



[12] 发明专利申请公开说明书

[21] 申请号 200510006220.6

[43] 公开日 2005年7月20日

[11] 公开号 CN 1641683A

[22] 申请日 2005.1.17

[21] 申请号 200510006220.6

[30] 优先权

[32] 2004.1.16 [33] US [31] 10/759,357

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 王 坚 陈黎勇 王 强 马晓旭

党映农

[74] 专利代理机构 上海专利商标事务所有限公司

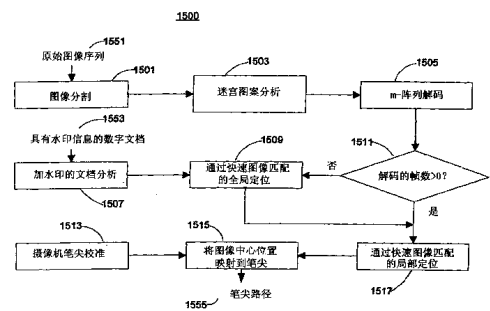
代理人 谢喜堂

权利要求书 5 页 说明书 24 页 附图 20 页

[54] 发明名称 通过 m 阵列解码和快速图像匹配的笔划定位

[57] 摘要

描述了当笔尖移过文档时确定笔尖路径的系统和方法。该文档为具有迷宫图案的水印文档，从该迷宫图案中确定编码的位置信息。利用位于笔中的摄像机来捕获图像序列。该笔尖路径是通过对该相应的迷宫图案解码和通过匹配该捕获图像与文档图像来确定的。如果与笔划相关的任何一帧的位置坐标不能根据 m - 阵列解码决定，那么这些帧被转换然后与文档图像的一个区域相配。一旦至少一帧的位置坐标被确定，其它帧的位置坐标则通过在相邻区域中匹配这些帧而进行确定。



1. 一种用于确定文档内部的笔尖路径的方法，其特征在于，所述方法包含：
- 5 (A) 对与捕获图像相关联的提取的比特解码，以便确定所述捕获图像的位置坐标；
- (B) 响应于 (A)，如果所述捕获图像的位置坐标不能通过解码确定，则将所述捕获图像与所述文档的图像信息进行匹配；以及
- (C) 将所述笔尖路径从所述捕获图像的位置坐标映射到所述笔尖的位置坐标。
- 10 2. 根据权利要求 1 所述的方法，其特征在于，所述图像信息是从加水印的文档图像信息中或者从对应于所述捕获图像的已解码位置的相邻区域的图像信息中选出的。
3. 根据权利要求 1 所述的方法，其特征在于，它还包含：
- 15 (D) 分析一迷宫图案以便从所述捕获图像中提取比特，其中，所述迷宫图案对应于一 m -阵列。
4. 根据权利要求 3 所述的方法，其特征在于，它还包含：
- (E) 从该捕获图像的一个遮盖成分中分割该迷宫图案，其中该遮盖成分遮盖该迷宫图案。
5. 根据权利要求 4 所述的方法，其特征在于，(E) 包含：
- 20 (i) 归一化所述捕获图像以便补偿不均匀的光照。
6. 根据权利要求 1 所述的方法，其特征在于，(B) 包含：
- (i) 将所述捕获图像与所述文档图像的一个区域进行比较，其中，所述区域是通过分析所述文档的图像或者一相邻图像帧的对应位置而估计的，并且其中，所述对应位置是通过 m -阵列解码或者全局定位来确定的。
- 25 7. 根据权利要求 6 所述的方法，其特征在于，(B) 还包含：
- (ii) 响应于 (i)，如果所述位置坐标不能被确定，则扭曲所述捕获图像；以及
- (iii) 将所述捕获图像与所述文档的一个区域进行匹配，其中，通过 m -阵列解码不能唯一地确定 x - y 位置。
- 30 8. 根据权利要求 7 所述的方法，其特征在于，(B) 还包含：

(iv) 响应于 (iii), 重复 (i)。

9. 根据权利要求 7 所述的方法, 其特征在于, 它还包含:

(D) 确定所述文档的区域, 其中, 所述区域被所述文档的内容所遮盖, 并且其中, 所述文档是加水印的。

5 10. 根据权利要求 1 所述的方法, 其特征在于, 它还包含:

(D) 校准所述笔尖以获得一校准参数, 其中, (C) 包含利用所述校准参数和一从局部定位中获得的透视变换。

11. 根据权利要求 10 所述的方法, 其特征在于, (D) 包含:

(i) 将所述笔尖固定在所述文档的一接触点上;

10 (ii) 改变一笔式摄像机中心的位置; 以及

(iii) 将所述笔式摄像机中心的位置映射到所述接触点。

12. 根据权利要求 10 所述的方法, 其特征在于, 所述校准参数表示一虚拟笔尖位置。

13. 根据权利要求 1 所述的方法, 其特征在于, (A) 包含:

15 (i) 获得与一捕获阵列相关联的提取比特;

(ii) 处理所述提取比特以便确定所述提取比特是否包含至少一个误差比特, 并且如果没有检测到误差比特, 则确定所述位置坐标; 以及

(iii) 如果检测到所述至少一个误差比特, 则进一步处理所述提取比特以便根据所述提取比特的一部分确定所述位置坐标,

20 其中, 所述位置坐标符合一局部约束。

14. 根据权利要求 13 所述的方法, 其特征在于, (ii) 包含:

(1) 从所述提取比特中选择第一子集;

(2) 解码所述第一子集; 以及

25 (3) 响应于 (2), 如果没有检测到误差比特, 则确定所述捕获阵列的位置坐标。

15. 根据权利要求 13 所述的方法, 其特征在于, (iii) 包含:

(1) 如果检测到一个误差比特, 则从所述提取比特中选择一个不同的子集, 其中, 所述不同子集的至少一个比特不是先前已正确解码的比特中的一个;

(2) 对所述不同子集的相关联的比特解码;

30 (3) 响应于 (2), 确定是否应该执行另一解码迭代;

(4) 如果应该执行另一解码迭代, 则从所述提取比特中选择另一子集, 其中,

- 所述其它子集的至少一个比特是从每一先前的迭代的一组错误解码的比特中选出的，并且重复（2）；以及
- （5）如果不应该执行另一解码迭代，则确定所述捕获阵列的位置坐标。
16. 根据权利要求 3 所述的方法，其特征在于，（D）包含：
- 5 （i）计算一图案参数，所述图案参数表征所述迷宫图案的网格线。
17. 根据权利要求 7 所述的方法，其特征在于，（ii）包含：
- （1）通过应用从迷宫图案分析中获得的一仿射变换来缩放和旋转一捕获的图像；以及
- （2）将所述捕获的迷宫图案对齐到一搜索区域的选择点。
- 10 18. 根据权利要求 17 所述的方法，其特征在于，（ii）还包含：
- （3）将所述扭曲的帧与一文档图像进行匹配，其中，所述扭曲的帧对应于一最大互相关值。
19. 根据权利要求 1 所述的方法，其特征在于，（C）包含：
- （i）利用一透视变换根据虚拟笔尖坐标计算笔尖位置坐标。
- 15 20. 根据权利要求 9 所述的方法，其特征在于，（D）包含：
- （i）获得一文档图像，其中，所述文档图像是加水印的。
21. 根据权利要求 20 所述的方法，其特征在于，（D）还包含：
- （ii）确定一像素的相邻窗口是否只包含迷宫图案单元，其中，所述文档图像是由多个子窗口来表示的；以及
- 20 （iii）如果所述子窗口不仅仅包含所述迷宫图案，则区分可见迷宫图案单元的一个度量。
22. 根据权利要求 21 所述的方法，其特征在于，（iii）包含：
- （1）将所述文档图像划分成实质上具有与迷宫图案单元相同大小的多个块；
- （2）如果对应的小块被文档内容遮盖，则以一像素作为相邻窗口的中心，对
- 25 在所述相邻窗口中完全可见块的数目进行计数；以及
- （3）以一表示可见块数目的指示符来标记所述像素。
23. 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可读介质。
24. 一种具有用于执行权利要求 3 所述的方法的计算机可执行指令的计算机
- 30 可读介质。
25. 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

可读介质。

26. 一种具有用于执行权利要求 9 所述的方法的计算机可执行指令的计算机可读介质。

27. 一种具有用于执行权利要求 10 所述的方法的计算机可执行指令的计算机可读介质。

28. 一种确定文档内部的笔尖路径的装置，其特征在于，它包含：

一解码模块，它对与一捕获图像相关联的提取比特解码；

一匹配模块，如果所述捕获图像的位置坐标不能由所述解码模块来确定，则它将所述捕获图像与所述文档的图像信息进行匹配；以及

10 一映射模块，它根据所述捕获图像的位置坐标映射所述笔尖路径，并且提供路径信息。

29. 根据权利要求 28 所述的装置，其特征在于，它还包含：

一校准模块，它根据笔相对于所述笔尖的移动信息确定一校准参数，其中，所述映射模块使用一变换和所述校准参数连同所述捕获图像的位置坐标一起来映
15 射所述笔尖的路径。

30. 根据权利要求 28 所述的装置，其特征在于，它还包含：

一加水印文档分析器，它确定文档的一个区域并提供有关所述区域的图像信息，其中，所述区域被所述文档的内容所遮盖，并且其中，所述文档是加水印的。

31. 根据权利要求 28 所述的装置，其特征在于，它还包含：

20 一迷宫图案分析器，它提取与所述捕获图像的迷宫图案相关联的提取比特，并且向所述解码模块提供所提取的比特。

32. 根据权利要求 31 所述的装置，其特征在于，它还包含：

一图象分割模块，它从所述捕获图像的一遮盖成分分割所述迷宫图案，其中，所述遮盖成分包含遮盖所述迷宫图案的文档内容。

25 33. 一种用于确定文档内部的笔尖路径的方法，其特征在于，所述方法包含：

(A) 校准所述笔以便获得一校准参数；

(B) 分析一加水印的文档图像，以便确定其中 $x - y$ 位置不能由 m -阵列解码来唯一确定的区域；

30 (C) 从捕获图像的一遮盖成分中分割一迷宫图案，其中，所述遮盖成分包含遮盖所述迷宫图案的文档内容。

(D) 分析所述迷宫图案以便从一捕获图像的迷宫图案中提取比特；

(E) 响应于 (D)，确定一将图像位置坐标变换为实际位置坐标的变换，其中，所述实际位置坐标标识了与所述文档相关联的笔尖位置；

(F) 对与所述捕获图像相关联的提取比特进行解码；

5 (G) 响应于 (F)，如果所述捕获图像的图像位置坐标不能被确定，则将所述捕获图像与图像信息进行匹配，包含：

(i) 扭曲所述捕获图像；以及

(ii) 将所述扭曲的图像与所述文档的一个区域进行匹配，其中，对应的 $x - y$ 位置不能唯一地由 m -阵列解码确定；

(H) 确定与一笔划相关联的其它图像帧的图像位置坐标，包含：

10 (i) 将一对应的捕获图像与先前已解码的图像的图像位置坐标的一相邻区域进行匹配；以及

(ii) 重复 (i)，直至所述笔划的所有图像帧都被处理过；以及

(I) 根据所述变换、所述校准参数和所述图像位置坐标映射所述笔尖路径。

通过 m 阵列解码和快速图像匹配的笔划定位

5 技术领域

本发明涉及利用一数字笔对介质进行交互。更特别地，本发明涉及在与一个或多个表面相互作用期间确定数字笔的位置。

背景技术

10 计算机用户习惯于利用鼠标和键盘作为与个人计算机交互的方式。虽然个人计算机对于书面文档提供有若干种优势，但大多数用户继续利用打印纸张完成某些功能。这些功能包括读取并注释书面文档。就注解而言，由于此用户对该打印文档的批注，则此打印文档呈现更大的意义。然而，困难之一是，对于具有批注的打印文档来说，该批注需要输入为电子形式文档。这就要求原始用户或者另一用户费

15 劲地看完这些批注，并将它们输入到个人计算机里。在某些情况下，一用户会扫描这批注和这原文，从而创建一个新建文档。这种多个步骤使打印文档和电子版本文档之间的交互很难在一重复基础进行处理。此外，扫描输入图像通常是不可修正的。可能没有办法从原文中分隔这些批注。这使利用这些批注变得困难。从而，需要一种改进的处理批注的方法。

20 一种捕获手稿信息的技术是借助于一支笔，其位置在书写期间可以被确定。提供这种能力的笔是 Anoto 公司的 Anoto 笔。这种笔是通过利用摄象机来捕获以预定图案编码的纸张图像而起作用的。这种图像图案的一个示例如图 11 所示。此模式使用 Anoto 笔（Anoto 公司制造）确定笔在一张纸上的位置。然而，使用 Anoto 笔的系统来确定位置有怎样的效率是不清楚的。为了提供高效的确定捕获图像的位置，

25 需要提供一种高效的捕获图像解码的系统。

当复查文档时，针对该文档，用户可以通过移动笔尖来注解该文档。此笔尖的路径可以包含多个笔划，其中每个笔划对应于一系列捕获图像。因此，在本行业中真正需要识别笔的路径以便对文档进行处理。

30 发明内容

本发明的各方面提供了当笔尖横移过文档时测定笔尖路径的系统和方法。对于本发明的一方面来说，文档用迷宫图案（maze pattern）加水印，根据该迷宫图案确定编码位置信息。

5 本发明的另一方面，通过位于笔中的摄影机来捕获一序列的图像。笔尖路径是通过对相关迷宫图案(m-阵列)解码和通过将捕获的图像与文档图像进行匹配来确定的。

本发明的另一个方面，如果与笔画相关的任何帧（对应于一捕获图像）的位置坐标不能根据 m-阵列解码确定，那么这些帧被转换并且然后与文档图像的一个区域进行匹配。一旦至少一帧的位置坐标被确定，另外一些帧的位置坐标就可以通过匹配相邻区域中的这些帧来确定。

10 本发明的另一方面，笔尖路径（对应于一笔划）是利用一透视变换和校准参数从一帧中心中来映射的。透视变换是从迷宫图案分析中并通过将摄像机捕获的图像与文档图像进行匹配所获得的。

15 附图说明

当结合附图阅读本发明时，上述概述以及以下优选实施例的详细说明将更容易理解，附图以示例方式包括，而不是对所要求发明的限制。

图 1 表示可以用于结合本发明实施例的计算机的一般说明。

20 附图 2A 和 2B 表示根据本发明实施例的一个图像捕获系统和相应的捕获图像。

附图 3A 至 3F 表示根据本发明实施例的各种序列和折叠技术。

附图 4A 至 4E 表示根据本发明实施例的各种编码系统。

附图 5A 至 5D 表示与根据图 4A 和 4B 的编码系统相关的四个可能的合成角。

图 6 表示根据本发明实施例的捕获图像部分的旋转。

25 图 7 表示结合图 4A 至 4E 的编码系统使用的各种旋转角。

图 8 表示根据本发明实施例用于确定捕获阵列位置的过程。

图 9 表示根据本发明实施例用于确定捕获图像位置的方法。

图 10 表示根据本发明实施例用于确定捕获图像位置的另一种方法。

图 11 表示在根据现有技术文档中编码间隔的表述。

30 图 12 表示根据本发明实施例用于从确定捕获图像中解码提取的比特的一个流程图。

- 图 13 表示根据本发明实施例从捕获图像中提取比特的比特选择。
- 图 14 表示根据本发明实施例用于从捕获图像中解码提取的比特的一个装置。
- 图 15 表示根据本发明实施例用于从捕获帧序列中确定笔尖路径的一种方法。
- 图 16 表示根据本发明实施例用于从捕获帧序列中确定笔尖路径的一个装置。
- 5 图 17 表示根据本发明一个实施例的捕获图像的一个示例。
- 图 18 表示根据本发明一个实施例的包含文本成分的捕获图像的一个示例。
- 图 19 表示根据本发明一个实施例的捕获图像的一个迷宫图案分析。
- 图 20 表示根据本发明的一个实施例，包含文本成分的捕获图像的一个迷宫图案分析。
- 10 图 21 表示根据本发明的一个实施例，分析文档图像的一个结果。
- 图 22 表示根据本发明的一个实施例，用于一个典型笔划的全局定位结果。
- 图 23 表示根据本发明的一个实施例，用于一个典型笔划的局部定位结果。
- 图 24 表示根据本发明的一个实施例，重新获得的笔尖笔划和对应捕获图像中心的路径。
- 15 图 25 表示对应于图 24 所示的重新获得的笔划的一个实际笔划。

具体实施方式

本发明的一些方面涉及确定与较大图像有关的一个捕获图像的位置。在此描述的位置确定方法和系统可以结合多功能笔使用。

- 20 以下内容以副标题分隔以便于读者得益。副标题包括：术语、通用计算机、图像捕获笔、阵列编码、解码、纠错、位置确定、m -阵列解码、和用于确定笔的路径的体系结构。

