[19]中华人民共和国专利局

[11] 公开号 CN 1120868A

[12] 发明专利申请公开说明书

[21]申请号 94191312.0

[51]Int.Cl[6]

G09G 3/02

[43]公开日 1996 年 4 月 17 日

[22]申请日 94.1.31

[30]优先权

[32]93.2.1 [33]IL[31]104575

[86]国际申请 PCT/US94/01095 94.1.31

[87]国际公布 WO94/18663 英 94.8.18

[85]进入国家阶段日期 95.8.28

[71]申请人 埃尤德·巴伦

地址 以色列海法

共同申请人 爱德华·A·沃尔夫

[72]发明人 埃尤德·巴伦

亚历山大·普里施温

[74]专利代理机构 中国国际贸易促进委员会专利商标事务所

代理人 邓 迅

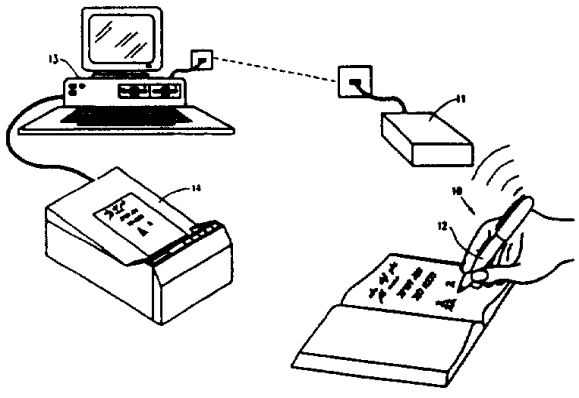
G08C 21/00

权利要求书 4 页 说明书 63 页 附图页数 6 页

[54]发明名称 图像通信装置

[57]摘要

用于手动成像的通信装置，包括：用于检测个人的手动成像特征的装置，这些特征是该个人高度特有的特征，但也包含与所代表的图像有关的信息；以及用于提供一个非个人独立的输出的装置，该输出根据所检测的特征指示图像。



权 利 要 求 书

1. 用于手动成象的通信装置，包括：

用于检测个人的手动成象特征的装置，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息；以及

用于提供一个非个人独立的输出的装置，该输出根据所检测的特征指示图象。

2. 根据权利要求 1 的通信装置，其中所述用于检测特征的装置包含在一个手持外壳中。

3. 根据权利要求 1 的通信装置，其中所述用于检测特征的装置包含在一个图形输入板组件中。

4. 根据权利要求 1 的通信装置，还包括用于传输非个人独立的输出的装置。

5. 根据权利要求 2 的通信装置，还包括用于传输非个人独立的输出的装置。

6. 根据权利要求 3 的通信装置，还包括用于传输非个人独立的输出的装置。

7. 根据权利要求 4 的通信装置，其中所述用于传输的装置用于传输能够用来重构个人的手动成象形式的信息。

8. 根据权利要求 2 的通信装置，其中所述用于检测的装置不需要图形输入板。

9. 根据权利要求 4 的通信装置，其中所述用于传输的装置包括一个调制解调器。

10. 根据权利要求 4 的通信装置，其中所述用于传输的装置用于以传真格式通信。

11. 根据权利要求 4 的通信装置，其中所述用于传输的装置用于以压缩的非光栅格式通信。

12. 根据权利要求 4 的通信装置，其中所述用于传输的装置用于有线通信。

13. 根据权利要求 4 的通信装置，其中所述用于传输的装置用于无线通信。

14. 根据前述任一权利要求的通信装置，其中所述用于检测特征的装置包括用于检测手动成象期间运动的瞬时角度的装置。

15. 根据权利要求 14 的通信装置，其中所述用于提供非个人独立的输出的装置用于提供手动成象期间所产生的笔划的输出指示。

16. 用于手动成象的通信装置，包括：

用于检测个人的手动成象特征的装置，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息；以及

用于提供一个输出的装置，该输出根据所检测的特征指示图象，并且其中

所述用于检测特征的装置包括用于检测手动成象期间运动的瞬时角度并提供所产生笔划的输出指示的装置。

17. 用于传输手动成象的装置，包括：用于检测运动并提供压缩形式的输出的手持装置，该输出能够由常规调制解调器、局域网或其他通信媒体发送。

18. 根据权利要求 1 至 13 和 16 至 17 中任一项的通信装置，

还包括：用于接收所传输的笔划内容信息并用于从中重构手动成象信息的装置。

19. 根据权利要求 14 的通信装置，还包括：用于接收所传输的笔划内容信息并用于从中重构手动成象信息的装置。

20. 根据权利要求 15 的通信装置，还包括：用于接收所传输的笔划内容信息并用于从中重构手动成象信息的装置。

21. 根据权利要求 18 的通信装置，其中所述用于接收的装置用于重构三维手动成象信息。

22. 根据权利要求 19 的通信装置，其中所述用于接收的装置用于重构三维手动成象信息。

23. 根据权利要求 20 的通信装置，其中所述用于接收的装置用于重构三维手动成象信息。

24. 用于手动成象的通信装置，包括：

用于检测手动成象期间的运动并提供压缩形式的笔划内容的输出指示的装置；以及

用于接收所传输的笔划内容信息并用于从中重构手动成象信息的装置。

25. 根据权利要求 24 的通信装置，其中所述用于接收的装置用于重构三维手动成象信息。

26. 用于手动成象的通信方法，包括：

检测个人的手动成象特征，该特征是该个人高度特有的特征，但也包含与所代表的图象相关的信息；以及

提供非个人独立的输出，该输出根据所检测的特征指示图象。

27. 用于手动成象的通信方法，包括：

检测个人的手动成象特征，该特征是该个人高度特有的特征，但也包含与所代表的图象相关的信息；以及

提供根据所检测的特征指示图象的输出，且其中

所述检测步骤包括检测手动成象期间运动的瞬时角度并提供所产生笔划的输出指示。

28. 用于传输手动成象的方法，包括：用于检测运动并提供压缩形式的输出的手持装置，该输出能够由常规调制解调器、局域网或其他通信媒体发送。

29. 用于手动成象的通信方法，包括：

检测手动成象期间的运动并提供压缩格式的笔划内容的输出指示；以及

接收所传输的笔划内容信息并从中重构手动成象信息。

说　　明　　书

图象通信装置

本发明一般涉及手写体和绘图通信装置。

在联机手写体分析领域中有大量活动。*Charles C. Tappert* 等人在1990年8月第12卷第8期《*IEEE* 论文集：模式分析和机器智能》的论文“联机手写体识别的技术状态”中回顾了截止到1990年的现有技术。

一般而言，当前联机手写体分析用于两种不同的应用中：等同检验以及手写字母和数字的计算机输入。这两种应用具有明显相反的工作要求和目标。用于等同检验的手写体分析检测对每个个人不同的手写体特征，因而能用来明确地标识给定的个人。相反，用于字母数字数字输入的手写体分析试图将对于等同检验是重要的特殊特征的影响最小化，而集中于与独立于个人书写者的给定符号相关联的通用手写体特点。

提供用于字母数字计算机输入的手写体分析的现有系统一般适于识别一个符号的外观而不是它是如何建立的。因此，这样的系统采用数字化仪或图形输入板。

另一方面，签名检验系统试图标识书写者的生物统计特点，并采用诸如书写过程中的压力和加速度这样的指示。

美国专利4,345,239采用笔加速度用于签名检验系统中。美国专利5,054,088采用手写体的加速度和压力数据特点用于等同

检验。如上述专利所指出的，笔加速度用于签名检验，因为它是个人特征，是每个个人的特点。于是，笔加速度还未用于手动成象(***hand imaging***)的通信中。

美国专利4,817,034描述一种采用数字化板的计算机化的手写体复制系统。美国专利4,641,354描述用于识别和显示手写字符和图形的装置，其中未识别的笔划信息保留在显示屏幕上。美国专利4,715,102描述一种涉及模式识别的方法和装置。美国专利4,727,588描述一种用于自动调节和编辑手写文本图象的系统，它将格式信息保存在手写文本中。美国专利4,703,511描述一种书写输入和动态再产生装置，其中将一个时间依赖代码嵌在书写路径中。

本发明目的在于提供改进的手写体和绘图通信装置。为简便和简明起见，说明书和权利要求书通篇使用术语“手动成象”来指手写活动和绘图活动，以及任何其他产生二维或三维图象的手的移动。

这样，根据本发明的最佳实施方式，提供一种用于手动成象的通信装置，它包括：用于检测个人的手动成象特征的装置，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息，还包括：用于提供一个非个人独立的输出的装置，它根据所检测的特征显示图象。

词组“非个人独立的输出”是指一个包含非个人特点以及可选择地包含个人特点的输出。非个人特点可以包括手动成象结果的几何表示。非个人独立的输出一般是能够(例如由能以标准格式读数据的任何设备)通用地传输和读取的形式。

根据本发明的最佳实施方式，用于检测的装置包含在一个手持外壳中。用于检测的装置最好包括用于传输非个人独立的输出的装置。

根据本发明的最佳实施方式，用于传输的装置用来传输能用来重构个人的手动成象形式的信息。

根据本发明的最佳实施方式，用于检测的装置不需要图形输入板。另一方案是，可以包括一个图形输入板。另外根据本发明的最佳实施方式，用于获取和编码的装置经过一个调制解调器通信。该通信可以是传真格式或者是压缩的非光栅格式。最好，该通信是无线通信。

根据本发明的最佳实施方式，也提供一种用于手动成象的通信装置，它包括：用于检测个人的手动成象特征的装置，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息，还包括：用于提供一个非个人独立的输出的装置，它根据所检测的特征显示图象。

另外根据本发明的最佳实施方式，用于检测特征的装置包含在一个手持外壳中。

仍然根据本发明的最佳实施方式，用于检测特征的装置包含在一个图形输入板组件中。

另外根据本发明的最佳实施方式，该通信装置还包括用于传输非个人独立的输出的装置。

仍然根据本发明的最佳实施方式，该通信装置用来传输能用来重构个人的手动成象形式的信息。

另外根据本发明的最佳实施方式，检测装置不需要图形输入

板。

另外根据本发明的最佳实施方式，用于传输的装置包括一个调制解调器。

仍然根据本发明的最佳实施方式，用于传输的装置用于以传真格式通信。

另外根据本发明的最佳实施方式，用于传输的装置用于以压缩的非光栅格式通信。

仍然根据本发明的最佳实施方式，用于传输的装置用于有线通信。

仍然根据本发明的最佳实施方式，用于传输的装置用于无线通信。

另外根据本发明的最佳实施方式，用于检测特征的装置包括用于检测手动成象期间的运动的瞬时角度的装置。

进一步根据本发明的最佳实施方式，用于提供一个非个人独立的输出的装置用于提供在手动成象期间所产生的笔划的输出指示。

根据本发明的另一最佳实施方式，还提供一种用于手动成象的通信装置，它包括：用于检测个人的手动成象特征的装置，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息，还包括：用于提供输出的装置，它根据所检测的特征显示图象；并且其中用于检测特征的装置包括用于检测手动成象期间的运动的瞬时角度并提供所产生的笔划的输出指示的装置。

根据本发明的另一实施方式，还提供一种用于传输手动成象的装置，它包括：用于检测运动并提供一个压缩形式的输出的手持

装置，该输出能够由常规调制调解器、*LAN* 或其他通信媒体传输。

另外根据本发明的最佳实施方式，该装置包括：用于接收所传输的笔划内容信息并用于从中重构手动成象信息的装置。

仍然根据本发明的最佳实施方式，用于接收的装置用于重构三维的手动成象信息。

根据本发明的另一实施方式，还提供一种用于手动成象的通信装置，它包括：用于检测手动成象期间的运动并提供压缩形式的笔划内容的输出指示的装置，还包括：用于接收所传输的笔划内容信息并用于从中重构手动成象信息的装置。

另外根据本发明的最佳实施方式，用于接收的装置用于重构三维的手动成象信息。

根据本发明另一实施方式，还提供一种用于手动成象的通信方法，包括：检测个人的手动成象特征，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息，还包括：提供一个非个人独立的输出，该输出根据所检测的特征显示图象。

根据本发明另一实施方式，还提供一种用于手动成象的通信方法，包括：检测个人的手动成象特征，这些特征是该个人高度特有的特征，但也包含与所代表的图象有关的信息，还包括：提供一个根据所检测的特征指示图象的输出，并且其中检测特征的步骤包括检测在手动成象期间的运动的瞬时角度并提供一个所产生的笔划的输出指示。

根据本发明的另一实施方式，还提供一种传输手动成象的方法，包括：用于检测运动并提供一个压缩形式的输出的手持装置，该输出能够由常规调制调解器、*LAN* 或其他通信媒体传输。

根据本发明的另一实施方式，还提供一种用于手动成象的通信方法，包括：检测在手动成象期间的运动并提供压缩形式的笔划内容的输出指示，并包括：接收所传输的笔划内容信息并从中重构手动成象信息。

通过以下详述并结合附图，更能理解和体会本发明。

图 1 是工作环境中的根据本发明的最佳实施方式构造和工作的用于获取手动成象并对之编码的装置的图示。

图 2 是图 1 的获取和编码装置的最佳结构的简要图示；

图 3 是在工作环境中根据本发明的最佳实施方式构造和工作的用于获取手动成象并对之编码的另一装置的图示；

图 4 是图 3 的获取和编码装置的最佳结构的简要图示；

图 5*A* 和图 5*B* 是采用图 1 和图 2 的装置的袖真通信机部件和笔的图示；

图 6 是根据本发明的最佳实施方式的用于传输三维手动成象信息并对之构重的装置的简要图示。

以下是对附录的说明，附录有助于理解和体会本发明。

附录 *A* 是本发明人对与经笔划手动成象的特点有关的发现的详细公开；

附录 *B* 是包括微控制器 26 编程部分的最佳实现的计算机程序列表；

附录 *C* 是对附录 *B* 的计算机程序列表的解释；

附录 *D* 是包括微控制器 46 编程部分的最佳实现的计算机程序列表；

附录 *E* 是对附录 *D* 的计算机程序列表的解释；

附录 *F* 是包括计算机 13 编程部分的最佳实现的计算机程序列表；

附录 *G* 是对附录 *F* 的计算机程序列表的解释；

附录 *H* 是包括计算机 13 编程部分的最佳实现的计算机程序列表；

附录 *I* 是对附录 *H* 的计算机程序列表的解释。

本发明人已经发现，每个书写者利用在一个含有大约 12 至 14 个笔划的集合中选择的笔划产生常规字母数字字符和图形图象，即手动成象，该集合是每个个人所特有的。本发明采用这种认识来提供用于手动成象的通信装置。附录 *A* 含有对本发明人的发现的详细公开。

现在参照图 1，示出了在一般工作环境下的根据本发明的最佳实施方式构造和工作的用于传输手动成象 10 的笔，它通过无线通信与通信机 11(例如，具有相联接收机的电话或射频调制解调器，如在市场上可从德克萨斯州 *Dallas* 的 *RF Monolithics* 公司购得的 *RB*1023 型射频接收机)通信。另一方案是，经过任何其他适用的通信媒体(例如，局域网和蜂窝电话网)进行通信。

手动成象通信笔 10(在图 2 中有更详细的图示)，可以用在任何书写表面上，或者可替代地，不用书写表面且不需任何特殊的板或书写衬底。手写体输入装置最好包括一个普通笔一般大小和形状的外壳 12。

通信机 11 可经过电话或同轴电缆或无线设备与任何适用的接收机通信，如具有相联打印机 14 的计算机终端 13。应懂得，用户利用装置 10 产生的手动成象几乎可以瞬时地显示在远程的打印硬拷

贝上或计算机屏幕上。相同的一般类型的装置可以用于电视屏幕的显示或用于传真接收机的输出。

如图 2 所示，设在外壳 12 中的是一个墨水贮存胆和输出笔尖组件 16，它可以根据任何常规方式构造和工作。另一方案是，不提供墨水输出。根据本发明的最佳实施方式，设置在外壳 12 前部的是一个加速度表 20，最好以三维方式工作。满足本发明的大小和功率要求的一般加速度表包括三个相互正交安装的 3031—002 型加速度表，在市场上可从英国伦敦 ***Kirby*** 大街 20—24 号的 ***EuroSensor*** 购得。另一方案是，可以使用三个以上的 3031—002 型加速度表。

