

标准键盘布局比较与优化设计^①

张俊琳^{1,2}, 崔勇¹, 王弘毅¹

¹(清华大学 计算机科学与技术系, 北京 100084)

²(清华大学 水利水电工程系, 北京 100084)

摘要: 键盘作为基本的人机交互设备之一, 在多个领域有广泛的应用。对于标准键盘, 现有的 Qwerty 键盘布局并不是为提高工作效率而设计的。而键盘输入作为一个复杂的人体工程学问题, 依赖于操作者的习惯、不同手指的易用性差异以及不同的文本信息等诸多因素。在键盘输入过程分解、手指参数假定和语言文本分析之上, 通过定义距离矩阵和文本矩阵, 为标准键盘布局比较和优化设计建立较为客观的评价模型, 并对现有键盘布局进行比较。作为应用, 给出了针对汉语拼音输入的新键盘布局, 并利用实际文本进行了测试与比较。

关键词: 键盘; 布局比较; 优化设计; 距离矩阵; 文本矩阵

Layout Comparison and Optimal Design of the Standard Keyboard

ZHANG Jun-Lin^{1,2}, CUI Yong¹, WANG Hong-Yi¹

¹(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

²(Department of Hydraulic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: As one of the basic human interface devices, the keyboard has a versatile application in many fields. To standard keyboard, the commonly used Qwerty keyboard layout is not designed for improving work efficiency. The keyboard entry is a complicated human engineering problem, which varies with the factors such as typists' habits, usability divergence of fingers and different text information. By decomposing the keyboard entry process, assuming parameters about fingers and analyzing text information, the paper defines the Distance Matrix and the Text Matrix to build a more objective model for the layout comparison and optimal design of the standard keyboard. Comparison of the existent keyboard layouts is made and a new efficient layout for inputting Pinyin-based Chinese is established.

Key words: keyboard; layout comparison; optimal design; distance matrix; text matrix

标准键盘是电脑最基本的外设之一。目前最常见通用的布局方案是 Qwerty 键盘。该布局最初由 Christopher Latham Sholes 于 1868 年提出, 旨在解决打字速度过快导致的某些键组合卡键的问题。“由于 Qwerty 键盘被设计用于完成现在看来老式的机械需求, 所以打字速度最大化并不是其明确的目的。”^[1]为此, August Dvorak 依据平衡双手负担、常用字母出现在中排、更有力手指负担更多输入等原则, 在 1932 年设计了新的键盘布局, 从而提高了打字效率。但由于多方面原因^[1], Dvorak 键盘并没有被推广。近年来, 随着电子设备不断普及, 以及移动电子设备的发展,

关于键盘设计又有人提出很多新的想法。在标准键盘方面, 如 2006 年由 Shai Coleman 提出的 Colemak 键盘, 它没有改变原有快捷键(如 Ctrl+C 等)的位置, 同时提高了效率, 是 Qwerty 和 Dvorak 键盘的整合, 使人们更易接受, 促进了新的高效键盘的推广。

比较不同布局键盘的优劣, 应该以输入效率作为评判标准。最直接的手段是打字实验^[2-4]。但这种方法涉及太多人为的、主观的因素, 在不同环境下结果就会有所差异。比如早期对 Qwerty 和 Dvorak 键盘的几个比较实验的不同结果表明了这种方法的困难^[3]。另一种方法是建立评价模型, 将键盘输入过程按一定

① 收稿时间:2011-08-01;收到修改稿时间:2011-09-16

准则量化, 建立适当的评价函数。比如基于费茨法则 (Fitts's Law) 的比较和设计^[5,6]。费茨法则是人机交互领域里的重要模型, 用于计算快速移动到一个目标的所需时间。其常见形式是:

$$T = a + b \log_2(1 + D/W)$$

其中 T 为完成移动的平均时间, 常数 a, b 反映了设备固有的属性, D 代表开始点到目标中心的距离, W 代表了目标的尺寸。但是, 虽然费茨法则被广泛地应用, 它是否适用于标准键盘仍然存在诸多疑问。Richard I. Land 就曾质疑: “这种方法(指费茨法则)的有效范围仍然需要检验, 但是令人惊讶的广泛。”^[3] 还有如针对特定文本的算法优化。针对特定文本分析, 利用建立的比较函数, 对庞大的键盘布局空间, 基于一定算法进行比较, 是对一般语言输入的一种简化。比如 Jeffrey 等人将考虑了四项目标的函数作为遗传算法的适应度, 从而得到了因不同文本而迥异的优化布局^[7]。

