文章编号:1009-3486(2004)04-0048-05

基于图元识别的OCR文本图像倾斜矫正快速算法

* 1

张秀山[1],吴产乐[2]

(1.海军工程大学 电子工程学院,湖北 武汉 430033;2.武汉大学 计算机学院,湖北 武汉 430072)

摘 要:提出了一种基于文本图元识别以跟踪字符中心线的高精度矫正OCR图像倾斜的快速算法,该算法思想虽然简单,却具有高效和高精度的特点,同时还具有高可靠性和良好的抗噪特征.实验表明,该方法完全满足实时应用的需要.

关键词:光学字符识别;倾斜矫正;图元识别;图元标准包围盒

中图分类号:TP391.4 **文献标识码**:A

A rapid algorithm to OCR image slant correction based on primitive recognition

ZHANG Xiu-shan[1], WU Chan-le[2]

(1.Electronic Eng. College, Naval Univ. of Engineering, Wuhan 430033, China;
2.Computer School, Wuhan University, Wuhan 430072, China)

Abstract: This paper proposes a rapid algorithm to OCR image slant correction, which applies the graphic primitive recognition technique and then keeps track of the character center-line. Though being simple, the algorithm is so efficient, accurate, and reliable that it can be used in real-ime applications.

Key words: optical character recognition; slant correction; graphic primitive recognition; standard bounding box for a graphic primitive

扫描的文本图像的倾斜度往往会影响字符的识别率,因此需要使用某种软件方法对图像进行矫正.瞿洋等为此提出一种应用Hough变换[1,2]进行图像倾斜矫正的方法.利用Hough变换发现图像中的直线模式的方法的突出缺点是计算量很大,尽管文献[1]采用分辨率层次模型以降低计算量,但算法与不采用分层模型的时间复杂度是等量级的,它只是在常数意义上较优.假设Hough变换使用的累积矩阵$\boldsymbol{A}(\rho,\theta)$的大小为$m\times n$,则显然算法总的时间复杂度为$O(mnS)$,这里$S$是整幅图像的以像素为单位的像素面积.另外,Hough变换应用在像文字等这种"粗直线"图像模式识别,若不考虑水平边缘提取时,图像倾斜角度的精度取决于门限γ的选取,而门限γ的选取应依据字体大小以及字间距和行间距等排版参数.事实上,文献[1]给出的门限γ的动态范围很大,一般需要人工干预.

1 算法思想

一种简单的避免大量统计运算的跟踪直线的方法是跟踪同行同字号字符体的中心线,一旦确定了中心线就可通过简单的几何运算来确定光学字符识别(Optical Character Recognition, OCR)图像的倾

* **收稿日期**:2004-02-02;**修订日期**:2004-04-02
基金项目:海军工程大学科学研究基金资助项目(HGDJJ04034)
作者简介:张秀山(1968-),男,讲师,博士生.

斜角.确定中心线需要确定至少 2 个字符体的中心位置,这就需要识别同一行中 2 个或多个字符体,并检查其中是否有等高字符体.识别字符体的方法可以有多种,一种有效的识别方法是采用广度优先的图元识别技术.确定中心线的方法可以直接通过两端点求得,也可利用直线拟合的方法以提高精度.这种算法思想不仅简单高效,而且图元识别的结果还可进一步在图元分割时结合和利用.

我们知道,中文字符不会越过其字符体,但带桉[3]的小写西文字符可以越过字符体;虽然同一字体字号的小写英文字符逻辑上定义基线和顶线相同,但字符的实际高度可能并不相同,如这里的"a"和"g".为了确定字符的实际高度,我们需要引入图元(字符)标准包围盒的定义.

定义 1　一个四连通区域即是一个图元;包围一个图元的最小水平方向矩形称为该图元的标准包围盒;相似地,包围一个字符的最小水平方向矩形称为该字符的标准包围盒.

一个图元(字符)p 的标准包围盒 $B(p)$可用 2 个设备坐标序偶表示,一个是左上角坐标,一个是右下角坐标,此处记为$<(\text{left},\text{top}),(\text{right},\text{bottom})>$.

