

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06K 19/06 (2006.01)



[12] 发明专利说明书

专利号 ZL 02821501. X

[45] 授权公告日 2008 年 9 月 3 日

[11] 授权公告号 CN 100416603C

[22] 申请日 2002.10.29 [21] 申请号 02821501. X
[30] 优先权

[32] 2001.10.29 [33] SE [31] 0103589-8
[86] 国际申请 PCT/SE2002/001962 2002.10.29
[87] 国际公布 WO2003/038741 英 2003.5.8
[85] 进入国家阶段日期 2004.4.28

[73] 专利权人 阿诺托股份公司

地址 瑞典隆德

[72] 发明人 玛斯·佩特·彼得森

[56] 参考文献

EP0469868A2 1992.2.5

EP0350933A2 1990.1.17

US4817171 1989.3.28

WO0126032A1 2001.4.12

审查员 王艳妮

[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 李 玲

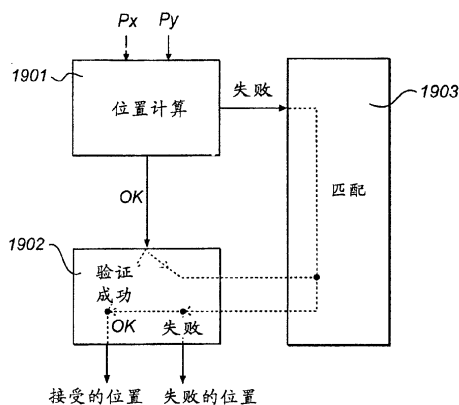
权利要求书 4 页 说明书 50 页 附图 17 页

[54] 发明名称

用于对位置编码图案进行解码的方法和设备

[57] 摘要

本发明涉及一种用于位置编码的方法，基于从位置编码图案的局部区域中确定的信息来计算一个位置，其中所述局部区域是由一个传感器成像的。如果位置计算失败，则将来自传感器记录的局部区域的信息与如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个所述局部匹配区域在位置编码图案中定义了一个位置。根据所述匹配步骤，匹配局部区域所定义的一个位置是位置编码图案中最有可能用于作为解码位置而被接受的成像局部区域的位置。此外在这里还描述了一种执行所述方法的设备和计算机程序。



1. 一种用于位置解码的方法，包括如下步骤：

计算步骤，基于从位置编码图案的局部区域中确定的信息来计算一个位置，其中所述局部区域是由传感器来成像的，由传感器成像的所述局部区域理想地定义位置编码图案中的位置；

其特征在于包括以下步骤：

匹配步骤，如果位置计算失败，则将来自传感器成像的局部区域的图像的信息与关于如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个所述匹配局部区域在位置编码图案中定义了一个位置；以及

接受步骤，接受匹配局部区域定义的多个位置中的一个位置，其中根据所述匹配步骤，所接受的位置是位置编码图案中最有可能用于由传感器成像的成像局部区域的位置。

2. 如权利要求 1 所述的方法，其中所述匹配是在与先前接受的位置接近的区域中执行的。

3. 如权利要求 1 所述的方法，其中所述匹配每次都是在位置编码图案中的一个维度上执行的。

4. 如权利要求 1 所述的方法，其中位置编码图案借助多个元素来对每个位置进行编码，其中每个所述元素可以采用至少两个不同的值，并且其中所述匹配步骤包括如下子步骤：

确定匹配局部区域的每个元素所采用的值；以及

对属于所述匹配局部区域的每个匹配局部区域来说，基于来自位置编码图案的成像子集的信息，确定成像局部区域元素与该匹配局部区域元素采用相同的值的概率量度。

5. 如权利要求 4 所述的方法，其中来自成像局部区域的信息包括一个概率值集合，其中每个概率值都表示成像局部区域元素之一采用一个可能值的概率，并且其中概率量度是基于所述匹配局部区域元素采用的值的概率值来确定的。

6. 如权利要求 4 所述的方法，其中概率量度是基于成像局部区域中与所述匹配局部区域中的相应元素采用相同值的元素的数量来确定的。

7. 如权利要求 4 所述的方法，其中位置编码图案中的每个位置都是由一个符号矩阵来编码的，其中每个符号都构成一个元素。

8. 如权利要求 4 所述的方法，其中位置编码图案中的每个位置是由一个符号矩阵来编码的，所述符号被划分成多个群组，并且其中每个群组构成一个元素。

9. 如权利要求 4 所述的方法，其中每个位置是由一个符号矩阵来编码的，所述符号被划分成多个群组，并且其中所述元素中的每一个都是由至少两个符号群组来定义的。

10. 如权利要求 7 所述的方法，其中每个符号包括一个标记，所述标记在四个预定方向中的一个方向上离开位置编码图案中的一个标称位置。

11. 如权利要求 8 所述的方法，其中一个群组中的符号值对应于循环数字序列中的一个局部序列，所述循环数字序列具有如下特性，那就是具有预定长度的每个局部序列都在数字序列中具有一个无歧义确定的位置。

12. 如权利要求 9 所述的方法，其中一个群组中的符号值形成循环数字序列中的一个局部序列，所述循环数字序列具有如下特性，那就是具有预定长度的每个局部序列都在数字序列中具有一个无歧义确定的位置，并且其中由群组成对定义的元素包括数字序列中各群组的位置之间的差分数字。

13. 如权利要求 1 所述的方法，还包括步骤：借助于来自成像局部区域的信息来验证基于成像局部区域所接受的位置。

14. 如权利要求 1 所述的方法，其中位置编码图案中的每个位置都是借助多个位置数字来编码的，所述位置数字成对定义多个确定所述位置的差分数字；其中关于如何构成位置编码图案的信息包括所述差分数字，来自成像局部区域的信息则包括从局部区域中确定的位置

数字；并且其中所述匹配包括确定从成像局部区域中确定的并在差分数字方面相互一致的位置数字的数目。

15. 一种用于位置解码的设备，所述设备包括一个适配成根据从位置编码图案的局部区域中确定的信息来计算一个位置的信号处理单元，其中所述局部区域是由传感器成像的，由传感器成像的所述局部区域理想地定义位置编码图案中的位置；

其特征在于：所述信号处理单元还适配成执行匹配步骤：如果位置计算失败，则将来自传感器成像的局部区域的图像的信息与关于如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个所述匹配局部区域在位置编码图案中定义了一个位置，此外所述信号处理单元还适配成接受匹配局部区域所定义的多个位置中的一个位置，其中根据所述匹配步骤，所接受的位置是位置编码图案中最有可能用于成像局部区域的位置。

16. 如权利要求 15 所述的设备，其中所述匹配是在与先前接受的位置接近的区域中执行的。

17. 如权利要求 15 所述的设备，其中所述匹配每次都是在位置编码图案中的一个维度上执行的。

18. 如权利要求 15 所述的设备，其中位置编码图案借助多个元素来对每个位置进行编码，其中每个所述元素可以采用至少两个不同的值，并且其中所述信号处理单元进一步包括：

确定匹配局部区域的每个元素所采用的值的装置；以及

对属于所述匹配其它局部区域的每个匹配局部区域来说，基于来自位置编码图案的成像子集的信息，确定成像局部区域元素与该匹配局部区域元素采用相同的值的概率量度的装置。

19. 如权利要求 18 所述的设备，其中来自成像局部区域的信息包括一个概率值集合，其中每个概率值都表示成像局部区域元素之一采用一个可能值的概率，并且其中概率量度是基于所述匹配局部区域元素采用的值的概率值来确定的。

20. 如权利要求 18 所述的设备，其中概率量度是基于成像局部

区域中与所述匹配局部区域中的相应元素采用相同值的元素的数量来确定的。

21. 如权利要求 18 所述的设备, 其中位置编码图案中的每个位置都是由一个符号矩阵来编码的, 其中每个符号都构成一个元素。

22. 如权利要求 18 所述的设备, 其中位置编码图案中的每个位置是由一个符号矩阵来编码的, 所述符号被划分成多个群组, 并且其中每个群组构成一个元素。

23. 如权利要求 18 所述的设备, 其中每个位置是由一个符号矩阵来编码的, 所述符号被划分成多个群组, 并且其中所述元素中的每一个都是由至少两个符号群组来定义的。

24. 如权利要求 21 所述的设备, 其中每个符号包括一个标记, 所述标记在四个预定方向中的一个方向上离开位置编码图案中的一个标称位置。

25. 如权利要求 22 所述的设备, 其中一个群组中的符号值对应于循环数字序列中的一个局部序列, 所述循环数字序列具有如下特性, 那就是具有预定长度的每个局部序列都在数字序列中具有一个无歧义确定的位置。

26. 如权利要求 23 所述的设备, 其中一个群组中的符号值形成循环数字序列中的一个局部序列, 所述循环数字序列具有如下特性, 那就是具有预定长度的每个局部序列都在数字序列中具有一个无歧义确定的位置, 并且其中由群组成对定义的元素包括数字序列中各群组的位置之间的差分数字。

27. 如权利要求 15 所述的设备, 还包括: 借助于来自成像局部区域的信息来验证基于成像局部区域所接受的位置的装置。

28. 如权利要求 15 所述的设备, 其中位置编码图案中的每个位置都是借助多个位置数字来编码的, 所述位置数字成对定义多个确定所述位置的差分数字; 其中关于如何构成位置编码图案的信息包括所述差分数字, 来自成像局部区域的信息则包括从局部区域中确定的位置数字; 并且其中所述匹配包括确定从成像局部区域中确定的并在差分数字方面相互一致的位置数字的数目。

用于对位置编码图案 进行解码的方法和设备

技术领域

本发明一般地涉及对位置编码图案进行解码，尤其涉及一种位置解码方法，其中包括根据从传感器所记录的位置编码图案局部区域中确定的信息来计算一个位置的步骤。

本发明还涉及一种可用以解码位置编码图案的设备、计算机程序和存储介质。

背景技术

目前存在具有如下特性的位置编码图案，其中对具有最小预定尺寸的位置编码图案的局部区域来说，可以借助局部区域中的信息和一种预定算法来计算所述位置编码图案中的绝对位置。在 US5,477,012 (O. Sekendur) 中公开了这种位置编码图案的实例，其中每个位置都是借助了一个符号来编码的，此外在 WO92/17859 (Hewlett Packard Co.)、WO00/73983 (Anoto AB)、WO01/26033 (Anoto AB) 和 US5,973,110 (Xerox Corp) 中也公开了这种位置编码图案的实例，在这些文献中，每个位置是借助多个符号来编码的。

举例来说，这种类型的位置编码图案可用于信息的数字记录，其中所述信息是借助具有位置编码图案的基底上的用户单元而被手写和/或手绘的。在移动用户单元的同时，在用户单元顶部连续记录了位置编码图案的图像。其中每个图像中的一个或多个符号都解码到一个位置。这些经过解码的位置共同构成了用户单元在基底上所进行的移动的数字描述。

只要在基底上正确再现了位置编码图案并且在记录的图像中完美成像了位置编码图案，则所述位置解码应该能够很好的工作。然而在实践中，在将位置编码图案应用于基底的时候，或多或少存在着导致位置编码图案失真的风险。此外还会出现由于用户单元成像系统故障和用户书写时倾斜用户单元而造成的用户单元无法正确成像位置编

码图案的情况，以及由此合乎比例地成像所述位置编码图案。这转而导致从位置编码图案中解码出错误的位置。

如果位置编码图案中包含了允许进行检错的冗余信息，则可以发现错误解码的位置。如果错误解码的位置与先前和/或随后解码的位置的距离不合理，那么也可以发现所述错误解码的位置。

如果检测到错误解码的位置，那么可以保留或滤去所述位置。如果保留该位置，那么这个以数字方式记录的位置不会正确表示用户单元在基底上的移动。如果改为滤去错误解码的位置，那么在以数字方式记录的位置中将会出现一个“间隙”。多个间隙将会妨碍到解译基底上所写内容的可能性。

发明内容

本发明的一般目的是解决这个问题并由此改善在经过位置编码的基底上生成的信息的数字表示的质量。

其中一个具体目的是增加可以从位置编码图案的图像中正确解码的位置数目。

另一个具体目的是在没有不恰当增加解码所需时间的情况下增加正确解码的位置数目。

而另一个具体目的则是在没有不恰当增加对用于解码的存储空间和/或处理器能力的需要的情况下，增加正确解码的位置数目。

这些和其他目的将会从以下描述中变得非常明显，它们全部或部分地借助于依照权利要求的方法、计算机程序、存储介质和设备来实现。

根据第一个方面，本发明尤其涉及一种用于位置解码的方法，其中包括根据从传感器成像的位置编码图案的局部区域中确定的信息来计算一个位置的步骤。所述方法还包括如下步骤：如果关于所述位置的计算失败，则将来源于传感器成像的局部区域的信息与如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个匹配局部区域都在位置编码图案中定义了一个位置；以及接受所述匹配局部区域定义的多个位置之一，其中依照所述匹配步骤，所述位置是位置编码图案中最可能与成像局部区域相对应的位置。

当位置计算失败时，可以通过执行匹配来获取一个位置，在大多

数情况下，与通过失败计算可能得到的位置相比，在对局部区域进行成像的时候，通过所述匹配获取的位置将会更好地反映出传感器的位置。此外还消除了数字表示中的间隙。

所述匹配可以在位置计算失败时立即进行。作为选择，所述匹配也可以在稍后记录了更多位置信息的时候执行，由此可以简化匹配。

举例来说，由于计算中的局部结果或最终结果明显出错，因此位置计算有可能会失败。当局部结果中包含了不应该在对位置编码图案进行解码的过程中得到的值的时候，所述局部结果明显是错误的。就时间而言，如果解码位置与直接在先或随后的解码位置不处在某个距离以内，或者如果将最终结果视为是极不可靠的，那么最终结果明显是错误的。

在这里应该指出，与匹配局部区域相比，传感器成像的局部区域可以具有更大的尺寸，尤其可以包括比启用位置计算严格所需要的信息还要多的信息。

所述匹配可以结合定义一个位置的位置编码图案中的所有可能的局部区域来加以执行。然而，如果位置编码图案很大，那么所述匹配有可能非常耗时。取而代之的是，由于将要确定的位置最有可能相对接近先前接受的位置，因此可以对所述匹配加以限制，以便只在与先前接受的位置相邻的有限匹配区域中执行匹配。如果存在若干个已接受的位置，其中这些位置基于那些在与位置计算失败的局部区域接近的时间上成像的局部区域，那么毫无疑问，可以使用一个以上的已接受位置来限定这个有限的匹配区域。

有限匹配区域的大小可以是预定的，或者举例来说，在对位置计算失败的局部区域进行成像之前和/或之后，可以通过传感器的速度和/或加速度来动态确定所述区域的大小。

如果假设位置计算失败的局部区域与先前接受的位置应该相隔一个距离 d 以内，其中 d 是位置编码图案中的位置或符号间的距离，那么在局部区域中，与失败的位置计算有关的信息必须与来源于位置编码图案中 $(2d+1)^2$ 的局部区域的信息匹配。如果改为在位置编码图案中每次只在一个维度上执行匹配，那么作为替换，局部区域中与失败的位置计算有关的信息只需要与来自 $2(2d+1)$ 的局部区域的信息相匹配。然而这种情况需要在位置编码图案中分别在不同维度上对

位置进行解码。此外还应该强调，所述匹配不必在所有维度上执行。如果位置计算在一个维度上取得成功，但在另一个维度上失败，那么只在位置计算失败的维度上执行所述匹配就已经足够。

此外还可以对位置编码图案进行设计，以便借助多个元素来对每个位置进行解码，其中每个元素都可以采用（assume）至少两个不同的值。然后在匹配中，成像的局部区域与匹配局部区域在元素级别上进行比较，并且将会确定所述元素在两个匹配局部区域中采用同一个值的概率量度。

