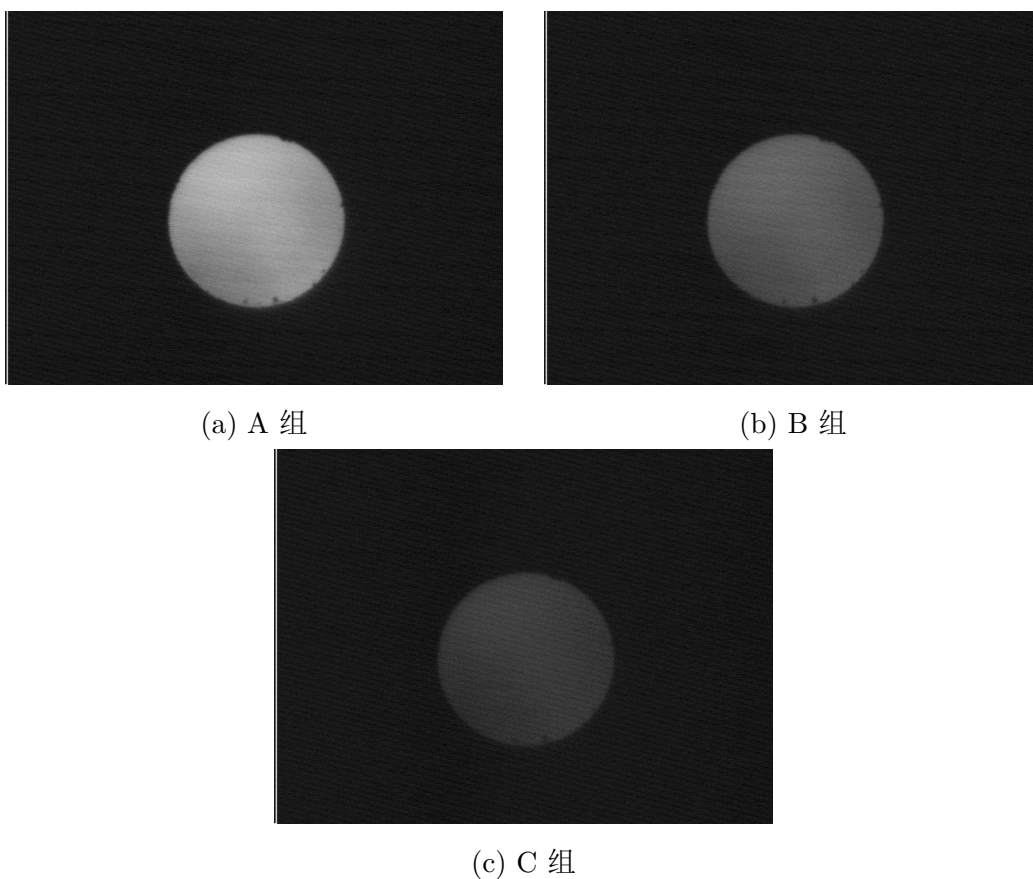


图 1: 采集图像

### 3 研究方法

---

图1 为 CCD 在不同曝光时间内采集到的微型圆孔图像。要求从图中提取出边缘信息，能够拟合出圆心和半径等参数，同时，要求能尽可能多的保留圆上的缺陷等细节信息，且处理算法应具有一定的兼容性，以便日后能够实际应用。