定义 2　设 p_1 和 p_2 是 2 个图元,它们的标准包围盒分别为 $B(p_1)=<(l_1,t_1),(r_1,b_1)>$和 $B(p_2)=<(l_2,t_2),(r_2,b_2)>$,定义图元标准包围盒的"并"运算如下:

$$B(p_1)\cup B(p_2)=<(\min(l_1,l_2),\min(t_1,t_2)),(\max(r_1,r_2),\max(b_1,b_2))>=B(p_1\cup p_2)$$

由上述标准包围盒的"并"运算的定义,可以立即得到如下定理.

定理 1　字符的标准包围盒是该字符所包含的所有图元的标准包围盒的"并".即 $B(\cup p_i)=\cup B(p_i)(i=1,\cdots,n)$,其中 n 为字符所包含的图元数.

证明:由定义 2 对 i 进行归纳,即可得证.

所有的大写英文字母和除 i,j 以外的所有小写英文字母均是单个图元构成的;而许多汉字都是多个图元构成的.因此,对汉字来讲,要识别字符体还需要对图元进行"并"运算处理.

定理 2　设 P_1 和 P_2 是识别出的任意 2 个同一行中的等高图元,且

$$B(p_1)=<(l_1,t_1),(r_1,b_1)>$$
$$B(p_2)=<(l_2,t_2),(r_2,b_2)>$$

则 P_1P_2 的中心线的倾斜角为:

$$\alpha_{p_1p_2}=\arctan\left(\left(\frac{b_2+t_2}{2}-\frac{b_1+t_1}{2}\right)/\left(\frac{r_2+l_2}{2}-\frac{r_1+l_1}{2}\right)\right)\tag{1}$$

证明从略.

(1)式计算的倾斜角度即可作为图像的倾斜角.显然,为了保证倾斜角识别的精度,要求这 2 个等高图元要相距最远.一般地,由于上式分母远大于分子,故(1)式还可表示为:

$$\alpha_{p_1p_2}\approx\left(\frac{b_2+t_2}{2}-\frac{b_1+t_1}{2}\right)/\left(\frac{r_2+l_2}{2}-\frac{r_1+l_1}{2}\right)\tag{2}$$

应该注意的是,倾斜的字符标准包围盒和未倾斜的该字符标准包围盒往往并不相同,面积可能变大也可能变小,甚至其中心点的水平和垂直位置都会发生偏移.但如下定理 3 可以确保一个实心矩形无论如何倾斜,它的标准包围盒的中心点不会偏离原中心点.中西文字符的笔画连贯性和像素分布正态性有利于保证中心点不会过度偏移.

定理 3　任意方向矩形的标准包围盒的中心点穿过原矩形中心点.

证明:如图 1 所示,$ABCD$ 是任意方向的矩形,$EFGH$ 是该矩形的标准包围盒,O 是 EG 与 BD 的交点.

在 RT$\triangle ADE$ 和 RT$\triangle CBG$ 中,

因为 $AD=CB$,所以 RT$\triangle ADE\cong$ RT$\triangle CBG\Rightarrow BG=DE$

因为 $BG\parallel DE\Rightarrow\angle OBG=\angle ODE$

又因为 $\angle DOE=\angle BOG$

$\Rightarrow\triangle OBG\cong\triangle ODE$

$\Rightarrow O$ 是 EG 和 BD 的中点.

同理可证,O 也是 FH 和 AC 的中点.

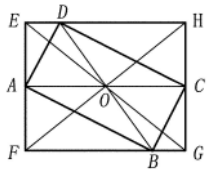

图 1　定理 3 示意图

O 即是 $ABCD$ 的中心点,也是 $EFGH$ 的中心点.于是,定理得证.

2 算法描述

基于以上思想,可以给出图像倾斜矫正算法的一般流程(见图 2).

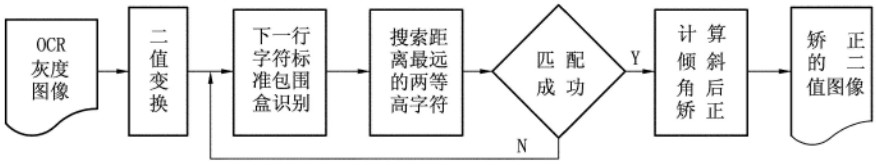

图 2 OCR 图像倾斜矫正算法的一般流程

字符的标准包围盒识别可根据定理 1 通过图元标准包围盒的"并"运算求得.于是,需要先实现图元标准包围盒的识别算法.下面用伪 C++代码[4]给出该算法的较为详细的算法描述.

