一种stroke滤波器文字分割算法

石振刚[1,2] 高立群[2]
(沈阳理工大学信息科学与工程学院 沈阳 110168)[1] (东北大学信息学院 沈阳 110004)[2]

摘 要 为解决复杂背景中准确地进行文字分割的问题,提出了一种应用stroke滤波器进行文本分割的新方法。首先进行stroke滤波器的合理设计,并应用所设计的stroke滤波器来判别文本的彩色极性,得到初次分割的二值图。然后进行基于区域生长的文字分割。最后,应用OCR(optical character recognition)模块提高文本分割的整体性能。将提出的算法与其他算法进行了比较,结果表明,所提算法更为有效。

关键词 文本分割,复杂背景,OCR,stroke滤波器
中图法分类号 TP391.43 **文献标识码** A

New Algorithm for Text Segmentation Based on Stroke Filter

SHI Zhen-gang[1,2] GAO Li-qun[2]
(College of Information Science and Engineering, Shenyang Ligong University, Shenyang 110168, China)[1]
(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China)[2]

Abstract In order to solve segment text accurately and robustly from a complex background, this paper proposed a new algorithm for text segmentation in images by using a stroke filter. First, we described the stroke filter briefly based on local region analysis. Second, the determination of text color polarity and local region growing procedures were performed based on the response of the stroke filter. Finally, the feedback procedure by the recognition score from an optical character recognition (OCR) module was used to improve the performance of text segmentation. The proposed algorithm was compared with other algorithms. The experimental results demonstrate that the proposed algorithm obtaines satisfactory results.

Keywords Text segmentation, Complex background, Optical character recognition(OCR), Stroke filter

文本分割是指在一个书面文档或视频序列中识别具有独立意义的单元之间的边界,其分割对象可以是视频流、网络动态数据或者书面静态文本。这种预处理技术在很多领域比如信息提取、文摘生成、文本解析以及文本导航等都有着极为重要的应用。目前,在简单背景下进行文本分割已经是一项较为成熟的技术,而如何在较为复杂的背景中进行文本分割还没有一个成熟的方法。因此,在复杂背景中准确地进行文本分割就成为当前文本分割中的热点问题[1,2]。

1 文本分割技术

文本分割主要有两种方法。第一种方法主要是根据分割目标与背景的差异进行分割,例如全局和局部的阈值分割方法及Otsu自动阈值分割方法等。通常,这类方法较为简单,算法运行速度快。但当分割目标与背景较为接近时,这类分割方法往往不能够进行准确的分割。第二种方法主要是根据像素的相似性进行分割,例如使用分裂合并的方法进行分割,使用边缘检测的方法进行分割及使用分水岭的方法进行分割等[3]。但这些方法由于在进行文本分割时对待分割文本的形状作了一些人为的规定,因此在实际应用中都具有不稳定性。

上述两种文本分割方法存在的主要问题是对待分割文本的色彩、大小、字体以及复杂背影较为敏感。针对这些问题,本文在充分考虑到文本固有特性的基础上,提出了一种应用stroke滤波器进行文本分割的新方法。本文第2节介绍stroke滤波器的设计;第3节详述本文的分割算法;第4节给出测试手段及实验结果,并就实验结果进行讨论;最后作出总结。

2 stroke滤波器设计

stroke定义为用于文本分割的一段直线或一段弧,而图像中的文本是由一个或几个stroke所组成的。stroke滤波器是基于局域分析法进行设计的,我们所设计的stroke滤波器如图1所示。

根据图1所示stroke滤波器来计算源图像中像素点滤波后的值。设源图像中任一点(x,y)为滤波器的中心点,其周围有3个矩形区域。图1中所示区域1表示中心区域,区域2和区域3为中心区域的两个长边侧面区域。这3个局部区域的方向和大小是由参数α和d决定的,其中d是由所要分割文本的先验知识所决定的。其他参数有如下关系:$d_1=d_2=d/2, w=2d$。

到稿日期:2009-03-05 返修日期:2009-05-04 本文受兵器预研支撑基金项目(62301110113)资助。
石振刚(1971-),男,博士生,主要研究方向为图像处理等,E-mail:szg888888@sina.com;高立群(1949-),男,教授,主要研究方向为图像处理等。

