

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06T 7/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 01803867.0

[45] 授权公告日 2006 年 6 月 28 日

[11] 授权公告号 CN 1261908C

[22] 申请日 2001.1.17 [21] 申请号 01803867.0

[30] 优先权

[32] 2000.1.18 [33] CA [31] 2,296,143

[86] 国际申请 PCT/CA2001/000042 2001.1.17

[87] 国际公布 WO2001/054068 英 2001.7.26

[85] 进入国家阶段日期 2002.7.18

[71] 专利权人 索威森公司

地址 加拿大魁北克

[72] 发明人 A·考奥贝 M·坎廷 L·贝拉尔

J·戈泽尔

审查员 李 菲

[74] 专利代理机构 北京市中咨律师事务所

代理人 李 峥 于 静

权利要求书 5 页 说明书 21 页 附图 16 页

[54] 发明名称

检测印刷电路板缺陷的方法和系统

[57] 摘要

这里描述了用于检测诸如印刷电路板 (PCB) 的电子电路的表面缺陷的一种方法和系统。该方法首先包括识别 PCB 的数字图像的轮廓。然后通过比较识别的轮廓与 PCB 矢量模型上的轮廓检测出 PCB 图像上的异常。每一检测出的异常与制造数据比较以检验其是否对应于一种缺陷。

1. 检测具有至少一层包含组件的印刷电路板(PCB)上表面缺陷的方法, 该方法包括:

捕获 PCB 的数字图像;

识别所述 PCB 图像上的边缘;

在计算机模型中表示所述 PCB 的至少一层上的每一组件;

通过比较所述被识别的边缘与所述计算机模型检测 PCB 图像上的异常; 以及

对于每一所述被检测的异常, 确定所述被检测到的异常是否对应于一个表面缺陷。

2. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 其中所述至少一层上每一组件的所述计算机模型包含线段。

3. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 其中所述边缘识别步骤包括:

计算所述 PCB 图像上的数值梯度以产生梯度图像;

阈值处理所述梯度图像以产生阈值处理后的图像;

对所述阈值处理后的图像骨架化; 以及

在骨架化的图像上找出边缘。

4. 如权利要求 3 所述用于检测 PCB 上表面缺陷的方法, 其中边缘找出步骤产生至少一串像素, 并包含:

作为像素坐标矢量保存所述骨架化的图像的所述至少一个像素串的每一个;

使用所述像素坐标估计所述至少一个像素串的二阶导数; 以及

估计所述二阶导数为零的通路的坐标;

从而, 零通路对应于所述边缘。

5. 如权利要求 3 所述用于检测 PCB 上表面缺陷的方法, 还包括校正 PCB 图像的光度。

6. 如权利要求 5 所述用于检测 PCB 上表面缺陷的方法, 还包括过滤已校正的图像。

7. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 其中所述的计算机模型包含所述至少一层中的每一层的矢量模型。

8. 如权利要求 7 所述用于检测 PCB 上表面缺陷的方法, 其中所述矢量模型包含所述至少一层的每层上每一组件的多边形表示。

9. 如权利要求 2 所述用于检测 PCB 上表面缺陷的方法, 其中在异常检测步骤之前所述计算机模型的所述至少一层的每一个与所述 PCB 图像对准。

10. 如权利要求 9 所述用于检测 PCB 上表面缺陷的方法, 其中对于计算机模型所述至少一层的每一个在 PCB 图像上选择至少一个基准点; 所述至少一个基准点被用来使对应的层与所述 PCB 图像对准。

11. 如权利要求 9 所述用于检测 PCB 上表面缺陷的方法, 其中在所述计算机模型的至少一层的每一层的每一线段上施加变换, 使所述至少一层的每一层与所述 PCB 图像对准, 以便补偿所述至少一层的每一层的旋转、平移、缩放和不良透视中的至少一个。

12. 如权利要求 9 所述用于检测 PCB 上表面缺陷的方法, 其中所述对准的层是叠加的。

13. 如权利要求 12 所述用于检测 PCB 上表面缺陷的方法, 其中所述提供的计算机模型是矢量模型, 包含每一层上每一组件的多边形表示; 所述多边形表示包含至少一个线段; 所述计算机模型的所述层是通过叠加来自一层的每一多边形与来自其它层的每一多边形而被叠加的。

14. 如权利要求 13 所述用于检测 PCB 上表面缺陷的方法, 其中所述叠加是对于每一层使用以下迭代公式进行的:

$$P_{v_{k+1}} = (P_{v_k} - P_{k+1}) \cup P_{k+1}$$

这里

P_k 是当前层中线段的矢量表示列表;

P_{v_k} 是通过先前层叠加产生的线段的矢量表示的列表;

Pv_{k+1} 是通过将先前的层的叠加进行叠加产生的线段的矢量表示的列表; 及

$$Pv = P_1.$$

15. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 其中所述计算机模型包含每一组件的轮廓。

16. 如权利要求 15 所述用于检测 PCB 上表面缺陷的方法, 其中所述计算机模型对于每一轮廓还包括围住所述轮廓的矩形的模型。

17. 如权利要求 16 所述用于检测 PCB 上表面缺陷的方法, 其中所述异常检测步骤包括: a) 对于 PCB 图像的每一像素, 检验每一组件的所述轮廓之中的一个轮廓是否有其对应的封闭所述像素的矩形; 如果否, 认为所述像素是一异常的一部分; 如果是, b) 验证所述像素是否是以其对应的矩形封闭所述像素的所述轮廓的一部分; 如果否, 认为所述像素是一异常的一部分。

18. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 其中所述异常检测步骤包括使用所述计算机模型生成所述计算机模型的图像, 并比较所述 PCB 图像与所述计算机模型的所述图像。

19. 如权利要求 2 所述用于检测 PCB 上表面缺陷的方法, 其中缺陷的确定包括将所述检测到的异常与设计规范比较。

20. 如权利要求 19 所述用于检测 PCB 上表面缺陷的方法, 其中所述检测 PCB 图像上的异常的步骤还包括, 检测 PCB 图像中相对于所述计算机模型中的所述线段处于超越的边缘; 对于所述检测的异常中的每一个, 通过以下确定缺陷 a) 确定所述计算机模型中位于所述检测的异常之一的预定距离内的第一线段; b) 对于与所述第一线段相对的第一线段, 确定所述第一线段与所述相对线段之间的距离是否在预定的阈值内。

21. 如权利要求 20 所述用于检测 PCB 上表面缺陷的方法, 其中当一个以上的异常位于所述第一线段与所述相对线段之间时, 作为它们之间有效的间隔计算所述第一线段与所述相对线段之间的所述距离。

22. 如权利要求 21 所述用于检测 PCB 上表面缺陷的方法, 其中所述

异常包括重叠及非重叠异常；作为所有非重叠异常与重新分组的重叠异常之间的区间的和计算所述有效间隔。

23. 如权利要求 20 所述用于检测 PCB 上表面缺陷的方法，其中所述预定的距离是由围住所述检测出的异常的一矩形定义的。

24. 如权利要求 23 所述用于检测 PCB 上表面缺陷的方法，其中所述围住矩形被膨胀。

25. 如权利要求 23 所述用于检测 PCB 上表面缺陷的方法，还包括检验所述围住矩形是否围住另一异常，如果是，则重新定义围住所述另一异常的一矩形。

26. 如权利要求 25 所述用于检测 PCB 上表面缺陷的方法，其中所述矩形重定义步骤是迭代的。

27. 如权利要求 20 所述用于检测 PCB 上表面缺陷的方法，其中所述阈值是比率的形式。

28. 如权利要求 13 所述用于检测 PCB 上表面缺陷的方法，其中所述异常检测步骤包括对所述计算机模型的每一层 k ，使用以下迭代公式，确定位于所述层 k 的狭窄区域中的多边形 T_k 的一列表：

$$T_{k+1} = (T_k - P_{k+1}) \cup ((D(P_{k+1}) \cap D(P_{V_k})) - D(P_{k+1} \cap P_{V_k}))$$

其中

P_{k+1} 是先前层中多边形的矢量表示的列表；

P_{V_k} 是通过先前层的叠加产生的多边形的矢量表示的列表；

$P_{V_{k+1}}$ 是通过与当前层的叠加产生的多边形的矢量表示的列表；

D 是引起多边形列表中所有多边形膨胀的函数；

以及

$$T_1 = D(P_1) \cap D(P_1);$$

所述多边形列表中两个相对的线段产生有一宽度的内部空间；如果相应的宽度不在预定的阈值内，则每一所述内部空间对应一缺陷。

29. 如权利要求 28 所述用于检测 PCB 上表面缺陷的方法，其中通过每一所述多边形的侵蚀确定所述内部空间的所述宽度。

30. 如权利要求 13 所述用于检测 PCB 上表面缺陷的方法, 其中所述异常检测步骤包括对于所述至少一层 k 的每一个, 使用以下迭代公式确定 PCB 的计算机模型上重叠区域中的多边形 O_k 的列表:

$$O_{k+1} = (O_k - P_{k+1}) \cup (P_{k+1} \cap P_{v_k})$$

这里:

P_{k+1} 是先前层中多边形的矢量表示的列表;

P_{v_k} 是通过先前层的叠加产生的多边形的矢量表示的列表;

以及

O_1 是空;

所述多边形列表中两个相对的线段产生有一宽度的内部空间; 如果相应的宽度不在预定的阈值内, 则每一所述内部空间对应一缺陷。

31. 如权利要求 30 所述用于检测 PCB 上表面缺陷的方法, 其中所述多边形列表包括可见的和隐藏的轮廓; 使用所述可见的轮廓外插所述隐藏的轮廓。

32. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 还包括在计算机文件中指明所述检测到的异常。

33. 如权利要求 1 所述用于检测 PCB 上表面缺陷的方法, 其中所述图像是位图形式的。

34. 如权利要求 33 所述用于检测 PCB 上表面缺陷的方法, 其中所述位图包含灰度级。

35. 用于检测 PCB 上表面缺陷的系统, 采用权利要求 1-34 的其中之一的方法。

检测印刷电路板缺陷的方法和系统

技术领域

本发明涉及用于检测诸如印刷电路板的电子电路上的缺陷的检查系统和方法。更具体来说，本发明涉及基于边缘检测的系统和方法。

背景技术

电子电路可以有很多形式，诸如 PCB(印刷电路板)，引线框架，混合电路。这些电路通常包含多个组成部分(组件)，例如导体，孔，焊点，介质，光聚合物保护膜等等。这些组成部分可以一层在另一层之上组装在各层。这种多层组装的例子是熟知的导体-孔-釉电路结构。在以下的说明中，我们将仅参照印刷电路板(PCB)为例。然而应当理解，本发明不限于电子电路的该实施例。