加速度表 20 的输出经一个运算放大器 24(例如，在市场上可从加利弗尼亚 ***Milpitas*** 的 ***Liner Technology*** 公司得到)加到微控制器 26(例如，包括 ***A/D*** 转换器的 ***Hitachi H***8/536 微控制器)。微控制器 26 用来抽取加速度表 20 所检测的加速度的多个预定特征。本发明的一个特别特征在于：已经发现在手动成象期间从所检测的加速度导出的相当小数量的离散特征足以绘制给定个人的字母数字符号和图形输出。应理解到，这种特征的特点因个人而变化，并且经常希望传输这些个人特点。于是，对微控制器 26 进行编程，以便不仅保留和传输该信息内容而且保留和传输书写者的个人手动成象特点。

提供微控制器 26 功能的 ***C*** 语言编程的软件的最佳程序列表示于附录 ***B***。以附录 ***B*** 的软件为例的微控制器 26 的功能所基于的原理的简述见附录 ***C***。

微控制器 26 用于抽取多个笔划并用常规通用代码(比如，

ASCII 码)对其进行编码，该常规通用代码在任何方式上都不依赖于给定个人的个人手写体特点，并易于为常规计算机、调制解调器等所接受。

微控制器 26 的编码的符号输出最好是与从常规计算机(如，*PC*)常规接收的输出或从调制解调输入的形式兼容或相同。

根据本发明的最佳实施方式，微控制器 26 的编码的输出由无线发射机 32 以无线的方式发射到通信机 11，比如，该通信机可以是在市场上可从 *RF Monolithics* 公司得到的 *MB*1003 型，并与接收机 12(图 1)通信。另一方案是，利用任何其他适用的 *IR* 发射机或无线电发射机。

另一方案是，可以设置非无线通信连接，如这里及下文参照图 5*A* 和 5*B* 所述。在这种情形下，最好设置如闪速 *RAM*34 的非易失性存储器，用以存储微控制器 26 的输出。设置一个适用的电池 33，用以向位于外壳 12 内的装置供电。

应理解到，本发明的装置最好是手持“笔”，可由用户携带并可结合任何适用的通信装置使用。通信装置和计算机以及与之通信的外设不必在任何方式上个人化，因为所有手写体识别硬件和软件都驻留在“笔”中。

现在参照图 3 和图 4，图中示出了本发明的另一实施方式，它采用图形输入板组件 40 代替了笔 10。该图形输入板组件 40，如图 4 所示，可以包括任何常规图形输入板 42，比如与专用笔 44 联用的并向微控制器 46 输出 *X*、*Y* 坐标和笔抬起信号的 *Summagraphics* 兼容图形输入板。

微控制器 46 可以具有上述用于笔划抽取和编码的微控制器

26 的所有相关功能，并可以直接与调制解调/传真部件 48(比如在市场上可从 *Rockwell* 或其他供销商处得到的)通信。提供微控制器 46 功能的 *C* 语言编程的软件的最佳程序列表示于附录 *D*。以附录 *D* 的软件为例的微控制器 46 的功能所基于的原理的简述见附录 *E*。

调制解调/传真部件 48 的输出端可以连接到普通电话或网络插座 50，用于以结合图 1 所述的方式通信。

附录 *F* 给出图形输入板组件 40 所传输的信息的重构功能的 *C* 语言编程的软件列表。该软件可以驻留在计算机 13 或可替代地驻留在任何输出装置中，它的功能是从所传输的手动成象信息中提供有用的输出。附录 *G* 是对附录 *F* 的程序列表中所含的重构功能所基于的原理的简述。

附录 *H* 给出笔 10 所传输的信息的重构功能的 *C* 语言编程的软件列表。该软件可以驻留在计算机 13 或可替代地驻留在任何输出装置中，它的功能是从所传输的手动成象信息中提供有用的输出。附录 *I* 是对附录 *H* 的程序列表中所含的重构功能所基于的原理的简述。

另外根据本发明的最佳实施方式，提供一种便携式信息存储和检索装置，它包括：一个便携式计算机存储器和输出装置，并具有一个上述类型的手持装置作为输入部件。

现在参照图 5*A* 和图 5*B*，图中示出了根据本发明另一实施方式的用于手动成象的获取和编码装置以及通信装置。这里，通信机 100 形成有一个插孔 101，用于可去除地接受获取和编码装置 102。数据通信触片 104 和 106 分别设在装置 102 的一端和插孔 101 中，

用于允许所写信息从装置 102 卸载到通信机 100。最好将通信机设计为袖真尺寸。

现在参照图 6，示出了本发明的用于传输三维信息的装置的使用。使用该装置的用户绘出三维物体(比如，飞机模型)的外形。用户可以如图所示沿着实际三维模型的线条，或可替代地随手画。由用户产生的手动成象经过通信机 11 传输到通信机 120，然后传到诸如计算机数控机床 122 或三维模型制造机 124 这样的应用装置，比如在市场上可从以色列 ***Ramat Hasharon*** 的 ***Cubital*** 有限公司得到。

根据本发明的手动成象信息的传输可以是两种不同方式。当在通信信道中需要压缩时，手动成象信息可以以笔划语言传送。在这种情形下，在远程需要手动成象重构器。

另一方案是，手动成象信息能够通过适用的重构装置在通信机 11 上游重构。在这种情形下，手动成象信息可以用常规 ***CAD*** 格式传输。

本领域熟练的技术人员应理解到，本发明并不限于以上所示和所述。本发明的范围仅由所附权利要求书限定。

附录 *A*

人脑告诉了人手什么

行为透视

Ehud Bar—On

E.B. 研究与发展有限公司

以色列，*Haifa* 32000，*Technion*，*Guttwirth* 大楼

1992 年 9 月 22 日

摘要

该项研究调查了手写过程中手移动的运动原型。这称为“手”和“脑”之间的“语言”，具有其自身的词汇和语法。“词汇”是笔划，语法是如何将笔划组合为笔划序列。将手写看成是交流的高级认识活动，表示为复杂的原动技能，且其调查洞悉到大量的自动过程。该项研究的主要发现在于笔划对于单个书写者是特有的，并刻划出书写者的唯一的原动控制机制的特点。分析了许多书写者所产生的成千个手写字符的动态数据。将时间信号分段为离散笔划单元，并表示为特征空间的向量。利用多种分类技术对那些向量进行分类。我们发现，除了书写期间手的移动能够采用任何形式或形状这一事实之外，一个特定的书写者只采用笔划的一个非常有限的集合。利用多种方法分类的结果产生一个有限的集合，只含有 12 到 14 个类型的笔划，它们构成 90 脑，它推测大量信息，以最少化所需注意资源。

1. 人类的原动行为

当我们谈到所谓“人脑告诉了人手什么”这一话题时，我们所讲的是可能存在于人脑与人手之间的有关“原动控制语言”这类问题。尽管这种“手脑之间的交流”可以从不同观点进行探讨，而我们则要从行为观点方面对其加以研究。也就是说，我们通过对手写体识别试验过程中所收集到的动力数据的分析，研究了量的形迹或称之为“原动程序”。尽管本文中所论述的所有结果只包括手写过程的输出信号，但是我们将尽量使所提出的方法从认知及生物角度看似乎是合理的。

我们先从认识的合理性角度开始论述。认识过程与人类原动行为之间有着很密切的联系。书写是人类通过使用复杂的原动技能表达其思想的一种方式。***Rosenbaum*** 将手写体描述为是若干次内部 ***l*** 翻译过程的极点。首先，一个抽象的信息或想法建立起来了。其次，它被转化为适当的语言表达方式，接着又被转化为一系列输出命令。在说话与写字之间存在着一种基本相似之处，因此我们可以假定它们两者有着相同的基本过程。说话时的音素对应的是写字时的笔划，而说话中的词素则对应写字时的字母。对于较高水平的抽象内容(如：单词，句法，专门词汇，语义学，韵律学以及论述)，情况或许也是相同的。

经验表明，行为排序能力，无论是在语言领域或是在绘画领域，都依靠中央模式机制。另一方面，我们所发现的有关原动控制语言方面的其它可能性可以归纳为其它认识活动。在手写体情况下，输出命令被表示为笔划。***Wright*** 发现，不同的生产过程可能会

被相同的高水平书写表现形式所控制。这种层次结构以及原动运动的“实质”表述是由经验为根据来支持的。人类能连贯地用小写字母在笔记本上书写，或者在黑板上用大的多的字体书写。此外，人们在用象手和脚这样的效应基因时，也能够进行连贯的书写。

如同在任何行为研究中一样，结果行为受书写体一般特性和单个书写者所具有的具体特性的影响。这就是说，大部分的变化可以归咎于个体的不同。许多研究者已经注意到手写体形式极具可鉴别性，以至书写者可以根据其笔迹被识别出来。这也是一个普通的知识，因此签字被当做对特定书写者唯一的鉴别工具。一些理论甚至将个人性格与其书写风格连系起来。正如我们准备在此篇文章中要说明的一样，对于单个书写者来说书写的原始方式是唯一的。

如同任何其它认识或原动活动一样，人类的原动控制经过了与智力发展相同的过程。一个大家熟悉而很有趣的现象就是儿童的图画(接着是写字)经过一个发展过程而逐渐趋于精美。许多研究人员提出，儿童早期绘画行为与其认知能力有关。***Vansommers*** 提出绘画可由高水平规则来支配，如同支配语言过程的规则一样，而且，绘画的发展过程与语言的发展过程的平行的。***Goodnow*** 和 ***Levine*** 甚至提出：一种“行为语法：儿童抄写的次序和句法”。他们在报告中阐述了几种绘画笔划次序规则。下面是有关这种规则的几个例子：“从最左端开始”，“从上端开始”，“ 用垂直笔划开始”，“从左至右划横线”等等。这种规则的进化基本原理可以简化原动计划。

一种手写方式生物方面的合理性包括两部分：假定神精控制

的合理性及人手生物力学方面的合理性。使用不同肌肉甚至器官时的书写方式储存是最能引起人们兴趣的问题之一。书写的自动性提出了一种数量机制，但是这种数量机制是不可能存在于人手的原动系统中的，而是存在于大脑的高层控制水平的某个地方。因此，当我们指"手"时，我们是含有隐喻的，也就是说，"手"代表的是完成原动控制的输出机制。最近，*Alexander* 等人提出有关"原动程序"的特殊概念是否就是人脑如何控制活动的生物合理方式的适当的发展基础呢？当我们所掌握的有关表层和基本神精原动区方面的知识与允许某种特殊方式相距甚远时，它可以提出什么方式比其它方式更具有神精方面的合理性。*Fischbach* 将他有关"表面细胞"和"原动命令细胞"的发现论述为人脑抽象化的根据。在猴子的视觉系统中，处于下层颞颥沟槽的"表面细胞"被当做描述一个高水平的抽象化而提出的。这些神经只对表面有反应而对其它视觉刺激则没有反应。表面细胞在原动方面有着与其极相似的东西。已验明"命令"神经存在于某种能引起固定行为方式的脊椎动物身上。*Georgopolos* 记录了单个神经的电子活动，在猴子将前肢活动方向转变成密码的原动皮层(脑回）中发现了命令神经。这些神经的刺激与特定的肌肉或相应运动的力量没有关系。*Geogopoulos* 通过总结许多神经的激发频率计算出一个向量，并发现它与运动方向的关系比与任何单个细胞的活动更为密切。在胳膊运动前几毫秒内向量的变化很明显。他将这种结果解释为原动神经方案的明显迹象。*Damasio* 研究了左后颞颥以及下层对称皮层受损病人的语言行为。发现这种病人在由现成的音素造词时会出现问题。通过对积累起来的对语言结构试验性发现的分析，加上

如同 ***MRI***(磁共振成象)以及 ***PET***(正级电子层面 ***X*** 线照相术)这样成象技术的辅助，表明了语言活动包括了原动皮层的激活作用以及左脑前部和后部语言中心。书写是一种语言活动，它包括构成单词和激活处于原动皮层的"命令细胞"写成笔划，字母和书面文字的制作中心。讲话也由一小套音素构成，同样我们则证明书写字母由一小套笔划构成。

除神精学方面的合理性之外，在人"手"部分还存在生物力学的强制因素。一些基本原则已被提出来支配这种控制机制。例如，***Flash*** 和 ***Hogan*** 提出的人类趋向用将肌肉反射减到最小程度的方式书写。也就是说，方位信号的第三次派生。一个更新的研究提出所谓"***Snap***"，这是方位的第四次派生，如同降到最小限度的成本作用。因为我们将发现施加于手写体上生物力学方面的强制力类型具有选择性假设。尽管支配手写体的原则可能是普遍的，每个书写者都有其自身独一无二的变化。未写出笔划(即，笔在动，但不接触书写平面)中的不同之处比起写出的笔划更为明显。笔在书写平面上的摩擦消弱了手的控制过程的特点，而这一点在笔被提起时会更为明显的表现出来。

文章是以回顾"原动程序"理论开始的，并且为反对这种词语以及它所意味的内容而展开争辩。我们将提出一个选择性的基本书写方式的连接模式并且说明它更具有生物和认识上的合理性。接着，我们将描述有关试验，数据的收集以及从这一分析中所得出的结论。最后一章将讨论有关结果，将我们的结论与其它进行比较。我们将以指出一些未来的方向和所提模式所包含的意义来结束我们这篇文章。

2. 注意力，数量和“原动程序”

“工作内存”的概念是从计算机的“工作内存”提法中模仿而来的，计算机中央处理系统(*CPU*)中的记存器具有相类似的作用。这就是为什么这一领域的研究人员喜欢谈论“原动程序”的原因。“原动程序”是用来储存资源的。根据这一方法，假定大脑通过执行“原动程序”来控制如同书写这类的活动，很像计算机中使用的软件。“原动程序”这一概念很具吸引力，它通过使用有限的普通原动命令(或路径)的预先程序指令序列控制大量的活动，这样就减少连续分析方法的复杂性。***Alexander*** 等人指出意味“硬件”与“软件”分离的“原动程序”的神经学方面合理性的种种困难。例如，在这种方式中软件由什么组成，当不使用时储存在什么地方，在使用前它们是如何组装的以及新程序是怎样编出的。关于“原动程序”方法的主要问题还包括执行的连续：有目的的运动假定被调到轨道中，之后又到关节运动。只有当相反的动力被计算出来后，肌肉的激活作用才能被计算出。因此，***Alexander*** 等人论证说，在皮质和基部的神经节里应该发现这种特化作用的信号。迄今为止，神经学方面的迹象似乎表明缺乏这种特化作用。

假定处理能力是有限的，因此，几项必须同时进行的工作在相同条件下展开竞赛。人类和其它生物体的主要问题或许是我们缺少相同的输出途径。所有的输出途径，无论是讲话，书写或是其它原动输出本质上都是连续的。或许这种连续的输出也意味着一个连续的认知过程。就是这种有关神经网的文献和这篇文章的假设，基本过程是相同的并且分布到上百万简单的处理单元(神经)中。因此，意味着运行在人脑计算机中连续符号过程的所谓“原动

程序”说法或许是一种误导。我们喜欢谈论“原动图解”，它们是被一个神经集合的激活作用引起的原动形态。

图解的联系性观点，即储存的知识原子在推理时被动态地组合到上下联贯的传感图解之中。***Rumelhart*** 和 ***Mcclelland*** 提出了一种说明有选择性的机制如何工作的技能。他们建议使用一种制图系统，这个系统可动态地产生程序化的连接并可在不同的时间集中于不同的特征上。***Smolensky*** 则坚持认为图解是连贯的知识原子组合，在此连贯和一致性在协调的名义下形成。他提出了协调原则：认知系统是原子连贯组合的激活过程，并且得出与被激活原子所代表的知识相一致的推理。