术语

- 25 笔 (Pen) - 可以包括或者不包括能够存储墨水的任何写入工具。在某些例子中，根据本发明的实施例，没有墨水能力的可以用作笔使用。

摄像机 (Camera) - 可以从纸张或者任何其他介质中捕获图像的一种图像捕获系统。

通用计算机

- 30 图 1 是常规通用数字计算环境示例的功能方块图，该数字计算环境可用来实

现本发明的多种方式。在图1中，计算机100包括处理单元110、系统存储器120和系统总线130，系统总线130将包括系统存储器的多种系统部件耦合到处理单元110。系统总线130可以是总线结构类型的任何几种类型，这些总线结构包括存储器总线或者存储控制器、外围总线和利用任何各种总线体系结构的本地总线。系统存储器120包括只读存储器(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ROM140中，包含有助于在计算机100内部元素之间传送信息的基本例行程序，诸如在启动期间。计算机100还包括硬盘驱动器170，用于从硬盘(未显示)读取和写入到硬盘；磁盘驱动器180，用于从可移动磁盘190中读取或者写入到可移动磁盘190；和一个光盘驱动器191，用于从可移动光盘192，诸如CD ROM或者其他光介质中读取或者写入到该可移动光盘192。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。这些驱动器和其相关的计算机可读介质提供计算机可读指令的非易失存储器、数据结构、程序模块和此个人计算机100的其他数据。本领域技术人员将意识到，在此示例操作环境中也可以使用能够储存计算机可存取数据的其他类型的计算机可读介质，诸如磁带盒、闪存卡、数字视频盘、Bernoulli盒式磁盘、随机存取存储器(RAM)、只读存储器(ROM)等等。

很多程序模块能够存储在硬盘驱动器170、磁盘190、光盘192、ROM140或者RAM150中，包括操作系统195、一个或多个应用程序196、其他程序模块197和程序数据198。用户能够经由输入设备诸如键盘101和定点设备(pointing device)102将命令和信息输入到计算机100里。其他输入设备(未显示)可以包括麦克风、操纵杆、游戏垫、卫星盘(satellite dish)、扫描器等等。这些和其他输入设备一般经由耦合到此系统总线的串行口接口106连接到处理单元110，但也可以通过其他接口诸如并行端口、游戏端口或者通用的串行总线(USB)连接。此外，这些设备仍然可以经由适当的接口(未显示)直接耦合到系统总线130。监视器107或者其他类型的显示设备也经由接口诸如视频适配器108连接到此系统总线130。除了此监视器外，个人计算机典型地包括其他外围输出设备(未显示)，诸如扬声器和打印机。在一优选实施例中，提供数字器165和伴随笔或者触针166以便于数字地捕获手画线输入。虽然显示了此笔数字器165和此串行口之间的直接连接，但实际上，笔数字器165可以经由并行端口或者其他接口和作为已知技术的130直接耦合到处理单元110。此外，虽然数字器165显示为与监视器107分离开，

但最好数字器 165 的可用输入区与监视器 107 的显示区是共同延伸的。此外，数字器 165 仍然可以是集成到监视器 107 中，或者可以作为重叠的分离装置或者在某些方面附加到监视器 107 中的形式而存在。

计算机 100 能够利用与一个或多个远程计算机诸如远程计算机 109 的逻辑连接，在网络环境下操作。远程计算机 109 可以是一个服务器、路由器、网络 PC、同级设备或者其他公用网络节点，典型地包括如上所述的有关计算机 100 的许多或者全部元素，虽然在图 1 中说明只有一个存储器设备 111。图 1 描绘的逻辑连接包括局域网(LAN) 112 和广域网(WAN) 113。这种网络环境在办公室、企业宽带计算机网络、内部网和因特网中是非常普遍的。

当在 LAN 网络环境使用时，计算机 100 经由网络接口或者适配器 114 连接到本地网络 112。当在 WAN 网络环境使用时，个人计算机 100 典型地包括一个调制解调器 115 或者其他装置，用于建立对此广域网 113 诸如因特网的通信。调制解调器 115 可以位于内部或者外部，经由串行口接口 106 连接到系统总线 130。在网络环境中，相对于个人计算机 100 或者其部分描述的程序模块可以存储在远程存储器存储设备中。

将意识到所示的网络连接只是说明性的，可以使用在计算机之间建立通信链路的其他技术。假定存在任何多种众所周知的协议，诸如 TCP/IP、Ethernet、FTP、HTTP，Bluetooth、IEEE 802.11x 802.11x 等等，并且该系统能够在客户-服务器配置下操作以便允许用户从基于服务器的 web 中检索网页。可以使用多种常规浏览器的任何一种来显示和操作网页上的数据。

图像捕获笔

本发明的特征包括将一个编码数据流放置在代表该编码数据流的显示的构成中。（例如，正如图 4B 将讨论的，该编码数据流用来创建一个图解图案）。该显示方式可以是打印的纸张(或者其他实际媒体)或者可以是投影连带另一个图像或者一套图像的编码数据流的显示器。例如，该编码数据流可以表示为纸张上的物理图解图像或者覆盖该显示图像的图解图像(例如代表文档文字)，或者可以是在显示器屏幕上的物理(不可修正的)图解图像(因此由笔捕获的任何图像部分位于该显示屏上)。

此捕获图像的位置确定可以用来确定用户与纸张、介质或者显示屏的交互作用的位置。在本发明的某些方面，此笔可以是一个在纸上书写的墨水笔。在其他

方面,此笔可以是用户在计算机显示器表面上书写的一个触针。借助于此文档的编码图像的知识或者支持显示在计算机屏幕上的文档,任何交互作用可以是提供回给此系统。当笔或者触针划过文档时,通过借助于此笔或者触针中的摄像机来重复地捕获图像,此系统能够跟踪由用户控制的此触针的运动。此显示的或者打印的图像可以是与空白或者富含内容的纸张有关的一种水印,或者可以是与显示的图像有关的或者与覆盖屏幕或内置在屏幕内的固定编码有关的一个水印。

图 2A 和 2B 表示具有摄像机 203 的笔 201 的一个说明性示例。笔 201 包括笔尖 202,可以包括或者不包括墨水储存器。摄像机 203 从表面 207 捕获图像 204。笔 201 此外可以包括辅助传感器和/或处理器,如在虚线框 206 所示的。这些传感器和/或处理器 206 可以同时包括向另一个笔 201 和/或个人计算机发送信息的能力(例如,经由 Bluetooth 或者其他无线协议)。

图 2B 表示由摄像机 203 观察的图像。在一个说明性的示例中,摄像机 203 的视觉字段(field of view)(即此摄像机的图像传感器的分辨率)是 32x32 像素(其中 $N = 32$)。在此实施例中,捕获图像(32 像素乘 32 像素)对应于摄像机 203 捕获的表平面大约 5 毫米乘 5 毫米的一个区域。从而,图 2B 表示 32 像素长乘以 32 像素宽的一次视场。N 的大小可调整,以致于较大的 N 对应于较高的图像分辨率。同时,虽然用于说明目的将摄像机 203 的视场显示为方形,此视场可以包括已知技术中的其他形状。

摄像机 203 捕获的图像可以定义为图像帧 $\{I_i\}$ 的一个序列,其中 I_i 在抽样时间 t_i 由笔 201 捕获。采样率可大可小,取决于系统配置和性能要求。此捕获帧的大小可大或小,取决于系统配置和性能要求。

摄像机 203 捕获的图像可以由处理系统直接使用,或者可以经受前置滤波。此前置滤波可以发生在笔 201 内或者可以发生在笔 201 外(例如,在个人计算机中)。

图 2B 的图像尺寸是 32x32 像素。如果每个编码单元尺寸是 3x3 像素,那么捕获的编码单元数目将大约为 100 个单元。如果每个编码单元尺寸是 5x5 像素,那么捕获的编码单元数目大约为 36。

图 2A 还表示图像平面 209,在其上图案图像 210 从位置 204 中形成。在目标平面 207 上,从此图案接收的光由 208 聚焦。透镜 208 可以是一个透镜或者是多部件透镜系统,但为简单起见这里以单镜头为代表。图像捕获传感器 211 捕获此图像 210。

图像传感器 211 可以足够大以捕获图像 210。做为选择,图像传感器 211 可以

足够大到捕获笔尖 202 在位置 212 处的图像,在位置 212 处的图像被称为虚拟笔尖 (pen tip)。注意到由于笔尖、透镜 208 和图像传感器 211 之间的恒定关系,此虚拟笔尖位置相对于图像传感器 211 是固定的。

以下换算 (transformation) $F_{S \rightarrow P}$ 将摄像机捕获的图像的位置坐标变换为纸张上实际图像的位置坐标:

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

在书写时,笔尖和纸张在同一平面上。从而,从虚拟笔尖到实际笔尖的转换也是 $F_{S \rightarrow P}$:

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

10 换算 $F_{S \rightarrow P}$ 可以估算为一个仿射变换 (affine transform), 其简化为:

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

对于估算 $F_{S \rightarrow P}$, 其中 θ_x , θ_y , s_x , 和 s_y 是在位置 204 捕获的图案的两个方向上的旋转和比例等级。此外,通过匹配捕获的图像与纸上对应的实际图像,可以细化 $F'_{S \rightarrow P}$ 。“细化 (Refine)”意思是利用一种被称为递归方法的最佳算法类型来得到对换算 $F_{S \rightarrow P}$ 更准确的估算。此递归算法将矩阵 $F'_{S \rightarrow P}$ 作为初始值对待。此细化的估算描述 S 和 P 之间的换算更为准确。

下面,通过校准可以确定虚拟笔尖的位置。

20 把笔尖 202 放在纸上固定位置 L_{penip} 。接着,使笔倾斜,借助于笔的不同姿势允许摄像机 203 捕获一系列图像。对于每个捕获的图像,可以获得换算 $F_{S \rightarrow P}$ 。从这个换算中,可以获得虚拟笔尖的位置 $L_{virtual-penip}$:

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

其中, L_{penip} 以(0,0)开始,并且

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

25 将从每个图像获得的 $L_{virtual-penip}$ 进行平均,可以确定虚拟笔尖的位置 $L_{virtual-penip}$ 。借助于 $L_{virtual-penip}$,可以得到更准确的 L_{penip} 估算。反复几次之后,可以确定更准确的虚拟笔尖位置 $L_{virtual-penip}$ 。

现在已经知道了虚拟笔尖的位置 $L_{virtual-penip}$ 。也可以从捕获的图象中获得换

算 $F_{S \rightarrow P}$ 。最终，可以使用此信息来确定实际笔尖的位置 L_{penip} ：

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

阵列解码

- 5 通过折叠一维序列可以构成二维阵列。包含足够大数目比特的二维阵列的任何部分可以用来确定其在完全二维阵列中的位置。然而，可能有必要从一个捕获的图象或者几个捕获的图象中来确定位置。至于要最小化与两个或多个二维阵列位置相关的捕获图像部分的可能性，可以使用非重复序列来创建该阵列。所创建序列的一个性质是该序列在在长度（或窗口） n 上不重复。以下描述了一维序列的创建，
- 10 然后是该序列折叠为一个阵列。

序列结构

- 一个序列号可以用作编码系统的起点。例如，一个序列（也被称为 m -序列）可以表示为以字段 F_q 设置的 q -元素。这里， $q=pn$ 其中 $n \geq 1$ ， p 是最初的序号。
- 15 该序列或 m -序列可以利用多种技术产生，包括但不限于多项式除法。利用多项式除法，该序列可以定义为以下：

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

- 其中 $P_n(x)$ 是字段 $F_q[x]$ 的 n 次本原多项式（具有 q^n 个元素）。 $R_l(x)$ 是字段 $F_q[x]$ 的 l 次非零多项式（其中 $l < n$ ）。可以使用两步骤迭代流程创建此序列：第一，两个多项式相除（结果是字段 F_q 的一个元素），第二，乘以余数 x 。当输出开始重复时，计算停止。此处理可以利用 Douglas W. Clark and Lih-Jyh Weng 在一篇文章中撰写的“Maximal and Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms,” IEEE Transactions on Computers 43.5 (May 1994, pp 560-568) 线性反馈移位寄存器来实现。在此环境下，建立序列循环移位与
- 20 多项式 $R_l(x)$ 之间的关系：仅仅循环移位该序列来改变 $R_l(x)$ 并且每一个循环移位对应于一个多项式 $R_l(x)$ 。合成序列的特性之一是，该序列具有 $q^n - 1$ 周期，并且在
- 25 一个周期内，宽度（或长度）为 n ，在该序列中任何部分存在一次而且只存在一次。这被称为“窗口特性”。周期 $q^n - 1$ 也被称为序列长度， n 作为该序列的阶数。

上述处理只是可以用来创建窗口特性序列的多种处理中的一种。

阵列结构

可以用来创建图像（其中部分可以由摄像机捕获）的阵列（或者 m -阵列）是一维序列或 m -序列的扩展。令 A 是周期 (m_1, m_2) 的一个阵列，即
 5 $A(k+m_1, l) = A(k, l+m_2) = A(k, l)$ 。当 $n_1 \times n_2$ 窗口在周期 A 移位时，所有非零 $n_1 \times n_2$ 矩阵在 F_q 出现一次而且只有一次。此特性也被称为“窗口特性”其中每个窗口是唯一的。

通过折叠该阵列可以得到二进制阵列（或 m -阵列）。一种方案是得到一个序列然后将之折叠为 $m_1 \times m_2$ 大小，其中该阵列的长度是 $L = m_1 \times m_2 = 2n - 1$ 。作为选择方式，可以以希望覆盖的预定间隔开始（例如一片纸，30 片纸或计算机监视器大小的纸）确定区域 $(m_1 \times m_2)$ ，然后使用该大小令 $L \geq m_1 \times m_2$ ，其中 $L = 2n - 1$ 。
 10

可以使用多种不同的折叠技术。例如，图 3A 到 3C 表示 3 个不同的序列。每个序列可以折叠为图 3D 所示的阵列。3 个不同的折叠方法显示为在图 3D 中有重叠，而且在图 3E 和 3F 中有光栅路径。我们采用图 3D 所示的折叠方法。
 15

为了创建图 3D 所示的折叠方法，创建长度为 L 和 n 阶的序列 $\{a_i\}$ 。接着，大小为 $m_1 \times m_2$ 的阵列 $\{b_k\}$ ，其中 $\gcd(m_1, m_2) = 1$ and $L = m_1 \times m_2$ 是从序列 $\{a_i\}$ 中通过令该阵列的每个比特为等式 1 所计算的结果：

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

20 此折叠方法可以另外表示为将序列布置在此阵列的对角线上，然后当到达一个边缘时从相反边缘继续。

图 4A 表示可以用来编码图 3D 阵列的抽样编码技术。意识到可以使用其他编码技术。例如，可以使用图 11 所示的替换编码技术。

参考图 4A，由深色墨水列代表 401 (例如 "1")。由深色墨水行代表第二比特
 25 402 (例如 "0")。将意识到任何彩色油墨可以被用来代表多个比特。选择墨水的唯一必要条件是提供与此介质背景有效的对比，以便由图像捕获系统可区分。图 4A 的比特由 3×3 矩阵单元所代表。矩阵的大小可以基于图像捕获系统的大小和分辨率修改为具有任何大小。比特 0 和 1 的替换表述如图 4C - 4E 所示。会意识到用于图 4A - 4E 的抽样编码的 1 或 0 的表述可以不受影响地相互转换。图 4C
 30 表示以交织排列的占用两行或者两列的比特表述。图 4D 表示以断线形式的行列像素的交错排列。最后图 4E 表示以不规则间距形式排列的行列像素表述(例如两

个深色点继之以一个空白点)。

回头查阅图 4A, 如果一比特由 3×3 矩阵代表, 并且一个成像系统检测检测此 3×3 区域内的一个深色行和白色行, 然后检测为一个零(或者一个一)。如果检测到一个图像具有一列深色和两列白色, 那么检测为一个一(或者一个零)。

5 这里, 使用多个像素或点来代表一个比特。使用一个像素(或者比特)代表一个比特是易被损坏的。纸上的灰尘、漂白点、表面不平等等在读取数据单元表述的一个比特时往往造成困难。然而, 将意识到可以使用不同的方案来用图表表示表面上的阵列。在图 4C 至 4E 中表示某些渐近弧。将意识到也可以使用其他方法。在图 11 阐述的一个方案是只使用空号移位点 (space - shifted dots)。

10 使用比特流来创建图 4B 的图解图案 403。图解图案 403 包括 12 行和 18 列。这些行和列由比特流形成, 可以转化为使用比特表述 401 和 402 的图解表示。图 4B 可以视为具有以下比特表述:

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

15 解码

当人员用图 2A 的笔书写或者在编码图案附近移动这些笔时, 摄像机捕获一个图像。例如, 当笔 201 按压在纸上和笔 201 横移过纸上的文档时, 笔 201 可以利用一个压力传感器。然后处理此图像以确定相对于此编码图像的完全表述, 此捕获图像的定位并提取构成此捕获图像的比特。

