

分类号：TP391.41

密 级：无

单位代码：10028

学 号：2130502033

首都师范大学硕士学位论文

一种彩色 QR 码的编解码设计与 防伪运用

研 究 生：袁沅祥

指导教师：李宏伟

学科专业：应用统计

学科方向：图像处理

二〇一六年五月七日

首都师范大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

导师签名：

日期： 年 月 日

首都师范大学学位论文授权使用声明

本人完全了解首都师范大学有关保留、使用学位论文的规定，学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版。有权将学位论文进行少量复制并允许论文进入学校图书馆被查阅。有权自行或许可他人将学位论文的内容编入有关数据库进行检索。有权将学位论文的标题和摘要汇编出版。保密的学位论文在解密后适用本规定。

学位论文作者签名：

导师签名：

日期： 年 月 日

摘 要

条码技术是一项新颖而充满生机活力的技术，二维码技术是条码技术新的发展方向与研究热点，是一项集信息编码、信息传递、图像处理、数据加密等技术于一体的综合性技术。

QR 码是一种矩阵形式二维条码，它由日本电装株式会社(DENSO CORPORATION)在 1994 年开发，并于 2000 年成为 ISO 国际标准。2001 年，QR 码成为我国国家标准。随着智能手机的普及，QR 码日益被大众所认知、接受，已广泛见于日常生活，而且作为一项新的防伪措施，也已被广泛用在电子票务、物流溯源等方面。但 QR 码容易被复制，防伪易被攻破。

本文研究了一种基于 QR 码的彩色二维码编码译码设计及实现，并与上转换发光材料结合起来，建立起 QR 码的防伪双屏障。一方面是软件屏障，不仅具有普通 QR 码的全部性能，还将色彩进行编码，使二维码携带更多的信息；另一方面是硬件屏障，化学材料不可复制。两道屏障使彩色 QR 码防伪性能得到提高，兼具时尚性与技术性。彩色 QR 码可以部分甚至完全隐藏，在人眼观看只是普通的 QR 码，用智能手机就能解析，得到基础信息。但用激光照射它会发出彩色光，经专用设备解码能获得双层信息，一是普通二维码信息，比如商品的生产信息，二是特殊信息，比如鉴别商品真假的信息。本文利用到的主要技术是图像处理算法和 RS 纠错编码理论，创新点是首次设计彩色 QR 码，并将化学材料与二维码结合起来用于防伪，这尚属首次。

本文前两章是知识储备。第一章介绍了条码技术和课题研究背景，第二章介绍了矩阵式 QR 码，并对 QR 码的编码解码过程进行了分析与研究。第三至四章是对彩色二维码的设计与实现。第三章提出了一种简单可行的彩色 QR 码的编码解码设计方案。第四章介绍了在 PC 端和安卓平台实现的彩色 QR 码编码解码程序。第五章介绍了作者基于 QR 码设计的隐形二维码与融合二维码，并对本文下了一个结论。

QR 码虽然得到了广泛的运用，但因为是日本的专利技术，因此可能存在安全隐患。本文在后期仿照 QR 码的设计思路，自行设计了一套二维码编码译码方法，并拟在这套技术之上研发彩色二维码。

关键词：QR 码、二维码防伪、图像处理、编码解码、纠错编码

Abstract

The bar code technology is a novel technology full of vitality, of which the two-dimensional barcode technology is the recent development direction and the research hot spot. It is a comprehensive technology which combines information coding, information transmission, image processing, data encryption and so on.

The QR Code is one kind of matrix form 2D barcode which researched and developed in 1994 by Japanese Denso Corporation, and became the ISO international standards in 2000. In 2001, the QR Code became China's national standards. With the popularity of smart phones, QR Code is increasingly being recognized and accepted by the public, and has been widely seen in daily life. As a new anti-counterfeiting measure, QR Code has been widely used in electronic ticketing, logistics, traceability, etc. However, QR Code is easy to be copied, and its anti-counterfeiting is easy to be breached.

In this paper, we study the encoding and decoding method of a kind of colored 2D barcode which is based on QR Code, and combining with transformation on the luminescence materials, we establish a QR Code anti-counterfeiting double barrier. On the one hand is the software barrier, it not only has the ordinary QR Code complete performance, but also it encodes the colors, thus causes 2D barcode carry more informations; On the other hand is the hardware barrier, chemical material cannot be duplicated. Two barriers enable the colored QR Code forgery-proof performance to obtain the enhancement, concurrently fashion and technical nature. The colored QR Code may be partly even completely hidden, in the human eye onlooking is only the ordinary QR Code. Using mobile phones to decode, we get basic information. But with laser it will emit light color, the proprietary equipment is able to win a double information by decoding it, one is ordinary QR Code information, such as the production of commodity information, the second is special information, such as identify true or false information. The main technologies used by this article are the algorithm of image processing and the RS error correction coding theory, the innovation spot is that we design the colored QR Code for the first time, and unifies chemistry material and the 2D barcode uses in the forgery-proof, this still was for the first time.

The first two chapters of this article are the knowledge reserves. First chapter introduced the bar code technology and the topic research background. Second chapter introduced the matrix form QR Code, and the encoding and decoding process of QR Code is analyzed and studied. Third to fourth chapters showed the colored QR Code design and the realization. Third chapter proposed one kind of simple feasible colored QR Code encoding and decoding design proposal. Fourth chapter introduced the colored QR Code encoding and decoding procedures on the PC and the android platform. In the fifth chapter, we introduced the stealth 2D code and the fusion 2D code based on the design of QR Code, and finally we drawn a conclusion to this article.

QR Code has been widely used, but because it is Japan's patented technology, there may be a security risk. Late in imitates the QR Code design, this paper designed a set of 2D barcode encoding and decoding method, and on this technology research we will develop a new colored 2D barcode.

Key words: QR Code, 2D barcode anti-counterfeiting, image processing, encoding and decoding, error correction of coding

目 录

摘 要.....	I
Abstract.....	II
目 录.....	III
图表目录.....	IV
第一章 前 言.....	1
1.1 条码技术简介.....	1
1.1.1 一维条码.....	1
1.1.2 二维条码.....	2
1.2 国内外研究现状综述.....	4
1.3 课题研究背景.....	5
1.4 选题的意义和价值.....	5
第二章 QR 码编码解码技术研究.....	7
2.1 编码过程中的主要技术.....	7
2.1.1 纠错编码技术.....	7
2.1.1.1 BCH 码.....	8
2.1.1.2 Reed-Solomon 码.....	8
2.1.2 掩码技术.....	10
2.2 解码过程中的主要技术.....	11
2.2.1 二值化.....	12
2.2.2 K-MEANS 聚类.....	14
2.2.3 图像滤波.....	15
2.2.4 寻象图形的定位.....	17
2.2.5 透视校正.....	20
2.3 QR 码编码解码库介绍.....	22
2.3.1 ZXing.....	22
2.3.2 ZBar.....	22
第三章 彩色 QR 码的编码解码设计.....	23
3.1 彩色编码.....	23
3.1.1 信息格式.....	23
3.1.2 密文编码.....	24
3.2 彩色解码.....	26
第四章 彩色 QR 码编码解码程序.....	28
4.1 程序概要设计.....	28
4.2 程序详细设计.....	29
4.3 安卓 APP.....	32
4.4 编码解码实例.....	32
第五章 结论与展望.....	38
5.1 作者的其他研究.....	38
5.1.1 隐形二维码.....	38
5.1.2 融合二维码.....	38
5.2 结论与展望.....	39
致 谢.....	40
参考文献.....	41

图表目录

图 1-1	常见的一维条码.....	1
图 1-2	国外二维码.....	2
图 1-3	国产二维码.....	2
图 1-4	彩色二维码示例.....	4
图 2-1	QR 码编码流程.....	7
图 2-2	QR 码掩码模式.....	11
图 2-3	QR 码译码流程.....	11
图 2-4	译码详细流程.....	12
图 2-5	一个图像的二值化.....	14
图 2-6	均值滤波图示.....	15
图 2-7	中值滤波图示.....	16
图 2-8	位置探测图形的结构 ^[38]	17
图 2-9	寻象第一种错误.....	18
图 2-10	寻象第二种错误.....	18
图 2-11	解决寻象第一类错误.....	19
图 2-12	解决寻象第二类错误.....	19
图 2-13	寻象函数性能图示.....	20
图 2-14	透视投影图解 ^[39]	20
图 2-15	透视畸变的 QR 码.....	21
图 2-16	透视校正的 QR 码.....	22
图 3-1	信息格式编码区(红色与蓝色).....	23
图 3-2	密文编码流程.....	24
图 3-3	密文解码流程.....	26
图 3-4	彩色解码流程.....	27
图 4-1	编码系统简要流程图.....	28
图 4-2	彩色 QR 码的控制信息.....	28
图 4-3	文档/视图框架.....	29
图 4-4	编码与解码模块.....	30
图 4-5	编码与解码测试.....	31
图 4-6	PC 端解码实例.....	31
图 4-7	PC 端解码实例(带 LOGO).....	32
图 4-8	APP 编码解码实例.....	37
图 4-9	APP 解码实例(带 LOGO).....	37
图 5-1	隐形二维码.....	38
图 5-2	QR-汉信融合码.....	39
表 2-1	Reed-Solomon 纠错编码参数选取.....	9
表 2-2	QR 码掩码模式.....	10
表 3-1	“数据头”与 LOGO 信息编码.....	24
表 3-2	彩色 QR 码 1.....	26
表 3-3	彩色 QR 码 2.....	26

第一章 前言

1.1 条码技术简介

1.1.1 一维条码

条形码简称条码，来源于其英文名字“Barcode”。人们现在口中说的条形码指的就是一维条码。一维条码从直观上看是一系列平行的黑色竖条，这些线条具有相同的高度，线条之间以空白分割。它们按照既定编码规则排列，组成表达信息的符号。这个符号之所以能被专门的工具识读，是由于白条和黑条反射率相差很大，用光源照射时接收到的反射光的强弱就被解释为“1”或“0”，即一个比特，形成比特流再经计算机破译即得到其蕴含的信息。

条码技术并不是一项特别古老的技术，而是一项比较年轻、众人皆知的技术，并处于高速发展的时期。条码技术最早产生在大约一百年前，最先出现在 Westinghouse 的实验室里^[1]。Kermode 发明了最早的条码标识，即一个“条”表示数字“1”，两个“条”表示数字“2”，三个“条”表示数字“3”，依次类推^[2]。在生产实践中，经常使用的一维码制包括 EAN-13 码，Code 39，ISBN 码，Codabar(库德巴码)，如图 1-1 所示，除此之外还有交叉 25 码、UPC 码、128 码、93 码等。



图 1-1 常见的一维条码

条码技术是建立在多个新兴技术基础上的一项信息自动识别技术，它综合了计算机技术、光电传感技术和通信技术。计算机技术体现在编码和解码，光电传感技术体现在条码获取，通信技术体现在数据后台处理。在众多自动识别技术中，条形码是最经济又实用的技术，具有成本低、输入快、可靠性高、使用灵活等优点。条码现在已经成为信息数据自动输入、识别的重要方法和手段，是物流管理现代化的重要技术手段。早期发明的一维条形码仅沿着黑白相互间隔的方向(横向)携带信息，编码量非常受限。如今，一维条形码已经广泛见于各类商品包装袋上。

1.1.2 二维条码

二维条形码是对一维条形码的发展，在英语中被称为“2-dimensional barcode”，用“2-dimensional”作修饰。二维码是用某些类型的几何结构(通常为正方形)，遵循一定的分配规则排列到平面上形成图像，来表达信息。二维码也利用了组成计算机内部逻辑基础二进制比特流的概念，在某些几何图形与二进制(0 或 1)之间建立对应关系，使用这些图形来表示文字数值信息，通过图像的读取输入装置或光电扫描设备自动识别，来实现信息的自动处理^[3]。

(一) 发展历程

国外学者对二维码技术的研究始于 20 世纪 80 年代末^[4]。二维码技术现在已经是一项非常成熟的技术^[5]。全球已经研制出来的一、二维码多达 250 种以上，其中常见的二维码有 QR Code、Data Matrix、PDF 417、Aztec 等，如图 1-2 所示。



图 1-2 国外二维码

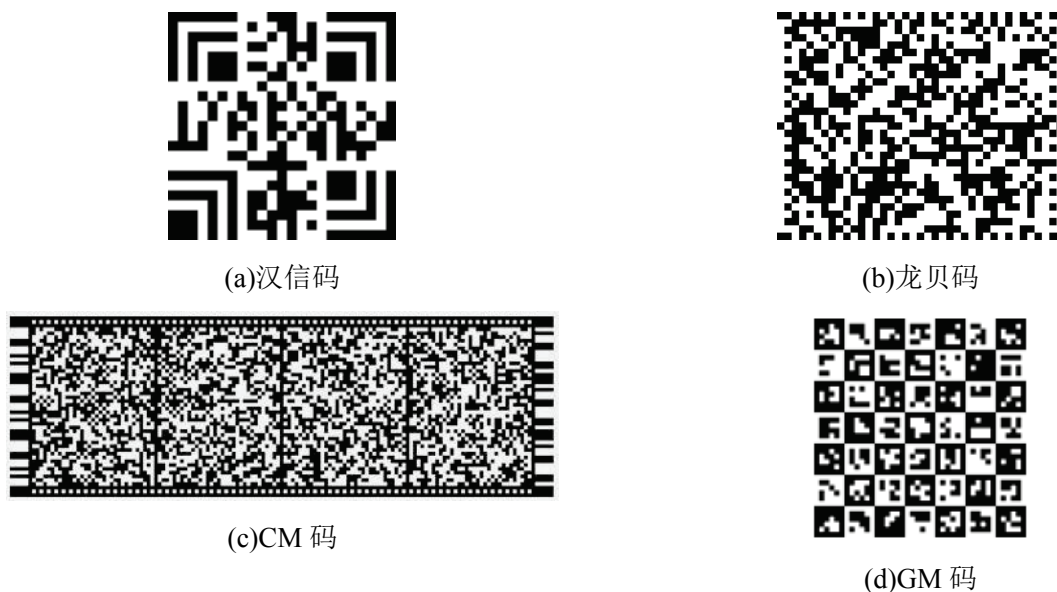


图 1-3 国产二维码

日本 Denso 公司^[6]在 1994 年 9 月成功研制出矩阵形式的 QR 码, 该符号的性能比一维条码更优, 不仅具备其他二维码的信息容量大、可靠性高、保密防伪性强等优点, 而且能高效地表示汉字。编码相同内容的 QR 码尺寸小于相同密度的堆叠式 PDF 417 条码。QR 码已经在我国得到了广泛的使用, 特别是近几年悄然火了起来, 几乎随处可见。目前, 市场上的大部分条码打印机与扫码器都支持 QR 码, 比如福建新大陆集团生产的扫描仪。

我国对二维码技术的研究开始于 1993 年, 首先采用了把国外技术先引进来的策略。隶属于国家质量监督检验检疫总局的中国物品编码中心, 对上述几种常用的国外二维码的技术规范进行了翻译和跟踪研究, 并借鉴 QR 码, 推出国产汉信码, 如图 1-3 (a)所示。我国的几项二维码国家标准是从美国的 PDF 417 码和日本的 QR 码翻译过来的。随着持续的人力物力投入和研究的推进, 国内取得了突破性的技术和创新二维码技术, 拥有自主知识产权的国产码制开始出现, 像龙贝码(图 1-3 (b))、CM 码(图 1-3 (c))、GM(图 1-3 (d))码都得到了国家的认同和推荐。

(二) QR 码

之所以将 QR 码取名为“Quick Response”, 是因为发明者原昌宏的初衷就是希望 QR 码能被快速解码。目前, QR 码已经在许多国家标准和国际中实现标准化, 任何人都可以随意查看该标准, DENSO INCORPORATED 已宣布, 不行使其就标准 QR 码拥有的专利权(专利第 2938338 号)^[6]。