在键盘优化设计时, 如果希望优化的键盘布局能成为一种较为普遍的、通用的标准, 依据某些特定类型的文本, 利用相关算法得到的“高速”输入布局是不可行的。另一方面, 键盘的设计原则也不应该仅是追求算法意义上的“速度”, 输入的准确性和舒适性也必须考虑到。因此, 一种通用的、较优的键盘布局, 应该建立在合理的文本分析之上, 并充分考虑到输入速度和输入习惯等因素。

本文首先对键盘输入过程进行分析, 据此定义“距离”作为评价函数, 并依据键位建立距离矩阵。距离矩阵是键盘布局比较的主要模型, 也是键盘设计分析的工具之一。与距离矩阵对应的文本矩阵, 将键盘布局设计所需的文本信息抽象出来。距离矩阵和文本矩阵为标准键盘的布局比较和优化设计提供了足够的信息。作为应用, 本文将对 Qwerty, Dvorak 和 Colemak 键盘布局进行比较。同时, 本文将突破 Dvorak 和 Colemak 只是针对英语的语言特征设计的局限, 给出一种针对汉语拼音输入的中文键盘布局优化方案, 并利用实际文本对新键盘布局进行一些测试与比较。

1 键盘输入过程分析

键盘输入是一个复杂的过程。打字者查看文本的同时也在输入。在输入过程中, 真正涉及到手与键盘交互的行为有两个动作: 手指对某个键位的敲击和手指的移动或准备移动以敲击下一键位。

现在来更细致地考察输入时的操作细节。深入分析两次按键之间, 从手指开始接触一个键并将要按下, 到手指开始接触第二个键的全过程, 可以将其简化分解为几个步骤, 以及对应的时间损耗(图 1)。

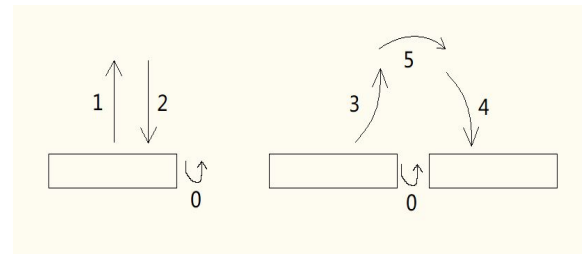


图 1 按键过程步骤分解

① 两次按下同一键位。手指按下该键(0), 按键损耗时间 T_0 。然后手指弹起(1), 耗时 T_1 。同一手指向下再次接触该键(2), 耗时 T_2 ;

② 两次按下不同的键位。同样手指按键时间 T_0 。然后手指弹起(3), 耗时 T_3 。在弹起(3)过程中, 其它手指(或者同一枚手指)将准备或已经开始向下运动以接触另一键。这里将时间损耗分为两部分: 手指向下运动时间 T_4 , 按键准备及手指反应时间 T_5 。对于 T_5 , 又可以将其分解为手指纵向移动的时间损耗 T_{5a} (如食指先敲击中排的键, 再敲击上排的键)和不同手指转换或横向移动时间损耗 T_{5b} (如食指先敲击中排的键, 再无名指敲击中排的键)。对于如食指先敲击中排的键, 在无名指敲击上排的键这类情况, 可以看作纵向移动和横向移动时间消耗的叠加。注意, T_3, T_4, T_5 在实际按键中并不是割裂的, 很大程度上, 这三段时间存在重叠, 甚至可能是同时发生的。这里只是为便于过程的分析而将它们独立出来, 作为一种简化的假设。

那么现在, 可以为键盘布局比较建立一定的模型。更优布局的键盘, 对于输入一定长度的文本, 应该花费更少的时间。从另一角度, 对于给定的时间, 应该能输入更长的文本。从这点出发, 比较键盘布局, 也可以从“移动长度”分析。下面将通过一些简化和假定, 引入距离的定义。

2 布局比较与优化设计模型

本节将通过考虑不同手指易用程度、左右手平衡、不同键位差异等输入因素, 将打字过程的手指运动参数化, 建立距离与距离矩阵的概念。然后利用距离矩

阵, 对不同键盘布局进行比较。对于新键盘的设计, 还需要考虑文本信息。与距离矩阵相对应, 根据文本信息, 可以得到文本矩阵, 对文本矩阵的分析为键盘布局优化设计提供了可行方法。通过距离矩阵和文本矩阵, 标准键盘布局比较与优化设计的模型就建立了。

