DOI:10.3979/j.issn.1673-825X.2019.06.019

基于特征向量和笔顺编码的字形相似算法研究

祁俊辉,龙　华,邵玉斌,杜庆治
(昆明理工大学 信息工程与自动化学院,昆明 650000)

摘　要:为充分利用汉字结构、轮廓、笔画、书写顺序等特征识别相似汉字,提出基于特征向量和笔顺编码的字形相似算法,用以解决形近字检索中准确度不高的问题。算法采用图像处理方法及五笔编码规则将汉字转化为特征向量形式和笔顺编码字符串,引入二值化差值算法和改进后的 Jaro-Winkler Distance 算法分别对其进行相似度计算,2 个相似度分别从不同方面反映汉字的相似程度,吸取 2 种方法的优势对其进行融合,得到最终字形相似度。实验结果表明,该算法在字形检索中较 3 元组递归算法准确率提高 27.8%,较模板匹配算法、结构方法、神经网络算法执行效率平均提高约 66.7%,该算法不仅可以有效解决形近字检索中的准确性问题,同时效率也得以优化。

关键词:特征向量;笔顺编码;差值算法;形近字检索
中图分类号:TP391.1　　**文献标志码**:A　　**文章编号**:1673-825X(2019)06-0885-07

Research on chinese character similarity algorithm based on eigenvector and stroke coding

QI Junhui, LONG Hua, SHAO Yubin, DU Qingzhi
(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650000, P. R. China)

Abstract: In order to make full use of Chinese characters structure, contour, stroke, writing order and other features to identify similar Chinese characters, a glyph similarity algorithm based on feature vector and stroke order coding is proposed to solve the problem of low accuracy in near word search. The algorithm uses image processing method and five stroke encoding rule to convert Chinese characters into feature vector form and stroke order coded string. Also the binary difference algorithm and the improved Jaro Winkler Distance algorithm are used to calculate the similarity. Two similarities are obtained. The degree of similarity of Chinese characters is reflected from different aspects, and the advantages of the two methods are combined to obtain the final glyph similarity. The experimental results show that the algorithm improves the accuracy of the triad recursive algorithm in glyph retrieval by about 27.8%, which is about 66.7% higher than the template matching algorithm, structure method and neural network algorithm. It proves that the algorithm can not only effectively solve the accuracy problem of the near word search, the efficiency is also optimized.

Keywords: eigenvector; stroke code; difference algorithm; near word search

0　引　言

作为象形文字的中文,拥有较多的形近字,从而导致在进行文字信息处理时常被错识。目前,印刷体识别技术已经有较高的准确度[1]。但对图片中形近字的字形仍存在识别率不高[2]的问题。这是

收稿日期:2018-05-14　**修订日期**:2019-10-11　**通讯作者**:龙　华　1670931890@qq.com
基金项目:国家自然科学基金(61761025)
Foundation Item: The National Natural Science Foundation of China (61761025)

因为很多汉字的字形差异非常细微,不容易区分,如“已”和“己”、“乌”和“鸟”等。因此,如何使识别系统利用这些细微差别将相似汉字准确区分,是目前文字处理中研究的热点,同时也是解决中文信息处理领域其它问题的前提[3-4]。

针对形近字的识别,林民等[5-7]利用计算机技术获取字库的笔画轮廓数据,为汉字相似研究奠定了基础,且后期针对复合字进一步完善了汉字部件轮廓库。栗青生等[8]将汉字笔画抽象为有向图形式,实现对汉字字形的部件描述,建立了动态汉字字形描述库。赵健等[9]对汉字的端点、折角点、交叉点等特征提取进行创新性的研究,但对字形复杂的汉字效果并不理想。胡浩等[10]结合汉字的构词和拼音属性,将其映射为32维空间向量形式,便于计算短文本的相似性。宋柔等[11]将汉字分为独体字和复合字分别进行讨论,取得较好的成果。孙华等[12]对5种汉字识别方法进行了分析并阐述了现阶段汉字识别算法中存在的问题。刘波等[13]将传统光学文字识别(optical character recognition,OCR)技术和文字图像结合,但识别率不高且所对比的汉字模板库较小。刘斌等[14]对含有噪声的汉字图片进行了矩阵分析,但只研究了单一特征的汉字分类。文献[15-16]对各种字符串相似算法,如Levenshtein Distance算法与Greeay String Tiling算法等进行了比较,并将传统的Jaro-Winkler Distance算法与Levenshtein Distance算法进行融合,从而提高字符串相似判定。

