

# 华侨大学

# 毕业论文

(科学研究报告)

题    目    面向机器人抓取的零件识别与

定位

院(系)别

机电及自动化学院

专    业

机械设计制造及其自动化

级    别

2014

学    号

1411121029

姓    名

彭诚

指导老师

林俊义 副教授

华侨大学教务处

2018年6月

## 摘 要

机器视觉技术作为工业自动化技术的重要组成部分,在过去几十年内获得了广泛的关注和发展,开始广泛地进入工业界。作为视觉技术的典型应用,机器视觉和工业机器人的组合方案具有极高的研究和实用价值。本文即聚焦于这样一个课题,具体为在一台用于上下料的工业机器人上,使用机器视觉技术在二维平面上对所处理零件的图像进行处理分析,以达到识别与定位目标零件的目的,为下一步对零件的抓取提供信息。

本课题的目的是针对特定零件开发出一种视觉识别与检测软件系统,以达到成功识别与定位该零件的目的,并在这一过程中对通用机器视觉技术进行了比较与探讨,以便厘清其原理与优劣性。软件的实现过程主要分为:(1)对所获取的工件图像进行预处理,包括图像的灰度化、图像对比度增强、噪声抑制等,在每一个预处理步骤中使用多个方法对比其优劣性,以期找到最佳的预处理方案。(2)使用 Sobel 算子、Canny 算子、霍夫圆变换等方法对预处理完毕的图像进行边缘检测,分析比较不同方法的特点,为下一步的算法设计做准备。(3)利用霍夫圆变换读取工件中心坐标,对 Canny 算子所提取的边缘进行角点检测,根据几何特性和不同角点检测的阈值,提出了一种计算工件相对转角的算法。最终实现的软件算法可以成功地读取和计算工件的圆心位置、半径、相对转角等参数,为之后工业机器人的抓取提供必要的信息。

本文中算法的开发环境为 visual C++ 2017,采用 OpenCV 库来对图像进行处理,最终成果为可实现预期效果的软件系统。最终确定的算法能够识别与定位所有样本图像中的零件,且已达到较好的精确度与识别速度。在实现与选择不同算法方案的过程中,本文也对主流的算法进行了分析与比较,对机器视觉的发展和基础理论进行了一定阐述,并在此基础上也提出了一种计算相对转角的算法,能够较为稳定且准确地运行。

**关键词:** 机器视觉,工业机器人,图像处理,识别与定位

## ABSTRACT

As an important component of the industrial automation technology, machine vision has acquired broad development and received highly attention in the past decades, being used by a large range of the industry. The scheme that integrates machine vision and industrial robot as a whole, as a typical implement, is regarded of highly research value and implement value. In this paper we proposed to focus on a problem in this area. Specially, the problem needs to be solved is using proper technology methods to process and analyze the two-dimension images of a part, to achieve the purpose of recognize the target part and find out its location and orientation. These information is used for the next step's grabbing of the manipulator.

In this paper, we developed a machine vision software system based on a particular part in order to recognize and localize it, and try to compare and discuss common machine vision methods, in order to have a clear view on their principle and advantages. The main process of the software includes: (1) put forward the pretreatment on the images, including transformation to gray scale, contrast enhancement, noise suppression, etc. In every pretreatment process, different methods are used and compared basing on their advantages and disadvantages, so that an optimized scheme can be found. (2) Methods like Sobel operator, Canny operator, Hough transformation are used to extract the features from the processed images, and the method with the best process result is chosen. (3) Use Hough transformation to get the coordinates of the part center. Apply corner detection on the edges of the part extracted by Canny operator. In addition, we proposed a kind of algorithm to calculate the rotation angle of the part according to the geometrical feature of the part and the threshold of the corners. The final version of the software can successfully confirm the center, radius and rotation angle of the part, which are used for the following grabbing.

All the algorithm in this paper is programmed in the SDE of Visual C++2017, and image processing is based on the OpenCV library. The final result of this topic is a software system which can achieve the expected result. The algorithm can recognize and localize the part in all the sample images, with a preferable accuracy and speed. During achieving and choosing the plan of the algorithm, we also analyzed and compared mainstream algorithm, and made an interpretation of the development and basic theory of the machine vision technology. Also, an algorithm is proposed in this paper to calculate the rotation angle of the part, which can work relatively robustly and accurately.

**Keywords:** Machine Vision, Industrial robot, Image processing, recognition and localization

# 目 录

<b>第一章 绪论</b>	<b>1</b>
1.1 机器视觉技术的地位与作用	1
1.2 机器视觉技术的发展趋势	2
1.2.1 国外机器视觉技术的发展	2
1.2.2 国内机器视觉技术的发展	3
<b>第二章 机器视觉平台软硬件构成</b>	<b>5</b>
2.1 硬件平台构成	5
2.1.1 通用硬件平台简介	5
2.1.2 系统的工作原理及结构布局	5
2.2 软件部分构成	6
2.2.1 Microsoft Visual C++	6
2.2.2 OpenCV	6
2.2.3 软件识别算法方案设计	6
<b>第三章 图像的预处理</b>	<b>8</b>
3.1 图像的灰度化	8
3.1.1 图像灰度化的基本原理	8
3.1.2 三种灰度化的方法介绍	8
3.1.3 灰度化算法的比较与选择	12
3.2 图像对比度增强	13
3.2.1 图像对比度增强的基本原理	13
3.2.2 两种对比度增强的方法介绍	13
3.3 图像的噪声抑制	16
3.3.1 图像滤波的基本原理	16
3.3.2 四种滤波的方法介绍	17
3.3.3 四种滤波的结果对比	21
<b>第四章 图像的特征提取</b>	<b>22</b>
4.1 边缘检测算法	22
4.1.1 边缘检测的基本原理	22
4.1.2 三种边缘检测算法	22
4.2 角点检测算法	25
<b>第五章 图像的相对坐标读取与计算</b>	<b>27</b>
5.1 圆心相对坐标读取	27
5.2 工件的相对偏转角计算	28

5.3 处理结果展示与分析.....	31
第六章 结束语.....	33
致谢	
参考文献	

## 第一章 绪论

### 1.1 机器视觉技术的地位与作用

工业机器人是柔性制造的重要构成，它的柔性以及灵活性使得它成为自动化制造中不可或缺的设备，广泛应用于上下料、工件的装卸和分拣。目前在全世界有数以万计的各类工业机器人在制造业、物流业等领域应用。为了保证这些机器人能够稳定且准确地完成这些工作，必须对它们进行精确的示教，同时还需要一个相对稳定且简单的工作环境。但是工厂的环境是复杂的，在流水线上零件的分布往往是杂乱无章的，实际目标工件的位姿与理想情况总是有差别的，这种情况下即使很小的偏差也会对工业机器人完成指定的任务造成障碍。这种由于实际情况的变化而导致工业机械手不能很好地完成指令的情况极大地限制了它们的通用性与稳定性。随着智能制造与敏捷制造的出现，进一步提高生产线的柔性及稳定性要求也日渐紧迫，对工业机械手在系统应用领域的通用性、自行性和灵活性要求也日渐提高。



图 1.1 生产线上的工业机器人

然而，机器人具备一定自行性的条件是对所在环境有足够的了解，这就使得增加传感器等装置以来提高机器对四周环境的感知变得十分必要。在这一个方面，视觉、听觉、距离感知、触感觉察和力度觉察具有很高的作用价值。在这些众多感知能力里面，视觉传感提供了大量容易获取且价值较高的讯息。类似地，对人类而言，用眼睛观察世界以及认知世界是最为方便与简易的。跟据统计我们不难发现，人们从外部拿到的信息大概 80% 是利用视觉或视觉器官来获取，这不仅表明了视觉获得的信息量大，而且说明了人类对视觉讯息有较高超的利用效率，同时也告诉我们人类视觉功能的重要意义。机器视觉是指的是将人类的视觉原理近似地算法化，并且在机器上得到的体现。利用视觉传感装置比使用其他传感装置来获取所在环境以及目标工件信息相比，存在以下几个方面的优势：第一，机器视觉能够迅速拿到大量信息，同时便于自动处理，同时也便于和加工控制信息的装置与

系统相互集成,在提升制造的柔性化以及自动化程度方向存在着无比重要的作用;第二,即便在损失了大部分的信息之后,机器视觉系统所能够提供的有关于周边环境的消息依旧比超声波与激光探测器来的更加多更加准确;再其次,视觉技术的采样时间依然比激光雷达以及超声波更加短,这代表着视觉传感系统的效率以及实时性要好得多,因此更加适用于物体的在线探测、识别以及定位等等。因为机器视觉技术具有上面这些长处,因而搭配了相关系统的工业机器人具有更加广阔的应用。总而言之,使用机器视觉来增强工业机器人的智能化水平,具有相当的现实意义和无比的研究价值<sup>[1]</sup>。

## 1.2 机器视觉技术的发展趋势

### 1.2.1 国外机器视觉技术的发展方向

机器视觉的大发展开始于 1980 年代,其产生和发展与机器视觉和机器人控制技术的发展密切相关。最近几年,视觉传感技术俨然成为高新技术领域的一个重要研究领域。机器视觉技术为人形机器人、工件装配机器人、抓取机器手连同其他种类的机器人解决现实问题提供了基础。其已经使得传统的生产状况发生了巨大的变化,对社会的生活和生产行为产生广泛而深远的影响。目前涌现出了很多关于机器视觉的基础理论和前沿应用方面的研究,一些诸如神经网络、深度学习等新的理论手段使得机器视觉有了新的发展,使其成为计算机领域热门的发展方向之一。机器视觉技术是在 1950 年代从统计学中的模式识别技术开始的,其时的研究主要集中在平面图像的分析与识别中,如光字符识别与认定、工件粗糙表面、纤维高清图片细节和航空细节图片的分析与解释。到了 60 年代,Roberts 利用相关的视觉技术实现了从平面化图像中提取出现实物体的空间结构,并且使得程序能够读取物体在空间中的位置。Roberts 的研究成果使得研究者们开始关注对三维场景的认知识别的算法与理论研究的研究。Roberts 的创造性成果给了科学界和学术界以极大的激励。1970 年代,当时世界上已经出现了一些视觉识别的应用系统。

在 1954 年,美国的 George C. Devol 研发了当时世界上首台机器人视觉识别装置。1960 年代机器人正式产品化,机器人的相关技术开始走向成熟。到了 1961 年,麻省理工学院的林肯实验室开始把一些装配有传感器的遥控机器的从动部分与计算机相互连接起来,以此形成的机器人可以凭借传感器感知目标物体的状态。之后,把摄像机器看作输入设备,把计算机图像处理技术以及物体识别等技术作为核心处理系统也一并引入到了机器人的系统中,机器视觉至此正式与机器人系统进行了结合<sup>[1]</sup>。

在 1970 年代,麻省理工学院的相关实验室聘请了众多相关领域学者共同进行了视觉领域的理论研究,算法以及相关集成系统设计的研究当中。到了 1973 年,David Marr 教授到该实验室主导一个研究课题小组。在 1977 年他们提出了不同于之前“积木的世界”的分析方法理论。这个理论在之后逐渐变成为机器视觉技术研究领域的一个不可或缺的框架。可以这么说,对视觉技术的研究兴起于 1980 年代,并在 1985 左右达到顶峰,在这一过程中,机器视觉技术获得了极大的发展,新的概念、新的方法、新理论都不断出现。例如,基于感知的特征群体的识别理论之框架、主动的视觉的理论之框架、计算机视觉的集

成理论框架等。

因为欧美世界对机器视觉的研究以及应用到机器人并与之相结合的案例都发生比较早,所以他们的技术发展的较完善,同时一部分商品化的集成计算机视觉系统都已经出现。在这方面开展的较早的是美国,早在 1970 年代,GM 公司就研制出了能精准识别流水线上零部件的视觉识别系统。到了 1980 年代,归功于 CPU 的普及,行业内对机器视觉集成系统表现出了特别大的兴趣。1980 年代国际市场中一个机器人集成系统 Puma/VC-100 成功实现了商品化。它的功能包括有物体识别,物体位姿坐标标定等。它使用的视觉技术能够在零件加工与装配流程中对物体进行检测。机器视觉系统是利用示教演示获得的。美国国家科学基金会为代表所研究的实用视觉系统迅猛发展。另一方面,日本的国家政策中,同样把大力发展工业化机器视觉技术放在了首要的地位。进入 90 年代之后,工业级别的视觉技术在汽车、物流、电子、食品、制药等效率需求高、劳动性密集化的产品线上已经成为必须的一环,达到了人眼所不能够起到之作用,有力的保障了产品的质量<sup>[2]</sup>。

斯坦福大学的研究所在早期所研发之机器人——Shkey,是一种较为典型的“车一眼”集成系统,其主要功能是能够在视野所及的范围之内识别出目标,所仰仗的是积累的经验构成的算法来求解他的行动和规划,同时利用逻辑推理来构成问答的能力。Shkey 能够穿梭房间之内,搜索和识别指定的物体,同时能够进行类似智能化的操作。GM 公司开发出了可在噪音环境之下工作的视觉识别系统。Adept 是全世界知名之工业机器人制造企业,其在研制其第一代智能机器人之时便考虑到了加入视觉识别技术,所以在二十几年的发展过程中,他们取得了较为丰富之经验和相对比较成熟之技术,以至于成为了美国最大的工业机器研发与制造企业。至此,美国涌现出了众多拥有核心技术的工业机械手企业,在国际上占据了很大一部分市场。之后,日本在利用后发优势,参考了其他国家研究的智能机器人的方面经验以后,在视觉技术与机器人技术上也获得了很大的发展。日立 central 研究所研发的集成了自主控制能力之智能机器人,能够用来完成按照图纸进行装配的操作。德国西门子公司在工业级别图像处理领域有多于 20 年的研发经验。他们的 SIMATIC VIDEOMAT 系统是全球首个高性能的能够同时处理单色和彩色图像的图像处理系统,同时成为 SIMATIC 自动化制造系统中及其重要之产品。在 1998 年西门子推出的 SIMATIC VS710 是行业内第首台具备智能化、一体化同时配备 PROFIBUS 接口,为分布式结构的灰度级别机器视觉系统。它将图像处理器件、CCD 光传感器、IO 接口集成为一个系统,同时提供了 PROFIBUS 的联网传输方式或者集成到一起的 RS232 接口。OMS 的视觉识别系统,是德国研制的商业级别的通用性系统软件,具有物体识别和边缘检测等功能。它的操作模式分为主动的以及被动的、交互的这三种,比较常见的应用在零件的分拣、装配和质量控制等方面。另外,加拿大和瑞士等国家,工业机器人和视觉领域的研究与产业化都有着不容忽视的实力<sup>[2]</sup>。

### 1.2.2 国内机器视觉技术的发展方向

我国对与机器视觉技术的研究其实是在改革开放之后才开始进行的,并同时在上



有了极大的发展。但同时，工业视觉的集成系统必须有其应用之市场才可以发展。改革开放之后，视觉系统的检查往往被制造企业所忽略。这个因素是造成产品质量不稳定的重要因素。伴随着企业们的深化改革，制造的产品质量相关的竞争已严重关系到企业的生存和发展。这就为工业视觉的发展创造了有利条件。这种情况下，视觉的研究才渐渐获得制造业的重视。

