



[12] 发明专利申请公开说明书

[21] 申请号 200510004179.9

[43] 公开日 2006年3月8日

[11] 公开号 CN 1744104A

[22] 申请日 2005.1.6

[21] 申请号 200510004179.9

[30] 优先权

[32] 2004. 1. 6 [33] US [31] 10/752,109

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 王 坚 王 强 马晓旭 李 越
林宙辰

[74] 专利代理机构 上海专利商标事务所有限公司
代理人 谢喜堂

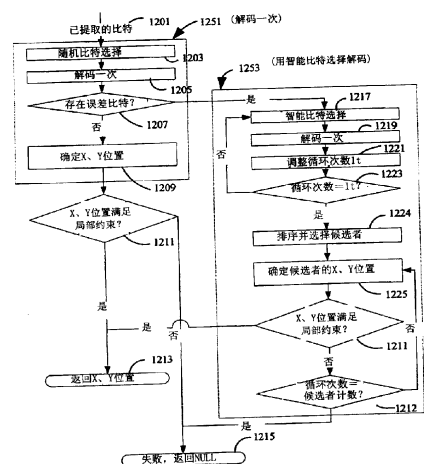
权利要求书 3 页 说明书 19 页 附图 13 页

[54] 发明名称

m - 数组解码和纠错的增强方法

[57] 摘要

描述了一种用于从较大的图像中确定所捕捉数组位置的过程和装置。一非重复序列可以被折叠成一非重复数组，其中，所述数组对于给定尺寸的每个邻近窗口都是唯一的。会捕捉到邻近窗口数组的一部分，并解码对应于所捕捉数组的所提取比特的子集来标识误差比特。通过进一步处理所述经解码的比特，以在非重复数组中确定所捕捉数组的位置。



1. 一种用于确定所捕捉数组位置的方法，其特征在于，所述的方法包括：
 - (A) 获取与所捕捉数组相关联的所提取的比特；
 - 5 (B) 如果不存在误差比特，则确定位置坐标；以及
 - (C) 如果存在误差比特，则通过非重复比特选择，从所提取的比特的一部分确定所述位置坐标，其中，所述的位置坐标符合一局部约束。
2. 如权利要求 1 所述的方法，其特征在于，所述 (B) 包含：
 - (i) 从所提取的比特中选择第一子集；
 - 10 (ii) 解码所述第一子集；以及
 - (iii) 响应于 (ii)，如果没有检测到误差比特，确定所捕捉数组的位置坐标。
3. 如权利要求 2 所述的方法，其特征在于，所述 (ii) 利用第一矩阵方程 $\tilde{\mathbf{b}}' = \mathbf{r}'\tilde{\mathbf{M}}$ ，其中 $\tilde{\mathbf{M}}$ 是 \mathbf{M} 的子矩阵，来确定向量 \mathbf{r} 。
- 15 4. 如权利要求 3 所述的方法，其特征在于，所述 (ii) 还利用第二矩阵方程 $\hat{\mathbf{b}}' = \mathbf{r}'\mathbf{M}$ 来确定经解码的比特。
5. 如权利要求 2 所述的方法，其特征在于，所述 (iii) 包含：
 - (1) 将经解码的比特与所提取的比特作比较。
6. 如权利要求 1 所述的方法，其特征在于，所述 (C) 包含：
 - 20 (i) 如果检测到误差比特，则从所提取的比特中选一择不同的子集，其中，所述不同子集中至少一个比特是来自先前的误差比特集；
 - (ii) 解码所述不同子集的相应比特；
 - (iii) 响应于 (ii)，确定是否应执行另一解码反复；
 - (iv) 如果应执行另一解码反复，则从所提取的比特中选择另一子集并重复
 - 25 (ii)；以及
 - (v) 如果不应执行另一解码反复，则确定所捕捉数组的位置坐标。
7. 如权利要求 6 所述的方法，其特征在于，由 $\hat{\mathbf{b}}' = \mathbf{r}'\mathbf{M}$ 计算所得的比特和所提取的比特是不同的。
8. 如权利要求 2 所述的方法，其特征在于，所述 (B) 还包含：
 - 30 (iv) 验证所述位置坐标在一目的区域内。

9. 如权利要求 6 所述的方法, 其特征在于, 所述 (C) 还包含:
(vi) 验证所述位置坐标在一目的区域内。
10. 如权利要求 2 所述的方法, 其特征在于, 所述 (i) 包含:
(1) 随机地从所提取的比特中选择所述第一子集的组成比特。
- 5 11. 如权利要求 6 所述的方法, 其特征在于, 所述 (i) 包含:
(1) 以根据经解码的比特是否在所述方法的前一次反复中满足矩阵方程 $\mathbf{b}^i = \mathbf{r}^i \mathbf{M}$ 的方式, 从所提取的比特中选择所述不同子集的相应比特; 以及
(2) 计算从 $\hat{\mathbf{b}}^i = \mathbf{r}^i \mathbf{M}$ 计算出的比特和所提取的比特之间有多少不同比特。
12. 如权利要求 6 所述的方法, 其特征在于, 所述 (v) 包含:
10 (1) 如果确定的向量对应于低于一阈值的误差比特数, 则选择所确定的向量 \mathbf{r}_i ; 以及
(2) 响应于 (1), 根据相应的误差比特数的升序对多个所确定的向量排序。
13. 如权利要求 12 所述的方法, 其特征在于, 所述 (v) 还包含:
(3) 响应于 (2), 找出对应于一目的区域内所述位置坐标的第一解。
- 15 14. 如权利要求 13 所述的方法, 其特征在于, 所述 (v) 还包含:
(4) 如果没有位于所述目的区域内的解, 则指示解码失败。
15. 如权利要求 6 所述的方法, 其特征在于, 所述 (iii) 包含:
(1) 根据误差比特的期望误差率, 调整所述方法所需的反复次数。
16. 如权利要求 6 所述的方法, 其特征在于, 所述 (ii) 包含:
20 (1) 通过利用第一矩阵方程 $\hat{\mathbf{b}}^i = \mathbf{r}^i \tilde{\mathbf{M}}$, 确定向量 \mathbf{r} ;
(2) 通过利用第二矩阵方程 $\hat{\mathbf{b}}^i = \mathbf{r}^i \mathbf{M}$, 计算经编码的比特, 其中, 所述向量由 (1) 确定; 以及
(3) 将经解码的比特和所提取的比特作比较, 以找出若干误差比特。
17. 一种具有用于执行权利要求 1 所述的方法的计算机可执行指令的计算机
25 可读媒质。
18. 一种具有用于执行权利要求 2 所述的方法的计算机可执行指令的计算机可读媒质。
19. 一种具有用于执行权利要求 6 所述的方法的计算机可执行指令的计算机可读媒质。
- 30 20. 一种确定所捕捉数组的位置坐标的装置, 其特征在于, 它包含:
(a) 一比特选择模块, 它应用一智能策略来选择比特的子集, 所述比特子集

含有来自先前误差比特集的至少一个比特,所述子集从对应于所捕捉数组的所提取比特中选择;

(b) 一解码模块,它处理所述比特的子集,以确定关于所提取比特的误差信息,并且从所述误差信息确定是否需要另一解码的反复;以及

5 (c) 一位置确定模块,它处理所述误差信息,以确定所捕捉数组的位置坐标,其中,所述位置坐标在一目的区域内。

21. 如权利要求 20 所述的装置,其特征在于,它还包含:

一输入接口,它接收所提取的比特,并将所提取的比特提交给所述比特选择模块用于处理。

10 22. 如权利要求 20 所述的装置,其特征在于,所述的解码模块为第 i 次反复计算位置矩阵 r_i ,并通过将根据 $\hat{b}' = r'M$ 所确定的经解码比特与所提取比特作比较来确定误差比特。

23. 如权利要求 20 所述的装置,其特征在于,它还包含:

15 一输出接口,它提供所捕捉数组的位置坐标,其中,所述位置坐标由所述位置确定模块确定。

24. 一种用于确定所捕捉数组的位置坐标的方法,其特征在于,所述方法包含:

(A) 接收与所捕捉数组相关联的所提取的比特;

(B) 从所提取的比特中选择第一比特子集;

20 (C) 解码所述第一比特子集;

(D) 如果没有检测到误差比特,确定所捕捉数组的位置坐标,所述位置坐标在一目的区域内;

(E) 从所提取的比特中选择一不同的子集,其中,所述不同子集的至少一个比特不满足矩阵方程 $\hat{b}' = r'M$;

25 (F) 解码所述不同子集的所选择的比特;

(G) 调整执行 (F) 的反复次数,所述次数按照 (F) 的结果来调整;

(H) 如果应当执行另一解码反复,重复 (E) - (G); 以及

(I) 如果不应执行另一解码反复,确定所捕捉数组的位置坐标,其中,所述位置坐标在所述目的区域内。

m-数组解码和纠错的增强方法

技术领域

- 5 本发明涉及使用数码笔与媒质交互,尤其涉及在与一个或多个表面交互期间确定数码笔的位置。

背景技术

- 10 计算机用户习惯于使用鼠标和键盘作为与个人计算机交互的方法。虽然个人计算机提供了优于书面文档的多种优点,但多数用户仍旧使用印刷纸来完成某些功能。这些功能中的一些包括阅读和注释书面文档。在注释的情况下,由于用户置于印刷文档上的注释,印刷文档表现出更大的重要性。然而,拥有具有注释的印刷文档的困难之一是以后需要将这些注释重新输入回文档的电子形式。这需要原用户或另一个用户辛苦地读完注释,并将它们输入到个人计算机中。在15 一些情况下,用户将扫描输入这些注释和原文本,由此建立新的文件。这些多个步骤使得印刷文档和电子版本的文档之间的交互难以用重复的方式来处理。此外,扫描输入的图像经常是不可修改的。没有将注释同原文本分开的方法。这使得使用注释变得困难。因此,需要一种处理注释的改进的方法。

- 20 捕捉手写信息的一种技术是使用笔,其位置会在书写期间被确定。提供这种性能的一种笔是 Anoto 公司的 Anoto 笔。该笔通过使用照相机捕捉用预定义图案编码的纸张的图像来起作用。该图像图案的例子在图 11 中示出。Anoto 笔(Anoto 公司)使用这种模式来确定一页纸张上该笔的位置。然而,不清楚 Anoto 笔所使用的系统对确定位置有多高的效率。为了提供对所捕捉图像位置的有效确定,需要一种提供所捕捉图像的有效解码的系统。

25

发明内容

- 30 本发明的各个方面提供了上述问题的至少其中之一的解决方案,由此使得可以在具有预定义图案的所察看的文档上定位所捕捉图像的一个或多个位置。所察看的文档可以在具有预定义图案的纸张、LCD 屏幕或其它任何的媒质上。本发明的各方面包括准许对所捕捉的图像进行有效解码的解码过程,为图像位置的有效确定作准备。

依照本发明的一个方面,解码过程巧妙地从不捕捉图像提取的比特中选择比特子集。依照本发明的另一个方面,一种过程调节该解码过程重复执行的次数。依照本发明的另一个方面,一种过程确定所提取比特位置的 X、Y 坐标,由此使得 X、Y 坐标符合诸如目的区域等局部约束。通过以下附图和所关联的描述,本发明的这些和其它方面将变得显而易见。

附图说明

结合附图阅读上述本发明的发明内容和以下详细描述的首选实施例,可以更好地理解它们,这些优选实施例对于本发明而言是作为示例而非局限包括于此。

- 10 图 1 示出了可以结合本发明的实施例使用的计算机的总体描述。
- 图 2A 和 2B 示出了依照本发明的实施例的图像捕捉系统和相应的所捕捉图像。
- 图 3A 到 3F 示出了依照本发明的实施例的各个序列和折叠技术。
- 图 4A 到 4E 示出了依照本发明的实施例的各个编码系统。
- 15 图 5A 到 5D 示出了与依照图 4A 和 4B 的解码系统相关联的四种可能的结果拐角。
- 图 6 示出了依照本发明的实施例的所捕捉图像部分的旋转。
- 图 7 示出了结合图 4A 到 4E 的编码系统使用的旋转的各个角度。
- 图 8 示出了依照本发明的实施例用于确定所捕捉数组位置的过程。
- 图 9 示出了依照本发明的实施例用于确定所捕捉图像位置的方法。
- 20 图 10 示出了依照本发明的实施例用于确定所捕捉图像位置的另一个方法。
- 图 11 示出了依照现有技术文档中编码空格的表示。
- 图 12 示出了依照本发明的实施例用于解码从所捕捉的图像中提取的比特的流程图。
- 图 13 示出了依照本发明的实施例对从所捕捉的图像中提取的比特进行比特选
- 25 择。
- 图 14 示出了依照本发明的实施例用于解码从所捕捉的图像中提取的比特的一种装置。