PrimitiveBoxRecognition 函数参数 bmp 是一幅灰度位图,c 为颜色阈值,current 为当前像素起点,函数中 box 为输出的图元标准包围盒.算法中引用的未定义函数可从函数名字及其参数理解其功能及实现.

算法 1 图元标准包围盒识别算法

```
CBox PrimitiveBoxRecognition (BMP& bmp, COLOR c, POINT current)
(1) MarkPixel(current) //标记 current 属于当前图元
(2) InsertQueue(current) //并将 current 像素插入待处理像素队列
(3) box.topleft= box.bottomright=current //然后初始化包围盒 box
(4) //识别包含 current 像素点的图元标准包围盒
(5) while !IsQueueEmpty() //像素队列非空
(6)     do current=ExtractQueue() //取队首像素
(7)       for each dir in {UP, DOWN, LEFT, RIGHT} //四连通图元
(8)         do neighbor=GetPixel(dir, current) //取四连通像素
(9)           if IsPixel(current, bmp, c) and !IsPixelMarked(current, bmp)
(10)              then MarkPixel(neighbor)
(11)                   InsertQueue(neighbor)
(12)                   UpdateBox(box) //更新 box
(13)  return box
```

计算字符的标准包围盒需要首先识别同一字符行中的所有图元,然后按水平方向坐标进行排序,接着就可进行"并"运算以求得各字符的已排好序的标准包围盒;最后,进行等高字符匹配,搜索最远的 2 个等高字符,并计算倾斜角.下面的算法将以上各步合并起来,给出总的 OCR 图像矫正算法.

算法 2 OCR 图像矫正算法

```
SlantCorrection(BMP& bmp, COLOR c)
(1) index=0 //初始化
(2) for current=(0,0) to (bmp.width-1, bmp.height-1) //从左到右从上到下扫描图像
(3) //搜索从当前像素开始的第一个未被标记的像素并识别其标准包围盒
(4)     do if IsPixel (current, bmp, c) and !IsPixelMarked(current,bmp)
(5)        then box[index]=PrimitiveBoxRecognition (bmp, c, current)
(6)          if index =0
(7)            then linebox= box[index++] //初始化 linebox
```

```
(8)                continue
(9)          else if IsInSameLine(linebox , box[index])
(10)                then linebox+=box[index++] //"并"运算
(11)                     continue
(12)    if index>1
(13)      then SortBox(box ,index) //排序 ,这里 index 为当前字符行图元总数
(14)           MergeBox(box ,index)//合并各字符的各图元 ,index 返回字符总数
(15)    //SearchSameHeightChar 搜索等高字符 ,返回距离最远的两个等高的 box 下标
(16)           if SearchSameHeightChar (box ,index ,i ,j) //搜索成功
(17)           then return CaculateSlantAngle(box , i , j) //计算倾斜角
(18)           else index=0 //扫描下一行
(19)  return "FAILURE !" //图元太少或未匹配成功 !
```

其中:index 为行中图元/字符个数;linebox 为字符行包围盒.

定理 4　SlantCorrection 算法在最好情况下的时间复杂度为 $O(L)+O(n^2)$,这里 L 是单行字符的像素面积,n 为单行字符数.

证明:显然,算法 PrimitiveBoxRecognition 的时间复杂度为相应图元 i 的标记次数即 $\Theta(C_i)$,其中 C_i 为相应图元 i 的像素面积.

SlantCorrection 算法的最好情况是首行匹配,第(1)~(11)行的算法时间应与该行的所有图元面积成正比,即为 $\Theta(\sum C_i)$;第(13)行可采用堆排序,其时间复杂度为 $\Theta(n\log\ n)$,其中 n 为行图元总数;第(14)行为 $\Theta(m)$,其中 m 是字符图元总数,显然 $m\leqslant n$;第(16)行为 $O(m^2)$.SlantCorrection 算法在最好情况下的时间即是上述各行时间的总和.即:

$$T_{首行}=\Theta(\sum C_i)+\Theta(n\log\ n)+\Theta(m)+O(m^2)=O(L)+O(n^2)$$

证毕.

一般情况下,算法能够在首行匹配成功.

