

非均匀光照文档图像快速二值化方法

曾凡锋 王战东 郭正东
(北方工业大学信息工程学院 北京 100144)

摘 要 拍照方式获取的文档图像在 OCR 识别中,可能因光照不均而导致识别率低下。针对此问题,提出一种基于分块的自适应文档图像快速二值化方法。根据文档区域亮度特征划分光照均匀区域、阴影区域及亮光区域,对不同区域自适应地选取最有效的二值化算法。为解决传统 White 算法自适应能力的不足,对其进行改进,有效减少了伪影和断笔的产生。实验结果表明,该方法能明显提高光照不均文档图像的 OCR 识别率,且校正速度快、鲁棒性好。

关键词 光照不均 图像块 二值化 区域亮度 区域 Otsu 算法 White 算法

中图分类号 TP391.1 文献标识码 A DOI:10.3969/j.issn.1000-386x.2015.11.043

FAST BINARISATION METHOD FOR UNEVEN ILLUMINATION DOCUMENT IMAGES

Zeng Fanfeng Wang Zhandong Guo Zhengdong
(College of Information Engineering, North China University of Technology, Beijing 100144, China)

Abstract The OCR recognition rate of camera-based document images may be low due to uneven illumination. To solve this problem, the paper puts forward a blocking-based fast adaptive binarisation method for document image. The method distinguishes the uniform illumination area, shaded area and strong light area according to regional lightness feature of the document, and selects the most effective binarisation algorithm adaptively for different areas. Traditional White algorithm has been improved to solve its shortage in adaptive ability, which effectively reduces the ghost artifacts and the phenomenon of missing strokes. Experimental results show that the method can significantly improve the OCR recognition rate of uneven illumination document images with fast correction speed and good robustness.

Keywords Uneven illumination Image block Binarisation Regional lightness Regional Otsu algorithm White algorithm

0 引 言

通过拍照方式获取的文档图像通常会存在光照不均的现象,影响 OCR 识别效果,为此,需要对原始文档图像加以校正后再进行识别。二值化作为图像预处理的主要方法,其处理结果对后续的目标识别有直接影响^[1]。

目前的二值化算法根据阈值求取方法不同,可以分为全局阈值二值化法和局部阈值二值化法^[2]两种。

全局阈值二值化法根据文档图像的直方图或灰度空间分布确定一个阈值,以该阈值来对整幅图像二值化。经典的全局阈值二值化算法有 Otsu 算法、最优阈值法、简单统计法、基于熵的阈值法等^[3]。这类方法具有运算简单、实时性好等优点。然而,全局阈值二值化法在求取阈值时只考虑了图像的整体灰度特征,对局部信息考虑不足,导致其适用范围局限于前景和背景有较好划分的图像^[4]。对于出现背景不均匀、模糊、或者噪声比较多的低质量图像,此类方法不再适用。

局部阈值二值化法又可分为分块局部阈值法和自适应阈值法^[5],针对图像的不同位置计算阈值并进行二值化。分块局部阈值法的基本思想是将原始图像划分成块,针对每一块使用类

似求取全局阈值的方法求取块阈值,然后对该图像块进行二值化处理;自适应阈值法考虑了以当前像素点为中心的若干个邻域像素点的局部灰度特征,根据局部灰度特征求取阈值来对中心像素点进行二值化。经典的局部阈值二值化算法有 Niblack 算法、Bernsen 算法、Sauvola 算法、White 算法等^[6]。局部阈值二值化法更能适应复杂的情况,但由于需要进行多次阈值的求取运算,比较耗时。对于不均匀光照图像,传统的局部阈值二值化算法也存在二值化过程中易出现笔画丢失和假目标(在背景区域受噪声干扰得到笔画结果)^[7]的问题。

通过对不均匀光照文档图像灰度分布及亮度特征的分析,结合全局阈值二值化法和局部阈值二值化法各自的特点,本文提出一种能够快速对不均匀光照文档图像进行二值化的方法。方法根据图像块的亮度特征自适应地选取最为合适的二值化算法,有效地减少了传统算法中易产生笔画丢失和假目标的现象,而且处理速度快。

