基于决策树的快速在线手写数字识别技术*)

姜文理　王　卫　孙正兴

(南京大学计算机软件新技术国家重点实验室　南京 210093)

摘　要　本文提出了一种快速的在线手写数字识别方法,该法采用书写笔划走势对手写数字进行建模,运用决策树学习算法进行数字分类识别。数字笔划走势特征提取简单、区分度高、对用户不敏感,实现了有限的资源条件下的高速识别,同时保证了方法的良好用户适应性;决策树学习算法分类情况全面,保证了方法的高识别率。实验结果表明:该方法既具有简单高效的特点,又具备很好的用户适应性。

关键词　笔划走势,方向码,决策树,if-then 规则,ID3 算法

Fast Online Handwritten Digits Recognition Based on Decision Tree

JIANG Wen-li　WANG Wei　SUN Zheng-Xing

(State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract　This paper proposed a fast method for handwritten digit recognition. The proposed method modeled the handwritten digits with the variation of stroke direction, and did the classification by decision tree learning. The extraction of stroke direction variation is simple, highly discriminable and insensitive to different users, implementing fast recognition under strict resource constraints, and the user adaptability is guaranteed at the same time. The decision tree learning is able to cover all the variance in user input, which guaranteed high recognition rate. The outcome of the experiments demonstrated the proposed method to be simple and effective with good user adaptability.

Keywords　Variation of stroke direction, Direction code, Decision tree, If-then rule, ID3 algorithm

1　引言

手写阿拉伯数字是世界各国通用的符号,手写数字识别属于模式识别和人工智能的范畴,对其进行研究具有十分广泛的意义[1]。尤其是 20 世纪 90 年代以来,小型化、便携式和无线设备的发展以及普适计算模式[2]的出现,使得以手写体(Handwritten)和手绘草图(Sketch)为核心的笔式交互(Pen-based User Interface)成了新一代人机交互技术中不可缺少的部分。在有限的资源条件下进行有效的识别,是手写数字识别技术研究的新热点。

按照识别方式,手写数字识别分为两种[3]:离线识别和在线识别。离线识别主要是将用户输入的手写数字以图像方式(手写数字扫描图或将实时输入转换成图像)来进行处理和识别,这种方法比较成熟,且已有大量的研究成果[4]。一方面,图像处理方式算法过程复杂,对存储要求较高。另一方面,图像处理方式可资利用的识别线索比较少,在用户适应性方面[5]很难满足不同用户输入习惯的差异以及同一用户不同时刻输入的差异。近年来兴起的在线识别 技术[6,7]直接利用用户输入笔迹的几何和时序等过程信息进行处理,这种方法对存储和处理的要求相对较低,且可有效利用多种线索来捕捉用户的输入习惯,具有很好的用户适应性和效果。如何有效地对手写数字建模并实现这种适应性成了这一方法面临的主要课题。

我们认为:人类视觉系统之所以能有效辨别各种手写数字,最根本的原因在于"形似"。这里所讲的"形似",不仅体现在外形轮廓上,更体现在整个数字书写的过程上。手写数字输入的多种变体之间存在的"书写走势"这个特征正好能体现这种"形似"。据此,本文提出了一种基于笔划走势的手写数字建模方法,并利用决策树学习方法[8],有效捕捉线索和各数字间的差异,实现了基于决策树的在线手写数字识别方法,既保证了识别算法的快速高效,又达到了良好的用户适应性。

2　基于走势的手写数字建模

2.1　手写数字建模的特征选择

手写识别研究的最终目标是零误识率和低拒识率的高速识别,手写数字的特征描述在其中具有举足轻重的地位。已有的手写数字特征描述主要包括两类:全局特征和结构特征。全局特征主要是指采用统计方法获得像素密度、矩、特征点、数学变换等特征,这种特征的获取算法比较复杂,计算开销大,且对输入类别的区分度不高。结构特征是指从字符的轮廓或骨架上提取字符形状的基本特征,包括圈、端点、节点、弧、突起、凹陷、笔画等,这些特征适合句法分析,提取算法简单,开销小,区分度高,但其描述过于繁杂。

