

浙江大学

本科生毕业论文（设计）



题 目 DataMatrix 快速定位算法研究

姓名与学号 任成斌 3090000375

指导教师 刘旭 教授

年级与专业 信息工程(光电) 09级

所在学院 光电信息工程学系

浙江大学本科毕业生毕业论文（设计）诚信承诺书

- 1.本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。
- 2.本人在毕业论文（设计）中引用他人的观点和参考资料均加以注释和说明。
- 3.本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。
- 4.在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

班级 光电0903 班级 3090000375

毕业论文（设计）作者签名 _____

2013 年 05 月 26 日

摘要

二维条码(2D Barcode)是用某种特定的几何图形按一定规律在平面(二维方向上)分布的黑白相间的图形记录数据信息的符号。不同于一维条码，其可以在纵向和横向两个方位同时表达信息，因此具有很大的信息密度。二维条码可以分为矩阵式和堆叠式二维条码。目前世界上主流的二维条码，有QR码(快速响应矩阵码)、Data Matrix码、PDF417码等。其中Data Matrix码由于可以在很小的空间内记录较多的信息，因此在工业流水线上有较多的应用。

Data Matrix码由于其编码特点，识别较慢。现有的基于Hough直线检测的定位算法速度较慢，不能满足工业上实时性要求。本文提出了一个快速定位算法，并且基于OpenCV完成了设计和测试。经过实验，证明本算法可以满足工业实时性要求。

同时，本文对二维码识别技术中涉及到的编码、图像处理以及开发用OpenCV都做了研究。

关键词： Data Matrix 二维码 定位算法 图像处理

Abstract

The two-dimension barcode (2D barcode) is a symbol which record data information using black and white graphics modules with a certain regular in two-dimension directions (horizontal and vertical). Unlike one-dimension barcode, 2D barcode can express information in two directions, which determines its higher data density. The 2D barcode can be categorized into two classes, matrix mode and stack mode. And the most popular 2D barcode present are QR code, Data Matrix code, PDF417 code, etc. Data Matrix Codes can store many information in small area, so it's very popular in industry.

Because of it's special decode rule, the reading speed of Data Matrix is slow. Those algorithms that based on Hough Transform cannot meet industries' time need. We develop a fast localizing algorithm and deployed on OpenCV. The experiment result shows that this algorithm can meet the real-time need.

Meanwhile, we also researched the topics of decoding rule, image processing and OpenCV that related to 2-D barcode reading.

Keywords: Data Matrix 2-D barcode

目 录

摘要	i
Abstract	ii
目录	iii
第一章 绪论	1
1.1 引言	1
1.2 二维条码技术简介	2
1.3 Data Matrix条码简介	2
1.3.1 Data Matrix条码结构	2
1.3.2 Data Matrix编码规则	4
1.4 二维条码研究现状	4
1.5 二维条码识别技术	6
1.5.1 图像预处理	6
1.5.2 条码初定位	7
1.5.3 条码精确定位	7
1.5.4 条码图像纠偏	7
1.5.5 RS纠错	7
1.5.6 0/1识别和解码	7
第二章 二维条码中的图像处理方法和定位技术	10
2.1 Ostu法（大津法）	10
2.2 边缘检测方法	10
2.2.1 微分算子法	11
2.2.2 拉普拉斯高斯算子法	11
2.2.3 Canny算子法	11
2.2.4 边缘提取效果对比	12
2.3 形态学运算	12
2.3.1 膨胀与腐蚀	12
2.3.2 开运算与闭运算	12

第三章 OpenCV机器视觉库介绍	17
3.1 OpenCV简介	17
3.2 OpenCV发展历史	17
3.3 OpenCV基本结构	18
3.4 OpenCV数据结构	18
3.5 OpenCV基本操作	19
3.5.1 矩阵操作	19
3.5.2 基本图像操作	19
3.5.2.1 图像读取	19
3.5.2.2 图像翻转	20
第四章 Data Matrix快速定位算法研究	21
4.1 主要思路	21
4.2 粗定位算法	22
4.3 精确定位算法	23
4.4 实验结果	24
4.4.1 定位效果	24
4.4.2 定位时间	27
第五章 总结	29
5.1 工作总结	29
5.2 研究展望	29
参考文献	31
致谢	34

第一章 绪论

1.1 引言

自动化数据采集技术是信息采集和处理的关键技术。随着工业自动化和数字化的发展，自动识别技术有了更加广泛的应用，极大地提高了各行业的生产效率。条码识别技术是计算机图像处理、模式识别、光电信息技术和通信技术为基础的综合技术，已经成为现在数据自动输入、识别的重要方法和途径[1]。

条码可以分为一维条码（One Dimensional Code）和二维条码（Two Dimensional Code）两类[2]。一维条码容量有限，仅能标识商品，不能存储更多信息，必须依赖数据库。二维条码相比之下，容量大大提升。同时，由于加入的纠错编码，鲁棒性更强，可靠防伪。二维条码可以用来表示多种语言、图像数据，不仅可以将字符信息、数字编译存储，而且可以将个人的生物信息：如人脸数据、指纹和虹膜等图像信息存入条码。[3]

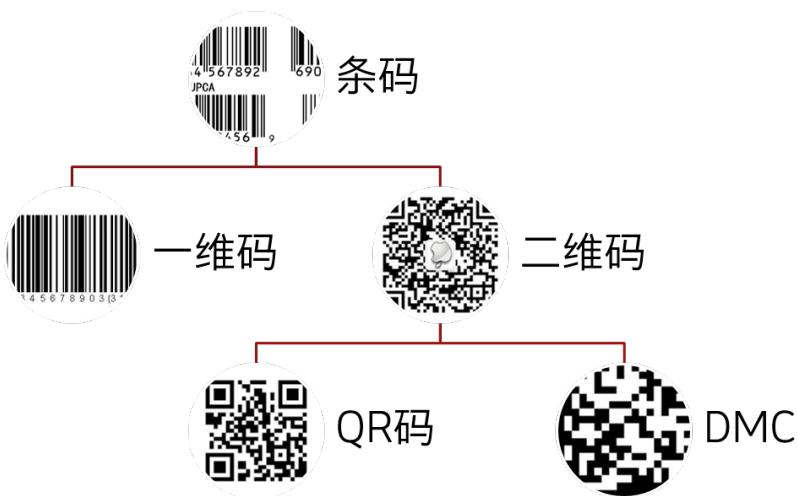


图 1.1: 常见条码

一维条码只有校验功能，即检验识别的信息是否正确。二维条码由于引入了纠错条码，具有很强的错误校验和纠正功能。在条码部分损毁、有污点的情况下，也可以被正确恢复和识别[4]。

二维条码按照不同的规定，按照一定规律在平面上分布黑白相间的图形用以记录数据信息。按照模块的深浅堆积方式不同，分为矩阵式和堆叠式两种。矩阵式条码将不同深浅的模块排列在图像中；堆叠式二维码采取行排列，将一维条码堆叠在一起。

在现阶段，应用较为广泛的二维条码主要有两类：QR码和Data Matrix码。在工业上应用较多的是后者[5]。

1.2 二维条码技术简介

从20世纪后期开始，二维条码由于其独特的优势，开始受到许多国家的关注。现在，已经有专门的机构负责制定条码标准。我国目前也在现在已经制定完成的二维码标准之上进行了改进，以适应不断更新的需求。

最早的堆叠式二维码——PDF147——由美国发明。之后，各种二维码相继面世：专门为邮政系统设计的CodeOne和QR码等[1]。现在，二维码不光用在证件信息管理中，在商业、餐饮、交通运输等部门的物流和工业生产线上的自动化管理部分也有着十分重要的应用。现在，还有金属条码、纤维织物条码以及三维条码等多种新型码都在研发之中，进一步扩大了条码的应用范围和前景。

我国对于二维码的研究开始于1993年，相对落后与发达国家。目前，国内二维码的标准还不够完善，也没有独立的解码技术。随着信息时代数据量的增加以及市场的驱动，国内对二维码技术的需求与日俱增，相关研究也日益增多。现在，我国的身份证件系统[6][7]、互联网身份识别等领域已经开始对二维码有大规模的使用。

1.3 Data Matrix条码简介

1.3.1 Data Matrix条码结构

Data Matrix原名Datacode，由美国国际资料公司(International Data Matrix, 简称ID Matrix)于1989年发明[8]。