关于机器视觉的研究也在我国的许多高校中开展，在华中科技大学的彭刚教授提出了利用视觉引导和超声波测距的方法来构建能够对运动物体的追踪和抓取的机器人系统。席文明团队在东南大学引入了一种利用基于机器视觉的引导技术的机器人来追踪复杂焊接轨迹的系统。基于双头摄像头获取的图像进行运算，最终能够求得焊接缝隙上的离散点，之后根据工艺的要求来对空间点进行插补运算。北京航空航天大学的孟僊团队提出了一种能够利用技术视觉的标识目标物体的方法，具有可以判断未知目标能否被抓取的条件，同时能够为抓取提供有效的信息。在多次定位的任务中，基于目标物体的先验信息快速定位，这种方法避免了重复移动机械手。华中科技大学的王敏等人提出了一种将视觉与超测量相耦合的理论，提出将平面数字图像和超声信号进行融合与检测，实现了在装配时对工件进行识别定位的目的，同时实现对工件边缘的提取。汕大的彭惠青团队通过对图像的二值化处理进行边缘检测，获得了较为理想的处理效果，但必须进行遍历搜索，耗费算力和时间。在清华大学的毛德柱等人成功利用遗传算法匹配了物体的位姿。这些国内的理论研究和算法研发都为我国的机器视觉技术在先进制造技术中的实际应用打下坚实基础<sup>[3]</sup>。

科学界对三维计算机视觉技术的研究的开展距今已有好几十年了，在理论与实现中均有了很大的发展。但是能够机器视觉系统在实际工业生产中的应用还不够广泛。最近几年，机器视觉系统一直通过堆积硬件及搭建庞大繁杂的软件库来实现功能的提升，它们几乎都存在高度专业化和高成本的问题。这种情况在国内也十分明显，不仅仅是由于其研发与硬件的成本特别高；同时还因实际问题的复杂性以及理论发展的不完善，很多问题都有待进一步研究，因此在图像信息处理中仍然存在瓶颈，并且适用范围也相对较窄。总之，当下对于机器视觉的算法实现和其与工业机器人控制技术相结合的实践，从各方面看都有其深入系统地研究的必要性。

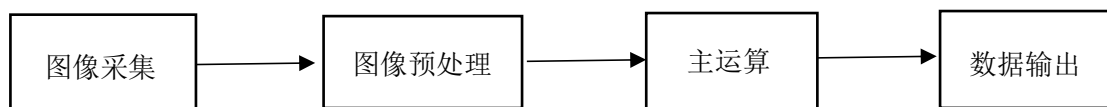
## 第二章 机器视觉平台软硬件构成

### 2.1 硬件平台构成

#### 2.2.1 通用硬件平台简介

机器视觉系统大致分为 PLC 式和 PC 式两种。前者多用于进行有无判别或形状匹配等效果。后者比较适用于各种复杂的环境，装配、定位、校准、识别等流程可以在其上实现，同时由于 PC 中可以使用 Windows 界面，开发者可以方便地进行编程，具有稳定性强，可扩展性强和通用性强等特点。随着工业自动化的不断发展，机器视觉系统在工况环境监控、缺陷和质量检测等方面有广阔的前景。

硬件是实现机器视觉算法的物理载体，也是其重要构成部分。机器视觉集成系统的基本组成如图 2.1 所示。一般来说，光传感器对工业相机发送指令信号；图像采集卡的作用是根据控制指令来存储采集完毕的图像；然后机器视觉算法软件进行相应的图像处理工作，包括图像信息的提取，如对工件的识别、尺寸与位置信息的读取等。系统的鲁棒性是对其进行评价的一个重要指标，照明系统、拍摄系统、处理系统、执行系统都对最后的识别效果起着非常重要的作用，因此子系统的性能好坏决定了最终的整体集成系统的性能。



2.1 视觉系统的信号基本流程框图

#### 2.2.2 系统的工作原理及结构布局

(1) 系统组成：系统的主要功能是利用机器视觉技术对当前时刻所拍摄的机械工件的图像进行相应处理，并进一步分析得出相应的结果，这些信息会传输到执行机构并进行下一步的决策。系统主要包括传送带 1、图像信号触发装置 2、光源子系统 3、图像采集卡 4、控制信号发生装置 5、执行机构系统 6、图像处理及分析软件 7。整个系统框架结构如图 2.2 所示。

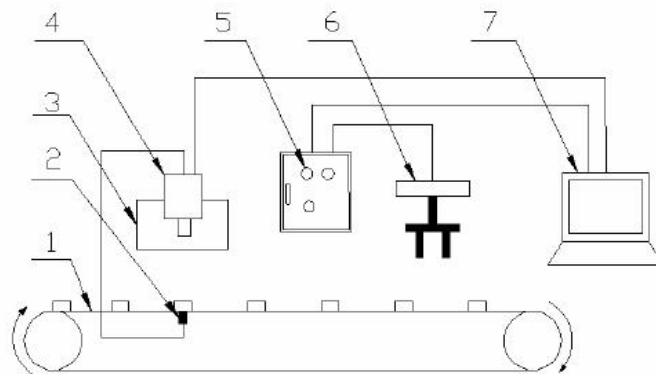


图 2.2 机器视觉系统整体框架组成图

(2) 工作原理：其原理是送料装置应该与传送装置采用相同的时钟，传送带上布置了随机位置的工件，当光感信号被触发并将信号向下一级传输后，工业摄像机被触发开始对正下方的工件进行拍摄，并对采集到的图像进行简单处理，去除冗余信息，之后将其传送给图像分析软件进行处理和判断，处理的结果会被反馈给执行机构进行下一步的操作。

## 2.2 软件部分构成

### 2.2.1 Microsoft Visual C++

Visual C++ 是 Microsoft 公司基于 Windows 平台、的面各对象的可视化 IDE。这一集成开发环境是开源的，与多种软件实现进行数据资源共享的同时进行库文件的传输也十分方便，可以进行各种函数库的调用使用。在软件中可以方便地使用构建好的软件框架，同时也有多种图形界面控件可供调用，节省了很多的开发时间，大大提升了软件的开发效率。同时该 IDE 的可扩展性也很轻，多数图形处理支持库比如 openCV，openGL 等等，只需经过简单的配置即可实现在 C++编译环境下的调用。同时使用 VC 还可利用 C++强大的接口和类函数功能，满足编写软件的多种需求。

### 2.2.2 OpenCV

OpenCV 全称 open source computer vision，是一个可以跨平台的计算机视觉函数库，包含了目前几乎所有能涉及到的视觉处理函数和算法。OpenCV 主要使用的编写语言是 C++，这也是本文为什么选用 VC++作为集成开发环境的原因之一。其接口也来自 C++，同时很多的 C 语言接口也被保存下来。本课题中使用 VC++和 OpenCV 这个方案同样也有处于成本的考虑，这二者都是开源程序或库，开发并不需要负担太高成本的同时，有能够获得异常丰富的视觉处理算法。相比之下，其他的计算机视觉软件开发语言和环境一是慢且不稳定，同时与其他库不兼容；二是需要比较高的花费才能获得正版权限；三是依赖于一些特别的解决方案因而不具有通用性。OpenCV 基于在真实世界的应用而不断改进，因为使用 C 代码来进行编写，故可实现较高的运算效率。

### 2.2.3 软件识别算法方案设计

本文要解决的课题为，利用上述提到的软件库与集成开发环境，识别流水线上的目标工件，读取出其位置信息和转角信息，为下一步的机械手抓取做准备。工件的图像已经由工业摄像机获取，如下图所示：

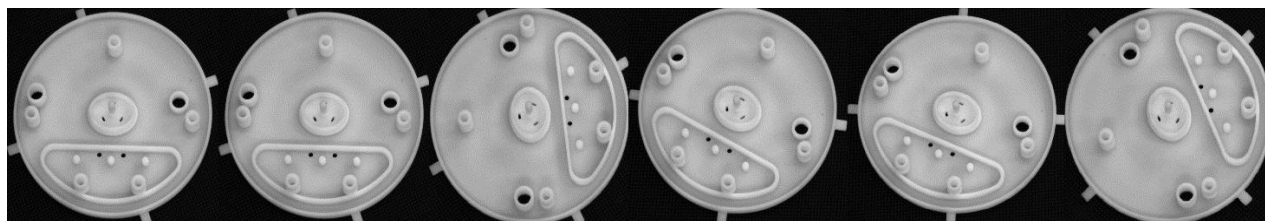


图 2.3 需识别的目标工件部分图像

为使机械手在下一步顺利安装该工件，视觉识别算法不仅需要读出当前工件相对于设定坐标原点的位置偏移量，而且要知道工件相对于设定模板位置的相对转角，以此才能确保此种非对称工件的准确安装。对零件位置信息的读取主要是依靠边缘检测算法提供的工件几何信息来确定的，因此实现较好的边缘检测效果是本文图像处理方案选择的一个重要指标。在进行图像分析和处理时，边缘检测常常是最为重要的一步，关系到后续整个算法的识别效果。边缘检测根据图像中局部像素的亮度的不连续性来识别与判断边缘，或者说，边缘检测算法检测的是局部亮度梯度最大的区域。为实现这一点，图像需要进行一些合适的处理过程，我们称之为图像的预处理，分为灰度化、图像对比度增强、滤波降噪这三个过程。这是机器视觉算法所通用的方案。边缘检测之后，需要对获得的几何特征进行提取和计算，以便获得相关的几何坐标。因此，本文确定的软件处理方案如下框图所示：

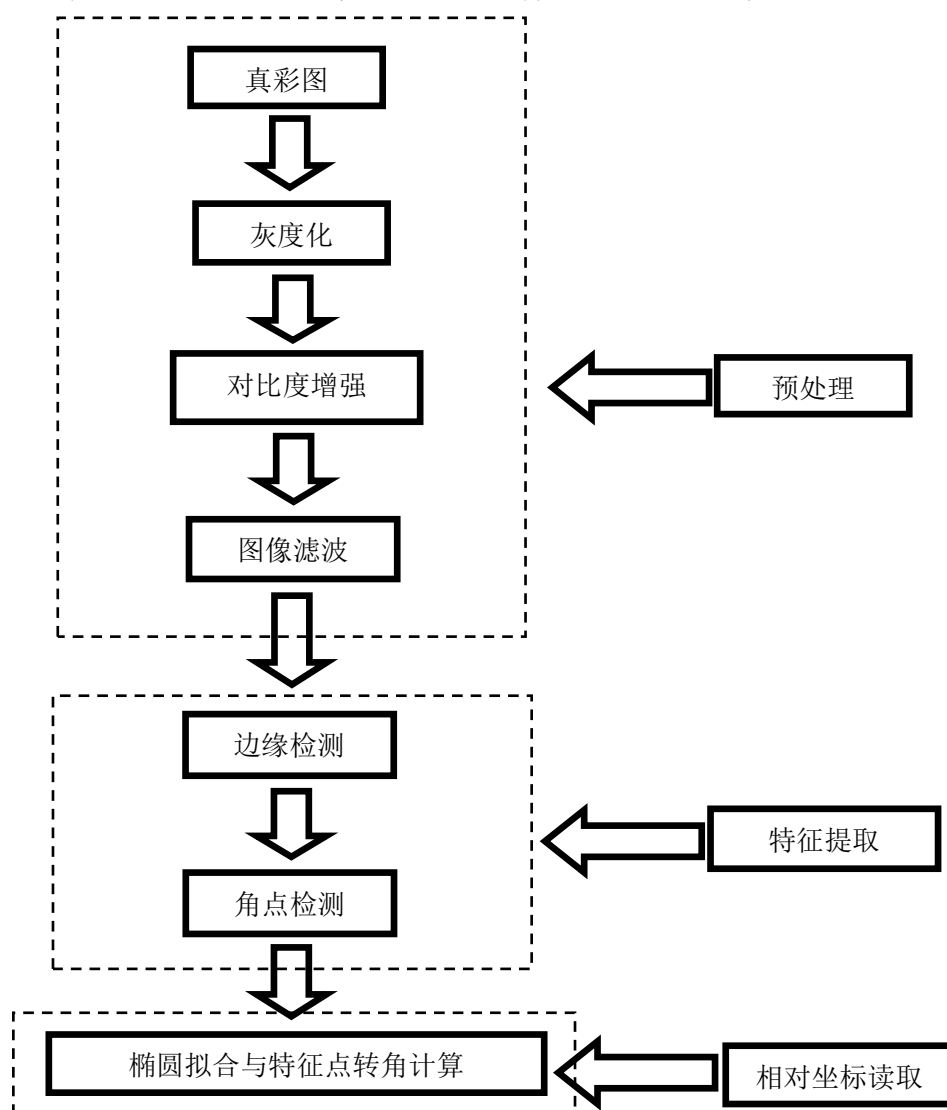


图 2.4 图像处理算法的主要步骤

## 第三章 图像的预处理

### 3.1 图像的灰度化

#### 3.1.1 图像灰度化的基本原理

工件的图像经过 CCD 相机拍摄后, 将其转化为数字信息存储到图像卡中, 这一过程成为图像的采集。由于计算机能够分析的只是数字信息, 所以一幅图像在进行处理前必须转化为合适的数字信息。其方法是将图像划分为数量足够且有限的小网格, 我们称之为像素。在每个像素里都有不同颜色通道的亮度值, 通常我们认为由 R、G、B 三种颜色通道的不同亮度组成了彩色图形里面的所有颜色。这种用一个值来表示像素的亮度的方法我们称之为像素离散化。完成上述的工作之后, 一张图片就被一个数值矩阵所表示, 以便于后续的计算处理。

由采集设备获取的彩色图像包含有较多信息。而对于本课题的研究内容而言, 许多被包含的信息都是对于物体识别来说不需要的。保留这些信息参与算法处理会大大拖慢图像处理的时间和效率, 这些冗余信息也会形成干扰, 降低图像识别的准确度。因此, 为了达到视觉识别系统对于识别的效率和实时性的要求, 我们需要对图像进行灰度化处理。具体为, 使得每个像素里 RGB 三个颜色通道的分量是相等或同比例的, 去除多余的彩色颜色, 只保留每个像素之间亮度上的不同。后续的计算则主要根据像素间这种亮度上的差异来实现识别算法。当然, 在这之前还需要提高亮度差异、增强对比度等一系列的操作。

#### 3.1.2 三种灰度化的方法介绍

灰度化处理的方法主要有以下三种:

(1) 加权值法: 赋予 R、G、B 三种颜色不同的权值来实现所有像素颜色的统一化, 权值的选取往往根据人眼的敏感度或者根据后续图像处理的要求来选取权值。用公式表达为:

$$G = W_R \times R + W_G \times G + W_B \times B \quad (3-1)$$

其中  $W_R$ 、 $W_G$ 、 $W_B$  分别为三个颜色通道的不同权值。

根据人眼对不同颜色的敏感度有不同的权值配比, 每一种权值的配比被成为颜色空间。在其中, YIQ 颜色空间为美国电视系统所采用, 构成了其 NTSC 彩色制式。这种制式分离了颜色与亮度信息。其中的 Y 表示亮度分量, 描述灰度信息, I 代表色调, Q 代表饱和度, 后两个分量表示彩色信息。将后两者分离, 将 RGB 颜色空间乘以 Y 分量的权值即可得到 YIQ 空间的灰度图。这种权值配比对于人眼较敏感, 也为最多采用的权值方案。转换公式如下:

$$RGB[A] \text{ to Gray} : Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (3-2)$$

将公式转换成代码如下：

```
float R, G, B;
for (int y = 0; y < src.rows; y++)
{
    uchar* data = src0.ptr<uchar>(y);
    for (int x = 0; x < src.cols; x++)
    {
        B = src.at<Vec3b>(y, x)[0];
        G = src.at<Vec3b>(y, x)[1];
        R = src.at<Vec3b>(y, x)[2];
        data[x] = (int)(R * 0.299 + G * 0.587 + B * 0.114); //利用公式计算灰度值(加权平均法)
    }
}
```

图 3.1 权值法灰度化的公式代码

用该方法灰度化处理之前的图像如下：

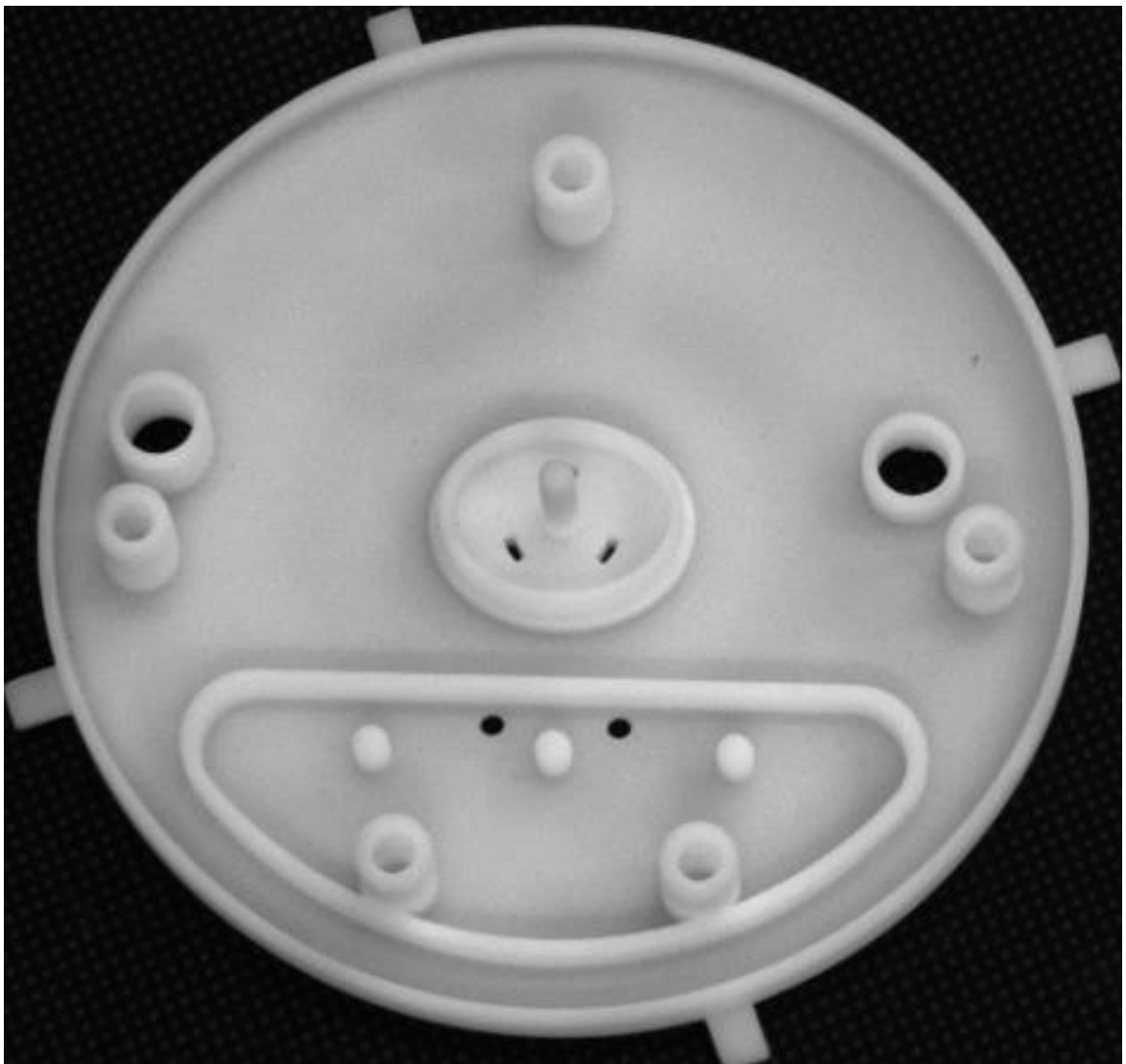


图 3.2 权值法灰度化处理之前

用该方法灰度化处理之后的图像如下：

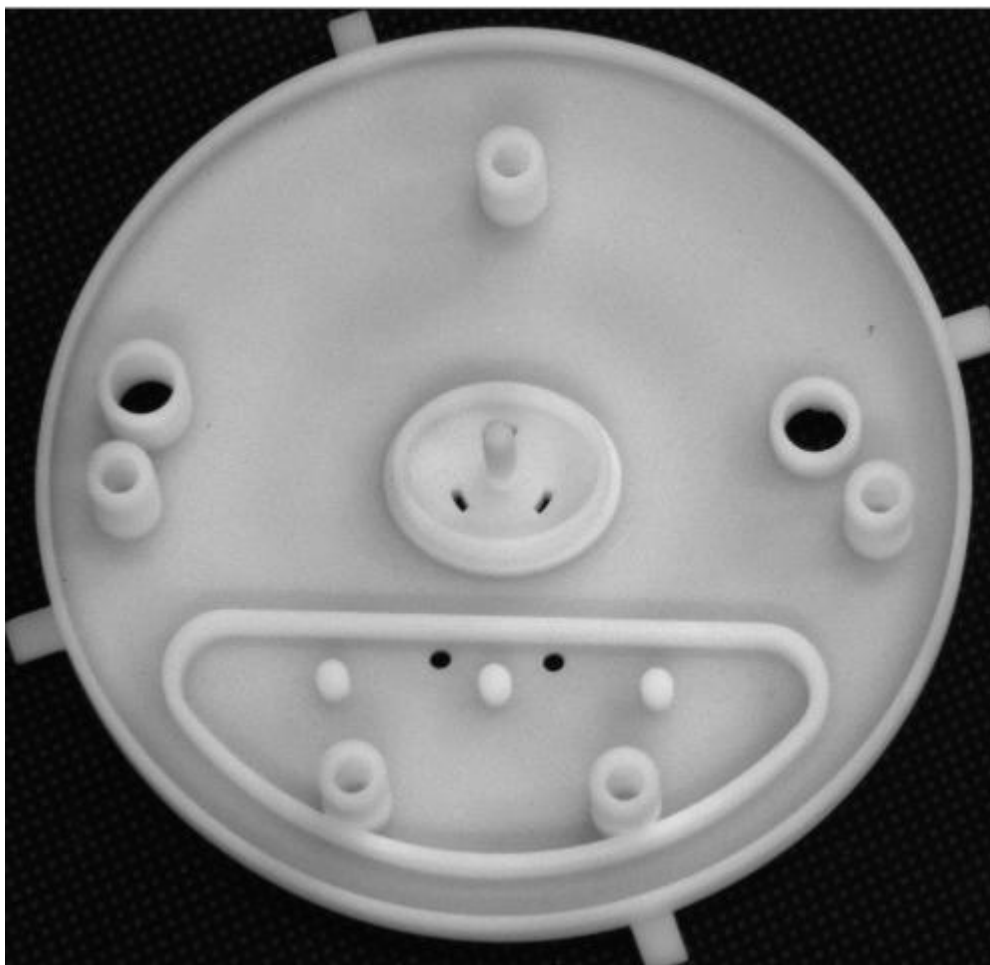


图 3.3 权值法灰度化处理之后

(2) 平均值法：这种灰度化方法是使得每个像素中 RGB 通道的值均等于三者的平均值，用这种方法灰度化会形成亮度较为均衡的灰度图。公式如下：

$$Gray = (R + G + B) / 3 \quad (3-3)$$

将公式转化为 C++ 代码为：

```
float R, G, B;
for (int y = 0; y < src.rows; y++)
{
    uchar* data = src0.ptr<uchar>(y);
    for (int x = 0; x < src.cols; x++)
    {
        B = src.at<Vec3b>(y, x)[0];
        G = src.at<Vec3b>(y, x)[1];
        R = src.at<Vec3b>(y, x)[2];
        data[x] = (int)(R + G + B) / 3; // 利用公式计算灰度值 (加权平均法)
    }
}
```

图 3.4 均值法处理公式代码



处理后的灰度图像如下:

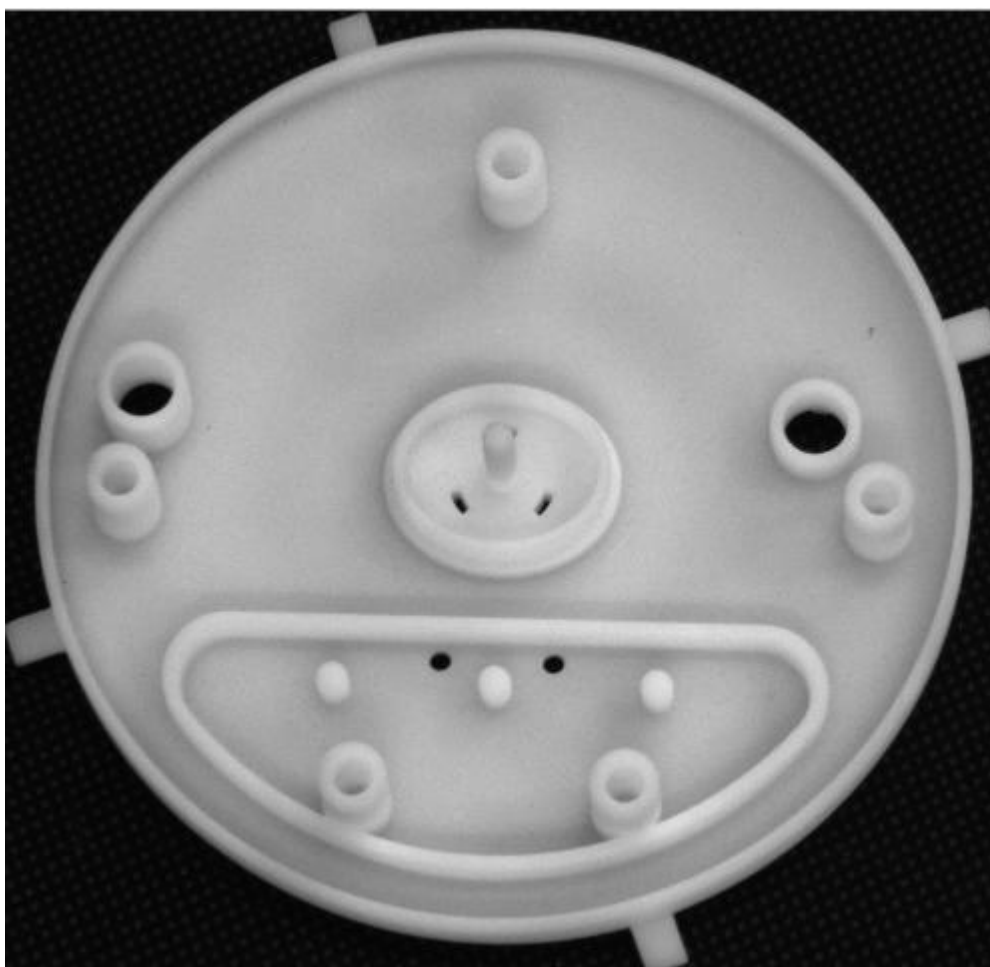


图 3.5 均值法灰度化处理后的图像

(3) 最大值法: 这种方法将 RGB 各通道的灰度值均等于三者中最大的那个值, 这样会形成亮度很高的图像。采用公式如下:

$$Gray = \max(R + G + B) \quad (3-4)$$

将公式转化为 C++ 代码为:

```
float R, G, B;
for (int y = 0; y < src.rows; y++)
{
    uchar* data = src0.ptr<uchar>(y);
    for (int x = 0; x < src.cols; x++)
    {
        B = src.at<Vec3b>(y, x)[0];
        G = src.at<Vec3b>(y, x)[1];
        R = src.at<Vec3b>(y, x)[2];
        data[x] = max(R, G, B); // 利用公式计算灰度值 (加权平均法)
    }
}
```

图 3.6 最大值法处理公式代码



处理后的灰度图像如下：

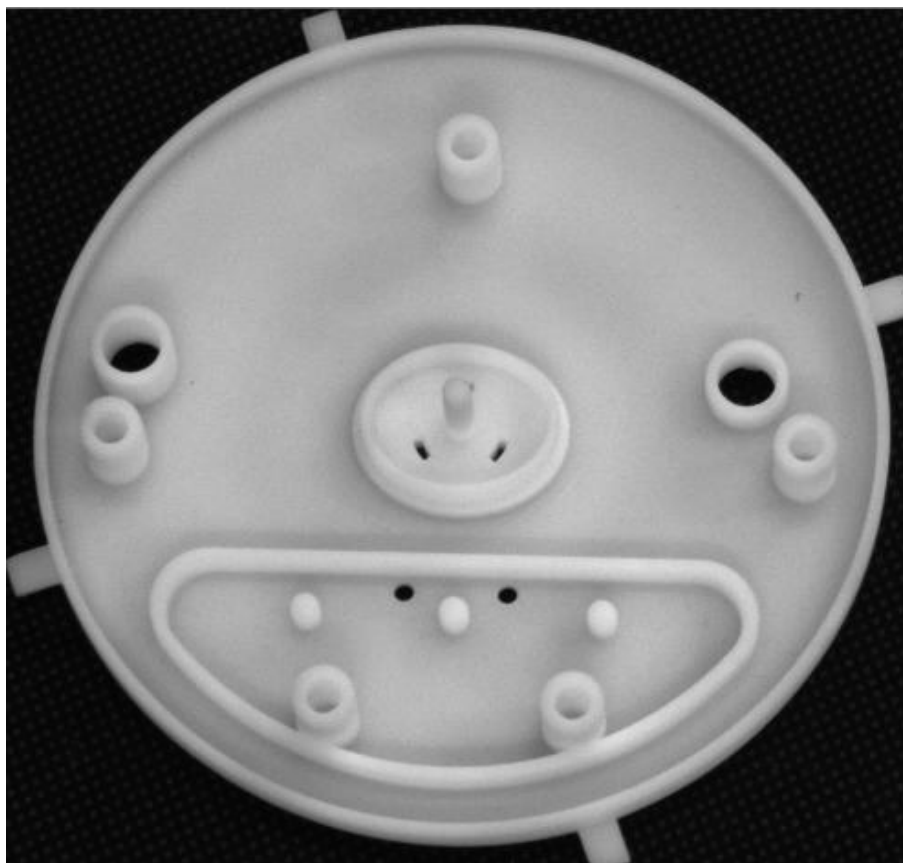


图 3.7 最大值法灰度化处理后的图像

### 3.1.3 灰度化算法的比较与选择

通过不同的灰度化方法我们获得了不同方法处理后的工件灰度图，但由于本来所使用的工件原图即以黑白色为主色调，故不同方法处理后的结果的差别对于肉眼观察并不是特别明显。因此，我们可选择一张色彩较为艳丽的图片进行不同权值的灰度化以获得更好的结果，如下图所示：