同样，我们计划谈论“原动图解”。这与我们的假设相一致，在所谓的“认知”和“原动”大脑机制之间没有本质的区别。连贯的图解模式与神经上的迹象也一致。即是说，不同皮质原动区中的特化与某些运动的顺序相关，而不与“原动程序”文献中提出的转化相关。根据我们的假设，预备系统和活动执行系统将属于相同的图解范畴。这是由结构上的事实证明的，即，三个原动区（***SMA***—补充原动区，***PMC***—主要原动 ***CORTEX*** 以及 ***PUTAMEN***），有着相同比例的目的从属原动细胞和肢体从属相关活动细胞。另一个有利的迹象是代表计算不同阶段的神经总数（根据“原动程序”观点）已经被说明是同时活动的。

3. 试验

3.1 数据收集

Rumelhart 开发了一种在书写者操作下能辨认草体笔迹的系

统。这个系统从许多书写者写做出的草体笔划的例子中进行模仿并且将其记录下来。他从每 58 名书写者中各挑选出大约 1000 个单词。这些词的平均长度约为 8 个字符。总共达到近 500,000 个手写草体的例样。他的试验结果很鼓舞人心并且被应用到该项试验中。由于 ***Rumelhart*** 主要对书写体识别感兴趣，这篇文章使用相同的数据来对书写过程进行调查。

这些数据是用以下方式收集的。每个词都被记录下来。之后放给按指示将该词写在数转换板上的书写者。每十毫秒 ***X*** 轴，***Y*** 轴结果以及有关笔是否接触在纸上的显示被提取一次。数据作为文件被储存起来，并且已被用于该篇文章中所提出的分析之中。

除 ***Rumelhart*** 的试验中的数据之外，几千个日文书写体笔划被收集起来。多数数据是从平假名手写体中收集而来，而一些数据则是在写 ***Kanji***(个人签名的)日文时收集来的。平假名具有英文手印刷体写出字的曲线形状，但是没有草写体的连字弧线。

为了抽取特征，手写原始资料已经过预先处理，目的是使所抽选出的特征用来区分和使笔划赋有特征性。一个笔划被规定为笔垂直运动速率的两个连续零位交叉点之间草写体符号的一部分。每个字被分为几部分或几笔划。英文最典型的书写速度是每秒两个字母。写日文的平假名需要大约相同的时间，一个典型的平假名能够在 0.3—0.5 秒的时间内被写出来。

3.2 字段和特征的抽取

字段及特征的抽取原则是将连续的信号分隔为不连续的部分并且用特征空间中的特征向量代表每一部分。

这个从连续信号中产生出笔划的所谓部分，取决于“笔划”这

个词的不同定义。尽管多数关于有线字体认知的文献都使用这个词，但是至今仍没有一个对“笔划”一词统一共认的定义。例如，人们经常发现只有笔的状态按唯一的规则变化。也就是说，笔划的定义即连续性的笔的上下运动。

一旦规定了“笔划”，就有许多用特征空间表示它的方式。有线字体认知研究使用了几种垂直的变换方式，如，与笔划对应的曲线部分的不连续的 ***Fourier*** 变换。即，一个笔划用从它的 $X(t)$ 和 $Y(t)$ 信号中得出的 ***Fourier*** 参数表示。本质上讲，任何垂直变换（如，***Walse*** 变换，***Karhune—Loeve***）在近似的笔划曲线中都能进行。即，水平曲线能近似于垂直函数（正弦曲线，多项式或甚至正方形曲线）。这种表述能被轻易地转变到频率领域，这曾在几次书写认知的研究中进行过。

当字体能够被少数参数表示时，这种将时间表示为参数领域的作法是有助的。因此，循环的平滑曲线通过协调作用产生更好的造型，因为它需要较少的参数。另一方面，直线型笔划则要求高级调性，因为它们包括高频分量。这就是为什么正弦曲线的近似值对于包含曲线型笔划的字体有用的原因，在英文中所发现的曲线型笔迹多于日文的 ***Kanji*** 字体（它们主要是由直线部分所组成）***K—L*** 展开式已经被证明是机械打印汉字认知中的一种成功运算方法。另一种成功试验是用经修改的 ***Hough*** 变换式来认知手写体汉字。***Hough*** 变换式是用于这类检测的一种方法并且已经被广泛用于任意形状。汉字是线型文字，因此它们很自然地对应 ***Hough*** 变换表示法。

当然，分割和特征抽取方式取决于目标。如果目标为形式认

知，那么，这种分割和特征抽取则应与区别各类形状相一致。我们寻找的是具有生物学角度合理性的分割和特征抽取。因此，我们仅调查那些可以被神经控制结构解释的特征，如，笔划的走向，它们的弯曲率等等。

3.3 ***Hollerbach*** 模式

所使用的分段和特征抽取过程是开发一种根本的书写过程模式并且以这种模式的参数方式对数据进行描述。所使用的方式来自 ***Hollerbach*** 模式，并且假设形成过程能用一对连结的振荡器来描述。这一对动作协调的振荡器吸是现有许多模式中的一种。实际上，关于这种速度测验图的匀称形状的基本假设也许过于简单。关于笔划的速度测验图的文献中通常呈现出的是一种不对称的铃状图。也就是说，由对数法线速度测验图所表示的以快速为目的的运动被认为是基本的单位(笔划)。更复杂的运动是以附加对数法线的方式来表示的。铃状速度测验图的这种不对称性来自于速度控制中大量处理过程的综合的随机行为。

尽管它是一种过于简化而不精确的模式，但是它具有很明显的优势，这就是它是以控制机制为基础的，并且从神经生物学角度是可以理解的。这种方式表现为：

$$\dot{x} = a\cos(\omega_x t + \phi) + c \qquad (1)$$

$$\dot{y} = b\cos(\omega_y t) \qquad (2)$$

总之，概念很简单，写字包括两个垂直的摇摆运动。如果我们针对在笔记本上用小写母写字，我们能够想到手腕的垂直运动(实

际上，更多的是弧形运动)以及手指的弯曲及水平延伸运动。这两个运动可以被认为是独立的。如果字母的大小超过一英寸，那么胳膊的肌肉就得参与进去。

根据这种模式，*Y* 轴包含一系列速度测验图呈正弦曲线的上 / 下笔划。*X* 轴也是摇摆的，并且右侧有一个连续的速度 *C*。通过调整相应的振幅 *a* 和 *b*，相应的方位，*phi* 以及 *X* 和 *Y* 轴方向相应有频率 *Wx* 和 *Wy* 来产生不同的字。因此，可以进一步假设只有当 *Y* 轴上的速度为零时(笔划的尾端)，参数才变化。这样，我们将在 *Y* 轴速度零交叉之间的笔划定义为 *Vy*。另外，当笔划状态变化时(从下至上或相反)，分段即可发生。

必须强调的是，*Hollerbach* 的模式是由第二命令机械系统为综合书写字体而设计的。这种模式并不试图模仿人类的原动控制或者用于人类书写的分析。然而，因为它是一种控制系统模式，一些参数可以按照人类生物力学系统转换。例如，代表位置变化的参数 φ，当表示神经肌肉控制系统的延迟时可以被转换。这样，它就能够对原动疾症具有一种重要的诊断价值。

当它被 *Rumelhart* 用来作书写分析时，它经受了一些倒退。其中之一是，这种模式不适合于这种图像，而适于笔划的速度测验图。由于速度信号 *Vy* 的阶段性，这种简化在多数英文曲线型书写情况下往往很奏效。因为我们所使用的 *Hollerbach* 模式主要基于速度信号，我们将从手写字体的 *X*—*Y* 领域到其相对应的速度描述所发生的变化。手写字母 *d*，对应的速度测验图以及重复制做的 *d* 的例子分别见图 1,2,3,和 4。

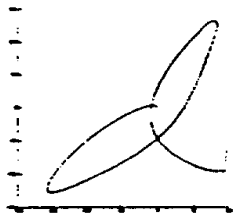


图 1:书写体字母 ***d***

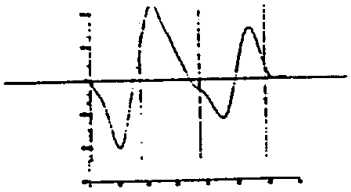


图 2:书写体字母 ***d*** 的 ***Vx*** 图

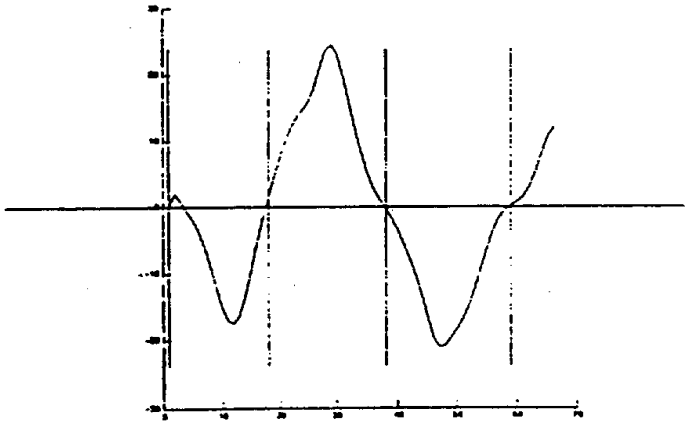


图 3:书写体字母 ***d*** 的 ***Vy*** 图

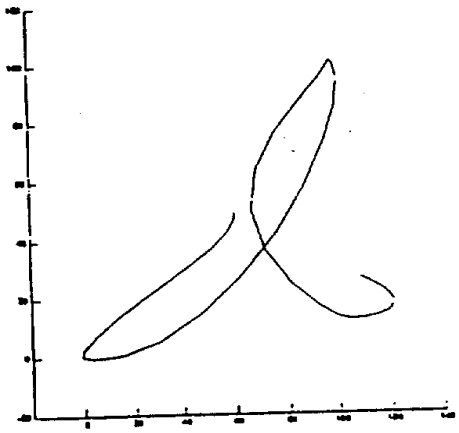


图 4：重构的字母 ***d***

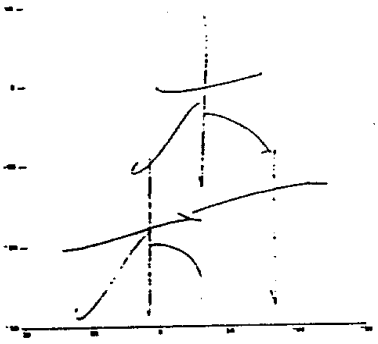


图 5：在日文 ***kanji*** 字中，分开的笔划更加明显。

这是 ***Kanji*** 字“***mori***”，意为：森林

y	x	中	速度	协调	向下的笔划
0.70	7.41	0.74	-2.95	4	1
3.44	-3.86	-2.77	-2.20	4	0
-10.56	-0.45	-0.01		2	1
0.71	-0.13	0.04	-0.33	2	1
4.40	0.08	-0.59	2.62	2	0
-4.20	-4.94	-1.88	-3.19	2	1
3.87	4.61	1.55	5.46	2	0
-2.67	5.51	3.63	-2.33	2	1
-5.74	-15.08	-7.31	-4.36	2	0
1.64	9.70	2.62	8.60	2	1
0.42	-1.46	0.23	-4.40	2	1
3.27	-1.99	-1.85	0.75	2	0
-9.71	-0.19	-0.04	-0.20	2	1
5.45	0.47	-1.20	3.70	4	0
-5.40	-5.85	-3.39	-2.74	4	1
4.29	4.89	0.84	10.50	2	0
-1.91	4.29	3.38	-1.56	2	1
3.91	-0.64	1.55	-3.30	4	0
1.82	9.72	3.70	6.74	2	1
0.07	-1.35	0.46	-5.25	2	1
2.00	-2.17	-2.12	1.86	2	0
-9.60	-0.15	-0.15	-0.28	4	1
0.98	-0.37	-0.09	-0.74	2	1
4.52	0.57	-0.33	-0.12	4	0
-4.22	-5.45	-2.72	-2.64	4	1
4.14	6.24	4.40	5.90	4	0
-5.10	7.23	4.61	0.54	4	1

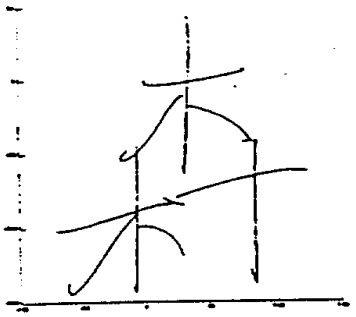


图 6：*Kanji* 字“*mori*”的 *Vy* 信号

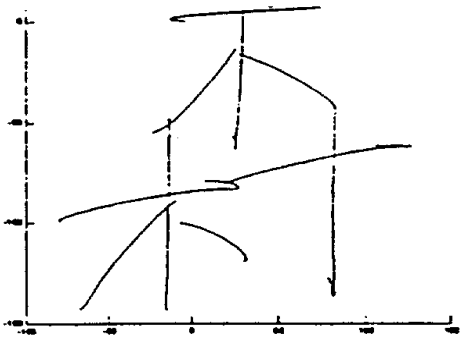


图 7：重构的 *Kanji* 字“*mori*”

可以看出，重新制做出的 *d* 不完美，字母 *d* 的第一笔的弯曲部分与原来的是相反的。这一结果说明这样一个事实，即，这种模式企图重新复制。

可以看出，重构并不完美，并且这个 *d* 的第一笔与原来的是相对的。这个结果说明这种方式试图重构速率，但却不是笔划所做出图像的效果。

这并不包括正弦曲线的近似值对 *Rumelhart* 曲线草写体的认

知奏效的事实。这说明在一些情况下(在写英文曲线的线段时),这种模式是相当成功的。

而另一方面,***Kanji*** 字体具有更多的短直线部分,这一点可以从下面的图中看出:

图中 ***Kanji*** 字,"***mori***"被分为 27 划(第三个"木"中最后两个向下的笔划缺了)。27 划中有 16 划的笔接触到纸,有 11 划只是将笔从线的一端移向另一端。向下笔划的 12 划对应该字中可看见的线段。

3.4 辨认笔划次序

辨认曲线型书写的关键问题是分段问题。***Rumelhart*** 已经发明了一种曲线书写辨认的学习演段方法,包括组合词的辨认和字母的辨认。字母辨认是以 ***PMP'S*** 辨认和 ***PMP'S*** 顺序构成字母辨认为基础的。这个系统同时包括学习辨认以及分隔字母。

尽管 ***Rumrlhart*** 的试验是为书写体辨认而进行的,但是从中仍能学到几招儿,包括 ***PMP'S*** 及其书写时的顺序。早在 60 年代初,就有人认识到原动知识能用来进行书写体辨认。一个称作综合分析的系统提出字体由其"原动程序"来进行辨认。通过推测原始程序并且根据综合及实际形式之间的不同反复使其更新的方式,这些"原动程序"会从想象上被推测出来。读与写过程之间的联系通过某种后天书写及诵读困难的相互发生而得以确证。与早些的探讨相对照,***Edelman*** 等人提出如果朗读者在朗读时使用原动知识,那么他们在用脑重复书写过程时似乎并非如此。当然,借助明确的模仿及含蓄的知识之间那些明显的区别,我们提出的联贯模式并不伤脑筋。这是"原动程序"这种隐喻说法所造成的误导影响的

另一个例子。

4. 集合的结果

4.1 固定范围的集合

集合的基本单位是笔划，每个笔划被表示为 N 范围内的一个点。每个笔划中我们选出的六个特征中的三个被采用。首先，对于这种模式我们只用一次，于是较协调的少数笔划就被移去。第二，我们不去区分上下笔划。向上的笔划包括较多高级协调，但是我们将我们的分析限制在基本的运动上，并且尽量忽略由生物机械控制机制所引起的振荡。未被采用的第三个特征是中间点。对于在空间领域的笔划的重构，每一笔划中点的 X 轴被计算出。然而，我们最初的分析表明这种变量与 X 变量有很大的关联。这种初步的分析产生出三个几乎不相关的变量：Δx，Δy 和速率。这些空间的尺寸为：