## 3 研究方法

虽然直接按传统方法难以准确有效地识别此类图像，但通过对传统方法的分析和组合，尝试出了一种行之有效的处理流程：

- 第一步，滤波降噪；
- 第二步，二值化处理；
- 第三步，边缘检测；
- 第四步，提取边缘，拟合圆，计算圆心和半径。

这四步基本每一步所用的都是传统算法，通过合理的排列使之得以有效的识别。

### 3.1 滤波算法

根据设计要求，滤波算法的选取应该既能够降低噪声，又可以保持图形的边缘形态。由于能力限制，最终在程序中提供的滤波算法只有四种：

**均值滤波** 线性滤波，边缘保持能力差。

**高斯滤波** 线性滤波，边缘保持能力较差。

**中值滤波** 非线性滤波，边缘保持能力较好。

**均值平移滤波** 非线性滤波，边缘保持能力好，但性能较差。

这其中，均值滤波基本就是拿来练手的，基本没有实用意义。

线性滤波基于卷积<sup>[1]</sup>实现。均值滤波使用权值相等的卷积核；而高斯滤波的所用卷积核按高斯分布分配权值。

### 3.1 滤波算法

---

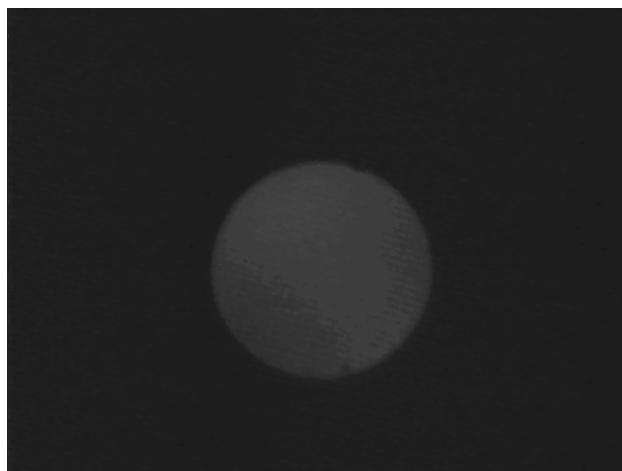
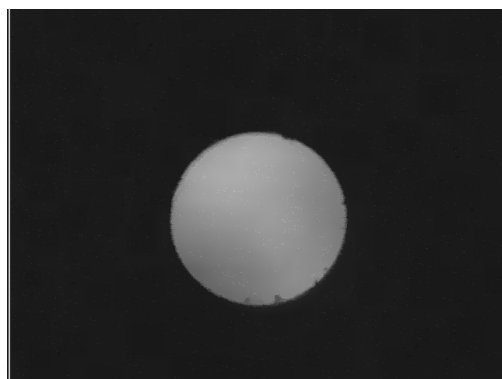
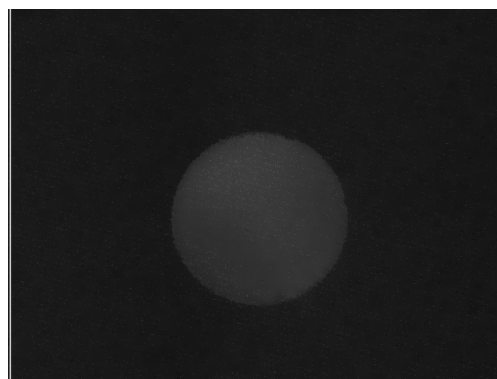


图 2: 中值滤波

中值滤波将每个像素用其周围像素的中位数替代。它不引入新的像素值，故具有一定的边缘保持能力。同时，它也是这几种方法中唯一能滤除图像左侧边缘的白线的方法，如图2所示。



(a) A 组滤波结果



(b) C 组滤波结果

图 3: 均值平移滤波

均值平移滤波<sup>[2]</sup>同时考虑了空间距离和灰度距离，只有同时在像素的空间邻域和灰度邻域内才计入平均值计算。在边缘处，像素的灰度变化通常较大，选取合适的灰度邻域范围，边缘另一侧的像素可以不被计入平均值计算，因而具有一定的边缘保持能力。但当图像曝光较差时，噪声的灰度可能和图形本身的灰度相当。此时若要继续保持边缘，则难以有效降噪。如

### 3.2 二值化之阈值分割算法

---

图3所示。

### 3.2 二值化之阈值分割算法

二值化本身的算法并没有什么特别的，就是普通的单一阈值分割。重要的是选取阈值所用的方法。

程序中尝试了以下方法：

- 灰度平均值法
- P-Tile 比例阈值法
- 力矩保持法<sup>[3]</sup>
- 大津法<sup>[4]</sup>
- 模糊度阈值分割法<sup>[5]</sup>

对经过均值平移滤波的 B 组图像进行二值化测试，结果如图4所示。可以看到，灰度平均值法得到的阈值偏低，在二值图像中引入了大量噪声；而力矩保持法得到的阈值偏高，导致二值图像中的目标图形显示不全。P-Tile 比例阈值法是半自动的阈值分割方法，选取合适的背景比例可以得到较好的二值图像，但如果比值划分不够细致，同样也不能保证获得非常准确的图像形态（如图4c）。而对于测试图像，大津法和模糊度阈值分割法得到的结果完全一致，也是效果最好的，因此程序中推荐方案使用的是这两种方法。

