

一种新的身份证号码定位方法

梁 艳, 潘保昌, 郑胜林, 甘汪雨

(广东工业大学 信息工程学院, 广东 广州 510640)

摘要: 提出了一种新的定位方法, 在分析身份证图片的 RGB 色彩分量的基础上, 利用红色分量对图片进行色彩的转换, 并采用差分法对目标进行定位, 然后根据灰度值的均值和方差对定位区域进行局部的二值化处理. 这种方法简单实用, 能有效地解决身份证号码定位的问题.

关键词: 定位; 身份证号码; 色彩转换; 差分法

中图分类号: TP391

文献标识码: A

文章编号: 1007-7162(2005)04-0062-05

身份证作为居民身份的象征, 它已与人们的日常生活息息相关, 成为人们生活中不可缺少的证明文件. 改革开放以来, 我国经济迅猛发展, 各城市、农村人口流动频繁, 身份证在经济社会生活中扮演着越来越重要的角色. 在很多地方, 例如机场、海关、边检、公安、旅馆等地方需要对其进行登记管理, 如果人工输入号码, 不但效率低, 而且容易出错. 如果能利用图像识别原理对身份证号码进行定位, 然后识别, 将大大提高工作效率. 对于身份证这种含有大量纹理及背景等强烈干扰的图像, 定位是实现号码自动识别的关键部分.

本文主要介绍一种新的定位方法, 该方法能准确、快速地实现身份证号码的定位.

1 色彩分析

身份证图片主要由字符、头像、激光防伪标识、网纹和印章构成, 在彩色图片里, 它们呈现出不同颜色, 互相交织在一起, 如果想直接从彩图里把字符取出来是很困难的, 所以要识别身份证号码, 首先要进行色彩转换.

在 RGB 彩色空间中, 每种颜色 C 由红、绿、蓝(R 、 G 、 B) 3 种成分组成^[1], 即

$$C = \{u, v, w\},$$

其中 $u \in R, v \in G, w \in B; R = \{0, 1, \dots, 255\}, G = \{0, 1, \dots, 255\}, B = \{0, 1, \dots, 255\}$. 这样, 根据 R 、 G 、 B 不同的组合就可以表示出各种不同的颜色.

通过扫描仪得到身份证图片的 24 bit 图像, 图像中的字符、激光防伪标识、网纹、印章分别呈现为: 黑色、浅紫(浅绿、浅蓝)色、绿色、红色. 在色彩转换中, 以号码、头像等呈现为黑色的图案为目标, 其它, 如激光防伪标识、网纹、印章等作为背景.

一般识别身份证号码时的色彩转换是把 24 bit 位图按照下式转换成灰度图^[2], 即

$$V_{\text{gray}} = 0.30R + 0.59G + 0.11B, \quad (1)$$

$$R = G = B = V_{\text{gray}}, \quad (2)$$

但这样得到的灰度图中, 目标和背景的灰度值相差不是很大, 不利于目标与背景的分离. 所以要采用另外的色彩转换方式.

只取图像的红色分量,即

$$C = \{u\}, \quad u \in R; R = \{0, 1, \dots, 255\},$$

作直方图,见图1.从图可以看出,目标的灰度值较低,主要分布在小于125的区域,而背景的灰度值较高,主要分布在大于125的区域.字符与背景的灰度值基本上没有交叉.

同理对图像的绿、蓝分量作分析,得到大致的RGB值分布如表1所示.从表看出,除了红色分量以外,在绿、蓝分量中,目标与背景的灰度值交叉,不利于目标与背景的分离.所以,单取图像的红色分量,把24 bit图转成8 bit的灰离图,如图2所示.此时得到的灰度图,字符偏暗,而背景偏亮,它们的灰度值相差比较大,这样的灰度图可以大大降低定位的难度.

表1 身份证中几种颜色的RGB值分布

部位(颜色)	R 值	G 值	B 值
字符(黑色)	< 125	< 100	< 100
激光防伪标识(浅紫、浅绿、浅蓝色)	> 125	> 100	> 150
网纹(绿色)	> 125	> 150	> 100
印章(红色)	> 125	< 100	< 100
背景(灰色)	> 125	> 100	> 100

