中山大学数据科学与计算机学院

物联网技术实验报告

二维码解码模块的设计与实现

小组成员: 陆\*\* 14353\*\*

指导教师： 胡建国

目 录

[第1章 二维码解码模块的设计概要 1](#_Toc468634353)

[1.1 二维码解码模块的设计概述 1](#_Toc468634354)

[1.2 二维码解码模块的设计安排 1](#_Toc468634355)

[第2章 二维码解码模块的详细设计过程 4](#_Toc468634356)

[2.1 OSTU二值化 4](#_Toc468634357)

[2.2 二维码截取 5](#_Toc468634358)

[2.3 Hough直线变换 6](#_Toc468634359)

[2.4 双线性变换 8](#_Toc468634360)

[2.5 临近插值 9](#_Toc468634361)

[2.6 码字提取 9](#_Toc468634362)

[2.7 译码 10](#_Toc468634363)

[第3章 二维码解码模块的功能测试 11](#_Toc468634364)

[3.1 OSTU二值化 11](#_Toc468634365)

[3.2二维码截取 12](#_Toc468634366)

[3.3Hough直线变换 13](#_Toc468634367)

[3.4双线性变换 14](#_Toc468634368)

[3.5临近插值 15](#_Toc468634369)

[3.6 码字提取 15](#_Toc468634370)

[3.7 译码 17](#_Toc468634371)

[第4章 二维码解码模块的功能创新 18](#_Toc468634372)

[4.1 基于连通域查找表的边界连通域去除方法 18](#_Toc468634373)

[4.2 基于积分图的局部自适应二值化 19](#_Toc468634374)

[第5章 二维码解码模块的项目总结 22](#_Toc468634375)

# 第1章 二维码解码模块的设计概要

## 1.1 二维码解码模块的设计概述

条码作为一种信息存储和传播的手段，广泛地应用于社会生产和生活的各个领域。物联网技术的不断发展，给条码技术的发展带来了新的契机。传统的一维条码由于信息容量限制已经不能满足应用需求，因此各种二维码应运而生。其中pdf417条码以信息容量大、编码信息范围广、译码可靠性高等优点在业界得到广泛应用。

对于pdf417条码的识别国外工业级别的软件已经达到了较高水平，但大多数受专利保护，而国内的二维码识别技术研究起步较晚，较不成熟。并且市面上的大多数pdf417条码识读器或是受技术限制或是出于实时性的需求，只能处理理想条件下的条码识别。对于各种复杂情况，如噪声、扭曲、低对比度等条件下的pdf417识别还是有很多研究价值。因此本文对复杂情况下的条码定位识别问题做了进一步研究。

## 1.2 二维码解码模块的设计安排

整个课程分为两段：前半段主要涉及图像处理，实现对图像中二维码的识别定位；后半段根据pdf417的编码规则完成对二维的解码。

**1、前半段（5-8周）：**

第五周：1）OSTU二值化算法：假定图像包含两类像素---前景像素和背景像素，计算能将两类分开的最佳阈值，使得类内方差最小，类间方差最大；2）线扫描算法：当图像只存在选择失真且图片中的干扰较小时通过扫描pdf417的左边线得到图形的旋转角度；3）旋转变换：。4）临近插值：如果某一点的与其水平或垂直方向上的临近点的灰度值不同，则把该点的灰度值重新赋值为其临近点的值。

第六周：1）形态学的两个基本操作膨胀和腐蚀。用结构元素B膨胀图像A的结果就是把结构元素B平移后使B与A的交集非空的点构成的集合。腐蚀的结果就是把结构元素B平移后使B包含于A的所有点构成的集合。2）基于膨胀操作的形态学重构，可用于去除图像中与边界相连的连通域；3）基于matlab自带的bwlabel函数建立连通域查找表，用于去除小面积连通域，实现对图像中二维码的识别；4）投影定位，通过统计每一行和每一列上的前景像素的数量来确定二维码所在的区域，实现对二维码的截取。

第七周：霍夫（hough）直线变换，用于检测构成二维码边框的四条直线的方程，从而确定二维码的四个顶点。 用累加矩阵以一定的精度(1deg)统计经过每个前景像素的每条直线的出现次数，取出四条局部出现次数最多的直线作为边框直线的方程。

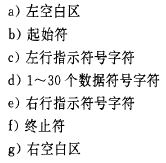
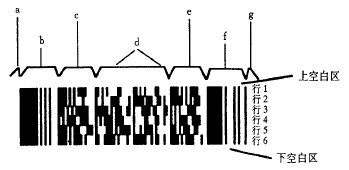
第八周：几何变换。1）仿射变换：

2）双线性变换：，。

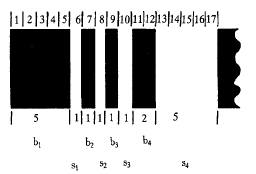
**2、后半段 （12-13周）：**

第十二周：码字提取。

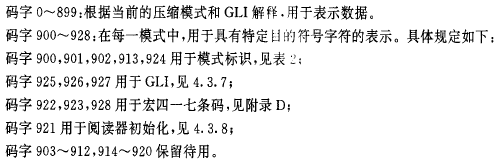
1）PDF417的符号结构：417条码符号为多行结构，上下为空白区，每行数据符号字符数相同，最小行数为3最大行数为90。每行构成如下图：



2）符号字符的结构：每个符号字符有4个条（黑）和4个空（白）构成，自左向右开始。每个条或空包含1~6个模块，并且一个符号字符中，4个条和4个空的总模块数为17，如下图：