2.1 手指运动参数假定

2.1.1 键位坐标化

为将时间转化为对应的“移动长度”, 首先对键位坐标化。记键位按顺序为 X1—X30, 并建立二维坐标系(图 2)。在此坐标系下, 每个键位都有一个坐标与之对应。如 X24(3,0)。而不同的键盘布局只是字符与 X1—X30 的对应关系, 如 Qwerty 键盘布局即为:

{ {X1:Q}, {X2:W}, ..., {X29: .}, {X30: /} }

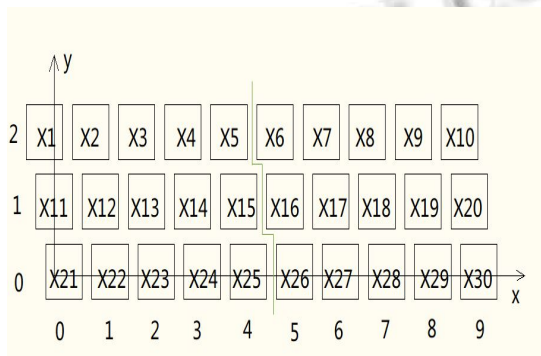


图 2 键位坐标化

2.1.2 距离与距离矩阵

考虑以上分析过的两次击键过程: 设先敲击键 $\alpha(x_1, y_1)$, 再敲击键 $\beta(x_2, y_2)$ 。

对于敲击任何一个键, 都包含时间 T_0 , 对不同布局没有差异, 故在计算比较时可以不考虑。

定义 1.

对于 T_1, T_2, T_3, T_4 , 由于它们反映的动作过程是类似的, 这里假定它们耗费时间相同, 对应有相同的移动长度 s 。对不同手指 s 取值不同, 其反映了不同手指的易用程度。

S 取值列表:

表 1 不同手指 S 取值

手指	小拇指	无名指	中指	食指
S (左手)	0.8	0.6	0.4	0.2
S (右手)	0.7	0.5	0.3	0.1

这样可取 $S(x)$ 与横坐标关系 x 如表 2:

表 2 $S(x)-x$ 关系表

x	0	1	2	3,4	5,6	7	8	9
$S(x)$	0.8	0.6	0.4	0.2	0.1	0.3	0.5	0.7

定义 2.1.

对 T_{5a} , 其反映手指离开中排去敲击上下排的纵向移动时间损耗, 对应有移动长度 D_a 。

在两次按键由不同枚手指完成时,

$$D_a = \begin{cases} 0.9, & y_2 = 2 \\ 0, & y_2 = 1 \\ 1, & y_2 = 0 \end{cases}$$

定义 2.2.

特别地, 如果两次敲击须要同一枚手指完成, 则 D_a 定义不同。在 $x_1 = x_2$ 或 $x_1, x_2 \in \{3,4\}$ (均为左手食指) 或 $x_1, x_2 \in \{5,6\}$ (均为右手食指) 时,

$$D_a = \begin{cases} 0.9(y_2 - y_1), & y_2 - y_1 \geq 0 \\ y_1 - y_2, & y_2 - y_1 < 0 \end{cases}$$

定义 3.

对 T_{5b} , 其反映不同手指间的转换所消耗的时间 (包括可能的水平移动), 对应有移动长度 D_b 。

$$D_b = \begin{cases} 0.6|x_1 - x_2|, & (x_1 - 4.5)(x_2 - 4.5) > 0 \\ 0.4, & (x_1 - 4.5)(x_2 - 4.5) < 0 \end{cases}$$

定义 4.

于是可以定义两次按键之间的距离:

$$D = S(x_1) + S(x_2) + D_a + D_b$$

比如敲击 X14 后敲击 X12, 则 $D = 0.2 + 0.6 + 0 + 1.2 = 2.0$; 敲击 X14 后敲击 X18, 则 $D = 0.2 + 0.3 + 0 + 0.4 = 0.9$ 。故后者距离较短, 费时较少, 效率较高。

如果将 X1 到 X30 的任意两键之间的距离列写在矩阵中, i 行 j 列权值 D_{ij} 表示先按下 X_i 后按下 X_j 的两键距离, 就得到了距离矩阵(图 3)。