### 3.3 边缘检测算法

在传统的 Canny 边缘算法<sup>[6]</sup>中，边缘检测算子是先于二值化进行的。但在本方案中，边缘检测是后于二值化进行的。因而几种边缘检测算子的实际效果相差无几。

边缘检测算子是基于卷积的线性滤波，程序中用到的几种算子如表1所示。

几种方法中，Laplacian 算子效果最好（边缘最细），因而在程序中推荐的方案配置都使用的是 Laplacian 算子。

### 3.3 边缘检测算法

---

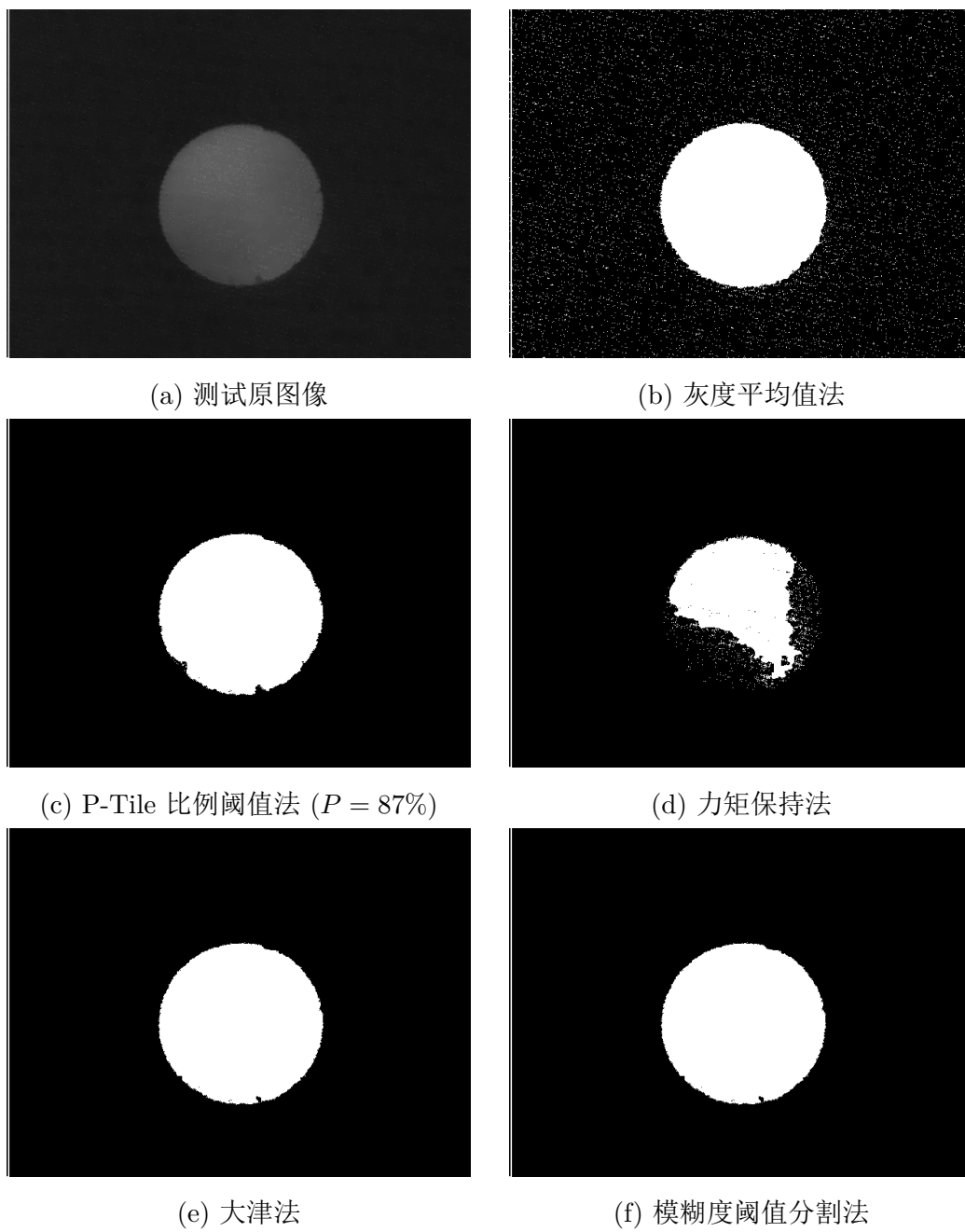


图 4: 阈值分割算法测试

### 3.4 圆拟合算法

---

表 1: 边缘检测算子

算子	卷积核
Sobel 算子	$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$
Prewitt 算子	$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$
Scharr 算子	$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}, \begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$
Laplacian 算子	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