笔划走势是人类视觉系统区分手写输入的一个重要线索,它是一个相对抽象的整体概念。在手写数字这个特定领域,笔划走势对类别间的差异具有很好的区分度,而对个体间的局部差异并不敏感,同时在一定程度上反映了输入过程的时间特性。这样既能保证对手写数字描述的简洁性,又能保证其具有很好的用户适应性,同时走势对旋转、平移、放大、缩小都满足不变性,将对完全没有限制的手写数字识别提供很

*)国家自然科学基金(编号:69903006、60373065)资助项目。**姜文理**　硕士生,研究方向为智能用户接口;**王　卫**　硕士生,研究方向为智能用户接口;**孙正兴**　博士,教授,博士生导师,研究方向为:智能人机交互、多媒体计算和计算机视觉。

好的支持。

图1是手写数字的简单示例:(a)中"2"的写法各不相同,但人眼一下子就能识别出它们都是"2",其原因就在于这3个字都依次包含了向右、向左下、向右的笔划走势;(b)中"8"都依次包含了向左、向右下、向左、向右上的笔划走势;(c)中"5"的第一笔都依次包含了向下、向右、向下、向左的笔划走势。

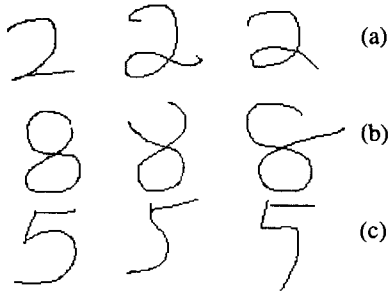
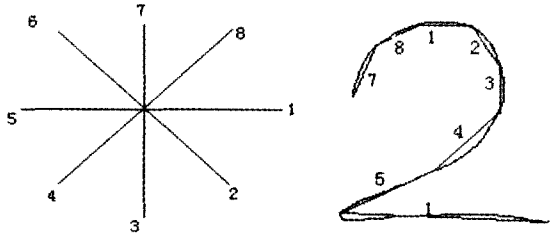


图1 手写数字示例

从例子可以看出笔划走势是数字不同写法所共有的一个特征,下面将对笔划走势进行具体的定义。

2.2 笔划走势的定义

这里把笔划走笔方向进行8方向编码,如图2(a)所示,将坐标平面均分为8个区依次编码,并将该编码称为笔划方向码。手写数字按笔划方向码编码处理后的效果如图2(b)所示,图中的方向码序列"78123451"所展现出来的方向码的变化趋势就是笔划的走势。



(a) 8方向编码示意图 (b) 笔划方向码示意图

图2 手写数字建模的方向码定义

根据输入点坐标,由下式产生笔划方向码:

$$k=(Y_{pt}-Y_{p(t-m)})/(X_{pt}-X_{p(t-m)})$$

其中,k是两点间的变化斜率,根据斜率来确定出相应的方向码;m是在计算t时刻的方向码时所取的前第m个点坐标。m的具体取值,要根据实际手写输入的采样频率来确定;m取得太小,很难克服抖动所带来的噪声,导致方向码频繁变化;m取得太大,容易漏掉关键坐标点,造成方向的错判。为了适应各种采样频率,本文取$m=1$,由此带来的噪声,将运用下文的滤波处理来消除。

先把笔划走势简单地划分为两大类,即单向笔划和变向笔划。单向笔划表示笔划的走向保持在某一方向上;变向笔划表示笔划的走向不只是出现在某个方向,可能会有两个或两个以上的方向。下面再给出几种变向笔划的分类。(1)顺笔划:笔划变向按顺时针变化;(2)逆笔划:笔划变向按逆时针变化;(3)混合笔划:笔划变向既有顺时针又有逆时针。