以上文献大多采用汉字组合结构和笔画对汉字部件进行描述,进而通过编辑距离等算法计算其字形相似度,亦或使用图像处理的方法进行相似度的计算。但由于汉字存在种类繁多、结构类型复杂等因素,目前没有较为完整的汉字结构库,致使在实现字形相似识别中存在困难;其次,将汉字描述为数学表达式形式后,其相似度的计算也需要进一步研究。

本文的主要工作包括对汉字字形的形式化结构进行描述,将汉字描述为特征向量和笔顺编码形式,通过二值化差值算法计算基于特征向量的字形相似度,通过改进后的Jaro-Winkler Distance算法计算基于笔顺编码的字形相似度,2个相似度分别从不同方面反映了汉字的相似程度,最后再将计算所得的2个相似度进行融合,得到最终相似度。

1 汉字形式化结构描述

在汉字计算机编码标准中,编码方式为Unicode的中日韩统一表意文字基本字符集收录汉字共20 902个(Unicode码为4E00~9FA5),以20 902个汉字作为数据源,对其进行形式化结构描述。

1.1 汉字特征向量

从字体文件中提取出汉字的图片形式,即汉字图片大小为 $l\times w$(单位为像素点),共计 N 个像素点。将汉字图片作为输入源,生成汉字的矩阵形式 $\boldsymbol{I}_{l\times w}$,矩阵 $\boldsymbol{I}_{l\times w}$ 中的元素值 $I(i,j)$,$i\in[1,l]$,$j\in[1,w]$ 即为该像素点的灰度值。以阈值 ξ 对矩阵元素进行二值化处理表示为

$$x_{(i-1)l+j}=\begin{cases}1, & I(i,j)\geq\xi\\ 0, & I(i,j)<\xi\end{cases},\quad i\in[1,l],j\in[1,w] \tag{1}$$

二值化处理后,将矩阵 $\boldsymbol{I}_{l\times w}$ 按照从左至右($j=1\rightarrow w$)、从上至下($i=1\rightarrow l$)的规则生成汉字的特征向量 $\{x_1,x_2,\cdots,x_N\}$。将编码方式为Unicode的基本字符集中的20902个汉字依照此向量产生规则生成其汉字特征向量并存入数据库,组建Unicode汉字特征向量数据库。

1.2 汉字笔顺编码

任何汉字都可根据书写笔画顺序分为横、竖、撇、捺、折,即五笔结构,故可按照编码规则对任意汉字生成其汉字笔画顺序编码字符串,简称汉字笔顺字符串,如表1。

表1 汉字笔画编码规则

Tab.1 Chinese character stroke coding rules

笔画	笔画编码
横(一)	a
竖(丨)	b
撇(丿)	c
捺(㇏)	d
折(㇀、㇄、㇂)	e

记汉字为 X,则该汉字 X 的笔顺字符串 $X=x_1x_2x_3\cdots x_z$,其中 z 为该汉字的笔画数,x_i 为该汉字第 i 笔的笔画,并且 $x_i\in\{a,b,c,d,e\}$,$i\in[1,z]$。

将编码方式为Unicode的基本字符集中的20902个汉字依照此编码规则生成其汉字笔顺字符串并存入数据库,组建Unicode汉字笔顺编码数据库。

2 汉字字形相似度算法

针对汉字字形相似度的计算,通过二值化差值算法计算基于汉字特征向量的字形相似度,通过改

进后的 Jaro-Winkler Distance 算法计算基于汉字笔顺编码的字形相似度,2 个相似度分别从不同方面反映了汉字的相似程度,吸取 2 种算法的优势对其进行融合,得到最终相似度。融合算法的整体流程框图如图 1。