通常的 QR 码呈正方形, 只有黑白两种颜色。在 4 个角落的其中 3 个(*Windows* 坐标系 *XOY*: 右上角、左上角、左下角), 印有较小、像“回”字形状的正方形图案。这 3 个“回”字图案用来帮助解码软件定位, 被称为“位置探测图形”。借助于它, 使用者不需要特别地对准, 无论以任何角度扫描, 资料仍可正确被读取^[7]。包括 QR 码、汉信码、DM 码等, 许多矩阵式二维码都有位置探测图形。

QR 码支持多种字符集编码, 包括中文, 这个特性使它优于一些不可编码中文的二维码, 比如 DM 码。QR 码具有信息容量大、可自定义纠错等级、识别速度快等优势, 已经成为最广泛使用的一种二维码, 以至于人们认为二维码就是 QR 码。

(三) 二维码防伪

随着二维码的普及, 特别是腾讯公司在其社交软件——微信, 在其中置入了扫码功能之后, 普通人对二维码也有了了解, 并尝到了扫码带来的便利。二维码的普及带动了一批新技术的发展, 二维码防伪作为一种新的防伪方法日渐兴起。相比其他防伪措施, 这种方法使用起来非常方便, 用户只需要扫描二维码标签就能验证真假, 而且获得了关于商品的详尽信息^[8]。正是因为二维码能够编码大量的信息, 所以能生成大批量且不重复的随机数码。这些随机数码能作为产品的唯一标识, 用于验证产品的真伪, 一个常见的使用场景就是用二维码存放产品的“序列号”。由于二维码技术是一项交叉技术, 入门门槛较高, 试图破解产品的编码方案是具备一定难度的。

二维码具有加密及防伪功能,可用于辨别商品的真假。消费者通过手机扫码辨识商品真假,并获得维权的证据,在某种程度上起到了遏制假冒伪劣产品的作用。但是,二维码防伪也面临新的造假技术的挑战。制假者一旦掌握了编码机制与一些真码的内容,就会大批量地进行复制,并将“真”的二维码贴到假冒的产品上。即使厂家提供联网验证的服务,其监控到假货的时间也是滞后的,不能完全避免损失。

1.2 国内外研究现状综述

目前国内普遍使用的黑白二维码技术因生成简单而被广泛运用,但在安全方面存在较大问题^[9]。文[10]提到,日本已将彩色二维码相关技术申请专利,期望与中方以技术入股、以合作生产的方式进行合作。彩色二维码实际上是一种三维码,是在二维码 X 轴- Y 轴坐标模型(二维平面)的基础上,增加了色彩维度(即 Z 轴)来表示信息,从而扩展到了三维空间。三维码所包含的信息量大幅增加,并具有更高的保密性和防伪能力。在欧美等发达市场,三维码技术已被广泛应用于印刷、户外屏幕乃以及互动电视等商业领域,我国也拥有完全自主知识产权的三维码技术,但尚未形成大规模应用^[11]。“视觉码”是一家以色列的创业公司,2015年1月20日获得了阿里巴巴投资^[12]。

国内文献研究的重点是如何提高解码效率,较少对彩色二维码进行深入研究。国内现在虽然产生了一些彩色二维码(如图1-4(c)视觉码),但都是根据QR码能纠正错误这一性质,并利用图像处理技术,在外形外观、色彩搭配等方面使得QR码更好看,而不像通常的黑白二维码那么单调,但这种彩色是不编码信息的。如图1-4(a)是一种微信二维码,图1-4(b)是一种创意二维码,这两个二维码都是彩色的,但彩色都是不编码信息的。

文[13]从增加编码信息量方面考虑,利用RGB色彩差异,在DM码的基础上设计了一种彩色二维码。文[14]仿照QR码的思路设计了一种彩色二维码,同样是利用了色彩差异,选用了五种颜色。



图 1-4 彩色二维码示例

国内彩色 QR 码的应用场景主要为广告创意,对产品美学要求高。我们设计的彩色 QR 码能部分甚至完全隐藏,可以用喷码方式打印到产品上而不破坏其美观,也可以制作成二维码标签,贴在产品上。一方面,我们的彩色 QR 码可以用彩色打印的方式,完整地打印到产品或以标签形式贴到产品;另一方面,我们的彩色 QR 码若与化学发光材料结合使用,彩色就不是颜料的色彩,而是用激光照射化学材料时发出的色彩。

1.3 课题研究背景

近年,我们常常遇到以“扫码送礼品,扫码享优惠”为主题的活动,推销员散发传单,传单上印着一个醒目的二维码;扫描超市农场品上面的二维码,可以得到其产原地、时间、物流等信息;2009 年 12 月起售的新版火车票上面也印着二维码,站务人员以此验证车票真伪;公交站台的许多广告都配有二维码……这些迹象表明,二维码已经深入人们的生活,几乎无处不在。

腾讯科技曾在 2012 年 11 月报道^[15],二维码市场规模在 2015 年将超 1 千亿。文中指出,伴随一些互联网巨头先后加入二维码行业,二维码行业发展形势如日中天,将走向全民普及时期。与此同时,二维码防伪技术在物联网时代随处可见^[16]。因为二维码可以编码大量的信息,所以能实现“一物一码”,用唯一的 ID 对物品进行标记。不同于一维条码需要联网及数据库支持才能获得数据表达的商品信息,二维码可以直接储存丰富的产品信息。将二维码印刷或者贴在产品包装上,用户只需要通过简便的扫码操作就能验证产品真伪,获得比较详细的产品信息,这使用户体验也较好。

二维码已经普遍存在于日常生活中,以日本的 QR 码为主。这些二维码往往只有黑白两种色彩,并不具备其他色彩。正所谓“道高一尺,魔高一丈”,二维码用于防伪遭遇到了一些新的问题,原本看起来靠谱的二维码防伪变得越来越不靠谱了。导致这种结果的一个重要原因就是二维码很容易被复制。易复制致使二维码防伪性能降低。一些特殊行业对二维码防伪提出了更高的要求。

本文基于 QR 码设计了一种彩色二维码,使得在不失去 QR 码原有性质的基础上加入了被编码的色彩信息,且可以与化学发光材料结合起来运用到防伪领域。化学材料不能被复制,较大地提高了二维码的安全性能。

1.4 选题的意义和价值

(一) 市场痛点

随着互联网的普及,网络购物成为许多人的首选。人们对于关乎人身安全和价值较高的产品特别关注,消费者都想买到真品,但假货横行。2015 年各大电商发起的“双十一”购物狂欢节交易额超 912 亿元,再创历史新高。有关报告显示^[17],2014 年质检总局开展 5 类 14 种电子商务产品质量的国家监督抽查,合格率为 73.9%,网购正品率仅为 58.7%。

(二) 国家战略

目前我国的二维码一共有五项标准,即 PDF 417 码、QR 码、GM 码、CM 码、龙贝码。早在二十一世纪初,我国就先后将美国的 PDF 417 码(由美籍华侨研制)及日本的 QR 码(由日本人研制)颁布为二维码国家标准。CM 码和 GM 码是武汉矽感公司分别针对 PDF 417 码和 QR 码设计的。国产龙贝码已经取得了多项技术突破,但多用于军工领域。大家在日常生活中见到的二维码码制九成以上是进口码制,尤其 QR 码最为常见。日常二维码中,既有 DM 码,也有 PDF 417 码,唯独见不到国产二维码。

二维码技术是物联网时代的一个核心技术。二维码对于物品,就好比身份证对于个人。如果物品的标识使用国外技术,关键技术受制于人,不仅打击民族自信,更对于国家会有许多潜在的安全隐患。

(三) 本文出发点

五个二维码国家标准中,除了从国外引进的 PDF 417 码和 QR 码,另外三种二维码都不是开放源码的,而这使得国产二维码更加被边缘化,几乎到了无人问津的地步。

TAG 是一种由互联网巨头微软公司自行研发的二维存储信息的技术,已经在 2015 年全面终止,其宣布的原因很简单——市场影响力根本无法和随处可见的二维码相提并论。由此,国产二维码当前跟 QR 码作战,基本死路一条,胜算非常渺小。

本文设计了一种与 QR 码“兼容”的二维码。我们主要的出发点是防伪,另一个初衷或者出发点是希望从敌人内部突破,逐步取得二维码战争的胜利。不需要花费额外的代价,还是使用原有的设备,就能对二维码进行解码,得到与 QR 码无异的信息,但若想要获取更多的信息,就需购买专用设备,或者在原有设备嵌入解码代码,或者安装专用软件。我们希望于无形中影响或更改消费者习惯,改变他们观念。等时机成熟时,再助力推广国产二维码。

(四) 特别说明

本篇论文是首都师范大学研究生创新创业实践基地第三期立项项目“善水防伪服务”的研究内容,项目得到了首都师范大学研工部的支持。文中彩色 QR 码的设计及应用程序的编码均由本人独立完成,化学发光材料由首都师范大学化学系光功能材料与器件实验室提供,激光装置由化学系刘东东同学联系厂商生产。彩色 QR 码图像打印及识别的相关实验由刘东东同学与本人合作完成。

此创业项目得到了北京水木九天科技有限公司董事长王晓庆老师的悉心指导。QR 码开发包来自谷歌公司的开源项目 ZXing。汉信码 SDK 得到了中国物品编码中心的授权。部分代码取自互联网,文章均指明了具体来源。

本文首先谨向创业团队以及上述单位或个人表示诚挚的感谢!

第二章 QR 码编解码技术研究

2.1 编码过程中的主要技术

QR 码的编码流程如图 2-1 所示。开始编码时，首先对用户输入的数据进行分析，选择适当的数据模式(比如数字模式或字节模式)，对数据进行编码，并计算纠错码字，一起构造出最终的信息。然后在矩阵中布置模块，对矩阵进行掩模，布置格式和版本信息，最终生成图像。该过程中比较重要的技术是纠错编码技术与掩码技术。

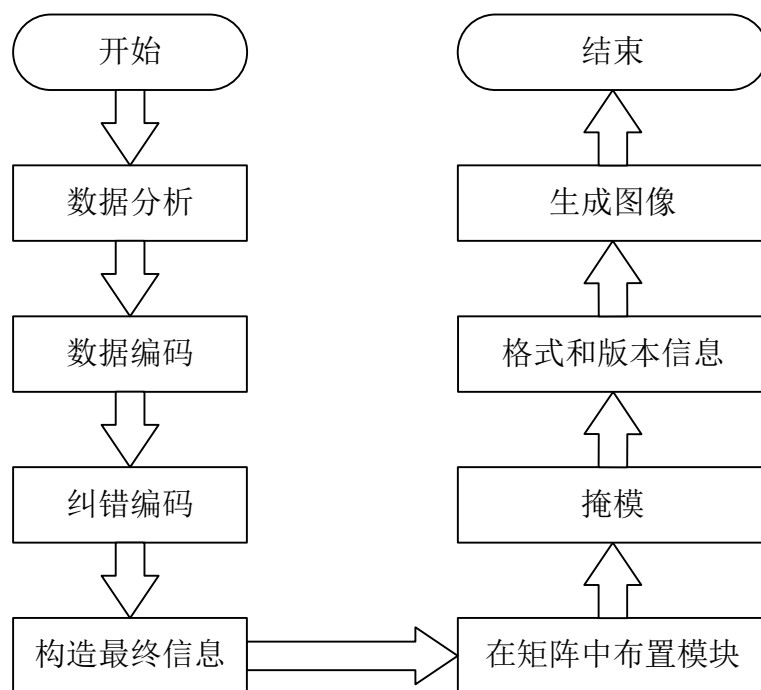


图 2-1 QR 码编码流程

2.1.1 纠错编码技术

纠错编码在英语中是“Error Correcting Coding”，简称 ECC。纠错编码技术的重要贡献是增强了数字通信的可靠性。纠错编码技术的理论极强，其原理涉及到深层的数学基础，比如信息理论、线性代数、抽象代数，这些理论是复杂的。鉴于纠错编码技术如此复杂，本文仅对其作简单介绍。

二维码在使用过程中难免会遭受污渍、撕损、变形等种种损坏，特别是当二维码用在物流溯源领域的时候。虽然图像预处理能有效地消除某些污染，并校正畸变，但对于缺失、覆盖等污染就无能为力。为了保证较高的识读准确率，我们需要在数据码字之后加入所谓的纠错码字用来恢复原始数据，添加的冗余信息占比越高，获得的纠错能力越强^[18]。QR 码拥有可供选择的四个级别的纠错能力，最大能纠正 30%的错误(H 级别)。

QR 码的格式信息采用了 $BCH(15,5)$ 码, 版本信息采用了 $BCH(18,6)$ 码, 数据信息采用了 RS 码。下面对 BCH 及 RS 两种纠错编码作简单的介绍。

2.1.1.1 BCH 码

BCH(Bose-Chaudhuri-Hocquenghem)码是 1960 年由博斯(Bose)和雷·查德胡里(Ray Chaudhuri), 1959 年由霍昆格姆(Hocquenghem)提出的纠正多个随机错误的循环码, 它在 QR 码中被用于格式信息和版本信息的纠错^[19]。BCH 码的取名正是根据上述学者的名称而取首字母的组合。

欲构造 BCH 码, 首先需要指定纠错个数。循环码的生成多项式通常具有像公式 2-1 的格式。

$$F(x) = LCM [f_1(x), f_3(x), \dots, f_{2t-1}(x)]$$

公式 2-1

公式 2-1 中, LCM 表示最小公倍式, t 为可纠错的个数, $f_{2i-1}(x), i=1, \dots, t$ 为素多项式, 则生成的循环码 $F(x)$ 就称为本原 BCH 码, 其最小码距 $d \geq d_0 = 2t+1$, 它能纠正 t 个随机独立的差错。其中 BCH 码的码长为 $m = 2^n - 1$ (本原 BCH 码)或是 $2^n - 1$ 的因子 (非本原 BCH 码)。

假设 $BCH(15,5)$ 码能纠正 3 个随机独立的错误, 那么有 $t=3$, $d \geq d_0 = 2t+1 = 7$, $m=15 = 2^n - 1$, 所以 $n=4$, n 是多项式的阶数。查不可约多项式表可得:

$$\left. \begin{aligned} f_1(x) &= (23)_8 = 010011 = x^4 + x + 1 \\ f_3(x) &= (37)_8 = 011111 = x^4 + x^3 + x^2 + x + 1 \\ f_5(x) &= (07)_8 = 000111 = x^2 + x + 1 \end{aligned} \right\} \Rightarrow \begin{cases} F(x) = LCM [f_1(x), f_3(x), f_5(x)] \\ = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) \\ = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1 \end{cases}$$

BCH 码的纠错原理如下^[19]: 若产生多项式可以被生成多项式整除, 则接受码等于发送码, 表明数据没有发生错误, 不用纠错; 若不能被整除, 则产生余项, 说明数据在传输途中发生了错误, 我们根据此余项通过计算纠正产生的错误。

2.1.1.2 Reed-Solomon 码

有限域被称为伽罗华域, RS 码的运算是建立在伽罗华域上的。因为 QR 码的码字为 8 个 bit, 所以对应的伽罗华域为 $GF(2^8)$ 。QR 码在编码的过程中用到的数据码字和纠错码字都是取自 $GF(2^8)$ 上的域元素^[20]。

RS 码是 BCH 码的一种特殊情况, 它是由 MIT Lincoln 实验室的 S.Reed 和 G.Solomon 于 1960 年在 “Journal of the Society for Industrial and Applied Mathematics” 上发表的一篇文章发展而来的, 因而被称为 Reed Solomon 码, 被简称为 RS 码^[21]。

