复杂背景 PDF417 解码相关技术的研究

汪小丰 刘 志 郑河荣 许晓辉 （浙江工业大学软件学院系统结构研究所 杭州 310014）

摘要：本文研究了 PDF417 二维条码解码的相关技术，涉及了条码在复杂背景下的准确定位，条码行数的准确确定，条码列数的准确的确定，以及每个码字的准确位置的确定。定义了一些数据结构来记录相关重要的信息。

关键词：PDF417 边缘检测 二维条码 目标定位 码字分割

1 引言

条码技术是计算机的应用实践中产生的一种自动识别技术，由于其输入速度快，成本低，可靠性好，因而发展迅速，应用广泛。由于一维条码由于所能存储的信息量非常的有限，二维条码应运而生。目前应用比较广泛的二维条码有 pdf417，Code49，Dada Matrix 等。pdf417 被许多国家和国际机构采纳。我国于 1998 年制定了 pdf417 条码国家标准。

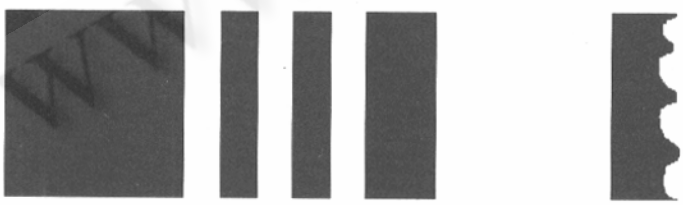

图 1

2 PDF417 结构的介绍

PDF417 被称为便携式数据文件（portable data file），它是一种多层，可变长的二维条码。每个 PDF417 二维条码都是由一堆垂直的对齐的行构成，行数最少三行，最多 90 行。每行最少包括一个条码字符（一个码字 Codeword），但开始字符，中止字符，行起始字符除外。每个字（码字）由四个条和四个空表示，从左向右，黑白条依次出现。每个条或空由一到六个模块组成（不包括开始符和中止符），模块的宽度是可变的，四个条和四个空总模块数为 17，故称为 417 条码。所有的 PDF417 二维条码的 开始字符和中止字符的信息都是一样的，开始信息：81111113 中止信息：711311121。下图符号字符为：51111125。

3 条码位置的确定

解码的第一步，就是要准确的确定条码在照片中的具体位置。由于拍照角度的原因，条码在照片中经常会发生一定的倾斜，在确定位置的时候我们要对他进行哈夫变换，矫正角度。对于一副比较干净的照片而言，我们比较容易的就可以准确的确定条码在照片中的具体的位置，但在实际的应用中，由于各种因素的影响，照片中会有各种各样的污点，有些污点和条码的开始符的特征非常的相似，在条码定位的时候我们必须要克服这些因素的影响。确定条码在照片中的具体位置的过程中，我们先确定条码的下边缘的位置，在确定条码的右边缘的位置，然后是条码的上边缘的位置，最后是条码的右边缘的位置。为了记录条码的位置，我们设计了一个结构来记录条码的位置信息。struct pos { int heighty; int widthx; }; heighty 记录的是象素点的垂直位置，widthx 记录的是象素点的水平位置，通过这个结构我们记录下条码四个顶点的位置，从而确定了条码的位置。

确定条码位置的时候，我们是从下向上，从左到右扫描的，如果遇到黑点，就判断是不是条码的左下端点。如果是就记录下该点的位置，如果不是就继续扫描。

确定某一个扫描到的黑点是不是左下端点，先取出该点右上区域的一块，进行扫描，看是否有白点，如果有则说明这点不是我们所要找的点，继续找一下黑点，否则继续判断该点右边的区域，看该区域是否满足 8:1:1:1:1:1:1:6 的特征，满足则是我们要找的点。然后从我们正在扫描的行最右端开始找到一个黑点使它的左边的区域满足 1:2:1:1:1:3:1:1:7 的特征，这样我

们就确定了条码的有下端的位置，我们再根据已确定的左右下端的位置确定左上端的位置，右上端的位置。pos Ldown，pos Lup，pos Rdown；，pos Rup，这四个数据用于记录端点的信息，试验结果表明这种方法能够准确快速的确定条码在照片中的位置。效果如图2。