1. Δy—垂直方向相应的位移。

2. Δx—水平方向相应的位移。

3. V—V 笔划末尾的速率 Vx 根据以下公式计算出。

对于近似于振荡频率的 a 与 b 之间的笔划，我们这样计算：

$$I_x = \int_a^b dx \cos(2\omega_x t) \tag{3}$$

$$v = [\frac{x_a - x_b}{l}] - [\frac{2I_x}{l}] \tag{4}$$

(5)

对于不同的 ***Wt*** 计算方式也不同，这是这种计算的一个例子。

从许多书写者那里得到的数据集合并没有产生令人满意的效果，但是个人书写者的集合则令人满意。例如，显示 13 个包括 90—14 紧凑的集合的特定书写者的集合与我们分析的所有书写者是一致的。这就证实了我们的推测，即，书写来自少数 ***PMPs***，它们对于个人书写者来说是唯一的。

这些集合的心迹线从特征空间被重构，并且在一个 2 维空间领域显示出来。从结果中可以清楚看到，不同的书写者具有不同的笔划类型：

因为每个书写者有大约 25,000 我们想收集的的笔划，我们从一种近似于 ***K*** —法算法的快速集合算法开始。对这种集合算法的主要要求是它能够处理大量的数据并且在几次(2—3)重复后找到满意的集合。另一要求，更为重要，就是这些心迹线将能很好的替代每组集合中的观测资料。这个要求导致对于不超出预定范围紧凑的超球形集合的寻找。那些集合通过一种等级集合算法在后一阶段才出现。

这个集合运用了一个双阶段方案。首先，一个“最近的心迹线分类”算法被采用以大的数据组展示集合。然后，集合心迹线的结果已经被转移到不同的等级集合方法中。第一阶段算法对局外人的笔划十分敏感，他们的笔划形成了独立的集合。这就是为什么我们得到许多小集合的原因。这些集合的数目小于 10 个观测资料。它们被当做杂乱的例外的笔划，并且已经被排开以免影响大集合的心迹线的表现。

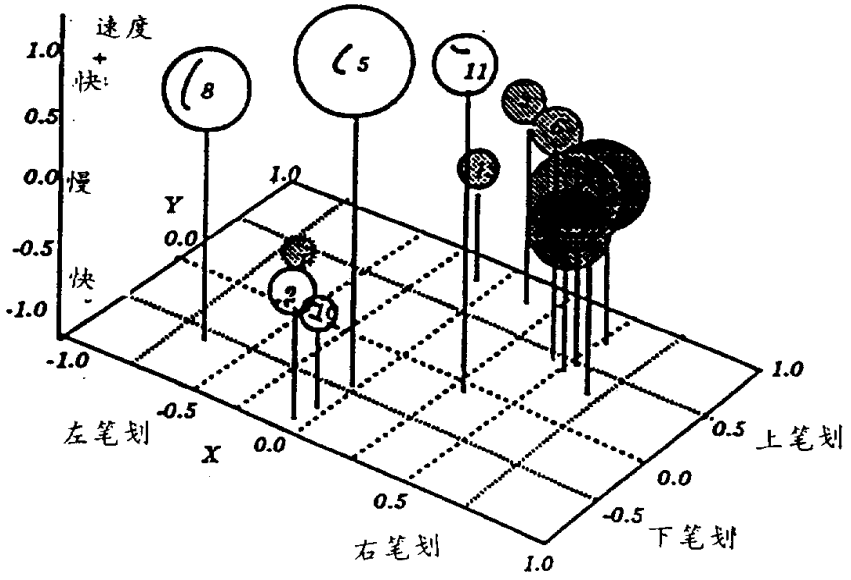


图 8：同一个书写者 25,000 笔划的集合。

灰色集合表示向下笔划。

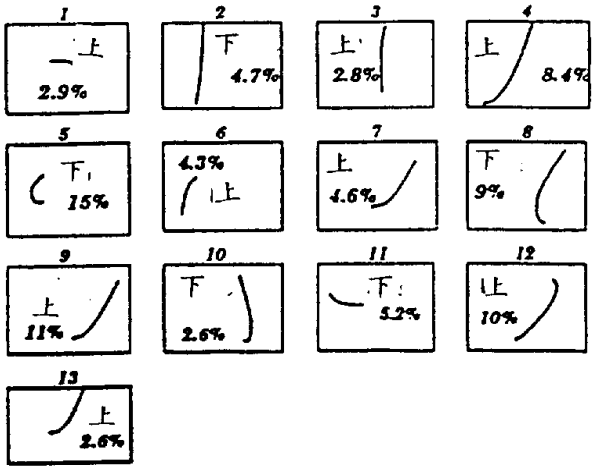


图 9：单个书写者的 13 心迹线笔划，包括其相应的频率。

第二阶段包括用十种不同方式计算的心迹线的结果的集合。我

们区别了这些产生紧凑超范围的集合的方法，以及那些能够验测延长的集合。我们从第一组的八个集合方法开始：

1. 平均联系集合分析
2. 心迹线等级集合分析
3. 完全联系集合分析
4. 相同变量最大值可能性方法
5. 可变数据集合分析
6. ***McQuitty*** 的相似性分析
7. 中等集合分析
8. ***Ward*** 的最小变量集合分析

不同的方法趋向支持如尺寸，形状或分散体等不同的特征。例如，以 ***K*** —法或 ***Ward*** 的最小变量方法这样的最小方形标准为基础的方法倾向寻找带有大体与每个集合数目相同的集合。平均联系集合则倾向寻找相等变量的集合。多数集合算法，除了个别联系和密度联系之外，倾向制做出紧凑的，粗略超范围的集合。这种如单个联系的以非参数密度估算集合方式为基础的集合方式奖在下一章讨论。

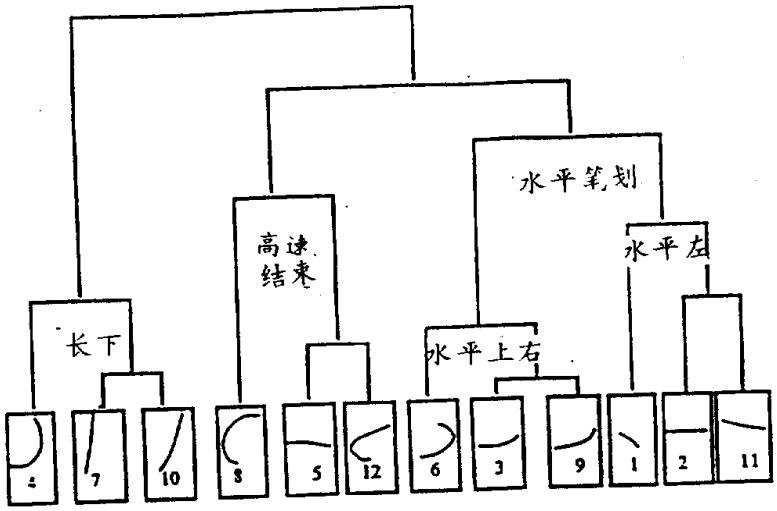


图 10:一个特定书写者的 12 个笔划心迹线的等级集合

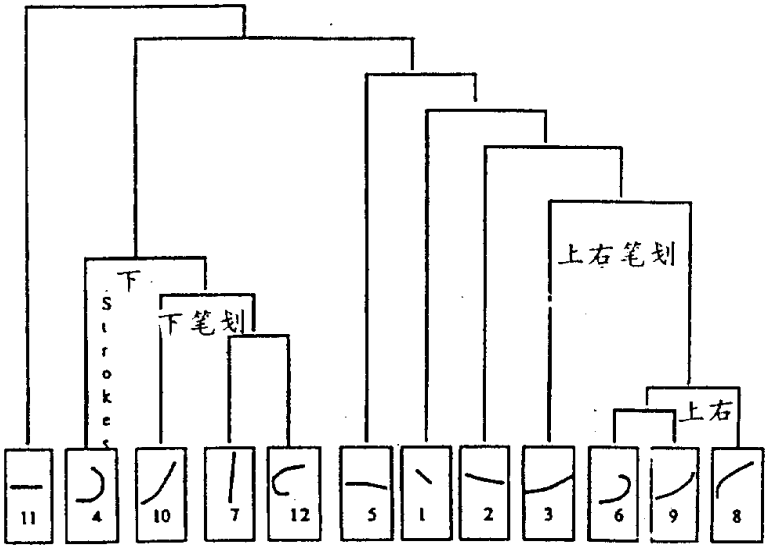


图 11:同一书写者的 12 个笔划的等级(密度联系)集合

所有上述集合方式产生非常相似的结果，并且基本部分从本质上也相同。许多不同的算法已经被用来调查不同等级集合方式下集合结构的健全性。所有上述方法的结果展现出下列基本的 12 个笔划(一名特定的书写者的)的基本部分。

从这个等级集合的结果中看，有一个很明显的超集合出现。水平的左向笔划在一个组，长的向下的笔划在另一个组。总之，我们

看到了水平笔划和垂直笔划的区别。水平方向的笔划又被进一步分为水平左向笔划，水平右向笔划和水平向上笔划。高速率的*C*状笔划是园形或椭园形的一部分。应当注意，对于一个特别的书写者来说，一种笔划总是以同样的方式来写出。例如，一个短横笔划，如*a t*连线，总是以左向笔划来完成的。另一些人只用水平右向笔划来写。但是，同一个书写者将用水平左向和水平右向笔划写是非常不可能的。对于垂直向长笔划来说，情况也是查同的。一旦书写者使用长的垂直向下笔划，他将总是把垂直线写成同样形状的向下笔划和速率图。这种笔划结构在我们以上所提到的所有等级集合算法中都是一致的。

使用非参数密度估算的集合方法，如“密度联系集合分析”，能够检测延长的集合形状。这些集合方法产生两个明显的超集合：即，向下的及长的笔划“和”向上的右向的笔划。向下的笔划就是英文字中构成主干的部分，而向上的右向笔划是那些典型的用作连字弧线的笔划。

4.1.1 笔划的“特征”形状

正如以上所讨论的，任何一个书写者都具有一套特别的表现书写者特征的笔划。虽然，同一个书写者具有相同的笔划，在书写不同语言时，一个特定的笔划出现的频率当然取决于这种语言。为了表示特定书写者的特征，就其笔划而言，我们提出“笔划顺序示意图”(*POD*)。该示意图在下面的图中表示出来。

图 12:一个书写者英文曲线书写的心迹线。

笔划根据其 *Vy* 值执行自上而下的命令。

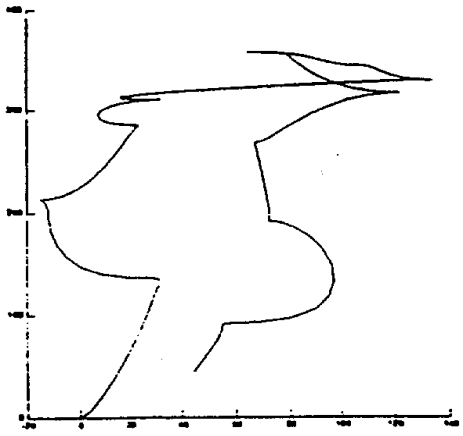


图 13:一位日本书写者写日文平假名时笔划的心迹线。

笔划根据其 *Vy* 值执行自上而下的命令。

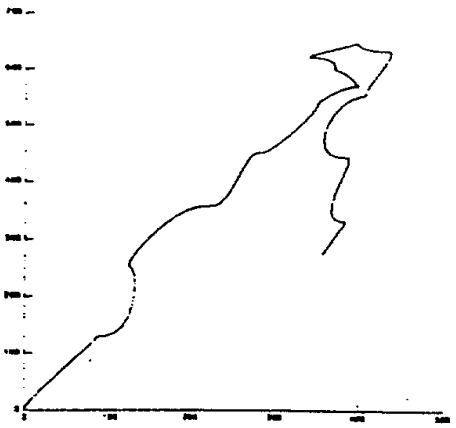


图 14:一个日本书写者写英文时笔划的心迹线。

笔划根据其 *Vy* 值执行自上而下的命令。

尽管它们看上很怪，那些示意图却十分有价值，并且反映了有关被分析书写者的笔迹的重要信息。

这两个书写者所使用的关键笔划之间有很明显的区别。这一点对其他书写者也一样。每个书写者使用一套独特的笔划：不同的斜率，不同的曲线，不同的速度轮廓图及不同的加速度。

5. 讨论以及未来的研究

我们将从比较 ***Rumelhart*** 书写辨认试验的结论和该项研究的结论来开始讨论。在 ***Rumelhart*** 的书写辨认试验中，从属的和独立的书写辨认者都得到过训练。两个网络系统已受到训练来辨认单个书写者的书写，一个网络系统已经被作为“从属书写者”辨认者以4位书写者进行训练。在从属书写者网络系统中 ***Rumelhart*** 发现对于1000字的词汇，在训练中从未见过的词有99个超过5分，大约90分。在独立的书写者数据方面，结果比较糟。大约70分。

根据这个研究结果，我们有了对这个结论置疑的依据。书写者间的变化性很大，而且较多的书写者未必会产生出更好的结果。

Rumelhart 的另一个发现是书写者可以很容易地被训练，从而写出可辨认的手迹。他开发了一种“在线”系统，在该系统中，网络系统在书写者在数字转换器上书写时开始辨认。如果正确地将独立书写者系统进行分类，对书写者稍加注意，就不难达到90分以上的好分数。对人适应辨认器的能力的仔细的试验还没有展开。通过将词汇限制在100或更少的方式，结果近乎可能接近完善（当然，这取决于词的混淆性）。

主要的结论是，将辨认器嵌入对书写者子集合进行训练的网络系统中的每一网络中会很有用处，或许一个用于打印机，一个

用于纯曲线型书写者等等。这种思路引出了目前的研究，即在此篇中所论述的研究。也就是说，研究书写者中个体的不同是很有用的。这种研究单个书写者不同的思路，作为一种书写识别的更好方法，最终开辟了一种新的研究领域，即，书写者独特笔划研究，这一研究与手脑交流中的无意识性主题有关。

这项研究从那个观点开始。我们提出的主要问题是单个书写者是否具有一套始终如一的明显的笔迹。基于这种观点的数据分析认为情况确实如此。人类书写者具有12—14种明显的笔划，这对于某个书写者来说都是具有特征的。这些笔划是最原始的原动方式，书写体就是由此而组成的。我们也从原始的笔划集合到高级集合进行了说明，从而揭示出控制机制的等级性质。这些发现与神经学文献的观点是一致的，这一点我们在简介中已经说明。也就是说，可能存在一种"命令细胞"，它们能够从大脑中的另一中心激活某一单词以及笔划机制。这个笔划由一个方位和振荡幅度所控制，它激活相应的原始原动方式(*PMPs*)。下一阶段将进入负责人脑原动领域激活过程的细胞领域。这项研究在磁共振成像方法(*MRI*)的支持，正在进行之中。我们将期待学习写字的过程将表明它是在形成这种原动激活中心，这与我们在研究中所发现的相对应。

这些发现对于无意识性及量的研究很有意义。须调查清楚的一个问题就是这种原动控制机制是否像前面研究者所提出的那样是为重要的论据和一种方式。这一点可以通过对方式间干扰的方法的研究来进行调查。例如，做一个试验，在试验中，试验主体按指示将一个字发出音来，而且同时将另一个字写出来。除了预料

的较长的反应时间之外，我们现在还预料了笔划方式与次序之间的干扰。另一个有趣的问题是原动方式是如何被储存的，并且在须要的时候又是如何出现的。我们的推测与神经学派方式是一致的，即，重现时间与次顺方式是无关的。一些对这种推测的支持是写规格大的字(***Kanji***)和写规格小的语言(平假名，英文)字体所用的时间是相同的。