突发错误是日常生活中常见的一种错误,任何系统难免遇到。比如在一个通信信道中,信号因为衰退而导致错误,而当这种错误连续发生时,就把这种错误称为突发错误。一个码元发生错误往往对前后码元的出错概率起影响。因为 RS 码的编码系统是基于字节设计的,所以它非常适合用来处理这种突发错误。

本文未对 RS 算法进行深入的研究,仅仅作简单介绍。RS 码是 BCH 码中一个特殊的子集,它的定义为: 对于一个 q 进制的 BCH 码,任意挑选一个正整数可以构造一个码长为 $n = qs - 1$ 的 BCH 码,而且码元的符号取自有限域 $GF(q)$,其中 q 为素数的幕^[22]。RS 码就是当 $s = 1$, $q > 2$ 时所建立的 BCH 码,此时码长为 $n = q - 1$ 。当 q 取值为 $2^m (m > 1)$,即码元取自有限伽罗华域 $GF(2^m)$ 时,RS 便能用来纠正突发性错误。

RS 纠错编码中的输入数据被划分为 $k \times m$ bit 一组,就是每个符号有 m bit,一共 k 个符号。可以纠正 t 个错误的 RS 码的参数如表 2-1 Reed-Solomon 纠错编码参数选取所示。

表 2-1 Reed-Solomon 纠错编码参数选取

符号位数 m	符号个数 $n = 2^m - 1$	数据个数 $k = n - 2t$	纠错个数 t	纠错率 $c = \frac{t}{n}$	码长 mn (bit)
2	3	1	1	$\frac{1}{3}$	6
3	7	5	1	$\frac{1}{7}$	21
		3	2	$\frac{2}{7}$	
		1	3	$\frac{3}{7}$	
4	15	13	1	$\frac{1}{15}$	60
		11	2	$\frac{2}{15}$	
		9	3	$\frac{1}{5}$	
		7	4	$\frac{4}{15}$	
		5	5	$\frac{1}{3}$	
		3	6	$\frac{2}{5}$	
		1	7	$\frac{7}{15}$	

注: $m = 4$ 的 RS 码在彩色 QR 码编码解码程序中用到。

2.1.2 掩码技术

首先介绍二进制位的异或运算。计算机位操作当中的“异或”用符号 XOR 或者 \wedge 表示。异或运算法则是^[23]：对于运算符两侧数的每一个二进制位，同值取 0，异值取 1。对异或运算的简单理解就是不进位加法，如：

$$1 + 1 = 0, 0 + 0 = 0, 1 + 0 = 1$$

异或满足下述四条性质：

(1)交换律：

$$a \wedge b = b \wedge a$$

公式 2-2

(2)结合律：

$$(a \wedge b) \wedge c = a \wedge (b \wedge c)$$

公式 2-3

(3)对于任何数 x ，都有

$$x \wedge x = 0, x \wedge 0 = x$$

公式 2-4

(4)自反性：

$$A \wedge B \wedge B = A \wedge 0 = A$$

公式 2-5

异或操作最重要的性质是自反性质(公式 2-5)，即，任给一个数 A ，使用相同的计算因子 B 作两次异或后得到其本身。这是一个非常有用的性质，该运算可以应用在加密、数据传输、校验等领域。二维码的掩码就是用到了异或运算。

假设有一张图片 I ，构造一个同大小的掩码图像 M ，则编码的过程如下：

$$I^c = I \wedge M$$

公式 2-6

图像 I 中色彩密集的区域，在被特定的掩码图像 M 作用之后会分散开来，之所以使二维码前景色与背景色分离开，是为了看起来美观，并便于机器识别。解码的过程如下：

$$I^c \wedge M = I \wedge M \wedge M = I \wedge 0 = I$$

公式 2-7

QR 码中设计了 8 种掩码图像，如表 2-2 及图 2-2 所示。

表 2-2 QR 码掩码模式

掩模图形编号	发生条件	掩模图形编号	发生条件
000	$(x+y)\%2$	100	$(y/2 + x/3)\%2$
001	$y\%2$	101	$xy\%2 + xy\%3$
010	$x\%3$	110	$(xy\%2 + xy\%3)\%2$
011	$(x+y)\%3$	111	$(xy\%3 + (x+y)\%2)\%2$

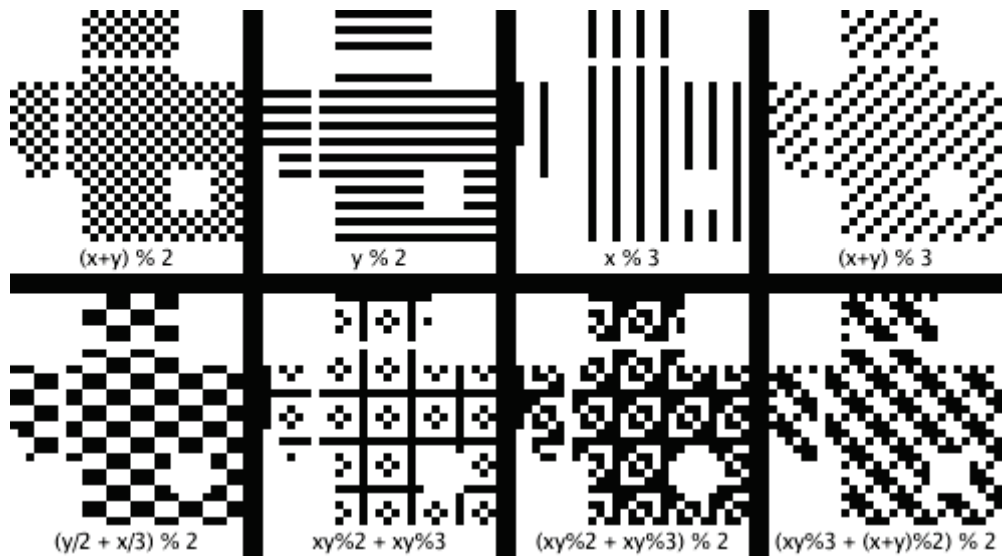


图 2-2 QR 码掩码模式

2.2 解码过程中的主要技术

QR 码解码整体来说就是三步走。

第一，获取 QR 码图像，对采集到的图像进行二值化处理；

第二，寻找定位符与校正符，然后将原图像中符号码部分取出(ZXing 中由 detector 代码实现，ZXing 的介绍参看 22 页)；

第三，对符号码矩阵按照编码规范进行解码，得到编码信息(ZXing 中由 decoder 代码实现，ZXing 的介绍参考 22 页)。

QR 码译码流程如图 2-3、图 2-4 所示。

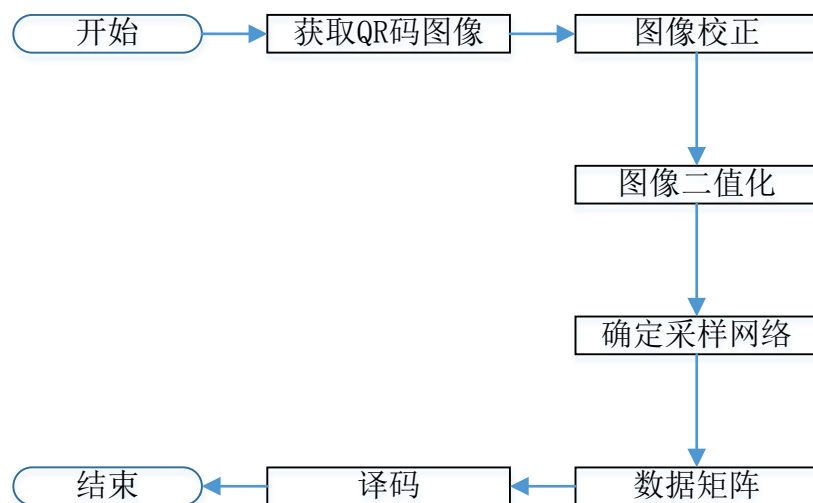


图 2-3 QR 码译码流程



图 2-4 译码详细流程

2.2.1 二值化

在 QR 码图像二值化之前,有许多文献指出需要先进行灰度化操作。灰度化操作是专门针对彩色 RGB 图像的,它将图像多个通道的灰度值加权得到一个最终灰度值。在 VS 2012 开发环境下,图像色彩用颜色索引 COLORREF 表示,需要将其转换为灰度值。

图像二值化是图像分析与处理中最常见也是最重要的处理手段之一,是进行一般图像处理之前必要的图像预处理过程^[24]。二值化操作将灰度图像转变为只有两个灰度值的图像,即二值图像。图像的二值化公式为

$$F(x,y)=\begin{cases} 1, f(x,y)\geq T \\ 0, f(x,y)<T \end{cases}$$

公式 2-8

其中, $f(x,y)$ 为原始图像的像素值, $F(x,y)$ 为二值化之后的像素值, T 是阈值。根据不同阈值选取方法,二值化方法可分为全局阈值、局部阈值和自适应阈值。

虽然二值化看起来特别简单,却受到了国内外学者在过去几十年的持续而广泛的关注,已经有数以百计的阈值选择方法。本文介绍以下几种获取图像二值化阈值的方法^{[25][26]},并仅针对 RGB 色彩空间。

方法一：极简方法

$$F(x, y) = \begin{cases} 255, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases}$$

公式 2-9

对 RGB 彩色图像灰度化以后，扫描图像的每个像素值，将值小于某个阈值 T 的像素值设为 0(黑色)，值大于等于 T 的像素值设为 255(白色)。该方法的优势是计算量少、速度快。该方法的缺点也很明显，一方面，阈值 T 不能被合理的解释，若是针对黑白二维码(通常可取 $T=128$)，却不失为一个可选择的方法；另一方面，该方法完全不考虑图像的像素分布情况与像素值特征。

方法二：灰度平均值方法

$$T = \frac{\sum_{g=0}^{255} g \times h(g)}{\sum_{g=0}^{255} h(g)}$$

公式 2-10

这个方法使用整幅图像的灰度平均值作为二值化的阈值，其结果可作为其他方法的初始猜想值。该方法操作过程如下，首先按照公式 2-10 计算像素的平均值 T ，然后对图像的像素逐个扫描，如果像素值大于 T ，那么像素值设为 255(白色)；如果像素值小于等于 T ，那么像素值设为 0(黑色)。

方法三：基于谷底最小值的阈值^{[27][28]}

该方法适用于具有明显双峰直方图的图像，即图像的灰度值集中在两个灰度中心附近出现。该方法寻找双峰的谷底最小值作为分割阈值，但是不一定能获得阈值，比如对于直方图平坦的图像或者单峰图像，该方法就不合适^[26]。该方法操作过程如下，首先计算图像的直方图，然后寻找直方图的两个最高的峰，最后取两个峰之间的峰谷最低处作为阈值。

方法四：基于双峰平均值的阈值

该算法和基于谷底最小值的阈值方法类似，只是最后一步不是取的双峰之间的谷底值作为阈值，而是取双峰的平均值作为阈值。

方法五：迭代最佳阈值

这是一种近似一维均值的方法，寻找二值化阈值，该方法的步骤如下：

1、设置一个初始化阈值，假设为 T ；

2、根据阈值 T ，把图像中每个像素归类为对象像素数据 G_1 或背景像素数据 G_2 ，即

$$F(x, y) = \begin{cases} (x, y) \in G_1, & f(x, y) \geq T \\ (x, y) \in G_2, & f(x, y) < T \end{cases};$$

3、计算 G_1 的平均值是 m_1 ， G_2 的平均值是 m_2 ，由此产生一个新的阈值 $T' = (m_1 + m_2)/2$ ；

4、回到第 2 步，用新的阈值继续分割像素，继续 2-4 步，直到计算出来的新阈值接近上一次阈值。

方法六：OTSU 方法

该算法是 1979 年由日本学者大津(OTSU)提出的，也叫作“大津算法”或“最大类间方差法”。该算法的主要思想是取某个阈值，使得前景和背景两类的类间方差最大，MATLAB 中的 `graythresh` 就是以该算法为原理执行的。

此外，基于直方图的阈值方法还有一维最大熵方法、力矩保持法^[29]、基于模糊集理论的阈值方法、Kittler 最小错误分类法^[30]、ISODATA(也叫做 intermeans 法)^[31]、Shanbhag 法^[32]、Yen 法^{[33][34]}，等等。

ZXing 对条码的二值化都是在 `Binarizer` 类中实现的，对一维码使用 `getBlackRow` 方法，对二维码使用 `getBlackMatrix` 方法。`Binarizer` 有两个派生类，`GlobalHistogramBinarizer` 和 `HybridBinarizer`；这两个类对 `getBlackMatrix` 方法的实现有所不同，`HybridBinarizer` 中的实现对于某些条件下的图像会做特殊的处理^[35]。

图 2-5 是对一个二维码图像进行二值化处理的结果。本文试验了两种方法。第一种方法采用灰度平均值方法，二值化结果如图 2-5 (b)所示；第二种方法采用 OTSU 方法，二值化结果如图 2-5 (c)所示。对比可知，OTSU 方法的效果更好。



图 2-5 一个图像的二值化

2.2.2 K-MEANS 聚类

“物以类聚，人以群分”，在自然科学和社会科学中，存在着大量的分类问题，聚类就是将实际或抽象对象的集合分成由类似的对象组成的多个类的过程^[36]。K-MEANS 聚类算法是一个简单而被广泛运用的算法，它的原理如下：

第一步，初始化 k 个类的中心，随机选取 k 个中心点，即 $C_k = rand(k)$ ；

第二步，遍历所有数据，计算每个数据到 k 个中心点的距离 $D_k = dis(P, C_k)$ ，它到哪个中心点距离最近，就将它归为该类，即 $\min D_k \Rightarrow P \in C_k$ ；

第三步，更新 k 个类的中心点，比如计算每个聚类的平均值，作为新的中心点，即

$$C_k = \frac{\sum_{i=1}^s P_i}{s}, P_i \in C_k;$$

第四步，重复步骤 2-3，直到这 k 个中心点不再变化(收敛了)，或者执行了足够多的迭代次数。

此算法的时间复杂度为 $O(I \cdot n \cdot k \cdot m)$ ，空间复杂度为 $O(n \cdot m)$ ，其中 m 为每个元素字段个数， n 为数据量， I 为迭代次数。通常情况下，我们可以把 I, k, m 均当作是常量，因此 K-MEANS 算法的时间与空间复杂度可以被简化成 $O(n)$ ，即线性的，算法速度快。

K-MEANS 算法具有计算时间短、速度快，容易解释，聚类效果还不错等优点，但也存在对异常值敏感、需要提前确定 k 值的缺点。下面讨论 K-MEANS 算法的收敛性。

数学上用误差项平方和 SSE 考察每个样本的各观测值的离散状况， SSE 又称为组内平方和或残差平方和。K-MEANS 的算法的 SSE 其实是一个严格的坐标下降过程。

$$SSE(c_1, c_2, \dots, c_k) = \sum_{i=1}^k \sum_{j=1}^M (x_j - c_i)^2$$

公式 2-11

SSE 采用欧式距离作为变量之间的聚类函数。每次朝一个变量 c_i 的方向找到最优解，也就是求偏导数，然后等于 0，可得 $c_i = \frac{1}{M} \sum x$ ，其中 M 是 c_i 所在的簇的元素个数，也就是当前聚类的均值就是当前方向的最优解(最小值)，这与 K-MEANS 的每一次迭代过程一样。所以，这样保证 SSE 每一次迭代时，都会减小，最终使 SSE 收敛。

由于 SSE 是一个非凸函数，所以 SSE 不能保证找到全局最优解，只能确保局部最优解。但是，可以重复执行几次 K-MEANS，选取 SSE 最小的一次作为最终的聚类结果。

2.2.3 图像滤波

因为实际扫描到的图像含有很多噪点，经过二值化的图像可能还是不能被正确解码，不能解码、解码出错。此时需要对图像进行滤波操作，以改善图片的质量。目前运用比较广泛的滤波算法是均值滤波和中值滤波。