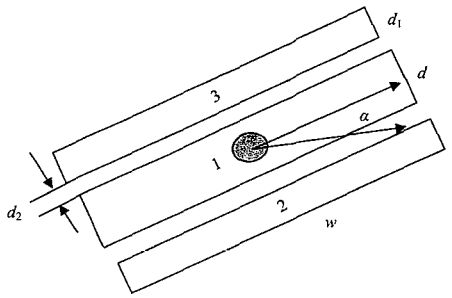


图 1　stroke 滤波器

根据所设计的 stroke 滤波器，定义源图像中像素点(x, y)滤波后具有亮(R^B)和暗(R^D)两类属性值，计算公式如下：

$$R^B_{\alpha,d}(x,y)=\frac{\mu_1-\mu_2+\mu_1-\mu_3-|\mu_2-\mu_3|}{\sigma} \tag{1}$$

$$R^D_{\alpha,d}(x,y)=\frac{\mu_2-\mu_1+\mu_3-\mu_1-|\mu_2-\mu_3|}{\sigma} \tag{2}$$

其中，μ_i 表示区域 i 的亮度均值，$i=1,2,3$。参数 σ 表示区域 1 的亮度标准差。对于滤波后所得到的属性值，提取式(3)—式(8)所示的特征值$(R^B, O^B, S^B, R^D, O^D, S^D)$作为进行复杂背景下文本分割的基础。

$$R^B(x,y)=\max_{(\alpha,d)}R^B_{\alpha,d}(x,y) \tag{3}$$

$$O^B(x,y)=\arg_{(\alpha)}\max R^B_{\alpha,d}(x,y) \tag{4}$$

$$S^B(x,y)=\arg_{(d)}\max R^B_{\alpha,d}(x,y) \tag{5}$$

$$R^D(x,y)=\max_{(\alpha,d)}R^D_{\alpha,d}(x,y) \tag{6}$$

$$O^D(x,y)=\arg_{(\alpha)}\max R^D_{\alpha,d}(x,y) \tag{7}$$

$$S^D(x,y)=\arg_{(d)}\max R^D_{\alpha,d}(x,y) \tag{8}$$

其中，R, O 和 S 分别代表 stroke 滤波器的响应、方向和尺度值。

接下来，在第 3 节详述基于上面所设计的 stroke 滤波器进行复杂背景下文本分割的算法。

3　基于 stroke 滤波器的文本分割算法

3.1　基本的文本分割

文本彩色极性的判定对于正确地进行文本分割是非常重要的[4]。为了自动地判别文本的彩色极性，我们使用了如下两个特征。

用第 2 节所设计的 stroke 滤波器对源像素点进行滤波，得到亮和暗两类属性值 R^B 和 R^D。然后，根据 R^B 和 R^D 计算出判别文本彩色极性的两个特征值：F_R 和 F_E。其中 F_R 代表像素点亮属性值总和与像素点暗属性值总和的比率，由式(9)计算得出：

$$F_R=\frac{\sum\limits_{(x,y)}R^{(B)}(x,y)}{\sum\limits_{(x,y)}R^{(D)}(x,y)} \tag{9}$$

F_E 代表像素点亮属性边缘点数量与像素点暗属性边缘点数量的比率，由式(10)计算得出：

$$F_E=\frac{N^{(B)}}{N^{(D)}} \tag{10}$$

其中，$N^{(B)}$ 和 $N^{(D)}$ 分别代表像素点亮属性边缘点数量和像素点暗属性边缘点数量。

最后，利用带有径向基核的支持向量基分类器进行分类[5,6]，得到基本的文本分割结果。

3.2　局部区域增长

由 3.1 节得到了基本的文本分割结果。但是这个分割结果丢失了许多文本像素。对于准确的文本分割来说，应该恢复这些丢失的文本像素。为此，对 3.1 节所得到的基本文本分割结果执行局部区域增长程序。