过去，PCB 的检查通过人工使用放大镜观看电路以视觉进行，试图找出与可接受的电路模型的差别。视觉检查有若干缺陷：主观性，速度慢，难以收集到缺陷的定量信息等。

随着计算机处理速度的增长近来出现了自动化的检查方法。这些自动检查方法在于提取待检查的 PCB 的数字图像，并分析该数字图像以确定缺陷的存在。这种自动化检查方法的例子是基于边缘检测的那些例子。

边缘是组件外形(轮廓)的片段。假设组件(或区域)之间的过度具有光学上不同的特征，则能够在视觉上区分数字图像上色调从一个组件到下一组件的变化。基于边缘的检测方法在于表征(characterize)并分析 PCB 图像的边缘，并通过比较这种边缘和已知数值和标准。

附图的图 1 示出可在 PCB 上或其它类型的电子电路上发现的缺陷的例子。例如，这种缺陷可能是 PCB 两层之间的不良重叠 10，两层未对准 12，两个相邻轨迹(track)之间间隙过小 14，两个组件之间桥接 16，轨迹太

窄 18 或轨迹断开 19。

在美国专利 No. 4,570,180, 标题为“用于自动光学检查的方法”(1986年2月11日授权, Baier 等人作为发明人)中描述了用于检测电子电路缺陷的基于边缘的方法的一个例子。该专利涉及使用数字图像处理技术用于二维图案的自动光学检查的方法和设备。该方法包括第一步骤, 在该步骤对于边缘或线条扫描灰度级数字图像, 并在图像存储中标记这些边缘。然后, 扫描并对允许的灰度测试图像存储中所有未标记的区域。

由于仅仅是通过对允许的灰度级比较未标记的区域对异常进行测试, Baier 方法的缺陷在于, 其检测的质量太依赖于数字图像的质量。实际上, 一个对象的数字化图像由于数字化的算法、放大倍率差等在特征的大小和位置上可以呈现出变化。

Baier 方法的另一问题在于, 几乎需要相同的计算量测试数字图像上不对应于边缘的所有像素, 这导致计算资源和时间的浪费。

Baier 方法的又一缺点在于, 要表征检测到的缺陷可能是困难的。这种表征对于找到电子电路制造过程中出现的错误模式可能是有用的。

在 1995 年 9 月 19 日颁发的标题为“检测半导体组件引线的方法”的美国专利 No. 5,452,368 中, LeBeau 提出一种检测物体中缺陷的方法, 这是通过比较第一物体的第一灰度级图像与第二物体的第二灰度级图像。更精确地说, 对第一图像的边缘特征骨架化 (skeletonize) 并与第二图像膨胀的边缘特征比较, 并反之亦然。与 Baier 方法相反, LeBeau 方法不测试数字图像上非边缘相关的区域。

然而 LeBeau 方法的缺点在于, 通过比较物体的两个图像来搜索缺陷, 而该物体可能在大致相同的位置被非常相似的缺陷改变。LeBeau 方法不允许检测这种缺陷。由于错误的制造过程可能引起这种反复出现的缺陷, 故这是一主要的缺点。

因而, 希望有一种方法和系统, 允许检测电子电路上的缺陷而不依赖于电子电路数字化图像的质量和分辨率, 并且不把灰度的变化解释为缺陷。

还希望有使用设计数据的检测缺陷的方法和系统。

进而希望有这样的方法和系统，它能提供两种水平的检查，一种是检测异常，而另一种是针对缺陷检查这些异常，这有助于使处理速度最大化。

发明内容

更具体地，根据本发明，提供了检测至少一层包含组件的印刷电路板(PCB)的表面缺陷的方法，该方法包括：

提供 PCB 的数字图像；

识别 PCB 图像上的边缘；

对 PCB 的至少一层上的每一组件提供对应的计算机模型；

通过比较所述被识别边缘与计算机模型检测 PCB 图像上的异常；以及对于每一被检测的异常，确定被检测到的异常是否对应于表面的缺陷。

根据本发明的另一方面，提供了用于检测 PCB 的表面缺陷的系统；该系统包括：

计算机，包含 PCB 模型并被配置为识别 PCB 图像边缘，通过比较被识别的边缘与该计算机模型来检测 PCB 图像上的异常，并对每一检测到的异常，确定检测到的异常是否对应于表面缺陷；

连接到计算机向 PCB 提供照明的照明部件；

连接到计算机的帧抓取器 (frame grabber)；

连接到帧抓取器以摄取 PCB 图像的照相机；以及

连接到帧抓取器的定位系统。

应当注意，术语“PCB”这里应解释为任何可能包含视觉可分辨的表面缺陷的电子电路。

在阅读以下仅以例子方式参照附图给出的本发明非限制性优选实施例的说明时，对本发明的优点和特征将更为显而易见。

附图说明

在附图中：

图 1 是表示印刷电路板(PCB)上的缺陷的例子的示意图；

图 2 是根据本发明的一实施例用于检测 PCB 上的缺陷的系统的框图；
图 3 是根据本发明的一实施例用于检测 PCB 上的缺陷的方法的流程图；

图 4 是图 3 轮廓识别步骤的流程图；

图 5 是 PCB 图像和从其所得的计算机模型的示意图，表示用于这些图像校准的基准点的选择；

图 6a 是来自三层 PCB 模型的重叠多边形的示意图；

图 6b 是图 6a 在减去随后的层的多边形的示意图；

图 6c 是图 6a 的多边形相交部分的示意图；

图 7a 是图 6a 的多边形的示意图，表示狭窄的区域；

图 7b 是来自图 7a 的 P1 和 P3 的多边形在膨胀之后的示意图；

图 7c 是图 7a 的多边形叠加结果的示意图；

图 8 是表示检测 PCB 区域上遗漏的轮廓的示意图；

图 9 表示在检测图 3 的异常之中的缺陷的步骤中定义两个有效半平面（active half-plan）图的示意图；

图 10 是表示当线段不平行时测量线段之间的距离的示意图；

图 11 是表示测量两个平行线段之间积累间隔的示意图；

图 12 是表示确定异常分析区域(AAR)的示意图；

图 13a、13b 和 13c 是确定异常是否违反允许的宽度或间距的方法的流程图；

图 14 是表示狭窄的空间区域中缺陷的示意图；

图 15 是检测狭窄空间区域中缺陷的决策树；

图 16 是来自矢量模型的表示侵蚀处理的多边形的示意图；以及

图 17 图 16 在侵蚀处理之后的多边形的示意图。

具体实施方式

一般来说，根据本发明的实施例的用于检测 PCB 表面缺陷的方法包括
1)识别 PCB 数字图像上的边缘，2)通过比较所识别的边缘与 PCB 的计算

机模型检测异常，以及3)通过每一异常的特征确定检测到的异常是否对应于表面的缺陷。

因而根据本发明的方法提出两种水平的检查：第一水平，允许快速识别异常并需要相对小的计算时间，以及第二水平，是更为精细的检查，比较识别出的异常与设计规范，以便表征检测到的缺陷。

为了实现这种方法，必须提供被检查的PCB的数字图像和类似的PCB的计算机模型。

参见附图的图2，将说明根据本发明的一实施例用于检测PCB表面缺陷的系统20。

一般来说，系统20允许聚集进行检查的PCB图像，并使用根据以下将更为详细说明的本发明的方法检测可能的缺陷。

系统20包括计算机22，帧抓取器24及照明组件26，这两者都连接到计算机22，连接到帧抓取器24的照相机28，及通过轴线控制器32与伺服控制器34连接到帧抓取器24的定位系统30。

计算机22最好是个人计算机形式，其配置为使得可存储由照相机28摄取的待检查的PCB的图像以及类似的PCB的计算机模型这两者。计算机22还最好被编程，以便控制照明组件26，并进行分析图像、及使用根据本发明的方法评估检查的PCB是否包含表面缺陷所需的计算。

照明组件26允许调节被检查的PCB上的光强。

帧抓取器24是连接照相机28与计算机22的一个卡的形式。帧抓取器24最好能够以两种模式任一种操作。按第一种模式，帧抓取器24逐行捕获图像，并向计算机22发送每一行。在第二这模式下，帧抓取器24在向计算机22发送图像之前一次捕获PCB的整个图像。

一种可选的传统图像处理卡36最好接口帧抓取器24和计算机22，并允许快速处理由照相机28摄取的图像。当没有包括卡36时，计算机22被配置成可处理图像。

照相机28允许摄取PCB的数字式图像。照相机28最好是可按线性或矩阵模式操作的CCD(电荷耦合器件)照相机形式。照相机28的分辨率可根

据所检查的 PCB 的性质及所需的精度而改变。

线性 CCD 照相机的使用要求定位系统 30 在垂直于其像素的方向扫描 PCB。然而垂直方向获得的分辨率实际上可以是无限的。其线分辨率大约为 8000 像素。

矩阵 CCD 照相机具有大约 4000×4000 像素的分辨率。最好在 PCB 的稀疏区域使用这种照相机。这种情形下，定位系统 30 可被设置为在区域之间移动。

定位系统 30 包括允许 PCB 位于其上的支架运动的轴和伺服电动机(未示出)。

轴控制器 32 允许定位系统 30 的运动与检查系统 20 的要求同步。

伺服控制器 34 控制定位系统 30 与反馈回路，该回路调节向定位系统 30 的电动机(未示出)供电所需的电流振幅。

由于认为帧抓取器，伺服控制器，定位系统和 CCD 照相机是业内熟知的，这里将不对它们作进一步的详细说明。

当然，在不背离本发明的精神和本质之下系统 20 可以有其它的配置。系统 20 显然可以是生产线的一部分。

现在转到图 3，将更为详细地说明根据本发明的一实施例用于检测 PCB 或其它电子电路上的缺陷的方法 200。

在步骤 202，向计算机 22 提供 PCB 的数字图像。该图像最好是具有灰度级的位图形式的。灰度级数可根据精度的需要的变化。根据本发明的一最佳实施例，系统 20 产生并分析具有 256 级灰度的数字图像。另外，原始的图像也可以有其它的格式，并增加转换步骤以提供位图图像。