图 3 距离矩阵

由于 D_a 项的存在, 距离矩阵并不是对称矩阵。它是基于手指运动参数假定而得到的结果。距离矩阵包含了键位分配足够的信息。对距离 D 所含项 $S(x)$ 、 D_a 、 D_b , $S(x)$ 反映了不同手指间的易用程度和左右手熟练度差异; D_a 反映中排较优且敲击上排键位比下排键位更易的原则; D_b 反映了按键位置差异和左右手的平衡分配。这些实际都是在设计键盘布局中须要考虑的。所以, 距离矩阵不只可以用于不同标准键盘之间的比较, 还是新键盘设计的一个分析工具。从根本上说, 比较与设计所追求的目的都是相同的, 即获得更高效的键盘布局。

分析距离矩阵, 可以得到很多有用的信息。如将 i 行所有元素绘成柱状, 可以清晰地看到先按 X_i 键后再按其它键的距离分布(图 4)。而将 i 列所有元素绘成柱状图, 可以看到先按其它键再按 X_i 键的距离分布。将每行元素总和绘成柱状图, 可以看到先按 X_i 键后按其它键的距离总和, 它从一定程度上反映了不同键位的总体输入难易情况。

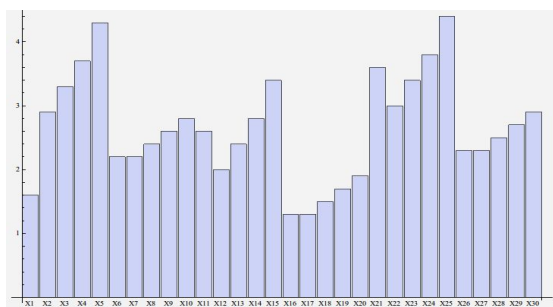


图 4 X_1 到其它键的距离分布

当然, 距离矩阵只是一种相当简化的模型。首先上述 $S(x)$ 反映的是一般右利手的打字者。其它的参数也只是一般假定, 更准确的结果应该通过实际的打字实验统计分析获得相对关系。而距离 D 也仅是一种相对简化的定义, 实际的键盘输入过程要复杂得多。但是简化模型已经可以反映出两次按键难易相对分布大致情况。比如距离矩阵中较小的距离值 $D_{24,16}$ 、 $D_{24,17}$ 、 $D_{14,16}$ 、 $D_{4,17}$ 、 $D_{14,17}$ 、 $D_{17,15}$ 等(均为 0.7), 正说明靠近键盘中部的按键过程距离较短, 输入效率较高。

2.2 布局比较

作为对距离矩阵的一个应用, 这里对 Qwerty, Dvorak 及 Colemak 键盘布局进行比较。由于三种键盘布局都是针对英语, 这里考虑他们对各个英语单词的

输入难易。对任一单词, 连续输入的过程实际上是上述分析中多个两次按键的组成。比如对单词 matrix, 输入整个单词的总距离应该是 m-a, a-t, t-r, r-i, i-x 的距离总和。而上面已经提及, 不同键盘布局只是英文字母和标点与 X_1 — X_{30} 的对应关系。这样, 搜索距离矩阵就可以获得不同布局下同一单词的不同距离。比如 matrix 在三种布局下距离分别为 10.7(Qwerty), 7.6(Dvorak), 10.1(Colemak)。

利用 Mathematica 提供的单词词库, 对其中 91926 个英文单词比较, 得到:

表 3 键盘布局比较

	Qwerty	Dvorak	Colemak
移动总距离	1.4141E6	1.1587E6	1.1443E6
较少单词数	/	78298(85%)	82781(90%)

由以上结果可以看到, 使用 Dvorak 和 Colemak 键盘, 移动总距离较 Qwerty 有大幅度减少, 也是多数单词的单个输入距离更短, 综合程度上提高了打字效率。而 Colemak 较 Dvorak 又有一定程度改善。考虑到人们已有的输入习惯, Colemak 键盘设计较 Dvorak 应该说是一种进步。

2.3 设计分析

实际上, 利用距离矩阵这一评价模型, 已经可以获得针对某一特定文本的最优布局, 即可以应用于新键盘的设计。但是, 依据某类特定的文本, 并不能成为普遍的、通用的标准。如果一个模型能充分地包含某一语言的输入情况, 那么依据这一模型设计的键盘布局才更为通用。

