

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.  
G06F 3/033 (2006.01)



# [12] 发明专利说明书

专利号 ZL 200610092486.1

[45] 授权公告日 2008 年 12 月 10 日

[11] 授权公告号 CN 100442214C

[22] 申请日 2003.9.29

[21] 申请号 200610092486.1

分案原申请号 03127217.7

[30] 优先权

[32] 2002.10.31 [33] US [31] 10/284,422

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 王建 林舟晨 李越

[56] 参考文献

JP2000137569A 2000.5.16

US5502568A 1996.3.26

JP2000222116A 2000.8.11

审查员 孙泽竑

[74] 专利代理机构 上海专利商标事务所有限公司  
代理人 张政权

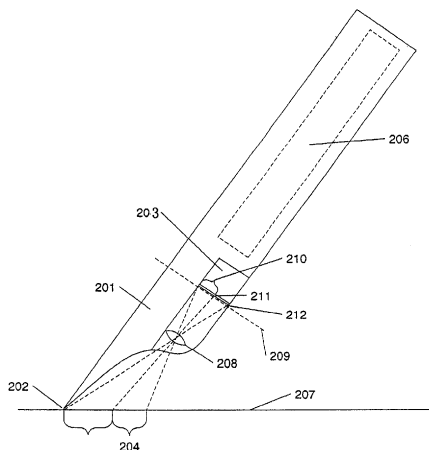
权利要求书 1 页 说明书 14 页 附图 10 页

[54] 发明名称

通过确定获取的图像的位置确定电子笔笔尖的位置的方法

[57] 摘要

描述了一种通过确定所获取的图像的位置来确定电子笔的笔尖的位置的方法，包括以下步骤：用电子笔的照相机获取图像；接收对应于所获取的图像的数据流；从所获取的图像中提取列；从所述列构建矩阵；根据距离将从所述列提取的代码排序；从所述代码确定距离矢量；确定所提取的代码的位置；以及根据所确定的所提取的代码的位置确定所述电子笔的笔尖的位置。



1. 一种通过确定所获取的图像的位置来确定电子笔的笔尖的位置的方法，其特征在于，包括以下步骤：

用电子笔的照相机获取图像；

接收对应于所获取的图像的数据流；

从所获取的图像中提取列；

从所述列构建矩阵；

根据距离将从所述列提取的代码排序；

从所述代码确定距离矢量；

确定所提取的代码的位置；以及

根据所确定的所提取的代码的位置确定所述电子笔的笔尖的位置。

2. 如权利要求 1 所述的方法，其特征在于，还包括以下步骤：

从所述列构建至少另一个矩阵；以及

从所述代码确定至少一第二矢量。

3. 如权利要求 1 所述的方法，其特征在于，还包括以下步骤：

在所述代码上进行错误校正。

## 通过确定获取的图像的位置确定电子笔笔尖的位置的方法

本申请是申请日为 2003 年 9 月 29 日、申请号为 03127217.7、发明名称为“2-D 阵列中的译码和差错校正”的专利申请的分案申请。

### 技术领域

本发明涉及使用数字笔和介质交互作用。本发明尤其涉及在和一个或多个表面交互作用时确定数字笔的位置。

### 背景技术

计算机用户习惯于使用鼠标和键盘作为和个人计算机交互作用的方法。虽然个人计算机与书面文件相比提供了大量便利,但多数用户继续使用打印的纸从事某些活动。一些这种活动包括阅读和标注书面文件。在标注的情况下,由于是由用户将标注记录到文件上,所以打印的文件变得更加重要。但是,给打印的文件标注的一个困难在于,之后需要将标注再次输入回文件的电子形式中。这要求原始用户或另一个用户辛苦地读完标注并将它们输入个人计算机中。在某些情况下,用户扫描入标注和原文,从而产生新的文件。这些反复步骤使打印的文件和文件的电子版本之间的交互作用变得在重复的基础上很难处理。此外,扫描入的图像常常是不能修改的。并没有办法将标注从原文中分开。这使得使用标注变得很麻烦。因此,需要一种处理标注的改进方法。

获取手写信息的一种技术是通过使用在书写时可以确定位置的笔。提供这种功能的一种笔是 Anoto 公司的 Anoto 笔。这种笔通过使用照相机来获取纸上用预定图形编码的图像。图像图形的一个实例如图 11 所示。Anoto 笔(Anoto 公司生产)用该图形来确定纸上笔的位置。但是,不清楚利用 Anoto 笔所采用的系统能使位置的确定多么有效。为了有效地确定获取的图像位置,需要一种提供所获取的图像的有效译码的系统。

### 发明内容

本发明的方面提供了至少一个上述问题的解决方法,从而能将所观察的图像

上的一个或多个位置定位。该所观察的图像可以是在打印的图像上或可以是显示的计算机图像上的图示。本发明的方面中包括允许有效的译码所捕获的图像的译码处理过程，它被用来有效地确定图像的位置。

本发明提供了一种通过确定所获取的图像的位置来确定电子笔的笔尖的位置的方法，包括以下步骤：用电子笔的照相机获取图像；接收对应于所获取的图像的数据流；从所获取的图像中提取列；从所述列构建矩阵；根据距离将从所述列提取的代码排序；从所述代码确定距离矢量；确定所提取的代码的位置；以及根据所确定的所提取的代码的位置确定所述电子笔的笔尖的位置。

通过附图和相关描述可以理解本发明的这些或其它方面。

#### 附图说明

结合附图可以更好地理解本发明的上述概述，以及以下较佳实施例的详细描述，这些附图作为实例，但不作为本发明的限制。

图 1 示出可以结合本发明的实施例使用的计算机的一般描述。

图 2A 和 2B 示出根据本发明的实施例的图像获取系统和相应的所获取图像。

图 3A 至 3F 示出根据本发明的实施例的各种序列和折叠技术。

图 4A 至 4E 示出根据本发明的实施例的各种编码系统。

图 5A 至 5D 示出和根据图 4A 和 4B 的编码系统相关的 4 种可能的合成边角 (resultant corner)。

图 6 示出根据本发明的实施例的所获取的图像部分的旋转。

图 7 示出结合图 4A 至 4E 的编码系统使用的各种旋转角度。

图 8 示出根据本发明的实施例的用于确定所获取阵列的位置的处理过程。

图 9 示出根据本发明的实施例的用于确定所获取图像的位置的方法。

图 10 示出根据本发明的实施例的用于确定所获取图像的位置的另一个方法。

图 11 示出文件中的编码空间的常规表示。

#### 具体实施方式

本发明的方面涉及确定相对更大的图像的所获取图像位置。这里描述的位置确定方法和系统可以结合多功能笔使用。

为便于读者阅读，用副标题隔开以下的描述。这些副标题包括：术语、常规用途的计算机、图像获取笔、阵列的编码、译码、差错校正和位置确定。

## 术语

笔-可以或不可以包括储存墨水的任何书写工具。在某些实例中，不含墨水的触笔可以用作根据本发明实施例的笔。

照相机-可以从纸或任何其它的介质获取图像的图像获取系统。

## 常规用途的计算机

图 1 是可以用来实现本发明的各方面的常规用途数字计算机环境实例的原理框图。图 1 中，计算机 100 包括处理单元 110、系统存储器 120 和将包括系统存储器 120 的各种系统部件连接到处理单元 110 的系统总线 130。系统总线 130 可能是多种类型总线结构的任一种，包括存储器总线或存储器控制器、外围设备总线、及使用各种总线结构的任一种的局部总线。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)，包括在 ROM140 中存储的有助于在诸如在启动期间，计算机 100 内的元件之间传送信息的基本程序 (routine)。计算机 100 还包括用于写入硬盘或从中读出 (未示出) 的硬盘驱动器 170、用于写入可移动磁盘 190 或从中读出的磁盘驱动器 180，和用于写入诸如 CD ROM 或其它光学介质的可移动光盘 192 或从中读出的光盘驱动器 191。硬盘驱动器 170、磁盘驱动器 180 和光盘驱动器 191 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 分别连接到系统总线 130。这些驱动器和其相关的计算机可读介质提供了用于个人计算机 100 的计算机可读指令、数据结构、程序模块和其它数据的非易失性存储。本技术领域内的熟练的技术人员将理解，可以存储由计算机访问的数据的其它类型计算机可读介质，诸如磁带盒、闪存卡、数字视频磁盘、伯努力盒式磁带、随机存取存储器 (RAM)、只读存储器 (ROM) 等等，还可以用于实例操作环境中。