图 3.8 用于对比不同灰度化方法的彩色图

上图经过处理后获得的结果如图所示：

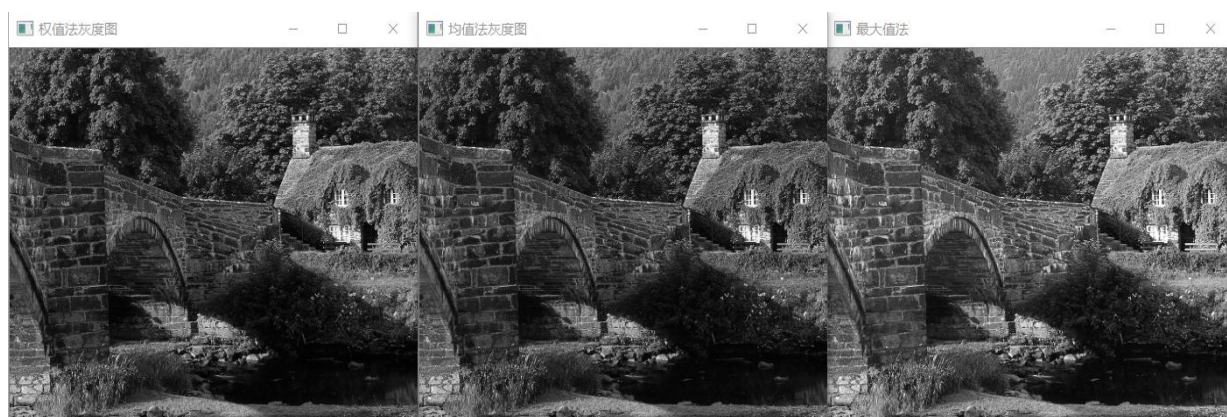


图 3.9 三种方法对彩色图的处理结果对比

不难看出，最大值法获得的灰度图亮度较大，这也意味这图像像素之间亮度的差异变小，即对比度的降低。这一点是不利于后面的边缘检测的。而均值法获得的图像较暗，且受图像自身的色彩分布影响较大，因此在处理不同的图像时获得的结果并不稳定。相比之下，权值法获得的图像亮度适中，且因为使用了固定的权值，不会受到图像自身色彩亮度的太多影响，结果较为稳定，故被选用为本文所述算法中的灰度化处理方案。

## 3.2 图像对比度增强

### 3.2.1 图像对比度增强的基本原理

对比度是图像处理中非常重要的一项指标，它指的是图像的不同像素区域在亮度等特征值上面的差异。符合实际且具有较高对比度的图像能够帮助研究者和开发者编写的算法更好地区分物体与背景环境、物体与物体之间的边界以及物体自身不同部分的边界。在灰度化的部分我们已经提到了使得亮度过强的最大值法和亮度不稳定的均值法因为不能构建较为优良的对比度而不被我们采用。经过权值法灰度化的图像虽然已经具备了一定的对比度，但是为了进一步提高软件的识别准确度，我们需要再次加强灰度化图像的对比度。此处要用到图像增强算法。

图像增强算法是一种简便而又常用的方法。它在保证所有像素位置关系不变的情况下，修改每一个像素点的灰度，使得数值高者变得更高，数值低者变得更低，是一种输入像素与输出像素之间点对点的运算，因此又被称之为点运算。设输入图像与输出像素的灰度分别为  $f(x,y)$ ， $g(x,y)$ ，图像增强则可以被表示为如下的函数：

$$g(x,y) = T(f(x,y)) \quad (3-5)$$

映射关系  $T$  决定了输出与输入像素的灰度值之间关系。

常见的对比度增强算法有两种，分别是线性变换和直方图均衡化。以下对这两种算法的原理以及处理效果作陈述。

### 3.2.2 两种对比度增强的方法介绍

### (1) 线性变换

线性变换是图像增强算法的一个重要方面，它通过设置阈值以及操作特定像素处理函数，对像素进行点对点操作，增大了目标图像的对比度与清晰度，使得特征变得易于提取。与之前类似，我们设输入和输出像素分别为  $f(x,y)$ ， $g(x,y)$ ，其中的  $x,y$  表示像素所在的坐标位置。线性变换的计算公式如下：

$$G(x,y) = \begin{cases} a' + \frac{b' - a'}{b - a} [f(i,j) - a] & a \leq f(i,j) \leq b \\ a' & f(i,j) < a \\ b' & f(i,j) > b \end{cases} \quad (3-6)$$

其中， $[a,b]$  为处理之前的像素的灰度所在域， $[a',b']$  为处理之后的像素灰度所在域。函数图如下

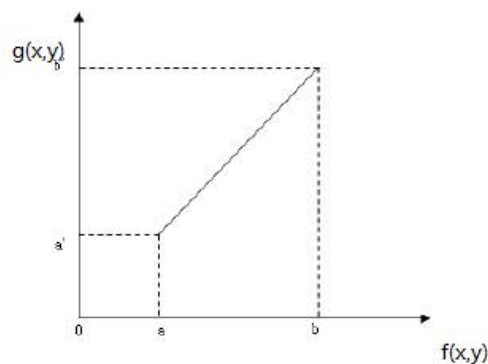


图 3.10 线性变换的函数分布图

在过高或过强亮度的图像中，所以像素的灰度会分布在一个局限的值域内。此时采用线性变换，将大大提高图像的层次感，强化对比度，有利于下一步的特征检测与提取。在编写具体的算法实现中，为了简化算法，提升算法的实时性和计算效率，我们根据经验将直线的斜率设置为 1.3，B 值设置为 60，这样避免了每次对新的图片进行处理时都要进行大量运算，而实际取得的效果如下：

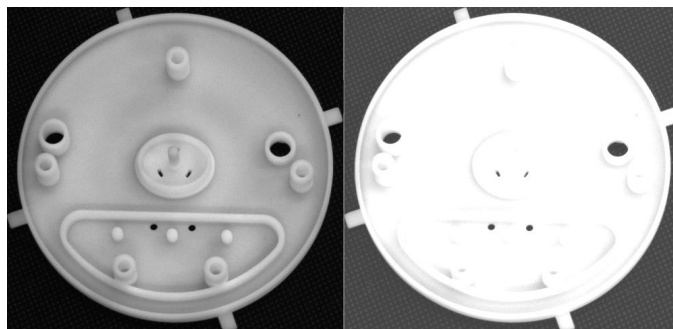


图 3.11 线性变换之前的灰度图与之后的处理结果

可以看到物体自身与背景之间的对比度大大增强，边缘变得更加明显。

## (2) 直方图均衡化

线性变换解决的时图像亮度分布过窄的问题，但是很多时候图像的缺陷不仅仅来自于灰度的分布区域过小，还可能来自于某些灰度值的使用频率远高于其他灰度值。即灰度值的分布不够均衡。如下图所示：

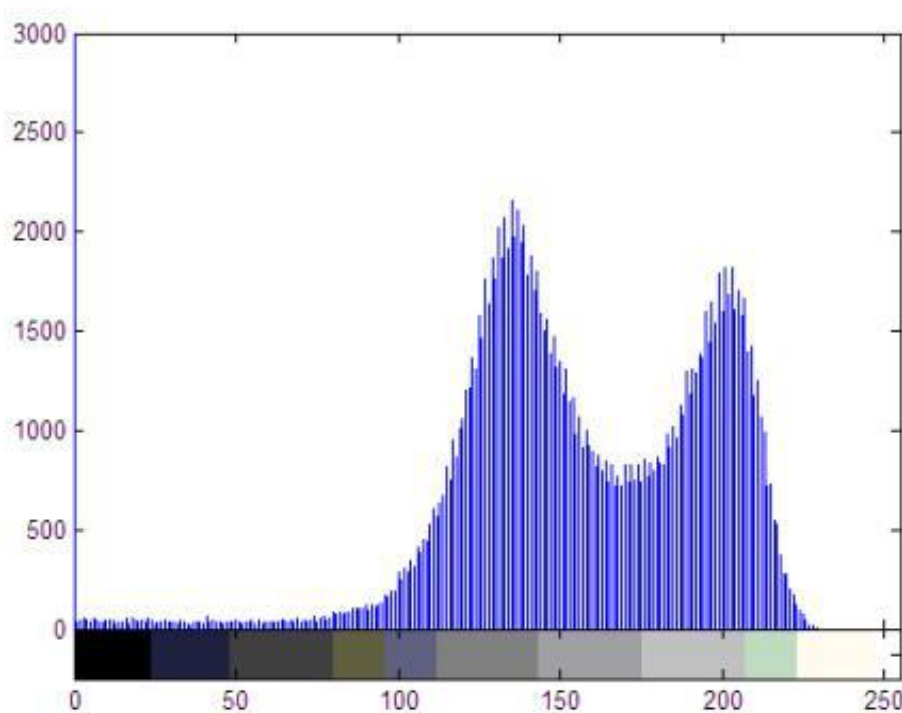


图 3.12 某图像的直方图分布

上图的直方图表示该图形在灰度值 120 和 200 处的像素过多，即这两个灰度的使用强度较大，而较明和较暗的灰度值使用较少。图像的灰度分布直方图统计了每个灰度级别的出现频率，反应了整幅图片的灰度分布和明暗对比，也是为下一步的图像处理进行评价的重要指标。在评价一幅用于图像处理的图片时，我们认为对所有的灰度强度够均衡得使用才是一幅质量较佳得图像。而作为图像增强技术之一的直方图均衡化，就是利用算法来使得图像的灰度使用频率更加均衡化的手段。

直方图均衡化的最终目的是让原图像的灰度分布更加均匀，而达到这种目的的手段如下：读取出原始图像各像素的灰度等级  $S_k$ ，然后统计该灰度等级的出现频率  $n_k$ ，利用下式来获得原图像的分布直方图： $P(s_k) = \frac{n_k}{n}$ ；之后利用以下公式进行直方图均衡化：

$$t_k = EH(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p(s_i) \quad 0 \leq s_k \leq 1, k = 0, 1, \dots, L-1 \quad (3-7)$$



即利用所处理像素之前的所有像素作为分母平均当前的像素值，以避免过为陡峭的变化率。当以此种方法递推完所有像素之后，极值点和均值点之间的差异变小，灰度值的分布也趋向于均匀。OpenCV 提供了方便的封装函数供开发者使用。直接调用 `equalizeHist` 函数来对之前的灰度图进行处理，得到了如下的对比图：

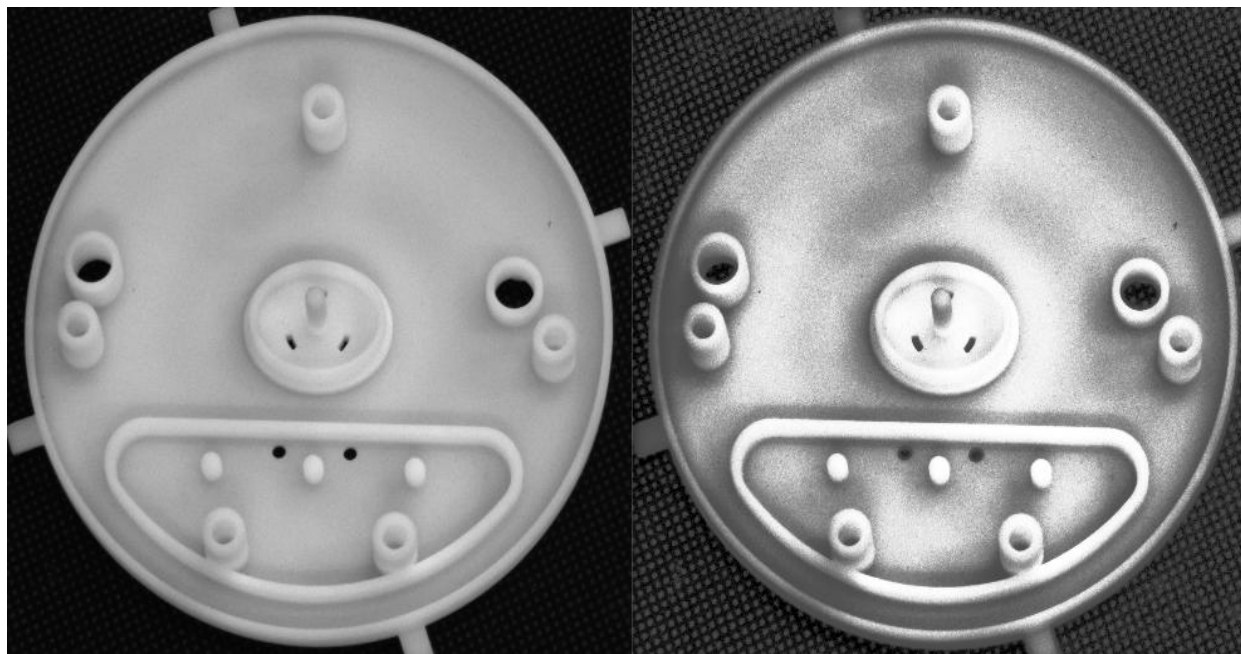


图 3.13 直方图均衡化前后图像的对比

可以看出，处理后图像的亮度分布更显均匀，过暗的背景和过亮的工件之间亮度的差异有所减小，图像变得更加平稳。但同时工件与背景之间的边缘也变得模糊，对后续的边缘检测有些影响。

### 3.3 图像的噪声抑制

#### 3.3.1 图像滤波的基本原理

机器视觉处理所用的图像一般是在自然光的条件下采集而来，通常来讲会产生失真、噪声等问题，这是由于光学系统、处理算法、自然光照等条件不可控地会出现随机性的问题。而在图像处理的过程中，过多的噪声会影响算法的判断。因此，对图像的去噪处理是不可或缺的。

滤波是去噪的一种重要手段，其手段是选择性地去除图像中某些极端的异常值，提取出其他方面的内容，使得图像中重要的信息得以保留下来。在极端情况（诸如仪器损坏导致某一区域出现大面积失真）以外的情况中，噪声点一般是随机分布的，这种情况下利用周围相邻像素的平均或加权平均值作为新的灰度值就可以有效消除噪声。这种做法本质上相当于建立了一个通道，它只许低频信号通过，而高频信号被检测出后则会被去除，实现对异常点的过滤。但同时，图像中的边缘轮廓也属于变化陡峭的高频信号，因此滤波会在不同程度上减弱图像中边缘信号的强度，使得边缘模糊化。所以在选择滤波方案时，要考