对应于距离矩阵, 这里提出文本矩阵的概念。考虑某一语言在键盘输入中所需要的全部字符, 将它们列写为矩阵的形式, i 行 j 列权值 T_{ij} 表示这一语言选取文本输入统计时所有出现先字符 c_i 后字符 c_j 这样形式的次数, 就得到了文本矩阵。足够数量的文本可以更接近这一语言的真实情况。通过填充足够数量的常用文本, 文本矩阵反映的是这一语言中相邻字符出现频率。而文本矩阵单个行或列的和可以反映单个字符在该语言中出现的频率。

距离矩阵包含了键位布局信息, 文本矩阵包含了语言频率信息, 两者都是同样阶数的权矩阵。通过字符与键位的映射关系, 正好将距离矩阵与文本矩阵联系起来。通过距离矩阵和文本矩阵, 应该说设计新的

更优的键盘布局的所需信息已经足够了。

3 针对中文输入的新键盘设计与比较

通过以上分析,对于任何一种可在标准键盘上输入的语言文字,只要收集到要分配于键位的字符以及依据该语言足够数量的文本建立这些字符的文本矩阵,通过综合分析文本矩阵,参照距离矩阵进行字符的键位分配,就可以得到新的优化的高效键盘布局。

使用由 OpenCC(开放中文转换)的作者提供的中文文本词库,对 136585 个含词频中文字词进行统计分析,得到中文的文本矩阵(图 5)。可以看到,中文文本矩阵的最大特征是有很多项为 0,即中文拼音输入中有些字符组合几乎不会出现。

图 5 中文文本矩阵

对文本矩阵同样可以作若干分析。将文本矩阵权值排序绘成柱状图,可以得到相邻字符出现频率分布;将各行综合排序绘成柱状图,可以得到单个字符出现频率分布。部分频率值如表 4。

然后依据距离矩阵的信息,可以设计出新的键盘

布局。这里按照如下做法进行设计:

- ① 保持常用快捷键及标点符号不变;
- ② 根据文本矩阵和距离矩阵,参考对应的柱状图信息,选取未分配字符中较高频率的单个字符分配到键位距离总和较小的未被分配的键位上。如有多个字符频率都较高,考虑其相邻字符频率情况,出现较高频率的相邻字符优先分配;
- ③ 确定某个字符分配后,将较高频率的相邻字符分配到该键位距离较小的其它键位中,同时考虑较高频率的单个字符的优先分配;
- ④ 分配过程中有多种选择的情况,如可以保持与 Qwerty 键盘字符分配相同,则尽量保持原位置不变。重复②、③分配完所有字符。

表 4 部分频率值(出现次数)

an	ng	ia	sh	in	ji
5.21E10	5.15E10	2.60E10	2.12E10	2.05E10	1.92E10
i	n	a	h	u	g
8.88E10	8.66E10	8.65E10	6.09E10	5.49E10	4.14E10

这样得到新的键盘布局如下:

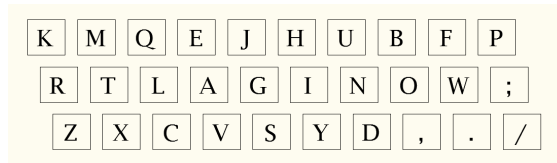


图 6 新键盘布局(针对中文)

对新键盘布局,给出针对实际文本的几个测试,其中移动总距离及部分分析结果如表 5。

表 5 结果分析

	移动总距离				新键盘与 Qwerty 比较	
	Qwerty	Dvorak	Colemak	New	移动距离减少比例	字词距离缩短个数比例
三体(约 16.7 万字)	652188	577933	572189	411206	36.9%	86.4%
北欧神话(约 6.4 万字)	223274	193341	193269	145494	34.8%	79.3%
徐志摩诗歌(32 首,约 1.1 万字)	43368.5	38821.6	37688	27054.8	37.6%	87.7%
The Little Prince(约 1.7 万词)	104820	84805.8	85644.7	90822.2	13.4%	63.7%
De Profundis(约 1.8 万词)	109148	88432.9	89265.8	94734.7	13.2%	62.2%

从这几份实际文本的比较可以看到,四种键盘中, Qwerty 的确是最低效的。Dvorak 和 Colemak 对五份文本移动总距离相差不大。对三份中文文本,新键盘布局相对其它三种键盘效率都有很大提高;对两份英文文本,新键盘输入效率不及 Dvorak 和 Colemak,但相对 Qwerty 仍有一定程度改善。