大量的程序模块可以存储在硬盘驱动器 170、磁盘 190、光盘 192、ROM140 或 RAM150 上，这些程序模块包括操作系统 195、一个或多个应用程序 196、其它的程序模块 197 和程序数据 198。用户可以通过诸如键盘 101 和定位设备 102 将命令和信息输入计算机 100。其它的输入设备 (未示出) 可以包括麦克风、操纵杆、游戏垫、卫星碟 (satellite dish)、扫描仪等等。这些和其它输入设备常常通过和系统总线连接的串行口接口 106 连接到处理单元 110，但也可以由其它接口连接，诸如并行口、游戏口或通用串行总线 (USB)。此外，这些装置可以通过适当的接口 (未示出)

直接连接到系统总线。监视器 107 或其它类型的显示器装置也通过诸如视频适配器 108 的接口连接到系统总线 130。除了监视器, 个人计算机通常包括其它外围输出装置(未示出), 诸如扬声器和打印机。在较佳的实施例中, 为了数字化地获取徒手输入而提供了笔形数字转换器 165 和附随笔或触笔 166。虽然示出了笔形数字转换器 165 和串行口之间的直接连接, 但实际上, 如该技术领域中已知的, 笔形数字转换器 165 可以通过并行口或其它接口以及系统总线 130 直接和处理单元 110 连接。此外, 虽然示出数字转换器 165 和监视器 107 分开, 但较佳地, 数字转换器 165 的可用输入区域和监视器 107 的显示器区域是协同扩展的。此外, 数字转换器 165 可以结合到监视器 107 中, 或可以作为分开的装置加入或另外被附加到监视器 107。

计算机 100 可以逻辑连接到一个或多个远程计算机, 诸如远程计算机 109 来在网络环境中工作。远程计算机 109 可以是服务器、路由器、网络 PC、对等装置或其它普通的网络节点, 并通常包括许多或全部以上所述和计算机 100 相关的元件, 虽然在图 1 中只说明了存储装置 111。图 1 中描述的逻辑连接包括局域网(LAN) 112 和广域网(WAN) 113。这些网络环境在办公室、企业范围的计算机网络、内联网和因特网中是很普通的。

当在 LAN 网络环境总使用时, 计算机 100 通过网络接口或适配器 114 和局域网 112 连接。当用于 WAN 网络环境时, 个人计算机 100 通常包括调制解调器 115 或用于建立和诸如因特网的广域网 113 的连接的其它装置。调制解调器 115 可以是内部或外部的, 并通过串行口 106 和系统总线 130 连接。在网络环境中, 描述中和个人计算机 100 相关的程序模块或其部分可以存储于远程存储器装置。

可以理解, 所示的网络连接是说明性的, 而可以使用用于建立计算机之间的通信连接的其它技术。假定存在各种已知的协议中的一种, 诸如 TCP/IP、以太网、FTP、HTTP、蓝牙、IEEE 802.11x 等等, 则系统可以以客户机-服务器结构工作来允许用户从基于网络的服务器检索网页。可以使用任何常规的网络浏览器来显示和处理网页上的数据。

### 图像获取笔

本发明的方面包括以显示的格式放置编码的数据流。该显示的格式可以是打印的纸(或其它物理介质)或可以是放映结合了其它图像或图像组的编码数据流的显示器。例如, 编码数据流可以表现为纸上的物理的图形图像或显示器屏幕上的覆盖所显示的图像的图形图像, 或可以是显示器屏幕上的物理的(不能修改)图形图像

(因此由笔获取的任何图像部分在显示器屏幕上是可以定位的)。

所获取的图像位置的确定可以用于确定用户和纸、介质或显示器屏幕交互作用的位置。在本发明的某些方面中，笔可以是在纸上书写的墨水笔。在其它方面，笔可以是用户用来在计算机显示器表面上书写的触笔。在已知文件上的编码图像或支持计算机屏幕上显示器的文件时，任何的交互作用可以提供回系统。通过重复地获取照相机的位置，系统可以追踪由用户控制的触笔的移动。显示的或打印的图像可以是和空白的或内容丰富的纸相关的水印或可以是和显示的图像或覆盖屏幕或构建于屏幕内的固定编码相关的水印。

图 2A 和 2B 示出具有照相机 203 的笔 201 的说明性实例。笔 201 包括可以包含或不包含墨水容器的笔尖 202。照相机 203 从表面 207 获取图像 204。笔 201 还可以包括另外的传感器和/或处理器，如虚线框 206 所表示的。这些传感器和/或处理器 206 也可以包含将信息(例如，通过蓝牙或其它无线协议)传送到其它笔 201 和/或个人计算机的能力。

图 2B 表示由照相机 203 所观察的图像。在一个说明性的实例中，照相机 203 的视场为  $32 \times 32$  像素(其中  $N=32$ )。因此，图 2B 示出的 32 像素长、32 像素宽的视场。N 的大小可以根据所需的图像分辨率程度来调整的。虽然为了便于说明将此处的照相机 203 的视场示为正方形，但该视场还包含本技术领域中的其它形状。

从照相机 203 到笔 201 的输入可以定义为一组图像帧  $\{I_i\}$ ,  $i=1, 2, \dots, A$ , 其中  $I_i$  由笔 201 在采样时间  $t_i$  时获取。采样率可以是固定的或可以根据文件的大小变化的。所捕获的图像帧的大小可大可小，这取决于文件的大小和所需的精确度。同样，照相机图像大小可以根据所查找的文件的大小确定。

由照相机 203 获取的图像可以直接由处理系统使用或可以进行预过滤。该预过滤可以在笔 201 内进行或可以在笔 201 之外进行(例如，在个人计算机内)。

图 2B 的图像大小为  $32 \times 32$  像素。如果每个编码单元的大小为  $3 \times 3$  像素，则所获取的编码单元数约为 100 个单元。如果编码单元的大小为  $5 \times 5$ ，则所获取的编码单元数约为 36 个。

图 2A 还示出了图像平面 209，在平面 209 上形成来自位置 204 的图形的图像 210。从目标平面 207 上的图形接收到的光由透镜 208 聚焦。透镜 208 可以是单个透镜或多部分的透镜系统，但这里为了简单只表示为单个透镜。图像获取传感器 211 获取图像 210。

图像传感器 211 可以大到足以获取图像 210。可供选择地，图像传感器 211 可以大到足以在位置 212 获取笔尖 202 的图像。作为参考，在位置 212 的图像被称作虚拟笔尖(virtual pen tip)。注意到相对图像传感器 211 的虚拟笔尖位置是固定的，因为笔尖、透镜 208 和图像传感器 211 之间的关系是恒定的。因为变换是从虚拟笔尖 212 的位置(由  $L_{\text{virtual-pen tip}}$  表示)到实际笔尖 202 的位置(由  $L_{\text{pen tip}}$  表示)，所以可以确定相对所获取的图像 210 的实际笔尖的位置。

以下的变换  $F_{S \rightarrow P}$  将由照相机获取的图像变换到纸上的实际图像：

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

在书写期间，笔尖和纸在同一平面上。因此，从虚拟笔尖到实际笔尖的变换也是  $F_{S \rightarrow P}$ ：

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

变换  $F_{S \rightarrow P}$  可以被称作透视变换。这简化为

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

来作为  $F_{S \rightarrow P}$  的估计，其中  $\theta$ 、 $s_x$  和  $s_y$  是在位置 204 获取的图形的旋转和两个方向上的伸缩比例。此外，可以通过将获取的图像和纸上相应的背景图像相匹配来将  $F'_{S \rightarrow P}$  的提炼为  $F_{S \rightarrow P}$ 。“提炼”意味着通过一种称为递归法的优化算法得到更精确的透视矩阵  $F_{S \rightarrow P}$  (8 个参数)。该递归法将矩阵  $F'_{S \rightarrow P}$  作为初始值。 $F_{S \rightarrow P}$  比  $F'_{S \rightarrow P}$  更精确地描述 S 和 P 之间的变换。

接着，可以通过校准确定虚拟笔尖的位置。

将笔尖 202 放置在纸上的已知位置  $L_{\text{pen tip}}$ 。接着，使笔倾斜，使照相机 203 能获取不同的笔姿态的一组图像。对于所获取的每个图像。可以得到变换  $F_{S \rightarrow P}$ 。从该变换，可以得到笔尖的虚拟图像的位置  $L_{\text{virtual-pen tip}}$ ：