在步骤 204，识别 PCB 图像上的轮廓。步骤 204 产生实际的轮廓图 (ACC)，它是 PCB 图像上轮廓的表示。正如以下将要解释的，将以比由系统 20 摄取的 PCB 图像的分辨率更大的分辨率计算轮廓的位置。步骤 204 概括在图 4 中。

在子步骤 204a(图 4)，校正 PCB 图像的光度 (luminosity)。更具体来说，最好使用以下方程式使图像的光度(像素亮度)规格化：

$$\text{lum} = 256 \times (\text{lum} - \text{LUM_MIN}) / (\text{LUM_MAX} - \text{LUM_MIN}) \quad (1)$$

其中 lum 是图像的每一像素的光度;

LUM_MIN 是图像的最黑暗区域中的平均光度; 以及

LUM-MAX 是图像的最明亮区域中的平均光度。

以上的方程式允许像素的强度分布在 LUM_MIN 与 LUM_MAX 之间, 以便覆盖所有 256 灰度级。

显然, 以上方程式可以根据图像的灰度级数校正。也可以使用其它规格化方程式。

虽然子步骤 204a 是可选的, 但发现校正光度对于增加确定阈值的子步骤(204d)的效率是有益的。

应当相信理解确定最黑暗和最明亮区域的算法是在业内专业人员的知识范围之内。

可选的子步骤 204b 中, PCB 图像被过滤, 以降低图像数字化所引起的噪声。以下高斯滤波器最好施加到 3 乘 3 像素窗口上(八连接窗口):

121

242

121

高斯滤波器可施加到具有其它尺寸的像素窗口。也可使用其它类型的滤波器。

在子步骤 204c, 使用通常的 Prewitt 方法在八连接窗口上计算数值梯度。另外, 也可以使用其它的方法。梯度的强度存储在一新的图像中。由于 Prewitt 方法被认为在业内是熟知的, 在不再详述。

然后比较梯度图像与预定的阈值, 以识别图像中光度变化迅速的区域中的像素(步骤 204d)。这些区域通常对应于图像中重叠轮廓的像素轨迹。

在子步骤 204e, 通过对在子步骤 204d 识别出的像素轨迹的骨架化而计算边缘的第一近似。实际上, 图像上每一边缘对应于一串连接的像素, 每一像素在八连接窗口中具有至少一个邻域。骨架化处理过程允许像素串呈现出具有一个像素宽度的轨迹。该结果最好被保存在具有与原始图像相

同尺寸但只有二值，即对于属于边缘的像素为 1，而对其它像素为 0 的图像文件(ACC_{image})中。

子步骤 204f 中，按形成 ACC 的像素的坐标列出 ACC_{image} 像素的每一串。如以下将要说明的，这些坐标只是实际坐标的第一估计。

更精确来说，该串遵循反时针方向，且每一连续的像素的坐标保存在文件中：ACC。因而，ACC 包含一系列矢量，每一个表示着图像轮廓之一的坐标。

然后逐像素分析每一轮廓，以便通过二阶导数上的零估计通路的位置，已发现该导数是光度转折点坐标的可靠指示。认为对应的像素是轮廓的一部分，因为光度外形的转折点是两种材料之间过度的指示。

更精确地，该处理过程首先在于在每个像素找出垂直于轮廓的方向的估计。为实现这一点，首先作一条线拟合在当前像素位置，包含在同一轮廓上之前的 2 个像素和之后的 2 个像素。拟合的线指示在该像素位置轮廓切线的角度。增加 90 度就获得垂直方向。然后在作为由系统 20 最初摄取的原始 PCB 图像中，抽取沿该方向并穿过当前像素的光度外形(profile)。对光度外形计算二阶导数。二阶导数的零通路对应于光度外形的转折点。这点的坐标保存在 ACC 中，代替坐标的第一估计。

当然，也可以使用其它数值方法找出在每一像素垂直于轮廓的方向的估计。

相信考虑从受检查的 PCB 数字图像抽取轮廓的其它方法，应是业内专业人员知识范围之内的事情。

正如在步骤 208 将更为详细说明的，通过比较保存在 ACC 中及 ACC_{image} 中的轮廓位置与所设计的轮廓位置，检测出受检查的 PCB 上的异常。实际上，如果其位置不同于其设计的位置，则轮廓或轮廓的一部分在 ACC 上是可怀疑的。

然后在方法 200(图 3)的步骤 206 中，提供受检查的 PCB 计算机模型。计算机模型最好包含组件轮廓每一层的多边形形式的矢量模型。除了作为轮廓的适当的表示，已发现多边形对于与 ACC 比较的有效的模型，这将

在以下说明。

PCB 传统的 Gerber 模型最好转换为矢量模型，其中每一层的每一组件表示为一多边形。当然，如果把弧形边缘以短角顶表示，即使包含圆弧的元素也可以表示为适当的矢量模型。另外，如果没有 PCB 的 Gerber 模型，也可使用 PCB 的其它多层模型以抽取相关的信息。

在比较 PCB 的矢量模型与 ACC 及 ACC_{image} 之前，可以对矢量模型进行变换，以便使该模型与图像对准。当然，PCB 制造期间，各层通常是由不同的机器在不同时刻施加的，各层的接正可能稍微不同。这时最好有 PCB 的多层模型，并根据受检查的 PCB 重新校准模型的各层，这样可产生较好的 PCB 模型。

对于每一层，选择预定的点作为校准处理的基准。从表示层 32 和 PCB 图像 34 的模型的图 5 能够较好看出，已经发现最好选择水平 36 和垂直 38 线段的交叉点 40 作为基准点。显然，没有被来自另一层的元件覆盖的层的区域提供了更可靠的基准点。应当注意，由于 PCB 图像 34 包含所有的层，PCB 图像 34 包含诸如组件 42 这样的不是出现在单独一层模型 32 上的组件。

一旦建立了基准点的列表，就能够在一层的每一点上施加以下的变换，以便根据 PCB 图像上的基准点校准该点：

$$u = \frac{a_u \times x + b_u \times y + c_u}{dd \times x + ee \times y + 1} \quad (2)$$

$$v = \frac{a_v \times x + b_v \times y + c_v}{dd \times x + ee \times y + 1} \quad (3)$$

其中 (u, v) 是 PCB 图像中的坐标，

(x, y) 是层模型中的坐标，

及 a_u, b_u, c_u, dd 和 ee 是允许补偿在 PCB 图像和模型之间旋转、平移、缩放及透视的系数。这些系数对每一层可能是不同的。

最好在检查 PCB 之前选择基准点，以将计算机延时最小化。

一旦模型的所有层都被校准，它们所有都被叠加。

由于模型所有的层包含多个多边形，2 层的叠加就在于把第一层的每一多边形与来自第二层的所有多边形叠加。例如如果模型包含第三层，则来自第三层的多边形与来自前两层叠加结果的新的多边形叠加。

这一叠加过程示于图 6a 到 6c。

图 6a 示出包含三个逐层叠加的多边形 P1、P2 和 P3 的三层模型。叠加在于把多边形 P3(Pv3)叠加到多边形 P1 和 P2 的叠加结果之上。

P2 在 P1 上的叠加产生：

$$Pv1 = P1 - P2 \quad (4)$$

$$Pv2 = P2 \quad (5)$$

在 P2 叠加到 P1 上之后，P2 保持原样不动，同时 P1 必须减去 P2 的重叠的部分。P3 在 P2 和 P1 的叠加上的叠加产生(参见图 6b)：

$$Pv1 = P1 - P2 - P3 \quad (6)$$

$$Pv2 = P2 - P3 \quad (7)$$

$$Pv3 = P3 \quad (8)$$

在某些情形下，2 个多边形的减去会产生孤立的多边形。Pv1 和 Pv2 可被看作是多边形列表。当按层有多于一个的多边形时，P1、P2 和 P3 也可被看作是多边形列表。

叠加的过程允许指出重叠的区域。在叠加新的层之后重叠区域上的多边形必须被更新。

在图 6c 上明显可见，O21 是在与 P3 叠加之后 P1 和 P2 求交所得的结果的多边形。

又由于两个多边形或两个多边形的相交或两个多边形之间的差可能产生一个以上的多边形，O21 这里必须解释为多边形列表。如果层 1 和 2 之间没有多边形重叠，该列表可以不包含多边形。

如以下将要描述的那样，上述的叠加过程还允许保持狭窄区域的踪迹。狭窄区域的一个例子是图 7a 中的区域 T31。区域 T31 是如下被抽取的。

从图 7b 可见, 多边形 P1 和 P3 被膨胀而产生 $D(Pv1)$ 和 $D(P3)$ 。这时发现它们之间的相交产生显然包含被抽取的地区的多边形 $D31$ 。多边形 $O31$ 被膨胀而产生从 $D31$ 抽取的 $D(O31)$, 产生狭窄区域 $T31$ (参见图 7c)。

图 7a 到图 7c 所示的操作概括为以下的公式:

$$T31 = (D(P3) \wedge D(Pv1)) - D(O31) \quad (9).$$

由于以下的原因, 这里 $T31$ 又可看作是多边形列表: 两个多边形的相交可以产生多于一个的多边形, 并且 $D(O31)$ 也可以产生一个以上的多边形。

总之, 层的叠加产生以下三种结果:

$$Pv_{k+1} = (Pv_{k+1} - P_{k+1}) \cup P_{k+1} \quad (10),$$

可见多边形的列表;

$$O_{k+1} = (O_k - P_{k+1}) \cup (P_{k+1} \cap Pv_k) \quad (11),$$

来自重叠区域的多边形列表; 以及

$$T_{k+1} = (T_k - P_{k+1}) \cup ((D(P_{k+1}) \cap D(Pv_k)) - D(P_{k+1} \cap Pv_k)) \quad (12)$$