今后的研究会将行为学分析与神经学研究结合起来，并可以对我们在介绍中所提出的许多问题作出回答。

附录 B

```
#include <conio.h>
#include <stdio.h>
#include <signal.h>
#include <math.h>
#include <stdlib.h>
#include <graphics.h>
#include <dos.h>
#include <string.h>
#include <values.h>
#include <iostream.h>
#include <alloc.h>
#include <time.h>
#include <ctype.h>
#include "q_lib.h"
#include "start.h"

#define DMAX 400
#define PERIOD_CONST 10

#define T 0.01
#define NUMBER_CHANNEL 5
#define del 20

pos_st POS_ST[DMAX];

FILE *point,*fp;

int pen_up=0;

int init_board (void);
extern unsigned _stklen=0x8000;
int read_acc (char *filename);
void ini_msg(void);
void next_point(void);
void read_smooth_point(int number);
void read_averag_point(float xyz[],long num);
void get_acc(void);

#ifdef __cplusplus
  typedef void (*fptr)(int);
#else
  typedef void (*fptr)();
#endif
unsigned int shift_cnt=0;
int Catcher(int *reglist)
{
  printf("Caught it!\n"); /* make return AX = 3 */
  return 0;
}
```

```
/* First filteration by Butterworth digital filter 4'th order and
   0.1 cutoff freq..
   Input : array of accel. POS_ST and index of point ind1.
   Output : smoothed values in array POS_ST */
void butter_filt4(pos_st POS_ST[],int ind1)
{
 static float a[5]={1.,-3.180639,3.861194,-2.112155,.438265};
 static float b[5]={.000416599,.001666397,.002499595,.001666397,.000416599};
 static float y[4][5],x[4][5];
 int i,j;
 float sum=0.;
 static int cnt=0;
 for(j=0;j<4;j++)
  {
    if ( cnt<5 )
       y[j][4-cnt] = POS_ST[ind1].AR[j];
    for(i=4;i>0;i--)
       x[j][i] = x[j][i-1];
    x[j][0] = POS_ST[ind1].AR[j];
    if ( cnt >= 5 )
      {
        sum = 0;
        for(i=4;i>0;i--)
           y[j][i] = y[j][i-1];
        for(i=0;i<5;i++)
           sum += b[i]*x[j][i];
        for(i=1;i<5;i++)
           sum -= a[i]*y[j][i];
        y[j][0] = sum;
        POS_ST[ind1].AR[j] = sum;
      }
  }
 if ( cnt < 5 ) cnt++;
}
```

```
float DIF[2]={0,0},DIF0[2]={0,0};
/* Second filteration by Butterworth digital filter 4'th order and
   0.02 cutoff freq.
   Input : X and Y accel.
   Output : smoothed values in global variable DIF */
void butter_filt0(float dif[2],int fl)
{
      s     t     a     t     i     c                 d     o     u     b     l     e
a[5]={1.e+00,-3.83582554064735e+00,5.52081913662223e+00,-3.53353521946302
e+00,8.48555999266478e-01};
      s     t     a     t     i     c                 d     o     u     b     l     e
b[5]={8.98486146372335e-07,3.59394458504525e-06,5.39091688001037e-06,3.59
394458371298e-06,8.98486146816424e-07};
 static float y[2][5],x[2][5];
 int i,j;
 float sum=0.;
 static int cnt=0;
  if (cnt == 5 && fl)
     cnt=0;
  for(j=0;j<2;j++)
   {
    if ( cnt<5 )
      {
       for(i=0;i<5;i++)
         {
          y[j][i] = dif[j];
          x[j][i] = dif[j];
         }
      }
    for(i=4;i>0;i--)
       x[j][i] = x[j][i-1];
    x[j][0] = dif[j];
    if ( cnt >= 5 )
      {
         sum = 0;
         for(i=4;i>0;i--)
            y[j][i] = y[j][i-1];
         for(i=0;i<5;i++)
            sum += b[i]*x[j][i];
         for(i=1;i<5;i++)
            sum -= a[i]*y[j][i];
         y[j][0] = sum;
         DIF[j]= sum;
      }
    }
 if ( cnt < 5 ) cnt=5;
}
```

```
/* Function for compensation of second filter delay,
   make delay for outputs of first filter.
   Input:d1 and d2 current difference of signals in a pair
   Output: delayed difference signal */
void set_delay(float d1,float d2,float &res1,float &res2)
{static h_cnt=0,t_cnt=0;
 static float diff_acc[2][50];
 diff_acc[0][t_cnt]=d1;
 diff_acc[1][t_cnt]=d2;
 if ( (h_cnt-t_cnt+del)%del == del-1 )
    {
      res1 = diff_acc[0][h_cnt];
      res2 = diff_acc[1][h_cnt];
      h_cnt = (h_cnt+1) % del;
    }
 t_cnt = (t_cnt+1) % del;
}

struct send_data
      {int period; //T
       float ampl; //A
       float incr_acc; //V
       };
send_data params[2][DMAX];
/*      This procedure gets as inputs two signals (after the filteration) for each
        accelerometer and the pen status.  The procedure performs segmentation of
        the signals and calculation of each segment's features  */
void get_pack_param(send_data params[2][DMAX])
{static int fl_opt=0,cnt_seg[2]={0,0};
 static int cnt_t[2];
 static float p_dacc[2]={0,0},beg_dacc[2]={0,0};
 static float max_ampl[2]={0,0},beg_xyacc[2]={0,0};
 int i;
 float dacc[2];
 set_delay(POS_ST[1].AR[0]-PEN.R[0]-POS_ST[1].AR[1] + PEN.R[1],
           POS_ST[1].AR[2]-PEN.R[2]-POS_ST[1].AR[3] + PEN.R[3],
           dacc[0],dacc[1]);
  if (!pen_up)
   {
   for (i=0;i<2;i++)
    {
    if ( fl_opt )
      {
        if ( dacc[i]*p_dacc[i] <= 0. )
          {
            params[i][cnt_seg[i]].period=cnt_t[i];
            params[i][cnt_seg[i]].ampl=max_ampl[i];
            params[i][cnt_seg[i]].incr_acc=DIF[i]-beg_xyacc[i];
            cnt_seg[i]++;
```

```
            params[i][cnt_seg[i]].period= -1;
            p_dacc[i]=dacc[i];
            beg_dacc[i]=dacc[i];
            beg_xyacc[i]=DIF[i];
            max_ampl[i]=0;
            cnt_t[i]=0;
           }
          else
           {
            p_dacc[i]=dacc[i];
            if (fabs(max_ampl[i])<fabs(dacc[i]))
                 max_ampl[i] = dacc[i];
            cnt_t[i]++;
           }
      }

    if (!fl_opt && (fabs(dacc[0])>5. || fabs(dacc[1])>5.))
      {
        fl_opt = 1;
        cnt_t[0]= cnt_t[1] = 0;
        memcpy(p_dacc,dacc,sizeof(float)*2);
        memcpy(beg_dacc,dacc,sizeof(float)*2);
        memcpy(beg_xyacc,DIF,sizeof(float)*2);
      }
    }
   }
    else
     {
        params[0][++cnt_seg[0]].period= -1;
        params[1][++cnt_seg[1]].period= -1;
     }
}
```

附录 C

微控制器得到加速度表的信号，作为输入，在时域内对其分段（分段程序的列表示于附录 B），并用几个传输到接收机的参数代表每个段。

每个信号的分段利用振荡中心的移动和振荡幅度和频率执行。分段过程和特征抽取由以下步骤组成：

—将原始信号分为两个分量：振荡中心移动信号和振荡信号。每一个信号是利用 *Butterworth* 数字滤波器以第 4 阶和 0.02 截止频率对来自加速度表的加速信号进行滤波得到的，如“数字滤波器设计”(*T. W. Parks and C. S. Burrus*，*John Wiley&Sons*，1987，第七章，7.3.3 节)中所述。第二个信号是利用 *Butterworth* 数字滤波器以第 4 阶和 0.1 截止频率对一对两个加速度表的不同信号进行滤波得到的。

—根据振荡信号的零值对信号进行分段。

—特征抽取—用于限定每个段的特征为：

—段边缘之间的间隔(T)。

—振荡幅度(A)。

—段边缘第一信号值之间的差值(V)。

附录 *D*

Fax Pen（传真笔）

手动成象的笔划发生器

```
#include <stdio.h>
#include <alloc.h>
#include <math.h>
#include <string.h>
#include <graphics.h>
#include <process.h>
#include "st.h"
int far arr_x1[15000],arr_y1[15000];
void send_p ( unsigned , char * , unsigned  );//sending from
com-port
int init_com ( int  , unsigned  );
  void segment(int num_p,unsigned int far p_x[],unsigned int far
p_y[],char *p)
{
extern float xmax,ymax;
extern shiftx;
extern char *arg_str[];
    int  v_x[2][1000];
    int  v_y[2][1000];
    int  extrxy[2][1000];
    int attr=0;
// structure of binary file of data from graphic tablet
    struct point
    {
     unsigned x :13;
     unsigned y :13;
     int pen :6;
    } pnt;
// structure of binary file of coefficients of the cubic spline
    struct stroke
    {
     unsigned x :13;
     unsigned y :13;
     signed alf1:11;
     signed alf2:11;
    } strk;
   char buff[20];
    int j;
   int newp;
   int num_points;
   int m;
   float lastx,lasty;
   int lvx,lvy;
   FILE *fp1;
   int count,i;
   unsigned long size_file = 0 ;
   int fl;
   int lpoint,fpoint;
   unsigned  base;
   char *p1;
   char *exten={".str"};
   int ascii=0;
```

```
/* test for extension ".str".If true - it is the ASCII format *
 * for creating a stokes file                                  */
    strlwr(arg_str[2]);
    if ((strstr(arg_str[2],exten))!=NULL) ascii=1;

      p1=p;
      newp=num_p;
  for(j=0;j<15000;j++){
   arr_x1[j]=0;arr_y1[j]=0;
   }
   /*--------------------------------------*/
 /*
  *   writing binary file with data from graphic tablet   *
  *                                                       */
    if ((fp1 = fopen (arg_str[1] ,"wb"))!=NULL)
    {
      for (count=0;count<num_p ;count++)

      {
      int p;
     if (p_x[count]<shiftx)
      {
      pnt.x=(unsigned)p_x[count];
      p=1;
      }
      else
      {
      pnt.x=(unsigned)(p_x[count]-shiftx);
      p=0;
      }
     pnt.y=(unsigned)p_y[count];
     pnt.pen=p;
     fwrite (&pnt,sizeof(pnt),1,fp1);
       }
       fseek ( fp1 , 0L , SEEK_END ) ;
       size_file = ftell ( fp1 ) ;
       sprintf (buff,"%lu",size_file ) ;
       setcolor(13);
       settextstyle(0,0,1);
       outtextxy(168,40,buff);

       fclose(fp1) ;
    }
  /*--------------------------------------*/
  lpoint=fpoint=0;
  fl=1;
  while (num_p>lpoint)
{
  for (i=fpoint;p_x[i]>=shiftx && i<num_p;i++);
  attr=(unsigned)(p_x[i-1]-shiftx);
  for (j=i;p_x[j]<shiftx && j<num_p;j++)
   {
   arr_x1[j-i]=(float)p_x[j];
   arr_y1[j-i]=(float)p_y[j];
   }
   num_points=j-i;
   newp=num_points;
   fpoint=lpoint=j;
```

```
   /************************/
    if (j<num_p)
    {

/*                                                    *
 *             process segmentation                   *
 *                                                    */

    lastx=arr_x1[num_points-1];
    lasty=arr_y1[num_points-1];
    extrxy[0][0]=arr_x1[0];
    extrxy[1][0]=arr_y1[0];

  j=0;
  for(i=0;i<=newp-1;i++)
  { arr_y1[j]=arr_y1[i];
   arr_x1[j]=arr_x1[i];
     if (arr_y1[i]!=arr_y1[i+1])
     //&& arr_x1[i]!=arr_x1[i+1])
        j++;
        else
        num_points--;
  }
  newp=num_points;
/*-------------------------------------*/
    lvx=arr_x1[num_points-1]-arr_x1[num_points-2];
    lvy=arr_y1[num_points-1]-arr_y1[num_points-2];
    v_x[0][0]=(arr_x1[1]-arr_x1[0]);
    v_y[0][0]=(arr_y1[1]-arr_y1[0]);
    j=0;m=0;
/*-------------------------------------*/
  j=0;
  for(i=0;i<newp-1;i++)
  { arr_y1[j]=arr_y1[i];
   arr_x1[j]=arr_x1[i];
     if (arr_y1[i]!=arr_y1[i+1]
     && arr_x1[i]!=arr_x1[i+1])
        j++;
        else
        num_points--;
  }
  j=0;
  newp=num_points;
  for(i=0;i<newp-1;i++)
  {arr_y1[j]=arr_y1[i];
   arr_x1[j]=arr_x1[i];
     if (arr_y1[i]!=arr_y1[i+1]
     && arr_x1[i]!=arr_x1[i+1])
        j++;
        else
        num_points--;
  }
  j=0;
  newp=num_points;
  for(i=0;i<newp-1;i++)
  { arr_y1[j]=arr_y1[i];
   arr_x1[j]=arr_x1[i];
     if (arr_y1[i]!=arr_y1[i+1]
```

```
     && arr_x1[i]!=arr_x1[i+1])
        j++;
        else
        num_points--;
  }
/*****************************/


/*****************************/
  newp=num_points;
     j=1;
     m=0;
     setcolor(13);
     setfillstyle(1,9);

fillellipse((float)extrxy[0][0]*639/xmax,479-(float)extrxy[1][
0]*400/ymax-28,1,1);


   for(i=0;i<num_points-2;i++)
   {
   if (arr_y1[i]<arr_y1[i+1] && arr_y1[i+1]>arr_y1[i+2])
           {
           extrxy[0][j]=arr_x1[i+1];
           extrxy[1][j]=arr_y1[i+1];

fillellipse((float)extrxy[0][j]*639/xmax,479-(float)extrxy[1][
j]*400/ymax-28,1,1);
           v_x[0][j]=(arr_x1[i+2]-arr_x1[i+1]);
           v_y[0][j]=(arr_y1[i+2]-arr_y1[i+1]);
           v_x[1][j]=(arr_x1[i+1]-arr_x1[i]);
           v_y[1][j]=(arr_y1[i+1]-arr_y1[i]);
           j++;
           }
           else
   if (arr_y1[i]>arr_y1[i+1] && arr_y1[i+1]<arr_y1[i+2])
           {
           extrxy[0][j]=arr_x1[i+1];
           extrxy[1][j]=arr_y1[i+1];

fillellipse((float)extrxy[0][j]*639/xmax,479-(float)extrxy[1][
j]*400/ymax-28,1,1);
           v_x[0][j]=(arr_x1[i+2]-arr_x1[i+1]);
           v_y[0][j]=(arr_y1[i+2]-arr_y1[i+1]);
           v_x[1][j]=(arr_x1[i+1]-arr_x1[i]);
           v_y[1][j]=(arr_y1[i+1]-arr_y1[i]);
           j++;
           }
           else
   if (arr_x1[i]<arr_x1[i+1] && arr_x1[i+1]>arr_x1[i+2])
           {
           extrxy[0][j]=arr_x1[i+1];
           extrxy[1][j]=arr_y1[i+1];

fillellipse((float)extrxy[0][j]*639/xmax,479-(float)extrxy[1][
j]*400/ymax-28,1,1);
           v_x[0][j]=(arr_x1[i+2]-arr_x1[i+1]);
```

```
          v_y[0][j]=(arr_y1[i+2]-arr_y1[i+1]);
          v_x[1][j]=(arr_x1[i+1]-arr_x1[i]);
          v_y[1][j]=(arr_y1[i+1]-arr_y1[i]);
          j++;
          }
          else
  if (arr_x1[i]>arr_x1[i+1] && arr_x1[i+1]<arr_x1[i+2])
          {
          extrxy[0][j]=arr_x1[i+1];
          extrxy[1][j]=arr_y1[i+1];

fillellipse((float)extrxy[0][j]*639/xmax,479-(float)extrxy[1][
j]*400/ymax-28,1,1);
          v_x[0][j]=(arr_x1[i+2]-arr_x1[i+1]);
          v_y[0][j]=(arr_y1[i+2]-arr_y1[i+1]);
          v_x[1][j]=(arr_x1[i+1]-arr_x1[i]);
          v_y[1][j]=(arr_y1[i+1]-arr_y1[i]);
          j++;
          }

  }
          extrxy[0][j]=lastx;
          extrxy[1][j]=lasty;

fillellipse((float)extrxy[0][j]*639/xmax,479-(float)extrxy[1][
j]*400/ymax-28,1,1);
          v_x[1][j]=lvx;
          v_y[1][j]=lvy;
          j++;
/*-----------------------------------*/
/*                                                              *
* Calculating & writing a stroke file in binary or ASCII format *
 *
   */
 {  FILE *ff;
   if (fl)
   {
   ff =ascii ? fopen (arg_str[2],"w"):fopen (arg_str[2],"wb");
   if (ascii)
                        f     p     r     i     n     t     f
(ff,"{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{{
{{{{{{{{{{{{{{{{{{{{\n");
   fl=0;
   }
     else
   ff =ascii ? fopen (arg_str[2],"a"):fopen (arg_str[2],"ab");
  if(j==2    &&    extrxy[0][0]==extrxy[0][1]    &&
extrxy[1][0]==extrxy[1][1])
   {int temp;
   j--;
   temp=j+(attr<<9);
   memcpy(p,&temp,2);
   p+=2;
   if (ascii)
   fprintf (ff,"%6d\n",temp);
     else
   fwrite(&temp,sizeof(j),1,ff);
    strk.x=(unsigned)extrxy[0][0];
```