Data Matrix是一种矩阵式二维条码，其发展的构想是希望在较



图 1.2: Data Matrix码

小的条码标签上存入更多的资料量。Data Matrix的最小尺寸是目前所有条码中最小的，尤其特别适用于小零件的标识，以及直接印刷在实体上。

Data Matrix码（以下称做DM码）的符号结构由寻边区和数据区组成。寻边区包括两部分：L字型实心边界和铁道线。寻边区是DM码的边界。L字型边界用来限制条码的物理边界，确定DM码的位置和角度。铁道线用来确定模块的单元结构，同时可以辅助确定条码的物理尺寸以及形变。条码中间的编码区包含着编码信息，按照一定的编码规则，可以包括数字、字母和汉字等信息[9]。

DM码由黑白两种颜色的模块构成，每一个模块被称作一个数据单位，分别代表着0和1。

DM码有ECC000-140和ECC200两种类型，ECC000-140具有多种不同等级的纠错功能，ECC200通过Reed-Solomo纠错码产生多项式计算出纠错码。根据尺寸的不同，包含的容量两也不同。

选项	特性
符号尺寸	ECC000-140: 9*9 49*49, 仅为奇数 ECC20: 10*10 144*144, 仅为偶数
可编码字符集	0 127, 全部128个ASCII字符 128 255, 拓展ADCII字符
数据类型与容量	数字数据: 3166个字符 文本字符: 2335个字符 8位字节数据: 1556个字符
数据类型与容量	ECC000-140: 四个等级卷积纠错 ECC200: Reed-Solomon纠错
附加性能	反转映像: 深色背景上的浅色区域或者浅色背景上深色区域均可 扩充解释: 仅ECC200可选, 使得符号进行针对性编码 结构化追加: 仅ECC200可选, 允许一个数据文件以最多16个字符表示

表 1.1: Data Matrix编码特性

ECC200由于其算法容易, 并且尺寸有弹性, 应用较为普遍。

1.3.2 Data Matrix编码规则

Data Matrix 符号的数据区由规则排列的方形深浅模块构成。数据区的四周是探测图形(Finder Pattern), 探测图形外则是空白区, 如图 (1.2) 所示。探测图形为一个模块宽度, 是数据区的边界, 其中两条邻边为暗实线, 形成了一个L型边界, 用于限定物理尺寸、定位和符号失真。另两条邻边由交替的深色模块和浅色模块组成, 主要用于限定符号的单元结构, 也能帮助确定物理尺寸及失真。

Data Matrix编码特性如表 (1.1) 所示。

1.4 二维条码研究现状

Data Matrix码是使用最为广泛的二维条码之一。由于其最小尺寸是目前所有二维码中最小的, 特别适合于小零件的标记以及直接印刷。因此现在Data Matrix在工业领域有着大量应用, 主要集中于

集成电路和药品的小物件以及制造业流水线的生产过程中。

之前SARJAKOSKI等人提出了将Data Matrix用于处理地图数据[10]，他们使用的DataMatrix算法主要是先进行图像滤波处理，再用Hough变换进行直线检测。基于这一思路，后续一部分工作在于对直线检测步骤的优化。

胡晓崇等人基于Java开发了一套DM码识别方案[5]，为以后将该系统移植到手机平台打下了基础。他们的算法主要包括：预处理、初定位和精确定位。采用Hough变换求取条码的旋转角度；他们的流程如图(1.3)所示。

邹沿新等人没有使用传统的边缘检测和直线检测手段。采用十字搜索法来得到“L”形边界[11]。此类算法受背景噪声、几何失真的影响较小，但时间复杂度较高。适用于小图像以及实时性需求不高的场所。但是在工业流水线上，此类算法不能满足速度要求。他们的算法识别包含单个条码的320*240的图像时间为5ms。

姚林昌等人同样没有采用Hough变换[12]。他们的算法首先进行矩形检测，大致确定Data Matrix条码位置，然后用距离为角度的函数将Data Matrix条码2D边界分析变成1D 波函数分析。他们的算法平均识别时间为12.06ms，识别率为99.22%，达到了工业现场实时性和可靠性的要求。

天津大学的刘峰（音译）等人提出了一个Data Matrix的鲁棒性很好的快速定位算法[13]。该算法采取两步定位，第一步通过Sobel滤波进行边缘检测，先大致确定条码所在区域，第二步对ROI进行扫描确定条码的精确边界。在精确定位过程中，他们采取了直接扫描的方式，对条码边界进行追迹。该方法速度较快，但是对于图像噪声水平有比较高的要求。

黄强等人也采用了两步定位算法[9]。他们首先采用L字型追迹方法确定条码的位置和角度，然后利用形态学处理确定条码的边界。他们的算法在复杂背景环境下定位精确度非常高，但是算法较为复杂，不能满足工业实时性要求。

除了传统的印刷Data Matrix，在流水线上主要使用的还有点状雕刻的Data Matrix（Dotted Data Matrix）。对于点状雕刻码，国内尚没有过多的研究。只有陈袁飞（音译）提出过一个扫描算法[14]。

他首先确定ROI，采用的是从图像中心向外扫描的方法。然后通过最远点确定条码边界。最后用了投影法对铁道线进行扫描，从而确定了条码模块的大小。该算法主要解决了工业照相机拍摄有角度的问题。

国外的Dita对这一问题有比较多的研究[8]。除了提出一个通用的算法[15] [16] [16] [17]，他还专门针对倾斜和变形的情况进行了研究[?]。他首先用形态学生长算法处理图像，从而使条码成为一个完整的矩形。然后通过对边界的追迹确定条码的倾斜角度，进而确定条码的顶点。对于由于工业照相机角度问题而造成条码倾斜的情况，他创造了一个映射点矩阵对条码进行覆盖，直接读取条码0/1信息。对于打印到曲面的条码，他利用曲面的角度和条码的大小等信息，可以构造一个映射矩阵，对条码进行覆盖读取。同时，他还提出了一种用Meanshift聚类算法对条码位置进行定位的方法。

Data Matrix码的传统定位方法需要通过L字型和铁道线对边界进行确定。常用的线检测算法往往耗时较长，并且在复杂背景下效果不好。

工业照相机由于环境限制，往往不能很好的对条码进行采集，经常会出现光照度不均匀、模糊、条码位置偏移等情况。这往往会给条码定位带来困难。

现在Data Matrix定位算法的一个限制在于时间。二维码识别速度是流水线效率的瓶颈。现在的算法定位300K图像中的一个Data Matrix 时间最快在5ms左右，解码时间在5ms左右。工业上对定位速度有更高的要求，以适应生产需求。

现在Data Matrix定位算法的另一个限制在于复杂环境下的识别。工业上采集系统的环境比较复杂，如图（1.4）特别是点状雕刻的条码，很难从背景中分离出来。因此对定位算法提出了很高的鲁棒性要求。

1.5 二维条码识别技术

1.5.1 图像预处理

主要是指对图像进行裁剪、降噪以及二值化操作，从而减少复

杂背景以及光照度不均匀对后期识别的影响。常用的降噪方法有邻域均值降噪、中值滤波和低通滤波等。在OpenCV实现中，还可以利用图像金字塔对图像进行降噪处理。二值化的方法主要是Otsu法（大津法）[18]。下文会详细介绍相关图像处理算法。

1.5.2 条码初定位

主要是在采集到的图像中初步确定二维条码所在的区域，并且将该区域从图像中分割开来，进行下一步处理。常用的初定位方法有连通域标记和形态学处理等。下文将会对相关算法进行详细介绍。

1.5.3 条码精确定位

利用不同条码各自的定位特征，对条码边界进行精确定位。在下文中对Data Matrix精确定位算法进行了更加深入的讨论

1.5.4 条码图像纠偏

在工业上，一般来说精确定位后的条码往往存在一定程度的偏转、变形等问题，需要通过一定的变换将条码调整到合适的位置和方向，方便下一步识别。常用的校正方法有仿射变换等。

1.5.5 RS纠错

Data Matrix采取了Reed-Solomon算法来进行校验和纠错，在很大程度上可以恢复由于损毁、污渍等状况造成的条码信息损失。由于本文主要是研究定位算法，因此对RS纠错算法没有过多的讨论。