虑两个方面：既要使得图像中影响后续处理的大部分噪声点被过滤掉，同时也要使得图像中的边缘依然能够保持足够高的变化强度。图像滤波的方法分为频域法和空域法两大类。空域法是指图像内不同区域内有不同的灰度等级分布，根据灰度强度来把每个像素与周围邻域进行对比去除特征点。更为常用的是频域法，它是指通过分析像素的灰度值的变化频率来描述图像的灰度值分布。它把图像分解为从低频到高频的频率成分，像素灰度值变化慢的区域包含低频率较多，而变化快的区域则包含更多高频率。由于图像是二维的，所以对于它的频率分析有水平和垂直两个方向。常用的滤波函数分为中值滤波、均值滤波、高斯滤波、双边滤波等。在下一节将着重比较这几种不同的滤波函数在原理和处理结果上的区别。

### 3.3.2 四种滤波的方法介绍

如果滤波器使用周围相邻像素的加权均值来替换原有像素，我们就称之为线性滤波器。把领域中每个像素位置对应的权值放在一个位置矩阵内，用来表示滤波器的权值分布，这被成为滤波器的内核。内核的规模也可以人为确定，可以是  $N \times N$  的矩阵，也可以是  $M \times N$  的矩阵。如下图是一个  $3 \times 3$  的内核，每个内核的位点所占权重是相同的：

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

以下将介绍两种线性滤波器：

#### (1) 均值滤波

如上面内核的示意图所示，即每个像素位点所占的权重是相同的，使用周围像素的均值取代原有像素值。它属于一种空域处理的方法。处理公式如下：

$$I_{OUT} = \sum_i \sum_j I_m(x-i, y-j) K(i, j) \quad (3-8)$$

在这个算法运算过程中，位于  $(x, y)$  的像素与  $K$  内核的中心点对齐。在算法实现的过程中我们将滤波器尺寸定位  $5 \times 5$ ，对之前线性变换之后的图像进行处理，得到的结果如图 3.14 所示。

#### (2) 高斯滤波

高斯滤波同样属于线性滤波，但不同于均值滤波，像素所对应的权值与它到目标像素之间的距离正相关。用它来处理高斯噪声是很合适的。对于二维图像来说，通常使用如下的函数：

$$\frac{g[i,j]}{c} = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (3-9)$$

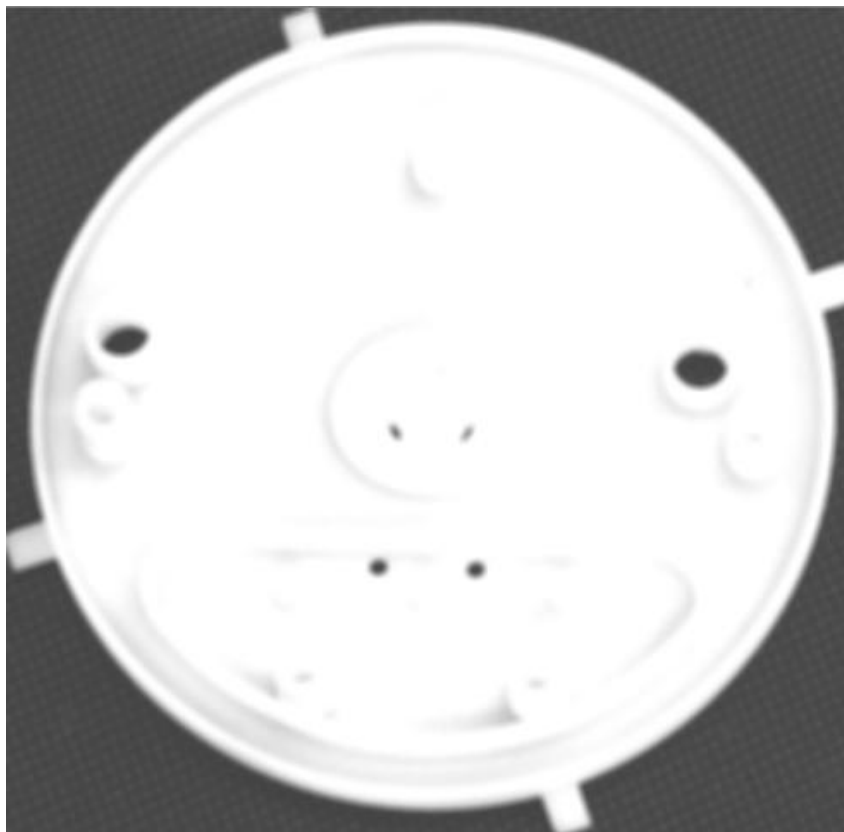


图 3.14 均值滤波处理后的图像

高斯滤波器的一些特点有利于我们进行滤波处理，如旋转对称性，单值函数性，高斯函数的傅立叶变换频谱是单半，高斯滤波器的宽度有参数  $\sigma$  表征，高斯函数具有可分离性等等。离中心点越远的像素的影响因子就越小，这样的处理逻辑使得像素之间的过渡更加平滑，减少了突变的成分。但从另一方面来将，高斯滤波并没有消除全部的噪音。多数实验也证明，在消除随机噪声上高斯滤波具有很好的效果，但同时由于使得图像变得过于平滑，图像中的一些特征也会变得更加模糊。在实现算法的过程中，我们依旧设定滤波器大小为  $5 \times 5$ ，方便与之前的方法进行比较，处理后的结果如图 3.15 所示。

### (3) 中值滤波

中值滤波不同于上述两个滤波算法，它属于非线性滤波，非常有利于消除椒盐类的噪声。中值滤波把当前像素和他的邻域组成一个集合，然后计算出这个集合的中间值，作为当前像素的值。它基于的是这样一种理论：噪点往往是局部像素值极大的点，它们对应的像素点极少，在与周围像素构成集合并排列计算中间值时，噪点很难成为中间值，因此被排除掉。简单的线性滤波器很难排除这种极大值的影响，而相比之下这就是中值滤波的优点。这种算法对于图像的干扰和扫描噪声很有效，但是对一些细节点比较多，比如角点、

线条多的图像则会产生部分失真的处理效果。在本文所处理的工件中，所有线条都较为圆润，因此是中值滤波适用的情况。其算法实现与公式详解如下：

在目标图像上取一个一维向量  $f_1, f_2 \dots f_n$ ，算法处理的尺度大小为奇数  $m$ ，从输入向量中以此取出相应个数的元素： $f_{i-v} \dots f_i \dots f_{i+v}$ ，其中  $f$  为处理窗口的中心值， $v$  为中值所在的序列号，再将这  $m$  个点的按数值大小排列，取其中间值作为滤波输出。中值滤波的表达式为：

$$Y_i = \text{med}\{f_{i-v}, \dots, f_i, \dots, f_{i+v}\} \quad i \in S, v = (m-1)/2 \quad (3-10)$$

对数字图像进行中值滤波，实质上对二维序列  $\{X_{ij}\}$  的中值滤波，滤波窗口也是二维的，但是窗口的像素分布可以由算法设计者来确定。

二维的中值滤波函数可以表示为：

$$Y_{ij} = \sum_A \text{med}\{X_{ij}\} \quad (3-11)$$

处理后的图像如下图 3.16 所示。

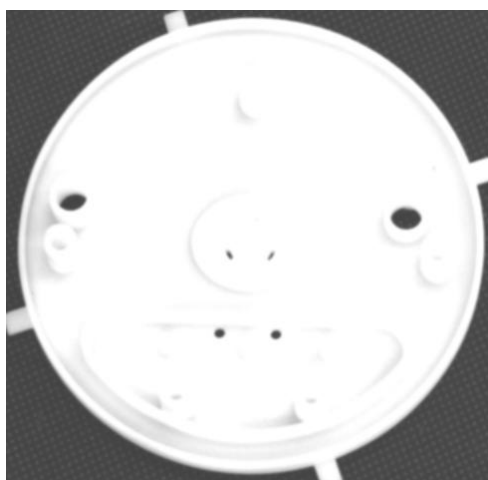


图 3.15 高斯滤波处理后的图像

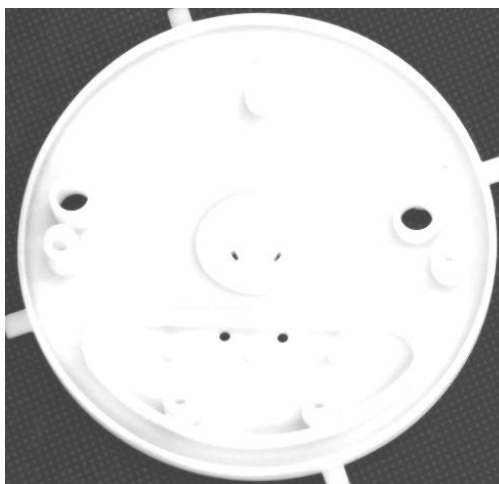


图 3.16 中值滤波处理后的图像



#### (4) 双边滤波

双边滤波也使用了非线性滤波器，能够实现锐化物体边界的同时进行有效降噪的效果。类似于其他线性滤波器，双边滤波也是采用赋予不同位点像素不同加权值来进行平均的方法，不同的像素亮度值的加权值代表某个像素的强度。不同的是，所用的加权平均值基于高斯分布，而并不是给定值。此外，双边滤波的权值不仅考虑了核子各部分到中心目标点的欧氏距离（普通的高斯低通道滤波只考虑和中心像素之间的距离），还把像素范围域中的辐射范围之间的差别（如卷积核中各像素与中心像素之间类似度、颜色的强度，深度距离）纳入了考虑范畴，在计算中心像素的时候同时将空域与像素范围同时进行考量。

总结起来，其基本思路是同时考虑要处理的像素点的空域信息和值域信息，是一种组合起来的滤波方式，因此叫做 双边，即同时考虑两边的数据。首先，图像滤波中常见的情况是：图像在空间中变化缓慢，因此相邻的像素点会更相近；在图像的边缘处变化剧烈，此时如果用周围像素平均化来处理则得到的结果中边界必然是模糊的，不利于下一步的算法处理。那么因此，考虑再把像素点的值的高低纳入考虑范围，因为边缘两侧的点像素值变化剧烈。这样子使得其加权的时候权值得差别很大，从而边缘不那么容易被模糊掉。可以理解为先根据像素 值的大小分割出要进行处理图像区域，再给该点所属的类别分配权重，然后进行邻域加权平均运算，得到最终处理的结果。在双边滤波中，两个要素即：临近度和相似度，分别处理空域和值域信息。其数学表达方式相近，如下：

$$h(x) = k_d^{-1}(x) \int \int_{-\infty-\infty}^{\infty \infty} f(\xi) c(\xi, x) d\xi \quad (3-12)$$

$$h(x) = k_r^{-1}(x) \int \int_{-\infty-\infty}^{\infty \infty} f(\xi) s(f(\xi), f(x)) d\xi \quad (3-13)$$

其中积分号前面为归一化因子，这里考虑对所有的像素点进行加权，c 和 s 是分别是二者的特征函数函数：

$$c(\xi, x) = e^{-\frac{1}{2} \left( \frac{d(\xi, x)}{\sigma_d} \right)^2} \quad (3-14)$$

其中：

$$d(\xi, x) = d(\xi - x) = \|\xi - x\| \quad (3-15)$$

$$s(\xi, x) = e^{-\frac{1}{2} \left( \frac{\sigma(f(\xi), f(x))}{\sigma} \right)^2} \quad (3-16)$$

其中

$$\delta(\phi, x) = d(\xi - x) = \|\xi - x\| \quad (3-17)$$

x 是要处理的点，f(x) 代表该点的像素值。h(x) 处理后的像素点像素值。合并后最终的滤波函数为：

$$h(x) = k_r^{-1}(x) \int \int_{-\infty-\infty}^{\infty \infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi \quad (3-18)$$

处理后的图像如下：

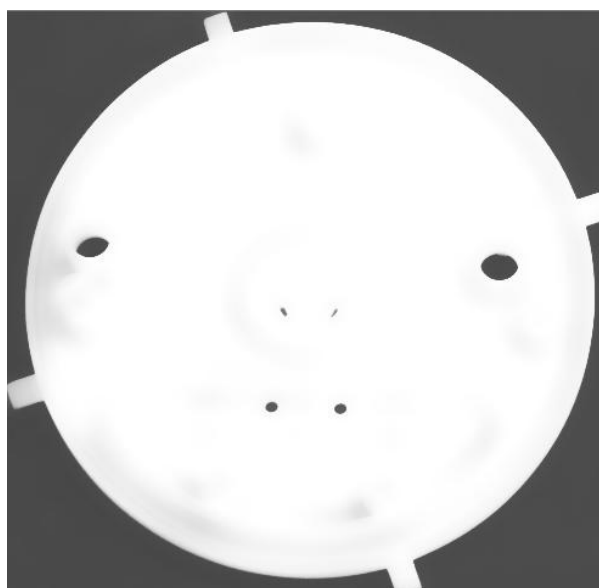


图 3.17 双边滤波处理后的图像

### 3.3.3 四种滤波的结果对比

在得到上述四种滤波算法的结果后，我们将其放在一起对比其效果，并选取最优的对于处理本工件来说最优的算法。

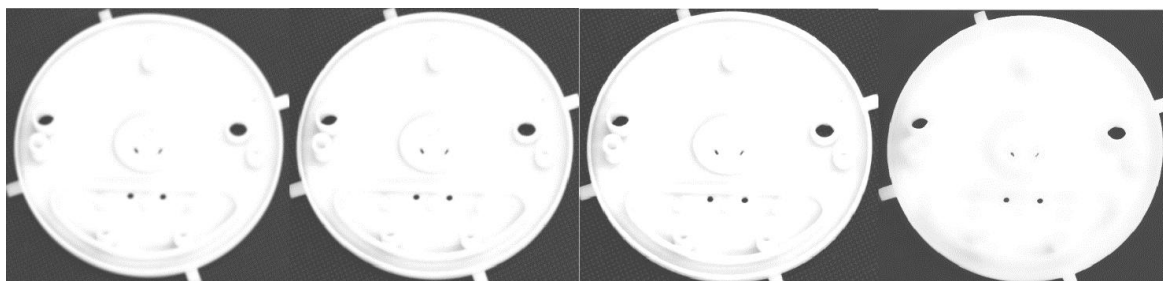


图 3.18 四种滤波处理后的图像

上图中的处理后图像从左到右依次对应之前先后使用的滤波算法。将图像放在一起分析对比不难发现，均值滤波和高斯滤波由于使用了线性滤波器，图像整体变得均匀，虽然去除了大量噪声，但同时也模糊了边界，使得边检变得难以识别；中值滤波和双边滤波由于不是单纯的线性滤波。所以在去除大量不相关的噪点的同时，还清晰化了图形的边界，增强了图像的对比度。这种效果在第四幅双边滤波的图像上最为明显，使得图像简单化为基本的几何形状和特征，大大简化了后续图像的处理，但同时使得图像的锐度过高，消除了很多小的形状特征。同时，该方法所耗费的计算时间明显地长于其他算法。综合以上实验结果，我们选用中值滤波和双边滤波作为比较好的方案。