首先，根据 3.1 节所得到的基本文本分割结果以及原始文本图像估计全局概率密度函数(PDF)。然后，若一个非文本像素满足如下 3 个条件，则将其恢复为文本像素。

条件 1　在此非文本像素的 3×3 邻域内，若文本像素的个数大于 3。

条件 2　此非文本像素的亮度概率值 $Pr(s)$ 大于阈值 Th1。

条件 3　此非文本像素与相邻像素的亮度值之差大于阈值 Th2。

通常，Th1 和 Th2 的值取为 0.16 和 32，$Pr(s)$ 由式(11)计算得出[7]：

$$Pr(s)=\mathrm{PDF}(s) \tag{11}$$

直到没有非文本像素要被变为文本像素后，局部区域增长过程才结束。

将本文所用局部区域增长算法总结如下。

输入：I—3.1 节所得到的基本文本分割结果；

　　S—源文本图像。

Step1　根据 I 和 S 计算彩色文本的全局概率密度函数(PDF)。

Step2　对于 I 中每一个白色像素(非文本像素)，若其 3×3 邻域内的白色像素数大于 3，则转到 Step3，否则进行下一个像素的 Step2 判断。

Step3　对于每个黑色像素(文本像素)，若其在 3×3 邻域内与相邻文本相似且具有较高的概率密度函数值，则将其标注为文本像素。重复 Step2 和 Step3，直到没有像素改变为止。

输出：文本分割结果。

3.3　文本分割结果修正

由 3.2 节所得的文本分割结果经实验得到的错误率大约在 7%～9%，其主要原因在于没有考虑到不同字体中不同字符的相似度不同。为了提高分割结果的准确率，增加了 OCR 模块作为最后修正分割结果的附加模块。

通常，不同字体中不同字符的相似度不同，对字体识别所起作用的大小必然也是随机变化的。为了能够反映这种变化，每个字符 h 的第 i 个字体识别模板中都包含一个 n 维向量：

$$\alpha_{hi}=[p(\theta_{hi}|w_1),p(\theta_{hi}|w_2),\cdots,p(\theta_{hi}|w_n) \tag{12}$$

其中，n 为待识别字体数目，θ_{hi} 表示具体识别模板，$p(\theta_{hi}|w_j)$ 表示输入字符 h 属于字体 w_j 时匹配模板为 θ_{hi} 的概率。$p(\theta_{hi}|w_j)$需要通过对实际样本的学习获得。同其它大多数基于样本的学习方法一样，训练样本越多，所得到的结果越逼近真正分布。

设用于字体识别的字符集合为 $h_a(a=1,2,\cdots,m)$，与之匹配的字体识别模板分别记为 $\theta_{h_a i_a}$，则根据贝叶斯公式即可得出字体识别结果为 w_j 的概率为：

$$p(w_j|\theta_{h_1 i_1},\theta_{h_2 i_2},\cdots,\theta_{h_m i_m})=\frac{p(\theta_{h_1 i_1},\theta_{h_2 i_2},\cdots,\theta_{h_m i_m},w_j)}{p(\theta_{h_1 i_1},\theta_{h_2 i_2},\cdots,\theta_{h_m i_m})} \tag{13}$$

为保证能够以最大的概率得到正确的结果，应该判断这

些字符所属字体为：

$$w_c=\arg\max_{r=1,\cdots,n}(p(w_r|\theta_{h_1 i_1},\theta_{h_2 i_2},\cdots,\theta_{h_m i_m})) \quad (14)$$

通过增加用于字体识别的字符数目 m，可以提高字体识别可信度。只有识别可信度高于预先定义的期望可信度时，才返回字体识别结果。

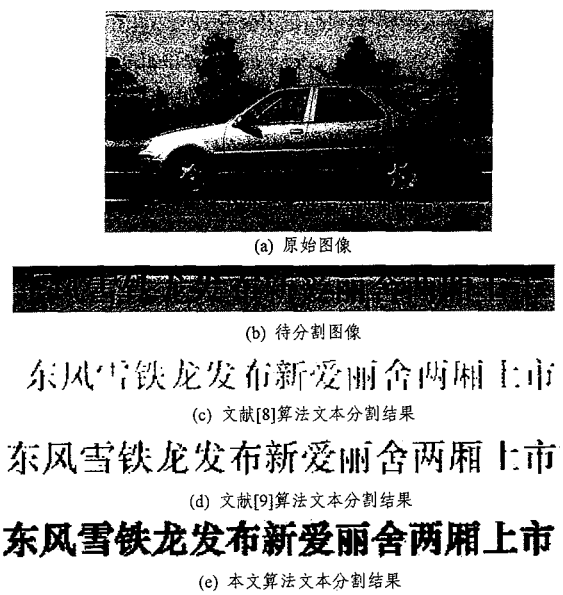
4 实验与分析

本节将对本文所提算法的性能进行仿真实验和分析。实验主要分为两部分：第一部分用本文算法对中文和英文文本进行分割，图 2 显示了分割结果。从图中可以看到，对于不同语言的文本，本文算法都有较好的分割效果。这表明本文算法对于不同语言具有很好的适应性。