3）码字集：码字集包含929个码字（0~928）：

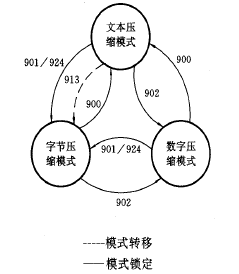


4）符号字符的簇：由3个簇（0、3、6）构成，每个簇包括以不同的条空形式表示的所有929个码字，并且每一簇中每个符号字符对应唯一的码字。同一行使用一个簇号，对于特定一行：簇号=[(行号-1)mod 3]\*3, 行号>=1。

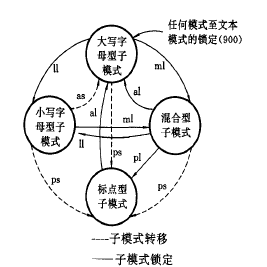
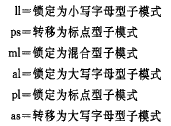
第十三周：译码。

1）三种数据压缩模式：文本压缩模式（TC）、字节压缩模式（BC）、数字压缩模式（NC）。

2）模式锁定与转移码字：模式锁定码字用于将当前模式切换到指定的目标模式，改薄至切换在下一个切换前一直有效。模式转移码字用于将文本压缩模式暂时切换为字节压缩模式，仅对切换后的第一个码字有效。个模式之间的切换关系如下图所示：



3）文本压缩的3个子模式及切换：文本压缩模式下一个码字表示一个字符对（H、L）：码字=30\*H+L。3个子模式：大写字母型子模式、小写字母型子模式、混合型子模式、标点型子模式。子模式之间的切换：



4）数字压缩模式：编码：先将数字分成44位为一组，然后在数字前面补1，再进行10至900基的转换。译码：将每15个码字从左到右分为1组，对于每一组码字先执行900基至10基的转换，再去掉最高位的1。

5）字节压缩模式：字节压缩模式通过256基至900基的转换将字节序列转换为码字序列。对于字节压缩模式有两个模式锁定：901和924，当索要表示的字节总数不是6的倍数时要模式锁定901，当索要表示的字节总数为6的倍数时要用模式锁定924。模式转移913用于从文本模式到字节模式的暂时性转移。

# 第2章 二维码解码模块的详细设计过程

## 2.1 OSTU二值化

OSTU算法假设图像只包含前景像素和背景像素，计算能将两类像素分开的最佳阈值，使得类内方差最小类间方差最大。

思路：假设整幅图像中前景像素和背景像素所占比例和期望值分别为w0,u0和w1，u1。那么整幅图像的期望值为u=w0\*u0+w1\*u1。当分割的像素点为t时，前景像素和背景像素的类间方差为。求使得g(t)最大的t。由于w1=1—w0,所以u1=(u-w0\*u0)/(1-w0)，所以g(t)简化为。

流程：

1）计算图像的灰度直方图，求得整幅图像的均值u；

2）找到直方图中的起止灰度值st,nd；

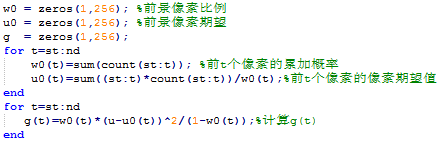
3）对于每个t (st<=t<=nd)计算相应的g(t)，并用一个数组g记录；

4）数组g中峰值所在坐标即为所求的最佳阈值

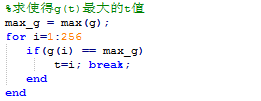
5）根据求得的阈值对图像进行二值化，低于阈值的判为0，高于阈值的判为1。

关键代码：

g(t)的计算：



阈值的选取



## 2.2 二维码截取

功能：将二维码从有复杂背景的图片中截取出来。

思路&流程：

1）先用一个合适的结构元素对输入的图像进行一定次数的闭运算增强二维码内部的连通性。

1）利用重构去除与边界相连的连通域。对图像的边界进行膨胀再与原图做与操作重复一定次数后便可重构出于边界相连的连通域。再用原图减去重构出来的边界得到去除与边界相连的连通域后的图像。

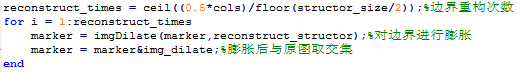
2）建立查找表去除小面积的连通域。函数[L,num]=bwlabel(src)用于对二值图像src中的连通域进行编号，L是一个记录原图各点所属连通域的编号的矩阵，num是连通域的数量，即L中的最大值。统计L中各个连通域的面积。把最大连通域面积的一般作为阈值，去掉面积小于该值的连通域。

3）投影定位确定二维码所在区域。统计每一行上前景像素的数量作为水平方向上的投影Sy，统计每一列上前景像素的数量作为垂直方向上的投影Sx。用公式S(k)=0.7\*S(k)+0.3S(k+1)分别随Sx和Sy进行平滑。把各个方向上的投影值的均值作为阈值，找到各个方向上大于阈值的区间，即为二维码所在区域。