均值滤波是一种线性滤波操作，适用于受高斯噪声污染的图像。均值滤波的操作过程是利用某个像素领域里的全部像素的平均值来代替该像素的灰度。均值滤波的过程图示见图 2-6。

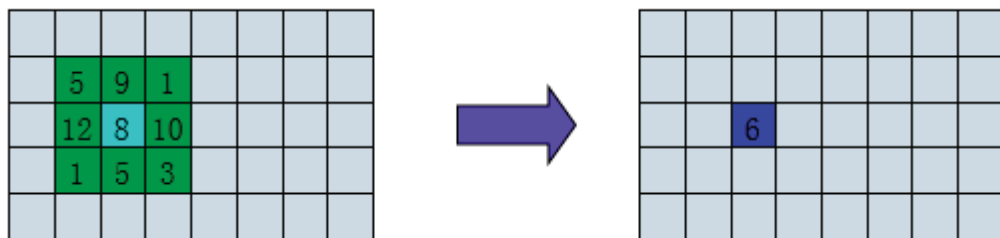


图 2-6 均值滤波图示

均值滤波的核是 $H_0 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, $\frac{1}{9}$ 表示权重, 从滤波核形式可知, 它对领域里所有的

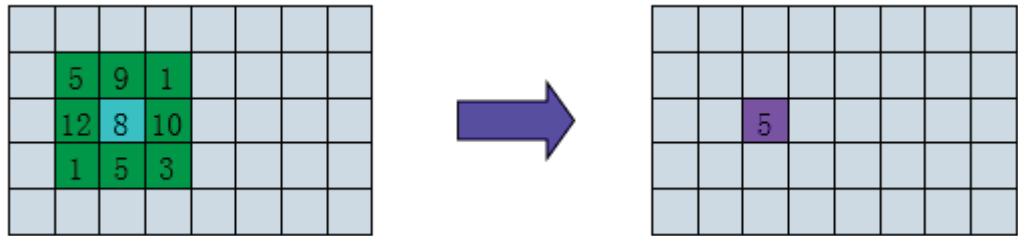
像素点同等对待, 在抑制噪声的同时也对图像造成了模糊效果。加权平均形式的滤波核分配给领域里像素点的权重不再相同, 对图像模糊有一定改善作用, 但是, 即使选用这种滤波核, 对图像质量的改善效果也是非常有限的。下面是几个常用的加权均值滤波器。

$$H_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

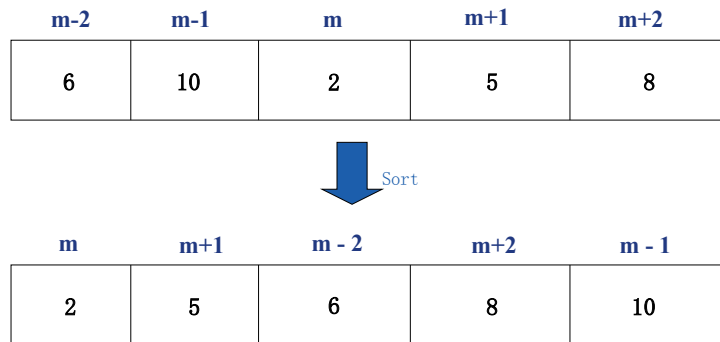
$$H_3 = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$H_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$H_4 = \frac{1}{2} \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$



(a)

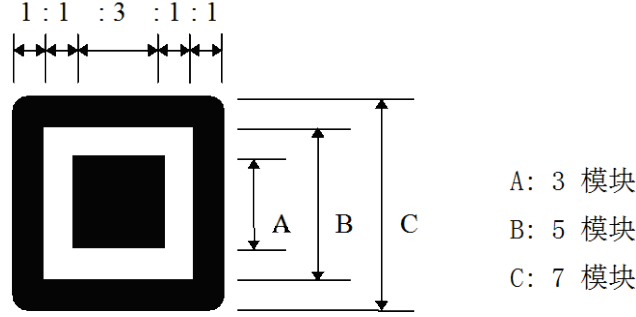


2 6

(b)

图 2-7 中值滤波图示

中值滤波是一种非线性滤波器, 具有一定的保边界能力。该算法把某像素相邻区域几个像素点的灰度值按值的大小进行排列, 排序过程如图 2-7 所示, 然后选取序列的中值作为当前像素点的灰度值。因为中值滤波算法是一种能保护图像边缘的算法, 而边缘是图像的重要属性, 因此中值滤波算法使用比较广泛, 对去除扫描噪声、椒盐噪声很有效果。中值滤波算法的设计思路简单明了, 时间复杂度低, 但学者表示, 对点、线和尖顶多的图像不宜采用中值滤波^[37]。

图 2-8 位置探测图形的结构^[38]

2.2.4 寻象图形的定位

QR 码的位置探测图形如图 2-8 所示。由定位图形的特征,对 QR 码图像进行逐行扫描,当检测到 1:1:3:1:1 这个比例时,说明发现了一个位置探测图形。假设待解码的 QR 码图像为 $I_{m \times n}$, 并且 I 已经经过了图像预处理。检测定位图形的方法如下:

第一步,求得图像 I 的导数图像并取绝对值,即 $D = |diff(I)|$;

第二步,对于第 i 行 $D_i = [d_{i1}, d_{i2}, \dots, d_{in-1}]$,找到导数值大于某个阈值 L (通常接近 255) 的列指标,即为 $Idx_i = find(D_i > L) = [c_{i1}, c_{i2}, \dots, c_{is}]$, s 小于 QR 码符号的大小;

第三步,对 Idx_i 求导,得到 $R = diff(Idx) = [r_{i1}, r_{i2}, \dots, r_{is-1}]$, R 的值就是白色或黑色像素块的长度,若存在相邻的 $r_{ij} : r_{i,j+1} : r_{i,j+2} : r_{i,j+3} : r_{i,j+4} = 1:1:3:1:1$ 就表示找到了一个位置探测图形。实际图像很难保证这个比例绝对准确,故应设置一个浮动区间。QR 码国家标准提供的参考译码算法中,每一元素宽度的允许偏差为 0.5。

根据 i, j 就能定位 QR 码图像的三个位置探测图形。这就是获取 QR 码位置探测图形的中心的总体思路,实际编程可能会更复杂。经过上述处理,并不是三个位置探测图形就得到三个中心坐标,而是一系列坐标,采用 K-MEANS 聚类算法将这些坐标聚为三类。

利用 MATLAB 验证此算法,编码生成用于检测的 QR 码图像 “test.jpg”(图 5-2 QR-汉信融合码),从 0 到 360 角度每隔 5 度旋转一次,获得 72 个图像。利用 MATLAB 自带的 K-MEANS 函数对于这个问题处理,在默认情况下,随机选择 3 个点作为初始中心,从结果看出,对于有的图像,该算法检测到的中心坐标是错误的。

如图所示第一种错误,孤立点 P 被归为同一个类 C 中,导致该类的中心发生偏移。如图所示第二种错误,原本应该分开的两个类 C_1, C_2 被归为一个类。本文分析产生这些错误的原因是, K-MEANS 随机选择初值导致的。于是本文提出选择初值的方法。

对于二维数据集 $Data$, 找到 x 维的最值 x_{\min}, x_{\max} , 提出两个点 $a_1 = (x_{\min}, y_1)$, $a_2 = (x_{\min}, y_2)$, 找到 y 维的最值 y_{\min}, y_{\max} , 也提出两个点 $a_3 = (x_1, y_{\min})$, $a_4 = (x_1, y_{\max})$, 从中选择 3 个作为中心坐标的初值,选择策略是使得两两之间的距离比较大,反过来说,两两距离最小的两个点中,只保留一个。这样,去掉了第二种错误。

针对第一种错误, 在数据已经被分为三类的基础上, 对每一类采用 $1-\delta$ 法则, 即将 $[\mu-\delta, \mu+\delta]$ 之外的点去除, μ 是均值, δ 是标准差。最后, 求出每一类的平均值作为该类中心。解决这两种错误之后, 程序对这 72 张 QR 码寻象的处理结果都是比较好的。