为了区分手写数字的具体类别,还必须抽取一些其它的走势分类特征作为决策树的分类属性。

2.3 笔划数据预处理

原始笔划数据中有大量的冗余和噪声,必须进行滤波处理,以消除这些冗余和噪声。滤波处理分两步:第一步是对原始点坐标的滤波;第二步是对方向码的滤波。

2.3.1 原始坐标数据的滤波

(1) 平滑滤波处理

同一笔划中相邻点之间具有一定的相关关系,可以采用一种简单的平滑处理方法来消除大部分的噪声点。

$$\begin{cases} X_{pt}=(X_{(t-n)}+X_{(t-n+1)}+\cdots+X_t+X_{(t+1)}+\cdots \\ \quad +X_{(t+n)})/(2n+1) \\ Y_{pt}=(Y_{(t-n)}+Y_{(t-n+1)}+\cdots+Y_t+Y_{(t+1)}+\cdots \\ \quad +Y_{(t+n)})/(2n+1) \end{cases}$$

其中,(X_t,Y_t)是笔划在t时刻的坐标,(X_{pt},Y_{pt})表示经过平滑后的数据。n是向前向后取点个数,具体取值要看采样点的疏密情况:比较密的时候取值可以大些,取值2或3时的平滑效果就很明显。实验显示这种处理在去除噪声点的同时并不改变笔划的整体走势。

(2) 数据冗余消除

平滑处理过的数据中还有不少冗余数据,消除这些冗余可以减少后续计算量和方向码中的噪声。采样点之间靠得太近,会使方向码的计算方式变得不可靠,容易出现噪声方向码。我们可以用下式来对数据进行冗余消除:

$$\Delta=\sqrt{(X_{pt}-X_{p(t-1)})^2+(Y_{pt}-Y_{p(t-1)})^2}$$

当Δ小于一定阈值时就认为点(X_{pt},Y_{pt})是冗余点。通过实验,阈值取5~8时效果最好。

2.3.2 方向码的滤波

在实际书写过程中,输入的笔划比较随意,在方向码序列中,会含有不少的噪声和人为的错笔,这就需要对方向码进行滤波。具体的滤波方法如下。

(1)笔划起始处和终止处的噪声:人们的书写习惯容易在落笔和起笔时引入噪声,一般去除同一笔划的前两个方向码和后两个方向码,就能消除这种噪声。

(2)笔划平直处的噪声:笔尖的抖动容易造成在一串相同的方向码中混有一个或两个不同的方向码,删除这些方向码,即可消除这种噪声。

(3)笔划变向处的噪声:在笔划变向处容易引入噪声,出现一个或两个方向码与前后方向码都不相同,删除它们来消除这种噪声。

2.4 笔划方向码的合并

通过预处理得到了笔划的有效方向码。在进行走势分类前,还需对方向码进行合并处理,使其简化。处理方法是:对同一笔划中连续相同的方向码进行合并,得到两组数据:第一组是笔划的方向码序列$M_i(i=1,2,\cdots,n)$,$1\leqslant M_i\leqslant 8$,序列包含的不同方向码个数为$n$,相邻方向$M_i$与$M_{i-1}$经合并后,不可能是同方向的码值;第二组数代表方向码序列M_i所对应的某方向上的方向码合并的个数$N_i(i=1,2,\cdots,n)$,N_i将用于后续的分类过程。

2.5 笔划类别属性生成

经过上述处理,产生了正确的笔划方向码序列M_i。根据M_i先将笔划区分为上面定义的4种类别。具体判别算法如下:

(1)如果方向码序列M_i包含的方向码个数n为1,则将该笔划分类为单方向的基本笔划,即单向笔划。

(2)如果方向码序列M_i包含的方向码个数$n>1$,则将该笔划分类为变方向的复合笔划。根据相邻方向码之间的码值的变化情况,进一步区分出笔划是顺笔划、逆笔划或混合笔