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

以及， $F_{P \rightarrow S} = 1/F_{S \rightarrow P}$

通过对从每个图像得到的  $L_{\text{virtual-pen tip}}$  取平均，可以确定虚拟笔尖  $L_{\text{virtual-pen tip}}$  的精确的位置。

现在知道了虚拟笔尖的位置  $L_{\text{virtual-pen tip}}$ 。还可以从所获取的图像得到变换  $F_{S \rightarrow P}$ 。最终，可以使用该信息来确定实际笔尖  $L_{\text{pen tip}}$  的位置：

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



### 阵列的编码

可以通过折叠一维序列来构建二维阵列。二维阵列的任何部分可以用来确定其在整个二维阵列中的位置。一个所关心的问题在于由照相机获取的图像部分在大小上受到限制。因此，需要能从所获取的图像或几个所获取的图像中确定位置。产生了额外的复杂性，即在所获取的图像中出现的错误损害在所获取的图像中的某些点的可用性。可以使用非重复的序列来创建阵列，从而使所获取的图像的一个部分和两个或多个二维阵列中的位置相关联的可能性最小。所创建序列的一个属性在于该序列在一长度(或窗口) $n$ 上不重复。以下将描述一维序列的创建，随后是将该序列折叠成为阵列。

### 序列构建

可以将一组数字用作编码系统的起始点。例如，一序列(也称作  $m$ -序列)可以表示成域  $F_q$  中的  $q$  元素集。这里， $q=p^n$ ，其中  $n \geq 1$  而  $p$  是质数。该序列或  $m$  序列可以通过各种不同的技术产生，这些技术包括，但不限于，多项式除法。使用多项式除法，序列可以定义如下：

$$\frac{R_l(x)}{P_n(x)} \quad (1)$$

其中  $P_n(x)$  是域  $F_q[x]$  (具有  $q$  个元素) 中  $n$  次的本原多项式(primitive polynomial)。  $R_l(x)$  是域  $F_q[x]$  中的  $l$  次(其中  $l < n$ ) 的非零多项式。序列可以用迭代过程产生，它使用两个步骤：第一，将两个多项式相除(得到域  $F_q$  的一个元素)，第二，将余项乘以  $x$ 。当输出开始重复时，该计算停止。可以使用线形反馈移位寄存器来实现该过程，如 Douglas W. Clark 和 Lih-Jyh Weng 的文章 "Maximal and Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms," IEEE 计算机学报 43.5 (1994 年 5 月, pp560-568) 中所述。在该环境中，在序列的循环移位和多项式  $R_l(x)$  之间建立一种关系：改变  $R_l(x)$  仅仅使序列循环地移位而每个循环移位对应一个多项式  $R_l(x)$ 。所造成的序列的一个属性在于，遍及宽度(或长度) $n$ ，任何部分在该序列中只存在一次。

以上所描述的处理过程仅是用来产生非重复序列的各种处理过程中的一个。

### 阵列构建

可以用于创建图像(该图像的一部分由照相机获取)的阵列(或  $m$  阵列)是一维序列(或  $m$  序列)的延伸。使  $A$  为  $(m_1, m_2)$  周期的阵列, 即  $A(k+m_1, l)=A(k, l+m_2)=A(k, l)$ 。当  $n_1 \times n_2$  窗口移位通过一个  $A$  周期, 则遍及  $F_q$  所有非零  $n_1 \times n_2$  矩阵出现一次且只有一次。该属性可以称作“窗口属性”, 其中每个窗口都是唯一的。随后,  $A$  可以表示成周期为  $(m_1, m_2)$  阶次为  $(n_1, n_2)$  的阵列( $m_1$  和  $m_2$  是阵列中水平方向和垂直方向的比特数)。

二进制阵列(或  $m$  阵列)可以通过折叠序列来构建。一种方法是得到一个序列, 然后将它折叠成大小  $m_1 \times m_2$ , 其中阵列的长度是  $L=m_1 \times m_2=2^n-1$ 。可供选择地, 可以用打算覆盖的空间的预定大小来开始(例如, 一张纸, 30 张纸或计算机监视器的大小), 确定面积  $(m_1 \times m_2)$ , 随后使用该大小且使  $L \geq m_1 m_2$ , 其中  $L=2^n-1$ 。

可以使用各种不同的折叠技术。例如, 图 3A 到 3C 示出 3 种不同的序列。它们每一个都可以折叠成图 3D 所示的阵列。这 3 种不同的折叠方法示为图 3D 中的覆盖, 并示为图 3E 和 3F 中的光栅路径(raster path)。图 3D 所示的折叠方法可以表示成当  $\gcd(m_1, m_2)=1$ , 而  $L=2^n-1$ 。

为了创建如图 3D 所示的折叠方法, 先创建长度为  $L$  且阶次为  $n$  的序列  $\{a_i\}$ 。接着, 从序列  $\{a_i\}$  通过使该阵列的每一比特通过方程式 1 计算来创建大小为  $m_1 \times m_2$  的阵列  $\{b_{kl}\}$ :

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

可供选择地, 该折叠方法可以表示为将序列置于阵列的对角线上, 随后, 当到达边界时从另一侧继续。

图 4A 示出针对可以和图 3D 的阵列一起使用的第一比特和第二比特的实例编码技术。可以理解, 也可以使用其它的编码技术。例如, 如图 11 所示的可供选择的编码技术。

参考图 4A, 第一比特 401(例如, “1”)由一列暗墨水表示。第二比特 402(例如, “0”)由一行暗墨水表示。可以理解可以使用任何颜色的墨水来表示各种比特。只要求所选的墨水的颜色提供与介质的背景呈强烈的对比, 以使图像获取系统能够区分。图 4A 中的比特由  $3 \times 3$  的单元格矩阵表示。可以根据图像获取系统的大小和分辨率将该矩阵的大小更改为任意大小。可供选择的比特 0 和 1 的表示如图 4C-4E 所示。可以理解, 用于图 4A-4E 的实例编码的零或一的表示可以切换而没有影响。图 4C 示出在占据交叉排列的两行或列的比特表示。图 4D 示出可供选择的在行和列

中以虚线形式的像素排列。最后，图 4E 示出不规则间隔形式的行和列的像素表示（例如，两个暗框接着一个亮框）。

参考图 4A，如果比特由 3 乘 3 的矩阵表示，而图像系统在该 3x3 区域中探测到一个暗行和两个白行，则表示 0(或 1)。如果图像被探测到具有一个暗列和两个白列，则表示 1(或 0)。

这里，使用超过一个块或像素或点来表示一个比特。使用单个像素(或块或比特)来表示一个比特是很脆弱的。灰尘、纸上的折痕、不平整的表面等为读取数据单元的单个比特表示增添了困难。但是，可以理解，可以将不同的方法用于图形化地表示表面上的阵列。图 4C 到 4E 示出了一些方法。可以理解，还可以使用其它的方法。图 10 中阐述了只使用空间移位点(space-shifted dot)的一种方法。

比特流被用来产生图 4B 的图形 403。图形 403 包括 12 行和 18 列。这些行和列由转换为使用比特表示 401 和 402 的图形化表示的比特流形成。图 4B 可以视为具有以下比特表示：

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

#### 译码

当人们用图 2A 所示的笔靠近编码的图形书写或移动时，照相机获取图像。随后，处理该图像来确定和编码图像的全部表示相关的所获取图像的方向，并提取构成编码图像的比特。

图 4B 包括来自图像 403 的像素块的表示。所显示图像 404-411 示出 5x5 的像素块。像素块 404 示出在白行之间的一暗行。像素块 405 示出在白列之间的一暗列。像素块 406 示出左下边角。像素块 407 示出右上边角。像素块 408 示出在左侧具有半个暗行的暗列。像素块 409 示出一暗行，该暗行上具有半个暗列。像素块 410 示出半个暗行。像素块 411 示出半个暗列。分析像素块的组合，可以理解，所有的像素组合可以通过像素块 404-411 中找到的图像部分来形成。

此外，可以尝试用更小的像素块集来表示所有可能的像素组合。不用更多的，可以预料图 5A-5D 所示的 4 个块中的每一个都可以在图像 403 中找到。但是，在 8 个像素块 404-411 中只存在 3 种边角。在该实例中，没有图 5A 所表示的像素块。通过一事实，即可以选择图像部分 401 和 402 来减少边角的类型，出现了根据所缺少边角的类型来确定所获取图像的方向的可能。

