

# 改进水印图片里文字识别效果的新方法

骆剑锋<sup>1</sup>, 谌 颀<sup>2</sup>

(1. 华南理工大学 计算机科学与工程学院, 广东 广州 510641;

2. 广东技术师范学院 天河学院计算机系, 广东 广州 510641)

**摘要:** 现今, OCR 文字识别技术已经飞快发展, 如一般的车牌识别、扫描文本识别等技术的识别率已达到非常高的水平, 但这些识别技术在带水印图片的文字识别这一方面表现一般。其原因主要在识别前后的处理还不够。主要讲述了文字识别的新预处理的方法和后期处理的实现。

**关键词:** USM 锐化; 最大类间方差; 阈值; 平均灰度值; 二值化阈值; 拼写与语法检查

中图分类号: TP391.43

文献标识码: A

## A new way to improve the effectiveness of character recognition from watermark picture

LUO Jian Feng<sup>1</sup>, SHEN Hang<sup>2</sup>

(1. College of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China;

2. Computer Department of Tianhe College, Guangdong Polytechnic Normal University, Guangzhou 510641, China)

**Abstract:** The technology of OCR has been developed very well so far, such as in the field of license plate recognition and book recognition, in which the recognition rate is very good. But these technologies are not good at recognizing the character from the watermark pictures. Because the technology lacks of pre-treatment and post-treatment of the recognition. In this article, we will introduce a new way to show how to fulfill the pre-treatment and how to take the post-treatment activity.

**Key words:** USM sharpen; maximum normalized square; threshold value; the average of gray value; the binary-conversion threshold value

所谓带水印的图片, 其实就是在文字的上再加上一层半透明的图像, 如图 1 所示。这层图像不会影响读者对文字的阅读, 但如果计算机要提取图片里面的内容就会非常的困难。



图 1 带水印的图片

为了能准确且方便地从带水印的图片中提取内容, 本文对图片的预处理分成三步: 第一步是进行灰度处理, 第二步是用自动获取阈值的 USM 锐化, 最后是对图片进行二值化处理。

在众多文字识别技术中, 暂时还没有一种识别方法能保证识别出来的内容是准确的, 因此, 在进行识别后

的检查和纠正就显得很重要。为了在识别错误后能尽量地纠正, 需要对识别结果进行拼写和语法检查及纠正。所以在本文的后部分还介绍一种简单的识别结果纠正程序。

### 1 文字识别的预处理

#### 1.1 灰度处理

灰度处理主要是对图片中每一像素的 RGB 值进行映射, 映射到 0 至 255 的灰度值。所以得出来的图片就像黑白照片一样。这样做就可以降低后面步骤的运算量, 并且使得图片更加易于辨认。效果图如图 2 所示。

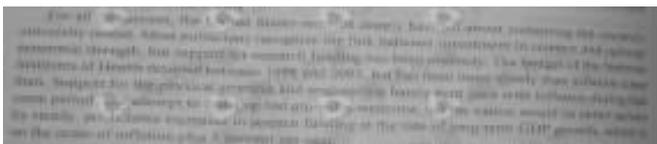


图 2 灰度处理后的图片

## 1.2 USM 锐化

由于水印是半透明地盖在文字资料上,使得水印下的文字的边缘变得模糊且难以辨认,因此要对图片锐化,这样会突显图片中的文字,有利于实现图片中的文字与水印分离。

在这里不用一般的锐化方法,因为一般的锐化方法只是对边缘进行增强,而 USM 锐化则不一样。USM 锐化前和 USM 锐化后的图分别为图 3 和图 4。



图 3 USM 锐化前



图 4 USM 锐化后

从图 3 和图 4 的对比中,可以清楚地看到,USM 锐化不单可以把文字一边的边缘增强,同时它还把文字边缘的另外一边明显地减弱。通过这种方法,文字就会被突显出来,而文字附近的图像就会被减弱,从而减少半透明水印的面积。并且文字的边缘还会出现白边(其实是由于边缘减弱造成的部份),这就为下一步二值化埋下了伏笔。

USM 锐化算法中的阈值是需要设定的,阈值过大,锐化就不能产生出效果;设定的阈值过小,那么就会同时把文字与水印的边缘都增强,什么都分不清楚了。由于阈值的设定至关重要,阈值是通过最大类间方差法定下来的,方法如下:

对于图像  $I(x, y)$ ,前景(即目标)和背景的分割阈值记作  $T$ ,属于前景的像素点数占整幅图像的比例记为  $w_0$ ,其平均灰度  $\mu_0$ ;背景像素点数占整幅图像的比例为  $w_1$ ,其平均灰度为  $\mu_1$ 。图像的总平均灰度记为  $\mu$ ,类间方差记为  $g$ 。

假设图像的大小为  $M \times N$ ,图像中像素的灰度值小于阈值  $T$  的像素个数记作  $N_0$ ,像素灰度大于阈值  $T$  的像素个数记作  $N_1$ ,则有:

$$\omega_0 = \frac{N_0}{M \times N} \quad (1)$$

$$\omega_1 = \frac{N_1}{M \times N} \quad (2)$$

$$N_0 + N_1 = M \times N \quad (3)$$

$$\omega_0 + \omega_1 = 1 \quad (4)$$

$$\mu = \omega_0 \mu_0 + \omega_1 \mu_1 \quad (5)$$

所以方差式为:

$$g = \omega_0 (\mu_0 - \mu)^2 + \omega_1 (\mu_1 - \mu)^2 \quad (6)$$

再把式(5)代入式(6),得式子:

$$g = \omega_0 \omega_1 (\mu_0 - \mu_1)^2 \quad (7)$$

然后,在程序中使  $T$  值遍历 0 到 255,每次都算出方差值  $g$ ,最后能使  $g$  最大的  $T$  值就是需要的阈值。但  $T$  值无需遍历那么多次,因为文字一般都接近黑色,所以