收稿日期:2014-03-21。国家自然科学基金资助项目(61371142);十一五国家科技支撑平台重点基金项目(2009BA171B02);北京市自然科学基金资助项目(4132026);北京市属高等学校人才强教计划资助项目(PHR201007121)。曾凡锋,副研究员,主研领域:图像处理,智能控制,系统辨识等。王战东,硕士生。郭正东,硕士生。

1 Otsu 算法^[8]与 White 算法^[9]简介

1.1 Otsu 算法

Otsu 算法是 1979 年由日本的大津展之在判断最小二乘原理的基础上提出的。其基本思想是首先把图像的灰度直方图在某一初始阈值处分割成两组,然后求两组间的方差并不断迭代,当某一阈值使得两组间方差最大时,即确定该值为整幅图像的分割阈值 T 。

$$T = Arg\{\max_{0 < i < L} \delta^2(t)\}$$
 (1)

其中, L 为图像灰度化后的最大灰度级, $\delta^2(t)$ 为两组间的方差。

$$\delta^2(t) = w_0(u_0 - u_T)^2 + w_1(u_1 - u_T)^2$$
 (2)

其中, w_0 和 w_1 分别为两组中像素出现的概率, u_0 和 u_1 分别为两组像素的灰度均值, u_T 为图像的总灰度均值。

Otsu 算法运算简单、实时性强,一直受到很高评价,被认为是全局阈值二值化方法中的最优方法,在图像分割领域有重要的意义和广泛的应用^[10]。

1.2 White 算法

White 算法的基本原理是遍历图像的每一个像素点 (i, j) , 根据式(3)计算以 (i, j) 为中心、尺寸为 $b \times b$ 像素的局部窗口内像素点的灰度平均值 $mean(i, j)$:

$$mean(i, j) = \frac{1}{b \times b} (\sum_{m=-b/2}^{b/2} \sum_{n=-b/2}^{b/2} f(i+m, j+n))$$
 (3)

然后根据式(4)计算像素点 (i, j) 二值化后的像素值。

$$G(i, j) = \begin{cases} 255 & \text{若 } mean(i, j) < f(i, j) \times bias \\ 0 & \text{其他} \end{cases}$$
 (4)

其中, $f(i, j)$ 为像素点 (i, j) 的灰度值,参数 $bias$ 取 $1 \sim 3^{[9]}$, 0 表示是黑色, 255 表示白色。

White 算法在局部阈值算法中有较好的评价,对光照不均、模糊的文档图像有一定的适应能力。文献[11]分析和比较了 40 多种局部阈值二值化方法,指出 Sauvola 方法^[12]和 White 方法的性能较优。相对于 Sauvola 算法,White 算法由于不需要对每一个像素点计算局部窗口内像素的平均方差,拥有更高的运行效率。

2 基于区域亮度特征的校正方法研究

2.1 区域亮度特征分析

在处理光照不均图像的问题上,均值、方差、梯度等图像特征受到了学者们的重视,但是图像的亮度特征并未得到充分利用。实际应用中文档图像的光照不均现象主要有以下表现形式^[13]: (1) 图像整体或部分处于阴影下而导致灰度值偏低,使得信息难以提取。(2) 图像整体或部分处于亮光下或者图像中含有反光特性的物体而导致灰度值偏高,像素信息受到损失。由于图像出现局部阴影或亮光的现象在实际应用中更为常见和难以校正,所以本文主要解决此类问题。

直观分析,阴影区域的平均亮度要低,亮光区域的平均亮度要高。通过对 100 幅光照不均图像的区域亮度特征统计测试与分析,发现通过区域亮度的差异来区分光照均匀区域、阴影区域及亮光区域是可行的。

如图 1 所示,研究中将灰度图像的 256 个灰度级均分为四个灰度区间,如果某像素点的灰度值在 $[0, 63]$ 之间则将该像素点归为低灰度像素点,记为 LowPixel;如果某像素点的灰度值在 $(191,$