1.5.6 0/1识别和解码

通过网格化，对条码信息进行读取，并且根据相应的编码规则，识别出条码中所包含的信息。

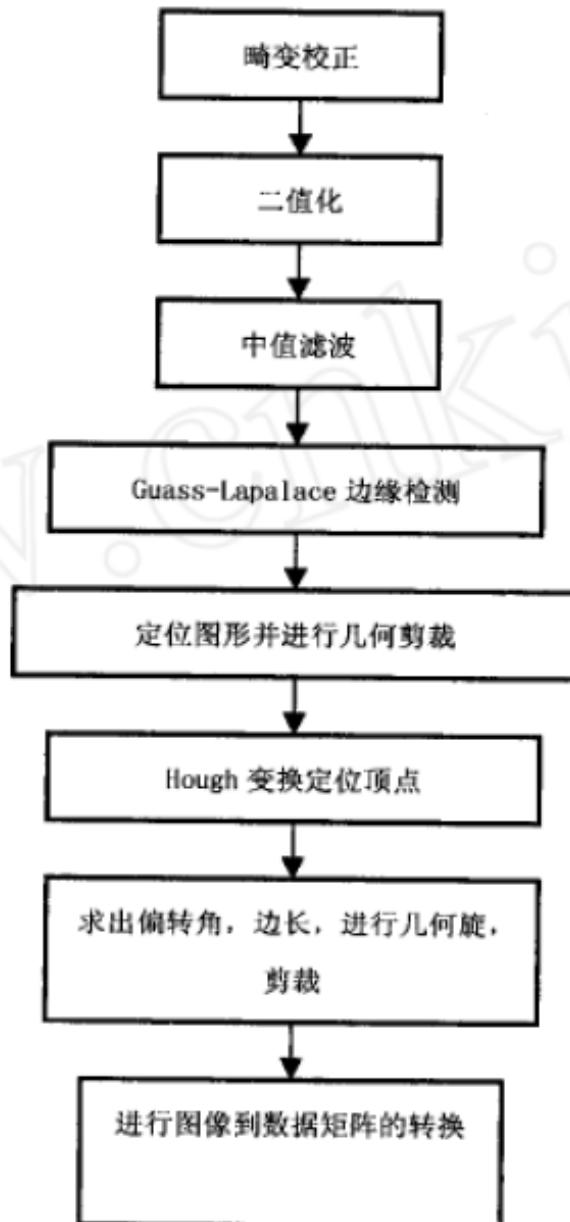


图 1.3: 胡晓崇算法流程图



图 1.4: 工业采集图像

第二章 二维条码中的图像处理方法和定位技术

2.1 Ostu法（大津法）

最大类间方差法是由日本学者大津展之于1979年提出的，是一种自适应的阈值确定的方法，又叫大津法，简称OTSU[18]。它是按图像的灰度特性，将图像分成背景和目标两部分。背景和目标之间的类间方差越大，说明构成图像的两部分的差别越大，当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。因此，使类间方差最大的分割意味着错分概率最小。

对于图像 $I(x,y)$ ，前景(即目标)和背景的分割阈值记作 T ，属于前景的像素点数占整幅图像的比例记为 w_0 ，其平均灰度 μ_0 ；背景像素点数占整幅图像的比例为 w_1 ，其平均灰度为 μ_1 。图像的总平均灰度记为 μ ，类间方差记为 g 。假设图像的背景较暗，并且图像的大小为 $M*N$ ，图像中像素的灰度值小于阈值 T 的像素个数记作 N_0 ，像素灰度大于阈值 T 的像素个数记作 N_1 ，则有：

$$\omega_0 = \frac{N_0}{M * N} \quad (2.1)$$

$$\omega_1 = \frac{N_1}{M * N} \quad (2.2)$$

$$N_0 + N_1 = M * N \quad (2.3)$$

$$\omega_0 + \omega_1 = 1 \quad (2.4)$$

$$\mu = \omega_0 * \mu_0 + \omega_1 * \mu_1 \quad (2.5)$$

$$g = \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu)^2 \quad (2.6)$$

将公式整理，可以得到：

$$g = \omega_0\omega_1(\mu_0 - \mu_1)^2 \quad (2.7)$$

采用遍历的方法得到使类间方差最大的阈值，即为所求。

2.2 边缘检测方法

边缘检测技术对于图像处理十分重要。通常边缘提取是将ROI (Region of Interest) 与背景分离出来的前提条件。现在的边缘检测

算法根据原理，可以分成三大类：一类是基于固定的局部算法，如微分算子法等。这类算法属于经典的边缘提取方法。第二类一般是以能量最小化为目标的全局提取方法。该类算法从全局最优的角度提取边缘，如神经网络分析法等。第三类则是最近发展起来的，以小波变换、形态学和分型理论等技术为代表。在二维码识别中，考虑到速度和实际需要，一般采取的是经典边缘提取方法。该类方法特点是处理迅速，虽然效果稍差，但已经能够满足工业上实际需要。

2.2.1 微分算子法

该类方法通过空域微分算子卷积来进行边缘提取。微分算子可以突出灰度变化。通过对导数值设置一定的门限，可以提取边界点集。

梯度模是一个连续函数 $f(x,y)$ 在 (x,y) 处方向导数的最大值。通过梯度模可以有效地进行边缘检测，它不仅仅具有导数不变性，还有各项同性。

2.2.2 拉普拉斯高斯算子法

拉普拉斯高斯（LOG）算法是一种二阶微分边缘检测方法[19]。通过寻找图像中灰度值二阶导数零点来检测边缘点。在实际应用中，一般会用高斯函数对图像进行滤波来降噪。

2.2.3 Canny算子法

边缘提取的重点在于增强边缘的同时不能引入更多的噪声。由于图像边缘和噪声在频域中都属于高频分量，因此简单的微分运算在增强边缘的同时往往也会引入更多噪声。一般在微分运算之前都会采取适当的平滑滤波。Canny 对此问题进行了分析[20]，并且从数学上推导出了4个指数函数线性组合形式的最佳边缘提取算子网。Canny 算子利用高斯算子的一阶微分，在增强边缘和抑制噪声之前找到了比较好的平衡。

2.2.4 边缘提取效果对比

从图（2.1）中可以看出，Roberts算子提取边缘的结果边缘较粗，不是十分精确。采用Sobel和Prewitt算子的结果对边缘的定位要好于Roberts算子。采用拉普拉斯高斯算子的结果明显优于前面三种算子，边缘较为完整，位置比较准确。Canny算子提取的边缘最为完整，连续性非常好。这主要是因为Canny算子采取了“非极大值抑制”和形态学操作。

2.3 形态学运算

形态学（morphology）一词通常代表生物学的一个分立。它是研究动物和植物的形态和结构的学科。数学形态学（mathematical morphology）作为工具可以从图像中提取出对于表达和描绘区域形状有用处的图像分量，如边界、骨架和凸壳等。形态学的常用操作包括膨胀、腐蚀、开/闭运算、击中与不击中运算等[21]。

2.3.1 膨胀与腐蚀

A和B是 Z^2 的集合，A被B膨胀可以表示为 $A \oplus B$ ，定义为：

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (2.8)$$

B集合通常叫做膨胀的结构元素。

图（2.2）是一个形态学膨胀过程。

A和B是 Z^2 的集合，A被B腐蚀可以表示为 $A \cdot B$ ，定义为：

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2.9)$$

同样的，B集合通常叫做腐蚀的结构元素。

图(2.3)是一个形态学腐蚀过程。

2.3.2 开运算与闭运算

膨胀使图像扩大，而腐蚀使图像缩小。在形态学中，另外还有另外两类重要的形态学操作：开操作与闭操作。开操作一般使得对

象的轮廓变得光滑，断开狭窄的间断和消除细的突出物；闭操作同样使得对象的轮廓变得更光滑，但与开操作相反，它通常消弭狭窄的间断和长细的鸿沟，消除小的空洞，并填补轮廓线中的断裂。