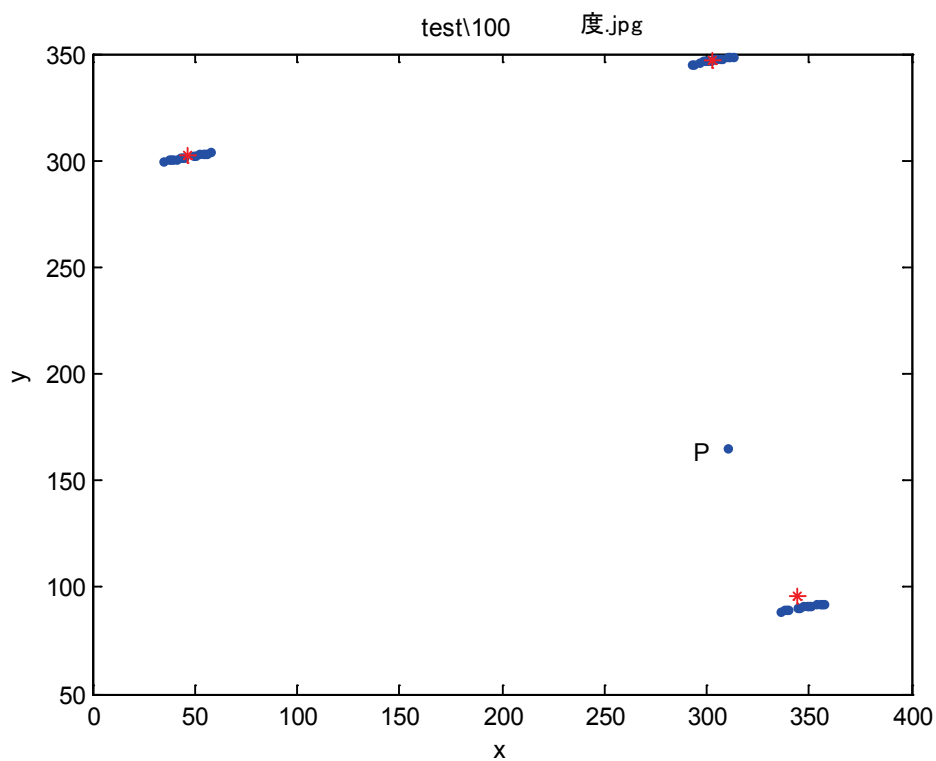


图 2-9 寻象第一种错误

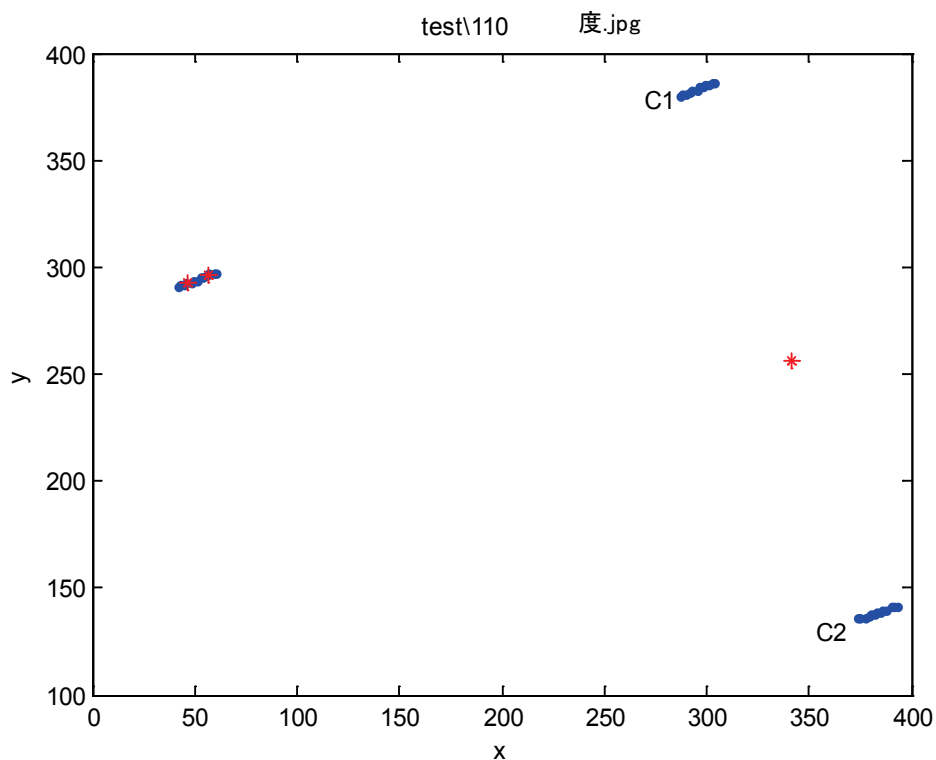


图 2-10 寻象第二种错误

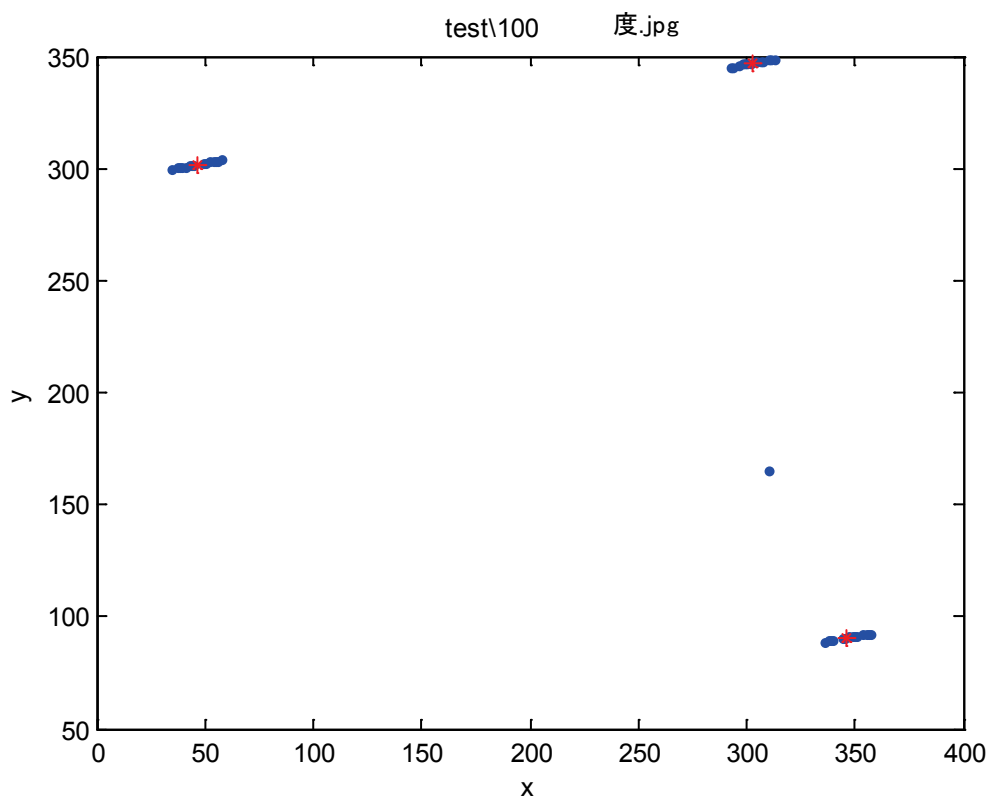


图 2-11 解决寻象第一类错误

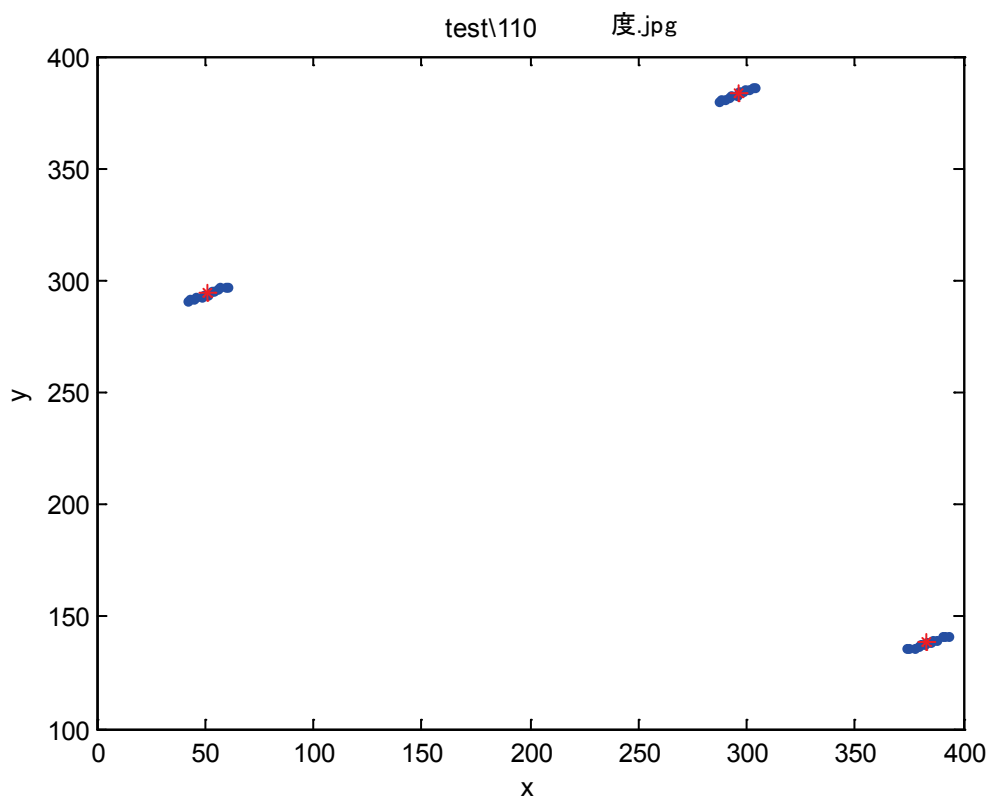


图 2-12 解决寻象第二类错误

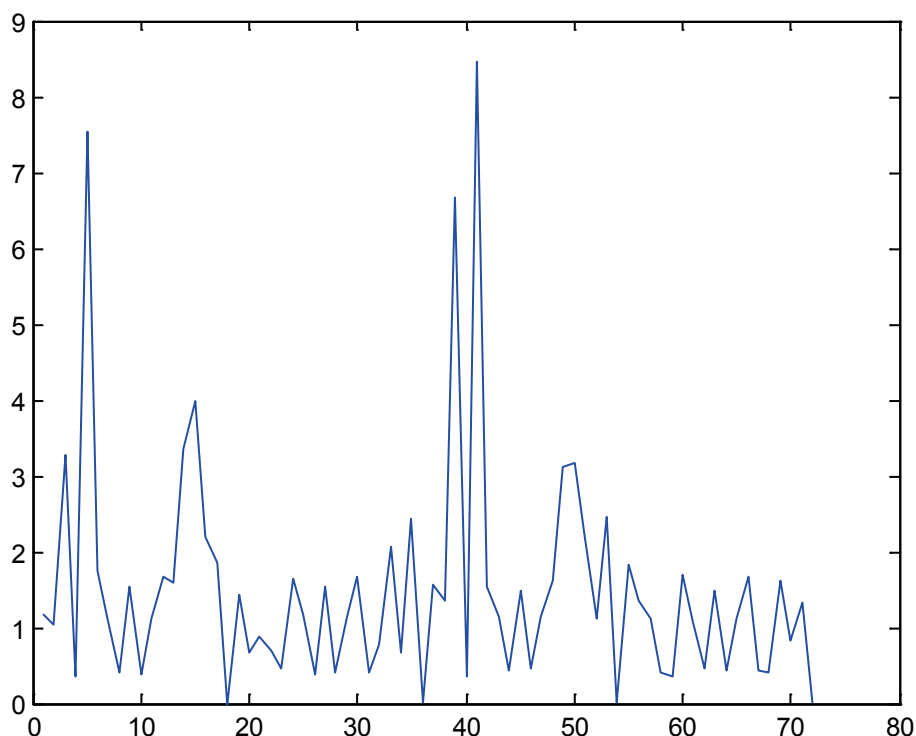
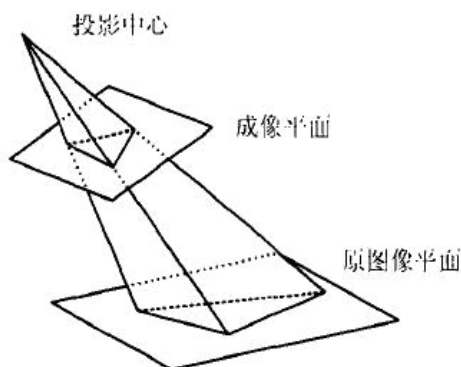


图 2-13 寻象函数性能图示

如图 2-13，最大误差是偏离 8.46 个像素，平均误差是偏离 1.5 个像素。

2.2.5 透视校正

实际采集的图像会失真、变形。透视变换是调整图像几何形状的一种方法，其实质和其他图像几何形状变换方法是一样的，是一种坐标变换。透视投影如图 2-14 所示。在文献 [39] 中，提供了两种思路求解目标图像：一是“前向映射”，把源图像的像素坐标 (x, y) 通过变形运算 $f(x, y) = (x', y')$ 转为目标图像上相应点坐标 (x', y') ；二是“逆向映射”，就是利用 f^{-1} 将目标图像像素坐标 (x', y') 映射到源图像 (x, y) 。这两种方法都会导致目标点的坐标会出现小数，需要进行插值处理。常用的插值方法有近邻插值(低复杂度)、线性插值、双线性插值以及样条插值等。

图 2-14 透视投影图解^[39]

当拍摄平面与 QR 码图像不平行时，例如拍歪情况下，获取到的图像就如图 2-15(或图 2-5 (a))所示。很明显，已经产生了透视畸变。



图 2-15 透视畸变的 QR 码

设 (x, y) 为图像上变换前的像素点，其对应的变换后的像素点为 (x', y') ，则透视变换公式为：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix}}{1 + \begin{bmatrix} g & h \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}}$$

其中， a, b, c, d, e, f, g, h 是 8 个透视变换参数。

为了确定原始图像到扫描图像之间的投影变换，必须首先确定公式中的 8 个参数，8 个参数需要 8 个方程。若能知道透视变换前后 4 对点的坐标就能解出这些参数。一种办法是通过计算 QR 码四个角的坐标来确定这些参数(文[39]中采用的方法)。假设 $(x_i, y_i), i=1, 2, 3, 4$ 是变换前的四个点， $(X_i, Y_i), i=1, 2, 3, 4$ 是变换后的四个点，那么， $B = [X_1 \ Y_1 \ X_2 \ Y_2 \ X_3 \ Y_3 \ X_4 \ Y_4]^T$ ，联立方程组 $Ar = B$ ，其系数为 A ，于是， $r = [a, b, c, d, e, f, g, h]^T = A^{-1}B$ 。

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -X_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -Y_1x_1 & -Y_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -X_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -Y_2x_2 & -Y_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -X_3x_3 & -X_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -Y_3x_3 & -Y_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -X_4x_4 & -X_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -Y_4x_4 & -Y_4y_4 \end{bmatrix}$$

不同于文[39]，ZXing 采用的是另外 4 个点，即通过上文寻找到的三个位置探测图形的中点坐标，计算出校正符的可能位置，然后去探求最接近右下角的校正符，寻找方法与寻找定位符的方法基本相同。如果找到，就返回校正符的中心坐标；否则，可用一个计算值来代替，解码程序可以继续执行。

ZXing 计算得到的 4 个点为 $P_0 = (215.0, 32.5)$, $P_1 = (212.5, 142.0)$, $P_2 = (104.5, 153.0)$, $P_3 = (116.0, 41.5)$, 校正后的图像如图 2-16 所示(采用最临近插值)。



图 2-16 透视校正的 QR 码

2.3 QR 码编解码库介绍

有关 QR 码编解码, 在开源条件下有几个常用的库, 如 ZXing、ZBar、libqrdecode、libqrencode 等。QR 二维码图像的解码过程涉及到许多图像处理知识, 图像预处理搞得不好直接关系解码的成功与否。

2.3.1 ZXing

ZXing(“zebra crossing”)是一个 Java 类库, 一个由 Google 主导的开源项目。ZXing 不仅能够对 QR 码、Data Matrix、PDF 417 等二维条形码进行解码, 还能对许多一维条形码进行解码。ZXing 提供了多种平台下的客户端, 包括 J2ME、J2SE 和 Android。ZXing 提供了 Android、C++、C#、iPhone、J2ME、J2SE、JRUBY、Symbian 等多种应用的类库, 被程序开发者广泛使用。ZXing 在 GITHUB 上的主页是 <https://github.com/zxing/zxing>, 任何遵守其开源协议的人都可以自由下载使用。

2.3.2 ZBar

ZBar 是一个一维码或二维码扫描工具, 支持手机相机和图像扫描, 还支持多种平台, 特别是它有利于 iPhone 手机的支持, 被 iOS 开发人员广泛使用。此外 ZBar 还提供二维条码扫描 API 开发工具包。ZBar 目前支持扫描, 除了 Windows 平台外, 还支持 Linux 及 iPhone 平台。可扫描以下常见类型的条码, 如: EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, QR Code。ZBar 可以在 GITHUB 获取: <https://github.com/ZBar>。ZBar 主要被 iPhone 平台的开发者使用。

对比 ZXing 与 ZBar 两个类库, 并经过调研, 本文采用的是基于 ZXing 的 C++开发包。

第三章 彩色 QR 码的编解码设计

3.1 彩色编码

在 QR 码图形中,我们先做如下假定,高反射率的方块(白色模块)称为前景色,低反射率的方块(黑色模块)称为背景色。彩色 QR 码编码分为两层,本文称之为明文编码和密文编码(或者说普通编码和彩色编码),前者是普通的 QR 码编码,后者则是将两种色彩进行编码。

彩色的编码是在 QR 码编码完成之后进行的,编码形成的比特流排列到 QR 码的前景色上面。这样,由于彩色是化学材料发光产生,因而打印出来的二维码看起来就是 QR 码,彩色被隐藏掉了。

3.1.1 信息格式

为了解码能方便进行,必须有一个“数据头”来说明密文是如何编码、携带的是何类信息等。QR 码有三个位置探测图形,已将 QR 码上下颠倒,使位置探测图形组成 XOY 坐标系。如图所示,右侧、左下角和顶端是三个形状一样 7×7 大小的位置探测图形。观察发现,每个位置探测图形前景色有 16 个模块(图中以红色标记),解释成二进制位即一共有 16 位,位置探测图形分割符共有 15 个模块(图中以蓝色标记),即 15 位。



图 3-1 信息格式编码区(红色与蓝色)

为了简单起见,将三个位置探测图形的 48 位编码成“数据头”,携带内层信息编码的参数。编码头采用 $RS(3,1)$ 纠错编码算法,表示三个符号中有一个是数据符号,其余两个是纠错符号,两个纠错符号可以纠正一个错误,符号是来自 $GF(2^2)$ 中的元,2 个二进制位是一个符号, $RS(3,1)$ 能够纠正 $\frac{1}{3}$ 的错误。

现在考虑将位置探测图形分隔符的 $15 \times 3 = 45$ bit 像素值取 42 bit 进行 $RS(3,1)$ 纠错编码, 可以编码的数据比特数是 $42 \times \frac{1}{3} = 14$ bit。我们利用这 14 bit 编码 LOGO 的信息, 这样, 彩色 QR 码即支持带有 LOGO 的二维码了。

表 3-1 “数据头”与 LOGO 信息编码

数据头 (16 位)	1-10 位	11-13 位	13-16 位	
	存放数据长度, 最长为 $2^{10} - 1 = 1023$	存放纠错等级, 共 7 个版本	存放彩码掩码版本, 共 7 个版本	
LOGO 信息 (14 位)	1 位	2-6 位	7-10 位	11-14 位
	是否有内嵌 LOGO 的标志, 取 0 或 1	编码版本, 从 0 到 31	LOGO 宽度, 从 0 到 15	LOGO 高度, 从 0 到 15

其中所谓的掩码, 与 QR 码中的掩码有本质不同, 这里指的是一种掩饰方法, 使彩色编码按某种规则分散开来, 避免色彩过于集中, 利于解码。

3.1.2 密文编码

彩色编码的方法采用 $RS(15, k)$ 编码, k 是数据符号个数, 最大能纠正 $\frac{7}{15}$ 的错误 ($k=1$)。4 个二进制位是一个符号, 一组符号需要 $4 \times 15 = 60$ 位。为了简化编程, 待编码的字串都先被转换为 UTF-8 模式。

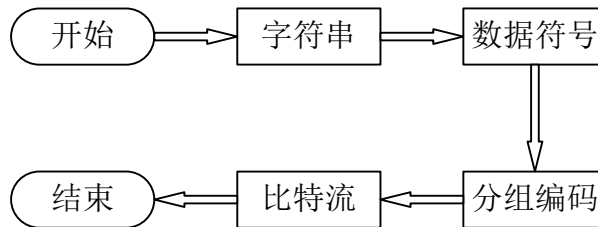


图 3-2 密文编码流程

第一步, 分析用户的输入信息, 转换为 UTF-8 模式的字符串, 字符串长度为 L , 纠错等级为 t ;

第二步, 将字符串以每 4 bit 进行分割, 得到数据符号, 个数为 $2L$, 将数据符号进行分组, 分组个数为 $B = \left\lceil \frac{2L}{k} \right\rceil$, 其中 $k = 15 - 2t$ 为每组数据符号个数;

第三步, 分组编码, 在 k 个数据符号后面需要补充 $2t$ 个纠错符号;

第四步, 获得纠错编码之后的比特流, 比特个数为 60。

彩色编码主要由 RS4Encode 函数完成, 该函数有四个参数, 分别是 BYTE* dataPtr, 一个指向编码字符串的 BYTE 型指针, int dataLen, 字符串长度, int &bitsLen, 返回数据长度, 即二进制位数, int ecLevel, 纠错等级。函数返回编码的二进制结果, 用 int 数组存放。

/* RS 纠错编码 */

```

int* RS4Encode(BYTE* dataPtr, int dataLen, int &bitsLen, int ecLevel)
{
    // 4bits 为一个符号，信息符号数
    int nMsgSymbols = 2 * dataLen;
    // 信息占比,一次能编码的符号数
    int nDataRate = 15 - 2 * ecLevel;
    // 15 个符号为一块，块数
    int nBlocks = ceil(1.f * nMsgSymbols / nDataRate);
    // 编码需要的符号总数
    int nRealMsgSymbols = nBlocks * 15;
    // 存放编码结果的位总数
    bitsLen = 4 * nRealMsgSymbols;
    int temp[15], bits[32];
    int *num = new int[nBlocks * nDataRate];
    int *result = new int[bitsLen];
    SplitByteBy4Bits(dataPtr, dataLen, num, nBlocks * nDataRate);
    reedSolomon rs(4, ecLevel);
    for (int j = 0; j < nBlocks; ++j)
    {
        ZeroMemory(temp, 15 * sizeof(int));
        int* cur = num + j * nDataRate;
        memcpy(temp, cur, nDataRate * sizeof(int));
        rs.set_msgfor_encode(temp);
        rs.encode();
        rs.get_encoded_msg(temp);
        for (int i = 0; i < 15; ++i)
        {
            Int2Bits(temp[i], bits, 4);
            result[60 * j + 4 * i] = bits[0];
            result[60 * j + 4 * i + 1] = bits[1];
            result[60 * j + 4 * i + 2] = bits[2];
            result[60 * j + 4 * i + 3] = bits[3];
        }
    }
    SAFE_DELETE(num);
    return result;
}