推论 1　若 OCR 图像的幅面宽度保持恒定,则 SlantCorrection 算法在最好情况下的时间复杂度为 $O(w)$,其中 w 是单个字符的平均高度(宽度).

证明从略.

应该注意到算法中引入了一个行包围盒 linebox,它是由同行中的所有图元的标准包围盒通过"并"运算得到的.它具有统计学特征,不仅有助于同行中的所有图元的可靠识别,进而确保字符标准包围盒的可靠识别,而且也提高了倾斜角发现的准确性和可靠性.

另外,同行中会有些随机噪声图元出现,它们的产生不是由于得不到正确的匹配而被忽略,就是被 MergeBox 过程所合并,但不管它们的产生是否影响字符标准包围盒中心点,一般都不会影响最终倾斜角计算结果.因而,算法也具有良好的抗噪特征.

3　实现与测试

作者采用 visual C++在Celeron466机器环境下实现了上述算法.图3是经过裁剪的原图,经二

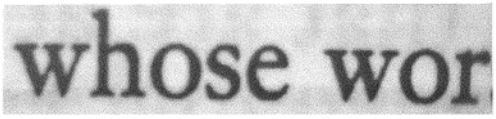

图 3　二值变换前的原始 OCR 文本图像

图 4　二值变换后的黑白 OCR 文本图像

值变换后得到图 4,其二值映射阈值为 $R,G,B=108$.经过对多幅不同字号的大小为 720×576 的未裁剪

原图进行测试,不但都成功发现图像的倾斜角,而且程序响应很快,一般在 100 ms 以内,完全满足实时检测的需要.如图 5 所示是图像幅面宽度恒定为 720 像素时的算法运行时间曲线.由图 5 可知,该曲线确实验证了推论 1.经程序计算和跟踪得到按图中字符出现顺序的 5 个字符"woewo"的各图元识别标准包围盒分别为<(83,205),(171,263)>,<(243,207),(302,266)>,<(355,209),(403,268)>,<(446,213),(532,270)>和<(533,213),(590,273)>.可分别利用 2 个 w、2 个 o 或第一个单词的首尾字符即 w 和 e 按(1)式计算得到图像的倾斜角 α_{ww}、α_{oo} 和 α_{we}:

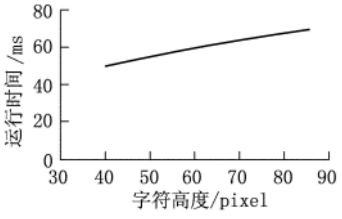

图 5 运行时间曲线

α_{ww} = arctan(7.5/362) = 1.187°,精度为 arctan(0.5/362) = 0.079°

α_{oo} = arctan(6.5/289) = 1.288°,精度为 arctan(0.5/289) = 0.099°

α_{we} = arctan(4.5/252) = 1.023°,精度为 arctan(0.5/252) = 0.114°

以上精度计算假定同行的 2 个等高字符的图元识别标准包围盒的中心垂直偏差为 1 个像素的情形.以上精度都足以满足 OCR 字符识别的需要.

用商用软件汉王 OCR 5.0 对原灰度图像进行倾斜矫正,它测得倾斜角度为 0°,即未检测到倾斜.由于应用 Hough 变换计算量较大(文献[1]给出 2 000×3 000 像素的图像在 Pentium 133 上测试时间为 2 s 左右),而且需要一些经验参数,因而也未能去实现和详细比较.

4 结 论

本文在充分分析 OCR 扫描文本特征基础上,提出了基于图元(字符体)识别的跟踪字符中心线以提取图像倾斜角的方法.该方法不仅算法思想简单,实时高效,而且具有较高的精度和可靠性,以及良好的抗噪特征.

参考文献:

[1] 瞿 洋,杨利平.Hough 变换 OCR 图像倾斜矫正方法[J].中国图像图形学报,2001,6(2):178—181.

[2] Ballard D H. Generalizing the Hough transform to detect arbitrary shapes [J]. Pattern Recongnition, 1981, 13(4): 111—122.

[3] Hearn D, Baker M P. Computer Graphics [M]. New York: Prentice Hall Press, 1997.

[4] Cormen T H, Leiserson C E, Rivest R L, *et al*. Introduction to Algorithms [M]. Massachusetts: MIT Press, 2001.