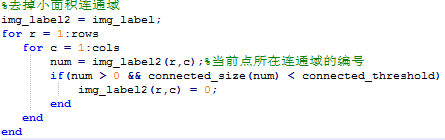
4）截取出二维码所在区域。

关键代码：

重构：



去除小面积连通域：



## 2.3 Hough直线变换

功能：通过对二维码的边框进行hough变换得到二维码四条边界方程进而求得二维码的四个顶点坐标。

思路：

1）提取边框：先用一个较大的结构元素对二维码进行闭运算，将二维码揉成团得到图像a。再用一个较小的结构元素膨胀a得到b。用b减去a得到二维码的边框。

2）对二维码边框进行hough变换：根据极坐标转换公式x=rho\*cos(theta),y=rho\*sin(theta)，将笛卡尔坐标系下的直线方程转换为极坐标的形式：rho=x\*cos(theta)+y\*sin(theta);根据这一公式建立一个累加矩阵H[2\*rho\_max+1][theta\_max]。然后以一定精度统计经过每个前景像素点的每条直线的参数(rho,theta)，即对于每个前景像素点(x,y)求每个theta取值下的rho，并记录到累加矩阵H中。

3）峰值搜索：先对H矩阵进行扩展。用一个4\*3的矩阵lines记录四条直线的参数（*lines(:,1)为在H中对应点的值; lines(:,2)为rho; lines(:,3)为theta*）。 遍历H找到最大值所在的坐标记录到lines中，并将此坐标-10到10领域的值全部置为0，同样的方法寻找其他三点（*前两个峰在1到90度的范围内找，后两个峰在91度到180度的范围内找*）。

4）找直线交点：用一个4\*4的矩阵记录四个交点（*points(:,1) 为到原点的距离，points(:,2)为x坐标，points(:,3) 为y坐标，points(:,4)为点的序号，用于确定点的相对位置*）。将lines中的不平行的直线的参数两两带入矩阵方程求得交点坐标。

流程：

1. 根据图像的大小row，col求得rho\_max=sqrt(row^2+col^2)
2. 建立累计矩阵H[rho\_max][theta\_max]
3. 对于每个前景像素点(x,y)以及给定的theta时，根据直线方程求出对于的rho值
4. 对应的累加器H[rho][theta]加1
5. 在1到90度的范围内搜索前两个峰值并记录到lines矩阵中
6. 在91到180度的范围内搜索后两个峰值并记录到lines矩阵中
7. 求交点坐标：

*for i=1:3*

*for j=i+1:4*

*if(lines(i)不平行于lines(j))*

*求他们的交点并记录*

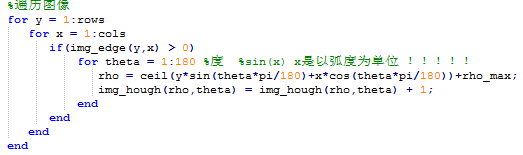
*end*

*end*

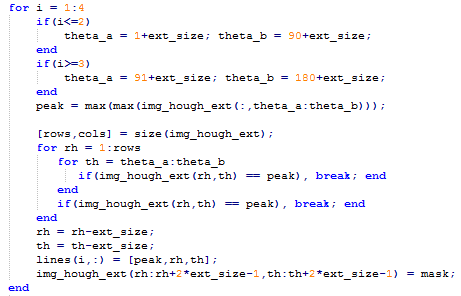
*end*

关键代码：

建立累加矩阵:



峰值搜索:



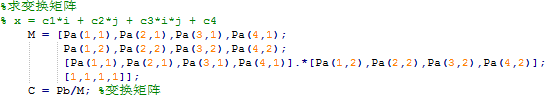
## 2.4 双线性变换

思路&流程：根据双线性变换公式： （(i,j)为变换前的坐标，(x,y)为变换后的坐标）

由于变换矩阵是2\*4矩阵有8个参数，因此需要四组对应点。

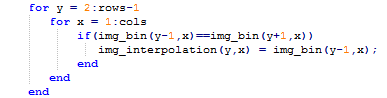
令(x1,y1),(x2,y2),(x3,y3),(x4,y4)为变换后二维码的四个顶点的坐标，(i1,j1),(i2,j2),(i3,j3),(i4,j4)为变换前二维码四个顶点的坐标，带入变换方程得到：

对上式运用矩阵除法求得变换矩阵。再用该矩阵作用于图像中的每个前景像素求得每个前景像素变换后的坐标。



## 2.5 临近插值

用于填补变换形成的空洞。插值规则：如果某一点与其水平或垂直方向上的临近点的灰度值不同，则把该点的灰度值重新赋值为其临近点的值。



## 2.6 码字提取

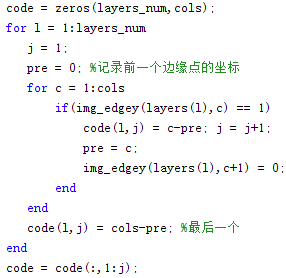
功能：从二维码中提取出码字。

思路&流程：

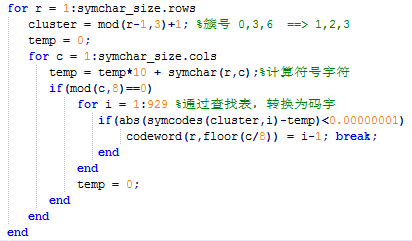
1. 提取图像边缘。
2. 消除垂直边缘，做水平投影，通过搜索峰值得到二维码每层的所在的行数。
3. 消除水平边缘，根据上一步中得到的每层所在的行数，计算每层上各个条和空的宽度。
4. 将第一个条的宽度除以8得到一个模块的大小。
5. 将3）中所得的各个条和空的宽度除以模块的大小得到每个条和空所占的模块数。
6. 将每8个模块数根据公式转换成一个符号字符。
7. 遍历PDF417符号转换表，在其中找出temp所对应的码字，用一个decode数组保存。

