



(12)发明专利

(10)授权公告号 CN 1618173 B

(45)授权公告日 2016.08.17

(21)申请号 02827546.2

迈克尔·R·隆志

(22)申请日 2002.06.21

(74)专利代理机构 北京律诚同业知识产权代理有限公司 11006

(30)优先权数据

代理人 徐金国

10/176,933 2002.06.20 US

(85)PCT国际申请进入国家阶段日

(51)Int.Cl.

2004.07.26

G06F 17/27(2006.01)

(86)PCT国际申请的申请数据

H03M 11/00(2006.01)

PCT/US2002/019926 2002.06.21

G06K 9/18(2006.01)

(87)PCT国际申请的公布数据

(56)对比文件

W02004/001979 EN 2003.12.31

US 5828991 A, 1998.10.27,

US 5586198 A, 1996.12.17,

US 5623261 A, 1997.04.22,

(73)专利权人 泰吉克通讯股份有限公司

地址 美国华盛顿州

审查员 王燕

(72)发明人 艾森·罗伯特·布莱德弗德

凯思·卡尔斯·布尔费什

戴维·J·凯 皮姆·凡·默尔斯

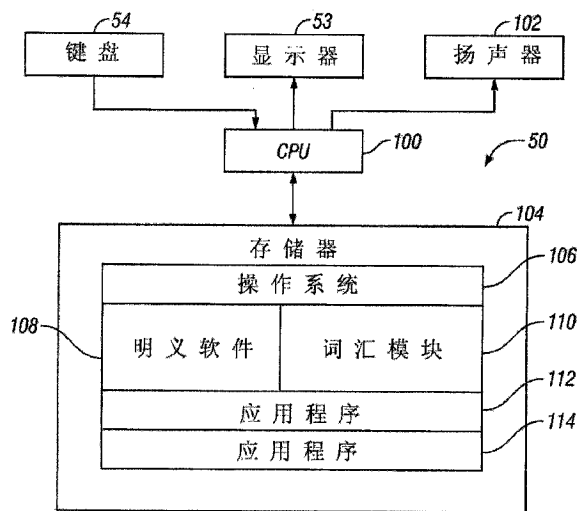
权利要求书5页 说明书37页 附图28页

(54)发明名称

多义性文字输入的明确字符过滤

(57)摘要

本发明涉及了一种用于在多义性文字输入中明确过滤的方法和装置。本发明提供了包括各种明确文字输入方法的实施例,诸如2键和长时间按下。本发明还提供了用于利用环绕构造方法、词干锁定方法、单词完整化方法、和n个字符列搜索来对数据库中的单词进行匹配的装置。



1. 用于多义性键输入序列的明确字符过滤的装置,所述装置包括:
显示器;
处理器,其中所述处理器管理向所述显示器的单词输出;
键盘,包括多个键,所述键经由合适的接口电路连接至所述处理器;
其中所述键的至少一些键具有多种含义,
其中与多种含义相关联的所述键的单一按键包括多义性键输入,以及
其中所述键盘允许用户不多义地指定每个字符作为按键的有序对;
其中所述处理器接受所述键盘的至少一个键输入序列,
其中所述至少一个键输入序列中的一个键输入序列包括来自单一按键的至少一个多义性键输入和至少一个明确输入的指定音节,其中所述音节包括一个或多个明确输入的指定字符,

其中所述一个或多个明确输入的指定字符被输入键输入序列中,并且用为每个字符的单个按键多义地指定剩余的字符;

其中所述装置被配置为从多个按键中识别每一所述一个或多个明确输入的指定字符,所述明确输入包括以下任意:

2键明确输入;
长时间按下一个键以显示并循环到一个字符序列;
长时间按下在一个键上以明确地输入一个数字/数;
改到数字模式并按下一个键以明确地输入一个数字/数;
改到多次轻敲模式并重复地按下一个键,以明确地输入一个字符;
通过把每对键的按下分组为2键明确输入,把多义性模式键的按下解释成明确字符;
通过把同一键的重复按下分组成多次轻敲输入,把多义性模式键的按下解释成明确字符;

利用多次转换键,由此允许关于简单按下的多义性输入,和关于不同种类按下的明确字符输入;

通过同时按下至少一个的键的联合,用主键表明一个多义性的字符组和一个副键表明要选择该组中的哪个字符;和

使用软键以根据与在前按键相关联的最有可能的字符的分析或根据在候选单词的当前列表中的单词,提供任一指定给一个键的字符;以及

存储器,连接至所述处理器,其中所述存储器包括单词数据库;和
明义过滤系统,所述明义过滤系统被配置用于相对于所述键输入序列中的多义性和明确字符来匹配所述单词数据库中的单词;以及

其中匹配的单词输出到所述显示器。

2. 根据权利要求1所述的装置,进一步包括语音合成软件和作为交流工具的扬声器。

3. 根据权利要求1所述的装置,其中所述装置进一步被配置用于通过2键明确输入来输入准确字符,所述用于通过2键明确输入来输入准确字符的手段包括使用以下至少一个:

矩阵显示;

标记,其包括相关联字符的预定组的一个子组;和

滚动列表。

4. 根据权利要求1所述的装置,其中所述装置被配置用于:
通过长时间按下键来输入准确字符以显示和循环字符序列;以及
至少以下之一:

在显示了所述字符序列之后,每按下所述键一次所述字符序列中的下一个字符就被高亮度显示,在一个键的按下和下一个键的按下之间经过了足够的时间后或当按下了不同的键时结束;或

在显示了所述字符序列之后,每次按下定位键,所述字符序列中的下一个字符就被高亮度显示。

5. 根据权利要求1所述的装置,其中所述明义过滤系统被配置用于只要可能就在所述单词数据库中对单词进行匹配。

6. 根据权利要求1所述的装置,其中所述装置进一步被配置用于从候选单词的所述过滤的组中选出想要的单词。

7. 根据权利要求6所述的装置,其中所述装置被配置为以预定的顺序呈现所述候选单词。

8. 根据权利要求1所述的装置,其中所述装置进一步被配置为围绕至少一个确定字符构造单词。

9. 根据权利要求8所述的装置,其中所述围绕至少一个确定字符构造单词进一步包括来自下列群组的至少之一:

通过对单词进行匹配而进行解释,其中所述单词包括为一特定序列而明确输入的字符;

通过对单词进行匹配而进行解释,其中所述单词包括至少一个单词,所述至少一个单词与在所述至少一个确定字符之前的多义性键相匹配;

通过对单词进行匹配而进行解释,其中所述单词包括至少一个单词,所述至少一个单词与以所述至少一个确定字符结尾的多义性键相匹配;

通过对单词进行匹配而进行解释,其中所述单词包括至少一个单词,所述至少一个单词以所述至少一个确定字符开始;或

通过对单词进行匹配而进行解释,其中所述单词包括至少一个单词,所述至少一个单词跟在所述至少一个确定字符之后。

10. 根据权利要求8所述的装置,其中所述围绕至少一个确定字符构造单词是通过按下下一键来驱动的,以获取由在所述单词数据库中的多次搜索所得到的下一组结果。

11. 根据权利要求1所述的装置,其中所述装置进一步被配置用于词干锁定,以通过把至少一个字符序列标记成明确输入的形式来避免重复解释所述至少一个字符序列。

12. 根据权利要求11所述的装置,其中所述装置进一步被配置用于包括来自下列群组的至少之一:

完全下一个锁定,其中下一个键锁定了一个所述至少一个字符序列;

通过利用方向键在一个所述至少一个字符序列上移动光标的无限制锁定;

通过在所述单词数据库中对单词进行匹配来利用所述至少一个键输入序列之前的先前明确输入的字符,其中所述单词以所述明确输入的字符开始并与所述至少一个键输入序列匹配;提出以所述明确输入的至少一个字符序列开始并紧挨着一个多义性键序列的单词

的完整形式；

允许按下一被指定为锁定功能的键；

允许按下一合适方向键以锁定正被提供的单词完整形成；

基于按下一个或多个所述键而锁定选择的词干；或

生成与存在所述单词数据库中的单词不同的单词。

13. 根据权利要求12所述的装置,其中所述完全下一个锁定还包括识别和使用单词边界。

14. 根据权利要求12所述的装置,其中所述生成与存在所述单词数据库中的单词不同的单词进一步包括:

接收包括一序列锁定的输入的第一组字符;

接收与第二组字符相关联的随后的多义性输入;

在所述单词数据库中搜索想要的单词,其利用所述接收的第一组字符和所述随后的多义性输入;

如果在所述单词数据库中发现所述想要的单词,那么允许对所述想要的单词进行选择;

如果在所述数据库中没有发现所述想要的单词,那么通过锁定执行至少一次后续搜索并在所述第二组字符的子组中进行搜索。

15. 用于多义性键输入序列的明确字符过滤的方法,所述方法包括如下步骤:

提供显示器;

提供处理器,其中所述处理器管理向所述显示器的单词输出;

提供键盘,所述键盘包括多个键,所述键经由合适的接口电路连接至所述处理器,其中所述键的至少一些键具有多种含义,其中与多种含义相关联的所述键的单一按键包括多义性键输入,而且

其中所述键盘允许用户不多义地指定每个字符作为按键的有序对;

其中所述处理器从所述键盘接受至少一个键输入序列,以及

其中所述至少一个键输入序列的键输入序列包括来自单一按键的至少一个多义性键输入和至少一个明确输入的指定音节,其中所述音节包括一个或多个明确输入的指定字符,

其中所述一个或多个明确输入的指定字符输入在所述键输入序列中,并且用为每个字符的单个按键多义地指定剩余的字符;

从多个按键中识别每一所述一个或多个明确输入的指定字符,所述明确输入技术包括以下任意:

2键明确输入;

长时间按下一个键以显示并循环一字符序列;

长时间按下在一个键上以明确地输入一个数字/数;

改到数字模式并按下一个键以明确地输入一个数字/数;

改到多次轻敲模式并重复地按下一个键,以明确地输入一个字符;

通过把每对键的按下分组为2键明确输入,把多义性模式键的按下解释成明确字符;

通过把同一键的重复按下分组成多次轻敲输入,把多义性模式键的按下解释成明确字

符；

利用多次转换键，由此允许关于简单按下的多义性输入，和关于不同种类按下的明确字符输入；

通过同时按下至少一个的键的联合，用主键表明一个多义性的字符组和一个副键表明要选择该组中的哪个字符；和

使用软键用于根据与在前按键相关联的最有可能的字符的分析或根据在候选单词的当前列表中的单词，提供任一指定给一个键的字符；

相对于键输入序列中的多义性和明确字符来对单词数据库中的单词进行匹配；以及将匹配的单词输出给所述显示器。

16. 根据权利要求15所述的方法，进一步包括提供语音合成软件和作为交流工具的扬声器。

17. 根据权利要求15所述的方法，进一步包括通过2键明确输入来输入准确字符，其中所述通过2键明确输入来输入准确字符的步骤进一步包括利用下列之至少一个：

矩阵显示；

标记，其包括相关联字符的预定组的一个子组；和

滚动列表。

18. 根据权利要求15所述的方法，进一步包括：

通过长时间按下键来输入准确字符以显示和循环字符序列；其中在显示所述字符序列之后，所述循环字符的步骤进一步包括至少以下步骤之一：

在显示了所述字符序列之后，每按下所述键一次所述字符序列中的下一个字符就被高亮度显示，在一个键的按下和下一个键的按下之间经过了足够的时间后或当按下了不同的键时结束；或

在显示了所述字符序列之后，每次按下定位键，所述字符序列中的下一个字符就被高亮度显示。

19. 根据权利要求15所述的方法，其中相对于多义性和明确字符匹配所述单词数据库中的所述单词的步骤包括只要可能就相对于多义性和明确字符在所述单词数据库中对所述单词进行匹配。

20. 根据权利要求15所述的方法，其中所述匹配的单词以预定的顺序呈现。

21. 根据权利要求15所述的方法，进一步包括：围绕至少一个准确字符构造单词。

22. 根据权利要求21所述的方法，所述围绕至少一个准确字符构造所述单词进一步包括由下列步骤组成的群组中的至少一个步骤：

通过对单词进行匹配而进行解释，其中所述单词包括在所述键输入序列中的第一的位置的所述至少一个准确字符；

通过对单词进行匹配而进行解释，其中所述单词包括至少一个单词，所述至少一个单词与在所述至少一个准确字符之前的多义性键相匹配；

通过对单词进行匹配而进行解释，其中所述单词包括至少一个单词，所述至少一个单词与以所述至少一个准确字符结尾的多义性键相匹配；

通过对单词进行匹配而进行解释，其中所述单词包括至少一个单词，所述至少一个单词以所述至少一个准确字符开始；或

通过对单词进行匹配而进行解释,其中所述单词包括至少一个单词,所述至少一个单词跟在所述至少一个准确字符之后。

23.根据权利要求21所述的方法,其中所述围绕至少一个准确字符构造所述单词的步骤是通过按下下一键来驱动的,以获取由在所述单词数据库中的多次搜索所得到的下一组结果。

24.根据权利要求15所述的方法,进一步包括:词干锁定,以通过把至少一个字符序列标记成明确输入的形式来避免重复解释所述至少一个字符序列。

25.根据权利要求24所述的方法,所述词干锁定进一步包括执行由下列步骤组成的群组中的至少一个步骤:

提供完全下一个锁定机制,其中下一个键锁定了一个所述至少一个字符序列;

提供通过利用方向键在一个所述至少一个字符序列上移动光标的无限制锁定;

通过在所述单词数据库中对单词进行匹配来利用当前键输入序列之前的先前明确输入的字符,其中所述单词以所述至少一个准确字符开始并与所述键输入序列匹配;

提出以所述明确输入的至少一个字符序列开始并紧挨着一个多义性键序列的单词的完整形式;

允许按下一被指定为锁定功能的键;

允许按下一合适方向键以锁定正被提供的单词完整形成;

基于按下一个或多个所述键而锁定选择的词干;或

生成与存在所述单词数据库中的单词不同的单词。

26.根据权利要求25所述的方法,所述完全下一个锁定机制进一步执行步骤:识别和使用单词边界。

27.根据权利要求25所述的方法,所述生成与存在所述单词数据库中的单词不同的单词的步骤进一步包括下列步骤:

接收包括一序列锁定的输入的第一组字符;

接收与第二组字符相关联的随后的键输入序列;

在所述单词数据库中搜索想要的单词,其利用所述接收的第一组字符和随后的键输入;

如果在所述数据库中发现所述想要的单词,那么允许对所述想要的单词进行选择;

如果在所述数据库中没有发现所述想要的单词,那么通过锁定执行至少一次后续搜索并在所述第二组字符的子组中进行搜索。

多义性文字输入的明确字符过滤

技术领域

[0001] 本发明涉及了文字输入技术。尤其是,本发明涉及了对带有有限的小键盘的无线通信装置的文字输入解决方案。

背景技术

[0002] 这些年来,便携式计算机已经变得越来越小。键盘,已经成了致力于生产更小型的便携式计算机的主要的尺寸限制部件。如果使用标准打字机大小的键,则便携式计算机必须至少象键盘那么大。已经在便携式计算机上应用了小型键盘,但是人们发现小型键盘的键太小了,以至于用户不能方便或迅速地进行操作。

[0003] 把实足尺寸的键盘组装在便携式计算机中,同样妨碍了计算机实际的便携式使用。如果不把它放到平坦的工作面上以便让用户用双手打字,则大多数计算都不能被操作。最新一代小型计算机,叫做个人数字助理(PDA)或掌上型计算机,多家公司已经通过在所述装置中结合手写识别软件来试图解决这个问题。用户可以通过在触控面板或荧光屏上书写来直接输入文字。接着,该手写的文字被识别软件转换成数字数据。可惜的是,除了下述事实以外,即用笔印下或写下通常要比打字慢,手写识别软件的准确性和速度现在并不令人满意。就日语来说,由于其大量复杂的字符,该问题变得尤为困难。使事情变得更糟的是,现在需要文字输入的手持式计算装置仍然在不断变小。在双向寻呼、蜂窝电话、和其他便携式无线技术方面的新进展,引发了对小型和便携式双向信息传递系统的需求、尤其是对即能够发送也能够接收电子信件(“e-mail)的系统的需求。

[0004] 从而,研制一种小型的和用户在用一只手拿着的同时可用另一只手进行操作的、用来向计算机输入文字的键盘,是有益的。先前的研制工作,已经考虑了采用一种减少了键数的键盘。如在按键式电话机的小键盘布局中所提出的,很多压缩键盘采用了3乘4的键序列。阵列中的若干键与多个字符相关联。从而,用户需要这样一种方法,即,指出哪个与给定键相关联的字符是想要的字符。

[0005] 一种提出的无多义性地指定在压缩键盘上所输入的平假名字符的方法,要求用户输入两个或多个按键以指定每个假名。可以同时地(和谐地chording)或按顺序地(多次敲击指定)输入所述按键。和谐地或多次敲击指定都不能产生一种使用起来足够简单且有效的键盘。多次敲击指定效率低,和谐方式通常复杂不易学习和使用。

[0006] 日语五十音图中的每个音节(syllable),都由单个元音或一个辅音跟随了一个元音组成。有两个例外是:没有元音的音节ん,和用来表示跟随辅音的发音的“双音(doubling)”或“促音(hardening)”的“小(small)”っ。这些音节能被写成平假名(通常在书写日本本国的单词时使用)或片假名(通常在书写外来单词时使用)。所述术语假名,是用于指平假名或片假名的。五十音图通常被表示成有行和列的表格(在表格1中示出),其中每行在相应于5个日语元音あ、い、う、え、和お的列中都有多达5项。每行都对应一个字首辅音,虽然给定的辅音可以为一行(例如,s(a)さ=>sh(i)し;t(a)た=>ts(u)っ;等等)中的某些项而经历变声。第8行由腭音化元音や、ゆ、和よ(在现代日语中不用YI和YE)。发音标记

ゝ和°用来表明辅音发音的变化,通常表明从清辅音到浊辅音的变化。表2给出了通过在表1中的音节上添加发音标记ゝ和°而构成的基本音节。小写形式的音节や、ゆ、和よ也被用来在第2位置处与音节结合,或表1和2“ゝ”列表示由相应辅音和腭音化元音组成的音节(例如,跟有“小”や的き表示きや)。因而,这些带有腭音化元音的音节被写成一对假名,如表3所示,其包括用发音标记所写的形式。

[0007]

がGA	ぎGI	ぐGU	げGE	ごGO
ざZA	じJI	ずZU	ぜZE	ぞZO
だDA	ぢJI	づZU	でDE	どDO
ばBA	びBI	ぶBU	べBE	ぼBO
ぱPA	ぴPI	ぷPU	ぺPE	ぽPO

[0008] 表3

きゃ KYA	きゅ KYU	きょ KYO
ぎゃ GYA	ぎゅ GYU	ぎょ GYO
しゃ SHA	しゅ SHU	しゅ SHO
じゃ JA	じゅ JU	じょ JO
ちゃ CHA	ちゅ CHU	ちゅ CHO
にゃ NYA	にゅ NYU	にょ NYO
ひゃ HYA	ひゅ HYU	ひょ HYO
びゃ BYA	びゅ BYU	びょ BYO
ぴゃ PYA	ぴゅ PYU	ぴょ PYO
みゃ MYA	みゅ MYU	みょ MYO
りゃ RYA	りゅ RYU	りょ RYO

[0009]

[0010] 日语的词典顺序,通常用表1第一列中的音节的顺序(按照元音A)表示:め、ガ、さ、た、な、は、ま、や、ら、わ、ん,其中这些字母中的每一个(除了ん)都表示多达5个音节的子类,以该顺序,从元音め、じ、う、え、お所组成。目前,需要日文输入的诸如蜂窝电话这样的产品,通常利用多次敲击指定的方法,其中9个键中的每一个都与前9行(め到ら)中的每一个相关联。用在键上的多次敲击,指示相应行的哪个字符是想要的,其中在键上的补充敲击,顺序地把将要输出的字符改变为表1或2的下一列中出现的字符。一种分开键或超时的方法被用来提供与同一键相关联的连续字符的输入。第十个键被用于音节わ、を、ん和片假名“bo”符号,其表明只有元音的音节重复在前音节的元音。“小”や、ゆ、上也与被关联到や键,其需要额外的按键以进行选择。通常用附加键来添加跟在音节后面的发音标记。

[0011] 利用压缩键盘输入日语平假名(或片假名),仍是个富有挑战性的问题。用如上所述的当前的多次敲击方法,生成单个假名音节需要平均来说最少3个按键。用两个字符表示的带有腭音化元音的音节(即,表3中由表1和2第二或“じ”列开始的音节后带有“小”や、ゆ或よ而组成的),可以需要多达8个按键来生成。因此,开发一种有助于将输入平假名所需要的按键数减为最小的、而且用起来简便直观的压缩键盘系统,将是人们所期望的。

[0012] 在压缩键盘上,键入除假名外还包括汉字字符(汉字)的标准日语文字,甚至是更富有挑战性的问题。在带有实足尺寸的键盘和大显示器的标准计算机上输入文字,通常通过下述方法来实现,即首先利用相应于表1-3所示的每个平假名的拉丁字母表中的字母(日

语中叫做“romaji(罗马字)”)来键入想要文字的发音。随着字母的键入,自动把输入转换为相应的平假名音节并显示在屏幕上。在很多情况下,接着用户需要把最初显示成平假名的所述文字转换成想要的特定文字解释。所显示的平假名表示了用户实际想要输入的汉字和平假名组合的语音读法(reading),其传达了用户的想要的意思。由于在日语中有大量的同音字,可以有多种相应于用户所输入的平假名的汉字和平假名的可能的有意义的组合。在标准计算机上,可以显示多个所述选择对象的转换,例如每个可选对象都与一个数字键相关联以便按下该键就把输入的平假名转换成所显示的汉字解释。当试图在小型手持式装置上实施该过程时,由于有限的显示尺寸和少量的可用键,又增加了复杂性。

[0013] 一种用于指定在压缩键盘上所输入的平假名的可选方法,允许用户用单个按键来输入每个平假名。压缩键盘的每个键都与多个平假名字符相关联。随着用户输入键序列,从而在所得到的输出中出现了多义性的问题,因为每个按键都可能表示几个平假名中的一个。因此,该系统必须提供一种手段,通过该手段用户能够有效地表明每个按键的可能解释中的哪一个是想要的。已经提出了几种方法,用于解决按键序列的多义性问题。

[0014] 在John L.Arnott和Muhammad Y.Javad于International Society for Augmentative and Alternative Communication杂志上发表的“Probabilistic Character Disambiguation for Reduced Keyboards Using Small Text Samples”(以下称为“Arnott一文”)中总结了多种所建议用来确定与多义性的按键序列对应的正确字符序列的方法。Arnott一文注意到大多数明义(也称去多义性或歧义消除)方法,利用了有关语言中字符序列的统计资料解决给定语境下的字符多义性。即,现有的明义系统统计性地分析正由用户输入的多义性按键分组以确定对这些按键的适当解释。Arnott一文还指出了几种明义系统,试图利用词级的去多义性对来自压缩键盘的文字解码。通过在接收到指示词的结束的明确字符之后,把所接收的按键序列和词典中的可能匹配进行比较,词级去多义性方法针对整个词去除多义性。Arnott论文讨论了词级去多义性的许多缺点。例如,由于在识别不常用单词上的显示以及不能解码词典中不包括的单词方面的限制,词级明义方法常常不能正确地对单词解码。由于解码上的限制,词级明义方法不能无错地对一次按键形成一个字符的无约束英语文本解码。从而Arnott一文倾力于字符级明义而不是词级明义,并且该文还指出了字符级明义看起来是最有前途的明义(或去多义性)技术。然而与字母语言相比,日语中的每个平假名字符都表示一个音节,而不是表示基本音素的单个字母。由于该原因,字符级明义在日语中是低效的,因为对平假名的可能序列几乎没有限制,并且平假名序列的可能性分布不够歪斜以使该方法能够有效。

[0015] 又一种提出的基于词级明义的方法是在1982年Academic Press出版的,作者为I.H.Written,书名为“Principles of Computer Speech”的教科书中(一下称为“Witten方法”)公开的。Witten讨论一种降低利用电话触摸垫输入文字的多义性系统。Witten认识到,对于一本24,500个词的词典中的大约92%的单词来说,在按键序列与词典进行比较时不会出现多义性。然而,Witten指出,一旦确实出现了多义性问题,其必须通过由系统将多义性提供给用户并要求用户在多个多义性项目中进行选择,来以交互方式进行选择。因此,用户必须对系统在每个单词末尾的预测做出反应。这种反应降低了系统的故障并增加了为输入给定的文字段落所需的按键数。

[0016] 在日语的情况下,由于语言中的大量同音字,单词处理软件的用户,习惯于随着单

词的输入从多个多义性解释中进行选择。假名的同一序列经常能够被转换成两个或多个不同的汉字解释。因此,在输入假名序列之后,通常要求所述用户从一组可能选择中选择想要的汉字转换,并经常还要求用户以某种方式确认所选的正确转换。当利用压缩键盘输入所述平假名时,还有关于用户实际想要作为将要被转换成汉字的平假名序列的多义性问题。结果,可能解释的数目被大大增加了。

[0017] 消除多义性按键序列的多义性,仍是个挑战性的问题。如上面的讨论中所指出的那样,使输入一段文字所需的按键次数为最小的满意解决办法不能达到可在便携式计算机的使用中可接受的所要求的效率。因此,人们想要开发一种明义系统,其解决了所述按键的多义性问题,同时使所需按键的总数最小,在简单并容易理解的用户接口的语境中。

[0018] 一种用于日语的高效的压缩键盘输入系统,必须满足所有的下述标准。首先,日语(假名)音节在键盘上的布局、和生成它们的方法,必须容易为讲本族语的人理解和容易使用。其次,所述系统必须有助于减少输入文字所需的按键数以提高压缩键盘系统的效率。第三,所述系统必须降低用户的认识负担,通过降低了在输入过程期间所需的注意力和决策的数量而减轻了用户的认知负担。第四,所述方法,应该减少在实施实际系统时所需的存储器和处理资源的数量。

[0019] Kisaichi et al. [JP 8-314920; US 5,786,776; EP 0 732 646 A2]公开了这样一种方法,其中电话小键盘的键1-0分别用平假名音节{あいうえお}、{かきくけこ}、{さしすせそ}、{たちつてと}、{なにぬねの}、{はひふへほ}、{まみむめも}、{やゆよ}、{らりるれろ}、和{わをん}标记。这符合日语电话小键盘的实际标准,其中电话小键盘的键1-9分别用平假名音节あ、か、さ、た、な、は、ま、や、和ら来标记。出现在每个键上的单个平假名,代表了指定给该键的全组的平假名,其相应于在表1中的整行平假名,其中在所述行中所述单个平假名出现在第一列。常常用{わをん}来明确地标记0键。

[0020] Kisaichi et al.公开了一种词级明义(或词级多义性消除)方法,其中用户通过按一次与每个字符相关联的键,来多义地(ambiguously)输入一个字符序列(平假名)。在每个输入序列的末尾,用户按下“Conversion/Next Candidate(转换/下一个候选项)”键来显示与所输入的键序列相关联的平假名的一个可能序列的第一个原文解释(textual interpretation)。Kisaichi et al.公开了一种词典结构,其中一个给定输入键序列的所有原文解释都被连续地存于相邻存储块中。如果有一个的话,每额外地按下“Conversion/Next Candidate”键就显示下一个存于所述词典中的原文解释。如果没有原文解释了,就会显示一条错误信息并可以执行任意的异常处理。当显示了想要的原文解释时,必须按下一特定的“Confirmation(确认)”键,以便在用户能够继续输入下一个文字对象前确认想要的文字已经被显示出来的。

[0021] Kisaichi et al.所公开的方法中存在若干难点。一个是,由于下述事实,即,在平假名字符串的指定和每个可能的平假名候选字符串的转换中,易于出现非常多的给定键序列的可能的原文解释。这就可能要求用户利用“Conversion/Next Candidate”键在非常多的解释中一步一步地来发现想要的解释。而且,在可能的解释中一步一步地进行的过程中,用户看见各种汉字和/或与由于输入的多义性而产生的各种平假名字符串相对应的平假名字符串。这能够分散人们的注意力,并需要用户在试图发现想要的解释的过程中额外地注意。此外,原文解释数据库被安排成所有数据都只由完整的单词组成,并且所有用于给定长

度的所有键序列的所有数据还被连续地存储于存储器的相邻存储块中。Kisaichi et al. 没有公开任何显示合适的词干的方法,其中所述词干相应于一个在输入序列中的那些点处不符合任一完整的单词的、较长的、仍不完整的单词。在输入中的这类点处,Kisaichi et al. 只能显示每个所述键的默认显示(indicate),诸如数字或默认的字母或字符。这就把用户搞糊涂了,并且不能提供有效地帮助用户确认想要的键已经被输入的反馈。最后,需要用户为每个所输入的单词按“Confirmation(确认)”键,需要用户必须为每个输入输入一个额外的按键。因此,Kisaichi et al. 所公开的系统没有满足上面所讨论过的标准。

[0022] 任何词级明义应用所面对的另一个重要挑战是,能否成功地将其应用于对其应用最为有利的各种硬件平台上。如上面所提到的,这种装置包括双向寻呼机、蜂窝电话、和其他手持式无线通信装置。这些系统使用电池供电,因此其硬件设计和资源利用都尽可能的节约。所设计的在这种系统上运行的应用程序,必须将处理器带宽的使用及存储器要求减小的最低。这两个因素的关系一般是相反的。因为词级明义系统要求有很大的单词数据库运行,并且必须能迅速地对输入的按键做出反应以提供令人满意的用户界面,所以能够压缩所需数据库而不致严重影响利用数据库所需的处理时间将是很大的有利之处。在日语的情况下,额外的信息必须被包括在数据库中以支持假名序列到用户想要的汉字的转换。

[0023] 任何词级明义应用所面对的另一个挑战,是要向用户提供有关正在输入的按键的足够的反馈。在使用普通的打字机或字处理器时,每个按键都代表一个可以在输入后尽快向用户显示的唯一字符。但是,在词级明义中这一点常常是不可能做到的,因为每个按键都代表多个字符,并且任何按键序列都可能与多个单词或词干匹配。因此,想要开发一种明义系统,其能使所输入按键的多义性最小,并且还使用户最高效地解决在文字输入时所出现的多义性问题。一种提高用户效率的方法,是在每次按键之后都提供合适的反馈,其包括在每次按键之后都显示最有可能的单词,并且倘若当前按键序列与一个完整的单词不符合,就显示仍不完整的单词的最有可能的词干。为了创造一种用于日语的有效的压缩键盘输入系统,已经设计了一种并不满足所有上述标准的系统。

[0024] 首先,日语(假名)音节在键盘上的排列、和产生它们的方法,对于说本国语的人来说是容易明白和使用的。其次,所述系统有助于减小输入文字时所需的按键次数。第三,所述系统通过降低了在输入过程期间所需的注意力和决策的数量而减轻了用户认知负担。第四,此处所公开的方法,有助于减少在实施实际系统时所需的存储器和处理资源的数量。