划。具体过程如下：

如果$-4\leqslant M_i-M_{i+1}\leqslant -1$或$4\leqslant M_i-M_{i+1}\leqslant 7$对于所有$i=1,2,\cdots,n$成立，那么输入笔划是顺笔划；如果$1\leqslant M_i-M_{i+1}\leqslant 4$或$-7\leqslant M_i-M_{i+1}\leqslant -4$对于所有$i=1,2,\cdots,n$成立，那么输入笔划是逆笔划；其它情况下，输入笔划是混合笔划。

2.6 其它分类属性的生成

笔划类别属性还不足以区分数字类别，不同类别的数字会存在笔划类别相似的情况。要完全区分开所有类别，必须抽取一些其它特征作为分类属性。先定义几个新属性。

(1)旋转方向。笔划的方向码按顺时或逆时针的变化定义为笔划的旋转方向。

(2)旋转幅度。同一个旋转方向上的所有相邻方向码差的绝对值总和定义为旋转幅度。差值的绝对值大于4时，用8减去这个绝对值作为取值。

(3)旋转力度。旋转幅度在相同距离上的变化快慢定义为旋转力度。

(4)首尾点距离。笔划的起点和尾点之间的距离定义为首尾点距离。

(5)旋转次序。混合笔划中“顺→逆”或“逆→顺”这种旋转变化定义为旋转次序。

图3是上述属性用于数字类别区分的具体示例。(a)和(c)所示情况，离散化旋转幅度属性，用于区分出数字类别“2”、“7”、“4”和“6”。(b)和(d)所示情况，离散化首尾距离属性，用于区分出数字类别“2”、“6”、“0”和(e)所示情况，离散化旋转力度属性，用于区分出数字类别“6”和“9”。(f)所示的混合笔划，存在旋转次序上的差异。图中“2”的旋转次序是“顺→逆”，“3”的旋转次序是“顺→逆→顺”，“8”的旋转次序是“逆→顺→逆”。(g)和(h)所示情况，计算笔划中不同旋转方向部分的旋转幅度，用于区分出数字类别“5”、“8”和“9”。

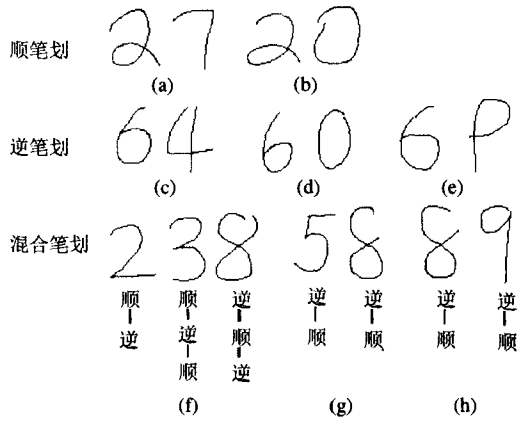


图3 手写数字的具体分类情况

通过例子可以看到，用笔划走势来分类数字是比较简单有效的，数字的字符域小且固定，选取7到8个分类属性，就可以有效地把数字区分。

3 基于决策树的分类

决策树也称为判定树，是一种有指导的学习方法。决策树代表着决策集的树形结构，可以根据训练集数据构造出决策树。如果该树不能对所有对象给出正确的分类，就选择一些例外加入到训练集数据中。重复该过程，直到形成正确的决策集。决策树方法首先对数据进行处理，利用归纳算法生成可读的规则和决策树，然后使用决策树对新数据进行分析，本质上是通过一系列规则对数据进行分类的过程。