具体实施方式

- 30 本发明的各方面涉及确定所捕捉图像相对于较大图像的位置。此处所描述的位置确定方法和系统可与多功能笔结合使用。

为了方便读者，用小标题将下列描述分开。所述小标题包括：术语、通用计算机、图像捕捉笔、数组编码、解码、纠错和位置确定。

术语

- 5 笔——一种书写工具，可具有或不具有存储墨水的能力。在某些例子中，不具有墨水存储能力的触针可被用作依照本发明实施例的笔。

照相机——一种图像捕捉系统，可从纸张或其它媒质捕捉图像。

通用计算机

- 10 图 1 是可用于实现本发明各个方面的常规通用数字计算环境的例子的功能性框图。在图 1 中，计算机 100 包括处理单元 110、系统存储器 120 以及把包括系统存储器在内的各种系统组件耦合至处理单元 110 的系统总线 130。系统总线 130 可能是若干总线结构类型中的任一种，包括存储器总线或存储器控制器、外围总线、以及使用多种总线体系结构的局部总线的任一种。系统存储器 120 包括只读存储器
15 (ROM) 140 和随机存取存储器 (RAM) 150。

- 基本输入/输出系统 160 (BIOS) 存储在 ROM 140 内，它包括如启动时帮助在计算机 100 内的元件间传输信息的基本例程。计算机 100 也包括用于从硬盘 (未示出) 读取或向其中写入的硬盘驱动器 170、从可移动磁盘 190 读取或向其中写入的
20 磁盘驱动器 180、以及用于向诸如 CD ROM 或其它光学媒质这样的可移动光盘 192 读写的光盘驱动器 191。硬盘驱动器 170、磁盘驱动器 180 和光盘驱动器 191 分别通过硬盘驱动器接口 192、磁盘驱动器接口 193 和光盘驱动器接口 194 连接到系统总线 130。这些盘驱动器及其相关的计算机可读媒质为个人计算机 100 提供了对计算机可读指令、数据结构、程序模块和其它数据的非易失性存储。本领域的技术人员可以理解，用于存储计算机可访问数据的其它类型的计算机可读媒质也可在示例
25 性操作环境中使用，这些其它计算机可读媒质诸如磁带盒、闪存卡、数字视频盘、Bernoulli 盒式磁带、随机存取存储器 (RAM)、只读存储器 (ROM) 等等。

- 许多程序模块可以存储在硬盘驱动器 170、磁盘 190、光盘 192、ROM 140 或 RAM 150 上，包括操作系统 195、一个或多个应用程序 196、其它程序模块 197 和程序数据 198。用户可以通过输入设备输入命令和信息到计算机 100 中，输入设备
30 诸如键盘 101 和定位设备 102。其它输入设备 (未示出) 可以包括麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等。这些和其它输入设备一般通过耦合到系统

总线的串行端口接口 106 连接到处理单元 110，但也可由其它接口连接，诸如并行端口、游戏端口或通用串行总线（USB）。此外，这些设备可以通过适当的接口（未示出）直接耦合到系统总线 130。监视器 107 或其它类型的显示设备也可以通过诸如视频适配器 108 这样的接口连接到系统总线 130。除监视器外，个人计算机一般包括诸如扬声器和打印机等其它外围输出设备（未示出）。在优选实施例中，为了数字地捕捉手画线输入，提供了笔数字化仪 165 和伴随的笔或触针 166。虽然示出了笔数字化仪 165 和串行端口间的直接连接，然而实际上，笔数字化仪 165 可通过并行端口或本领域已知的其它接口和系统总线 130 而直接耦合到处理单元 110。而且，虽然所示的数字化仪 165 离监视器 107 较远，但数字化仪 165 的可用输入区最好与监视器 107 的显示区域有共同的范围。此外，数字化仪 165 可被集成到监视器 107 内，或可作为叠加或附着于监视器 107 上的独立设备而存在。

计算机 100 可工作在使用到一个或多个诸如远程计算机 109 这样的远程计算机的逻辑连接的网络化环境内。远程计算机 109 可以是服务器、路由器、网络 PC、对等设备或其它公共网络节点，并且一般包括与计算机 100 相关的许多或所有上述元件，然而图 1 中仅示出了存储器存储设备 111。图 1 所述的逻辑连接包括局域网（LAN）112 以及广域网（WAN）113。这种网络环境常见于办公室、企业范围的计算机网络、内联网以及因特网。

当用于 LAN 网络环境时，计算机 100 通过网络接口或适配器 114 连接到局域网 112。当用于 WAN 网络环境时，个人计算机 100 一般包括调制解调器 115 或其它装置，用于在诸如因特网等广域网 113 上建立通信。调制解调器 115 可以是内置或外置的，它通过串行口接口 106 连接到系统总线 130。在网络化环境中，所述与个人计算机 100 相关的程序模块或其中的一部分可存储在远程存储设备内。

可以理解，所示网络连接是说明性的，并且可以使用其它用于在计算机间建立通信链路的技术。假设存在诸如 TCP/IP、以太网、FTP、HTTP、蓝牙、IEEE 802.11X 等各种已知协议，系统可以工作在客户机-服务器配置中，以允许用户从基于 web 的服务器中检索网页。可以用各种常规网页浏览器的任一种来显示和操纵网页上的数据。

图像捕捉笔

本发明的各方面包括将经编码的数据流置于表现该经编码的数据流的显示形式中。（例如，如将与图 4B 一起论述的，该经编码的数据流被用于创建图形图案。）

所述显示形式可以是印刷纸张（或其它物理媒质），或可以是结合另一图像或另一组图像投影该经编码的数据流的显示器。例如，该经编码的数据流可以被表现为纸张上的物理图形图像或叠加在显示图像上的图形图像（例如，表示文件的文本），或可以是显示屏上的物理（不可修改）图形图像（这样，任何由笔捕捉的图像部分在显示屏上是可定位的）。

对所捕捉图像位置的确定可以被用于确定用户与纸张、媒质或显示屏交互的位置。在本发明的某些方面中，笔可以在纸上书写的墨水笔。在其它的方面，笔可以是用户在计算机显示器表面书写的触针。在知道文档上经编码的图像或支持在计算机屏幕上显示的文档的情况下，任何交互可被提供回系统中。当笔或触针在文档上来回移动时，系统可以通过用笔或触针中的照相机反复地捕捉图像来跟踪由用户控制的触针的移动。该显示或印刷图像可以是与空白或内容丰富的纸张相关联的水印，或可以是与叠加在屏幕上或内建到屏幕中的显示图像或固定编码相关联的水印。

图 2A 和 2B 示出了具有照相机 203 的笔 201 的说明性例子。笔 201 包括可包括或不包括墨水贮液器的笔尖 202。照相机 203 从表面 207 捕捉图像 204。笔 201 还可包括如显示在虚线框 206 中的附加传感器和/或处理器。这些传感器和/或处理器 206 也可包含将信息发送到另一支笔 201 和/或个人计算机的能力（例如，通过蓝牙或其它无线协议）。

图 2B 示出了由照相机 203 所观看到的图像。在一个说明性例子中，照相机 203 的视图区（即，照相机图像传感器的分辨率）是 32×32 像素（其中 $N=32$ ）。在该实施例中，所捕捉的图像（ 32 像素 \times 32 像素）对应于由照相机 203 捕捉的表面平面图的大约 $5\text{mm} \times 5\text{mm}$ 的区域。因此，图 2B 示出了 32 像素长 \times 32 像素宽的视图区。N 的大小是可调整的，使得较大的 N 对应于较高的图像分辨率。同样地，虽然照相机 203 的视图区在这里为了说明性的目的而被显示为正方形，然而视图区可以包括其它本领域公知的形状。

由照相机 203 捕捉的图像可以被定义为图像帧序列 $\{I_i\}$ ，其中， I_i 是由笔 201 在采样时间 t_i 捕捉的。根据系统配置和性能要求，采样率可大可小。根据系统配置和性能要求，所捕捉图像帧的尺寸可大可小。

由照相机 203 捕捉的图像可以直接由处理系统使用，或可经过预过滤。所述预过滤可以发生在笔 201 内部，或发生在笔 201 外部（例如，在个人计算机中）。

图 2B 的图像尺寸是 32×32 像素。如果每个编码单元尺寸是 3×3 像素，那么

所捕捉的经编码单元的数目将大约为 100 单元。如果编码单元尺寸是 5×5 像素，那么所捕捉的经编码单元的数目大约为 36。

图 2A 也示出了图像平面 209，来自位置 204 的图案的图像 210 在其上形成。接收自对象平面 207 上的图案的光由透镜 208 聚焦。透镜 208 可以是单透镜或由多部分组成的透镜系统，但为了简化起见在这里被显示为单透镜。图像捕捉传感器 211 捕捉图像 210。

图像传感器 211 可以大到足以捕捉图像 210。或者，图像传感器 211 可以大到足以捕捉笔尖 202 在位置 212 处的图像。作为参考，在位置 212 处的图像被称为可视笔尖。需要注意，由于笔尖、透镜 208 和图像传感器 211 的固定关系，对于图像传感器 211 的虚拟笔尖位置是固定的。

以下的转换 $F_{S \rightarrow P}$ 将由照相机捕捉的图像的位置坐标转换为在纸上的实际图像中的位置坐标：

$$L_{paper} = F_{S \rightarrow P}(L_{Sensor})$$

在书写过程中，笔尖和纸在同一平面上。因此，从虚拟笔尖到实际笔尖的转换也是 $F_{S \rightarrow P}$ ：

$$L_{penip} = F_{S \rightarrow P}(L_{virtual-penip})$$

转换 $F_{S \rightarrow P}$ 可以被估算为仿射变换。这简化为：

$$F'_{S \rightarrow P} = \begin{pmatrix} \frac{s_x \sin \theta_y}{\cos \theta_x \sin \theta_y - \cos \theta_y \sin \theta_x}, & -\frac{s_x \cos \theta_y}{\cos \theta_x \sin \theta_y - \cos \theta_y \sin \theta_x}, & 0 \\ \frac{s_y \sin \theta_x}{\cos \theta_x \sin \theta_y - \cos \theta_y \sin \theta_x}, & \frac{s_y \cos \theta_x}{\cos \theta_x \sin \theta_y - \cos \theta_y \sin \theta_x}, & 0 \\ 0, & 0, & 1 \end{pmatrix}$$

作为 $F_{S \rightarrow P}$ 的估算，其中 θ_x 、 θ_y 、 s_x 和 s_y 是在位置 204 处捕捉的图案的两个方位的旋转和比例。此外，我们可以通过将所捕捉的图像与纸上相应的真实图像进行匹配来精确化 $F'_{S \rightarrow P}$ 。“精确化”意指通过一种被称为递归法的最优化算法，取得对转换 $F'_{S \rightarrow P}$ 更为精确的估算。递归法将矩阵 $F'_{S \rightarrow P}$ 作为初始值来处理。精确化的估算可以更精确地描述 S 和 P 之间的转换。

接着，我们可以通过校准来确定虚拟笔尖的位置。

我们将笔尖 202 放在纸上固定位置 L_{penip} 。接着，我们倾斜该笔，使照相机 203 可以就不同的笔的姿势来捕捉一系列图像。对于每个捕捉的图像，我们可以获取转换 $F_{S \rightarrow P}$ 。我们可以从这种转换获取虚拟笔尖 $L_{virtual-penip}$ 的位置：

$$L_{\text{virtual-pentip}} = F_{P \rightarrow S}(L_{\text{pentip}})$$

其中, L_{pentip} 被初始化为(0,0), 并且

$$F_{P \rightarrow S} = (F_{S \rightarrow P})^{-1}$$

通过对从每个图像获取的 $L_{\text{virtual-pentip}}$ 求平均值, 可以确定虚拟笔尖 $L_{\text{virtual-pentip}}$ 的
5 位置。有了 $L_{\text{virtual-pentip}}$, 我们可以取得对 L_{pentip} 更为准确的估算。在几次反复之后, 可以确定虚拟笔尖 $L_{\text{virtual-pentip}}$ 的准确位置。

现在知道了虚拟笔尖 $L_{\text{virtual-pentip}}$ 的位置。我们也可以从所捕捉的图像中获取转换 $F_{S \rightarrow P}$ 。最后, 我们可以使用该信息来确定实际笔尖 L_{pentip} 的位置:

$$L_{\text{pentip}} = F_{S \rightarrow P}(L_{\text{virtual-pentip}})$$

10

数组编码

二维的数组可以通过折叠一维序列来构造。包含足够大量的比特的二维数组的任一部分可用于确定它在完整的二维数组中的位置。然而, 会需要从一个捕捉的图像或一些捕捉的图像中确定位置。为了使所捕捉的图像与该二维数组中两个或多个
15 位置相关联的可能性最小, 可使用无重复序列来创建数组。所创建序列的一个特性是该序列不在长度 (或窗口) n 上重复。以下描述创建一维序列, 接着将该序列折叠成数组。

