

## 摘要

在本文的开始,对 PCB 的缺陷类型进行了分析,根据 PCB 的缺陷特征的不同,之后需要采用不同的识别方法进行操作。接下来总结了基于 OpenCV 的各种机器视觉算法,分析各种算法的处理结果,选取可供之后使用的算法。最后对之前的研究进行总结,将缺陷进行分类,根据不同的缺陷特征匹配合理的视觉算法,就可以得到检测某些特定缺陷的机器视觉检测缺陷算法。

用实验的方法对所设计的算法进行测试,使其去检测包含不同缺陷的 PCB 图像,实验结果证明,文中介绍的 PCB 缺陷检测算法能够对 PCB 上几种缺陷进行有效地检测,包括通孔堵塞、断路、缺损等。由于加入 Mark 点识别操作,在识别缺陷的同时可以将缺陷进行定位,并可以将检测的结果输出以便进行后续的处理工作。

关键字: 印刷电路板 (PCB); 机器视觉; 缺陷

## ABSTRACT

At the beginning of this paper, the defect types of PCB are analyzed. According to the PCB's defect characteristics, different recognition methods are needed to operate. Then, we summarize all kinds of machine vision algorithms based on OpenCV, analyze the processing results of various algorithms, and select the algorithms that can be used later. Finally, we summarize the previous research, classify the defects, and match the reasonable visual algorithm according to the different defect characteristics, then we can get the machine vision inspection defect algorithm to detect some specific defects.

To test the algorithm by using the experimental method, the detection of PCB images to contain different defects. Experimental results show that the PCB defect detection algorithm in this paper can effectively detect PCB of several defects, including a hole plug, circuit breaker, defect etc. With the addition of Mark point recognition operations, the defect can be located while identifying the defect, and the output of the detection can be output for subsequent processing.

**Keyword:** printed circuit board (PCB)、machine vision、defect

## 目录

<b>第一章 绪论</b>	<b>1</b>
1.1 课题的研究背景	1
1.2 机器视觉介绍	2
1.3 课题的主要研究内容	2
<b>第二章 PCB 缺陷类型分析</b>	<b>3</b>
2.1 PCB 的生产过程	3
2.2 PCB 缺陷定义和分类	3
2.3 PCB 缺陷识别算法介绍	7
2.4 小结	8
<b>第三章 常用视觉处理算法</b>	<b>11</b>
3.1 阈值操作	11
3.1.1 阈值介绍	11
3.1.2 阈值化的类型	11
3.1.3 阈值化处理结果	13
3.2 Sobel 导数	16
3.3 Canny 边缘检测	18
3.4 霍夫变换	19
3.5 直方图均衡化	21
<b>第四章 OpenCV 在 PCB 缺陷检测中的运用</b>	<b>23</b>
4.1 模板匹配法	23
4.1.1 模板匹配法介绍	23
4.1.2 模板匹配法实验	26
4.2 Mark 点识别	28
4.3 小结	30
<b>第五章 总结与展望</b>	<b>31</b>
<b>致谢</b>	<b>33</b>
<b>参考文献</b>	<b>35</b>



## 第一章 绪论

### 1.1 课题的研究背景

当今电子行业，基于 PCB 的电子产品已经成为的一个必要部分。而现代技术正在向着高度数字化的方向发展，可以预见，PCB 的需求以及产量也将会飞快地增长。在这一方面，各个国家都给予了充分的重视，例如马来西亚在全国范围内已经有 37 个生产 PCB 板的上市公司，而中国成为了全球第一大 PCB 生产基地。

从资料上来看，全球 PCB 产业产值占电子元件产业总产值的 25% 以上，2016 年全球 PCB 总产值已达 542.07 亿美元。从中国的 PCB 产业发展情况来看，从 2000 年到 2016 年的 16 年时间里，我国 PCB 产值的从 33.7 亿美元增长到 271 亿美元，增长率高达 704%，占全球的比重也从 8.2% 提高到 50.0%。以上数据表明，中国 PCB 的产值相当于全球 PCB 产值的一半，这是非常可观的。另外，现在的电子产品的越来越轻薄，体积也越来越小，这就要求 PCB 也朝着超薄型、小元件、细间距、多层数、高性能方向发展<sup>[1]</sup>。PCB 在制作的过程中可能由于某些原因导致焊盘出现缺陷或者线路断开等问题，若将这些电路板应用到电子产品中，很有可能会导致各种问题后期在电子产品中出现，从而造成重大损失，甚至整个电子产品也会因此而报废。

为了确保使没有缺陷的 PCB 转换为高质量、高可靠的电子产品，实现关于 PCB 投入使用时的零缺陷，PCB 在电子组装前的质量检测就变得尤为重要。

最初 PCB 缺陷检测主要借助人工方式，依靠人力观察每一块 PCB，检查出每一处的缺陷。这种检测方式会产生较高的工作成本，不仅浪费时间，而且检测的准确性也无法得到保证。随着 PCB 高集成度、排版复杂化的发展方向，人工检测方式对于 PCB 的检测来说已经不再适用，因此有必要寻找一种新的自动而有效的方法，使得检测工作更加规范与智能，这便产生了检测 PCB 缺陷的机器视觉检测技术。

## 1.2 机器视觉介绍

机器视觉检测技术(Machine vision detection technology),是集合了计算机学、图像处理学、数字信号处理学等的综合技术,是应用于现代精密检测技术的领域的技术中最具有发展潜力的。由于机器视觉检测技术可以通过采取图像间接测量的方法进行检测,不需要与物体进行接触,而且检测速度快、效率高,因此常常被应用于工业检测中,来对物体(产品或零件)空间尺寸或位置进行快速检测与测量。根据其检测性质和应用范围的不同,机器视觉检测技术通常分为定量检测和定性检测两种<sup>[2]</sup>。机器视觉检测系统在生活中或者生产中的各个地方都有十分活跃的应用,例如:贴片机 Mark 点检测、印刷电路板的视觉检测、工件平行度和垂直度测量、机械零件毛刺检测、各类物品的自动识别分类和几何尺寸测量等。除了在可观物体的检测上的运用外,在许多使用直接检测的方法难以检测的场合,利用机器视觉系统可以有效实现,例如纳米级尺寸的检测。机器视觉检测系统的应用正在越来越多地代替人力劳动,这在很大程度上就会提高了生产自动化的水平和智能检测系统的水平

## 1.3 课题的主要研究内容

本文研究的主要内容是用于检测 PCB 在电子组装前缺陷的检测算法,所使用的工具为 OpenCV。运用 OpenCV 来编写视觉检测处理的算法用以检测缺陷。

首先,需要讨论的是 PCB 缺陷的类型。因为 PCB 缺陷种类非常多,难以使用一种算法同时检测所有的缺陷,而有些算法也是无法使用视觉检测的方法与识别的。因此需要对众多的缺陷进行分类整理,选择便于使用视觉算法处理的缺陷类型进行设计。接下来就是对于几种常见算法的介绍。OpenCV 是非常实用的视觉处理工具,通过调用内部的函数即可达到处理各种图像的目的。文章中主要介绍了几种实用的算法,包括算法函数的介绍以及使用的说明。最后就是介绍针对 PCB 缺陷检测的几种方案,用以检测几种常见缺陷。其中包含了对于采集到的包含不同缺陷的 PCB 图像进行的实验以及检测的结果,并且对于结果进行总结分析,做出对于进一步研究的展望。

## 第二章 PCB 缺陷类型分析

### 2.1 PCB 的生产过程

PCB 缺陷产生于 PCB 制作的过程中，可能是因为设计的问题，或者因为制作时工艺及条件没有达到制作的要求。因此要了解 PCB 的缺陷以及产生的原因首先需要了解 PCB 的制作过程。PCB 的制作方法很多，例如积层法、减去法、增层法、加成法等等，但各种方法制作 PCB 的过程可以归为同一个流程，如图 2.1 所示。在由图 2-1 表示的流程中，每一个步骤都有相应的工艺要求，而每一个工艺的差错都可能造成 PCB 出现缺陷。如果从避免浪费的角度出发，在每一个步骤都是需要进行检测的。但在实际生产中，若在每一步进行一次检测会消耗大量的时间从而减慢制作的速度，因此通常在 PCB 制作完成后进行统一的检测工作。因此本文的检测工作也集中在 PCB 制作完成后，在电子组装步骤前的缺陷检测设计。

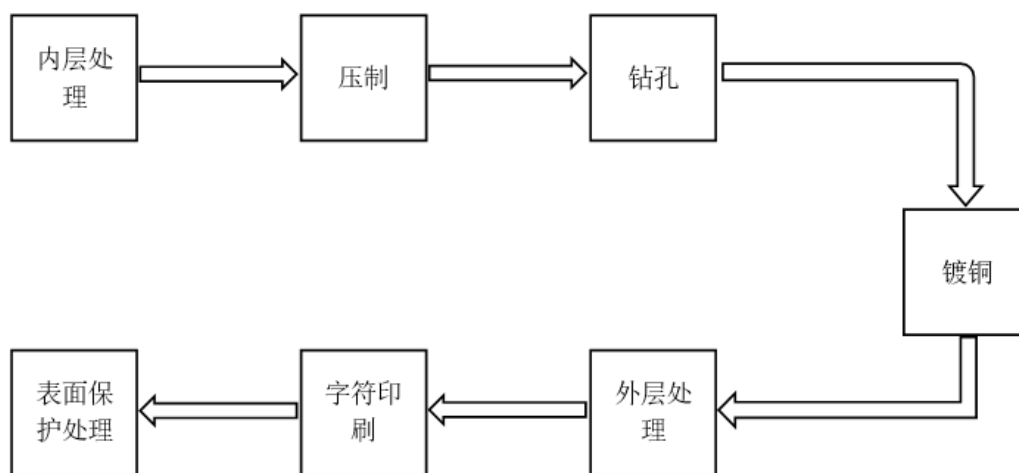


图 2.1 PCB 的生产过程

### 2.2 PCB 缺陷定义和分类

PCB 的缺陷特征非常多，各行业根据自己的需要对缺陷进行分类，所以就产生了许多种对于缺陷分类的方法。其中最明显的一种分类方法就是根据缺陷产生于哪个步骤来对 PCB 的缺陷进行分类。如图 2-2 展示的就是在镀铜阶段由于过度腐蚀或者腐蚀不足而产生的缺陷。该缺陷的特点为覆铜层内出现空洞或者覆铜层超出设计的范围，这类缺陷必然会造成电路的电性能方面的影响，从而导致 PCB