(a) 中文原始图像　(b) 中文待分割文本图像　(c) 中文文本分割结果

(d) 英文原始图像　(b) 英文待分割文本图像　(c) 英文文本分割结果

图 2　中英文文本分割结果



(a) 原始图像　(b) 待分割图像　(c) 文献[8]算法文本分割结果　(d) 文献[9]算法文本分割结果　(e) 本文算法文本分割结果

图 3　不同算法文本分割结果比较

在实验的第二部分，对本文所提算法与文献[8,9]所提算法的分割效果从主观和客观上进行了比较。主观上，图 3 显示了各种算法的分割结果，从中可以看到本文的分割效果是最好的。客观上，选取分割准确率（计算公式见式(15)）与算法执行速度这两个指标进行比照，结果如表 1 所列。从表 1 中可以看出，本文所提算法的分割准确率均高于所对比的 3 种文本分割算法。但从执行速度上来看，本文算法较文献[8]的 Otsu 方法慢，与文献[9]的 Song 方法相当。主要原因在于本文算法为提高文本分割的准确率执行了将 OCR 模块作为最后修正分割结果的附加模块，这也是本文算法在今后所要改进和提高的地方。

$$\text{分割准确率}=\frac{\text{文本中正确分割的字符数}}{\text{文本中的字符总数}} \quad (15)$$

表 1　不同算法文本分割准确率与执行速度比较

方法	分割准确率	算法执行时间
文献[8]算法	86.3%	9.8ms
文献[9]算法	90.5%	13.3ms
本文算法	96.8%	13.7ms

结束语　本文提出了一种应用 stroke 滤波器进行文本分割的新方法。该算法通过合理地构造 stroke 滤波器，使其在进行文本分割时能够更好地反映出待分割文本的本质特征。为消除 stroke 滤波器分割文本时所产生的过分割现象，本文算法对通过 stroke 滤波器所分割的文本执行了局部区域成长程序。最后，通过执行 OCR 附加修正模块，使得复杂背景下文本的分割准确率达到了 97%左右。

但是，由于本文所提算法具有较高的计算复杂度，因此算法执行速度较其他常规方法略慢。为解决此问题，下一步将深入研究在保证文本分割准确率的前提下，改进算法有关模块，以降低算法计算复杂度。另外，在未来的工作中，将考虑如何把基于 stroke 滤波器的文本分割技术应用于文档摘要及信息检索等系统中。

参 考 文 献

[1] Dimitrova N, Zhang H J, Shahraray B, et al. Applications of video content analysis and retrieval[J]. IEEE Multimedia, 2002, 9(3): 43-55

[2] Wang Y, Liu Y, Huang J C. Multimedia content analysis using both audio and visual clues[J]. IEEE Signal Process, 2000, 17(6): 12-36

[3] Jung K, Kim K I, Jain A K. Text information extraction in images and video: a survey[J]. Pattern Recognition, 2004, 37(5): 977-997

[4] Tseng Y H, Lee H J. Recognition - based handwritten chinese character segmentation using a probabilistic Viterbi algorithm [J]. Pattern Recognition, 1999, 20(9): 791-806

[5] 陈东，刘希玉. 基于支持向量基的条码分类研究[J]. 山东师范大学学报：自然科学版，2007，22(4)：24-26

[6] 钟将，温罗生，冯永，等. 基于近似支持向量机的 Web 文本分类研究[J]. 计算机科学，2008，35(3)：167-169

[7] Tao D, Li X, Wu X, et al. General tensor discriminant analysis and Gabor features for gait recognition[J]. IEEE Trans. Pattern Analysis, 2007, 29(10): 1700-1715

[8] Otsu N. A threshold selection method from gray-scale histogram [J]. IEEE Trans. Syst. Man Cybern, 1979, 9: 62-66

[9] Song J, Cai M, Lyu M R. A robust statistic method for classifying color polarity of video text[C]// Proceedings of the IEEE International Conference on Acoustics. 2003: 581-584