关键代码：

计算每个条和空的宽度：



解码码字转换：



## 2.7 译码

功能：将码字解码成字符串

思路：先将码字序列根据模式锁定符切分，将子序列放到相应的子模块中解码。

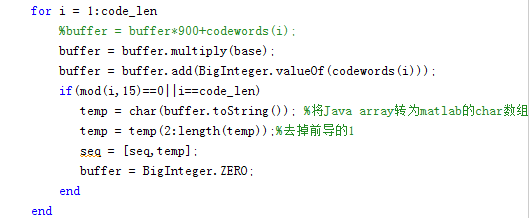
流程：

数字压缩模式解码：将序列按每15个码字分组，每个组进行10基到900基的转换，再将前导的1去掉。

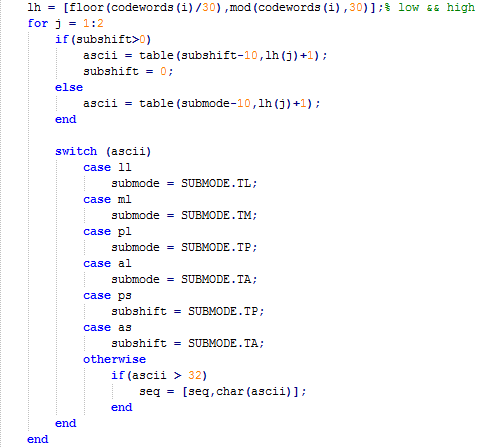
文本压缩模式解码：根据码字表建立各个子模式下对应的字符表。根据公式：码字=30\*H+L 将码字分解成字符对，对于每个字符遍历对应模式下的字符表查找出相应的ascii值，如果是子模式转换符则设置相应的模式，否则通过char函数转换为字符。

关键代码：

数字解码：



文本解码：



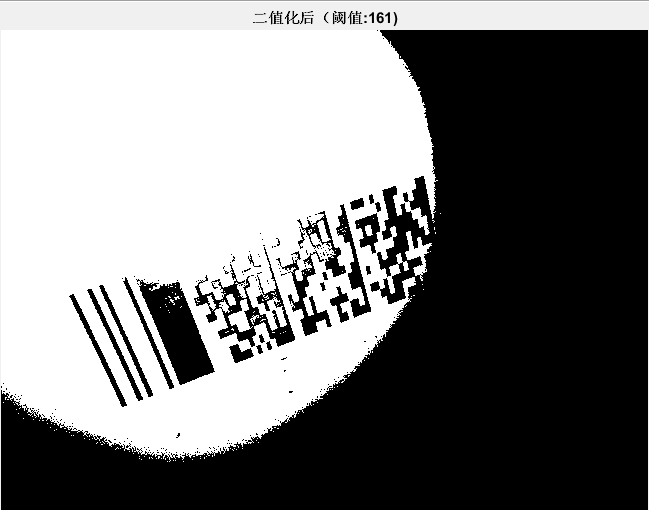
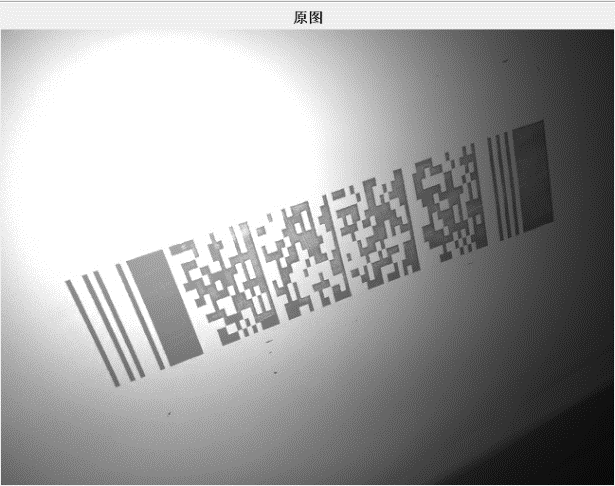
# 第3章 二维码解码模块的功能测试

## 3.1 OSTU二值化

1）当所处理图像光照均匀时，分割效果较好



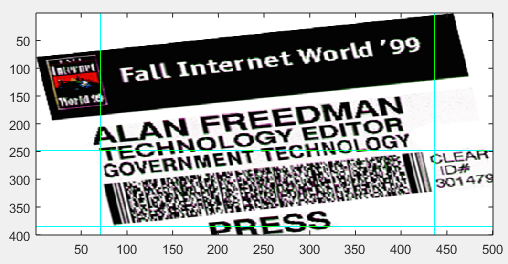
2）当处理的图像存在高照不均匀时，则ostu处理效果不是很好



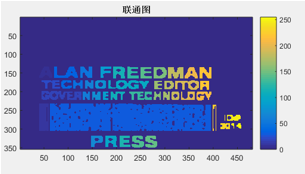
可以看到，强光提高了整幅图的均值，使得计算出来的阈值也随之提高，导致光线较暗部分被置成了黑色。

## 3.2二维码截取

1）当二维码与周围的距离较大，且在去掉与边界相连的连通域后二维码是最大的连通域时效果较好。

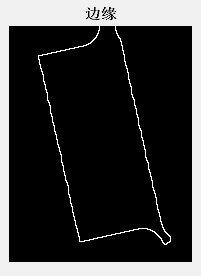
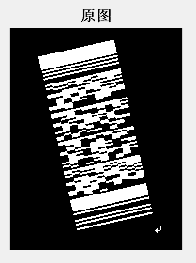