### 3.4 圆拟合算法

程序里提供了 3 种圆拟合算法：

幼稚的拟合法：

通过平均值计算拟合，仅当数据点在圆上均匀分布时大致有效，参考意义不大。

简单代数拟合法：

通过使指标函数

$$f = \sum_i ((x_i - x_c)^2 + (y_i - y_c)^2 - R^2)^2 \quad (1)$$

最小，对数据进行拟合。

### 3.5 误差校正算法

---

相比标准的最小二乘拟合的残差平方和最小化

$$\begin{aligned} g &= \sum_i \nu_i^2 \\ &= \sum_i \left( \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - R \right)^2 \end{aligned} \quad (2)$$

此方法存在一定的系统偏差<sup>[7]</sup>。

$$\begin{aligned} f &= \sum_i \left( (x_i - x_c)^2 + (y_i - y_c)^2 - R^2 \right)^2 \\ &= \sum_i (\nu_i(\nu_i + 2R))^2 \\ &\approx 4R^2 \sum_i \nu_i^2 \end{aligned} \quad (3)$$

如式(3)所示，此方法可能会倾向于使  $R$  减小而不是单单使残差平方和最小来最小化指标函数，从而导致系统偏差。但此方法思路简单，实现容易，处理速度快，而且实测误差基本在半个像素以内，因此仍然使用此方法。

<sup>Hyper</sup>  
超级代数拟合法<sup>[7]</sup>：

结合了 Pratt 拟合法<sup>[8]</sup> 和 Taubin 拟合法<sup>[9]</sup>，拟合具有无偏性，但处理速度比前面的简单代数拟合法要稍慢。

### 3.5 误差校正算法

误差校正，准确地说是粗大误差的剔除，在原本的方案中并不存在。但伴随着程序的调试，发现图像左侧边缘的白线对拟合处理的影响极大，而且常规的滤波操作无法将其滤除，故考虑在拟合过程中加入校正环节。

最终在程序中使用了两种校正方法：

**中位误差校正：**

迭代式算法。在得到初步拟合结果的条件下剔除误差大于误差中位数的点，按剩下的点重新拟合，如此往复，直到整体误差小于给定值为止。此方法受初始情况影响大，如果最初的拟合曲线就穿过了大量干扰点，则很可能无法收敛。



## 4 实验结果

### 基于连通性的误差校正：

考虑到边缘算子得到的边缘图形必定是连通的，而干扰点与目标图形通常不会连通，故利用连通性作为判据可以分离干扰点和目标图形。再选取合适的判据分析所有连通区域中哪个最可能是目标图形，并以此区域进行拟合即得到结果。当干扰点较多时，此方法处理速度相对较慢，但准确性较好。

## 4 实验结果

如表2所示，程序附带的推荐方案有三份，分别用于 A、B、C 三组。

表 2: 推荐方案配置

方案	滤波算法	阈值分割	边缘检测	圆拟合	误差校正
a	中值滤波 滤波半径： 2	模糊度阈 值分割	Laplacian 算子	简单代数 拟合法	中位误差 校正
b	均值平移 滤波 空间滤波 半径：1 色彩滤波 半径：0.02 最大等级： 20	大津法	Laplacian 算子	简单代数 拟合法	基于连通 性的误差 矫正
c	高斯滤波 滤波半径： 2 标准差：1.5	模糊度阈 值分割	Laplacian 算子	超级代数 拟合法	基于连通 性的误差 校正

