

# DataMatrix 条码的畸变校正

## The Correction of DataMatrix Barcode Distortion

袁远松 赵小敏 杨东勇 (浙江工业大学 软件学院 浙江 杭州 310014)

**摘要:** 条码在使用过程中出现的污染、缺损、仿射变换和边缘模糊等畸变给条码识别带来困难。为提高条码码字的切割和识别的准确度,本文提出了一种矩阵式二维条码的畸变校正算法,采用凸包运算精确定位条码图像的边界和四个顶点,并采用控制点法校正条码图像。实验表明,该方法简单、易于实现并且效果好。

**关键词:** 二维条码 DataMatrix 连通区域 条码识别 校正

### 1 概述

条码技术已经成为当今主要的计算机自动识别技术之一,是信息数据自动识别、输入的一种重要方法和手段,现被广泛应用于工业、商业、交通运输、邮电及办公室自动化等各个领域。目前,条码分为一维条码和二维条码。二维条码的密度远大于一维条码,识别难度也相对较高。条码在使用过程中出现的污染、缺损等因素将导致图像的缺陷,在条码有边缘部分存在半像素问题,导致边缘模糊,图像在拍摄时会出现仿射变换。这些因素加大了条码识别的难度<sup>[1]</sup>。图1是一张实际应用中需要识别的 DataMatrix 条码图像,它存在着上述的畸变。如何快速准确地识别条码一直是条码技术研究的热点问题,条码图像的畸变校正整个条码识别过程中最重要的环节之一。

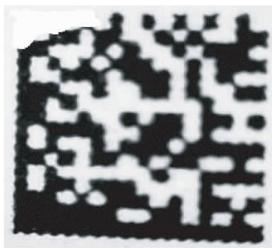


图1 DataMatrix 条码图像

传统的 Datamatrix 图像校正方法是先通过 Hough 变换检测出 DataMatrix 定位图形“L”型边界,然后再进行图像校正<sup>[2]</sup>。由于 Hough 变换是全局检测算法,计算量非常大。条码在流通过程中往往会出现锯齿状等

边缘模糊,这也加大了 Hough 变换检测难度,并且降低了检测的准确度。

本文针对 DataMatrix 条码图像,研究了一种矩阵式二维条码的畸变校正算法。该算法先对图像二值化处理,接着通过连通区域标记的方法初定位整个条码区域,然后利用最小凸多边形精确定位条码边界和条码图像的四个顶点,最后根据条码的四个顶点采用控制点法校正图像。

### 2 DataMatrix 结构简介

Datamatrix 是矩阵式二维条形码的一种,它以一个矩形图案表示数据信息。通常,黑色模块表示“1”,白色模块表示“0”,或者使用相反的表达方法。另外,在实际应用中,可以采用不同的印制方式。

DataMatrix 图形可分为两个部分:定位图形(寻边图形)和数据区。定位图形由两条实线边组成的“L”型和与其相对的两条虚线边组成;数据区是由1、0模块组成的矩形。由于 DataMatrix 定位图形的特性,将其与背景图形区别开来,并对“L”形的顶点进行精确定

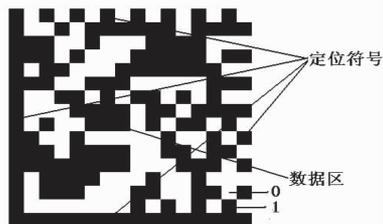


图2 DataMatrix 条码及其结构示意图

位。图 2 为 DataMatrix 条码及其结构示意图<sup>[3]</sup>。

### 3 条码图像校正原理

条码图像的校正,可采用控制点法把失真图像与校正后图像建立连接点控制。控制点法<sup>[4]</sup>是通过测定若干特定坐标点来确定坐标变换方程的系数。输入失真图像  $f(u, v)$  从  $uv$  失真图像坐标系变换到  $xy$  校正后图像坐标系上数字输出校正后图像  $g(x, y)$  并满足:

$$f(u, v) = g(x, y) = [f(p(x, y), q(x, y))] \quad (1)$$

其中,数字图像上每个像素的坐标均为整数  $(u, v)$  坐标与  $(x, y)$  坐标之间的变换函数关系为:

$$u = p(x, y), v = q(x, y) \quad (2)$$

若已知失真条码图像与校正后图像上四对应点对的位移量,可求解下列坐标变换方程的系数  $a \sim h$ 。

$$u = p(x, y) = ax + by + cxy + d \quad (3)$$

$$v = q(x, y) = ex + fy + gxy + h \quad (4)$$

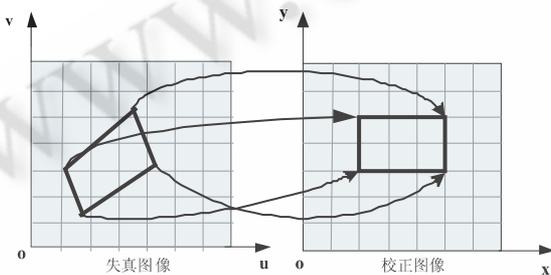


图 3 失真图像与校正图像“控制点”

如何准确地找出失真条码图像的四个顶点是条码图像运用控制点法校正的关键。根据 DataMatrix 条码图形结构观察 DataMatrix 条码的黑色模块主要集中在几个较大的连通区域中,很少有孤立的黑色模块存在。即使有孤立的模块,在条码校正时可忽略,只要提取较大的连通区域即可,整个条码区域主要集中在条码定位图形周围。因此,对于整个条码边界的初步提取可采用连通区域标记算法。采用连通区域标记算法还可以去除条码图像中出现的一些图像小噪声。

在实际应用中条码往往会出现缺损或齿轮状,加大了精确定位条码的四顶点的难度。通过凸包运算,可以求出整个图像的最小凸多边形,由于条码齿轮状出现,有可能出现少许不是条码边界的顶点,这里称为“伪顶点”,与“伪顶点”相邻的边称为“伪边”。与“伪

顶点”相邻的两条“伪边”的夹角通常较小或者边长较短。根据“伪边”和“伪顶点”的特征,为凸多边形的相邻边的夹角和边长分别指定一个阈值  $\beta$  和  $b$ ,把夹角小于阈值  $\beta$  的相邻边连成直线并且去除长度小于  $b$  的边。去除“伪顶点”和“伪边”,可以有效地消除条码齿轮状对图像校正的影响。条码图像边界有可能出现部分缺损,运用凸包运算求出的最小凸多边形的顶点并不能确定是条码的四个顶点,凸多边形最长的四边相互相交的交点就是条码图像的两个顶点。在 DataMatrix 条码识别过程中,查找定位图形“L”型对于码字切割至关重要,通过定位图形可以确定每小块 DataMatrix 图形的排列顺序和大小。通过计算条码图像的每个顶点的 Rating 值可以准确地确定条码的定位图形,即条码的“L”型边界,Rating 值为该点与它相邻两点组成的两条直线在二值化后的条码图像中所占黑点数的乘积。Rating 值最大的一个点为“L”型的交点,Rating 值较大的三个点为“L”型黑边的顶点。

### 4 条码图像校正实现

对于一幅出现污染、缺损、仿射变换和边缘模糊等畸变的非标准的 DataMatrix 条码图像的校正,本文设计了条码校正的实现流程,主要包括预处理、初定位、精确定位和图像校正,整个条码图像校正流程如图 4 所示。

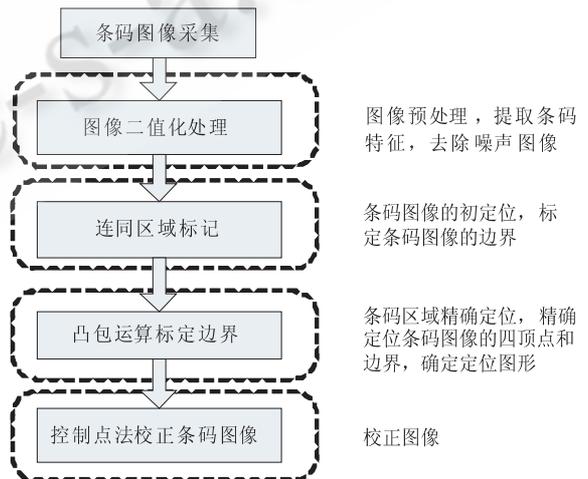


图 4 DataMatrix 条码图像校正流程

#### 4.1 图像的预处理

图像预处理是为了加强条码图像的边缘、区域等

有意义的特征信息,同时去除一些小噪声和污染,是整个条码图像识别、分析和校正的基础。图像的预处理,主要是对图像进行二值化处理。根据条码图像与背景图像有不同的灰度集合,且两个不同的灰度集合可用一个灰度级阈值  $T$  进行分割。如果图像中某像素的灰度值小于阈值,则该像素的值置为 0,否则灰度值置为 255,按下式对图像处理可以得到二值化图像  $g(x, y)$ 。

$$g(x, y) = \begin{cases} 0, & f(x, y) < T \\ 255, & f(x, y) \geq T \end{cases} \quad (5)$$

