

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 3/033 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510129610.2

[43] 公开日 2006 年 6 月 7 日

[11] 公开号 CN 1782973A

[22] 申请日 2002.6.25

[21] 申请号 200510129610.2

分案原申请号 02812703.X

[30] 优先权

[32] 2001.6.25 [33] SE [31] 0102235-9

[71] 申请人 阿诺托知识产权许可贸易公司

地址 瑞典隆德

[72] 发明人 马茨·彼得·彼得森

安德烈亚斯·比约克隆

[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 王 萍

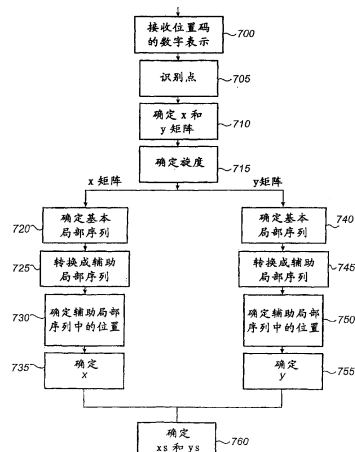
权利要求书 5 页 说明书 30 页 附图 5 页

[54] 发明名称

用于获取位置码和解码位置码的方法

[57] 摘要

本发明涉及在一种用于进行位置编码的方法中，在一个表面上，位置是根据基本数字序列而在第一维度编码的，其中所述基本数字序列具有如下特性，即第一预定长度的各个局部序列的基本数字序列是无歧义确定的。第一维度中的各个位置是由局部序列中的一个进行编码的。基本数字序列是由至少两个辅助数字序列构造的，所述序列具有比基本数字序列基数更小的基数并且被用于确定与第一维度中的位置相对应的基本数字序列的局部序列。由于辅助数字序列总体需要的存储器要比基本数字序列更少，因此尤其有可能实现具有有限存储器容量的设备中的方法。辅助数字序列也可用于解码所述位置码。



1. 一种用于位置编码的方法，包括如下步骤：

在一个表面上，根据一个基本数字序列而对第一维度中的位置进行编码，所述序列具有如下特性，即第一预定长度的各个局部序列的基本数字序列中的位置都是无歧义确定的，第一维度中的各个位置都是使用局部序列中的一个来编码的，

基本数字序列是由至少两个辅助数字序列构造的，所述辅助数字序列的基数少于所述基本数字序列的基数并且用于确定与第一维度中的位置相对应的基本数字序列的局部序列。

2. 根据权利要求1的方法，还包括：

使得可用于位置编码的基本数字序列中的不同数字的数目可以分解成至少两个因数，

使用与因数数目具有相同数目的辅助数字序列，以及

使得每个因数形成相应辅助数字序列中的基数。

3. 根据权利要求1或2的方法，还包括使得辅助数字序列的长度成对互质。

4. 根据前述任何一个权利要求的方法，其中每个辅助数字序列都具有如下特性，即每一个具有第一预定长度的局部序列在相应的辅助数字序列中都具有一个无歧义确定的位置。

5. 根据前述任何一个权利要求的方法，其中辅助数字序列的基数至多是5，优选至多是3。

6. 根据前述任何一个权利要求的方法，其中基本数字序列的局部序列可以双射映射在来自各个辅助数字序列的局部序列组合上。

7. 根据前述任何一个权利要求的方法，其中基本数字序列中的数字是使用周期性主数字序列的第二预定长度的局部序列来编码的，所述序列具有如下特性，即每一个第二预定长度的局部序列的周期性主数字序列中的位置是无歧义确定的，周期性主数字序列的局部序列是以这样一种方式而在所述表面上设置的，其中定义了形成基本数字

序列中的数字的差分数字。

8. 根据权利要求 7 的方法，其中周期性主数字序列的局部序列是以这样一种方式而在所述表面上设置的，其中每一个局部序列是在一个第二维度上延伸的。

9. 根据权利要求 7 或 8 的方法，其中仅仅使用了一个在理论上可以使用主数字序列的局部序列来进行定义的差分数字的子集，所述子集排除了这种等于或接近于零的理论上可以定义的差分数字。

10. 根据权利要求 7-9 中任何一个权利要求的方法，其中主数字序列是二进制的，并且所述序列是以这样一种方式选择的，其中在主数字序列中并没有以后向或反转形式出现预定长度大于第二预定长度的局部序列。

11. 根据权利要求 7-10 中任何一个权利要求的方法，其中主数字序列是二进制的，并且所述序列是以这样一种方式选择的，其中在主数字序列中，并没有以一个二进制数字反转的形式出现预定长度大于第二预定长度的局部序列。

12. 根据权利要求 7-11 中任何一个权利要求的方法，其中主数字序列中具有第二预定长度的局部序列包括六个数字，并且基本数字序列中具有第一预定长度的局部序列包括五个差分数字。

13. 根据权利要求 7-12 中任何一个权利要求的方法，其中周期性主数字序列 M 如下所示：

M=0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0,
1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,
0, 1, 1.

14. 根据前述任何一个权利要求的方法，其中基本数字序列包括 54 个不同数字，辅助数字序列则具有 4 个数字，其中三个具有基数三，并且其中一个具有基数二。

15. 根据前述任何一个权利要求的方法，其中辅助数字序列如下所示：

**A1=0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0, 1,
0, 0, 2, 0, 0, 0, 1, 1, 0, 0, 0, 1, 2, 0, 0, 1, 0, 2, 0, 0, 2,
0, 2, 0, 1, 1, 0, 1, 0, 1, 1, 0, 2, 0, 1, 2, 0, 1, 0, 1, 2, 0,
2, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 2, 1, 0, 1, 0, 2, 1, 1,
0, 0, 1, 2, 1, 0, 1, 1, 2, 0, 0, 0, 2, 1, 0, 2, 0, 2, 1, 1, 1,
0, 0, 2, 1, 2, 0, 1, 1, 1, 2, 0, 2, 0, 0, 1, 1, 2, 1, 0, 0, 0,
2, 2, 0, 1, 0, 2, 2, 0, 0, 1, 2, 2, 0, 2, 0, 2, 2, 1, 0, 1, 2,
1, 2, 1, 0, 2, 1, 2, 1, 1, 0, 2, 2, 1, 2, 1, 2, 0, 2, 2, 0, 2,
2, 2, 0, 1, 1, 2, 2, 1, 1, 0, 1, 2, 2, 2, 2, 1, 2, 0, 0, 2, 2,
1, 1, 2, 1, 2, 2, 1, 0, 2, 2, 2, 2, 2, 0, 2, 1, 2, 2, 2, 1, 1,
1, 2, 1, 1, 2, 0, 1, 2, 2, 1, 2, 2, 0, 1, 2, 1, 1, 1, 1, 2, 2,
2, 0, 0, 2, 1, 1, 2, 2**

**A2=0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0, 1,
0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 2, 0, 0,
0, 1, 2, 0, 1, 0, 1, 2, 1, 0, 0, 0, 2, 1, 1, 1, 0, 1, 1, 1, 0,
2, 1, 0, 0, 1, 2, 1, 2, 1, 0, 1, 0, 2, 0, 1, 1, 0, 2, 0, 0, 1,
0, 2, 1, 2, 0, 0, 0, 2, 2, 0, 0, 1, 1, 2, 0, 2, 0, 0, 2, 0, 2,
0, 1, 2, 0, 0, 2, 2, 1, 1, 0, 0, 2, 1, 0, 1, 1, 2, 1, 0, 2, 0,
2, 2, 1, 0, 0, 2, 2, 2, 1, 0, 1, 2, 2, 0, 0, 2, 1, 2, 2, 1, 1,
1, 1, 1, 2, 0, 0, 1, 2, 2, 1, 2, 0, 1, 1, 1, 2, 1, 1, 2, 0, 1,
2, 1, 1, 1, 2, 2, 0, 2, 2, 0, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2,
0, 1, 2, 2, 2, 0, 2, 0, 2, 1, 1, 2, 2, 1, 0, 2, 2, 0, 2, 1, 0,
2, 1, 1, 0, 2, 2, 2, 2, 0, 1, 0, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1,
2, 0, 2, 2, 2,**

**A3=0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0,
0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,**

**A4=0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2, 0, 0, 2, 0, 1,
0, 0, 0, 1, 1, 2, 0, 0, 0, 1, 2, 0, 0, 2, 1, 0, 0, 0, 2, 1, 1,
2, 0, 1, 0, 1, 0, 0, 1, 2, 1, 0, 0, 1, 0, 0, 2, 2, 0, 0, 0, 2,**

2, 1, 0, 2, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 2, 0,
1, 1, 1, 1, 0, 0, 2, 0, 2, 0, 1, 2, 0, 2, 2, 0, 1, 0, 2, 1, 0,
1, 2, 1, 1, 0, 1, 1, 1, 2, 2, 0, 0, 1, 0, 1, 2, 2, 2, 0, 0, 2,
2, 2, 0, 1, 2, 1, 2, 0, 2, 0, 0, 1, 2, 2, 0, 1, 1, 2, 1, 0, 2,
1, 1, 0, 2, 0, 2, 1, 2, 0, 0, 1, 1, 0, 2, 1, 2, 1, 0, 1, 0, 2,
2, 0, 2, 1, 0, 2, 2, 1, 1, 1, 2, 0, 2, 1, 1, 1, 0, 2, 2, 2, 2,
0, 2, 0, 2, 2, 1, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 1, 0, 0,
2, 1, 2, 2, 1, 0, 1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 0,
1, 2, 2, 1, 2, 2, 0, 2, 2, 2, 1, 1.