用结构元素B对A进行开操作，表示为 $A \circ B$ ，定义为：

$$A \circ B = (A \ominus B) \oplus B \quad (2.10)$$

因此，用B对A进行开操作就是用B对A进行腐蚀，然后用B对结果进行腐蚀。

图(2.4)是一个形态学开操作过程。

同样，使用结构元素B对集合A进行的闭操作，表示为 $A \bullet B$ ，定义如下：

$$A \bullet B = (A \oplus B) \ominus B \quad (2.11)$$

这个公式说明，使用结构元素B对集合A的闭操操就是用B对A进行膨胀，而后用B对结果进行腐蚀。

图(2.5)是一个形态学闭操作过程。

开操作有一个简单的解释。假设我们将结构元B看做一个（扁平的）“转球”。 $A \circ B$ 的边界通过B中的点完成，即B在A的边界内转动时，B中的点所能到达的A的边界的最远点。这个开操作的几何拟合特性使我们得出了集合论的一个公式。这个公式说明用B对A进行开操作是通过求取B在拟合A时的平移的并集得到的。就是说，开操作可以表示为一个拟合操作：

$$A \circ B = \cup\{(B)_z | (B)_z \subseteq A\} \quad (2.12)$$

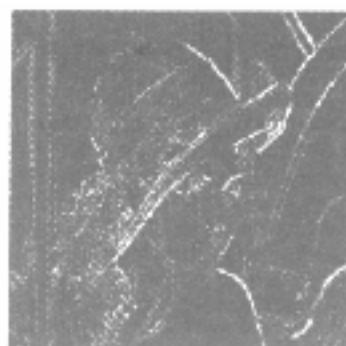
闭操作有着相似的几何解释。只是我们现在在边界的外部转动B。简而言之，开操作和闭操作是一对对偶操作，所以闭操作在边界外部转动球是预料之中的事情。



(a) 原始的 Lena 图像



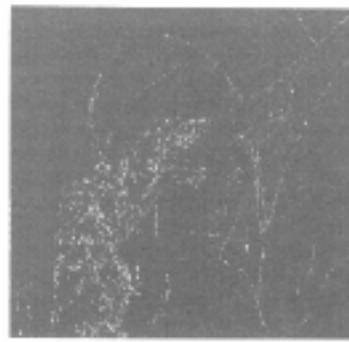
(b) Roberts 算子提取边缘图像



(c) Sobel 算子提取边缘图像



(d) Prewitt 算子提取边缘图像

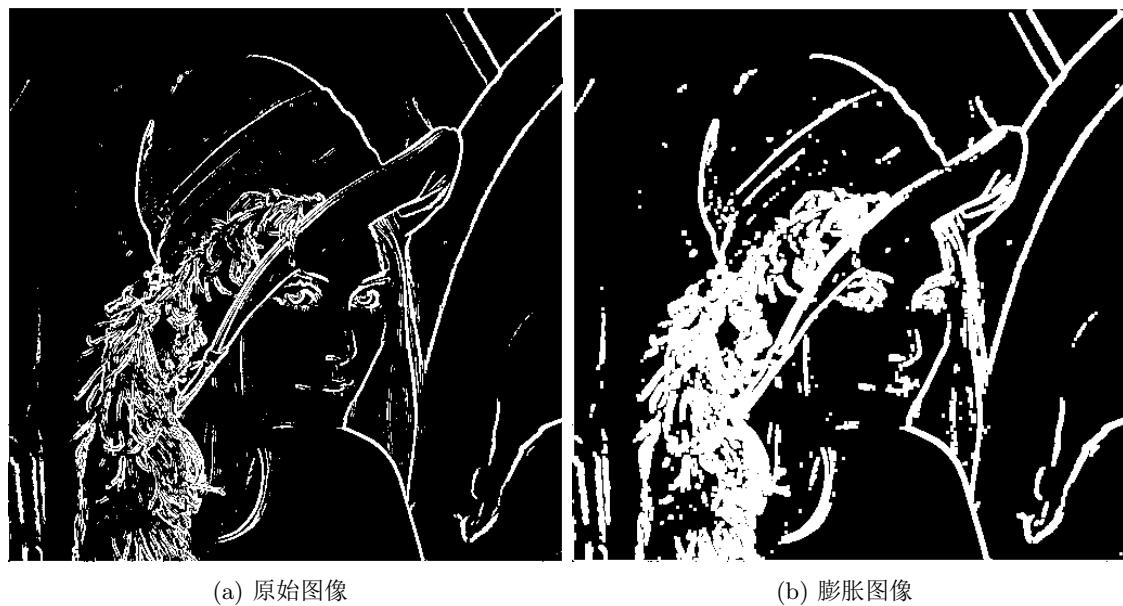


(e) LOG 算子提取边缘图像



(f) Canny 算子提取边缘图像

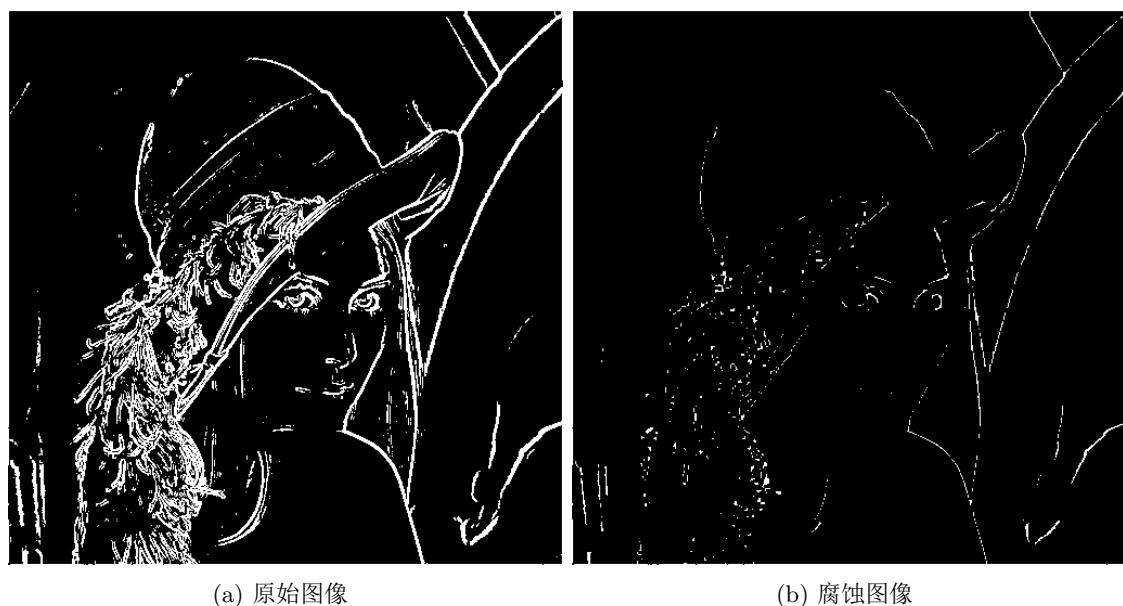
图 2.1: 经典边缘检测算子效果对比



(a) 原始图像

(b) 膨胀图像

图 2.2: 形态学膨胀效果



(a) 原始图像

(b) 腐蚀图像

图 2.3: 形态学腐蚀效果

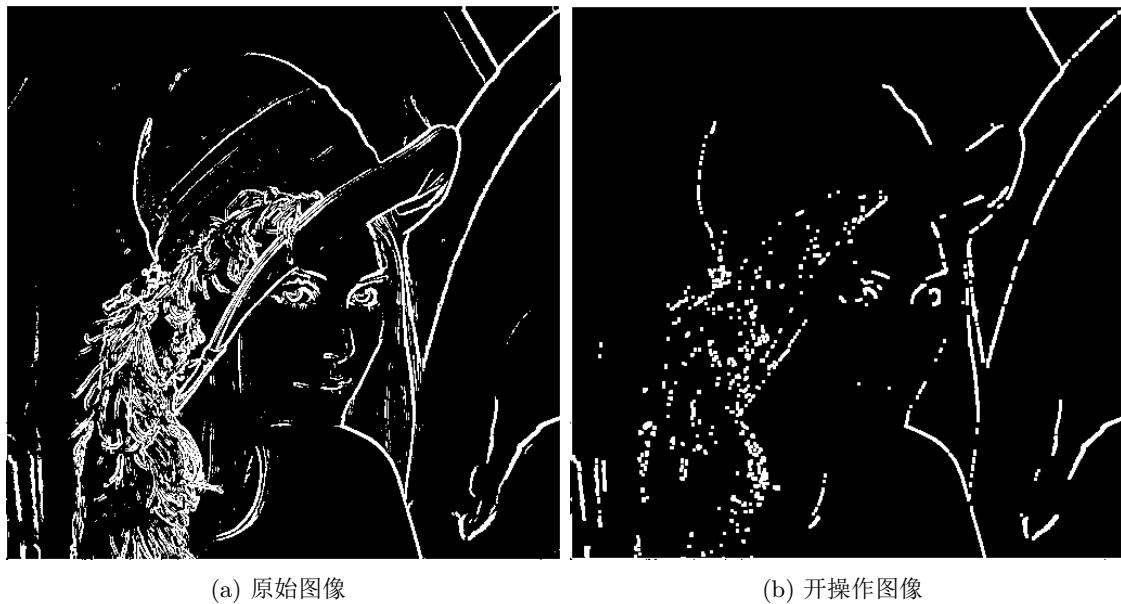


图 2.4: 形态学开操作效果



图 2.5: 形态学闭操作效果

第三章 OpenCV机器视觉库介绍

3.1 OpenCV简介

OpenCV是一个开源（开源定义）的计算机视觉库，项目主页为：（项目主页）。OpenCV采用C/C++语言编写，可以运行在Linux、Windows 和Mac等操作系统上。OpenCV 还提供了Python、Ruby、MATLAB以及其他语言的接口。