20 对于确定此捕获图像相对于整个编码区域的取向来说, 人们可能注意到并不是图 5A - 5D 所示的四个可能的边角都存在于此图解图案 403 中。事实上, 利用此正确的方位, 图 5A 所示的边角类型存在于图解图案 403 中。因此, 图 5A 所示的边角类型定位在右方向上丢失。

25 继续图 6, 可以解析摄像机 601 捕获的图像和其确定的方向, 由此可解释图像 601 实际。首先复查图像 601 以确定需要旋转图像的角度 θ , 以此水平地和垂直地定位这些像素。注意到替换的网格校准是可能的, 包括旋转这些基础的格子为不水平和垂直排列(例如 45 度)。当用户在其他图案之前倾向于注意水平和垂直图案时, 使用非水平和垂直排列可以提供消除用户直观干扰的可能的利益。为简明起

见，此格子方向(基础格子的水平和垂直以及任何其他旋转)被集合地称为预定义的格子方向。

接着，分析图像 601 以确定哪个角丢失了。需要旋转图像 601 为解码 603 准备好的图像的旋转量 α 被显示为 $\alpha = (\theta + \text{一旋转量}\{\text{由丢失角定义的}\})$ 。此旋转量由图 7 的等式表示。再查阅图 6，角度 θ 首先由此像素的布局确定，以便到达此像素在水平和垂直(或者其他预定义格子方向)的排列，以及此图像被如图 602 所示进行旋转。然后分析得出结论来确定丢失的角和旋转为图像 603 的图像 602，以建立解码的图像。这里，图像反时针旋转 90 度，图像 603 具有正确的方向并且能被用来解码。

将意识到可以在图像 601 旋转之前和之后适用旋转角 θ ，以占据丢失角。同样会意识到，由于考虑了此捕获图像中的噪声，可以呈现全部四个类型的角。我们可以统计每个角的数目，并选择具有最小的类型作为丢失的角类型。

最后，读出图像 603 中的代码并与用于创建图像 403 的原始比特流相关。此相关可以以许多方式实现。例如，可以通过递归法实现，其中比较恢复的比特流与原始比特流之内的全部其他比特流片段。第二，通过使用两个比特流之间的汉明间距，可以在此恢复恢复比特流和原始比特流之间执行统计分析。将意识到也可以用其它方案来确定原始比特流内部所恢复的比特流。

一旦具有恢复的比特，则必须定位此原始阵列内部的捕获图像(例如图 4B 所示的一个)。确定整个阵列内部比特段位置的流程是由若干选项完成的。首先，捕获的实际比特可以是模糊的(例如，摄像机可以捕获具有模糊代码文档内容的一个图像)。第二，灰尘、漂白点、反射等等也可能引起此捕获图像的误差。这些误差使得定位处理更加困难。在这点上，此图像捕获系统可能必须具有从此图像中提取的非时序比特功能。以下表示用于利用此图像的非时序比特进行操作的方法。

令序列 (m-序列) I 对应于幂系列 $I(x) = 1/P_n(x)$ ，其中 n 时该 m-序列的阶数，捕获的图像包含 $I\mathbf{b} = (b_0 \ b_1 \ b_2 \ \dots \ b_{K-1})'$ 的 K 比特，其中 $K \geq n$ ，上标代表矩阵或矢量的转置矩阵。 K 比特的的位置 s 正是 I 的循环移位数目，所以 b_0 被移位到该序列的起始处。接着该移位的序列 R 对应于幂系列 $x^s/P_n(x)$ 或 $\mathbf{R} = T^s(\mathbf{I})$ ，其中 T 是循环移位算子。我们不能直接找到此 s 。多项式模块形成了一个字段。保证 $x^s \equiv r_0 + r_1x + \dots + r_{n-1}x^{n-1} \pmod{P_n(x)}$ 。因此，我们可以找到 $(r_0, r_1, \dots, r_{n-1})$ 并且解 s 。

$x^s \equiv r_0 + r_1x + \dots + r_{n-1}x^{n-1} \pmod{P_n(x)}$ 的关系意味着 $\mathbf{R} = r_0 + r_1T(\mathbf{I}) + \dots + r_{n-1}T^{n-1}(\mathbf{I})$ 。以

二进制等式表示为:

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

其中 $\mathbf{r} = (r_0 \ r_1 \ r_2 \ \dots \ r_{n-1})'$ 和 $\mathbf{A} = (\mathbf{I} \ T(\mathbf{I}) \ \dots \ T^{n-1}(\mathbf{I}))$, 形成 \mathbf{I} 从 0 移位到 (n-1) 移位的循环移位。现在在 \mathbf{R} 中只有稀少的 K 比特可用来解决 \mathbf{r} 。令 \mathbf{R} 中的 b_i 和 b_0 之间索引差值为 k_i , $i=1,2,\dots,k-1$, 那么 \mathbf{R} 的第一和第 (k_i+1) 个元素正是 b_0, b_1, \dots, b_{k-1} , $i=1,2,\dots,k-1$ 。通过选择 \mathbf{A} 的第一和第 (k_i+1) 列, $i=1,2,\dots,k-1$, 可以形成以下二进制等式:

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

其中 \mathbf{M} 是一个 \mathbf{A} 的 $n \times K$ 子矩阵

10 如果 \mathbf{b} 无差错,那么 \mathbf{r} 的解决方案可以表示为:

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

其中 $\tilde{\mathbf{M}}$ 是 \mathbf{M} 的任何非退化 $n \times n$ 子矩阵, $\tilde{\mathbf{b}}$ 是 \mathbf{b} 的相应的子矢量

借助于已知的 \mathbf{r} , 我们可以使用 Pohlig-Hellman-Silver 算法, 正如 Douglas W. Clark and Lih-Jyh Weng 在“Maximal and Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms,” IEEE Transactions on Computers 43.5 (May 1994, pp 560-568) 中所述的那样, 以发现 s , 使得 $x^s \equiv r_0 + r_1x + \dots + r_{n-1}x^{n-1} \pmod{P_n(x)}$ 。

20 由于矩阵 \mathbf{A} 可以很大 (具有 n 除 L 的大小, 其中 $L=2n-1$), 我们应当避免存储整个矩阵 \mathbf{A} 。事实上, 正如我们在上述处理看到的, 假定提取的比特具有索引差值 k_i , 那么只有 \mathbf{A} 的第一和第 (k_i+1) 列与计算相关。给定了捕获的图像之后, 这种 k_i 的选择是非常受限的。因此, 只有涉及到计算的那些列需要保留。这些列的总数目比 L 要小得多 (其中, $L=2n-1$ 是该 m -阵列的长度)。

纠错

25 如果在 \mathbf{b} 中存在误差, 那么解决方案 \mathbf{r} 变得更加复杂。可能不能轻易地使用具有纠错解码的传统方法, 因为与捕获比特相关的矩阵 \mathbf{M} 可能从一个捕获图像变为另一个。

30 我们采用随机逼近法。假定在 \mathbf{b} 、 n_e 中的误差比特数目与 K 比较相对为小, 那么从 \mathbf{b} 的 K 比特中选择正确的 n 比特的概率以及非退化的相应 \mathbf{M} 子矩阵 $\tilde{\mathbf{M}}$ 就比较高。

当所选的 n 比特全部正确时, \mathbf{b}' 和 $\mathbf{r}'\mathbf{M}$ 之间的汉明间距或者与 \mathbf{r} 有关的差

错比特数目应该为最小，其中， r 利用等式 (4) 来计算。重复该处理几次，很可能造成最小差错的正确的 r 可以被识别。

如果只有一个与最小数目的差错比特相关的 r ，那么就被认为是正确的解决方案。否则，如果多于一个 r 与最小数目的差错比特有关，那么ⁿ。超过 M 产生的代码的纠错能力的可能性就高，并且解码处理失败。此系统然后移动以便处理下一个捕获的图像。在另一个示例中，可以考虑有关笔先前位置的信息。也就是说，对于每个捕获的图像来说，笔接着所希望的目的域可以被识别。例如，如果用户在摄像机捕获的两个图像之间没有抬起笔，那么第二捕获图像确定的位置不应当离第一位置太远。与最小数目差错比特相关的每个 r 接着被检测，以便看是否从 r 计算的位置 s 满足 该本地限制，即是否该位置在指定的目的区域。

如果位置 s 满足本地限制，那么在每个阵列提取的 X , Y 位置被返回。如果不满足，那么解码处理失败。

图 8 描绘了在捕获图像的序列 (或 m -序列) 中用来确定位置的一个处理流程。首先，在步骤 801，接收涉及捕获图像的数据流。在步骤 802，相应的列从 A 中提取出来，并构成矩阵 M 。

在步骤 803， n 独立列向量是从矩阵 M 中随机选择，并且向量 r 由等式 (4) 确定。在步骤 804，此流程进行 Q 次 (例如，100 次)。以后将在“循环次数计算”中讨论确定循环次数。

在步骤 805，根据其相关的差错比特数对 r 进行分类。可以利用多种已知技术中的分类算法进行排序。例如，可以使用选择分类算法。当数目 Q 不大时选择分类算法是有益的。然而，如果 Q 变大，可以更有效地采用处理更大数目项的其它分类算法 (例如合并分类)。

接着在步骤 806，通过检测是否多个 r 与最小数目的差错比特相关，系统确定是否成功地完成纠错。如果是，那么在步骤 809 返回一个错误，表明解码失败。如果否，那么在步骤 807 计算在序列 (或 m -序列) 中提取的比特的位置 s ，例如，通过使用 Pohig-Hellman-Silver 算法。

下面，阵列中 (X , Y) 位置被计算为: $x = s \bmod m_1$ 和 $y = s \bmod m_2$ 在步骤 808 返回该结果。

30 位置确定

图 9 表示用于确定笔尖位置的一个处理流程。此输入是摄像机捕获的一个图

像，输出可以是此笔尖的位置坐标。同时，此输出可以包括(或不包括)其他信息诸如此捕获图像的旋转角。

在步骤 901，接收来自摄像机的一个图像。接着，在步骤 902 选择性地预处理此接收的图像(由步骤 902 的间断线表示)，以便调整深浅像素等等之间的差别对比。

接着，在步骤 903，分析此图像以确定其内部的比特流。

接着，在步骤 904，多次从比特流中随机选择 n 比特，并确定该原始序列(或者 m -序列)内部接收的比特流的位置。

最后，在步骤 904 一旦确定了此捕获图像的位置，在步骤 905 确定笔尖的位置。

图 10 给出关于 903 和 904 的更多详情，并表示提取捕获图像内部比特流的方案。首先，在步骤 1001 接收来自于摄像机的一个图像。在步骤 1002 此图像接着可以选择性地经历图像预处理(由步骤 1002 的间断线表示)。在步骤 1003 提取此图案。这里各个线上的像素可以被识别，并且穿过此像素的图案方向(对应于角度 θ)能够被估算。

接着，在步骤 1004 分析接收的图像以确定此基础网格线。如果在步骤 1005 发现网格线，那么在步骤 1006 从此图案中提取代码。接着在步骤 1007 对代码解码，在步骤 1008 确定笔尖的位置。如果在步骤 1005 未发现网格线，那么在步骤 1009 返回一个错误。

20

增强型解码和纠错算法的概要

利用图 12 所示的本发明实施例，假定从捕获图像(对应于捕获阵列)和目的地区域中提取比特 1201， m -阵列解码的变型和纠错处理对此 X,Y 位置解码。图 12 表示此增强型方法的处理处理 1200 的流程图。处理流程 1200 包含两个部分 1251 和 1253。

- 解码一次。部分 1251 包括三个部分。
 - 随机比特选择：随机选择此提取比特 1201 的一个子集(步骤 1203)
 - 解码此子集(步骤 1205)
 - 确定具有局部约束的 x,y 位置(步骤 1209)
- 利用智能比特选择解码。部分 1253 包括四个部分。
 - 智能比特选择：选择此提取比特 1201 的另一个子集(步骤 1217)

30

- 解码此子集 (步骤 1219)
- 调整步骤 1217 和步骤 1219 重复(循环时间)的数目(步骤 1221)
- 确定具有局部约束的 x,y 位置 (步骤 1225)

本发明的实施例利用谨慎的策略来选择比特、调整循环迭代数目并根据局部
5 约束确定 X,Y 位置(位置坐标), 流程 1200 提供。借助于部分 1251 和 1253, 步骤
1205 和 1219 (解码一次) 利用等式 (4) 计算 r

令 \hat{b} 为解码的比特, 即

$$\hat{b}' = r'M$$

b 和 \hat{b} 之间的差值是与 r 相关的差错比特。

10 图 12 表示根据本发明的实施例, 用于对从捕获图像中提取的比特 1201 解码
的处理流程 1200 的流程图。处理流程 1200 包含部分 1251 和 1253。部分 1251
获得与捕获图像相关(对应于捕获阵列)的提取比特 1201 (包含 K 比特)))。在步骤
1203, 从提取的比特 1201 中随机选择 n 比特。在步骤 1205, 处理流程 1200 解码
一次并计算 r 。在步骤 1207, 处理流程 1200 确定对于 b 是否检测出误差比特。如
15 果步骤 1207 确定没有误差比特, 那么在步骤 1209 确定此捕获阵列的 X,Y 位置坐
标。对于步骤 1211, 如果 X,Y 坐标满足局部约束即目的区域内部的坐标, 那么
在步骤 1213 处理流程 1200 提供此 X,Y 位置(达到另一个处理或者用户接口)。否
则, 步骤 1215 提供一个失败指示。

如果步骤 1207 检测出 b 中的误差比特, 执行部分 1253 以便对误差比特解码。
20 步骤 1217 从提取的比特 1201 中选择另一套 n 比特(与在步骤 1203 选择的 n 比特
至少有一个比特不同)。步骤 1221 和 1223 确定对此提取比特解码所必须的重复
数目(循环时间)。步骤 1225 通过测试在步骤 1219 获得的哪个候选者满足此局部
约束来确定此捕获阵列的位置。将更详细地讨论步骤 1217 - 1225。

25 智能比特选择

步骤 1023 从提取的比特 1201 (具有 K 个比特) 中随机选择 n 个比特, 并
解答为 r_1 。利用方程式 (5), 能计算出解码比特。令 $I_1 = \{k \in \{1,2,\dots,K\} | b_k = \hat{b}_k\}$,
 $\bar{I}_1 = \{k \in \{1,2,\dots,K\} | b_k \neq \hat{b}_k\}$, 当 \hat{b}_k 是 \hat{b} 的第 k 个比特时, $B_1 = \{b_k | k \in I_1\}$ 并且
 $\bar{B}_1 = \{b_k | k \in \bar{I}_1\}$, 即, B_1 是与原始比特一样的解码结果的比特, \bar{B}_1 是于原始比
30 特不同的解码结果的比特, I_1 和 \bar{I}_1 是与这些比特相关的索引。当从 B_1 中选择
任何 n 个比特时可以获得相同的 r_1 是可以意识到的。因此, 如果不谨慎的选取下

一个 n 比特,就可能导致选择的比特是 B_1 子集,因而发生导致获得相同的 r_1 结果。

为了避免这样的情况发生,步骤 1217 按照下面的 步骤选取下一个 n 个比特。

1. 在 \bar{B}_1 1303 中选取至少一个比特并且从 B_1 1301 中随机的选取剩余的比特,如图 13 所示的与比特排列 1351 一致。通过计算 $\hat{b}'_2 = r'_2 M_2$, 处理流程 1200 解出 r_2 并且得到 B_2 1305, 1309 和 \bar{B}_2 1307, 1311。
2. 重复步骤 1。当选择下 n 个比特时,对于任何 \bar{B}_i ($i = 1, 2, 3, \dots, x-1$, 其中 x 是当前循环数),至少有一个从 \bar{B}_i 中选择的比特。当没有这样的比特子集可被选择时或者当循环次数达到时该迭代终止。

10 循环次数的计算

误差校正部分 1253,需要的迭代数(循环次数)是每次循环后调整的。循环的次数是由预期的误差率来决定。即不是全部的被选择的 n 比特是正确的预期的误差率 p_e 是:

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

- 15 其中 lt 代表循环次数并且被一个常量初始化, K 是从捕获的阵列中提取的比特数, n_e 代表在处理流程 1200 的迭代期间招致的最小误差比特数, n 是多维阵列的阶数, C_K^n 是从 K 比特里选择 n 比特的组合数目。

在实施例中, 我们想要 p_e 小于 $e^{-5} = 0.0067$. 结合 (6),我们有:

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

- 20 调整循环次数可能明显地减少需要误差校正的处理流程 1253 的迭代次数。

用本地限制确定 X, Y 的位置

- 25 步骤 1209 和 1225,解码位置应该在目的区域内。 目的区域是算法的一个输入部分,根据不同的应用,它可以是各种大小和位置或者只不过是整个 m -阵列。例如,如果确定了先前的位置,考虑书写速度,当前笔尖的目的区域应该接近先前的位置。然而,如果笔被抬起,那么下一个位置可能是任何地方。因此在此情况下,目的区域应当是整个 m -阵列。正确的 X, Y 位置由以下步骤来确定。

在步骤 1224, 处理流程 1200 选择其对应的差错比特数小于以下的 r_i

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

其中 lt 是实际的循环次数, lr 代表由以下公式计算的本地限制:

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

其中 L 是 m -阵列的长度.