3.1 决策树的生成算法

通过上面的分析，构造决策树所需要的测试属性基本确定。考虑到走势分类并不是十分复杂，本文采用了ID3算法的基本思想来学习构造决策树。算法ID3的策略是：

(1) 最初，所有的训练样本都在树的根部；

(2) 如果属性值连续，进行离散化；

(3) 基于选择的属性递归地划分样本；

(4) 使用启发式或统计式的策略(信息增益)选择最优划分；

(5) 当以下条件为真时，划分停止：

a) 分入当前节点的所有样本属于同一类C，类标签为C；

b) 没有属性可用于进一步的样本划分，取节点中样本归属最多的类别为类别标签；

c) 无样本分入当前节点，取父节点中样本归属最多的类别为类别标签。

其中，第(4)步中最优划分的属性选择策略是决策树生成的核心。

3.2 属性选择方法

ID3使用信息增益作为属性选择标准。它使得在各级非叶节点选择的属性具有最大的信息熵增益率，从而保证该非叶节点到达后代的叶节点的平均路径最短。设定S表示训练样本集，s_i表示类别$C_i(i=1,2,\cdots,m)$的训练样本，那么要对一个给定数据对象进行分类所需要的信息量为：

$$I(S_1,S_2,\cdots,S_m)=-\sum_{i=1}^{m}p_i\log_2(p_i)$$

其中，p_i是任意一个数据对象属于类别C_i的概率，可以按S_i/S计算。而其中的log函数以2为底是因为在信息论中信息都是按位进行编码的。

设一属性A取v个不同的值$\{a_1,a_2,\cdots,a_v\}$。利用属性A可以将集合S划分为v个子集$\{S_1,S_2,\cdots,S_v\}$，其中S_j包含了S集合中属性A取a_j值的数据样本。若属性A被选为测试属性(用于对当前样本集进行划分)，设S_{ij}为子集S_j中属于C_i类别的样本数，那么利用属性A划分当前样本集合所需要的信息(熵)可以计算如下：

$$Ent(A)=\sum_{j=1}^{v}\frac{S_{1j}+S_{2j}+\cdots+S_{mj}}{S}I(S_{1j},\cdots,S_{mj})$$

其中，$\frac{S_{1j}+S_{2j}+\cdots+S_{mj}}{S}$项被当作第$j$个子集的权值，它是由所有子集中属性$A$取$a_j$值的样本数之和除以$S$集合中的样本总数。$Ent(A)$计算结果越小，就表示其子集划分结果越“纯”(好)。而对于一个给定子集S_j，它的信息为：

$$I(s_{1j},s_{2j},\cdots,s_{mj})=-\sum_{i=1}^{m}p_{ij}\log(p_{ij})$$

其中，$p_{ij}=\frac{s_{ij}}{|S_j|}$，即为子集$S_j$中任一个数据样本属于类别$C_i$的概率。

这样，利用属性A对当前分支节点进行相应样本集合划分所获得的信息增益$Gain(A)$为：

$$Gain(A)=I(s_1,_2,\cdots,s_m)-Ent(A)$$

决策树算法计算每个属性的信息增益，并从中挑选出信息增益最大的属性作为给定集合S的测试属性，并由此产生相应的分支节点。所产生的节点被标记为相应的属性，并根据这一属性的不同取值分别产生相应的分支，每个分支代表

一个被划分的样本子集。

3.3 树枝修剪

在决策树生成过程中，由于训练数据中噪声和孤立点的存在，会使得有些分枝反映的是训练数据的异常，即出现“过配”现象，因此需要通过剪枝方法剪去不可靠的分枝，提高树独立于训练数据正确分类的能力。

通常来说，有两种常用的剪枝方法：

(1) 预剪枝(prepruning)

预剪枝通过制定停止节点分裂的条件而提前停止树的构造。

(2) 后剪枝(postpruning)

后剪枝方法从完全生长的树中剪去分枝。

本文采用了计算复杂度和时间开销相对小的预剪枝方法，在建造决策树的过程中进行适当的剪枝。

4 实验设计

4.1 实验内容

选择 20 个不同用户，共采集了约 20000 个样本，每个数字约 2000 个。每个用户在 1 周内书写约 1000 个数字，考虑到环境对书写的影响，采集分 10 次进行，每次选择上午、中午、下午或晚上等不同的时间段。