16.根据前述任何一个权利要求的方法,还包括在所述表面上根据第二基本数字序列对第二维度中的位置进行编码,其中所述第二基本数字序列具有如下特性,即无歧义确定具有第三预定长度的每一个局部序列的第二基本数字序列中的位置,第二维度中的各个位置是由第二基本局部序列中的一个局部序列来进行编码的,并且第二基本数字序列是由至少两个第二辅助数字序列构造的,所述第二辅助数字序列具有小于第二基本数字序列的基数并被用于确定与第二维度中的位置相对应的第二基本数字序列的局部序列。

17. 根据权利要求 16 的方法,其中用于在第一和第二维度中进行位置编码的基本和辅助数字序列都是相同的。

18. 根据前述任何一个权利要求的方法,还包括将结合了图形编码的位置码打印在所述表面上。

19. 一种计算机程序,其中包括用于使计算机执行位置编码的指令,所述计算机程序包括用于在一个表面上根据基本数字序列来对第一维度中的位置进行编码的指令,所述基本数字序列具有如下特性,即无歧义确定具有第一预定长度的各个局部序列的基本数字序列中的位置,第一维度中的各个位置是由局部序列中的一个来进行编码的;

基本数字序列是由至少两个辅助数字序列构造的,所述辅助数字序列的基数小于基本数字序列的基数,并且该序列由计算机程序使

用，以便确定与第一维度中的位置相对应的基本数字序列的局部序列。

20. 一种用于编码位置的设备，所述设备包括一个处理器和一个存储器，所述存储器保存的是根据权利要求 19 的计算机程序。

21. 一种计算机程序，包括用于使计算机执行根据权利要求 21 - 32 中任何一个权利要求的方法的指令。

22. 一种计算机程序，包括用于使计算机执行根据权利要求 35 - 42 中任何一个权利要求的方法的指令。

23. 一种用于解码一个位置码的设备，所述设备包括一个处理器和一个存储器，所述存储器保存一个根据权利要求 22 而对位置码进行解码的计算机程序。

24. 一种产品，具有一个位置码，所述位置码至少对一个第一点和一个第一维度的第一坐标进行编码，第一点的第一坐标是由基本数字序列的局部序列来编码的，所述局部序列具有第一预定长度，所述基本数字序列具有如下特性，即预定长度的各个局部序列的基本数字序列中的位置都是无歧义确定的，并且所述基本数字序列还是通过至少两个辅助数字序列构造的，所述辅助数字序列具有比基本数字序列基数更小的基数。

用于获取位置码和解码位置码的方法

本申请是申请号为 02812703.X、申请日为 2002 年 6 月 25 日、发明名称为“用于获取位置码和解码位置码的方法”的专利申请的分案申请。

技术领域

本发明涉及用于获取位置码的方法，一种对位置码进行解码的方法，一种用于这些目的的计算机程序，一种具有位置码的产品，以及用于位置编码的设备以及对位置码进行解码的设备。

背景技术

在一个表面上，将位置码用作一种以电子形式记录那些写入或绘制在所述表面上的信息的工具，这种技术已经是已知的。

US5,477,012 中描述了一种笔，所述笔在一个表面上以光学方式检测一个包含不同符号的位置码。所述笔对位置码进行解码并且生成那些对所述笔穿越表面的移动进行描述的坐标对。每一个位置都是由一个特定符号来编码的。如果位置码对许多位置进行编码，则会导致出现符号变得复杂并且由此难以对其进行正确检测的缺点。另一方面，如果使符号变得更大，则分辨率也会降低。

WO92/17859 中描述了一种位置码，其中每一个位置都是借助多个符号来进行编码的，并且每一个符号都有助于一个以上位置的编码。因此，相邻位置是通过局部共享的符号来进行编码的。

WO92/17859 的附录 A 中给出了以下关于如何构造位置码和如何对位置进行解码的实例。

从以下的 m-序列开始： $s = (0, 0, 1, 0, 1, 1, 1)$ 并且 $t = (0, 1, 1)$ 。通过让图案 (pattern) 中的第一列与序列 s 相同，从

而构造一个位置编码图案。为了构造以下的列，可以查看 t-序列。如果 t-序列中的第一个元素是 0，则第二列包括 s-序列。取而代之的是，如果第一个元素是 1，则第二列包括具有单步循环移位的 s-序列。并且后续的列也是根据 t-序列中元素的值而以相应方式构建的。由此得到了以下图案：

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

现在假设我们想要找到具有以下图案子集的局部表面的位置。

1	0	0
0	1	0
1	0	1

在这个子集中，第一列是 (1, 0, 1)。这个子序列出现在 s-序列中的位置 2。子集的循环移位是 (1, 1)。这个子序列出现在 t-序列中的位置 1。图案中的累积移位是 (0, 0, 1, 2)，因此，子集的垂直位置是 $2 + 0 = 2$ 。这样，在局部表面上，子集的位置是 (1, 2)。

这个图案具有如下优点，即可以使用简单符号对其进行编码，例如用于 0 的第一符号和用于 1 的第二符号。

然而，这类位置编码图案的一个有趣特性是能够使用众多唯一位置来对一个很大图案进行编码，以便在一个尽可能大的表面上执行位置确定。在以上实例中，垂直方向的大小受到 s-序列长度的限制，水平方向的大小则受到 t-序列长度的限制。这些序列长度不能无限增加，因为所述序列必须具有如下特性，即如果得到一个 k 位子序列，那么这个子序列必定出现在序列中的一个单独位置。因此，增加

序列长度可能造成子序列长度增加，由此造成了为确定位置而需要记录的局部表面的增大。

在转让给本申请的申请人的 WO01/26032、WO01/26033 和 WO01/26034 中描述了另一种位置编码图案，所述位置编码图案与 WO92/17859 中的图案基于相同的基本原则，但是所述位置编码图案能对数目更多的位置进行编码。

在这种图案的一个实施例中，使用了第一数字序列，该序列具有如下特性，即无歧义地确定每一个具有预定长度的子序列或局部序列在数字序列中的位置，以及在所述表面上的列中打印具有不同旋度或循环移位的数字序列。然而，与使用一个二进制的第二序列（WO92/17859 中的 t-序列）所不同，在这里允许在相邻列间执行更大的循环移位。

所述表面进一步分成编码窗口（code window）。每个编码窗口包括至少三个具有数字序列的列并与相邻编码窗口重叠了一个数字序列。通过在相邻列之间使用移位，对沿着 x 轴的编码窗口位置进行编码。这样一来，在一个编码窗口中的数字序列之间所进行的移位定义了一个位置，其中第一编码窗口具有位置 0，下一个则具有位置 1，依此类推。

然而，在读出位置码的时候，读取设备常常会记录这两个编码窗口的某些部分。为了确定涉及位置码的读出图像的编码窗口位置，在这里使用第一范围中的移位来指示用于该位置的最不重要移位，并且将第二范围中的移位用于所述位置中的其他移位。

而 y 坐标的编码也是以相应方式执行的。

目前已经发现，这个位置码可以很好地工作并且允许对很多位置进行编码。然而还存在一个提高位置码纠错特性的期望。

例如，假设对 x 方向的位置进行无歧义编码的最小子集包含五个移位。从这五个移位中可以确定编码窗口的位置。因此，两个相邻编码窗口的位置通常只会在最后一个数字（最后一个移位）上存在差异。例如，对三个连续编码窗口来说，可以如下进行移位：

25, 15, 34, 18, 3; 25, 15, 34, 18, 4; 25, 15, 34, 18, 5;

普通的检错方法基于：读出比所涉及任务需要的信息还要多的信息。而结合上述编码则很难读出适用于检错的额外信息。例如，假设读取设备记录了以上三个编码窗口中第一个编码窗口的最后两个移位以及后续窗口的前三个移位，即 18, 3, 25, 15, 34。此外，假设出现了一个意味着记录了 18, 4, 25, 15, 34 的错误。那么，举例来说，从这五个移位两端一端一个地读出的两个另外的移位并没有提供可以检错的额外信息。

当在 x 方向进行编码的时候，如果使用一个具有如下特性的数字序列，其中预定长度的各个局部序列（子序列）在所述数字序列中都具有无歧义确定的位置，则可以简化纠错。而另一方面则出现了一个实际问题，那就是这个数字序列必须具有很大的基数，也就是说，如果它可以对许多位置进行编码，则必须允许很大的移位。然后则需要一个很大的表，以便在数字序列中找到与读出的局部序列相对应的位置，这转而需要大量的存储器。然而举例来说，较为理想的是能在一个具有有限存储器和处理器容量的手持读取设备中执行解码。并且较为理想的是能在具有有限存储器和处理器容量的设备中执行编码。

发明内容

因此，本发明的一个目的是通过提供一个位置码来减少上述问题，所述位置码可以对大量位置进行解码，允许进行纠错，但不需要为了进行编码和解码而保存大量的表，由此可以在具有有限数量的存储器和/或有限处理器容量的设备中得以实现。

另一个目的则是可以有效正确地对一个位置码进行解码，其中所述位置码对很多位置进行编码。

根据本发明，提供了一种在编码设备中用于确定位置码的方法，所述位置码对一个表面上的第一点以及第一维度的至少一个第一坐标 x 进行编码，包括：接收至少第一坐标 x ，作为发送到编码设备的输入信号，基于接收到的坐标 x ，确定用于编码第一维度中的位置的基本数字序列的一个局部序列，所述局部序列具有第一预定长度并

且在基本数字序列中具有一个与接收到的坐标 x 相对应的无歧义确定的位置，基本数字序列是由至少两个辅助数字序列构造的，所述辅助数字序列的基数小于基本数字序列并由编码设备使用，以便确定与坐标 x 相对应的基本数字序列的局部序列。

根据本发明，提供了一种用于确定位置码的编码设备，所述位置码对一个表面上的第一点以及第一维度的至少一个第一坐标 x 进行编码，所述编码设备包括：处理器，所述处理器被配置用于：接收至少第一坐标 x 作为输入信号并基于接收到的坐标 x 确定用于编码第一维度中的位置的基本数字序列的一个局部序列，所述局部序列具有第一预定长度并且在基本数字序列中具有一个与接收到的坐标 x 相对应的无歧义确定的位置，以及存储器，用于存储具有基数小于基本数字序列的至少两个辅助数字序列，所述处理器被配置用于通过使用所述至少两个辅助数字序列，确定与坐标 x 相对应的基本数字序列的局部序列。

根据本发明，提供了一种用于对位置码进行解码的方法，所述位置码对一个表面上的第一点和第一维度的一个第一坐标($x:y$)进行编码，该方法包括：识别第一预定数量的数字，所述数字基于位置码的数字表示而形成了基本数字序列的局部序列，所述局部序列在基本数字序列中具有一个与第一坐标相对应的无歧义确定位置 P ；以及确定基本数字序列中局部序列的位置 P 包括使用至少两个辅助数字序列，所述序列构造了基本数字序列，并且其中每一个辅助数字序列都具有一个比基本数字序列基数更小的基数。

根据本发明，提供了一种用于对位置码进行解码的设备，所述位置码对一个表面上的第一点和第一维度的第一坐标进行编码，所述设备包括一个用于提供解码的位置码的数字表示的传感器，用于保存至少两个辅助数字序列的存储器，其中所述辅助数字序列使用一个比辅助数字序列基数更大的基数来构造一个基本数字序列，以及一个处理器，所述处理器被调整成基于位置码的数字表示来识别第一预定数量的数字，这些数字形成了基本数字序列的一个局部序列，所述局部序列在基本数字序列中具有一个与第一坐标相对应的无歧义确定的位置 P ，并且所述处理器还被调整成借助辅助数字序列来确定基本数字序列中的位置 P 。

因此,本发明的第一个方面涉及一种用于位置编码的方法,包括如下步骤:在一个表面上,根据一个基本数字序列而对第一维度中的位置进行编码,其中所述序列具有在基本数字序列中无歧义确定第一预定长度的各个局部序列位置的特性,第一维度中的每一个位置是由任何一个局部序列编码的,基本数字序列则由至少两个基数小于基本数字序列基数的辅助数字序列构造,并且使用这两个数字序列用于确定与第一维度中的位置相对应的基本数字序列的局部序列。

由于基本数字序列是由基数更小的辅助数字序列构造的,其中所述基数优选为不超过5,因此,对于将基本数字序列中的位置转换为这个序列的局部数字序列来说,所需要的表可以由若干个较小的表所取代,这些表合起来只需要较少的存储器。由此可能在实践中对数量相当多的位置的坐标进行编码。