序列的构造

20 一数字序列可被用作编码系统的开始点。例如, 一序列 (也被称为 m -序列) 可被表示为在 F_q 域中的 q -元素集。这里, $q=p^n$, 其中 $n \geq 1$ 且 p 是质数。该序列或 m -序列可以由多种不同的技术生成, 这些技术包含但不限于多项式除法。使用多项式除法, 序列可以被定义如下:

$$\frac{R_l(x)}{P_n(x)}$$

25 其中 $P_n(x)$ 是域 $F_q[x]$ 中次数为 n 的本原多项式 (具有 q^n 个元素)。 $R_l(x)$ 是域 $F_q[x]$ 中次数为 l 的非零多项式 (其中 $l < n$)。该序列可以使用反复的过程用两步创建: 第一, 将这两个多项式相除 (得出域 F_q 里的一元素), 第二, 将余数乘以 x 。当输出开始重复时, 计算停止。该过程可以使用线性反馈移位寄存器来实现, 如关于计算机的 IEEE 学报 43.5 (IEEE Transactions on Computers 43.5) (1994 年 5 月, 第
30 560 - 568 页) 中 Douglas W.Clark 和 Lih-Jyh Weng 的论文 "Maximal and

Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms”所提出的。在这个环境中，在序列的循环移位和多项式 $R_l(x)$ 之间建立起一种关系：改变 $R_l(x)$ 只循环地移位该序列且每个循环移位对应于一多项式 $R_l(x)$ 。所得序列的特性之一是，该序列具有 $q^n - 1$ 的周期，且在一个周期内，宽度（或长度） n 上，任一部分在序列中出现一次且只有一次。这被称作为“窗特性”。周期 $q^n - 1$ 也被称为序列的长度，且 n 被称为序列的次数。

上述的过程只是可用于建立具有窗特性的序列的多种过程之一。

数组的构建

10 可用于创建图像（其一部分可用照相机捕捉）的数组（或 m -数组）是一维序列或 m -序列的扩展。设 A 为周期 (m_1, m_2) 的一数组，即 $A(k + m_1, l) = A(k, l + m_2) = A(k, l)$ 。当 $n_1 \times n_2$ 窗口通过 A 的周期位移时， F_q 上的所有非零 $n_1 \times n_2$ 矩阵都出现一次且只有一次。该特性也被称为“窗特性”，因为每个窗口是唯一的。接着，窗口可被表示成周期为 (m_1, m_2) 数组（ m_1 和 m_2 为出现在数组中比特的水平和垂直数目）、阶数为 (n_1, n_2) 的数组。

15 可以通过折叠序列来构建二元数组（或 m -数组）。一种方法是获取一序列，接着将它折叠成 $m_1 \times m_2$ 的尺寸，其中数组的长度是 $L = m_1 \times m_2 = 2^n - 1$ 。或者，我们可以从我们希望覆盖的预定尺寸的空间开始（例如，一页纸，30 页纸或计算机监视器的尺寸），确定区域 $(m_1 \times m_2)$ ，接着使用该尺寸使得 $L \geq m_1 \times m_2$ ，其中 $L = 2^n - 1$ 。

20 可使用多种不同的折叠技术。例如图 3A 到 3C 示出了三种不同的序列。这些序列的每一种可被折叠成图 3D 所示的数组。这三种不同的折叠方法被显示为图 3D 中的重叠，以及图 3E 和 3F 中的光栅路径。我们在此采用图 3D 中所示的折叠方法。

25 为了创建如图 3D 所示的折叠方法，我们创建一长度为 L ，阶数为 n 的序列 $\{a_i\}$ 。接着，通过使该数组的每个比特如方程 1 所示地计算，从序列 $\{a_i\}$ 创建尺寸为 $m_1 \times m_2$ 的数组 $\{b_{kl}\}$ ，其中， $\gcd(m_1, m_2) = 1$ ，且 $L = m_1 \times m_2$ ：

$$b_{kl} = a_i, \text{ 其中, } k = i \bmod(m_1), l = i \bmod(m_2), i = 0, \dots, L-1 \quad (1)$$

这种折叠方法或者可表示为将序列置于数组的对角线上，接着当达到边缘时，从对侧的边缘继续。

30 图 4A 示出了可被用于编码图 3D 的数组的示例编码技术。应该理解，可以使用其它的编码技术。例如，图 11 示出了一种可供选择的编码技术。

参见图 4A，第一个比特 401（例如，“1”）可由一列黑墨水表示。第二个比特 402（例如，“0”）可由一行黑墨水表示。应该理解，任何颜色的墨水可被用于表示各种比特。对墨水颜色唯一的要求是它要提供与媒质背景的显著对比，使得图像捕捉系统可以辨认。图 4A 的比特可用单元的 3×3 矩阵表示。基于图像捕捉系统的尺寸和分辨率，矩阵的尺寸可以被更改为任何尺寸。图 4C-4E 中示出了比特 0 和 1 的交替表示。应该理解，图 4A-4E 的示例编码对 1 和 0 的表示可以不受影响地交换。图 4C 示出了以隔行排列的占据 2 行或列的比特表示。图 4D 示出了在行和列中以虚线的形式对像素的可选排列。最后，图 4E 示出了在行和列中以不规则间隔格式的像素表示（例如，两个黑点后接着一个白点）。

- 10 再参见图 4A，如果一个比特由 3×3 的矩阵表示，且成像系统在 3×3 的区域中检测到一个黑的行和两个白的行，那么就检测到一个 0（或是 1）。如果检测到图像具有一个黑的列和两个白的列，那么就检测到一个 1（或是 0）。

这里使用一个以上像素或点来表示一个比特。使用单个的像素（或比特）来表示一个比特是易受到破坏的。灰尘、纸上的折痕、不平的表面等等造成了读出数据单元的单个比特表示的困难。但是，应该理解，可以使用不同的方法来图形地表示平面上的数组。图 4C 到 4E 中示出了一些方法。应该理解，也可以使用其它的方法。在图 11 中提出了一种方法：仅使用空间一位移（space-shifted）的点。

- 20 比特流被用于创建图 4B 的图形图案 403。图形图案 403 包括 12 行和 18 列。该行和列由被使用比特表示 401 和 402 转换为图形表示的比特流形成。图 4B 可被看作具有以下比特表示：

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

解码

- 25 当人们使用图 2A 的笔书写或将笔移动到接近已编码图案之处，照相机就捕捉图像。例如，当笔 201 压住纸张并且笔 201 在纸张上的文档上来回移动时，笔 201 会利用压力传感器。接着，处理图像以确定所捕捉图像对于经编码图像的完整表示的方向，并提取组成所捕捉图像的比特。

为了确定所捕捉图像对于整个编码区域的方向，我们会注意到并非图 5A-5D 显示的所有四个能想象得到的拐角会出现在图形图案 403 中。实际上，就正确的定

位，图 5A 所示的拐角类型不会存在于图形图案 403 中。因此，缺少图 5A 所示的拐角类型的方向是正确的方向。

5 继续到图 6，可分析由照相机 601 所捕捉的图像，并确定其方向，以使其可按照由图像 601 实际表示的位置来解释。首先，考察图像 601 以确定旋转图像所需要的角度 θ ，使得像素是水平和垂直对齐的。应该注意，可选的栅格对齐是可能的，包括将底层的栅格到非水平和垂直排列的旋转（例如，45 度）。使用非水平和垂直对齐会提供从用户消除视觉上分心的可能的好处，因为用户往往在注意到其它图案之前注意到水平和垂直图案。为了简化起见，栅格的方向（水平的和垂直的，和对底层栅格的任何其它的旋转）被总称为预定义的栅格方向。

10 接着，分析图 601，以确定缺少了哪个拐角。将图像 601 旋转至为解码做好准备的图像 603 所需的旋转量 ϕ 被显示为 $\phi = (\theta \text{ 加 旋转量 } \{ \text{由缺少的拐角定义} \})$ 。该旋转量由图 7 中的方程示出。再参见图 6，角度 θ 最初由使像素达到像素的水平或垂直（或其它预定义的栅格方向）排列的布局来确定，且图像如 602 中所示的那样旋转。接着，执行分析以确定缺少的拐角，并将图像 602 旋转至图像 603，以建立
15 用于解码的图像。这里，图像被逆时针旋转了 90 度，使得图像 603 就具有了正确的方向并且可被用于解码。

应该理解，旋转角度 θ 可以在旋转图像 601 之前或之后应用，来解决缺少的拐角。也应该理解，考虑到所捕捉图像中的噪声，可能会存在所有的四个拐角。我们可以计算每种类型的拐角的数目，并选择数目最少的那种类型作为缺少的拐角类型。
20

最后，图像 603 中的代码被读出并与用于创建图像 403 的原始比特流进行相关。这种相关可以用多种方法实现。例如，可以使用递归法来执行，其中，恢复的比特流与原始比特流中的所有其它比特流片断作比较。第二，可在恢复的比特流和原始比特流之间执行统计分析，例如，通过使用两种比特流之间的汉明距离。应该理解，
25 可以使用多种方法来确定原始比特流中恢复的比特流的位置。

一旦我们得到了恢复的比特，我们需要在原始数组中定位所捕捉的图像（例如，图 4B 中所显示的那个）。由于许多项目使得在整个数组中确定比特片段位置的过程复杂化。第一，实际要捕捉的比特可能是模糊的（例如，照相机可能捕捉手写的图像，这使得原始代码变得模糊）。第二，灰尘、折痕、反射等等也会在所捕捉的
30 图像中引起误差。这些误差使得定位程序更加困难。在这点上，图像捕捉系统会需要用从图像中提取的非连续比特来运作。以下显示了用来自图像的非连续比特来运

作的方法。

5 设序列（或 m -序列） \mathbf{I} 对应于幂级数 $I(x) = 1/P_n(x)$ ，其中 n 是 m -序列的阶数，且所捕捉的图像包含 K 比特的 $\mathbf{Ib} = (b_0 \ b_1 \ b_2 \ \dots \ b_{k-1})'$ ，其中 $K \geq n$ ，且上标 t 表示矩阵或向量的转置。 K 比特的位置 s 正好是 \mathbf{I} 的循环位移的次数，使得 b_0 被位移至序列的开始处。那么该位移序列 \mathbf{R} 对应于幂级数变量 $x^s/P_n(x)$ ，或 $\mathbf{R} = T^s(\mathbf{I})$ ，其中 T 是循环位移算子。我们间接地得出该 s 。多项式模数 $P_n(x)$ 形成一个域。确保 $x^s \equiv r_0 + r_1x + \dots + r_{n-1}x^{n-1} \pmod{P_n(x)}$ 。因此，我们可得出 $(r_0, r_1, \dots, r_{n-1})$ ，接着求出 s 。

关系 $x^s \equiv r_0 + r_1x + \dots + r_{n-1}x^{n-1} \pmod{P_n(x)}$ 意味着 $\mathbf{R} = r_0 + r_1T(\mathbf{I}) + \dots + r_{n-1}T^{n-1}(\mathbf{I})$ 。写成二元线性方程，就成为：

$$10 \quad \mathbf{R} = \mathbf{r}'\mathbf{A} \quad (2)$$

其中 $\mathbf{r} = (r_0 \ r_1 \ r_2 \ \dots \ r_{n-1})'$ ，且 $\mathbf{A} = (\mathbf{I} \ T(\mathbf{I}) \ \dots \ T^{n-1}(\mathbf{I}))$ ，它由 \mathbf{I} 从 0-位移到 $(n-1)$ -位移的循环位移组成。现在，在 \mathbf{R} 中只有稀疏的 K 比特可用来解出 \mathbf{r} 。设 \mathbf{R} 中 b_i 和 b_0 之间的下标差为 k_i ， $i = 1, 2, \dots, k-1$ ，那么 \mathbf{R} 的第一和第 $(k_i + 1)$ 个元素（ $i = 1, 2, \dots, k-1$ ）正好是 b_0, b_1, \dots, b_{k-1} 。通过选择 \mathbf{A} 的第一和第 $(k_i + 1)$ 列

15 （ $i = 1, 2, \dots, k-1$ ），形成了以下的二元线性方程：

$$\mathbf{b}' = \mathbf{r}'\mathbf{M} \quad (3)$$

其中， \mathbf{M} 是 \mathbf{A} 的 $n \times K$ 子矩阵。

如果 \mathbf{b} 是无误差的，则 \mathbf{r} 的解可被表示为：

$$\mathbf{r}' = \tilde{\mathbf{b}}'\tilde{\mathbf{M}}^{-1} \quad (4)$$

20 其中 $\tilde{\mathbf{M}}$ 是 \mathbf{M} 的非退化 $n \times n$ 子矩阵，且 $\tilde{\mathbf{b}}$ 是 \mathbf{b} 的相应的子向量。