继续到图 6, 可以分析由照相机获取的图像 601 并确定其方向, 从而可以说明图像 601 实际表示的位置。首先, 检查图像 601 来确定需要旋转该图像的角度  $\theta$  以便将像素水平方向和垂直方向对准。注意到, 可供选择的格子对准可能包括将底线格子旋转到非水平和垂直排列(例如, 45 度)。使用非水平和垂直排列可以提供降低来自用户的视觉迷惑(distracton)的可能的的好处。简单起见, 格子的方向(水平的和垂直的和将底线格子倾斜到其它方向)共同被称作预定格子方向。

接着, 分析图像 601 来确定所缺少的边角。将图像 601 旋转到准备用来译码 603 的图像所需的旋转量  $\phi$  示为  $\phi = (\theta \text{ 加上旋转量 } \{ \text{由所缺少的边角确定} \})$ 。该旋转量由图 7 中的方程式示出。再参考图 6, 角度  $\theta$  首先由像素的布局(layout)确定以便到达像素水平的和垂直的(或其它预定格子方向)排列, 而图像如 602 所示来旋转。随后, 进行分析来确定所缺少的边角, 并将图像 602 旋转成 603 所示的图像, 从而形成用于译码的图像。这里, 图像被逆时针旋转了 90 度, 从而图像 603 的位置可以恢复。

可以理解, 可以在图像 601 旋转之前或之后应用旋转角度  $\theta$  来考虑所缺少的边角。

最后, 图像 603 中的代码被读出并和用来创建图像 403 的初始数据流相关联。这种关联可以以多种方法进行。例如, 可以通过递归方法进行, 在递归方法中, 恢复的比特流和初始比特流中所有其它的比特流片段比较。其次, 可以在恢复的比特流和初始比特流之间, 例如, 通过使用这两个比特流之间的汉明间距进行统计分析。可以理解, 可以使用各种方法来确定在初始比特流中所恢复的比特流的位置。

一旦具有所恢复的比特, 就需要将所获取的图像在初始阵列中(例如, 图 4B 中所示的)定位。由于具有大量的操作, 因此在整个阵列中确定比特部分的位置的过程是很复杂的。首先, 可以从照相机的理想获取来使实际获取的比特模糊(例如, 照相机可以获取具有使原始代码模糊的笔迹的图像)。其次, 灰尘、折痕、反射等也可以产生获取的图像上的错误。这些错误使定位过程变得更加困难。在这方面, 图像获取系统需要使用从图像中提取的非顺序比特来运作。以下示出了用于使用来自图像的非顺序比特运作的方法。

首先, 只看序列, 使该序列(或  $m$  序列)  $I$  对应幂级数  $I(x) = 1/P_n(x)$ 。同样,  $R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{n-1}x^{n-1}$  是任何非零多项式, 其次数小于  $n$ 。序列  $R$  对应  $R(x)/P_n(x)$ 。如上所述,  $R$  是  $I$  的移位型式(shifted version)。距离  $D$  定义为  $I$  和  $R$  之间的距离。如果  $R$  是  $I$  的第  $s$  个移位, 则这可以表示为  $D(R) = s$ 。距离  $D$  可以使

用各种不同的技术计算。一项技术是通过确定两者之间的汉明间距。另一个方法是通过使用由 Douglas W. Clark 和 Lih-Jyh Weng 的文章“Maximal and Near-Maximal Shift Register Sequences: Efficient Event Counters and Easy Discrete Logarithms,”IEEE 计算机学报 43.5(1994 年 5 月, pp560-568)中介绍的 Pohlig-Hellman-Silver 算法。还可以使用本技术领域已知的其它的距离确定技术。使用 Pohlig-Hellman-Silver 算法时,  $R(x) \equiv x^s \pmod{P_n(x)}$ 。

但是, 尝试使用折叠的序列计算距离是更加复杂的, 因为所提取的代码可以是非连续的并还可能包括错误。用以上的阵列分析开始, 可以从图像确定位置。可以用以下的关系开始:

$$R = r^t A \quad (2)$$

其中,  $r = (r_0 \ r_1 \ r_2 \ \dots \ r_{n-1})^t$ , 而  $A = (I \ T(I) \ \dots \ T^{n-1}(I))^t$  由 I 的循环移位构成。因此, 任何 I 的循环移位都是  $r^t$  和 A 的乘积。如果笔读取 k 个比特(I 的  $b = (b_0 \ b_1 \ b_2 \ \dots \ b_{k-1})^t$  其中  $(k \geq n)$ ), 则 R(x) 可以用于将  $b_0$  循环移位到序列(或 m 序列)的起始部分。随后, 该移位调整 A 的列来产生 A 的  $n \times k$  的子矩阵 M。因此,

$$r^t M = b^t \quad (3)$$

其中 T 表示矩阵或矢量的变换。

如果 M 的所有行都是独立的, 则可以确定 r 而 D(R) 通过调整方程式(1)的一维位置造成 I 中  $b_0$  的绝对位置。这意味着可以确定  $b_0$  比特的二维位置。该位置信息允许确定笔尖的位置。

接着, 可以确定和检查所有 A 的调整。

可供选择的, 因为矩阵 A 很大, 所以不是所有 A 的调整都需要保留。事实上, 在某些执行过程中, 只有列之间的相对位置是相关的。因此, 可以保存的列(在将  $b_0$  移位到序列(或 m 序列)的起始部分后)是那些当笔(也就是, 照相机)旋转时其对应的二维位置在照相机的接收域内的列。这些列可以存储于具有二维位置信息的阵列(例如, 矩阵)中。一个实例包括:

“元件-类型 CoreVector(核心矢量)[行大小][列大小]”

这种列的总数远小于 L(其中  $L = 2^n - 1$ )。如果 b 是无错误的, 则 r 的解可以表示为:

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

其中  $\tilde{M}$  是 M 的非简并(non-degenerate)的  $n \times n$  子矩阵而  $\tilde{b}$  是相应的 b 的子矢量。

## 错误校正

如果在  $b$  中存在错误, 则  $r$  的解将变得更加复杂。如果  $b$  中的错误数(表示为  $n_e$ ) 小于  $d/2$ , 其中  $d$  是由  $M$  所产生代码的最小汉明距离, 则  $b$  可以被校正。不幸的是, 由于矩阵  $M$  对应不同的笔姿变化, 且矩阵  $M$  的结构是不规则的, 则不便应用具有错误校正的常规译码方法。在某些情况下, 当出现错误时, 认为不能精确地确定  $d$ 。参见 Ilya Dumer, Daniele Micciancio 和 Madhu Sudan, "Hardness of Approximating the Minimal Distance of a Linear Code", IEEE 计算机科学基础的论文集 1999, pp. 475-485。

但是, 使用随机方法实现了积极的结果。假定  $n_e$  相比  $k$  相对较小, 则当随机选择了  $n$  个比特而它在  $M$  中所对应的列是独立的时, 则从  $b$  的  $k$  比特选择正确的  $n$  个比特的概率很大。

系统可以在图像中重复选择随机比特。如果系统重复该选择和分析很多次, 则它一般能分离出序列中的合适距离。换句话说, 当选择了正确的比特时,  $b'$  和  $r'M$  之间的汉明距离将最小, 其中  $r$  是通过方程式(4)计算的。重复几次, 正确的  $r$  将比其它的距离出现更多次, 并因此很容易识别。其结果可以单独使用或可以结合之前的位置信息使用。例如, 如果照相机获取可以位于该页顶部的线上的一串图像, 且从照相机的当前图像提供, 例如, 从之前的串有一个很近、一个很远的两个距离, 则更加可能的是, 在很短的时间内用户只将笔移动较小的量而不是移动较大的量。可以使用之前的图像获取和当前的图像获取之间的其它类型的分析。

如果存在不同的  $r$  以使  $b'$  和  $r'M$  之间的汉明距离都最小, 则  $n_e$  超过由  $M$  所产生代码的错误校正能力的概率很大, 且译码过程失败。随后, 系统可以移到来自照相机的下一个所获取的图像。

最后, 可以确定正确的  $b$ , 而在序列( $m$  序列)  $I$  中的  $r$  的位置可以通过各种技术(包括, 但不限于 Pohlig-Hellman-Silver 算法)来计算。随后可以定位笔尖。