举例来说，所述概率量度可以包括相应的元素值数目。作为另一个实例，来自成像局部区域的信息可以包括一个概率值集合。更具体地说，所述集合中的每个概率值可以表示成像局部区域中某个元素采用其中一个可能值的概率。所述概率值集合可以包括关于各个元素可能值的概率值，也可以只包括关于某些元素和值的概率值。所述概率量度是根据概率值确定的，其中概率值针对的是匹配局部区域的元素所采用的值。举例来说，如果匹配局部区域的元素采用的是位置编码图案组合已知而导致已知的值的某种组合，则可以将相应元素值的概率值相乘，以便确定成像局部区域中的元素采用这些值的组合概率值。与只确定了用于局部区域的元素的值的情况相比，通过这种基于概率值的匹配，可以将这其中的信息用于一个更大范围，其中所述概率值针对的是从成像局部区域中确定的元素。

举例来说，上述元素可以与那些组成位置编码图案并且在传感器使用位置编码图案的时候成像的符号相对应。这些元素也可以对应于更高或已处理等级的信息。举例来说，这些符号可以划分到群组中，其中每个群组都构成了一个元素，所述元素可以根据用于群组中的符号的值而采用不同的值。此外，这些元素也可以由两个或更多的群组来共同定义。由此不必将成像局部区域与具有在位置编码图案中发现的相同分辨率的匹配局部区域相匹配，这样可以节省匹配过程中的时间，并且可以在为了等待匹配而要保存那些来自成像局部区域的信息的时候节省存储空间。

附加的匹配优化可以用不匹配所有可用信息的方式来实现。

在执行了匹配之后，基于成像局部区域所接受的位置可以借助于来自成像局部区域的信息来进行验证，所述区域通常包含了比位置计

算/匹配所需信息还要多的信息。由于匹配总是产生一个位置，因此所述验证步骤将是非常有利的。如果选择了与先前接受的位置相邻的匹配区域，则不能使用与这个位置相差的距离来确定通过匹配所获取的位置，然后较为有利的是借助成像局部区域中可用的附加信息来验证所述位置。在这里，较为有利的是通过概率计算来执行所述验证。

非常有利的是，以上用于位置解码的方法可以与 Anoto AB 开发的位置编码图案一起使用，并且举例来说，在 WO01/26033 中对所述图案进行了描述。在当前申请的附录 A 中描述了这种图案的一个变体。位置编码图案恰恰具有这样一种性质，那就是可以在各个维度上独立解码所述位置编码图案中的一个位置。

上述方法可以用硬件或是软件的形式来实施。因此，本发明还涉及一种计算机程序，其中包含了计算机程序代码，当在计算机中执行所述代码的时候，该代码将导致计算机执行一个依照任何一个方法权利要求的方法。

计算机程序可以由安装在与传感器相同的单元或是安装在另一个单元中的处理器执行，其中所述传感器对位置将被解码的局部区域进行成像。

计算机程序可以用存储介质的形式来保存和分发，例如 RAM、ROM、光盘、磁带或是市场上可以买到的其他存储介质。所述存储介质也可以是一个传送信号。

根据另一个方面，本发明涉及一种设备，其中包括一个适配成根据从传感器成像的位置编码图案的局部区域中确定的信息来计算一个位置的信号处理单元。所述信号处理单元还适配成：如果位置计算失败，则将来自传感器成像的局部区域的信息与如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个匹配局部区域都在位置编码图案中定义了一个位置，以及接受匹配局部区域定义的多个位置中的一个位置，根据所述匹配步骤，所述局部区域是位置编码图案中最有可能用于成像局部区域的位置。

根据一个方面，本发明涉及一种用于位置解码的方法，包括如下步骤：基于从位置编码图案的局部区域中确定的信息来计算一个位置，其中所述局部区域是由传感器来成像的，由传感器成像的所述局部区域理想地定义位置编码图案中的位置；其特征在于包括以下步骤：如果位置计算失败，则将来自传感器成像的局部区域的图像的信息与关

于如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个所述匹配局部区域在位置编码图案中定义了一个位置；以及接受匹配局部区域定义的多个位置中的一个位置，其中根据所述匹配步骤，所接受的位置是位置编码图案中最有可能用于由传感器成像的成像局部区域的位置。

根据另一个方面，本发明涉及一种用于位置解码的设备，所述设备包括一个适配成根据从位置编码图案的局部区域中确定的信息来计算一个位置的信号处理单元，其中所述局部区域是由传感器成像的，由传感器成像的所述局部区域理想地定义位置编码图案中的位置；其特征在于：所述信号处理单元还适配成：如果位置计算失败，则将来自传感器成像的局部区域的图像的信息与关于如何在多个匹配局部区域中设计位置编码图案的信息相匹配，其中每个所述匹配局部区域在位置编码图案中定义了一个位置，此外所述信号处理单元还适配成接受匹配局部区域所定义的多个位置中的一个位置，其中根据所述匹配步骤，所接受的位置是位置编码图案中最有可能用于成像局部区域的位置。

从以上关于方法的论述可知，所述设备的优势是很明显的。

举例来说，所述信号处理单元可以由一个经过恰当编程的处理器来实现，也可以由 ASIC（专用集成电路）或 FPGA（现场可编程门阵列）这类经过特殊适配的硬件来实现，此外还通过数字或模拟电路或是二者任意组合来实现。因此，所述信号处理单元具有用于执行解码方法中各个步骤的装置。

附图说明

现在参考附图并借助实施例来对本发明进行更详细的描述，其中：

图 1 是可以使 3 用本发明的系统的示意图；

图 2 示意性地描述了纸上局部区域的解码和匹配；

图 3 示意性地显示了一个带有符号的局部区域；

图 4 示意性地显示了如何在 X 方向上执行编码；

图 5 示意性地显示了如何由四个辅助数字序列 A1 ~ A4 来构造主数字序列 PD；

图 6 显示的是对位置编码图案进行图形编码的一个实例；

图 7 显示了一个位置编码图案经过放大的产物；

图 8 显示的是如何读取附加符号而将其用于纠错;

图 9 是显示了如何对位置编码图案进行解码的一个实例的流程图;

图 10 是一个显示了如何将位置编码图案的局部区域图像解码到一个位置的实例的流程图;

图 11a 显示了一张具有位置编码图案的纸;

图 11b 是与图 11a 中位置编码图案的部分放大;

图 12 显示了位置编码图案中的一个标记的四个理想位置;

图 13 显示了如何对标记的值概率进行计算;

图 14a ~ c 显示了如何对部件的值概率进行计算;

图 15a ~ c 描述的是第一和第二矩阵的创建;

图 16 显示的是一个循环主数字序列;

图 17 描述的是关于序列概率的计算;

图 18 描述的是对位置编码图案进行解码的过程中的步骤;

图 19 描述的是位置编码图案的旋转;

图 20 是一个显示位置解码的流程图;

图 21 是显示位置解码的示意框图; 以及

图 22 示意性地显示了一个可以执行位置解码的设备。

具体实施方式

图 1 示意性地显示了一个用于信息管理的系统, 其中在所述系统中可以使用本发明。所述系统包括一个纸张形式的基底 1、用户单元 2 和外部单元 3。所述纸具有一个位置编码图案 4, 其中只有一小部分位置编码图案 4 是以放大状态示意表示的。在这里可以使用用户单元 2 而以等同于普通的笔的方式而在纸 1 上进行书写, 同时以数字形式来记录书写内容。以数字方式记录的信息可以在用户单元 2 和/或外部设备 3 中得到处理, 其中可以将所述信息自动(联机)发送到所述装置, 也可以由用户用信号通知所述装置发送所述信息。对正确反映了纸上内容并以数字方式记录的信息来说, 尽可能正确和完整地解码所述位

置，这一点是非常重要的。

下文中将会更详细的描述位置编码图案 4，所述位置编码图案 4 由图形符号构成，这些符号是根据预定规则排列和设计的，这意味着如果正确成像了位置编码图案中的局部区域，其中所述局部区域将会具有预定的最小尺寸，则可以借助于成像局部区域中的符号以及一种预定解码算法来明确判定位置编码图案中的局部区域的位置。所述位置可以作为笛卡尔或另一个坐标系中的一对坐标来获取。

当用户在纸 1 上借助用户单元 2 来进行书写的时候，所述用户单元会在用户单元顶端连续记录位置编码图案的局部区域图像。其中每一个图像都解码到一个位置。然后，这种经过解码的位置的序列构成了用户单元 2 在纸上所做移动的数字表示，由此构成了纸上所写内容的数字表示。

关于位置编码图案的解码可以在用户单元 2 中进行，也可以在外单元 3 中进行，此外还可以在某些其他单元中进行，其中位置编码图案的图像是以一种经过了某种程度的处理的形式发送到所述单元的。

如上所述，局部区域中的信息是借助预定解码算法而被解码的。只要对局部区域恰当成像，那么这种处理可以很好地发挥作用。然而在实际情况中，举例来说，由于不良的照明条件，或者用户将笔保持在一个倾斜位置，从而导致合乎比例地成像所述局部区域，或者所述成像机制致使图像失真，因此用户单元经常不能完全正确地成像局部区域。然后会出现解码算法曲解符号或遗漏符号的情况，这将导致不正确地解码所述位置。如果没有在纸上恰当再现位置编码图案，而是在将其施加到纸上的时候就已经发生了某些程度的失真，那么很有可能会出现差错。如果因为种种原因而不能接受一个位置，例如将其认为是极其不可靠的，那么也可能将所述位置视为是不正确的。

在一定条件下，举例来说，如果位置编码图案包含了能够执行检错的冗余信息，或者相对于先前或随后解码的位置而言，所述解码位置并不合理，则可以发现差错。

图 2 示意性地显示了如何解决那些关于错误解码的位置的问题。纸 200 具有一个位置编码图案，其中为了清楚起见，在所述图中并未显示所述图案。第一局部区域 A 对一个位置 P1 进行编码，其中所述位置 P1 已经在未曾检测到任何解码错误的情况下进行了解码并且由此已经作为一个正确解码的位置而被接受。第三局部区域 C 则对一个已作为正确解码的位置而被接受的位置 P3 进行编码。在用户单元成像了局部区域 A 之后并且在其成像局部区域 C 之前，所述用户单元已经成像了具有位置 P2 的局部区域 B。然而由于图像失真，因此将局部区域 B 中的信息解码到一个错误位置 P2'，所述位置对应于虚线表示的局部区域 B'。

在这里执行了一个匹配操作，而不是滤去或排除那些不可用的局部区域 B。所述匹配操作不需要立即执行，而是可以在稍后正确解码了一个或多个位置的时候才执行。在这个实例中，所述解码是在正确解码了局部区域 C 之后才得以执行的。

更为特别的是，从成像局部区域 B 中确定的信息与如何在局部区域 A 与 C 周围的匹配区域 M 的不同局部区域中组成所述位置编码图案的信息进行匹配和比较。执行所述匹配时所在区域的一个实例是由点划线给出的。在这里可以将所述区域选作一个预定区域，也可以动态选择所述区域。在选择匹配区域的时候，可以对一个或多个已经接受的位置加以考虑，其中可以在位置计算失败之前和/或之后记录所述位置的相应局部区域。特别地，在对位置计算失败的局部区域进行记录之前和/或之后，可以确定用户单元的速度和加速度并且将其用作一个评定所述匹配区域的基准。

所述匹配可以在一个或两个维度上进行。根据位置编码图案的组合，在某些情况下可以分别为各个维度执行位置解码。然后则会出现这样一种情况，那就是第一维度的位置计算成功并为这个维度产生一个已接受的位置，而第二维度的位置计算失败并且在第二维度产生了一个明显错误的位置。于是在大多数情况下都只需要在错误的维度上执行所述匹配。

举例来说, 假设关于图 2 中位置 P2 的 x 坐标的计算成功, 并且只有关于 y 坐标的计算是失败的, 那么将会得到一个与局部区域 B”相对应的位置 P2”。在 y 方向上, 由于错误位置远离已接受的位置 P1 和 P3, 因此将会检测到所述错误位置。然后, 成像局部区域 B 中的信息只需要与来自不同局部区域的信息进行匹配, 其中所述局部区域带有一个在范围 M' 中发生变化的 y 坐标。

这样一来, 匹配是在两方面的信息之间执行的, 其中一方面的信息来自一个成像局部区域并且由此并非始终是 100 % 正确的, 而另一方面的信息则是关于不同匹配局部区域中的位置编码图案外部特征的已知信息。在成像局部区域与匹配局部区域之间进行的每次匹配中都会为成像局部区域确定一个对所述匹配局部区域进行成像的概率。在匹配了所述匹配区域中的所有匹配局部区域时, 则选择一个与得到最大匹配概率的匹配局部区域相对应的位置。

毫无疑问, 较为理想的是匹配尽可能快, 由此可以实时以数字方式记录纸上记述的内容并且所述匹配将会需要尽可能少的处理器能力和存储空间。当在用户单元中执行匹配操作时, 所述用户单元通常应该只具有有限的处理器能力和有限的存储空间, 因此后者尤其是合乎需要的。这样一来, 较为理想的是以多种方式来优化所述匹配。

在以上实例中, 错误位置 P2' 和 P2" 都是纸上的位置。然而, 它们也可以是那些不在纸上的位置。

此外还应该指出, 为了清楚起见, 在图 2 中相对一个应用实施例而言, 其中极大地放大了局部区域相对纸的大小。

图 3 显示了局部区域 300 的一个实例, 其中所述区域对一个位置进行编码并且包含 6*6 个符号, 这些符号分别是用点 310 来描述的。为了清楚起见, 在这里并未显示符号的不同的值, 所有符号都是用相同的点来显示的。每个符号可以采用预定数量的不同值, 例如“0”和“1”。所述符号值可以借助不同的参数而被编码, 例如点的大小、点的位置或点的形状。很明显, 也可以使用那些符号数量不同的局部区域。

局部区域 300 与已知内容的局部区域的匹配可以在不同级别上进

行。举例来说，所述匹配可以在符号级别上进行。在这种情况下，从局部区域 300 中确定的每个符号值都可以与已知内容的局部区域中的相应符号值比较，并且可以将正确符号值的数目用作局部区域 300 符合已知内容局部区域的概率指示。作为选择，在对局部区域 300 中的符号的值进行解码的时候，可以确定每个符号采用“0”和“1”之类的各个可能值的概率。对每个匹配的局部区域而言，可以确定局部区域 300 中的符号恰好采用匹配局部区域中的值的组合的组合概率。具有最大组合概率的匹配局部区域的位置将被选为局部区域 300 的解码位置。

作为选择，所述匹配可以在一个“更高”级别上进行。所述符号可以排列在一个群组中，例如排列在列或行矢量中，此外还可以为每个矢量确定一个矢量值。图 3 显示了用 320 标记的一个群组的实例。这样一来，在各个局部区域匹配中可以对群组值或矢量值进行比较，而不是对符号值进行比较，对每个局部区域而言，这种处理将会产生更少的比较。与在符号级别上的匹配相似，在这里还有可能确定每个群组/矢量采用每个可能值的概率，并且为经过匹配的每个局部区域确定组合概率。这意味着只需要在各个局部区域匹配中组合更少的概率。对与如何获取不同匹配局部区域中组成位置编码图案相关的信息来说，由于保存或计算群组值而不是符号值就已足够，因此群组级别上的匹配还意味着对其进行简化。

所述匹配还可以结合那些来自成像局部区域并进行了更高程度处理的信息来进行。举例来说，在匹配中可以对来自两个或更多群组的信息进行比较。

在下文中将参考申请人开发的一个具体的位置编码图案来对所述位置编码进行更详细的例示和描述，其中特别要求经由国际专利申请 PCT/SE02/01243 而对所述图案加以保护，该申请在提交本申请的时候并未公开。在附录 A 中将会参考图 4~9 来对位置编码图案及其解码进行详细描述。但在下文中将会给出关于所述图案的一个扼要和简化的描述。

位置编码基于一个数字序列，所述序列在下文中称为差分数字序

列。这个差分数字序列具有如下特性，如果采用预定长度的任意局部序列，例如具有五个差分数字的局部序列，那么这个局部序列在差分数字序列中始终具有一个无歧义确定的位置。换句话说，它在差分数字序列中只会出现一次，由此可以将其用于位置确定。更具体的说，差分数字序列是沿着位置编码图案的 x 轴和 y 轴“行进”的。