把 20 个用户分成 2 个小组，每组 10 个，第一次从 1 组的数据中抽取一定数量作为训练样本，2 组的数据作为测试样本；第二次反过来进行。

实验同时进行了另外两项测试：第一，训练样本总量从 500 个(即每个数字约 50 个)开始逐渐增加，分成 500、1000、2000、3000、5000 几个规模进行，并每次记录下各个数字在这些训练样本规模下的测试识别率；第二，加入人工预测走势的 if-then 分类规则，对数字进行分类识别，并记录下测试结果，然后和决策树方法进行比较。

4.2 实验结果

决策树方法实验结果如表 1 所示。

表 1 手写数字在线识别实验结果

(%)	0	1	2	3	4	5	6	7	8	9	平均
用户 1	98	100	97	98	96	94	95	97	94	95	96.3
用户 2	97	99	96	97	97	92	93	97	93	94	95.7
用户 3	96	98	96	96	95	92	94	96	95	94	95.2
用户 4	96	100	97	96	100	95	97	100	94	95	97.1
用户 5	100	100	98	99	97	95	96	99	96	94	97.4
用户 6	100	100	100	100	99	96	98	100	100	98	99.1
用户 7	94	98	98	95	91	93	93	98	92	93	94.5
用户 8	93	99	96	95	95	91	95	97	93	94	94.7
用户 9	97	100	95	94	95	94	96	97	94	95	95.5
用户 10	98	100	98	98	97	95	96	99	100	100	98.1
用户 11	96	99	95	94	96	94	93	97	93	91	94.8
用户 12	97	100	100	98	95	95	96	100	95	97	97.3
用户 13	100	100	98	97	92	91	95	97	93	95	95.8
用户 14	94	97	93	95	94	93	92	97	91	91	93.7
用户 15	96	97	95	95	96	92	94	98	94	94	95.1
用户 16	100	100	100	99	98	96	98	100	97	96	98.4
用户 17	96	100	96	96	95	92	94	100	96	97	96.2
用户 18	95	98	95	94	96	93	91	93	91	92	93.8
用户 19	96	99	96	96	94	93	95	100	95	95	96.0
用户 20	98	100	97	97	96	94	95	98	95	95	96.5
平均	96.8	99.2	96.8	96.4	95.7	93.5	94.8	98.0	94.6	94.8	**96.06**

表 1 中，每一行表示的是一个用户在 0～9 这 10 个不同数字上的识别率，每一列表示的是 0～9 这 10 个数字在 20 个不同用户下的识别率。最后一列是每个用户手写数字的平均识别率，最后一行是每个数字在 20 个测试用户下的平均识别率。

从表中可以看出：(1)同一数字在不同测试用户下，识别率存在一定差异；(2)同一测试用户对不同数字的识别率有一定差异。产生上述现象的主要原因是笔划走势特征提取的精准直接影响到了决策树的生长，进而影响到识别时的精准。笔划走势简单的数字，容易捕捉到完整的特征，从而识别率较高。相反，笔划走势复杂的数字，容易丢失部分走势特征，导致识别率略低。

综合观察实验结果，所有测试用户手写数字的平均识别率均高于 93%，说明这个方法的识别率是比较令人满意的，同时说明了该方法具有良好的用户适应性。测试中，在 PⅢ机器上识别的平均响应时间不到 1ms，在 PDA 上的平均响应时间约为 50ms，说明该方法很快速。