图 2　定位条码

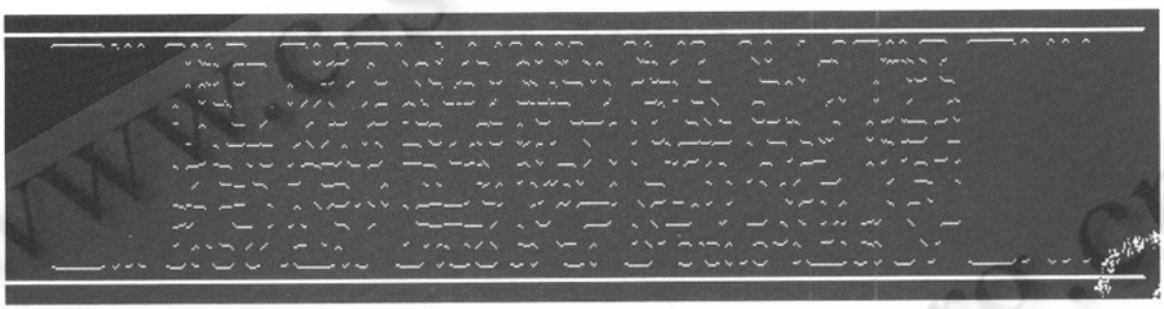

图 3　水平边缘检测

4　条码的行和列几码字的确定

4.1　条码的行确定

将图像转为灰度图像后，我们采用基于方差的大津（otsu）算法求出阈值 T，将图像转为二值图像，在此基础上定位出条码的具体位置，再求出条码的行数和列数。w0 ，w1 代表前景点和背景点占象数点的比例，u0，u1 为前景和背景的平均灰度值。U 为平均灰度值。找到一个灰度值 T 使得类间方差 g：

$$u = w0 * u0 + w1 * u1。 \quad (1)$$

$$g = w0 * (u0 - u)2 + w1 * (u1 - u)2 \quad (2)$$

最大，从而就确定了阈值 T。

$$f(x) = \begin{cases} 0 & f(x) <= T \\ 255 & f(x) > T \end{cases} \quad (3)$$

条码行数的确定是基于水平边缘检测，利用投影算法我们能准确的确定条码的行数。投影算法的数学基础如下：

G(x,y) = f(x,y) − f(x,y−1) x ，y 为象素点所在的行和列　(4)

该算法使用的卷积核如下：

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

对应用了投影算法的图像区域进行逐行扫描，并记下每行白点数，然后再对这些数据进行差分运算，从而确定条码的行数 mzhx

4.2　条码的列确定

在定位条码的时候我们记录下了条码左上角和右上角的位置，我们就可以很容易求出条码的宽 width =Rup. widthx − Lup. width ，根据条码的起始符可以求出条码的每个码字的宽 mzwidth，从而确定了条码的列数。

4.3　条码码字位置的确定

能否准确的解码和码字能否准确的确定有着直接的关系。为了确定码字的位置，我们必须要知道码字的高（mzheight）和宽（mzwidth），容易求出条码高：

height = Lup. heighty - Ldown. heighty,然后再根据已求的条码的行数就可以知道码字的高度,根据条码的行数和列数,码字的高度和宽度就可以的定位出每个码字的位置。

```
mzheigh = height / mzhx
    for( int i = Ldown. heighty ; i < = Lup. heighty ; i
+ = mzheight)
    for( int j = Ldown. widthx ; j < = Rdown. widthx
; j + + )
    {
      data[i * linebytes + j * 3] = 0;
      data[i * linebytes + j * 3 +1] = 255;
      data[i * linebytes + j * 3 +2] = 0;
    }
```

发现,每个信息码字都是由八个一到六的阿拉伯数字组合而成,所以我们可以定义一个新的结构来存储每个码字所代表的信息,用这种结构数组来存储正个条码的信息。结构如下:

```
struct Hbwidth {
    int h1;int b1;
    int h2; int b2;
    int h3;int b3;
    int h4;int b4;
    };
```