实际位置编码图案由简单的图形符号组成，这些符号可以采用 0~3 这四个不同的值。参见图 6，其中每个符号都包括一个标记，其中举例来说，所述标记可以具有点的形状，并且可以相对于光栅点的标称位置而在四个不同方向中的一个方向上位移一定距离。所述标称位置包括不可视光栅中光栅线之间的一个交叉点。符号值则是由位移方向确定的。每个符号值 0~3 都可以转换成一个用于对 x 坐标进行编码的比特和一个用于对 y 坐标进行编码的比特，也就是转换成比特对 0, 0; 0, 1; 1, 0 和 1, 1。因此，所述编码分别在 x 和 y 方向上进行，而图形编码则是结合一个为 x 和 y 比特所共有的符号来进行的。

每个位置均由 $6*6$ 个符号编码，由此可以转换成一个用于所述位置的 x 坐标的 $6*6$ 比特矩阵和一个用于所述位置的 y 坐标的 $6*6$ 比特矩阵。假设考虑的是 x 比特矩阵，那么可以将这个矩阵划分成各具有 6 比特的 6 个列。一列中的每个比特序列都在一个长为 63 比特的循环主数字序列中构成一个局部序列，该序列具有如下性质：如果选择了长度为 6 比特的局部序列，那么这个序列在主数字序列中具有无歧义确定的位置。由此可以将这六个列转换成与主数字序列中的 6 个位置相对应的 6 个位置数字或序列值。在这六个位置数字之间，可以成对形成五个差分数字，这五个差分数字构成了差分数字序列的一个局部序列并且由此在同一差分数字序列中具有无歧义确定的位置，进而在 x 轴上具有无歧义确定的位置。对某个 x 坐标来说，位置数字的变化取决于 y 坐标。另一方面，由于位置数字始终是根据主数字序列变化的，因此不管 y 坐标是什么，所述差分数字都是相同的，其中在整个位置编码图案中，所述主数字序列将会在列中周期性地重复。

相应地， y 比特矩阵中的六个行在主数字序列中定义了六个位置

数字。这六个位置数字定义了五个差分数字，这些差分数字构成了差分数字序列的一个局部序列并且由此在 y 轴上具有无歧义确定的位置。

在使用位置编码图案的时候，位置编码图案中不同部分的图像是如上记录的。通常，图像包含的符号明显多于 6×6 个。事实上，解码通常是以图像中最好的 8×8 个符号为基础的。其中尤其将附加符号用于检错和/或纠错，在附录 A 和 B 中对此进行了更详细的描述。

由于位置编码图案并没有始终在表面上得到完美再现，并且由于用户单元有时会在成像过程中导致位置编码图案失真，因此要相当可靠地确定点的位移将是非常困难的，由此很难确定符号的值。因此在一个实际的实施例中改为确定每个符号采用四个不同的可能数值的概率。由此将成像的局部区域解码到一个矩阵中，所述矩阵具有矩阵中各个矩阵元素的四个概率值，其中每个矩阵元素对应于一个符号。这个矩阵转而可以分成两个矩阵，一个对应于 x 坐标，另一个对应于 y 坐标。其中每一个矩阵中都包含了矩阵中每个矩阵元素的两个概率值。其中一个概率值对应于矩阵元素具有值“1”的概率，另一个则对应于矩阵元素具有值“0”的概率。对于位置编码图案中各个图像的处理由此导致产生了两个概率值矩阵 P_x 和 P_y 。参考图 10~19，在附录 B 中更详细地描述了如何确定这些矩阵。其中特别要求在国际专利申请 PCT/SE02/01246 中保护附录 B 的内容，在提交当前申请的时候，所述申请并未公开。

下文中参考图 20 中的流程图来对基于上述 Anoto 位置编码图案的位置解码进行描述。优选地，所述位置解码借助一个处理器和适当的程序代码来执行，其中所述处理器和程序代码可以存在于图 1 的用户单元 2 或外部单元 3 中，也可以存在于某些其他单元中。

针对程序代码的输入信号包含了以上两个概率值矩阵 P_x 和 P_y （与图 18 的矩阵 34 和 35 相对应），由此所述矩阵分别涉及的是 x 坐标和 y 坐标。根据这些概率矩阵，在步骤 1800 中，对一个 x 位置和一个 y 位置进行计算。在附录 A 和 B 中更详细地描述了如何执行这个

计算。简要的说,就是为矩阵 P_x 和 P_y 中的每一个矩阵确定各行和各列中的比特序列在主数字序列中具有的最有可能的位置。根据由此确定的位置或位置数字(在附录 B 中也称为“序列值”),分别为 x 和 y 方向确定差分数值。所述差分数值分别为 x 方向和 y 方向形成了差分数值序列中的一个局部序列。在差分数值序列中,每个局部序列都具有一个无歧义确定的位置。所述位置可以用附录 A 中描述的方式来确定。 x 数字序列的位置确定了 x 坐标,而 y 数字序列的位置则确定了 y 坐标。

所述坐标计算可以产生三种情况。其中一种是这两个坐标的计算取得成功,即 1801,另一种则是其中一个坐标的计算取得成功,但是其他坐标的计算失败,也就是 1802,还有一种情况是所有坐标的计算都失败,即 1803。

如果在所述坐标计算过程中没有检测到差错,则认为所述计算取得成功,并且在步骤 1804 中将其保存为一个接受的位置。然后如果没有对所述位置进行其他检查,则滤去所有与成像局部区域有关的其他信息。

如果针对某个坐标的计算失败,则保存这个坐标的信息,以便在匹配中加以使用,同时在步骤 1805,如果没有对所述位置进行其他检查,则保存那些成功计算的坐标的信息并且滤去相关信息。

如果关于这两个坐标的计算都是失败的,那么毫无疑问,在步骤 1806 中将会保存这两个坐标的信息。

保存的信息可以是位置计算所基于的全部信息,即一个或多个概率值矩阵 P_x 和/或 P_y ,此外所述信息也可以是或多或少进行过处理的形式的信息。举例来说,可以计算和保存各个列采用每个可能位置数字的概率值。举例来说,作为选择,也可以只保存位置数字最可能的组合,或者将其与相关的概率保存在一起。此外还可以保存上述信息的不同组合或子集。

所述匹配可以立即进行,也可以在稍后对更多位置进行了解码的时候进行。

现在假设针对 x 的计算取得了成功,但是 y 坐标的解码则是失败的。 x 坐标是在具有用于 y 坐标矩阵中的行定义的所有可能位置数字的概率值的时候得到保存的。

在执行匹配的时候,首先在步骤 1807 中确定将要匹配的区域。在这种情况下, x 坐标是已知的,因此所述匹配只需要沿着 y 轴进行。沿着 y 轴的范围可以根据先前与随后接受的 y 坐标来加以确定。

然后执行的是实际匹配。由于 x 坐标是已知的,因此对匹配区域中的每个 y 坐标来说, y 坐标对应的位置数字是已知的。在步骤 1808,从存储器中读取为这些位置数字保存的概率,并且将组合概率作为与所论述位置数字的概率产物来加以计算。在为匹配区域中的所有 y 坐标计算了组合概率的时候,在步骤 1809 中将会选择得到最大组合概率的 y 坐标。在步骤 1810 中,将先前保存的 x 坐标和选定的 y 坐标作为一个已接受的位置来加以保存,如果不对所述位置进行其他检查,则滤去所有那些为匹配保存的信息。

在一个替换实施例中,只有最有可能用于每一行的位置数字才会得到保存。在这种情况下将会比较用于各个 y 坐标的相应位置数字的数目。在这里将会选择为最大数量的相应位置数字所获取的 y 坐标。这个实施例的优点是保存的信息量将会更小。另一方面,就 y 方向上的位置而言,这将产生一个成像局部区域符合匹配局部区域的更粗略的概率量度。

而在另一个替换实施例中,除了最有可能的位置数字之外,还保存了关于这些位置的概率。然后可以基于相应的位置数字、这些位置的组合概率以及非对应位置数字的组合概率来选择最可能的 y 坐标。

在步骤 1803,举例来说,如果针对这两个坐标的位置计算全都失败,则保存 y 坐标矩阵 P_y 中每个行的最可能位置数字以及 x 坐标矩阵 P_x 中每个列的最可能位置数字。

所述匹配每次只为一个坐标执行。假设所述匹配始于 x 坐标。在步骤 1811,首先将匹配区域确定成一个沿着 x 轴的范围。所述区域的大小可以根据如上所述的相同参数来加以确定。随后在步骤 1812 中执

行实际匹配。

由于 y 坐标在这个实例中并非已知，因此不同的 x 坐标对应的位置数字也不是已知的，只有差分数字才是已知的。在这里，一种可能的匹配策略是计算存储位置数字之间的差分并且将这些差分与匹配区域中不同 x 坐标的差分相匹配。然而由于错误的位置数字会导致两个错误的差分，因此这种策略并不是一种最佳策略。

另一种可能的解决方案是测试位置数字的所有可能组合，其中所述组合为匹配的 x 坐标产生了差分数字。对每一个匹配差分数字的位置数字组合来说，所述位置数字的概率值将被选择并且相乘。此外还确定了取得最高概率值的 x 坐标。这种解决方案能够很好的运作，但是需要大量计算，由此花费了很长时间。此外，它还需要存取所有位置数字的概率值。

为了优化匹配，所述匹配可以改为借助一种算法来执行，所述算法根据所匹配的 x 坐标的差分数字来计算对应于至少一个后续位置数字的位置数字的数目，其中将位置数字视为是从右到左。

假设已经从成像局部区域中确定了 8 个可以访问的位置 P_0 到 P_7 。这些位置与七个差分数字 $s_0 \sim s_6$ 相匹配，其中所述差分数字构成了差分数字的局部序列，所述局部序列对与来自成像局部区域的信息相匹配的 x 坐标进行编码。所述差分数字则可以通过组合已知的位置编码图案来加以确定。

因此，所述算法是如下运作：

Numbermatch = 0

For k=6 down to 0

For i=0 down to 6 -k

If $p_{k+i+1}-p_k=\text{sum}(j=k:k+i)$ (s_j)

Numbermatch=Numbermatch+1;

Interrupt innermost loop;

End

End

End

在执行了算法的时候，变量 **Numbermatch** 包含了位置数字的数目，依照差分数字局部序列中的差分，这些数字对应于至少一个后续的位置数字。

在执行了所有匹配的时候将会选择 **Numbermatch** 最大的 **x** 坐标。所述坐标是作为一个成功坐标来保存的。此外，现在很容易确定 **x** 坐标的位置数字，以便在以下将要论述的可能进行的后续验证步骤中使用。

现在借助实例来描述以上算法。假设已经成像了一个局部区域，所述局部区域包括为 **x** 坐标产生位置数字的后续局部序列 **P** 的符号。这些位置数字定义了差分数字的一个局部序列 **D**。进一步假设没有正确成像局部区域并且已经根据来自成像局部区域的信息而在位置计算中得到了错误的局部序列 **P***。最终假设主数字序列长为 63 比特，这意味着位置数字之间的差分是 63 为模来进行计算的，其中所述位置数字表示主数字序列中的位置。

P= 23 12 54 43 7 18 11 35
D= 52 42 52 27 11 56 24

P*= 23 55 54 43 7 42 11 35

这样一来，错误解码的是第一个（55，也就是 **P*** 中左起第二位置数字）和第五个（42，也就是 **P*** 中左起第六个位置数字）位置数字。在这个实例中，上述算法是如下运作的：

Numbermatch=0

P6:

(35-11) mod63=24. P6 matched . Numbermatch=1

P5:

(11-42) mod63=32 (should be 56) No match

**(35-42) mod63=56 (should have been 17=
(56+24) mod63. No match. P5 not matched.**

P4:

**(42-7) mod63=35 (should have been 11) No match.
(11-7) mod63 = 4 =(11+56) mod63. P4 matched.
Numbermatch=2**

P3:

(7-43) mod63=27. P3 matched. Numbermatch=3

P2:

(43-54) mod63=52. P2 matched. Numbermatch=4

P1:

**(54-55) mod36=62 (should have been 42) No match.
(43-55) mod63=51 (should have been 31=(42+52) mod63.
No match.
(7-55) mod63=15(should have been
58=(42+52+27) mod63. No match.
(42-55) mod63=50(should have been
6=(42+52+27+11) mod63.No match.
(11-55) mod63=19(should have been
62=(42+52+27+11+56) mod63.No match.
(35-55) mod63=43(should have been 23)=
(42+52+27+11+56+24) mod63.No match.
P1 not matched.**

P0:

(55-23) mod63=32(should have been 52).No match.

(54-23) mod63=31=(52+42) mod63.P0 matched.

Numbermatch=5

所述算法由此得出一个事实，那就是根据差分数字局部序列中的差分，有五个位置数字与至少一个后续位置数字相一致。相应位置数字的数目则用作了匹配概率的一个量度。

如果关于两个位置坐标的位置计算全都失败，那么毫无疑问，上述匹配 x 坐标的实例同样能在所述计算始于 y 坐标而不是 x 坐标的时候发挥作用。

在通过匹配确定了第一坐标的时候，可以使用与如上所述只有一个坐标的位置计算失败的情况相同的方法来确定第二坐标，也可以使用与用于两个失败计算中的第一坐标的情况相同的方式来确定第二坐标。

上述算法可以通过同时评估若干种匹配以及更有效处理差分数字之和来进行优化。因此，所述算法如下所示，其中关于每个匹配的 ss_i 都等于 $\text{sum}(s_0, \dots, s_i)$ 。

对各个匹配来说

For k=0 to 6

For i=0 to 6-k

dp= $P_{k+i+1}-P_k$

If dp= ss_i

Numbermatch_k=Numbermatch_k+1

Interrupt innermost loops;

End

End

End

Check whether Numbermatch₀ is the greatest so far, in that case save value of Numbermatch₀ and he corresponding x ccordinate.

For j=0 to 5

Numbermatch_j=Numbermatch_{j+1}

End

Numbermatch₆=0

d=the next preceding difference number in the difference number sequence.