并且,通过在至少一个第一维度中使用基本数字序列来对位置进行编码,还有可能得到一个具有良好检错和纠错特性的位置码。更具体地说,基本数字序列中的每个数字都是使用一个周期性主数字序列的局部序列来进行编码的。如果现在记录的是比位置确定所需序列更长的主要数字序列的局部序列,则可以通过使用以这样一种方式设计的主数字序列来实现检错和纠错,其中更长的局部序列不会在某些错误版本中出现,例如具有一个错误数字的版本。

同样,对解码而言,其中使用了这样一个实际情况,那就是基本数字序列是由基数更小的辅助数字序列构造的,所述解码能够使用几个更小的表以及中国余数定理来执行。由此可以在例如具有有限存储器和处理器容量的手持设备中实现解码。

这种位置编码可用于对一个和两个维度中的位置进行编码。并且,可以使用相同的规则以及结合与第一维度编码所用序列不同或者相同的数字序列来执行第二维度的编码。

附图说明

现在将使用一个示范性实施例并且参考附图来描述本发明,其中:

图 1 示意性显示了如何在第一方向执行位置 x 方向的编码,

图 2 示意性显示了如何使用四个辅助数字序列 A1 - A4 来构造一个基本数字序列 PD,

图 3 显示了位置码的图形编码的一个实例,

图 4 显示了一个具有增大的位置码的产品,

图 5 显示了如何记录用于纠错的额外符号,

图 6 是一个流程图并且显示了如何确定位置码的一个实例,

图 7 是一个流程图并且显示了如何解码位置码的一个实例, 以及

图 8 示意性显示了一个可用于读出和解码位置码的设备。

具体实施方式

在下文中将使用一个特定实例来描述根据本发明的用于位置编码的普遍原理。此后则描述如何在实践中执行编码和解码。之后将会描述可用于对位置码进行解码的读取设备的一个实例。最后描述的是替换实施例。

用于位置编码的普遍原理

位置码被用于对一个或多个维度中的位置进行编码。为了简化描述, 首先设想所述位置码位于第一维度, 在这个实例中, 第一维度处于 x 方向。在这个方向上使用了一个周期性的主数字序列 (main number sequence) 来进行编码, 所述周期性的主数字序列具有在周期性主数字序列中无歧义确定预定长度的各个局部序列位置的特性。在这个实例中, 预定长度是 6。因此, 如果从周期性的主数字序列中的任何位置提取六个连续的数字, 那么这六个数字在主数字序列中只以这个顺序出现一次。如果主数字序列的末端连接到主数字序列的开端, 那么这种特性同样适用。由此将主数字序列称为是周期性的。在这个实例中, 使用了二进制的主数字序列。如果要无歧义确定具有六个数字的局部序列位置, 那么主数字序列由此具有 $2^6 = 64$ 的最大长度, 并且在主数字序列中, 长度为 6 的局部序列可以具有位置 0 -

63. 然而如下文将要描述的那样, 如果选择了长度为 63 的主数字序列, 则可以得到改进的纠错特性。由此在下文中假设主数字序列的长度是 63 并且由此定义了范围 0 - 62 中的唯一位置。实践中, 在具有这个长度的主数字序列的情况下, 有可能具有一个将各个局部序列转换成主数字序列中的位置的表, 反之亦然。

假设主数字序列开端如下所示:

0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0...

然后, 例如在主数字序列中, 局部序列 0, 0, 0, 0, 0, 0 具有无歧义位置 0, 局部序列 1, 1, 1, 1, 1, 0 具有无歧义位置 9, 局部序列 1, 1, 1, 0, 1, 0 具有无歧义位置 11。

位置编码基于使用周期性主数字序列的不同旋转或循环移位。为了在 x 方向对位置进行编码, 在穿越表面的列中以某些其他方法打印或排列主数字序列, 以多种方式对其进行旋转或循环移位, 其中所述列处在 y 方向上, 该方向正交于对位置进行编码的方向。可以在同一列中重复打印出主数字序列, 如果在 y 方向编码的位置比对应于主数字序列长度的位置更多, 那么这个操作将是必需的。然后则使用主数字序列的相同旋转来进行所有重复。然而也可以在不同的列中使用不同的旋转。

图 1 示意性显示了一张具有 7 个列 $x \sim x+6$ 的纸 1。在主数字序列的不同旋转中, 仅仅打印出了第一数字。整个主数字序列由方框 B 做出示意性标记。另外在图 1 中, 在每一列中, 具有六个数字的第一局部序列 S 是用虚线绘制的方框来进行标记的。如上所述, 在主数字序列中, 每一个局部序列 S 都具有一个无歧义确定的位置。

每一对相邻的列都定义了一个差分数字 (difference number) d 。差分数字 d 是由各个列中第一局部序列的主数字序列位置之间的差分给出的。作为替换, 如果在所述列中向下一步的局部序列位置之间选取差分, 由于使用相同方式来移动所述位置, 因此结果将会是相同的。这样一来, 不管在列中以何种“高度”比较主数字序列中的局部序列位置, 差分数字 d 都是相同的。因此, 对每一对列来说, 差

分数字 d 在 y 方向上都是常数。

在列 $x+5$ 中, 主数字序列的第一局部序列是 0, 0, 0, 0, 0, 0, 它对应的是主数字序列中的位置 0。在列 $x+6$ 中, 主数字序列的第一局部序列是 1, 0, 0, 0, 1, 1, 它对应了主数字序列中的位置 57。因此, 这些列中的差分或循环移位是 57, 所以 $d(x+5)=57$ 。应该指出的是, 所述差分数字是以主数字序列长度为模来确定的。

可以用这种方式编码的差分数字处于范围 $0 \sim K-1$, 其中 K 是主数字序列的长度, 在本范例中 $K=63$, 借助于此, 有可能对范围 $0 \sim 62$ 中的差分数字进行编码。通过选择主数字序列的不同旋转, 有可能创建一个差分数字序列, 下文中将其称为基本数字序列 (primary number sequence) 或基本差分数字序列 (primary difference number sequence), 所述序列具有这样一种特性, 即预定长度的各个局部序列在基本数字序列中都具有无歧义确定的位置。在本实例中, 预定长度是 5。由于包含 5 个差分数字的各个局部序列在基本差分数字序列中都具有一个无歧义确定的位置, 因此可以将其用于在 x 方向对位置进行编码。因此, 在图 1 中, 局部序列 $S(x) \sim S(x+5)$ 对五个差分数字 $d(x) \sim d(x+4)$ 进行编码, 这些差分数字指定了基本数字序列的一个局部序列。

并且应该指出, 在实践中通常不会将局部序列与其确切值一起打印出来, 而是将其与一个图形编码一起打印。

主数字序列的二进制数字的列构成了一个矩阵, 下文中将其称为 x 矩阵。

如果主数字序列长度为 K , 那么基本差分数字序列中的基数等于 K , 并且其最大长度是 K^5 , 在本范例中则是 63^5 。然而在实践中, 对如此大的基数而言, 使用一个表来把局部序列转换成基本差分数字序列中的局部序列, 这种处理并不是切实可行的, 反之亦然。但是, 如果为了能够使用可管理表格而减小基数, 则会减少可以编码的位置数目。

这个问题是通过构造更短的差分数字序列的基本差分数字序列

来解决的，在下文中将其称为辅助差分数字序列（secondary difference number sequence）或辅助数字序列（secondary number sequence），对辅助差分数字序列而言，在对位置码进行编码和解码的时候，通过使用这种辅助差分数字序列，可以在更小的表格中调整局部序列及其位置，以便在基本差分数字序列中确定与特定 x 坐标相对应的局部序列，反之亦然。

可以如下确定辅助差分数字序列：

首先，只允许差分数字处于这样一个范围，其中不同差分数字的数目可以分解成至少两个因数。在长度为 63 的基本数字序列的实例中，只允许差分数字处在一个长度为 54 的范围以内。实际上，数字 54 可以分解成 $2 \times 3 \times 3 \times 3$ 。作为选择，也可以选择 60 个差分数字，也就是 $5 \times 3 \times 2 \times 2$ 个差分数字，然而为使表格尽可能小，因此选择那些给出尽可能小的因数的数字将是非常有利的。

其次，形成了数目与不同差分数字可以分解的因数数目相同的辅助差分数字序列。因此在这个实例中，辅助差分数字序列的数目等于四。

另外，我们让每个因数在其相应的辅助差分数字序列中形成一个基数。在这个实例中，我们由此得到一个基数为 2 的辅助差分数字序列以及基数为 3 的三个辅助差分数字序列。

因此，如果辅助差分数字序列还具有如下特性，即长度为 5 的局部序列在辅助差分数字序列中具有唯一位置，那么，辅助差分数字序列的最大长度分别是 32 和 243。在具有这种长度的辅助差分数字序列的情况下，实践中可以将局部序列转换成位置，反之亦然。作为替换，如果我们选择使用 60 个差分数字，那么我们将会得到一个基数为 5 并且由此最大长度是 3125 的辅助差分数字序列，这就给出了一个表，所述表占用了大得多的存储器，但仍旧是可以管理的。

最后，对辅助差分数字序列的长度进行选择，以使所述长度成对互质。这意味着对每对辅助差分数字序列而言，一个差分数字序列长度不会与第二差分数字序列长度具有任何相同的因数。另外，这还

意味着如果重复每一个差分数字序列，那么，在经过 $L = l_1 * l_2 * \dots * l_m$ 个位置之前，不会从各个辅助差分数字序列中产生相同的局部序列组合。其中 l_1 是辅助差分数字序列 1 的长度， l_2 是辅助差分数字序列 2 的长度，依此类推直到 l_m ，它是最后一个辅助差分数字序列的长度。当然，如果只有两个辅助数字序列，那么 $L = l_1 * l_2$ 。

图 2 示意性描述了这个操作。在顶端显示了基本数字序列 PD，所述序列在一个具有差分数字的长序列中行进，在这里，所述差分数字由 X 表示。在下方示意性显示了辅助数字序列 A1 ~ A4。竖线显示了数字序列开始的位置。而基本数字序列的局部序列以及辅助数字序列的相应局部序列则是由虚线表示的。可以看出，局部序列对应于辅助数字序列中的不同位置。

在这个实例中，通过组合辅助差分数字序列的局部序列而显示的基本差分数字序列的局部序列是双射的。然而这并不是必需的。

在这种情况下，在 y 方向这种第二维度上的位置编码可以根据等同于第一维度中的位置编码的规则来执行。然后在所述表面上，通过不同的循环移位而在行中排列基本数字序列，所述行即为图 1 的 x 方向。差分数字是在相邻的行之间定义的，这些差分数字形成一个基本差分数字序列，该序列可以由辅助差分数字序列构造。在第二维度中也可以使用另一个主数字序列、另一个基本差分数字序列以及其他辅助差分数字序列。不同数字序列中的基数也可以不同于第一维度中的编码所使用的基数。然而在这个实例中，在 x 方向和 y 方向上使用了相同序列。在一种与 x 方向上的编码相对应的方式中，y 方向上的编码产生一个矩阵，在这个矩阵中，行的值包含了主数字序列的二进制数值。