OpenCV的设计目标是执行速度尽量快，主要关注实时应用。它采用优化的C代码，能够充分利用多核处理器的优势。

OpenCV的一个目标是构建一个简单易用的计算机视觉框架，以帮助开发人员更加便捷的设计复杂的计算机视觉应用程序。OpdnCV中包含500多个函数，覆盖了计算机视觉的许多应用领域，例如：工厂产品检测、医学成像、信息安全、用户界面、摄像机标定、立体视觉和机器人等。因为计算机视觉和机器学习密切相关，所以OpenCV还提供了Machine Learning Library。该库侧重于统计方面的模式识别和聚类。ML除了用在计算机视觉相关人物中，也可以方便的应用于其他机器学习场合。

3.2 OpenCV发展历史

OpenCV诞生于Intel研究中心，其目的是为了促进CPU密集型应用。为了达到这个目的，Intel启动了多个项目，其中包括实时光线追踪和三维显示墙。一位在Intel工作的OpenCV 作者在访问一些大学的时候，发现许多顶尖大学的实验室（如MIT的多媒体实验室）拥有很好的内部使用的开发计算机视觉库——在学生间相互传播的代码，这会帮助一个新生从高的起点开始计算机视觉研究。

因此，OpenCV的目的是开发一个普遍可用的计算机视觉库。在Intel 的性能库团队的帮助下，OpenCV实现了一些核心的代码以及算法，并且发给Intel俄罗斯的库团队。这就是OpenCV的诞生之地：在与软件性能库团队的合作下，它开始于Intel的研究中心，并且在俄罗斯得到实现和优化。

3.3 OpenCV基本结构

OpenCV主要结构可分成五部分，其中四个模块如图（3.1）所示。OpenCV的CV模块包含基本的图像处理函数和高级的计算机视觉算法。MLL是机器学习库，包含一些基于统计的分类和聚类工具。HighGUI包含图像和视频的I/O函数。CXCore 包含OpenCV的一些基本数据结构和相关函数。

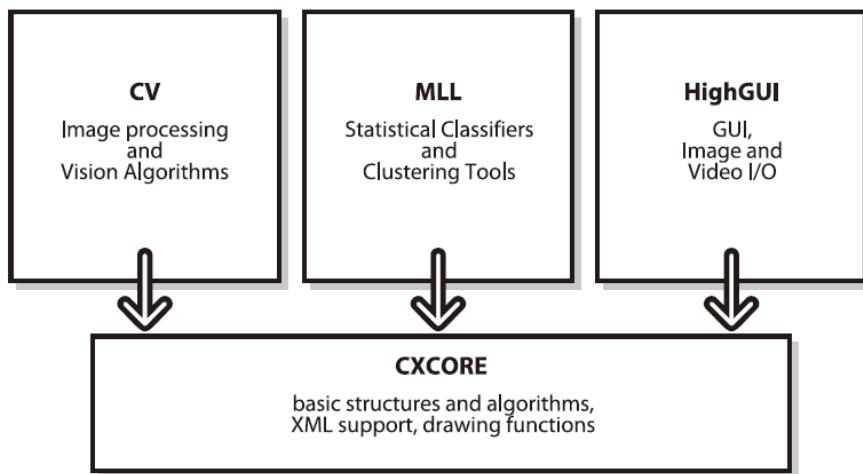


图 3.1: OpenCV基本结构

该图中没有包含CvAux模块。该模块中一般存放一些即将被淘汰的算法和函数（如基于嵌入式隐含马尔科夫模型的人脸识别算法），同时还有一些新出现的实验性的算法和函数（如背景和前景的分割）。

3.4 OpenCV数据结构

在OpenCV中，图像主要是以cv::Mat类的进行存储和处理的。

Opencv2.0版本发布后，其新的C++接口，cv::Mat代替了原来c风格的CvMat和IplImage.目前，2.0版本对c的接口也是支持的。

相对于c的接口，c++的cv::Mat统一了矩阵和图像这两个概念。事实上，矩阵和图像其实是一样的。由于cv::Mat是c++的类，所以也具备了相关的一些特征。例如，内存的释放。在C++中，一个对象超出其使用范围后，会自动调用析构函数进行销毁。而在c中，如果给CvMat类型的变量使用函数cvCreateImage 等函数分配了内存空

间，那么必须调用相应的函数进行释放，而不会自动销毁。如果没有相应的释放，则会造成内存泄漏。

3.5 OpenCV基本操作

3.5.1 矩阵操作

在使用c++接口前，先包含相应的OpenCV namespace。

在使用#include语句包含相应头文件后，使用下面语句即可包含相应的opencv命名空间：

```
using namespace cv;
```

如果没有这个语句，那么在这个命名空间的相关资源就需要带上cv前缀，如cv::Mat，表示的是使用命名空间cv中的Mat；而有了using namespace cv这个语句后，就可以直接写Mat。

这个新的类型Mat支持类似于matlab风格的矩阵代数运算，例如：

```
Mat A = Mat(3,4,CV_32FC1);
Mat B = Mat(4,3,CV_32FC1);
Mat C = 2*A*B;
```

那么，矩阵C是一个3*3矩阵，是矩阵A和矩阵B 做矩阵乘法后，乘上因子2的结果。这种方式比c接口好像要更直观一点。

Mat还有其他对矩阵操作的方法，例如：

```
Mat C = C.inv(); //Now C is its own inverse matrix
Mat D = A.t(); //D is the transposed matrix of A **
```

3.5.2 基本图像操作

3.5.2.1 图像读取

在OpenCV中，读取的图像都存放在cv::Mat类的一个对象中：

3.5.2.2 图像翻转

利用OpenCV中的cv::flip函数，调节相关参数，我们可以很方便的实现图像各方向翻转功能。

```
cv :: Mat result ;  
cv :: flip (image , result , 1);  
cv :: namedWindow(" Output Image " );  
cv :: imshow (" Output Image " , result );  
cv :: waitKey (0);  
cv :: imwrite (" output . bmp " , result );
```

第四章 Data Matrix快速定位算法研究

4.1 主要思路

快速定位算法设计的一个要点是：快。经过分析和实验对比，选择能够满足条件的最快速算法。在图像处理领域，一般来说，效果与速度难以兼顾，效果好的算法往往计算量比较大，耗时较长，快速算法在复杂环境下往往效果不好，对噪声等情况的抑制不够理想。在快速算法设计过程中，不断地进行对比测试就显得十分必要。

在算法设计中，我采取了“分而治之，逐个击破”的策略。首先设计一个大致的算法框架，然后对每一个步骤的算法进行优化。对于可能的方案，进行单元测试，优化参数。比如，在二值化过程中，可供选择的二值化算法有Otsu法，自适应阈值法等。其中Otsu法是比较普遍使用的，可以同时兼顾效果和速度。但是在实验过程中，我发现，Otsu法不能很好的处理光照分布不均匀，存在阴影的情况下。在这种情况下，选择自适应阈值能够取得比较好的效果。但是自适应阈值处理时间较慢，在普通情况下Otsu法已经可以满足下一步处理的需求。因此，我在设计中加入了自动选择算法的功能，如果发现Otsu法不能为下一步识别提供必要的信息，算法会重新调用自适应算法重新处理。这样同时兼顾了定位速度与定位效果。具体的粗定位实现效果会在下文中详加介绍。

在定位方法上，考虑到速度限制，我没有采用传统的Hough直线检测，而是直接对连通域进行矩形识别，通过加入阈值限制的方法来提取出可能存在二维码的ROI。通过实验，发现这种方法极大地提高了定位速度，并且能够很好的对多条码区域进行提取，具有很好的可拓展性。