For j=6 to 1

ss_j=ss_{j-1}+d

End

ss₀=d

End

非常有利的是，所述匹配可以用一个验证步骤即步骤 1815 来终止，该步骤在图 20 中由虚线表示。实际上，由于匹配算法总是发现一个与先前已接受位置接近的位置，因此将会存在增加差错的风险。这意味着不可能通过查看从已接受位置开始的距离来评定所匹配的位置是否正确。所述验证步骤可以包括借助成像局部区域中可用的全部信息来为获取的位置计算概率。通常在这个步骤中包含了比位置计算中使用的 8*8 个符号多出很多的符号。举例来说，所述操作可以包括 16*16 个符号。对其中每个符号来说，在这里已经为符号可以采用的每个值都确定了一个概率值（在附录 B 中也将其称为值概率）。当借助最好 8*8 区域确定一个坐标对时，这些 8*8 符号的值是已知的，因此，分别处于相应的 8*8 比特的 x 和 y 比特矩阵中的列和行的比特也是已知的。由于列和行组成了二进制循环主数字序列的局部序列，因此在知道差分数字序列的情况下确定的是如何在更大的 16*16 区域中延伸列和行。然后则可以借助 16*16 区域中的比特的概率值来计算整个 16*16 区域的概率度量并且将其与一个门限值进行比较，以便确定对于最终要接受的解码位置而言，所述 16*16 区域的总概率是否足够大。

在图 21 的框图中概述了整个方法。首先执行一个位置计算 1901。

如果这个计算成功，则位置继续进行验证 1902。如果所述验证成功，则接受所述位置。如果位置计算失败，则执行一个匹配 1903。即使对一个位置计算成功的位置来说，如果验证失败，那么也将会执行这个处理。而所述匹配则始终产生一个位置。这个处理进行到验证 1902。如果为通过匹配获取的位置所进行的验证取得成功，则接受这个位置，否则位置解码失败，并且不为成像的局部区域记录位置。在位置计算取得成功的特定范例中，如果随后立即进行的验证与匹配之后的验证一样是失败的，但是计算得到位置与所匹配的位置是相等的，那么仍然可以接受所述位置。

为了进行位置解码，可以使用不同类型的设备来读取和解码位置编码图案。这种设备可以具有一个用于产生位置编码图案的数字表示的传感器和一个用于在数字表示中识别位置编码图案并且解码所述图案的信号处理单元。所述信号处理单元可以包括一个带有存储器和适当程序的处理器，也可以包括特定的硬件，此外还可以包括数字和/或模拟电路或者这些设备的适当组合。

所述传感器可以是适合对位置编码图案进行成像的任何类型的传感器，因此所述标记的一个图像是以黑白、灰度或彩色形式获取的。所述传感器可以是固态的单芯片或多芯片设备，它对任何适当波长范围中的电磁辐射都很敏感。例如，所述传感器可以包括一个 CCD 部件（电荷耦合器件）、CMOS 部件（互补金属氧化物半导体）或 CID 部件（电荷注入器件）。作为选择，所述传感器可以包括一个用于检测所述标记的一个磁性性质的磁传感器阵列。更进一步，在这里可以对所述传感器进行设计，以便形成所述标记的任何一种化学、声学、电容或电感性质的图像。

在图 22 中显示了用于位置解码的设备的一个实例。所述设备可以构成图 1 中的用户单元。它包括一个与笔具有大致相同的形状的外壳 2011。在外壳短的一端具有开口 2012。短的一端紧靠表面或者与之保持一个很短距离，其中位置测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。

光学部分包括一个用于照射将要成像的表面的发光二极管 2013, 并且包括一个用于记录二维图像的感光区域传感器 2014, 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。

设备的电源是从电池 2015 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部电源(未显示)获取电源。

电子线路部分包括一个处理器单元 2016, 所述处理器单元具有经过编程而读取来自传感器的图像并且根据这些图像执行位置计算、匹配和验证的处理器, 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 2017, 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 2017 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 2018, 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 2019, 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 2020。

所述设备可以在不同的物理外壳之间分开, 其中用于捕获位置编码图案的图像以及将上述信息发送到第二外壳所需的传感器和其它组件位于第一外壳中, 而用于执行位置解码所需的信号处理单元和其它组件则位于第二外壳中。

附录 A

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循环主数字序列来进行编码，所述循环主数字序列具有在循环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，如果从循环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。由此在下文中采用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。

位置编码基于使用循环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 4 示意性显示了一张具有 7 个列 $x \sim x+6$ 的纸 A100。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 AB 做出示意性标记。另外在图 4 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，下文将其称为基本数字序列或基本差分数字序列，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 4 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或辅助数字序列，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排列局部序列及其位置，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数字序列中具有唯一位置，那么，辅助差分数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数字序列而言，一个差分数数字序列长度不会与第二差分数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 是辅助差分数数字序列 1 的长度， l_2 是辅助差分数数字序列 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 5 示意性地描述了这个操作。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数字的长序列中行进，在这里，所述差分数数字由 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数字序列的局部序列而显示的基本差分数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主数字序列，所述行即为图 4 的 x 方向。差分数数字是在相邻的行之间定义的，这些差分数数字形成一个基本差分数数字序列，该序列可以由辅助差分数数字序列构造。在第二维度中也可以使用另一个主数字序列、另一个基本差分数数字序列以及其他辅助差分数数字序列。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 6a ~ d 中显示的方法来对这些不同组合进行图形编码，其中点 A30 在四个方向中的任何一个方向上从光栅中的标称点 A31 开始移动，各个标称点对应于光栅中的线条 A32 之间的交点。WO01/26033 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 7 示意性地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 7 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 7 中标记的。然而，通常并不将所述光栅打印出来。

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循环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, 我们为循环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 8 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 度，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。在那些误差坐标范围很大的地方，连续误差要比其他误差更加难以检测。

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。更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，已经使用以下方法选择了主数字序列和辅助差分数字序列。

主数字序列 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, 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

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。

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。

以下描述引用了图 9 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 WO01/26033 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 0~3。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 9 中是通过并行流程对此加以指示的。

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，在图 9 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部序列。更具体地说，如下所示来改写各个差分数字：

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

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：

$$P = p_1 \pmod{l_1}$$

$$P = p_2 \pmod{l_2}$$

.

.

$$P = p_m \pmod{l_m}$$

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 中详细描述实例中那样，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 $L = \prod_{i=1, m} l_i$ 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 ：

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

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， $xs = 0$ 。 $s(x, 0)$ 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 xs 。

根据图 9 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 ys 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 xs ，并且在 $y = 0$ 的条件下将 y （与主数字序列的长度相模）从 xs 中减去以及在 $x=0$ 的条件下将 x 从 ys （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 ys 。

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_i: 它将 n 个辅助差分数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 9 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。

附录 B

以下参考图 10 中的流程图来对一个如何借助概率计算来解码一个编码图案的实例进行说明。将要解码的编码图案具有在申请人的 WO 01/26033 中描述的类型。

图 11a 显示了一张有一个表面 16 的纸 15, 其中表面 16 上具有一个处于光可读位置编码形式 17 的编码图案下方的位置编码图案。所述位置码包括标记 18, 并且为了清楚起见而被极大地加以放大。图 11b 显示了图 11a 中的位置码 17 的一个进一步放大的部分 19。设备被设计为记录位置码的一个局部区域的一个图像 (步骤 A), 标识图像中的多个标记 18 (步骤 B), 以及用一个光栅形式的参考系统去适配图像 (步骤 C), 其中光栅具有在光栅点 22 相交的光栅线 21。所述适配以这样一种方式实现: 每个标记 18 与一个光栅点 22 相关。例如, 标记 23 与光栅点 24 相关。因此通过光栅适配可以确定每个标记属于哪个光栅点。在本例中, 光栅是方格的形式, 但也可能是其他形式。此处结合其作为参考的申请人的申请 WO 01/75783、WO 01/26034 和 SE 0104088-0, 更详细揭示了光栅对图像中的标记的适配。

在“理想”编码图案中有且仅有一个标记与每个光栅点相关。由于编码图案的图像中的缺陷和不足, 因此有可能很难在编码图案的一个图像中确定哪些标记属于该编码图案以及多个标记中的哪一个是与某个特定光栅点相关的那个。由于此原因, 在本例中可能在解码数据时将多个标记与同一个光栅点相关联。与一个光栅点光联的标记一起形成属于该光栅点的一个标记元素。

在本例的编码图案中, 标记 18 的值由其相对于其相关的光栅点 22 的位移定义。更具体地, 定义标记值的是一个标记相对于一个光栅点的一个点位置。此点通常是标记的主要点。在本例的编码图案中, 每个标记有四个理想位置。这些位置位于四条光栅线 21 中的每一条上, 这四条光栅线从与该标记相关的光栅点 22 延伸出来。这些位置位于与光栅点相等的距离处。图 12a-d 中放大显示了一个标记的理想位置 25。他们在图 12a 中的值为“0”, 在图 12b 中的值为“1”, 在图 12c

中的值为“2”，在图 12d 中的值为“3”。这样每个标记代表四个不同的值“0-3”。

由于多种原因，在记录的图像中标识的标记通常不具有理想位置。因此在许多情况下，可能难以明确地确定一个标记的值。由于此原因，设备被设计为为每个标识的标记计算该标记定义“0-3”中每个值的有关的价值概率（步骤 D）。每个标记的值概率是标记 27 到其每个理想位置 25 的距离 26 的减函数，参见图 13，或者更具体地，通常是从标记 27 的主要点到每个理想位置的距离的减函数。标记可被假定为在理想位置周围正态分布。这意味着值概率 $P(d_i)$ 可由公式 $P(d_i) = k \exp(-(d_i)^2/v)$ ，其中 k =常数， d_i =从标记到理想位置的距离，以及 v =常数，在本例中为距离的方差。方差通过经验确定。从而可对每个标记计算四个值概率。有时会出现标记在光栅点中央的情况。在这种情况下，标记的四个值概率将是相等的，因为标记到每个理想位置的距离是一样大的。没有标记与某个光栅点相关的情况，即相关的元素包含零个标记的情况，被看作标记位于光栅点中央的情况，即值概率是相等的。

如果有多个标记与一个光栅点相关，例如三个，即在相关的标记元素中有三个标记，则光栅点或标记元素共有 $3*4$ 个值概率。由此将设备设计成为每个光栅点或标记元素以及每个值确定一个与光栅点相关的多个标记一起定义此值的值概率（步骤 E）。从而这些为光栅点确定的值概率也可被称为光栅点概率。通过计算光栅点的值概率，可在位置确定中考虑记录图像中的所有标记，并可最小化丢失信息的危险。由于上述装置对于每个标记元素，计算该标记元素定义每个值的值概率，因此在以下说明中值“0”-“3”被称为标记元素值。

一个标记元素的值概率可通过被比较的标记元素中的标记的值概率确定，为每个可能的标记元素值选择最高的值概率。作为替换，标记元素的值概率可为标记元素中的标记的每个可能标记元素值的值概率加权和。当然标记元素的值概率也可以通过与上述方法不同的方法确定。

在本例中计算一个标记元素的值概率是通过图 14 中的数字例子说明的。图 14a 显示了一个具有两个相关的标记 28 和 29 的光栅点 22。标记 28 和 29 一起组成了属于光栅点 22 的标记元素。图 14b 中的表 30 和 31 包含各标记的可能值的值概率 P_1 。图 14c 中的表 32 包含了得出的标记元素的可能标记元素值的值概率 P_2 。在本例中，标记元素的值概率是相对的。作为替换，它们也可以以一种适当的方式被归一化。如果只有一个标记相关到一个光栅点，即相关的标记元素中只有一个标记，则显然标记和标记元素的值概率是相等的。

当记录一个图像时，设备到表面的距离影响被记录的位置编码的部分的大小，从而也影响能适配到图像中的光栅的大小。为了将图像转化为一个位置，使用预定数目的标记元素，本例中为 $8*8$ 个标记元素。如果多于 $8*8$ 个光栅点适配到了图像中，则标识额外的标记元素。因此设备被进一步设计为从所有标识的标记元素中选择提供最多关于表面上的位置的信息的标记元素集合（步骤 F）。此标记元素集合是（但并不必须是）连续的。在本例中，标记元素集合中的标记元素对应于一个光栅点矩阵，该光栅点矩阵具有适配到图像的光栅点，但这不是必要条件。换句话说，目的是选择多个标记元素，每个标记元素值具有相关值概率，其能够最大化记录图像的信息量度。为了此目的，对于每个标识的标记元素计算一个熵。然后选择给出最小的熵和的 $8*8$ 个标记元素，其中最小的熵对应了记录的图像的最大信息量度。如果标记元素的值概率被归一化以使得

$$\sum_i P_{2,i} = 1,$$

则标记元素的熵 H 根据以下公式计算：

$$H = -\sum_i P_{2,i} \log_2(P_{2,i})$$

其中 $P_{2,i}$ 是标记元素值 $i(i=0,1,2,3)$ 的标记元素值概率， $\log_2$ 是 2 为底的对数。这样一个标记元素的熵在其值概率相等的时候最大，而在除一个值概率外其他值概率均为零时最小。另一个通过熵计算来选择 $8*8$ 个标记元素的替换方法是用每个标记元素的最高值概率作为一个信息值。在这种情况下，选择连续的 $8*8$ 个标记元素，其最大化由

这 8×8 个标记元素的信息值的和组成的信息量度。

正如已提到的，本例中使用的编码图案对表面 4 上的一个点的两个坐标进行编码。这些坐标可分别被解码。因此，它们可被称为两个维度中的数据。更具体地，编码图案中的每个标记编码一个被用于解码第一坐标的第一比特以及一个被用于解码第二坐标的第二比特。

因此在对记录的图像中的编码图案进行解码时，一个标记元素的每个可能的标记元素值“0”-“3”被转换成一个第一和一个第二解码值，在本例中它们是二进制的。从而设备被设计为对于标记元素集合中的 8×8 个标记元素中的每一个标记元素将标记元素值“0”-“3”转化为四个不同的比特组合 (0,1) (0,0) (1,0) 和 (1,1)。对于每个标记元素，比特组合具有属于每个标记元素值的值概率，参见图 15a 的表 33 中所示的前述数字实施例的后续。在比特组合中，第一比特，即第一解码值，表示第一维度，第二比特，即第二解码值，表示第二维度。相应的标记元素值的值概率 P_2 与第一和第二解码值相关。这样标记元素集合可被用于创建第一维度的具有相关的值概率的第一解码值的第一集合，以及第二维度的具有相关的值概率的第二解码值的第二集合（步骤 G）。表 33 说明了标记元素集合中的一个标记元素。表 33' 和 33'' 分别包含了具有相关的值概率的第一集合中的相应的第一解码值，以及具有相关的值概率的第二集合中的第二解码值。显然从上述说明中可知，第一和第二解码值中的每一个值为 0 或 1。

设备被设计为对于标记元素集合中的每个标记元素，将第一集合中的每一个不同的可能第一解码值中与一个值概率相关，并将第二集合中的每一个不同的可能第二解码值与一个值概率相关。由于本例中可能的第一和第二解码值为 0 和 1，因此上述操作在第一和第二集合中对标记元素集合中的每个标记元素得到了解码值 1 的一个值概率和解码值 0 的一个值概率。以下解码值 0 的值概率称为 0 概率，而解码值 1 的值概率称为 1 概率。

参见表 33'，在本例中，通过比较第一集合中的对应于第一解码值为 0 的值概率来对标记元素集合中的每个标记元素实现上述操作。

然后最高的值概率被选为 0 概率并保存在一个第一矩阵 34 中。以相同的方法比较第一集合中对应于第一解码值为 1 的值概率。然后最高的值概率被选为 1 概率并保存在第一矩阵 34 中。参见表 33”，随后对于第二集合中的值概率、第二解码值和一个第二矩阵 35 重复上述程序。从而第一和第二集合被用于创建一个具有 0 和 1 概率的第一和第二矩阵（步骤 H）。在图 15b 中的数字例子后续说明了该结果。作为替换，通过将第一集合中对应于第一解码值为 0 的值概率相加，所得的和被存储为 0 概率，以及通过将第一集合中对应于第一数字为 1 的值概率相加，所得的和被存为 1 概率，来创建对于元素集合中的 8×8 个元素的具有 0 和 1 概率的一个第一和一个第二矩阵。然后对于第二集合中的值概率和第二解码值重复此程序。

这样标记元素集合中的 8×8 个标记元素现在对应于两个矩阵 34 和 35，每个具有 8×8 个矩阵元素，其中每个矩阵元素包括一个 0 概率和一个 1 概率。通过这些第一和第二矩阵，可确定位置的坐标。

在确定记录的图像中所有标记元素的值概率后选择标记元素集合的一个替换方法是，等到已为所有被标识的标记元素确定对应于矩阵 34 和 35 的矩阵。在这种情况下，则可根据相应的 0 和 1 概率选择每个矩阵中的 8×8 个矩阵元素。执行此操作的一个方法是选择 8×8 个矩阵元素，其中 0 和 1 概率中的一个为高而另一个为低。在这种情况下，不必选择对应于相同标记元素的矩阵元素用于两个坐标的确定，计算通过两个矩阵的不同的相应标记元素进行。

在本例中位置码在基于一个第一循环主数字序列的第一维度中。这给出了标记元素的标记元素值之间的关系的条件。第一循环主数字序列具有这样的性质：其中每个预定长度的局部序列的位置是明确地确定的。在本例中预定长度是 6。如果在第一循环主数字序列中的任意位置取这样 6 个顺序的数字，则这六个数字只在第一主数字序列中以此顺序出现一次。如果第一主数字序列的末尾连续到第一主数字序列的开始，该性质仍适用。因此，第一主数字序列被称为循环的。在本例中使用一个二进制主数字序列。如果要明确确定一个具有六个数