的性能无法达到预期设想。由于这类缺陷通常出现在 PCB 的表面，容易捕捉到缺陷的图像，因此，可以通过视觉检测算法进行识别检测。图 2.2 圈出部分就是 PCB 上的缺陷。

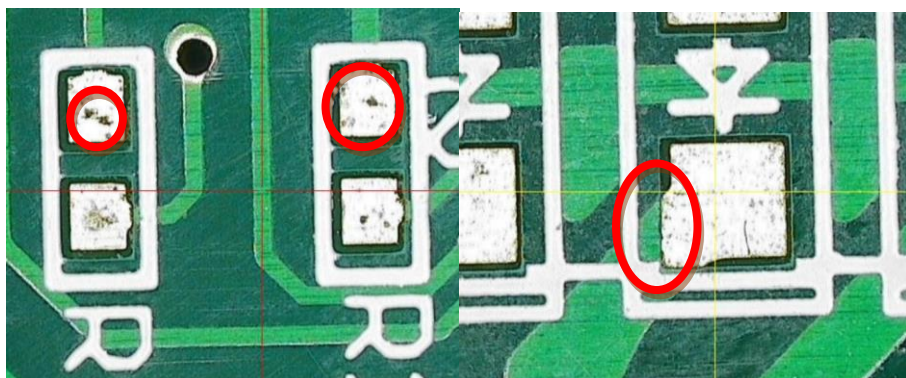


图 2.2 过度腐蚀或者腐蚀不足而产生的缺陷

一块没有缺陷的 PCB 如图 2.3 所示，图 2.4 中的 PCB 示意图与图 2.3 中的线路布局是一样的，但不同的是其包含了多种缺陷，其缺陷的名字在示意图边上也有标注。各种缺陷对 PCB 功能方面损害严重性是不相同的，以损害的严重性为依据，我们可以将缺陷分为本质性缺陷和潜在性缺陷。可能会导致 PCB 的功能完全无法实现的缺陷称为本质性缺陷，会导致 PCB 在使用的过程当中可能出现故障的缺陷称为潜在性缺陷<sup>[3]</sup>。图 2.5 中按照此依据对常见的缺陷进行了分类，进行了这个步骤才能根据不同类型缺陷设计不同的检测算法。

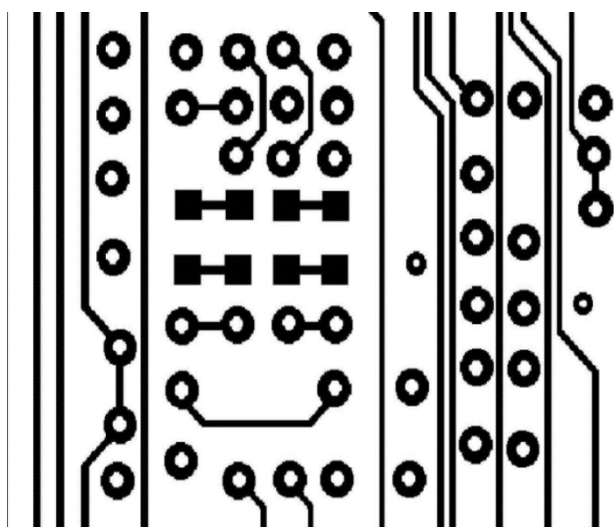


图 2.3 没有缺陷的 PCB 示意图



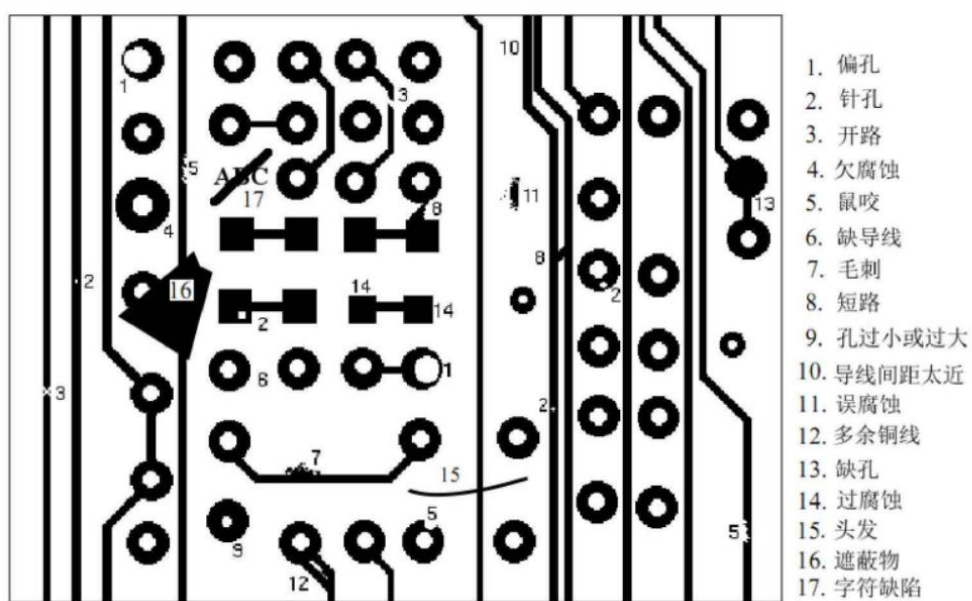


图 2.4 包含多种缺陷的 PCB 示意图

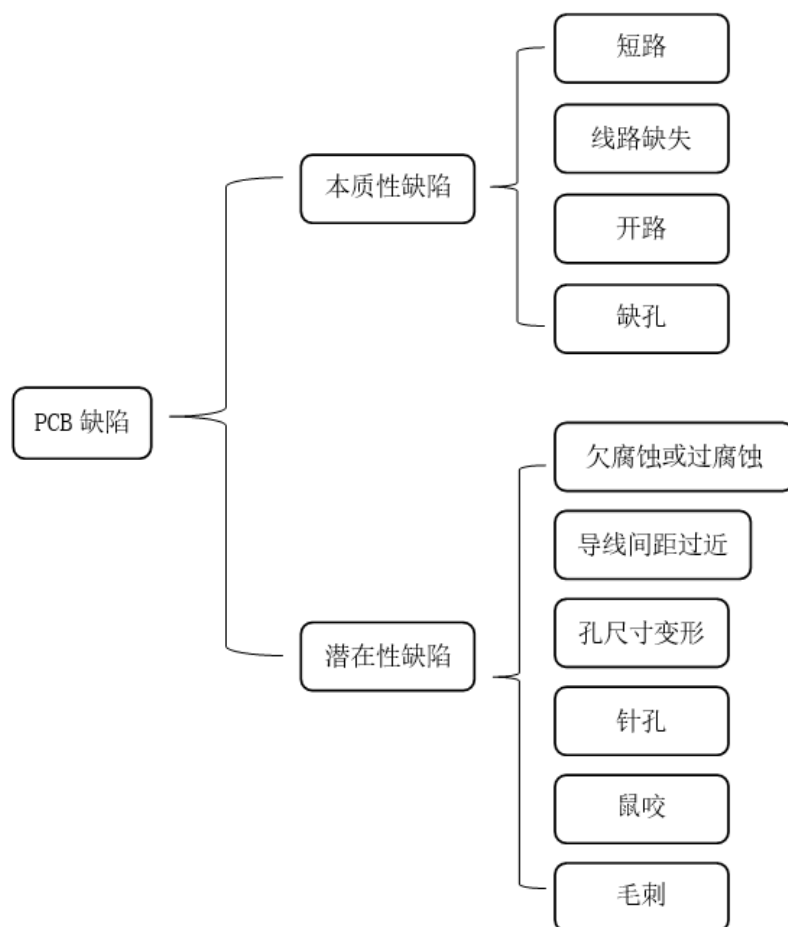


图 2.5 PCB 缺陷分类

在实际情况下,没有两块 PCB 是完全相同的,而在一块 PCB 上或多或少会出现一些缺陷,因此只要把缺陷控制在可接受的范围内,或者当缺陷可以被简单修复的时候,可以将该 PCB 视为合格的产品。在检测的过程中,需要将待检 PCB 与标准模板进行匹配。由于视觉检测系统是根据图像进行检测的过程,在与标准模板匹配中就需要加上一定的容错空间,而不是完全与模板保持一致。

经过对缺陷的分类,就可以从以上的缺陷类型中挑选可以采用视觉检测算法识别的缺陷。首先讨论的是本质性缺陷。由于本质性缺陷会导致 PCB 完全无法工作,这项内容的检测就变得尤为重要。如图 2.5 所示,本质性缺陷包含短路、线路缺失、开路和缺孔,下面就针对这四种缺陷类型进行分析。