[0025] 多年以来,便携式计算机已经变得越来越小。键盘,已经成了致力于生产更小型的便携式计算机的主要的尺寸限制部件。如果使用标准打字机大小的键,则便携式计算机必须至少象键盘那么大。已经在便携式计算机上应用了小型键盘,但是人们发现小型键盘的键太小,以至于用户不能足够准确地方便或迅速地进行操作。把实足尺寸的键盘应用在组装在便携式计算机中,同样妨碍了计算机实际的便携式使用。大多数计算机如果不被放到平坦的工作面上以便让用户用双手打字,则不能被操作。用户不能在站立或移动的时候容易地使用便携式计算机。

[0026] 最新一代的小型便携式计算机,叫做个人数字助理(PDA),各公司已经通过在PDA中结合手写识别软件来试图解决这个问题。用户可以通过在触控面板或显示屏上书写来直接输入文字。接着,该手写的文字被识别软件转换成数字数据。可惜的是,除了下述事实以外,即用笔印下或写下通常要比打字慢,手写识别软件的准确性和速度现在并不令人满意。

[0027] 目前,在无线产业中的极大增长,已经生产了大量可靠、方便、和非常流行的、普通消费者可以买的起的移动通信装置,诸如蜂窝电话、双向寻呼机、PDA,等等。这些需要文字输入的手持式无线通信和计算装置,仍在变得越来越小。在双向寻呼机、蜂窝电话和其他便携式无线技术方面的新进展,引发了对小型和便携式双向信息传递系统的需求,尤其是对能够发送也能够接收电子信件(“e-mail”)的系统的的需求。一些无线通信装置厂商,也想向客户提供客户能够用拿着装置的手进行操作的装置。

[0028] 明义背景

[0029] 先前的研发工作已经考虑到了使用减少键数的键盘。正如按键式电话机的小键盘布局所建议的,很多压缩键盘已经采用3乘4的键阵列。所述键阵列中的每个键都含有多个字符。因此,在用户输入一个按键序列时就会出现多义性,因为每个键都可以表示几个字母中的一个。曾经提出过若干方法欲解决按键序列的多义性问题,其被称作明义。

[0030] 所提出的一种用于无多义性地指定在压缩键盘上所输入的字符的方法,平均起来需要用户输入两个以上按键以指定每个字母。按键可以同时输入(联合敲键)或顺序输入(多次敲键指定)。无论是联合敲键还是多次敲键指定都产生了一个足够简单并且使用有效的键盘。多次敲键指定效率低,而联合敲键复杂、不便学习和使用。

[0031] 所提出的另一种用于判断对应于多义性按键序列的正确字符序列的方法,被总结在下面的文章中:题目为“Probabilitstic Character Disambiguation for Reduced Keyboards Using Small Text Samples”,由John L.Arnott和Muhammad Y.Javad发表在Journal of International Society for Augmentative and Alternative Communicaiton中(以下称之为“Arnott一文”)。Arnott一文指出大多数明义方法应用关联语言中字符序列的公知统计数据来解决给定给定上下文中的字符多义性问题。

[0032] 又一种所提出的基于词级明义的方法,公开在标题为Principles of Computer Speech的一本教科书中,作者为I.H.Witten,并由Academic Press在1982年出版(以下称之为“Witten方法”)。Witten讨论的是一种减小利用电话触控垫输入文字的多义性的系统。Witten承认对于一本24500个单词的词典中的大约92%的单词来说,在按键序列与词典进行比较时不会出现多义性。然而,Witten指出,一旦确实出现了多义性问题,其必须通过由系统将多义性提供给用户并要求用户在多个多义性项目中进行选择,来以交互方式进行选择。因此,用户必须对系统在每个单词末尾的预测做出反应。这种反应降低了系统的故障并增加了为输入给定的文字段落所需的按键数。

[0033] H.A.Gutowitz,在名为“Touch-Typable Devices Based on Ambiguous Codes and Methods to Design Such Devices”、号码为W0 00/65091(2000年6月15号)的专利申请中公开了一种可键入装置的设计,特别是,包括有多义性代码的触摸键入装置,其提出了大量的人机工程学上的问题并建议了对这种问题的一些解决方案。Gutowitz讲解了从用力触摸可键入的多义性代码和基本最佳的多义性代码的分类中选择多义性代码的方法,其用于诸如计算机、PDA等类似装置和其他信息设备的触摸-键入装置,所述装置给出了在诸如尺寸、外型、该设备的计算性能上的设计限制、此类装置的一般性使用以及例如遵循英文字母排列顺序或是Qwerty式键盘的排列等的常规限制。

[0034] Eatoni Ergonomics公司,提供了一种叫做WordWise的系统(版权2001 EatoniErgonomics公司),其改编自常规键盘,并且其中在常规键盘上键入大写字母,并且

在按下带有想要的字母的键的同时按住诸如shift键这样的辅助键。WordWise背后的键思想,是从所述电话小键盘上的每个键上的每个字母组中选择一个字母。如此选择的字母,是通过在按下带有想要字母的键的同时按住一个辅助键来键入的。WordWise没有使用词汇数据库/词典来搜索单词以解决多义性、没有多义性、或其项目的组合的问题。

[0035] Zi公司讲解了一种预测文字的方法,eZiText^(R)(2002,Zi公司),但是并没有讲解或在任何地方建议在多义性模式中的明确文字过滤,也没有与2-键明确输入、词干锁定、或n个字符列搜索相结合。

[0036] 改善当前明义方法的需要

[0037] 多义性按键序列的明义仍是一个挑战性的问题。明义所面对的一个特殊挑战,是向用户提供有关正在输入按键的足够的反馈。在使用普通的打印机或字处理器时,每个按键都代表一个可以在输入后尽快向用户显示的唯一的字符。但是,在词级明义中,例如,这一点经常是不可能做到的,因为每个输入都代表多个字符,并且任一输入序列都可能与多个诸如单词或词干这样的对象相匹配。在用户拼写或按键出错而又没有把握判断是否出错,必须一直等到整个按键序列输入完毕而想要的单词未出现才知道时,这种多义性尤其是个问题。在其他例子中,原先的利用词级明义的系统,在被系统认定为结束选择,诸如特定键(例如space键)选择,的预定选择没有做出之前,不提供反馈。

[0038] 然而,有些字母表(alphabet),诸如泰国语和阿拉伯语,含有比英语字母表多的字母,这就导致了在数目减少的键方面存在甚至更大的多义性。这些语言的有效输入,要求一种在需要时降低所述多义性的机制。

[0039] 因此,提供一种在输入期间降低显示器上的明显的多义性、并提高用户检测和更正拼写和/或输入错误的能力的明义系统,将是有益的。

[0040] 提供一种下述这样的明义系统也是有益的,即,其通过明确过滤性能降低多义性并提高效率以便候选单词、词干列表和符号序列等类似的是缩小的,并且随后能够较迅速地向用户提供一单词的完整形式或序列的完整形式。尤其是,允许锁定一部分或全部先前所输入字符的解释的当前状态将是有益的,其包括但不限于对象和/或单词词干、所明确输入的字符,由此防止先前输入的重新解释。

[0041] 在明确输入字符的周围建立锚(anchor),用于新对象、单词、或单词词干的起始或结束,也将是有益的。

[0042] 提供合理的猜测以通过先前所明确输入的字符的n个字符列(n-gram)分析来扩展对象或单词,也将是有益的。

[0043] 降低多义性并提高在消除诸如单词或单词词干这样的语言对象的多义性的过程的效率,例如通过防止先前输入的重新解释,也将是有益的。

[0044] 识别多义性地或明确地输入的公共定界符作为建议的连接两个独立字符组的点,以建议按键解释将在何处重新开始,也将是有益的。

发明内容

[0045] 本发明涉及了一种用于在多义性文字输入中明确过滤的方法和装置。本发明提供了包括各种明确文字输入方法的实施例,诸如2键和长时间按下。本发明还提供了用于利用环绕构造方法、词干锁定方法、单词完善方法、和n个字符列搜索来对数据库中的单词进行

匹配的装置。

附图说明

[0046] 当结合附图参照下面详细描述来更好地理解本发明时,本发明前述各方面和许多相伴的优点将更为明显,其中:

[0047] 图1a,是含有本发明的压缩键盘明义系统的蜂窝电话的实施例的示意图;

[0048] 图1b,是蜂窝电话小键盘的示意图,其类似于图1a,但其中多个键中的每一个都与一个或多个罗马字符相关联。

[0049] 图1c是含有本发明的受限或没有显示性能的压缩键盘系统的实施例的蜂窝电话小键盘的示意图;

[0050] 图1d是含有本发明的有显示性能的压缩键盘系统的实施例的蜂窝电话小键盘的示意图,其示出了随着键2的触发的显示;

[0051] 图2a是图1a的压缩键盘明义系统的硬件方框图;

[0052] 图2b是含有本发明压缩键盘系统的便携式计算机触摸屏的实施例的示意图;

[0053] 图2c是图2b的触摸屏的示意图,其示出了随着与作为按键有序对中的第一个的音节か、き、く、け、和こ相关联的键的激活而显示的小键盘;

[0054] 图3给出了用于日语的压缩键盘明义系统的词级明义软件的实施例的流程图;

[0055] 图4是结合了本发明压缩键盘系统的便携式计算机的触摸屏的备选实施例的示意图;

[0056] 图5是结合了具有9个键的本发明压缩键盘系统的便携式计算机的触摸屏的又一备选实施例的示意图;

[0057] 图6比较了键与符号的实际相关性及包含出现在实际键上的字符的重音符号变化的逻辑相关性的情况;

[0058] 图7是关联了逻辑符号与键索引的表的例子;

[0059] 图8A示出了词汇模块的树节点中的数据的内部布局;

[0060] 图8B示出了构造语音拼写(Yomikata)文字对象的主指令的实施例的语义成分;

[0061] 图8C示出了用来构造词条(midashigo)文字对象的四个不同类型的副指令的实施例的语义成分;

[0062] 图8D示出了用来构造词条文字对象的四个不同类型中的两个的副指令的又一优选实施例的语义成分;

[0063] 图9示出了实施例中节点结构中的可能内部数据项目的四个例子;

[0064] 图10示出了一种未压缩词汇模块的优选树形结构;

[0065] 图11示出了从词汇模块内检索出来的过程中用于对象的中间存储的优选实施例的对象列表的状态的例子;

[0066] 图12是用于在按键列表给定时从词汇模块中检索文字对象的软件过程的优选实施例的流程图;

[0067] 图13是在单个按键给定了时用于遍历词汇模块的数形结构及改变对象列表状态的软件过程的实施例的流程图;

[0068] 图14是用于建立折叠压缩词汇模块的软件过程的实施例的流程图;

- [0069] 图15是折叠词汇模块的数形结构的软件过程实施例的流程图；
- [0070] 图16是用于定位词汇模块的树中与给定节点相比较具有最大冗余度的第二节点的软件过程的实施例的流程图；
- [0071] 图17是用于计算词汇模块的树中两个节点之间的冗余度的软件过程的优选实施例的流程图；
- [0072] 图18是示出了本发明数据库的部分内容的图表,用于在一个键上的三个连续按键的序列,其多义性地与音节か、き、く、け和こ相关联；
- [0073] 图19给出系统操作的三个代表性例子,其示出了图1a所示系统在输入文字的同时随着在所输入的键序列上的每个按键的文字显示内容；
- [0074] 图20是根据本发明的明确字符过滤系统的硬件方框图；
- [0075] 图21是根据本发明的无线手持式装置的图片,其示出了与键相关联的英语字母表的每个字母；
- [0076] 图22是根据本发明,0-9键上的欧洲字母图表和一些特殊符号及其各自的Shift键；
- [0077] 图23是根据本发明一个2键矩阵输入方法的示例性屏幕快照；
- [0078] 图24是用于在图20的多义性文字输入系统中的明确字符过滤的2键明确输入的流程图；
- [0079] 图25是根据本发明优选实施例在多义性模式中明确过滤的流程图；
- [0080] 图26是根据本发明用于生成没有存于数据库中的单词的方法的流程图；
- [0081] 图27示出了根据本发明通过长时间按住一个键来明确输入的一个例子；
- [0082] 图28根据本发明描述了环绕构造的例子的流程图。

具体实施方式

[0083] 用于日语的压缩键盘文字输入系统

[0084] 1. 系统结构和基本操作

[0085] 参照图1a,描述了根据本发明所构建的一种压缩键盘明义系统,其作为包含在具有显示器53的便携式蜂窝电话52。便携式蜂窝电话52含有在标准电话键上所实施的压缩键盘54。出于本申请的目的,术语“键盘”被广泛地定义,以包括任何输入部件,所述输入部件包括具有为键所定义的区域触摸屏、离散的机械式键、膜式键等等。在图1a中描述了键盘54中每个键上的假名的排列,其符合已经成为日本电话的实际标准的排列。应该注意到,由此键盘54与标准QWERTY键盘或包括至少46个键、为表1所示出的基本日语五十音图中的每个假名指定一个键的键盘相比,具有数目减少了的数据输入键。

[0086] 表1

[0087]

あA	いI	うU	えE	おO
かKA	きKI	くKU	けKE	こKO
さSA	しSHI	すSU	せSE	そSO
たTA	ちCHI	つTSU	てTE	とTO
なNA	にNI	ぬNU	ねNE	のNO

はHA	ひHI	ふFU	へHE	ほHO
まMA	みMI	むMU	めME	もMO
やYA	-	ゆYU	-	よYO
らRA	りRI	るRU	れRE	ろRO
わWA	-	-	-	をWO
んN	-	-	-	-

[0088] 更准确地说,在该实施例中所示出的优选键盘包括从1到0以3乘4阵列所排列的10个数据键,加上Select(选择)键60,和Convert(转换)键62。任选地,所述键盘还可以包括Clear(清除)键64以删除先前的按键;用于输入不多义的字符、数字、和符号的模式Mode键67;以及向先前所输入的假名添加浊音符号(dakuten)和半浊音符号(handakuten)的Diacritic键68。

[0089] 通过在压缩键盘54上的按键向明义系统输入数据。在第一优选实施例中,当用户利用键盘输入按键序列时,文字就被显示在电话显示器53上。当在显示空间受限的装置上实施所述系统时,诸如在图1a-d中所描述的蜂窝电话,只有当前所选或最有可能的单词对象被显示在正在生成的文字的插入点88处。当依次按下键以输入想要的单词时,就以某种区别格式显示相应于所述输入序列的最有可能的单词。在图1a所描述的优选实施例中,用下划的虚线显示了当前单词。如在下面的更详细的解释,当Select键60或Convert键62被按下后,所述下划的虚线被变为下划实线。

[0090] 在图2b和2c所示出的第二优选实施例中,在显示器53上定义了两个区以向用户显示信息。文字区,正如上面对第一实施例的描述,显示用户所输入的文字,起到文字输入和编辑的缓冲器的作用。如图2b和2c所示,选择列表区77,通常位于文字区66之下,其显示了相应于用户所输入的按键序列的单词和其他解释的列表。所述选择列表区,通过同时给出按频率的降序所显示的最常出现的输入按键序列的解释和另一个较少常出现的候选解释,帮助用户解决所输入按键中的多义性问题。

[0091] 在图2中给出了压缩键盘明义系统硬件的方框图。键盘54和显示器53通过合适的接口电路被连至处理器100。任选地,扬声器102也被连至所述处理器。处理器100接收来自键盘的输入,并管理所有向显示器和扬声器的输出。处理器100被连至存储器104。所述存储器包括诸如随机存储器(RAM)这样的临时存储器介质,和诸如只读存储器(ROM)、软盘、硬盘、或CD-ROM这样的永久存储器介质的组合。存储器104含有所有软件例行程序以管理系统的操作。优选地,所述存储器包括一个操作系统106、明义软件108、和在下面的其他细节中所讨论的相关的词汇模块110。任选地,所述存储器可以包括一个或多个应用程序112、114。应用程序的例子包括单词处理程序、软件词典、和外语翻译程序。语音合成软件还可以作为一个应用程序来提供,以允许所述压缩键盘明义系统充当交流工具。

[0092] 回到图1a,压缩键盘明义系统允许用户只用一只手快速输入文字或其他数据。利用压缩键盘54输入数据。从1到0的每个数据键,都具有多种在键的顶面上用多个字符、数字、和其他符号表示的含义。(处于本公开的目的,用数据键上的数字和字符(s)来标识每个数据键,例如,用3s来标识右上数据键。)因为每个键都具有多种含义,所以按键序列在其含义上是多义的。当用户输入数据时,在显示器的多个区中显示各种按键解释,以帮助用户解决任何多义性问题。在大屏幕显示器上,在选择列表区向用户显示提供输入按键序列的

可能解释的选择列表。选择列表中的第一个项被选为默认默认解释并且显示在文字区66的插入点88处。在优选实施例中,在插入点88处(并在大屏幕装置的所述选择列表区中)用在其下所画出的下划虚线来显示该项。这种格式表示,根据当前选择列表中最常出现的对象隐含地选择该对象。如果显示器包括如图2b中的选择列表区,这种格式就建立了插入点88处的对象和显示在选择列表区77中的同一对象之间的直观关系。在图1a中,没有显示列表,并且只在插入点88处显示默认对象(在Select键的任一触发之前首先被显示在选择列表中的对象),或者如果已经隐含地选择了一个的话就只显示当前所选对象。

[0093] 可以按几种方式对输入按键序列的可能解释的选择列表进行排序。在普通操作模式下,按键最初被解释成假名项目以拼出相应于想要单词的语音拼写(以下称之为“单词解释”)。例如,在图2b中所示出的,用户已经输入了按键序列カ、カ、和カ。当键被输入时,同时进行词汇模块查找以确定与按键序列相匹配的语音拼写。根据使用频率从词汇模块中返回所述语音拼写,首先列出最常使用的语音拼写。利用所述示例性按键序列,从词汇模块中确定出语音拼写“かか<”、“きか<”、和“きこ<”作为相应于所述按键序列的三个最有可能的语音拼写。在该选择列表中的这八个所确定出的单词中,“かか<”是最常使用的,因此把其作为默认解释和暂定地作为平假名文字公布在在插入点88处。如在图1a中所示出的,在按下Select键60之前,把第一个语音拼写作为默认解释并利用与众不同的格式公布在插入点88处。该格式表示在一个数据键上的随后的按键将添加到当前键序列上,而不是启动一个新序列。例如,如在图1a,所述与众不同的格式包括用下划虚线显示作为平假名文字的语音拼写。其他可能与语音拼写匹配的列表被保存在存储器中,并根据其相关频率而被排序。

[0094] 在优选实施例中,随着相应于想要的语音拼写的按键序列的输入,所述用户简单地按下Select键60。在插入点88处默认语音拼写“かか<”下面的下划虚线,就被下划实线所取代。如果所显示的默认语音拼写,不是想要的语音拼写,则Select键60就被重复按下直到想要的语音拼写出现为止。在一个优选实施例中,在通过重复触发Select键60已经显示了存储器中与键序列相匹配的所有语音拼写之后,所述键序列被解释成一个数字(number),其中每个按键都产生在键标记上的数(digit)。这就允许了无需单独的数字模式就能够产生数字,并且还起到了可容易识别语音拼写解释的选择列表末端标记的作用。Select键60的下次按下,循环回选择列表内的第一个语音拼写。

[0095] 一旦显示了想要的语音拼写,如果想要的词条(原文解释)实际上与已经以平假名文字所显示的语音拼写相同,则用户就继续按下相应于将要输入的下一个语音拼写的第一个假名的数据键。另一方面,如果想要的词条包括汉字、汉字加平假名、片假名、或它们的某一组合,用户就按下Convert键62。这就使得所显示的语音拼写被在词汇模块中与该语音拼写相关联的最常出现的词条所取代。Convert键62的重复按下,按照频率的降序来用其他相关联的词条取代所显示的词条。在一优选实施例中,在已经通过重复触发Convert键62显示了存储器中所有与所选语音拼写相关联的词条之后,所选语音拼写作为片假名而被显示。这就使得片假名单词能够无需单独模式就被产生,并且还起到了可容易识别词条解释的选择列表末端标记的作用。在又一优选实施例中,如果用户希望选择与第一个(默认的)和所输入的键序列相关联的语音拼写相关联的词条,则可以立即按下Convert键62以获取想要的词条而不需先按下Select键60。

[0096] 在一次或多次按下Select键60和/或Convert键62之后,被按下的任一数据键除去

所显示的语音拼写或词条的特殊格式,并成为系统所解释的新的键序列的第一个按键。不需特殊的按键来确认在先按键序列的解释。

[0097] 在上面所描述的优选实施例中,按下Select键60向前循环通过存储器中与当前键序列相关联的所有语音拼写(按频率的降序)。在又一优选实施例中,按住Select键60一段预定时间阈值后,按照频率的降序向后循环通过存储器中所有的语音拼写。因此,当数字解释被包括在如上所述的存储器中相关联的语音拼写序列的末端时,在任何常规按下Select键60之前按住Select键60,立即向后循环到所述数字解释。重复按住Select键,则按频率的升序倒退循环通过所有相关联的语音拼写。同样地,按住Convert键62,按照频率的升序向后循环通过所有与当前所选语音拼写相关联的词条。类似地,在任何常规按下Convert键62之前第一次按住Convert键62,立即向后循环到片假名解释。

[0098] 仍参照图1a,在又一优选实施例中,当输入数据键时,所述Clear键64能够被按下以删除先前所输入的数据键。如果在当前键序列中的所有数据键都如此被删除了,按下Clear键64就会删掉文字显示器53上在当前选择列表为空时显示标准文字光标之处的插入点88左侧的字符。在一次或多次按下Select键60和/或Convert键62之后,按下Clear键64用当前键序列的默认语音拼写解释来代替插入点88处的当前所选文字解释,但是不从键序列中删除任何所述数据键。换句话说,在按下Select键60和/或Convert键62任意次数后第一次按下Clear键64,会有效地“删除”所有对Select键60和/或Convert键62的触发,在第一次按下Select键60或Convert键62之前立即把所述系统返回到所述状态。在又一优选实施例中,在一次或多次按下Convert键62之后,按下Select键60用与词条相关联的语音拼写代替在插入点88处的当前所选的词条。再次按下Select键60继续向前循环,从该点直到存储器中与当前键序列相关联的其他语音拼写(按频率的降序)。

[0099] 在又一优选实施例中,任一其他装置的触发起到了结束当前键序列的作用,其中所述其他装置明确地产生一个不多义的字符(诸如输入一个特殊Symbol模式并按下与单个特定字符无多义地相关联的键)。结果,除掉了在插入点88处所显示的语音拼写或词条的任一特定格式(在优选实施例中的虚或实的下划线),并且所述特定不多义的字符被添加到在新插入点88处的输出单词。

[0100] 暂时地把所选语音拼写或词条公布在所述文字区的插入点88处,以允许用户把其注意力保持在所述文字区上,而不需查阅选择列表。根据用户的选择,还可以对所述系统进行配置,以便一旦收到对Select键60(或Convert键62)的第一次按下,暂时地在插入点88处显示的所述语音拼写(或词条)就能够扩展(垂直或水平)以显示一份当先选择列表。所述用户可以选择在该份选择列表中所要显示的最大数量的单词。可选地,所述用户可以决定让选择列表总是显示在插入点88处,甚至是在Select键的第一次触发之前。所述明义系统把下一个单词的开始(多义数据键的触发或明确的不多义字符的产生所示意的)解释成是对当前所选项目是想要的项目的确认。因此,所选单词保留在插入点88处,作为用户的选择,所述下划线完全消失并且重新以不带特殊格式的标准文字重新显示所述单词。在大多数文字输入中,用户想拿按键序列作为构成语音拼写的假名。然而,应该明白,与每个键相关联的所述多个字符和符号允许单个按键和按键序列具有几种解释。在优选压缩键盘明义系统中,自动确定各种不同的解释并在解释按键序列的同时向用户进行显示,且作为单词列表向用户显示。

[0101] 例如,根据词干解释所述按键序列,其中所述词干相应于用户可能正在输入的假名的 可能的有效序列(在下文为“词干解释”)。不像单词解释,词干是不完整的单词。通过显示上个按键的可能解释,所述词干允许用户容易地确认已经输入了正确的按键,或者当他的注意力已经被转向所述单词中间时继续键入。有这样的键序列,即其符合长单词或短语的部分项目,但不符合任一完整单词或短语。在这种情况下,能够提供给用户的最有用的反馈,将示出相应于已经被输入直到该点的该单词词干的假名。在图2b中所示出的例子,按键序列カ、カ、カ能够被解释成构成了有效的词干“かけき”(产生了单词“かけきん”)。因此该词干解释作为选择列表中的项而被提供。优选地,按照所有可能单词的集合的组合频率,对该词干解释进行排序,其中所述可能的单词可以是根据每个词干由在数据键上的其他按键所生成的。用户可以在系统中选择或配置将被显示的最大数目和最小组合频率,以便有些词干解释不会被显示。当在选择列表中列出了词干解释时,如果词干解释与前述选择列表中的单词重复了,就忽略该词干。然而,当所述词干被忽略了时,可以用一个符号来标记相应于该被忽略的词干的单词,以显示还有用该单词作为其词干的更长的单词。词干解释,通过让用户确认已经输入了正确的按键来向用户提供反馈,来输入该用户想要的单词。

[0102] 图3是明义软件主例程序的流程图,其处理选择列表并确定在插入点88处将要显示什么以帮助用户消除多义性按键序列的多义性。在方框150,所述系统等待接收来自键盘54的按键。一收到按键,就在判定方框151进行检查,以确定所收到的按键是否是模式选择键。如果是,在方框172,所述系统就设置一个标志,以指示当前系统模式。在判定方框173,进行检查,以确定所述系统模式是否已经变了。如果是,在方框171,所述显示就被更新为反映当前系统模式所需要的显示。如果在方框151确定所述按键不是模式选择键,就在判定方框152进行检查,以确定所收到的按键是否是Select键。如果所述按键不是Select键,则在判定方框152A,进行检查以确定所收到的按键是否为Convert键。如果所述按键不是Convert键,则在判定方框153,进行检查以确定所述系统是否处于诸如明确Symbols(符号)模式这样的特定的明确字符模式。如果是,在判定方框166,就执行一个检查以确定任一暂时所选项是否出现在选择列表中。如果是,在方框167,所述项就被接收并被作为标准文字输出。接着,在方框168,相应于所述按键的明确的字符,被输出到文字区域中。接着,在判定方框169,进行检查以确定所述系统模式是否应该自动改变,就象在Symobls模式的情况下一样。如果是,执行进行到方框170并且所述系统模式被返回到先前的现用(active)模式,否则执行返回到方框150。

[0103] 如果在方框153,没有明确字符模式是现用的(active),在方框154所述按键就被添加到所存储的按键序列中。在方框156,从系统的词汇模块中,识别出相应于所述按键序列的对象。词汇模块是与按键序列相关联的对象库。对象是一段已存储的数据,根据所接收的按键序列对其进行检索。例如,词汇模块组内的对象包括数字、字符、单词、单词的组成部分、词干、短语、或系统函数和宏指令。下面的表中简要说明了每种对象:

[0104]

对象	相应的数据
数字	某数字,每个数都相应于单词按键,例如,两个数的序列“42”。

字符	一个相应于成对按键的字符或字符序列,例如,三个字符的序列“きか<”。利用每对按键以便使用无多义性地输入字符的两次敲击指定方法来消除多义性。
单词	相应于单词或多次按键的语音拼写或词条,例如,四个字符单词“かけきん”。
词干	一假名序列,表示构成一个单词的一个更长的假名序列的有效部分,例如,“かけき”是“かけきん”的词干。
短语	用户定义的或系统定义的相应于单次或多次按键的短语,例如,“おねがいします”。
系统宏指令	说明系统或用户定义的词及相关代码,例如,“<clear>”,其用于消除当前文字区域。除了描述单词,在词汇模块中系统宏指定对象和为执行规定函数所需的可执行代码相关联。

[0105] 虽然上面讨论了优选的词汇对象,但是可以明白其他对象也可被设想。例如,可以把图形对象与所存储的图形图象相关联,或可以把语音对象和所存储的语音片段相关联。还可以设想拼写对象,其把常见的拼错的单词以及键入错误的按键序列和单词的正确拼写连接起来。为了简化处理,每个词汇模块最好包含类似的对象。然而应该能够明白,在一个词汇模块中可以混合各种对象。

[0106] 返回到图3,在方框156,在每个词汇模块中识别出那些相应于所收到的按键序列的对象。在方框158-165,通过在词汇模块中进行查找而被发现的对象,被按照优先顺序进行排列(prioritize),以确定向用户显示对象的顺序。为了确定在选择列表内显示对象的顺序,在每个词汇模块之间和从每个词汇模块所返回的对象之间,建立优先次序。

[0107] 为了对从各种词汇模块中所识别出来的对象列表按照优先顺序进行排列,在方框158,对压缩键盘名义系统的操作模式进行检查。如上所讨论的,在标准操作模式中,所述单词解释(语音拼写和词条)首先被显示在选择列表中。因此,由单词词汇模块所生成的对象列表,将被指定一个高于由其他词汇模块所生成的对象列表的优先级。相反地,所述所述明义系统处于数字操作模式,则数字解释将被指定一个高于其他词汇模块的优先级。因此明义系统的模式规定了词汇模块对象列表之间的优先级。应该明白,在某些模式中,由某一词汇模块所生成的对象列表可以被完全从选择列表中忽略。

[0108] 由词汇模块所生成的对象列表,可以只包含单个项目,或其可以包含多个项目。由此,在方框160,如果所述对象列表含有多个项目,就解出由同一词汇模块所生成的对象之间的优先级。也对与特定按键序列相匹配的、在给定词汇模块中所查到的所述对象赋予优先级,该优先级确定了它们之间的相对显示(关系)。如上所提到的,默认显示顺序最好是在有代表性的惯用法语料库中的递减使用频率。从而,与和每个对象相关的优先级用于对选择列表中的对象排序。

[0109] 通过访问适当的系统菜单,很多与在词汇模块中所找到的对象的显示相关联的特性,是用户可编程的。例如,所述用户能够指定单个对象或各类对象在选择列表中的顺序。所述用户还可以设置优先等级,以便确定词汇模块之间的以及从每个词汇模块所识别出的对象之间的优先级。

[0110] 在解决了对象之间的优先级之后,在方框165,由识别出的对象构造选择列表并显

示给用户。作为用户输入多义性按键序列的默认解释,如在图1a和1c中所示,暂时在文字区53的插入点88处公布选择列表中的第一个项目并增强其亮度。接着,所述明义软件例行程序返回到方框150以等待下一个按键。