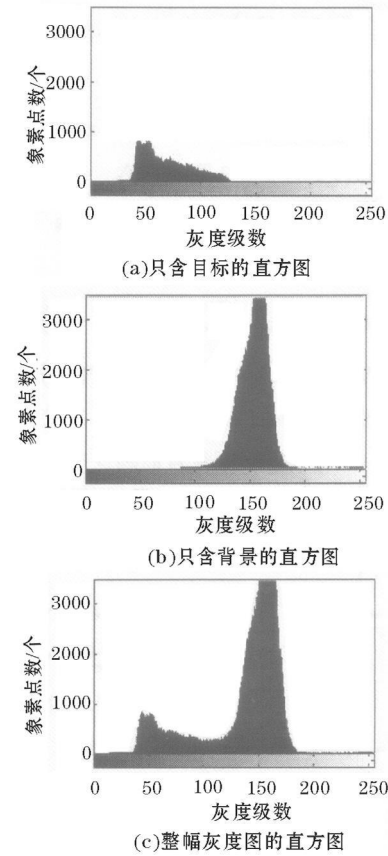


图1 取红色分量后作的直方图



图2 两种不同方法得到的灰度图

2 采用差分法进行号码的定位

通常采用的定位方法是:直接搜索灰度图的直方图,令波谷作为阈值对图像进行二值化,然后利用投影法对号码进行定位^[3].这种方法的关键在于找阈值.要找到适当的阈值并不容易,若选取的阈值偏大,就会有过多的目标象素点被错判为背景,造成号码不明显,甚至某些数字消失掉,严重影响了定位的效果;若阈值偏小,又会产生大量噪声,令定位不够准确.另外,由于号码相对于整幅图来说所占的面积不是很大,目标不够突出,这就更加大了搜索阈值的难度.所以在此提出一种差分法进行定位.

由本文的色彩分析部分可以知道,通过取红色分量得到的灰度图中,号码与背景之间的灰度值相差很大,特别是在交界的地方差值更大,可以利用相邻像素间的灰度值之差,构造出一幅差分图,根据图中包含的边缘信息,定出号码与背景的交界位置.

定义 设 $F = (f(x, y))$ 是一幅灰度图像的矩阵点集, $x = 1, 2, \dots, m; y = 1, 2, \dots, n$. 点 $f(x, y)$ 为一个像素, $f(x, y) \in \{0, 1, \dots, 255\}$.

定义 设 $S = (s(x, y))$ 是由灰度图 F 得到的差分图, 则

$$s(x, y) = \begin{cases} f(x, y) - f(x-1, y) & x = 2, 3, \dots, m, y = 1, 2, \dots, n; \\ f(x, y) - f(x, y-1) & x = 1, 2, \dots, m, y = 2, 3, \dots, n. \end{cases} \quad (3)$$

求 x 轴方向的灰度值之差时用公式(3), 求 y 轴方向的灰度值之差时用公式(4).

在差分图 S 中, 若 (x, y) 是位于号码与背景边缘的像素点, 那么 $s(x, y)$ 的值比较大, 否则, $s(x, y)$ 比较小.

设定某个阈值 T , 利用 T 可把差分图 S 进行二值化, 得到二值图 Bw , 其定义为

$$Bw(x, y) = \begin{cases} 0, & s(x, y) > T, \\ 1, & s(x, y) \leq T, \end{cases} \quad x = 1, 2, \dots, m; y = 1, 2, \dots, n. \quad (5)$$

当 $s(x, y)$ 的值大于 T 时, $Bw(x, y)$ 为黑像素; 当 $s(x, y)$ 的值小于等于 T 时, $Bw(x, y)$ 为白像素. 这样, 就可以把边缘点用黑像素表示, 而非边缘点用白像素表示. 然后再通过水平投影和垂直投影就可判断号码的具体位置了.

差分法实现的具体步骤如下:

1) 以灰度图 F 为对象, 把图中 x 轴方向相邻像素的灰度值之差存入矩阵 S_x 中.

2) 判断矩阵 S_x 中每个像素点的值, 如果大于阈值 T_x , 令 $Bw_x(x, y)$ 的灰度值为 0, 否则为

1. 其中, 阈值 T_x 的选取与图 S_x 的平均差分有关, 它可由下式表示:

$$T_x = \sum_{y=1}^n \sum_{x=2}^m s_x(x, y) \times 4 / [(m-1)n]. \quad (6)$$

3) 对 Bw_x 做水平投影, 得到投影图 $\text{plot}X$ (如图 3).

4) 根据投影图 $\text{plot}X$ 的极值和号码所在位置先验知识, 将号码的起始和终止纵坐标取出来, 分别存放在 X_1 和 X_2 中.