各方案对各组图像的处理结果如图5~13所示。可以看到，方案 a 和方案 c 对各组图像都有效，而方案 b 对 C 组图像无法有效处理（图12）。

## 4 实验结果

---

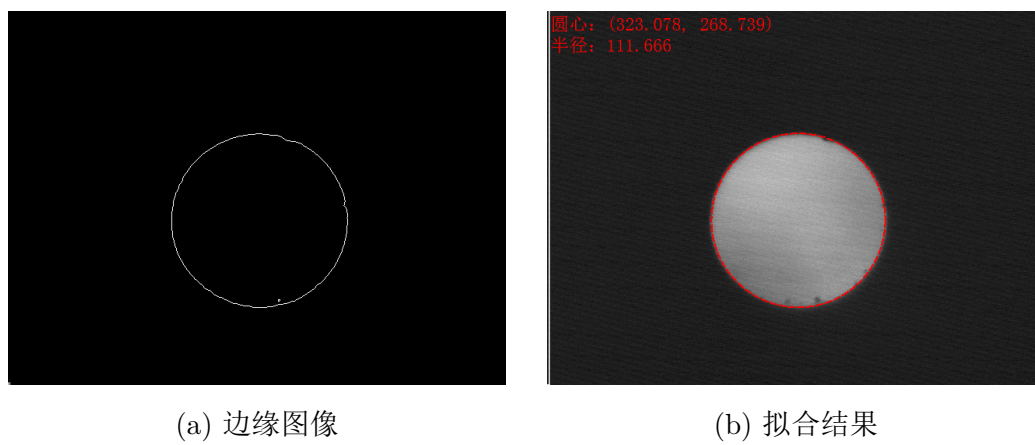


图 5: A 组-方案 a

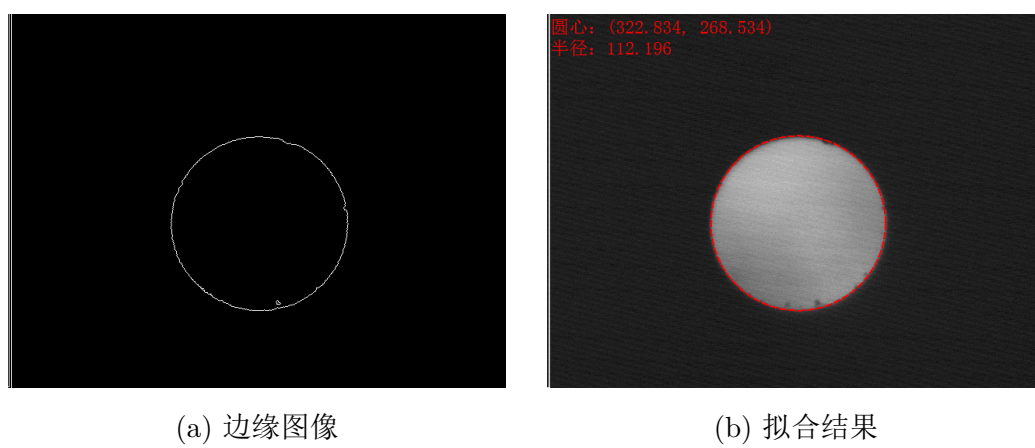


图 6: A 组-方案 b

## 4 实验结果

---

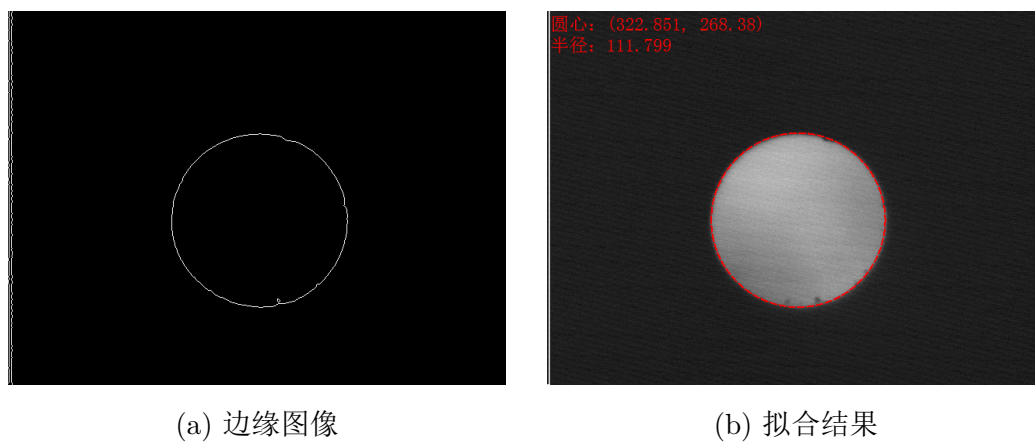