[0111] 如果检测到的按键是Select键60,则从判定方框152取“是”分支到达判定方框163,在方框163确定当前选择列表是否为空。如果是,则执行返回到方框150。如果在判定方框163选择列表不是空的,则取“否”分支到方框174。在方框174,在暂时被公布在插入点88处的所显示的默认语音拼写下面的下划虚线,被变为下划实线。接着在方框175,所述系统等待检测由用户输入的下一个按键。一收到按键,在判定方框176,就进行检查以确定下一个按键是否为Select键。如果下一个按键是Select键,在方框178,所述系统就把选择列表中的下一个语音拼写提前并把其标记为当前所选项目。在方框179,在插入点处暂时显示带有下划实线的当前所选项目。接着,所述例行程序返回到方框175,以检测用户所输入的下一个按键。应该明白,由方框175-179所形成的循环,允许用户通过多次按下Select键选择所输入的多义性按键序列的使用频率较小的各种语音拼写解释。

[0112] 如果下一个按键不是Select键,就在判定方框177进行检查以确定下一个按键是否为Convert键。如果所检测到的按键实Convert键,则从判定方框177取“是”分支到方框190,在方框190与当前语音拼写相关联的第一个词条被标记为所选项目并且在插入点88处暂时地显示带有下划实线的词条文字。接着在方框191,所述系统等待检测用户输入的下一个按键。一收到按键,就在判定方框192,进行检查以确定下一个按键是Select键。如果所述下一个按键是Select键,在方框196,所述系统就把当前所选项目改变到与当前所选词条相关联的语音拼写,并把其标记为当前所选项目,并且接着在方框179象以前一样进行。如果在判定方框192,下一个按键不是Select键,就在判定方框193进行检查,以确定文字按键是否为Convert键。如果是Convert键,接着,在方框194,所述当前所选对象就被提前到下一个与当前语音拼写相关联的词条,并被标记为所选项目。所述系统接着返回到方框191,以等待检测用户所输入的下一个按键。

[0113] 如果在判定方框177或193,下一个按键不是Convert键,则所述例行程序继续到方框180,在方框180暂时显示的项目被选为按键序列解释并被转换成文字取中标准文字格式。在方框184,从系统存储器中清除老的按键序列,因为在Select键或Convert键之后的多义性按键序列的接收,向系统表示了一个新的多义性序列的开始。接着,在方框154,新收到的按键被用来启动新的按键序列。因为把使用频率最高的语音拼写解释作为默认选择显示,所以明义软件的主例行程序允许用户在需要额外地触发Select键的情况最少的条件下连续地输入文字。

[0114] 图1b,是蜂窝电话小键盘的示意图,与图1a类似。压缩键盘54'包括多个数据输入键21'-30'。一个或多个数据输入键与多个罗马字符(用来按照发音拼写日语假名字符的发音的拉丁字符)相关联,并且用每个与所述键相关联的罗马字符对其进行标记。每次由输入装置的用户操作选择了一个输入,都会产生一个输入序列。所产生的输入序列具有一原文解释,该原文解释由于多个罗马字符与一个或多个数据输入键相关联而是多义性的。所述系统的该实施例,在概念上与图1a所示出的非常类似,但是其不需要发音键68,因为通过使用不同的拉丁字母而用罗马字表示带有发音标记的假名。例如,假名か用罗马字表示为“KA”,而带有浊音发音符的同一假名(が)用罗马字表示为“GA”。

[0115] 在标准操作模式中,所述按键最初被解释成相应于假名的罗马字序列的输入,以拼写相应于想要的单词解释的语音拼写。例如,如在图1b中所示出的,已经由用户输入了按键序列5KLM、4HIJ、5KLM、1ABC、5KLM和9TUV。在键被输入时,同时执行词汇模块查找,以确定与按键序列相匹配的语音拼写。根据使用频率从词汇模块中返回所述语音拼写,首先列出最常用的语音拼写。利益哦那个示例性的按键序列,所述语音拼写KIKAKU(“きかく”)和MIKAKU(“みかく”),作为相应于所述按键序列的两个最有可能的语音拼写,而被从词汇模块中识别出来。在选择列表中的两个已识别出的语音拼写中,KAKAKU“きかく”是最常用的,因此把其作为默认解释并暂时在插入点88'处被公布为平假名文字。如在图1b中所示出的,在按下Select键60'之前,该第一个语音拼写被看作是默认解释并被用与众不同的格式公布在插入点88'处。用罗马字指定语音拼写,需要大约平均起来象用假名的相应的指定的两倍之多的字符(以及由此两倍之多的键选择)。因此,通常在图1b中所示出的系统将会比图1a中所示出的产生较少的多义性选择,因为当在分布在同样数目的输入键上(10)上输入了两倍之多的按键时在统计学上指定更多信息。

[0116] II. 明义文字输入方法

[0117] 本发明还提供了一种方法,用于日语的压缩键盘,其允许用户不多义地指定每个想要的假名作为按键的有序对。所述日语五十音图包括108个音节(从标准大小的つ、や、ゆ、和よ开始,把“小”的つ、や、ゆ、和よ都算为独立的音节,因为他们书写和发音方式不同)。有一些其他不常用的音节,诸如“小”形式的元音音节あ、い、う、え、和お,其主要地仅仅用于片假名中。这些不常用的音节,在连同象在下面所讨论的显示一起使用时,还可以容易地由系统产生。在108个标准音节中,有37个是通过简单地把发音标记”或°添加到其他71个音节中的一个而产生的。这些不带发音标记的71个音节,能够被逻辑地组织成9或10行8或10列的单个矩阵,如在下面的详细解释。用两个假名标记本发明键盘上的多个键,一个表示与矩阵中给定行相关联的辅音,第二个假名表示与矩阵中给定列相关联的元音。

[0118] 对于用108个音节中的106个的日语的说本族语的人来说,所述组织结构是合乎逻辑并且直观的,并且用来生成剩余的两个音节(“小”つ和ん)的方法也是简单易学的。每个音节都由单对按键生成,包括带有用腭发音的元音的音节,该音节用两个单独的假名表示(例如,K YA,KYU,和KYO)。这就使得所需按键数显著地少于通过当前所使用的在压缩键盘上多次敲击输入假名的方法所需的按键数。因此,本发明提供了一种压缩键盘,其能被讲日语的本族人容易理解和快速掌握,并且其在降低输入按键序列的长度方面是有效的。

[0119] 在优选实施例中,日语五十音图中的71个音节组织在表4a中所示出的矩阵中。在通常情况下,其包括出现在表4a前8列的全部69个音节,相应于有序对的第一个按键,确定了将要输出的音节的辅音,第二个按键确定了元音。两个剩余的音节(“小”つ和ん)是在下面所讨论的例外的情况。通过生成在表4a的矩阵中所出现的相应的基础音节,来输出没有在表4a中示出的剩余37音节,接着利用单独的键添加发音标记。图1c给出了受限的或不具备显示能力的蜂窝电话的示意图,其具有结合了本发明压缩键盘系统具体实施例的用于日语的小键盘。10个键121到130中的每一个,都用一个假名标记来在键区的左上方进行标记,该假名标记是表4a中标记为“Key1”的列中的行标题(在下文为“行标记假名”),同时用一个假名标记来在键区的右下方进行标记,该假名标记是表4a中标记为“Key2”的行中的列标题(在下文为“列标记假名”)。在优选实施例中,当输入两个键序列时产生出现在表4a中的

音节,其中第一个键是在左上方标记了行标记假名的键,该行标记假名相应于音节所在的行,第二个键是在右下方标记了列标记假名的键,该列标记假名相应于音节所在的列。所述10个行标记假名,以标准词典顺序出现在图1c所示键盘的10个键121-130的左上方。前5个列标记假名以其标准的词典顺序出现在键盘的前5个键121-125的右下方,接下来“小”的や、ゆ、和よ(也以标准的词典顺序)出现在后三个键126-128上。最后,“小”つ出现在下一个键129上,之后的ん出现在键130上。通过触发Diacritic(发音)键131一次生成发声音符[˘]和连着触发两次以生成发声音符[˚],可以把发音标记添加到任一音节上。当把发声音符添加到腭音化元音音节(由两个输出假名表示,所述假名包括一个由表4中的“い”列后面带有“小”や、ゆ、和よ的音节构成)上时,所述发声音符就紧根在两个假名中的第一个假名的后面而被在其正确位置处添加到所述输出上。

[0120] 因此,对于出现在表4a前8列中的69个音节来说,相应于有序对的第一个按键,确定了将要输出的音节的辅音字母,第二个按键确定了元音。在其他优选实施例中,显示器用来向用户提供反馈。一收到有序对中的第一个按键,系统就显示能够由每个键所生成的、作为有序对的第二个按键有效的各种音节。哪个键生成哪个音节之间的关联,可以通过用与产生所述音节的键相关联的数字(或其他标识符)来标记每个音节,来表示。作为选择,可以用符合相应键的布局的几何布局的方式来显示(带或不带数字标记)音节。例如,在诸如带有文字显示的蜂窝电话这样的装置的情况下,音节能够被显示成相应于电话小键盘的1到9个键的布局的3乘3的矩阵。当以该方式使用显示器(display)时,甚至很少用到的音节,诸如“小”形式的元音音节あ、い、う、え、和お,都能被很容易地生成。例如,利用表4b中所示出的矩阵,一按下有序对的第一个键,其中第一个键相应于表4b顶行中只有元音的音节,所述显示器显示了与键1到5上的第二按键相关联的标准大小形式的元音音节あ、い、う、え、和お,并且“小”形式的元音音节あ、い、う、え、和お与键6到0相关联。

[0121] 图1d给出了包括文字显示性能的蜂窝电话的示意图,其具有结合了本发明压缩键盘系统具体实施例的用于日语的小键盘。10个键121到130中的每一个,都只用表4b中的一个行标记假名,在键区的中央进行标记。所述10个行标记假名,以标准词典顺序出现在图1d所示键盘的10个键121-130上。图1d给出了在触发了键122以作为按键的一个有序对的第一个键(与か、き、く、け和こ多义性地相关联)之后的显示器的外观。在优选实施例中,当触发了用行标记假名所标记的键时,就显示在表4b的相应行中出现的音节。所述显示器示出了假名和每个键之间的关联,其中所述假名是即将产生的假名,所述键是可能被按下作为按键有序对的第二个键的键。表4b中的列,是用数字进行标记的,所述标记还出现在图1d所示出的蜂窝电话的每个键上。应该注意,在该优选实施例中,带有腭音化元音的三个音节(きや、きゆ、和きよ)与键127-129(用数7、8、9标记的)相关联,并因此这些键能够显示在显示器的单行上。这就使用户更容易把这些相关的音节与将要产生它们的键关联起来,因为不仅是根据键上的数字标记和所显示的音节进行关联,还根据显示器上相同的3乘3音节矩阵布局和小键盘中的键进行关联。如前所述,通过触发Diacritic键131一次以产生发声音符[˘]和连续触发Diacritic键131两次以产生发声音符[˚],来把发音标记添加到任一音节上。当把发声音符添加到带有腭音化音节(与图1d所示出的例子中的键127-129上的第二按键相关联)上时,所述发声音符就紧根在两个假名中的第一个假名的后面而被在其正确位置处添加到所述输出上。

[0122] 图2b和2c给出了本系统的优选实施例,当作为在便携式计算机触摸屏上所显示的键盘而被实施时,有可能出现该实施例。图2b给出了键盘150,其是在按键的有序对的第一个按键之前所显示的键盘。图2c给出键盘170,其是在触发了图2b中的与音节か、き、く、け和こ相关联的键152以作为按键有序对的第一个之后所显示的。在图2c中,键171-179分别用假名进行标记,其中所述假名是当键被触发时将为输出所生成的。键和音节的该种布局,依赖于和图1d的显示器上所示出的相同的3乘3矩阵布局,利用表4b所示出的音节的同一概念上的组织结构把小键盘的键和音节关联了起来。

[0123] 在又一优选实施例中,相应于表4a或4b中的空单元的两个键序列的输入,产生了非输出(no output),且根据用户的选项,产生一可听错误信号。在一替换实施例中,相应于双击图1c中的键129(用列标记假名“小”つ标记)的单元,还产生一“小”つ。在又一替换实施例中,标记了列标记假名“小”つの列中的所有单元,也产生“小”つ。同样地,在又一替换实施例中,标记了列标记假名“小”んの列中的所有单元,产生ん。对于本领域普通技术人员来说,很显然地,能够对表4a和4b做出各种改变而不脱离本发明的范围。例如,“小”つ和ん是可以各种方法进行指定的特殊情况,诸如通过把他们指定到单独的键上进行指定,其中该键被触发一次就产生“小”つ以及被连续触发两次就产生ん。本发明的更重要的方面,是表4a前8列(或表4b的列1-5和7-9)中的音节指定的方法。

[0124] 表4b

Key 1 \ Key 2	1	2	3	4	5	6	7	8	9	0
ア	あ A	い I	う U	え E	お O	あ (A)	い (I)	う (U)	え (E)	お (O)
カ	か KA	き KI	く KU	け KE	こ KO		きゃ KYA	きゅ KYU	きょ KYO	
サ	さ SA	し SH	す SU	せ SE	そ SO		しゃ SHA	しゅ SHU	しょ SHO	
タ	た TA	ち CH	つ TSU	て TE	と TO	っ (TSU)	ちゃ CHA	ちゅ CHU	ちよ CHYO	
ナ	な NA	に NI	ぬ NU	ね NE	の NO		にゃ NYA	にゅ NYU	にょ NYO	
ハ	は HA	ひ HI	ふ FU	へ HE	ほ HO		ひゃ HYA	ひゅ HYU	ひょ HYO	
マ	ま MA	み MI	む MU	め ME	も MO		みゃ MYA	みゅ MYU	みょ MYO	
ヤ	や YA		ゆ YU		よ YO		ゃ (YA)	ゅ (YU)	ょ (YO)	
ラ	ら RA	り RI	る RU	れ RE	ろ RO		りゃ RYA	りゅ RYU	りょ RYO	
ワ	わ WA			え WO			ん N		ん (NN)	

[0126] 对于本领域普通技术人员来说,很显然地,尽管对图1c中所示具体实施例的键的行标记假名和列标记假名的指定是最自然的,但是替换实施例也是可能的,其中行标记假名和列标记假名以其他组合被指定给键而不脱离本发明的范围。图4和5给出了很多可能的替换实施例中的一些,其中行标记假名和列标记假名以一替换布局指定给键。

[0127] 在一替换实施例中,本发明的系统还可以在如图5所示的只有9个键的键盘上实施。表4a 71个不同音节中的68个,还可以被象表5所示的那样组织,其中已经删除了相应于行标记や的行。在这种9个键的系统中,三个音节や、ゆ和よ由有序对键序列生成,其中第一个键是相应于表第一行中的单元的行标记あ。音节“小”つ和ん,通过相应于最后一列的单元的键序列而生成,其包括被合并起来的表4a的最后两列。

[0128] 表4a

[0129]

Key 1 \ Key 2	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ン
ア	あ A	い I	う U	え E	お O				
カ	か KA	き KI	く KU	け KE	こ KO	きゃ KYA	きゅ KYU	きょ KYO	
サ	さ SA	し SHI	す SU	せ SE	そ SO	しゃ SHA	しゅ SHU	しょ SHO	
タ	た TA	ち CHI	つ TSU	て TE	と TO	ちゃ CHA	ちゅ CHU	ちょ CHO	っ(TSU)
ナ	な NA	に NI	ぬ NU	ね NE	の NO	にゃ NYA	にゅ NYU	にょ NYO	
ハ	は HA	ひ HI	ふ FU	へ HE	ほ HO	ひゃ HYA	ひゅ HYU	ひょ HYO	
マ	ま MA	み MI	む MU	め ME	も MO	みゃ MYA	みゅ MYU	みょ MYO	
ヤ	や YA		ゆ YU		よ YO	ゃ (YA)	ゅ (YU)	ょ (YO)	
ラ	ら RA	り RI	る RU	れ RE	ろ RO	りゃ RYA	りゅ RYU	りょ RYO	
ワン	わ WA				を WO				ん N

[0130] 表5

[0131]

Key 1 \ Key 2	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ン
ア	あ A	い I	う U	え E	お O	や YA	ゆ YU	よ YO	
カ	か KA	き KI	く KU	け KE	こ KO	きゃ KYA	きゅ KYU	きょ KYO	
サ	さ SA	し SHI	す SU	せ SE	そ SO	しゃ SHA	しゅ SHU	しょ SHO	
タ	た TA	ち CHI	つ TSU	て TE	と TO	ちゃ CHA	ちゅ CHU	ちょ CHO	っ(TSU)
ナ	な NA	に NI	ぬ NU	ね NE	の NO	にゃ NYA	にゅ NYU	にょ NYO	
ハ	は HA	ひ HI	ふ FU	へ HE	ほ HO	ひゃ HYA	ひゅ HYU	ひょ HYO	
マ	ま MA	み MI	む MU	め ME	も MO	みゃ MYA	みゅ MYU	みょ MYO	
ラ	ら RA	り RI	る RU	れ RE	ろ RO	りゃ RYA	りゅ RYU	りょ RYO	
ワン	わ WA				を WO				ん N

[0132] 在本发明的又一方面中,系统向用户提供了这样的指示,其关于下一按键是否作为有序对的第一个按键(确定将要输出的音节的辅音)、或作为有序对的第二个按键(确定将要输出的音节的元音)而将要被处理。例如,该指示可以作为一对被交替点亮的标记LED而被实施。替换地,可以在显示器上交替地显示两个不同的图标。对于本领域普通技术人员来说,很显然地,在不脱离本发明范围的情况下,还可能有这种指示的很多其他替换实施例。

[0133] 如图1c和2b所示把适当标记指定给键的综合影响;根据按键有序对音节的产生;如图1d和2c所示在有序对的第一个按键之后的合适的反馈的提供;在键标记和标准排列次序及日语五十音图布局之间的自然对应;和向用户的指示的包括,其中所述指示关于下一个按键是否相应于指定了音节辅音的行标记假名或指定了元音的列标记假名,产生了一种既对于不多义的输入文字来说效率高又容易被讲本族语的日本人理解和迅速掌握的系统。因此,利用具有少量实足尺寸的键且易学易用的键盘,实现了文字的高速不多义的输入。

[0134] III. 结合了多义性和不多义性按键的文字输入方法

[0135] 仍在本发明的其他方面,结合了音节的多义性和不多义性指定,以实现输入方法中的更高的效率。在一个优选实施例中,根据用户的选项,利用上面所公开的两个按键的方法,不多义地指定将要输入的单词的第一个音节。应该注意,在带有腭音化元音的音节的情况下,所述前两个按键产生了对两个平假名字符(包括表示腭音化元音的小や、ゆ和よ)的指定。用为每个音节的单个按键,多义地指定单词或短语剩余的音节。在象一个或两个音节那么短的单词的情况下,这种组合方法能够显著地降低输入序列的多义性,并由此降低用户可能需要对其进行检查以发现想要的读法(reading)的候选读法的数量。在又一替换实施例中,如在本领域中公知的多次敲击指定方法,可以被用来只不多义地指定单词或短语的第一个音节,和多义地指定所述音节的剩下的部分。

[0136] 仍在又一替换实施例中,利用本发明的多义性指定方法,所述两次敲击方法可以被用来指定单词或短语中想要的音节(而不是只指定第一个)。例如,把键按住超过某一时间阈值,可以表示在紧接着之后的按键是两个按键中的第二个,以便不多义地指定想要的

音节。该方法有两个优点。一是,任一音节(而不是只有单词或短语的第一个音节)都可以被不多义地指定。二是用户能够选择不多义地指定一个音节,而且是仅在其被认为这样将提高输入效率时。对于一些少见的单词和短语(尤其是对于较短的单词来说)来说,最常用的单词和短语的默认显示,可能需要多次触发Select键以选择想要的语音拼写,当所有的音节被多义地指定了时。在这种情况下,不多义地指定至少一个音节,能够使产生想要的语音拼写所需的按键明显较少,因为用户只需在那些在相同位置(s)处共享相同的被不多义地指定的音节(s)的语音拼写中进行选择。

[0137] IV. 提供明义或多义按键的数据库结构

[0138] 利用一种或多种数形数据结构,把用来消除输入序列多义性的单词和短语的数据库存储在词汇模块中。相应于特定按键序列的单词,是由存于数形结构中的数据构造的,所述数形结构呈指令形式,所述指令对紧接在前按键序列的单词和词干的组进行修改。因此,随着序列中的每个新按键的按下,与该按键相关联的指令集被用来创建与所述按键序列相关联的单词和词干的新组,其中所述按键序列已经添加了新的按键。用该方法,单词和词干不被明确地存于数据库中,他们是根据用来对其进行访问的按键序列来构造的。

[0139] 在是日语的情况下,数形数据结构包括两个类型的指令。主指令创建存于词汇模块中的单词和短语的语音拼写,所述词汇模块由相应于所述单词和短语的发音的假名序列组成。与每个语音拼写相对应的是一列副指令,该指令创建了与语音拼写相关联的词条。每个语音拼写都由主指令创建,其对与紧接在前按键序列的一个语音拼写相关联的进行修改。同样,每个词条都由副指令创建,其对与语音拼写相关联的一个词条进行修改,其中所述语音拼写由与该副指令相关联的主指令对其进行修改。

[0140] 因为每个主指令都指一个已知键,所以关于哪个特定假名将被添加的信息,作为逻辑索引而被优选地存储在与该键相关联的假名集(the set of kana)中。在图6中,画出了单个键540的有代表性的图。在优选实施例中该键的内部逻辑表示,不必反映物理布局。例如541,是在日语词汇模块中对与“か”键相关联的键的优选逻辑描述。4个附加的符号(き、く、け、こ)也与该键相关联。同样地,按照其在日语词典中(く、き、け、こ)使用频率的降低的顺序而对所述符号进行索引。通过扩展,图7是把逻辑符号的索引与键索引联系起来的优选表格,所述键索引在消除按键变成日语单词的多义性中被使用。图7给出了相应于图1a所示出的优选实施例的表格,其具有单独的Diacritic键68,当在前的字符将要带着浊音或半浊音符号出现时,就用该键进行指定。按下Diacritic键68,相对于浊音或半浊音符号是否将被添加到在前的假名上来说,是多义的(或有歧义的)。在一替换实施例中,Diacritic键68是不多义的,并为了浊音符号而被按下一次以及为了指定半浊音符号而被连续按下两次。在又一优选实施例中,给定假名的所有形式(带或不带浊音或半浊音符号),都与同一按键相关联,并且还将出现在Logical Symbol Index Table(逻辑符号索引表)的同一行中。这就允许利用数据库的系统,被建立,以便Diacritic键68 就被接受并且把新字母附加在其上。在这种系统中,如果要求使用Diacritica键68的选项被取消(turn off)了,指定带有发音符号的假名的添加的指令就被略过去,直到Diacritica键68已经被按下一次(为浊音符号)或两次(为半浊音符号)。

[0141] 图10示出了单词对象词汇模块1010中的树的代表性图。根据相应的按键序列使用树形数据结构来组织词汇模块中的对象。如图10所示,词汇模块中的每个节点N001、

N002、……N011,代表一个特定按键序列。树中的节点通过路径P001、P002、……P011连接。因为在所述明义系统的优选实施例中,有11个多义性数据键,所以词汇模块树中的每个父节点可与11个子节点相连。通过路径连接的节点代表有效按键序列,而一个节点缺少路径代表无效按键序列,即不与任何所存储的单词相对应的按键序列。应该注意,在无效输入按键序列的情况下,优选实施例的系统仍将能够构造并显示输入键序列的数字解释。在插入点的数字解释的出现(没有按下Select键),向用户表明了词汇模块中含有相应于所述输入键序列的非单词。

[0142] 词汇模块树根据收到的按键序列进行遍历。例如,按下自根节点1011开始的第二数据键,使与第一键相关联的数据被从根节点1011内部取出并被评估,之后遍历到节点N002的路径P002。第二次按下第二数据键,使与第一键相关联的数据被从节点N002内部取出并被评估,之后遍历到节点N102的路径P102。如将在下面更加详细地描述的,每个节点都与相应于按键序列的多个对象相关联。当收到每个按键并处理相应的节点时,由相应于所述按键序列的对象生成一个对象列表。明义系统的主例程序利用每个词汇模块的对象列表来生成一个选择列表。

[0143] 图8A是与每个节点相关联的优选数据结构400的框图。数据结构包含在词汇模块树中连接每个父节点到子节点的信息。所述树结构还包含用于识别与节点所代表的特定按键序列相关联的对象的信息(指令)。

[0144] 在节点数据结构400中的第一字段,是有效键位字段402,其标明了连至父节点的子节点的数目和标志,和11个可能键中的哪些与所述信息相关联,以识别(构造)与由所述节点代表的特定按键序列相关联的对象。由于在该优选实施例中有11个数据键,所以最多可以有11个子节点与任一父节点相连,因此有效键位字段中提供了11个有效键以表示子节点的存在或不存在。每个有效键位与包含词汇模块中的相应各子节点数据结构的指针的指针字段404a、404b、……404n相关联。因为只有与子节点相关联的按键是与父节点相关联的按键序列的有效继续时,子节点才存在,所以指针字段数目对每个节点来说是不同的。例如,有效键位字段402可以表示可能的11个按键中只有6个,产生了有效子节点。因为只有6个有效路径,所以在父节点的数据结构中只包含6个指针字段。有效键位字段402是用来查明包含在所述节点数据结构内的指针字段的标志。如果按键没有产生有效子节点,则相关联的指针字段被从节点数据结构中去掉,以便节省用来存储词汇模块所需要的存储空间。

[0145] 与每个节点相关联的有多个与节点所代表的按键序列相对应的对象。每个对象都由附在具体有效键上的分组(包)408中的406中的指令描述,该有效键由包含在节点数据结构中的有效键位字段402中的位模式表明。

[0146] 每个分组406中的每个指令,都描述了一个与每个节点所代表的按键序列相对应的对象。描述一个对象需要维持两个对象列表。图11描述了明义软件过程由词汇模块树中的父与子动态创建的代表性对象列表。对象列表430,是一个含有与代表两个按键的节点相关联的对象1-N1的对象列表。对象列表440,是一个含有与代表三个按键的节点相关联的对象1-N2的对象列表。每个对象列表都包含与每个节点相关联的所有对象的列表。对象列表430与由图1a的键盘所生成的按键序列2办2办的父节点相关联。对象列表440与按键序列2办2办的子节点相关联。应该明白,对象列表的大小是根据每个节点相关联的大小的实际数目而变化的。

[0147] 与子节点相关联的每个语音拼写对象,是通过一个主指令构造的,该指令在由父节点所构造的对象上添加一个假名字符。因此,在图8a中的指令分组406中的每个指令,都含有一个在图8B中示出的OBJECT-LIST-INDEX字段556,其从父节点对象列表中识别出一个用来构造子节点对象的对象。例如,参见图11,在老对象列表430中的第三对象“きか”,被用来构造新对象列表440中的第二对象“きか<”。因此,原来的对象标识符字段OBJECT-LIST-INDEX556提供了一个连接给老对象列表中的项目,以识别用来构造新对象的老对象。

[0148] 在指令558中含有一个LOGICAL-SYMBOL-INDEX(逻辑符号索引)字段555以指示要添加到识别出的对象上的符号,以便构造新对象。因此,所述LOGICAL-SYMBOL-INDEX字段就从节点按键序列中的最后的键指定将要添加以构造新对象的字母。例如,参见图11,利用老对象列表430中的第四对象“かか”并添加一个附加按键以指定<,来构造在新对象列表440中的第一对象“かか<”。在图7中的逻辑符号索引表中,“<”是在2か键上的第二逻辑字母,因此生成对象“かか<”的指令的LOGICAL-SYMBOL-INDEX字段,被置为2,以指示此表中的第二字母。以这种方式对对象进行编码,是利用了与每个节点相关联的已知按键序列及字母到键的已知关联,以便大大减少每个词汇模块所需的存储空间。

[0149] 所述词汇模块编码技术,也允许不需检索就访问词汇模块项目。一收到每个新的有效按键,系统就执行与当前节点上的键相关联的指令以便从老对象构造新对象,接着按照单个指针到达适当的子节点。而且,不需要把每个对象存储在词汇模块中,新对象是利用LOGICAL-SYMBOL-INDEX字段定义而添加到老的解释上的。因此,词汇模块中多个对象所共享的词干,只存储一次并能用来产生所有由它所派生出来的对象。所公开的存储方法,要求维持词汇模块树中的父节点对象列表以便构造子节点的对象列表。

[0150] 诸如图7中所描述的逻辑符号索引列表中的项目,不必是单个字符-任一序列都可占据一个单个项目。例如,假名序列“なかつた”可以被添加到老对象列表的第三对象“きか”而形成“きかなかつた”。以该种方式,所输入按键序列的长度,不必与相关联对象的长度直接对应。存储于符号索引列表内的项目中的字符序列,将允许词汇对象由任意键序列来识别,即存储于词汇模块树内的任意位置。

[0151] 图8B的每个指令558中还可以包括对象类型字段,以指定关于正在构造的对象的附加信息。所述对象类型字段可以包含一个指定所生成的对象是单词、词干、或任一其他对象的代码。因此,对象类型字段允许将不同类型的对象在给定的词汇模块中混合。而且,对象类型字段还可以包含关于单词语音部分的信息,或构造各种词形变化和词尾所需的信息。利用具有语音信息部分的词汇模块的压缩键盘明义系统,可以利用附加信息进行语法分析来改进多义性消除的过程。对象类型字段也可以包含独特的代码以容许传输压缩形式的文本。所述独特的代码,代替传输输入的按键序列或相关联的消除了多义性的字符,将被传输到远程终端。

[0152] 所述优选词汇模块数形数据结构的关键特点之一,是与每个节点相关联的对象被按照其使用频率存储在节点数据结构400中。即,由指令分组406中的第一个指令所构造的对象,具有比由406中的第二个指令(如有的话)所构造的对象高的使用频率,而后者又具有比第三个指令(如有的话)所构造的对象高的使用频率。用该种方式,自动把对象置于对象列表中以便按照使用频率的降序对其进行排序。为了说明的目的,单词对象的使用频率,指的是在具有代表性的应用语料库内使用给定单词的概率,其与每个单词在语料库中出现的

次数成正比。在词干对象的情况下,使用频率由共享该相同词干的所有单词的频率总和确定。

[0153] 把使用频率和其他等级信息存储在每个节点,这就避免了在使用系统时需要确定和按每个对象的等级排序。这一点在单词对象词汇中有着重要的含义,因为所存的对象可以包括大量更长的单词所共有的共享词干。动态地判断这些词干的相对等级,将需要遍历子节点的整个树并存储(accumulate)关于每个词干的信息,为便携式计算装置增加明显的处理开销。事先确定该信息并把其存储在词汇数据中,这样就降低了处理的开销。然而,当使用频率或节点所创建的对象等级,由创建他们的指令406的排序隐含地表示时,该信息就不需要另外的存储空间。