字的局部序列的位置，则第一主数字序列最长可具有 $2^6=64$ 的长度，长度为 6 的局部序列可占有第一主数字序列中的位置 0-63。但是如果选择一个长度为 63 的第一主数字序列，则可提供改进的纠错性质，这可从下面的说明中看到。从而在下面的说明中将假定第一主数字序列的长度为 63，并从而定义了 0-62 范围中的唯一位置。

图 16 显示了一个可结合位置码使用的第一循环主数字序列的例子。例如，局部序列 0,0,0,0,0,0 具有第一主数字序列中的明确位置 0，局部序列 1,1,1,1,1,0 具有明确位置 9 而局部序列 1,1,1,0,1,0 具有明确位置 11。为确定表面上的一个位置，必须在记录的图像中标识 $6*6$ 个标记元素。但是，如上文所讨论的，位置确定采用的是 $8*8$ 个标记元素，其原因将从下面的说明中看出。如上文所述，第一维度中的位置码所基于的第一循环主数字序列具有这样的性质：它只包含相互不同的长度为 6 的局部序列。因此，每个长度为 8 的局部序列在第一循环主数字序列中的位置是明确确定的。在确定表面上的位置的坐标时利用了此事实。

设备被设计为将第一循环主数字序列中每个长度为 8 的唯一局部序列与第一矩阵 34 中的每列匹配（步骤 I）。该方法在图 17 中说明。该图显示了一个长度为 8 的二进制局部序列 36 以及第一矩阵 34 中的一列 37（图 15b），所述列具有矩阵元素，每个元素包含一个 0 概率和一个 1 概率，分别对应于第一解码值为 0 和为 1 的概率。对于每个矩阵元素，根据局部序列 36 中相应的数字选择 0 和 1 概率之一。例如，局部序列 36 中的第一个数字为 0，意味着选择 0 概率作为列 37 中的第一个矩阵元素。局部序列中的第二个数字为 1，意味着选择 1 概率作为列 37 中的第二个矩阵元素。对于第一主数字序列中的每个局部序列，对于第一矩阵 34 中的每一列，设备还被设计为通过将相应选择的矩阵元素的 0 和 1 概率相乘计算一个第一序列概率（步骤 J）。在图 17 中，已计算了对应于局部序列 36 和列 37 的第一序列概率 38。在进行此操作后，将有 63 个第一序列概率，它们对于第一矩阵 34 中的每一列具有一个相关的唯一的序列值。这些序列值由相应的局部序列在

第一循环主数字序列中的位置定义。设备被设计为对每一列选择最高的第一序列概率以及相应的序列值，并保存它们。

此处第二维度中的位置码是基于一个第二循环主数字序列的，在本例中它与第一循环主数字序列具有相同的性质。

设备被进一步设计为以相应于上述方式的方式，将第二循环主数字序列中每个长度为 8 的唯一局部序列与第二矩阵 35 中的每行匹配。正如矩阵 34 中的列一样，矩阵 35 中的行具有矩阵元素，每个元素包含一个 0 概率和一个 1 概率，分别对应于第二解码值为 0 和为 1 的概率。对于每个矩阵元素，根据第二循环主数字序列中的局部序列中相应的数字选择 0 和 1 概率之一。对于第二主数字序列中的每个局部序列，对于第二矩阵 35 中的每一行，设备还被设计为通过将相应选择的矩阵元素的 0 和 1 概率相乘计算一个第二序列概率（步骤 J）。在进行此操作后，将有 63 个第二序列概率，它们对于第二矩阵 35 中的每一列具有一个相关的唯一的序列值。这些序列值由相应的局部序列在第二循环主数字序列中的位置定义。设备被设计为对每一行选择最高的第二序列概率以及相应的序列值，并保存它们。

本例中使用的位置码是基于循环主数字序列的不同旋转或循环移动的使用。例如，为编码 x 方向的位置，第一主数字序列在表面上按列向不同的方向旋转或循环移动以某种其他的方式印刷或排列，即，在与要编码位置的方向正交的 y 方向，从上往下。主数字序列可在同一列中重复印刷，如果要在 y 方向编码多于对应于主数字序列的长度的位置则这一点是必要的。然后在所有重复中使用相同的主数字序列旋转。这意味着在不同的列中可使用不同的旋转。

每对相邻的列定义一个差分数字 D。差分数字 D 由每列中第一局部序列的在主数字序列中的位置之间的差分给出。如果局部序列的位置之间的差分在列中往下一步，则结果将是相同的，因为位置将以相同的方式偏移。从而差分数字 D 始终是相同的，不论在列中的哪个“高度”比较主数字序列中的局部序列的位置。从而对于每对列，差分数字 D 在 y 方向是恒定的。相邻列之间的差分数字形成一个差分数字集合，

它可用于获取表面上一个位置在第一维中的坐标。

第二方向的位置码，例如在此情况下为 y 方向，可基于与第一维度中的位置码相同的原理。然后第二主数字序列被排列为在表面上的行中不同的循环移动，即，在 x 方向，从左往右。差分数字在相邻行之间定义，且这些差分数字形成一个差分数字集合，它可用于获取表面上的一个位置在第二维度的坐标。

从而位置码由第一方向的一个局部位置编码和第二方向的一个局部位置编码组成。

从上文可见，局部序列不是以其显值写入的，而是以其图形编码写入的。在图形编码中，标记定义了局部位置编码图像的叠加。

由于位置码是基于在表面上以预定方向排列的主数字序列的，因此标记必须在这些方向上被解码，以便正确确定位置。如上文所提到的，正确的解码方向是从上往下和从左往右的。

用于数据解码的设备在记录图像时可以相对于表面和位置编码的不同位置旋转。如图 18 中的箭头 40 所示，有四个可能的记录旋转。位置码的记录图像本身不显示位置码和设备之间的相对旋转，因为如果位置码旋转 0, 90, 180 或 270 度其外观实质上是一样的。但是当位置码被旋转时，每个标记相对于其相关的光栅点的位移的方向将会改变。这导致了编码标记的位移的比特组合（第一解码值，第二解码值）改变。通过位置码的“正确”旋转，标记以正确的解码方向在列中从上往下以及在行中从左往右排列。如果位置的正确旋转为 0，则以下内容对于错误的旋转成立：

- 顺时针 90 度：具有“正确”旋转的标记的列，即标记从上往下排列的列，将成为标记从右往左排列的行，即为错误的解码方向，而具有“正确”旋转的行，即标记从左往右排列的行，将成为标记从上往下排列的列，即为正确的解码方向。

- 180 度，具有“正确”旋转的标记的列，将成为标记从下往上排列的列，即为错误的解码方向，而具有“正确”旋转的标记行，将成为标记从右往左排列的行，即为错误的解码方向。

·顺时针 270 度：具有“正确”旋转的标记的列，将成为标记从左往右排列的行，即为正确的解码方向，而具有“正确”旋转的标记的行，将成为标记从下往上排列的列，即为错误的解码方向。

如果列和行中的标记以错误的解码方向排列，则解码时每个元素的 0 和 1 概率将被颠倒。

因此如下文将要说明的，设备被设计为测试记录的图像中的位置码的局部区域的不同旋转。在旋转 180 度并“倒置”的第一和第二矩阵 34 和 35 上（在图 15c 中分别标示为 34' 和 35'）还执行分别在第一和第二矩阵 34 和 35 上执行的操作，即将循环主数字序列中的局部序列分别与矩阵中的列和行进行匹配（步骤 I），计算序列概率（步骤 J），以及分别选择具有相应的列和行的序列值的最高序列概率。这些旋转的、倒置的矩阵 34' 和 35' 对应记录的图像中的位置码的颠倒的局部区域。在显示记录的图像中的一个位置码的一个局部区域的一个例子的图 11 中解释了其原因。在该图中，为了简便只使用了 9 个标记，每个与一个光栅点相关以方便说明。位置码 45 是记录在图像中的那个。位置码 45' 是同一个颠倒的位置码。矩阵 46 和 47 分别对应于向右旋转的位置码 45 的矩阵 34 和 35，而矩阵 48 和 49 分别对应于倒置的位置码 45' 的矩阵 34' 和 35'。如果倒置的位置码的矩阵 48 和 49 旋转 180 度并倒置，则可获得向右旋转的位置码的矩阵 46 和 47。这里倒置表示每个矩阵元素中的 0 和 1 概率改变位置。

经过上述程序后，对于矩阵 34 和 34' 中的每一列和对于矩阵 35 和 35' 中的每一行有一个具有相应的序列值的序列概率。然后对于矩阵 34, 34', 35 和 35'，设备被设计为通过将相应的最高序列概率相乘计算一个旋转概率（步骤 K）。根据对应于矩阵 34 和 34'（对应于最高旋转概率）的最高序列概率的序列值，以及对应于矩阵 35 和 35'（对应于最高旋转概率）的最高序列概率的序列值，可计算位置的坐标。

如上文所说明的，不必检查所有四个旋转。这可由一个例子简单地说明。现采用记录的一个图像中的一个位置码的局部区域是图 19 中显示的那个（45）。再采用位置码的“正确”旋转是相对于记录的编

码顺时针旋转 90 度。根据上述说明, 其中正确旋转被采用为旋转 0 度, 这意味着记录的图像中的位置码 45 相对于“正确”旋转顺时针旋转了 270 度。因此, 位置码 45' 相对于“正确”旋转顺时针旋转了 90 度。以上文所述的方式, 现在在图 19 中解码位置码 45, 45'。如上文所说明的, 具有“正确”旋转 0 度的标记的列排列在位置码 45 中正确的解码方向。正确的方向引起对应于这些行的序列概率为高, 从而引起旋转概率为高。如上所述, 具有正确旋转的标记的行是在位置码 45 中以错误解码方向排列的列。错误的方向与倒置结合引起对应于这些列的值概率为低, 从而引起旋转概率为低。对于位置码 45' 关系将是相反的。如上文所说明的, 具有正确旋转 0 度的标记的列将为位置码 45' 中以错误的解码方向排列的行。错误的方向与倒置结合引起对应于这些行的序列概率为低, 从而引起旋转概率为低。如上所述, 具有正确旋转的标记的行是在位置码 45' 中以正确解码方向排列的列。正确方向引起对应于这些列的序列概率为高, 从而引起旋转概率为高。如上文所提到的, 当记录位置码的“正确”旋转时, 列和行将在图像中的“正确”方向延伸。这意味着对于图 15 中的例子, 矩阵 34 和 35 的旋转概率将均高于矩阵 34' 和 35' 的旋转概率。这表示已记录了位置码的“正确”旋转。从而可根据对应于矩阵 34 的最高序列概率的序列值计算一个第一坐标, 并根据对应于矩阵 35 的最高序列概率的序列值计算一个第二坐标。

当记录相对于“正确”旋转旋转 180 度的位置码时, 行和列将在图像中的“错误”的方向延伸。这意味着对于图 15 中的例子, 矩阵 34' 和 35' 的旋转概率将均高于矩阵 34 和 35 的旋转概率。这表示位置码已经以相对于“正确”旋转 180 度的旋转记录。从而可根据对应于矩阵 34' 的最高序列概率的序列值计算一个第一坐标, 并根据对应于矩阵 35' 的最高序列概率的序列值计算一个第二坐标。

相对于“正确”旋转顺时针旋转 90 或 270 度的位置码的记录由不属于相同旋转的最高旋转概率表示。如果采用图 15b 中的矩阵 34 和 35 来自相对于正确方向顺时针旋转 90 度的一个位置码, 则矩阵 34 的

旋转概率将高于矩阵 34' 的旋转概率，而矩阵 35 的旋转概率将低于矩阵 35' 的旋转概率。在这种情况下，根据对应于矩阵 34 的最高序列概率的序列值计算第二坐标，并根据对应于矩阵 35' 的最高序列概率的序列值计算第一坐标。如果采用图 15b 中的矩阵 34 和 35 来自相对于正确方向顺时针旋转 270 度的一个位置码，则根据对应于矩阵 34' 的最高序列概率的序列值计算第二坐标，并根据对应于矩阵 35 的最高序列概率的序列值计算第一坐标。

允许检测记录的图像中的位置码的旋转的是以下事实：当一个矩阵旋转 90, 180 或 270 度时，该矩阵的旋转概率会改变。如果位置码的旋转不为 0，即如果记录的图像中的位置码的旋转错误，则如上文所述，旋转概率将为低。这取决于以下事实：循环主数字序列中长度为 8 的唯一局部序列不会倒置或颠倒地出现在主数字序列中。如果对于长度为 6 的局部序列的主数字序列满足此条件，则意味着主数字序列将显著减少，这意味着能编码的位置更少。因而这是为什么虽然理论上只要求 6×6 个元素但在位置确定中要使用 8×8 个标记元素的原因之一。

旋转检测中使用的基本原理可用在纠错中。例如，可选择主数字序列，使得长于位置确定所要求的预定长度的局部序列不会以一比特倒置出现在主数字序列中。这样，如果能确定地检测到这样一个较长的局部序列中除一个比特外的所有比特，则可以纠正错误的比特。这是为什么虽然理论上只要求 6×6 个元素但在位置确定中要使用 8×8 个标记元素的另一个原因。

这样，通过适当地选择主数字序列，可相当显著地改进编码图案的检错和纠错属性。

但是，对于 64 比特长的主数字序列，可能不具备循环主数字序列的长度为 8 的局部序列不出现倒置或颠倒的这一属性，这是为什么主数字序列的长度选为 63 的原因。

因此，在解码中使用冗余信息获得纠错属性。在上文说明的例子中， 8×8 个标记元素被用于解码中，虽然位置信息可根据 6×6 光栅点

提取，即对于确定位置有 56 比特的冗余信息 $[(8^2-6^2) \times 2]$ 。在解码中，在使用属于当前图像的值概率的同时，按列和按行将当前图像中的信息与可能出现在位置码中的不同局部序列匹配。关于冗余信息、概率和标记元素值之间的关系已知条件的组合给出对当前图像中的干涉的良好不敏感性。这样每个标记的值的的重要性降低了，因为单个标记的值必须对应于给出最高序列概率的局部序列中的其他值。

如果对于纠错的需求有限，则设备可被替换地设计为，为第一矩阵中的每一列和第二矩阵中的每一行，选择一个对应于每个标记元素的 0 和 1 概率的最高者的序列，以及从而选择一个序列值。

当已建立了记录的位置码相对于“正确”旋转的旋转 40° 时，则可确定位置的第一和第二坐标（步骤 L）。如上文所述，此操作是根在图 18 中对于第一坐标标示为 $Sx_1-Sx_8(41)$ 和对于第二坐标标示为 $Sy_1-Sy_8(42)$ 的序列值执行。

设备被设计为对序列值 Sx_1-Sx_8 和 Sy_1-Sy_8 计算相邻序列值之间的差，它产生七个差分数字 Dx_1-Dx_7 和 Dy_1-Dy_7 的两个集合 43 和 44。然后这些差分数字被用于产生一个第一坐标和一个第二坐标。

但是，如上文所说明的，对于第一坐标的计算，只有序列值 Sx_1-Sx_8 中的六个，即差分数字 Dx_1-Dx_7 中的五个是必须的。因而根据本例使用了序列值 Sx_2-Sx_7 和差分数字 Dx_2-Dx_6 。这一点也适用于第二坐标，它由序列值 Sy_2-Sy_7 和差分数字 Dy_2-Dy_6 计算出。也可分别对每个方向只确定六个序列值， Sx_2-Sx_7 和 Sy_2-Sy_7 。

从差分数字到坐标的换化可以多种方式实现，例如以在这里被结合进来作为参考的申请人的申请 WO 01/26033 中说明的方式实现。

在上文说明的例子中，已经为数据解码在一个记录的图像中标识了 8×8 个标记元素。但是，有时候会出现不可能标识这么多标记元素的情况。则额外的“空”标记元素被加到标记元素中，它们可在图像中被标识以获得总共 8×8 个标记元素。如先前所说明的，“空”标记元素的值概率均为相等的。