第一个是短路的问题。PCB 的短路造成的影响非常大。短路会使电路中电流不经过某一用电器,由于导线的电阻很小,因此电流会异常大,这样的电流电池或者电源都是无法承受的,会造成电源的损坏。轻的情况下,电路短路会使电路工作出现问题,严重的情况下,导线的温度升高,会烧穿 PCB 甚至引发安全问题。PCB 的短路原因很多,设计上的原因由于在采集的图像上没有明显特征,我们是无法通过图像进行识别的。另外一种短路原因就是在 PCB 制作或者存放过程中,导电物质连接了两个线路导致短路问题的出现<sup>[4]</sup>。例如图 2.6 所示,在圈出的位置由于焊锡连接了两条导线,导致短路。这类的缺陷可以通过把缺陷模板导入算法。以图 2.6 为例,这类缺陷的特征是在两条平行线之间有异常的物体,将这类缺陷特征与待检 PCB 进行比对,即可检测出有这类问题的 PCB。

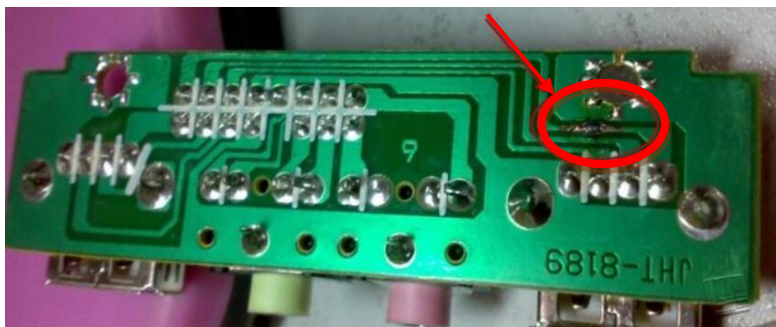


图 2.6 PCB 短路

其次讨论的是线路缺失这类 PCB 缺陷问题。此类缺陷的特征就是制作出的 PCB 与设计的不符合,有部分的线路未按照要求制作出来。这种缺陷直接造成了该线路连接的电路无法正常工作。从可行性角度来看,若把 PCB 与标准模板进行

比对, 然后找出与标准模板有差异的地方即可找出此类缺陷, 但在实际过程中, 操作难度较大。因为从与标准模板差异性来看, 存在差异的不一定是这类缺陷, 任何一种缺陷都会造成与 PCB 标准模板的不同, 在检测之后也不易区分, 因此暂时不对这种缺陷类型进行分析。

接下来分析的是开路缺陷问题。在 PCB 制作的过程中, 若腐蚀出现问题, 就有可能造成 PCB 上的线路断开。或者在制作的过程中进入了某些不导电的杂质, 引起开路的问题。电路开路即电流无法正常经过用电器, 这部分电路就失去了预期功能, 这类缺陷也是绝对不允许出现在用于生产电子产品的 PCB 上的。这类缺陷的特征为在线条形的导线上出现端口, 即导线不连续。从图像检测的角度出发, 可以将断开部分的特征导入缺陷库内, 通过将缺陷特征与 PCB 图像进行比对, 找出 PCB 开路缺陷。

PCB 的缺孔有两方面的原因。首先是由于操作上的原因忘记对 PCB 进行开孔, 另一个原因是开孔后在后续操作的时候堵塞了。PCB 缺孔致使插装元器件无法正常焊接在 PCB 上, PCB 就无法完成工作需求。由于图像检测是的检测依据是灰度值, 而缺孔的 PCB 在开孔内圆的灰度值与正常开孔是不同的, 可以考虑采用圆检测算法, 然后检测内圆灰度值, 之后与正常的开孔进行比对以检测此类缺陷。

最后讨论的是潜在性缺陷的检测。潜在性缺陷的种类非常多, 而且大多不会直接影响到电路的正常工作。在缺陷检测方面, 此类缺陷的特点大多不太明显。例如毛刺和导线间距过近这两类问题, 从图像检测的角度出发无法很好地尽心识别。但是潜在性缺陷也不是所有都无法进行检测的。过腐蚀和欠腐蚀以及孔尺寸变形这些有明显特征的缺陷还是可以进行识别检测的。在 PCB 上, 焊盘的形状大多为标准圆形或者矩形, 而孔的形状都为标准的圆形。若是在形状上出现缺陷, 可以直接把矩形以及圆形的模型导入匹配模板, 通过对比检测缺陷。

## 2.3 PCB 缺陷识别算法介绍

PCB 缺陷识别算法有很多种, 一般情况下我们可以将其分为两大类, 分别是参考比较法和非参考比较法<sup>[5]</sup>。在检测 PCB 缺陷的时候通常需要同时使用两种方法同时对 PCB 进行检测以节省检测时间。

参考比较法是通过图与图的对比来检测待测 PCB 的缺陷。首先需要得到一块

标准 PCB 的图像。在实际情况下,制作 PCB 的过程中难免出现一些工艺误差导致 PCB 小缺陷的产生,因此标准 PCB 图像可以通过对图像进行处理得到。例如通过视觉算法,对图像的毛刺、断线或缺损的部分进行修复,从而得到完整的 PCB 模板图像。之后将此标准 PCB 图像与待测 PCB 进行图像特征对比,与标准 PCB 图像不一致的地方就是待测 PCB 存在缺陷的位置。之后,将存在缺陷的位置进行二次测定,以缺陷的特征来判定缺陷的类型。

无参考比较法不需要采集标准的 PCB 图像,而是通过测定一些参数来判断 PCB 上是否存在缺陷。PCB 在设计及制作的时候都是遵循一定的参数原则的,这个原则可能是线与线之间的距离,也可能是开孔的大小等等。而存在缺陷的 PCB 在尺寸参数上与标准的 PCB 有差距,以此为依据,可以通过视觉算法,对采集到的待测 PCB 图像内各个参数进行测定,若参数不符合标准设定,则可将其判断为缺陷。这种方法中,需要去测定的参数包括开孔的尺寸、线距、边缘是否规则等等。通常情况下,由于设计 PCB 时根据要求不同会有不同的设计方法,因此采用无参考比较法不能精确地测定出 PCB 上的缺陷,但是检测的结果可以作为一个参考,方便进行进一步的检测工作。

两种方法的优劣性不加以评判,但是在实际使用的时候可以根据缺陷的特征去选用适合的方法加以识别。在上文中已经介绍了 PCB 缺陷的分类方法,每一种缺陷检测的复杂程度都是不相同的。对于较为简单的 PCB,可以采用无参考比较法进行粗检测,然后对检测结果进行分析以得到 PCB 缺陷的位置。若想得到更为精确的结果,或者 PCB 的结构较为复杂,采用参考比较法是更为妥当的选择。以参考比较法进行缺陷的识别具有针对性,及以待测板的标准板作为模板对待测板进行匹配,通过一一对比的方式检测出 PCB 上存在的缺陷。尽管这种方法也会存在误差,但相对于无参考比较法,以这种方法检测的结果在后续操作中会更加方便。

## 2.4 小结

以上对 PCB 的各类缺陷进行了一下总结和分类,目的就是 PCB 的各类缺陷有更加清晰的了解,以及方便对之后的操作做准备。PCB 的缺陷整理是一个复杂的过程,在分类的过程中需要考虑多方面因素,而以上的分类则是从对电路功能

的影响以及缺陷识别可行性方面进行考虑的。从以上的分析中可以得出，PCB 的缺陷对 PCB 的影响是各不相同的，有些缺陷可以导致电路完全无法正常工作，而有些缺陷仅仅是形态上的缺陷，并不会直接对电路的功能造成影响。所以，在缺陷检测的实施上，也可以进行轻重缓急的划分，对于影响严重的缺陷需要进行首先检测。

本章还对各类缺陷的检测可行性进行了分析。每一种缺陷都有其特征，而本文讨论的仅仅是以图像处理的方法对缺陷进行检测，因此把各类缺陷的特征都归于图形化。有些特征虽然对电路影响很大，但是由于其在图像上无特别的特征，因此无法使用图像处理进行检测，而需要之后以其他方法检测其缺陷是否存在，这类缺陷在下文就不再提及。而在图像上有检测特征的缺陷又可以分为两类，一种是固定缺陷特征的缺陷，一种是有明显特征的缺陷，对于缺陷检测的算法也就是围绕这两方面展开。



## 第三章 常用视觉处理算法

### 3.1 阈值操作

#### 3.1.1 阈值介绍

阈值操作可以说是图像处理算法中最为简单也最为重要的图像分割的方法。无论是用于检测还是测量，都需要过滤掉图形中一些无关像素点或者根据灰度值对图像中的像素点进行划分，这时候就需要进行阈值操作。举一个简单的例子，每张图片中各部分的颜色都是有差异的，而颜色的差异从视觉处理算法上来看可以归为灰度值的差异，换一句话说图像中物体（部分或者整体）相对于背景颜色的差异以相应算法进行换算，得到的就是灰度值。通过对图像的阈值化，就可以对灰度值不同的像素点进行划分，分割出我们需要的物体部分。而且此分割属于像素级的分割<sup>[6]</sup>。

阈值相当于一个量化标准，为了从一张图片中取出我们需要的部分，我们首先需要得到每一个像素点的灰度值，然后将此灰度值与标准进行比较，并根据比较的结果进行相应操作。将所有的像素点都进行过比对之后，将有相同特征的像素点进行归类，同类的像素点提取出来即可分割所需物体部分。通常这些像素点会被在程序中被设定一些特定的值，例如灰度值 0(黑色)以及灰度值 255(白色)。当然灰度值可以设置为任意值，但为了视觉上的强烈对比以及方便后续操作，采用对比度较强的值更加有利。

结合本课题开看，在 PCB 缺陷识别上，板上有各种各样的图形特征，但不是所有的特征都为缺陷。想要在识别过程中对缺陷进行一次判定，就必须加上阈值处理，才能更为精确地得到缺陷的部分。

#### 3.1.2 阈值化的类型

在 OpenCV 中提供了 threshold 函数，可以通过调用此函数进行阈值化的操作。而在该函数中，所提供的阈值化类型有以下五种。

##### 1. 二进制阈值化

该阈值化类型如式（3-1）所示：

$$\text{dst}(x,y) = \begin{cases} \text{maxVal} & \text{if } \text{scr}(x,y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases} \quad (3-1)$$