[0154] 虽然优选地,对象是被按照其使用频率而存于节点数据结构400内的,但是可以意识到使用频率字段还可以与每个指令相关联。使用频率字段将包含与该相关对象的使用频率相对应的代表号码。不同对象间的使用频率,将通过每个对象的使用频率字段进行比较而确定。使用将使用频率字段与每个对象分组相关联的这后一种构造方法的有点是,使用频率字段可由明义系统来改变。例如,该系统可以改变使用频率字段以反映用户在代表性文字输入期间使用词汇模块内的某些对象的频率。

[0155] 图12,是子例行程序600的流程图,其用于分析所收到的按键序列以便在具体词汇模块中识别出相应的对象。所示子例行程序600构造了特定按键序列的对象列表。方框602,清除了新的对象列表。方框604在其根节点1011开始遍历图10的树1010。方框606获得第一键的按下。方框608到612形成了处理所有可获得的键的按下的循环。方框608调用图13的子例行程序。判定方框610确定是否已经处理了所有可获得的键的按下。如果任一键的按下仍尚未处理,方框612就转到下一个键的按下。如果所有键的按下都已经被处理了,方框614就返回所述已完成的对象列表。将意识到,如果主例行程序在处理新按键序列时重复调用子例行程序600,每个新按键序列都比最后一个多个键并且除了最后一个以外的所有的键都与上一次启用(invocation)相同,则如果子例行程序620直接被调用来只处理最新一次键的按下,可以绕过初始化步骤602、604。

[0156] 图13是从子例行程序600调用的子例行程序620的流程图。如上所提到的,为构造一个新对象列表,所述明义系统从一份老的对象列表开始。因此,在方框626,存储由先前的节点所生成的对象列表,以便其用来构造所述新的对象列表。

[0157] 在图3中所示的主例行程序中,在方框150系统检测按键。如果存在通到与按键相对应的子节点的有效路径,则收到新的按键就会导致向下遍历词汇模块树。由此在图13的方框621,检查节点400的数据结构的有效键位字段,以确定有效指令和指针是否与所收到的按键相对应。在判定方框622,对有效键位字段进行检查,以便确定是否存在由指令406和诸如404a这样的指针字段所组成的有效分组408与输入的按键相对应。如果没有有效分组与按键相对应,在方框624所述老的对象列表将返回到主例行程序以生成选择列表。因为所收到的按键是不与词汇模块中任一对象相对应的无效按键序列的一部分,所以所述按键被忽略并且把当前对象列表返回到主例行程序作为由词汇模块所生成的对象列表。因此包括方框622和624的子例行程序620的分支,忽略所有无效按键序列并返回在父节点所生成的对象列表,以便在明义系统所生成的选择列表中进行可能的插入。

[0158] 如在判定方框622存在与所收到的按键序列相对应的有效分组,则子例行程序转

到方框626,在该方框新对象列表被复制到老对象列表。方框628取出与给定键相关联的第一有效指令。方框630把迭代器NEW-INDEX初始化为1,以便第一指令将在新对象列表中生成第一项。接着,所述子例行程序进入由方框632到642所组成的循环,以构造与有效指令相关联的对象列表。在方框632,检查所述OBJECT-LIST-INDEX(对象列表索引)字段556,并从老对象列表装入相应的对象。在方框634,检查所述LOGICAL-SYMBOL-INDEX字段555,并将合适的符号(通过象图7的550这样的逻辑符号索引表与所收到的按键相关联)添加到所识别出的对象的末端。将意识到,在方框634,如果在给定键551向符号表550和逻辑符号索引552的输入持有一个字符序列,则方框634多于一个的字符可以被添加到所识别出的对象上。在方框636,组合对象和符号作为新对象存储在新对象列表中。在方框638,进行检查以确定子例行程序是否已经处理了与在给定节点的给定键相关联的最后的的有效指令。如果最后的的有效指令尚未处理,则在方框640取出下一个指令。在方框642,NEW-INDEX递增(加1)。

[0159] 如在判定方框638中的检查表明已经为所述节点构造了所有的对象,则子例行程序转向方框644并按照相关联的指针到子节点。在方框646,把新对象列表返回给主例行程序,以生成选择列表。应该意识到,为每个从用户处收到的按键,都执行用于产生与每个节点相关联的对象列表的图12的子例行程序600。在用户输入新按键序列时,不进行词汇模块的“搜索(searching)”,因为每个按键只是使子例行程序又在词汇模块树内前进一级。因为不是为每个按键都执行搜索,所以词汇模块以最小的处理开销返回与每个节点相关联的对象列表。

[0160] 对图10所描述的树形数据结构1010起作用的图13所描述的子例行程序620的对象识别软件过程的按键组合中固有的几种新手段,可用来检索更多的词汇对象而同时又可利用较少的词汇模块存储器,并且不会增加子例行程序620的处理时间。

[0161] 根据输入词典中符号的使用频率对给定词汇模块的逻辑符号索引表550(图7)的每一行中的符号进行排序,可以使树形数据结构1010中的所有节点400的指令558(图8B)的大多数具有等于1的LOGICAL-SYMBOL-INDEX字段555。类似地,通过将所有节点400中的所有指令分组406的指令558进行排序,以便使词干和单词对象在对象列表中以其在语言中的使用频率的降序生成,可以使树形数据结构110中的所有节点400的指令558的大多数具有等于1的OBJECT-LIST-INDEX字段556。因此,在树1010中的很多数据是冗余的。系统地识别冗余度并通过将连接父

[0162] 节点与子节点的路径重新指向并删除不再被引用的子节点来消除冗余度,可导致比原始树包含少得多的节点、少得多的指令和少得多的连接仍然可从原始树检索到可检索的每个对象的折叠或卷绕数据结构。而且,经过其指令在对象列表440中生成类似对象的原始树的清楚的路径实例,被合并为折叠树的公共路径,此后它们将作为通用的(与专用的相对)对象构造规则,使简化结构可以比原来用于定义给定词汇模块树1010的结构多得多的对象。例如,在一个优选实施例中,由具有30,000英文单词的列表生成的非折叠词汇树可能包含多于78000条指令。通过如上所述的折叠过程的优选实施例的折叠之后,假设多义性按键序列和检索过程是以图12的流程图中的优选实施例来实现的话,则经过修改的树可能包含少于29000条的指令,数目比此结构可检索的单词对象的数据要少。此结果是显著并新颖的,因为对于一个键的按下,每个指令的反应是通过添加单个符号来只改变对象列表430中的一个对象。这是折叠树和重复使用公共指令序列作为通用对象构造规则的软件检索过程

所产生的结果。修改后的树结构的又一方面,是用于把对象与按键序列关联起来的自动识别通用规则。利用这种规则,词汇模块能够,高成功率地把按键序列与最初在创建它时没有被用到的单词和词干对象关联起来。例如,构建词汇模块的输入单词列表可以包括单词“sun”、“run”和“runing”而不是单词“sunning”,但是由所述算法创建的折叠树结构仍将常见单词“sunning”作为用于相应键序列的选择列表内的一个对象。

[0163] 图9所描述的节点是例子。节点560具有两个有效键,在其有效键字段562“0101000000”中以“1”表示。在优选实施例中“1”的位置表示第二个和第四个键是有效路径并且分别具有通到与其相关联的子节点的指令分组和指针566和568。分组566含有三个指令“(1,1,0)”、“(1,2,0)”和“(2,1,1)”,跟在其的指针“P”连接560到子节点。如果图12的子例行程序600,已经处理了通往节点560的按键列表,则图13的子程序620被调用以处理“2”键(该键在优选实施例中是2か键),之后将发生下述过程。指令561将把键2か(“か”)的第一逻辑符号添加到索引1中的老对象,以构建索引1中的新对象。561的第三字段,“0”是STOP-FLAG(停止标志)557(图8B)的伪值,表示这不是当前分组的指令,所以对下一个指令进行解释。指令563将把键2か的第一逻辑符号添加到索引2中的老对象,以构建索引2中的新对象。新对象的索引将是2,因为正在构建的新对象的索引是隐含地以指令本身的顺序建立的,例如第二指令永远建立第二对象。指令563的第三字段,“0”,是STOP-FLAG 557(图8B)的伪值,所以对下一个指令进行解释。指令567将把键2かの第二逻辑符号(图7中的“<”)添加到索引1中的老对象,以构建索引3中的新对象。指令567的第三字段,“1”,是STOP-FLAG 557的真值,表示这是当前分组的最后指令,所以子例行程序620的执行将从步骤638转到步骤644。

[0164] 有可能把两个或多个包含不同指令分组408的节点组合成为单个节点,该节点可用于与多个节点独立时一样的目的,这就意味着词汇树110中的某些节点在新的意义上是冗余的。对于本发明单词“冗余”用于两个节点的含义是通过示于图14-17中的优选实施例中的软件过程的操作可以将一个节点去掉。

[0165] 例如,比较图9中的节点560和574。键2かの指令分组566和571完全相同,但是节点560键4た上的指令570与节点574键4た上的指令572矛盾,因此,两者不能互相代替,两个节点也不能组合成为一个来完成两者的工作。对节点560和576进行比较。与每个节点上的键2か相关联的指令分组566和577完全相同。指令分组569和578的STOP-FLAG字段 557的设定不同,但该差异不会使它们发生矛盾。图13中的子例行程序620的对象检索过程的根本结果是通过给定键在一个节点执行一组指令生成的新对象列表。附加的对象可后接到对象列表的尾部而不会对正确处理该节点的任何子节点造成损害。这样,在578之后执行附加的指令不会在节点576的子节点的处理中造成错误。此过程的本质只有在执行错误的指令或执行的指令过少才会遭到破坏。与此类似,在节点576的键9ら上存在有效键不会与在节点560中缺少键9ら矛盾。因此,节点560和576是冗余的,并且可以合并成为一个新节点582,该节点可实现两者的实际结果,并且其作用与两者的子节点的父节点相同。应该意识到,在确定冗余度过程中指针也起了作用。在一个优选实施例中,在树中的与不继续形成更长单词的词干的单词相关联的按键序列的最后按键中,在有效键分组408中的指针具有特殊值“NULL(零)”,这表示再没有子节点了。此类节点称为“终止节点”。对于具有为两个节点所共有的有效键的子节点上的两个节点,相应的子节点必定是冗余的,因为其父节点是冗余的,并且

对从子节点下行的节点也同样,一直到到达终止节点或对正在比较的节点所共有的有效键序列上不再有后代为止。

[0166] 图14至17给出了用于类似于1010的词汇模块树的压缩和折叠的软件过程的优选实施例的流程图。图14为用于建立压缩词汇模块的软件过程的优选实施例的流程图。在方框652中,扫描词典以识别在如图6中针对日语词汇模块所示的实际键上出现的符号以外的任何必需的附加多义性符号。在方框654-656中,对各符号在其相应的各键上根据其在输入的词典中的使用频率的降序指定其逻辑索引,与图7示例中的相同。假定具有频率的对象词典,本领域技术人员很清楚方框658如何建立1010形式的词汇树。在方框660中,识别冗余节点并将其合并以减少数据的重复,并从而将与单个对象相关联的孤立的指令序列转变为用于检索多个对象的通用规则。此过程详细地在图15中示出。方框662从终止节点识别所有保持为NULL(零)的指针,并将其改变为指向具有最大父节点数的节点,从而增加模块中的规则数。将意识到,其他规则也可以适用于把子节点指定给NULL指针,并且此种规则可根据正在处理的按键的相关因子(factor)在对象检索时动态地应用。在方框664中,计算剩余的各独特指令558及指针404a,以便其可以编码为独特的位图而将较短的位图指定给较高频率的指令和地址以节省空间。一优选实施例是本领域内所公知的Huffman编码。另外,是多个父节点的子节点的节点,可以以特殊的顺序存储以便可快速检索和减少对其寻址所需的位(比特)数。

[0167] 在方框658中建立树的同时选择用来描述将要存储的对象的指令558,可以理解,当对象是单词或单词词干时,其字符序列包含可以被方便地用来增加树1010中的节点的冗余的附加数据。比如,日语中并非所有的假名对同样常见,比如“ま”通常后接“ず”。假名对(或双字母)的统计,可用于从前一个假名预测对象中的最可能的下一个假名。采用这种预测,在图7的逻辑符号索引表550中的多义性符号的逻辑顺序,可被动态地改变以进一步优化第一位置的使用。这种预测可扩展到假名的三个一组(三字母组),并且广泛地可扩展到n字母组。

[0168] 图15为折叠词汇模块的树1010的软件过程的优选实施例的流程图。从图14的流程图的方框660进入方框670。方框672从图10的树1010根节点1011下的第一个节点开始启动过程。方框674调用示

[0169] 于图16中优选实施例的流程图的子例行程序690来定位与当前节点冗余度最大的节点,如该节点存在的话。如目标节点找到,则判定方框676将过程指向方框678,在该方框中合并冗余节点,取消树中的重复数据,将多个单独的指令归类为共享序列,成为可用于将按键序列与对象关联起来的通用规则。如判定方框676失败,则判定方框680检查过程是否完结。如尚有更多的节点要处理,流程就转到方框682去识别另一节点。

[0170] 图16为用于在树1010中找寻与给定节点相比较具有最大冗余度的节点的软件过程的优选实施例的流程图。从图15的流程图的方框674进入方框690。方框692启动一个位置标志符MAX-SAVING用于测得的冗余度。方框694从图10的树1010根节点111开始启动过程。方框696调用示于图17中优选实施例的流程图的子例行程序710,来计算给定节点与当前节点的冗余度。判定方框698检查冗余度是否大于报告的MAX-SAVING。如是,方框700记录临时发现的节点的身份BEST-NODE为相对给定节点冗余度最大的节点并且报告的冗余度定为MAX-SAVINGS。判定方框702检查是否所有的节点都已经估算完毕。如否,则流程继续转到方

框704,在该方框中从当前节点转向下一个节点。从方框704流程返回到方框696。如在判定方框702中检查结果是最后一个节点已经评估,则方框706将节点的身份作为冗余度最大者,如没有,返回给定节点转到图15的子例行程序670的方框674。

[0171] 图17为用于计算两个特定节点之间的数字冗余度的软件过程的优选实施例的流程图。从图16的子例行程序690的流程图的方框696进入方框710。方框712初始化重复指令的计算。方框714将KEY-INDEX初始化为1。方框716,从指定作为参数的两个节点400之中的第一个,将与键KEY-INDEX相关联的指令分组406读出到子例行程序,并将其置于临时列表LIST-A中。如键KEY-INDEX不是有效键,不读指令。方框718,对指定作为参数的两个节点400之中的第二个,将与键KEY-INDEX相关联的指令分组406读出到子例行程序,并将其置于临时列表LIST-B中。如键KEY-INDEX不是有效键,不读指令。判定方框720判断LIST-A或LIST-B是否为空。如否,方框722从LIST-A和LIST-B中各取出一条指令并将各自的剩余指令数减1。判定方框724检查在其LOGICAL-SYMBOL-INDEX和OBJECT-LIST-INDEX字段中指令是否相同。如否,向图16的子例行程序690的方框696返回一个表示无冗余的失败代码。如判定方框724的判断为“是”,则方框728将计数SAVED INSTRUCTIONS增加1。控制又转到方框720。如判定方框720的检查是“真”,则控制转到判定方框730,该方框检查是否两个节点的所有可能的键都已经比较过。如否,方框732使KEY-INDEX加1,并且控制转到方框716。如方框730的判断是是,则控制转到方框734以重置KEY-INDEX为1。方框736检查与两个节点的KEY-INDEX相关联的指针。判定方框738检查是否两个指针有的为空(NULL),即终止节点或任何无效键的指针为空。如都不为空,控制转到方框740,该方框利用子例程710分别检查由两个不空指针指向的子节点是否为冗余的。在判定方框742中检查方框740的结果。如两个子节点不是冗余的,就返回一个失败代码。否则认为两个子节点为具有某一数字分数的冗余节点,该数字分数由方框746累计。判定方框748检查是否与最后键(在优选实施例中的键9)相关联的指针已经检查过。如否,方框752将KEY-INDEX增加1并将控制转到方框736。如在判定方框748中的检查判断所有的指针都已经检验过,则在子例程进入方框710时将原来所识别出的两个节点的累计冗余度值返回。

[0172] 将意识到,冗余度的数值计算可被加权以便考虑附加的因素,如在每个节点的分支数及作为子节点指向节点的父节点的数。将意识到,如两个节点因为与某一个键相关联的指令的排序而不是冗余的,则可以将与输入词典中的低频单词相关联的指令重新排序而不会影响与较高频率对象相关联的指令的优先级,从而可增加了树的冗余度。

[0173] 已经相对于由主指令组成的树,对通过消除冗余节点来减少树的大小的在上面所公开的方法进行了描述。把每个主指令与一系列0个或更多的副指令关联起来,提高了将要合并的对象的复杂性,但是所公开的过程仍能被用来识别和消除包括主和副两个指令的树中的冗余。而且,主和副两个指令不必存于存储器中的同一实际的树中。可以是主和副指令的两组指令,还可以存于存储器中的同一实际的树中。只要主指令树的节点中的主指令序列,与在副指令树的相应节点中的副指令组的序列一致,两组指令就最好存于单独的树中。类似地,当主和副指令存于存储器中的同一实际树中时,它们可以被相互混合,以便每个主指令后面都紧跟着(0或多个)相关联的副指令序列。作为选择,节点中的所有主指令都可以顺序地存于连续的块中,其后以与主指令相同的顺序紧跟着连续块的第二个序列,每个块都含有(0或多个)相关联的副指令。在该可选形式中,可以根据其主指令合并冗余的节点,和

合并相关联的副指令的两个或多个单独的组。接着,在每个转到被合并的节点的父节点,只需添加少量的附加信息以指定将要执行副指令的哪个候选组。

[0174] 此外,仍在本发明的又一方面,在数据库压缩方面的较高效率,是通过在数据库结构中为任一特定相关联的读法对每个汉字字符只存储一次。通常,数据库可以包括多个读法相同的相同汉字的不同例子,例如動物(读作どうぶつ)和植物(读作しよくぶつ)中的汉字物(读作ぶつ)。为了便于各种不常见汉字字符的组的输入,诸如没有明确地包括在数据库中的不常见的名字或单词,例如官物(かんぶつ),通过只输入想要的相关联的读法就能指定想要的汉字。因此,在一个优选实施例中,直接从树结构的根开始,给定汉字的每个读法,连同包括一个对相关联的汉字的代码的指定的副指令,都包括在数据库中。在所有这种情况中,相应的副指令把汉字添加到空文字对象上,因为构造相关读法的节点序列,起始于数据库树的根。因此,所有把汉字添加到空文字对象上的副指令,都能具有与众不同的格式,其中不需要索引来指定先前词条的索引,并且用来指定汉字字符代码的位数可以足够用来包含并区别数据库中的大量有效汉字字符。

[0175] 相对地,当汉字字符出现在数据库中的一个单词或短语中时,除了作为单词或短语的第一个字符之外的任何位置,相关联的副指令把汉字添加到已有词条文字对象上,其中该词条文字对象已经被添加到用于紧接在前节点的主对象列表中的相应的语音拼写的副对象列表中。从而,在该情况下,指令格式必须包括一个指定了词条文字对象的副对象索引的字段。同样地,当已有词条必须“保存”以便汉字能被添加在随后的节点中时,或当假名(在主指令中所指定的)必须被添加在、副对象索引字段必须被包括在指令中时。这样,为把数据库的大小减到最少,这两种类型副指令的格式,一个带有副对象索引而另一个带有这种将被区分的索引。

[0176] 在后面的情况中,即,当指令中的副对象索引是非零(non-zero)时,就把汉字添加到词条,与该将要添加的汉字相关联的读法可以由系统进行确定,因为系统记住哪个音节已经被添加到主对象列表中相关联的读法上,因为字符是最后被添加到由指令中的索引所指定的词条文字对象上的。如上所述,每个汉字字符的每个读法都从树的根部开始存于数据库中,其中用于汉字字符的代码完全由相关联的指令指定。因此,在汉字字符将要被添加到先前已有词条上的指令中,通过存储相应于被充分指定的汉字字符的位置的索引,来指定所述汉字,其中被充分指定的汉字字符在由树形结构的根开始所确定的读法的副对象列表中。该方法使在数据库中的汉字字符的大量出现,能够用比正常情况下所需要的少得多的位来进行指定,因为所存的索引将属于在数据库被创建时能够被确定的非常有限范围的值。例如,在编码了大约89,000个单词的数据库中,大约89%的汉字字符最好以该方式进行存储。在一个优选实施例中,如图8C所示,有四种类型副指令,每个都以被设置为四个不同值中的一个的两位格式字段开始例如:

[0177]

00	保存指令5600:保存由SECONDARY-OBJECT-LIST-LINDEX在当前节点所创建的列表中所指定的先前列表的词条(主指令所添加的假名,将是用来识别将要添加到该词条上的下一个汉字码元的语音拼写的一部分)
01	假名指令5601:把主指令所附加的假名(作为原文假名)添加到先前的由SECONDARY-OBJECT-LIST-LINDEX所指定的词条上,并把该修改后的词条添加到当前列表中。

10	间接汉字指令5602:主指令所附加的假名使用于将被附加到先前由SECONDARY-OBJECT-LIST-INDEX所指定的词条上的汉字的语音拼写变得完整;单独的N位字段KANJI-INDEX-FROM-ROOT含有列表索引,其中将要附加的汉字被认为是由根节点开始沿着所述完整的语音拼写所到达的在节点中完全指定的词条。应该注意:一个或多个这些N位代码可以被保存以表示其他将被附加到先前由SECONDARY-OBJECT-LIST-INDEX所指定的词条上特殊的专用对象(诸如在大量数据项目之前的一个转义序列或汉字重复字符)
11	直接汉字指令5603:下面的N位字段KANJI-CODE完全指定了汉字的编码(作为词条的第一个码元),以便与该汉字相关联的读法包括所有附加到所述语音拼写上的假名,从根节点开始直到包括相关联的主指令所附加的假名。

[0178] 因此,如图8C所示,指令5600-5602,格式字段代码5610置为00,01和10,全都包括字段5611 SECONDARY-OBJECT-LIST-INDEX,其指定了由紧接在前的节点所创建的在列表中先前已有的misashigo的索引。在PRESERVE指令5600中经索引的词条是保存在列表中的一个,以便其由一个或多个指令在子节点中进行修改。在KANA指令5601中,经索引的词条是由附加在由相关联的主指令所指定的假名进行修改。在一个INDIRECT KANJI指令5602中,所述用于将被附加的汉字的假名读法,根据指令的在前序列和指令5602所关联的主指令来进行确定。所述将被附加的汉字是由KANJI-INDEX-FROM-ROOT字段5613来指定的,其是与主指令相关联的第二指令列表中的指令的索引,所述主指令附加在起始于根节点1011的序列中所确定的读法的最后的假名。在DIRECT KANJI指令5603中,包括可一个更大的KANJI-CODE位字段5614,其含有足够的信息以恢复给定汉字的完全编码。所有指令格式5600-5603还包括STOP-FLAG字段5612,其起到了区分与给定主指令相关联的组的最后的指令的作用。

[0179] 在图18中给出了各种情况的例子,其示出了在数据库树形结构的三个节点中的一些可能的指令列表的代表。应该注意,是以说明为目的,图18给出了“解码”形式的主指令,其示出了要附加的实际假名而不是将实际存于数据库指令中的Logical Symbol Index(逻辑符号索引)(如图7)。也是以说明为目的,由每个指令所创建的所述列表对象,在指令右边的波形括号中示出。图18示出了用于“2办”键(例如,图1d的键122)的三个连续触发的节点的主和副指令列表的部分清单。左手边的列给出了用于第一级节点(起始于树形结构的根节点)的指令,表示由序列中键的第一次触发所创建的文字对象。接下来的两列给出了用于键的第二和第三次触发的指令。列表中的每个主指令都被标记为Y1,……,Yn(其中“Y”代表“语音拼写”),并且每个主指令都与其相关联的副指令列表一起示出,每个副指令被标记为M1,……,Mn(其中“M”代表“词条”)。在图18中,Y0和M0代表“null”文字对象。因此,如预期一样,在根级(级1)所有的主指令都指定了Y0并且所有的副指令都指定了M0。因此,在该级所表示的副指令是DIRECT KANJI指令(即,没有包括SECONDARY-OBJECT-LIST-INDEX字段),并且用于在图18中示为“M0+X”的每个汉字字符“X”的字符代码,在指令的KANJI-CODE字段中,被充分指定。由于DIRECT KANJI指令只被用来指定作为词条的第一个字符出现的汉字,即使当这种指令出现在树的第一级一下时,也不需要在前PRESERVE指令。这样,没有“M0+P”形式的指令出现在级1中,因为它们能被推出,并不需保存一个“null(空)”文字对象。例如,在中将的列(级2)中,主指令Y1把假名<附加到前一级(即办)的语音拼写Y1上,由此在第二级上创建了一个语音拼写对象办<。第五个副指令M5是DIRECT KANJI指令,并且由此相应于完

整的读法か<,其包括全部从根节点开始向下到当前级所附加的假名。该指令还包含足够的信息以为字符画确定正确的输出代码。该同一汉字字符还是词条的一部分,其中所述词条是由与主指令Y2相关联的副指令M1在级3上所创建的。所述主指令Y2把假名<附加到前一级(即きか)的语音拼写Y11 傻姑娘,由此在第三级创建了语音拼写对象きか<。与该主指令Y2相关联的副指令M1被示为“M5+[M5]”。该指令是一个INDIRECT KANJI指令,并且由此包括一个SECONDARY-OBJECT-LIST-INDEX字段,其指定了一个值5(图18中标为“M5+”)。这指的是在前一级2与Y11(由主指令的OBJECT-LIST-INDEX字段所引用的)相关联的第五词条对象。该在级2上的词条对象,是由主指令Y11“Y2+か”所创建的,并且其相关联的副指令M5“M7+P”(一个PRESERVE指令,表明所述词条对象企由与前一级(级1)的语音拼写Y2相关联的指令M7创建和充分指定)将被复制(保存),作为在当前级(级2)的列表中的第五词条对象。所述“preserve”指令起到了位置标志符的作用,并表明在该级(か)所添加的假名是将由随后的指令所附加的汉字的读法的开始。这在级3的INDIRECT KANJI指令Y2:M1:“M5+[M5]”中发生,其中与经索引的汉字[M5]相关联的读法,是作为保存的か加上由Y2在级3上所添加的<(か<)而被知道的。在从根开始直到级2的Y1来查找该读法か<,其中索引[M5]产生了充分指定的汉字画。把该汉字附加到由级1所保存的词条上,从而生成了完全指定的词条企画。

[0180] 在又一优选实施例中,一个备选方法被用来实施“preserve”指令(在上述实施例中的格式代码00)所提供的功能。在该备选方法中如图8D所示,带有格式代码00的备选指令格式5604,类似于指令格式5602,其中其指定了指定了先前已有词条的索引的SECONDARY-OBJECT-LIST-INDEX字段和5611KANJI-INDEX-FROM-ROOT字段5613。然而,在该备选实施例中,所述SECONDARY-OBJECT-LIST-INDEX不是指向紧跟在前的节点,而是指向在当前节点之前比由NUMBER-OF-KANA字段5615所指定的节点数多一个的节点。相应于所要附加的汉字的读法,通过向后通过主指令链从所指向的节点直到当前节点中的相关联的主指令,来进行确定。该方法的优点是,在插入的节点处不需要单独的PRESERVE指令,所以能够使数据库更紧凑。所述缺点是在处理按键序列的过程中必须保持先前节点的多个对象列表,而不是只有一个先前节点的对象列表。类似地,在备选形式的假名指令5605中,所述SECONDARY-OBJECT-LIST-INDEX指的是在当前节点之前由NUMBER-OF-KANA字段5615所指定的节点数的节点。一个或多个所要附加的假名序列,通过向后通过主指令链从所指向的节点直到当前节点中的相关联的主指令,来进行确定。其与备选的PRESERVE指令的优点和缺点相同。对于本领域普通技术人员来说很明显,可以用各种方法来合并所述FORMAT CODE字段(5610)、SECONDARY-OBJECT-LIST-INDEX字段(5611)、和NUMBER-OF-KANA字段(5615),例如使用Huffman编码,以实现在所存指令中的更大的压缩。

[0181] 在本发明的又一优选实施例中,通过下述机制,实现了数据库的进一步压缩,而没有基本附加的处理负担。在很多情况下,当与给定汉字相关联的读法出现在单词中的非首写位置时,所述读法就会改变。在绝大多数这样的情况下,该语音变化是在前汉字的发音的共发音效应的结果,并导致非首写汉字的第一个辅音由不发音变为发音,或由摩擦音变为爆破音。这相当于下过那非首写汉字读法的第一个假名添加了浊音或半浊音。在上面所公开的数据存储和检索方案中,这将需要为数据库安排一个附加项目以存储所述充分编码汉字,其中所述充分编码汉字与存于起始于数的根部的节点序列中的备选读法(带有浊音或半浊音)相关联。在该优选实施例中,当带有这种备选读法的汉字出现在非首写位置时,就

用INDIRECT KANJI指令来指定关于备选读法的想要汉字。为了减小数据库的大小,所引用的汉字不从与备选读法相关联的而只从与标准读法(没有浊音或半浊音)相关联的根进行存储。为了识别想要的汉字,存于INDIRECT KANJI指令的SECONDARY-OBJECT-LIST-INDEX字段中的值,按如下方式进行修改:设N代表从与备选读法相关联的根开始进行存储的汉字数,并且设X代表SECONDARY-OBJECT-LIST-INDEX值,其相应于从与标准读法(没有浊音或半浊音)相关联的根开始进行存储的想要汉字的位置。接着,把INDIRECT KANJI指令的SECONDARY-OBJECT-LIST-INDEX字段设为值(N+X)。在处理INDIRECT KANJI指令的过程中,系统首先搜索从与备选读法相关联的根开始所存的汉字,并确定其中只存有N个汉字,其小于字段值(N+X)。确定相关联的读法开始于带有浊音或半浊音的假名,生成相应的标准读法,从字段值(N+X)中减去所确定的汉字数N,并且发现想要汉字在所确定的标准读法之下的所确定的索引位置X。

[0182] V. 利用明义按键的代表性系统的操作

[0183] 图19示出了图1a所示系统的操作的三个代表性例子。这些例子示出了系统的操作,以及在实施例中所提供的反馈的特性,包括数据键21-30、Select键60、Convert键62、以及Diacritic键68的使用。所述例子给出了将在文字区域66中出现的文字,包括在插入点88处的对象的特殊格式(下划虚线或下划实线),作为每个按键的结果。每个按键都通过图1a中所示出的键的数字标记进行识别。