255] 之间则将该像素点归为高灰度像素点,记为 HighPixel。

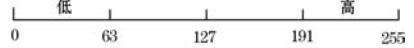


图 1 等分灰度级

本文将图像的区域亮度特征定义为灰度图像某一区域内 LowPixel 的比例和 HighPixel 的比例。

对于文档图像,无论某一区域内的笔画像素多或是少,正常排版及光照情况下背景像素点所占比例要远大于笔画像素点所占比例,也就是说 HighPixel 一般要占很大比例。为总结阴影和亮光对图像区域亮度特征造成的影响,研究中统计分析了光照均匀图像和光照不均图像灰度直方图之间的差异。

如图 2 所示,第 1 列为光照均匀图像及其灰度直方图,可以看出其直方图分别在 $[0, 63]$ 和 $(191, 255]$ 出现了一个峰值,呈双峰态。第 2 列为第 1 列中图加上局部阴影后的图像及其灰度直方图,其直方图出现了三个峰值,而且和第 1 列相比,HighPixel 的比例大幅降低,LowPixel 的比例变化不大。第 3 列为第 1 列中图加上局部亮光后的图像及其灰度直方图,其直方图呈单峰态,大部分像素点的灰度值集中在 $(191, 255]$ 区间内。通过对多幅类似以上三类图像及其灰度直方图的对比分析,研究发现不均匀光照对文档图像的区域亮度特征具有以下影响:

- (1) 一般情况下,亮光或阴影主要影响的是灰度值处于 $(63, 255]$ 之间的像素点的比例,使之增加或减少。
- (2) 阴影会造成区域内 HighPixel 的比例下降,但对 LowPixel 的比例影响很小,因为阴影一般不会使背景像素点的灰度值降低到与笔画像素相接近的程度。
- (3) 亮光会造成区域内 HighPixel 的比例大幅增加,同时使 LowPixel 的比例减少。

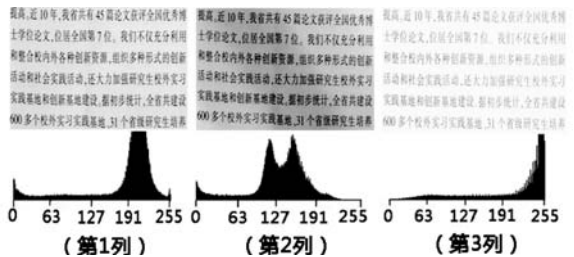


图 2 灰度直方图对比

通过提取图像区域 LowPixel 和 HighPixel 所占的比例,然后根据以上规律可以区分出光照均匀区域、阴影区域和亮光区域。

需说明的是,上述规律是在文档图像无过亮亮光以致文字笔画过浅,也无过暗阴影以致文字笔画过于模糊的情况下统计总结得出的。

2.2 传统二值化算法的改进

2.2.1 区域 Otsu 算法

前文已提及,Otsu 算法有其自身的不足。该算法使用单一阈值对整幅图像进行二值化处理,当图像出现局部对比度变换较大的情况时,Otsu 算法无法得到满意的二值化效果^[14]。

文献[15]根据灰度图像的水平投影智能地将图像划分成很多不同尺寸的小图像块,计算各块内像素的灰度均值 μ 及标准差 σ ,根据二者与设定参数 μ_T 和 σ_T 的大小关系来判定块中目标与背景的分布,对非全背景块采用 Otsu 算法的原理求取块阈值进行二值化。这种方法总体分割性能较好,但在实际应用中 μ_T 和 σ_T 的选择难以控制,造成方法鲁棒性较弱。而且该方法分块过细,容易产生较多的边界效应^[10]。文献[16]将图像均

等分块,计算各块标准差并按大小排序,然后与用户交互将图像块归为只含背景或目标、含少量背景或目标、背景目标均衡分布三个不同的区域,再分别进行二值化处理。对于第三类区域同样也采用了区域 Otsu 算法求取块阈值。该方法分割能力较强,但需要用户给出交互信息,不能满足自动化处理的需求。

研究发现,对于光照不均文档图像,在光照均匀区域和笔画像素较多的阴影区域使用区域 Otsu 算法能获取较好的分割效果。但在亮光区域和笔画像素较少的阴影区域 Otsu 算法会失效,主要表现为对亮光区域容易产生较多的断笔现象,对笔画像素较少的阴影区域则产生大面积伪影,如图 3、图 4 所示。