图 8 示出了可以用于在所获取的图像的序列(或  $m$  序列)中确定位置的过程。首先, 在步骤 801 中, 接收到和所获取的图像相关的数据流。在步骤 802 中, 从所获取的图像提取列和由所提取的列构建矩阵。在步骤 803 中, 根据距离将代码排序。可以使用各种本技术领域内已知的排序算法来将代码排序。例如, 可以使用选择排序算法。选择排序算法是有利的, 其中  $m$  不是很大。但是, 如果  $m$  变得很大, 则可以使用其它更有效地处理更多项的排序算法(例如, 合并排序)。

在步骤 804 中, 从步骤 802 中构建的矩阵中选择  $M$  行矢量, 而从 802 的矩阵

到阵列(或  $m$  阵列)的起始部分来确定距离矢量  $r$ 。

所获取的图像可以包括比确定矩阵  $M$  的位置所需的更多的信息。例如, 如果每个比特由 9 个像素表示而照相机图像系统包括  $32 \times 32$  比特的矩阵, 则可以得到约 100 个比特 ( $10 \times 10$ )。这些图像比特中,  $n$  被用于确定位置。所以, 例如, 如果阵列  $m_1, m_2$  的大小是  $255 \times 255$  (65025), 则将所获取的图像定位所需的比特数将为  $n$ , 其中  $m_1 \times m_2 \leq L = 2^n - 1$ 。这里,  $n = 16$ 。因此, 所获取的图像所需要用来产生能够定位图像的比特数为  $n$ 。为了提供各种不同的选择来考虑错误, 需要从图像中提取超过  $n$  个比特。例如, 如果所需的  $n$  的数量为 16, 则照相机将需要产生 18 或者更多的比特。

该分析在步骤 805 中进行  $K$  次(例如, 10 到 100 到 500 次)。该方法尝试确定具有最多重复结果的位置, 且其错误比特的数量小于一个值(例如, 3)。随后, 系统确定在步骤 806 中错误校正是否进行得顺利。如果否, 则将错误返回到步骤 809。如果是, 则在步骤 807 的序列(或  $m$  序列)中计算提取的代码的位置(例如, 通过 Pohlig-Hellman-Silver 算法)。该步骤还可以包括计算  $y_i = y_n / m_i$  并在组列表中查找系数(factor)以便得到  $R_i$  的剩余值的子步骤。这可以表示为:

$$L = \sum \left( \frac{n}{m_i} \cdot R_i \cdot V_i \right) \bmod n \quad (5)$$

$L$  表示序列的长度。接着, 阵列中的  $(X, Y)$  位置表示为  $x = L \bmod n_1$  和  $y = L \bmod n_2$  且结果返回到步骤 808。

#### 位置确定

图 9 示出用于确定笔尖位置的过程。输入是由照相机获取的图像, 而输出可以是该图像相比总图像的位置坐标。而且, 输出可以包括(或不包括)所获取的图像的旋转角度。

在步骤 901 中, 从照相机接收图像。接着, 所接收的图像可以选择性地在步骤 902 中预处理(如步骤 902 中由虚线标出的)来调整亮和暗像素之间的对比度, 旨在确定所获取的图像的旋转角度等等。

接着, 在步骤 903 中, 分析图像来确定其中的比特流。

接着, 在步骤 904 中, 比特流和初始比特流比较并确定在初始比特流中所接收的比特流的位置。当系统意识到初始比特流如何卷绕来创建图像 403 时, 可以进行该比较。最后, 一旦在步骤 904 中确定了所获取的图像的位置, 则在步骤 905

中可以确定笔尖的位置。

图 10 示出用来确定笔尖位置的可供选择的方法。首先，在步骤 1001 中从照相机接收图像。随后，该图像可以选择性地进行步骤 1002 中的预处理(如步骤 1002 中的虚线所示)。在步骤 1003 中提取图形。这里，可以提取各种线上的像素来找到图形的方向并确定角度  $\theta$ 。

接着，在步骤 1004 中分析所接收的图像来确定像素和网格线基础分布。如果在步骤 1005 中找到网格线，则在步骤 1006 中从图形提取代码。随后，在步骤 1007 中将该代码译码并在步骤 1008 中确定笔尖的位置。如果在步骤 1005 中没有找到网格线，则在步骤 1009 返回错误信号。

虽然用所附权利要求定义了本发明，但这些权利要求是说明性的，其中，本发明旨在包括此处说明的组合或子组合中的单元和步骤。因此，存在任何数量的可供选择的组合来定义本发明，它们结合一个或多个说明书中的单元，包括描述、权利要求和附图，在各种组合或子组合中。对于相关技术领域中的熟练的技术人员来说明显的是，根据本说明书，本发明的各方面的可供选择的组合，无论是单独或者结合一个或多个此处定义的单元或步骤，可以用作本发明或本发明的一部分的修改或变化。包含在这里的本发明的书面描述旨在覆盖所有这样的修改和变化。