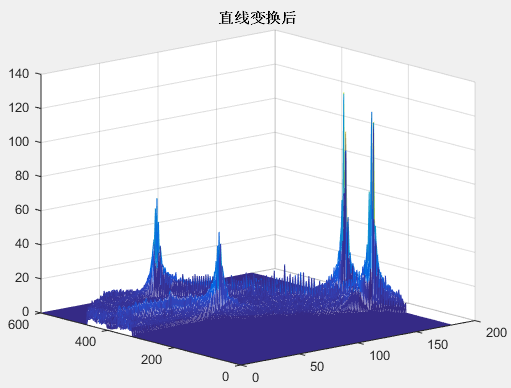
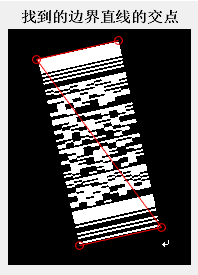
2）如果预处理的时候膨胀次数不够，没能将与二维码主体较远的定位条与主体相连，则定位条会被当成小面积连通域去除。



## 3.3Hough直线变换

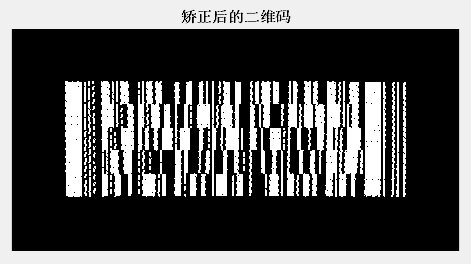
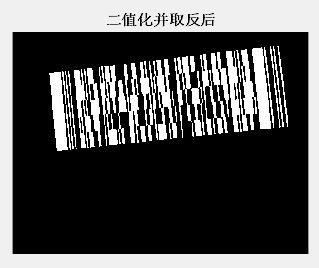
在二维码扭曲不是特别厉害的情况下都可以正常提取到四个顶点。



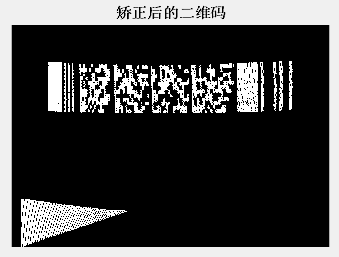
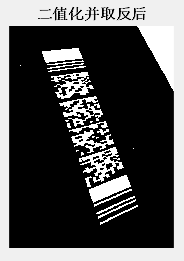


## 3.4双线性变换

在Hough模块获取到的顶点正确的情况下都能够正确的将图片矫正

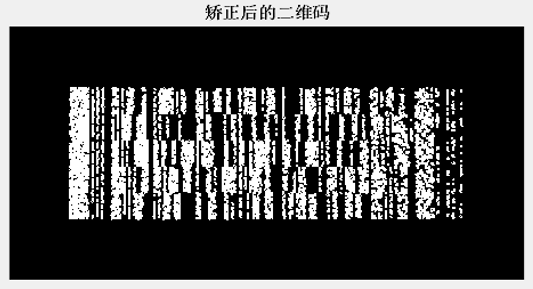
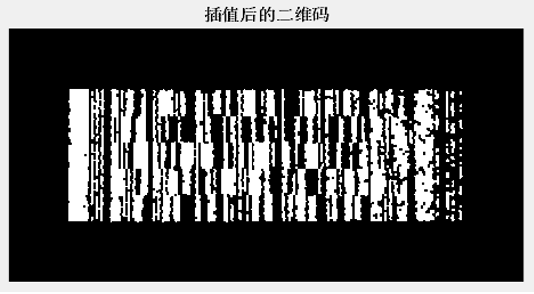