在用不同规模的训练样本进行测试的实验中，得到如图 4 所示的结果。

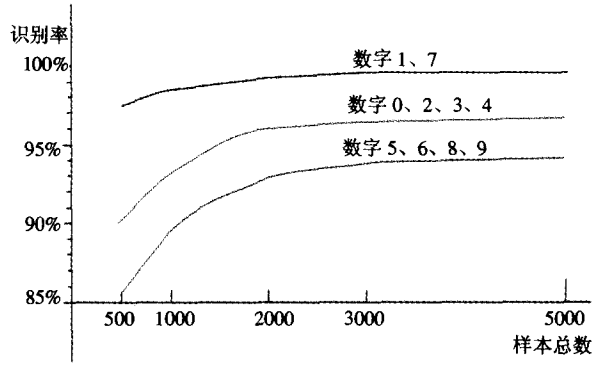


图 4 不同样本规模的数字识别率曲线

从图 4 中可以看到，不同数字的用户识别率趋于稳定所需要的样本数量是不一样的，而且每个数字最后趋于稳定的识别率上限也有所不同，主要原因是不同数字笔划走势的复杂度存在差异。走势简单的数字，分支情况少，在很小的样本量下，识别率就趋于稳定，而且具有较高的识别率。

决策树方法和 if-then 规则的测试结果比较如图 5 所示。

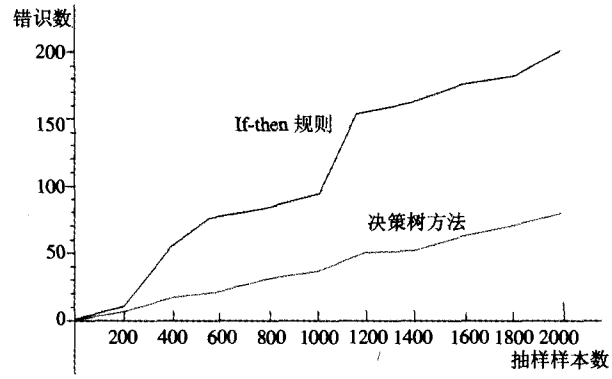


图 5 抽样统计两种方法错识数曲线

图 5 显示的是一个抽样统计的结果。从每个测试用户数据中随机抽取 100 个测试样本，累加统计所有用户在这 100

(下转封三)

址信息(内码),当用户从拼音相同的汉字中选出所需的汉字时,将该字在字库中的地址作为参数传递到联想子程序,并以该信息作为检索的条件,再从根结点到分支结点或是叶子结点进行查询,类似于在生成树中查询,然后得到由该字联想到的汉字的地址信息,再从字库中取出汉字供用户选择。以词语"地球"的输入为例,根据第一个汉字内码和用户选择的汉字在这组汉字中的位置,可以获得用户选择的汉字的内码,如"地"的内码为2080,以该内码为参数,在联想码表生成树中查找该内码的对应汉字的联想汉字,从其取出由"地"联想到的汉字,以一个静态数组记录其内码,"球"的内码为8980,再显示该组汉字,"地"联想的汉字常用的就一个"球"字,若有多个则顺序显示。至此,完成用户对于"地球"的输入。

由于联想码表生成树中,一条路径表示联想汉字的地址,又因为一个汉字在字库中占两个字节,所以,有效的路径其分支结点或是叶子结点都是偶数。

改进的Trie树的查找过程都是从根结点出发,走一条从根到叶子结点的路径,其查找时间依赖于数的深度,在拼音生成树中,树的深度为6,因为最长的拼音由6个字母组成;而在联想码表生成树中,常用汉字有5000多个,根据每个汉字占2个字节,其地址信息需要6个数字表示,树的深度也为6。

3.2 Makefile文件

系统在uClinux下开发所使用的Makefile文件内容为:

```
EXEC=pinyin
OBJS=pinyin.o graphic.o
all:$(EXEC)
$(EXEC):$(OBJS)
    $(CC) $(LDFLAGS)-o $@ $(OBJS) $(LIBM) $(LDLIBS)
    $(LIBGCC)
romfs:
    $(ROMFSINST)/bin/$(EXEC)
    $(ROMFSINST)/pinyin/pymb
    $(ROMFSINST)/pinyin/lxmb
    $(ROMFSINST)/pinyin/ziku
clean:
    -rm -f $(EXEC) *.elf *.gdb *.o
```