如果重叠 x 矩阵和 y 矩阵，对于合成的 xy 矩阵中的各个点而言，将会具有一个来自 x 矩阵的二进制数字和一个来自 y 矩阵的二进制数字。由此可以在各个点获取下面这四个二进制数字组合：0, 0; 0, 1; 1, 0; 和 1, 1。举例来说，在一个表面上，可以使用图 3a ~ d 中显示的方法来对这些不同组合进行图形编码，其中点 30 在四个方

向中的任何一个方向上从光栅中的标称点 31 开始移动，各个标称点对应于光栅中的线条 32 之间的交点。WO01/26032 中更详细地描述了这种类型的图形编码，然而，实际的基础（underlying）位置编码是以一种不同方式实施的。本实例中使用的编码则如下所示：

数值	位移	二进制数字对
1	右	0, 1
2	上	0, 0
3	左	1, 0
4	下	1, 1

所述光栅矩阵可以是虚拟的，在这种情况下，不会将其与位置码一起明确打印在表面上。

图 4 示意性显示了采用了以具有位置码 410 的纸张 400 为形式的产品的一部分。应该强调的是，举例来说，图 4 中的位置码相对于适合例如数字化字迹的版本有了极大增加。另外，光栅 420 是在图 4 中标记的。然而，通常并不将其打印出来。

当然，对这四对二进制数字而言，也可以选择其它类型的图形编码，例如不同大小的四个点，具有不同形状四个标记或是具有不同旋度的四个标记。

由于位置编码基于主数字序列中不同的局部序列位置之间的差分，并且可以使用不同的局部序列对来得到相同差分，因此可以使用多种方式获取基本差分数字序列。更具体地说，根据选择哪一个局部序列来启动 x 矩阵中的第一列（ $x=0$ ）和 y 矩阵的第一行（ $y=0$ ），可以得到 63 个不同“版本”的基本差分数字序列，其中选择局部序列即为选择主数字序列的旋度。由此可以产生 $K \times K$ （在这个实例中是 $63 \times 63 = 3969$ ）个不同“版本”的位置码，这些位置码使用相同的基本差分数字序列来进行 x 方向和 y 方向的编码。在下文中，位置码的这些不同版本称为分段。用于 x 方向的位置码的段编号 x_s 以及用于 y 方向的位置码的段编号 y_s 可以分别指定成 x 方向和 y 方向的附加坐

标。

在以上实例中，仅仅使用了在理论上可以使用主数字序列进行编码的差分数字的一个子集，更具体地说，只用到了 63 个可能差分数字中的 54 个。理论上可行的差分数字的局部范围可以用多种方式来加以选择。在这个实例中，接近 0（以基本数字序列长度为模）的差分数字可能在位置码中产生不必要的规律性。因此在这个实例中，差分数字是在范围（5，58）中选择的。这必须结合基本数字序列中的局部序列与辅助数字序列中的局部序列之间的换算来加以考虑，下文将会对此进行描述。现在假设位置码排列在一个表面上。读取设备可以具有一个区域传感器，所述传感器检测与至少 6×6 个点相对应的位置码子集。当读出位置码的时候，可以使用多种方式而使读取设备相对所述位置码而保持旋转。位置码图像本身不会显示位置码与读取设备之间的旋度，因为无论它是否旋转了 0、90、180 还是 270 度，位置码大体看来都是一样的。然而，在旋转位置码的时候将会改变各个点的位移方向，这转而导致改变了那些通过点的位移而被编码的二进制数字对。所述变化如何出现则依赖于位移与二进制数字对之间的换算。在下文中，假设所述换算是如上进行的。则其中存在三种不同的情况：

1) 在位置码图像旋转了 180 度的情况下，当试图基于图像来解码所述位置时，向后读取周期性主序列，所述主序列对处于未旋转位置码的 x 和 y 位置进行编码。在如上所述将位移与二进制数字对相结合的情况下，由于倒转读取位置码，因此将会反转所解码的二进制数字。

2) 在顺时针旋转 90 度的情况下，在试图解码旋转矩阵中的 y 位置的时候，将会后向读取一个对无旋转矩阵中 x 位置进行编码的二进制数字序列，并且将会倒转所述二进制数字。

3) 在顺时针旋转 270 度的情况下，在试图解码旋转矩阵中的 x 位置的时候，将会后向读取一个对无旋转矩阵中 y 位置进行编码的二进制数字序列，并且将会倒转所述二进制数字。

这意味着如果在右上方的无旋转矩阵中的局部序列从未在周期性主数字序列中出现倒转和反向，则可以检测到 90、180 和 270 度的旋转。如果为主数字序列满足这个条件，则意味着将会显著减小其长度，这转而意味着只能对更少的位置进行编码。

为了解决这个问题，取而代之的是，我们为周期性主数字序列的更长局部序列满足所述条件。这意味着必须读出比实际位置测定所需序列更长的局部序列。在某些情况下，这种更长局部序列在读出过程中就已经可用。如果对位置坐标进行编码的位置码的最小子集是二次的，与在这个实例中的情况一样，则总是必须读出一部分位置码，这部分位置码很大，并且包括整个最小子集，即使是在相对于位置码而把读取设备旋转了 45 度的时候。因此，始终会包含至少一个比位置确定所需序列更长的局部序列。这在图 5 中得到了描述，其中每个正方形 50 对应于一个点，而涂成灰色的正方形则说明，始终可以读出具有两个额外点的至少一行和一系列。

在用于位置确定的主数字序列的局部序列长度为 6 的当前实例中，可以使用长度为 8 的局部序列来检测旋转。因此，这些八位的局部序列必须具有不会在主数字序列中出现反向和反转的特性。对一个 64 位的长主数字序列而言，这一点是无法实现的，这也就是转而将长度改选为 63 的原因。

在上述两种旋转即 90 和 270 度的旋转中，将会读出一个沿正确路线旋转的序列和一个沿错误路线旋转的序列。在使用用于旋转检测的上述方法的情况下，只会在这两个读出方向（x 和 y）中的第一个方向检测到一个旋转误差。在第二个读出方向上则会看到对正确旋转图像的第一读取方向上的位置进行编码的旋转移位。

在顺时针旋转 90 度的情况下，当在 x 方向进行解码的时候，将会看到对正确旋转图像的 y 坐标进行编码的循环移位。因此，所看到的是 y 方向的基本差分数字序列的局部序列，但是所述序列沿着错误路线旋转（从右到左而不是从左到右，这是 x 方向的基本差分数字序列的局部序列的情况）。

在相应的路线上可以看到，当在 y 方向进行解码的时候，如果旋转了 270° ，那么 x 方向的基本差分数字序列的局部序列沿着错误路线旋转。

局部序列沿着错误路线旋转的原因是由于 x 和 y 方向的基本差分数字序列的相对定位。从总的位置编码图案边缘开始， x 方向的基本差分数字序列是在顺时针方向运转的，而 y 方向的基本差分数字序列则是在逆时针方向运转的。如果它们在相同方向运转，那么在上述情况中，它们不会沿着错误路线旋转，而是会沿着正确路线旋转。

在相反方向运转基本差分数字序列则具有很多优点。在这些情况中可以看到，基本差分数字序列的局部序列沿着正确路线旋转并且对“正确”坐标进行解码，但却是在错误方向上。这个坐标不但符合从解码所需基本差分数字序列开始的六个循环移位，而且周围的移位证明所解码的坐标是正确的。在基本差分数字序列沿着相反方向运转的情况下，也对来自基本差分数字序列的六个循环移位进行解码，但却是以错误顺序进行的，由此导致产生一个坐标，从统计上讲，这个坐标不会符合错误旋转的基本差分数字序列的周围部分。这个事实导致产生更好的检错概率。

另一个优点则在于，如果基本差分数字序列沿着相反方向行进，那么连续错误即解码到相邻位置的多个连续图像的误差将会更少。更为特殊的是，在一个基本差分数字序列中沿着错误路线旋转的不同相邻组的六个循环移位并不有助于对相邻坐标进行编码。在那些错误坐标范围很大的地方，连续错误要比其他错误更加难以检测。

与用于检测旋转的原理相同的基本原理也可以用于纠错。例如，可以使用如下方式来选择主数字序列，其中不会在比位置确定所需序列更长的某些预定长度的局部序列中出现一个二进制数字的反转。如果确信可以在这种更长的局部序列中检测到除一之外的所有二进制数字，则可以纠正错误位。

位置码的检错和纠错特性可以通过智能选择主数字序列而得到显著改善。进一步的提高则可以通过选择辅助差分数字序列来实

现。

在以上实例中，已经使用附录 A 中显示的方法选择了主数字序列和辅助差分数字序列。

辅助差分数字序列具有以下特殊的纠错特性：

假设正好只有一个主数字序列的局部序列没有得到正确解码，并且这将导致主数字序列中错误位置的解码。由于各个位置都被用于计算两个相邻的差分数字，因此这些差分数字都会受到不正确解码的影响。如果任何一个差分数字超出了所用差分数字范围（5，58），则会立即检测到所述错误。然而，如果并非如此，那么，在由基本差分数字序列的局部序列所产生的四个辅助差分数字序列的局部序列组合中，在两个相邻位置之间会有至少一个局部序列失真。由于这两个第一辅助差分数字序列 A1 和 A2 都具有基数 3 并且未曾使用的差分数字数目是 $9 = 3 \times 3$ ，因此这两个序列中的任何一个序列的失真都具有如下特性，那就是这两个受影响数字的总和始终具有以 3 为模的相同的值。两个辅助数字序列 A1 和 A2 则具有如下特性，即，对长度为 7 的各个局部序列而言，在辅助数字序列中只会发现单个错误位置解码所产生的 14 个可能的局部序列失真中的至多一个。另外，第三辅助数字序列 A3 还具有如下特性，对长度为 7 的各个局部序列而言，在辅助数字序列中将会发现由单个错误位置解码所产生的 13 个可能的局部序列失真中的至多一个。而对于 28 个可能失真中的 7 个失真而言，第四辅助数字序列 A4 也具有相同的特性。因此，所检测的主数字序列的局部序列的单个错误解码概率是非常大的。

结合选定的序列，在一个分段的各个维度上，可以对总共 410815348 个不同位置进行编码。如所述，不同分段的数目可以是 63^2 。因此，可以编码的位置总数是 $63^2 \times 410815348^2 = 6.7 \times 10^{20}$ 个位置。

如果每个位置都与一个 $0.3 \times 0.3 \text{ mm}^2$ 的表面相对应，那么这相当于能够在六千万平方公里的表面上编码的唯一位置。从理论上讲，可以借助位置码来对这个由所有唯一点构成的表面进行编码，并且可以