码字通过黑白条不同的宽度来表达信息,为了能够准确的解码,我们必须能够准确的确定各个黑白条的宽度,再把这些宽度除以条码模块的宽度从而得到码字所表达信息的编码,再匹配数据库就可以解出码字所要表达的信息。与传统的记录码字信息方式相比(传统的记录码字信息使用字符串来表示),大大地简化了后面的解码工作。

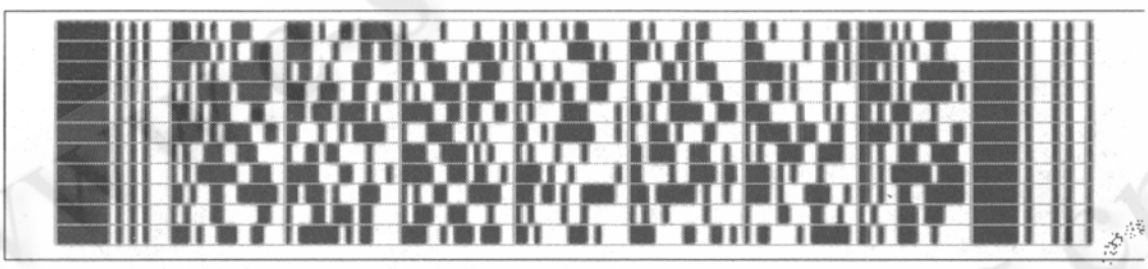

图 4　每个码字位置的确定

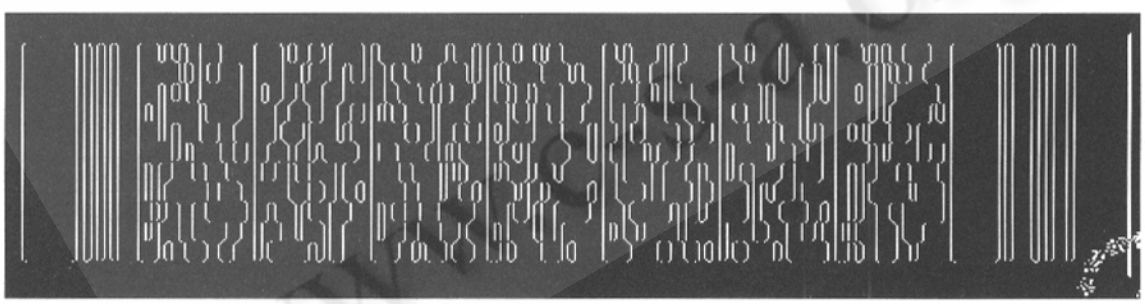

图 5　垂直边缘检测

5　码字信息的确定

在确定码字信息之前,我们必须选择一种合适的方式来存储这些信息,以往都是用字符串数组来存储这些信息,每一个字符串存储一个码子信息,但是用这种方式存储的信息,在后面匹配数据库的时候会非常的不方便,效率也比较低。我们研究码字的特点不难

我们采用垂直边缘检测来确定黑白条的宽度。垂直边缘检测的原理如下:

G(x,y) = f(x,y) - f(x-1,y) x ,y 为象素点所在的行和列　(5)

(下转第 60 页)

(上接第 56 页)

所采用的积分核：

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

条码经过垂直边缘检测以后，会有非常明显的垂直的白线，两条白线之间就是一个白条或者黑条，测出两条白线之间的象素点的个数，从而知道了白条或者黑条的宽度，将得到的数据与模块宽度相比较，得到一个数字，将其存入上述的一个结构中。

6 结论

通过边缘检测算法，准确的确定了条码的行数，但确定的码字黑白条的宽度的精度有待提高。通过定义新的结构来存储码字信息，极大的方便了后面的数据库查询，也提高了解码速度。

参考文献

1 Sriram T , Rao V K Applications of Barcode Technology in Automated Storage & Retrival Systems IECON Proceedings 1996 1.

2 杨淑莹，vc + + 图像处理程序设计，清华大学出版社，2005.

3 何斌，Visual C + + 数字图像处理，人民邮电出版社，2002.

4 刘宁钟，基于投影算法的二维条码识别，博士论文，2002.