整个算法的流程图如图（4.1）所示：

整个定位过程分成两部分：粗定位和精确定位。粗定位输入信息是原始图像，输出是所有可能包含二维码的矩形区域。精确定位接受粗定位过程的输出信息，输出是在原始图像上标记所有二维条码的精确边界。

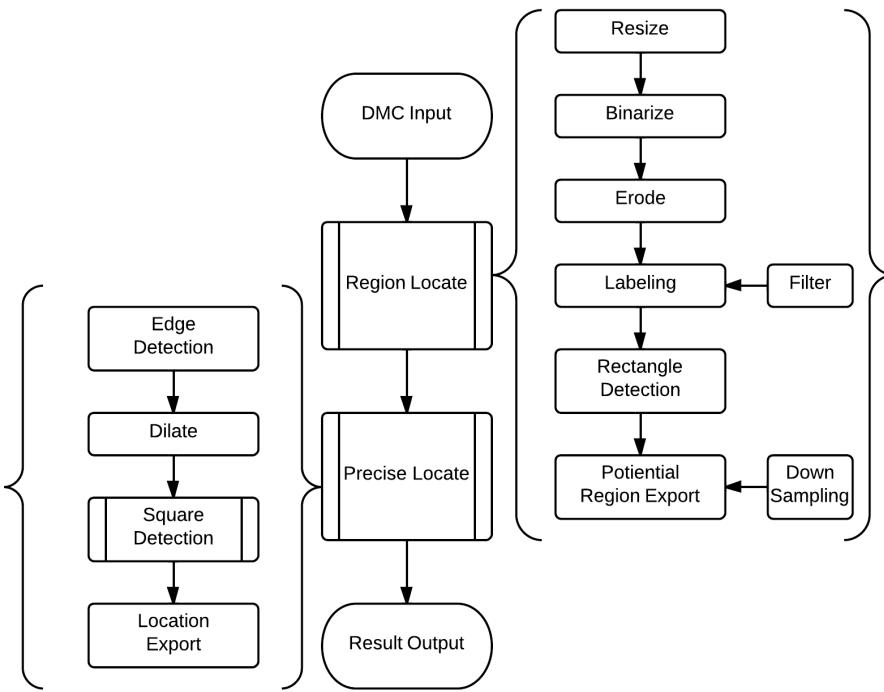


图 4.1: 快速定位算法流程图

4.2 粗定位算法

读取进图像后，首先对图像进行缩放处理。这样一方面可以减少数据量，提高处理速度，另一方面也使得程序可以处理多种尺寸的图像。同时，由于在图像尺寸压缩过程中带来的信息损失，这在一定程度上也起到了降噪的作用。接下来对图像进行二值化操作，默认采取的是Otsu法，适用于绝大多数的场景，如果碰到光照分布不均出现阴影的情况，会自动调用自适应阈值算法。接下来进行形态学上腐蚀处理，这一过程的目的在于将二维码区域填充起来。经过腐蚀的图像会进行连通域标记，将整个图像划分成包含不同形状的连通域。通过阈值对所有连通域进行过滤，可以很好的提取出二维条码所在连通域。

关于阈值的选择，在样本较少时，可以手动进行测试调整。对于以后大规模在工业上部署使用，可以选择合适的消费函数，进行计算机自动优化。实现方法如下：首先人工定位条码精确边界，作为训练数据。计算机自动选择不同阈值，测试成功定位的条目数。优化结束后，在识别过程中，阈值也会随着识别的准确率进行动态微调。

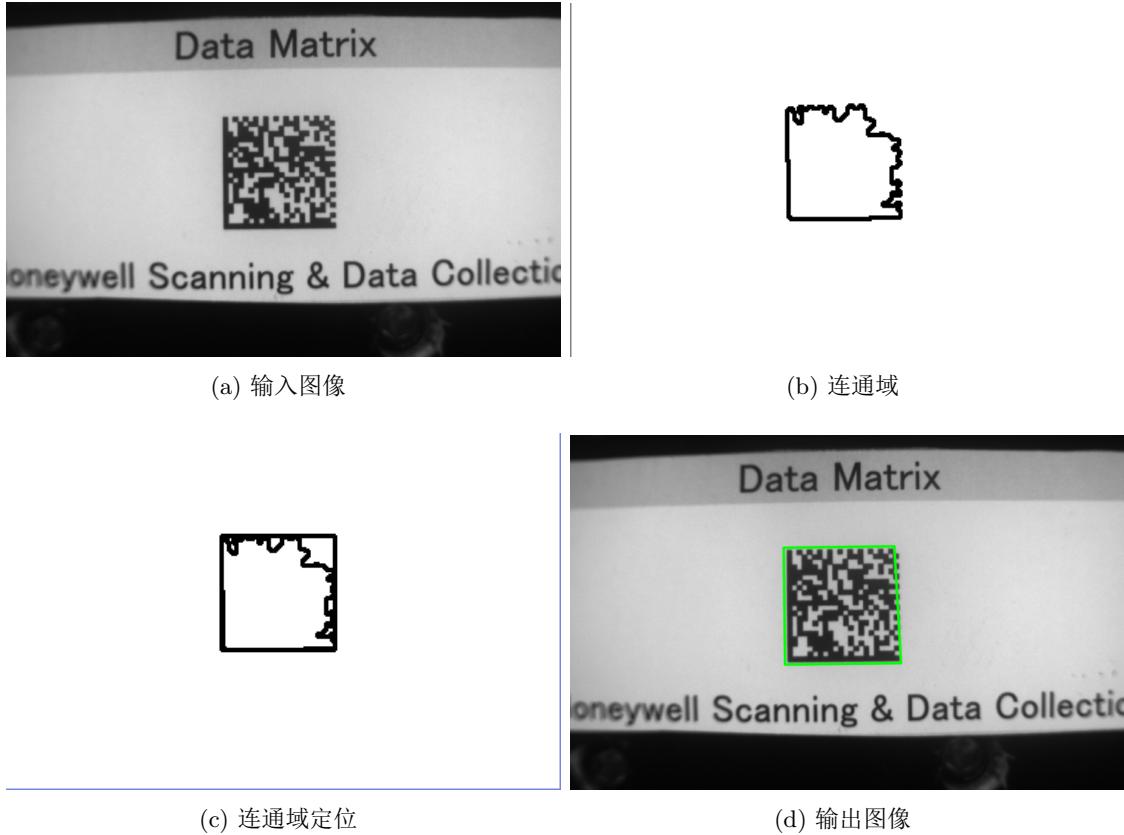


图 4.2: 粗定位效果

4.3 精确定位算法

粗定位过程中会生成包含二维条码的区域，精确定位会对这些区域进行更进一步的处理。由于粗定位过程中已经排除了大部分不含二维码的区域，因此在精确定位过程中需要处理的数据量已经小了很多。这进一步提升了定位速度。

在精确定位的第一步，首先会进行边缘检测。同二值化过程的处理方法一样，在边缘检测时同样整合了多个算法，可以进行自动选择。默认使用的是Roberts算子，这是最简单也是最快速的边缘检测算法。如果Roberts算法不能很好的定期条码边界，程序会自动调用Canny算子重新进行检测。这种多算法自动选择的方法，虽然在处理复杂环境时会多花一些时间，但是在绝大多数时间采用的都是非常迅速的算法，同时也兼顾了对环境的适应性。