来自狭窄区域的多边形列表,

其中

$$T_1 = D(P_1) \cap D(P_1) \quad (13); \text{ 以及}$$

O_1 是零(空)。

该过程是迭代的, 即每一个新的层被叠加到通过以前层的叠加所得的 Pv_k, O_k, T_k 。K 指示已叠加的层的数目。

多边形列表上进行的运算是含并 (union)、相交 (intersection)、减 (subtraction) 和膨胀 (dilatation)。合并 (\cup) 在于把两个多边形列表组合成单个列表。两个列表相交 (\cap) 在于来自第一列表的每一多边形与来自第二个列表的每一多边形相交。类似地, 当两个列表减时, 对第一个列表的每一个多边形减去第二个列表的所有多边形。

可见的多边形 Pv_k 列表描述了应当在 PCB 图像上出现的所有轮廓。然而, 上述的列表 Pv_k 并不是适合于与 ACC 快速比较的格式。实际上, ACC 中的每一点应当与来自 Pv_k 的至少一个多边形段对应, 以保证快速比较。

转换列表 Pv_k 以便以它同下一个顶点形成的线段代替每一多边形顶点

产生特征轮廓图(ACC)。这样 ACC 的线段被划分为多边形列表, 包含所有它们的线段而不是整个顶点的描述, 如同在 Pv_k 列表那样。更具体地, ACC 中的每一线段由一直线的方程式、直线的端点坐标, 及该线段允许的距离阈值表征。此外, ACC 对所有的多边形包含接近封闭该多边形的矩形的表征。这矩形被膨胀到保证 ACC 中没有被封闭在该矩形中的每一点不对应于多边形的线段。

异常的检测

一旦生成 ACC, 方法 200(参见图 3)的步骤 208 在于通过比较 ACC 中的轮廓与现在为 ACC 形式的计算机模型而检测 PCB 上的异常。更具体来说, 异常最好在以下两种图中被指出和表征:

ECC, 超越轮廓图 (Excess Contour Chart), 及
MCC, 遗漏轮廓图 (Missing Contour Chart)

按超越检测轮廓的第一步是要找到对应于 ACC 中每一点的 ACC 中的线段。如果一点不充分靠近 ACC 的任何线段, 它被看作是处于超越并被保存在 ECC 中。计算的距离与预定的阈值比较。ECC 的格式与 ACC 相同; 它是轮廓的一个列表, 每一轮廓由连接的点的一个列表表征。

更具体来说, 对于 ACC 中每一点:

- ACC 被扫描直到其列出的矩形之一围住该点;
- 对应的多边形被扫描直到找到该点可以对应的点的线段;
- 如果没有线段对应于该点, 扫描另一多边形;
- 如果检测到一线段, 处理过程继续到一个点, 但是最好以相同的线段开始。实际上, 由于形成多边形的点是相继的, 下一个点对应于相同线段的概率很大;
- 不对应于任何线段的点在 ECC 中被指出。如果该点是连续的一系列的点的一部分, 它则在 ECC 中作为这一系列的部分被记录。

对于 ACC 的每一轮廓, 在 ACC 和 ACC 中各点最好以相同的方向被扫描。

当然, 可以使用其它的方法检测 ACC 中的超越轮廓。

现在更为详细地说明遗漏轮廓的检测。

遗漏轮廓的检测也意味着对 CCC 中轮廓位置偏离太多的轮廓的检测。对于高密度的电子电路，轮廓的允许偏离带相对是狭窄的。但是由于 CCC 是矢量的形式，其分辨率不限于图像的分辨率。类似地，ACC 的分辨率是图像分辨率的 7 到 8 倍，因为在以子像素分辨率检测轮廓之后 ACC 已被编译过。这一点是有益的，因为这允许即使当最大允许偏离狭窄时也能检测放错的轮廓。

应当注意，遗漏轮廓的检测就精密度来说是相对没有约束的。遗漏轮廓检测过程的目的只是要验证，CCC 上的轮廓是否在 PCB 的图像上没有。其实，出现在 PCB 图像上但相对于计算机模型放错的轮廓，将通过超越轮廓过程的检测而检测，并在 ECC 上指出。因而，遗漏轮廓的检测能够以图像的分辨率实现，这样就能够通过比较图像进行。

为了获得 MCC，CCC 上的轮廓首先被绘制到二元图像上，即与 ACC_{image} 有相同的分辨率的 CCC_{image} 。比较 CCC_{image} 与 ACC_{image} ，产生 MCC，这概括为以下的方程式：

$$MCC = D(ACC_{image}) \otimes CCC_{image} \cap CCC_{image} \quad (15)$$

这里 $D(ACC_{image})$ 是膨胀之后的 ACC_{image} 。计算 MCC 的处理过程还总结在图 8 中。

MCC 和 ECC 包括对应于 PCB 图像中异常的区域描述。这些区域在方法 200 的步骤 210 中被检查(参见图 3)，以检测缺陷。

异常的检查

由于异常能够在 ECC 中产生一个以上的轮廓，但是 ECC 的每一个轮廓只对应于一个异常，故 ECC 的每个轮廓最好看作是独立的异常并分别被检查。当 ECC 中所有的轮廓已经被检查时，检查过程就结束。被发现超过预定的制造标准的 ECC 中的轮廓最好被在文件中指明并表征。

在不背离本发明的精神和性质之下该文件可以采取许多形式，例如可以包括缺陷的位置和尺寸，以及由这种缺陷被弯曲的 PCB 的描述。

能够在 PCB 上发现的大部分缺陷对应于在 PCB 上印刷的元件的错误

的宽度，或这些元件之间不正确的间隔。当元件的宽度或两个元件之间的距离与设计规范差异太大时，其它的异常，诸如污染物，沟槽，及毛刺可能成为问题。

当以下两个之一超过制造标准时，异常中的缺陷被检测：其上有异常的元件的宽度，及如果异常不在元件上，两个相邻元件之间的距离。如以下所述，这两个条件是独立地被检验的。

这两个条件的验证很类似。一般来说，在两种情形下，可在异常的邻域中找到的 CCC 的所有线段都存储在一列表形式的计算机文件中，这里称之为邻域线段列表(NSL)。对于异常附近的每一线段，找到与第一线段相对的 NSL 的每一其它线段。验证第一线段与所有这些相对的线段之间的距离，以便评估该距离是否在设计或制造标准之内。那些标准显然是预先确定的并预先存储在计算机文件中，以便可由计算机 32 访问。

如以下将解释的那样，两个线段之间的距离的验证和它们之间的宽度的验证的区别仅在于怎么考虑两个相对的线段的定义。

如图 9 中所见，轮廓的每一线段，诸如线段 44，可被解释为 2 个半平面 46 和 48 的边界。在线段 44 的与轮廓 50 相对的一侧的平面 46，是在线段 44 的“外”侧。在线段 44 与轮廓 50 相同的一侧的平面 48，被称为线段 44 的“材料”（material）侧。当验证线段 44 与“相对”（opposed，对立）线段 52 之间的距离时，材料侧设置为“有效”（active）的。

当段 44 与 56 之间的轨迹 54 的宽度被检验时，外侧是有效的。两个线段，如果它们彼此面对并如果它们在相同的有效平面内，则被配对分析。在图 9 的例子中，线段 52，58 和 60 可选地与线段 44 配对。

检测过程使用的阈值最好以百分数表示，这样提供统一的方法，对于每一新的电路类型，不需要对宽度和距离新的阈值进行编程。当然，百分数是根据设计值表示的。例如，当如果两个线段之间的距离小于设计的两倍将成为问题时，则阈值被设置为 50%。

显然，基于绝对值的阈值也可以用来检验线段之间的宽度。

可以选择不同的方法确定两个线段之间的间隔，这取决于线段的取向

及是否两个线段之间有一个或多个异常。

如果是在两个平行的线段之间检测到异常，这两个线段之间的间隔就被简单地定义为这两个线段之间的距离。

如果两个线段 62 和 64 不平行(参见图 10)，该间隔被任意定义为它们之间的角度。这种情形下，一个点到基准线段 62 的距离被如下测量：测量穿过两个线段之间的交叉点 68 与该点的直线 66。该点与基准线段之间的距离定义为直线 66 与基准线段 62 之间的角度 72。

在两种情形下，都要计算异常的每一点与基准线段(相对的线段)之间的距离。记录的最大和最小距离定义了包含异常的区间(interval)。在异常存在时，两个线段之间的有效间隔是两个线段之间的间隔减去该区间。规范化的间隔是有效间隔与两个线段之间的间隔的比率。

图 11 示出，当在两个线段 80 和 82 之间检测出多个异常 74、76 和 78 时，间隔计算的一个例子。

这种情形下，对于每一异常计算区间。如果一个以上的异常的区间重叠(例如参见异常 74 和 76)，这些区间被再分组，以形成单个的累积区间(a-c)。累积的(a-c)和非累积的(d-e)区间被相加而产生有效的间隔。再次，规范化的间隔是两个线段之间有效间隔与实际间隔之间的比率。

间隔和宽度的检验只是在异常附近的 PCB 图像区域中进行。以下的描述说明了这些区域是怎样被确定的。这些区域这里将称为异常分析区域(AAR)。

现在转向图 12，首先作为围住异常 86 的矩形 84 确定 AAR。如果已知形成异常 86 的轮廓的点的坐标，可以使用熟知的算法确定这种矩形。

然后通过由以下定义的膨胀因子(DF)膨胀矩形 84：

$$DF = \frac{1 + \% \text{ res}}{1 - \% \text{ res}} \quad (14)$$

这里 % res 是规范化的间隔或规范化的宽度上的预定阈值，以产生有效 AAR 88。

在矩形 88 围住的区域之外的元素最好不考虑作进一步分析，因为预先

假设它们不会与异常 86 的组合引起缺陷。然而,对按上述配对并在 AAR 88 中的所有的线段 89-91 和 89-91'要检查缺陷。

当在步骤 208(参见图 3)检测到异常并保存在 ECC 中时,最好对每一异常确定 AAR。

如果第二个异常的 AAR 与第一个异常的 AAR 重叠,则设想这两个异常的组合能够引起缺陷。当检查异常的缺陷时,当作似乎其 AAR 没有重叠其它异常的 AAR 那样进行检验。如果是这样,则生成一新的围住所有涉及到的异常的矩形。新的矩形被膨胀而产生一新的 AAR,且重复这最后的处理直到新的 AAR 不重叠任何其它的 AAR 为止。在该 AAR 中,再对所有的线段对检验其缺陷。