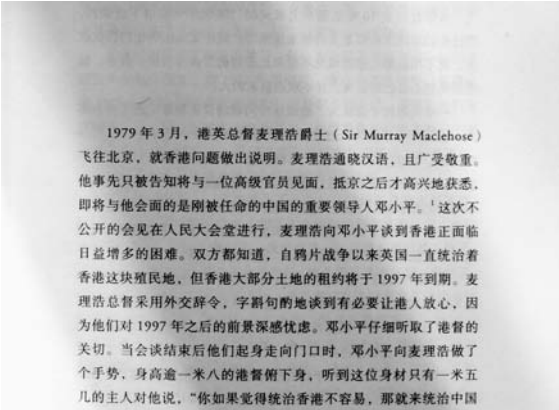


图 3 原图局部

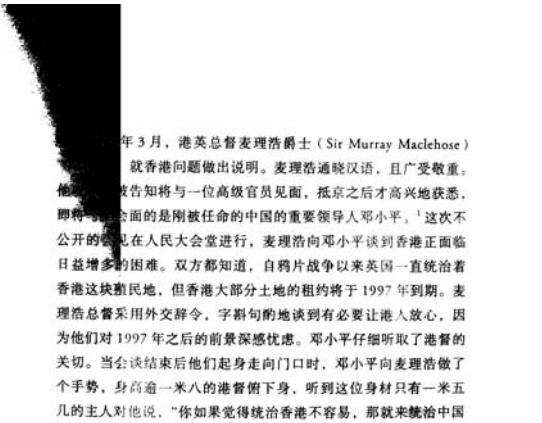


图 4 区域 Otsu 算法处理图

将图 3 均分为 2 行 4 列共 8 个区域块,从左到右下依次记为第 1~8 块,分别对每块采用区域 Otsu 算法进行二值化处理。从处理后的结果图 4 可以看出,第 1 块为笔画像素较少的阴影区域,在对其二值化时产生了大面积的伪影;同为阴影区域的第 2、6、7 等图像块等由于笔画像素较多,取得了较好的分割效果。因此,可以将区域 Otsu 算法应用于光照均匀区域和笔画像素较多的阴影区域的二值化处理中。由于笔画像素点的灰度值一般很低,可以根据区域内 $LowPixel$ 的比例大概判定该区域内的笔画像素多或是少。

2.2.2 White 算法的改进

White 算法需要对全部像素点逐个进行局部阈值运算,速度较慢。选取的局部窗口越大耗时越长,窗口过小又容易造成局部窗口内全是背景或全是前景,导致伪影或断笔的产生。文献[11]指出 White 算法局部窗口大小一般取 15×15 , $bias$ 通常取 2。

研究发现,实际应用中 White 算法对系数 $bias$ 的选取非常敏感, $bias$ 选取偏大容易出现笔画丢失,尤其在亮光区域,亮光

会使部分笔画像素的灰度值增大而导致该像素被错判为白色像素; $bias$ 选取偏小容易在阴影区域出现伪影,阴影会使部分背景像素点的灰度值降低而引起错判。经大量样张统计测试,当处理对象是文档图像时, $bias$ 以取 $1.05 \sim 1.4$ 为宜。

基于 White 算法的以上不足,本文从如何针对不同区域亮度选取合适的 $bias$ 出发,对 White 算法提出以下几点改进:

(1) 对于亮光区域,式(4)中取较小的 $bias$ 以减小 $mean(i, j) < f(i, j) \times bias$ 的概率,从而增大 $G(i, j) = 0$ 的概率,减少笔画像素的丢失。即:如果某一区域被判定为亮光区域,则使用式(5)对该区域进行二值化。

$$G(i, j) = \begin{cases} 255 & \text{若 } mean(i, j) < f(i, j) \times bias_Low \\ 0 & \text{其他} \end{cases} \quad (5)$$

其中, $bias_Low$ 取 $1.05 \sim 1.15$ 。

(2) 对于阴影区域,式(4)中取较大的 $bias$ 以增大 $mean(i, j) < f(i, j) \times bias$ 的概率,增大 $G(i, j) = 255$ 的概率,从而减少伪影的产生。即:如果某一区域被判定为阴影区域,则使用式(6)对该区域进行二值化。

$$G(i, j) = \begin{cases} 255 & \text{若 } mean(i, j) < f(i, j) \times bias_High \\ 0 & \text{其他} \end{cases} \quad (6)$$