将这个表面称为虚表面 (imaginary surface)。因此, 由位置码编码的坐标是虚表面上的点的绝对坐标。一部分位置码可以应用在一个物理基底或表面上。然后, 所述位置码对这个基底上的位置进行编码。然而, 通常并没有将所述坐标设计成物理基底上的位置的绝对坐标, 而是将其设计成虚表面上的点的坐标。

位置编码的实际实例

下文将对如何在实践中实现位置编码进行描述。通常, 用户没有兴趣打印那些来自坐标 0 及其上方的整个位置码, 反之, 倒不如说是为用户分配了一个特定坐标区域, 所述区域可用于一个特定应用。于是, 其中一个实际问题是如何确定位置码是用于这个特定坐标区域的。

可以在一个编码设备中确定或产生位置码, 所述设备包括某种适当类型的处理器单元。然后, 处理器单元包括实际处理器、工作存储器和程序存储器, 其中保存了具有确定位置码的指令的计算机程序。处理器单元可以引入到典型的个人计算机、打印机或某些其他合适的设备之中。作为选择, 编码设备包括专用硬件, 例如 ASIC (专用集成电路)、FPGA (现场可编程门阵列) 或是可以经过修改而适于特定任务的相似单元, 此外还可以是数字和/或模拟电路或其某些适当组合。

产生位置码的有意义的最小部分即为对单个点的坐标进行编码的部分。在这个实例中, 这相当于产生一个矩阵, 所述矩阵具有 6×6 个位移值, 这些位移值 (displacement value) 描述了从它们在一个规则光栅中的相应光栅点开始的 6×6 个点的位移。

然而, 通常需要产生一个用于对多个点的坐标进行编码的位置码的子集, 也就是一个更大的矩阵。

在一般情况下, 输入编码设备的信号包括六个参数:

x: 它是位置码左边与之相距最远的列的 x 坐标。

y: 它是用于最上的行的坐标。

xs: 它表示的是哪些位置码分段被用于 x 位置码。

ys: 它表示的是哪些位置码分段被用于 y 位置码。

宽度: 它表示位置码中列的数目, 也就是 x 方向的坐标区域大小。

高度: 它表示位置码中行的数目, 也就是 y 方向的坐标区域大小。

这些参数是由用户或从外部单元输入编码设备的。可选地, 还可以包含一个指向以位移值填充的矩阵的指针。并且为矩阵分配一个另外的存储器。

编码设备的任务是将矩阵中的各个点与一个位移值 o 相关联, 其中所述位移值指示了将会在基底上印制的位置码中移动一个点的方向。

如上所述, 位移值 o 可以由二进制数字对中的二进制数字确定, 在这个实例中, 位移值 o 可以是 $0 \sim 3$, 其中第一个二进制数字构成了当前点的 x 位置码的值 m_x , 而第二个二进制数字则构成了当前点的 y 位置码的值 m_y 。如果二进制数字对是已知的, 则可以在一个把二进制数字对转换为位移值的表格 O 中查找位移值。

当然, x 位置码的值 m_x 依赖于我们所设置的点, 即 x 和 y , 并且还依赖于位置码的当前分段。因此, $m_x = m_x(x, xs, y)$ 。在一种对应方式中, y 位置码的值 m 依赖于我们设置的点, 即 x 和 y , 并且还依赖于 y 位置码的当前分段。所以 $m_y = m_y(y, ys, x)$ 。

如果我们确定了周期性主数字序列是在分段 xs 的列 x 中的什么位置 s 开始, 并且我们添加了 y (其中通过以主数字序列长度为模来得到该结果), 那么我们会在用于 x 编码的点 x, y 获得我们所在的主数字序列的位置。然后, 我们可以在主数字序列表 M 中查找主数字序列中的哪一个数字 (二进制数字) 具有这个位置, 由此在点 x, y 组成 x 编码的值。这可以表述为 $m_x(x, y, xs) = M[(s(x, xs) + y) \pmod{63}]$ 。

通过执行对应操作, 可以在点 x, y 发现 y 编码的值。这样一来, 我们可以使用所述二进制数字对, 从中确定将要保存在矩阵中的

点 x, y 的位移 o 。

因此, 编码设备首先需要计算的是: 对列 x 而言, 主数字序列始于哪个位置 s , 其中列 x 是在对点 x, y 的坐标进行编码的位置码的子集中距离左边最远的列。换言之, 可以说循环移位是为列 x 中的主数字序列而计算的。

所述操作可以如下执行。

如先前所述, 差分数字 $d(x)$ 是由列 x 中第一局部序列以及相邻列 $x+1$ 中第一局部序列的主数字序列中的位置定义的。更具体地说, 事实是

$$d(x) = (s(x+1, xs) - s(x, xs)) \pmod{\text{主数字序列长度}}$$

其中 $s(x, xs)$ 是主数字序列中的位置或关于位置码分段 xs 的列 x 中的主数字序列的循环移位。

作为一种定义, 还包括了这样一种情况, 即 $s(0, xs) = xs$, 也就是说, 用于 x 位置码最前列中的主数字序列的循环移位定义了 x 分段。

以此为基础, 可以将 $s(x, xs)$ 确定为

$$s(x, xs) = (xs + \sum_{j=0}^{x-1} d(j)) \pmod{\text{主数字序列长度}}$$

在实践中, 由于不可能为那些对多个点坐标进行编码的位置码保存定义了所有 x 的差分数字值 $d(x)$ 的基本差分数字序列, 因此, 现在使用的事实是基本差分数字序列中的各个差分数字都可以在一个来源于各个辅助差分数字序列的一个数字的组合上映射。在这里, 通常基于:

$$d(x) = dc(1, x) + b_1 * dc(2, x) + \dots + b_1 * \dots * b_{n-1} * dc(n, x),$$

其中 n 是辅助差分数字序列数目, b_i 是差分数字序列中的基数, $i=1 \sim n-1$, 并且可以通过结合差分数字序列 i 中的数字而在表格 DC_i 进行查找, 由此确定 $dc(i, x)$, 因此, $dc(i, x) = DC_i[x \pmod{\text{差分数字序列 } i \text{ 的长度}}]$ 。应该指出的是, 如果只有两个辅助差分

数字序列，那么当然只会使用上述表达式中的前两项。

在所讨论的情况中，由此我们得到

$d(x)=5+dc(1, x)+3*dc(2, x)+9*dc(2, x)+18*dc(3, x)$ 其中常数 5 是来源于使用了范围 5~58 中的差分数字这一事实。

由于来自不同辅助差分数字序列的作用是独立的，因此可以分别为每个辅助差分数字序列计算对相当于差分 x 的所有差分数字总和。在图 2 中可以将其视为分别为每个序列求出相当于位置 x 的所有差分数字的总和。

在实践中，很容易使用常数和表格来确定这个总和。实际上，辅助差分数字序列的整个周期的作用是恒定的，由此可以预先将其确定为常数并且加以保存。在这里，这些常数是指定的 $DCCS_i$ 。也可以预先确定差分数字序列的不完整周期的作用并且将其保存在表格中，对于相应辅助差分数字序列而言，所述作用给出了上至这个位置并且包括这个位置的差分数字的总和。在这里将表格表示为 $DCICS_i$ 。来自常数项 5 的作用则是 $5*x$ 。

由此可以将用于每一个辅助差分数字序列的项 $\sum(j=0...x-1)d(j)$ 确定为 $DCCS_i*(x \text{ div } l(i)) + DCICS_i[x(\text{mod } l(i))]$,

其中 $l(i)$ 是辅助差分数字序列 i 的长度。

当确定了 $s(x, xs)$ 的时候，如上所述，可以使用 y 的已知值以及查表来确定处于点 x, y 的 x 位置码的值。

对 x 位置码中的后续列而言，当然，不需要如上所述来确定 $s(x, xs)$ ，但是取而代之的是，使用了 $s(x, xs)$ 已经已知这个事实，并且使用以下关系式来计算 $s(x+1, xs)$ 。

$d(x)=(s(x+1, xs)-s(x, xs))$ (以主数字序列长度为模)

$d(x)=5+dc(1, x)+b_1*dc(2, x)+...+b_1*b_2*...b_{m-1}*dc(m, x)$

$dc(i, x)=DC_i[x \text{ 与差分数字序列 } i \text{ 的长度相模}]$ 。

然后则根据输入参数“宽度”来为所需要的数目的列重复进行这个过程。

而 $s(y, ys)$ 则是以一种相应方式确定到 $s(x, xs)$ 的，并且

通过使用输入参数 x ，还可以确定处于点 x, y 的 y 位置码的值。

在得到了这两个值的时候，可以通过在表 O 中进行查找来确定点 x, y 的位移。

为了对位置进行编码，编码设备可以使用以下表格和常数：

M: 序列中不同位置上的主数字序列中的数字。

O: 不同的二进制数字对的位移 (0~3)

DC_i: 处于序列中不同位置的辅助差分数字序列 i 中的数字；

DCICS_i: 辅助差分数字序列 i 的每一个不完全周期中的数字之和。

此外，还可以保存以下常数：

DCCS_i: 辅助差分数字序列 i 中的所有数字之和

l(i): 辅助差分数字序列 i 的长度

K: 主数字序列的长度。

图 6 显示了一个用于进行位置编码的简单流程图。首先在步骤 600，接收那些形式为 x, y, xs, ys ，宽度 (Width)，高度 (Height) 的输入信号。此后，可以并行确定 x 矩阵和 y 矩阵，也可以先 x 后 y 或以相反顺序确定 x 矩阵和 y 矩阵。这在流程图中是用并行的流程来表示的。在步骤 605，首先为 x 位置编码中的第一列确定循环移位 $s(x, sx)$ 。此后，在步骤 610，为第二列确定循环移位 $s(x+1, xs)$ 。随后在步骤 615 应用 $x=x+1$ 并且在步骤 620 调查是否已经为整个宽度即所有列确定了循环移位。如果并非如此，则流程返回到步骤 610，并且重复这个步骤和步骤 615。如果情况就是这样的话，则在步骤 625 中查表确定 x 矩阵中的二进制数字的值。在一种对应的方式中，用于 y 矩阵的二进制数字值是在步骤 630~650 中确定的。最后，关于 xy 矩阵的位移是在步骤 655 中查表确定的。

当然，这仅仅是一个如何执行位置编码的粗略的示范性实例，在实践中，它可以借助多种途径来改变。例如，一旦确定了点的循环移位，则可以连续确定一个二进制数字对以及所述点的相应位移。由此没有必要缓存 x 和 y 矩阵。

所述方法也可以在一个计算机程序中执行。来自计算机程序的输出信号没有必要正好是具有位移值的 xy 矩阵，取而代之的是，它可以组成关于构成位置码的主数字序列的局部序列的某些其他指示。

位置解码的实际范例

可以在一个解码设备中对位置码进行解码，所述解码设备可以包括一个用于读出位置码的传感器和一个某种恰当类型的处理器单元。所述处理器单元包括实际处理器、工作存储器和程序存储器，其中保存了具有对所述位置码进行解码的指令的计算机程序。处理器单元可以引入到一个典型的个人计算机之中，也可以引入手持读取设备或其他一些合适的设备之中。作为选择，解码设备可以通过专用硬件来实现，例如 ASIC（专用集成电路）、FPGA（现场可编程门阵列）或是可以经过修改而适于这个特定任务的相似单元，并且也可以通过数字和/或模拟电路或是二者的某些恰当组合来实现所述解码器。