- 5 步骤 1224 以差错比特数目的升幂来分类 r_i 。然后步骤 1225, 1211 和 1212 找出其相应的 X, Y 位置位于目的区域的第一 r_i 。步骤 1225, 1211 和 1212 最后返回 X, Y 位置作为结果(经由步骤 1213), 或者指示解码处理失败(经由步骤 1215)。

用于确定笔划路径(笔划定位)的体系

- 10 图 15 表示根据本发明的一个实施例, 用于从捕获帧 1551 的序列中确定笔尖路径的一种方法。在步骤 1501, 处理一帧, 以此诸如正文的文档内容就从只包含迷宫图案单元的其他区域分离开。此外, 此图像(灰度级的)被归一化以补偿不均匀照明。在步骤 1503, 从一帧(捕获图像)中的可见迷宫图案条形图(bar)中提取 m -阵列。在步骤 1505, 如果提取的比特数目(在步骤 1503 确定的)大于此嵌入的 m -阵列阶数, 那么通过 m -阵列解码可以获得唯一的位置坐标(对应于此帧中心的 $x-y$ 位置)。

- 15 步骤 1507 分析数字文档 1553, 以便确定由文档 1553 的内容遮盖的迷宫图案的文档区域 1553。(利用此实施例, 数字文档 1553 不包括用户产生的批注)。如果文档内容 1553 遮盖迷宫图案单元, 那么步骤 1505 不能提取足够的 m -阵列比特来确定此帧的 $x-y$ 位置。通过分析此整个水印文档 1553, 处理流程 1500 能够查明 $x-y$ 位置不能被唯一确定的文档区域 1553。如将要讨论的, 在步骤 1509 采用此分析。

- 20 在步骤 1511, 如果不能根据 m -阵列解码确定与笔划相关的任何帧的 $x-y$ 位置(对应于步骤 1505), 那么处理流程 1500 不能用于确定此帧的位置。在此情况下, 步骤 1509 执行全局定位, 其中捕获的图像能够被一个通过迷宫图案分析(步骤 1503)获得的仿射变换弄弯, 然后能够与文档图像 1553 相配, 文档图像 1553 中迷宫图案单元的重要数量被此文档内容所遮盖。在此情况下, 步骤 1509 执行全局定位, 其中捕获的图像能够被一个通过迷宫图案分析(步骤 1503)获得的仿射变换弄弯, 然

后能够与文档图像 1553 相配,文档图像 1553 中迷宫图案单元的重要数量被此文档内容所遮盖。一旦成功地匹配好一帧,那么使用局部定位(将要讨论的步骤 1517)来定位整个笔划(对应于一系列帧)。

5 在步骤 1517,其 $x-y$ 位置不能利用步骤 1505 解码或者不能利用步骤 1509 进行全局定位确定的帧还要再处理。这种帧的位置应该靠近其 $x-y$ 位置被 m -阵列解码或者全局定位确定的那些帧的位置。这种帧的位置是通过匹配此帧与已经确定了 $x-y$ 位置的相邻区域来确定的。也可以获得捕获图像和文档图像之间的透视变换 $F_{S \rightarrow P}$ (如以前所讨论的)。

10 从 m -阵列解码(对应于步骤 1505)和全局局部定位(分别对应于步骤 1509 和 1517)获得的 $X-y$ 位置表示捕获图像的中心的位置($x-y$ 位置)。(本的实施例可以一维、二维和三维。在本实施例中,支持二维空间以此位置定位对应于一个 $x-y$ 位置)。为了获得笔尖的 $x-y$ 位置,需要校准笔尖(例如图 2 所示的笔尖 202)和关联的摄像机(例如图 2 所示的摄像机 203)之间的关系。步骤 1513 支持将要讨论的校准(calibration)。

15 在步骤 1515,通过使用从局部定位和校准参数处获得的透视变换来映射图像中心的 $x-y$ 位置,以此确定笔尖的 $x-y$ 位置。(确定笔尖路径 1555 的一个例子如将要讨论的图 26 所示。)

20 图 16 表示根据本发明的一个实施例,用于从捕获图像(帧)的序列中确定笔尖路径的一个装置 1600。在本发明的一个实施例中,图像分割 1601 执行步骤 1501、迷宫图案分析器 1603 执行步骤 1503、 m -阵列解码模块 1605 执行步骤 1505、匹配模块 1609 执行步骤 1511、1509 和 1517、水印文档分析器 1607 执行步骤 1507、校准模块 1611 执行步骤 1513、和映射模块 1613 执行步骤 1515。

25 文档图像分析和校准可以脱机执行、或者在在线图像捕获和处理之前分离。可以联机或者脱机执行另一部分(例如图像分割模块 1601、迷宫图案分析器 1603 m -阵列解码模块 1605、匹配模块 1609、和映射模块 1613)。下面将详细描述这部分的功能。

图像分割

30 捕获图像可以包含文档内容诸如正文或者附图,这是在预处理时最初评价的。图 17 表示根据本发明的一个实施例,捕获图像 1700 的一个例子。图 18 表示根据本发明的一个实施例,包含正文部分包括正文字符 1801 和 1803)的一个捕获图

像示例。如果一个图像包含正文/附图区域，那么此正文/附图区域从只包含迷宫图案或者空白像素的另一个区域中分离开。

此外，此捕获图像例如捕获图像 1700 和 1800)被标准化以补偿不均匀照明。图 19 表示根据本发明的一个实施例，对应于对捕获图像 1700 (图 17 所示)照明的
5 标准化的处理过的图像 1900。图 20 表示根据本发明的一个实施例，对应于对捕获图像 1800 照明的标准化的处理过的图像 2000。

迷宫图案分析

迷宫图案分析的任务是在捕获图像中，从可见迷宫图案条形(例如图 4A 所示
10 的条形图 401 和 402)中提取 m -阵列比特。图 19 表示根据本发明的一个实施例，捕获图像 1700 (图 17 所示)的一个迷宫图案分析。比特 1901 (对应于比特值 1902 的“0”)和比特 1903 (对应于比特值 1904 的“1”)是 m -阵列比特的两个成员。在此迷宫图案中根据迷宫图案网格线例如网格线 1905 和 1907 来组织 m -阵列。图
20 表示根据本发明的一个实施例，捕获图像 1800 (图 18 所示)的一个迷宫图案分
15 析 2000。(在此例子中不能确定在正文字符 1801 和 1803 附近的某些迷宫图案单元的 m -阵列比特)。

图 19 和 20 表示对迷宫图案单元和迷宫图案条形图的一个说明。首先计算迷宫图案网格线的参数(沿着每个维度即仿射变换的刻度等级和旋转)，然后确定嵌入
20 到此迷宫图案中的原始方向(或者象限)。从而，基于此网格线和条形图方向确定此 m -阵列比特信息。

m -阵列解码

如果在步骤 1505 (图 15 所示的)获得的提取比特数目大于嵌入的 m -阵列的阶数，那么通过 m 阵列解码可以获得唯一的 $x - y$ 位置。
25

水印文档图像分析

迷宫图案可能被此文档内容所遮盖，这意味着没有能够从捕获图像中提取解码的足够的 m -阵列比特。通过分析此整个水印文档图像，处理流程 1500 或者装置 1600 能够确定在哪个区域能够唯一地确定此 $x - y$ 位置和在哪个区域不能唯一
30 地确定 $x - y$ 位置。在全局定位时使用此分析结果。图 21 表示根据本发明的一个实施例，分析文档图像 1553 的结果 2100。在此实施例，文档图像(例如文档

1553)上的像素被称为四个类型中的一种。根据以下关系标记像素:

类型 I: 32 乘 32 像素相邻窗口(考虑此中心的像素)只包含迷宫图案单元

类型 II: 32 乘 32 像素相邻窗口包含 60 个或更多个迷宫图案单元

类型 III: 32 乘 32 像素相邻窗口包含 36 至 60 个迷宫图案单元。

5 类型 IV: 32 乘 32 像素相邻窗口包含 35 或者小于 35 个迷宫图案单元。

在此实施例, 如果捕获图像的中心位于类型 I 或者类型 II 区域, 则能够确定此捕获图像的 x - y 位置, 如果此中心位于类型 III 区域, 则可以确定确定此捕获图像的 x - y 位置。在图 21, 区域 2101 对应于类型 I 象素面积, 区域 2103 对应于类型 II 象素面积, 区域 2107 对应于类型 III 象素面积, 区域 2105 对应于类型 IV 象素面积。 (分析和标记此整个文档, 但在图 21 只显示一部分)。

在此实施例, 通过产生一个位图 (bitmap) 电子文档或者通过扫描纸件然后再以适当分辨率抽样此相应位图, 处理流程 1500 就可以得到数字文档 1553。分辨率的确定是基于以下考虑的: (1) 文档图像的分辨率不应小于此捕获图像, 因为此水印的灰阶文档图像将与摄像机捕获的图像相配, 以便确定此捕获图像的位置; (2) 一个打印的迷宫图案单元应该映射为文档图像像素的整数目, 以此该匹配算法会更有效地工作。例如, 如果摄像机的分辨率是 0.15mm / 像素, 即物理世界中的 0.15mm 被映射为一个摄像机像素, 那么该印刷的迷宫图案单元尺寸是 0.45mm * 0.45mm, 即在该摄像机传感器上一个印刷的迷宫图案单元被映射为 3 * 3 像素, 该文档图像的分辨率还应该设置为 0.15mm 像素, 以此在该文档图像中一个印刷的迷宫图案单元将映射为 3 * 3 象素面积。

快速图像匹配的全局定位

如果一个笔划的捕获图像的 x - y 位置不能从 m - 阵列解码中确定, 那么位于该不能唯一确定 x - y 位置区域的所有图像的中心是可预言的。在这种情况下, 通过由迷宫图案分析得到的仿射变换来弄弯此捕获图像, 然后与其中不能唯一确定 x - y 位置的文档图像区域相配。一旦成功地匹配了一帧, 那么就采用局部定位算法来定位此整个笔划。

对于文档 1553 (图 15 所示), 在 32 乘 32 像素子窗口中的可见迷宫图案单元数目典型地从 10 到 100 变化。此文档中的像素被标记为水印文献分析(图 15 所示的步骤 1507) 的四个类型之一。一个搜索区域被设置为类型 III 和类型 IV 区域的集合。

图 22 表示根据本发明的一个实施例，用于示例性笔划 2201 的全局定位结果。借助于全局定位(对应于步骤 1509)来确定笔划 2201 上的点 2203 的 $x - y$ 的位置。部分 2202 被放大为放大的部分 2204 以便更好地表示笔划上的点 2203。(在此实施例，借助于全局定位只确定笔划 2201 上的一点，其余的点由局部定位确定)。

5 利用全局定位，在此搜索区域的选择点处将一帧与此文档图像匹配。通过对迷宫图案分析(对应于图 15 所示的步骤 1503)进行刻度定标和旋转(仿射变换)，可以将此原始帧弄弯(变换)。对于在匹配帧中确定选择点来说，一个偏移可能有用。

在此实施例，一帧匹配成功对应于捕获图像和此文档图像之间的最大互相关值。(此互相关值交替变换为 0 到 1)。此最大互相关值应该比其他互相关值和
10 此门限值充分地大，即大到多少，可以由脱机训练确定。例如，全局定位可以要求最高互相关值和第二高互相关值之间的差值要大于 0.1。

快速图像匹配的局部定位

对于其 $x - y$ 位置不能通过 m -阵列解码/全局定位确定的那些帧来说，此位置
15 应该靠近其中 $x - y$ 位置被确定的帧的位置。从而，相应的位置是通过匹配此图像与已经确定了 $x - y$ 位置的相邻区域来确定的。还得到捕获图像和文档图像之间的透视变换。

图 23 表示根据本发明的一个实施例，用于示例性笔划 2301 的局部定位结果。借助于 m -阵列解码确定笔划 2301 上点(例如点 2305a 和 2305b)的子集的存储单元
20 位置，和借助于局部定位确定剩余点(例如点 2303a 和 2303b)的位置。

在此实施例，局部定位利用起始点和中枢起始点。起始点是由 m -阵列解码(对应于图 15 所示的 1505)或者由全局定位(对应于图 15 所示的步骤 1509)成功地定位了的那些帧的位置。一帧序列被分割为一些段，其中逐段执行本地搜索。在局部定位期间，一个段可以被分成几个分段。在此实施例中，本地搜索被限制在基
25 于笔尖移动的速度和加速度限制之内的搜索区域。

下面的程序描绘局部定位的一个实施例：

- 帧序列分割

(a) 选择中枢起始点

一笔划中的第一和最后一个起始点是中枢起始点。

30 对于两个中枢起始点 p_1 、 p_2 之间的起始点来说，具有距穿过 p_1 、 p_2 的线路 L 的最大距离 D_{max} 的一个，和大于门限(典型地设置为 0.5)的 D_{max} 作为中枢起始

点。

(b) 细化此中枢起始点处的透视变换，通过比较摄像机捕获的图像与此文档图像来得到更精确的透视变换。

(c) 由中枢起始点对此帧序列分段

5 每个分段是从中枢起始点或者此笔划的第一点开始，在中枢起始点或者此笔划的最后一点结束。

- 确定完成的分段

当以下情况发生时，宣布一个分段为完成的分段：

10 (a) 这分段中至少有一个起始点不是中枢起始点，即在此分段上不能找到更多的中枢起始点，或者换句话说，此分段非常像一条直线。在这种情况下，所有的点是插入的并且此分段被宣布完成。

或者

(b) 对于此分段中的每个点，相应的帧已经被处理。

- 为未完成的分段的一帧找到一个搜索区域

15 (a) 此分段的第一点是一个起始点，并且不必进行处理。

(b) 对于此分段的第二个点，此搜索区域的中心设置为第一点，并且搜索区域的大小受最大速度的限制。

20 (c) 对于此分段的其他点，能够估算已经处理过的前一点的速度。能够从此位置和前一点的速度中计算搜索区域的中心，并且搜索区域的大小受到最大加速度的限制。

- 在搜索区域的模板匹配

此步骤是基于这样的假定：在较短时间内笔的姿势仅仅有一点点改变，其意味着相邻的帧之间的仅有一点透视变换差值。

(a) 借助于前一处理过的帧的透视矩阵弄弯此帧。

25 (b) 通过匹配此帧与作为中心点的文档图像，确定在此搜索区域每个点的互相关。

(c) 具有最高互相关的点应该为此帧的正确位置。

- 细化对于此点的透视变换

通过比较此摄像机捕获的图像与此文档图像来细化此透视变换。

30 • 切断 (Prune) 此搜索结果

几个因素可以造成获得的错误结果，诸如不正确的初始透视变换参数和移动

混乱。通过限制移动的速度和加速度，能够切断此错误结果。

(a) 所有此起始点将不会被切断。

(b) 从第一点到最后点经历整个笔划。如果一个点不能满足前一点的移动限制，那么此点一定是错误的并应该切断。

5 (c) 从最后一点到第一点经历整个笔划。如果一个点不能满足前一点的移动限制，那么此点一定是错误的并应该切断。

(d) 切断之后，笔划中保持的点是正确的点。此切断的点被根据内插法确定的点替代。

10 摄像机-笔-尖校准

从 m -阵列解码和全局/局部定位获得的 $X - y$ 位置表示此捕获图像中心的位置。为获得笔尖 202 的 $x - y$ 位置，笔尖 202 和摄像机 203 的中心之间的关系必须要校准。快速和准确校准是有效的，因为墨水盒可以频繁地改变。

15 通过以不同姿势在书写表面上的固定触点上接触笔尖，可以捕获到一些图像。通过使用一个限制来确定校准参数，该限制是此图像的所有 $x - y$ 位置应该借助于正确的校准参数被映射为相同点(固定触点)。

在此实施例中下面的过程是用来校准笔尖。该过程用来估算校准参数

$L_{virtual-pentip}$:

- 20 a) 把实际的笔尖放在纸张上的固定位置 L_{pentip}
- b) 使笔尖在象素位置 L_{pentip} 上直立,并且捕捉一系列具有不同姿势的图象。
- c) 对于每一个捕获图象,通过曲径图案分析,m-阵列解码和匹配捕获图象和图象文档来计算该转换 $F_{S \rightarrow P}$ (把摄像机捕获的图象的位置坐标转换为纸张上的实际图象的位置坐标)和 $F_{P \rightarrow S} = F_{S \rightarrow P}^{-1}$ ($F_{S \rightarrow P}$ 的逆变换),因而:

$$L_{pentip}^i = F_{S \rightarrow P}^i \cdot L_{virtual-pentip}, \quad i = 1, 2, \dots, N$$

$$25 \quad L_{virtual-pentip}^i = F_{P \rightarrow S}^i \cdot (L_{pentip} + \Delta L_i), \quad i = 1, 2, \dots, N$$

在那里 N 是实验中捕获图象的数量, ΔL_i 是在第 i 个 帧中的笔尖确切位置和 L_{pentip} 之间的偏移。

d) 初始化 $\bar{L}_{virtual-pentip}$ 为 $s(0, 0)$, 这里 $\bar{L}_{virtual-pentip}$ 是 $L_{virtual-pentip}$ 的估算值。

30 e) 使用中(c)的第一个方程式, 设定 $L_{virtual-pentip}$ 为 $\bar{L}_{virtual-pentip}$ 求得 L_{pentip}^i , $i = 1, 2, \dots, N$. 通过平均 L_{pentip}^i , L_{pentip} 被估算为:

$$\bar{L}_{pentip} = \frac{1}{N} \sum_{i=1}^{i=N} L_{pentip}^i$$

f)使用中(c)的第二个方程式, 设定 L_{pentip} 为 \bar{L}_{pentip} 求得 $L_{virtual-pentip}^i$, $i=1,2,\dots,N$.
通过平均 $L_{virtual-pentip}^i$, $L_{virtual-pentip}$ 被估算为:

$$\bar{L}_{virtual-pentip} = \frac{1}{N} \sum_{i=1}^{i=N} L_{virtual-pentip}^i$$

5 g)重复步骤 e.在数次迭代后, $L_{virtual-pentip}$ 和 L_{pentip} 将会分别的聚合到更充分精确的结果, 这可以被称为 $L_{virtual-pentip}^{accurate}$ 和 $L_{pentip}^{accurate}$.