5) 同理, 再对 F 中纵坐标为 X_1 和 X_2 之间夹着的矩形部分做 y 方向的差分运算, 存入矩阵 S_y 中.

6) 判断矩阵 S_y 中每个像素点的值, 如果大于阈值 T_y , 令 $Bw_y(x, y)$ 的灰度值为 0, 否则为

1. 其中, 阈值 T_y 的计算公式是.

$$T_y = \sum_{x=1}^m \sum_{y=X_1}^{X_2} s_y(x, y) \times 4 / [m(n-1)]. \quad (7)$$

7) 对 Bw_y 做垂直投影, 得到投影图 $\text{plot}Y$ (见图 4).

8) 根据投影图 $\text{plot}Y$ 的极值, 将每个数字的横坐标取出来, 保存横坐标.

因为噪声的影响, 在搜索号码的横、纵坐标时要设定最低高度, 当投影值低于这个高度就认为是波谷. 投影图 $\text{plot}X$ 的最低高度 $X_{\min}$ 、投影图 $\text{plot}Y$ 的最低高度 $Y_{\min}$ 分别由下式决定:

$$X_{\min} = \sum_{i=1}^m \text{plot}X(i) / m, \quad (8)$$

$$Y_{\min} = \sum_{i=1}^n \text{plot } Y(i) / n, \tag{9}$$

定位的过程及效果图见图 3、图 4、图 5。

3 局部二值化

利用差分法定位后, 只能确定号码在灰度图中的具体位置, 为了便于识别, 还需要作后处理, 对号码作局部二值化^[4]。

对于身份证号码中单个数字的二值化过程, 要选择动态的阈值以保证每个数字二值化的准确性。如果用搜索直方图谷值的方法, 会非常费时。考虑到分割后的数字图像中, 目标的像素点占较大面积, 目标与背景之间的灰度值相差比较大, 而且背景的亮度变化不大, 可采取基于均值和方差的方法搜索阈值^[5], 即对每幅数字图像求均值和方差, 然后利用均值和方差进行二值化。定义数字图像中像素点灰度的均值为 m , 方差为 s , 则阈值的公式为

$$T = w_1 m + w_2 s, \tag{10}$$

T 为图像的阈值, w_1 和 w_2 为预先设定的修正系数, 实验中分别取值 0.98 和 -0.2。假设图像为 $H \times W$ 像素大小, 点 (x, y) 的灰度值为 $f(x, y)$, 则 m 、 s 的计算公式如下:

$$m = \frac{1}{HW} \sum_{y=1}^H \sum_{x=1}^W f(x, y), \tag{11}$$

$$s = \sqrt{\frac{1}{HW} \sum_{y=1}^H \sum_{x=1}^W f^2(x, y)}. \tag{12}$$

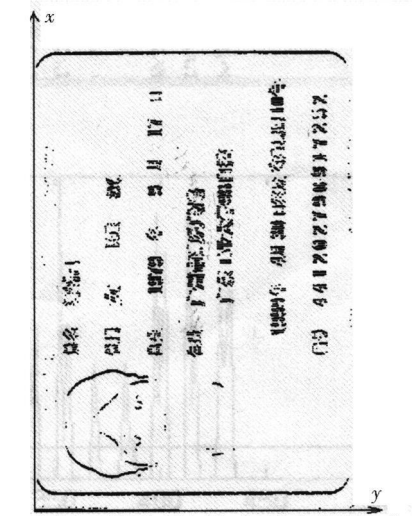
4 实验结果

取 100 幅身份证扫描图像进行号码定位测试, 实验结果: 单个数字定位的正确率为 99.4%, 整张通过率 (即整张身份证的号码全部定位正确) 为 95%。把定位后的单个数字图像归一化成 16×10 像素的字符点阵, 然后输入三层的 BP 神经网络进行识别^[6], 结果为: 单个数字的识别率为 99.2%, 整张通过率 (即整张身份证的号码全部定位正确) 为 92%。

对产生定位误差的图像进行分析, 发现产生误差的数字一般为下列两种情况: 一是由于套膜磨损严重, 造成数字信息模糊不清, 给定位增加了难度; 二是身份证上人为污迹遮盖了部分数字, 提取有效信息比较困难。

5 结论

本文提出了利用红色分量进行色彩转换、采用差分法进行号码的定位等新方法, 这些方法简单实用, 对身份证的定位非常有效。从实验结果可以看出, 这些方法完成了自动定位身份证号码的功能, 准确率高, 是一种不错的方法。