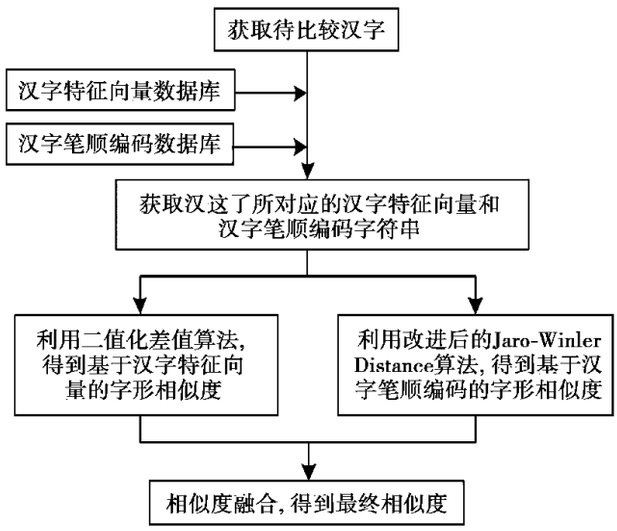


图 1　算法整体流程框图

Fig.1　Overall flow chart of the algorithm

2.1　基于特征向量的字形相似度

为衡量汉字之间基于特征向量的字形相似度,引入二值化差值算法对汉字特征向量进行运算。

算法 1　基于特征向量的字形相似度算法

输入:汉字 A,B。

输出:汉字 A,B 之间基于特征向量的字形相似度 $Sim_1(A,B)$。

Step 1　从 Unicode 汉字特征向量数据库中调取汉字 A,B 所对应的汉字特征向量 $\boldsymbol{A}=\{a_1,a_2,\cdots,a_N\}$ 和 $\boldsymbol{B}=\{b_1,b_2,\cdots,b_N\}$。

Step 2　定义 $c_i=a_i-b_i, i\in[1,N]$,生成汉字 A、B 所对应的差值特征向量 $\boldsymbol{A}-\boldsymbol{B}=\{c_1,c_2,\cdots,c_N\}$。

Step 3　求汉字 A、B 之间基于特征向量的字形相似度 $Sim_1(A,B)$ 可以表示为

$$Sim_1(A,B)=\frac{N-\sum_{i=1}^{N}|c_i|}{N}, i\in[1,N] \quad (2)$$

2.2　基于笔顺编码的字形相似度

为衡量汉字之间基于笔顺编码的字形相似度,要使用字符串相似算法对其进行计算。通常使用 Cosine Similarity, Levenshtein Distance, Hamming Distance 等算法对字符串相似进行判定,但 Cosine Similarity 和 Hamming Distance 算法并不适用于笔顺编码字符串,Levenshtein Distance 算法没有考虑到字符换位的情况,综合考虑后引入改进后的 Jaro-Winkler Distance 算法对汉字笔顺字符串进行运算。

算法 2　基于笔顺编码的字形相似度算法

输入:汉字 A,B。

输出:汉字 A,B 之间基于笔顺编码的字形相似度 $Sim_2(A,B)$。

Step 1　从 Unicode 汉字笔顺编码数据库中调取汉字 A,B 所对应的汉字笔顺字符串 str_A 和 str_B。

Step 2　计算匹配窗口值 MW 为

$$MW=\frac{Max(|len_A|,|len_B|)}{2}-1 \quad (3)$$

(3)式中:len_A 为笔顺编码字符串 str_A 的长度;len_B 为笔顺编码字符串 str_B 的长度。

Step 3　由笔顺编码字符串 str_A,str_B 及匹配窗口值 MW,判决字符是否匹配可以表示为

$$|pos(a)-pos(b)|<MW \quad (4)$$

(4)式中:a 是字符串 str_A 中的某一字符;$pos(a)$ 为 a 在字符串 str_A 中的位置下标;b 是字符串 str_B 中的某一字符;$pos(b)$ 为 b 在字符串 str_B 中的位置下标。遍历字符串 str_A 和 str_B,计算匹配字符数 m,即所有满足(4)式的字符总个数。应注意,在匹配过程中,需排除被匹配过的字符,若找到匹配过的字符,则需跳出此次匹配,进行下一字符的匹配。

Step 4　由笔顺编码字符串 str_A 和 str_B 的匹配字符集确定匹配字符换位数 n,若匹配字符集中字符顺序一致,则 $n=0$,否则 n 为换位数目的一半。

Step 5　计算笔顺编码字符串 str_A 和 str_B 之间的 Jaro Distance 表示为

$$Dis_j=\begin{cases}0, & m=0\\ \frac{1}{3}\left(\frac{m}{len_A}+\frac{m}{len_B}+\frac{m-n}{m}\right), & m>0\end{cases} \quad (5)$$