由于 ARR 可能包含一个以上的异常或在大小上可能是相对重要的,可以进行测试以检验是否每一异常都位于两个线段之间的中间间隔中。最好实现以下两个规则进行该测试。

第一个条件是说,定义 ARR 的矩形应当重叠两个配对的线段的两个有效半平面。

第二个条件意思是,两个线段应当充分靠近。更具体地,使用以下的准则:第一和第二线段之间的异常的距离之和应当小于它们之间的最大距离的一半。两个线段之间的最大距离被保存并在这里将称为 d_{segmax} 。虽然已经发现这种准则可产生良好的结果,其它的准则也可以用来确定两个线段是否充分靠近来被检查。

当第一和第二个条件满足时,认为异常在两个线段之间的中间间隔内,并认为相对的一对线段被检查缺陷。

现在转向图 13a、13b 和 13c,将说明方法 300,该方法确定异常是否由于违反允许的宽度或间隔而引起缺陷。

诸如以下将要说明的那样,在狭窄空间区域和重叠区域的情形下,将要另外进行两个附加的检查。

最好把异常划分为两个组:可能引起宽度缺陷的元件上的异常,以及不在元件上但可能引起间隔缺陷的异常。

元件上的异常相当容易选择。实际上，ECC 上每一轮廓对应于一系列连续的点。与 CCC 轮廓重叠的 ECC 轮廓这时一分为二。由此，如果 ECC 的轮廓的任何点在元件上，则认为轮廓的所有的点也在这一元件上。如果只要有 ECC 轮廓的一个点不在元件上，则不认为该轮廓在元件上。因而只检验 ECC 上每一轮廓的一个点即允许区分在两个组中的异常，这里称为 $ECC_{spacing}$ 和 ECC_{width} 。方法 300 在两个组上被独立使用。

由于方法 300 在两个组每一个上的应用之间的差别很相似，并为了简介的目的，将只是参照整个 ECC 说明方法 300。

第一步骤 302 是要提取列在 ECC_{width} 中下一个没有检查的异常。显然，第一次执行步骤 302，下一个异常是第一个列出的异常。现在这一异常将称为当前异常(CA)。

步骤 304 是找到 ECC 中的所有异常，该 ECC 具有重叠 CA 的 AAR 的其对应的 AAR。

在步骤 306，这些异常在 ECC 列表中被标记为被检查，并被放置在当前异常列表(CAL)中。按以上所讨论的那样 ARR 被重新评估(步骤 306)，并重复步骤 304 和 306 直到没有未检查的异常使其 ARR 重叠新的 ARR 为止。

在步骤 308，被封闭在 AAR 中的 CCC 中所有的轮廓被列在新的 NSL 中。

在步骤 310，检验 NSL 中所有的线段被标记为已分析。如果它们不是这样，则在 NSL 中下一个未被分析的线段成为当前线段(CS)，CS 在 NSL 中被标记为被分析(步骤 311)，且处理过程继续进行到步骤 316。

如果这时 NSL 中所的线段被标记为被分析，在步骤 312，对 ECC 中所有的异常已经被标记为被检查的情形进行检验。如果不是，对下一个异常重复步骤 302。如果是，则异常的检查结束并停止整个处理过程(步骤 314)。

步骤 316 在于使用上述准则验证 NSL 的下一个未分析线段是否相对于 CS。如果是这种情形，则处理过程继续进行步骤 320。如果不是这种情形，

且已达到 NSL 的末端(318)，则重复步骤 310。否则，重复步骤 316。

在步骤 320，考虑到 CS 与相对于 CS 的线段，如上所述计算 dsegmax。

在步骤 322，进行检验看 CAL 是否包含另一个未检查的异常。如果是这种情形，CAL 中下一个未检查异常设置为 CA(步骤 324)，且处理过程继续进行步骤 326。否则该方法前进到步骤 330。

在步骤 326，如果 CA 的 ARR 没有重叠 CS 和与其相对的线段两者的有效半平面，则重复步骤 322。如果重叠，则方法前进到步骤 328。

在步骤 328，对于 CA 的每一点计算到 CS(MD1)与到相对线段(MD2)的最小距离，以及最大和最小间隔(MAS 和 MIS)。

在所有被检测的异常之中，确定在两个线段的内部空间中的那些异常(步骤 330)。测试如下：如果

$$MD1+MD2 > DESGMAX* \%RES$$

则不认为异常在两个线段内部空间中。从 CAL 中除去不在内部空间中的异常(步骤 332)。

如果 CAL 是空的，则处理过程返回步骤 310(步骤 334)。

在步骤 336 中，由每一异常占据的区域被保存在所有被占据区域(AOR)的列表中，该列表最好是区间的形式。AOR 列表包括对每一异常的 MAS 和 MIS。如果两个区域重叠，则它们的区间相加。

在步骤 338，如上所述计算规范化的间隔。实际上，要计算 CS 及其相对的线段之间的有效间隔。这等于两个线段之间的间隔减去 AOR 中的区间之和。规范化的间隔等于有效间隔对两个线段之间的实际间隔的比率。

如果规范化的间隔劣于值 %RES，则在步骤 340 检测到缺陷。

在步骤 342，对检测到的缺陷被表征并保存到缺陷列表中。更具体地，保存异常和两个线段(CS 和相对于 CS 的线段)。然后处理过程返回步骤 310。

狭窄空间区域的检查

由于以下两个原因，最好对来自狭窄区域的多边形 T_{k+1} 进行检查：

- 多层模型上它们的尺度可能太小；以及

- 即使模型上两个元件之间的内部空间正确，图像上对应的区域可能不正确。

一般来说，多边形 T_{k+1} 列表中两个相对的线段产生具有一宽度的内部空间。如果对应的宽度不在预定的阈值之内，则这些内部空间的某些对应于缺陷。

更具体来说，多层模型上太小的尺度可能指示着重要的层间校准间隙。这通过以预定的准则测试内部空间进行检测。例如，两种不同材料之间的间隔可能被检验，以保证其不劣于例如由 PCB 制造者所设置的最小间隔值。如果小于，则在两层之间检测出校准的问题。

有可能，根据方法 300 的异常的检测可能无法检测到 PCB 两个元件之间错误的间隔。

实际上，如以上所讨论，异常的检测是在模型中的轮廓与在 PCB 图像中检测出的轮廓之间预定的容差内进行的。然而，内部空间中允许的缺陷的最小尺度与内部空间的尺度有关。例如，如果关于轮廓位置的容差是模型上宽度的 25%，且区域的最小宽度是模型宽度的 50%，则按算法 300 将不能检测到异常。

现在转到图 14，第一异常 90 能够引起轮廓 91 在公差容限内偏差并达标称宽度 92 的 25%，且在相对的轮廓 93 附近的第二异常 94 也是如此。按方法 300 两个异常都不能检测到，因为它们都在公差容限 92 和 96 之内。然而，这两个异常的组合使内部空间变得狭窄到其标称宽度的 50%。然而，这就对应着一个缺陷。

如图 15 所示，以下的准则是根据本发明的方法设置的，以便检测狭窄空间(内部空间)区域的缺陷。

在步骤 400，如果计算的内部空间的宽度劣于例如由制造者规定的阈值，则检测到层的校准错误。

如果不是，且宽度劣于轮廓位置上的四倍公差容限(步骤 402)，则可能出现缺陷，并测量 PCB 图像上的内部空间宽度。

如果测量的宽度劣于模型上的宽度的 50%(步骤 404)，则元件之间的

区域(内部空间)不正确并检测到缺陷。

如图 16 和 17 所示,最好通过使多边形 100 逐步变窄(参见箭头 101)直到多边形的线段交叉(参见图 17 中 102),进行矢量模型上内部空间 98 的测量。这一过程称为侵蚀。

侵蚀率是内部空间 98 宽度的指示。例如,直到出现交叉 102 之前侵蚀率最好设置得高。然后,降低该比率来膨胀该多边形 100。然后再降低该比率重新开始侵蚀直到另一交叉出现。重复该处理过程直到侵蚀率等于或劣于所需的分辨率。该比率是以像素的分数表示的,因为如上所讨论,分辨率最好按子像素的数量级。

图像上内部空间的测量最好通过划分出内部空间的轮廓的抽取进行。应当注意到,这些轮廓不一定对应于相同的元件或不一定产生封闭的轮廓。然后轮廓被封闭以形成多边形的顶点。以上对于矢量模型中多边形的侵蚀所讨论的方法被用来测量轮廓的宽度。