[0184] 图19中的例子1,给出了在图1a的系统的一个优选实施例中短语お願ひします是如何输入的。在键21、25和22上的每前三个按键之后,显示最常见的语音拼写,每个都分别相应于长为1、2和3个假名的单词。在Diacritic键68上的第四个按键,指定了在键22之前的按键的假名相应于带有浊音符号的假名,并且所显示的语音拼写变为最常见的、与键序列21、25、22相匹配的并且在第三位置处的假名带有浊音符号三假名的语音拼写(うなぎ)。在键21上的第5个按键之后,最常见的语音拼写はおねがい,其相应于想要输入的短语的第一个单词。在Convert键62上的随后按键把所显示的文字变为最常见的相应于所显示的语音拼写的词条,其在这种情况下也相应于想要输入短语的第一个单词。在数据键23上的下一个按键,启动了新按键序列的输入以消除歧义,因为它是在一个或多个Select键60或Convert键62的触发(在这种情况下,Convert键62的一个激活)之后发生的。在键23上的按键之后,接着的两个按键在键27和23上,象以前一样显示最常见的语音拼写。在这种情况下,在这些按键之后所显示的单词,相应于输入短语的想要单词します。在Select键60上的最后的按键,表明当前单词对象的按键序列已经完成,以便在数据键21-30上的下一个按键将启动一个新的输入序列。

[0185] 图19的例子2,示出了包括在数据库中的短语ありがとうございます的输入期间的显示。在7的按键(键22上)之后,与所述输入键序列相匹配的数据库中唯一的对象,就是想要的输入短语本身。因此,即使在该点的序列不相应于完整的单词或短语,也显示最常见的匹配词干(在该情况中为ありがとう)。而且,由于在数据库中只有一个潜在的匹配对象,并且该对象的当前位置中的假名包括浊音符号,所以该假名被显示成带有浊音符号,尽管并没有触发Diacritic键68。因此,在该情况下,触发所述Diacritic键68是可选的,并且例子2示出了在没有为ご或ざ假名触发Diacirtic键68的情况下所述短语的生成。为从所述序列的第7直到第10个按键显示了词干,直到通过第11个按键来完成该短语的输入。

[0186] 例子3给出了在实施例中的与Select键60和Convert键62相关联的各种功能。在键22和键25上的前两个按键之后,与语音拼写所匹配的最常见的单词被确定为この。在Select键60上的下一个按键,表示了当前单词对象的数据键序列的结束,并由下划虚线变为下划实线来表示。第二次按下Select键60选择第二最常见的语音拼写くに。在Convert键62上的随后的按键,选择相应于所选和所显示的语音拼写くに的最常见的词条。在Convert键62上的随后两个按键,循环通过较少常见的单个汉字字符的词条,以及以片假名形式示出了所选语音拼写的序列中的最后的词条。在Convert键62上的随后的按键,循环回到最初所显示的平假名,作为与该语音拼写相同的词条序列中的第一个原文解释。在Convert键62上的附加按键重复该循环,再次示出最常见的汉字解释。在那之后,按下Select键60还原当前所选语音拼写くに的显示。第2次按下Select键60,提升到下一个语音拼写かね。在Convert键62上的随后的两个按键,循环通过与该语音拼写相关联的前两个(最常见的)词条,其中每一个都是单个汉字字符。最后按住Convert键62,循环回到词条列表以重选先前所显示的汉字。

[0187] 在此所公开的压缩键盘日语文字输入系统,减小了结合了该系统的计算机或其他装置的尺寸。减少数量的键,允许所构造的装置被用户用一只手拿着,同时用另一只手进行操作。所公开的系统尤其是有利于蜂窝电话、PDA、双向寻呼机、或其他受益于准确高速文字输入的小型电子装置的使用。当在基于触摸屏的装置上或带有有限数目的、还可以有有限的显示屏幕区域的机械按键的装置上实施时,所述系统能够提供高效性和简便性。本发明的系统产生了由日语语音音节所组成的假名形式的输入,结合了把输入假名转换成想要的汉字或其他相应于假名读法的文字解释的处理。而且,本发明的系统提供了用于把系统操作所需的信息存储在非常紧凑的数据库中的方法,其在执行中只需要极少的处理花销。

[0188] 本领域中的技术人员,还将认识到,能够对键盘布局的设计和基础数据库的设计做出局部改变,而不显著地脱离本发民的基本规则。因此,在所附权利要求的范围内,应该意识到,本发明还可以以除了在此所描述的特定方式之外的方式实施。

[0189] 多义性文字输入的示例性明确字符过滤

[0190] 本发明涉及了一种用于在多义性文字输入中明确过滤的方法和装置。本发明提供了包括各种明确文字输入方法学的实施例,诸如,两键和长时间按下。本发明还提供了用于在数据库中对单词进行匹配的手段,其利用了环绕构造方法学、词干锁定方法学、单词完整化方法学、和n个字符列搜索。

[0191] 更特别地,本发明涉及了一种方法和装置,用来在多义性文字输入模式中进行明确过滤,用来扩展并解释对象、单词、单词词干、和类似的,以及用来锁定词干(stem-locking)。利用各种明确输入手段降低多义性的明确过滤的一个特定实施例,包括2键明确输入特性,其中利用结合了矩阵显示的键的按下。预定的字符组被“环绕构造”成锚(anchor)以结束或开始一个新单词或词干。通过在前明确字母的“n个字符列”分析提供合理的猜测,以扩展单词。尤其是,提供了三个利用词干锁定的特定实施例,其中第一实施例利用了想要对象的n个字母的组合,诸如单词、和下一个键。第二实施例锁定了对象的词干部分,并求出对象的非锁定部分。第三实施例利用了n个字符列搜索。

[0192] 压缩键盘明义系统硬件的优选实施例的方框图在20中示出。键盘54和显示器53经由合适接口电路连至处理器100。任选地,扬声器102也连至所述处理器。处理器100从键盘

接收输入,并管理所有的对显示器和扬声器的输出。处理器100连至存储器104。所述存储器包括诸如随机存储器(RAM)这样的暂时存储器介质、和诸如和诸如只读存储器(ROM)、软盘、硬盘、或CD-ROM这样的永久存储器介质的组合。存储器104含有所有软件例行程序以管理系统的操作。优选地,所述存储器包括一个操作系统106、明义软件108、和在多义性文字输入中相关的明确字符过滤软件和/或在下面的其他细节中所讨论的扩展和解释软件110。所述存储器还包括词汇数据库30。任选地,所述存储器可以包括一个或多个应用程序112、114。应用程序的例子包括单词处理程序、软件词典、和外语翻译程序。语音合成软件还可以作为一个应用程序来提供,以允许所述压缩键盘明义系统充当交流工具。因此,在这种情况下,输出可以是由扬声器中用声音输出。

[0193] 本领域中的普通技术人员应该意识到并明白,所述讨论也适用于当对对象或部分对象进行了组合时的符号和符号序列,符号的一个典型例子,是所有语言中的字符,诸如字母、数、标点符号、或任一其他语言的符号。对象或部分对象的一个典型例子,是单词或部分单词。然而,在此的讨论同样地适用于日语假名和韩语字母(jamos)。而且,应该注意,所述对象不必是语言,如在此所请求保护的明义系统可以被用来查找图标、电话号码、或存货记录,只要提供一种类型的符号串表示。因此,应该意识到,诸如字母、单词、单词词干等项的使用并步局限于这些应用中,还被用来使在此所讨论的阅读和理解更容易。

[0194] 明确输入

[0195] 根据本发明以过滤单词选择为目的的生成明确字符的一个优选实施例,提供了2键明确输入特性。在很多语言中,诸如英语,所有与该键相关联的字母都被印在所述键上,如在图21中所描述的,其中图21是无线手持式装置的例子图片,其根据本发明给示出了与键相关联的英语字母表中的每个字母。

[0196] 然而,其他语言,诸如法语,包括比能够可见地置于键上的字母多得多的字母,诸如蜂窝电话键。图22是示于0-9键和其各自的Shift键上的欧洲字母和一些特殊符号的示例性图表。例如,如果用户按下5键,就出现带有字母数字式字符j、k、l和5的显示矩阵。图23是根据本发明的一个2键矩阵输入方法的示例性屏幕快照。在第一键按下之后,选择矩阵就显示在装置的面板上。应该意识到,印在键上的其他标记也是可能的,诸如每个键都包括相关联字母的预定组的子组,诸如显示相关联字母的预定组的第一和最后字符。

[0197] 图24是用来在图20的多义性文字输入系统中的明确字符过滤的2键明确输入的流程图。根据本发明的优选实施例,就用户明确地选择指定给键的很多字母中的一个来说,用户通过诸如按下的方式选择相关联的键(5000)。所述对想要的键的选择,例如按下想要的键,使系统提供第二组字母,诸如以矩阵形式,其相应于所选的键(5110)。在矩阵中的每个方框都包括与刚选择/按下的键相关联的字母组中的一个字母。图23示出了由于按下了1键而显示的例子。从该点来说,用户所全部需要的是,对包括想要字符的矩阵方框的键数进行选择(5200)。该明确选择过程,为字母的选择最多只需要2键的选择/按下。

[0198] 对于本领域普通技术人员来说很显然地,以过滤为目的的2键明确输入法,能够被扩展到任一其他语言。对于本领域普通技术人员来说很显然地,可能有诸如滚动列表这样的备选配置。

[0199] 根据一种备选配置,所述用户通过长时间按下来对键进行选择,以提供并循环到所述键上的字符序列。如图27所示,只要正在按下所述键,一些或全部字符就可以在预选窗

口内滚动过去。当放开所述键时,就接受了最后被高亮度显示的字符。

[0200] 仍是在又一配置中,在通过在所述键上长时间按下来提供字符序列之后,每按下所述键一次所述字符就出现或被高亮度显示。当在一个键的按下和下一个键的按下之间经过了足够的时间时或当按下了不同的键时,每个明确字符的提供可以结束。

[0201] 仍是在又一配置中,在通过在所述键上长时间按下来提供字符序列之后,每按下定位键一次所述字符就出现或被高亮度显示,其中所述定位键是,例如,方向键或滚动轮。

[0202] 下面是其他用于过滤的明义输入手段(means)的列表。应该意识到,所述列表只是用来列举的,而不是无遗漏的:

[0203] • 长时间按下一个键来明确地输入数字/数;

[0204] • 改为数字模式并按下一个键来明确地输入数字/数(紧接着返回到用于过滤的多义性输入模式);

[0205] • 改为多次轻敲(multi-tap)模式并重复地按下一个键,以明确地输入一个字符(紧接着返回到用于过滤的多义性输入模式);

[0206] • 通过把每对键的按下分组为2键明确输入,或通过把同一键的重复按下分组为多次轻敲输入,把多义性模式键的按下解释成明确字符;

[0207] • 利用多次转换键,诸如5向摇杆,其允许关于简单按下的多义性输入,和关于不同种类按下的明确字符输入;和

[0208] • 通过“联合”,其指的是通过同时按下多于一个的键,用主键表明一个多义性的字符组和一个副键表明要选择该组中的哪个字符(例如,在视频游戏控制器上)。

[0209] 用于过滤的明确输入的又一手段是,当指定给一个键的任一字符通过副手段被提供给用户时,诸如显示在如果被选择了就会被输入的程控键或“软键”标记上。所述软件机构允许对所输入的最后按键进行更正,例如,如果用户正在用额外的字符扩充单词或希望在输入整个单词之前更正所显示的解释。所提供的额外的字符,将基于与在前按键相关联的最有可能的字符的分析,或基于在当前单词候选列表内的单词。假定最有可能的字符已经显示在多义性单词选择中,那么第二最有可能的字符将被提供。例如,如果用户想要键入“ram”并输入了键序列726,则出现单词“ran”。在与726相关联的单词候选列表中的该位置处最有可能的字母是“m”,因而“m”能够被提供在软键上。当用户按下软键时,“m”就取代了单词“ran”中的“n”。

[0210] 在多义性模式中的明确字符过滤

[0211] 在多义性模式中明确字符过滤,是当在输入一序列多义性项目期间明确地输入一个字符时。已经讨论的明确输入的一个实施例,是利用如上所述的矩阵的2键明确输入方法。根据本发明的又一优选实施例,字符的明确输入,是通过长时间相关联的键来完成的。当字符序列中的一个或多个字符被明确地输入了时,只检索在相应的位置(s)带有该明确所选字符(s)的所存的单词。

[0212] 根据本发明的一个优选实施例一种检索想要的单词的方法,是在多义性模式中输入了明确字符时,然后只要可能明义过滤系统就相对于多义性和明确字符继续对数据库单词进行匹配,而不是根据第一个明确字母接受默认单词。如果没有匹配,所选的最后的的多义性单词 就被接受并且把新字母附加在其上。

[0213] 作为例子,假设用户想要用英语T9输入法输入单词“home”。用户长时间在第3位置

处按下6键以明确地选择“m”，从而产生了唯一的“home”，并且，此外，单词词干“imme”和“gome”在4个按键之后出现。

[0214] 又一例子是利用象“cu2night”这样的混合字母数字单词，其也可存于数据库中。如果所述用户长时间按在2键上，那么明确地输入一个“2”，象“cub”这样的单词就这样被从单词候选列表中过滤掉。

[0215] 明确过滤，是迅速提供单词完整形式的又一方法。如果“cu2night”在数据库中，并且“cu”明确地跟在“2”之后被多义性地输入，按键序列“282”的所有多义性解释将被过滤掉，剩下“cu2night”作为更可能的选择以用单词的完整形式而被提供。

[0216] 图25是根据本发明优选实施例，在多义性模式中用于明确过滤的流程图。一个字符是想要的，并多义性地或明确地输入其相关联键(6000)。接着，根据多义地选择的键(即，其相关联的字符)和明确地选择的字符解释想要的单词。在数据库中搜索带有下述这样的字符的单词，即该字符在所述单词的相应的位置处与多义地选择的项目(例如，键)和明确地输入的字符相关联(6100)。接着以预定优先顺序提供并显示所搜到的单词(6200)，诸如以用户最常用的顺序。如果想要的单词在所提供的搜到的单词组之中(6300)，则选择想要的单词避过那结束该过程。否则，所述处理不会结束并从选择想要的键(600)开始重复所述步骤。

[0217] 环绕构造(Build Around)

[0218] 明义过滤系统的又一方面中，数据库中的单词能够被环绕构造明确或多义性地输入的字符，或预定字符组中的附加字符。根据一个解释，所述匹配单词，如有的话，包括用于特定序列的明确字符。然而，该序列的附加解释可以包括：与多义性键相匹配的以明确字符开始和/或结束的一个或多个单词；一个或多个以明确字符(如有的话)开始的匹配单词；和一个或多个与跟在明确字符后面的多义性键相匹配的单词。例如，如果输入了“gifts4less”键序列，由于明确输入的数“4”，单词“gifts”与明确字符之前的多义性键序列相匹配以及“less”与其后的键序列相匹配，即使“gifts4less”、“gifts4”、和“4less”没有在数据库中发现。类似地，“mobile.com”可以作为一个由数据库项目“mobile”和“.com”或如果有多义性标点键的话由项目“mobile”和“.”和“com”自动构建的单词而被键入；在这两种情况下所述句点可以不必明确输入。

[0219] 环绕构造概念的一个实施例，可以参照图28来进行描述，图28是描述了环绕构造例子的流程图。所述环绕构造方法启始于(9000)正输入的键序列，例如[k1 k2 k3 k4 k5 k6k7 k8]。所述k#’s指的是多义性输入的键，kd指的是可以是明确输入的字符或多义性地输入标点字符的限定键。接着该方法确定数据库是否发现了一个匹配(9100)。如果是，就显示一个单词(9200)。如果用户想要转到续数据库所发现的单词列表中的下一个选择，用户就按下next(下一个)键以请求数据库进一步进行查找(9300)。尤其是，在所述数据库进行复杂搜索与键序列相匹配的项目。键序列在定界符两端被分成两组。两组序列的各种分组的例子被定界符如下划分：

[0220] • [k1 k2 k3 kd][k5 k6 k7 k8]；

[0221] • [k1 k2 k3][kd k5 k6 k7 k8]；和

[0222] • [k1 k2 k3]kd[k5 k6 k7 k8]。

[0223] 如果在任一组中发现了匹配(9400)，那么就选择并显示想要的单词(9500)。如果

用户想要环绕构造该序列,所述用户能够按下下一个键以获得下一组由对数据库的复杂搜索所得到的结果(9600)。

[0224] 如果数据库没有发现首写字母的匹配(9100),那么就在数据库中进行复杂搜索键序列的匹配(9700),其中键序列在定界符两端被分成两组,在上面给出了对其的描述和例子。

[0225] 应该意识到,仍在本发明的范围和精神内可以改变搜索和显示/提供可能想要的单词组的手段。

[0226] 词干锁定

[0227] 在本发明的优选实施例中,词干锁定是在序列中锁定一个或多个字符的子序列或子串。例如,可以锁定在单词字符的前n个序列。所述词干锁定方法,是只搜索那些带有锁定字符的单词。例如,假设用户利用T9技术选择了无线蜂窝电话的4并接着选择了6键。在显示器上提供单词“go”。如果“go”被锁定,那么根据随后的键选择,只有在前两个位置中带有“go”的单词被选择。

[0228] 可以用多种方法来执行字母的锁定。例如,两个这样的方法是利用“完全下一个锁定(full next locking)”机制和“无限制的(unbounded)”方法,通过在字符上移动光标来进行选择。作为实施“完全下一个锁定”的例子,使用了“下一个(next)”键。即,根据本发明的一个实施例,选择“next”键锁定一个先前所提供的字符序列,由此通过重新解释锁定的字符序列来消除系统重新解释对象的可能性。现在所选字符序列被标记为明确输入。根据本发明一个实施例的在字符(无限制的)上移动光标的例子,通过用户选择右方向键(或左方向键,例如,希伯来语)来移动所述光标以开始一个序列,光标在其上所移动过的所有字符都被锁定。

[0229] 完全下一个锁定的又一方面,是当可能的单词边界被识别了时锁定一个先前所提供的字符序列。可以根据某交互序列来识别这种边界,诸如,当用户“下一个(nexts)”并接着选择符号或标点符号时;或当用户“下一个”并接着输入一个明确数字时。

[0230] 应该意识到,其他“锁定”方法包括,但不限于下述手段:

[0231] • 按下指定了“锁定”功能的键,诸如一个OK或Select键;和

[0232] • 按下合适的方向键以“锁定”所提供单词的完整形式。

[0233] 词干锁定方法,诸如上面所讨论的那些,可以与上面所描述的“环绕构造”相关。即,一旦锁定了一个序列,其就可以被“环绕构造”。

[0234] 日语假名的例子

[0235] 在日语中明义过滤方法的又一例子。在假名(语言)字符形式中的子短语,能够被转换成汉字(语音拼写),并且那些转换的子短语可反过来过滤所转换子短语之前或之后的剩余多义性假名的解释,除了不能被用来构造有效短语的解释之外。

[0236] 在本发明的又一实施例中,采用了锁定的分级机制。例如,右方向键可以锁定光标左边的所有字符。因此,在该实施例中,右方向键锁定了第一选择词干,而“下一个”键锁定了不是第一选择的词干。

[0237] 又一种实施是,代替结合了“next”键的右方向键利用“select”键。例如,所述“select”键可以被用在每个多义性键序列的结尾,以用来选择和接受默认选择(select键被按了一次)或用来选择一个备选选择(select键不止被按下一次,直到想要的选择出现了

或被高亮度显示)。在该情况下，“select”键被用来词干锁定第一或备选选择，而不是利用右方向键用于第一选择和“next”键用于其他选择(或反过来也一样，作为键的选择是任一的，但是预定的)。

[0238] 单词的扩充和完整

[0239] 本发明的其他方面，是系统在锁定位置自动开始一个新单词。例如，所述用户输入“car”并将其锁定，并接着为“pet”输入所述键。示出“Carpet”，因为它是数据库中的完整单词。还示出了所述单词“carset”，因为它是“car”+“set”，其是“环绕构造”概念的又一例子，其中多义性序列可以跟在一明确字符的输入之后，这就使一个解释成为新多义性单词的开始。

[0240] 作为选择，当一个或多个先前所接受的字符在(没有间隔地挨着)当前现用多义性键序列之前时。所述系统利用在前的字符来对单词和短语(例如，在用户数据库中)进行匹配，其中所述单词和短语以那些字符开始并与跟在那些字符后面的多义性键序列相匹配。任选地，提供了一个或多个单词的完整形式。下面给出了一个例子。用户输入并锁定(接受)所述字符，“con”。接着用户为“stan”键入多义性键，并被提供“consantinople”作为单词的完整形式。在该例子中，所述用户可以利用一个左/右方向键来接受将被用来过滤的“con”并提出以所接受的字符开始并紧挨着一个多义性键序列的单词的完整形式。希伯来和其他语言走另一个方向，从此只右边的不是假定的。

[0241] 图26是根据本发明用于生成没有存在数据库中的单词的一种方法的流程图。即，在同等优选的备选实施例中，如果输入序列被锁定(7000)并且随后的多义性字符被输入(7100)以及相应的所存单词没有在数据库中发现(7200)，那么根据先前的n个锁定字符来搜索可能的组合(7300)。在本发明的优选实施例中，一个n个字符列被用来发现所存储的单词。这与利用在字母中锁定的整个组相反。例如，假设用户已经输入并明确选择了单词“flight”，但所述用户，无论如何都想输入不在数据库中的单词“flightplan”。在本发明的该实施例中，所述用户接着明确地锁定“flight”之后的字母“p”和“l”。接着，该系统利用最后明确输入的两个字母(p和l)来搜索单词。这样，就生成了没有存在数据库中的单词。