图 7: A 组-方案 c

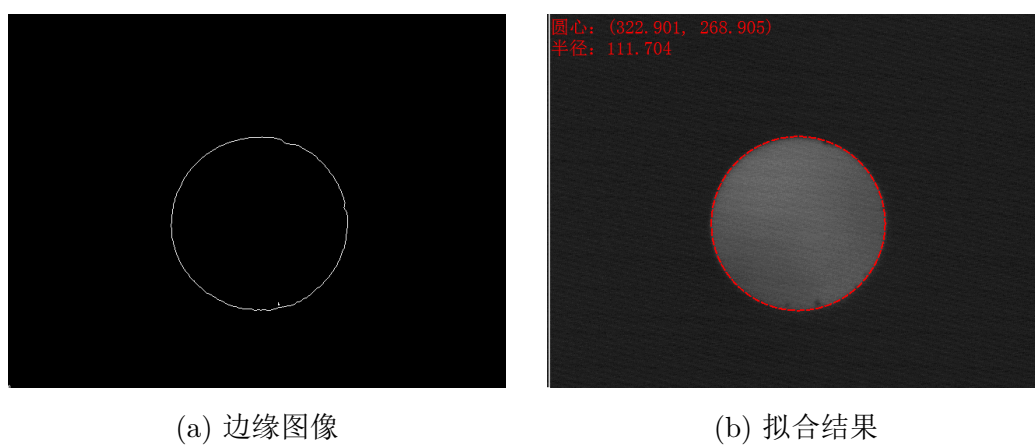


图 8: B 组-方案 a

## 4 实验结果

---

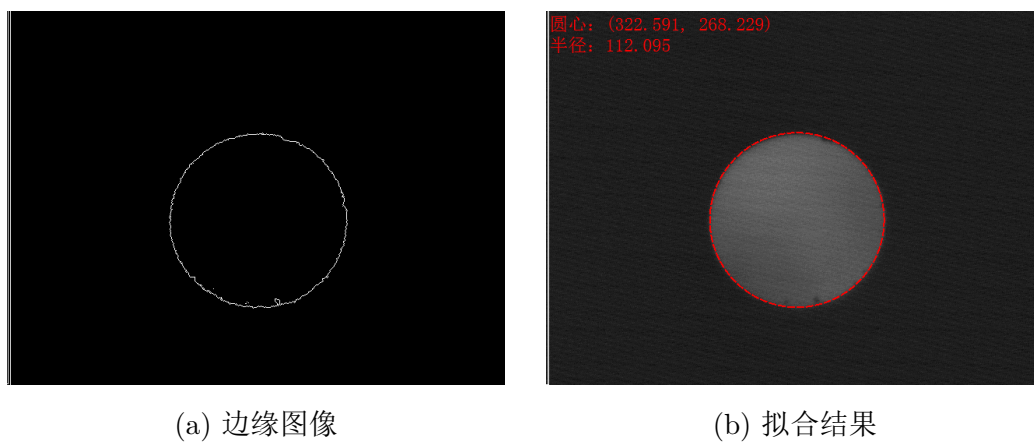


图 9: B 组-方案 b

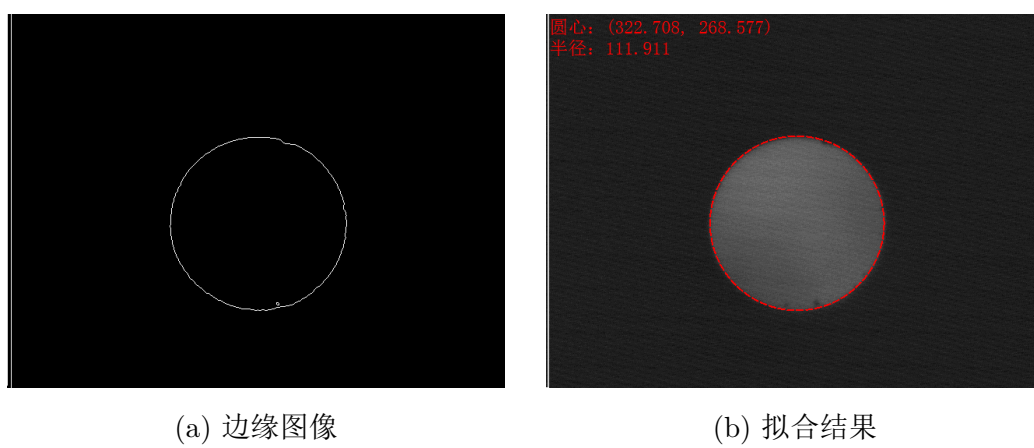


图 10: B 组-方案 c

## 4 实验结果

---

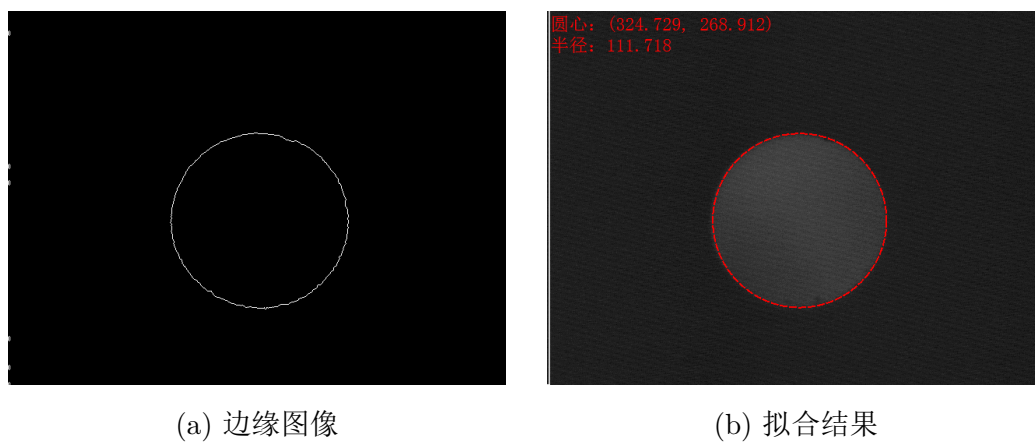