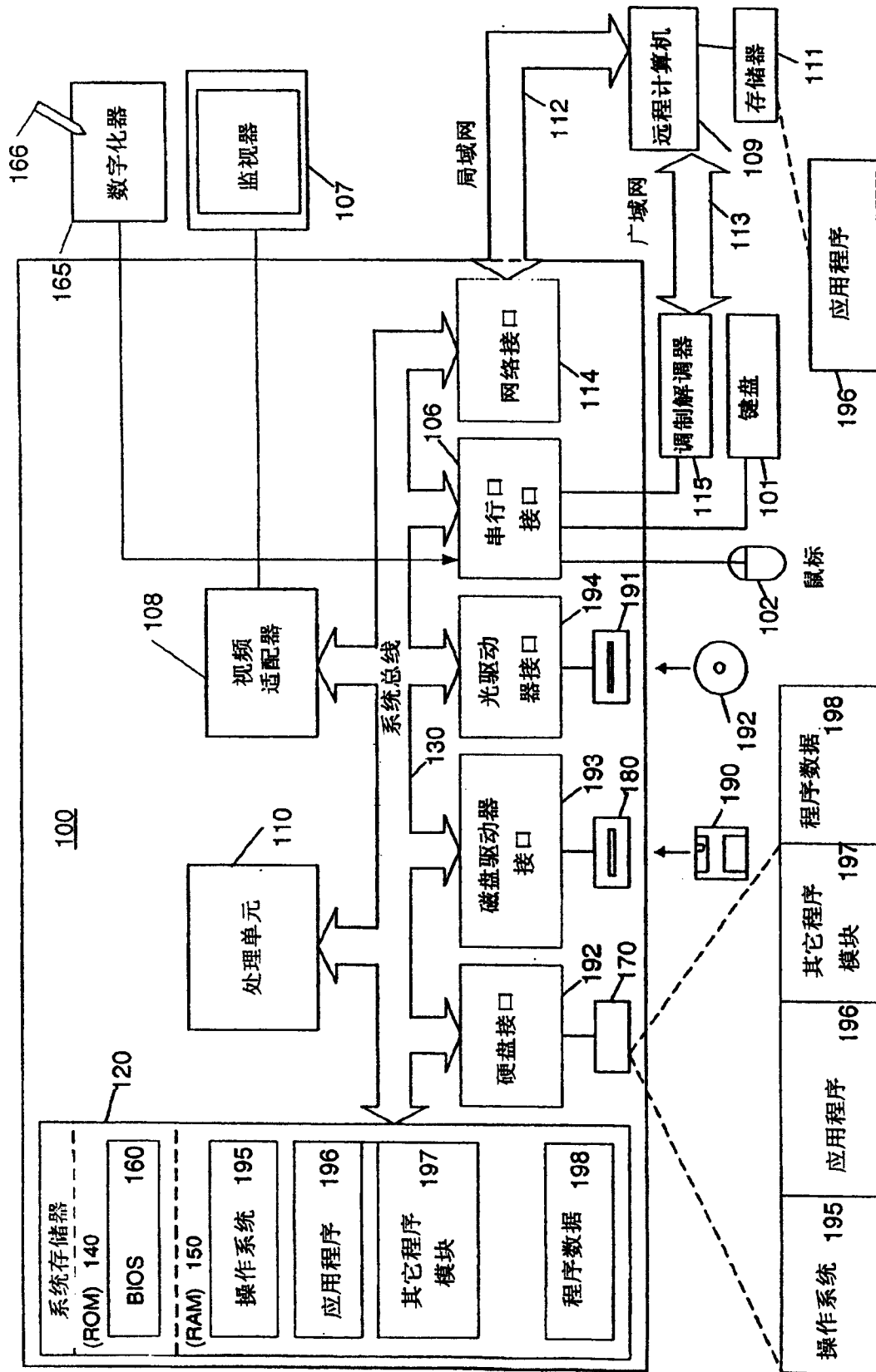


图 1

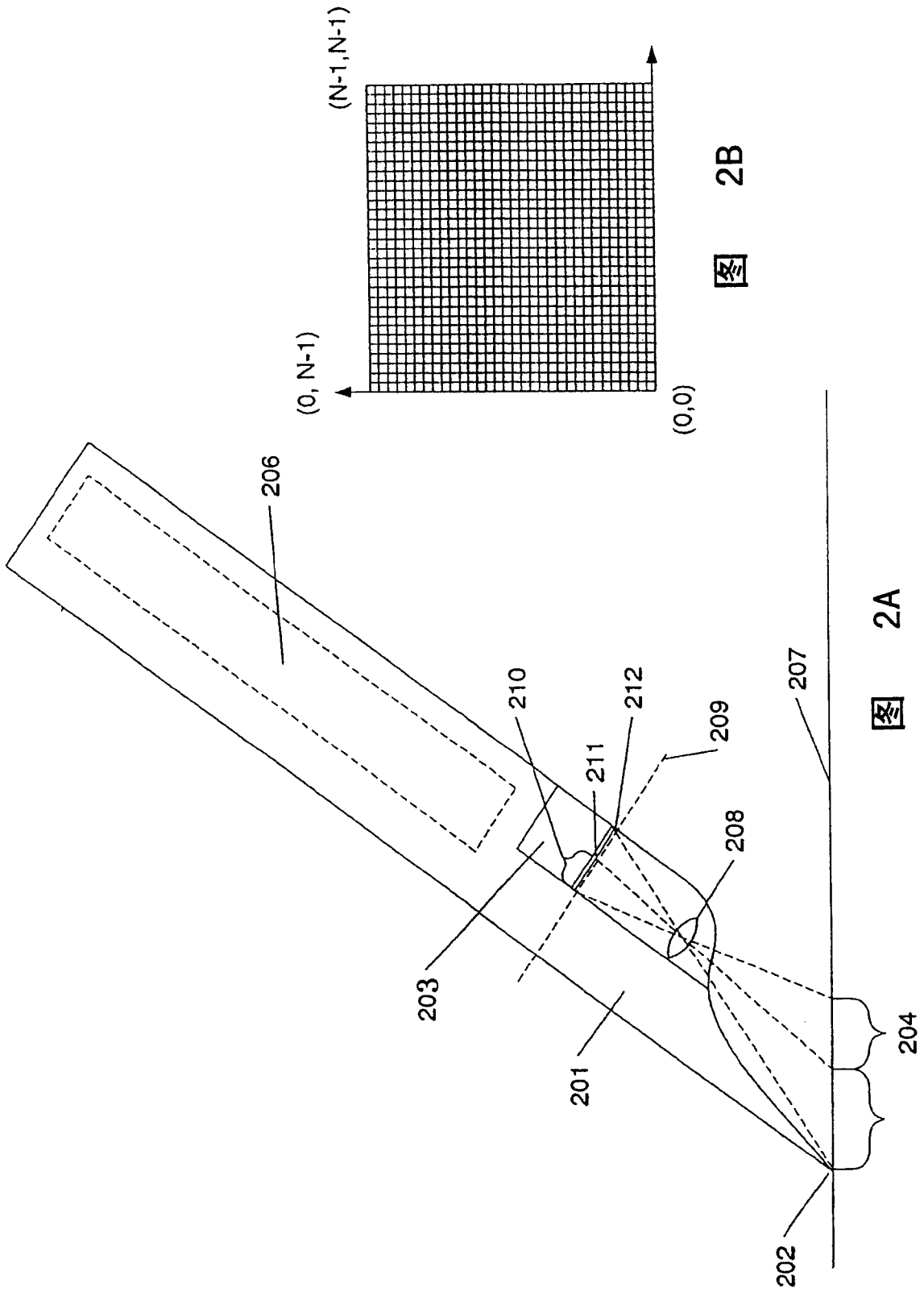


图 2B

图 2A

0111 0000 0101 1101 1001 0111 1111 1000 1010

图 3A

010101110 110110010 001010011 111101100

图 3B

0101 1101 0011 1101 1001 1001 1110 0010

图 3C

$m_1 = 9$

0	1	0	1	0	1	1	1	1	0
1	1	0	0	1	1	0	0	1	0
0	0	1	0	0	1	0	0	1	1
1	1	0	1	1	0	1	1	0	0

$m_2 = 4$

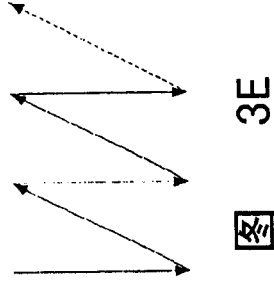


图 3E

图 3D

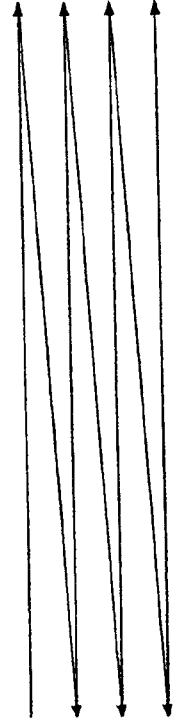
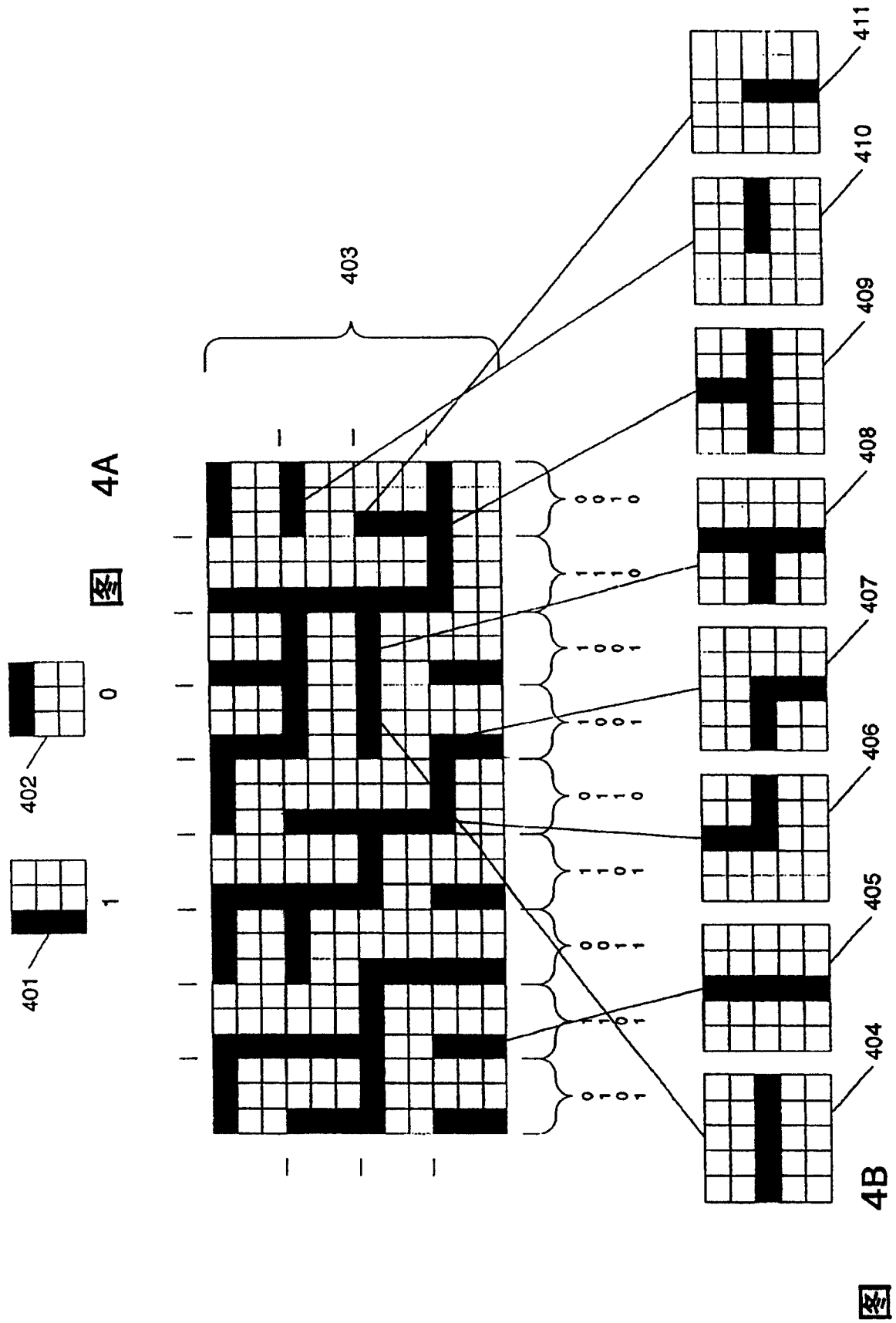