其中, $bias_High$ 取 $1.2 \sim 1.4$ 。

(3) 在增加了(1)和(2)两步处理的基础上,将 White 算法局部窗口的尺寸缩小为 7×7 或 9×9 。

改进(1)和改进(2)使得 White 算法具有了更强的适应局部亮光和局部阴影的能力, $bias_Low$ 和 $bias_High$ 的取值范围经由大量测试统计得出,可操作性强。改进(3)通过缩小局部窗口提升算法的运行效率。假设在传统 White 算法中设置局部窗口为 15×15 ,则对每个像素点需要计算其周围 255 个像素点的灰度均值。改进后如果设置局部窗口尺寸为 9×9 ,则每个像素点只需计算其周围 81 个像素点的灰度均值,运行效率提高了 68.24%。但窗口不宜过小,否则容易导致窗口内全是背景或前景,降低算法性能。

2.3 本文校正方法设计

由于光照不均图像灰度分布及局部对比度变化的复杂性,不可能使用单一阈值进行分割处理;单独使用局部阈值算法又会比较耗时,无法满足实时性的需求。实际应用中,光照不均图像中的光照不均匀区域往往只占一小部分,本校正方法对于占居图像大部分的光照均匀区域以及笔画像素数多的阴影区域采用区域 Otsu 算法进行二值化以节省处理时间;其他区域采用本文改进的 White 算法以提取更多的图像细节。

首先,将灰度化后的图像均分为 $N \times N$ (N 取 2~8) 个子图像块。 N 过小不易将光照均匀区域和不均匀区域划分开; N 过大会导致部分图像块内全是前景或全是背景,且分块过细对最终的处理效果也无明显提升,反而会增加处理时间。对于 1944×2592 像素的文档图像, N 取 4 或 5 时能使得在二值化效果和处理速度上同时取得最佳。此外,鉴于处理对象是文档图像,且以 OCR 识别为主要应用目的,研究中没有过多考虑分块边界效应,而是简单的对图像进行区域均分,以降低分块的复杂程度和算法耗时。测试结果表明这种划分方法是有效的,边界效应产生很少,几乎没有对最终的 OCR 识别造成影响。

分块完毕后,遍历子图像块并统计子图像块 $block[i]$ 的 $LowPixel$ 所占比例 $LowPixel_Ratio[i]$ 和 $HighPixel$ 所占比例 $HighPixel_Ratio[i]$,然后根据 $LowPixel_Ratio[i]$ 和 $HighPixel_Ratio[i]$

$Ratio[i]$ 自适应地选取相应的二值化算法。图 5 为本文校正方法的基本流程图。

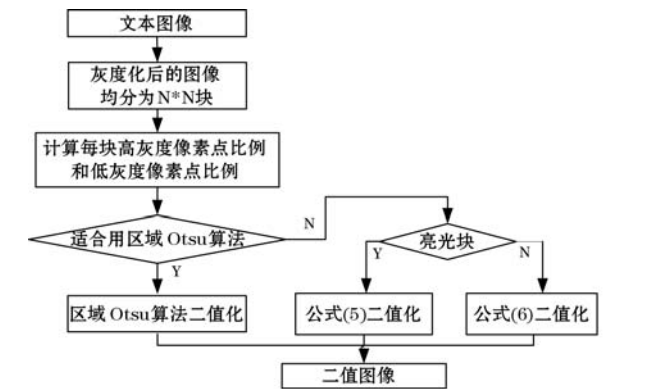


图5 校正方法基本流程图

2.4 算法步骤描述

根据前文论述,本校正方法的详细步骤如下:

(1) 利用公式 $L = 0.299 \times R + 0.587 \times G + 0.114 \times B$ 将原始文档图像灰度化,然后将灰度图像均分为 $N \times N$ 个子图像块,转步骤(2)。