最后, 获得一个 $L_{virtual-pentip}^{accurate}$ 作为校准参数 $L_{virtual-pentip}$.

把图像中心位置映射到笔尖

10 笔尖 202 的位置 X-y 是通过映射图象中心的 x-y 位置确定, 该图象中心的 x-y 位置是从局部定位和校准参数利用透视变换获得。图 24 和 25 图解说明了笔尖校准结果的一个示例。图 26 展示了依照本实施发明笔尖 202 恢复的笔划。图 25 展示了与图 24 所示的恢复的笔划有关的实际笔划 2501。在图 24 中的笔划 2403 展示了捕获图象的中心位置路径。就是说, 如果没有校准, 笔划 2501 就可能被错误地
15 恢复(恢复为笔划 2403), 反之如果有适当的校准, 笔划 2501 就可被正确地恢复(恢复为笔划 2401)。

正如本领域技术人员可以意识到的, 可以使用具有包含指令用于控制计算机系统的相关计算机可读介质的计算机系统来实现这里公开的示例实施例。此计算机系统能够包括至少一个计算机诸如微处理器、数字信号处理器、和有关外围电子电
20 路。

虽然本发明被所附权利要求所限定, 但这些权利要求只是说明本发明希望包括这里以任何组合和子组合描述的元素和步骤。从而, 还具有任何数目的可替换组合用来限定本发明, 这些替换组合了结合说明书包括描述说明、附图和权利要求的一个或多个元素。对于相关领域技术人员很明显的是, 根据本发明的说明书, 本发
25 明一些方面的替换组合, 或者单独或者与这里限定的一个或多个元素或步骤组合, 可以被用来修改或替换本发明或者本发明的部分。希望本发明这里包含的书写描述覆盖所有这种修改和替换。

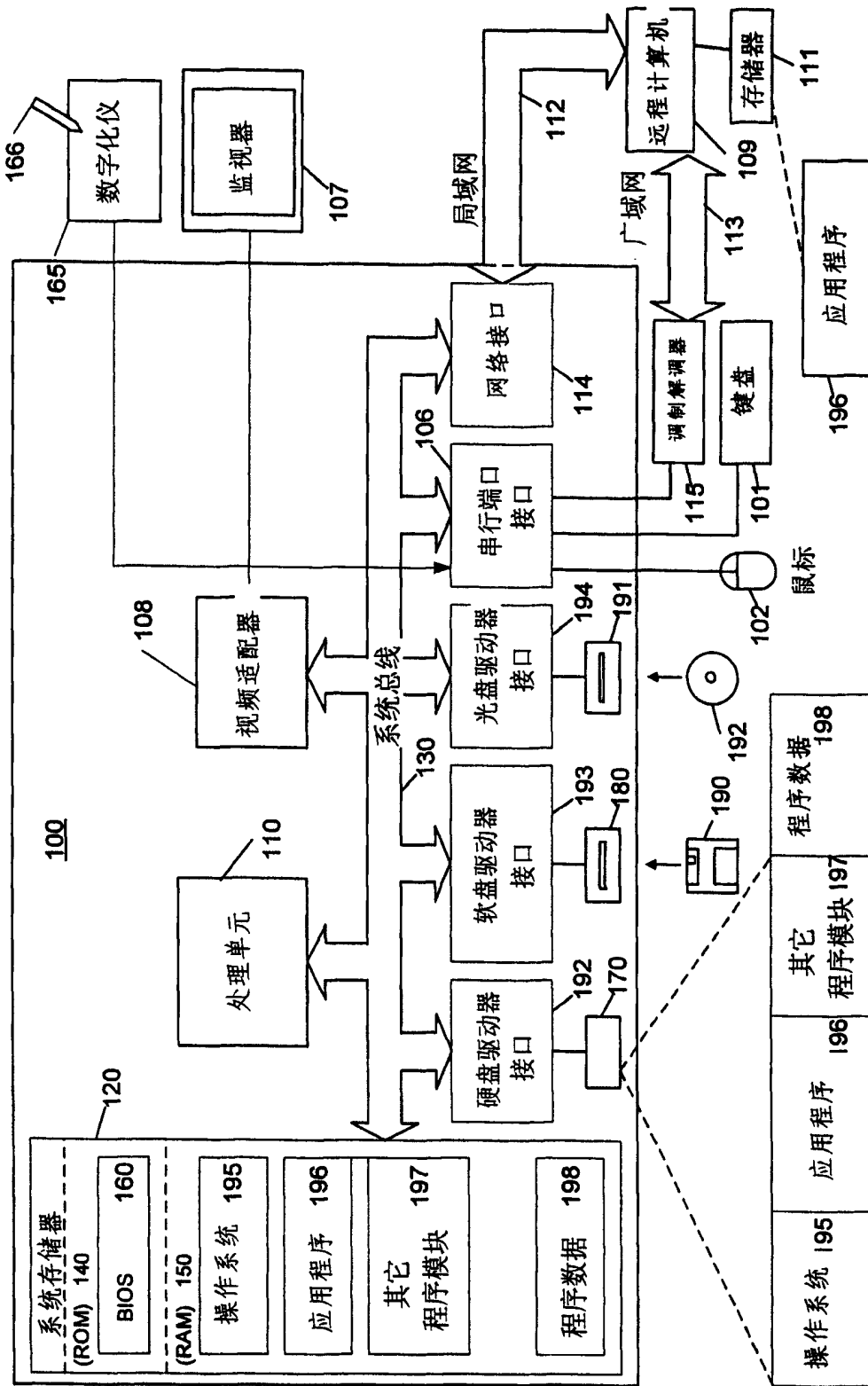


图 1

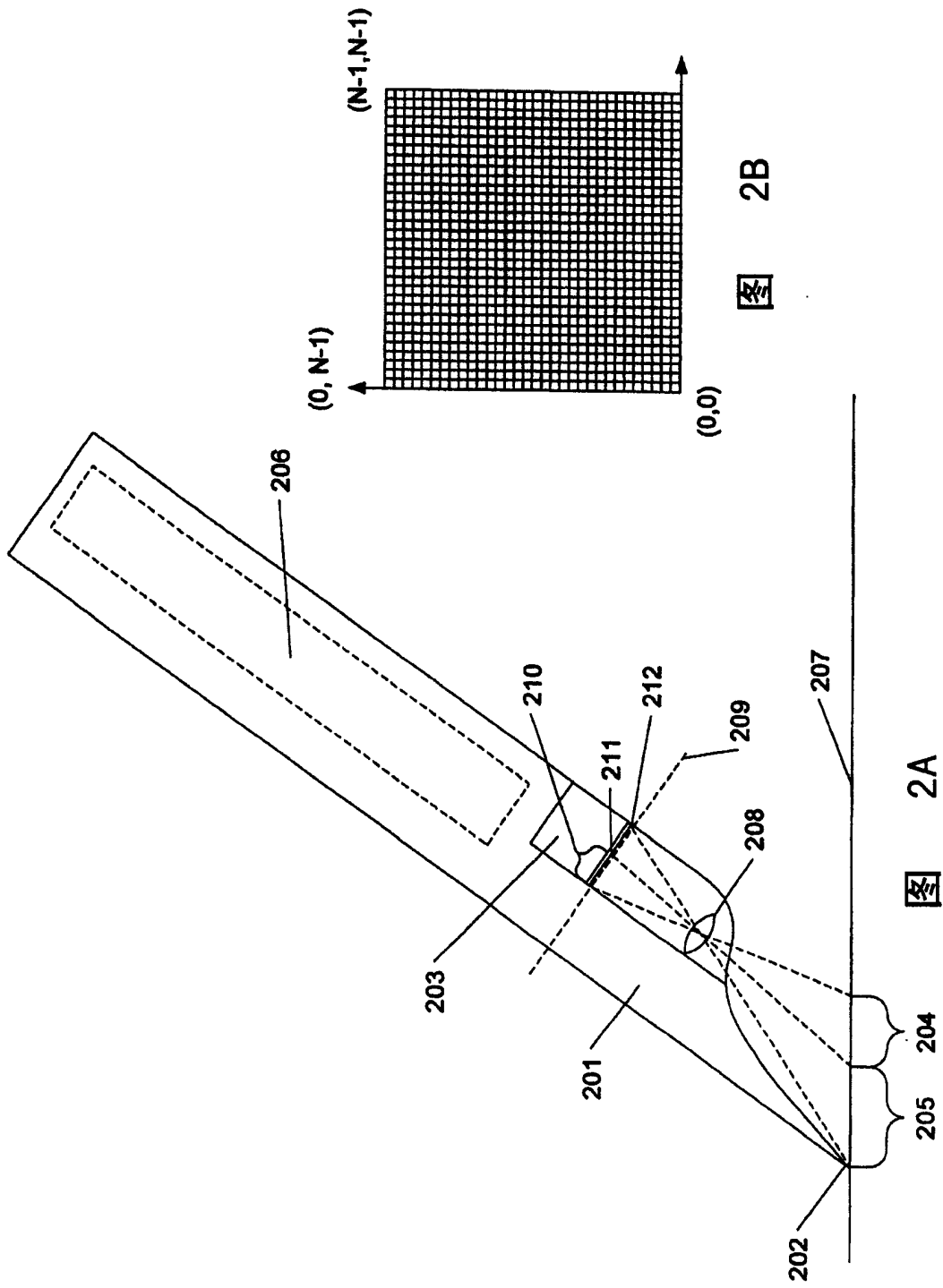


图 2B

图 2A

图 3A 0000010 00 01100 0101 001 111010 0 0111001 0 010110 111 0110 01101 01 0111111