已知 \mathbf{r} 的情况下，我们可使用 Pohlig-Hellman-Silver 算法，如关于计算机的 IEEE 学报 43.5 (IEEE Transactions on Computers 43.5) (1994 年 5 月，第 560—568 页) 中 Douglas W. Clark 和 Lih-Jyh Weng 的论文“Maximal and Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms”所提及

25 的，来解出 s ，使得 $x^s \equiv r_0 + r_1x + \dots + r_{n-1}x^{n-1} \pmod{P_n(x)}$ 。

因为矩阵 \mathbf{A} (其尺寸为 $n \times L$, 其中 $L=2^n - 1$) 可能会非常大, 我们应避免存储整个矩阵 \mathbf{A} 。实际上, 如我们在上述过程中所见的, 给定下标差为 k_i 的提取的比特, 只有 \mathbf{A} 的第一和第 $(k_i + 1)$ 列是和计算相关的。给定所捕捉图像的尺寸, 对 k_i 的这种选择是相当有限的。因此, 只需要保存那些在计算中涉及的列。这种列的总数远小于 L (其中 $L=2^n - 1$ 是 m -序列的长度)。

纠错

如果在 \mathbf{b} 中存在误差, 那么 \mathbf{r} 的解法变得更为复杂。因为与所捕捉的比特相关联的矩阵 \mathbf{M} 会在各个捕捉的图像之间变化, 因此不能轻易地应用传统的带有纠错的解码方法。

我们采用一种随机的方法。假设 \mathbf{b} 中误差比特的数目 n_e 与 K 相比是相对小的, 那么从 \mathbf{b} 的 K 比特中选择出正确的 n 比特, 且 \mathbf{M} 的相应子矩阵 $\tilde{\mathbf{M}}$ 为非退化的可能性是高的。

当所选择的这 n 比特都正确时, \mathbf{b}' 和 $\mathbf{r}'\mathbf{M}$ 之间的汉明距离, 或与 \mathbf{r} 相关联的误差比特的数目, 应该是最小的, 其中 \mathbf{r} 通过方程(4)来计算。重复几次该过程, 可能可以标识出可推出最小误差比特的正确的 \mathbf{r} 。

如果只有一个 \mathbf{r} 与最小误差比特数相关联, 那么就将它作为正确的解。否则, 如果有一个以上 \mathbf{r} 与最小误差比特数相关联, 则 n_e 超过由 \mathbf{M} 生成的代码的纠错能力的可能性是高的, 且解码过程失败。接着, 系统会移到下一个捕捉的图像。在另一个实现中, 会考虑关于该笔的前一位置的信息。即, 对于每个所捕捉的图像, 可标识下一次会期望该笔的目的区域。例如, 如果用户在照相机的两次图像捕捉之间没有提起笔, 那么由第二图像捕捉确定的笔的位置不应该离第一位置太远。接着可以检查与最小误差比特数目相关联的每一 \mathbf{r} , 来看由 \mathbf{r} 计算所得的位置 s 是否满足局部约束, 即, 该位置是否在规定的目的区域之内。

如果位置 s 满足局部约束, 就返回数组中所提取比特的 X、Y 位置。如果不满足, 解码过程就失败。

图 8 描述了可被用于确定所捕捉的图像在序列 (或 m -序列) 中的位置的过程。第一, 在步骤 801 中, 接收到涉及所捕捉图像的数据流。在步骤 802 中, 从 \mathbf{A} 提取相应的列, 并构建矩阵 \mathbf{M} 。

在步骤 803 中, 随机地从矩阵 \mathbf{M} 中选择 n 个独立列向量, 并且通过解方程 (4) 来确定向量 \mathbf{r} 。在步骤 804 中执行该过程 Q 次 (例如, 100 次)。对循环次数的确

定在循环次数计算一节中讨论。

在步骤 805 中， r 根据其相关联的误差比特数目来排序。可以使用本领域公知的多种排序算法来完成这个排序。例如，可使用一选择排序算法。当数目 Q 不大时，选择排序算法是有利的。然而，如果 Q 变大，可使用可更有效地处理较大项目数目的其它排序算法（例如，合并分类）。

接着，系统在步骤 806 中通过检查是否有多个 r 与最小误差比特数相关联，来确定是否成功地执行了纠错。如果是，在步骤 809 中返回误差，指示该解码过程失败。如果不是，在步骤 807 中计算序列（或 m -序列）中被提取的比特的位置 s ，例如，通过使用 Pohig-Hellman-Silver 算法。

接着，计算数组中(X,Y)位置： $x = s \bmod m_1$ 且 $y = s \bmod m_2$ ，并在步骤 808 中返回结果。

位置确定

图 9 示出了用于确定笔尖位置的过程。输入是由照相机捕捉的图像，且输出可以是笔尖的位置坐标。同样地，输出可以包括（或不包括）其它信息，诸如所捕捉图像的旋转角度。

在步骤 901 中，从照相机接收到图像。接着，可在步骤 902 中可任选地预处理接收到的图像（如步骤 902 的虚线框中所示），以调整浅色和深色像素之间的对比度等等。

接着，在步骤 903 中，分析该图像以确定其中的比特流。

接着，在步骤 904 中，多次从所述比特流中随机地选择 n 比特，并且确定接收到的比特流在原始序列（或 m -序列）中的位置。

最后，一旦在步骤 904 中确定了所捕捉图像的位置，可在步骤 905 中确定笔尖的位置。

图 10 给出了关于步骤 903 和 904 更多的细节，并且示出了在捕捉的图像中提取比特流的方法。第一，在步骤 1001 中从照相机接收到图像。接着，该图像在步骤 1002 中可任选地经历图像预处理（如步骤 1002 的虚线框所示）。在步骤 1003 中提取该图案。这里，可提取不同行上的像素以找出该图案的方向和角度 θ 。

接着，在步骤 1004 中分析接收到的图像，以确定底层的网格线。如果在步骤 1005 中找到了网格线，那么就在步骤 1006 中从该图案中提取代码。接着在步骤 1007 中解码该代码，并且在步骤 1008 中确定笔尖的位置。如果在步骤 1005 中未找到网

格线，那么在步骤 1009 中返回出错。

增强的解码和纠错算法概述

采用如图 12 所示的本发明的实施例，给定从所捕捉的图像中提取的比特 1201
5 （对应于所捕捉的数组）和目的区域， m -数组解码和纠错过程的变体解码 X、Y 位置。图 12 示出了该增强方法的过程 1200 的流程图。过程 1200 包括两个组成部分 1251 和 1253。

- 解码一次。组成部分 1251 包括三部分。
 - 10 ➤ 随机比特选择：随机地选择被提取比特 1201 的子集（步骤 1203）
 - 解码该子集（步骤 1205）
 - 使用局部约束确定 X、Y 位置（步骤 1209）
- 使用智能比特选择来解码。组成部分 1253 包括四部分。
 - 智能比特选择：选择被提取比特的另一个子集（步骤 1217）
 - 15 ➤ 解码该子集（步骤 1219）
 - 调整步骤 1217 和步骤 1219 反复的次数（循环次数）（步骤 1221）
 - 使用局部约束确定 X、Y 位置（步骤 1225）

本发明的实施例利用了一种谨慎策略，依照局部约束来选择比特、调整循环次数和确定 X、Y 位置（位置坐标），所述局部约束被提供给过程 1200。在组成部分
20 分 1251 和 1253 中，步骤 1205 和 1219（“解码一次”）都使用方程（4）来计算 \mathbf{r} 。
设 $\hat{\mathbf{b}}$ 为被解码的比特，即：

$$\hat{\mathbf{b}}' = \mathbf{r}'\mathbf{M} \quad (5)$$

\mathbf{b} 和 $\hat{\mathbf{b}}$ 之间的差别是与 \mathbf{r} 相关联的误差比特。

图 12 示出了依照本发明的实施例，用于解码从所捕捉的图像中提取的比特
25 1201 的过程 1200 的流程图。程序 1200 包含组成部分 1251 和 1253。组成部分 1251

取得与所捕捉的图像（对应于所捕捉的数组）相关联的提取比特 1201（包含 K 比特）。在步骤 1203 中，从提取比特 1201 中随机地选择 n 比特（其中 n 是 m -数组的阶数）。在步骤 1205 中，过程 1200 解码一次并计算 \mathbf{r} 。在步骤 1207 中，过程 1200 确定对 \mathbf{b} 是否检测到误差比特。如果步骤 1207 确定没有误差比特，则在步骤 5 1209 中确定所捕捉数组位置的 X 、 Y 坐标。对于步骤 1211，如果 X 、 Y 坐标满足局部约束，即，坐标在目的区域内，则过程 1200 在步骤 1213 中提供 X 、 Y 位置（诸如给另一个过程或用户接口）。否则，步骤 1215 提供失败指示。

如果步骤 1207 检测到 \mathbf{b} 中的误差比特，为了在误差比特存在的情况下解码，就执行组成部分 1253。步骤 1217 从所提取的比特 1201 中选择另一组 n 比特（它与步骤 1203 中选择的 n 比特至少相差一个比特）。步骤 1221 和 1223 确定为了解码所提取的比特所需要反复的次数（循环次数）。步骤 1225 通过测试在步骤 1219 中获取的哪些候选者满足局部约束来确定所捕捉数组的位置。将更详细地讨论步骤 1217—1225。

15 智能比特选择

步骤 1203 从所提取比特 1201（具有 K 个比特）中随机地选择 n 比特，并求出 \mathbf{r}_1 。可使用方程（5）来计算被解码的比特。设 $I_1 = \{k \in \{1, 2, \dots, K\} | b_k = \hat{b}_k\}$ ， $\bar{I}_1 = \{k \in \{1, 2, \dots, K\} | b_k \neq \hat{b}_k\}$ ，其中 \hat{b}_k 是 $\hat{\mathbf{b}}$ 的第 k 个比特， $B_1 = \{b_k | k \in I_1\}$ 且 $\bar{B}_1 = \{b_k | k \in \bar{I}_1\}$ ，即， B_1 是解码结果与原始比特相同的比特，且 \bar{B}_1 是解码结果与原始比特不同的比特， I_1 和 \bar{I}_1 是这些比特相应的下标。应该理解，当从 B_1 中选择任意 n 比特时，将 20 获得相同的 \mathbf{r}_1 。因此，如果不仔细选择下 n 个比特，所选择的比特可能是 B_1 的子集，这会导致获得相同的 \mathbf{r}_1 。

为了防止这样的情况，步骤 1217 依照以下的过程选择下 n 个比特：

1. 从 \bar{B}_1 1303 中选择至少一个比特，其余的比特随机地从 B_1 1301 和 \bar{B}_1 1303 25 中选出，如对应于比特排列 1351 的图 13 所示。接着过程 1200 通过计算 $\hat{\mathbf{b}}'_2 = \mathbf{r}'_2 \mathbf{M}_2$ 得出 \mathbf{r}_2 ，并找出 B_2 1305、1309 和 \bar{B}_2 1307、1311。
2. 重复步骤 1。当对每个 $\bar{B}_i (i = 1, 2, 3, \dots, x-1)$ ，其中 x 是当前循环数目，选择下 n 个比特时，至少有一个比特是从 \bar{B}_i 中选出的。当没有这样的比特子集可 30 被选择或达到循环次数时，该反复终止。

循环次数计算

使用纠错组成部分 1253, 在每次循环后调整所需反复次数 (循环次数)。由期望的误差率来确定循环次数。其中并非所有被选择的 n 比特都是正确的期望误差率 p_e 是:

$$p_e = \left(1 - \frac{C_{K-n_e}^n}{C_K^n}\right)^{lt} \approx -e^{-lt\left(\frac{K-n}{K}\right)^{n_e}} \quad (6)$$

- 5 其中 lt 表示循环次数, 它由一个常数初始化, K 是从所捕捉数组中提取的比特的数目, n_e 表示在过程 1200 的反复期间出现的最小数目的误差比特, n 是 m -数组的阶数, 且 C_K^n 是组合的数目, 其中 n 比特是从 K 比特中选取的。

在该实施例中, 我们希望 p_e 小于 $e^{-5} = 0.0067$ 。与 (6) 结合, 我们得到:

$$lt_i = \min \left(lt_{i-1}, \frac{5}{\left(\frac{K-n}{K}\right)^{n_e}} + 1 \right) \quad (7)$$

- 10 调整循环次数会显著地减少纠错所需的过程 1253 反复的次数。

使用局部约束来确定 X、Y 位置

- 15 在步骤 1209 和 1225 中, 被解码的位置应该在目的区域内。所述目的区域是算法的输入, 且根据不同的应用, 它可以是各种尺寸和位置, 或简单地是整个 m -数组。一般地, 它可以由应用预知。例如, 如果已确定先前的位置, 考虑到书写速度, 当前笔尖的目的区域应该接近先前的位置。然而, 如果笔被提起, 那么它的下一个位置可以是任何地方。因此, 在这种情况下, 目的区域应该是整个 m -数组。由以下步骤确定正确的 X、Y 位置。