图 3F



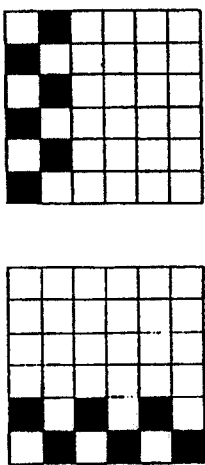


图 4C

0

1

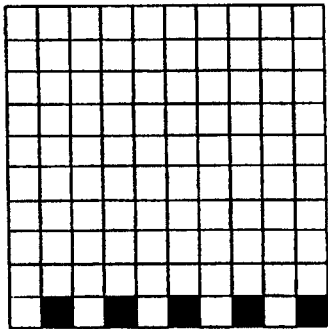


图 4D

0

1

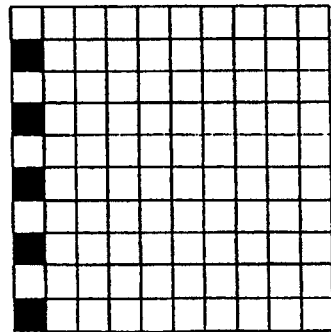


图 4E

0

1



5A

图



5B

图



5C

图



5D

图

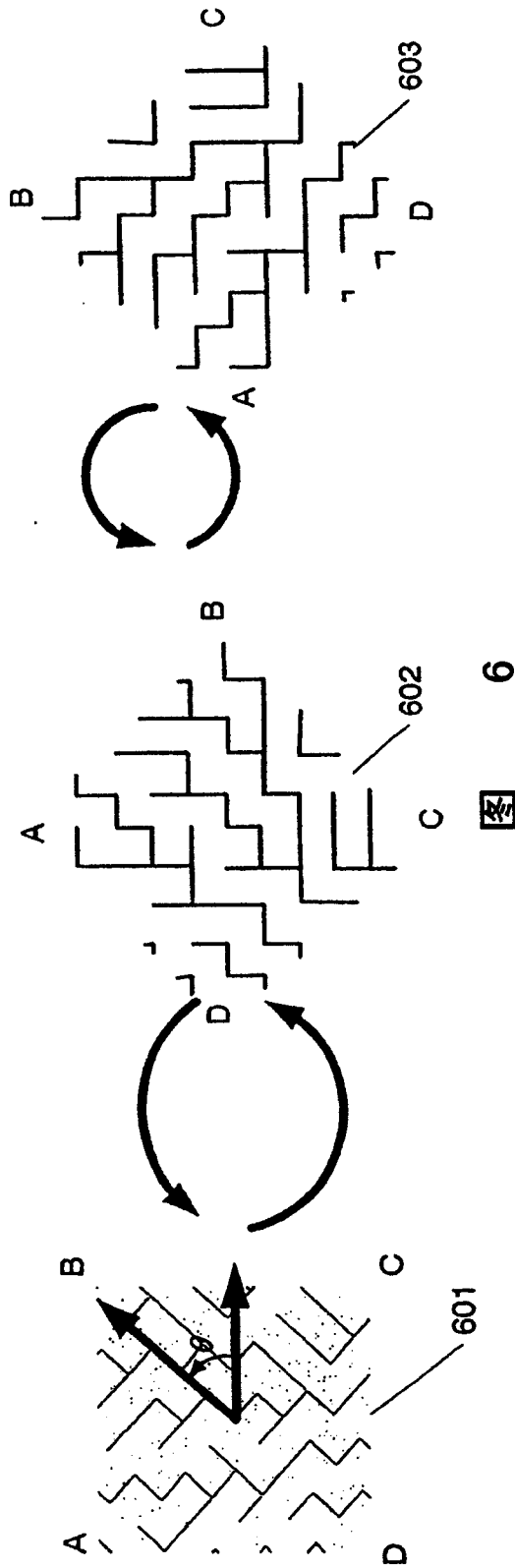


图 6