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

图 3C 000100100 00111110 01011101 010011001 011100111 001011010 011000011

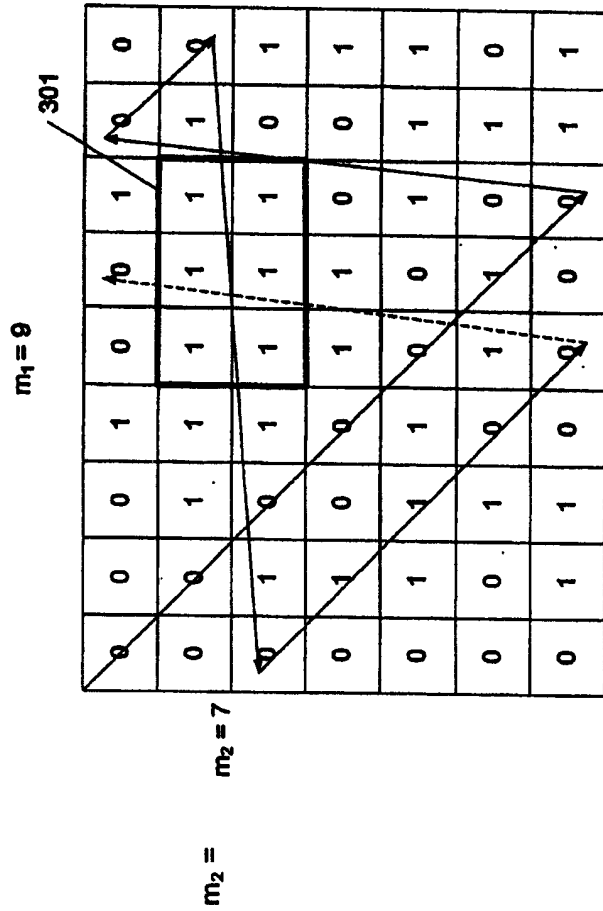


图 3D

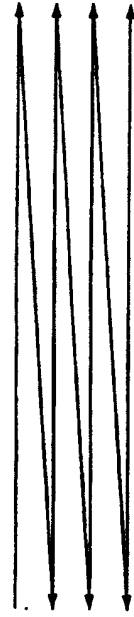
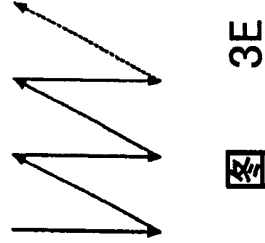
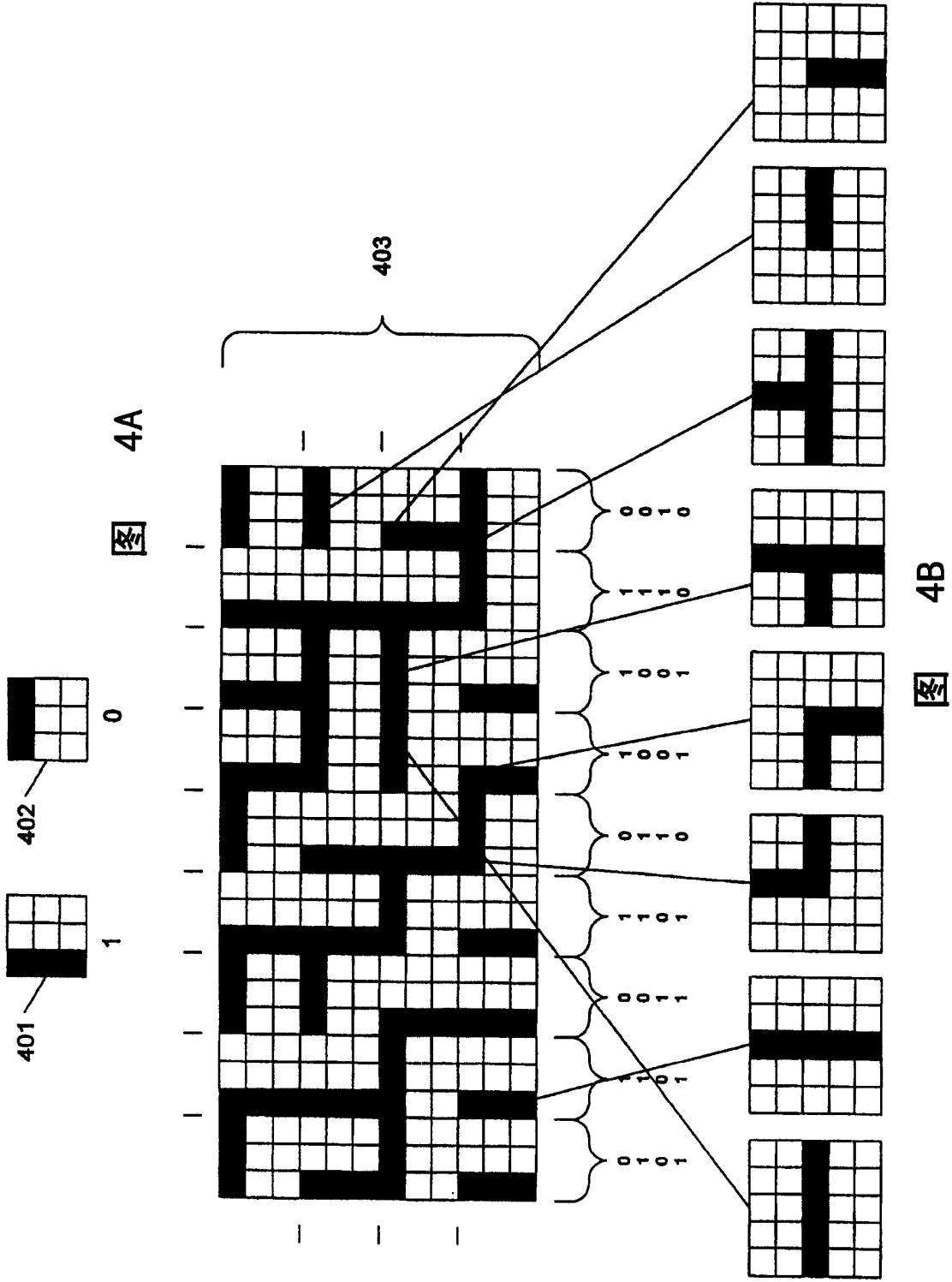
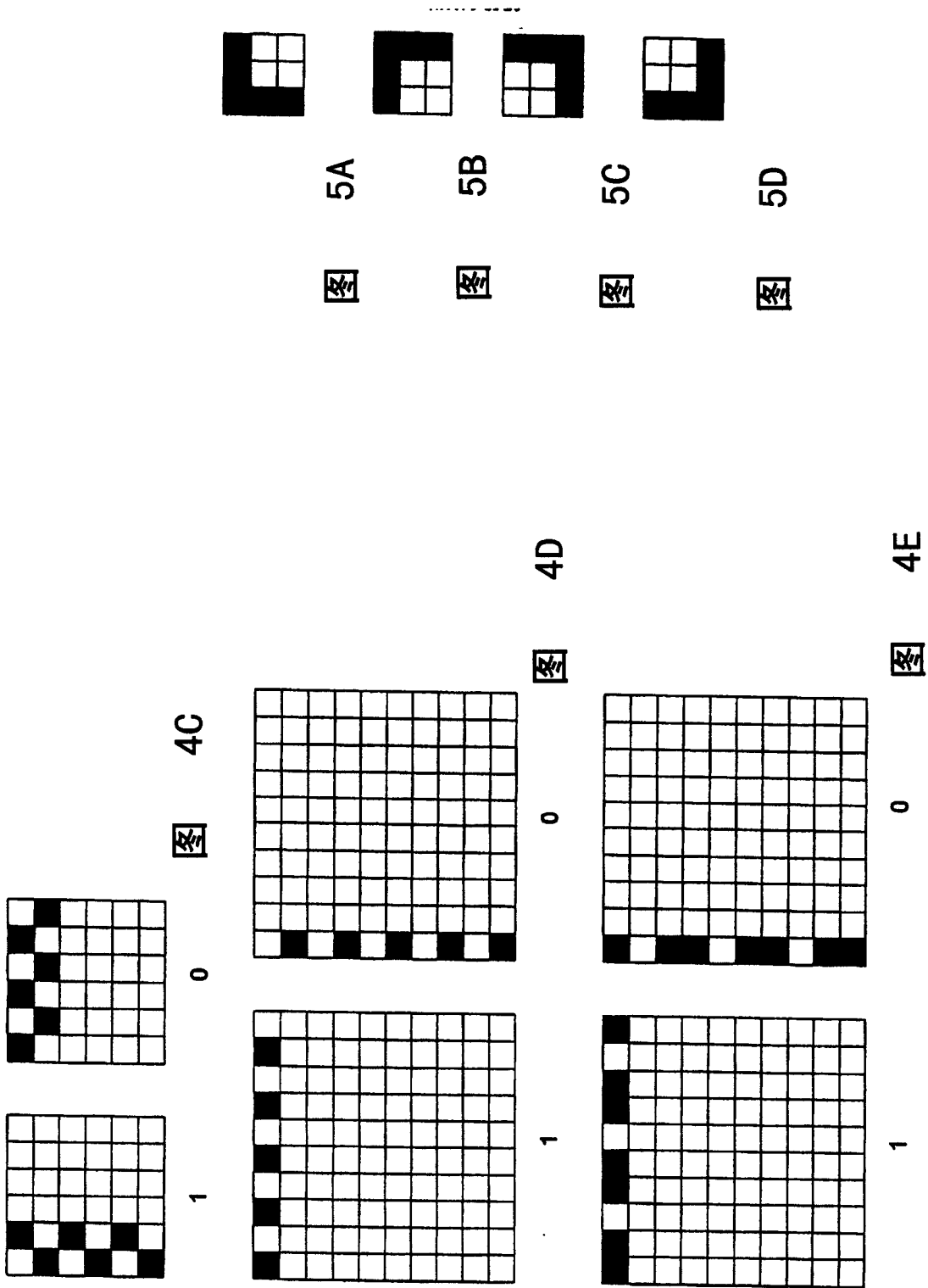


图 3F





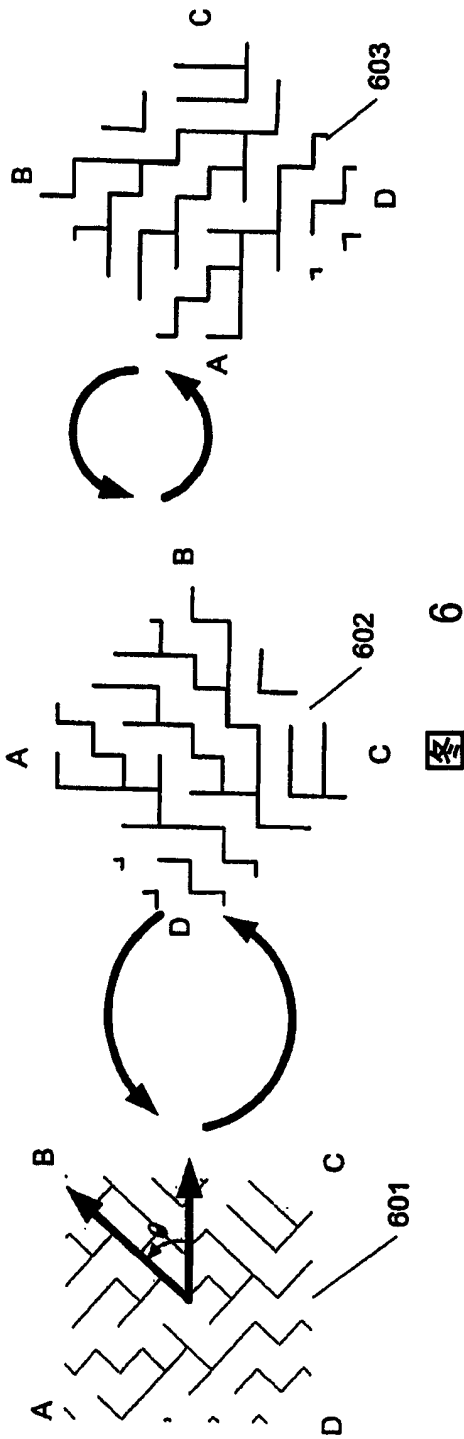


图 6

$$o = \theta + \begin{cases} 0 - \pi o a \\ \frac{\pi - \pi o b}{2} \\ \frac{\pi - \pi o c}{2} \\ \frac{3\pi - \pi o d}{2} \end{cases}$$