## 第四章 图像的特征提取

### 4.1 边缘检测算法

#### 4.1.1 边缘检测的基本原理

边缘检测顾名思义，就是利用算法检测图像中物体的边缘，实现图像分割和目标区域提取等目的，是图像分析的重要手段。边缘检测是本文中的核心处理算法，关系到最终结果的准确度，在工程中占有十分重要的目的。

在图像中，物体与背景之间的边界被成为边缘，是图像亮度变化最为显著的部分。主要存在于物体与背景、物体与物体、不同区域之间，用来分析纹理、分割图像、提取形状特征。将图像数值化后，边缘往往是图像灰度值一阶导数的局部极值，即变化梯度最大值的区域。我们常常用离散化的梯度函数来逼近这些峰值，达到检测的目的。

在算法处理中，我们常常用一些滤波器来反响放大差异值，实现突出变化点、削弱背景区域的目的，使得边界区域进一步凸显。这种反向放大的滤波器被称为算子，常用的算子有 sobel 算子、canny 算子等。

#### 4.1.2 四种边缘检测算法

##### (1) sobel 算子

sobel 算子是一种典型的用于边缘检测的线性滤波器，基于水平和垂直方向两个 3\*3 的内核，结构如下：

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

这两个内核在两个方向上卷积获得响应，二者之比被称作梯度。如果把图像看作二维函数，sobel 算子就是图像在垂直和水平方便变化的速度。作为梯度的 sobel 算子是一个二维向量，包含水平垂直两个方向的函数的一阶导数：

$$grad(I) = \left[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T \quad (4-1)$$

Sobel 算子在水平和垂直方向计算像素值得差分，得到像素梯度得近似值。它在像素周围得一定范围内进行运算，以减少噪声带来得影响。由于梯度是一个二维向量，具有模

和方向。它得模表示变化的幅度，计算时通常被当作使用欧几里和模，但为了减少计算量，这里我们使用 X、Y 方向模的绝对值之和来表示，称为 L1 模。同时，梯度向量的方向与像素值变化最快的地方一致，在边缘识别中，这意味着该方向和边缘垂直。因此，在边缘检测中我们利用的是作为二维向量的 sobel 算子的属性来确定边界。

对梯度进行阈值化，可以得到二值分布边缘图像。阈值如果设定得过高就会导致边界不完整，不能整个保留下来；如果设定得过低就会留下太多不相干得的边缘，影响算法的下一步判断。因此，找到一个合适的阈值十分重要。以下是 sobel 对不同滤波处理后的图像进行处理的结果：

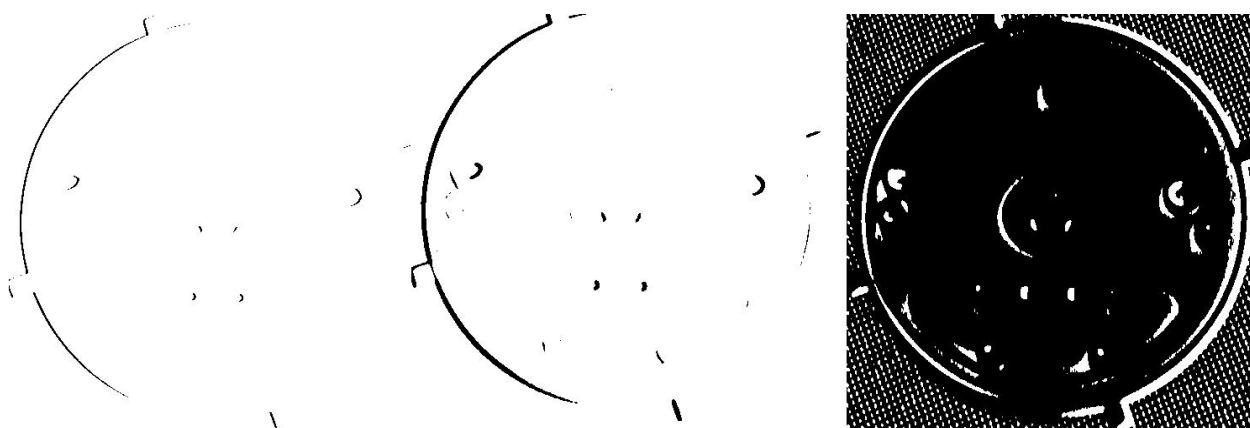


图 4.1 sobel 算子对不同滤波处理图像的输出结果

```
// sobel算子的模
Mat sobelX, sobelY;
Sobel(dst3, sobelX, CV_16S, 1, 0, 3, 0.4, 128);
Sobel(dst3, sobelY, CV_16S, 1, 0, 3, 0.4, 128);
Mat sobel;
sobel = abs(sobelX) + abs(sobelY);
//过滤
double sobmin, sobmax;
cv::minMaxLoc(sobel, &sobmin, &sobmax);
Mat sobelImage;
sobel.convertTo(sobelImage, CV_8U, -255./ sobmax, 255);
//阈值化
Mat sobelthresholded;
cv::threshold(sobelImage, sobelthresholded, 125, 255, cv::THRESH_BINARY);
dst4 = sobelthresholded;
imshow("Sobel算子", sobelthresholded);
waitKey(0);
destroyWindow("Sobel算子");
```

图 4.2 sobel 算子算法实现

可以看出, 对不同的滤波图像 sobel 算子算法可能需要不同的阈值来处理, 在处理过程中很容易使得边缘过于细小, 产生断裂; 或者使得边缘过于厚重, 识别出过多的不必要特征。

## (2) Canny 算子

Canny 算子是通常基于 sobel 算子, 利用高低两个不同的阈值来判断当下的点是否属于轮廓。这样就避免了 sobel 算子使用一个阈值时会出出现无法兼顾过滤不相干元素和保证边缘完整性的状况。选择低阈值时, 保证其能包含属于重要边界的边缘像素。而较高的阈值的作用就是界定重要轮廓的边缘, 排除异常的几何特征。具体做法是分别使用高低两个阈值处理图像, 得到两个结果图像。之后结合这两张图的边缘生成最优的轮廓检测图。具体做法是在低阈值的边缘分布图上只保留具有连续路径点的边缘, 同时把那些边缘点连接到属于高阈值的边缘分布图上。这样, 最后生成的图像上边缘都被保留下来, 而不相关的杂乱边缘都被过滤掉。只要选择了适当的阈值, 就可以获得较高质量的轮廓。此外, canny 算子还包括了优化边缘分布图的质量的算法。在进行滞后阈值化之前, 只有梯度幅度是梯度方向上最大的点, 该对应的边缘点才会被保留。Canny 算子的识别结果如下:

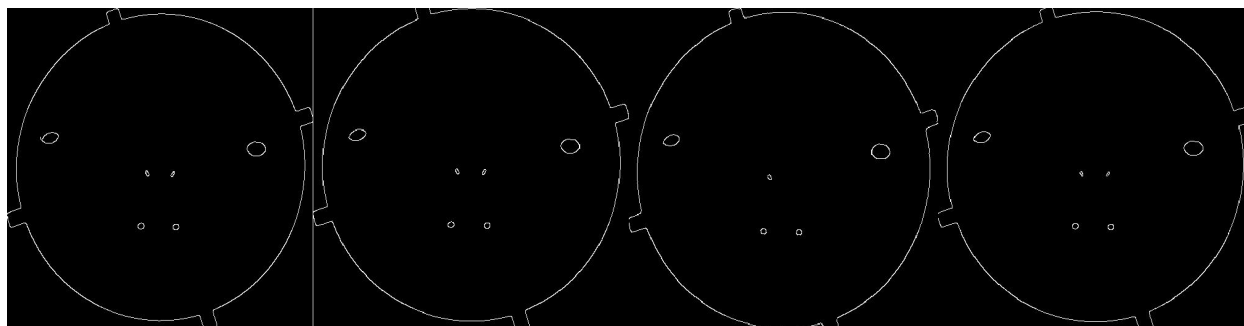


图 4.3 sobel 算子对不同滤波图形的处理结果

可以看出, 无论对于哪种滤波方式, canny 算子都能够稳定地检测出其边缘, 而且图像较为干净, 没有杂乱的边界形成干扰。相比于 sobel 算子需要针对不同的图像来调整不同的阈值, canny 算子具有更好的适应性和更高的结果准确度。

## (3) Hough 圆检测

Hough 变换是一种常用于检测几何特征的经典算法, 起初用于检测直线, 后来也用于检测圆等其他几何结构。在本例中, 我们使用 Hough 变换来检测工件的圆轮廓和圆心。我们知道圆的方程为:

$$(x-a)^2 + (y-b)^2 = r^2 \quad (4-2)$$

此方程包含三个参数, 这样的圆映射到三个参数构成的坐标系中的集合中对应的是一条曲线。在 XY 坐标系中同一个圆被检测到的次数越多, 他们在三维坐标系中映射点的被统计次数就越多, 当大于设定的阈值的时候, 这个点所代表的圆就是检测出的圆。这里我们需要使用三个累加器来统计三个参数被标识的次数。但是累加器的个数越多, Hough 变

换的准确度越低。为了提高算法的准确度，在圆检测中加入了两轮筛选，第一轮使用二维累加器找出圆心的位置，第二轮筛选利用一维累加器找出圆的半径。

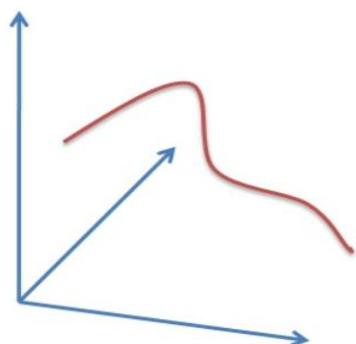


图 4.4 在三维空间三个参数构成的点形成曲线

得到存放圆的向量之后，迭代遍历该向量，在图像中画出这些圆。所处理之后的图像如下：

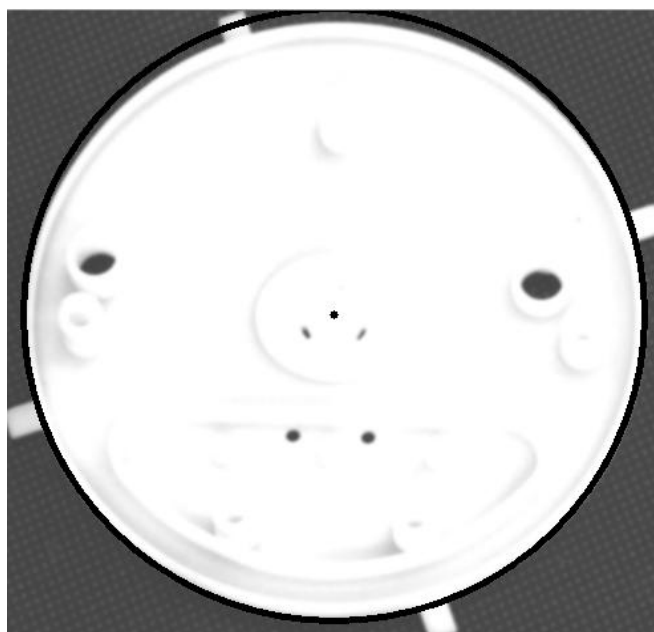


图 4.5 Hough 圆检测处理后的结果

从结果可以看出，算法较准确的定位出了圆心，且对于图像边缘的贴合是比较准确的，但是由于圆检测是在二维平面中进行的，而图像实际所处的位置和摄像机的角度等三维参数的影响存在，所以在平面上工件并不是一个严格规整的圆，而是近似于椭圆，所以 Hough 圆检测只是检测出了最接近的圆，但并不能贴合所有的边缘。即便如此，用此算法检测圆心和半径能够达到比较高的准确度，是一种比较好的方案。

## 4.2 角点检测算法

在提取特征点的过程中，对于角点的检测是一种非常常见的方法。角点是指边缘相交的区域，通常在它的四周像素值的变化非常剧烈。角点是一种二维特征，可以很容易地被精确定位。因此角点常常作为特征点成为判断与识别物体位姿的依据。

在本文中我们使用的是 Harris 角点检测方法，在这种算法中，通常会在要检测的目标点周围放置检测窗口，用来检测像素周围各个方向上强度值的变化幅度。得到所有计算方向上平均强度的变化值之后，进行比较，如果多个方向的变化之值都很高，就认为这个点是角点。通常我们用一个协方差矩阵来表示目标位点在各个方向上像素值强度的变化速率：

$$R = [u, v] \begin{bmatrix} \sum (\frac{\delta I}{\delta x})^2 & \sum \frac{\delta I}{\delta x} \frac{\delta I}{\delta y} \\ \sum \frac{\delta I}{\delta x} \frac{\delta I}{\delta y} & \sum (\frac{\delta I}{\delta y})^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (4-3)$$

这个协方差矩阵的两个特征值，分别表示最大平均强度值和水平方向的平均强度值变化。如果被检测点在边缘上，这两个特征值会出现一个大一个小的情况；如果两个都很大，则说明该位置在角点上。而判断大小是的手段是该矩阵的特征值必须都大于设定的阈值。在实现本课题的算法过程中，为了使得被选作特征点的角点一定能被选出来，我们设定了比较低的阈值，而后会根据欧几里得距离来进行筛选。我们选用边界效果最为良好处理的结果图像如下：