(2) 遍历 $N \times N$ 个子图像块,使用式(7)和式(8)统计子图像块 $block[i]$ 的 $LowPixel_Ratio[i]$ 和 $HighPixel_Ratio[i]$ 。

$$LowPixel_Ratio[i] = \frac{\sum_{(u,v) \in block[i]} \{ 0 \leq f(u,v) \leq 63 \} }{\sum_{(u,v) \in block[i]} \{ 1 \} } \quad (7)$$

$$HighPixel_Ratio[i] = \frac{\sum_{(u,v) \in block[i]} \{ 191 < f(u,v) \leq 255 \} }{\sum_{(u,v) \in block[i]} \{ 1 \} } \quad (8)$$

其中 $f(u,v)$ 为像素点 (u,v) 的灰度值。转步骤(3)。

(3) 若 $LowPixel_Ratio[i] = 0$, 则认为 $block[i]$ 内全是背景,将该块内所有像素点全置为白色像素。否则转步骤(4)。

(4) 如果 $HighPixel_Ratio[i] > Threshold1$ ($Threshold1$ 取 90% ~ 95%), 则判定该块为亮光块,采用式(5)进行二值化。否则转步骤(5)。

(5) 如果 $HighPixel_Ratio[i] < Threshold2$ ($Threshold2$ 取 70% ~ 80%), 并且其 $LowPixel_Ratio[i] < Threshold3$ ($Threshold3$ 取 3% ~ 6%), 则判定该块为笔画像素少的阴影块,采用式(6)进行二值化。否则转步骤(6)。

(6) 对该子图像块采用区域 Otsu 算法进行二值化,转步骤(7)。

(7) 利用四邻域法对以上几步处理后的图像进行去噪,即遍历二值化后的图像,如果某像素点灰度值 $f(i,j) = 0$, 则:

If $f(i-1,j) = 255 \cap f(i+1,j) = 255 \cap f(i,j-1) = 255 \cap f(i,j+1) = 255$
Then $f(i,j) = 255$
Else $f(i,j) = 0$

转步骤(8)

(8) 输出二值图像。

3 实验结果与分析

3.1 测试环境

本方法在 Microsoft Visual Studio2005 环境下采用 C++ 语言实现,测试环境为: Intel (R) Core (TM) 2 Duo CPU P7450 2.13 GHz,内存 2 GB;操作系统为 Windows XP。测试样张根据

实际应用主要从图书和杂志中选取,大小均为 1944×2592 像素,内容类型包括满页文字、图文混排、表格、英文等几种。选取的样张数目为 100 张,包括 50 张局部阴影图片,50 张局部亮光图片。采集图像的设备为数码相机,OCR 采用汉王 OCR2.0 版本。

3.2 二值化效果及算法性能对比

实验用本校正方法与其它四种常用的二值化算法包括 Otsu 算法、Niblack 算法^[17]、Sauvola 算法和 White 算法做了对比分析,各算法中参数及局部窗口的设置主要参考了各算法原文、文献^[4]和文献^[11]等。

图 6 左上方为原图像中含有局部阴影的区域。从实验结果可以看出,全局 Otsu 算法对于阴影区域的处理效果很差;Niblack 算法和 White 算法在处理阴影区域时产生了大量伪影;Sauvola 算法产生的伪影较少,但该算法十分耗时,表 1 中有进一步对比。本校正方法产生的伪影很少,效果比较理想。

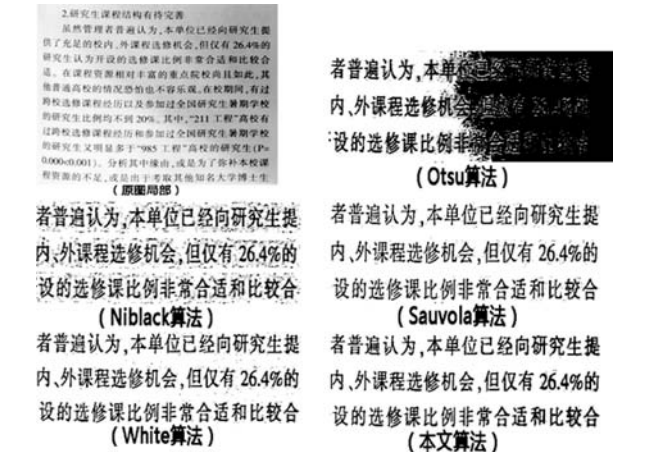


图6 局部阴影图像处理效果对比

图 7 左上方是原图像中的局部亮光区域。可以看出,Niblack 算法在文字周围产生了很多噪声;White 算法产生了大量的断笔;Otsu 算法和 Sauvola 算法的处理效果差强人意,但也有断笔出现,如第 2 行字中的“大”字、“生”字和“习”字等;本校正方法产生的断笔少,效果更优。