Step 6　为凸显笔顺编码字符串中相同部分的重要性,提取笔顺编码字符串 str_A 和 str_B 的最长公共子串 str_{AB},并获取其长度 len_{AB},进一步计算笔顺编码字符串 str_A 和 str_B 之间的 Jaro-Winkler Distance,该值即为汉字 A、B 之间基于笔顺编码的字形相似度 $Sim_2(A,B)$,表示为

$$Sim_2(A,B)=Dis_j+\left[len_{AB}\cdot\frac{1}{Max(len_A,len_B)}\cdot(1-Dis_j)\right] \quad (6)$$

2.3　相似度融合

由算法 1 计算所得的基于特征向量的字形相似度 $Sim_1(A,B)$ 和算法 2 计算所得的基于笔顺编码的

字形相似度 $Sim_2(A,B)$ 取值为[0,1],反映了汉字 A,B 之间的字形相似程度,数值越大则说明相似程度越高,但 2 个相似度是从不同方面反映了汉字的相似程度,基于特征向量的字形相似度 $Sim_1(A,B)$ 主要从汉字结构、轮廓等角度反映汉字的相似程度,而基于笔顺编码的字形相似度 $Sim_2(A,B)$ 主要从汉字笔画、书写顺序等角度反映汉字的相似程度。

由于单独使用基于特征向量的字形相似度算法或基于笔顺编码的字形相似度算法对汉字字形相似进行衡量并不严谨。例如,以汉字“未”和“末”为例,若仅以基于笔顺编码的字形相似度算法进行衡量,则两者的相似度为 1,但实际上两者是存在差异的;再以汉字“由”和“甲”为例,若仅以基于特征向量的字形相似度算法进行衡量,则两者的相似度很小,但实际上两者只是上下反转关系。故本文分别吸取 2 种算法的优势,对其进行均衡融合。

算法 3 相似度融合算法

将基于特征向量的字形相似度 $Sim_1(A,B)$ 与基于笔顺编码的字形相似度 $Sim_2(A,B)$ 进行加权平均,即可得到相似度融合算法,表示为

$$Sim(A,B)=\frac{Sim_1(A,B)+Sim_2(A,B)}{2} \quad (7)$$

3 实验与结果

为验证本文提出的基于特征向量和笔顺编码的字形相似算法的有效性,本文设计了 3 个对比实验:①实验 1 使用本文提出的汉字字形相似算法,对任意输入的 2 个汉字进行相似度的计算,并与文献[2]进行比较,考察本算法对汉字字形相似度的准确度;②实验 2 使用本文提出的汉字字形相似算法,对任意输入汉字提取出与其相似的汉字,并与文献[11]、文献[12]进行比较,考察本算法是否能够对汉字更准确、真实地反映其字形的相似程度以及执行效率问题;③实验 3 使用本文提出的汉字字形相似算法,遍历编码方式为 Unicode 的基本字符集中 20 902 个汉字,提取出最为相似的 3 个字,考察本算法对任意汉字的字形相似是否支持。

实验用 Python 3.6 编程语言实现。微软雅黑字体作为输入源,提取 64×64 像素的汉字图片,取灰度二值化阈值 $\xi=1$,即使图片中除了纯白色像素点外将其他像素点都赋予黑色以凸显效果。

实验 1

1)实验设计和评测方法。实验采用文献[2]的实验样本,运行本文所提出的基于特征向量和笔顺编码的字形相似算法并与文献[2]进行对比,通过比较 2 个字之间的相似度,考察本算法对汉字字形相似度识别的准确度。

2)实验结果和分析。实验 1 所采用的汉字样本如表 2。

表 2 实验 1 的汉字样本

Tab.2 Chinese character sample of experiment 1

样本编号	样本
1	玻 // 坡
2	玻 // 波
3	玻 // 披
4	伧 // 沧
5	伧 // 伦
6	伧 // 论
7	伧 // 沦

以编号为 1 的样本为例,执行算法 1 的流程,2 个汉字的特征向量及差值特征向量的形式化描述如图 2,2 个汉字之间基于特征向量的字形相似度 $Sim_1(A,B)=0.885\ 3$;根据算法 2 所述执行流程,计算所得 2 个汉字之间基于笔顺编码的字形相似度 $Sim_2(A,B)=0.984\ 3$;执行算法 3 得 2 个汉字之间最终字形相似度 $Sim(A,B)=0.934\ 8$。