图1

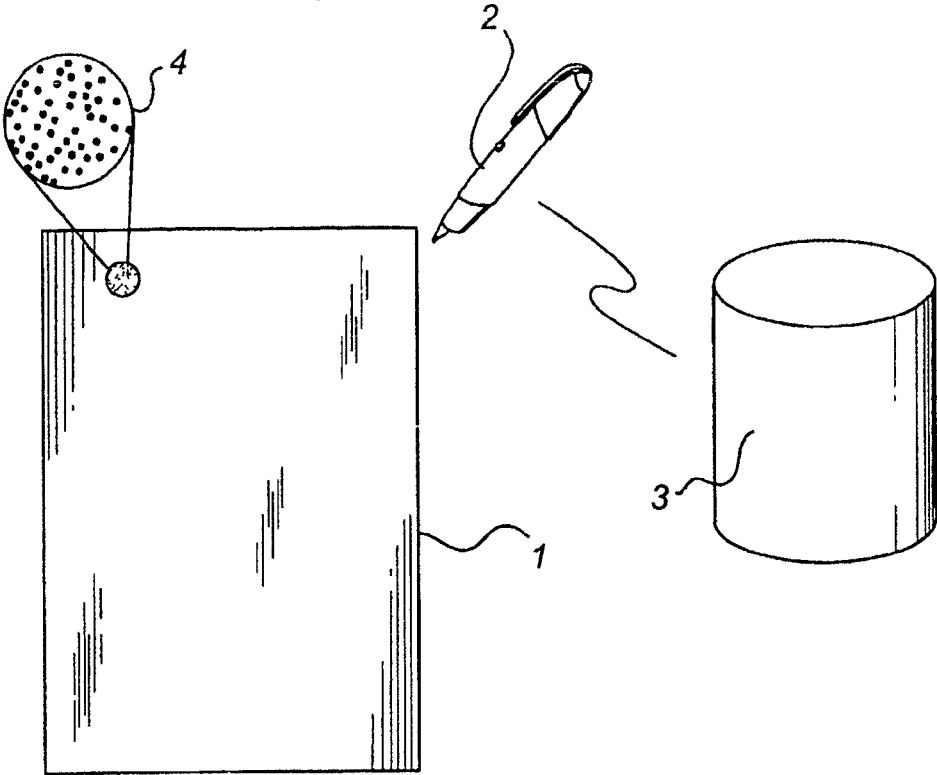


图2

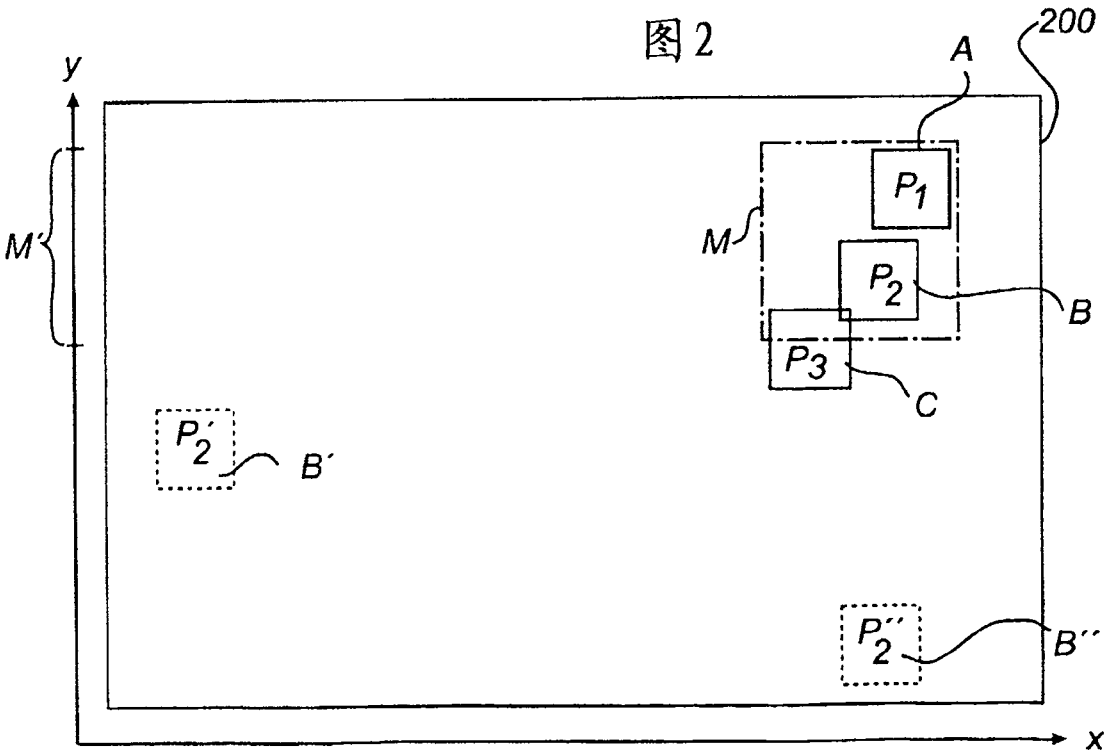


图3

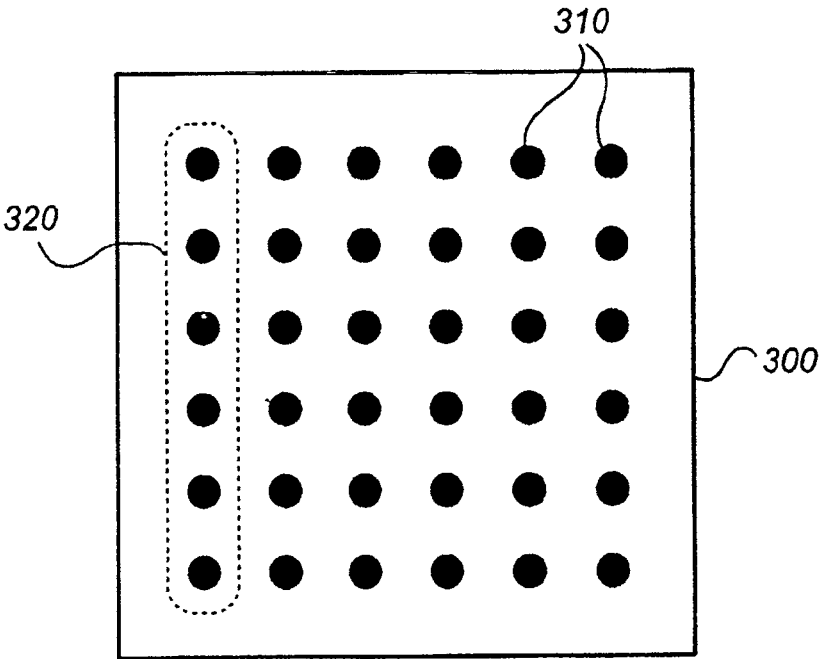


图 4

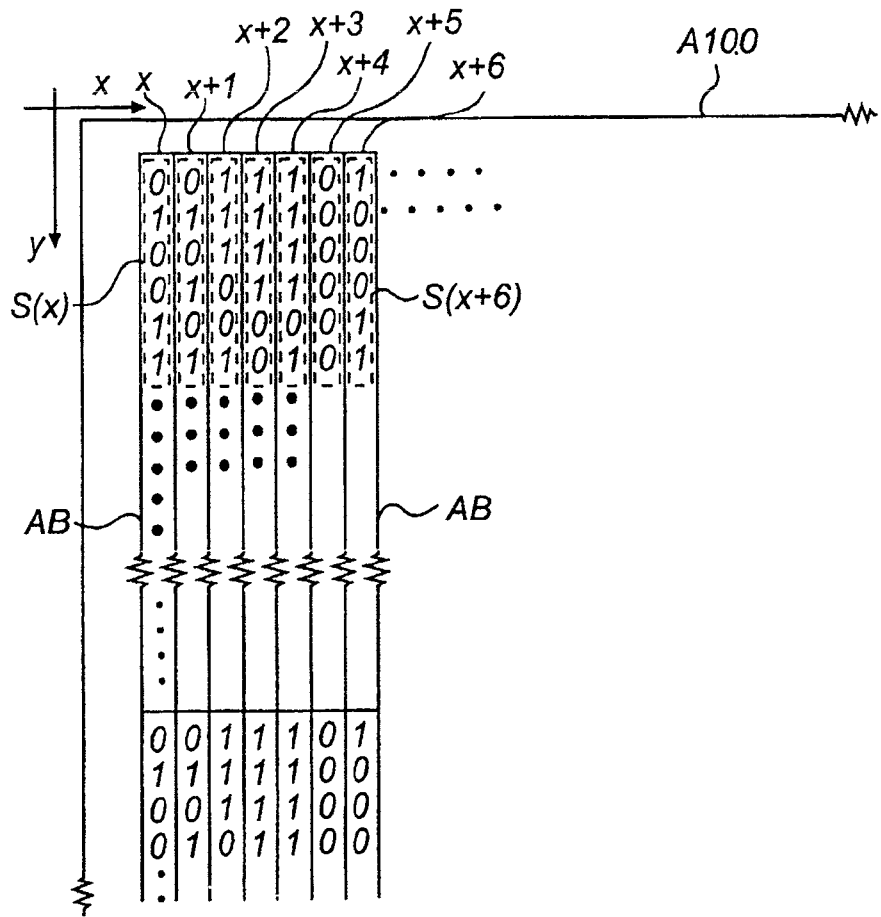
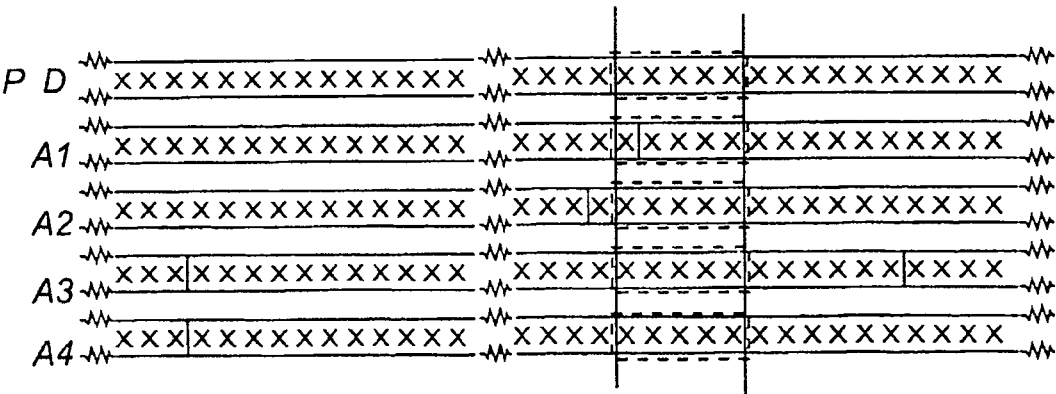


图 5



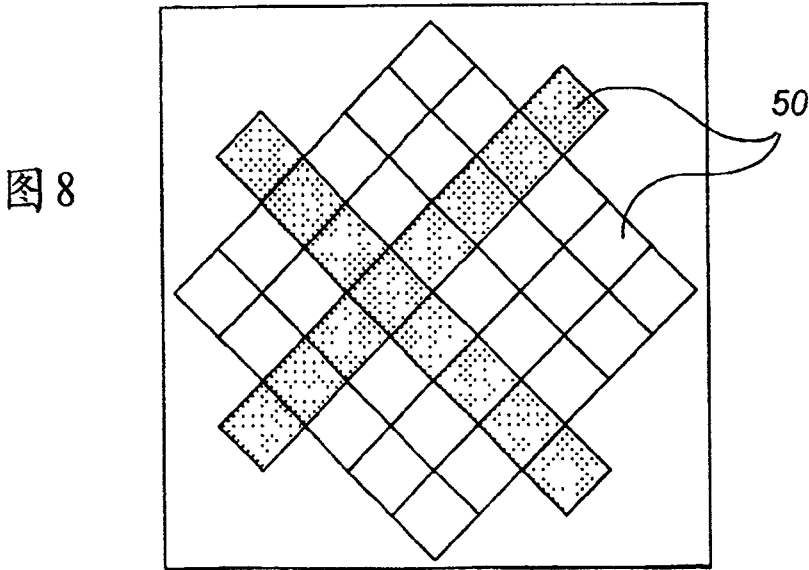
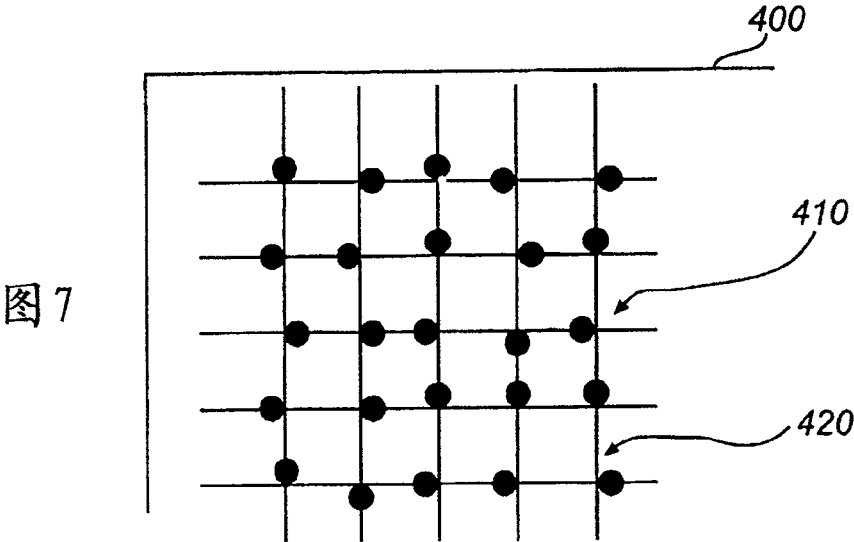
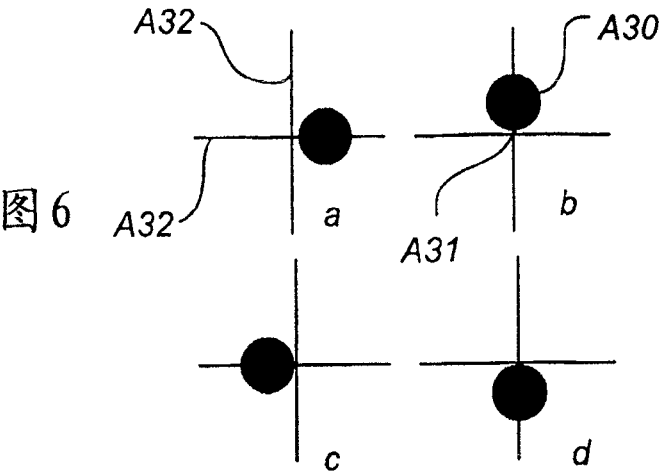