```

3.2 彩色解码

密文解码是建立在明文解码成功的基础上。

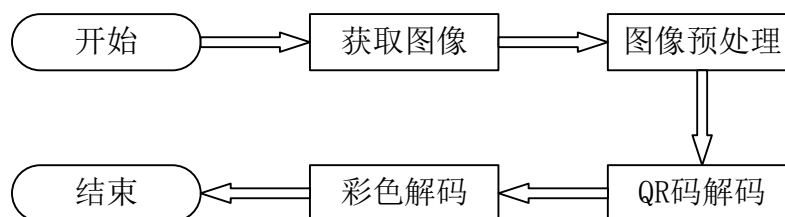


图 3-3 密文解码流程

由于 QR 码不再是黑白色彩，直接调用原有的解码方法可能不会成功。实际上，彩色图像的颜色在内存中是用 COLORREF 表示，COLORREF 是一个 32bit 整型数值。RGB(r,g,b) 是一个设置像素 COLORREF 值的宏，两者之间具有下述关系

$$rgb(r, g, b) = COLORREF = r + g \cdot 256 + b \cdot 255 \cdot 256$$

公式 3-1

因为彩色编码于 QR 码的前景色上面，根据二值化原理，我们可以猜想当彩色的 RGB 值比较接近白色 RGB(255,255,255) 时，彩色 QR 码能够解析成功。实验表明确实如此，下述几种彩色 QR 码用微信都可以解码。

表 3-2 彩色 QR 码 1

彩色 QR 码			
彩色 1	(0, 255, 0)	(255, 255, 0)	(0, 255, 0)
彩色 2	(0, 255, 255)	(0, 255, 255)	(255, 255, 0)

对于彩色 1 取红色、彩色 2 取绿色，微信则不能解析。本文采用 K-MEANS 聚类算法将背景色提取出来，则能顺利解析 QR 码。

表 3-3 彩色 QR 码 2

	=		+	
彩色 QR 码	=	彩色	+	QR 码

彩色解码的步骤为：

第一步，利用 K-MEANS 算法将 QR 码抓取出来；

第二步，解析 QR 码，并得到此 QR 码的前景色；

第三步，解析彩色信息。先解析得到“数据头”与 LOGO 信息，再解析彩色。彩色解码主要由 RS4Decode 函数完成，它有三个参数，int* dataSrc，彩色经过二值化之后得到的比特流，以整型数组存放，int dataLen，数据长度，int ecLevel，纠错等级。函数返回解码的二进制结果，以整型数组存放。

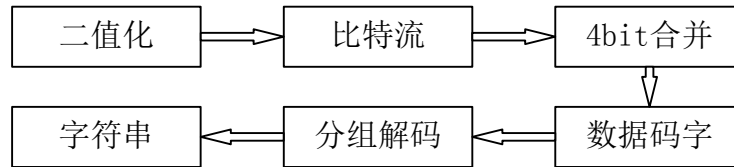


图 3-4 彩色解码流程

/ RS 纠错译码 */*

```

int* RS4Decode(int* dataSrc, int dataLen, int ecLevel)
{
    int nMsgSymbols = 2 * dataLen;
    int nDataRate = 15 - 2 * ecLevel;
    int nBlocks = ceil(1.f * nMsgSymbols / nDataRate);
    int nRealMsgSymbols = nBlocks * 15;
    int bitsLen = 4 * nRealMsgSymbols;
    MergedIntBy4Bits(dataSrc, bitsLen);
    int temp[15], bits[32] = { 0 };
    int *result = new int[nBlocks * nDataRate * 4];
    reedSolomon rs(4, ecLevel);
    for (int j = 0; j < nBlocks; ++j)
    {
        memcpy(temp, dataSrc + 15 * j, 15 * sizeof(int));
        rs.set_msgfor_decode(temp);
        rs.decode();
        rs.get_decoded_msg(temp);
        for (int i = 0; i < nDataRate; ++i)
        {
            Int2Bits(temp[2 * ecLevel + i], bits, 4);
            result[4 * nDataRate * j + 4 * i] = bits[0];
            result[4 * nDataRate * j + 4 * i + 1] = bits[1];
            result[4 * nDataRate * j + 4 * i + 2] = bits[2];
            result[4 * nDataRate * j + 4 * i + 3] = bits[3];
        }
    }
    return result;
}

```

第四章 彩色 QR 码编码解码程序

本文设计的彩色 QR 码编码解码程序以 Microsoft Visual Studio 2012 为开发工具，使用了 C++ 接口的 ZXing 类库，并参考了 GITHUB 等网站的一些资料，详情请看 4.2 节。PC 端程序基于微软基础类库 MFC。所谓 MFC，就是被一些 C++ 类封装了的 *Windows* 应用程序接口，里面包含了开发应用程序常用的框架，大大减少了开发人员的工作量，有关 MFC 请参看《深入浅出 MFC》。Android 平台的 APP 借助于 NDK 实现移植。所谓 NDK，就是 Google Android NDK，有关 NDK 开发环境的搭建、开发 NDK 程序等事项，可以参考文[40]。

4.1 程序概要设计

在布局程序界面之前先进行程序的概要设计，明确应用程序的输入输出，确定软件设计的目标、限制以及模块之间的接口。程序编码模块的简要流程图如图 4-1 与图 4-2 所示。

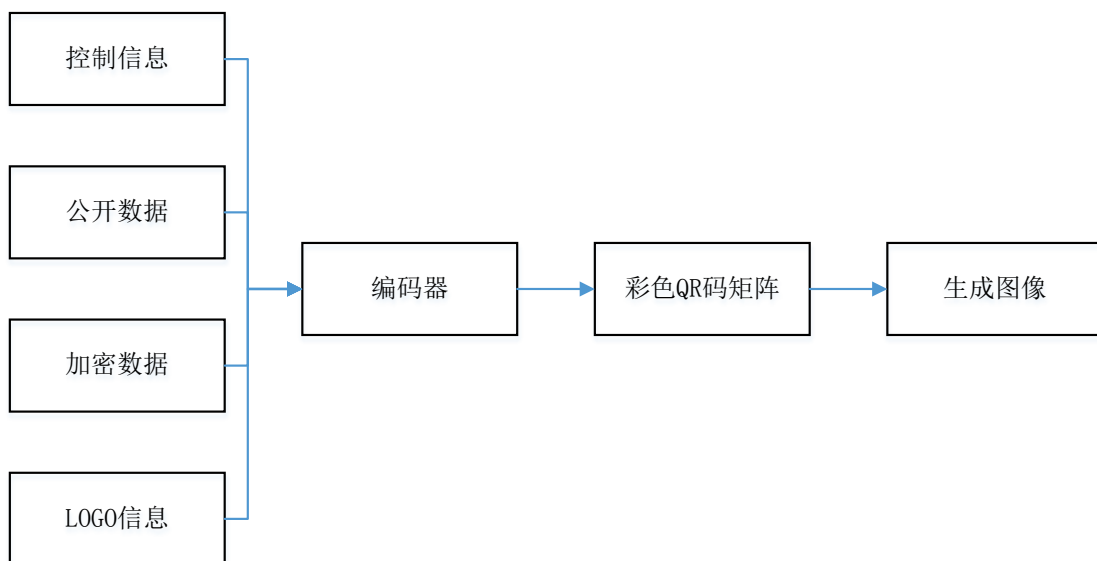


图 4-1 编码系统简要流程图

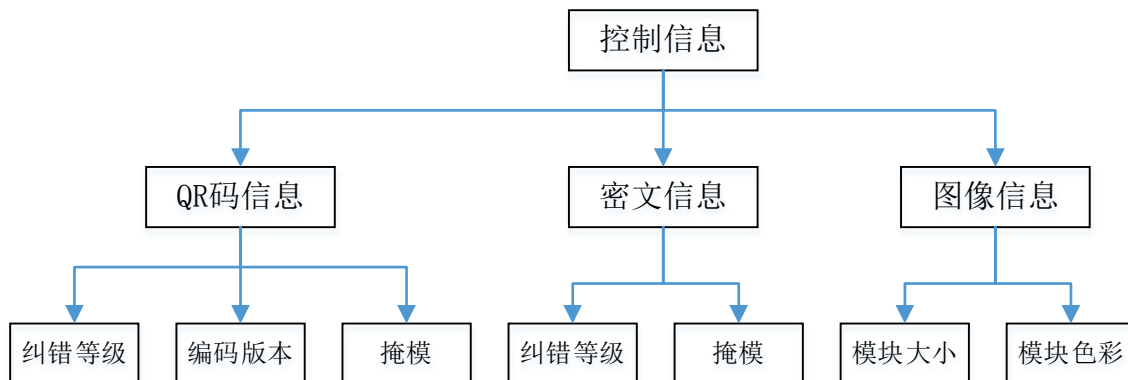


图 4-2 彩色 QR 码的控制信息

解码是编码的逆过程，解码模块上用来展示的条目与编码模块的条目一样，只不过控制信息、公开数据、加密数据与 LOGO 信息此时是作为输出。解码可能也需要提供一些参数，使得解码效果更高，使操作更方便。

4.2 程序详细设计

1 程序目标

彩色 QR 码编码解码程序应能把用户输入的信息编码成彩色 QR 码，能对彩色 QR 码图像进行解码，并输出此二维码的编码信息。

2 参考资料

编写本应用程序，作者参考了下述资料。

[1] 侯俊杰. 深入浅出 MFC(第二版)[M]. 武汉: 华中科技大学出版社, 2001-1.

[2] GITHUB. ZXing Project: <https://github.com/zxing/zxing>

[3] codeproject. Simple C/C++ QRCode Library:

<http://www.codeproject.com/Articles/593591/Simple-C-Cplusplus-QRCode-Library>

[4] GITHUB. reed_solomon: https://github.com/nceruchalu/reed_solomon

3 总体设计

彩色 QR 码编码解码软件采用了文档/视图结构，相关内容请参看文[41]和[42]。

在 Visual Studio 2012 中使用 MFC 开发基于文档的应用程序非常方便，MFC 为开发人员提供了几种基本框架，文档/视图结构是用的比较广泛的一种。顾名思义，文档类就是负责管理数据，用户对数据所做的任何改变由之完成，而视图是可视化之后的文档，用户通过视图能方便地实现对数据的访问和更新。本文中二维码图像及其属性作为文档类的成员，视图负责显示图像，如图 4-3。

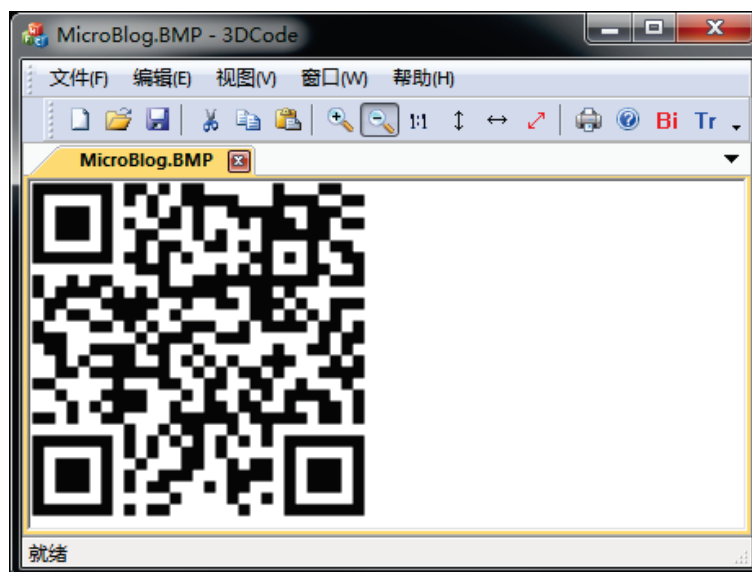


图 4-3 文档/视图框架

4 模块布局

如图 4-4 所示是编码与解码模块。需要特别说明的是，两个模块均能方便地独立出来，集成在一起是考虑到测试更方便一些。

当用户输入公开数据时，生成普通的 QR 码，如果继续输入加密数据，则生成彩色 QR 码。可以利用有关控件控制编码参数，比如 QR 码掩码模式，比如密文纠错等级，比如模块大小。当用户打开一副二维码图像时，会在解码模块中预览，点击“解码”按钮就能尝试进行解码，如果成功了，则会输出二维码的编码信息及编码参数。



图 4-4 编码与解码模块

5 编码解码测试

下面是对明文“首都师范大学”和密文“CNU”进行的彩色编码，QR 码参数及彩色编码参数可从图 4-5 看到。从解码的结果来看，解码模块正常工作，所得到的 QR 码版本、纠错等级、掩码版本与编码时的设定一致，密文的纠错等级及掩码版本也是对的。

图 4-5 解码测试所用的二维码图像是由程序生成的，几乎没有任何杂点，因此这个试验只是为了验证解码模块能否正常工作。

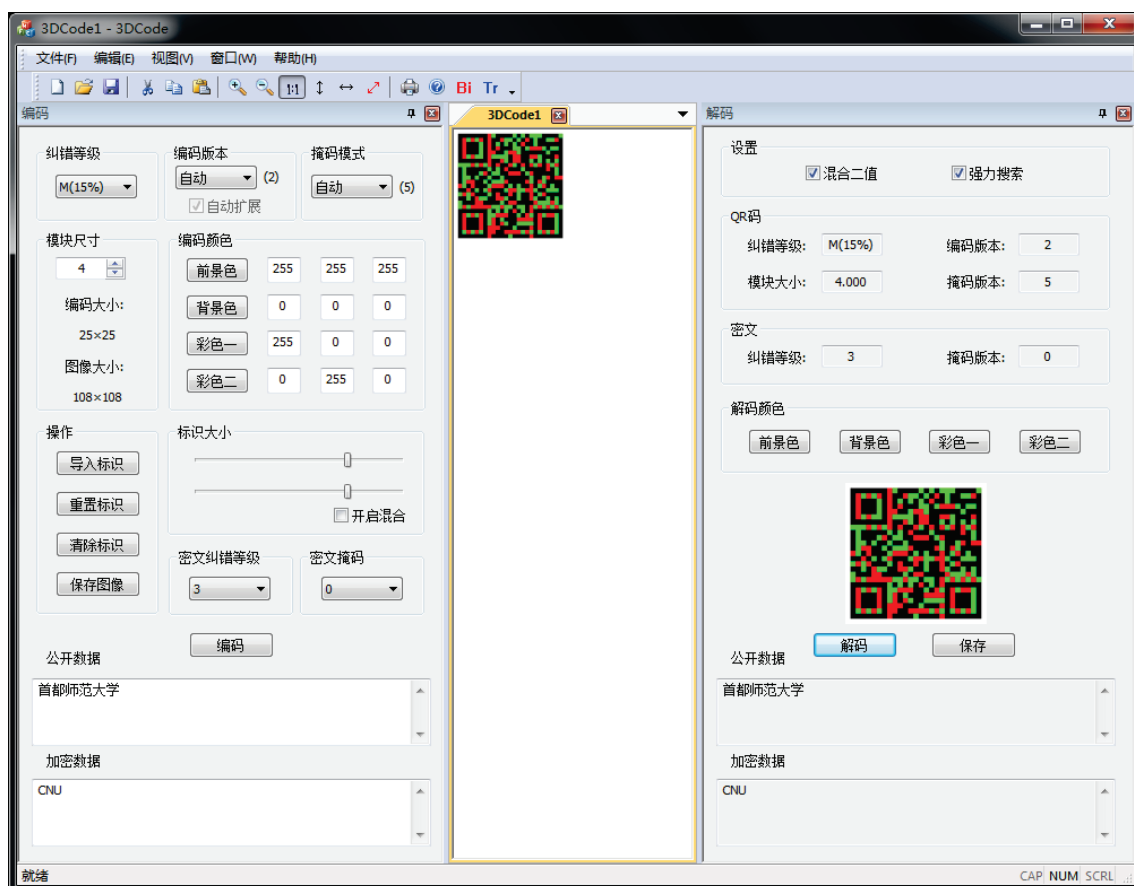


图 4-5 编码与解码测试

如图 4-6 是对一张用手机拍摄的彩色 QR 码图像进行解码的程序截图，这表明解码程序具有一定的可行性。图 4-7 是对编码明文“我心向阳，你心向月”和密文“CNU”的图像进行解码的实例，该二维码携带有首都师大 LOGO 标志。