当设定一个基准之后，数值的类型自然被分为了两种，高于此数值的或者不高于此数值的。应用在阈值时，就是像素点的灰度值是否高于阈值。阈值通常设定为一个常数，或者在程序中供更改的数值变量。当图像中某个像素点的灰度值较大，高于了设定阈值，那么就将此点灰度值设为 **maxVal**（如最大灰度值 255）。反之，则将灰度值设为 0。处理结果可以由图 3.1 表示，其中红色区域表示图像中灰度值较大的部分，白色区域表示图像中灰度值较小的像素点，加粗黑色线条表示的是经过处理之后重新为像素点赋的灰度值。

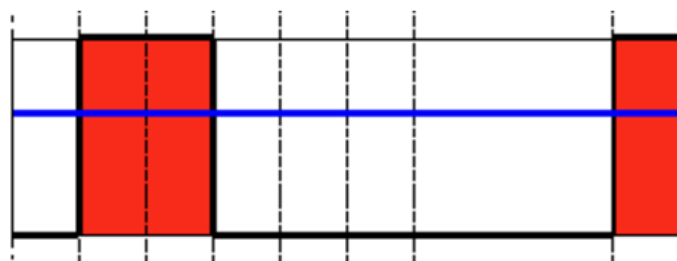


图 3.1 二进制阈值化

## 2. 反二进制阈值化

该阈值类型如下式（3-2）所示：

$$\text{dst}(x,y) = \begin{cases} 0 & \text{if } \text{scr}(x,y) > \text{thresh} \\ \text{maxVal} & \text{otherwise} \end{cases} \quad (3-2)$$

反二进制阈值化与二进制阈值化在处理像素点的思路上是相似的。在这种处理方式中，同样需要拟定一个特定的灰度值来作为阈值，将图像中的各个像素点的灰度值与该阈值进行一一对比。不同的是，反二进制阈值化在结果处理上与二进制阈值化相反。反二进制阈值化中，函数将灰度值小于阈值的像素点的灰度值设为预先设定的最大值，而将大于阈值的像素点的灰度值置为 0。处理的结果如图 3.2 所示。

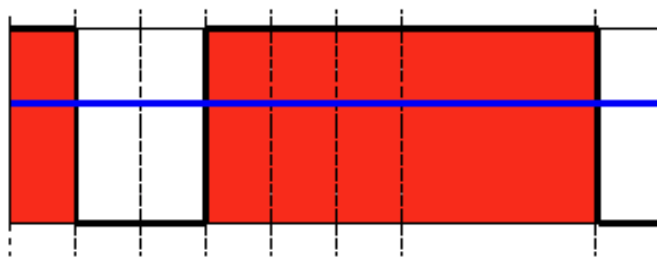


图 3.2 反二进制阈值化



### 3. 截断阈值化

该阈值化类型如式(3-3)所示：

$$\text{dst}(x,y) = \begin{cases} \text{threshold} & \text{if } \text{scr}(x,y) > \text{thresh} \\ \text{scr}(x,y) & \text{otherwise} \end{cases} \quad (3-3)$$

在截断阈值化中，也需要输入一个灰度值数值作为阈值。当图像中像素点的灰度值大于这个输入数值，将该像素点的灰度值设为阈值，在其他情况下的像素点则保持不变。处理的结果如图 3.3 所示。

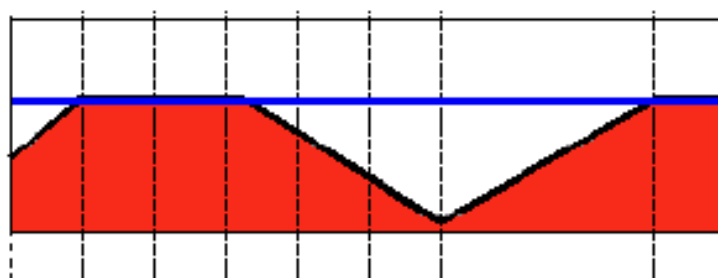


图 3.3 截断阈值化

### 4. 阈值化为 0

该阈值类型如式(3-4)所示：

$$\text{dst}(x,y) = \begin{cases} 0 & \text{if } \text{scr}(x,y) > \text{thresh} \\ \text{scr}(x,y) & \text{otherwise} \end{cases} \quad (3-4)$$

顾名思义，反阈值化为 0 的操作结果就是与阈值化为 0 相反的。与其他阈值处理类似，使用这种类型的阈值化也需要设定一个基础阈值。通过对比图像中每一个像素点的灰度值和所设定的阈值大小，对图像中像素点灰度值的数值进行相应的操作，即置 0 或保持不变。处理的结果如图 3.5 所示。

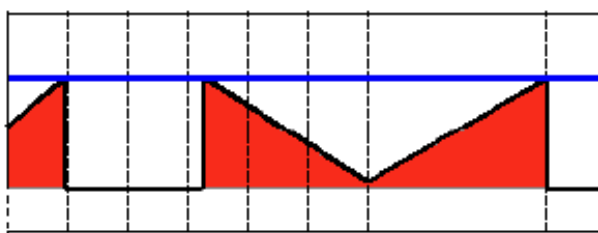


图 3.5 反阈值化为 0

#### 3.1.3 阈值化处理结果

##### 1. 二进制阈值化

图 3.6 是未经过处理的 PCB 上采集的 Mark 点部分的图像。图 3.7 表示的是二

进制阈值化处理的结果，第一张图片为将阈值设为 50 时，运用函数处理图片所得的结果，第二张图片为将阈值设为 185 时的结果。可以看出，运用二进制阈值化处理图片之后，黑色的部分为灰度值较大的部分，而灰度值小于阈值的部分为白色。

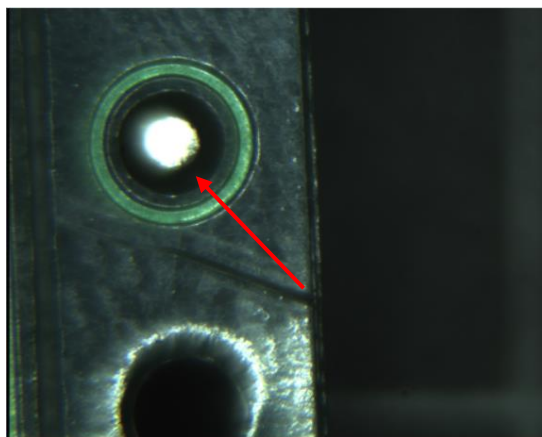


图 3.6 未经处理 Mark 点图像

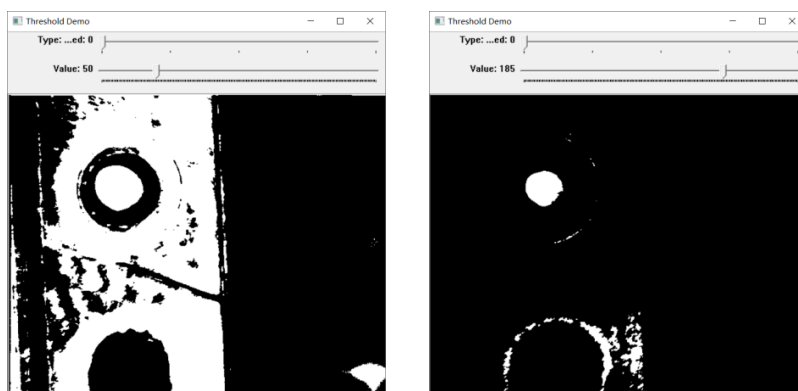


图 3.7 二进制阈值化

## 2. 反二进制阈值化

图 3.8 表示的是以反二进制阈值化处理的图像。同样，这部分处理的原图像为图 3.6，后面两张为经过算法处理过的图像，两张图片设定的阈值也分别为 50 和 185。可以很清楚地看出，用反二进制阈值化处理的图像在色彩上与二进制阈值化是完全相反的。在灰度值较高的地方显示的颜色成了白色，而灰度值较低的地方显示黑色。

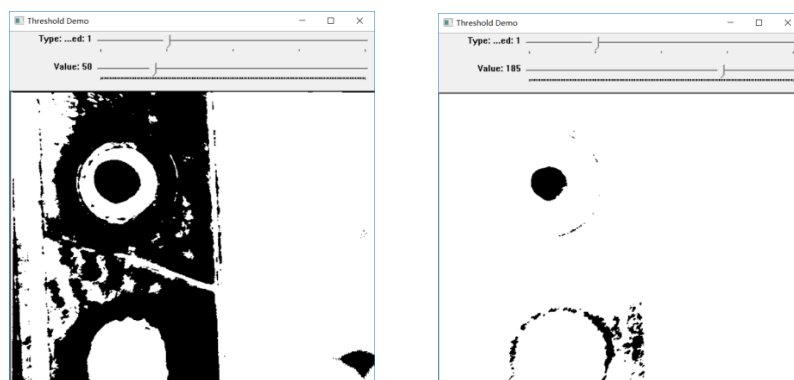


图 3.8 反二进制阈值化

### 3. 截断阈值

在图 3.9 中，处理后的结果就不像之前两种方法处理的结果了。在截断阈值操作之后会根据设定阈值的高低保留图片中原有的部分，而不是将所有图像中的像素点重新赋予新的灰度值。根据设定阈值的高低，保留部分也有所不同。在第一张图片中，由于设定阈值为 50，保留内容较少，而第二张图片中设定阈值为 185，保留部分就增加许多。