(a) 差分图 Bu_x (为了与图(b)对应, 把图像逆时针旋转 90°)

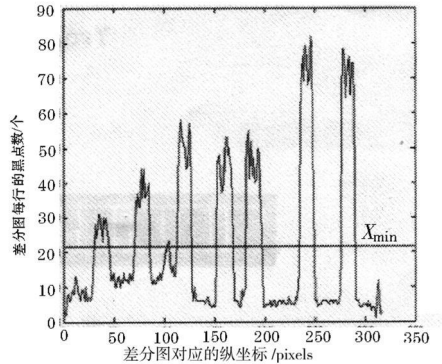


图 3 对差分图作水平投影

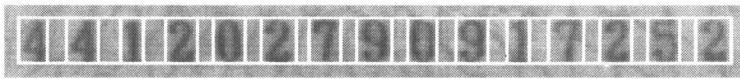
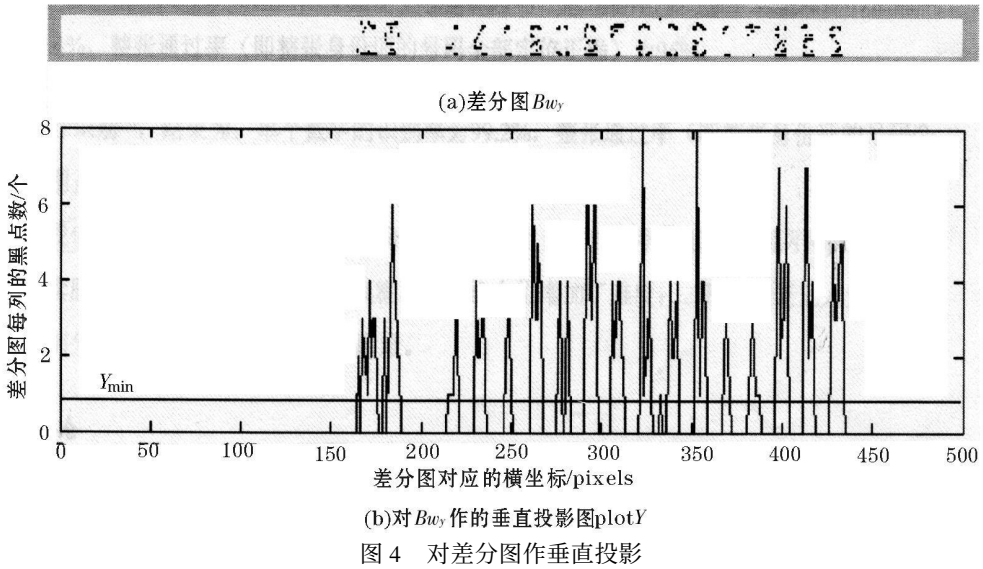


图 5 定位的效果图

另外, 该定位方法不仅针对身份证号码取得了相当好的效果, 同时对其它类型的图像中字符区域的定位也具有重要的参考价值.

参考文献:

- [1] 章毓晋. 图象处理和分析[M]. 北京: 清华大学出版社, 2001.
- [2] 周长发. Visual C++ .net 图像处理编程[M]. 北京: 电子工业出版社, 2002.
- [3] 黄山, 李岱璋, 尚建华. 一种身份证图像自动识别中文字定位与分割的方法[J]. 计算机应用研究, 2002, (10): 80-81.
- [4] 殷瑞祥, 李国华. 身份证号码的自动识别系统[J]. 华南理工大学学报, 2002, 30(2): 94-96.
- [5] 龙钧宇, 金连文. 一种基于全局均值和局部方差的图像二值化方法[J]. 计算机工程, 2004, 30(2): 70-72.
- [6] 贺光时, 刘宇. BP 神经网络算法在数字识别中的应用[J]. 西北纺织工学院学报, 2000, 14(4): 356-362.

New Location Method of ID Card Numbers

LIANG Yan, PAN Bao-chang, ZHENG Sheng-lin, GAN Wang-yu

(Faculty of Information Engineering, Guangdong University of Technology, Guangzhou 510640, China)

Abstract: A new location method is proposed in this paper. Based on the analysis of RGB color field, the image of ID card is made color conversion by red component and located with difference method. And then the located areas are separately binarized based on mean and standard deviation. This method is simple and practical. It can effectively solve the problem of location of ID card numbers.

Key words: orientation; ID card number; color conversion; difference method