图 11: C 组-方案 a

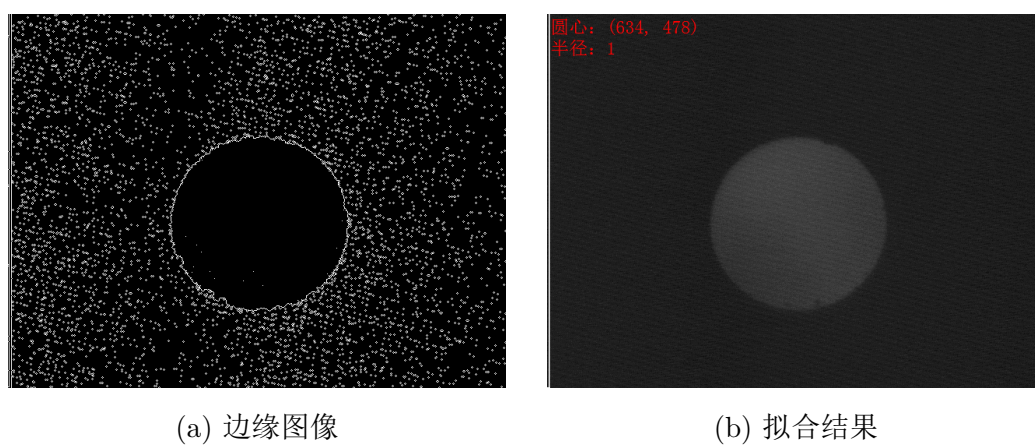


图 12: C 组-方案 b

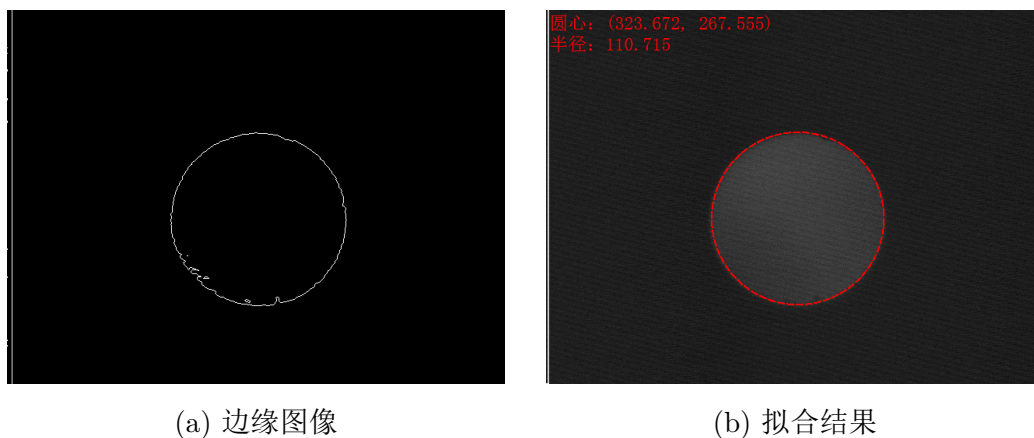


图 13: C 组-方案 c

## 5 程序说明

程序源码较长，就不附在报告里了。完整源码可以在 GitHub 仓库 <https://github.com/miRoxx/MEMS-oriented-image-testing-technology> 下的子目录 `cpp-qt` 找到。