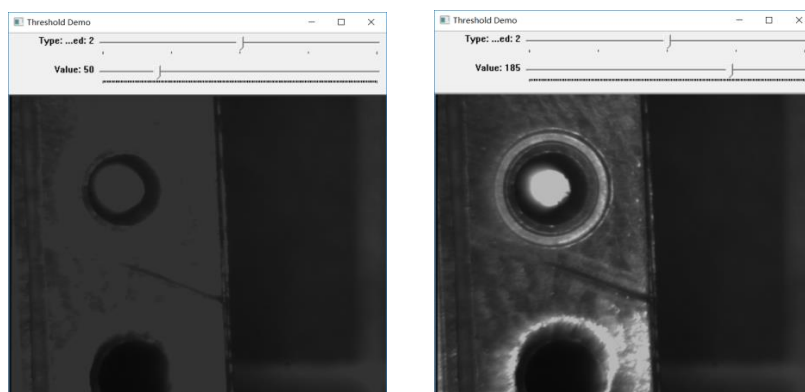


图 3.9 截断阈值

### 5. 阈值化为 0

图 3-10 中表示的是经过 0 阈值处理后的图像。由于该阈值化类型中将灰度值小于阈值的像素点灰度值置 0（黑色），因此若阈值设定不高，则图像中显示原本内容就较多。当阈值设定为 185 时，由于图片很多部分灰度值未达到，图片黑色部分就比较多。

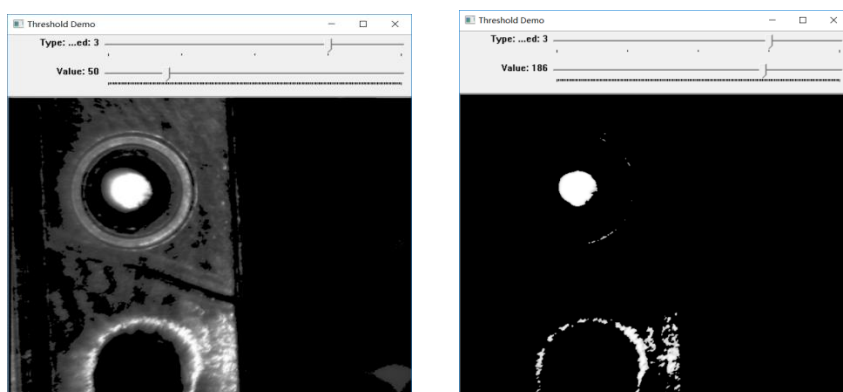


图 3.10 阈值化为 0

## 6. 反阈值化为 0

反阈值化为 0 操作后的图像如图 3.11 所示，此结果与阈值化为 0 相反。当阈值为 50 时图片原本的内容基本被黑色取代，而阈值为 185 时显示原本内容较多。

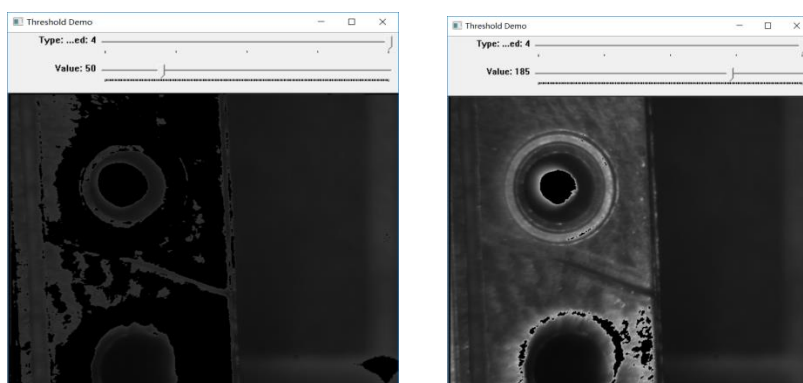
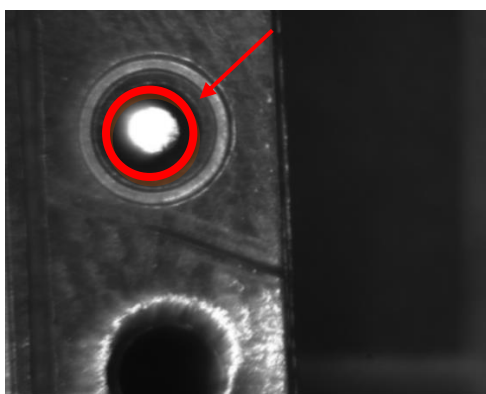


图 3.11 反阈值化为 0

## 3.2 Sobel 导数

对图形的卷积运算是图像处理中很重要的一个环节，而在卷积运算中，对图像进行求导又是最重要的。对于图像的求导计算的用处很多，而在缺陷检测方面，可以用于检测图像的边缘。如图 3.12 所示，在圈出部分有很明显的颜色变化,从白色过渡到黑色，即在边缘部分像素点的灰度值发生了显著的变化。表示这一显著变化的一个方式就是使用导数，梯度值大变化预示图像中可能存在边缘<sup>[7]</sup>。



3.12 边缘

假设被作用图像为  $I$ ，首先要在水平和垂直方向上进行求导。以水平方向上的求导过程为例，将图像  $I$  与一个奇数大小的内核  $G_x$  进行卷积。例如，当内核大小为 3 的时候， $G_x$  的计算结果可以表示为式(3-5)。

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \cdot I \quad (3-5)$$

而在垂直方向的计算结果为式(3-6)。

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \cdot I \quad (3-6)$$

运用在水平以及垂直方向上处理的结果  $G_x$  及  $G_y$  即可运用公式(3-7)求出某个像素点的近似梯度。

$$G = \sqrt{G_x^2 + G_y^2} \quad (3-7)$$

Sobel 算子函数的作用就是提供一个内核，以该内核为依据对原图像进行寻边操作。依旧以图 3.6 为例进行图像处理操作，在进行处理之后，图片如 3.13 所示。可以看出，经过处理之后的图片保留下了图片边缘的部分，但是由于这仅仅是初步的寻边操作，在处理之后也保留了很多其他有灰度值变化的部分。即便如此，这次初步的寻边操作可以为之后的各种操作提供极大的便利。

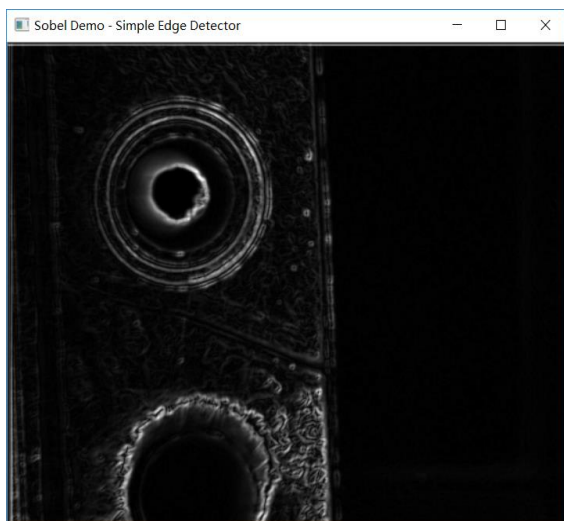


图 3.13 Sobel 导数

### 3.3 Canny 边缘检测

John F. Canny 在 1986 年研究出了一种多级边缘检测算法,也就是现在的 Canny 边缘检测算法。这种算法在边缘检测上有很多的优越性,同时也容易操作,因此被很多人认为是边缘检测的最优算法。

通常情况下,在图像处理领域对于最优边缘检测的主要评价标准有以下三点:

- (1) 低错误率: 图像处理中噪声的影响是很大,图像中的噪声可能在像素点的判断上对算法造成干扰导致错误的出现。在 Canny 边缘检测中,对噪声的抑制做了很好的设计,这样保证了更多的边缘能够被正确地识别出来。
- (2) 高定位性: 在识别出边缘之后,需要将所识别的边缘标出以便后续的操作。若识别的准确性不错但是边缘的标识不够准确,在图像处理过程中也会带来很大的误差。Canny 边缘检测在识别边缘之后可以很方便地定点到识别的位置。为了达到尽可能准确,将标识的边缘和图像中的边缘尽可能贴近。
- (3) 最小响应: 由于图像中很多不可控因素,很多边缘可能会被多次识别,带来了数据的冗余。在 Canny 边缘检测中最大化地排除了干扰,让同一个边缘只能被一次识别。

在执行 Canny 边缘检测时,第一步是对图片进行噪声的消除。使用高斯平滑滤波器卷积降噪是一般情况下使用的方法<sup>[8]</sup>,以式(3-8)所示的五阶的高斯内核作为例子进行演示:

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (3-8)$$

接下来需要计算梯度的幅值和方向，这一步骤可以参考 3.2 中介绍的方法。对这一步之后的图片进行非极大值抑制，用以排除边缘像素，这一步可以排除无关的像素点，保证识别出的边缘不会很粗或者多次识别同一边缘。最后一步是阈值操作。在图像中，若一个像素灰度超过高阈值,认为这个像素在边缘上。如果这一个像素位置的灰度幅值比低阈值小，该像素被排除。如果某一像素位置的灰度幅值在两个阈值之间，该像素仅仅在连接到一个高于高阈值的像素时被保留。一般来说，使用 Canny 算子时使用的高:低阈值比在 2:1 到 3:1 之间。

图 3.14 表示的为经过 Canny 检测之后的图像。可以看出经过 Canny 边缘检测的图像过滤了很多无关内容，并且检测出边缘效果上比 Sobel 算子好出许多。由于加入了阈值操作，可以通过调整阈值来调整处理过程中过滤掉的像素点数量。在将阈值拉到最高之后，所得到的图像近乎就是边缘图像。当然，需要得到更为精确的边缘还需要进行后续的操作处理。