图7 局部亮光图像处理效果对比

由于增加了对阴影区域和亮光区域的针对性处理,实验结果表明,本校正方法取得了更为令人满意的二值化效果。

表 1 对比了五种算法的性能,Otsu 算法对阴影区域和亮光区域的处理效果很差,造成 OCR 识别率很低。Niblack 算法容易在字符周围产生大量噪声,识别率也受到严重影响。Sauvola 算法识别率较高,但需要对每个像素点计算局部均值和方差,十分耗时。White 算法对系数 $bias$ 的选取十分敏感,对图像区域亮度变化复杂的情况适应能力不强。本校正方法速度比全局阈

(下转第 196 页)

4 结 语

本文利用人眼视敏度特性选择 Rank 模板,可以一定程度上避免模板选择的盲目性,使得计算机处理结果与人眼视觉效果趋于一致。而基于 LSD 算法中区域生长策略的直线特征检测能在线性的时间内得出亚像素级精度的检测结果,其检测结果快速、准确。此外,本文方法对于晴天、雨天等不同的天气情况下的图像的扣件定位适应性好,定位精度高。其定位出来的扣件图像尺寸较小,不仅缩小了后续扣件识别过程中的计算量,而且大大地减小了图像中的冗余信息对扣件识别的影响。从图 5 中可得,该方法提取的扣件边界相比于图 2 中传统的边界提取算子的扣件轮廓更完整,也为后续的扣件识别过程提供了重要的基础。

参 考 文 献

[1] Hsieh H Y,Chen N,Liao C L. Visual recognition system of elastic rail clips for mass rapid transit systems[C]//ASME/IEEE 2007 Joint Rail Conference and Internal Combustion Engine Division Spring Technical Conference. American Society of Mechanical Engineers, 2007: 319 - 325.

[2] Marino F,Distante A,Mazzeo P L, et al. A real-time visual inspection system for railway maintenance: automatic hexagonal-headed bolts detection[J]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2007, 37(3): 418 - 428.

[3] Deutschl E,Gasser C,Niel A, et al. Defect detection on rail surfaces by a vision based system [C]//Intelligent Vehicles Symposium, 2004 IEEE. IEEE, 2004: 507 - 511.

[4] Xia Y, Xie F, Jiang Z. Broken Railway Fastener Detection Based on Adaboost Algorithm [C]//Optoelectronics and Image Processing (ICOIP), 2010 International Conference on. IEEE, 2010, 1: 313 - 316.

[5] 张洋. 巡道车视频处理系统关键技术的研究[D]. 北京: 北京交通大学, 2008.

[6] Feng Hao, Jiang Zhiguo, Xie Fengying, et al. Automatic Fastener Classification and Defect Detection in Vision-Based Railway Inspection Systems[J]. Instrumentation and Measurement, 2013, 63(4): 877 - 888.

[7] 李永波, 李柏林, 熊鹰. 基于 HOG 特征的铁路扣件状态检测[J]. 传感器与微系统, 2013, 32(10): 110 - 113.

[8] 于方杰. 光照条件变化下立体视觉研究[D]. 青岛: 中国海洋大学, 2013.

[9] 尚婷婷. 基于边缘特征的立体匹配算法研究[D]. 北京: 北京理工大学, 2010.

[10] 杨中雷. 基于眼睛生理的凝视效果模拟[D]. 浙江: 浙江大学, 2011.

[11] Von Gioi R G, Jakubowicz J, Morel J M, et al. LSD: A Fast Line Segment Detector with a False Detection Control[J]. Pattern Analysis and Machine Intelligence, 2010, 32(4): 722 - 732.

[12] Von Gioi R G, Jakubowicz J, Morel J M, et al. LSD: a line segment detector[J]. Image Processing On Line, 2012.

[13] 马姗. 铁路扣件识别的研究[D]. 北京: 北京交通大学, 2013.