算法默认将离原点最近的点作为矫正后二维码的左上角，离左上角最近的点作为左下角，因此在如下图的情况中会出现矫正后的二维码上下颠倒的情况。



## 3.5临近插值

因为算法在考察某一点的临近点的时候只是考察了该点一个方向上与其**相邻**的点，所以在图片内部的空隙较小的情况下效果较好，并且只能在一个方向上填补空隙。

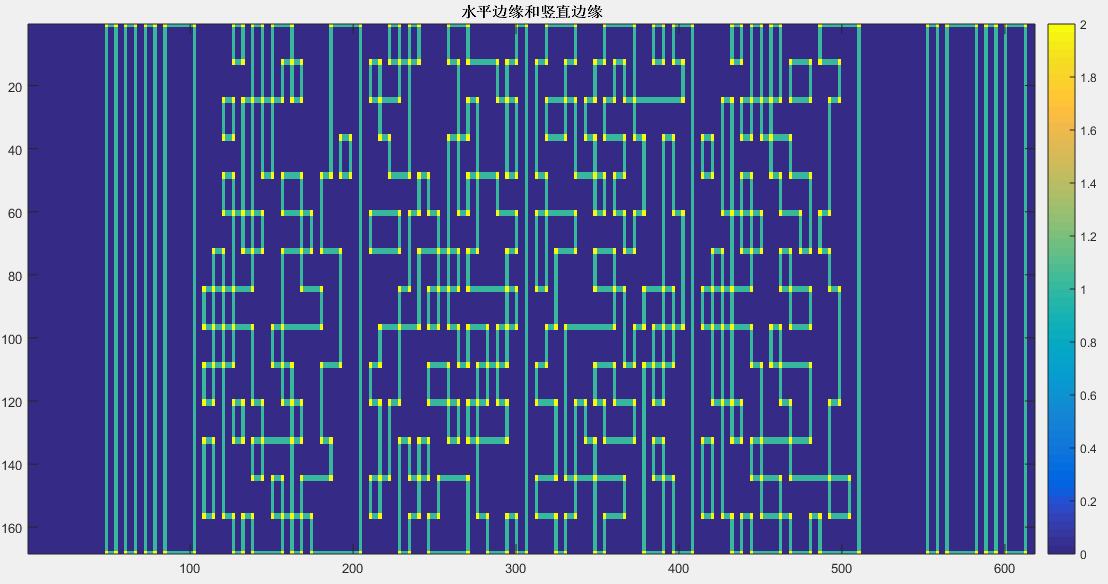
 

上图为只考察竖直方向上的临近点时候的差值效果，可以看到，二维码内部较小的空隙都被填补，但是较大点的空隙无法填补，并且在边缘上的差值效果不是很好。

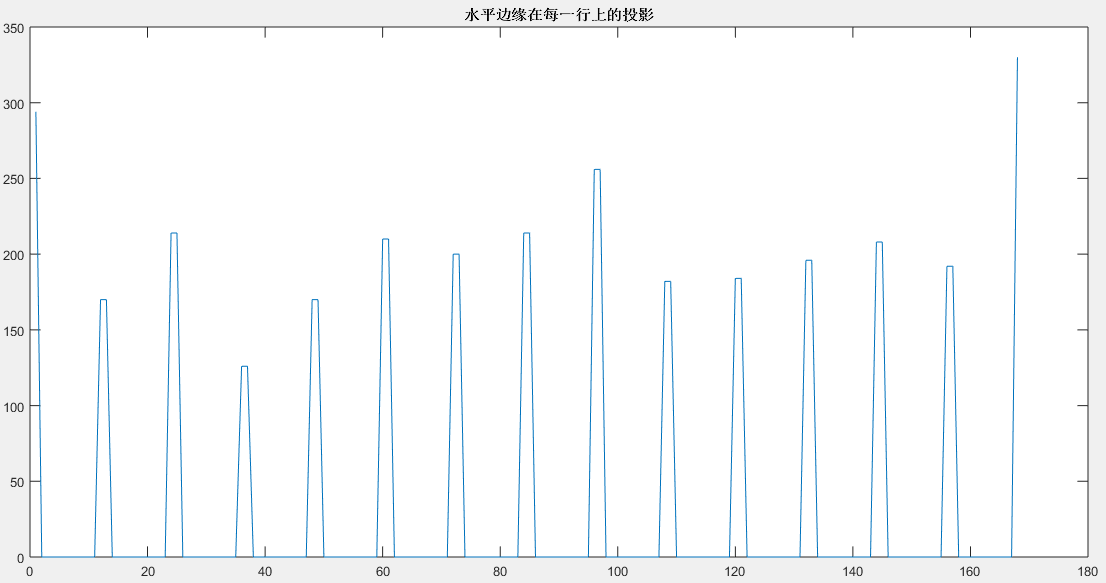
## 3.6 码字提取

test2.bmp:

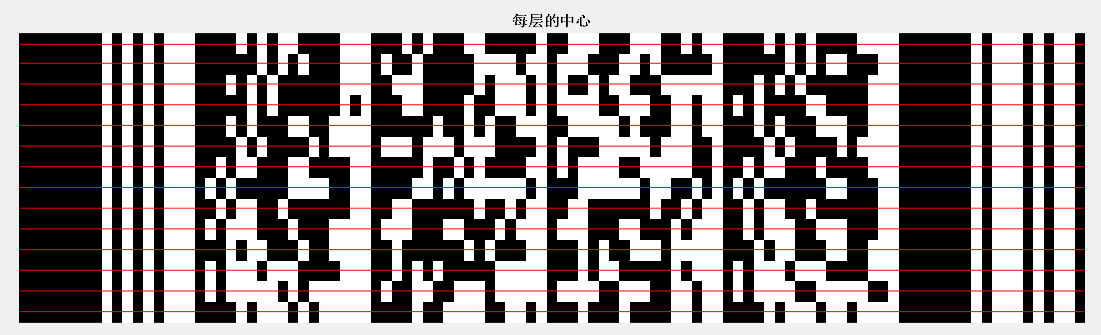
水平边缘与竖直边缘相加的效果图：



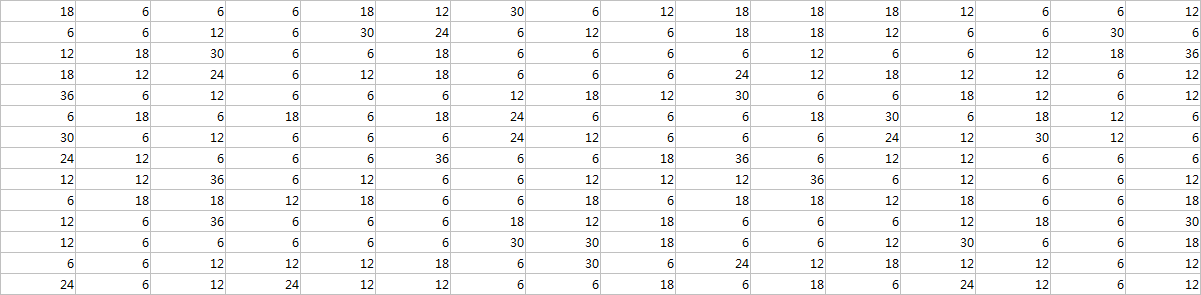
对水平边缘做竖直投影得到每层的起始坐标：



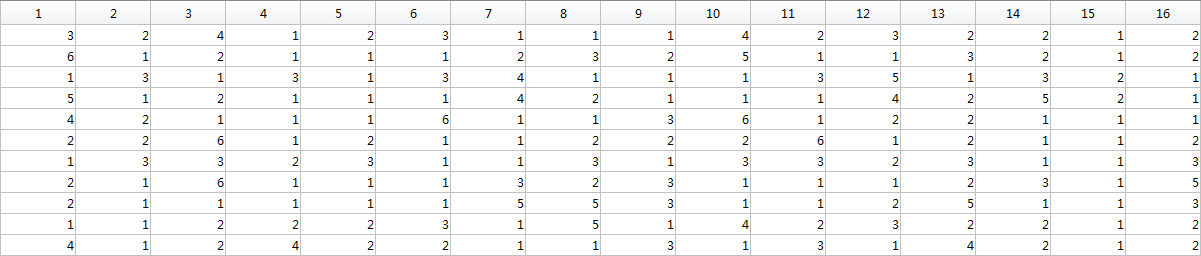
根据每层的起始坐标，计算出每层的中心所在的行数：



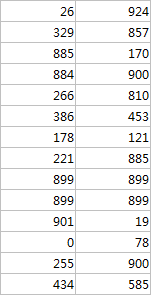
每个符号字符的条空的宽度：



除以一个模块的宽度6后得到的符号字符：

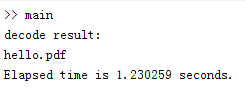


根据符号字符-码字表转换成码字序列：

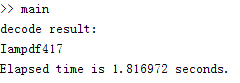


## 3.7 译码

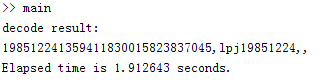
test1解码结果：



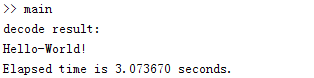
test2解码结果：



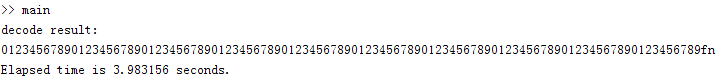
test3解码结果：



test4解码结果：



test5解码结果：



由于前3个测试为bmp格式不需要经过二值化，而后2张为png格式解码前需要先经过二值化，因此后两张解码总时间较前3张长。

# 第4章 二维码解码模块的功能创新

## 4.1 基于连通域查找表的边界连通域去除方法

原先在二维码截取的时候去除与边界相连的连通域是通过将边界作为marker不断地进行重构，再用原图减去重构后的marker。考虑到每次对边界的重构都需要将整幅图像遍历一遍，并且在结构元素的大小为(2n+1)\*(2n+1)的情况下，每次边界只能拓展n个像素点，因此在结构元素较小的时候需要进行很多次的重构，因此对时间的消耗会很大。

针对以上情况，结合去除小面积连通域所用的方法，提出本方法。

思路：先将图像进行膨胀增强二维码内部的连通性，再为其建立查找表，通过遍历图像边界确定出与边界相连的连通域，并将该连通域的编号置为0。

流程：

1）对图像进行膨胀

2）建立查找表 [L,num]=bwlabel(src,8)

3）将marker赋为图像的边界

4）

*for marker中的每个前景像素(r,c)*

*if(L(r,c)>0)*

*for(L中的每个点(r1,c1))*

*if(L(r1,c1)==L(r,c))*

*L(r1,c1)=0*

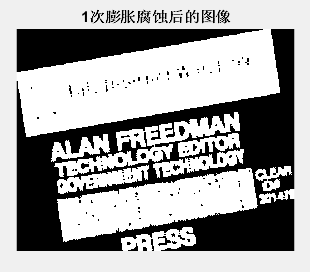
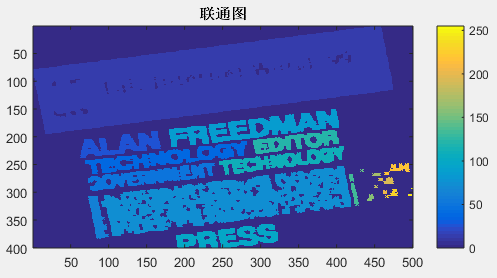
*end*

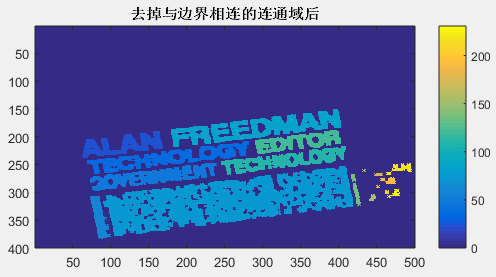
*end*

*end*

*end*

效果：

左侧为用重构方法去除效果， 右侧为用查找表方法去除的效果。可以看到，二者的差别不是很大。因为对于一张8连通的连通图，其与边界相连的连通域与用一个3\*3的全为1的结构元素对边界进行重构所得到的连通域是一样的。

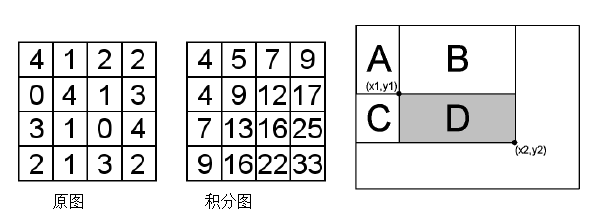
## 4.2 基于积分图的局部自适应二值化

因为OSTU二值化是基于全局阈值的二值化，所以只适用于光照均匀的图像，当光照不均匀时，光照强的部分会将整幅图像的阈值提高，导致光照较暗部分的前景像素被错误地当成了背景像素处理。