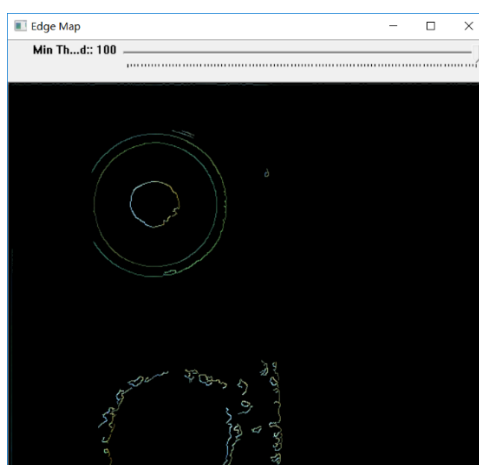


图 3.14 Canny 边缘检测

### 3.4 霍夫变换

霍夫变换分为两种，即霍夫线变换以及霍夫圆变换。霍夫线变换一般是用来寻找图像中的直线的，检测的方法类似阈值操作，采集灰度值超过预定值的像素点，之后进行连线。若要采用霍夫变换，要先用各种方法对图像二值化，并进行

一些初步的边缘检测，即霍夫线变换的输入只能是边缘二值图像。

根据所学的知识，要表示一条在二维空间的直线可以借由两个变量。例如标准坐标系中可由变量参数  $(x, y)$  表示，在极坐标系中可由参数极径和极角  $(r, \theta)$  表示<sup>[9]</sup>。在霍夫变换中，用的是极坐标系来表示一条直线。因此，直线的表达式如式 (3-9) 所示：

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right) \quad (3-9)$$

化简如式 (3-10) 所示：

$$r = x \cdot \cos\theta + y \cdot \sin\theta \quad (3-10)$$

一般来说，对于所有通过点  $(x_0, y_0)$  的直线，我们可以将它们统一用式 (3-11) 表示：

$$r_0 = x_0 \cos\theta + y_0 \sin\theta \quad (3-11)$$

式 (3-11) 中，每当给出一个  $(r_0, \theta)$  值就可以得到一条通过点  $(x_0, y_0)$  的直线。如果给出一个定点  $(x_0, y_0)$  借由式子 (3-11)，我们可以得到所有通过该点的直线，并且还可以得到一条正弦曲线。对于图像上的所有的点，都采用以上的方法进行处理。若在图像中两个不同的点经过处理后所得到的的曲线相交于平面  $r-\theta$ ，它们肯定是通过同一直线的。一般来说，我们可以统计在平面  $r-\theta$  上汇聚在一点上的曲线数量，这样就可以得到检测直线的结果。当检测到相交于同一点的曲线数量越多，就表示组成这一条直线的点也就越多。通过对阈值的调整，我们可以规定一个范围，当高于这个数值的点被检测到，那么就认为所检测的是一条线，这就是霍夫变换的原理。它检测的是图像中每个点对应的在平面上的曲线间交点，当相交于同一点的曲线数量超过了所设置的阈值，这个交点就可以被认为所是代表参数对  $(\theta, r_0)$  在原图像中为一条直线。用霍夫线变换对图像处理之后的结果如图 3.15 所示。



图 3.15 霍夫线变换



霍夫圆变换的原理可以参考之前提到的霍夫线变换的原理，但需要注意的是点对应的 $(r, \theta)$ 的空间被三维的圆心点  $x$ 、 $y$  还有半径  $r$  空间取代。我们可以用参数极径极角 $(r, \theta)$ 来表示一条直线。但如果想表示一个圆，这两个参数是不够的。若想用参数表示圆，所需要的必须三个参数，而三个参数正如上文提到过的就是 $(x, y, r)$ ，这三个参数可以用来表示空间中的任意一个曲线。参考二维空间上的霍夫线变换，对于多个边缘点，越多这些点对应的三维空间曲线交于一点那么他们经过的共同圆上的点就越多，参照霍夫线变换，我们也可以通过设置一个合理的阈值来检测圆，这就是我们所说的霍夫圆变换的原理。但是这会带来极大的计算量，为了能够能快地进行检测圆的工作，OpenCV 中所运用的是比霍夫圆检测更加方便快速的方法：霍夫梯度法。霍夫梯度法的依据是圆心一定是在圆上的每个点的模向量上，这些圆上点模向量的交点就是圆心。霍夫梯度法的第一步就是找到这些圆心，这样三维的累加平面就又转化为二维累加平面。第二步根据所有候选中心的边缘非 0 像素对其的支持程度来确定半径。图 3.16 为霍夫圆变换前后的对比图。在处理后的图像中，圆形部分用线条标出，并且在圆心以绿色标出。

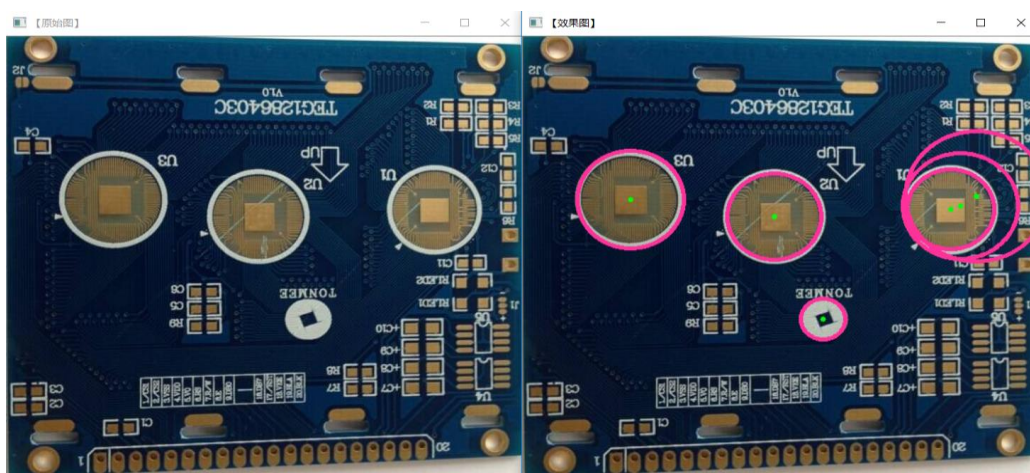


图 3.16 霍夫圆变换

### 3.5 直方图均衡化

对于直方图均衡化，首先要了解的是图像直方图的定义。图像直方图是采用图像化的形式展示了图像中各像素灰度值的分布，它直观地表示出每一个强度的像素灰度值含有的像素点数量。直方图均衡化的作用就是统计出主体的像素分布规律并进行针对性的强化，用以加强图像对比度<sup>[10]</sup>。在一张图像中，每个像素点

的灰度值大小分布是不同的，但总有大部分像素点灰度值集中在一定区域上，而直方图均衡化就是扩展这个范围，使图像的对比度更高。

均衡化指的是把一个分布(给定的直方图)映射到另一个分布(一个更宽更统一的强度值分布)，所以强度值分布会在整个范围内展开。要想实现均衡化的效果，映射函数应该是一个累积分布函数。要使用其作为映射函数，我们必须对最大值为 255(或者用图像的最大强度值)的累积分布进行归一化。最后，使用一个简单的映射过程来获得均衡化后像素的强度值。

图 3.17 所表示的就是经过直方图均衡化的图像。在经过处理之后，亮暗交界的地方出现了更加明显的边界，为之后的处理做准备。

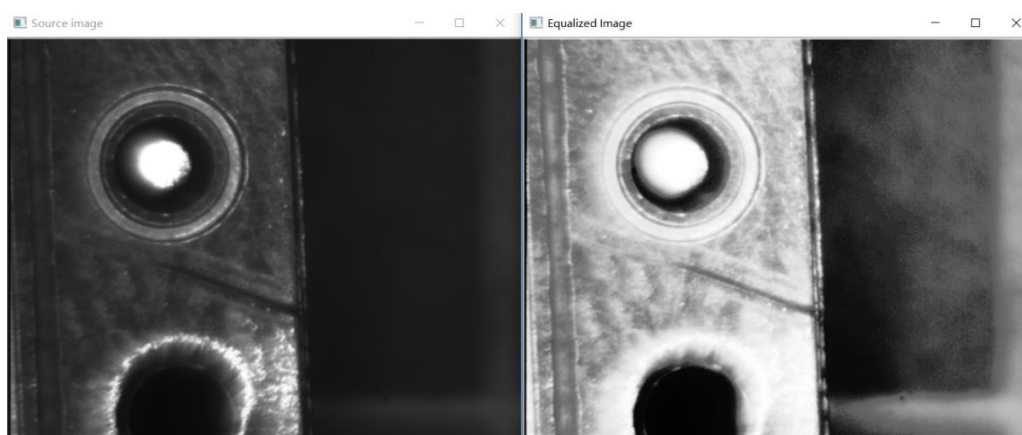


图 3.17 直方图均衡化

## 第四章 OpenCV 在 PCB 缺陷检测中的运用

### 4.1 模板匹配法

#### 4.1.1 模板匹配法介绍

模板匹配法检测 PCB 缺陷就是通过将已有 PCB 缺陷的模型图像导入算法中，通过将模板与待测 PCB 上的图像特征进行对比来找出 PCB 上存在的缺陷。要使用模板匹配来找 PCB 上的缺陷，首先需要得到标准的缺陷图像。每一块 PCB 上的缺陷都会有不同的特征，因此我们需要找到最有代表性的特征，这样才能便于我们更加准确地找到 PCB 上这类缺陷的存在。在得到缺陷图像之后，通常会将缺陷的模板图像进行一些处理，包括图像的二值化、消除噪声等操作，同时通过阈值操作过滤掉一些不必要的图像特征<sup>[11]</sup>。大致处理过程如流程图 4.1 所示：