在步骤 1224 中, 过程 1200 选择 r_i , 其相应的误差比特的数目少于:

$$20 \quad N_e = \frac{\log_{10}\left(\frac{3}{lt}\right)}{\log_{10}\left(\frac{K-n}{K}\right) \times \log_{10}\left(\frac{10}{lr}\right)} \quad (8)$$

其中 lt 是实际的循环次数, 且 lr 表示局部约束率, 由以下计算:

$$lr = \frac{\text{目的区域的面积}}{L} \quad (9)$$

其中 L 是 m -数组的长度。

- 25 步骤 1224 以误差比特数目升序来对 r_i 排序。接着, 步骤 1225、1211 和 1212 找出第一 r_i , 在该 r_i 中相应的 X、Y 位置在目的区域内。最后, 步骤 1225、1211

和 1212 返回 X、Y 位置作为结果（经由步骤 1213），或返回解码程序失败的指示（经由步骤 1215）。

增强的解码和纠错程序的说明性例子

- 5 一说明性例子说明了如由组成部分 1251 和 1253 所执行的过程 1200。假设 $n=3$ ， $K=5$ ， $\mathbf{I}=(I_0 \ I_1 \ \dots \ I_6)'$ 是阶数 $n=3$ 的 m -序列。那么

$$\mathbf{A} = \begin{pmatrix} I_0 & I_1 & I_2 & I_3 & I_4 & I_5 & I_6 \\ I_6 & I_0 & I_1 & I_2 & I_3 & I_4 & I_5 \\ I_5 & I_6 & I_0 & I_1 & I_2 & I_3 & I_4 \end{pmatrix} \quad (10)$$

- 同样假设所提取比特 $\mathbf{b}=(b_0 \ b_1 \ b_2 \ b_3 \ b_4)'$ ，其中 $K=5$ ，实际上是 m -序列的第 s 、 $(s+1)$ 、 $(s+3)$ 、 $(s+4)$ 和 $(s+6)$ 比特（这些数目实际上是 m -数组长度 $L=2^n-1=2^3-1=7$ 的模数）。因此

$$\mathbf{M} = \begin{pmatrix} I_0 & I_1 & I_3 & I_4 & I_6 \\ I_6 & I_0 & I_2 & I_3 & I_5 \\ I_5 & I_6 & I_1 & I_2 & I_4 \end{pmatrix} \quad (11)$$

它是由 \mathbf{A} 的第 0、1、3、4 和 6 列组成。唯一地确定 m -数组中 b_0 的 X、Y 位置的数字 s 可在解出 $\mathbf{r}=(r_0 \ r_1 \ r_2)'$ 后计算出，所述 $\mathbf{r}=(r_0 \ r_1 \ r_2)'$ 被期望实现 $\mathbf{b}' = \mathbf{r}'\mathbf{M}$ 。由于 \mathbf{b} 中可能的误差比特，可能不能完全地实现 $\mathbf{b}' = \mathbf{r}'\mathbf{M}$ 。

- 15 过程 1200 利用以下的程序。从 \mathbf{b} 中随机地选择 $n=3$ 比特，为 $\tilde{\mathbf{b}}'_1=(b_0 \ b_1 \ b_2)$ 。求出 \mathbf{r}_1 ：

$$\tilde{\mathbf{b}}'_1 = \mathbf{r}'_1 \tilde{\mathbf{M}}_1 \quad (12)$$

其中 $\tilde{\mathbf{M}}_1$ 由 \mathbf{M} 的第 0、1 和 2 列组成。（注意， $\tilde{\mathbf{M}}_1$ 是 $n \times n$ 矩阵，且 \mathbf{r}'_1 是 $1 \times n$ 向量，这样 $\tilde{\mathbf{b}}'_1$ 是所选比特的 $1 \times n$ 向量。）

- 20 接着，计算被解码的比特：

$$\hat{\mathbf{b}}'_1 = \mathbf{r}'_1 \mathbf{M} \quad (13)$$

- 其中 \mathbf{M} 是 $n \times K$ 矩阵，且 \mathbf{r}'_1 是 $1 \times n$ 向量，使得 $\hat{\mathbf{b}}'_1$ 是 $1 \times K$ 向量。如果 $\hat{\mathbf{b}}'_1$ 等于 \mathbf{b} ，即，没有检测到误差，那么步骤 1209 确定 X、Y 位置，且步骤 1211 确定解码的位置是否在目的区域内。如果是，解码是成功的，就执行步骤 1213。否则，如图 1215 所示，解码失败。如果 $\hat{\mathbf{b}}'_1$ 不同于 \mathbf{b} ，那么在 \mathbf{b} 中检测到误差比特，就执行组成部分 1253。步骤 1217 确定集合 B_1 ，为 $\{b_0 \ b_1 \ b_2 \ b_3\}$ ，其中解码的比特与原始比特相同。因此， $\bar{B}_1 = \{b_4\}$ （对应于图 13 中的比特排列 1351）。循环次数 (lt) 被初始化为一常量，例如，100，根据应用，它是可变化的。注意，对应于 \mathbf{r}_1 的误差比特数等于

1. 接着，步骤 1221 根据方程 (7) 来更新循环次数(lt)， $lt_1 = \min(lt, 13) = 13$ 。

步骤 1217 接着从 \mathbf{b} 选择另外的 $n=3$ 比特。如果这些比特都属于 B_1 ，即 $\{b_0, b_2, b_3\}$ ，那么步骤 1219 将再次确定 \mathbf{r}_1 。为了防止这样的重复，步骤 1217 会从 \bar{B}_1 选择（举例而言）一个比特 $\{b_4\}$ ，并保留来自 B_1 的两个比特 $\{b_0, b_1\}$ 。

5 所选择的三个比特形成了 $\tilde{\mathbf{b}}'_2 = (b_0, b_1, b_4)$ 。步骤 1219 求出 \mathbf{r}_2 ：

$$\tilde{\mathbf{b}}'_2 = \mathbf{r}'_2 \tilde{\mathbf{M}}_2 \quad (14)$$

其中 $\tilde{\mathbf{M}}_2$ 由 \mathbf{M} 的第 0、1 和 4 列组成。

步骤 1219 计算 $\hat{\mathbf{b}}'_2 = \mathbf{r}'_2 \mathbf{M}$ 。找出这样的 B_2 ，例如 $\{b_0, b_1, b_4\}$ ，使得 $\hat{\mathbf{b}}'_2$ 和 \mathbf{b} 是相同的。那么 $\bar{B}_2 = \{b_2, b_3\}$ （对应于图 13 中的比特排列 1353）。步骤 1221 根据方程
10 (7) 更新循环次数(lt)。注意与 \mathbf{r}_2 相关联的误差比特的数目等于 2。代入 (7)， $lt_2 = \min(lt_1, 32) = 13$ 。

因为需要执行另一反复，步骤 1217 从 \mathbf{b} 中选择另外的 $n=3$ 比特。所选择的比特应该不都属于 B_1 或 B_2 。因此，步骤 1217，举例而言，可从 \bar{B}_1 选择一个比特 $\{b_4\}$ ，从 \bar{B}_2 选择一个比特 $\{b_2\}$ 并保留一个比特 $\{b_0\}$ 。

15 继续所述的对 \mathbf{r} 求解、比特选择和循环次数调整，直至我们不能选出任何新的 $n=3$ 比特，它们不完全属于任何先前的 B_i ，或达到了最高循环次数 lt 。

假设过程 1200 计算 5 个 $\mathbf{r}_i (i=1,2,3,4,5)$ ，其误差比特的数目分别对应于 1、2、4、3、2。（实际上，对于这个例子，误差比特数目不能超过 2，但说明性例子示出了更大数目的误差比特以说明该算法）。举例而言，步骤 1224 选择 \mathbf{r}_i 的例如
20 $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_4, \mathbf{r}_5$ ，其对应的误差比特数低于(8)中所示的 N_e 。

步骤 1224 以所选择向量 $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_4, \mathbf{r}_5$ 误差比特数的升序来对它们排序： $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_5, \mathbf{r}_4$ 。步骤 1225、1211 和 1212 从已排序的候选列表中找到第一向量 \mathbf{r} ，例如 \mathbf{r}_5 ，其对应的位置在目的区域以内。接着，步骤 1213 输出对应的位置。如果没有一个位置是在目的区域以内的，解码程序失败，如步骤 1215 所示的。

25

装置

图 14 示出了依照本发明的实施例对从所捕捉的数组提取的比特 1201 进行解码的装置 1400。装置 1400 包含比特选择模块 1401、解码模块 1403、位置确定模块 1405、输入接口 1407 和输出接口 1409。在这个实施例中，接口 1407 会从不同的
30 来源接收提取的比特 1201，所述的来源包括支持照相机 203 的模块（如图 2A 中所

示)。比特选择模块 1401 依照步骤 1203 和 1217 从提取的比特中选择 n 比特。解码模块 1403 依照步骤 1204 和 1219 解码所选择的比特(如由比特选择模块 1401 选择的, 从 K 个提取比特中选取的 n 比特), 以确定检测到的比特误差和相应的向量 \mathbf{r}_i 。解码模块 1403 将确定的向量 \mathbf{r}_i 提交给位置确定模块 1405。位置确定模块 1405 依照步骤 1209 和 1225 确定所捕捉数组的 X、Y 坐标。位置确定模块 1405 将结果提交给输出接口 1409, 如果成功, 所述的结果包括 X、Y 坐标, 如果不成功, 所述的结果为一失败指示。输出接口 1409 会将结果提交给另一模块, 所述另一模块可执行进一步的处理或可显示该结果。

10 装置 1400 可以设想不同形式的实现, 包括利用计算机可读媒质的模块和利用诸如专用集成电路(ASIC)等专门硬件的模块。

本领域的技术人员会理解, 可以利用带有包含用于控制计算机系统的指令的相关联的计算机可读媒质的计算机系统来实现这里所揭示的示例性实施例。所述计算机系统会包括至少一台计算机, 诸如微处理器、数字信号处理器和相关联的外围电子线路。

15 虽然使用所附的权利要求来定义本发明, 但这些权利要求是说明性的, 本发明意在包含这里所描述的元素和步骤的任何组合或子组合。因此, 有许多可替换组合可用于定义本发明, 它们包含来自本说明书的一个或多个元素的各种组合和子组合, 包括描述、权利要求和附图。按照本发明, 本发明各方面的可替换组合, 无论是单独或与这里所定义的一个或多个元素或步骤相组合, 可被用作本发明的修改和
20 替换, 或作为本发明的一部分, 这对于相关领域的技术人员是显而易见的。这里所包含的对本发明的书面描述意在包括所有这样的修改和替换。