程序基于 Qt 开发。结构上，采用视图与模型分离的模式，使程序结构清晰，开发与调试便利；行为上，操作和响应是异步进行的，将耗时的图像处理过程放置于单独的线程中，以避免界面操作被阻塞。

源文件介绍：

- `mems.pro` - qmake 项目文件，提供项目管理和自动构建。
- `main.cpp` - 主函数所在文件，程序入口。
- `mainpanel.h`, `mainpanel.cpp` - 主面板，界面响应与控制。
- `mainpanel.ui` - 基于 Qt Designer 设计的主面板界面，构建时自动生成 `ui_mainpanel.h`。
- `processor.h`, `processor.cpp` - 处理器，负责处理图像。
- `algorithms.h` - 算法包装，包含了所用到的全部图像处理算法。
- `imagefilter.h`, `imagefilter.cpp` - 图像滤波与卷积算法。

## 参考文献

---

- [binarize.hpp](#) - 二值化算法, header only。
- [thresholding.h](#), [thresholding.cpp](#) - 阈值分割算法。
- [edgedetect.h](#), [edgedetect.cpp](#) - 边缘检测算法。
- [circlefit.h](#), [circlefit.cpp](#) - 圆拟合与误差校正算法。
- [configuration.h](#), [configuration.cpp](#) - 参数配置类, 参数配置的包装。
- [thumbnailview.h](#), [thumbnailview.cpp](#) - 缩略图预览控件。
- [progressupdater.h](#), [progressupdater.cpp](#) - 进度条控制辅助类。
- [utils.h](#) - 杂项工具。
- [images.qrc](#) - Qt 资源文件, 图像资源。
- [mems\\_zh\\_CN.ts](#) - Qt 翻译文件, 中文翻译。
- [config.ini](#) - 配置文件, 提供算法的默认配置。

## 参考文献

- [1] Kernel (image processing)[EB/OL]. (2018-05-22) [2018-06-10]. [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)).
- [2] Mean shift[EB/OL]. (2018-05-16) [2018-06-10]. [https://en.wikipedia.org/wiki/Mean\\_shift](https://en.wikipedia.org/wiki/Mean_shift).
- [3] TSAI W H. Moment-preserving thresholding: A new approach[J/OL]. Computer Vision, Graphics, and Image Processing, 1985, 29(3):377-393. <http://www.sciencedirect.com/science/article/pii/0734189X85901331>. DOI: 10.1016/0734-189X(85)90133-1. ISSN: 0734-189X.
- [4] OTSU N. A Threshold Selection Method from Gray-Level Histograms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1):62-66. DOI: 10.1109/TSMC.1979.4310076. ISSN: 0018-9472.

## 参考文献

---

- [5] HUANG L K, WANG M J J. Image thresholding by minimizing the measures of fuzziness[J/OL]. Pattern Recognition, 1995, 28(1): 41-51. <http://www.sciencedirect.com/science/article/pii/0031320394E0043K>. DOI: 10.1016/0031-3203(94)E0043-K. ISSN: 0031-3203.
- [6] CANNY J. A Computational Approach to Edge Detection[J/OL]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1986, 8: 679-698. <https://doi.ieeecomputersociety.org/10.1109/TPAMI.1986.4767851>. DOI: 10.1109/TPAMI.1986.4767851. ISSN: 0162-8828.
- [7] AL-SHARADQAH A, CHERNOV N. Error analysis for circle fitting algorithms[J/OL]. Electron. J. Statist., 2009, 3: 886-911. <https://doi.org/10.1214/09-EJS419>. DOI: 10.1214/09-EJS419.
- [8] PRATT V. Direct least-squares fitting of algebraic surfaces[J/OL]. Comput. Graphics, 1987, 21(4): 145-152. <https://doi.org/10.1145/37402.37420>. DOI: 10.1145/37402.37420. ISSN: 0097-8930.
- [9] TAUBIN G. Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation[J/OL]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1991, 13: 1115-1138. <https://doi.ieeecomputersociety.org/10.1109/34.103273>. DOI: 10.1109/34.103273. ISSN: 0162-8828.