以下描述的是解码设备的一个特定实例。

以下描述参考了图 7 中的流程图。

输入到解码设备的信号包括图像或是测得的位置码子集的某些其他数字表示，所述子集对至少一个点的坐标进行编码，步骤 700。在这个实例中，如上所述，位置码是由相对光栅中的光栅点而在预定路线上移动的点来进行图形表示的。

在解码的第一个步骤中，举例来说，解码设备通过使用阈值处理来识别图像中的点并且将光栅适合所述点，步骤 705。例如，可以借助于在申请人的专利申请 WO01/26034 中描述的方式并且使用不同点对之间的距离来确定所述光栅，也可以借助 WO01/75783 中描述的方式并且使用傅立叶变换来确定所述光栅。对各个光栅点而言，相关点的位移将被确定并且根据所述位移而被给出一个数值 $0 \sim 3$ 。并且至少将会选择一个大小为 $(n_1+1) * (n_1+1)$ 的矩阵，其中 n_1 是用于定位的主数字序列的局部序列长度，并且其位置是在主数字序列中无歧义确定的。而在所涉及的实例中则选择了一个 $8*8$ 的矩阵

$((n_1+2)*(n_1+2))$ 。通过在把位移值转换成二进制数字对的表 OI 中进行查找而将位移值转换成二进制数字对，由此将这个矩阵分成了一个 x 矩阵（也称作 x 位置码）和一个 y 矩阵（也称为 y 位置码），步骤 710。每个二进制数字对中的第一个二进制数字形成了 x 矩阵，而每个二进制数字对中的第二个二进制数字则形成了 y 矩阵。

在下一个步骤，对 x 和 y 矩阵的四种可能类型的旋转（0、90、180 和 270 度）进行调查并且确定当前旋度，步骤 715。在这里使用了先前描述的事实，那就是如果错误旋转矩阵，则会出现主数字序列上丢失的长度为 8 的局部序列。

同时，可以基于长度为 8 的局部序列来执行某种程度的纠错。

在确定了正确旋转的时候，后续步骤中仅仅使用了 x 和 y 矩阵，这两个矩阵处于正确路线之中并沿着正确路线旋转，其大小为 n_1*n_1 ，在这个实例中则是两个 $6*6$ 的矩阵。 x 矩阵和 y 矩阵可以并行解码，也可以按照先 x 后 y 的顺序或者与此相反的顺序来解码，图 7 中通过并行的流程指示了这一点。

在解码 x 矩阵的第一个步骤 720，确定一个基本差分数字序列的局部序列，图 7 中将其称为基本局部序列。更具体地说，在 x 矩阵中，所述矩阵的列中的二进制数字构成了主数字序列的局部序列。主数字序列中的相应位置是通过在表 MI 中进行查找来确定的，其中表 MI 将局部序列转换成位置。此后，确定那些由主数字序列的局部序列来进行编码的差分数字。在这个实例中，它们的数目是五。更具体地说，将差分数字确定为相邻局部序列位置之间的差分 d ，所述差分是通过以主数字序列长度为模来确定的。

这样一来，由此得到的差分数字序列是基本差分数字序列的一个局部序列，所述序列被用于 x 方向的编码，但是由于其大小，因此并没有将所述序列保存在解码设备中。取而代之的是使用了基本差分数字序列的特性，即各个局部序列都可以转换成来自各个辅助差分数字序列的局部序列的组合。由此在步骤 725 中将基本局部序列转换成辅助局域序列。更具体地说，如下所示来改写各个差分数字：

$$d=d_1+b_i*d_2+\dots+b_1*b_2 \quad *b_{n-1}*d_n,$$

其中 n 是辅助差分数字序列的数目, b_i 是辅助差分数字序列 i 中的基数。在所涉及的实例中, 这将会变成:

$$d=5+d_1+3*d_2+9*d_3+18*d_4$$

其中可以通过整除或查表来确定 $d_1 \sim d_4$ 。

因此, 在基本差分数字序列的已解码局部序列中, 5 个数字中的每一个都会产生四个数字的 d_1, d_2, d_3, d_4 。因此将会得到五个 d_1 的数字, 所述数字形成了第一辅助差分数字序列的一个局部序列, 此外还获得了五个 d_2 的数字, 它们形成了第二辅助差分数字序列的一个局部序列, 并且获得了 5 个 d_3 的数字, 它们形成了第三辅助差分数字序列中的一个局部序列, 以及五个 d_4 的数字, 它们形成了第四辅助差分数字序列的一个局部序列。以下给出了一个实例:

d	23	45	51	9	37
d1	2	0	0	0	1
d2	1	0	2	0	0
d3	0	1	1	1	0
d4	1	2	2	0	2

在辅助差分数字序列中, 局部序列的相应位置是通过在四个表格 DCI_i 中进行查找来获取的, 步骤 730, 其中表 DCI_i 把相应的辅助差分数字序列中的局部数字序列转换成相应的辅助差分数字序列中的位置。

对每个局部序列而言, 由此得到了一个位置 p_i 。对这些位置而言, 事实是:

$$P=p_1(\text{mod } l_1)$$

$$P=p_2(\text{mod } l_2)$$

.

.

$$P=p_m(\text{mod } l_m)$$

其中 P 是对应于坐标 x 的基本差分数字序列中的位置, p_i 是辅

助差分数字序列 i 中的位置, l_i 是辅助差分数字序列 i 的长度, m 是辅助差分数字序列的数目。

举例来说, 如在 Niven Suckerman, Introduction to the theory of numbers, Springer Verlag 中详细描述的实现中那样, 这个方程组可以使用众所周知的中国余数定理 (chinese remainder theorem) 求解。由此在步骤 735 中确定 x 。

定义 $L = \text{prod}(i=1, m)l_i$ 以及 $q_i * (L/l_i) = l \pmod{l_i}$ 。

然后可以得到基本差分数字序列中的位置 P :

$$P = (\text{sum}(i=1, n)((L/l_i) * p_i * q_i)) \pmod{L}$$

在所涉及的实例中,

$$L = l_1 * l_2 * l_3 * l_4 = 236 * 233 * 31 * 241 = 41081534$$

并且得到了

$$q_1 = 135$$

$$q_2 = 145$$

$$q_3 = 17$$

$$q_4 = 62.$$

例如, 如果为辅助差分数字序列的局部序列得到了位置 $p_1=97$; $p_2=176$; $p_3=3$ 以及 $p_4=211$, 那么根据中国余数定理, 基本差分数字序列中的相应位置 P 是 170326961。

当确定了 P 的时候, 下一个步骤是确定所述位置属于哪个 x 分段。首先, 这是在首先假设 $y=0$ 并且通过计算 $s(x, 0)$ 来执行的, 也就是说, 它是列 x 的局部数字序列的主数字序列中的位置, $x_s = 0$ 。 $s(x, 0)$ 则是通过与如上所述相同的方式并且结合位置 x 的位置码判定而使用辅助差分数字序列来计算的。如果从用于 x 方向定位的六个局部数字序列中的第一局部数字序列的位置中减去 $s(x, 0)$, 则可以得到了 $y=0$ 的 x_s 。

根据图 7 的步骤 740 ~ 755, 在一种与关于 x 方向的上述方式相对应的方式中, y 矩阵是在解码 x 矩阵之前、之后或者与之并行解码的, 基本差分数字序列中的一个位置是为 y 方向确定的。此外, 所涉

及的 y 分段 y_s 是以一种与为 x 分段的上述方式相对应的方式而计算的, 但所述计算是在假设 $x=0$ 的情况下进行的。最后在步骤 760, 可以通过 y 加以考虑来确定 x_s , 并且在 $y=0$ 的条件下将 y (与主数字序列的长度相模) 从 x_s 中减去以及在 $x=0$ 的条件下将 x 从 y_s (与主数字序列的长度相模) 中减去, 由此可以对 x 加以考虑, 从而确定 y_s 。

由此可以使用以下表格来对位置码进行解码:

MI: 它将主数字序列的局部序列转换成主数字序列中的位置,

DCI: 它将 n 个辅助差分数字序列中的每一个序列的局部序列转换成 n 个辅助差分数字序列中的位置, 以及

OI: 它将位移值转换成二进制数字对。

应该强调的是, 图 7 的流程图是一个如何在例如计算机程序中实现位置解码的粗略的示意性实例。

手持读取设备的实例

位置码可以由不同类型的读取设备读取和解码。这种设备可以具有一个用于获取位置码的数字表示的传感器, 并且具有用于识别数字表示中的位置码并且解码所述位置码的装置。如上所述, 这些装置可以包括一个具有存储器和适当程序的处理器, 也可以包括专用的硬件或数字和/或模拟电路, 还可以包括这些设备的恰当组合。

图 8 显示了读取设备的一个实例。所述设备包括一个与笔近似具有相同形状的外壳 11。在外壳短的一端具有开口 12。短的一端紧靠表面或者与之保持一个很短距离, 其中位置确定是在所述表面上执行的。

所述外壳实质上包括一个光学部分、一个电子线路部分和一个电源。

光学部分包括至少一个用于照射将要成像的表面的发光二极管 13, 并且包括一个用于记录二维图像的感光区域传感器 14, 例如 CCD 或 CMOS 传感器。可选的, 所述设备还可以包括一个光学系统, 例如镜子和/或透镜系统。发光二极管可以是一个红外线发射二

极管并且传感器可以感知红外线。

设备的电源供应是从电池 15 获得的，所述电池安装在外壳中的分离隔间。此外也可以经由电缆而从外接电源（未显示）获取电源供应。

电子线路部分包括一个处理器单元，所述处理器单元具有经过编程而对来自传感器的图像进行记录并且根据所述图像执行位置确定的处理器，此外所述电子线路部分还包括一个工作存储器和一个程序存储器。

在这个实施例中，所述设备还包括一个笔尖 17，借助于此，可以在将要执行位置确定的表面上执行基于普通颜料的书写。笔尖 17 可以是可扩展和可伸缩的，由此用户可以控制是否对其进行使用。但在某些应用中，所述设备根本不需要具有笔尖。

基于颜料的书写具有一种类型，该类型对红外线而言是透明的，并且所述标记会适当吸收红外线。通过使用发出红外线的发光二极管以及对红外线敏感的传感器，可以在不受到上述与图案在一起的写入干扰的情况下执行图案检测。

所述设备还可以包括按钮 18，并且可以借助于此来激活和控制设备。并且该设备还具有一个收发信机 19，以便进行往返于设备的信息的无线传输，例如使用红外线，无线电波或超声。该设备还可以包括一个用于显示位置或记录信息的显示器 20。

所述设备可以在不同的物理外壳之间分开，其中第一外壳包括记录位置编码图案的图像所需要的组件，并且还包括用于将上述信息发送到由第二外壳包含并且根据所记录的一个或多个图像来执行位置确定的组件所需要的组件。