在不背离本发明的精神之下,其它方法也可以用来测量矢量模型和图像这两者上的内部空间。

当然,也可以设置其它的阈值或准则来检测狭区域的中的缺陷。

重叠区域的检查

由于前节所述相同的两个原因,对通过方程式(11)获得的重叠区域进行检查。

然而在重叠区域的情形下,可能出现另一问题:一层的一个或多个轮廓可能被重叠层隐藏在 PCB 图像上面。这种情形下,存在两种可能性。

线段的一部分是可见的,允许进行上层的轮廓的外插。否则,只能检测到可见的轮廓。

由于重叠区域中缺陷的检测一般基于图 15 中所述相同的准则,故这里不再详述。

由狭窄区域或重叠区域引起的缺陷最好被表征并保存在文件中,供进一步分析。该文件可以是与用来存储和表征其它检测到的缺陷的相同的文件。

虽然以上就本发明优选实施例对本发明进行了说明，但在不背离所附权利要求中定义的主题发明的精神和本质之下，它可以被修改。

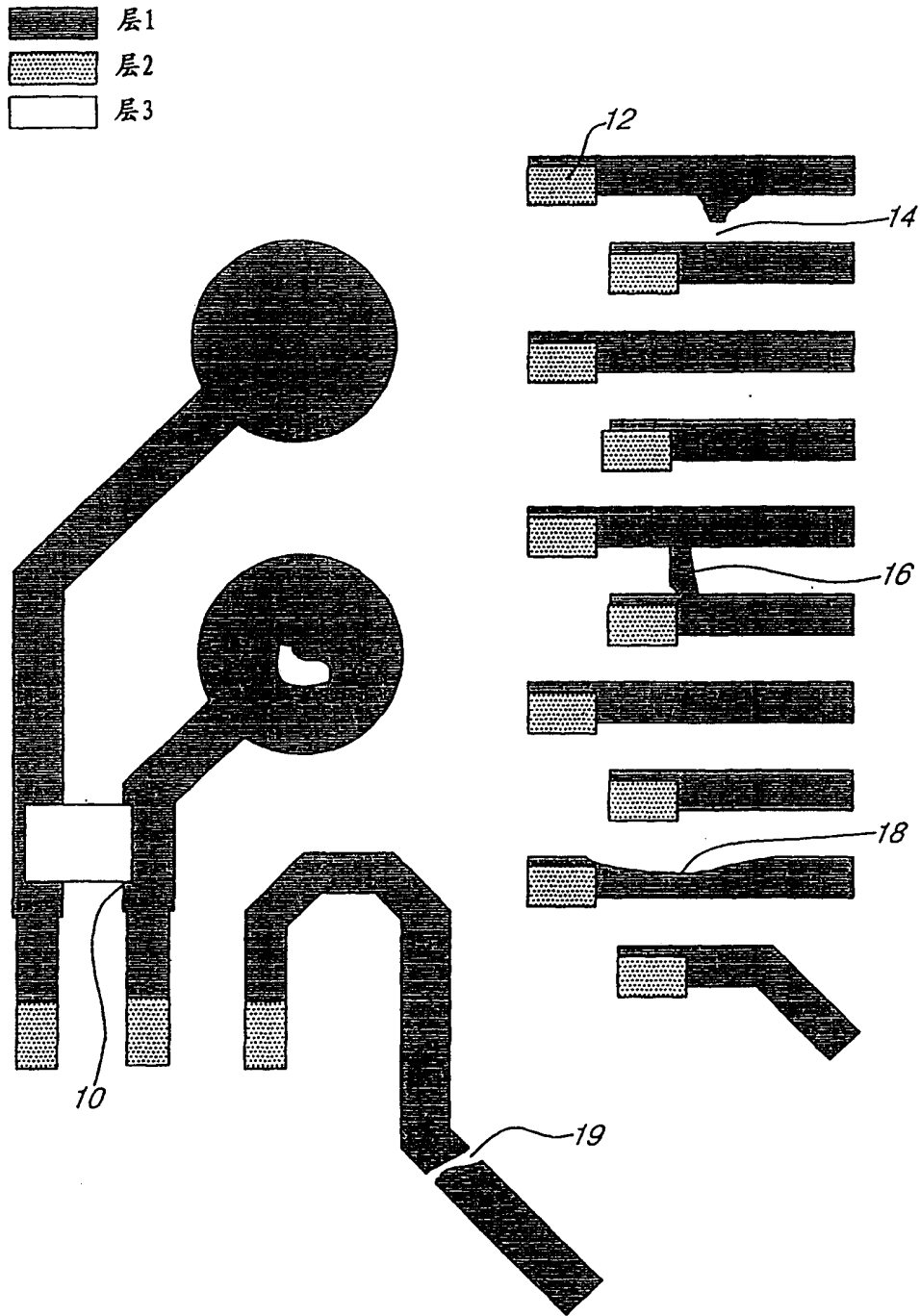


图1

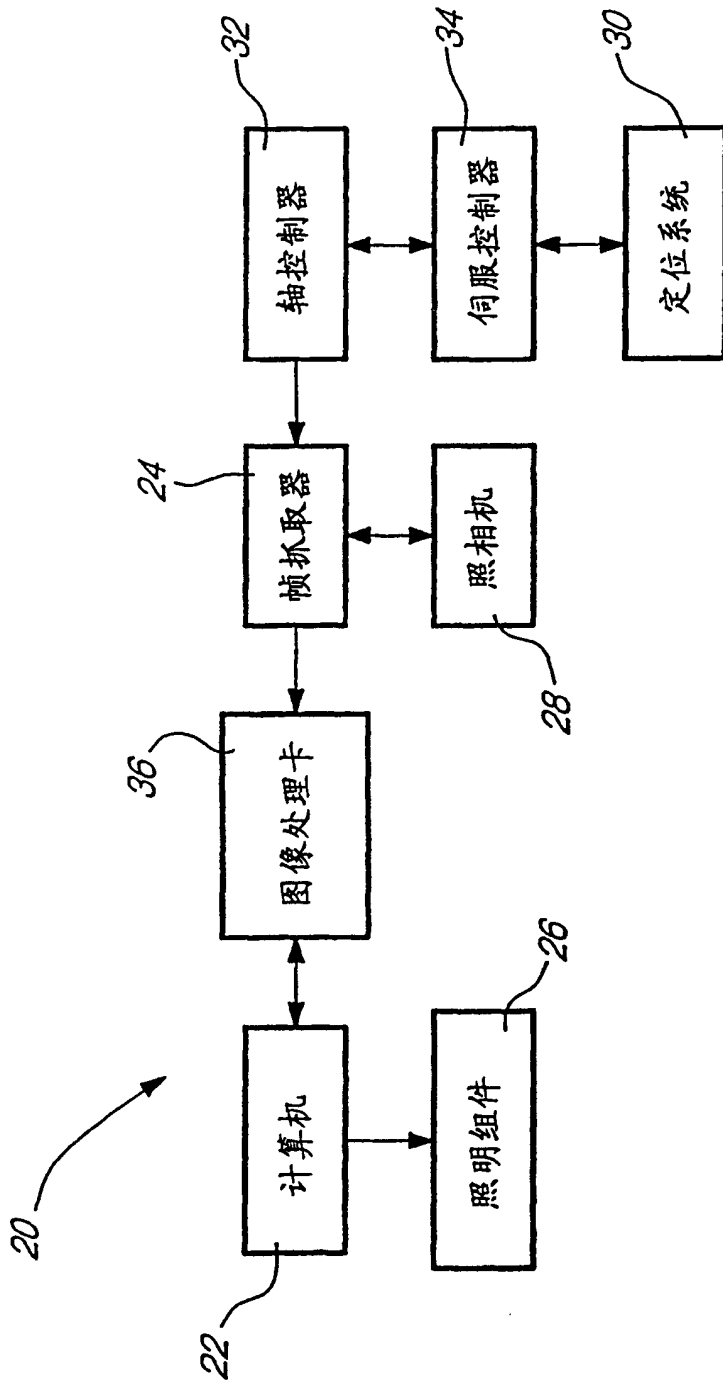


图2

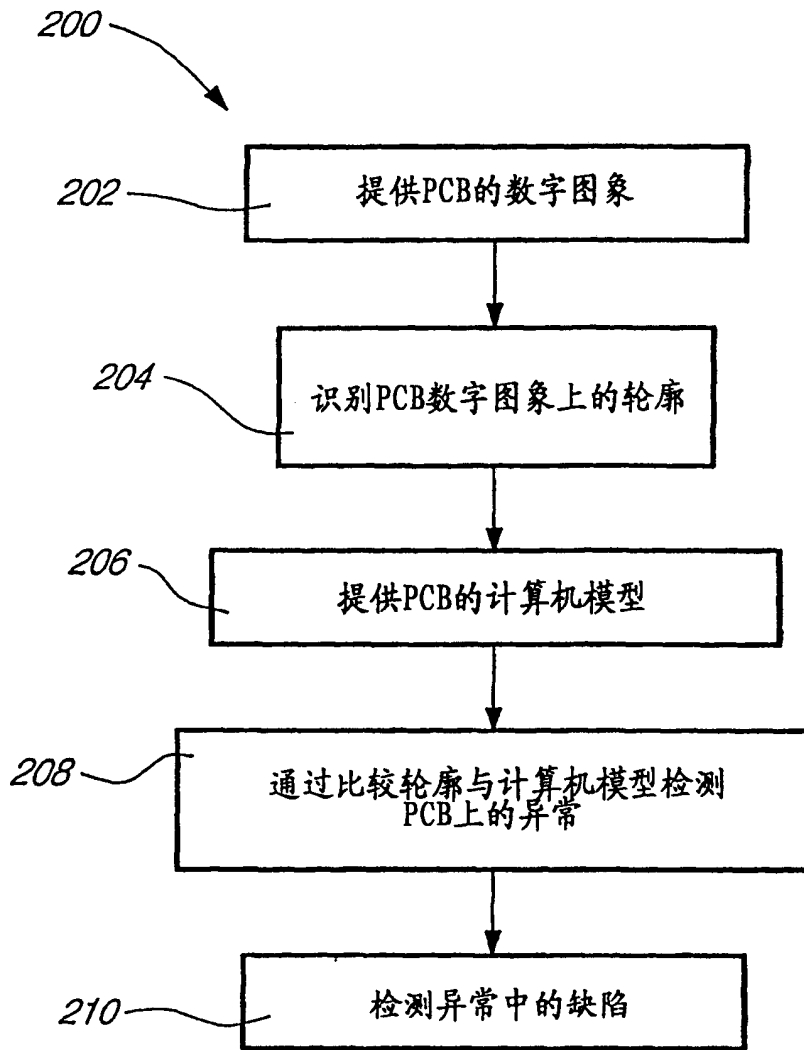


图3

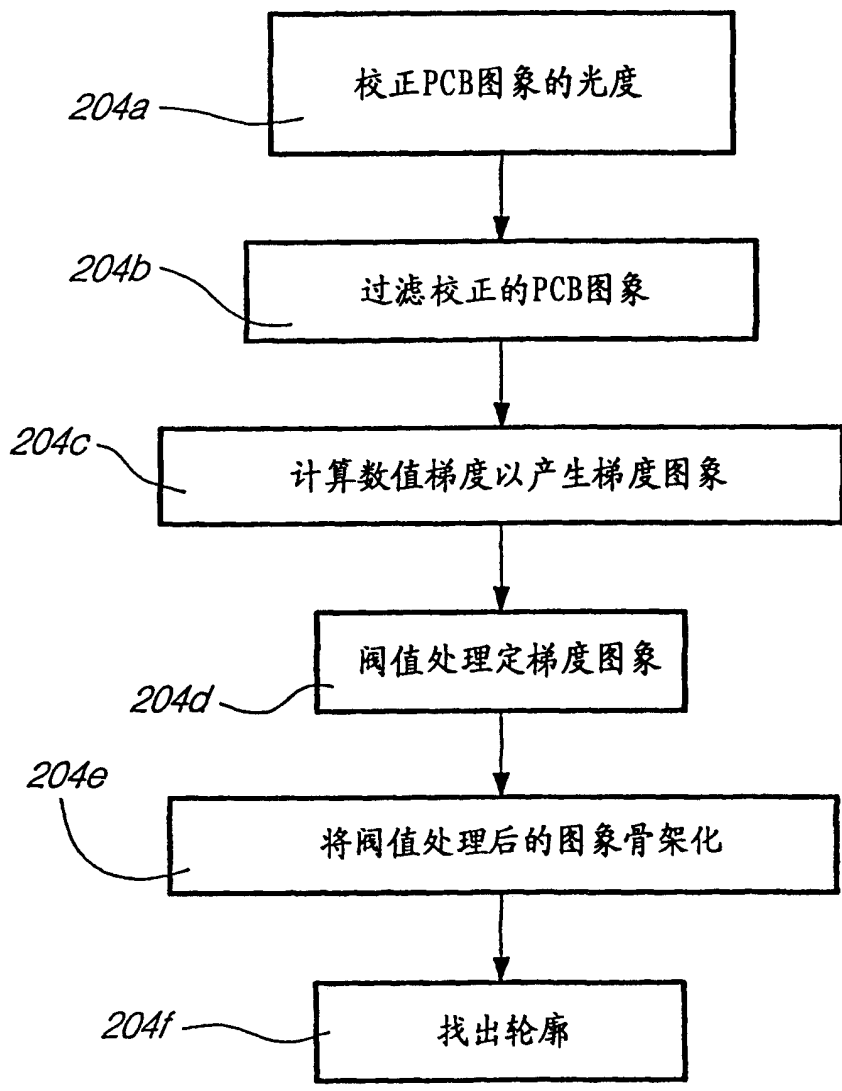


图4

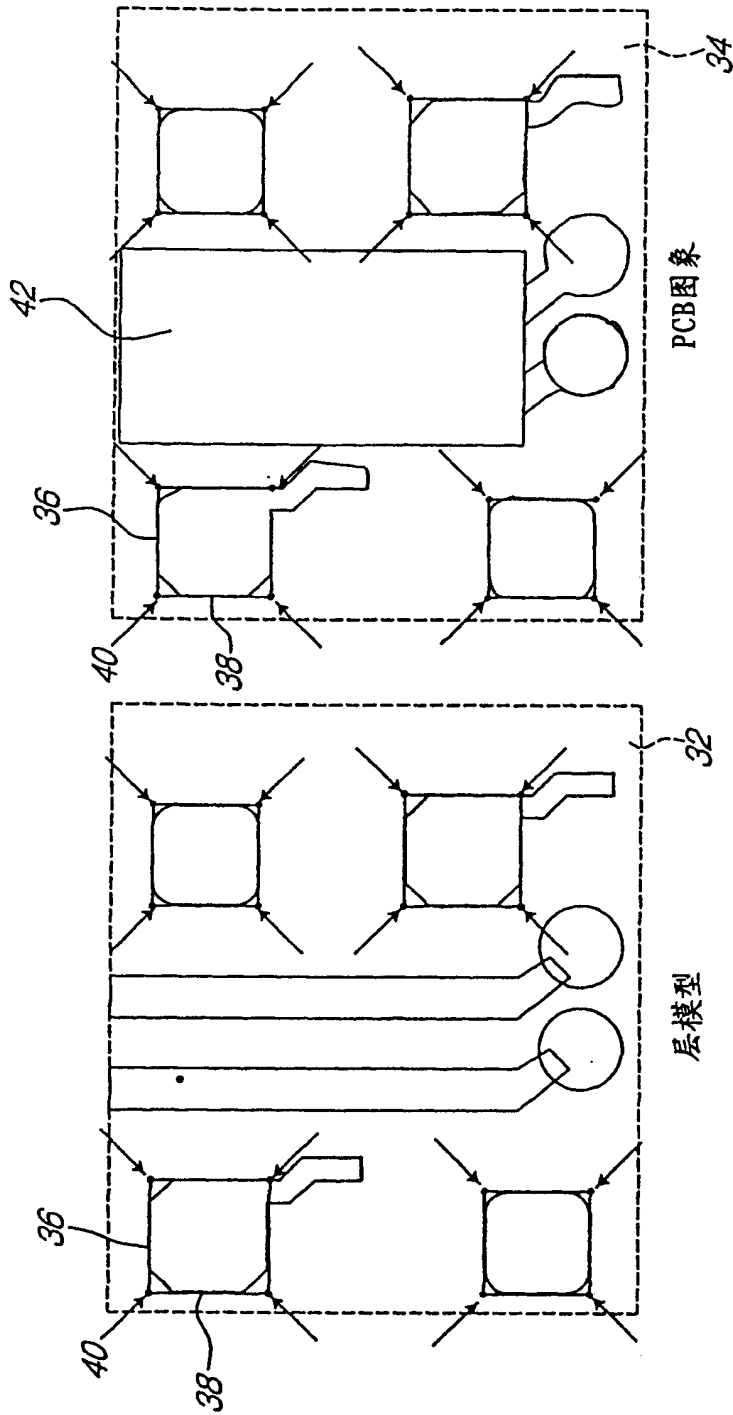


图5

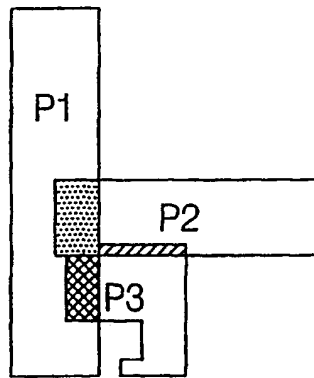


图 6a

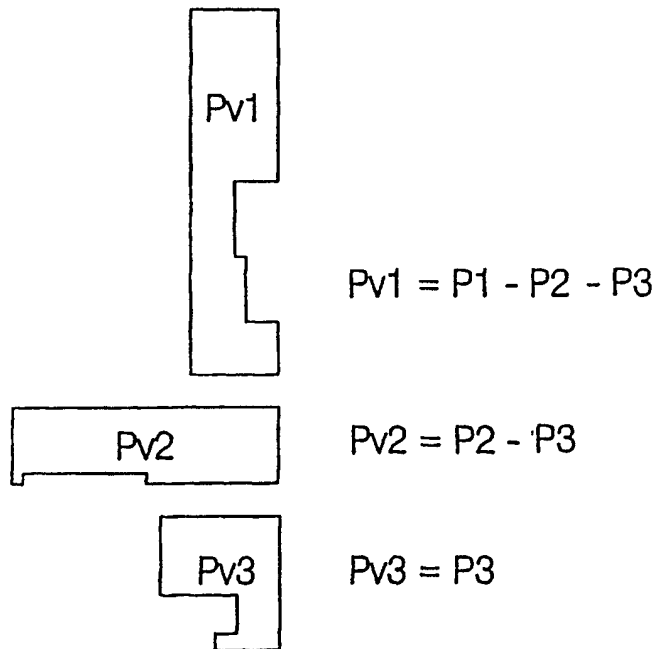


图 6b

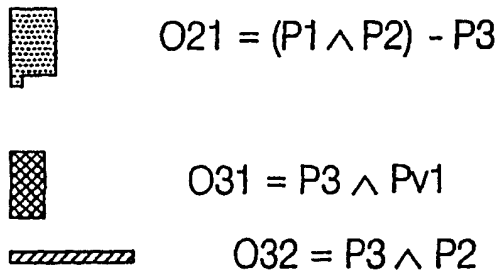


图 6c

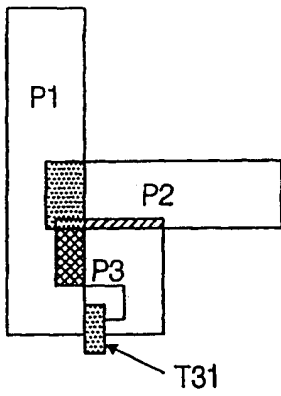