最简单的阈值选取方法是根据直方图来进行的,图像的灰度图是图像各像素灰度值的 1 的统计度量,将直方图看作一曲线,背景和目標物两区域部分所对应的直方图呈双峰且有明显的谷底,将谷底点所对应的灰度值作为阈值  $T$ 。

## 4.2 条码图像初定位

条码图像的初定位是初步标定整个条码图像的境界,有效地区分条码图像与背景图像。本文主要采用连通区域标记算法初步定位条码的区域。文献 [5] 中采用一种类似区域增长的思想,避免连通区域标记中重复标记的问题。该方法的基本思路是每次标记一整个连通区域,然后再标记下一个区域,直到标记完所有的连通区域,但是,该文献对算法的文字描述不够准确,伪代码描述存在条件语句和循环语句的作用范围不明确的问题,本文对该算法和伪代码进行改进,具体描述如下:

首先,对输入的二值图像逐行扫描,找到一个未标记区域的第一点,将该点新增到种子点集;从种子点集中取出一个种子(在种子集中删除该种子),先标记该种子,然后将该种子点的八邻域点满足标记要求且未被标记的点新增到种子点集中,在后续标记过程中,不断从种子点集中取出一个种子,标记该种子,采取上操作添加该点的八邻域点到种子集中,如此循环,直到种子点集为空,说明一个连通区域标记结束;依照上述方法,标记完二值图像的所有连通区域。假定  $I$  表示输入的二值图像,  $U$  表示未标记的连通区域点集,  $x, y$  表示点,  $V$  表示种子点集。该算法的伪代码描述如下:

- ① for each point  $x \in I$  do
- ② begin
- ③ if  $x \in U$  then
- ④ begin

- ⑤ AddPointsTOSeedPoints( $x, V$ );  
/\* 将新增的标记点  $x$  加到种子点集  $V$  中 \*/
- ⑥ while not empty of  $V$  do
- ⑦ begin
- ⑧ SeedPoint :=  $V$ . FirstElem();  
/\* 得到种子集  $V$  中的第一个种子点 \*/
- ⑨ SignPoint(SeedPoint); // 标记该种子点
- ⑩  $V$ . deleteFirstElem();  
/\* 种子集  $V$  中删除该种子点 \*/
- ⑪ while each point  $y \in$  EightNeighborHoodPoints(SeedPoint) do  
/\* 标记该种子点的八邻域点,将新增标记加入种子点集  $V$  中 \*/
- ⑫ begin
- ⑬ if  $y \in U$  then
- ⑭ AddPointsTOSeedPoints( $y, V$ );
- ⑮ end
- ⑯ end
- ⑰ end

条码图像经检测有可能得到多个连通区域,根据条码区域点的数量小于 180 和最大长度或宽度小 35 个像素两原则,排除一些非条码区域或者噪声图像,最后留下几个主要的条码区域,本文中图 1 经检测主要有四个连通区域,对条码图像的不同区域分别采用不同颜色,如下图 5 所示。

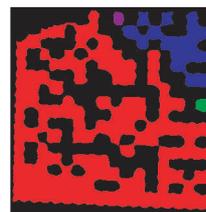


图 5 条码检测出的连通区域

## 4.3 条码的精确定位

条码图像的精确定位是精确定位条码图像的四个顶点和边界,并确定条码图像的定位图形,本文通过凸包运算实现。为了让凸包上的点能按照边界上出现的次序自左向右地排序,本文把凸包分成上凸包和下凸包两个部分。记平面有限点集  $P$  的凸包的所有顶点沿顺时针方向组成的列表  $\rho(P)$ ,上凸包各顶点列表

$\rho_{upper}$ , 下凸包各顶点列表  $\rho_{lower}$ , 可以得到如下计算凸包的算法<sup>[6]</sup>:

- ① 根据  $x$  坐标, 对所有点进行排序, 若  $x$  坐标相同, 则再按  $y$  坐标排序, 得到序列  $p_1, p_2, \dots, p_n$ ;
- ② 把  $p_1, p_2$  加入  $\rho_{upper}$ ;
- ③ 把  $p_3, \dots, p_n$  逐个加入  $\rho_{upper}$ , 每加入一个顶点, 当  $\rho_{upper}$  至少还有三个点, 而且最末尾的三个点所构成的不是一个右拐时, 将倒数第二个顶点从  $\rho_{upper}$  删除掉;
- ④ 在  $\rho_{upper}$  中加入  $p_n$  和  $p_{n-1}$ ;
- ⑤ 用与 3 类似的方法计算得到  $\rho_{lower}$ ;
- ⑥ 将第一个点和最后一个点从  $\rho_{lower}$  中删除, 将  $\rho_{lower}$  连接到  $\rho_{upper}$  后面得到  $\rho(P)$ , 完成凸包的计算。

该算法简单且易于实现, 算法复杂度较低。由于实际识别的条码多数会出现齿轮状, 计算的最小凸多边形可能有少许“伪边”和“伪顶点”。设定凸多边形的相邻边的夹角和边长的阈值分别  $13^\circ$  和 10 个像素点, 把夹角小于  $13^\circ$  的相邻边连成直线并且去除边长小于 10 个像素点的边, 最后得到只有少数几个有序的顶点组成的多边形  $P'$ 。对每个区域的最小凸多边形的顶点  $P'$ , 计算每个顶点的 Rating 值, 可确定“L”型的三个顶点。用凸包运算找到的条码的边界和四个顶点(顶点用红点表示, 边界用粉红色表示), 如下图 6 所示。

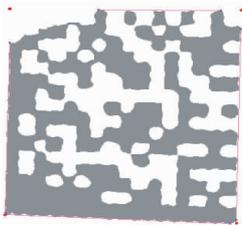


图 6 条码边界和四个顶点

#### 4.4 图像几何校正

将原条码图像映射到  $1000 \times 1000$  像素的校正正方形图像中, 根据条码图像的四顶点与  $(0, 0)$   $(0, 1000)$   $(1000, 0)$   $(1000, 1000)$  四点建立一一对应坐标关系, 其中条码图像“L”型的交点与点  $(0, 1000)$  对应, 其他点分别与对应的另外三点对应。根据公式(4)和(5), 通过四对控制点, 计算出公式中的系数  $a \sim h$ , 建立原条码图像点和校正图像点的变换关系。校正图像每点的值根据变换关系求出该点在原条码图像是对应

像素的像素值。图 7 为校正后的条码图像, 图中红线为条码图像根据每小块条码图像所作分割线。在码字切割时, 判断每小块条码图像所表示的信息是 1 还是 0, 只需统计校正后图像中每块小正方形中黑点和白点的个数, 如果黑点的个数大于等于白点的个数, 说明该小块条码图像表示的数据信息 1, 否则为 0, 即统计图 7 中红色分割线切分成的小正方形块中黑点和白点的个数。条码图像可能存在部分缺损的情况或者某个小块无法正常识别, 在译码时可以根据条码中的纠错算法纠错, 整个条码图像不需要每小块都完全识别。

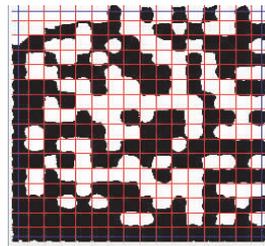


图 7 校正后图像

## 5 结论

本文采用凸包运算和控制点法, 提出了一种条码图像的畸变校正算法, 可以有效地校正条码图像在实际应用中出现的各种畸变, 并能快速确定条码的定位符号和定位符“L”型交点, 为条码识别时码字的准确切割打下了基础。该方法简单、易于实现并且效果好。

## 参考文献

- 1 刘宁钟, 杨静宇. 基于中点检测的二维条码识别. 小型微型计算机系统, 2004, 25(2): 283.
- 2 陈媛媛, 施鹏飞. 二维条形码的识别及应用. 测控技术, 2006, 25(12): 17-18.
- 3 ISO/IEC 16022: 2000(E), Information technology - International symbology specification - Data matrix.
- 4 杨帆等. 数字图像处理与分析. 北京: 北京航空航天大学出版社, 2007. 131-132.
- 5 陈柏生. 一种二值图像连通区域标记的新方法. 计算机工程与应用, 2006, 25: 46-47.
- 6 周利华. 手机端二维条码技术的研究与应用[硕士学位论文]. 浙江: 浙江工业大学, 2008.