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

图 7

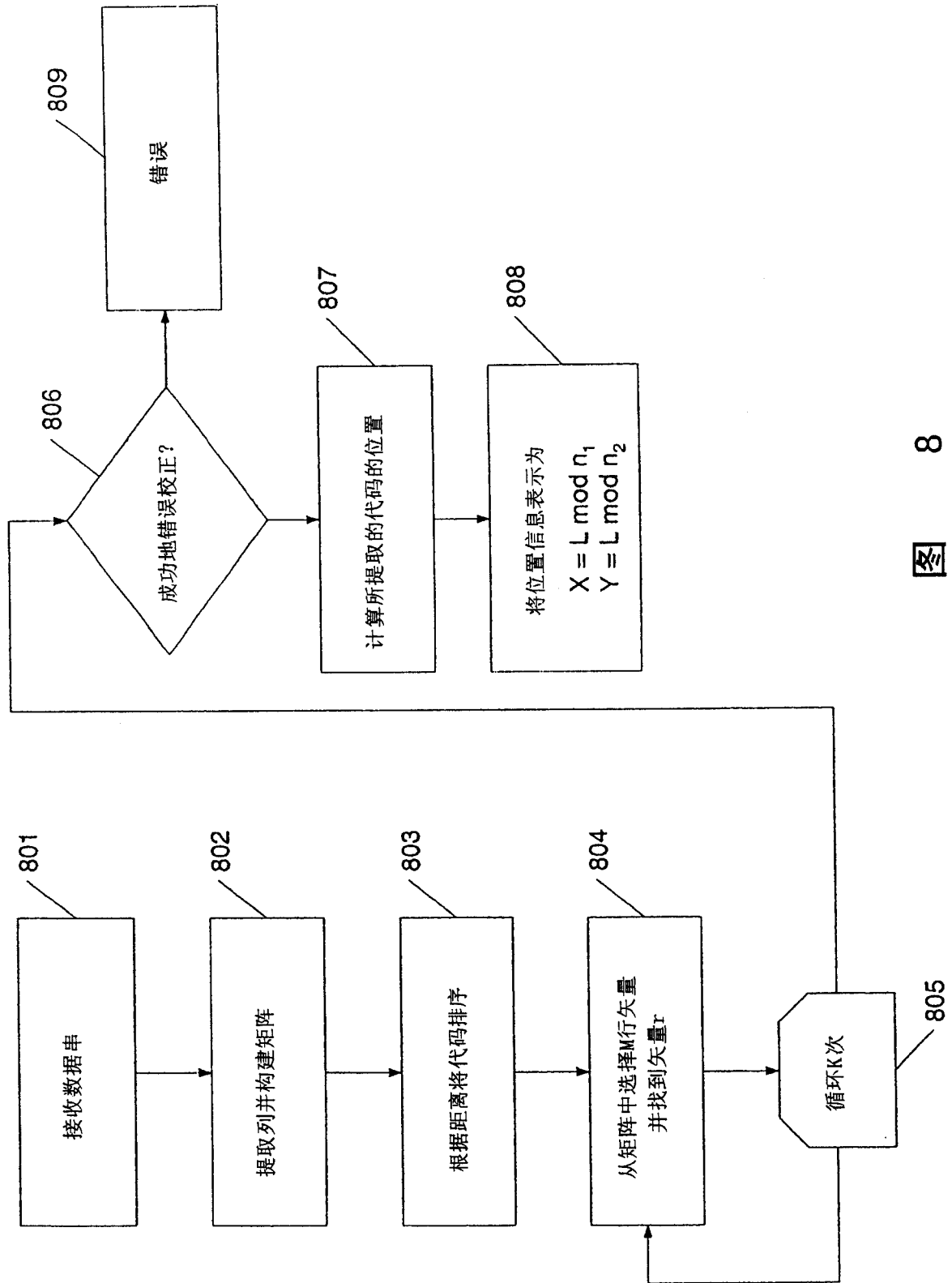


图 8

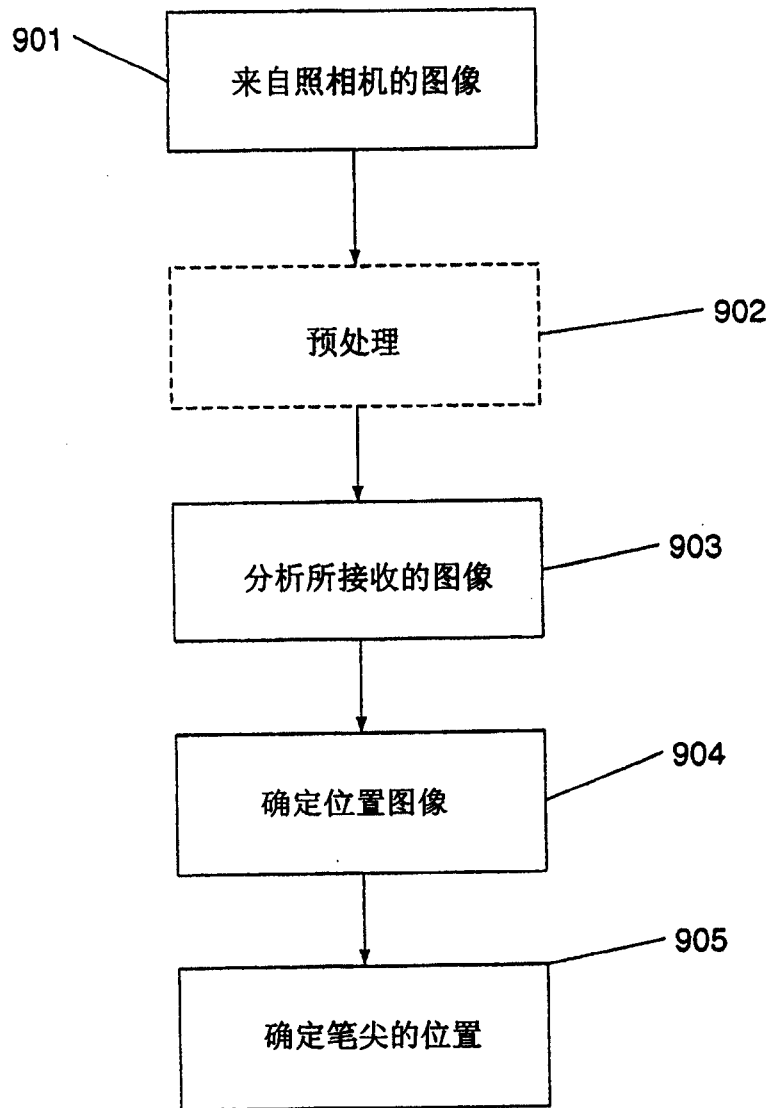


图 9



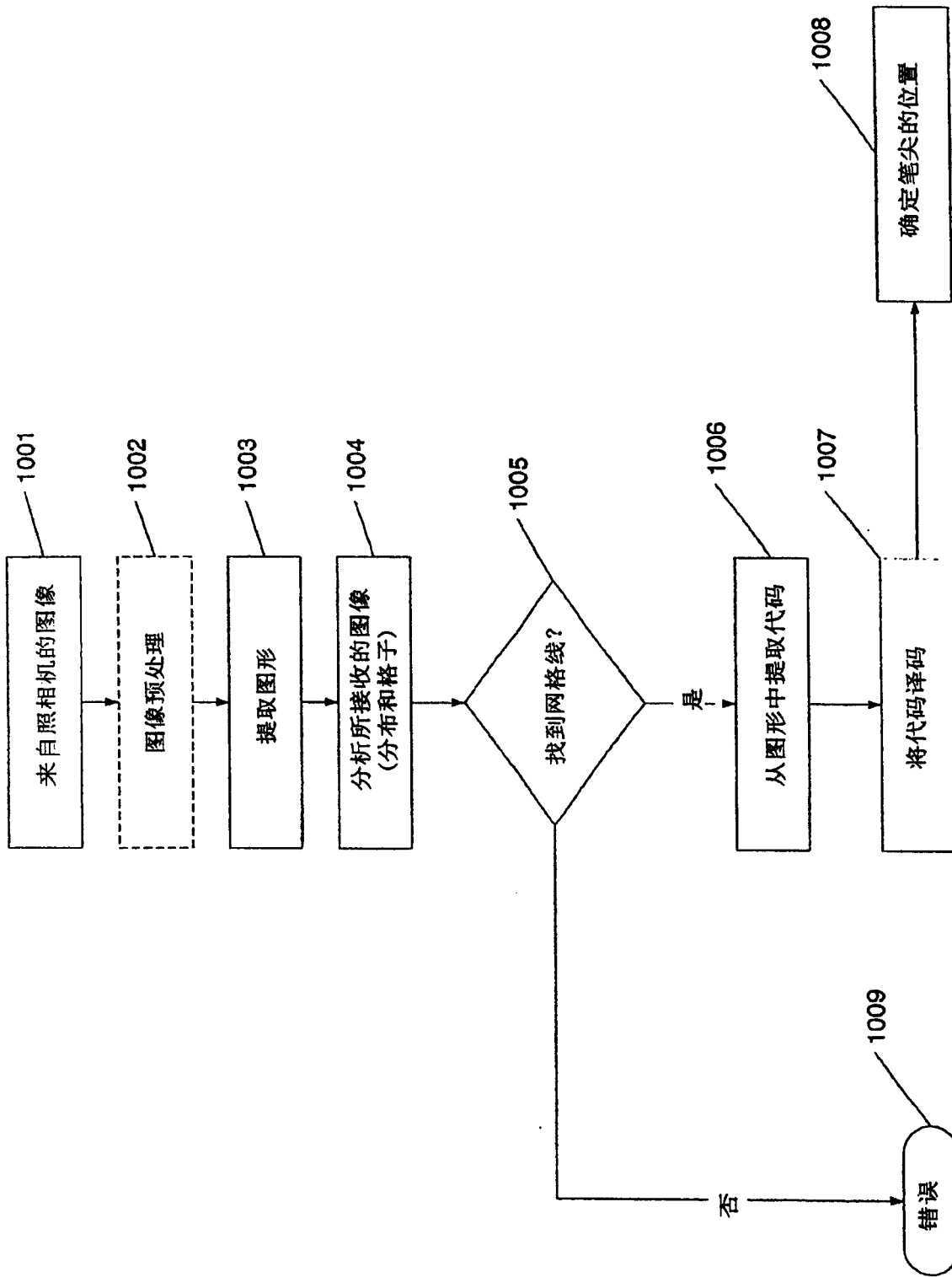


图 10

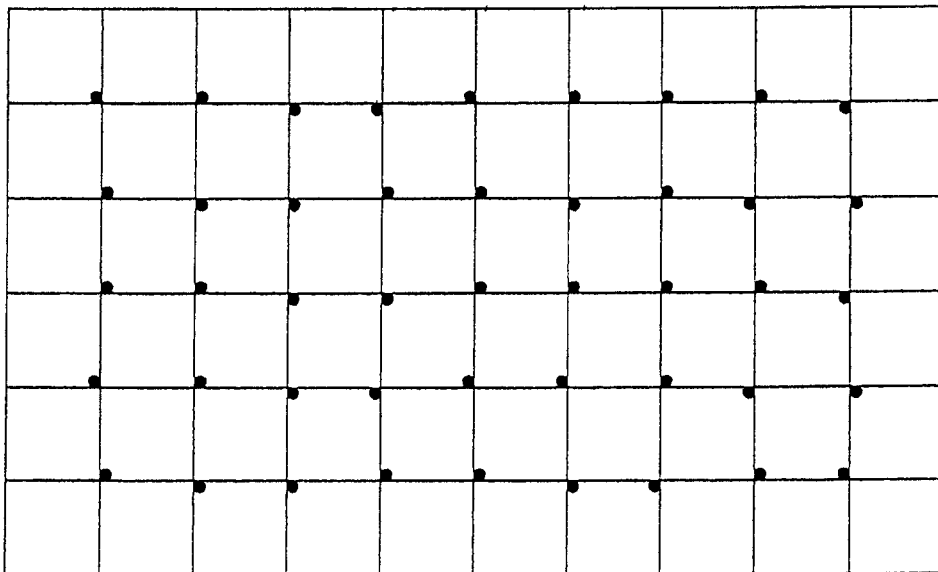


图 11