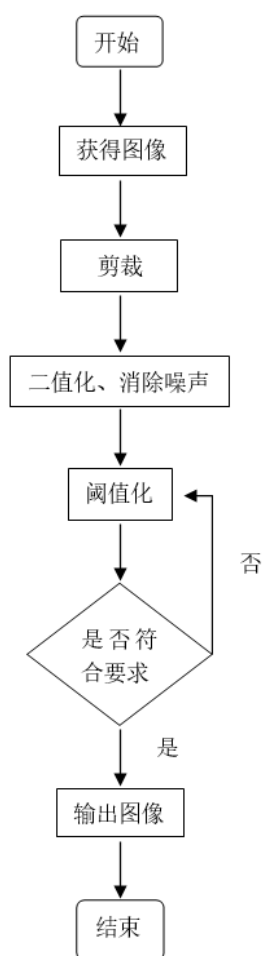


图 4.1 模板匹配流程图

模板匹配就是要根据模板图片找到待测 PCB 中最符合特征的部分。为了将模板图像与原图像进行匹配，需要将模板图像进行移动，即一个像素点一个像素点地将模板图像滑动，然后对比两块区域的像素点特征。在模板图像扫完整幅原图像后，若得到一个相似特征区域，则该 PCB 存在此类缺陷。

模板匹配算法一共有 6 中模式。接下来介绍这六种模式的使用情况，在以下公式中，以  $I$  表示原图像，以  $T$  表示模板图像。第一种为平方差匹配。在平方差匹配中，将 `method` 赋值为 `CV_TM_SQDIFF`。平方差匹配就是通过像素点灰度值的平方差数值对比来判定两个区域图像的相似度，所采用的公式如式(4-1)所示：

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2 \quad (4-1)$$

在平方差操作之后，每次对比的平方差值会进行对比，保留较小的一个，当对比完所有区域之后，仅保留最小的平方差区域。最理想情况下，平方差为 0，也就表示两个区域图像是完全相同的。匹配度越差，平方差也就越大。当平方差达到一定值的时候，就代表区域内没有与模板匹配的缺陷。在实际检测的时候，由于光源、待测板等多种因素的影响，需看情况设定范围，保证使用该算法不会误把正常区域检测为缺陷区域。

第二种匹配的模式为标准平方差匹配。在标准平方差匹配中，需要将 `method` 赋值为 `CV_TM_SQDIFF_NORMED`。标准平方差匹配的原理与平方差匹配基本是相同的，只是使用的公式略有差别，但经过多次实验发现，两种方法的检测在结果上并没有太多不同之处，基本同时锁定在同一个区域。在标准平方差匹配中，采用的公式如式(4-2)所示：

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (4-2)$$

与平方差匹配法的不同之处就是所处理的结果是用标准平方差表示的。同样，用这种方法匹配之后，得到的处理结果为值最小的标准平方差，即相似度最高的区域部分。

第三种模式为相关匹配法。使用相关匹配法时，`method` 的赋值为 `method=CV_TM_CCORR`。在相关匹配法中，采用的是将原图区域与模板图像相乘的操作，通过对比乘积的大小来比较两个区域的相似程度。在相关匹配法中，采用的公式如式(4-3)所示：

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y')) \quad (4-3)$$

在相关匹配法中由于采用的是相乘操作，在相乘结果越大的时候表示这两个区域灰度值较大与灰度值较小的部分重合度比较大。但也存在着一个很大的问题，若原图中整体的灰度值都偏大，则检测的结果未必是与模板匹配的部分，因为该算法中仅仅计算了灰度值的乘积。在实际使用这种算法的时候，发现这种算法在灰度值有明显差别的图像中表现较好，在差别不大的图像中经常出现识别错误的情况。

接下来是标准相关匹配。在标准相关匹配中 `method` 的值为 `CV_TM_CCORR_NORMED`。标准相关匹配与相关匹配的关系与标准平方差匹配和平方差匹配的关系相同，都只是在处理结果的数值上存在一些差异。而这两种匹配方式与之前两种匹配方式在预期结果上则是相反的，之前两种希望得到的是较小的数值，而这两种方式希望得到的则是更大的数值。在标准相关匹配中，用到的公式如式（4-4）所示：

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}} \quad (4-4)$$

由于标准相关匹配与相关匹配在思路基本一致，因此在识别的模式上也是相同的。由于数值上的差异会导致识别的位置有偏差，但基本差别不大。同时，因为之前分析的缺陷依旧存在，在识别准确率上，标准相关匹配也不是用与整体灰度值较大的图像。

第五种模式为相关系数匹配。在相关系数匹配中，`method` 设定值为 `CV_TM_CCOEFF`。进行相关系数匹配的时候，首先对模板的灰度值取均值的相对值，然后取待测图像区域的均值相关值，通过这两个值进行匹配。在相关系数匹配中，采用的公式如下所示：

$$R(x, y) = \sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y')) \quad (4-5)$$

其中  $T'(x', y')$  和  $I'(x + x', y + y')$  分别如式（4-6）和式（4-7）所示：

$$T'(x', y') = T(x', y') - 1/(w \cdot h) \cdot \sum_{x'', y''} T(x'', y'') \quad (4-6)$$

$$I'(x + x', y + y') = I(x + x', y + y') - 1/(w \cdot h) \sum_{x'', y''} I(x + x'', y + y'') \quad (4-7)$$

在相关系数匹配中，匹配结果为 1 时表示完美匹配，结果为 -1 是表示匹配结





运行程序之后，由于之前导入过相关模板，在面板上自动定位到了缺陷位置并进行显示，结果如图 4.3 中的 (a)、(b)、(c)、(d)、(e)、(f) 所示。

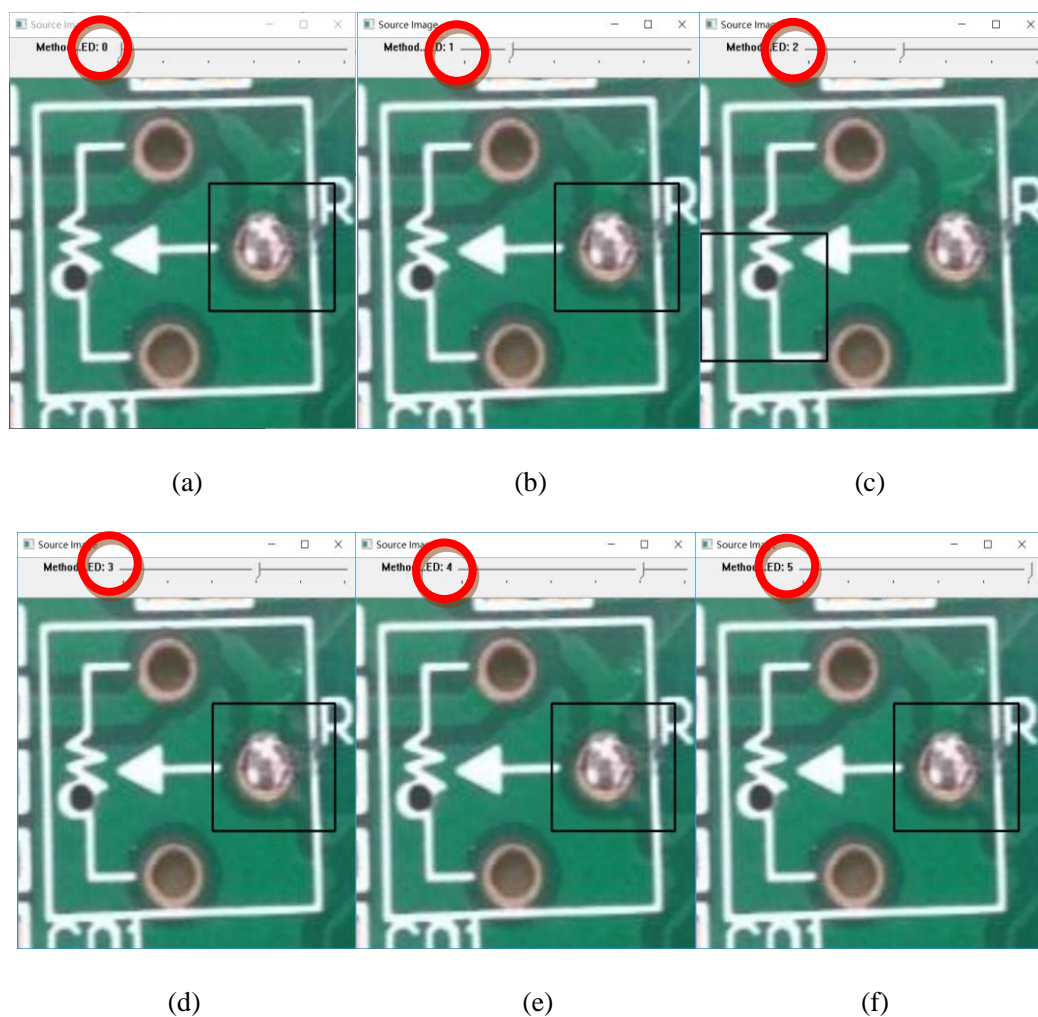


图 4.3 通孔堵塞检测

以上的六个结果是采用不同的方法对缺陷进行定位的结果，可以看出，除了在采用相关匹配法（图 4.3 中的 (c)）的时候定位出现了偏差，其他方法中都对缺陷进行了有效的定位，而且定位的结果基本是相同的。经过了对不同的 PCB 采用同样这几种方法进行检测，结果基本相同。因此表明，我们可以通过采用模板匹配的方法对 PCB 上的通孔堵塞缺陷进行有效地定位检测。但需要注意的是不宜采用相关匹配函数处理图像。

接下来讨论的是该算法对于短路的检测工作。在上文中解释过，由于 PCB 设计问题的短路无法通过图像检测识别，在这里检测的 PCB 短路为焊锡连接两条线路导致的短路问题。PCB 的短路问题比较难以准确识别，因为缺陷的特征不是很明显。众所周知，在 PCB 上通常会印上很多标注，而这些标注通常是白色的。在