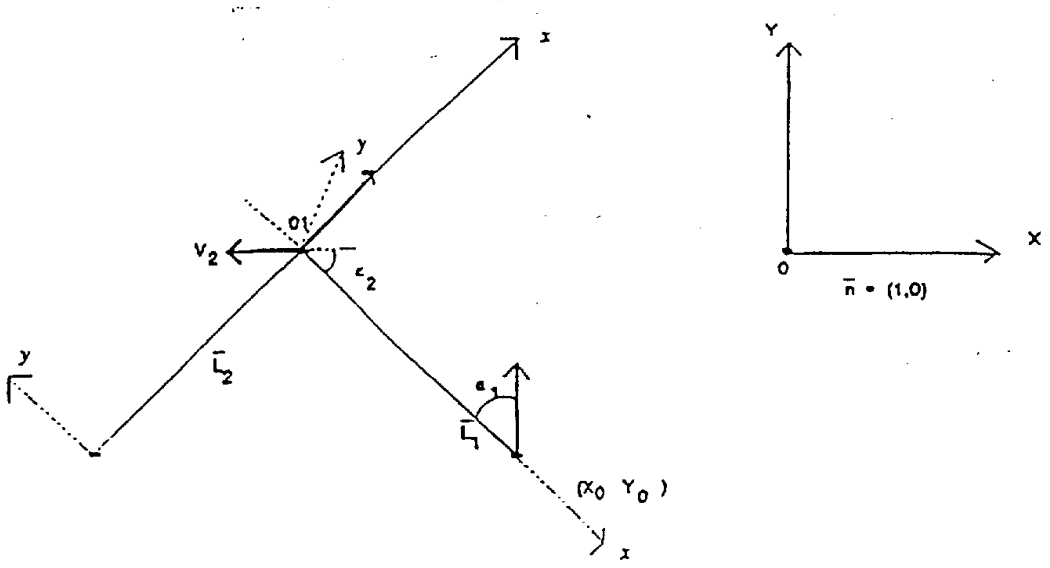
```
      strk.y=(unsigned)extrxy[1][0];
      strk.alf1=0;
      strk.alf2=0;
      memcpy(p,&strk,sizeof(strk));
      p+=sizeof(strk);
     if (ascii)
    f p r i n t f    ( f f , " % 4 d    % 4 d    % 6 d
%6d\n",extrxy[0][0],extrxy[1][0],0,0);
      else
    fwrite (&strk,sizeof(strk),1,ff);
     }
       else
     {int temp;
     temp=j+(attr<<9);
     memcpy(p,&temp,2);
     p+=2;
     if (ascii)
    fprintf (ff,"%6d\n",temp);
      else
    fwrite(&temp,sizeof(j),1,ff);
     for (m=0;m<j-1;m++)
      {float vx,vy,vx1,vy1,al,al1,al2,mods,modv,dlx,dly;
      dlx=extrxy[0][m]-extrxy[0][m+1];
      dly=extrxy[1][m]-extrxy[1][m+1];
      vx=v_x[0][m];
      vy=v_y[0][m];
      mods=sqrt(dlx*dlx+dly*dly);
      if(mods==0.0) mods=1;

      modv=(float)sqrt((vx*vx+vy*vy));
      if(modv==0.0) modv=1;
      al1=(dlx*vx+dly*vy);
      if(al1==0.0) al1=0.01;
      al2=(-dly*vx+dlx*vy);
      al1=al2/al1;
      /*-------*/
      vx1=v_x[1][m+1];
      vy1=v_y[1][m+1];
      modv=(float)sqrt(vx1*vx1+vy1*vy1);
      if(modv==0.0) modv=1;

      al=(dlx*vx1+dly*vy1);
      if(al==0.0) al=0.01;
      al2=(-dly*vx1+dlx*vy1);
      al=al2/al;
      al2=al;
      /*********/
      if (m==j-2) al2=0.0;
      if (m==0) al1=0.0;
      if (sign(al2)*al2<0.35 && sign(al1)*al1>1) al1/=2.0;
      if (sign(al1)*al1<0.35 && sign(al2)*al2>1) al2/=2.0;
      if (sign(al1)*al1>5.0) al1=sign(al1)*1.0;
      if (sign(al2)*al2>5.0) al2=sign(al2)*1.0;
      strk.x=(unsigned)extrxy[0][m];
      strk.y=(unsigned)extrxy[1][m];
      strk.alf1=(signed)(al1*1023.0/5.0);
      strk.alf2=(signed)(al2*1023.0/5.0);
      memcpy(p,&strk,sizeof(strk));
```

```
    p+=sizeof(strk);
    if (ascii)
   fprintf (ff,"%4d %4d %6d %6d\n",extrxy[0][m],extrxy[1][m],
            strk.alf1,strk.alf2);
     else
   fwrite (&strk,sizeof(strk),1,ff);
    }
    strk.x=(unsigned)extrxy[0][m];
    strk.y=(unsigned)extrxy[1][m];
    strk.alf1=0;
    strk.alf2=0;
    memcpy(p,&strk,sizeof(strk));
    p+=sizeof(strk);
    if (ascii)
   fprintf (ff,"%4d %4d %6d %6d\n",extrxy[0][m],extrxy[1][m],
            strk.alf1,strk.alf2);
     else
   fwrite (&strk,sizeof(strk),1,ff);
   }
  fclose (ff);
 }
}
}
      if(!(base=init_com(2,9600))) exit(-1);
      send_p( base , p1 , (unsigned)(p-p1) );
      if (ascii)
     {int leng;
      char ch[20]={""};
   fp1 = fopen (arg_str[2],"a");
   fseek ( fp1 , 0L , SEEK_END ) ;
   f        p        r        i        n        t        f
(fp1,")))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
))))))))))))))\n");
   fclose (fp1);
   strcpy(ch,arg_str[2]);
   leng=strlen(ch);
   strcpy(&ch[leng-3],"bst");
   fp1 = fopen (ch,"wb");
   fwrite(p1,(unsigned)(p-p1),1,fp1);
   fclose (fp1);
     }
fp1 = fopen (arg_str[2],"r");
      fseek ( fp1 , 0L , SEEK_END ) ;
      size_file = ftell ( fp1 ) ;
      sprintf (buff,"%lu",size_file ) ;
      setcolor(13);
      if (ascii)
   {
    outtextxy(580,40,buff);
    sprintf (buff,"%lu",p-p1 ) ;
    outtextxy(368,40,buff);
   }
      else
    outtextxy(368,40,buff);
fclose(fp1) ;
      free(p);
      free(p1);
}
```

3.1 坐标系的定义



oyx ———内部(局部)坐标系，描述笔划。

OYX ———外部(全局)坐标系，描述外部(如计算机屏幕)。

附录 E

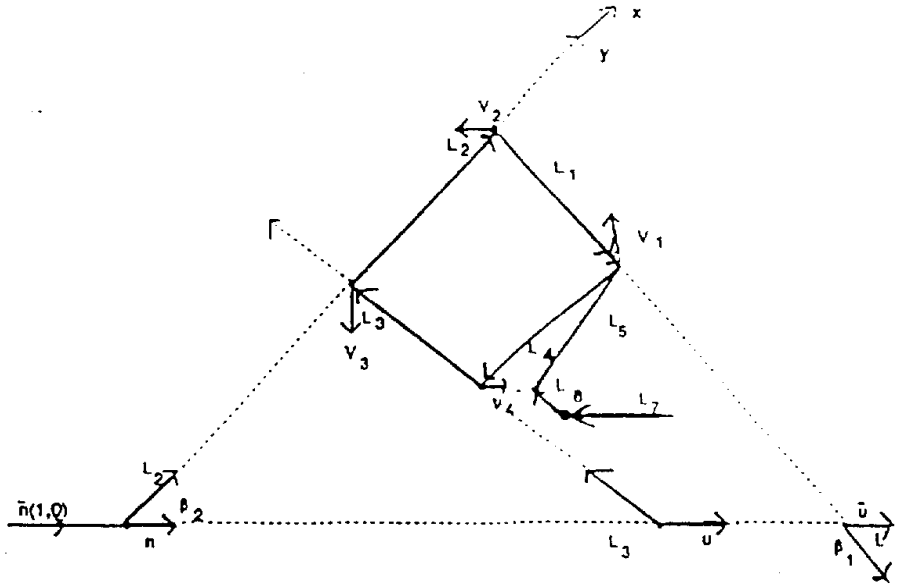
重构目标：

在局部坐标系中产生三阶(三次)样条序列，并从局部 oyx 坐标系转换为全局 OYX 坐标系。

限定符号分段的 OYX 向量

$$\bar{L}_1, \bar{L}_2, \bar{L}_m$$

的方向根据 OX 轴方向(局部坐标系)选择。



根据图 2，重构一个符号所需的信息包括：

$$\begin{pmatrix} \beta_1 \beta_2 \cdots\cdots \beta_m \\ \bar{L}_1 \bar{L}_2 \cdots\cdots \bar{L}_m \\ \bar{V}_1 \bar{V}_2 \cdots\cdots \bar{V}_m \end{pmatrix}$$

其中 $\beta_1 \ldots \beta_m$ 是相应向量

$$\overline{L}_1, \overline{L}_2, \ldots, \overline{L}_m$$

与向量 $\bar{n}$

之间的角度，

向量

$$\overline{V}_1, \overline{V}_2, \ldots, \overline{V}_m$$

是符号的“骨架”的向量速度(图 2 的例子中，字符为“a”)。

附录 *F*

Fax Pen（传真笔）

手动成象的重构

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <string.h>
#include <process.h>
#include <graphics.h>
#include "st.h"
float xmax=913.*5;
float ymax=594.*5;
int per=50;
float v_x[2][1000];
float v_y[2][1000];
int  dotted=0;
unsigned char attr=0;
int cols[5]={8,9,10,12,14};
int extrxy[2][1000];
//calculation of spline function with 2 derivatives and 2 points
float spl(float ,float ,float ,float ,float , float ,float  );
void main (int num_arg, char *arg_st[])
{
// structure of binary file of cubic spline coefficients
   struct stroke
   {
    unsigned x  :13;
    unsigned y  :13;
    signed alf1 :11;
    signed alf2 :11;
   } strk;
  int gdriver = DETECT, gmode, errorcode;
  int j=0;
  int m;
  int i;
  int x;
  FILE *ff;
   float seg,lseg;
   char *exten={".str"};
   int ascii=0;

/* test for file extension ".str".If true - it is the ASCII
format *
 * for creating stokes file                                    */
    strlwr(arg_st[1]);
    if ((strstr(arg_st[1],exten))!=NULL) ascii=1;

   if (num_arg<=1) {printf("Requires 1 parameter"); exit(1);}
        if ((ff=fopen(arg_st[1],"r"):fopen(arg_st[1],"rb"))==NULL)
   {printf("File not found"); exit(1);}

   registerfarbgidriver(EGAVGA_driver_far);
   registerfarbgifont(sansserif_font_far);
   initgraph(&gdriver, &gmode,"" );
   errorcode = graphresult();
   if (errorcode != grOk)
```

```
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1);
   }
    setbkcolor (15);
    setcolor(1);
    rectangle(0,0,639,479);
    line(0,35,639,35);
    setfillstyle(1,8);
    bar(1,1,638,34);
    settextstyle(3,0,4);
    setcolor(11);
    outtextxy(142,-5,"The Stroke Interpreter");
/************************************************/
   if (ascii)
                                   {       c       h       a       r
*str1="((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
((((((((((((((((((((((((";
      int res=1;
        while (res!=0)
   { char *str2;
    fscanf(ff,"%78s\n",str2);
    res=strcmp(str1,str2);
   }
    }
   while (1)
   {int temp;
   if (ascii)
   i=fscanf(ff,"%d\n",&temp);
   else
   i=fread(&temp,sizeof(j),1,ff);
    j=temp & 0x1FF;
    attr=temp>>9;
  if (i<=0)
  {
  fclose(ff);
  if (!getch()) getch();
  closegraph();
  exit(1);
  }
   i=0;
   {int vx,vy;
   while( i<j )
     {
   if (ascii)
     {
          f  s  c  a  n  f  (  f  f  ,  "  %  d     %  d     %  d
%d\n",&extrxy[0][i],&extrxy[1][i],&vx,&vy);
     v_x[0][i]=((float)vx*5.0/1023.0);
     v_x[1][i]=((float)vy*5.0/1023.0);
     }
   else
     {
     (fread(&strk,sizeof(strk),1,ff));
     extrxy[0][i]=(signed)strk.x;
     extrxy[1][i]=(signed)strk.y;
```

```
      /*                                                    *
       * preparation from integer to float    *
       *                  */
      v_x[0][i]=(strk.alf1*5.0/1023.0);
      v_x[1][i]=(strk.alf2*5.0/1023.0);
      }
      i++;
      }
    }
j=i;
/*-----------------------------------*/
  setcolor(cols[attr&0x07]);
  setlinestyle(0,0,2*((attr&0x08)>>3)+1);
  dotted=(attr&0x10)>>4;
  if(i==1)
     putpixel(extrxy[0][0],extrxy[1][0],cols[attr&7]);
     else
  {
  lseg=seg=0;

    //drawing splines

  for (m=0;m<j-1 ;m++)
    {float y,x1,y1,al1,al2,betc,bets,mods,dlx,dly;
    dlx=extrxy[0][m+1]-extrxy[0][m];
    dly=extrxy[1][m+1]-extrxy[1][m];
    al1=v_x[0][m];
    al2=v_x[1][m];
    mods=sqrt(dlx*dlx+dly*dly);

    x1=extrxy[0][m];
    y1=extrxy[1][m];
    /*------*/
    /*********************************
     *
     * cos and sin of angle between local and global system
     *
    *********************************/
    betc=dlx/mods;
    bets=-dly/mods;
    /*********/
    for(i=0;i<((int)mods+1);i+=2)
    {float x0,y0;
   x=(float)i;
    /*********************************
     *
     *    calculation of local spline function y=f(x) with
 parameters:
     *   (start x & y,finish x & y,angels in start & finish,
 argument x.
     *
    *********************************/
   y=spl(0.0,0.0,mods,0.0,al2,al1,mods-x);
    /*********************************
     *
     * rotation of local system to global system
     *
    *********************************/
```