在进行完边缘检测后，接下来会进行形态学膨胀过程。通过这一过程，二维条码内部绝大部分区域会被重新填充。接下来的过程

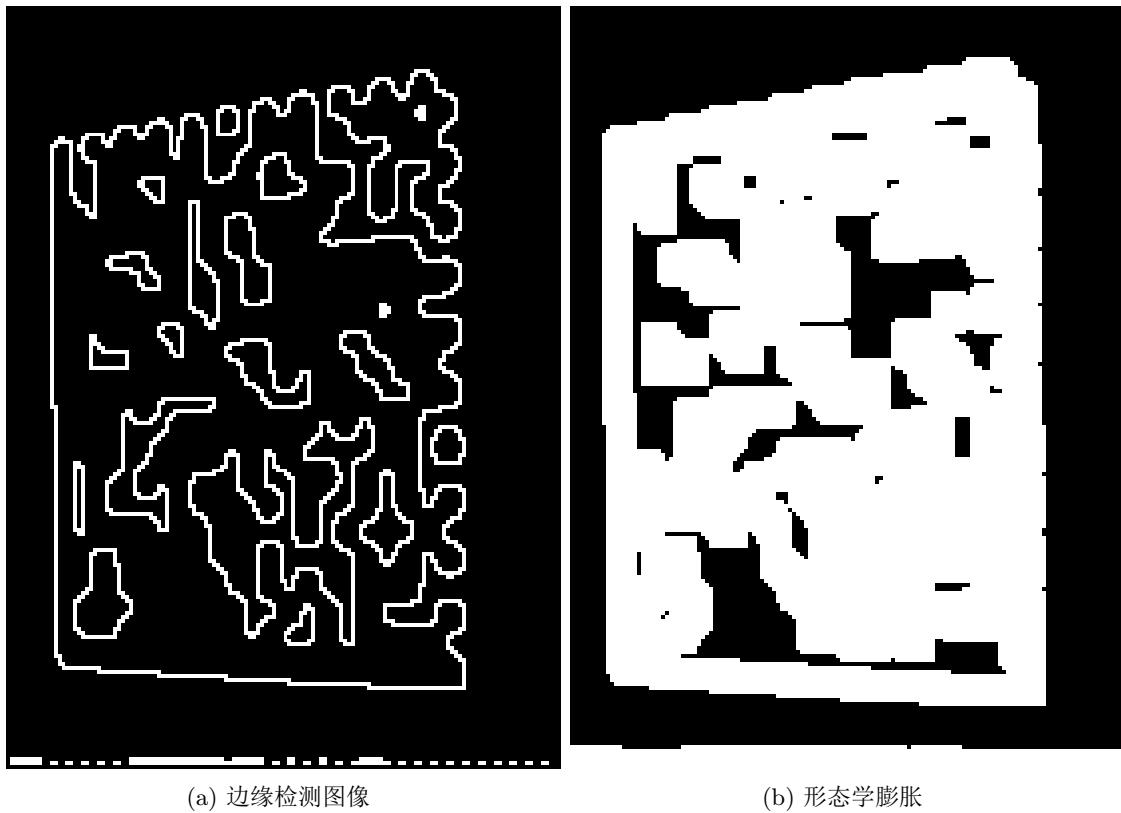


图 4.3: 形态学膨胀效果

会把该区域当做一个连通域进行处理，这可以防止定位时出现的错误。比起边缘检测算法等其他过程，形态学处理相对耗时较长，因此在这个过程中进行优化就十分必要。主要的步骤是选择合适的结构元素，在保证能够将条码内部填充的前提下，尽可能选择简单而面积小的结构元素对提升定位速度有很大的帮助。

4.4 实验结果

4.4.1 定位效果

经过实验，该快速定位算法能够很好的处理噪声、阴影、模糊、光照过强/过暗、旋转、倾斜等复杂环境。定位效果如图(4.4)和图(4.5)所示。

在复杂度非常高背景下，该算法也有着不错的表现，如图 (4.6) 所示。

最后，我测试了该算法在极为复杂的工业环境中的表现，如图

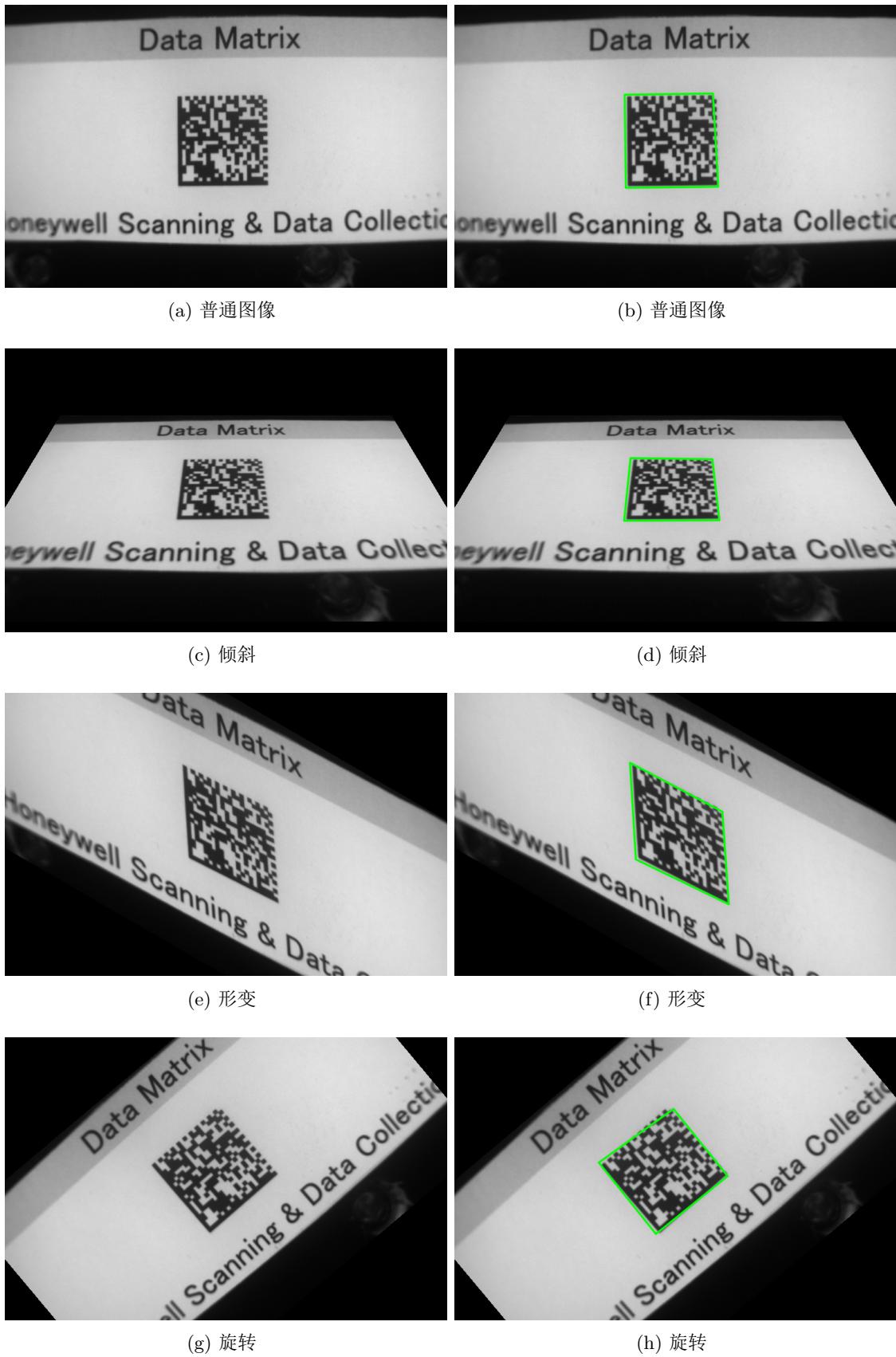


图 4.4: 快速算法定位效果(1/2)



图 4.5: 快速算法定位效果(2/2)