(上接第 188 页)

值 Otsu 算法略慢,对于 1444 × 2592 像素的图像相差约 150ms;与性能相对较好的 Sauvola 算法和 White 算法相比,OCR 识别率比 Sauvola 算法提高了 2.5%,比 White 算法提高了 4.8%;耗时比 Sauvola 算法减少了 90.7%,比 White 算法减少了 57.3%。

表 1 校正速度和 OCR 识别率对比

	Otsu 算法	Niblack 算法	Sauvola 算法	White 算法	本文 算法
平均耗时 (ms)	109	2788	2800	609	260
平均识别率 (%)	10.1	43.5	94.8	92.5	97.3

4 结 语

在对不均匀光照文档图像的区域灰度分布和亮度特征深入分析的基础上,本文提出了一种快速的校正方法。相对于单纯地使用局部阈值二值化方法耗时更少;校正后图像的 OCR 识别率得到明显提高。处理过程中无需与用户交互、预设参数少且容易控制,具有较高的应用价值。

目前,对于拍摄的局部光照不均文档图像本方法已能取得较为令人满意的校正效果。但如果要处理阴影(或亮光)面积过大、或者程度过深以致人眼难以辨识文字的图像,本方法还有待完善,以求进一步扩大适用范围。

参 考 文 献

[1] 许新征,丁世飞,史忠植,等. 图像分割的新理论和新方法[J]. 电子学报, 2010, 38(2A): 76 - 82.

[2] 李良华, 罗彬杰. 针对特定目标提取的二值化算法[J]. 计算机科学, 2009, 36(12): 282 - 284.

[3] 胡丽娜. 低质量文档图像的二值化研究[D]. 南京: 南京理工大学, 2012.

[4] 樊汝策, 王庆, 翟正军, 等. 一种改进的针对退化文本图像的二值化方法[J]. 测控技术, 2013, 32(5): 29 - 31.

[5] 罗松, 王俊峰, 唐鹏, 等. 面向条码识读的自适应二值化改进算法[J]. 计算机工程与设计, 2013, 34(4): 1324 - 1329.

[6] 龙建武. 基于 Otsu 的图像阈值分割算法的研究[D]. 长春: 吉林大学, 2011.

[7] 孟立娜, 韩其睿. 一种全局和局部相结合的二值化方法研究[J]. 计算机技术与发展, 2012, 22(11): 116 - 119.

[8] Otsu N. A threshold selection method from gray-level histogram[J]. IEEE Transactions on Systems, Man and Cybernetics, 1979, 9(1): 62 - 66.

[9] White J M, Rohrer G D. Image thresholding for optical character recognition and other applications requiring character image extraction [J]. IBM Journal of Research and Development, 1983, 27(4): 400 - 411.

[10] 刘纪红, 王毓媛. 一种基于自适应阈值的图像二值化算法[C]// 2009 中国控制与决策会议论文集(3): 3958 - 3962.

[11] Sezgin M, Sankur B. Survey over image thresholding techniques and quantitative performance evaluation[J]. Journal of Electronic Imaging, 2004, 13(1): 146 - 165.

[12] Sauvola J, Pietaksinen M. Adaptive document image binarization[J]. Pattern Recognition, 2000, 33(2): 225 - 236.

[13] 梁琳, 何卫平, 雷蕾, 等. 光照不均图像增强方法综述[J]. 计算机应用研究, 2010, 27(5): 1625 - 1628.

[14] 徐奕奕, 刘智琦, 刘琦. 基于文本图像的自适应补偿二值化处理算法[J]. 计算机仿真, 2011, 28(10): 240 - 243.

[15] Chang Y F, Pai Y T, Ruan S J. An efficient thresholding algorithm for degraded document images based on intelligent block detection [C]// IEEE International Conference on Systems, Man and Cybernetics (SMC 2008). 2008: 667 - 672.

[16] 龙建武, 申铨京, 陈海鹏. 基于图像区域的交互式文本图像阈值分割算法[J]. 计算机研究与发展, 2012, 49(7): 1420 - 1431.

[17] Niblack W. An introduction to digital image processing [M]. Prentice-Hall, 1986: 115 - 116.