可以遍历从 1 到 90。灰度图片 USM 锐化后的效果如图 5 所示。

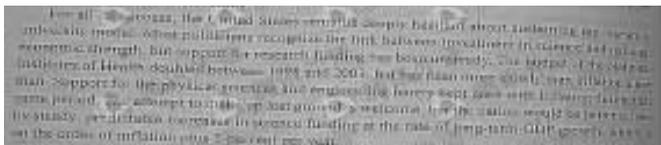


图 5 USM 锐化后的图片

## 1.3 二值化处理

通过二值化处理就可以得到主要的文字的图片了。而这一步最重要的地方是二值化处理时要选取的阈值,阈值决定了能否把图片中的水印图除掉。通过以下方法算出阈值:

(1)横向扫描全图,把两边为浅色、中间为深色的所有点(如图 6 中 F 被横框选中处)的灰度值累加(设总值为  $sum_1$ ),同时也累记点数(设总数为  $p_1$ )。

(2)纵向扫描全图,把上下两边为浅色、中间为深色的所有点(如图 6 中 F 被纵框选中处)的灰度值累加(设总值为  $sum_2$ ),同时也累记点数(设总数为  $p_2$ )。

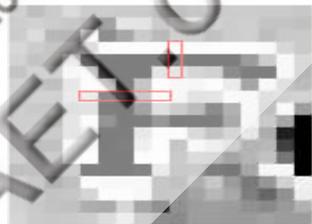


图 6 字 F 的放大图片

(3)得出的二值化的阈值就为:  $(sum_1 + sum_2) / (p_1 + p_2)$ 。

简单来说,就是把被白边围上的黑点的灰度值进行累加,它的和与黑点的数量相除,就得到了这些黑点的平均灰度值。这也正是在第二步用 USM 锐化的原因。用该阈值对图片进行二值化后的效果如图 7 所示。

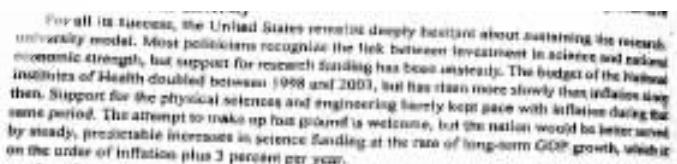


图 7 二值化后的图片

## 2 文字识别的后期处理

经过预处理后,就可以用很多现有的 OCR 文字识别算法把图 7 中的文字提取出来,这里就不再进行阐述了。

不难发现,无论预处理做得有多好,识别算法技术有多高,都难免有出错的时候,因此要对识别的结果进行拼写和语法的检查和纠正,这里直接给出实现该功能的程序。众所周知,微软 office 中的 word 里有拼写和语法的检查纠正功能,并且比较完善。

使用 VB.NET 实现拼写和语法检查纠正功能的过程如下:

(1)确定已安装 office 2003。VB 中已建立 windows 窗体,窗体中有 TextBox,它是存放 OCR 识别的结果,还有 Button,Button 中的事件就是拼写与语法检查与纠正代码。

(2)在 VB 中引用 Word 组件:在解决方案浏览器中的“引用”上右键->添加引用->标签页中选“COM”->选中“Microsoft Word 9.0 Object Library”,单击“确定”即可。

(3)在代码的最前端添加三个语句:Imports Microsoft.Office.Core, Imports Microsoft.Office.Interop.Word 和 Imports System.Runtime.InteropServices,然后在 Button 的单击事件中加如下代码即可。

```
Dim objWord, objTempDoc As Object
    ' 创建 Word 对象和临时文档
Dim iData As IDataObject
    ' 声明 IDataObject 存放从剪贴板返回的内容
objWord=New Word.Application
objTempDoc=objWord.Documents.Add
    ' 实例化 Word 对象
Clipboard.SetDataObject(TextBox1.Text)
    ' 复制文本框中的识别结果到剪贴板
With objTempDoc
.Content.Paste() ' 把剪贴板中的内容粘贴到临时文档
.Activate()
.CheckSpelling() ' 拼写检查
```

```
.CheckGrammar() '语法检查
.Content.Copy()
iData=Clipboard.GetDataObject()
TextBox1.Text=CType(iData.GetData(DataFormats.Text),
String) ' 利用剪贴板把修改后的内容返回给文本框
.Close()
End With
objWord.Quit()
```

本文中所讲述的最大类间方差定 USM 锐化阈值法、基于 USM 锐化后求黑点平均灰度值的图像二值化处理法和对识别结果进行检查纠正,经过多次实验,针对带水印图片的文字识别率可高达 98.64%。而前面两种处理图像的方法及它们的巧妙结合正是本文的亮点,并且是原创的。但本算法的运行速度并不理想,在以后的研究中希望有新的突破。

#### 参考文献

- [1] 杨柳,牛秦洲.啤酒瓶凸性模号图像预处理算法[J].电脑知识与技术,2007(11):1089-1091.
- [2] 王勇智.数字图象的二值化处理技术探究[J].湖南理工学院学报(自然科学版),2005(1).
- [3] 吕学强,迟呈英.英文光学字符识别的后处理[J].鞍山钢铁学院学报,2002(25):192-196.

(收稿日期:2009-03-09)