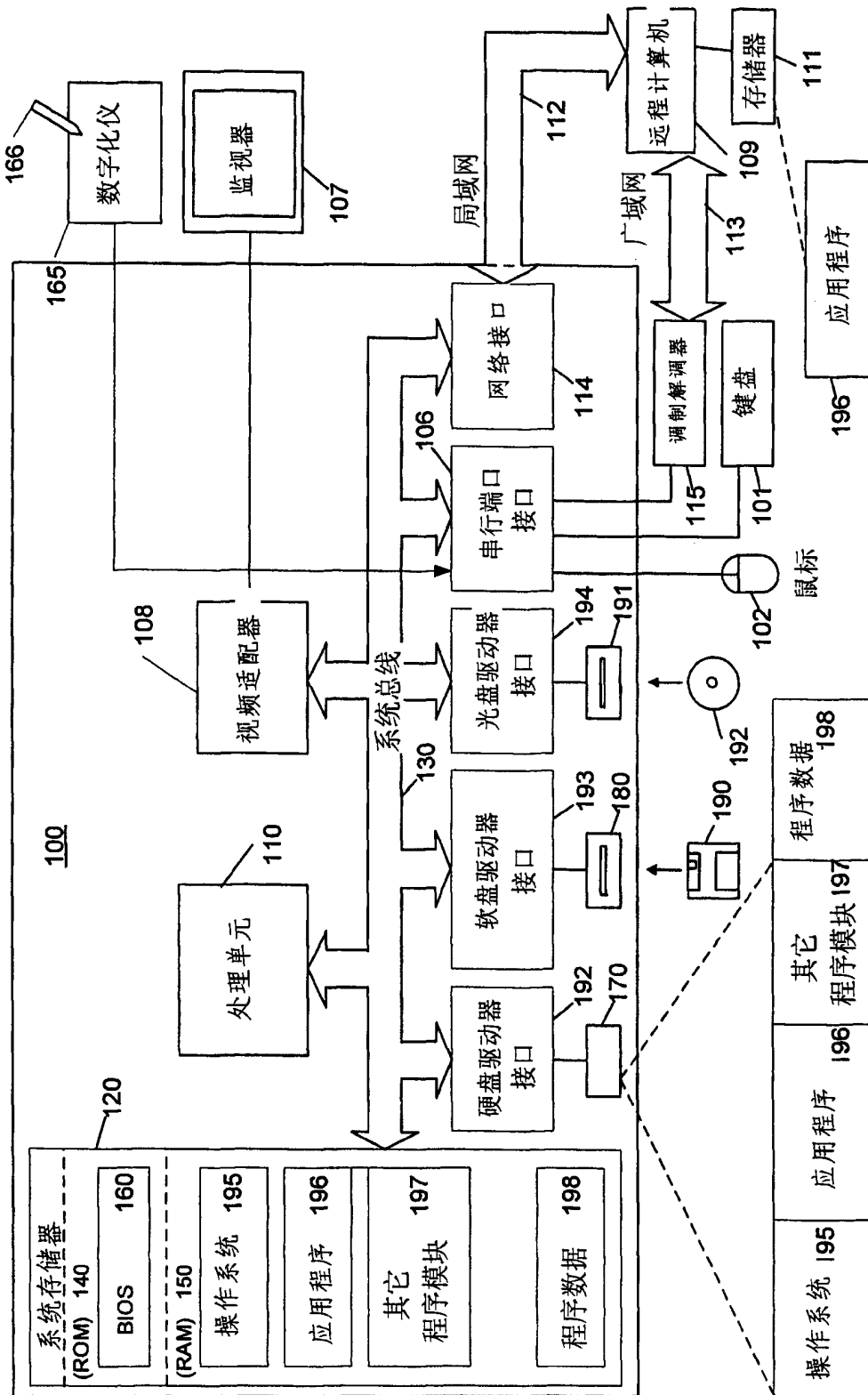


图 1

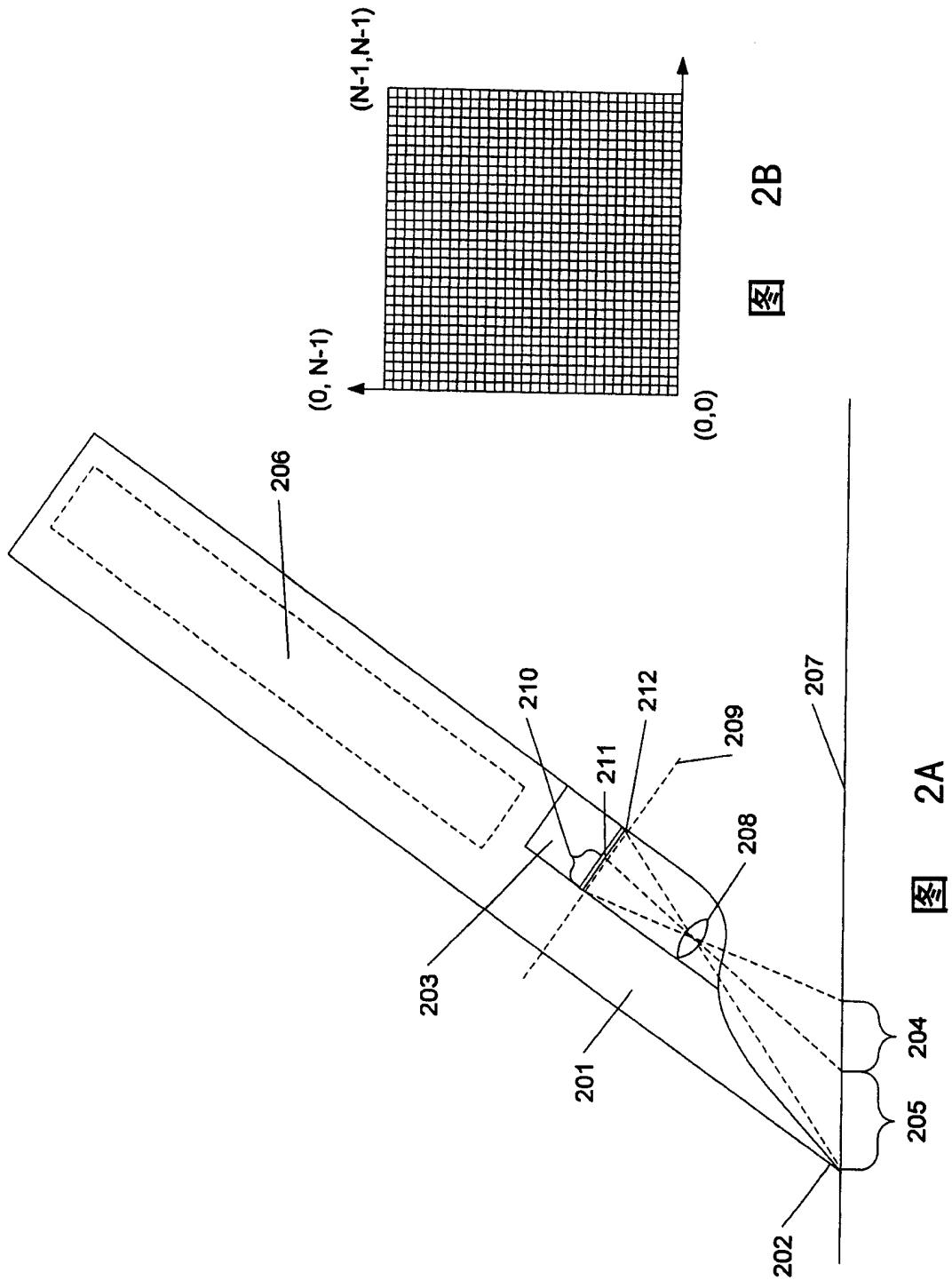


图 2B

图 2A

图 3A 0000010 00 01100 0101 001 111010 0 0111001 0 010110 111 0110 01101 01 01111111

图 3B 0000000 0011101 0100111 1110100 0111010 1000101 1110100 0100111 0011101

图 3C 000100100 00111110 010111101 010011001 011100111 001011010 011000011

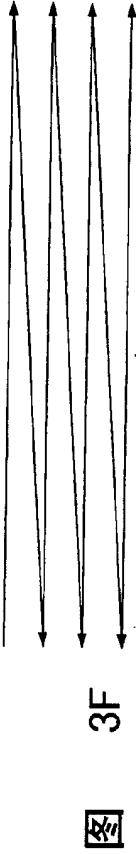
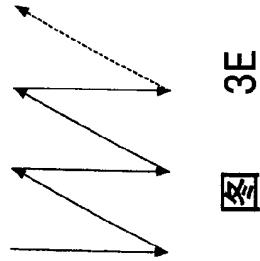
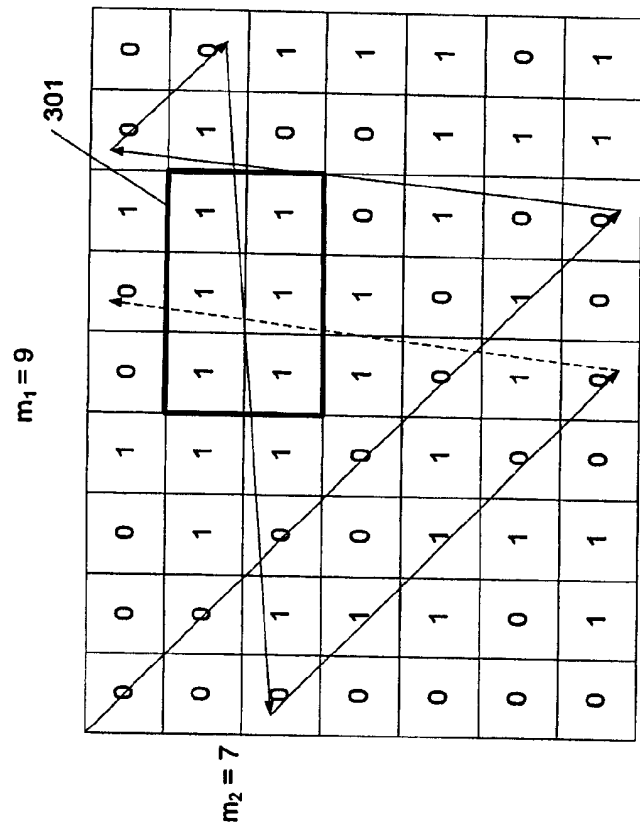


图 3D

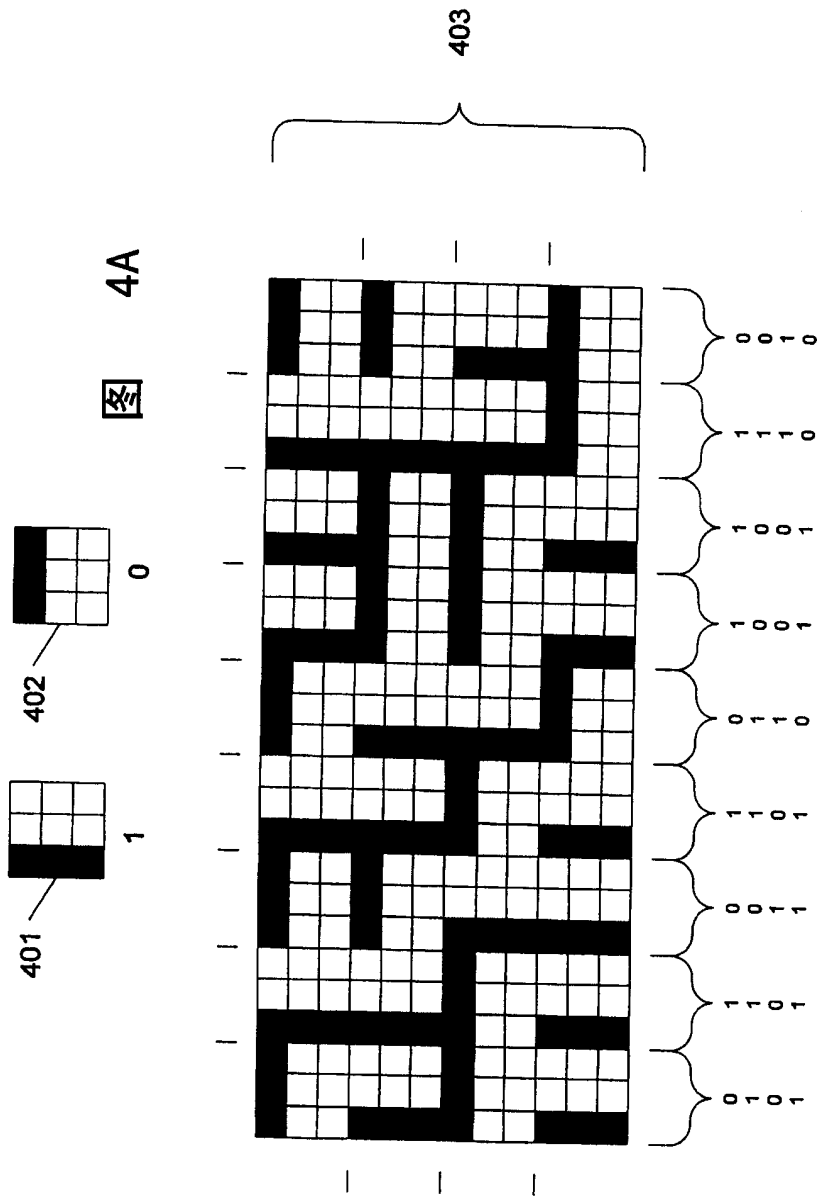


图 4B

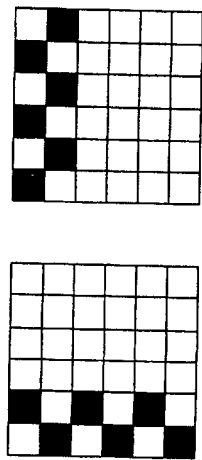
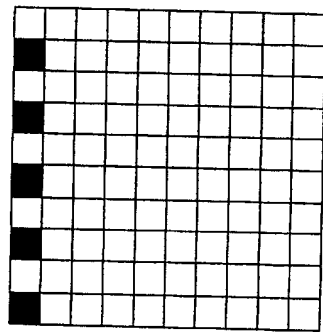
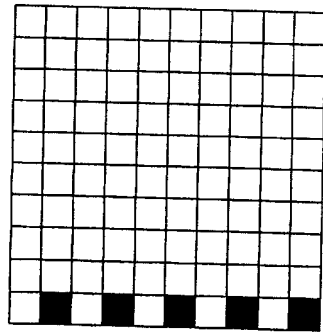