图9

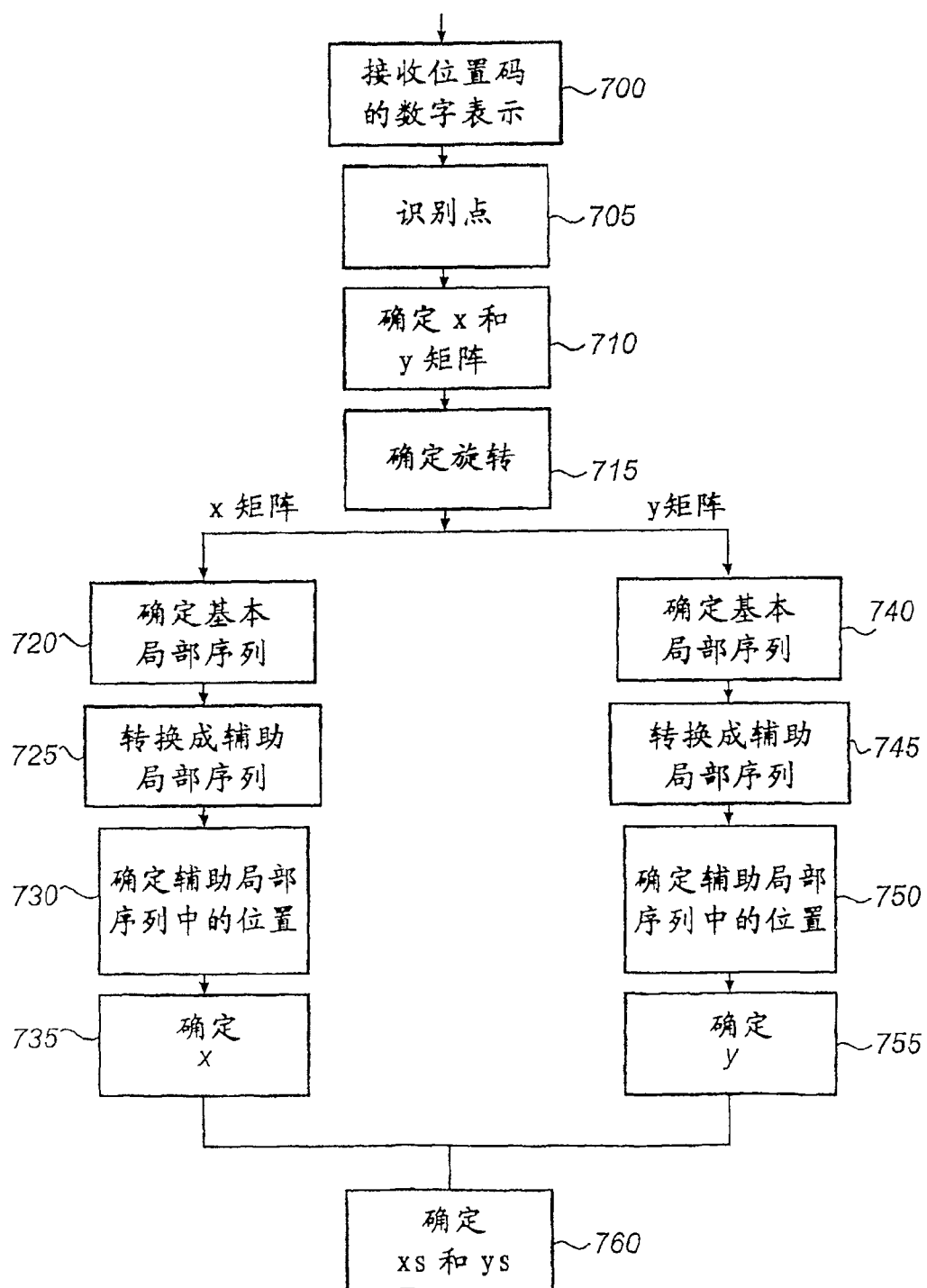


图10

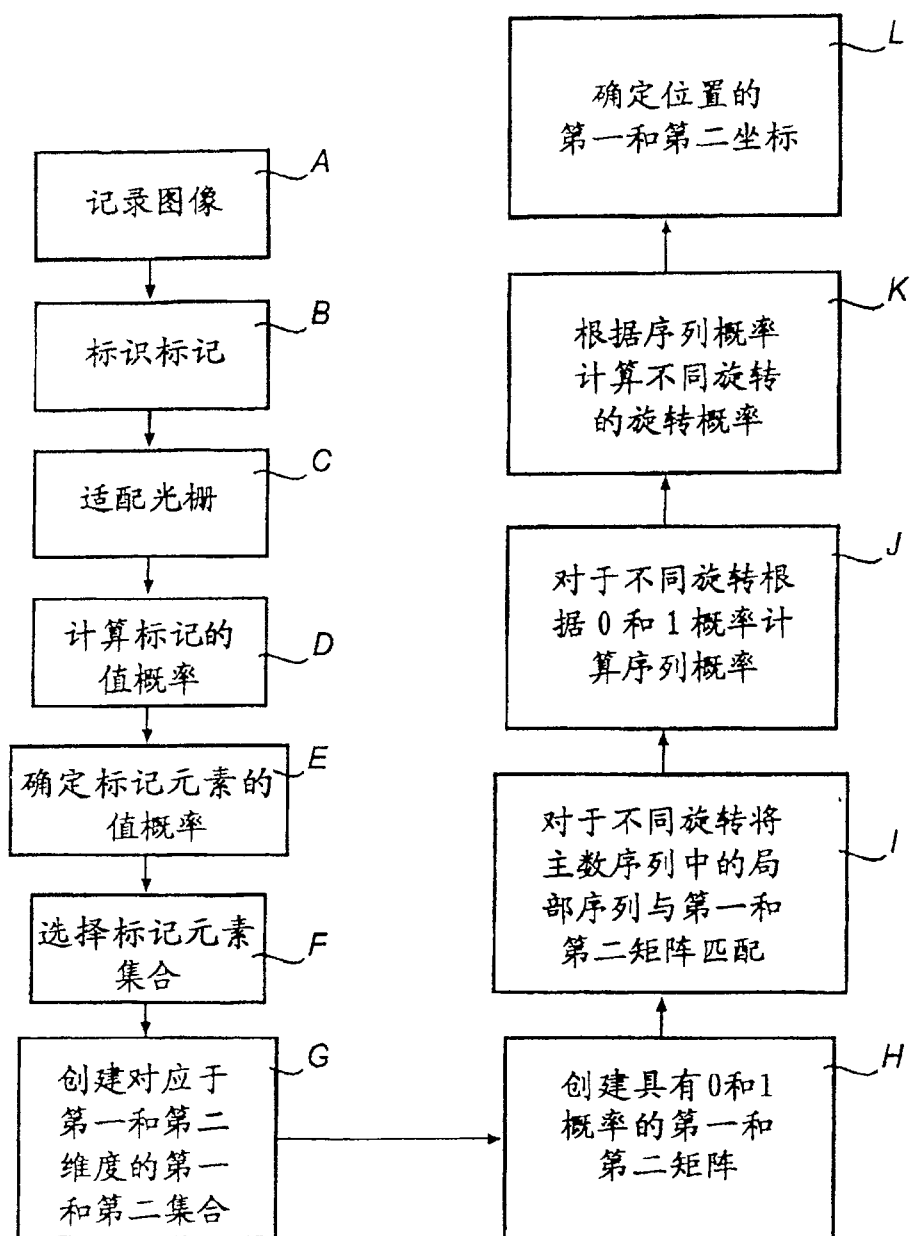


图 11a

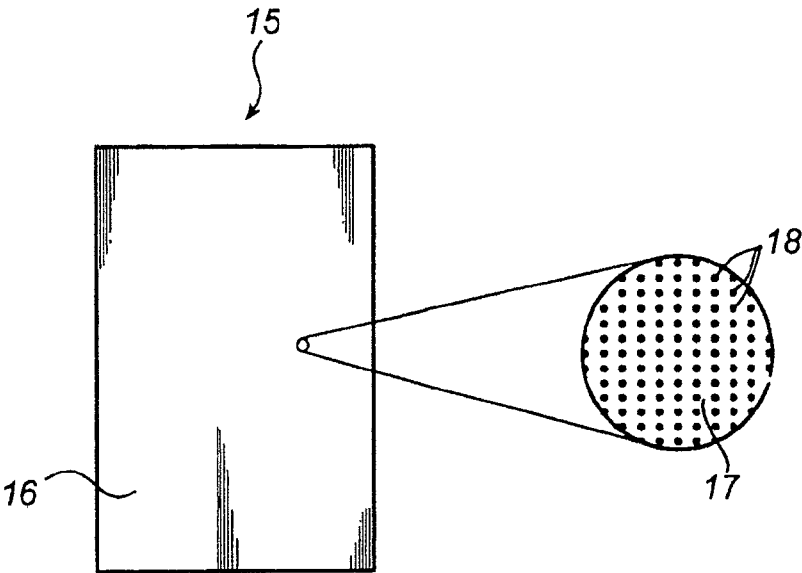


图 11b

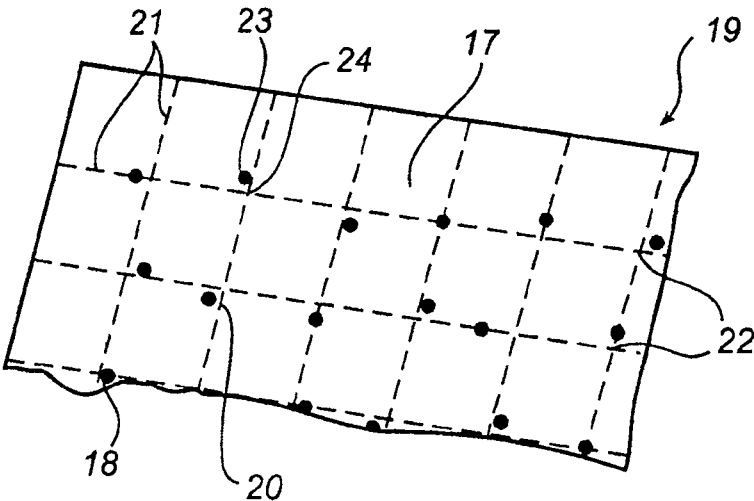


图 12

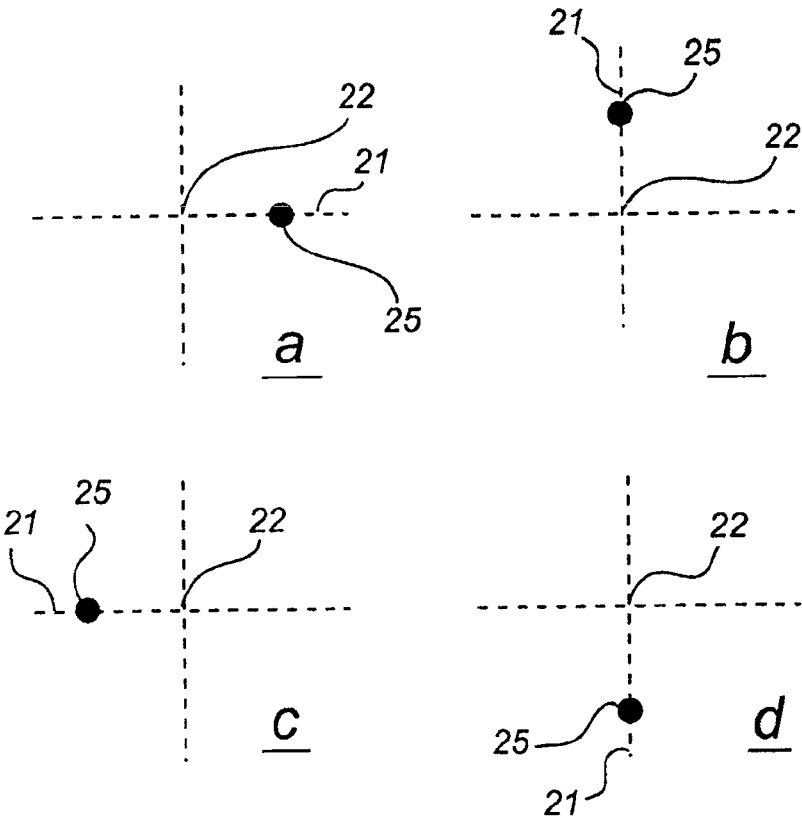


图13

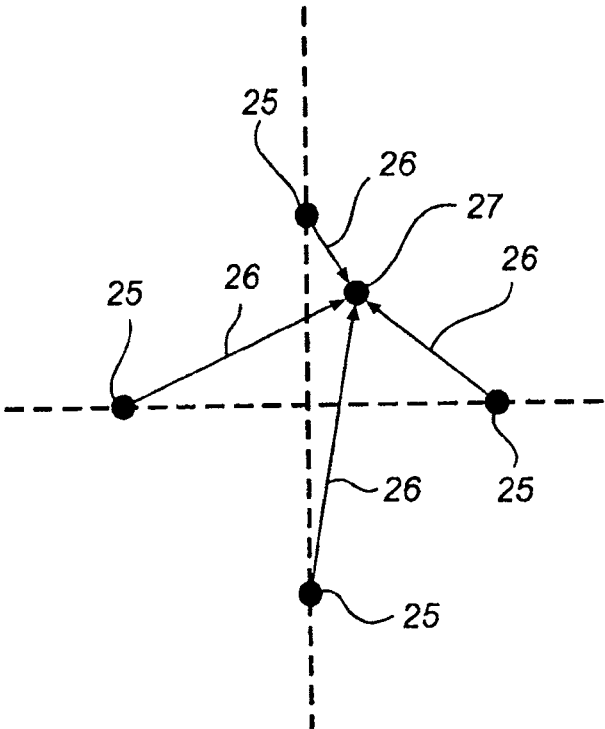


图14a

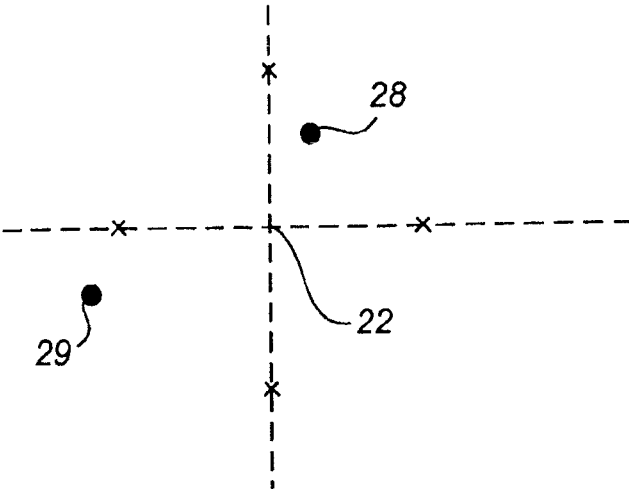


图14b

30

值	P_1
"0"	0,26
"1"	0,45
"2"	0,16
"3"	0,13

31

值	P_1
"0"	0,11
"1"	0,13
"2"	0,57
"3"	0,19

图14c

32

值	P_2
"0"	0,26
"1"	0,45
"2"	0,57
"3"	0,19

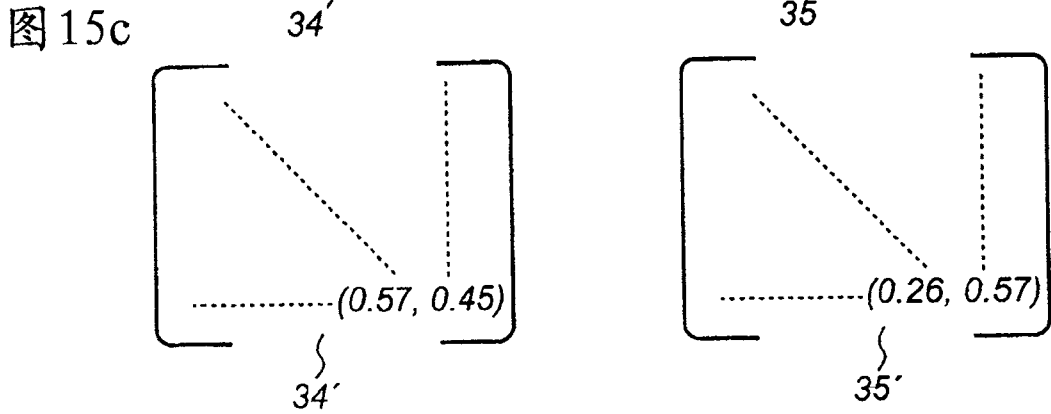
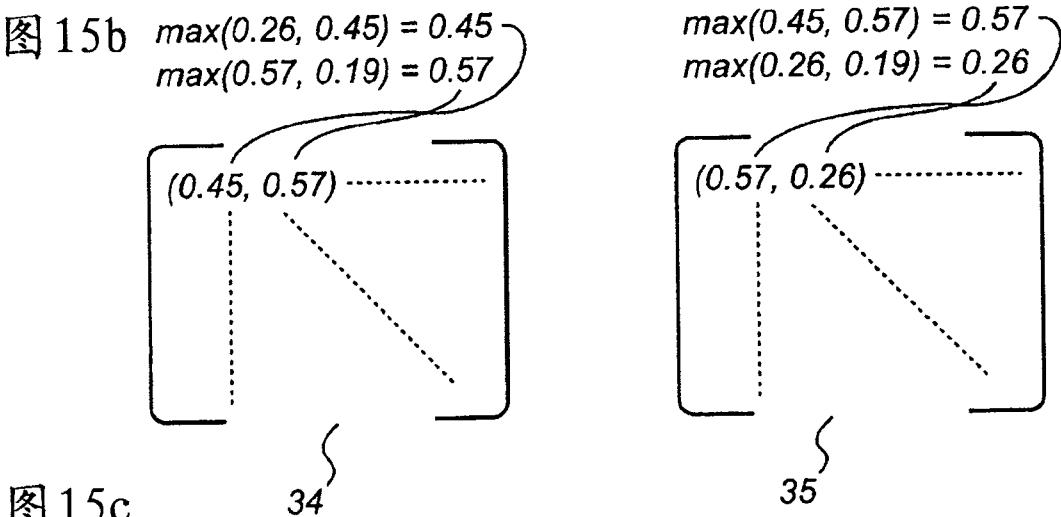
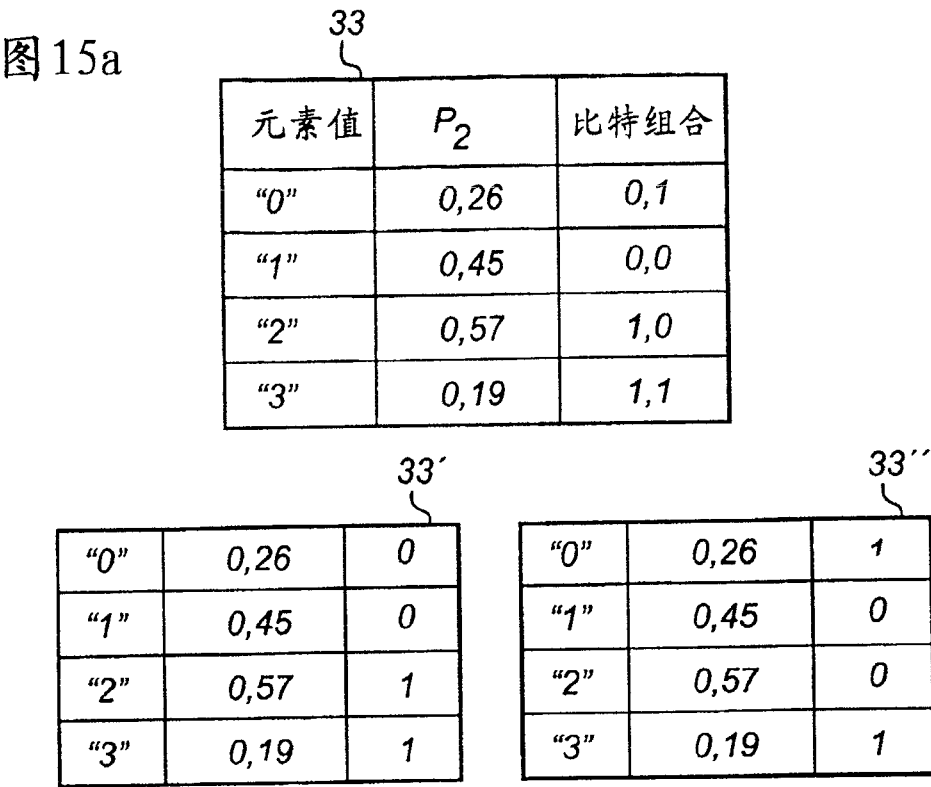


图17

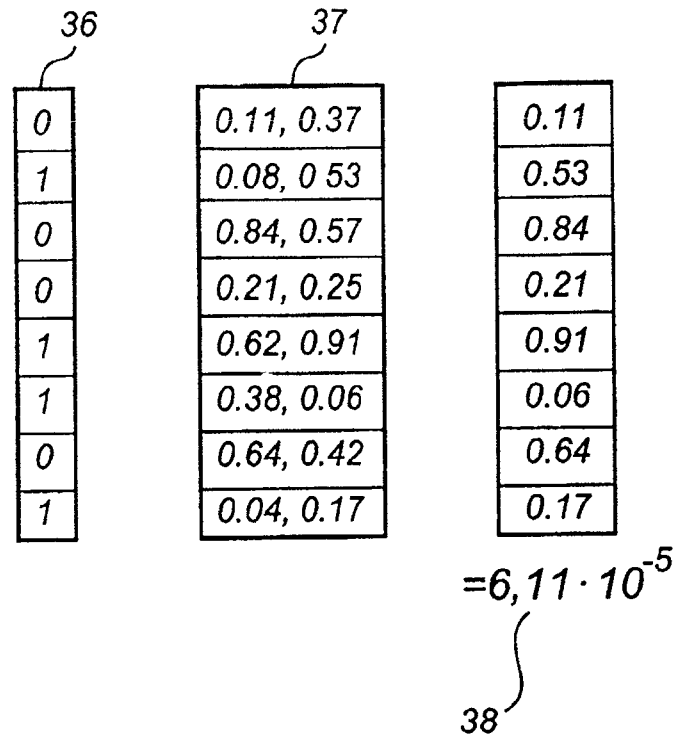


图16

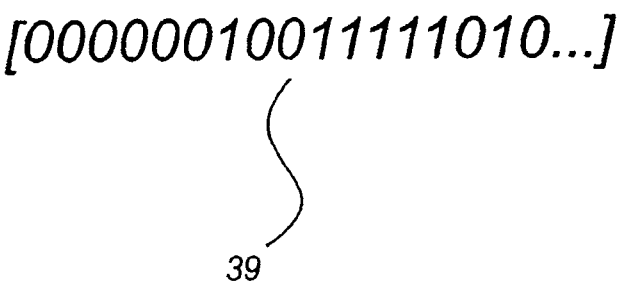