对所有样本依次计算其相似度,得到结果如表 3。

表 3 算法对汉字样本得到的相似度

Tab.3 Similarity of the algorithm to chinese character samples

样本编号	样本	基于特征向量的字形相似度	基于笔顺编码的字形相似度	融合相似度	文献[2]所计算的相似度
1	玻 // 坡	0.885 3	0.984 3	0.934 8	0.764
2	玻 // 波	0.643 8	0.808 8	0.726 3	0.528
3	玻 // 披	0.768 1	0.984 3	0.876 2	0.528
4	伧 // 沧	0.845 9	0.662 7	0.754 3	0.667
5	伧 // 伦	0.804 0	0.933 3	0.868 6	0.833
6	伧 // 论	0.742 7	0.694 4	0.718 6	0.500
7	伧 // 沦	0.751 7	0.531 7	0.641 7	0.500

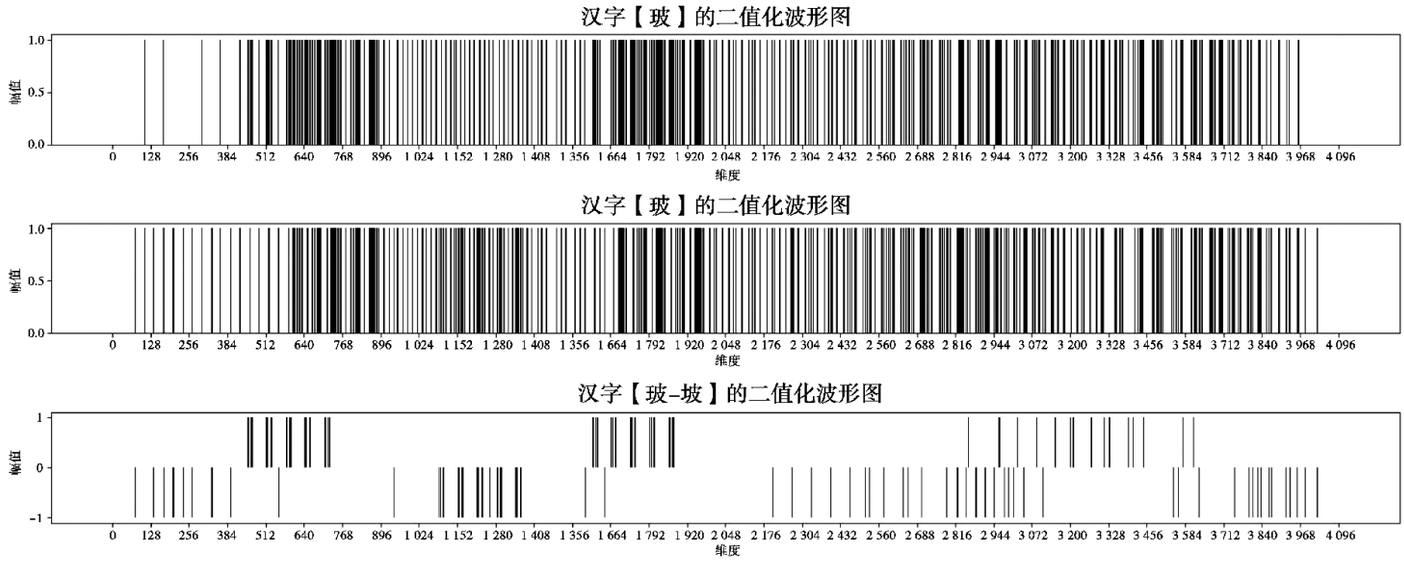


图 2　汉字“玻”和“坡”的特征向量及差值特征向量

Fig.2　Feature vectors and difference feature vectors of chinese characters “glass” and “slope”

从表 3 可知,基于笔顺编码的字形相似度因为只考虑到笔画顺序,假如 2 个汉字的笔顺相同则计算结果相同,如“坡”和“披”,但这样是不合理的,故加入基于特征向量的字形相似度对其进行融合,实现进一步区分。将本实验结果与文献[2]所计算的结果进行比较,发现文献[2]对样本 2、样本 3 的计算结果相同,但通过人眼识别,事实上“披”相对“波”更与“玻”相似,本算法也验证了这一点;同样文献[2]的算法对样本 6、样本 7 存在相同问题。综合实验结果,本算法对汉字字形相似度的识别较合理,相似度较文献[2]提高约 27.8%。