光源照射情况下,缺陷位置显示也是白色的,所以通常会错误检测。图 4.4 显示的是 PCB 短路的检测结果。



图 4.4 PCB 短路检测

从检测的图像中可以看出,还是在使用相关匹配法的时候识别出现了偏差。不仅如此,在采用不同方式对同一出缺陷进行识别是,识别的位置会有些许差别。图中的部分边上就有标注的白字,容易在识别的时候照成干扰,导致一些误差。但是,几种算法基本可以将位置定在缺陷部位,若对算法加以完善,应该可以避免这些误差,因此可以考虑在检测短路缺陷的采用模板匹配算法。

## 4.2 Mark 点识别

Mark 点是 PCB 上非常重要的部分,PCB 上每一个元器件的位置都是根据 Mark 点进行放置的。在识别出缺陷之后,需要根据该部分到 Mark 点的距离来反馈缺



陷的坐标，因此 Mark 点的识别工作也不可忽视。通常情况下，Mark 点为实心圆，以裸铜为材料进行制作。Mark 点的表面非常平整、光滑，在光源之下，会与背景色有明显的差别，因此很方便用视觉检测的方法进行识别。Mark 点识别的流程如图 4.5 所示：

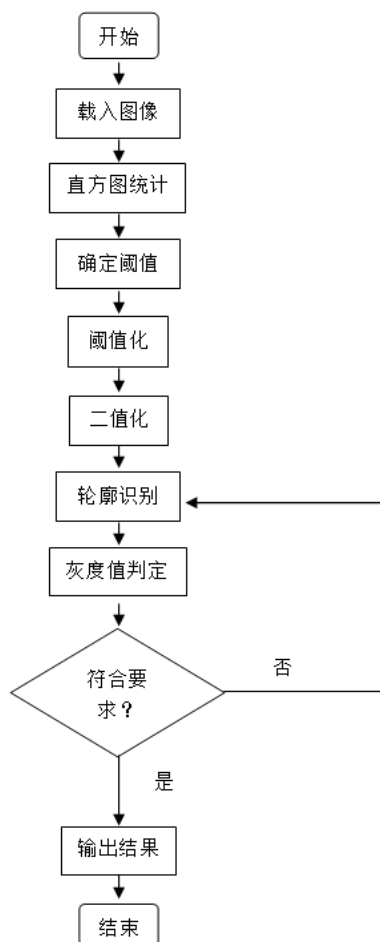


图 4.5 Mark 点识别流程图

识别的结果如图 4.6 所示。

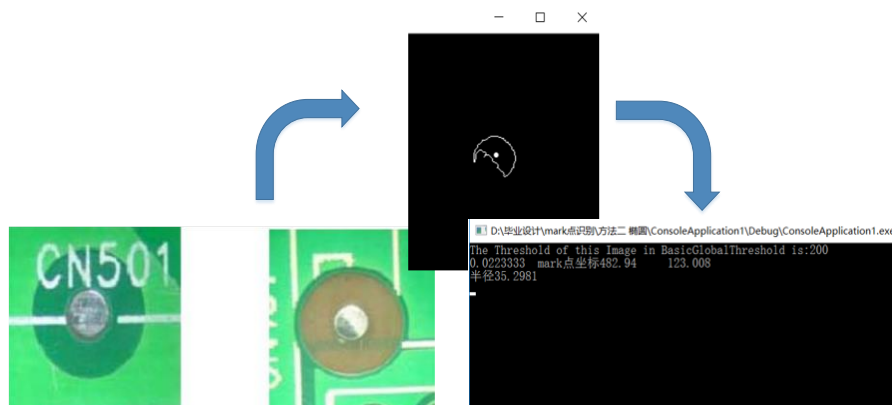


图 4.6 Mark 点识别

运用此算法，可以准确检测出 Mark 点的位置并且记录下 Mark 点的坐标。之后检测其他 PCB 缺陷的时候，就可以以这个坐标作为基准，记录并输出缺陷位置距离 Mark 点的相对坐标，便于之后的检修工作。

### 4.3 小结

PCB 的缺陷检测是一件复杂的工作，在图像处理上会遇到各种各样的干扰因素。在图像获得后，图像的噪声、污点、光源等等都会对检测结果造成影响。

在本章中，主要介绍了以模板匹配的方法对 PCB 上的缺陷进行定位识别。模板匹配主要是通过标准模板的灰度值特征对待测 PCB 图像进行扫描对比，在匹配到相似度高于要求的区域时，确定该区域存在 PCB 缺陷。在识别前，需要对图像中的干扰因素做相应的处理，例如对图像进行消除噪声、二值化等操作，达到更高识别精度的目的。从实验结果上来看，六种模式的识别方法中，除了相关匹配法的使用效果不佳外，其他几种方法均能有效地识别出 PCB 上的缺陷位置，并进行标识。此外，由于检测的目的是为了之后的检修工作，所以需要将缺陷的位置进行确定。这就需要获得一个坐标体系，用于记录缺陷的坐标。在此，就参考了贴片流程中的 Mark 点定位方法，通过对 Mark 点的识别以及定位，确定了 PCB 的坐标体系，在检测出缺陷之后，可以以 Mark 点位参考，确定缺陷的坐标位置。

## 第五章 总结与展望

在本文中，主要介绍的是基于视觉检测技术的 PCB 缺陷识别。首先 PCB 的缺陷类型非常多，想要通过视觉检测的方法进行识别，必须选择有图像特征的缺陷才行，因此在文章刚开始就对缺陷进行了分析和分类。在对缺陷分析之后，确定了积累可以采用视觉检测识别的缺陷类型。这一步工作是必不可少的，因为若想采用一种方法检测所有的缺陷类型，必然会造成检测的准确性无法达到要求。只有充分了解了各种 PCB 的缺陷，并将其分类，才能更好地完成后面的设计工作。

接下来介绍了几种常见的视觉算法。这几种算法经常在各类应用中出现，他们可以同时被运用来处理图像。而在之后的 PCB 缺陷检测中，也综合使用了这几种算法。可以说视觉算法是运用视觉检测系统的基础，了解各种算法的功能以及使用方法才能进行自己的功能设计工作。

最后介绍的是基于视觉检测技术的 PCB 缺陷识别，主要介绍的是模板匹配方法。这种方法就是以导入的缺陷模板，获得模板中缺陷类型的特种，然后与待测 PCB 进行像素匹配。若匹配到相似度高的区域，则认为检测到了 PCB 缺陷。在实际实验中，发现检测的准确性基本满足要求，但容易受到环境以及 PCB 本身的影响。

上述方法虽然能对几种 PCB 缺陷进行有效识别，但是 PCB 缺陷种类众多，检测出来的部分仅占全部缺陷类型中的小部分。在模板匹配识别中，需要导入各种各样缺陷类型的模板以匹配不同的缺陷，并且对模板有很高的要求，模板的质量越高，识别的准确性就越高。后续可通过修改模板，优化计算步骤，以达到更快识别的目的。除了模板匹配外，还有其他算法可以使用，例如采用颜色的差别识别缺陷，在某些缺陷类型也是适用的。PCB 的缺陷检测是非常值得进一步研究的技术。由于现在电子科技发展越来越快，致使 PCB 设计要求越来越高，对于 PCB 的制作精度也越来越高。达到高精度 PCB 的前提就是对制作出的 PCB 有很好的方法进行缺陷的检测。在本文提出的缺陷检测算法仅仅是一个基础，若想更快、更精准地定位 PCB 缺陷，需要做更多的工作。相信在不久的将来，PCB 能够实现自动全板识别，为电子制造领域带来更多的便利。



## 致谢

这次毕业设计是在本专业的指导老师高老师的指导之下完成的，感谢高老师为我这段时间提供实验室以及实验器材，这些为我的工作带来了很大的方便。每周老师都会来实验室检查并指导我的工作，使我不会在设计的过程中出现方向的错误，在我没有思路的时候辅导我找到灵感。

感谢学校提供了这么好的机会供我学习，也感谢与我一起并肩作战的同学们，谢谢！



## 参考文献

- [1] 陈亮, 基于机器视觉的 PCB 裸板缺陷检测方法研究, 浙江工业大学硕士学位论文, 2011 年 12 月
- [2] 张珂, PCB 缺陷自动检测系统的研究与设计, 北京交通大学硕士学位论文, 2008 年 6 月
- [3] 姚忠伟, 基于机器视觉的 PCB 缺陷检测算法研究, 哈尔滨工业大学硕士学位论文, 2012 年 12 月
- [4] 孟繁丽, 基于机器视觉的 PCB 裸板自动检测方法研究, 2016 年 9 月, 第 9 期, page: 13
- [5] 高宏伟、张大兴、王卫东等, 电子制造装备技术, 西安电子科技大学出版社, 2015 年 9 月
- [6] 孙晓婷, PCB 视觉检测系统研究, 中南大学硕士学位论文, 2008 年 6 月
- [7] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 电子工业出版社, 2014
- [8] 张铮、徐超、任淑霞, 数字图像处理与机器视觉, 人民邮电出版社, 2014 年 5 月
- [9] 刘雨, 基于机器视觉的 PCB 表面缺陷检测, 安徽工业大学硕士学位论文, 2016 年 6 月
- [10] 胡文娟, 基于机器视觉的 PCB 光板缺陷检测技术研究, 武汉理工大学硕士学位论文, 2007 年 4 月
- [11] 刘百芬、李海文、张姝颖, 基于机器视觉 PCB 裸板缺陷自动检测研究, 2014 年 7 月, 27 卷, page: 8
- [12] MA Can, Mao JianXu, Mao JianPing, Research and develop on PCB defect intelligentvisual inspection robot , 2010
- [13] Malge P.S., Nadaf R.S., PCB Defect Detection, Classification and Localization using Mathematical Morphology and Image ProcessingTools, February 2014, Volume 87 – No.9