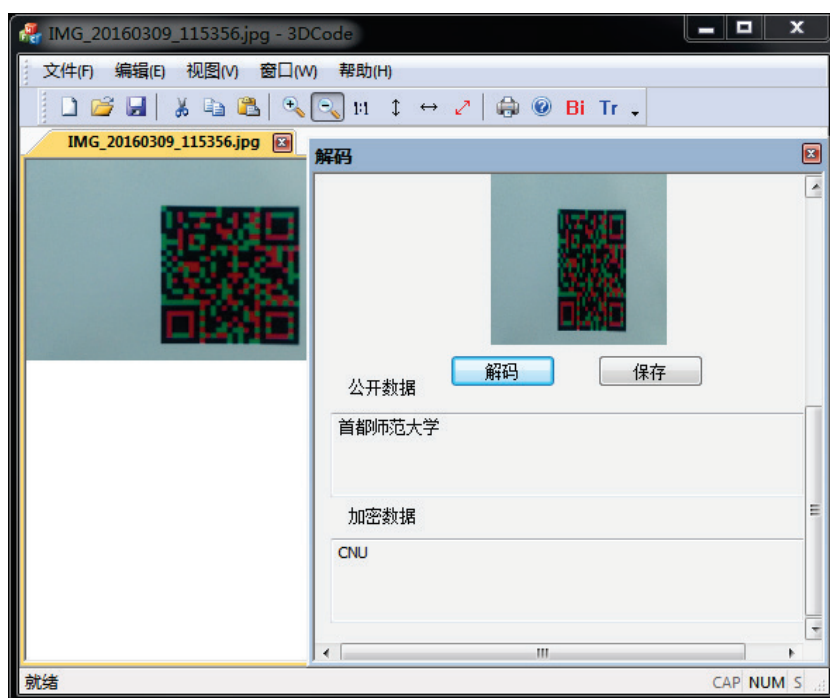


图 4-6 PC 端解码实例

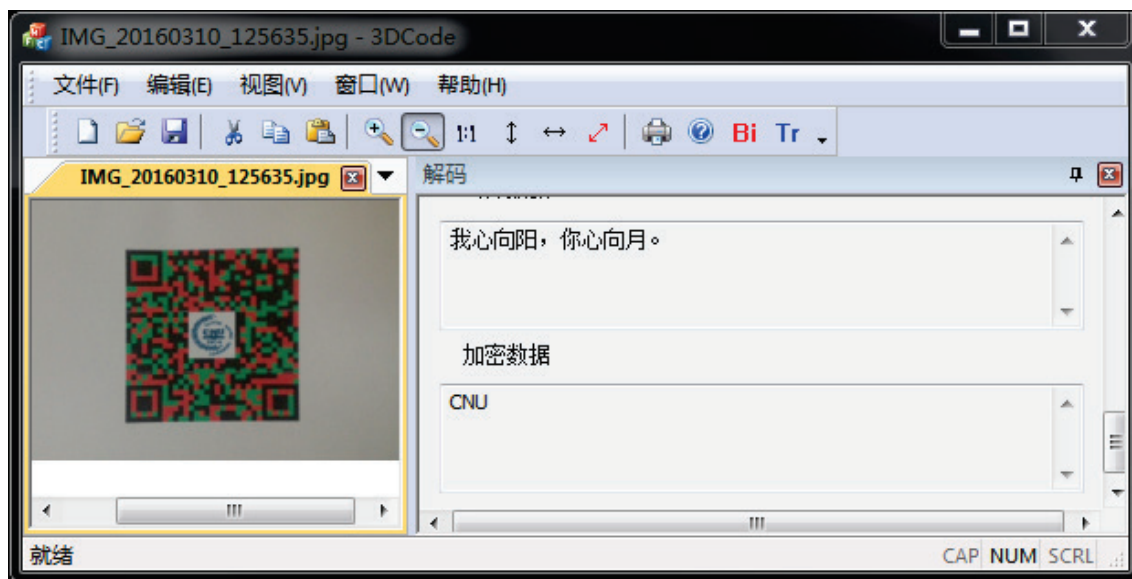


图 4-7 PC 端解码实例(带 LOGO)

4.3 安卓 APP

安卓 APP 采用 Eclipse 结合 ADT 开发。Eclipse 是一个基于 Java 开源的著名的集成开发环境(IDE)，可以跨平台自由使用。ADT 的全称为“Android Development Tools”，就是安卓开发工具。利用 NDK，将 C++源码编译成安卓语言动态链接库，现已证明编码和解码模块都可以移植到智能手机的安卓平台。

4.4 编码解码实例

(1) MK 文件

在使用 NDK 时，有两个关键的.mk 文件，这是移植 C/C++代码到安卓上面非常重要的文件。这两个文件是用来描述工程的本地模块，通常放在工程目录 jni 文件夹下面。*Application.mk* 文件指定了 ABI、平台、使用 STL 等，*Android.mk* 文件指明了将从 PC 移植过来的 CPP 文件进行编译。详情请参阅文[43]与文[44]。

Application.mk 文件

```
APP_ABI := all
APP_PLATFORM := android-10
APP_STL := stlport_static
APP_CPPFLAGS += -fexceptions
APP_CPPFLAGS += -frtti
```

Android.mk 文件

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := ColorCode
```

```

MY_CPP_LIST := $(wildcard $(LOCAL_PATH)/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/common/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/common/detector/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/common/reedsolomon/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/qrcode/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/qrcode/decoder/*.cpp)
MY_CPP_LIST += $(wildcard $(LOCAL_PATH)/zxing/qrcode/detector/*.cpp)
LOCAL_SRC_FILES := $(MY_CPP_LIST:$(LOCAL_PATH)/%=%)
include $(BUILD_SHARED_LIBRARY)

```

(2) CPP 接口实现

除此以外，需要单独另写一个 CPP 文件，用来编写接口函数，本项目中此文件为 *ColorCode.cpp*。鉴于该文件至关重要，本文附上该文件中编码解码函数。

```

extern "C" {

    /* 编码函数 */

    JNIEXPORT
    jint JNICALL Java_cnu_yuanyuanxiang_colorcode_EnDeCode_Encode(
        JNIEnv* env,
        jobject thiz,
        jintArray nResultArray,
        jint nEcLevel,
        jint nQRVersion,
        jboolean bIsAutoExtent,
        jint nMasking,
        jbyteArray utf8,
        jint bytesNum);

    /* 解码函数 */

    JNIEXPORT
    jobjectArray JNICALL Java_cnu_yuanyuanxiang_colorcode_EnDeCode_SimpleDecode(
        JNIEnv* env,
        jobject thiz,
        jintArray pHead,
        jint nWidth,
        jint nHeight);
};

```

```

/* 编码函数 */
JNIEXPORT jint JNICALL Java_cnu_yuanyuanxiang_colorcode_EnDeCode_Encode(
    JNIEnv* env,
    jobject thiz,
    jintArray nResultArray,
    jint nEcLevel,
    jint nQRVersion,
    jboolean bIsAutoExtent,
    jint nMasking,
    jbyteArray utf8,
    jint bytesNum)
{
    jint *pResults = env->GetIntArrayElements(nResultArray, (jboolean*)false);
    if(pResults == NULL) return 0;
    jbyte *pUtf8 = env->GetByteArrayElements(utf8, (jboolean*)false);
    if(pUtf8 == NULL) return 0;
    int ncLength = bytesNum + 1;
    char *temp = new char[ncLength];
    memcpy(temp, pUtf8, bytesNum);
    temp[bytesNum] = 0;
    CQR_Encode* pEnCodeQR = new CQR_Encode;
    pEnCodeQR->EncodeDataUtf8(nEcLevel, nQRVersion, bIsAutoExtent, nMasking,
temp, ncLength);
    SAFE_DELETE(temp);
    for (int i = 0; i < pEnCodeQR->m_nSymbleSize; ++i)
    {
        for (int j = 0; j < pEnCodeQR->m_nSymbleSize; ++j)
        {
            pResults[j + i * 177] = pEnCodeQR->m_byModuleData[i][j];
        }
    }
    delete pEnCodeQR;
    env->ReleaseIntArrayElements(nResultArray, pResults, 0);
    env->ReleaseByteArrayElements(utf8, pUtf8, 0);
    return (pEnCodeQR->m_nSymbleSize);
}

```



```

/* 解码函数 */
JNIEXPORT jobjectArray JNICALL Java_cnu_yuanyuanxiang_colorcode_EnDeCode_SimpleDecode(
    JNIEnv* env,
    jobject thiz,
    jintArray pHead,
    jint nWidth,
    jint nHeight)
{
    jint *pIntbytes = env->GetIntArrayElements(pHead, (jboolean*)false);
    float2 m_fPos[4];
    float m_fModuleSize = 0.f;
    int nEcLevel = 0, nQRVersion = 0, nMasking = 0;
    int ncLength = 0, _ncLength = 0;
    int strLen = 0, nInnerecLevel = 0, nInnerMask = 0;
    char* pDst[2] = { 0 };
    DecodeWholeImage(pDst, pIntbytes, nWidth, nHeight,
        m_fPos, m_fModuleSize, nEcLevel, nQRVersion, nMasking,
        TRUE, TRUE,
        RGB(255, 255, 255), RGB(0, 0, 0),
        RGB(255, 0, 0), RGB(0, 255, 0),
        ncLength, strLen, nInnerecLevel, nInnerMask);
    jobjectArray results = (env->NewObjectArray(2,
(env->FindClass("java/lang/String"), 0);
        if (pDst[0])
        {
            env->SetObjectArrayElement(results, 0, Cstr2Jstring(env, pDst[0]));
        }
        if (pDst[1])
        {
            env->SetObjectArrayElement(results, 1, Cstr2Jstring(env, pDst[1]));
        }
        env->ReleaseIntArrayElements(pHead, pIntbytes, 0);
        SAFE_DELETE(pDst[0]);
        SAFE_DELETE(pDst[1]);
        return results;
    }
}

```

(3) JNI 接口类

ColorCode.cpp 中函数的名字之所以很长, 这是因为它是根据 JNI 接口类(class EnDeCode)中的函数名及 JNI 方法命名规则命名的。在 EnDeCode 类中有多少个标记为 native 的方法, 相应地, 在 *ColorCode.cpp* 当中就应当有多少个实现。编译成功的 SO 库, 在 MainActivity 通过

```
static {System.loadLibrary("ColorCode");}
```

加载进来, 然后在主程序中声明一个 EnDeCode 类的对象, 就可以调用它的编码解码方法了。

EnDeCode 类的声明如下。

```
package cnu.yuanyuanxiang.colorcode;

public class EnDeCode {

    /* 编码函数*/
    public native int Encode(
        int nResult[],           // 编码结果
        int nEcLevel,           // 纠错等级
        int nQRVersion,         // 编码版本
        boolean bIsAutoExtent,  // 自动扩展
        int nMasking,           // 掩码版本
        byte utf8[],            // 编码数据
        int bytesNum);          // 数据字节数