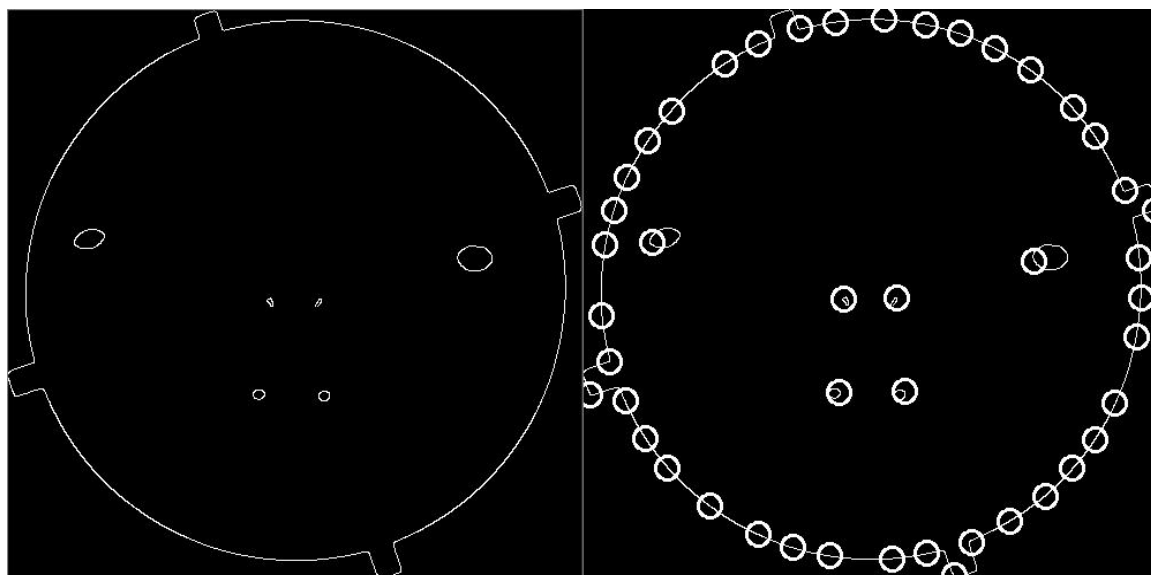


图 4.6 角点检测处理后的结果

可以看到由于设定了较低的阈值，所以图像中工件边缘上几乎所有的角点都被检测出来了，符合我们对算法的预期。

## 第五章 图像相对坐标读取与计算

### 5.1 圆心的相对坐标读取

在 Hough 检测的论述中我们提到由于摄影角度等三位空间位置关系对最终处理的图像有影响, 所以 Hough 圆检测绘制出的圆并不能完整贴合实际工件轮廓。但同时, 对于轮廓提取效果较好的 canny 算子实际的提取轮廓为椭圆, 再进行圆拟合后精度损失较高, 因此依旧采用有瑕疵的 Hough 检测来提取圆的圆心和半径, 使用的代码如下:

```
// 圆心相对坐标
//std::vector<cv::Vec3f>circles;
cv::HoughCircles(dst3, circles, CV_HOUGH_GRADIENT, 1, 50, 200, 100, 0, 0);
AllocConsole();
freopen("CONOUT$", "a+", stdout);
std::cout << "x      y" << std::endl;
for (size_t i = 0; i < circles.size(); i++)
{
    Point center(cvRound(circles[i][0]), cvRound(circles[i][1]));
    int radius = cvRound(circles[i][2]);
    // circle center
    circle(dst3, center, 3, Scalar(0, 255, 0), -1, 8, 0);
    // circle outline
    circle(dst3, center, radius, Scalar(0, 0, 255), 3, 8, 0);
    u3 = circles[i][0];
    u4 = circles[i][1];
    float x = circles[i][0]-x0;
    float y = circles[i][1]-y0;
    cout << x << "\t" << y << endl; //在控制台输出圆心坐标和半径
}
imshow("圆心检测", dst3);
waitKey(0);
destroyWindow("圆心检测");
```

图 5.1 圆心坐标与半径读取算法

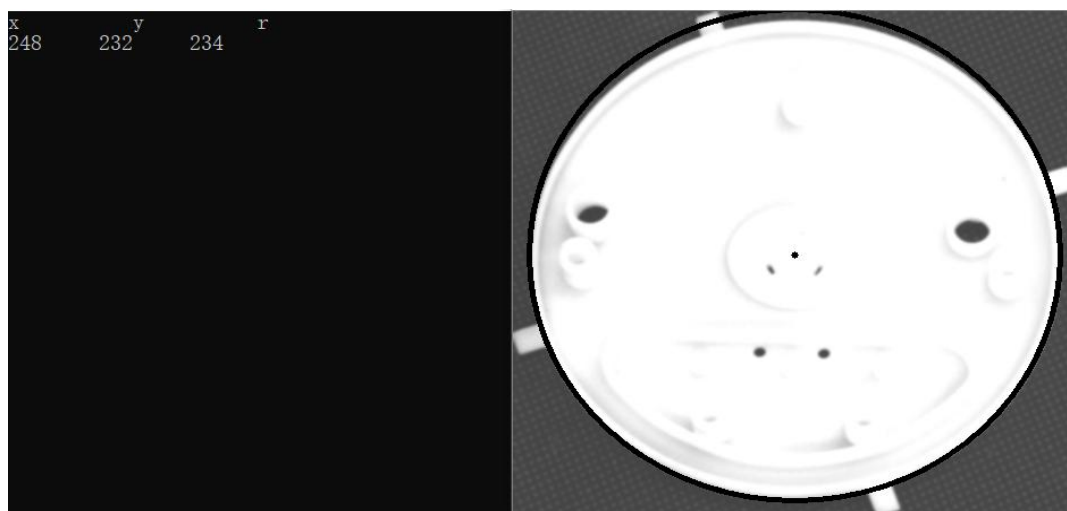


图 5.2 圆心检测算法运行结果

可以看到, 圆心坐标与拟合圆的半径在弹出的后台窗口内输出, 算法运行良好。



## 5.2 工件的相对偏转角计算

在设计相对偏转角算法时，我们采用了通过提取的特征点与圆心之间连线的夹角来进行计算。由于角点检测处理后提取了很多的特征点，所以首先需要对这些特征点进行筛选，选择在不同阈值情况下都较容易检测出、相对位置固定且较容易判断准的特征点作为依据。为此我们首先对 canny 算子提取后的边缘进行分析：

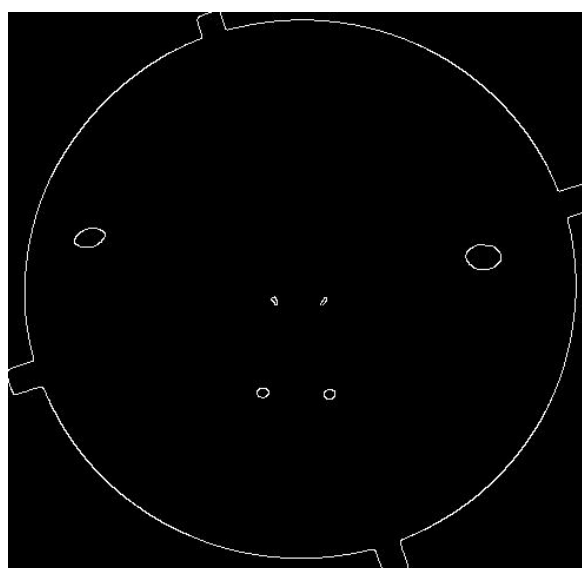


图 5.3 canny 算子处理后的结果

在对主要特征点进行标记并将两个算法处理前后的图像放在一起作对比，如下图所示：

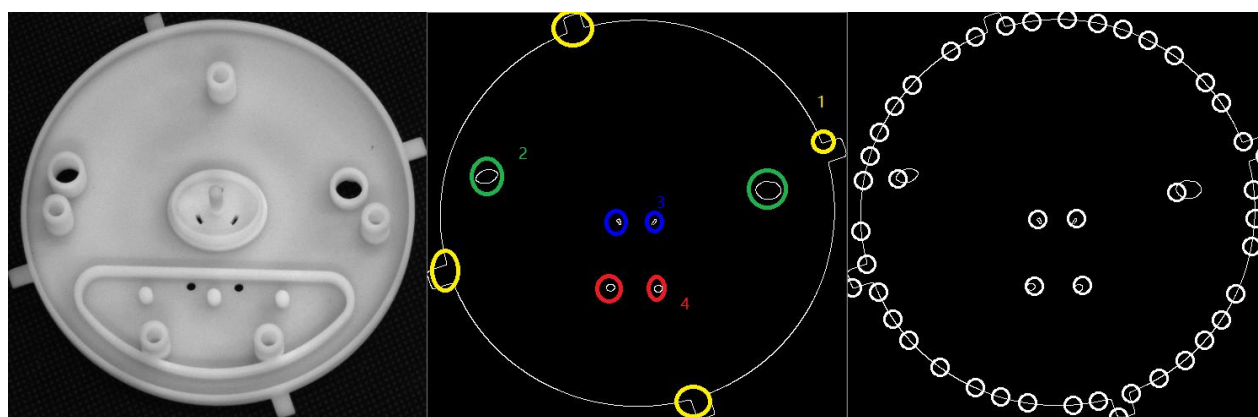


图 5.4 特征点的对比

根据位置不同对角点检测出的特征点进行分类，只要分为上图中间所示的四类。标号为 1 的各角点在工件边缘上，十分容易被检测出来，具有作为特征识别点的稳定性的要求。但同时，1 类角点具有定位价值的角点处在工件突出部分和圆形边缘相交的区域，该区域具有四个比较容易检测出来的点，且距离较近，在实际进行筛选的时候算法难

以区分，造成混淆之后会使得最终的定位精度产生较大的误差。同时，四个突出部分是对称的，并不具有特殊性，使用他们作为鉴定转角的特征点会增大算法鉴别的难度。

标号为 2 的各类角点处在工件圆内两个较大的椭圆孔洞上，相对于圆心具有特殊的位置，便于定位相对转角。同时，椭圆孔洞由于孔径较大的关系，与孔径后的背景形成了较强的对比度，使得在较低的阈值情况下依然可以被检测出，具有稳定性。但是，孔径过大使得第二类角点分布在较长的边界上，尤其是椭圆的两个尖点，所以检测出的角点位置并不是特别稳定，容易形成较大的误差。

标号为 3 的角点在实际的实验中是最易被检测出的。在高阈值的情况下第 3 类角点最先被提取出来。因此稳定性最强。但是通过观察最左边的工件彩图不难发现，第 3 类角点距离圆心出的突出部件太近，在实际拍摄时可能会由于角度问题被突出部分遮挡，因此很难把三个特征点同时识别出来。此外，由于三个特征点相对于圆心对称分布，不具有特殊性，也很难确定工件的相对转角。

标号为 4 的特征点由于孔径较小，在实验中并不是特别容易被检测出来，需要降低阈值才能实现稳定的检测结果。但是这两个孔洞是相对于圆心非对称分布的，因此具有特殊性，便于确定工件的相对转角。同时孔洞距离中心点处的突出部分较远，不存在被遮挡的情况。孔洞的直径较小，不会有过长的边缘，因此检测出的角点位置的偏差不会太大，对最终结果的精度影响会小很多。

综合上述的分析，确定用于检测相对转角的特征点采用标号 4 的两个角点。下面分析具体实现的算法。

为了确定工件的相对转角，我们需要找到一个独一无二的特征点。之前我们选定了标号 4 的两个特征点，但是因为这两个点是对称的，所以只有这两个点的坐标我们还是无法确定一个唯一的点。所以，我们对这两个点连线，取连线上的中点，这样的点可以是唯一的且确定的。我们以工件圆心点为原点建立笛卡尔坐标系，获得中心点的坐标后，该点与原点之间的连线在坐标系上会形成一个角度，该角度可以通过中点与圆心点之间的坐标运算可以获得。

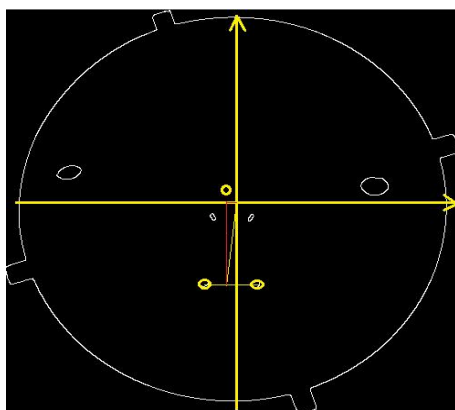


图 5.5 相对偏转角计算的算法示意图

在实现这个算法的时候，我们首先利用角点和原点的距离来筛选出被选为特征点的角点，这里我们使用欧式距离，即  $x$ 、 $y$  坐标的根号下平方和。在角点检测时我们可以在后台打印出所有检测出的角点的绝对坐标，如下所示：

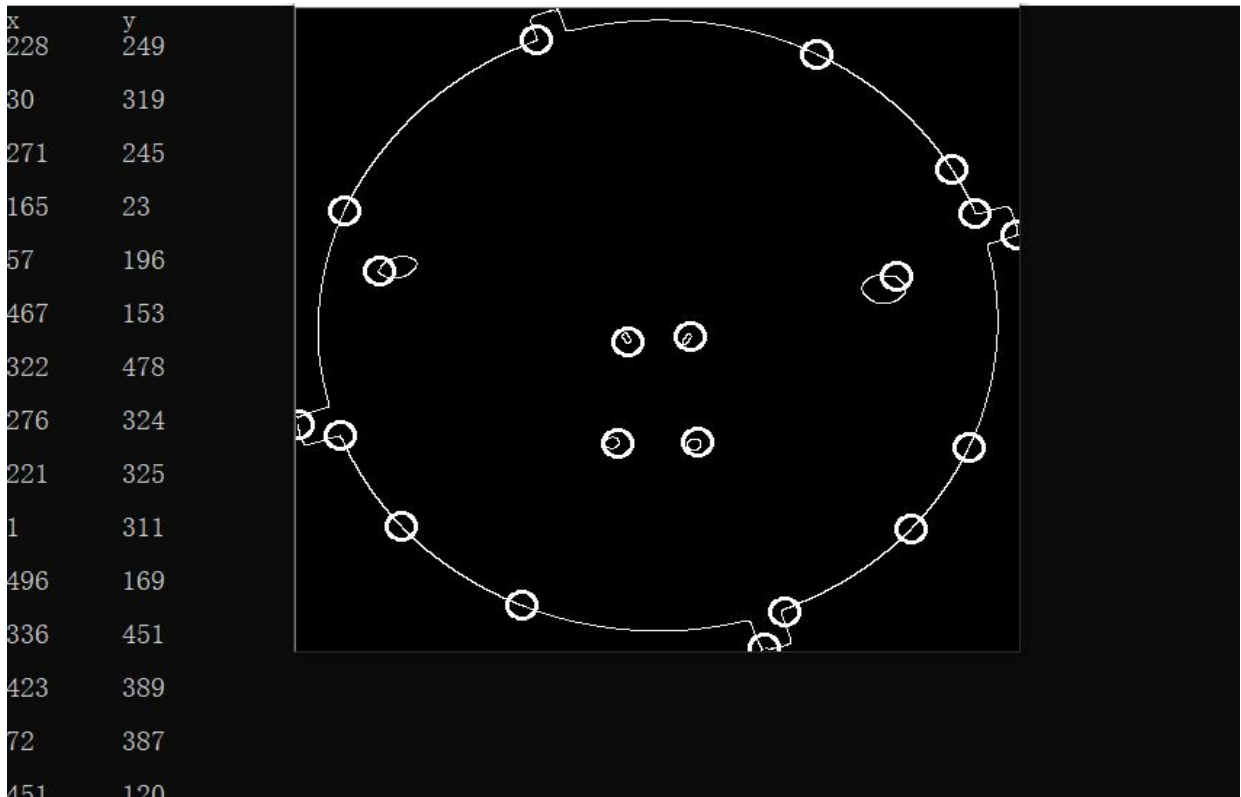


图 5.6 打印所有的角点的绝对坐标

在圆检测时我们已经有了目标圆的圆心绝对坐标，于是可以求解二者之间的欧式距离，设为  $L$ ， $L$  的输出如下：

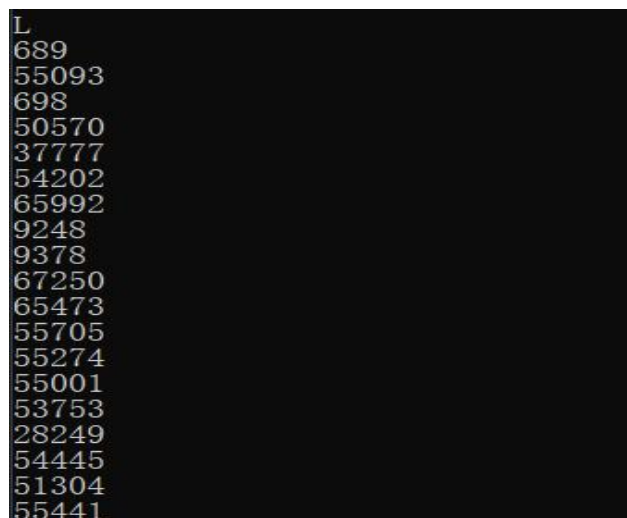


图 5.7 所有的角点相对于圆心的距离

通过观察发现所有角点根据里原点的距离大小，可以进行大致分类。这些数据在一定范围内聚集。通过这些特性和对样本图面的统计，我们可以大致确定目标特征点距离原点

的距离大致在 9000-12000，在不会包含其他范围内的角点的前提下，我们将范围扩大为 5000-15000，这样可以确保目标角点被检测出来。当角点坐标的 L 值在这个范围内时，我们将其 X、Y 坐标值分别存储在两个数组内，然后求其中点坐标  $(x, y)$ ，设原点坐标为  $(x_0, y_0)$ ，利用公式  $\theta = \text{atan2}((y - y_0), (x - x_0))$  即可得到相对转角。处理的代码如下：

```
float x = 0.5*(a[0] + a[1]);
float y = 0.5*(b[0] + b[1]);
float theta;
//cout << "x" << x << "\t" << "u3" << "\t" << u3 << "\t" << "y" << y
float m = x - u3;
float n = y - u4;
if (m > 0 && n <= 0)
{
    theta = atan(n / m) * 180 / 3.1415 - 90;
}
if (m <= 0 && n <= 0)
{
    theta = atan(n / m) * 180 / 3.1415 + 90;
}
if (m <= 0 && n > 0)
{
    theta = atan(n / m) * 180 / 3.1415 + 90;
}
if (m > 0 && n > 0)
{
    theta = atan(n / m) * 180 / 3.1415 - 90;
}
std::cout << "theta" << std::endl;
std::cout << theta << endl; //在控制台输出圆心坐标和半径
```