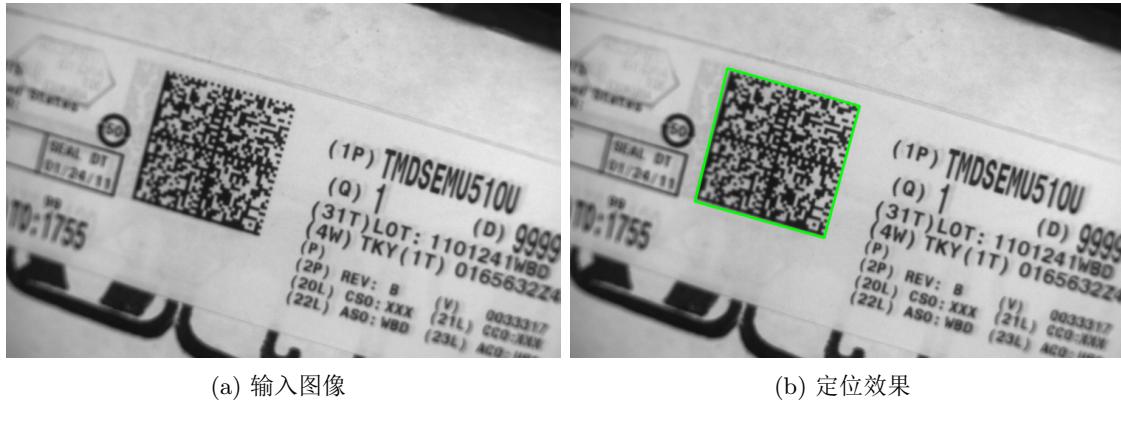


图 4.6: 复杂环境定位效果

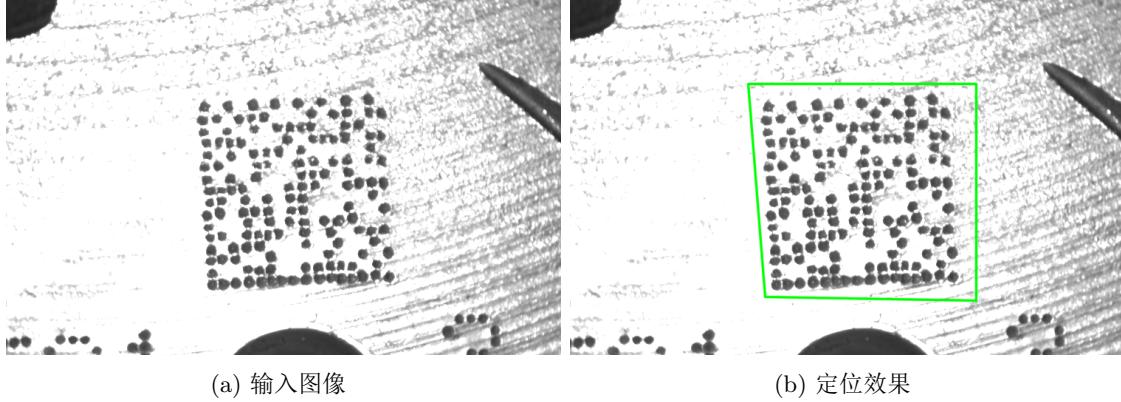


图 4.7: 工业环境定位效果

(4.7) 所示。

4.4.2 定位时间

对于定位时间的估计，我采取的方法是：在每种环境下，分别处理500张图像，计算其定位时间。

详细定位时间统计见表 (4.1)。

环境	粗定位 (ms)	精确定位(ms)	总时间(ms)
普通	0.5	0.5	1.0
倾斜	0.4	0.5	0.9
形变	0.5	1.0	1.5
旋转	0.6	0.7	1.3
高亮度	0.5	0.5	1.0
低亮度	0.5	0.5	1.0
模糊	0.5	0.5	1.0
噪声	0.5	0.5	1.0
复杂环境	1.5	0.5	2.0
工业环境	0.8	0.8	1.6

表 4.1: 不同环境平均定位时间

第五章 总结

5.1 工作总结

本文主要从图像处理的角度出发，研究了二维码识别相关领域技术，提出了具有高容错性的快速定位算法。本文主要解决了如下几个问题：

(1) 二维条码的粗定位

利用二维码本身是矩形的特点，在预处理之后采用连通域标记的方法提取出可能存在二维码的潜在区域。该技术使得二维码识别可以应用于各种复杂工业环境下，极大的扩大了二维码识别技术的应用范围。

(2) 二维条码的精确定位 本文没有采用传统的直线检测定位方法，而是直接运用四边形检测。这种创新性方法将定位速度成倍提高，是本算法成功的关键之处。同时，该方法同样适用于复杂环境，完全满足工业上使用条件。

(3) 对二维条码识别技术进行研究 二维码识别相关技术包括图像二值化、边缘检测、图像去噪、图像校正和形态学处理等多种图像处理方法。本文对相关领域已有的算法进行了比较和优化，构造出快速定位算法。

5.2 研究展望

二维条码具有许多优点：密度高、容量大、纠错能力强、可表示文字图像信息、可加密等。现在二维码已经广泛应用于现代自动化生产线、医疗行业、商品管理系统、超市物流管理等各个领域。随着物联网概念的推广和发展，将来二维码势必将在未来的商业和社会中扮演越来越重要的角色。

Data Matrix码由于其编码特点，可以在最小的空间内储存最多的信息。同时，比起其他二维码，它更适合在各种产品表面直接印刷。因此，Data Matrix码在工业流水线上有着十分重要的作用。但是，相比去QR码，Data Matrix码的识别更加困难，现在已有的识别算法中，识别Data Matrix码的速度远远慢于QR码。这已经成为

了限制生产效率的瓶颈之一。今后势必会有更多的有关快速定位算法的研究。

同时，由于工业上的复杂环境给二维码识别带来的挑战，如何在提升定位速度环境下同时兼容复杂环境识别能力也是对研究者提出的新挑战。广阔的前景与简单的挑战并存，这对研究二维码技术的广大科研人员既是挑战也是动力。

参考文献

- [1] 戴扬. 二维条形码编译码原理与应用研究. 南京航空航天大学, 5(26), 2004.
- [2] 周晓伟. 二维条码识别技术研究. PhD thesis, 上海: 上海交通大学, 2007.
- [3] 陈媛媛and 施鹏飞. 二维条形码的识别及应用[j]. 测控技术, 25(12):17–19, 2006.
- [4] 刘宁钟. 高维条码识别技术和编码理论的研究[D]. PhD thesis, 南京: 南京理工大学, 2003.
- [5] 胡晓崇and 何加铭. Data matrix 码识别技术研究. 杭州电子科技大学学报, 28(5):124–126, 2008.
- [6] 陈哲and 张永林. 基于二维条码技术的证件管理系统. 计算机应用与软件, 22(4):121–124, 2005.
- [7] 肖建芳and 林东岱. 二维条码和数字签名技术及其在毕业证书中的应用. 计算机工程与应用, 41(4):195–197, 2005.
- [8] I Dita, Marius Otesteanu, and Franz Quint. Data matrix code – a reliable optical identification of microelectronic components. In *Design and Technology in Electronic Packaging (SIITME), 2011 IEEE 17th International Symposium for*, pages 39–44. IEEE, 2011.
- [9] Qiang Huang, Wen-Sheng Chen, Xiao-Yan Huang, and Ying-Ying Zhu. Data matrix code location based on finder pattern detection and bar code border fitting. *Mathematical Problems in Engineering*, 2012, 2012.
- [10] Tapani Sarjakoski, Janne Kovanen, M Ro?nneberg, J Ka?hko?nen, and L Tiina Sarjakoski. Data matrix technology for linking mobile maps in a web-based multi-channel

- service. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2010, pages 1–5. IEEE, 2010.
- [11] 邹沿新and 杨高波. Data matrix 二维条形码解码器图像预处理研究. *计算机工程与应用*, 45(034):183–185, 2009.
- [12] 姚林昌, 白瑞林, 钱勇, and 徐义钊. 一种data matrix 条码的快速识别方法倡. *计算机应用研究*, 28(11), 2011.
- [13] Feng Liu, Anan Liu, Meng Wang, and Zhaoxuan Yang. Robust and fast localization algorithm for data matrix barcode. In *Optoelectronics and Image Processing (ICOIP)*, 2010 International Conference on, volume 2, pages 356–359. IEEE, 2010.
- [14] Yuan-Fei Cheng. A scanning method for dotted data matrix. In *Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on*, volume 2, pages 179–183. IEEE, 2008.
- [15] AI Dita, BM Otesteanu, and CF Quint. Scanning industrial data matrix codes. In *Telecommunications Forum (TELFOR)*, 2011 19th, pages 1211–1214. IEEE, 2011.
- [16] Ion-Cosmin Dita, Marius Otesteanu, Franz Quint, and Nagaraju Annamdevula. A scanning method for industrial data matrix codes marked on spherical surfaces. In *publishing at 11th WSEAS International Conference on TELECOMMUNICATIONS and INFORMATICS (TELE-INFO 2012)*, 2012.
- [17] Ion-Cosmin Dita, Vasile Gui, Marius Otesteanu, and Franz Quint. Using mean shift algorithm in the recognition of industrial data matrix codes. In *publishing at 11th WSEAS International Conference on SIGNAL PROCESSING (SIP' 12)*, 2012.
- [18] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

- [19] 季虎, 孙即祥, 邵晓芳, and 毛玲. 图像边缘提取方法及展望. 计算机工程与应用, 14(1):70–73, 2004.
- [20] 董梁and 王忠民. 图像边缘检测算法研究. 现代电子技术, 30(15):179–180, 2007.
- [21] 阮秋琦. 数字图像处理学. 电子工业出版社, 2007.

致 谢

值此论文完成之际，谨在此向大学四年来自予我关心和帮助的老师、同学表示衷心的感谢！

.....

谨把本文献给我的父母！