    /* 解码函数*/
    public native String[] SimpleDecode(
        int pHead[],            // 图像指针
        int nWidth,             // 图像宽度
        int nHeight);           // 图像高度
}
```

(4) 实例

如图 4-8 (a)是对文字“首都师范大学”编码的结果, 图 4-8 (b)是彩色 QR 码扫描解码实例, 均达到了预期效果。

如图 4-9 (a)是对带 LOGO 的 QR 码图像进行解码得到的结果, 图 4-9 (b) 是对彩色 QR 码图像进行解码得到到结果, 均达到了预期效果。

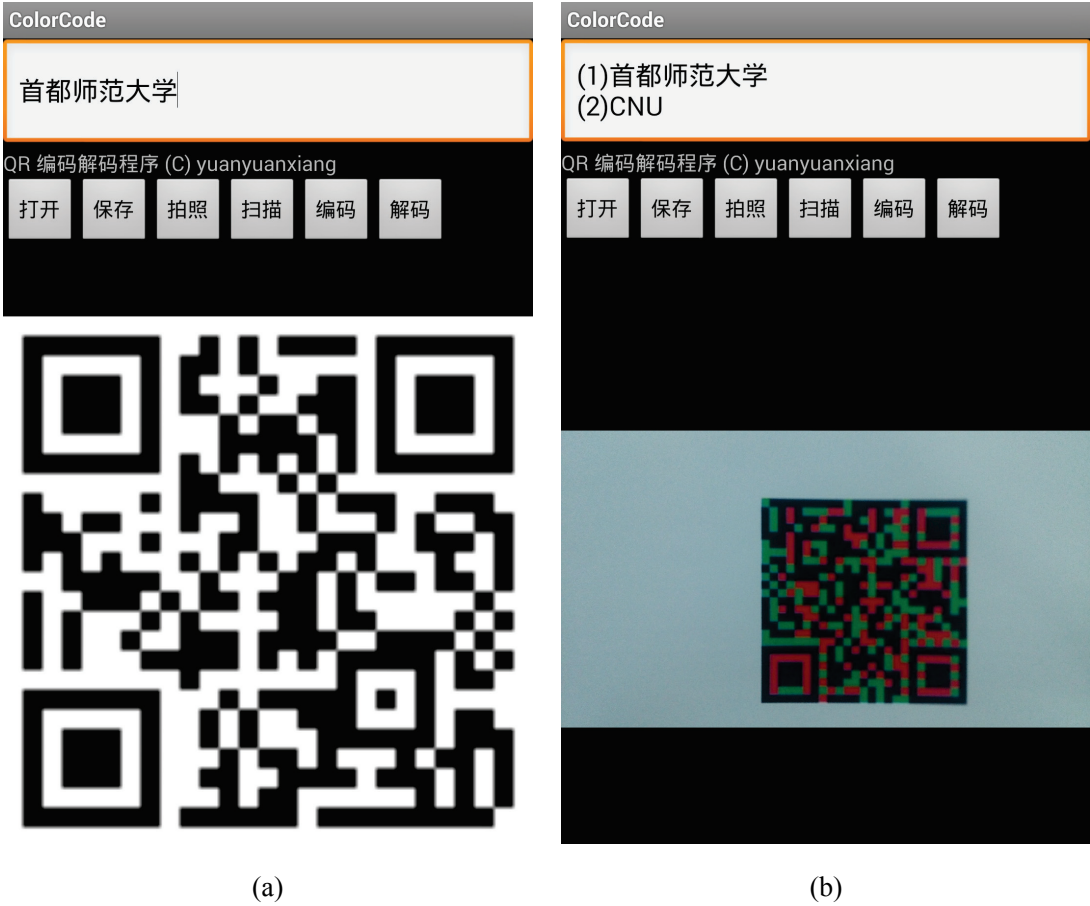


图 4-8 APP 编码解码实例

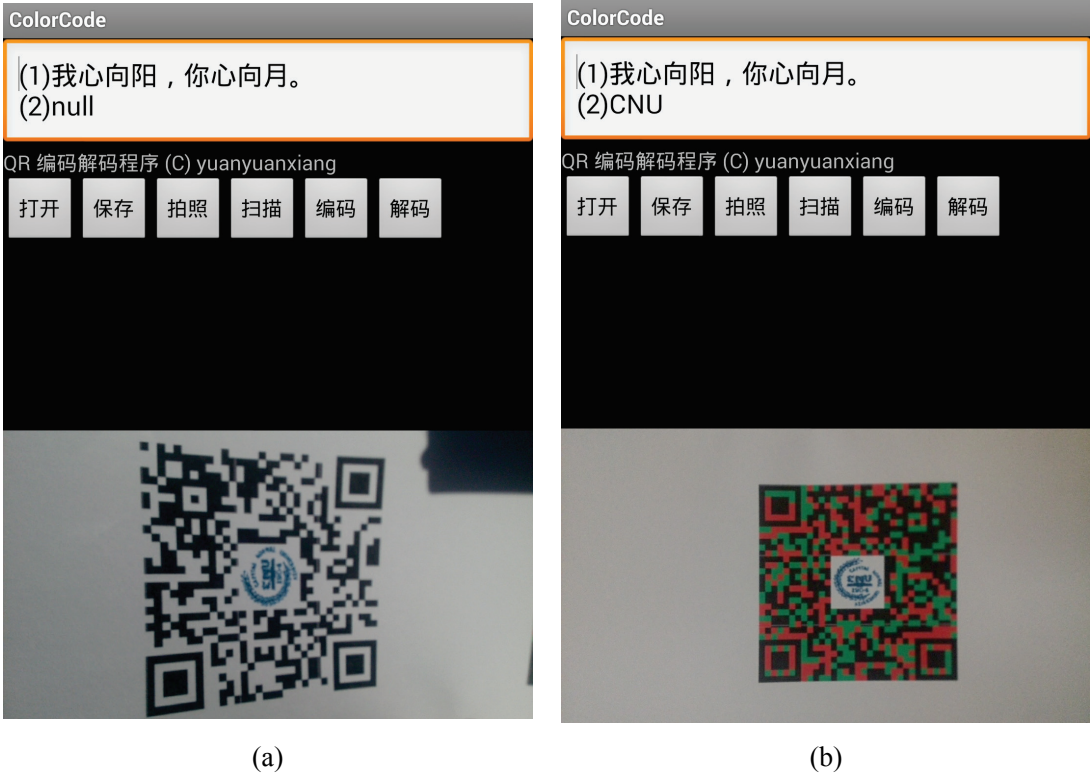


图 4-9 APP 解码实例(带 LOGO)

第五章 结论与展望

5.1 作者的其他研究

本文的创作过程，伴随着创业一起前进，作者也未能预料到今天的局面。我们认识到 QR 码是日本的技术，难免受制于人，基于 QR 码的彩色二维码存在劣势。因为 QR 二维码专利权归属于日本，所以存在专利之争的风险；国家目前正在大力推广国产软件、国产标准等，这为推广国产二维码提供了一个良好的契机。

为此，我们尝试设计了两种二维码。隐形二维码，属于初期探索阶段；融合二维码，将投入使用。

5.1.1 隐形二维码

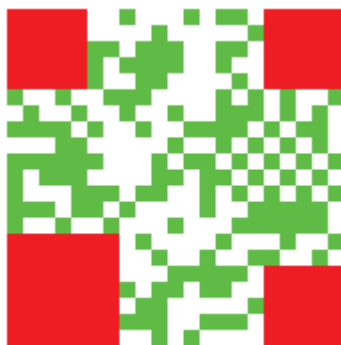


图 5-1 隐形二维码

QR 码是一种将数据编码和图像处理综合起来的技术，在对其进行研究之后，考虑到化学材料的特殊性，设计一种专用二维码，如图 5-1 隐形二维码所示是编码“首都师范大学”的二维码。此二维码包含三种颜色，红色、绿色和背景色(白色)，之所以称为隐形二维码，是因为红色、绿色只有在激光照射下才会出现，在人眼看来就是一张白纸。

将二维码和稀土上转换发光材料结合，我们创造性的开发出隐形二维码，这种二维码负载在艺术作品上不会影响艺术作品的外观，在特有检测仪器的检测下可读取，实现了防伪的功能。

5.1.2 融合二维码

融合二维码由 QR 码和国产汉信码组成(如图 5-2 QR-汉信融合码)，由于国产汉信码具有完备的知识产权，破解了 QR 码潜在的专利风险。我们创新性将 QR 码和汉信码结合起来，使消费者较容易接受，降低国产二维码推广难度。

融合二维码充分利用了 QR 码的纠错功能，在中间置入汉信码，使得粗看是个 QR 码，细看还有汉信码。当前形势下，国产二维码正面较量日本 QR 码，几乎没有成功的可能。融合码的推出，旨在从 QR 码内部突破，步步为营，先让大众对国产汉信码有个印象，再引导消费者使用汉信码，逐步赢得这场二维码战争。



图 5-2 QR-汉信融合码

消费者扫描 QR 码，得到一层信息，扫描汉信码，得到另一层信息，这两层信息是有区别的，甚至给予不同级别的权限，让汉信码能反馈更丰富的信息，在 QR 码反馈信息的页面推广使用汉信码。需要注意的是，QR 码在遭遇污染、损坏时，可能导致不能识别，这种情况下只有扫描汉信码。

5.2 结论与展望

本文首先要承认彩色二维码的编码译码程序还存在一些不足之处。首先，彩色编码先将数据格式转换为 UTF-8，未考虑数字模式、字母模式或别的模式。其次，由于时间仓促，程序未作过多的优化，彩色解码只有在条件较好的情况下才能成功。最后，由于本人能力有限，在方案设计及编程方面有许多不足的地方。

通过我们的实践，在打印机的墨盒掺入相应的隐形发光材料，打印出来的彩色二维码，在肉眼下只显示黑白两种色彩，普通解码程序能识别，但只能得到所谓的明文，彩色所隐含的信息是无法知道的。二维码甚至可以完全隐形，只有在近红外激光照射下才能显示出图形。经过设计专门的拍照装置，获取图像，再用专门设计的软件对图形解码，即可得出明文和密文。

因此，本文所设计的基于 QR 码的彩色二维码具备较大的技术可行性，可以用于防伪，具备普通二维码防伪的功能，更重要的是可以隐形，不会破坏载体的外观。已经有合作方生产出了定制的扫描设备。

最后，展望未来，我们希望二维码技术有新的创新，希望二维码在防伪领域有新的突破，也希望国产二维码占据市场主导地位。

致 谢

匆匆那年，有你也有我。在我毕业之际，我要感谢很多人。

本篇论文是首都师范大学研究生创新创业实践基地第三期立项项目“善水防伪服务”的研究内容，得到了研究生工作部、研究生院的大力资助。大众创业，万众创新，过程比结果更重要，创业使我在多方面有所收获，我首先要感谢一直坚持不懈的自己！

在首都师范大学求学的时光是美丽而充实的，在检测成像实验室生活的日子是快乐而难忘的。张朋教授学识渊博，为人和蔼可亲，处事严谨细致，把我们当成自己的孩子一样看待，给我们的关怀无微不至。张老师的讲话折射出他对人生的态度和独到的感悟，他作为长辈，我感觉出他的讲话很有道理，并受益匪浅。李宏伟老师讲话幽默，为人很好相处，培养学生的理念先进，作为我的导师，他给予我很大的自由，因为他高瞻远瞩的指导，使我才能在图像处理方向上做出一点成绩。赵星老师的每句话都很经典很有道理，每听他讲话我都会有新的领悟。

古人云：滴水之恩，当以涌泉相报。师恩重于山，师恩难报答。漫漫人生，道阻且长，我将在今后的学习、工作中，始终谨记恩师的教诲，发挥自己所学的专长，努力做出点成绩，以博恩师一笑。

还有，我可爱的师姐、可敬的师兄，他们在学业上始终是我学习的榜样，他们热心帮助我，使我避免走不少弯路。特别是陈德锋师兄，精通算法和编程，把我领入 MFC 编程的门里。大师兄张慧滔，与我们打成一片。二师兄朱溢佺，博学多才，他严谨的求学态度影响了我们，纠正了我们的马虎。赵云松师兄精通 CT 理论，非常擅长 MATLAB 编程，教会我 CT 基础知识，指导我编写算法。李孟飞师兄，在我迷茫、坠落的时候，找我谈话，给我施压，为我消散心头的云团。罗婷师姐和石昌荣师兄，常常陪我们在一起玩耍，总是给我们带来欢声笑语。最后，师弟师妹们，他们带给我无限的快乐。在此，我谨向检测成像实验室的老师们、同学们说声谢谢！

另外，我必须感谢自己的父母。他们生我养我，付出了无限劳苦、汗水及泪水。我能如愿完成自己的研究生学业，得到了他们的大力支持、鼓励及关爱。

时光如梭，岁月不居。回想在首都师范大学求学的时光，我心中有一份感激和不舍之情，感谢母校给我们提供的美好学习环境，感谢实验室给我们配置的珍贵学习资源，使我能沉下心来，安心学知识，踏实做科研。

来日方长，后会有期。祝愿老师们工作顺利、身体健康，祝愿同学们事业有成、前程似锦！

参考文献

- [1] 李俊宏, 湛邵斌. 条码技术的发展及应用[J]. 计算机与数字工程. 2009,12:115-118,154.
- [2] 李俊宏, 湛邵斌. 条码技术的发展及应用[J]. 计算机与数字工程, 2009,12:115-118+154.
- [3] 中国物联网. 二维码的概念与分类[EB/OL]. <http://www.netofthings.cn/ErWeiMa/2013-11/106.html>. 2013-11-19/2015-9-1.
- [4] 康春颖. 基于二维码技术的电子票务系统的研究与实现[D]. 东北大学, 2007.
- [5] 高峰. 二维条形码手机识别系统的设计与实现[D]. 北京邮电大学, 2008.
- [6] QRCode.com. QR 码的标准化[EB/OL]. <http://www.qrcode.com/zh/about/standards.html>. /2015-9-1.
- [7] 鲁洪军. 浅谈二维码识别在防伪系统中的应用[J]. 计算机光盘软件与应用. 2012, 22 期.
- [8] 张纪兵, 罗敏. 有机产品认证标志防伪技术现状及发展前景[J]. 认证技术. 2013.
- [9] 加拿大滑铁卢大学项目: 一种用于第四代无线通信网络的轻量级的安全系统(2013—047—多伦多—009)[J]. 电脑与电信. 2013.
- [10] 日本彩色二维码技术[J]. 电脑与电信, 2013, 03: 21.
- [11] 三维码[J]. 世界电信, 2014, 12: 8.
- [12] 彩色二维码也疯狂[J]. 标签技术, 2015, 05: 10.
- [13] 袁远松. 具有高压压缩比汉字编码能力的彩色二维条码的设计与实现[D]. 浙江工业大学, 2009.
- [14] 牛万红, 颜惠琴, 葛永斌. 一种彩色 QR 码的设计原理及编解码实现[J]. 北京理工大学学报, 2015, 10: 1067-1073.
- [15] 罗小卫. 二维码的诱惑: 2015 年市场规模将超 1 千亿[EB/OL]. <http://tech.qq.com/a/20121103/000008.htm>. 2012-11-3/2016-2-1.
- [16] 比特网. 物联网时代 二维码防伪技术随处可见[EB/OL]. http://news.xinhuanet.com/mobile/2012-12/01/c_132011976.htm. 2012-12-1/2016-2-1.
- [17] 施京京. 质检系统加强对电子商务产品质量安全监管. 中国质量技术监督[J]. 2014, 12: 14-15.
- [18] 李雅静. DataMatrix 二维条码图像识别的算法研究与实现[D]. 北京交通大学, 2009.
- [19] 胡秋玉. QRcode 识别算法的研究与改进[D]. 西南石油大学, 2014.
- [20] 张兴华. 矩阵式快速 QR 码的研究和应用[D]. 学位授予单位: 电子科技大学, 2009.
- [21] 王宗宝. rs 编/解码器的 soc 集成设计和验证[D]. 合肥工业大学. 2006.
- [22] 林萍. cmmB 智能卡系统研究[D]. 北京邮电大学. 2008.
- [23] 唐有斌. 位运算在程序设计中的应用. 计算机光盘软件与应用[J]. 贵州商业高等专科学校. 2014, 12: 265-266.
- [24] 谢锦妹. 铁路货车货物装载端部窜出检测的研究[D]. 中国铁道科学研究院. 2007.
- [25] jia20003. 图像处理之常见二值化方法汇总[EB/OL]. <http://blog.csdn.net/jia20003/article/details/8074627>. 2012-10-15/2016-3-1.
- [26] carekee. 十三种基于直方图的图像全局二值化算法原理、实现、代码及效果(转)[EB/OL].

- <http://www.cnblogs.com/carekee/articles/3643394.html>. 2014-4-3/2016-3-1.
- [27] J. M. S. Prewitt and M. L. Mendelsohn. "The analysis of cell images," *Annals of the New York Academy of Sciences*. vol. 128. pp. 1035-1053. 1966.
- [28] C. A. Glasbey. "An analysis of histogram-based thresholding algorithms," *CVGIP: Graphical Models and Image Processing*. vol. 55. pp. 532-537. 1993.
- [29] W. Tsai. "Moment-preserving thresholding: a new approach," *Comput. Vision Graphics Image Process.*. vol. 29. pp. 377-393. 1985.
- [30] Kittler. J & Illingworth. J (1986). "Minimum error thresholding". *Pattern Recognition* 19: 41-47
- [31] Ridler. TW & Calvard. S (1978). "Picture thresholding using an iterative selection method". *IEEE Transactions on Systems, Man and Cybernetics* 8: 630-632.
<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4310039>
- [32] Shanbhag. Abhijit G. (1994). "Utilization of information measure as a means of image thresholding". *Graph. Models Image Process. (Academic Press. Inc.)* 56 (5): 414-419. ISSN 1049-9652. DOI 10.1006/cgip.1994.1037
- [33] Yen J.C., Chang F.J., and Chang S. (1995) "A New Criterion for Automatic Multilevel Thresholding" *IEEE Trans. on Image Processing*. 4(3): 370-378
- [34] Sezgin M. and Sankur B. (2004) "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation" *Journal of Electronic Imaging*. 13(1): 146-165
- [35] mihenyinhua . ZXing 中 QR 码相关部分了解[EB/OL].
<http://blog.csdn.net/mihenyinhua/article/details/17224019>. 2013-12-9/2016-4-9
- [36] 孙汝杰 张宇光. 基于时间序列的个性化信息协同过滤技术研究[J]. *情报杂志*. 2006.
- [37] 张继尧. 张焜. 刘晓. 遥感图像自适应去噪方法研究[J]. *大气与环境光学学报*. 2011.
- [38] CN-GB.快速响应矩阵码[S].2000.
- [39] 卢钺. QR 码识别方法研究及应用[D]. 南京理工大学. 2013.
- [40] Devin Zhang. Android 之 NDK 开发[EB/OL].
<http://www.cnblogs.com/devinzhang/archive/2012/02/29/2373729.html>. 2012-2-29/2016-3-1.
- [41] 侯俊杰. 深入浅出 MFC(第二版)[M]. 武汉: 华中科技大学出版社, 2001-1:339-406.
- [42] yuanlong_zheng. VC 计划之二 MFC 文档/视图体系结构(1)[EB/OL].
http://blog.sina.com.cn/s/blog_62ce00c90100ld9h.html. 2010-9-17/2016-3-1.
- [43] 长平狐. NDK Android.mk 手册[EB/OL]. http://www.oschina.net/question/565065_93974.
2012-4-8/2016-3-1.
- [44] 长平狐. NDK Application.mk 使用手册[EB/OL]. http://www.oschina.net/question/565065_93983.
2012-4-6/2016-3-1.