实验 2

1)实验设计和评测方法。实验采用文献[11]的实验样本,利用本文提出的基于特征向量和笔顺编码的字形相似算法对样本进行相似字的提取,并将结果与文献[11]进行比较,同时将程序运行时间与文献[12]进行对比。通过比较算法对比同一样本得到的相似字及运行时间,考察本算法是否能够对汉字更准确、真实地反映其字形的相似程度以及执行效率问题。

2)实验结果和分析。实验 2 所采用的汉字样本如表 4。

表 4　实验 2 的汉字样本

Tab.4　Chinese character sample of experiment 2

样本编号	1	2	3	4	5	6	7
样本	蔼	闷	恋	货	椅	钻	察

分别利用文献[11]所介绍的汉字字形相似算法和本文提出的基于特征向量和笔顺编码的字形相似算法对汉字样本进行相似字的提取,并得到本算法的运行时间,结果如表 5。

表 5　算法对汉字样本提取的相似字及程序运行时间

Tab.5　Algorithm similar words extracted from chinese character samples and program running time

样本编号	样本		相似字	执行时间/ms
1	蔼	文献[11]	霭蕴落藩范莎薄藻蒲落藻菏菠范萍	147.3
		本算法	葛獦噶藹谒揭藒蒐磕蕩蔓揭堨偈葭	
2	闷	文献[11]	闰刊闲阂间阂阅闫闭问闯闸闹阅闪	104.8
		本算法	闪闵阆阂吣闵丽阌闩闼闶阙阂问门	
3	恋	文献[11]	挛栗妾孪弯莺变育蛮峦惩恐恕恙忽	94.3
		本算法	恣峦侦态忐恣怂怼忘孪您变栾悠总	
4	货	文献[11]	贷货赁贤赞资贸贺华赘赞责贪贯员	132.8
		本算法	贷赁貨贪贸赁负贫贤贺贯役役贠俛	
5	椅	文献[11]	特椅倚猜琦畸绮骑崎捺掩棒椿揍挎	99.4
		本算法	祹掎徛埼倚犄绮琦楂棟崎桍榷植猗	
6	钻	文献[11]	钻错铝铬铭错错惦拈沾陆将错援杭	117.4
		本算法	钻铀铂铦钳钟玷铝钇钿铓销钙贴铅	
7	察	文献[11]	蔡寮蔡察寄签寥茶寡寨窖赛寰寞富	120.1
		本算法	蔡案窠窓寮嚓窖窦客瘵寀窥寖檫宇	

从表 5 可知,本算法对样本所提取的相似字与文献[11]所介绍的算法结果相比,还是有很大的差异。文献[11]的结果并不理想,比如对样本 1 来说,汉字“蔼”和“落”、“范”、“莎”等通过人眼识别,除了偏旁外并无相似之处,而本算法所提取的汉字“葛”、“獦”、“噶”确实与“蔼”有相似之处,而且结果与样本都包含有相同汉字部件。

此外,本算法在提取相似汉字时所花费的平均时间为 116.6 ms,文献[12]中的模板匹配算法需要 0.2~1.0 s 才能识别一个字符,结构方法用时为 0.2~0.5 s,神经网络算法用时约 0.1 s。通过比较可知,本算法在执行效率上较其他算法提高约 66.7%。

实验结果表明,本算法可以更准确、真实地反映汉字字形的相似程度,同时执行效率较高。

实验 3:

1)实验设计和评测方法。实验采用双层遍历编码方式为 Unicode 的基本字符集中 20 902 个汉字的方式,利用本文的基于特征向量和笔顺编码的字形相似算法对其进行相似度计算及排序,并将与对比汉字相似度最高的 3 个汉字提取出来。通过对结果进行分析,考察本算法对任意汉字的字形相似是否支持。

2)实验结果和分析。实验利用本文基于特征向量和笔顺编码的字形相似算法对其进行相似度计算及排序,并将与对比汉字相似度最高的 3 个汉字提取出来。得到的部分结果例举如下:

一:十#0.8430/干#0.8270/亏#0.8209

二:三#0.8669/午#0.8092/干#0.7988

人:入#0.9441/太#0.8681/八#0.8616

木:术#0.9406/朩#0.9406/本#0.8827

不:丕#0.8900/下#0.8900/丁#0.8604

贝:页#0.8676/央#0.8627/见#0.8600

从:丛#0.8771/朳#0.8726/认#0.8498

宂:冗#0.8982/宄#0.8970/穴#0.8925

我:牦#0.8740/伐#0.8350/找#0.8330

地:扡#0.8956/坉#0.8876/忯#0.8625

怨:怒#0.8387/忍#0.8288/愍#0.8283

勉:勀#0.8659/葂#0.8251/勃#0.8214

剑:刽#0.8585/釗#0.8306/钊#0.8254

俑:俑#0.8858/涌#0.8823/诵#0.8711

本实验统计结果表明,相似度在 0.99 以上的汉字有 4 对,相似度在 0.98~0.99 的汉字有 12 对,具体如表 6;另外,相似度在 0.95 以上的汉字有 297 对,相似度在 0.90 以上的汉字有 6 665 对,相似度在 0.85 以上的汉字有 23 726 对。

表 6 相似度在 0.98 以上的相似字对

Tab.6 Similarity of similar word pairs above 0.98

相似字对	相似度	相似字对	相似度	相似字对	相似度	相似字对	相似度
陕 // 陕	0.990 1	贼 // 贼	0.990 4	菓 // 菓	0.993 4	壿 // 壿	0.993 0
劦 // 劦	0.981 6	余 // 余	0.981 8	釩 // 釩	0.984 1	圈 // 圈	0.982 2
鈇 // 鈇	0.985 7	彙 // 彙	0.985 5	嶂 // 幛	0.980 8	獎 // 獎	0.980 0
媸 // 媸	0.980 7	膧 // 膧	0.982 4	騷 // 騷	0.980 1	囉 // 囉	0.983 6

通过人眼识别表 6 所列出的相似汉字,确实具有很大的字形相似程度,说明本文所提出的汉字字形相似算法对任意汉字(独体字、复合字)都能提供较好的支持。

4 结束语

本文从汉字自身特征出发,提出了基于特征向量和笔顺编码的字形相似算法,通过实验 1 验证了本算法的准确率比文献[2]提高 27.8%;通过实验 2 验证了本算法能够比文献[11]、文献[12]更准确、真实地反映汉字字形的相似程度以及执行效率;通过实验 3 验证了本算法对任意汉字之间的字形相似度判断提供支持。本文可以得到几个结论:①从汉字结构、轮廓等角度出发,将汉字转化为特征向量形式,采用二值化差值算法对其进行计算,能够充分刻画汉字之间在汉字结构、轮廓方面的字形相似程度;②从汉字笔画、书写顺序等角度出发,将汉字进行笔顺编码,采用改进后的 Jaro-Winkler Distance 算法对其进行运算,能够更好地刻画汉字之间在笔画、书写顺序方面的字形相似程度;③通过将基于特征向量的字形相似度和基于笔顺编码的字形相似度进行融合,减少了单独使用某一算法对汉字字形相似带来的误差,同时也能吸取 2 种算法的优势,从而获得更好的综合效果。本文算法对汉字字形相似度计算的准确度高、运行时间短,较 3 元组递归算法准确度提高约 27.8%,较模板匹配算法、结构方法、神经网络算法执行效率平均提高约 66.7%,同时对任意汉字(独体字、复合字)的字形相似都有较好的支持度。

参考文献:

[1] 马飞,吕海莲,石果.基于最短欧氏距离匹配的印刷体汉字识别[J].平顶山学院学报,2012,27(02):70-73.
MA F, LV H L, SHI G. The research on printed chinese characters recognition based on nearest euclid distance template-matching[J]. Journal of Pingdingshan Universi-

ty,2012,27(02):70-73.

[2] 王东,熊世桓.一种新颖的汉字字形相似度计算方法[J].计算机应用研究,2013,30(08):2395-2397.
WANG D, XIONG S H. New algorithm for similarity calculation of chinese character glyph[J]. Application Research of Computers,2013,30(08):2395-2397.

[3] 刘徽,黄宽娜.中文密文数据库正则查询的研究与实现[J].重庆邮电大学学报(自然科学版),2011,23(02):247-252.
LIU H, HUANG K N. Research and implement of regular query of chinese encrypted database[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition),2011,23(02):247-252.