```
    x0=betc*x-bets*y+(float)extrxy[0][m];
    y0=(-bets*x-betc*y+(float)extrxy[1][m]);
    if (dotted) {float dx,dy;
          dx=(float)(x0-x1);
          dy=(float)(y0-y1);
          dx*=dx;
          dy*=dy;
          seg=sqrt(dx+dy);
          lseg+=seg;
           }
    if (!dotted || (((int)lseg)%(2*per))<per)
      line((float)x0*639/xmax,479-(float)y0*443/ymax,
           (float)x1*639/xmax,479-(float)y1*443/ymax);
    x1=x0;y1=y0;
     }
          if (dotted) {float dx,dy;
          dx=(float)(extrxy[0][m+1]-x1);
          dy=(float)(extrxy[1][m+1]-y1);
          dx*=dx;
          dy*=dy;
          seg=sqrt(dx+dy);
          lseg+=seg;
          }
    if (!dotted || (((int)lseg)%(2*per))<per)
       line((float)x1*639/xmax,479-(float)y1*443/ymax,

(float)extrxy[0][m+1]*639/xmax,479-(float)extrxy[1][m+1]*443/y
max);
         }
   }
  i=i;
  }
/*-----------------------------------*/



  }
```

附录 G

样条重构：

阈值条件(边际条件)：

间隔[0,11]边缘处的样条导数，向量 $\bar{L}1$ 的 $\bar{l}1$ 模

$$\dot{y}(l_1)=tg\alpha_1$$
$$\dot{y}(0)=tg\alpha_2$$
$tg\alpha_1, tg\alpha_2$ are calculated as:
$$\cos\alpha_1=\frac{\bar{V}_1\bar{L}_1}{|\bar{L}_1||\bar{V}_1|};\quad \sin\alpha_1=\frac{|[\bar{V}_1\bar{L}_1]|}{|\bar{V}_1||\bar{L}_1|}$$
$$\cos\alpha_2=\frac{\bar{V}_2\bar{L}_1}{|\bar{V}_2||\bar{L}_1|};\quad \sin\alpha_2=\frac{|[\bar{V}_2\bar{L}_1]|}{|\bar{V}_2||\bar{L}_1|}$$
$$tg\alpha_1=\sin\alpha_1/\cos\alpha_1;\quad tg\alpha_2=\sin\alpha_2/\cos\alpha_2$$

其中[.]向量积定义为：

$$[\overline{VL}]=\begin{pmatrix} i & j & k \\ V_x & V_y & V_z \\ L_x & L_y & L_z \end{pmatrix}$$
where V_x, V_y, V_z U L_x, L_y, L_z

其中向量 $\bar{V}$ 和 $\bar{L}$ 根据区间[$O1$，$l1$]中样条的有效位数在轴 Ox、Oy、Oz 上的投影 $V_xV_yV_zUL_xL_yL_z$ 是：

$$y(0)=y(l1)=0$$

一个样条定义为：

$$y(x)=a_1x^3+a_2x^2+a_3x+a_4$$

其中 a_1,a_2,a_3,a_4 是边条件限定的系数。

等分系统相对不知道的值

$a_1,i=1+4$

或

$$\begin{gathered}y(0)=a_4=0\\ y(l_1)=a_1l_1^3+a_2l_1^2+a_3l_1+a_4=0\\ \dot{y}(l_1)=3a_1l_1^2+2a_2l_1+a_3=tg\alpha_1\\ \dot{y}(0)=a_3=tg\alpha_2\end{gathered}$$

$$\begin{pmatrix}C_1^3 & l_1^2\\ 3l_1^2 & 2l_1\end{pmatrix}\begin{pmatrix}a_1\\ a_2\end{pmatrix}=\begin{pmatrix}-tg\alpha_2l_1\\ tg\alpha_1-tg\alpha_2\end{pmatrix}$$

对于 $a_4=0$，$a_3=tg\alpha_2$

到全局坐标系的转换根据下式进行：

$$\begin{pmatrix}X\\ Y\end{pmatrix}=\begin{pmatrix}\cos\beta_1 & \sin\beta_1\\ -\sin\beta_1 & \cos\beta 1\end{pmatrix}\begin{pmatrix}x\\ y_1(x)\end{pmatrix}$$

图 3 给出不同字母重构的例子。

附录 *H*

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <dos.h>
#include <stdlib.h>
#include <graphics.h>
#include "q_lib.h"
#define DMAX 400

#define T 0.01

struct send_data
      {int period;
       float ampl;
       float incr_acc;
       };

int reconstruct(send_data par[2],int t_count[2],float prev_dif[2])
{
static float px=-10000,py=-10000;
float DIF[2];
int i;
float  acc[4];
float u1;
float u2;
float shift[2];
const float a11=1.071524;
const float a12=0.11965;
const float a21=0.075;
const float a22=-0.22502;
const float b11=13.333333;
const float b12=13.333333;
const float b21=6.666667;
const float b22=6.666667;

  {
   float dx;
   float dy;
   float x,y;
   setcolor(15);

 /* Restoration of oscillation signal */
    u1=par[0].ampl*sin(t_count[0]*M_PI/par[0].period);
    u2=par[1].ampl*sin(t_count[1]*M_PI/par[1].period);

 /* Restoration of the movement of the center of oscillations */
    DIF[0]=t_count[0]/par[0].period+prev_dif[0];
    DIF[1]=t_count[1]/par[1].period+prev_dif[1];
```

```
/* reconstruction of the position of the pen's tip */
   dx=(a11*u1+a12*u2);
   dy=(a21*u1+a22*u2);
   shift[0] = (b11*DIF[0]+b12*DIF[1]);
   shift[1] = (b21*DIF[0]+b22*DIF[1]);
   if (px > -10000 && py > -10000)
     {
// Addition of the movement of the center of oscillations to obtain XY coordinates.
  x=100+(dx+shift[0])*2;
   y=200+(dy+shift[1])*4;
       if (i)
         {
          line(px,py,x,y);
         }
       // Saving coordinates of previouse point
       px=x;
       py=y;
     }
    else
     {
       px=100+(dx+shift[0])*2;
       py=200+(dy+shift[1])*4;
     }
  }
 return 0;
 }
```

附录 *I*

重构过程分两阶段执行：

—恢复信号。

—重构笔尖的位置。

从笔所传输的数据恢复加速信号(每个段的 T、A、V)是根据下式执行的。

$$u_x(i,t) = u_{x0} + u_{x1} ;$$
$$u_{x0} = u_x(i-1, T_{xi-1}) + (V_{xi}/T_{xi}) * t ;$$
$$u_{x1} = A_{xi} * \mathrm{SIN}((PI/T_{xi}) * t) ;$$

$$u_y(i,t) = u_{y0} + u_{y1} ;$$
$$u_{y0} = u_y(i-1, T_{yi-1}) + (V_{yi}/T_{yi}) * t ;$$
$$u_{y1} = A_{yi} * \mathrm{SIN}((PI/T_{yi}) * t) ;$$

$$u_z(i,t) = u_{z0} + u_{z1} ;$$
$$u_{z0} = u_z(i-1, T_{zi-1}) + (V_{zi}/T_{zi}) * t ;$$
$$u_{z1} = A_{zi} * \mathrm{SIN}((PI/T_{zi}) * t) ;$$

u_x, y_y, u_z 是加速度的恢复的信号。

u_{x0}, u_{y0}, u_{z0} 是振荡中心的恢复的移动。

u_{x1}, u_{y1}, u_{z1} 是振荡的恢复的信号。

XY 平面内笔尖位置的重构是通过分解两个分量进行的：XY 平面(x_0, y_0)内振荡中心的移动和振荡移动(所写符号的外形)(x_1, y_1)。这些值的计算根据下式进行：

$$x_0 = a_{11} * u_{x0} + a_{12} * u_{y0}$$
$$y_0 = a_{21} * u_{x0} + a_{22} * u_{y0}$$

$$x_1 = b_{11} * u_{x1} + b_{12} * u_{y1}$$
$$y_1 = b_{21} * u_{x1} + b_{22} * u_{y1}$$

参数 a_{ij}, b_{ij} 因个人而变，在对话开始时作为书写者的个人手动成象特点而接收。

图.1

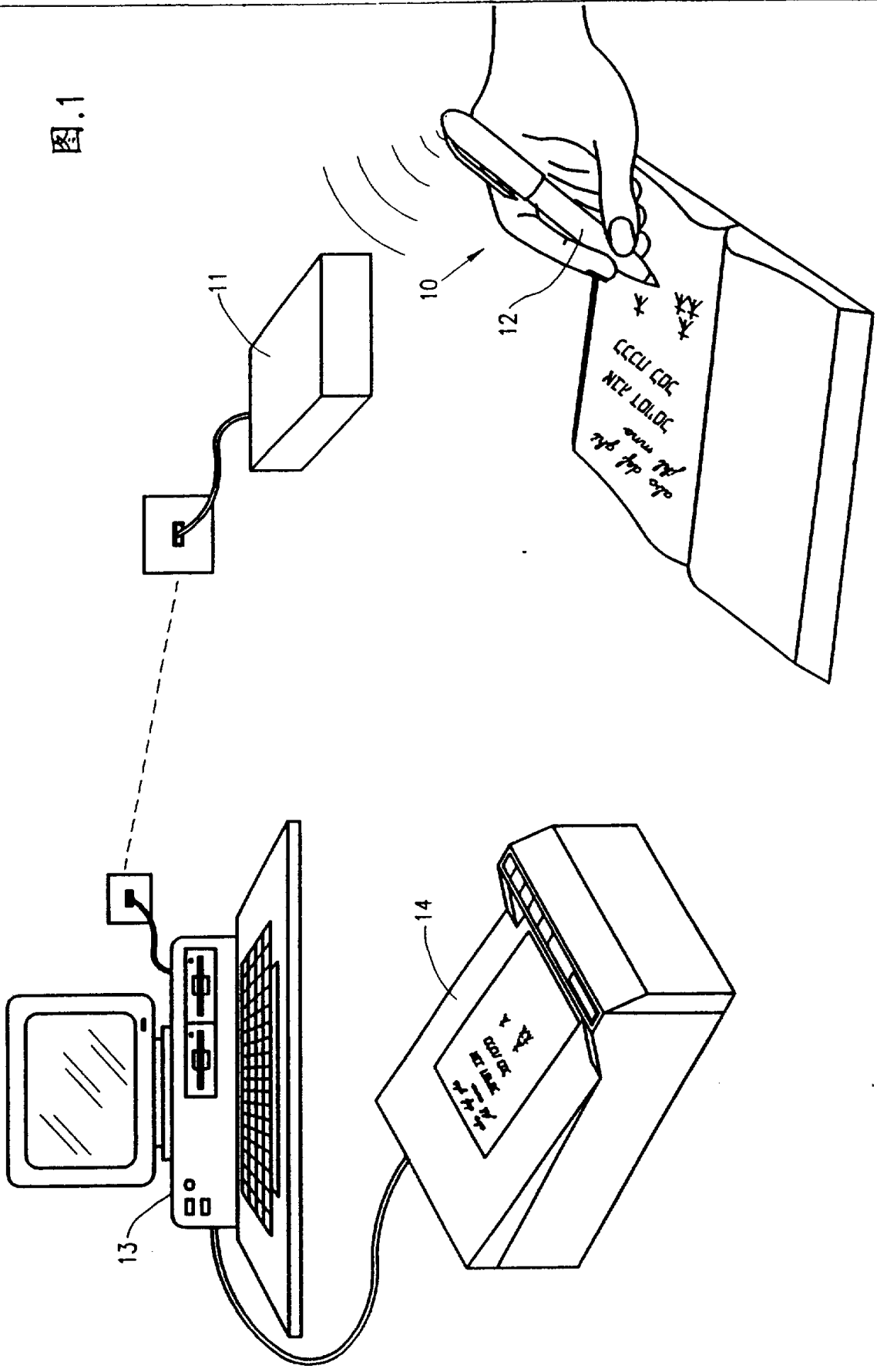


图.2

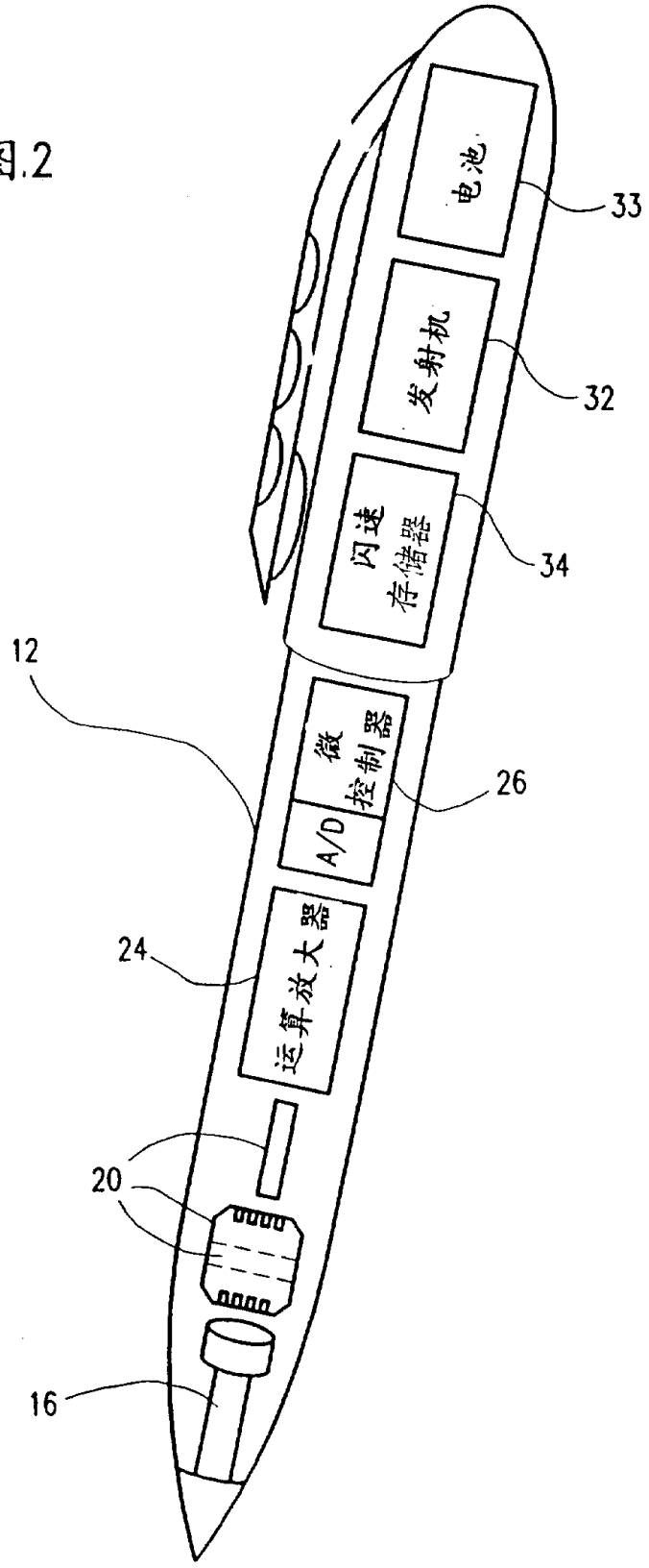
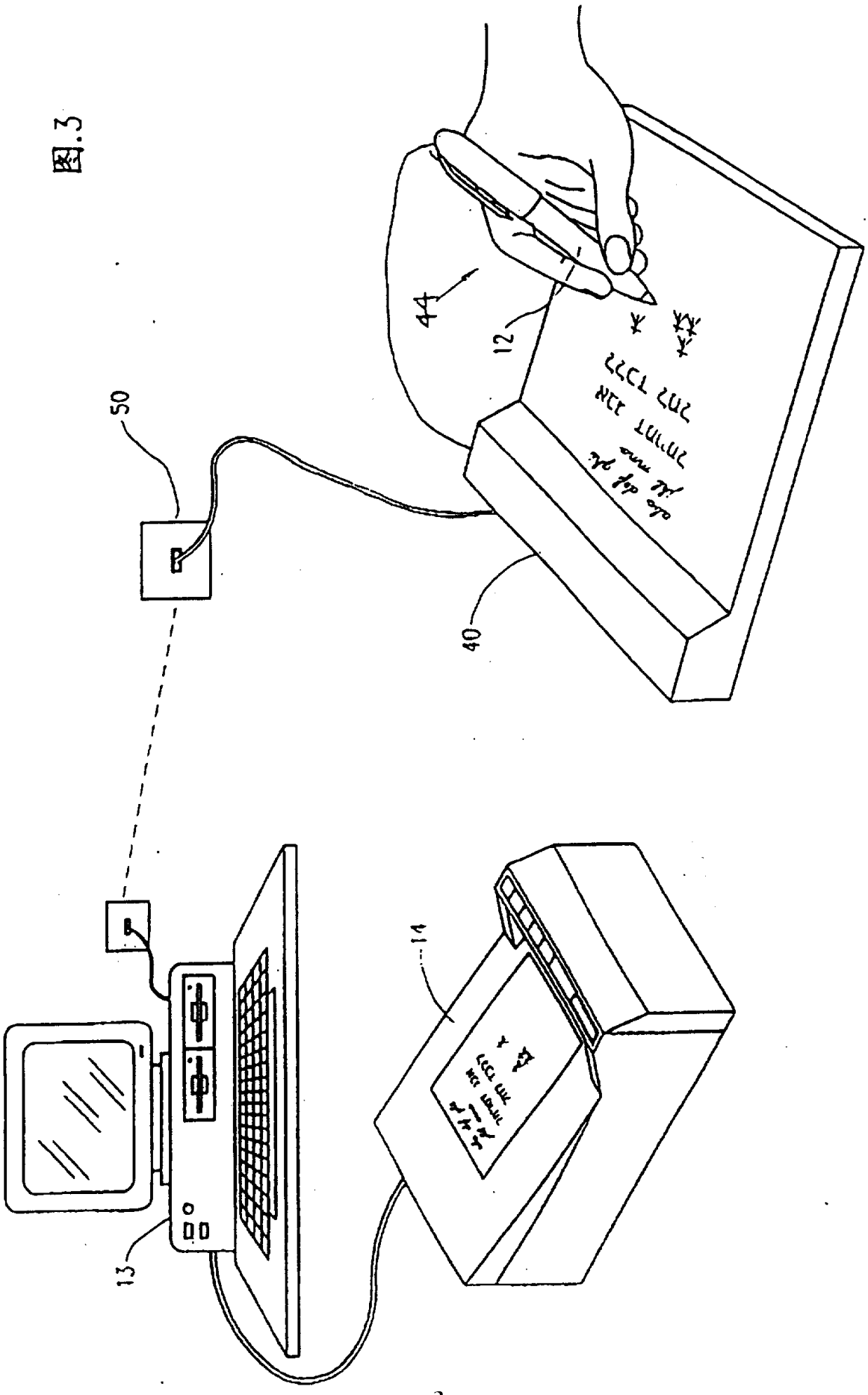