替换实施例

上述实施例中描述了一个在两个维度对位置进行编码的位置码。然而也可以使用位置码在一个维度对位置进行编码。例如在上述实例中，可以沿 x 轴并在 $y = 0$ 的情况下使用一个位置码“条带”。作为选择，可以仅仅使用一个 x 编码矩阵。在这种情况下，矩阵中的各

个点只具有一个值，因此，在二进制的主数字序列的情况下，我们只需要对一个 0 和一个 1 进行图形编码。

一维位置编码的一个应用可以是充当条码的等价物。

上述实施例中使用的是二进制的主数字序列。然而，也可以为主数字序列使用一个不同的基数。然后这将导致产生若干个需要在位置编码矩阵中的各个点进行编码的不同值。

在上述实施例中，在 x 方向和 y 方向使用相同的主数字序列和相同的辅助差分数字序列进行编码。然而这并不是必要的。不同的序列也可用于在不同的维度进行编码。然而，考虑到需要提供存储空间来保存用于这些序列的表，因此较为有利的是使用相同序列来进行各个方向的编码。

在上述实施例中使用了 6×6 个点或符号来进行位置编码。当然也可以使用更少或更多的符号。并且在两个维度上没有必要使用相同数目的符号。

在上述实施例中，基本差分数字序列的局部序列是双射映射在局部数字序列组合之上的，并且辅助差分数字序列的局部数字序列组合双射映射在基本差分数字序列的局部序列上。然而这并不是必要的。基本差分数字序列的不同局部序列也可以映射在辅助差分数字序列的同一局部序列组合上。并且有可能会存在并未映射在局部序列组合上的基本差分数字序列。此外还可以具有比基本局部序列更多的局部序列组合。在上述实施例中，光栅是一个正交网格。但是它还可以具有其它形式，例如 60 度角的菱形网格、三角形或六边形格子等等。

并且也可以在少于或多于四个方向上使用结合位移的图形编码，例如沿着六边形虚拟光栅的三个方向的位移。在一个正交光栅中，为了更易于创建光栅，仅仅可以使用两次位移。

在上述实施例中，为了避免差分数字接近 0，使用了范围 (5, 58) 中的差分数字。但是也可以使用其他的范围，其中包括例如差分数字 3 和 4。

在上述实施例中，可以使用光学方式读取图案，因此传感器是光学的。然而，所述图案也可以基于一个与光学参数不同的参数。在这种情况下，传感器当然必须具有可以读出所涉及参数的类型。这种参数的实例是化学、声学或电磁参数。并且也可以使用电容或电感参数。

在上文是借助一张纸来例示所述产品的。然而，所述产品可以是使用任何一个上述参数而被应用了所述位置码的产品。举例来说，如果位置码是光学的，那么可以使用打印机将其打印出来或者在一个使用正常打印技术的产品上进行打印。

附录 A

基本数字序列 M:

0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0,
1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1,
0, 0, 0, 1, 1

辅助差分数字序列:

A1=0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0,
1, 0, 0, 2, 0, 0, 0, 1, 1, 0, 0, 0, 1, 2, 0, 0, 1, 0, 2, 0,
0, 2, 0, 2, 0, 1, 1, 0, 1, 0, 1, 1, 0, 2, 0, 1, 2, 0, 1, 0,
1, 2, 0, 2, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 2, 1, 0, 1,
0, 2, 1, 1, 0, 0, 1, 2, 1, 0, 1, 1, 2, 0, 0, 0, 2, 1, 0, 2,
0, 2, 1, 1, 1, 0, 0, 2, 1, 2, 0, 1, 1, 1, 2, 0, 2, 0, 0, 1,
1, 2, 1, 0, 0, 0, 2, 2, 0, 1, 0, 2, 2, 0, 0, 1, 2, 2, 0, 2,
0, 2, 2, 1, 0, 1, 2, 1, 2, 1, 0, 2, 1, 2, 1, 1, 0, 2, 2, 1,
2, 1, 2, 0, 2, 2, 0, 2, 2, 2, 0, 1, 1, 2, 2, 1, 1, 0, 1, 2,
2, 2, 2, 1, 2, 0, 0, 2, 2, 1, 1, 2, 1, 2, 2, 1, 0, 2, 2, 2,
2, 2, 0, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 0, 1, 2, 2, 1,
2, 2, 0, 1, 2, 1, 1, 1, 1, 2, 2, 2, 0, 0, 2, 1, 1, 2, 2

A2=0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0,

1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 2,
 0, 0, 0, 1, 2, 0, 1, 0, 1, 2, 1, 0, 0, 0, 2, 1, 1, 1, 0, 1,
 1, 1, 0, 2, 1, 0, 0, 1, 2, 1, 2, 1, 0, 1, 0, 2, 0, 1, 1, 0,
 2, 0, 0, 1, 0, 2, 1, 2, 0, 0, 0, 2, 2, 0, 0, 1, 1, 2, 0, 2,
 0, 0, 2, 0, 2, 0, 1, 2, 0, 0, 2, 2, 1, 1, 0, 0, 2, 1, 0, 1,
 1, 2, 1, 0, 2, 0, 2, 2, 1, 0, 0, 2, 2, 2, 1, 0, 1, 2, 2, 0,
 0, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 0, 0, 1, 2, 2, 1, 2, 0, 1,
 1, 1, 2, 1, 1, 2, 0, 1, 2, 1, 1, 1, 2, 2, 0, 2, 2, 0, 1, 1,
 2, 2, 2, 2, 1, 2, 1, 2, 2, 0, 1, 2, 2, 2, 0, 2, 0, 2, 1, 1,
 2, 2, 1, 0, 2, 2, 0, 2, 1, 0, 2, 1, 1, 0, 2, 2, 2, 2, 0, 1,
 0, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 0, 2, 2, 2,

A3=0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1,
 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,

A4=0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2, 0, 0, 2, 0,
 1, 0, 0, 0, 1, 1, 2, 0, 0, 0, 1, 2, 0, 0, 2, 1, 0, 0, 0, 2,
 1, 1, 2, 0, 1, 0, 1, 0, 0, 1, 2, 1, 0, 0, 1, 0, 0, 2, 2, 0,
 0, 0, 2, 2, 1, 0, 2, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1,
 0, 1, 2, 0, 1, 1, 1, 1, 0, 0, 2, 0, 2, 0, 1, 2, 0, 2, 2, 0,
 1, 0, 2, 1, 0, 1, 2, 1, 1, 0, 1, 1, 1, 2, 2, 0, 0, 1, 0, 1,
 2, 2, 2, 0, 0, 2, 2, 2, 0, 1, 2, 1, 2, 0, 2, 0, 0, 1, 2, 2,
 0, 1, 1, 2, 1, 0, 2, 1, 1, 0, 2, 0, 2, 1, 2, 0, 0, 1, 1, 0,
 2, 1, 2, 1, 0, 1, 0, 2, 2, 0, 2, 1, 0, 2, 2, 1, 1, 1, 2, 0,
 2, 1, 1, 1, 0, 2, 2, 2, 2, 0, 2, 0, 2, 2, 1, 2, 1, 1, 1, 1,
 2, 1, 2, 1, 2, 2, 2, 1, 0, 0, 2, 1, 2, 2, 1, 0, 1, 1, 2, 2,
 1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 0, 1, 2, 2, 1, 2, 2, 0, 2, 2,
 2, 1, 1.

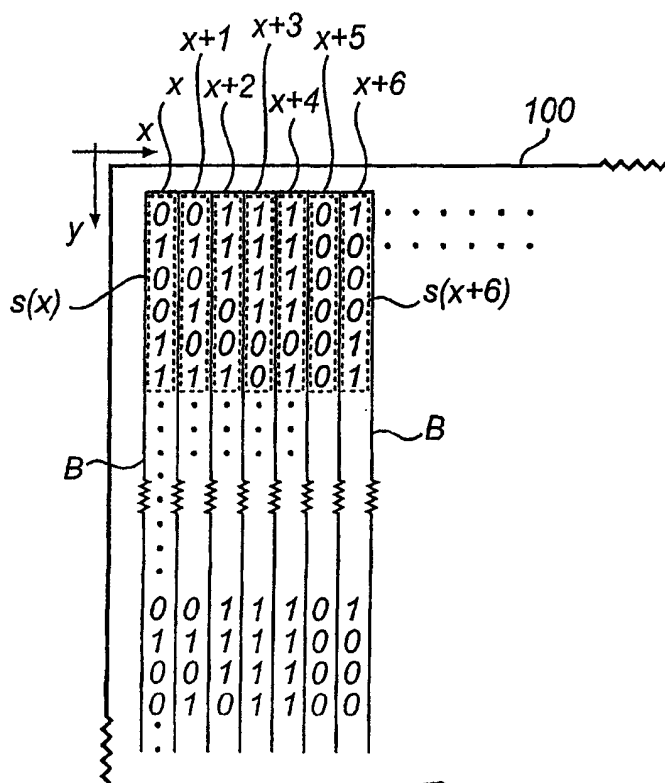


图1

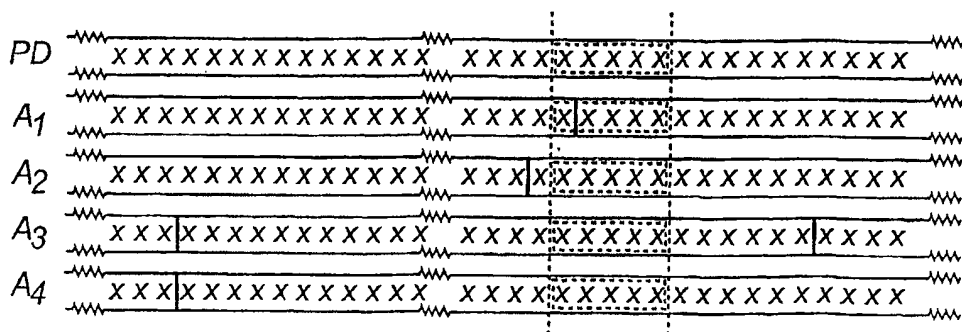
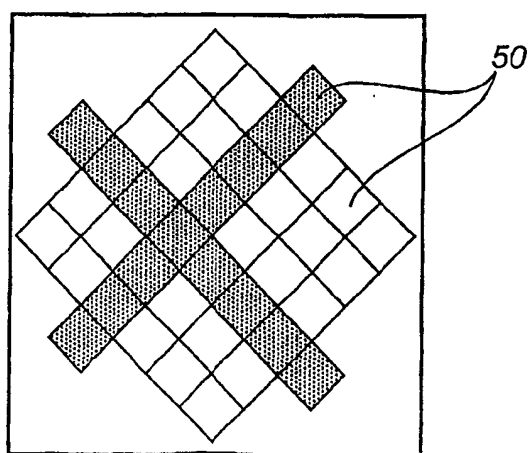
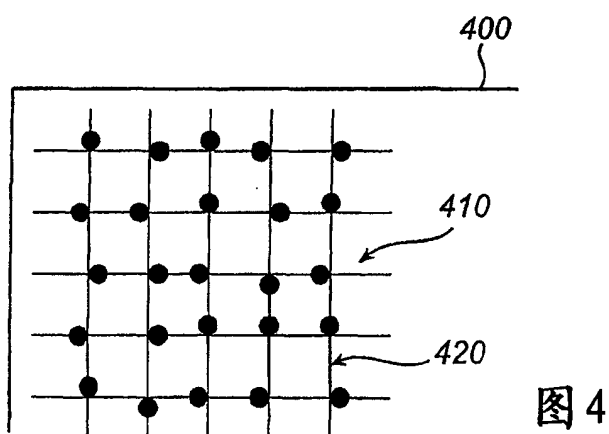
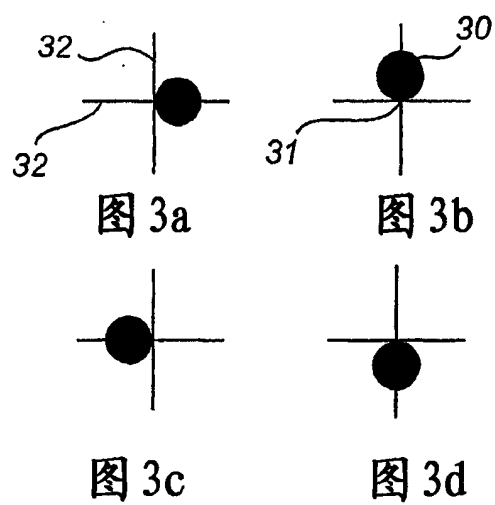