[0242] 虽然在此参照优选实施例对本发明进行了描述，但是本领域中的技术人员将很容易地想到可以取代在此所列出的其他应用，而不脱离本发明的精神和范围。

[0243] 因此，本发明将只由包括在下面的权利要求进行限定。

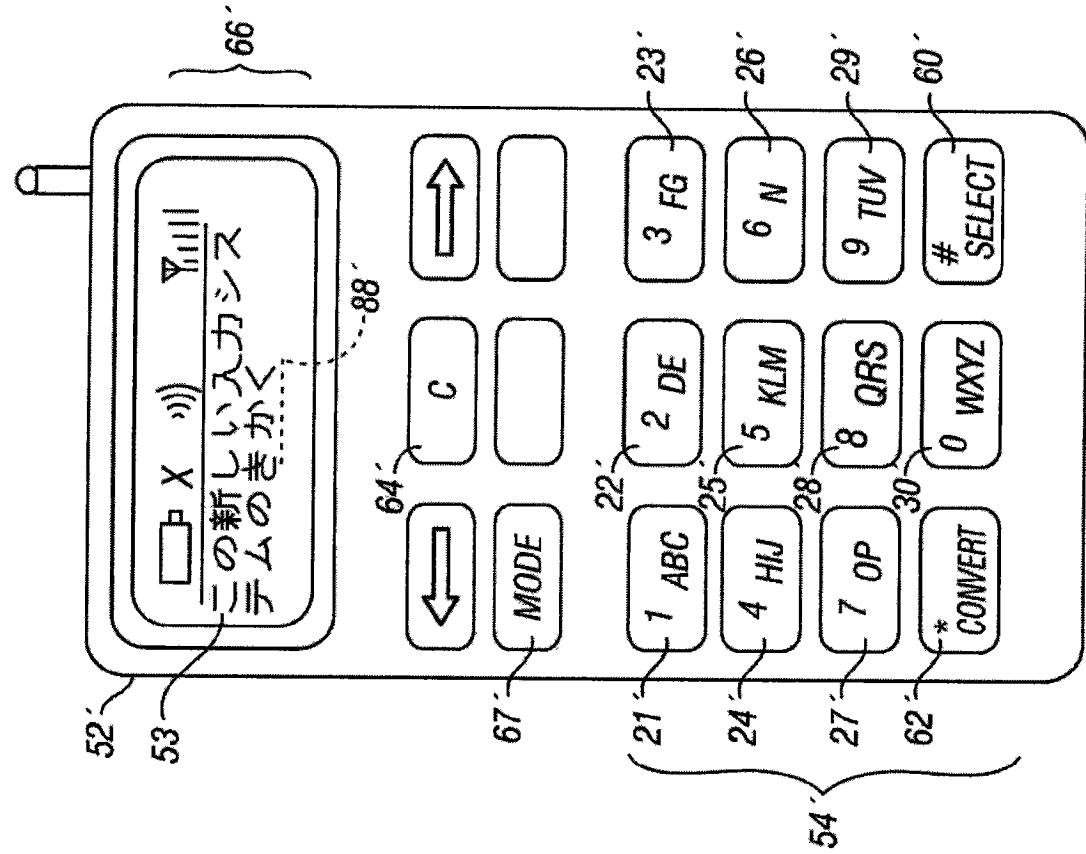


图 1B

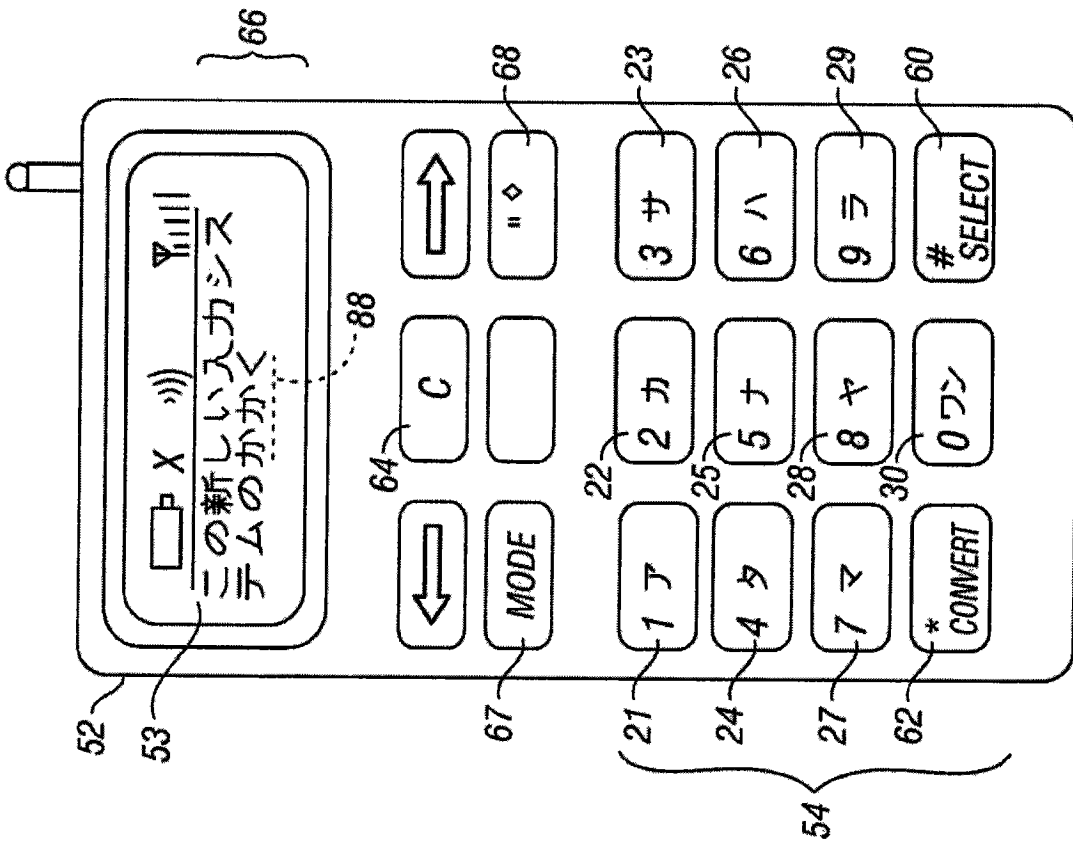


图 1A

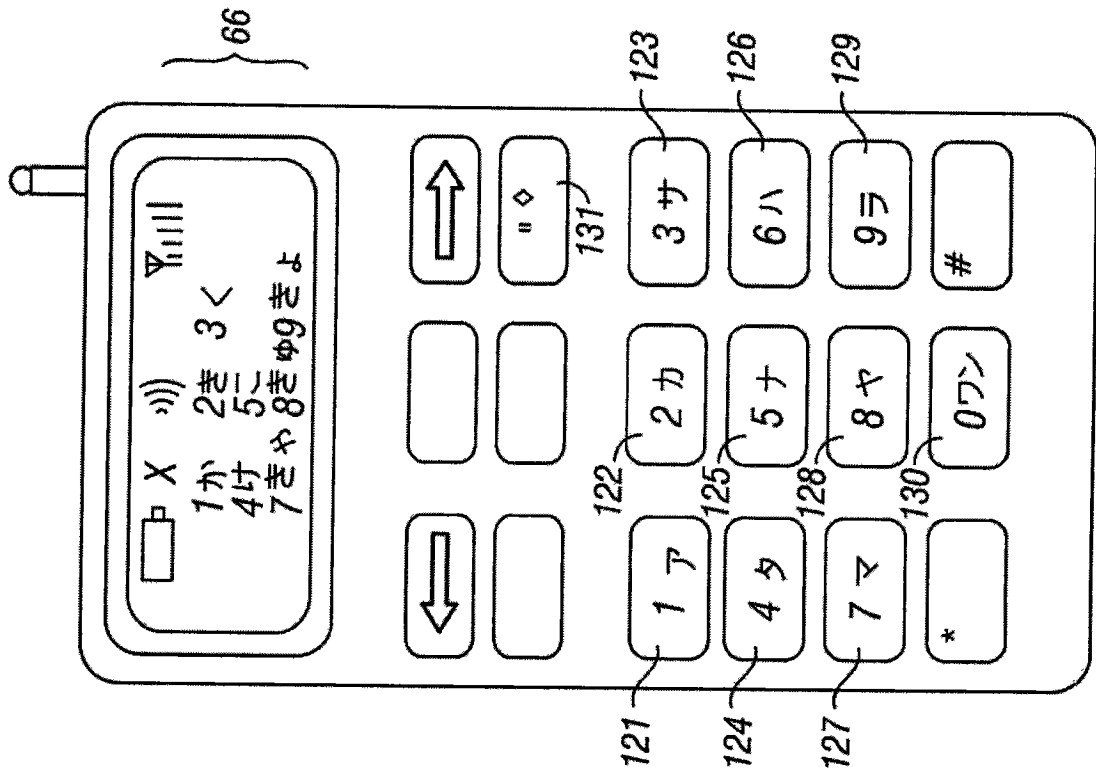


图 1D

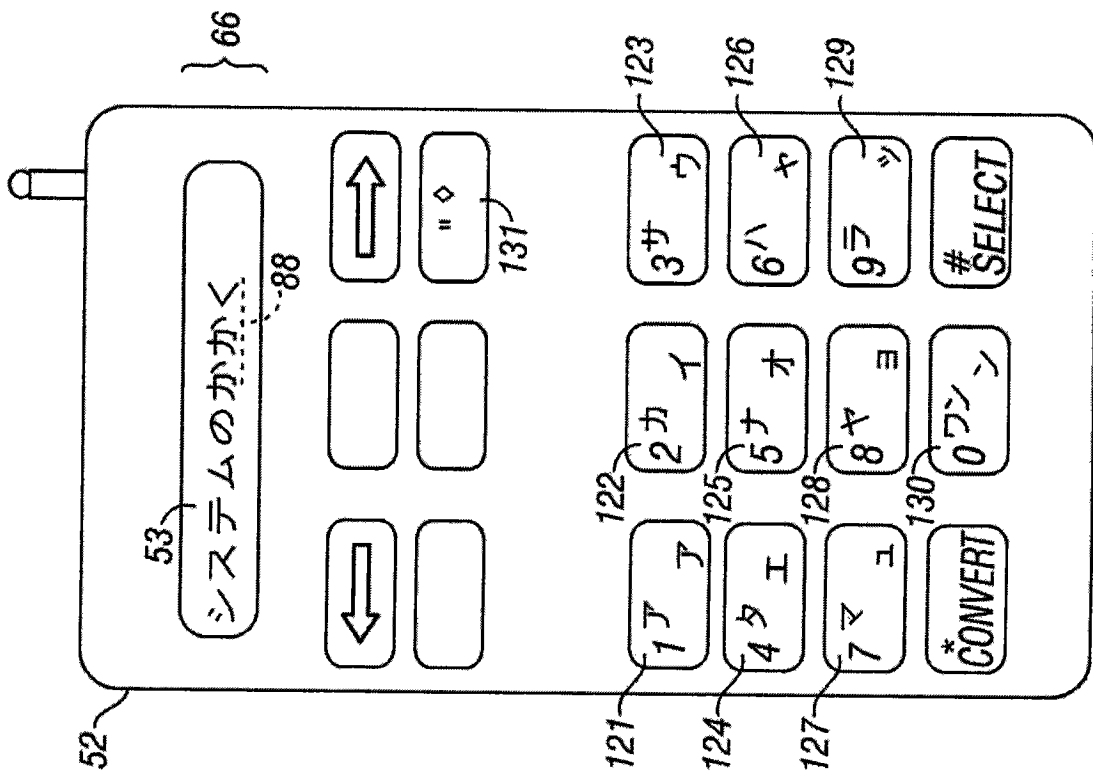


图 1C

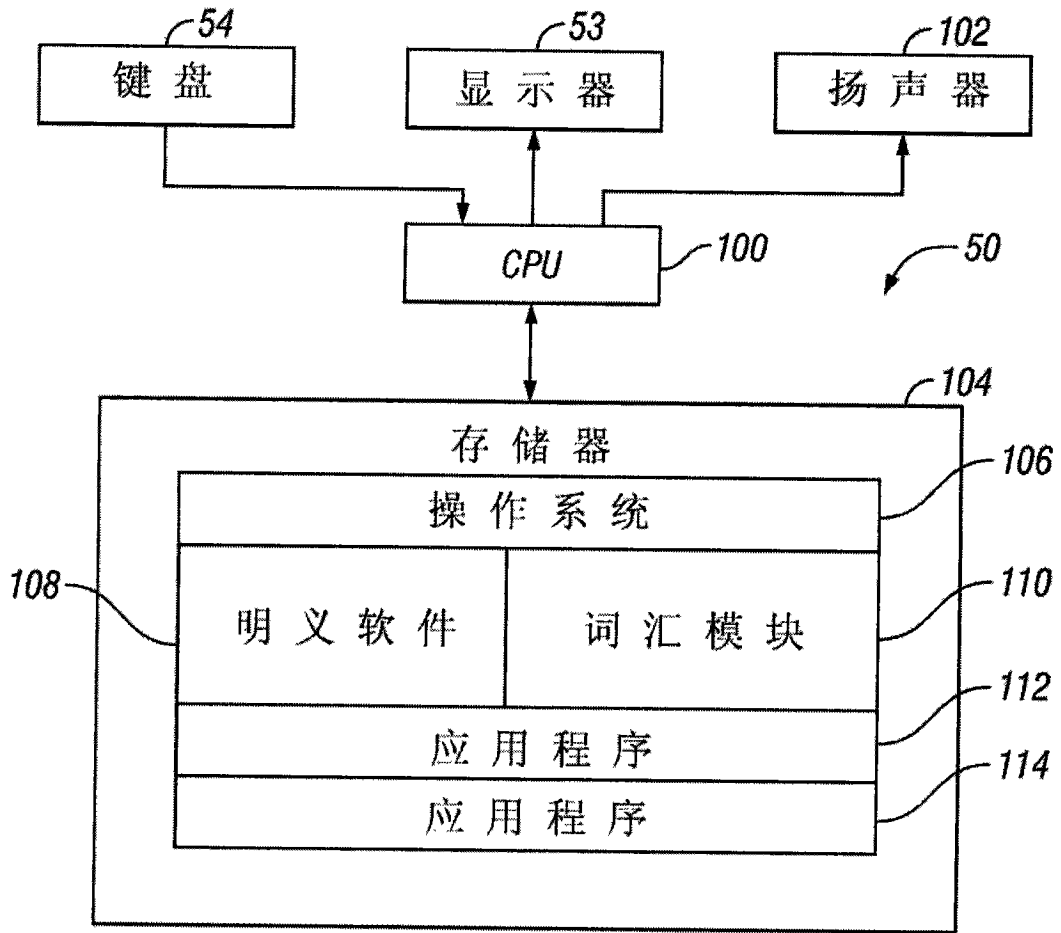


图2A

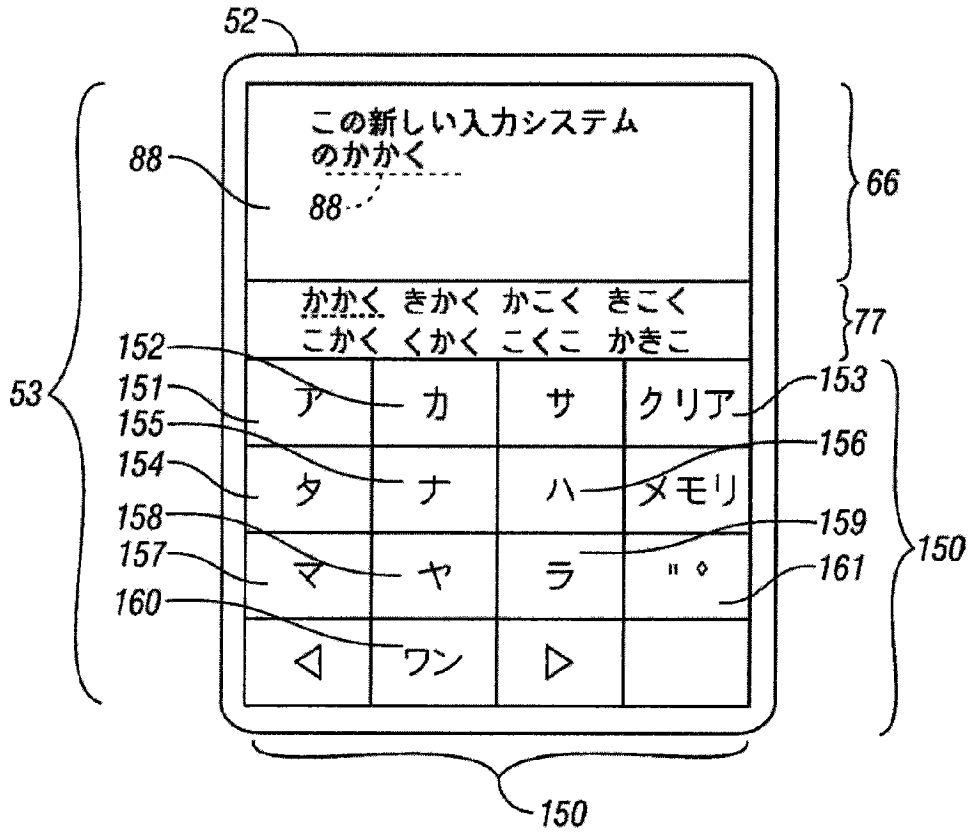


图 2B

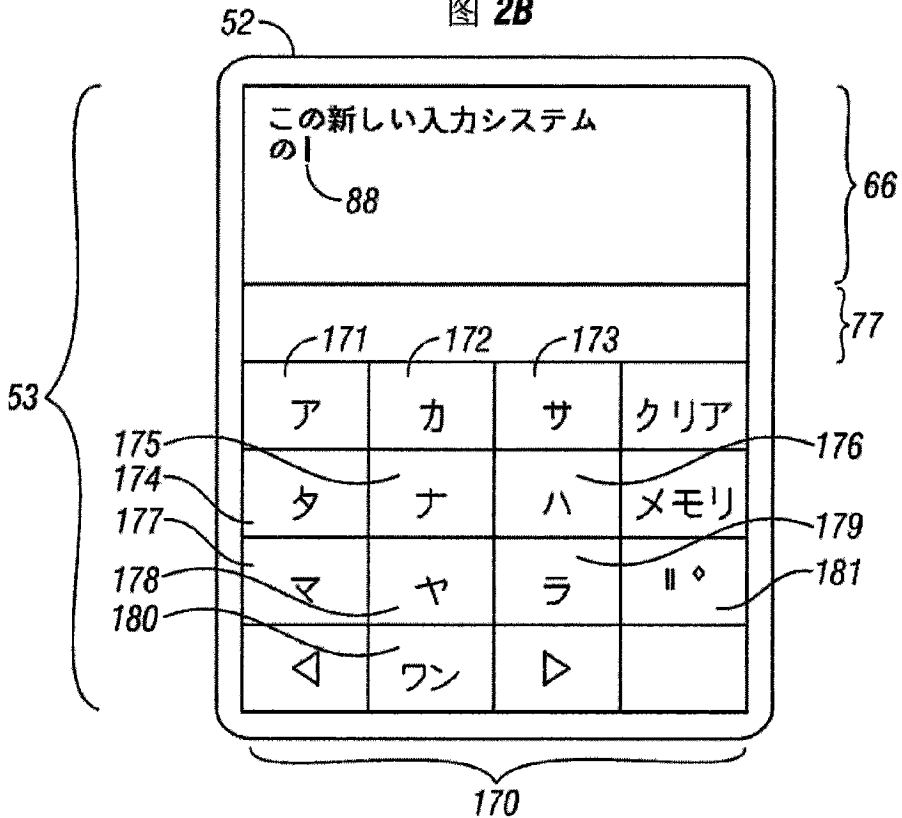


图 2C

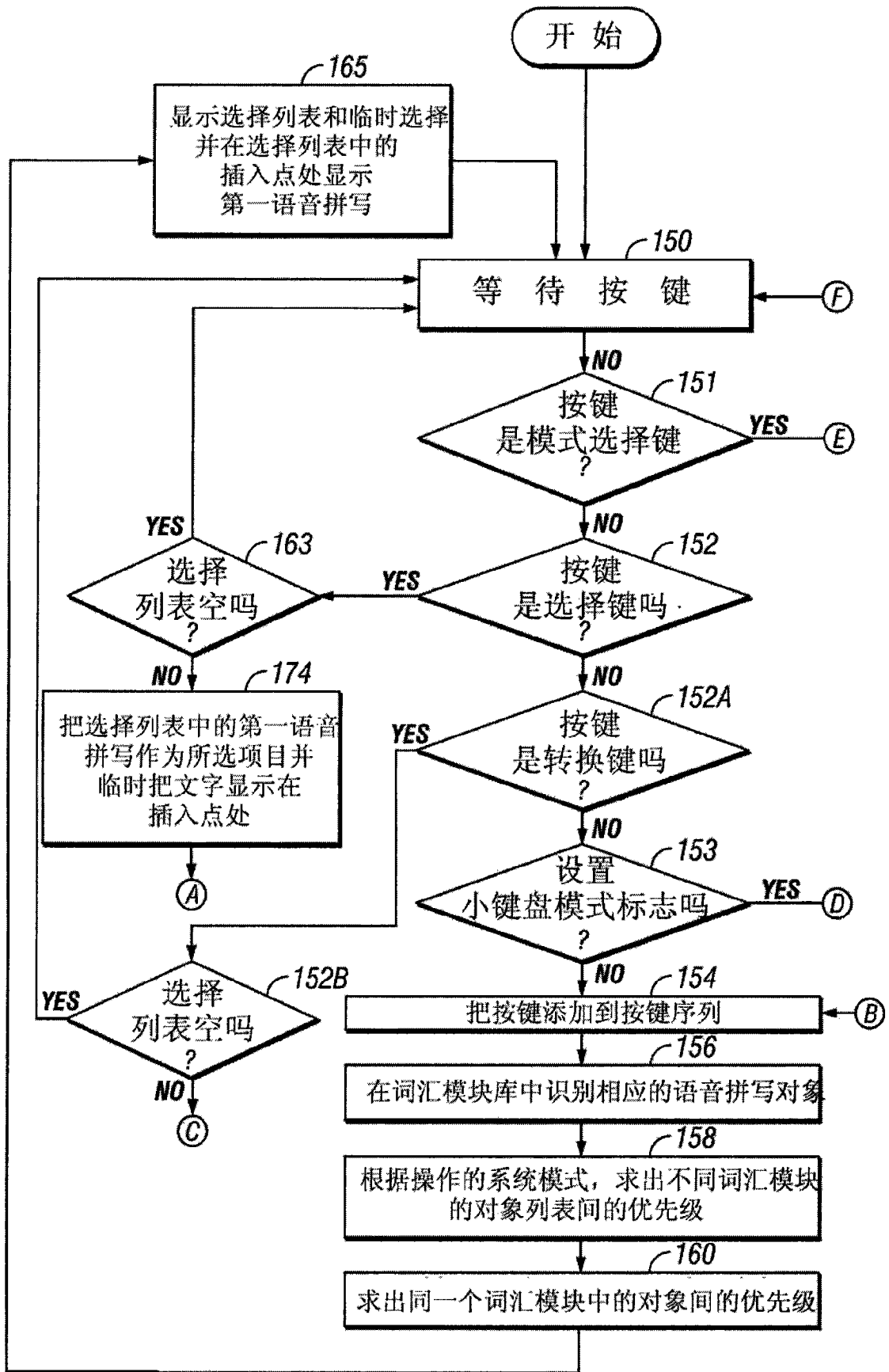


图3A

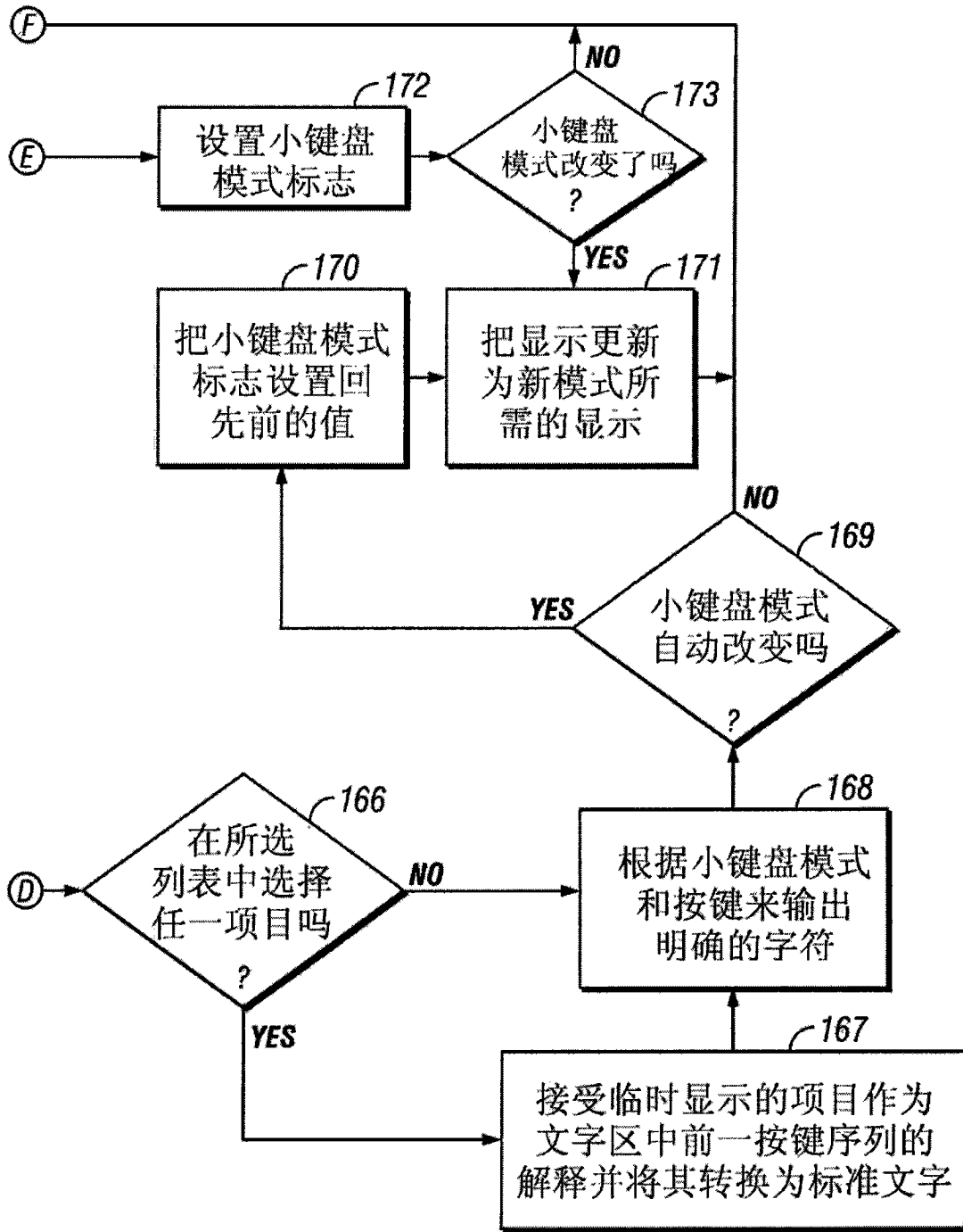


图3B

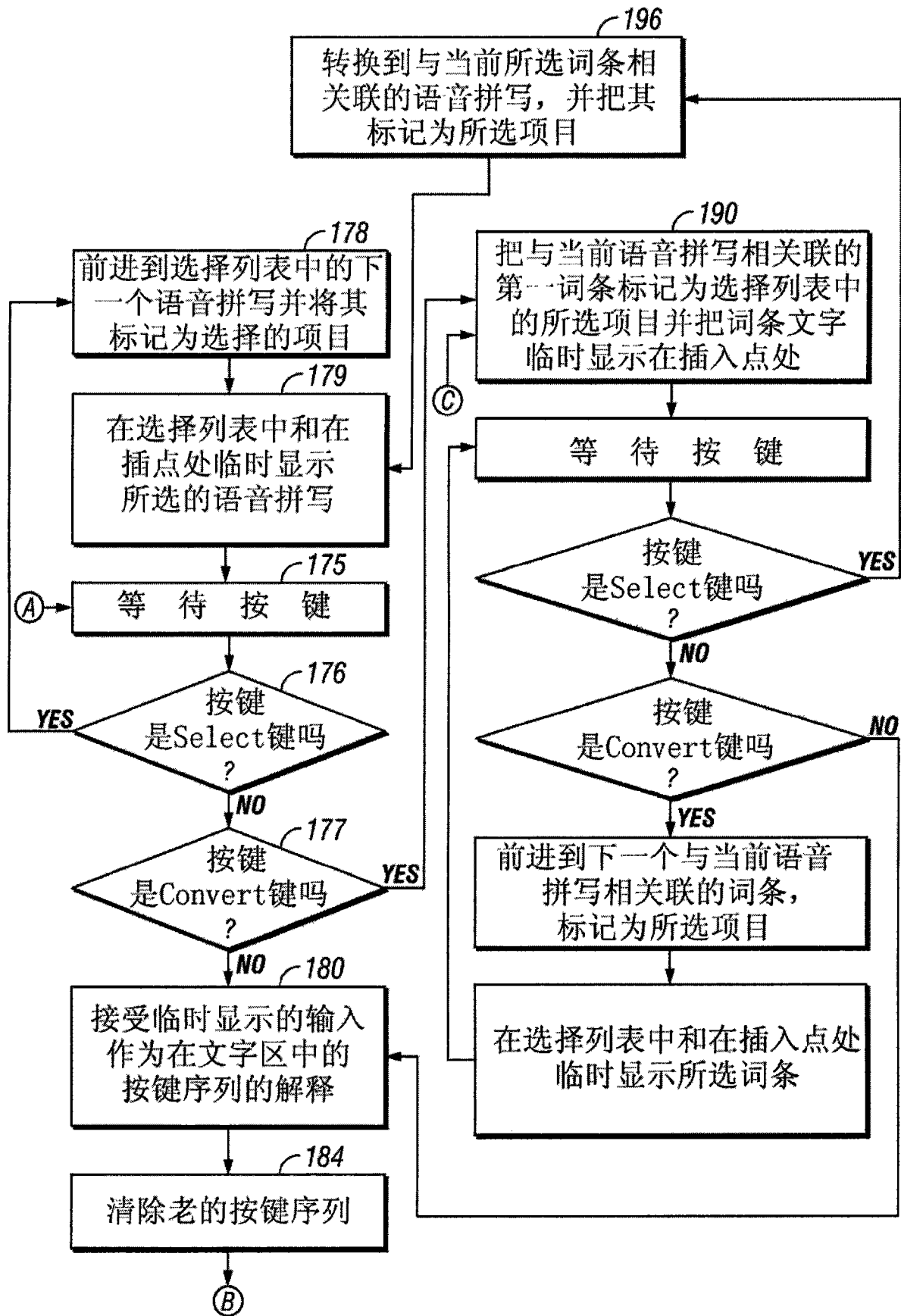


图3C

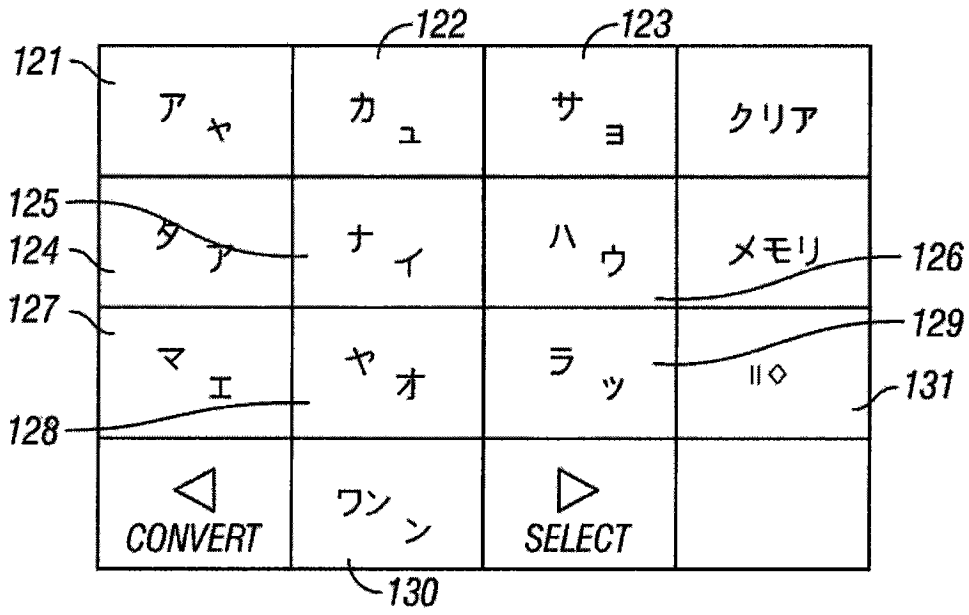


图4

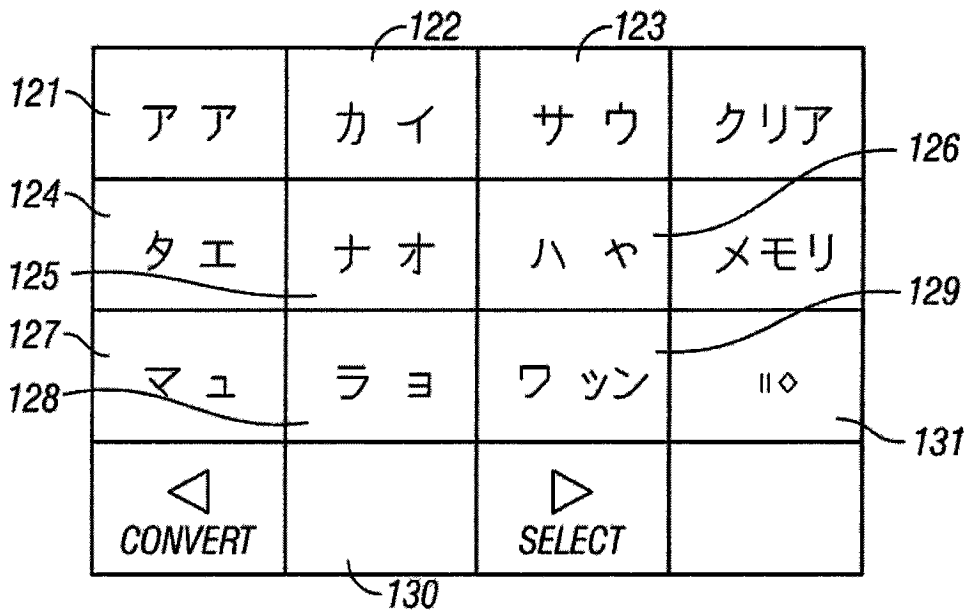


图5

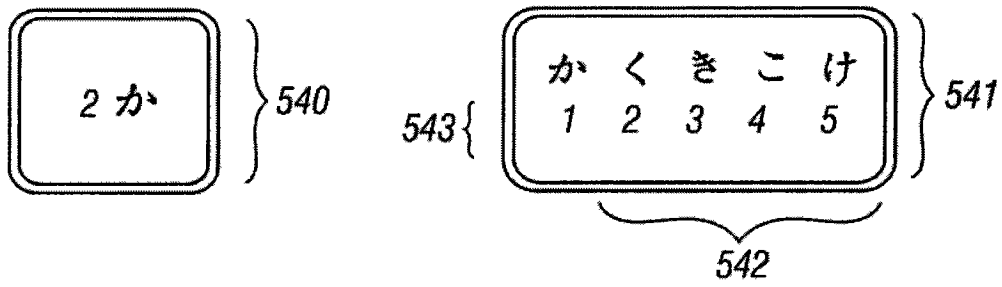


图6

552 {		1	2	3	4	5	6	7	8	9	10
550 {	1	い	う	お	あ	え	い	え	あ	お	う
	2	か	く	き	こ	け					
	3	し	さ	す	せ	そ					
	4	て	つ	と	つ	ち	た				
	5	な	に	の	ね	ぬ					
	6	は	ひ	ふ	ほ	へ					
	7	ま	み	も	め	む					
	8	よ	ゆ	よ	や	や	ゆ				
	9	り	る	ら	れ	ろ					
	10	ん	一	わ	を						
	11	〃	〇								
	551										

图7

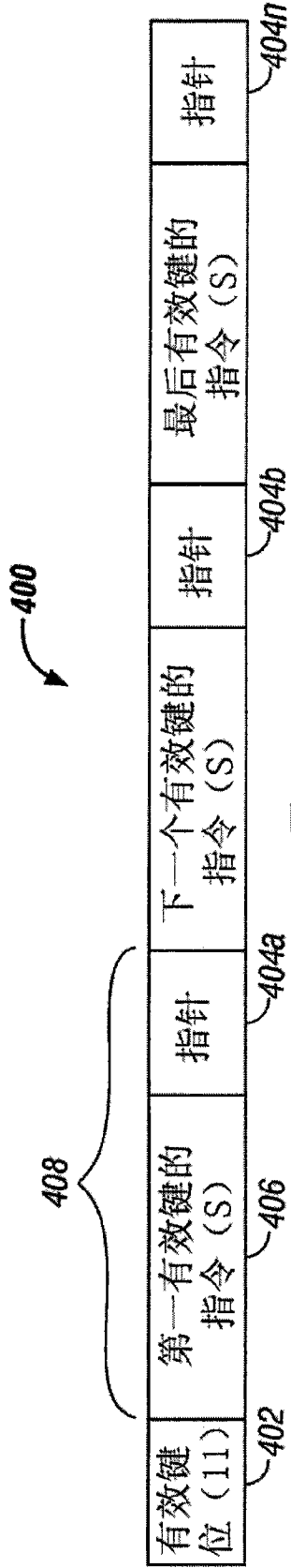


图 8A

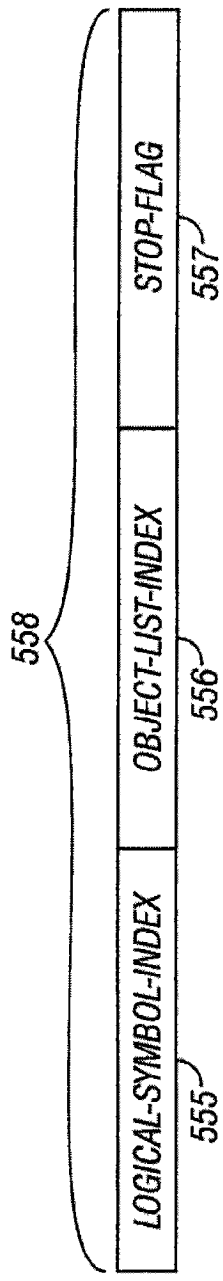


图 8B

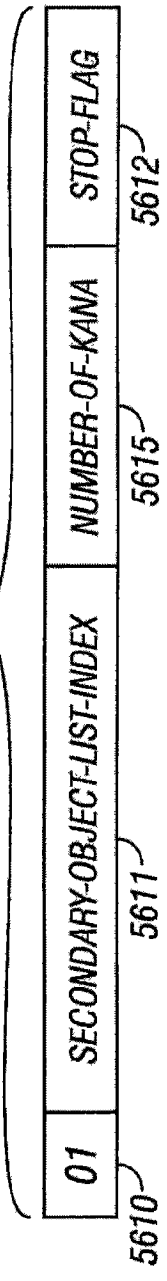
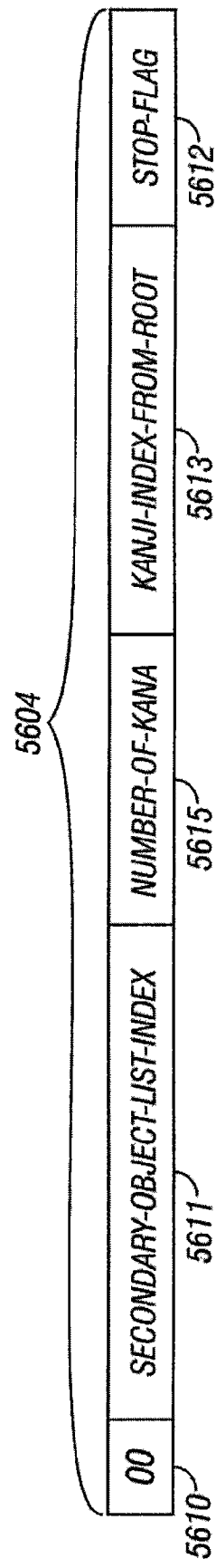