图 4C

0

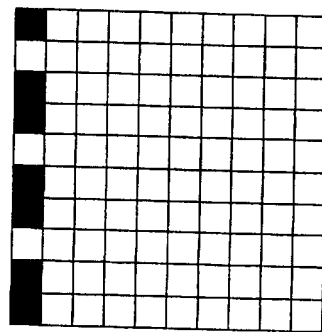


1

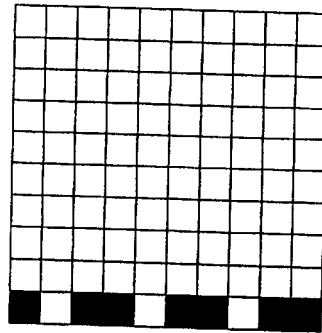


0

图 4D

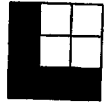


1



0

图 4E



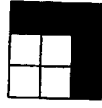
5A

图



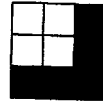
5B

图



5C

图



5D

图

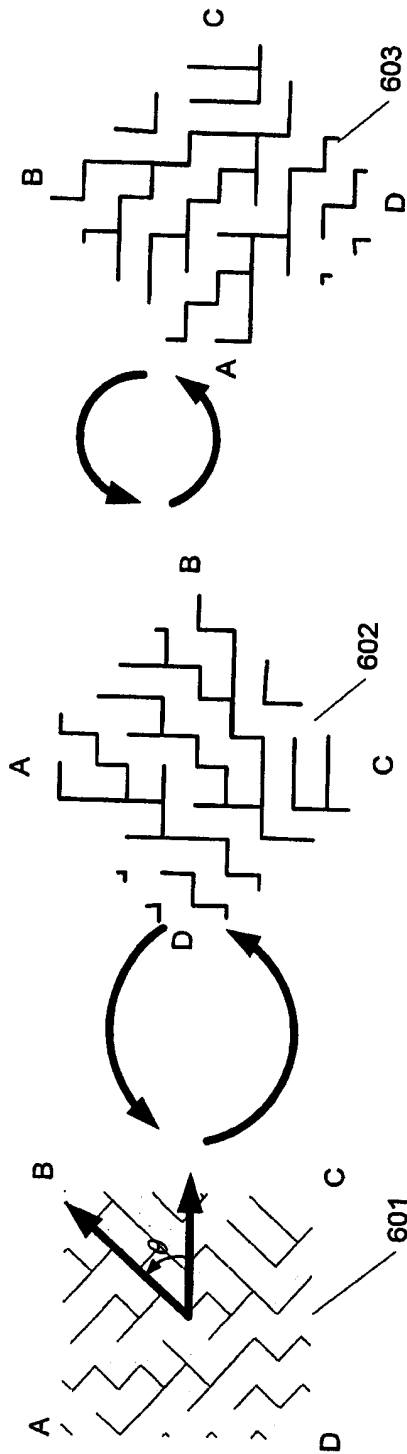
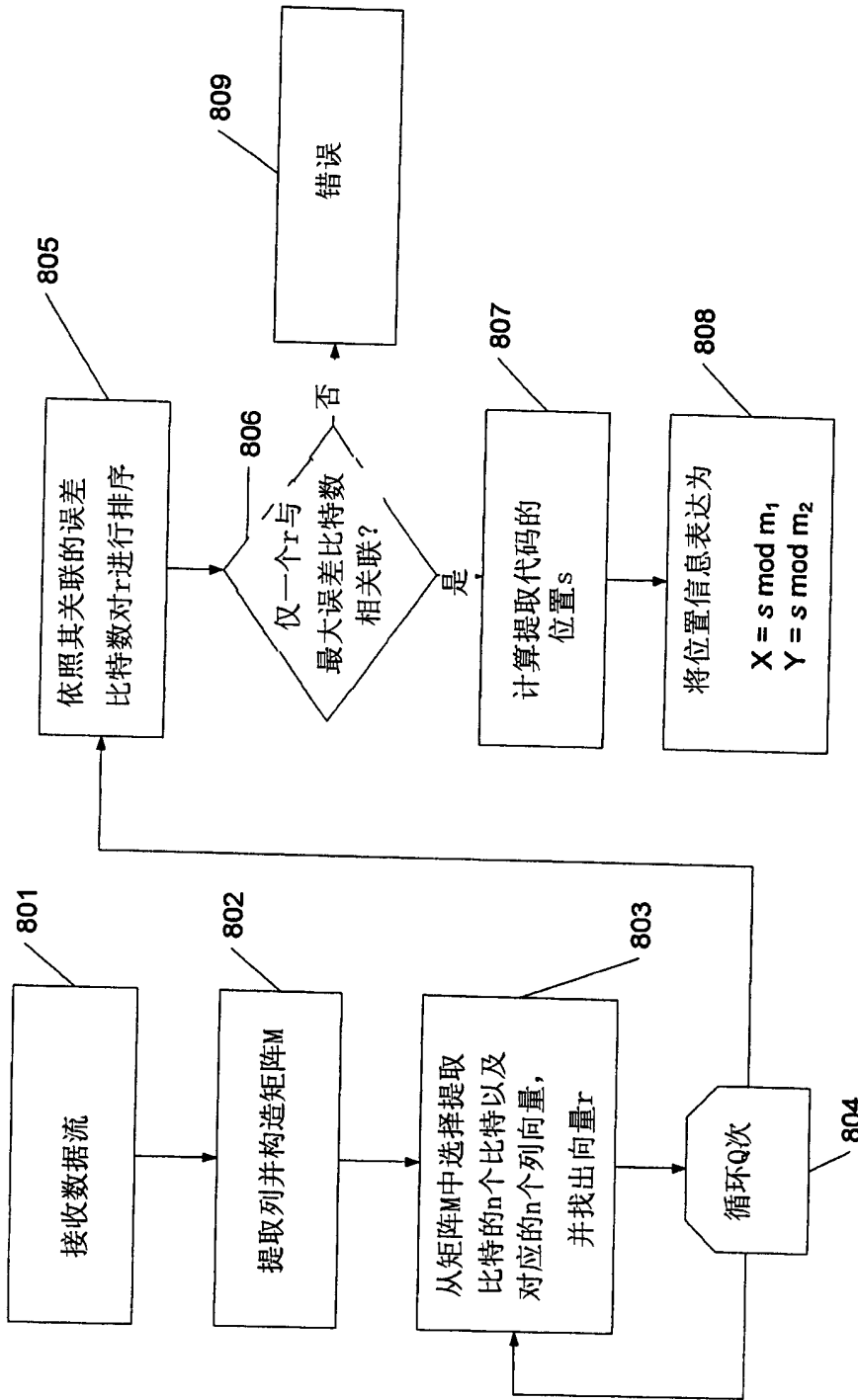


图 6

$$o = \theta + \begin{cases} 0 - no a \\ \frac{\pi - no b}{2} \\ \pi - no c \\ \frac{3\pi - no d}{2} \end{cases}$$