图 18

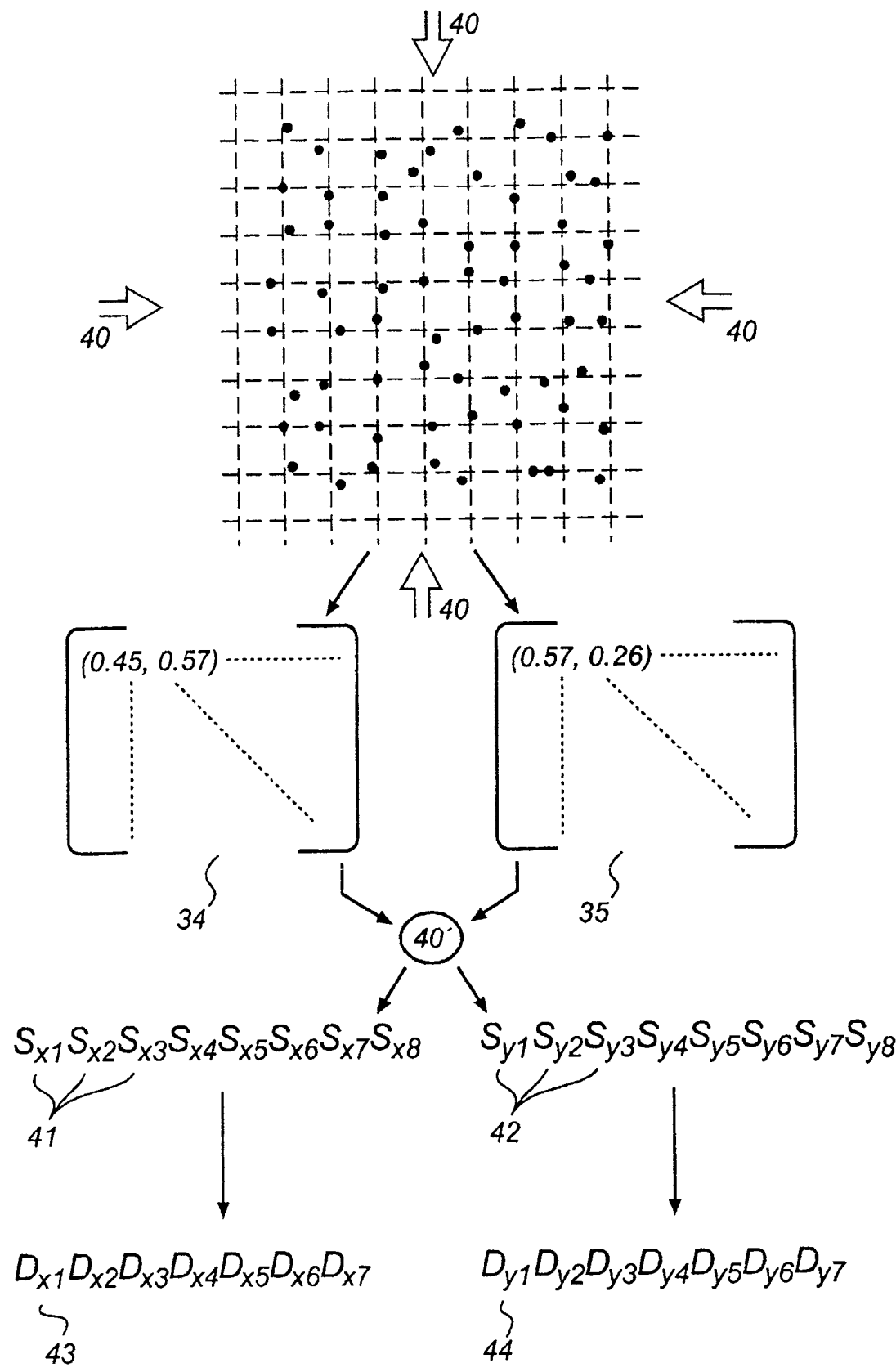
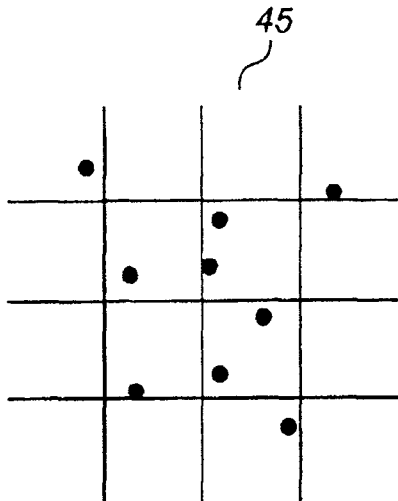


图 19

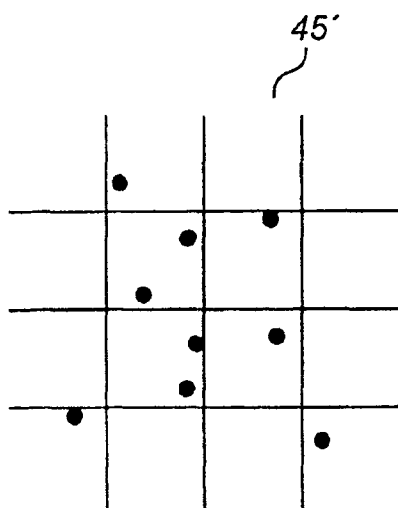


46

(0.45, 0.27)	(0.31, 0.41)	(0.73, 0.09)
(0.36, 0.14)	(0.73, 0.09)	(0.16, 0.47)
(0.72, 0.09)	(0.55, 0.12)	(0.12, 0.55)

47

(0.45, 0.15)	(0.15, 0.41)	(0.11, 0.73)
(0.36, 0.36)	(0.73, 0.11)	(0.47, 0.24)
(0.12, 0.72)	(0.55, 0.22)	(0.22, 0.55)



48

(0.55, 0.12)	(0.12, 0.55)	(0.09, 0.72)
(0.47, 0.16)	(0.09, 0.73)	(0.14, 0.36)
(0.09, 0.73)	(0.41, 0.31)	(0.27, 0.45)

49

(0.55, 0.22)	(0.22, 0.55)	(0.72, 0.12)
(0.24, 0.47)	(0.11, 0.73)	(0.36, 0.36)
(0.73, 0.11)	(0.41, 0.15)	(0.15, 0.45)

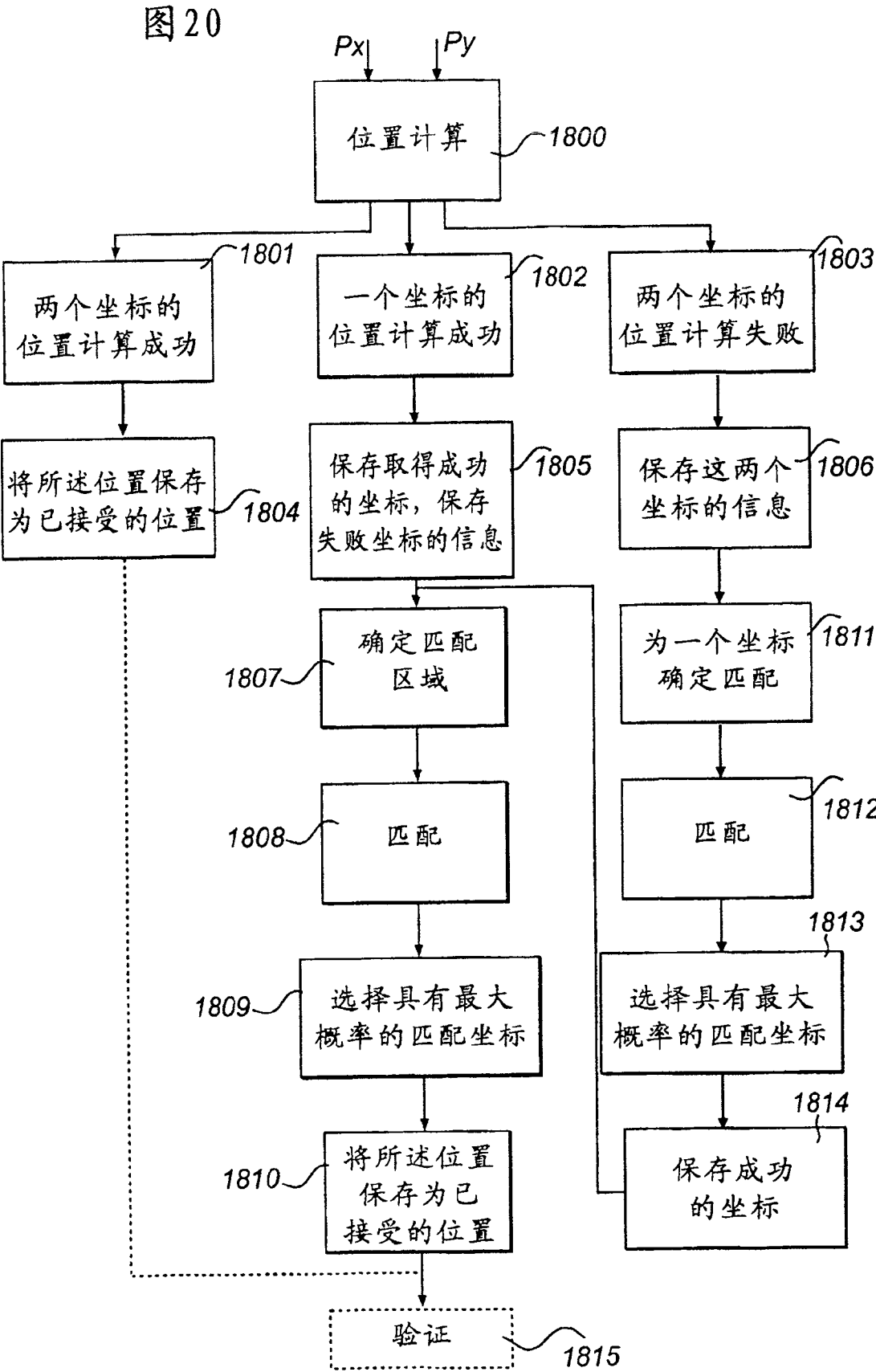


图 21