4 结语

在键盘输入过程步骤分解的基础上,本文通过定义距离,建立距离矩阵,给出了标准键盘布局比较的较为客观可行的方法。针对某一语言的足够数量的文
(下转第 87 页)

```
<iq id="NLz49-14" from="duwei@10.0.0.3/Smack" type="set">
  <query xmlns="jabber:iq:privacy">
    <active name="myPrivacyList"/>
  </query>
</iq>
```

图 6 用户请求激活隐私列表

通过以上信息分析可知,该隐身可见机制符合最初设计目标。

5 总结

在研究流媒体互动播控平台的基础上,针对其中的消息过滤和隐身可见机制进行了业务需求分析,结合 RFC3921、XEP-0045、XEP-0126 和 XEP-0016 等协议,设计了基于 XMPP 协议的人工审核和系统自动过滤的消息审核机制、隐私列表的基本单元和使用机制、针对群组管理员的隐身可见机制,并使用 smack 库验证了该设计方案。

(上接第 258 页)

本,可以建立文本矩阵。依据距离矩阵和文本矩阵,对于任一可在标准键盘输入的语言,其键盘布局优化设计的信息已经足够。最后本文给出了针对中文拼音输入的新键盘布局设计,并通过几份实际的文本测试,进行了相关效率比较。而要使新键盘布局方案得到推广,根本上还是改变人们长久以来的习惯,如果要广泛普及还有很多工作要做。

致谢

在此,向中文词库提供者郭家宝表示感谢。

参考文献

- 1 Liebowitz SJ, Margolis SE. The Fable of the Keys. *Journal of Law and Economics*, 1990,33(1):1-25.
- 2 Anderson AM, Mirka GA, Sharon M, Joines B, Kaber DB. Analysis of Alternative Keyboards Using Learning Curves. *Human Factors*, 2009,51(1):35-45.
- 3 Land RI. Keyboard Entry-Can it be Simplified. *ACM SIGSOC Bulletin*, 1982,13(3).
- 4 O'Brien MA, Rogers W, Fisk AD, Richman M. Assessing

参考文献

- 1 Peter SA. RFC3920: Extensible Messaging and Presence Protocol (XMPP): Core. Jabber Software Foundation <http://www.ietf.org/rfc/rfc3921.txt>. October 2004.
- 2 Peter SA. RFC3921: Extensible Messaging and Presence Protocol Protocol(XMPP): Instant Messaging and Presence. Jabber Software Foundation. <http://www.ietf.org/rfc/rfc3921.txt>. October 2004.
- 3 Peter SA, Smith K, Troncon R. XMPP: The Definitive Guide, O'Reilly: USA, April 2009.
- 4 Peter SA. XEP-0045: Multi-User Chat. Jabber Software Foundation. <http://xmpp.org/extensions/xep-0045.html> 2008
- 5 Peter SA. XEP-0126:Invisibility. XMPP Standards Foundation.<http://xmpp.org/extensions/xep-0126.html>. August 2005.
- 6 Peter M, Peter SA.XEP-0016:Privacy Lists. XMPP Standards Foundation. <http://xmpp.org/extensions/xep-0016.html>. Feb 2007.
- 7 Smack. Ignite Realtime: Smack API. <http://www.Ignitereal-time.org/projects/smack/2009>.

design features of virtual keyboards for text entry. *Human Factors*, 2008,50(4):680-698.

- 5 Zhai SM, Sue A, Accot J. Movement model, hits distribution and learning in virtual keyboarding. Dennis Wixon. *The SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves*. NY: ACM, 2002:17-24.
- 6 Francis G, Oxtoby C. Buiding and testing optimized keyboards for specific text entry. *Human Factors*, 2006, 48(2):279-287.
- 7 Goettl JS, Brugh AW, Julstrom BA. Call me e-mail: Arranging the keyboard with a permutation-coded genetic algorithm. In: Liebrock LM, ed. *The 2005 ACM symposium on Applied Computing*. NY: ACM, 2005:947-951.
- 8 卢亚军.藏文计算机通用键盘布局与输入法研究. *中文信息学报*,2006,20(2):78-86.
- 9 郝玉哲.人机工程型键盘. *中国安全科学学报*,1995,5(6):40-44.
- 10 方敏,金一庆,刘丹凤.英文输入用键盘布局的改进. *计算机工程与设计*,2004,25(1):153-160.