图 8D

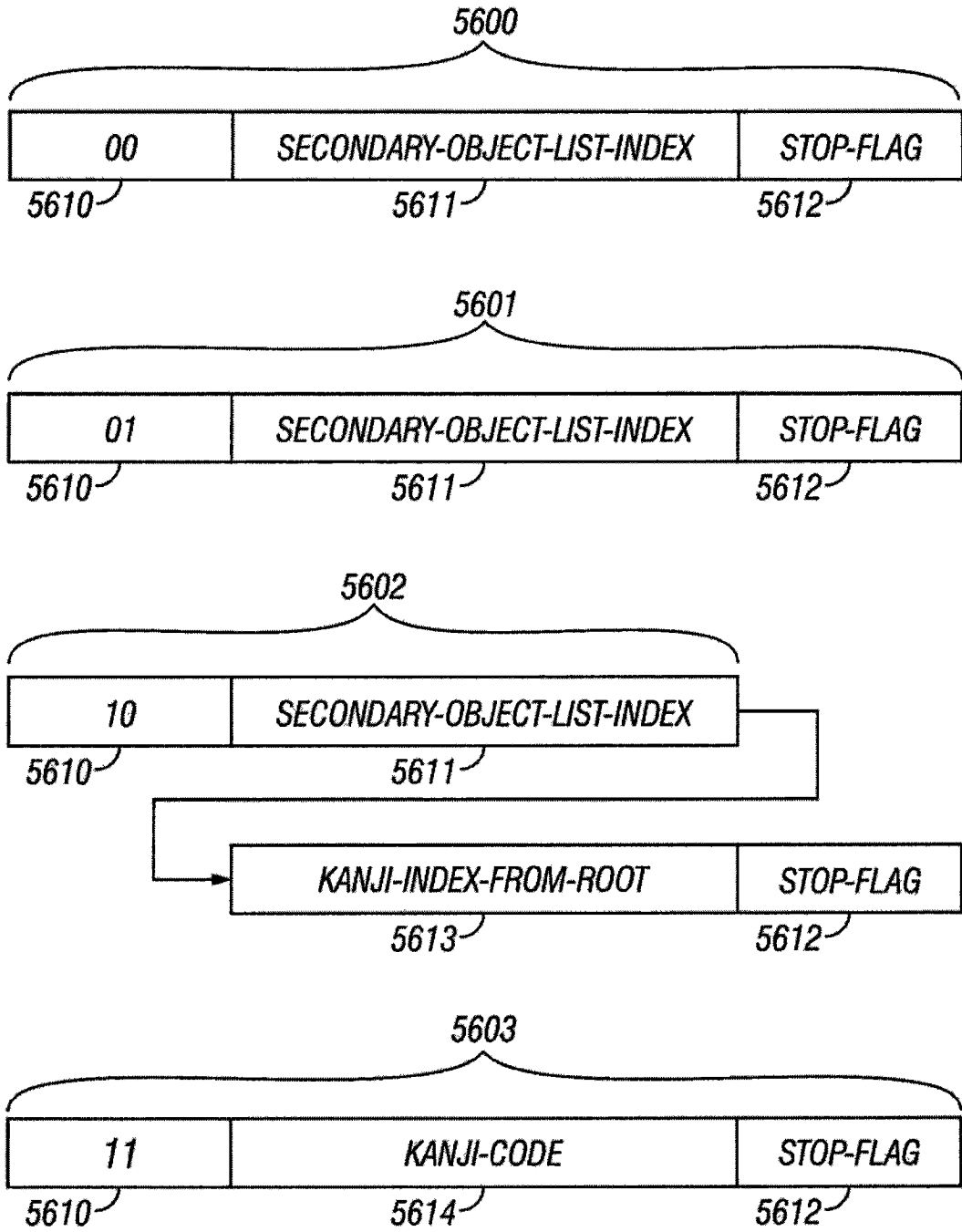


图8C

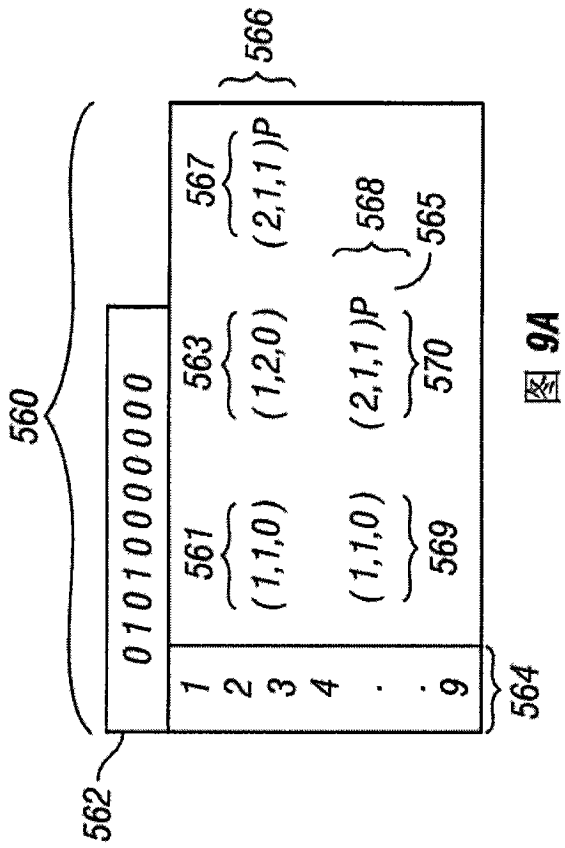


图 9A

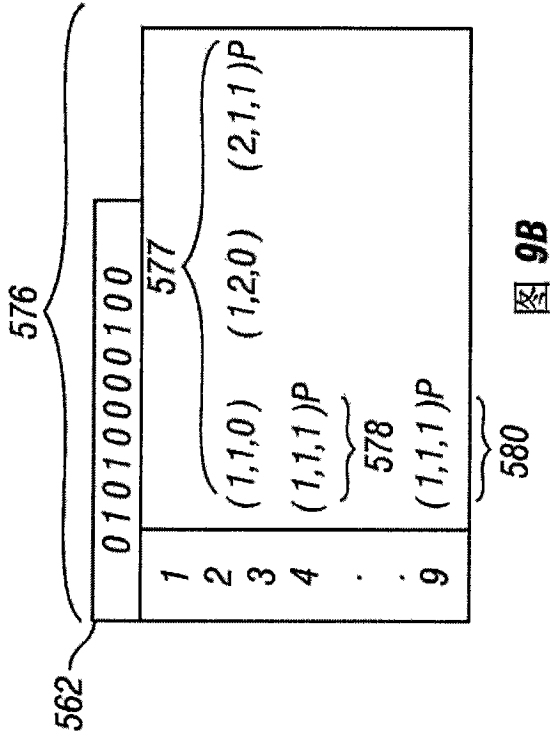


图 9B

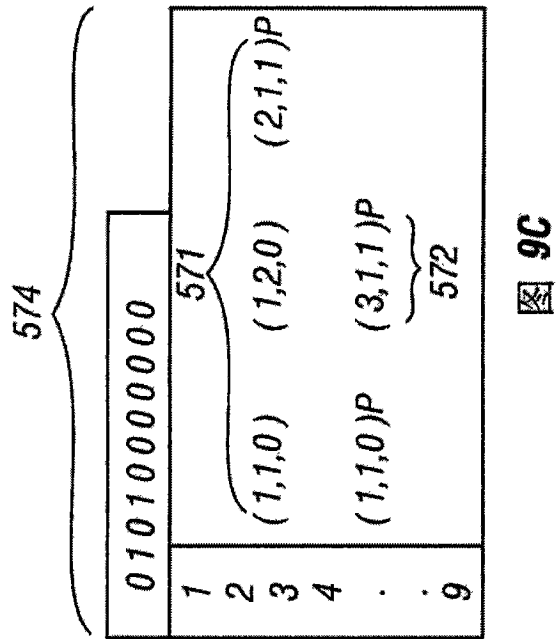


图 9C

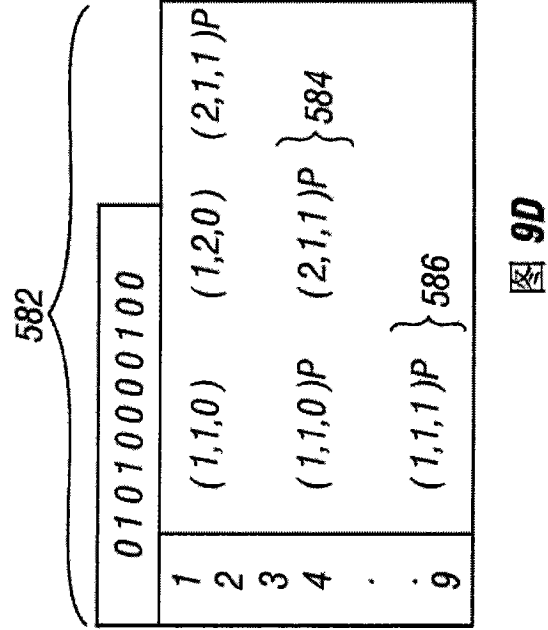


图 9D

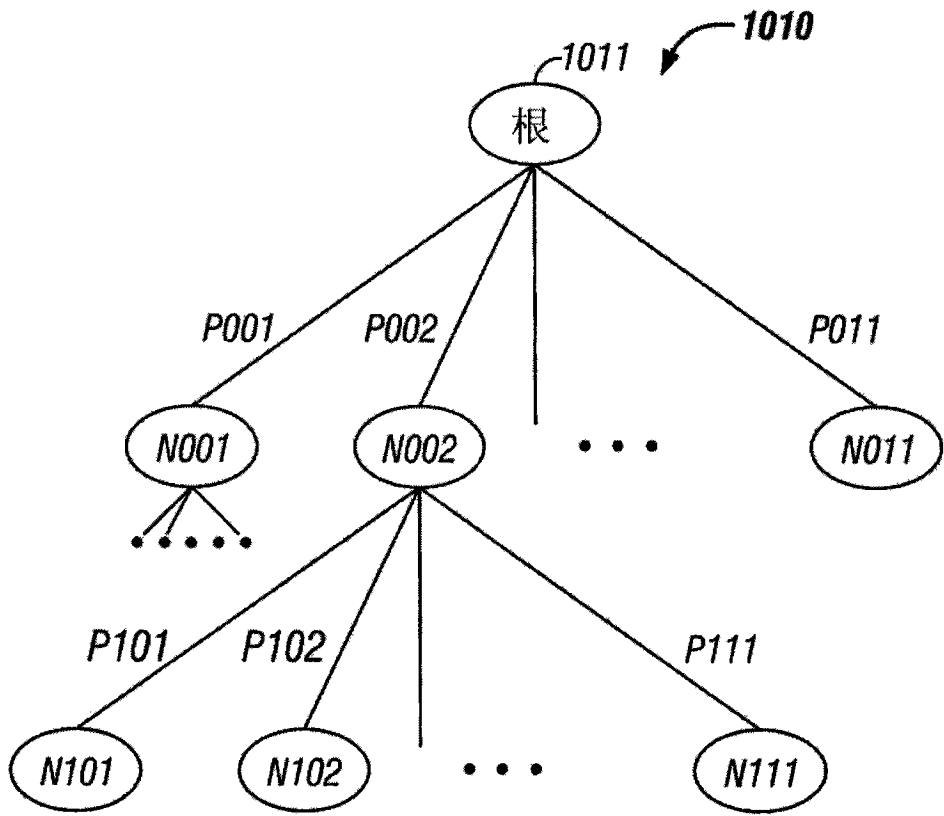


图10

键	未转换的字符	转换的字符
1	. , - ? ! ' . , - ? ! ' . , - ? ! ' .	. , - ? ! ' . , - ? ! ' .
	@ : 1	@ : 1
2	a b c ç à â Á Á Ä 2	A B C Ç À Â Á Á Ä 2
	á ä 2	Á Ä 2
3	d e f é è ê D E F É È Ê	D E F É È Ê
	ë 3	Ë 3
4	g h i î ï í G H I Î Ï Í	G H I Î Ï Í
	4	4
5	j k l 5	J K L 5
6	m n o ô ó ö M N O Ô Ó Ö	M N O Ô Ó Ö
	ñ 6	Ñ 6
7	p q r s ß 7	P Q R S 7
8	t u v ù û ú T U V Ù Û Ú	T U V Ù Û Ú
	ü 8	Ü 8
9	w x y z 9	W X Y Z 9
0	0	0