图2



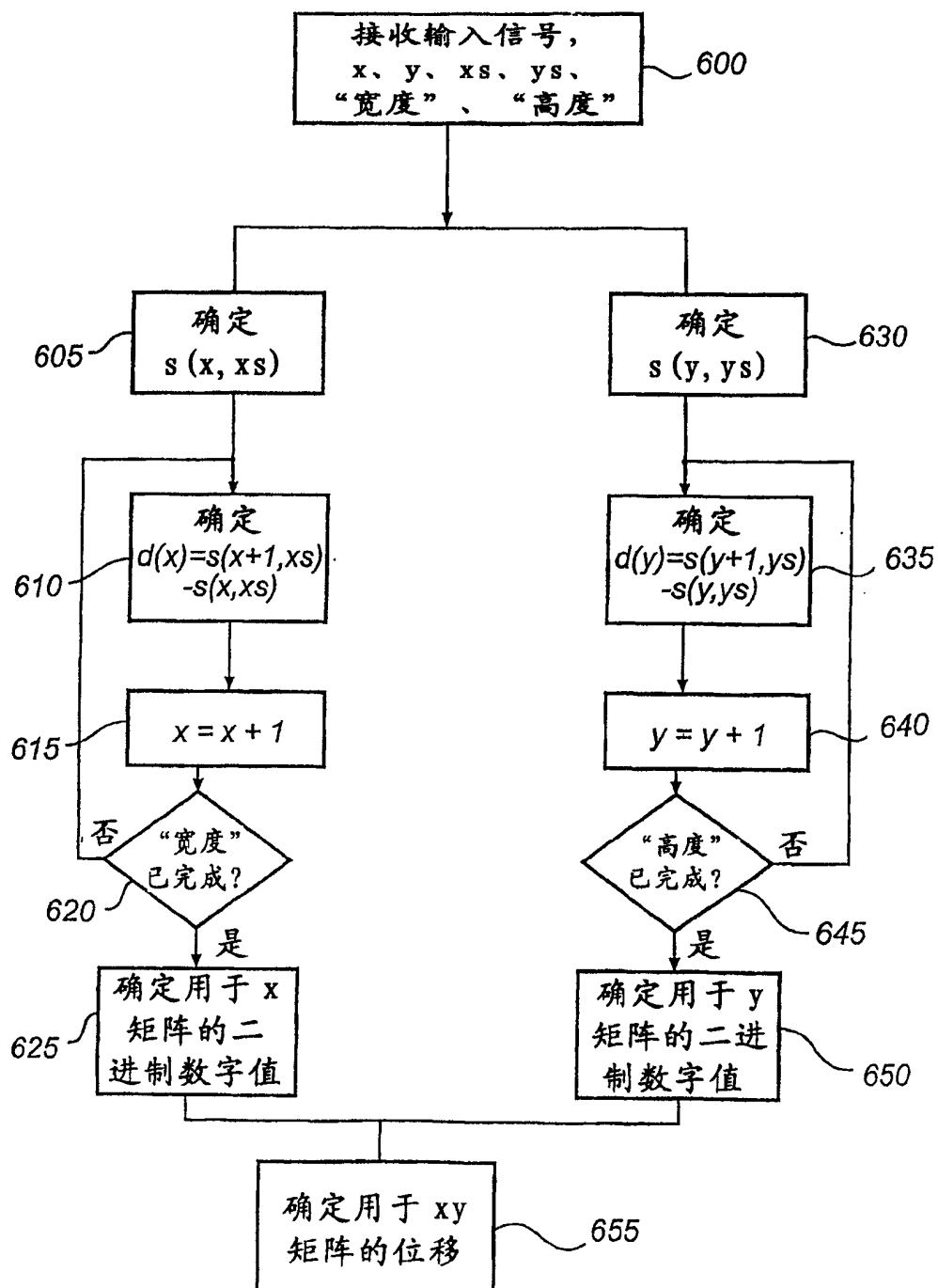


图6

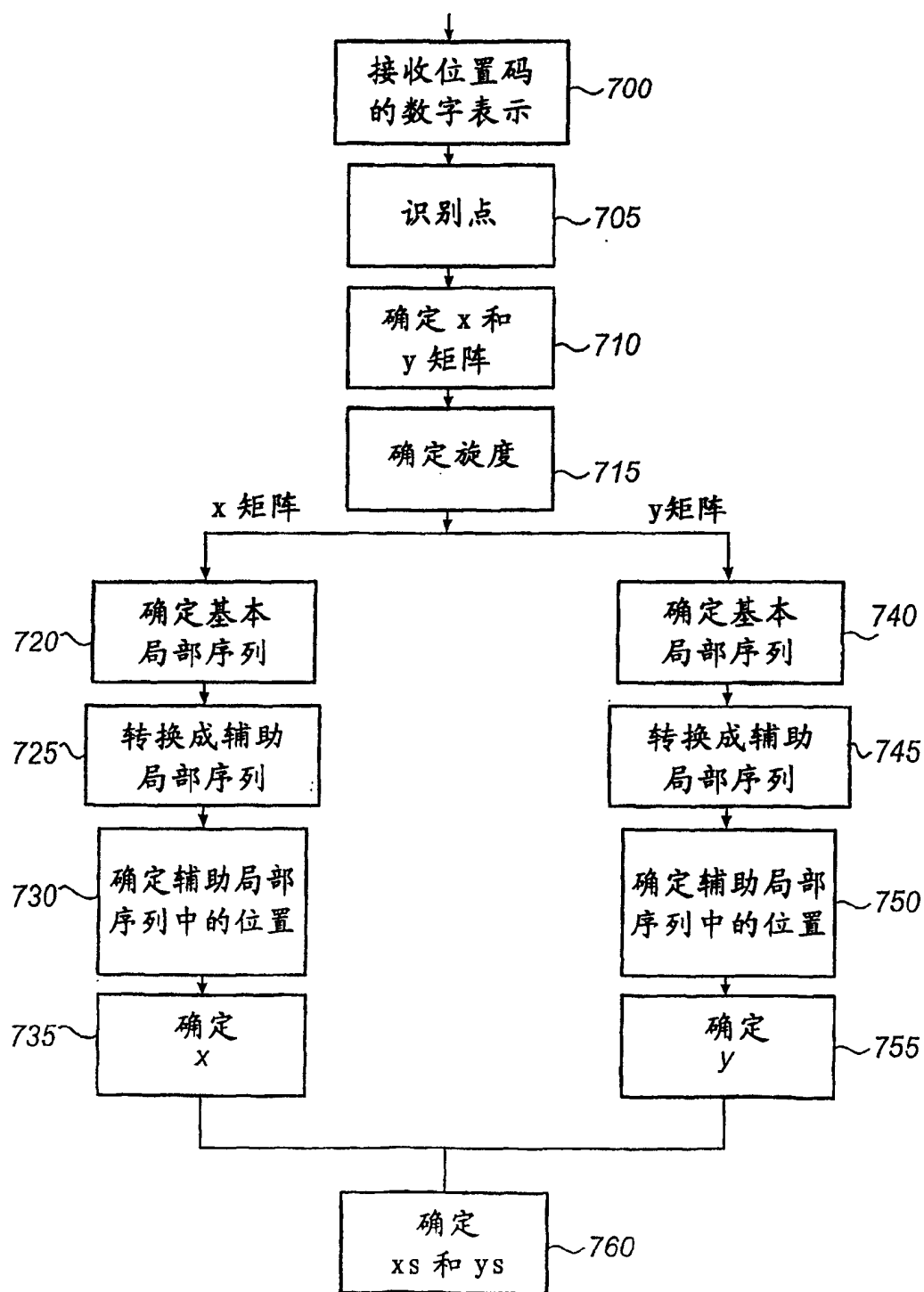


图7

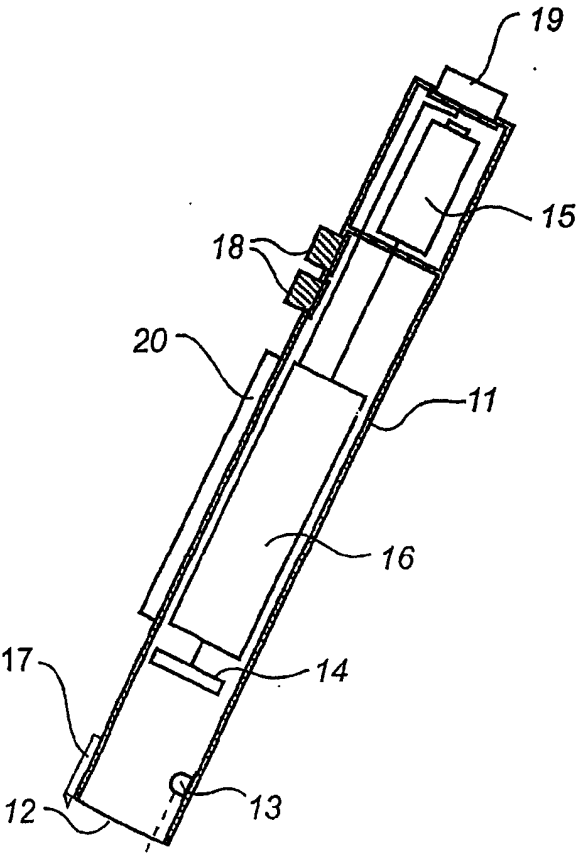


图 8