[4] 李丹,张蒙,孙海涛,等.一种改进的 KAZE 特征检测算法[J].四川大学学报:自然科学版,2015,52(3):523-528.
LI D, ZHANG M, SUN H T, et al. An improved algorithm for feature detection based on KAZE[J]. Journal of Sichuan University(Natural Science Edition), 2015, 52(3): 523-528.

[5] 林民,韩冬妹,宋柔.基于 GDI+路径技术的汉字笔顺和部件自动绘制[J].计算机应用研究,2007,24(08):228-230.
LIN M, HAN D M, SONG R. Automatic drawing chinese character stroke orders and radicals based on GDI+ paths[J]. Application Research of Computers,2007,24(08):228-230.

[6] 林民,宋柔.一种面向构形计算的汉字字形形式化描述方法[J].中文信息学报,2008(03):115-123.
LIN M, SONG R. Pattern computing-oriented formal description of chinese character glyph[J]. Journal of Chinese Information Processing,2008(03):115-123.

[7] 林民.汉字字形形式化描述方法及应用研究[D].北京:北京工业大学,2009.
LIN M. The stduy of formal description of chinese character glyph and application[D]. Beijing: Beijing University of Technology, 2009.

[8] 栗青生,张莉,刘泉,等.一种基于云端信息保护的汉字计算模型[J].计算机科学,2015,42(11):73-79.
LI Q S, ZHANG L, LIU Q, et al. Chinese character computing model based on cloud information protection[J]. Computer Science,2015,42(11):73-79.

[9] 赵健,冯乔生,何娟娟.面向汉字识别的新特征及其提取方法[J].软件,2015,36(03):31-36.
ZHAO J, FENG Q S, HE J J. New features for chinese character recognition and its extraction method[J]. Software,2015,36(03):31-36.

[10] 胡浩,李平,陈凯琪.基于汉字固有属性的中文字向量方法研究[J].中文信息学报,2017,31(03):32-40.
HU H, LI P, CHEN K Q. Research on Chinese character embedding based on its inherent attributes[J]. Journal of Chinese Information Processing,2017,31(03):32-40.

[11] 宋柔,林民,葛诗利.汉字字形计算及其在校对系统中的应用[J].小型微型计算机系统,2008,29(10):1964-1968.
SONG R, LIN M, GE S L. Similarity calculation of chinese character glyph and its application in computer aided proofreading system[J]. Journal of Chinese Computer Systems,2008,29(10):1964-1968.

[12] 孙华,张航.汉字识别方法综述[J].计算机工程,2010,36(20):194-197.
SUN H, ZHANG H. Survey on Chinese character recognition method[J]. Computer Engineering,2010,36(20):194-197.

[13] 刘波. 改进的图像匹配方法在汉字识别中的应用[D].广州:暨南大学,2015.
LIU B. The application of improved image registration method in chinese character recognition[D]. Guangzhou: Jinan University, 2015.

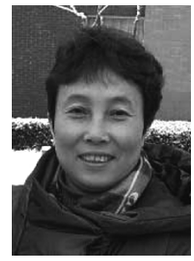
[14] 刘斌,肖惠勇.基于不可分小波变换与 Zernike 矩的印刷体汉字识别方法[J].计算机应用与软件,2018,35(04):227-236.
LIU B, XIAO H Y. Printed chinese character recognition based on non-separable wavelet transform and zernike moments[J]. Computer Applications and Software,2018,35(04):227-236.

[15] 牛永洁,张成.多种字符串相似度算法的比较研究[J].计算机与数字工程,2012,40(03):14-17.
NIU Y J, ZHANG C. Comparation of string similarity algorithms[J]. Computer & Digital Engineering,2012,40(03):14-17.

[16] 吴凌芬,杨小渊,叶添杰,等.改进 Jaro-Winkler 算法在迎宾机器人语音交互中的应用[J].现代计算机:专业版,2015(03):8-13.
WU L F, YANG X Y, YE T J, et al. Application of improved Jaro-Winkler distance in speech interaction of reception robot[J]. Modern Computer: Professional Edition,2015(03):8-13.

作者简介:

祁俊辉(1994—),男,硕士研究生,主要研究方向为中文信息处理。E-mail:qcehui@qq.com。



龙　华(1963—),女,云南大理人,教授,博士,主要研究方向为网络通信、信息处理。E-mail:1670931890@qq.com。

(编辑:陈文星)