针对OSTU的以上不足，提出局部自适应阈值二值化方法。每个像素点的二值化阈值都是根据其一定邻域内的像素的均值计算得到。这样每个像素的二值化阈值就可以随着邻域的亮度的变化而变化，从而减少了前景像素被误判的可能。

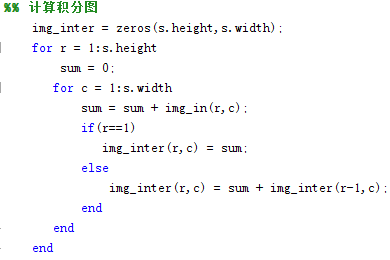
思路：通过图像的积分图计算图像的局部均值：  
每个点(x1,y1)的值为原图像中所有位于(x1,y1)左侧和上面像素的和。这样原图像中区域(x1<x<x2,y1<y<y2)的所有像素的和即可表示为：

如下图所示：

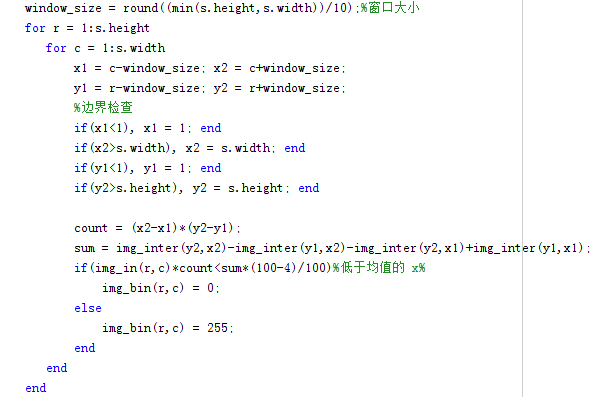


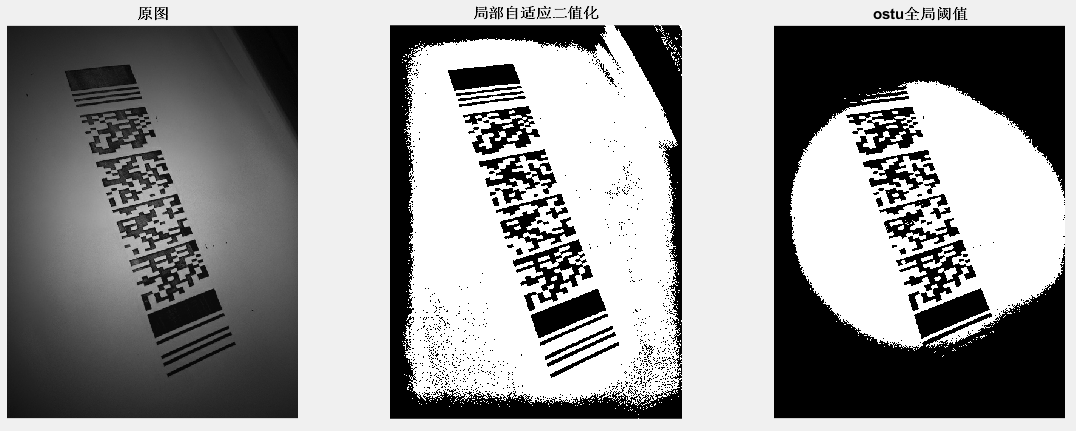
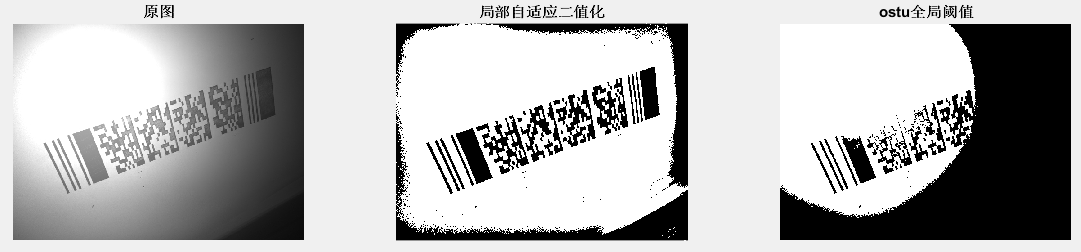
关键代码：

计算积分图：



二值化：



测试结果：

可以看到局部自适应二值化的效果明显优于全局阈值。另外局部自适应只需要对图像进行两次遍历即可完成，并且基于积分图的阈值计算方法使得局部阈值的计算可以在常数时间内完成。

# 第5章 二维码解码模块的项目总结

Pdf317作为一种应用最为广泛的二维码之一，具有信息容量大、编码信息范围广、译码可靠性高等诸多有点。本文对条码的识别和解码进行了研究，整个工作分为条码的识别和条码的解码两段。

对条码的识别工作研究了在复杂背景下条码的提取方法以及条码的矫正方法。提出了基于连通域的二维码提取方法和通过双线变换对二维码进行校正的方法。条码的解码工作中学习了pdf417条码的结构、编码规则等。实现了通过边缘检测提取码字的方法，以及文本压缩模式和数字压缩模式下的解码模块。