图22

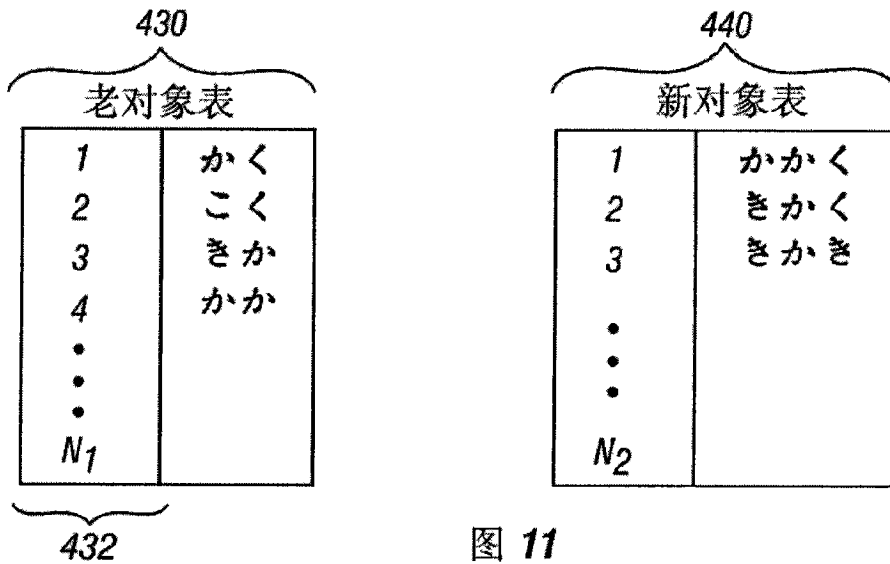


图 11

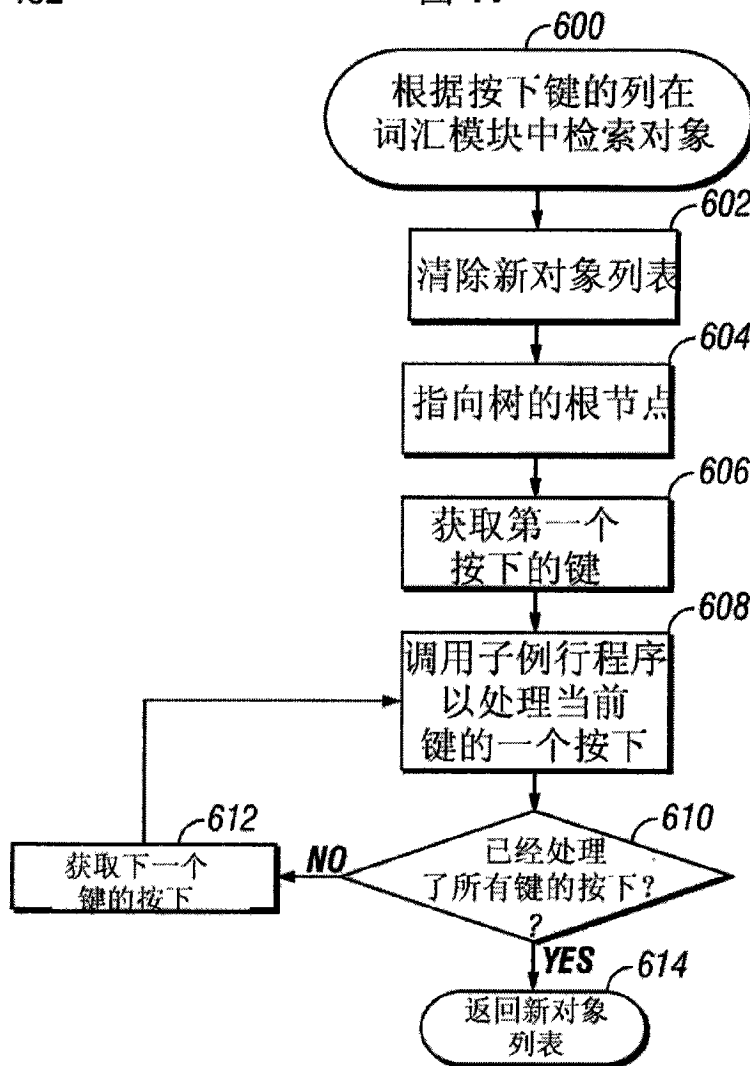


图 12

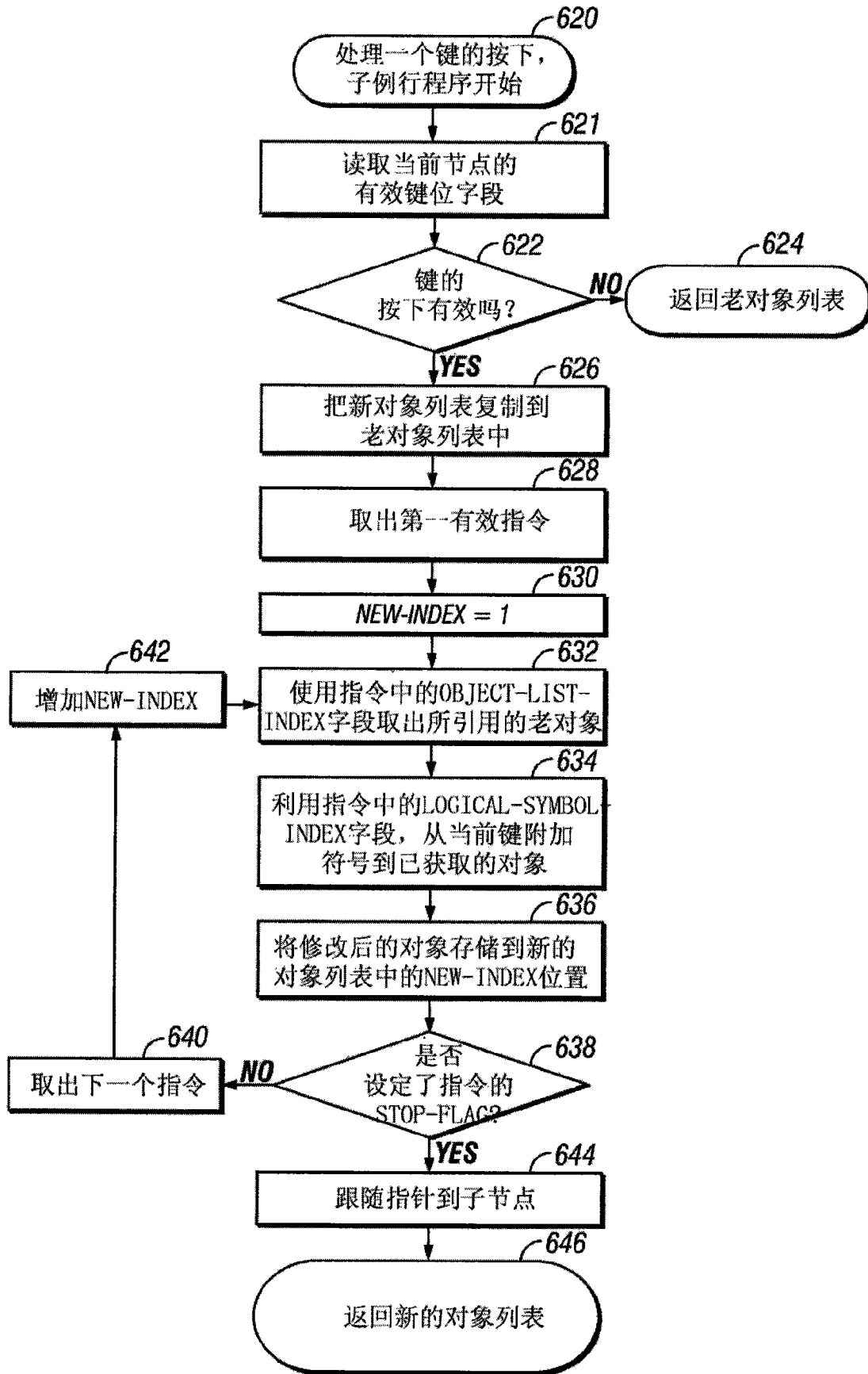


图13

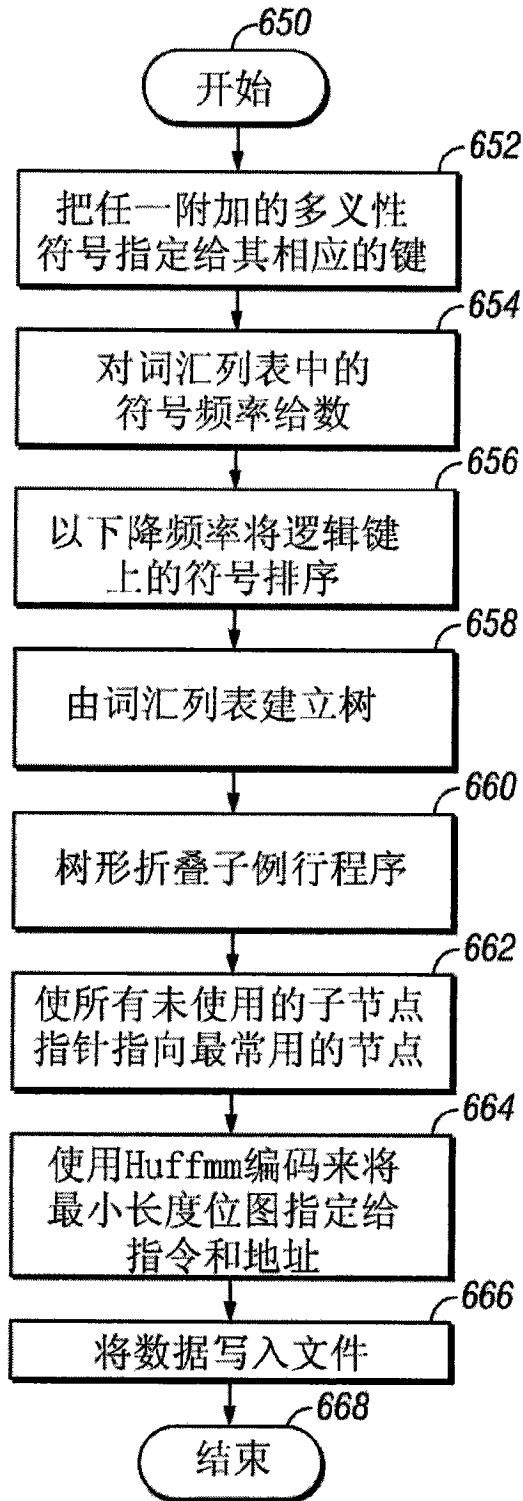


图 14

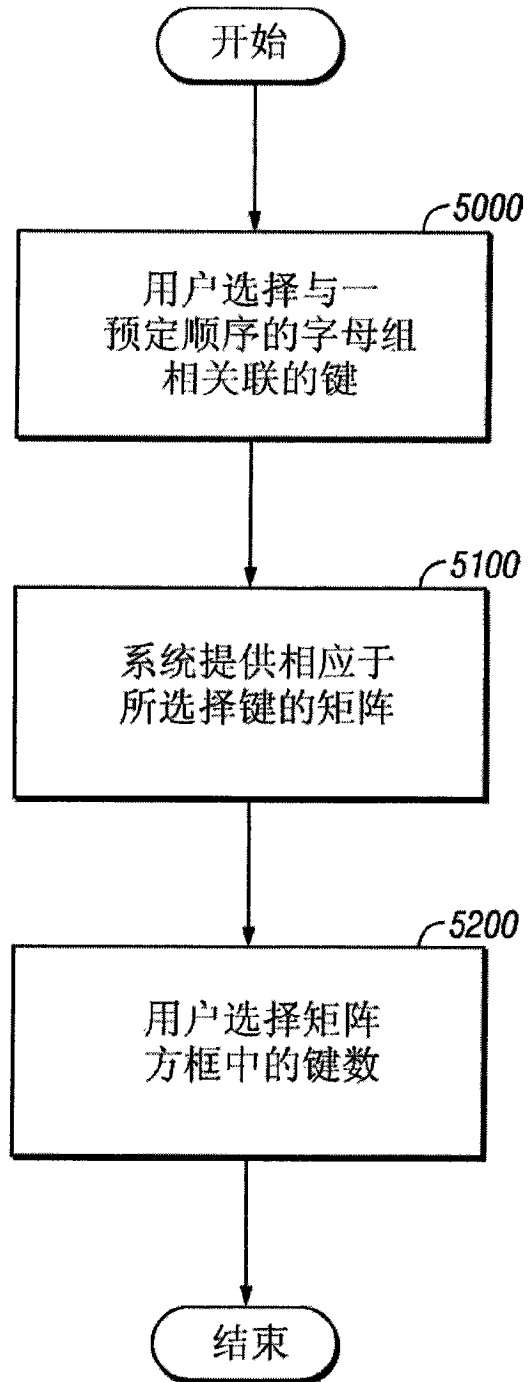


图 24

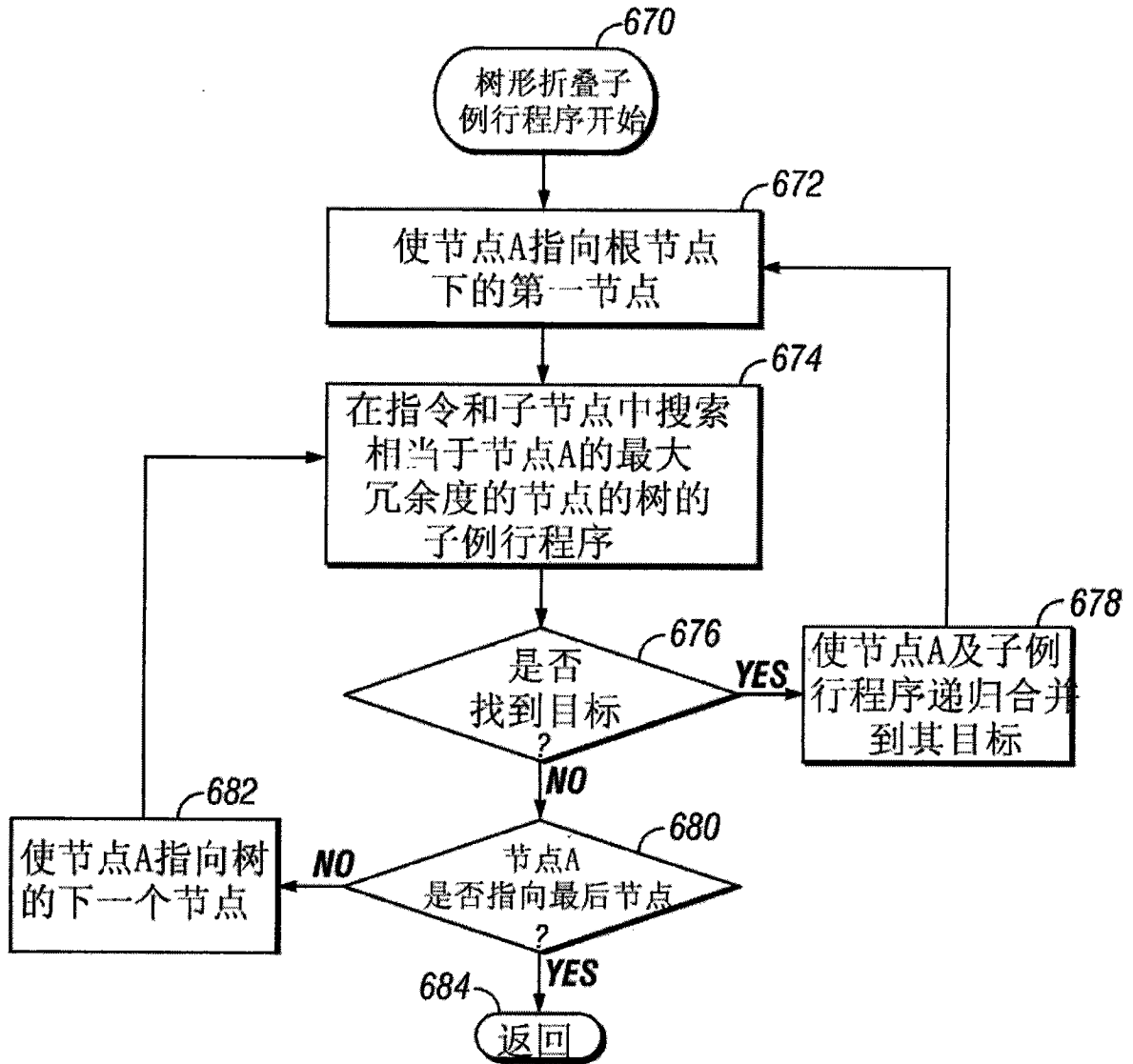


图15

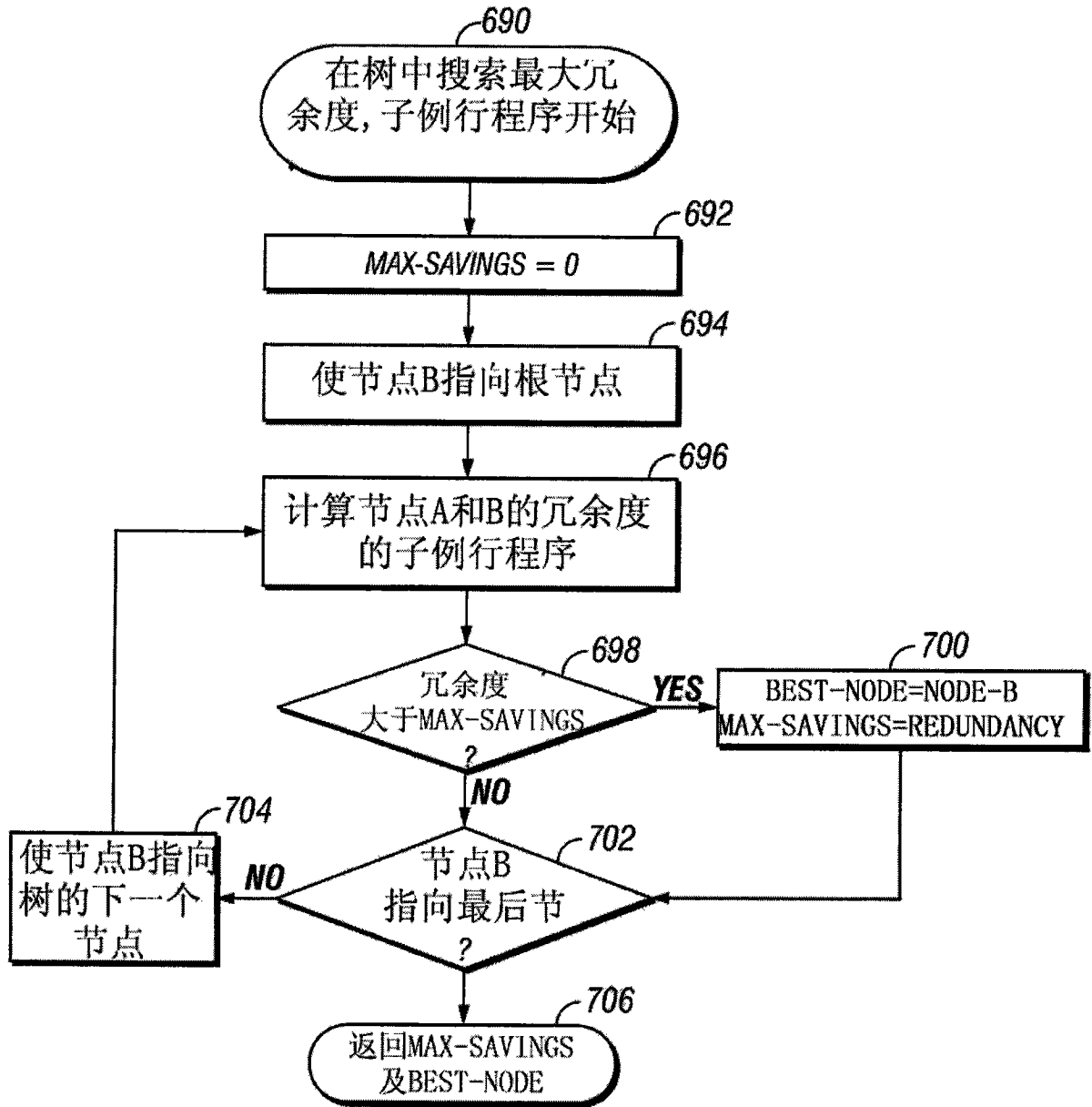


图16

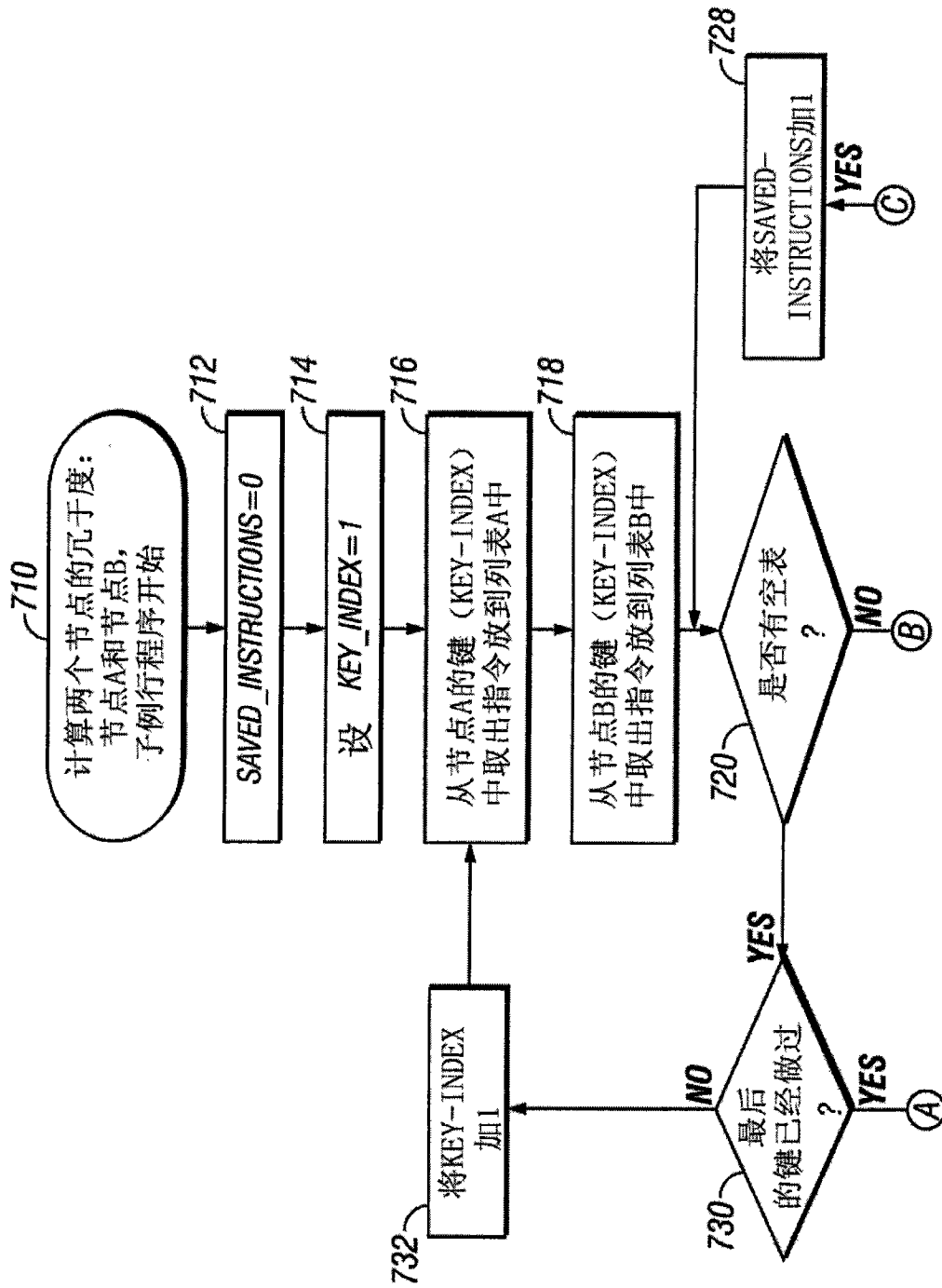


图 17A

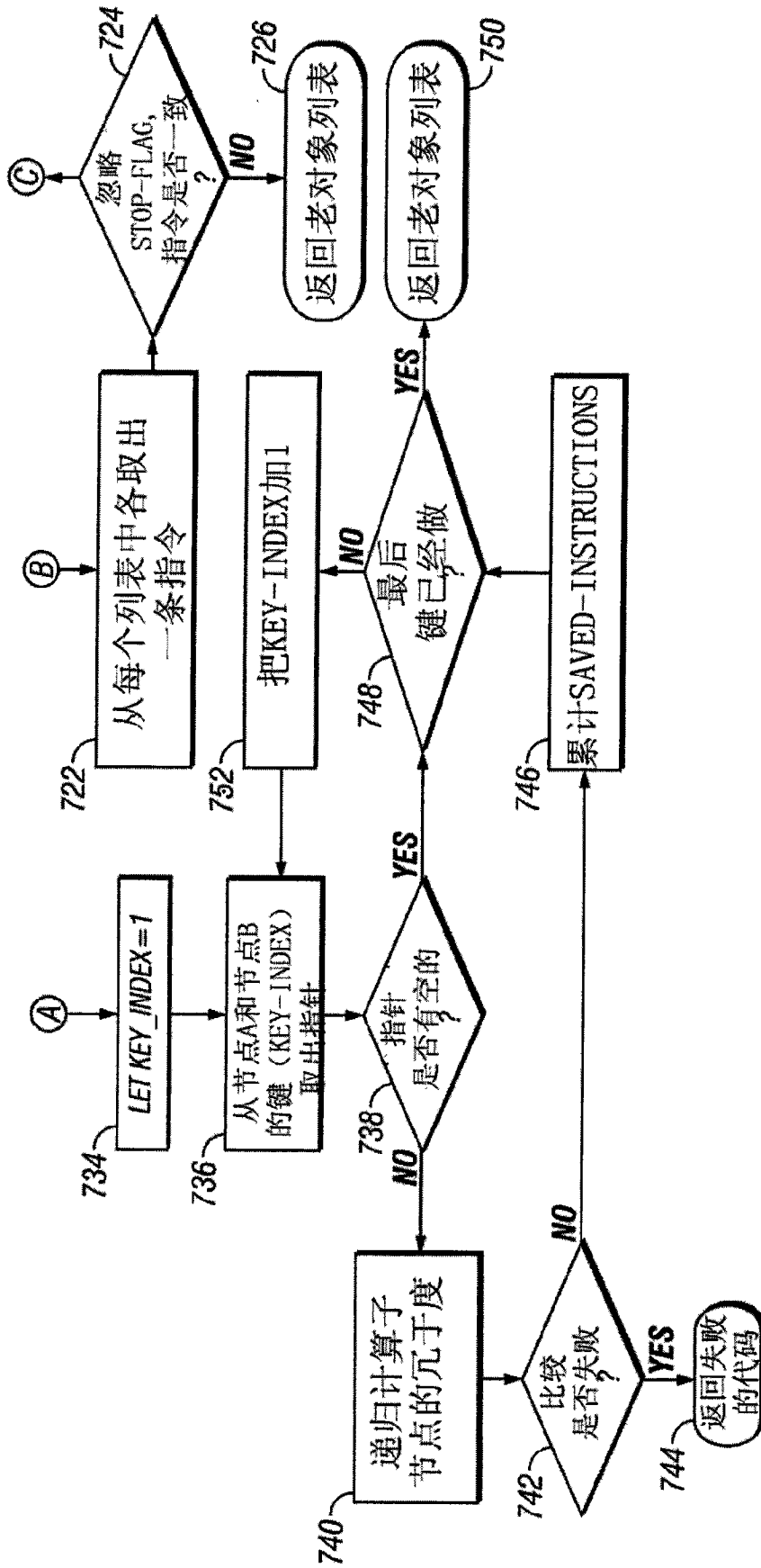


图 17B

级 1 2か	级 2 2か	级 3 2か
<p>Y1: Y0+か M1: M0+ 化 M2: M0+ 講 M3: M0+ 父 M4: M0+ 科 M17: M0+ 価 M23: M0+ 稼 M24: M0+ 働 M25: M0+ 描 M26: M0+ 何 M27: M0+ 欠 M28: M0+ 書 M29: M0+ 賃 Y2: Y0+き M1: M0+ 期 M2: M0+ 器 M3: M0+ 機 M4: M0+ 危 M5: M0+ 気 Y2: Y0+き (CONT.) M6: M0+ 帰 M7: M0+ 企 M8: M0+ 幾 M9: M0+ 規 M10: M0+ 聞 Y3: Y0+< M1: M0+ 丸 M2: M0+ 区 M3: M0+ 来 Y4: Y0+こ M1: M0+ 個 M2: M0+ 子 M3: M0+ 小 Y5: Y0+(十) M3: M0+ 気 M1: M0+ 毛 M2: M0+ 消 :</p>	<p>Y1: Y1+< {か<} M1: M0+ 各 M2: M0+ 核 M3: M28+ < {書<} M4: M0+ 閣 M5: M0+ 画 M6: M25+ < {描<} M7: M0+ 格 M8: M0+ 角 M9: M27+ < {欠<} M10: M0+ 確 M11: M0+ 括 M12: M0+ 殻 M13: M0+ 獲 M14: M29+[M3] {賀来} M15: M0+ 劃 M16: M0+ 摺 Y4: Y2+< {こ<} M1: M0+ 国 M2: M0+ 石 Y3: Y2+ き {きき} M1: M3+[M2] {機器} M2: M4+[M3] {危機} Y4: Y4+ {ここ} M1: M1+々 {個々} Y11: Y2+か {きか} M1: M6+[M1] {帰化} M2: M5+[M1] {気化} M3: M10+か {聞か} M4: M8+[M26] {幾何} M5: M7+P {企□} M6: M9+P {規□} Y16: Y1+か {かか} M1: M28+か {書か} M2: M17+P {価□} Y17: Y2+こ {きこ} M1: M6+P {帰□}</p>	<p>Y1: Y16+< {かか<} M1: M2+[M7] {価格} Y2: Y11+< {きか<} M1: M5+[M5] {企画} M2: M6+[M7] {規格} Y3: Y17+< {きこ<} M1: M1+[M1] {帰国} Y9: Y1+ {かかか} M1: M1+[M2] {各課} Y13: Y11+き {きかき} M1: M1+[M2] {気化器}</p>

图18

例 1

OBJECT DISPLAYED AT
INSERTION POINT 88

AFTER KEY 21: お
 AFTER KEY 25: あの
 AFTER KEY 22: おなか
 AFTER KEY 68: うなぎ
 AFTER KEY 21: おねがい
 AFTER KEY 62: お願い
 AFTER KEY 23: お願いし
 AFTER KEY 27: お願いしま
 AFTER KEY 23: お願いします
 AFTER KEY 60: お願いします

例 2

OBJECT DISPLAYED AT
INSERTION POINT 88

AFTER KEY 21: お
 AFTER KEY 29: いる
 AFTER KEY 22: いら
 AFTER KEY 68: ありが
 AFTER KEY 24: いりぐち
 AFTER KEY 21: ありがとう
 AFTER KEY 22: ありがとうご
 AFTER KEY 23: ありがとうござ
 AFTER KEY 21: ありがとうござい
 AFTER KEY 27: ありがとうございま
 AFTER KEY 23: ありがとうございます

例 3

OBJECT DISPLAYED AT
INSERTION POINT 88

AFTER KEY 22: か
 AFTER KEY 25: この
 AFTER KEY 60: この
 AFTER KEY 60: くに
 AFTER KEY 62: 国
 AFTER KEY 62: 邦
 AFTER KEY 62: クニ
 AFTER KEY 62: くに
 AFTER KEY 62: 国
 AFTER KEY 60: くに
 AFTER KEY 60: かね
 AFTER KEY 62: 金
 AFTER KEY 62: 鐘

PRESS-AND-HOLD

OF KEY 62: 金

图19

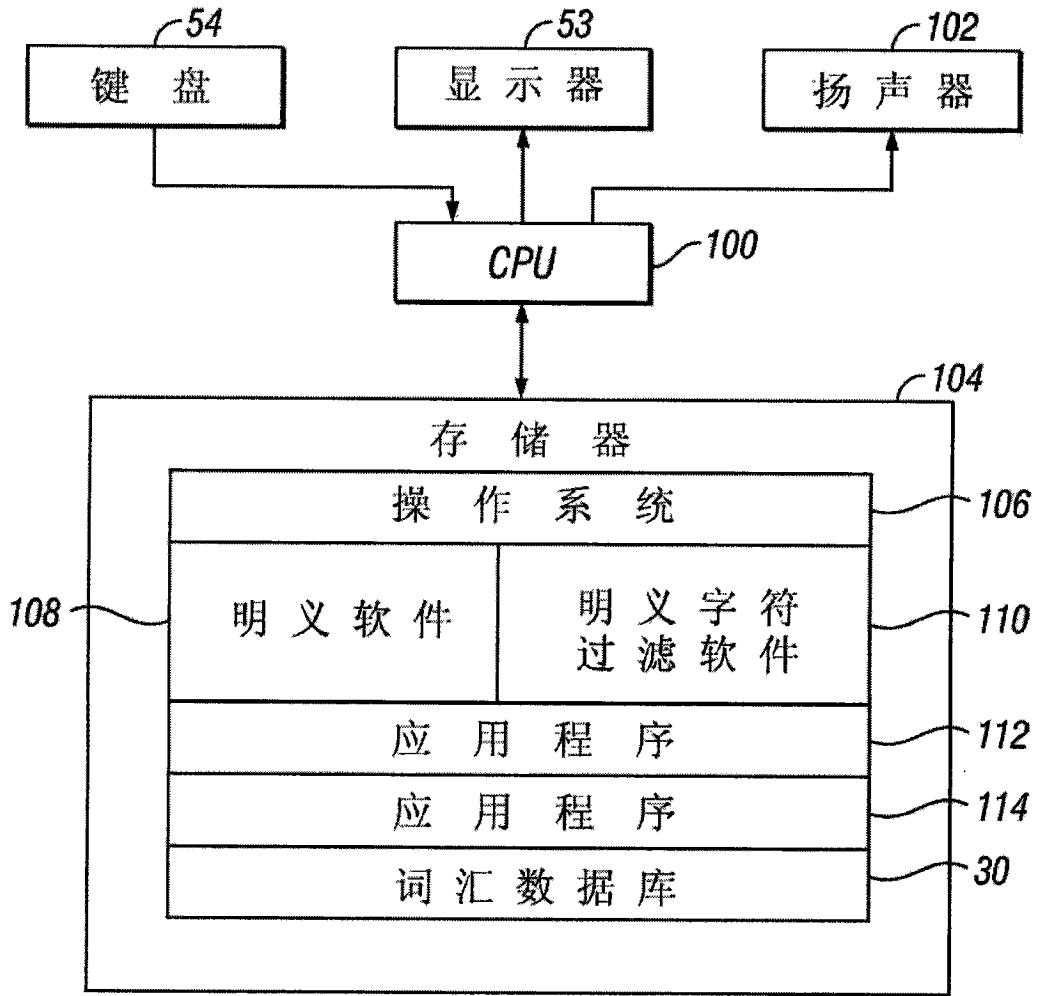


图20

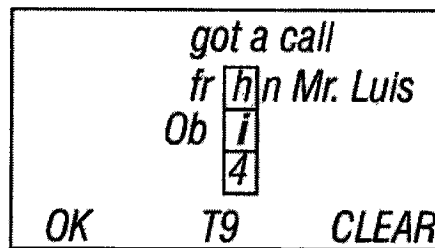


图27

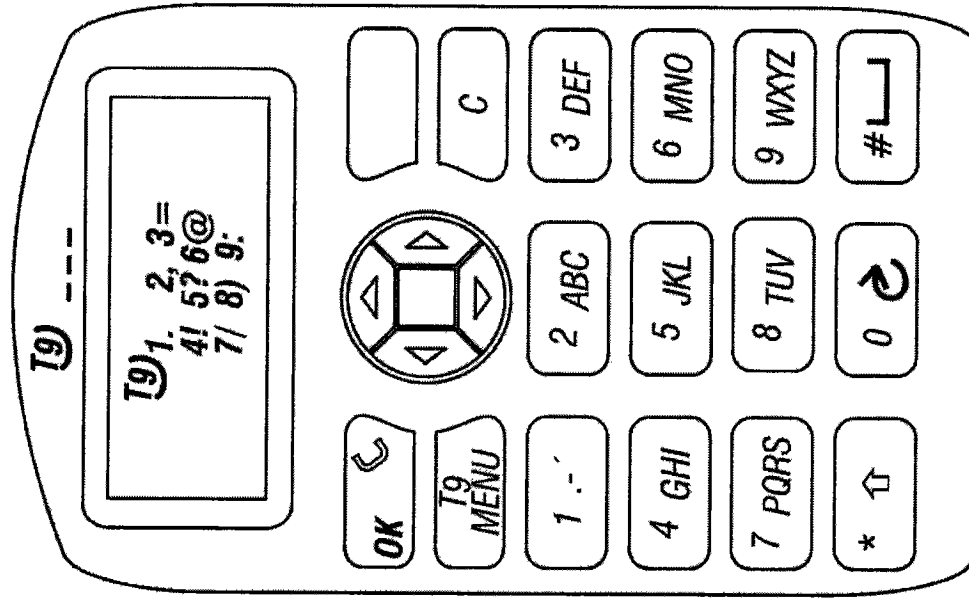


图 21

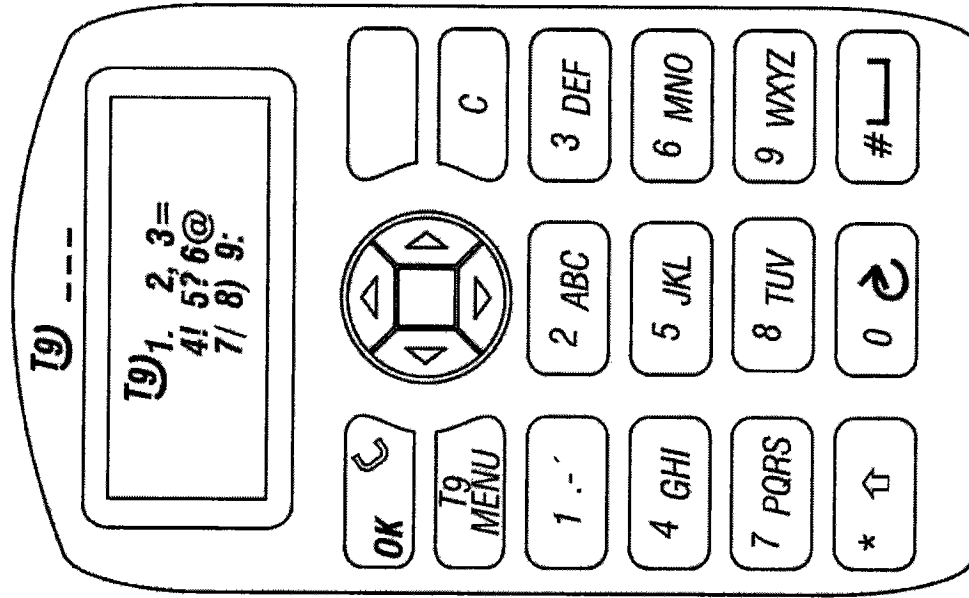


图 23

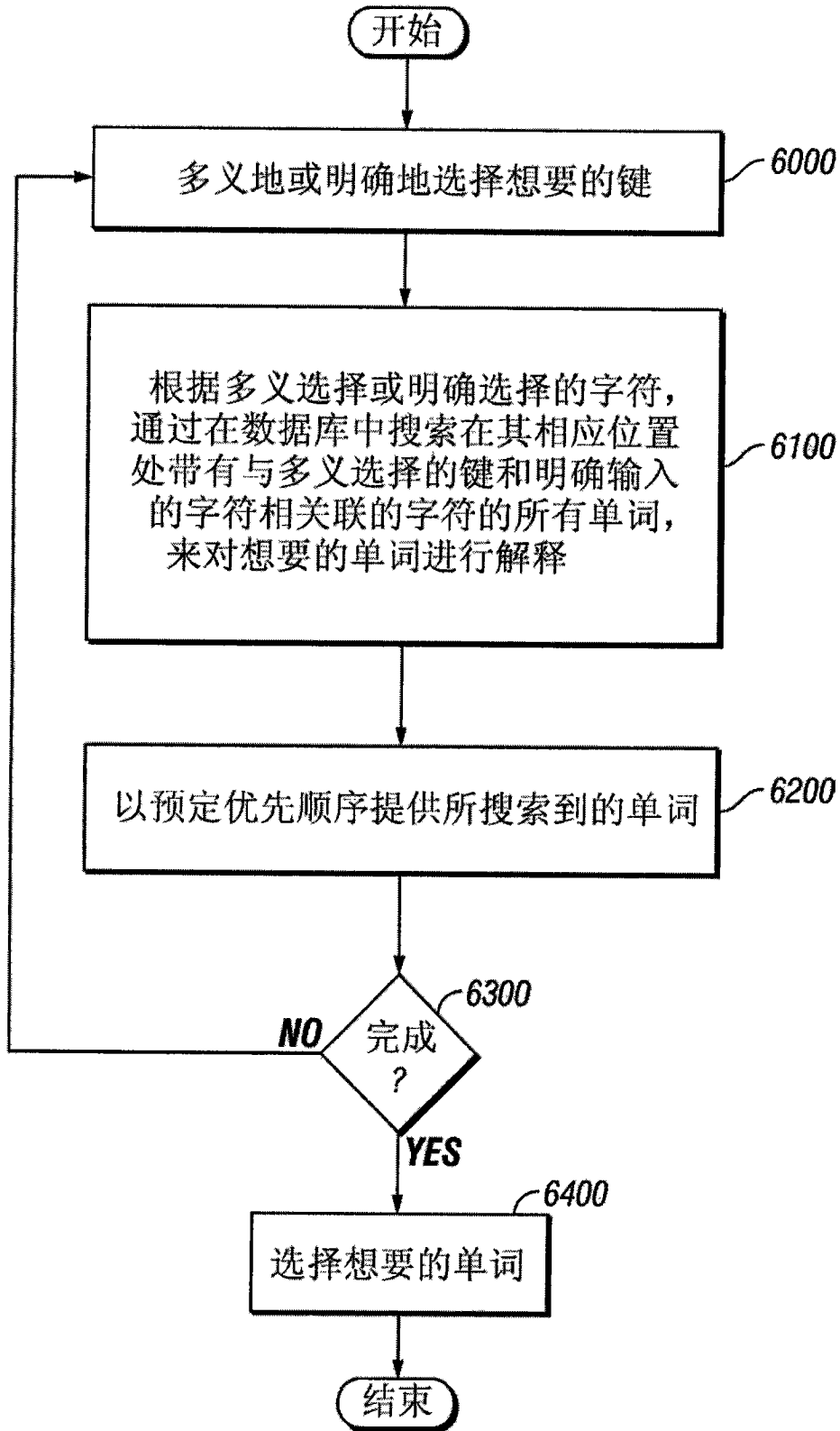


图25

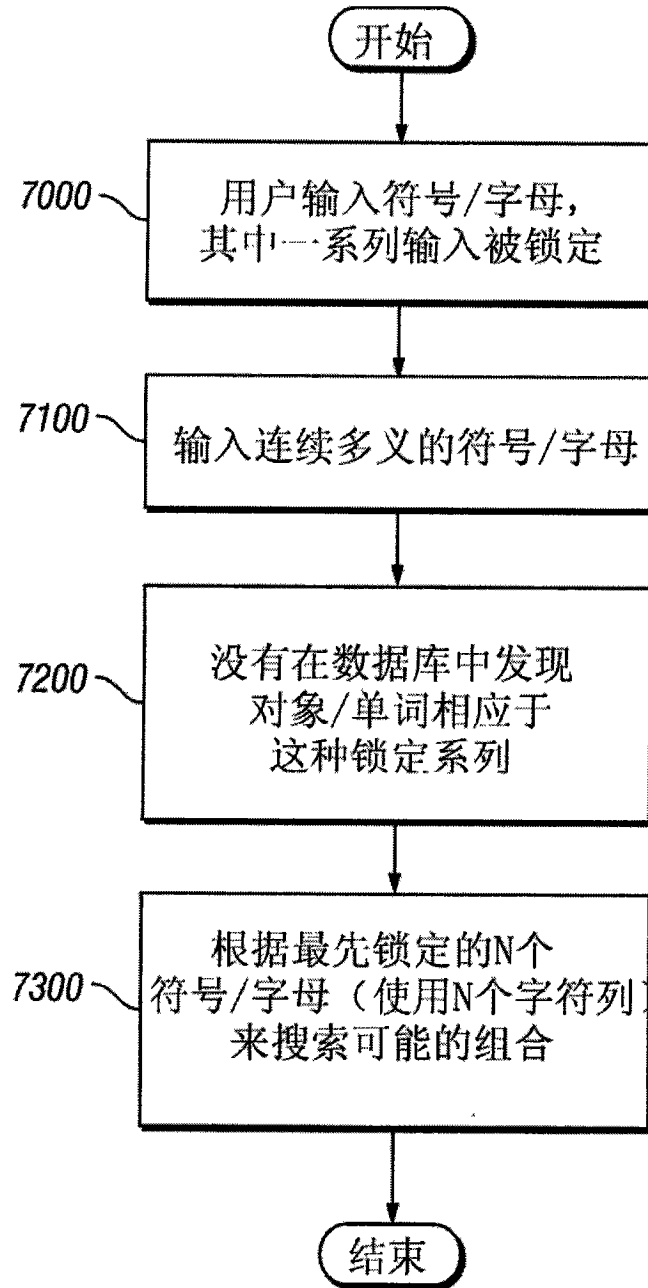


图26

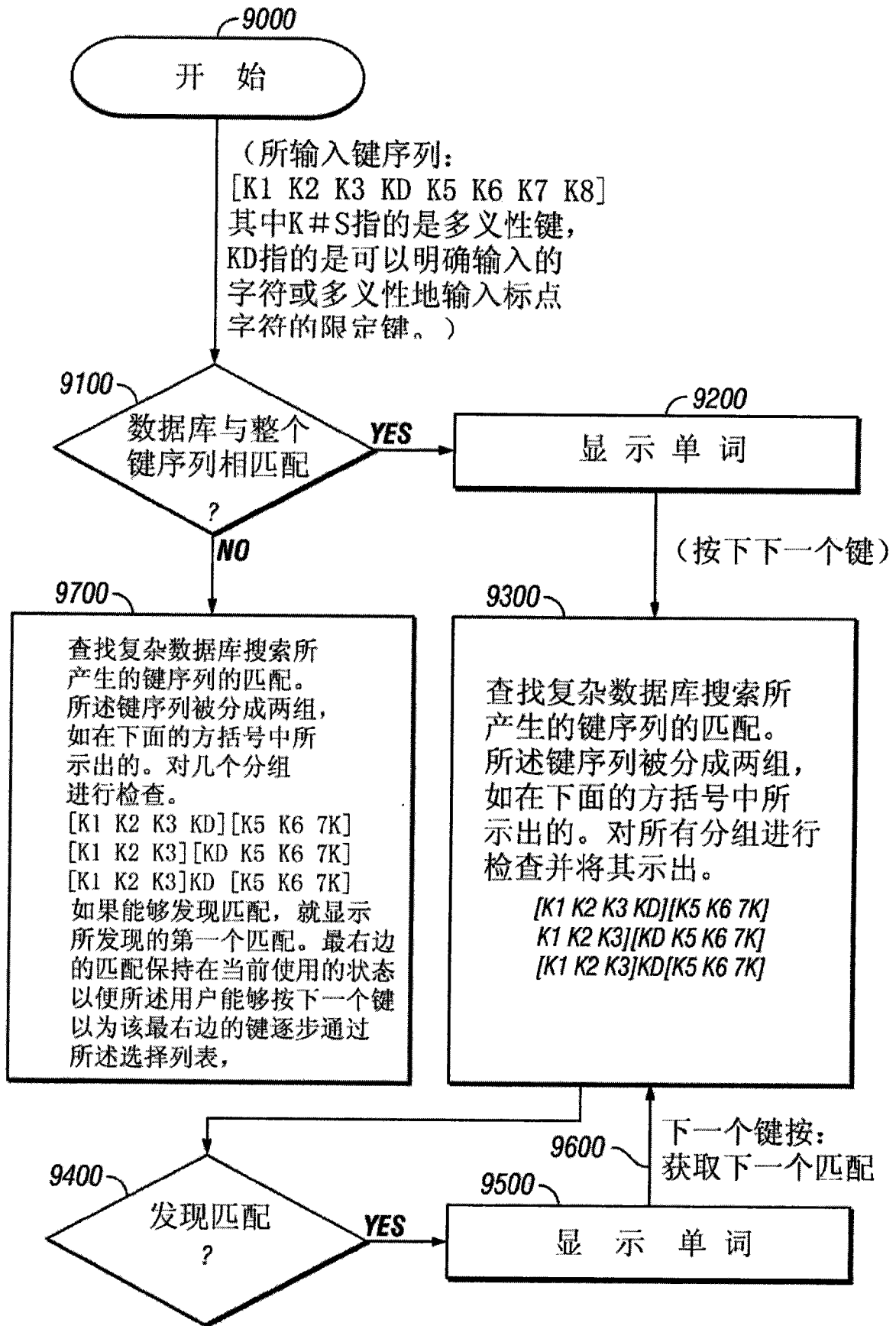


图28