图.3



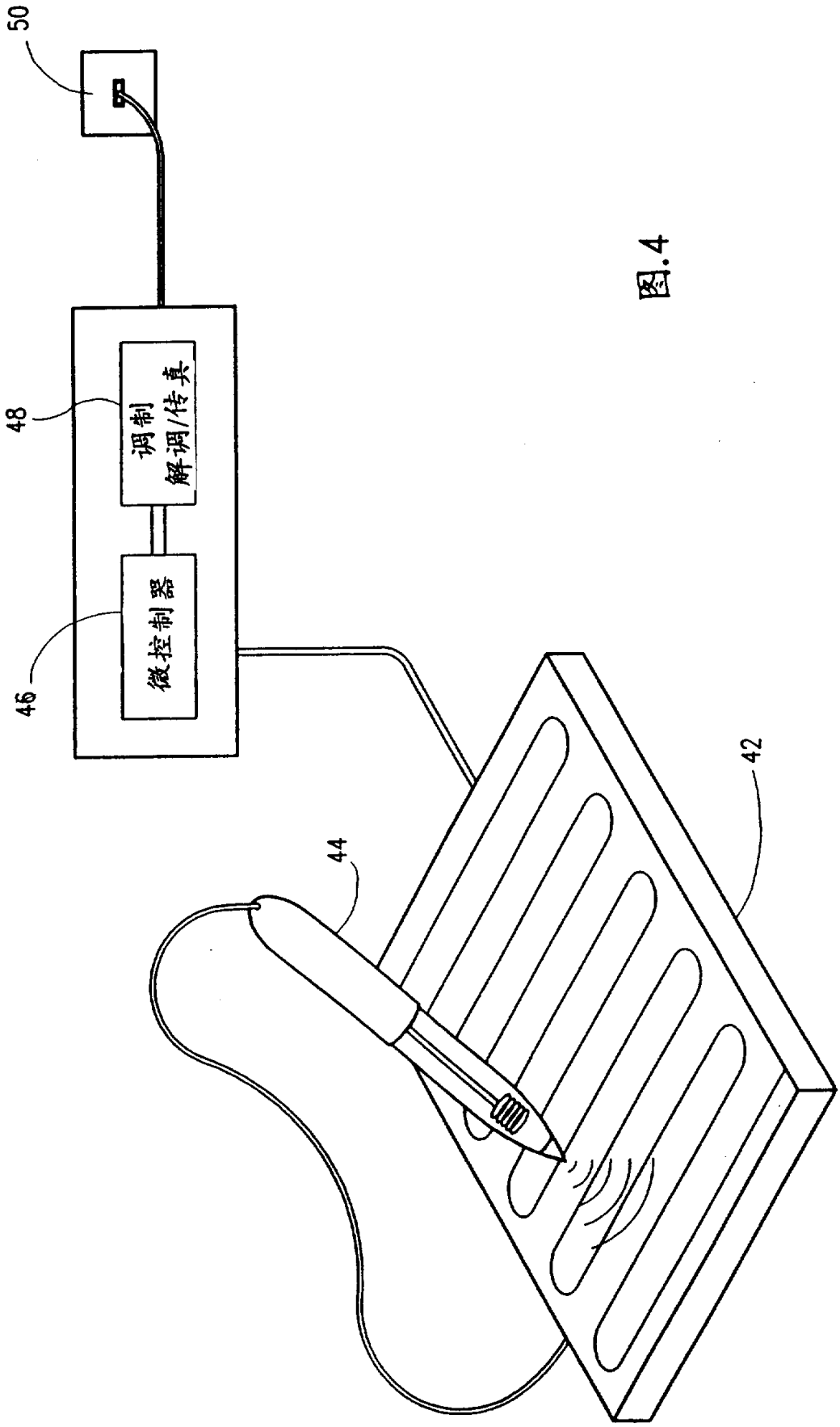


图.4

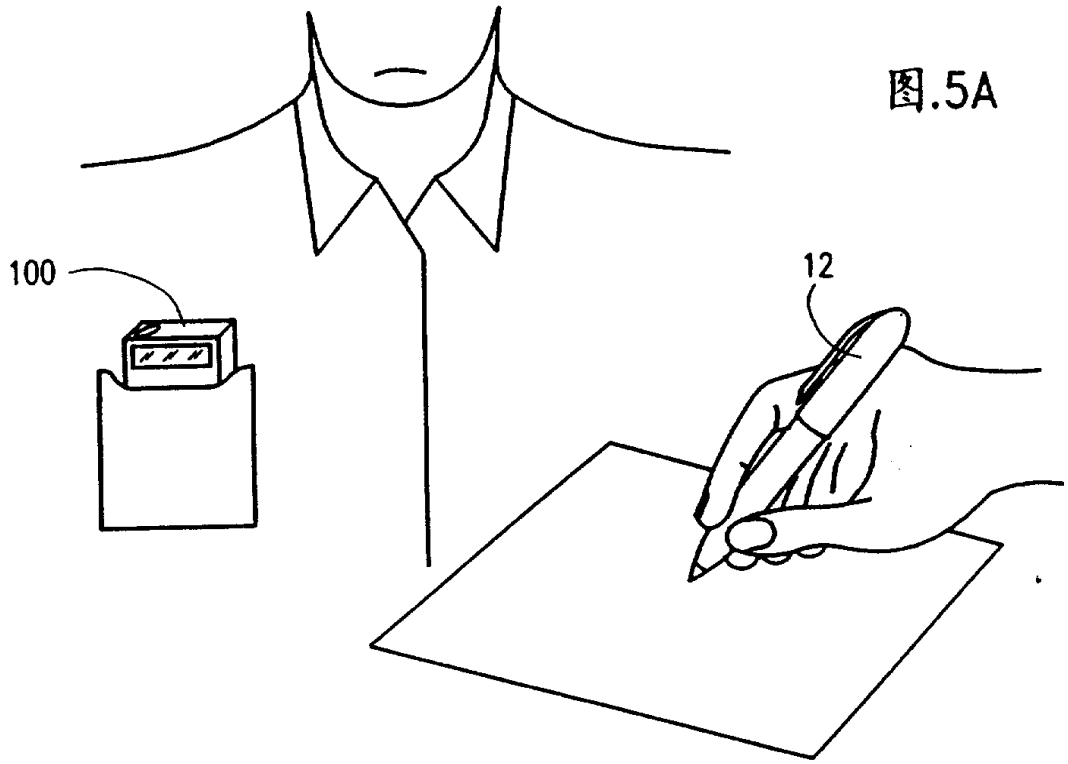


图.5A

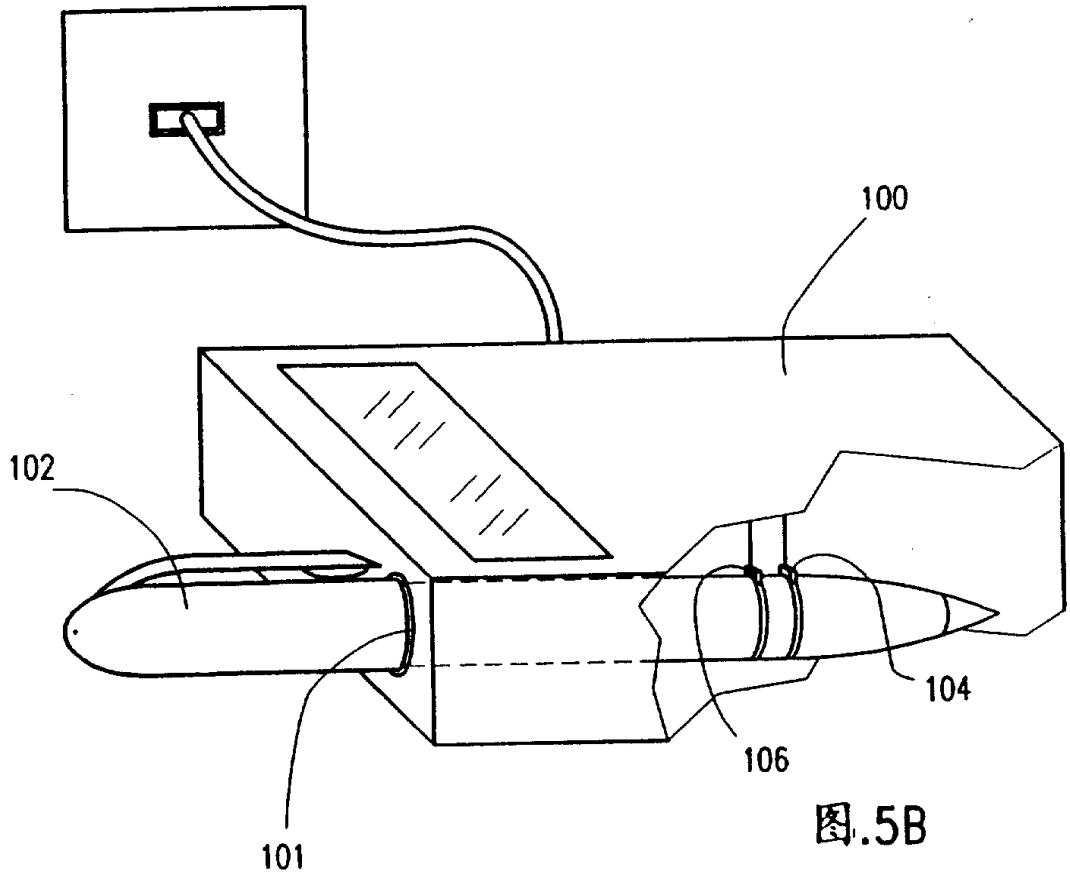


图.5B

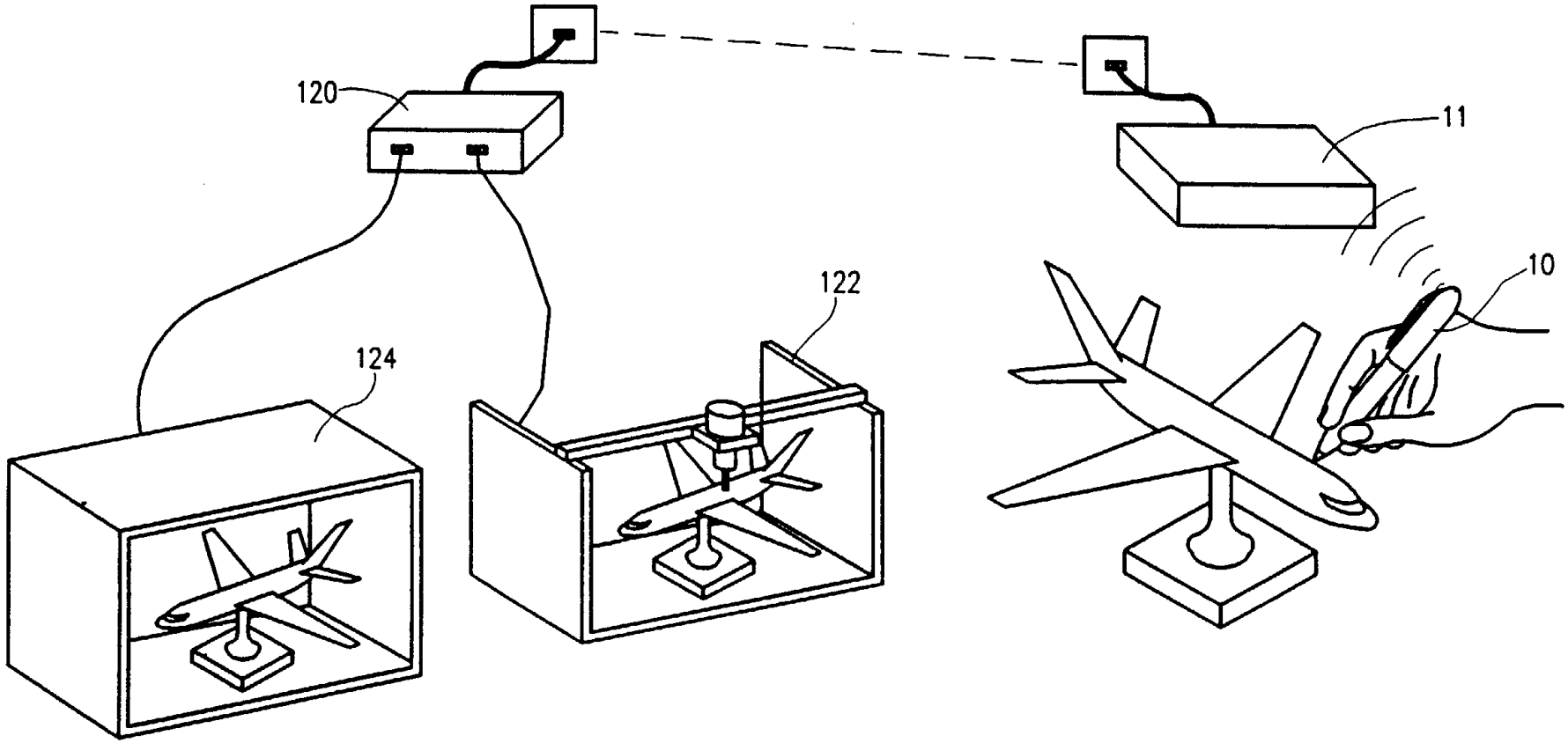


图 .6