图 7



8

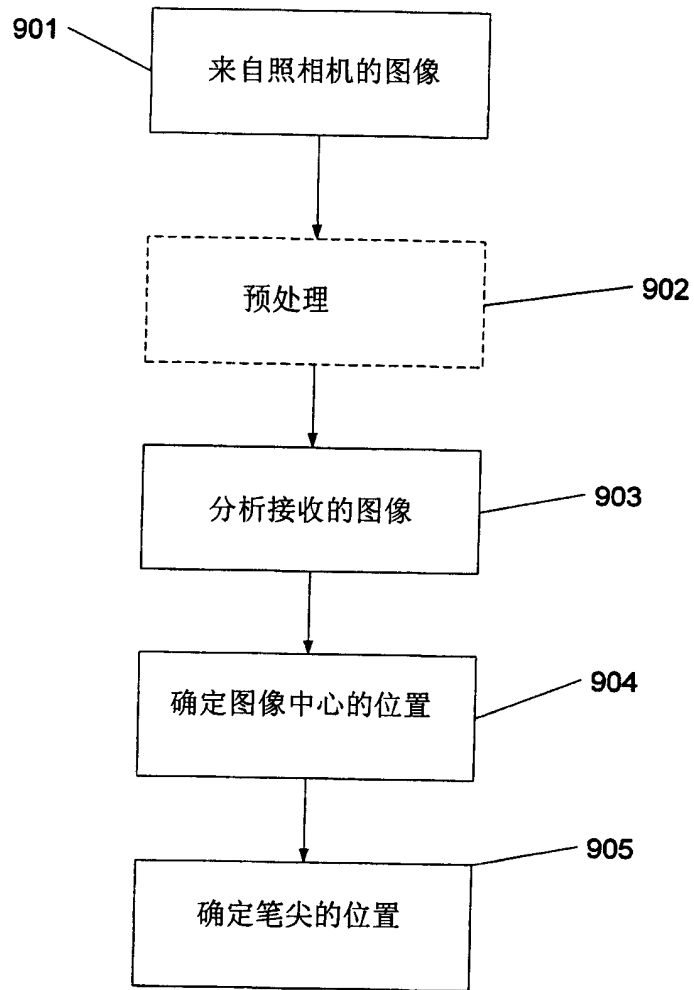


图 9

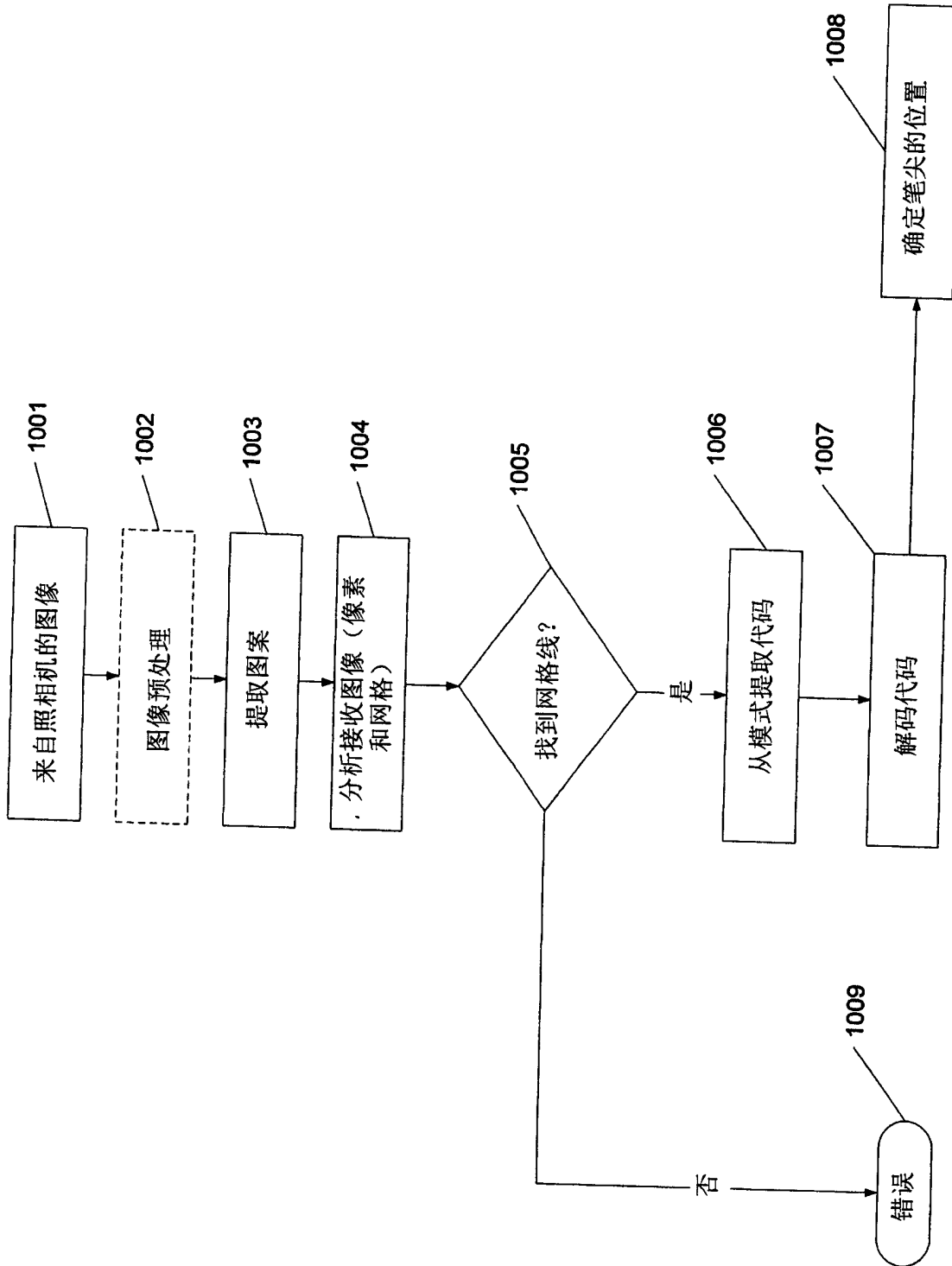


图 10

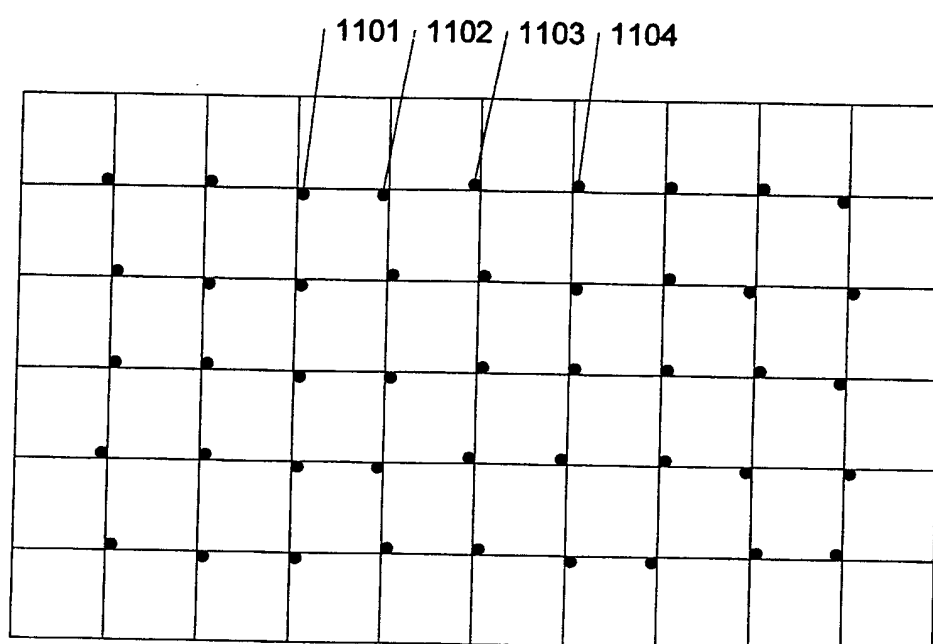


图 11

1200

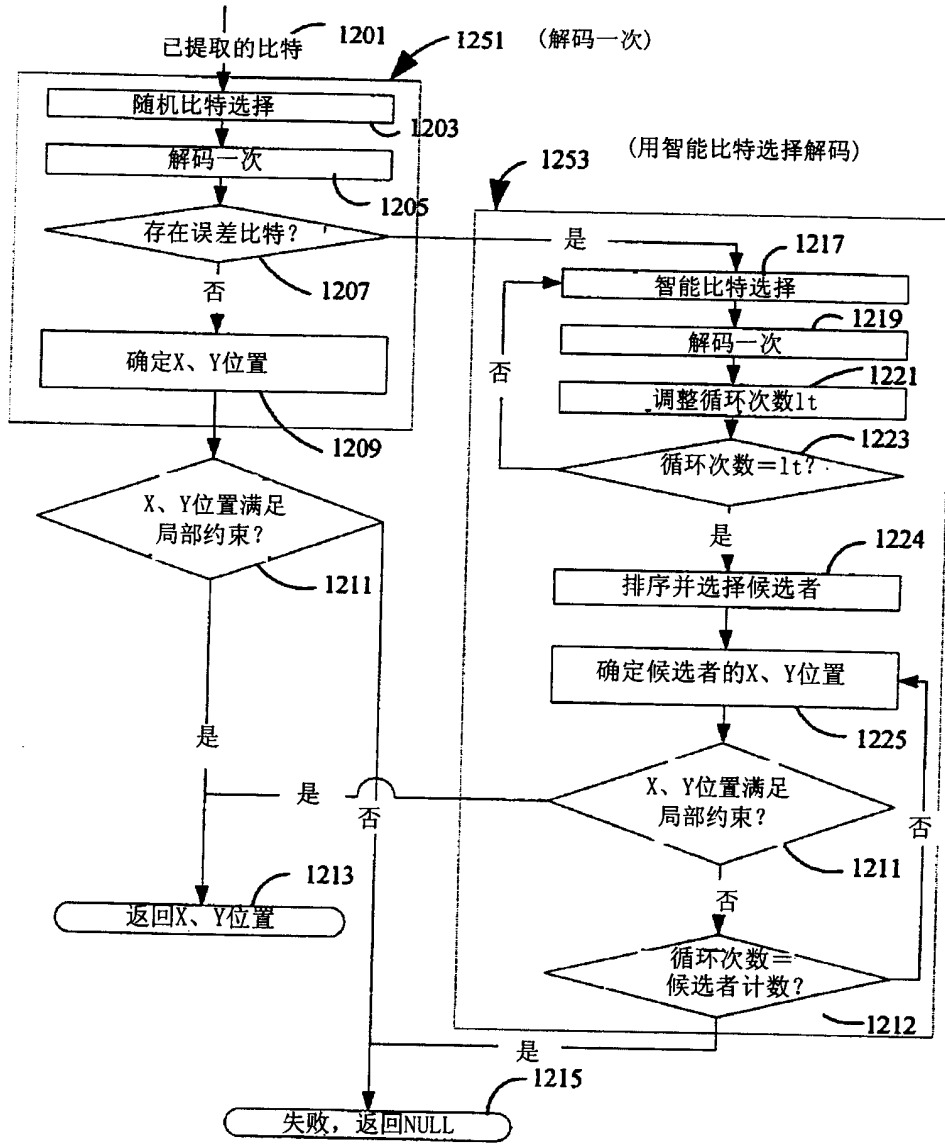


图 12

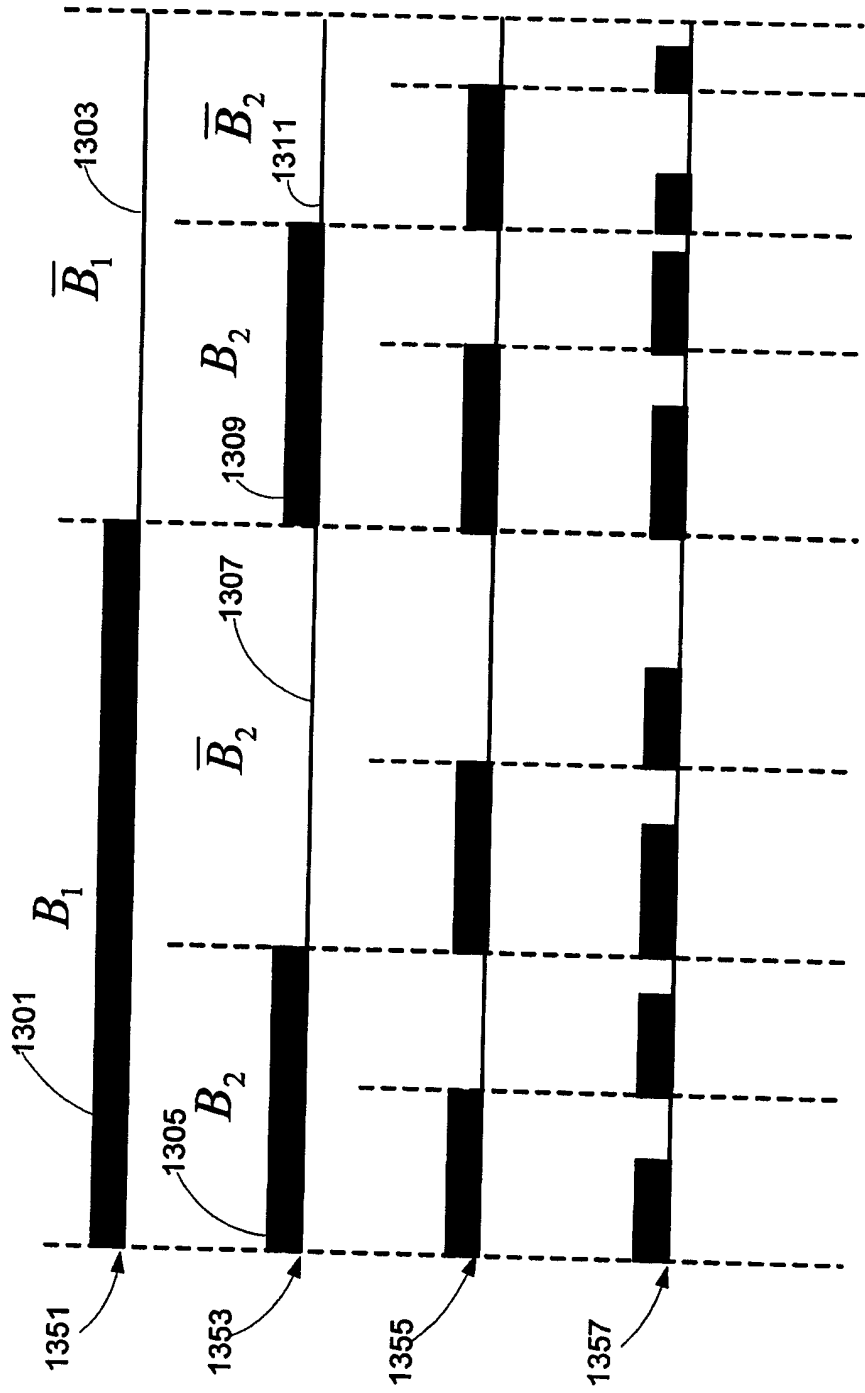


图 13

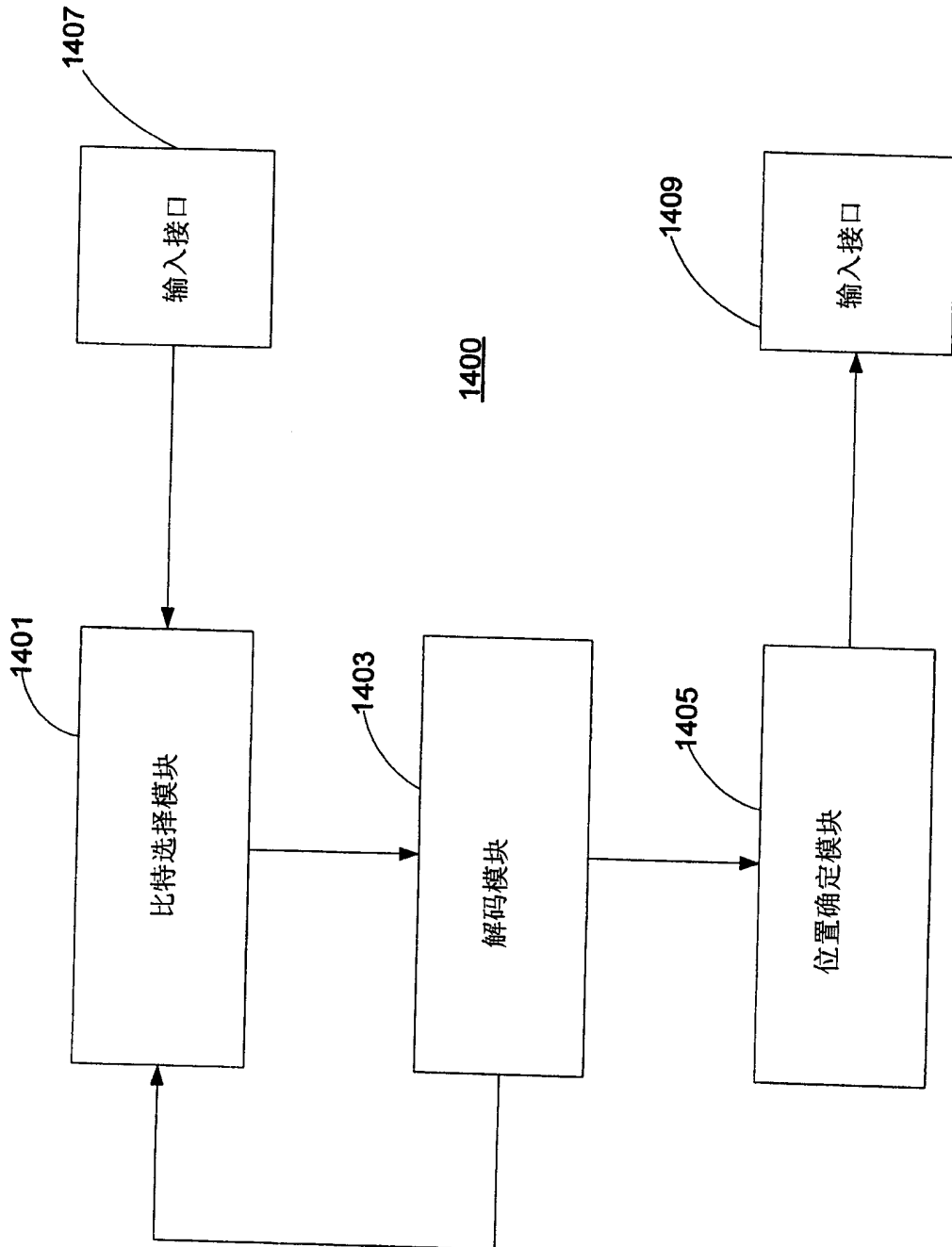


图 14