执行make romfs时调用"romfs"部分,ROMFSINST是一个宏调用,其功能是将生成的可执行文件复制到目标机上的romfs目录,供打包烧写使用。

当执行make clean时调用该clean部分,清除编译过程中生成的中间代码。

3.3 应注意的几个问题

第一,一个汉字占两个字节,因此,在计算汉字起始位置时必须将前面的汉字个数乘以2。

第二,汉字拼音中没有以i、u、v开头的,在生成树模块中,构造的拼音码表也就没有该内容,在生成树中也就没有相应的结点。在处理用户输入过程中首先加以判断处理,若是以i、u、v开头的输入可以不到树中检索,以节约时间,否则,再从根结点开始检索。

第三,字库的构造可以采用Windows中的码表生成器。选择合适的码表进行逆转换后得到字库,再根据系统实际的需要,对字库进行相应的改造处理。

结论 汉字数组的实现方法简单,采用数组的方法要将汉字数组加载进内存,占用资源较大,适用于汉字不多的场合,比如在手持设备中,要求的汉字不多,或是经常需要的汉字比较固定,可以采用该方法。在数组实现方法中,汉字被以静态数组的形式加载到内存,没有字库,不易实现汉字的联想输入。并且在数组中,依靠下标顺序检索汉字,效率较低。

采用拼音树的方法,查找主要依靠指针传递地址,效率高,并且在程序运行之初,汉字库并不加载到内存,比较节约资源。同时字库的扩充比较方便,只需要对码表进行修改,而不需要修改源代码,便于应用程序的移植。

本文在uClinux环境下开发了拼音输入法,通过运行证明该算法可行,检索速度较高。

参考文献

1 黄俊贵,倪波.汉字与汉字排检方法[M].书目文献出版社,1990

2 严蔚敏,吴伟民.数据结构.清华大学出版社[M],1997

3 http://www.uclinux.org

(上接第210页)

个样本上使用两种方法的错识个数。决策树方法的错识个数明显少,原因是训练充分时它生成的判定分支要比人工预测全面。比较两条曲线,决策树方法的曲线波动比较小,说明其性能比较稳定,即对用户并不敏感,进一步说明了本文采用的方法具有不错的用户适应性。

结论 实验结果说明了本文采用的在线手写数字识别方法是快速高效的,识别的正确率和响应速度都是令人满意的,方法的良好用户适应性也通过实验得到了验证。

下一步的研究工作是如何得到更高的识别率和更好的用户适应性。有两个方面可以完善:第一,优化笔划走势特征的提取算法,更好地捕捉手写数字变体间"形似"的特性。第二,适当增加分类属性特征,以适应更多的手写数字变体。

参考文献

1 Guyon I,Warwick C. Handwriting as computer interface. Survey of the State of the Art in Human Language Technology, NSF. 1995

2 Hansmann U,Merk L,Nicklous M S,et al. Pervasive Computing. Second Edition. Berlin Heidelberg:Springer-Verlag,2001

3 Liu Cheng-Lin, Nakashima K, sako H, et al. Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition, 2003,36:2271~2285

4 Khorsheed M S. Off-Line Arabic Character Recognition - A Review. Pattern Analysis & Application, 2002,5:31~45

5 孙正兴,彭彬彬,丛兰兰,等,在线草图识别中的用户适应性研究.计算机辅助设计与图形学学报,2004,16(9):1207~1215

6 Vuori V,laaksonen J,Oja E,et al. On-Line Adaptation in Recognition of Handwritten Alphanumeric Characters. In: Fifth International Conference on Document Analysis and Recognition, Bangalore, India, 1999. 792

7 Artieres T, Marchand J-M, Gallinari P, et al. Stroke level modeling of on-line handwriting through multi-modal segmental models. IWFHR, 2000

8 Mitchell T M. Machine Learning. In: The McGraw-Hill Companies. Inc, 1997,39