图 7

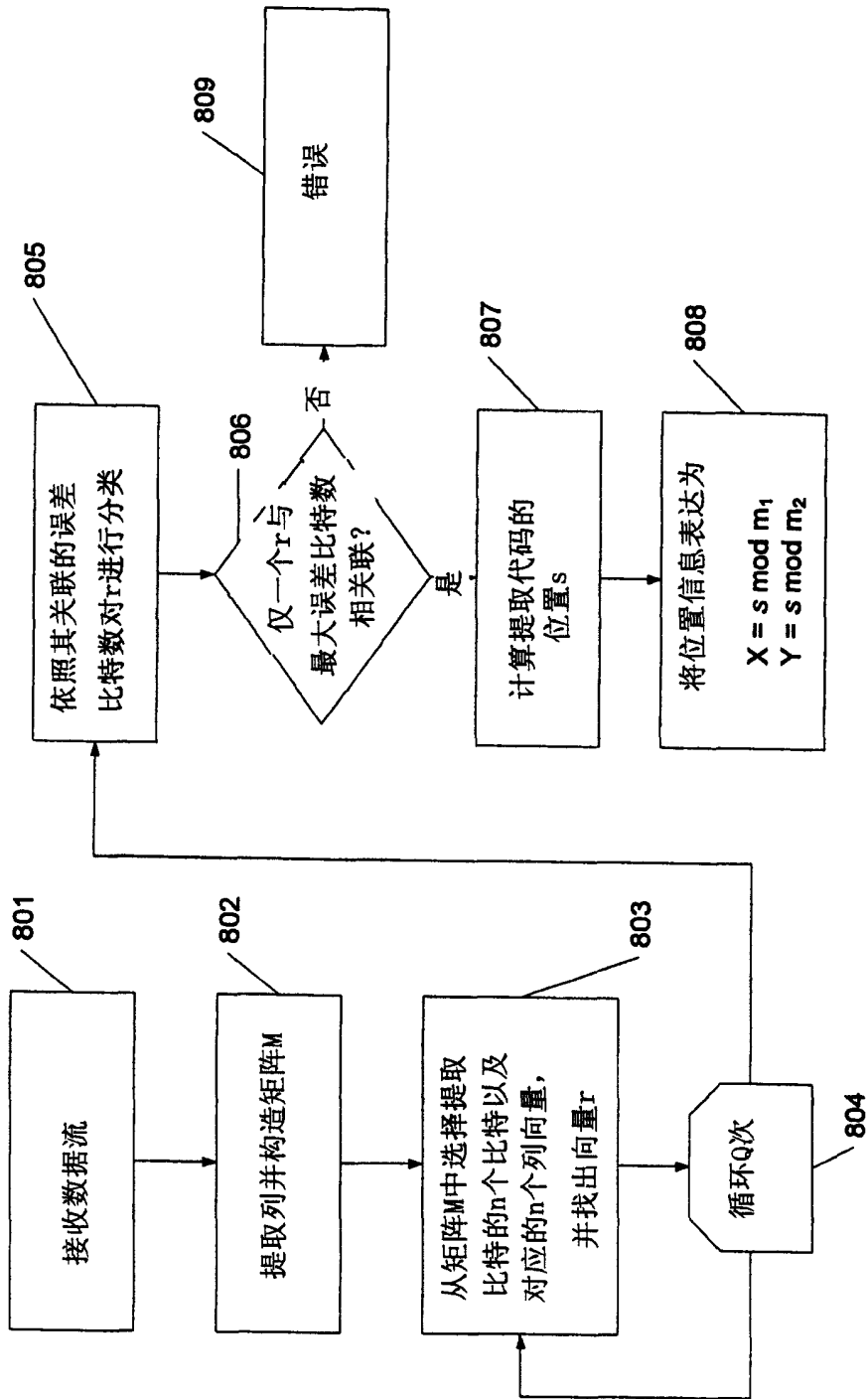


图 8

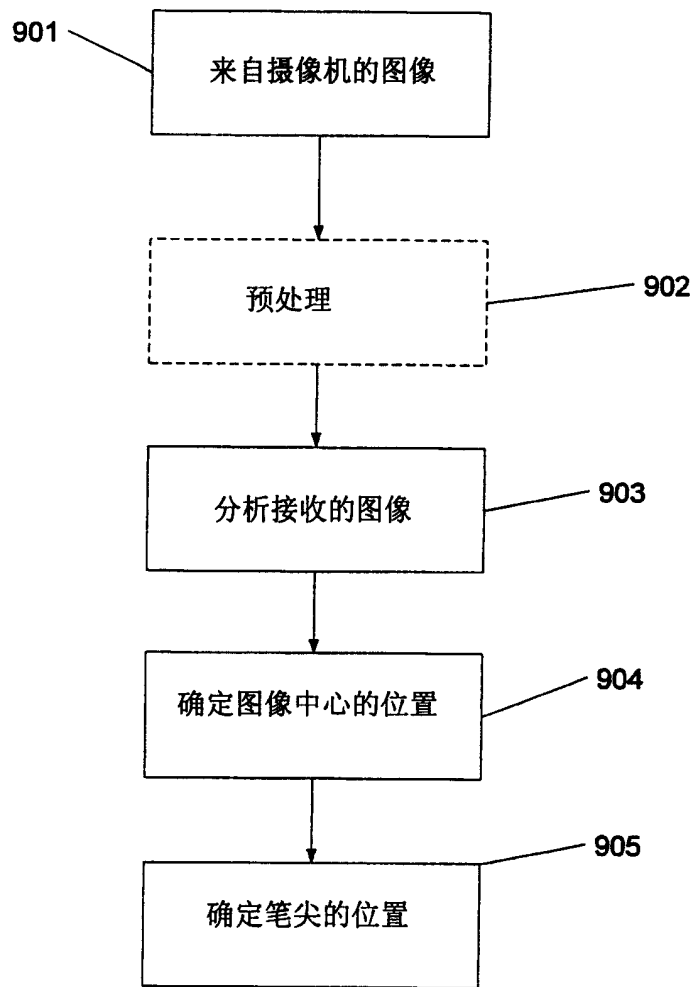


图 9

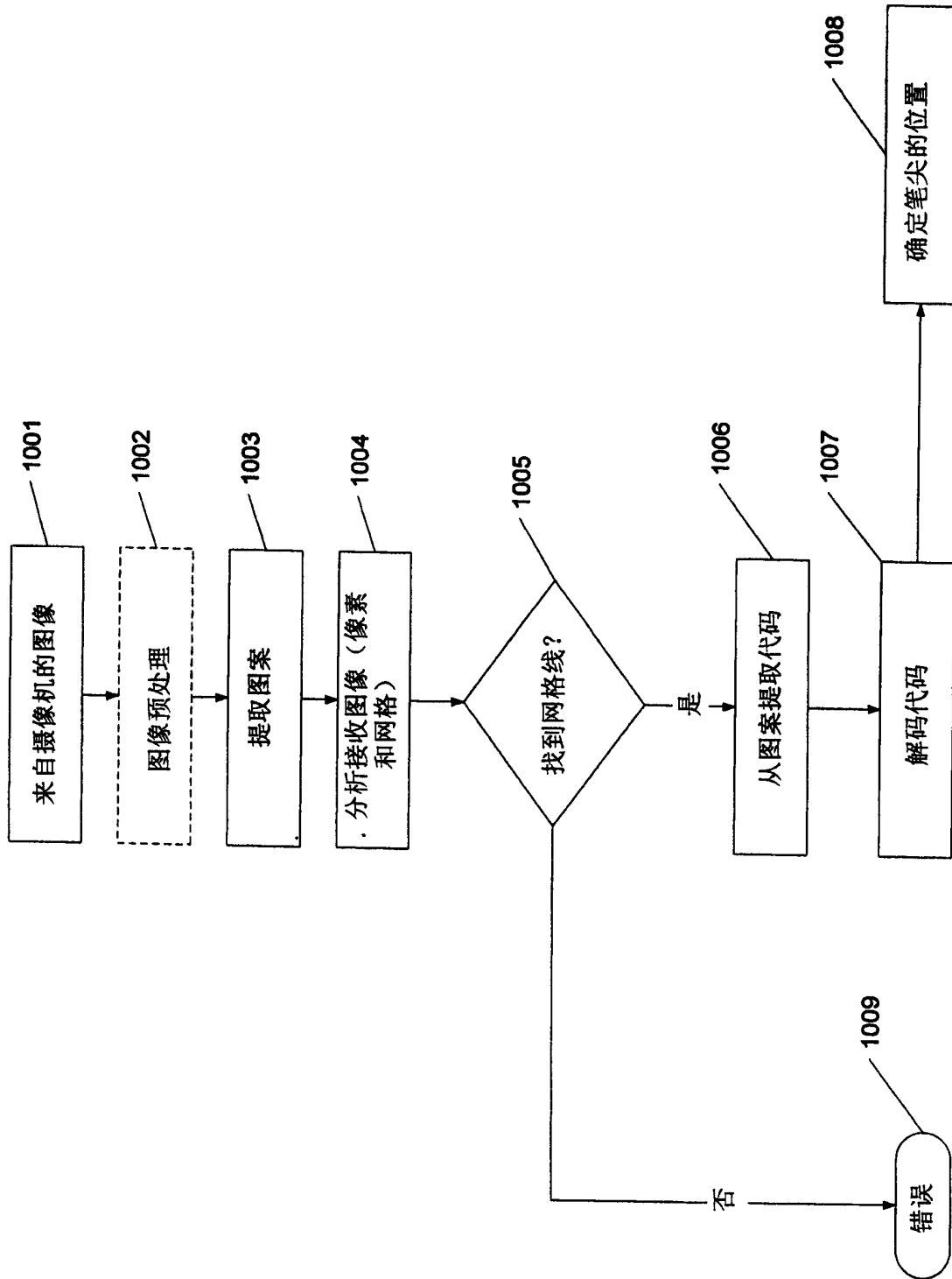


图 10

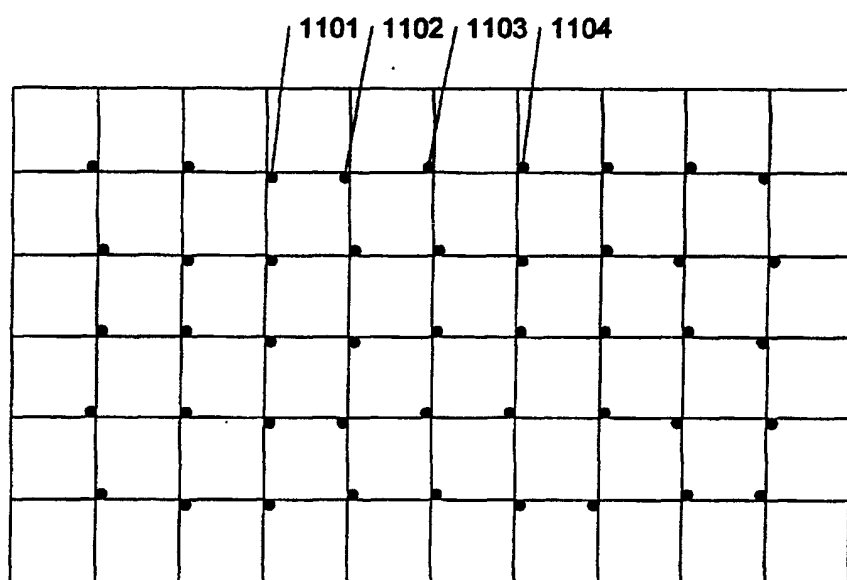


图 11
现有技术

1200

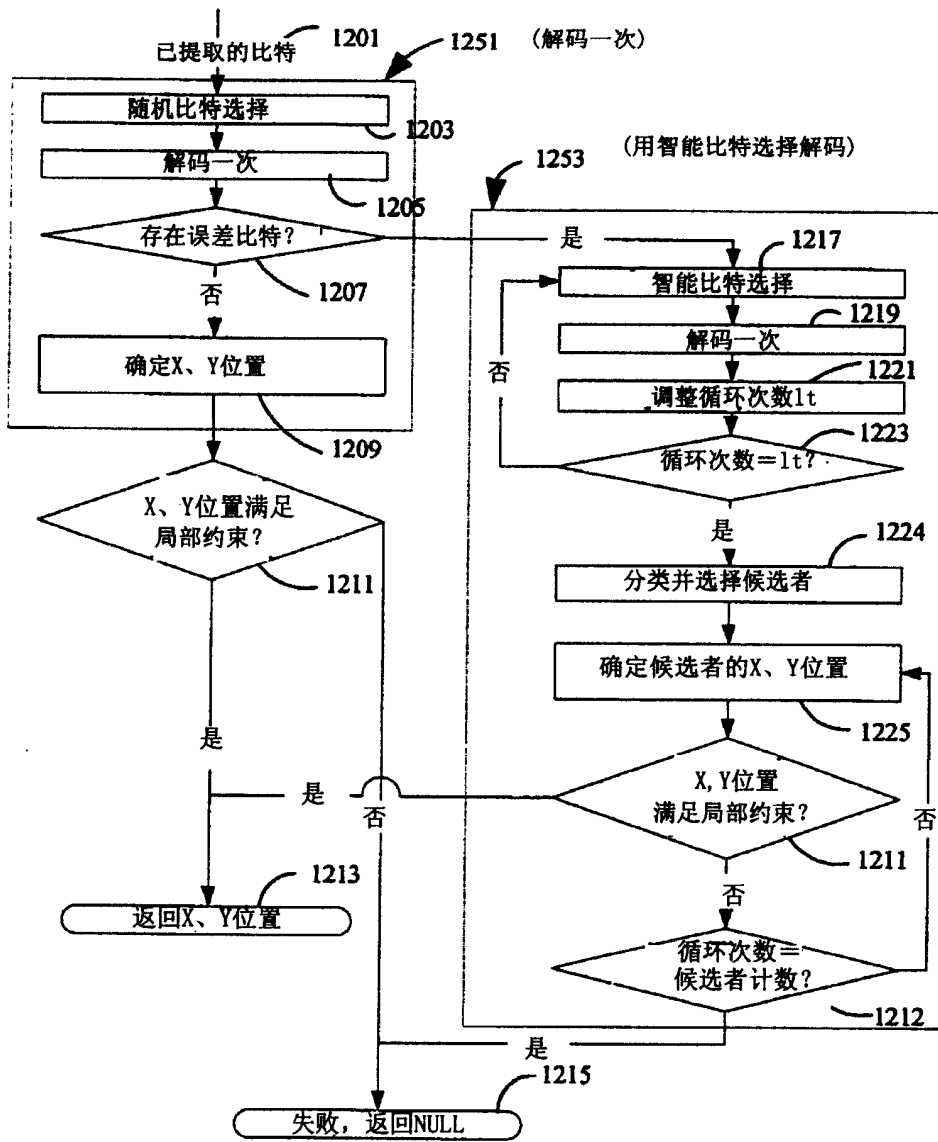


图 12

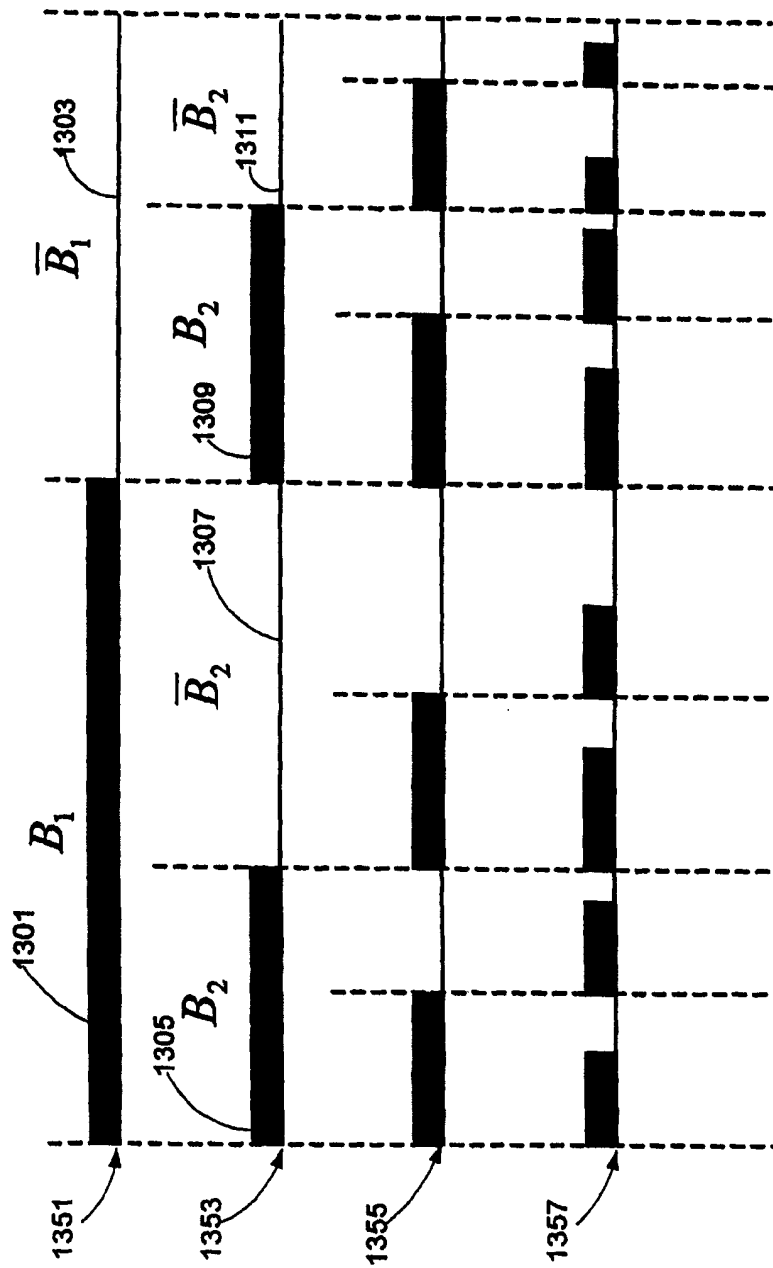


图 13

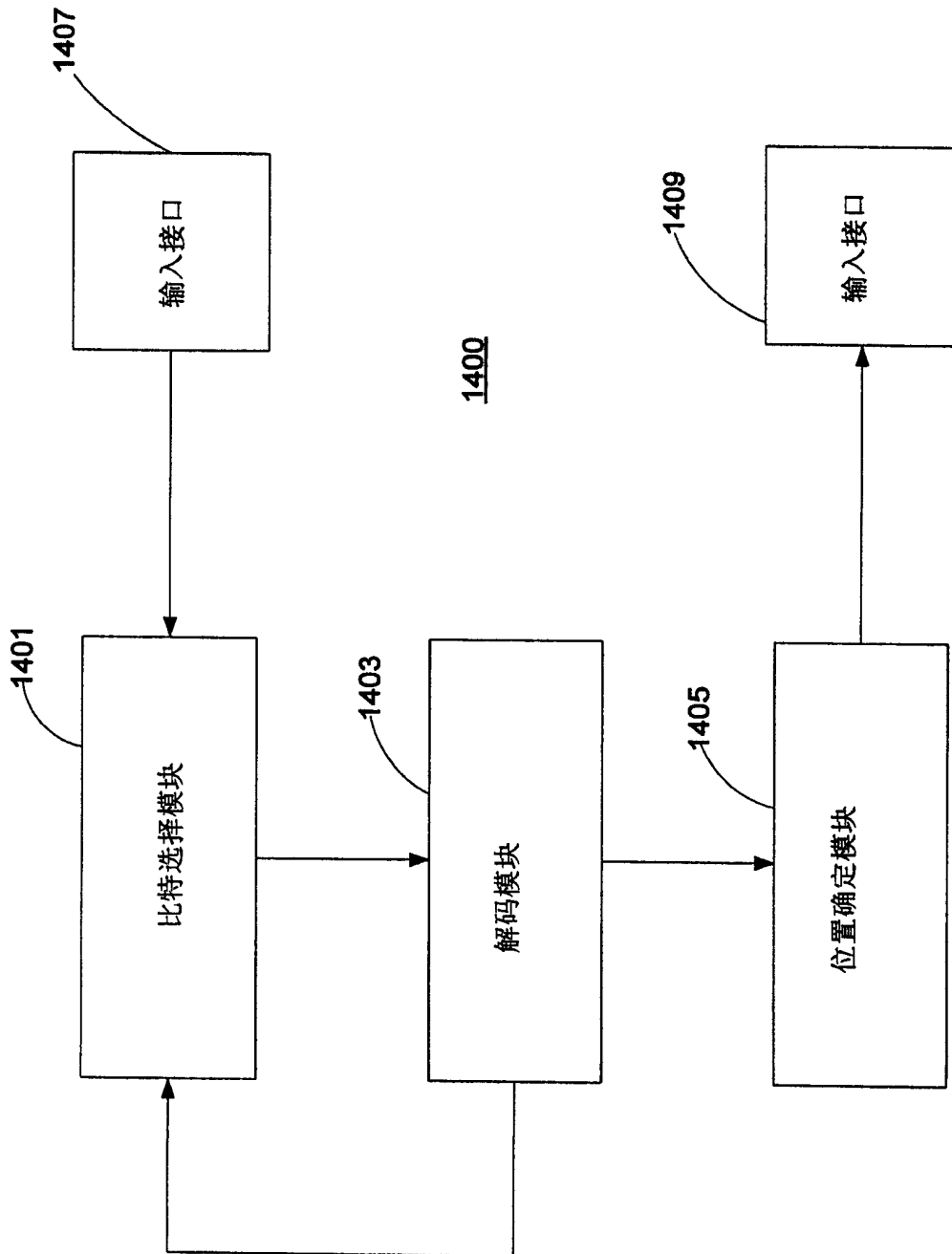


图 14

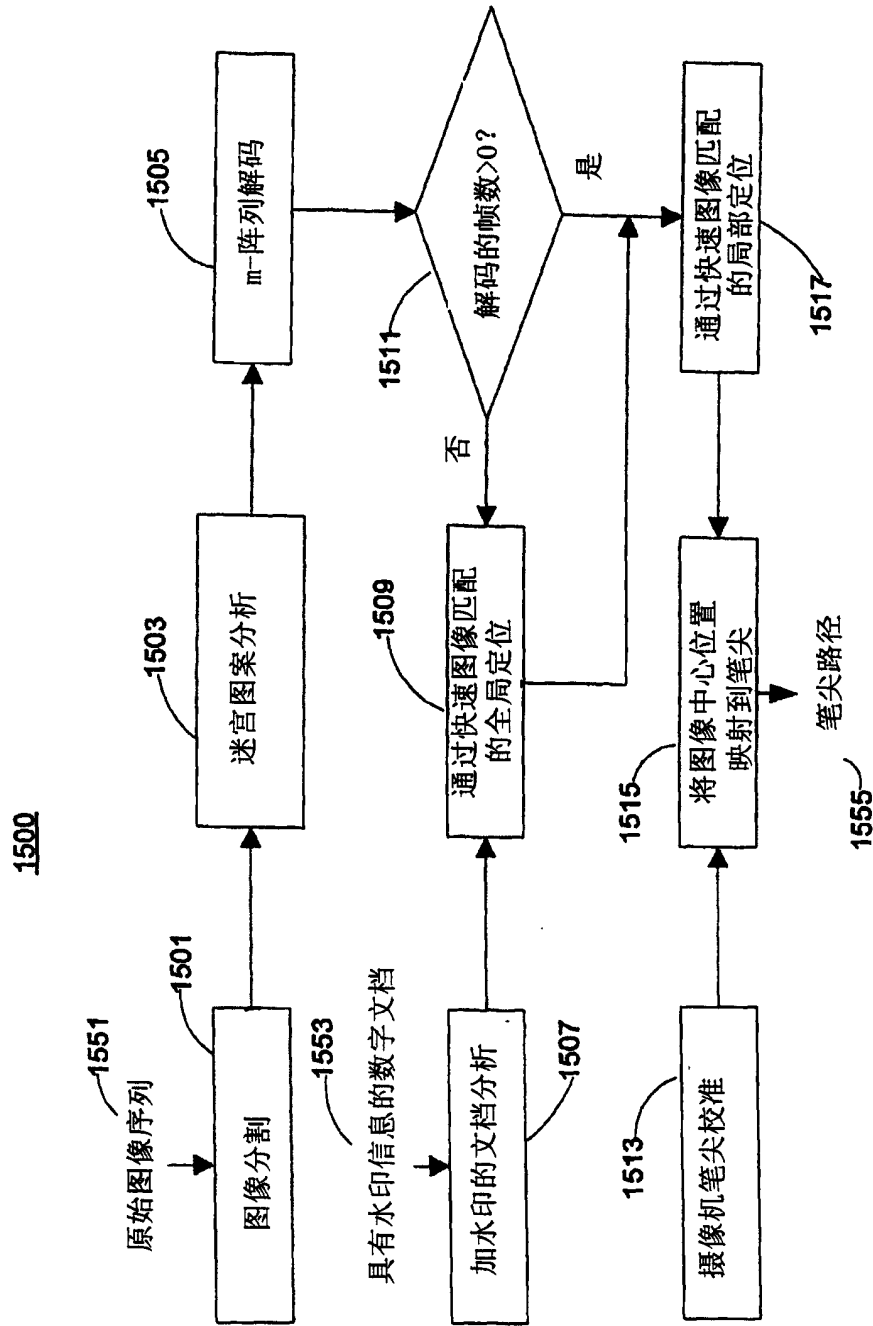


图 15

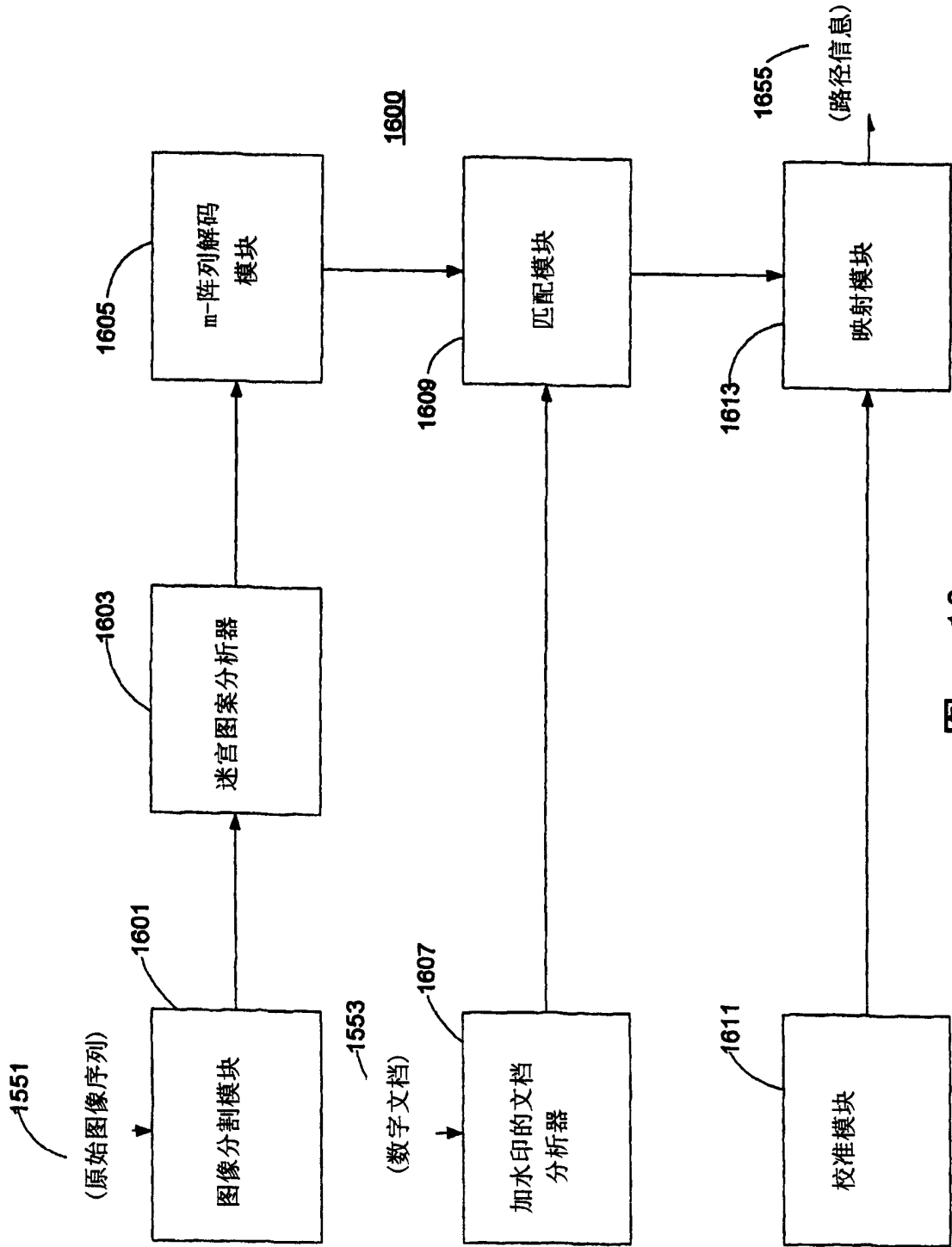