图 5.8 计算相对转角算法实现代码

### 5.3 处理结果展示与分析

通过实验验证，算法可以准确输出工件的相对转角，算法成功实现。部分结果如下。在结果展示中我们使用左上角的图片作为相对位置比较的模板，当工件相对模板顺时针旋转时转角为正，逆时针旋转时转角为负。X, Y 轴的正方向指向右方和下方。

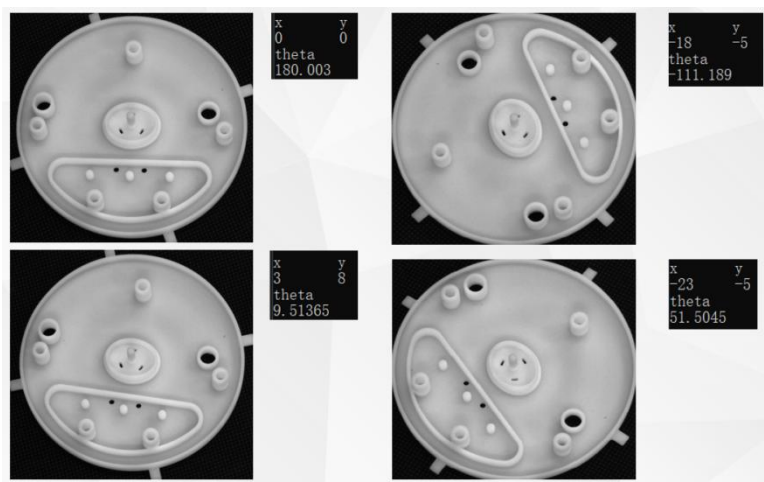


图 5.9 计算相对转角算法结果

通过对结果的分析可知软件计算的结果与我们肉眼观测的结果相符且精度较高,符合我们对算法输出的预期。通过对以下所有样本图片的分析计算,得出了和上面展示的处理结果都类似的数据。算法的可行性与稳定性得到了验证。

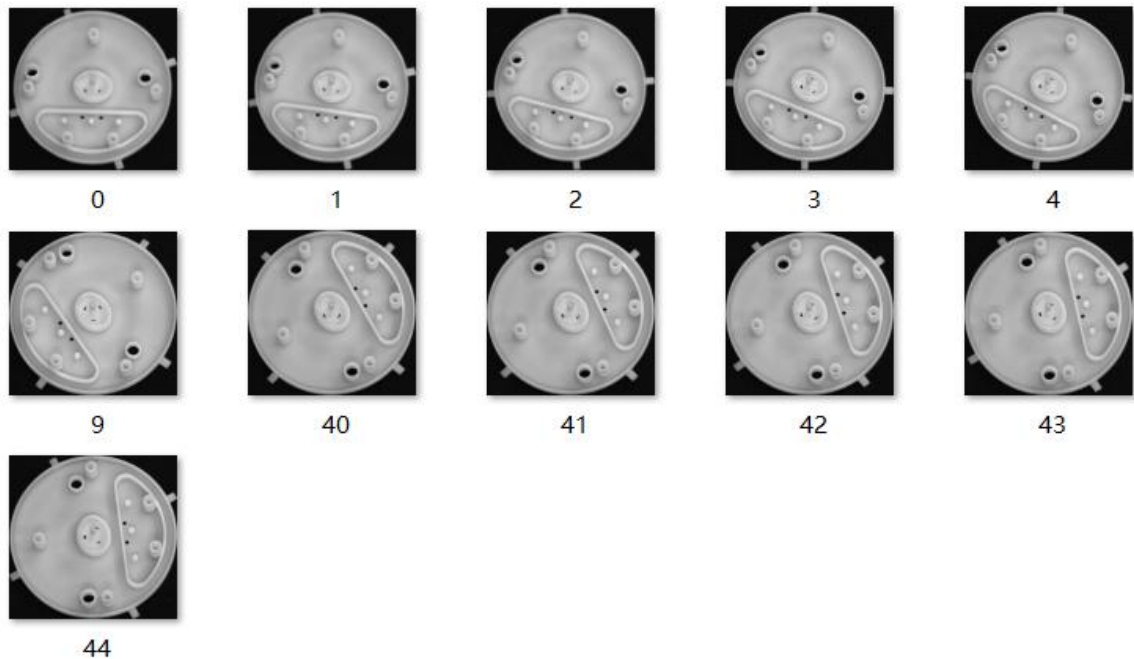


图 5.10 验证算法所用的样本集

本文在 visual studio 搭建的软件界面如下所示,加载图片后,按键执行所需要的算法,处理的图像结果会以弹出窗口的形式展示,数据结果显示在弹出的控制台里。

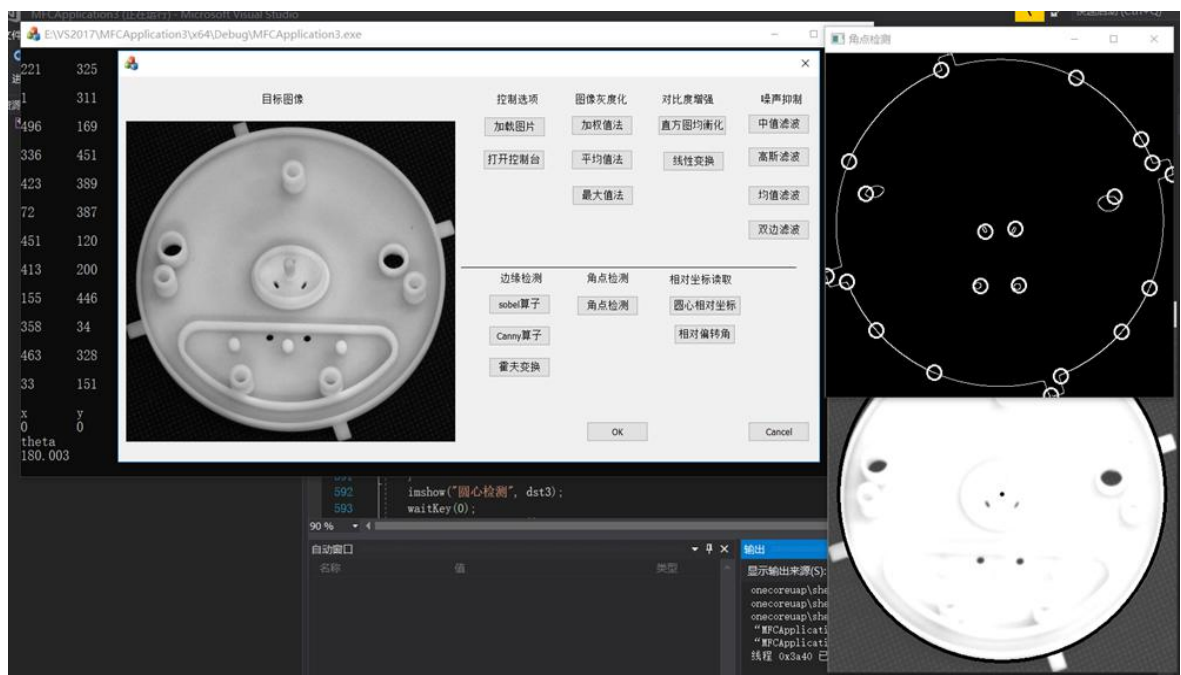


图 5.11 软件平台最终效果

## 第六章 结束语

本课题对于要进行抓取的工件进行视觉识别,提取其位置信息。在这一过程中对主流的工业机器人机器视觉算法进行的介绍与展望,利用了灰度化、图像增强、图像滤波、边缘提取、角点检测、霍夫圆检测等算法对给定的工件的样本图像进行了处理。在每个处理步骤中使用了多种算法平行进行,结合理论特点和实际算法处理的结果比较了不同算法的特点和优劣,选择其中最优的方法加入到最终方案的处理步骤中。在最后的相对坐标读取中,根据不同算法的特点结合工件自身的结构形态,我们提出了一种基于以上基础的识别算法来读取相关的转角坐标,具有一定的创新性。但同时也存在一些不足,比如本算法只能用来处理特定工件,专一性太强而通用性不足。同时,在霍夫圆检测和转角计算过程中采用了一些折衷的办法,虽然对于本工件的抓取已经达到了要求,但在一定程度上牺牲了精确度,有待提出更好的方法来进行计算。

通过本次课题的学习我了解了机器视觉技术的发展历史与其在当下的发展。通过对算法的实现我掌握了在 Visual Studio C++集成开发环境下进行编程,特别是编写 MFC 程序。我还学习到了如何调用 OpenCV 函数库,更为重要的是,学习了其中许多图像处理函数的调用和理论基础。同时我也学习到了很多图像处理中的基础理论,对它们在实际中应用的效果、优点和缺点都有所了解。这对我以后的研究、工作都会有很大的帮助。同时,在写作毕业论文时我也学会了如何进行学术写作、如何查阅相关文献等技能,对我以后的学习都有莫大的帮助。

## 致 谢

在这个课题的进行过程中我得到了许多人的帮助，没有他们的帮助我是无法这么顺利地完成任务的。首先在这里要感谢我的导师林俊义的悉心指导和陪伴，您的讲解让我在前期对于整个题目的了解更加深入、思路更加清晰，通过您对我的算法设计的分析让我很快的找到最初的问题的所在，在和您讨论的过程中我学到很多知识。谢谢老师！其次我还要感谢研究生童磊学长，你提供给我的样本图片给了我很大的帮助，在最初 MFC 编程上遇到的很多问题也是你帮助我一起解决的。和你一起对于最终算法的讨论开阔了我的思路，使得我能够解决这些问题，十分感谢！最后还要感谢学院，能够让我们在选择毕设课题的时候有充分的自由，让我能够选择自己感兴趣的课题并学到相关的知识于技能。

机电及自动化学院 机械设计制造及其自动化专业 2014 级

彭诚

2018 年 5 月 26 日



## 参 考 文 献

- 1 王红涛. 基于视觉的工业机器人目标识别定位方法的研究[D]. 西安理工大学, 2007.
- 2 朱春艳, 荣丽红, 仝志民, 等. 计算机视觉技术的发展与应用[J]. 科学技术创新, 2013(29):174-175.
- 3 胡玮. 机械手自动抓取钢卷系统的研究[D]. 武汉理工大学, 2009.
- 4 李传红, 沈敏德, 崔元波. 一种自动分拣机构的改进设计与运动仿真[J]. 齐鲁工业大学学报, 2012, 26(3):59-62.
- 5 孙雪晨, 姜肖楠, 傅瑶, 等. 基于机器视觉的凸轮轴表面缺陷检测系统[J]. 红外与激光工程, 2013, 42(6):1647-1653.
- 6 易松松. 基于机器视觉的手机面板缺陷检测方法研究[D]. 哈尔滨工业大学, 2016.
- 7 J.R Parker 著, 景丽 译. 图像处理与计算机视觉算法及应用. 北京, 清华大学出版社, 2012.5. 215-241
- 8 Rafael C. Gonzalez 著, 阮秋琦译. 数字图像处理(第二版). 北京, 电子工业出版社, 2001.5. 59-108
- 9 A. Mital, M. Govindaraju, B. Subramani, A comparison between manual and hybrid methods in parts inspection, Integrated Manufacturing Systems 9 (1998) 344-349.
- 10 P. Kopardekar, A. Mital, S. Anand, Manual, hybrid and automated inspection literature and current research, Integrated Manufacturing Systems 4 (1993) 18-29.
- 11 朱铮涛, 黎绍发. 视觉测量技术及其在现代制造工业中的应用[J]. 现代制造工程, 2004 (4) : 59-61.
- 12 刘继胜, 钟良. 机器视觉技术在质量检测中的应用[J]. 现代制造工程, 2004 (2) : 99-101.
- 13 唐向阳, 张勇, 李江有, 等. 机器视觉关键技术的现状及应用展望[J]. 昆明理工大学学报 (理工 版), 2004, 29 (2) : 36-39.
- 14 范祥, 卢道华, 王佳. 机器视觉在工业领域中的研究应用[J]. 现代制造工程, 2007(6):129-133.
- 15 李立宗 著. OpenCV 编程案例详解. 北京, 电子工业出版社, 2016.10. 39-47
- 16 袁欢, 基于计算机视觉技术的零件自动分检系统[D]. 浙江理工大学, 2015.



- 17 韦衡冰, 邓小林, 潘明华. 沙糖桔分拣分级装置研究和设计[J]. 中国农机化学报, 2016, 37(1):77-82.
- 18 赵彦硕. 药板自动排序分拣机的设计与研究[D]. 长春理工大学, 2014.
- 19 胡兴军, 蔡叶菁, 王健. 机器视觉技术及其在包装印刷质量检测中的应用[J]. 中国包装工业, 2004(11):35-37.
- 20 李彩花. 基于 SCARA 机器人控制系统的视觉集成技术与研究[D]. 哈尔滨工业大学, 2005.