图 16

原始图像

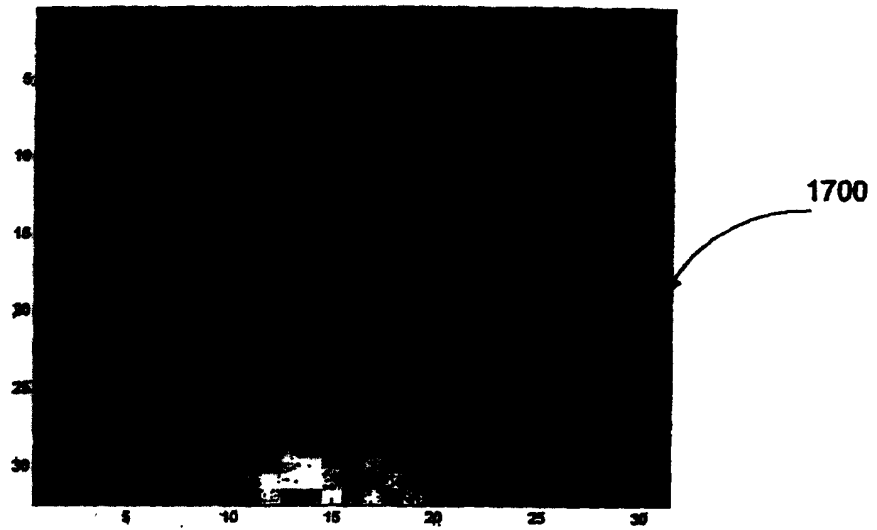


图 17

1803

原始图像

1801

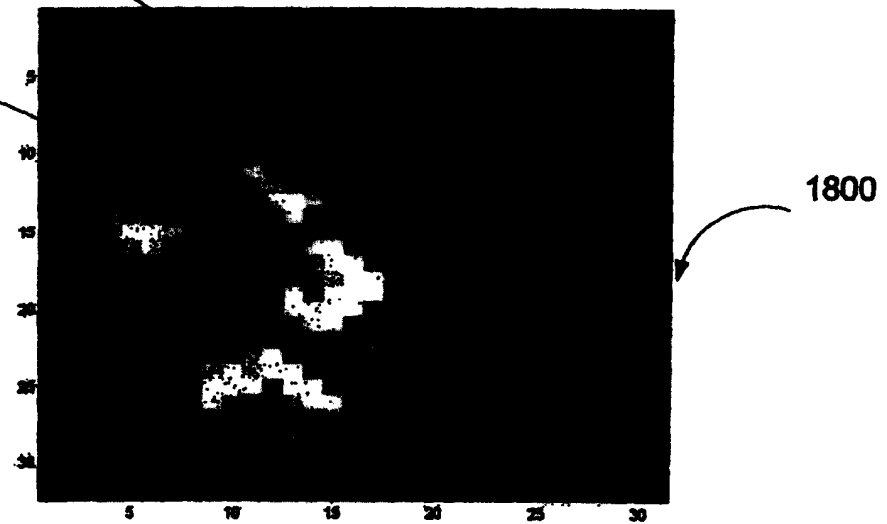


图 18

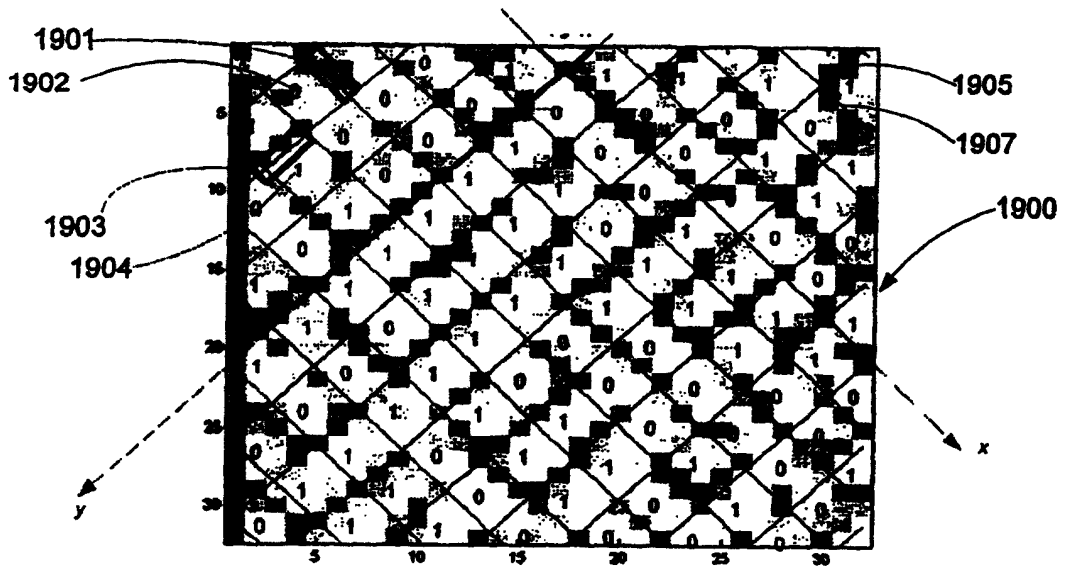


图 19

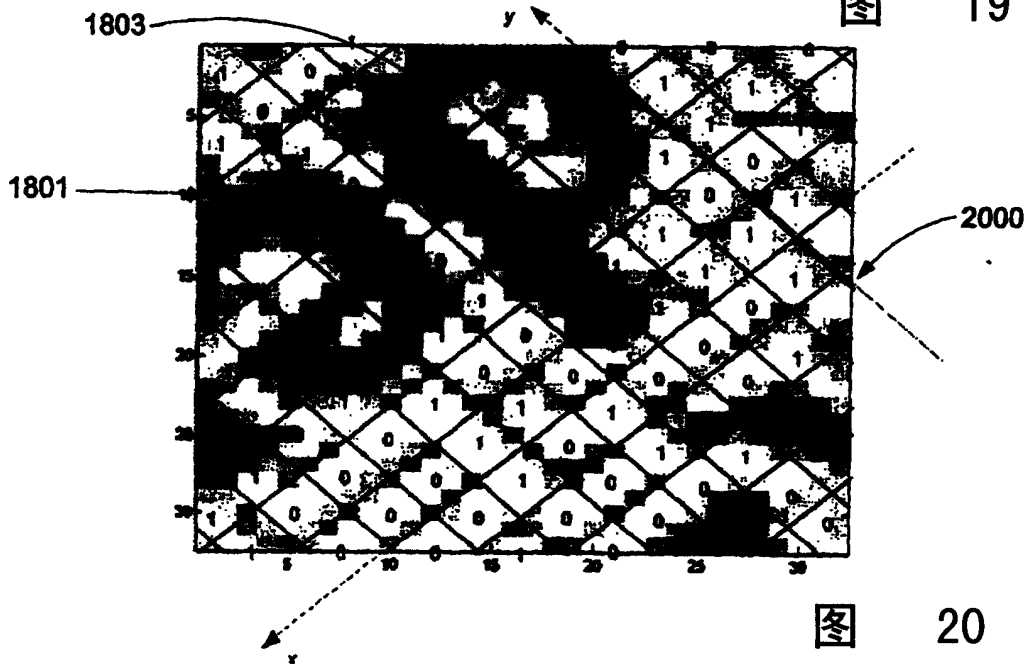


图 20

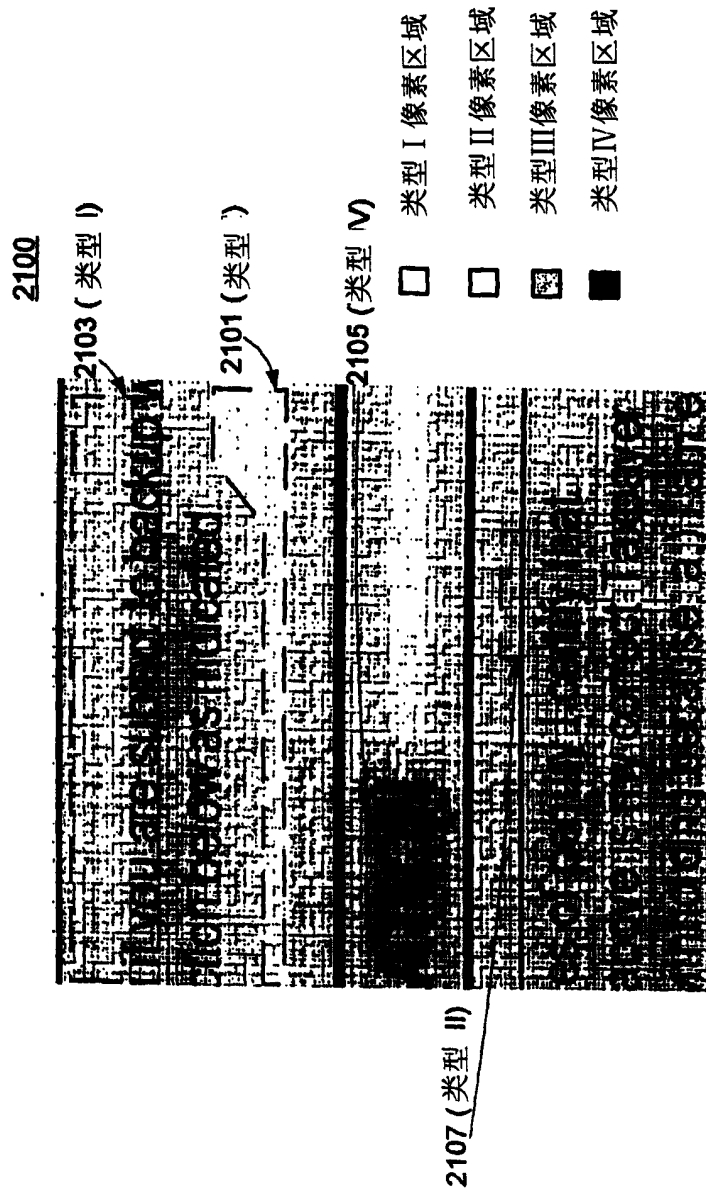


图 21

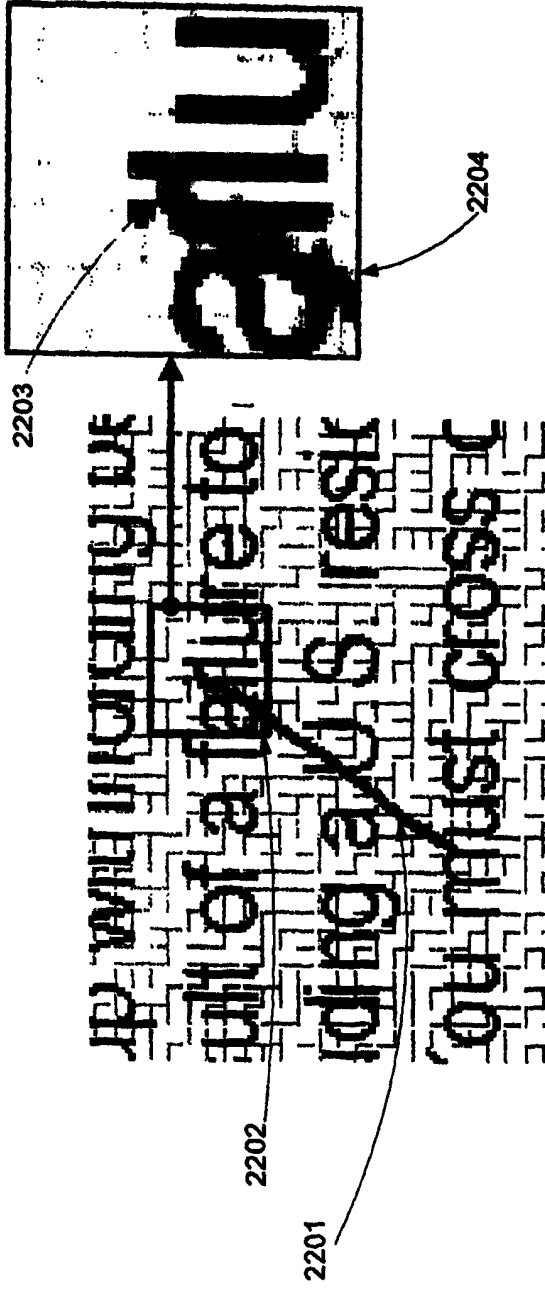


图 22

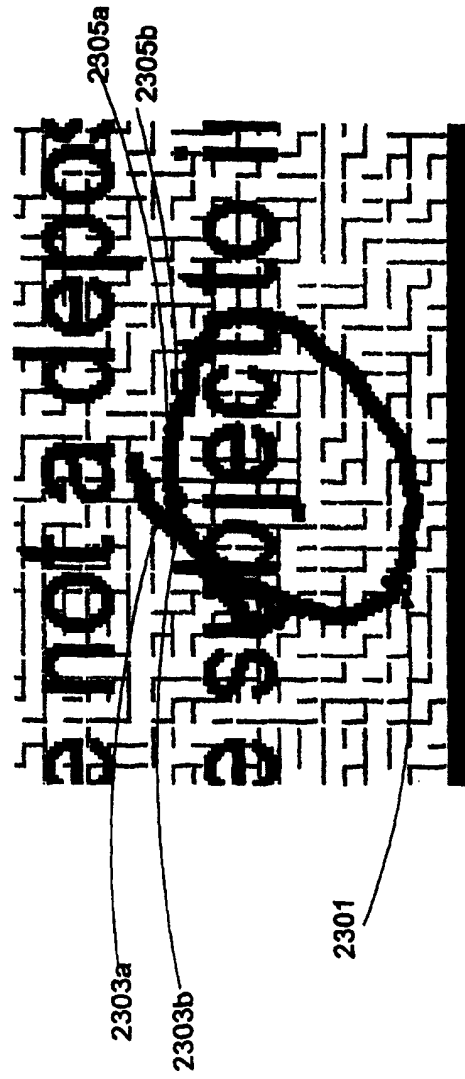


图 23

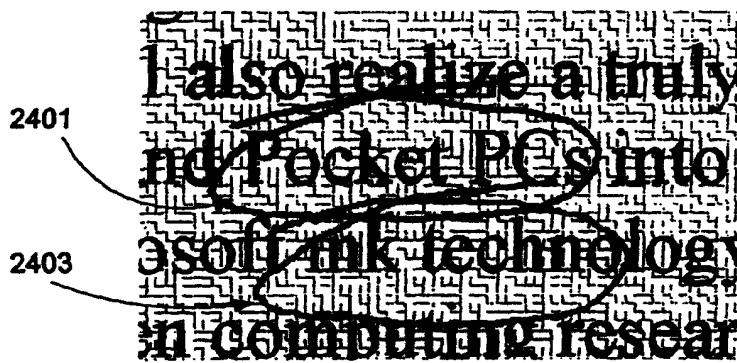


图 24



图 25