图 7a

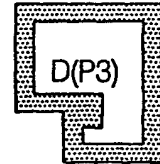


图 7b

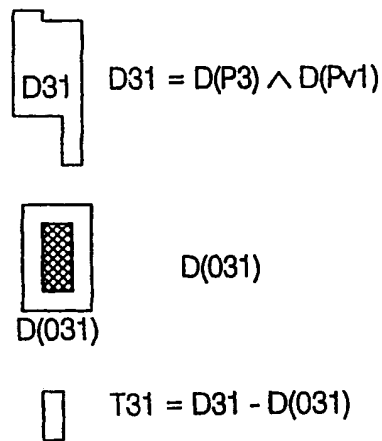


图 7c

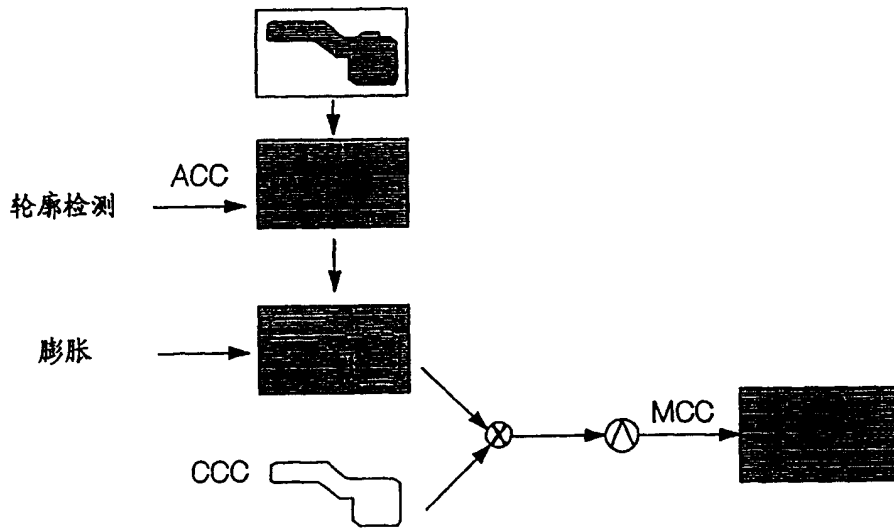


图8

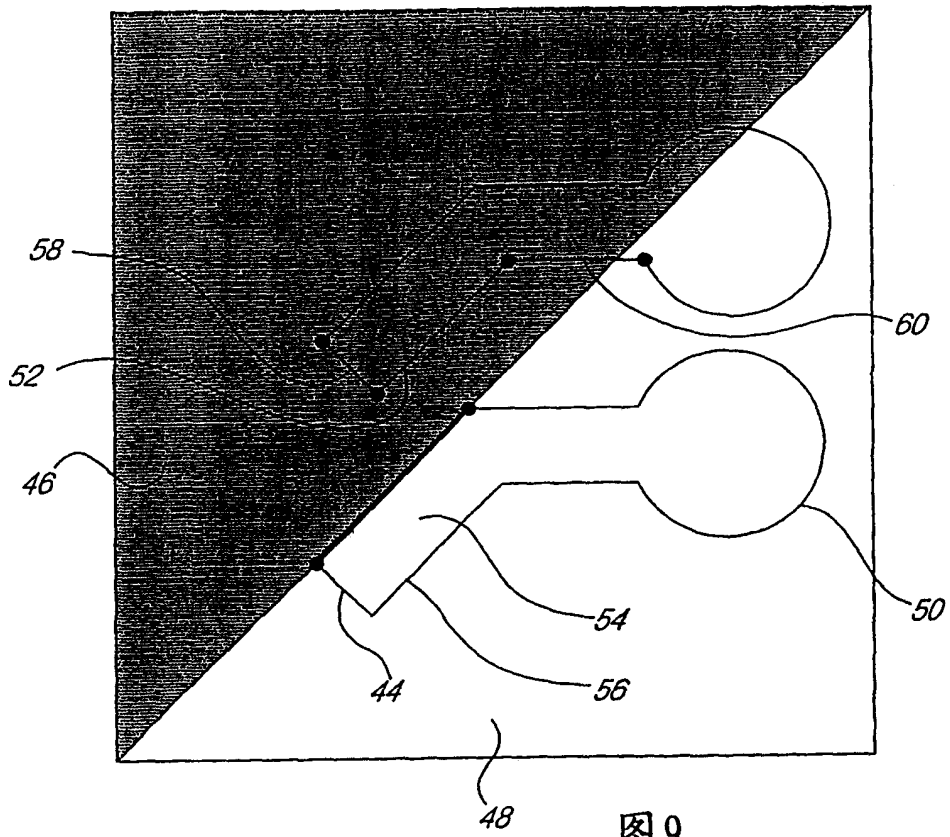


图9

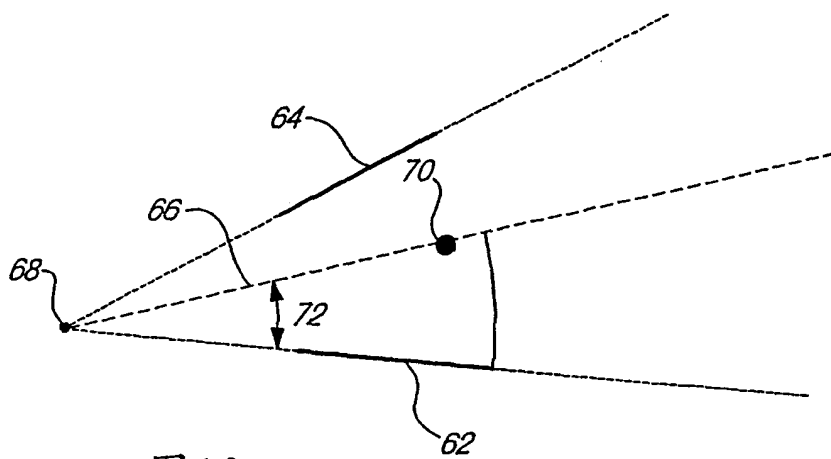


图10

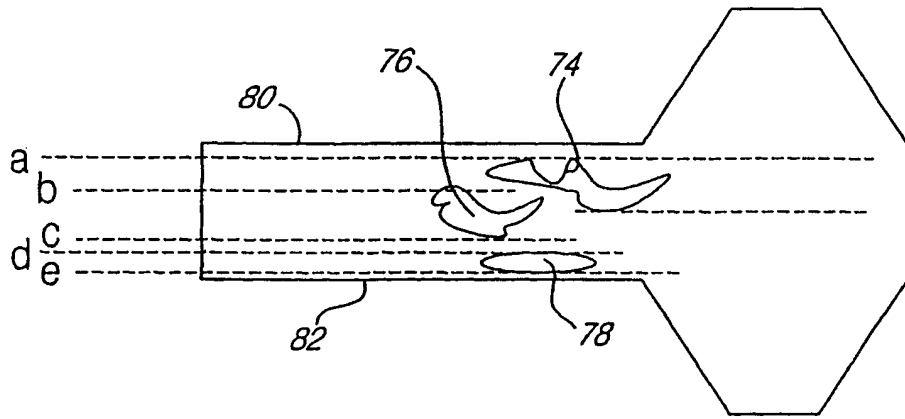


图11

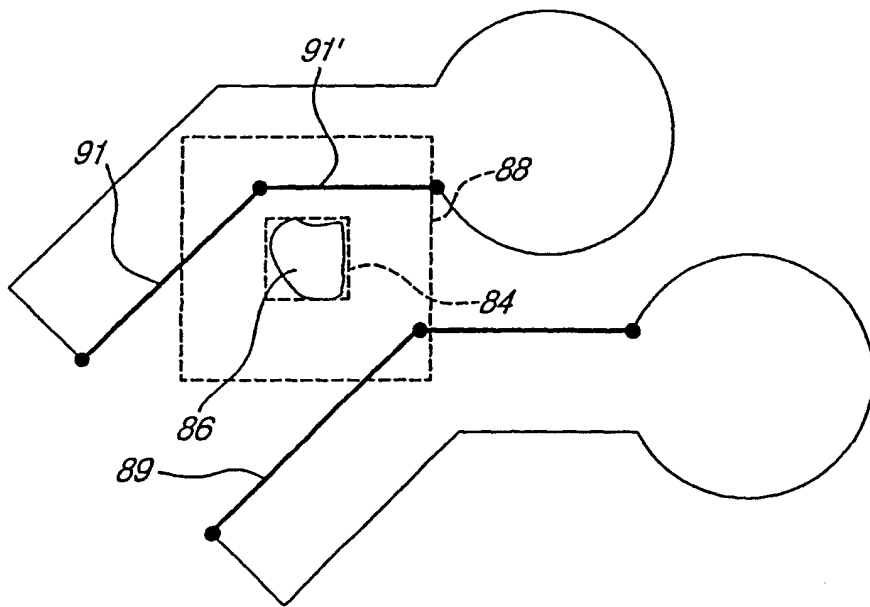


图12

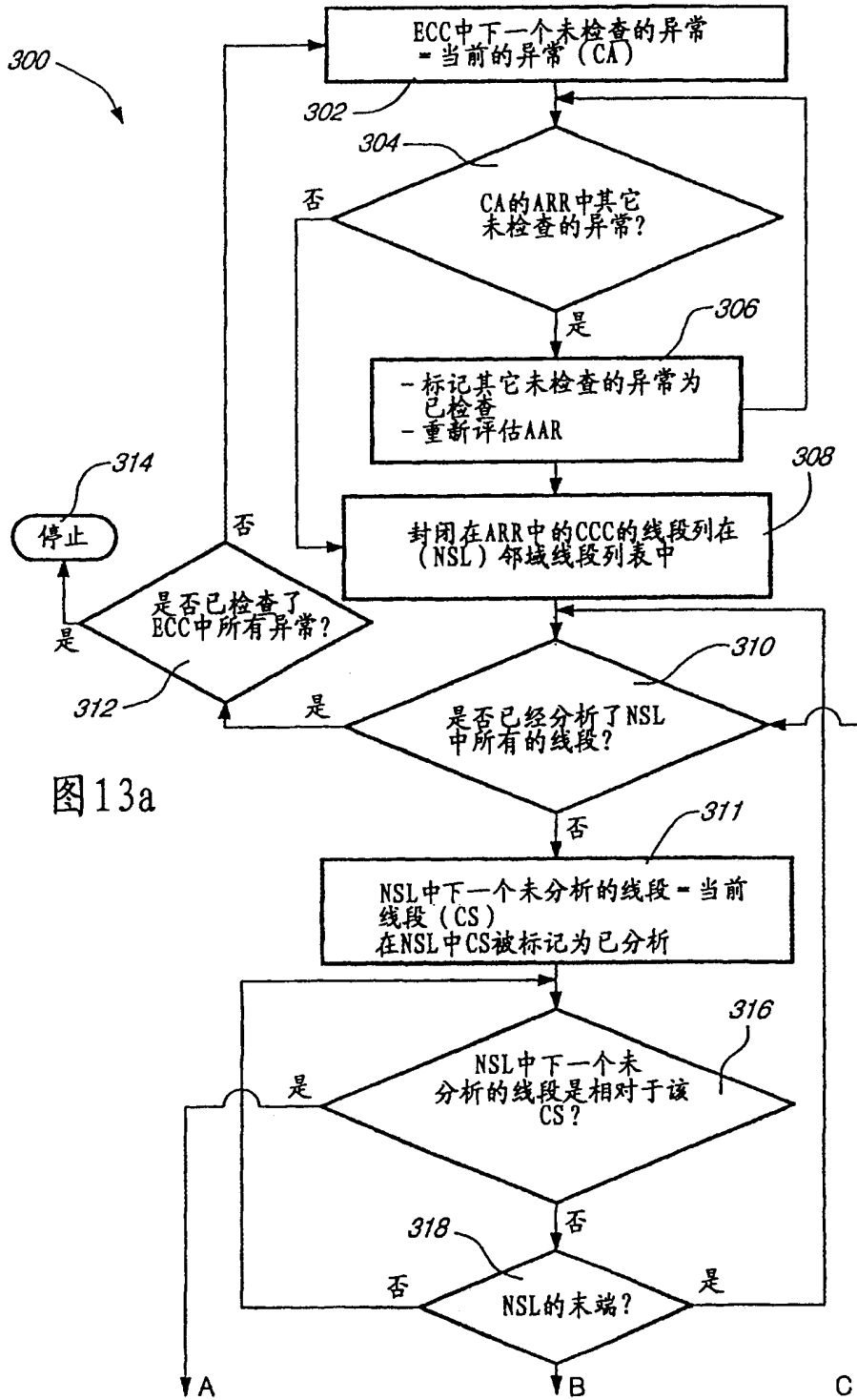
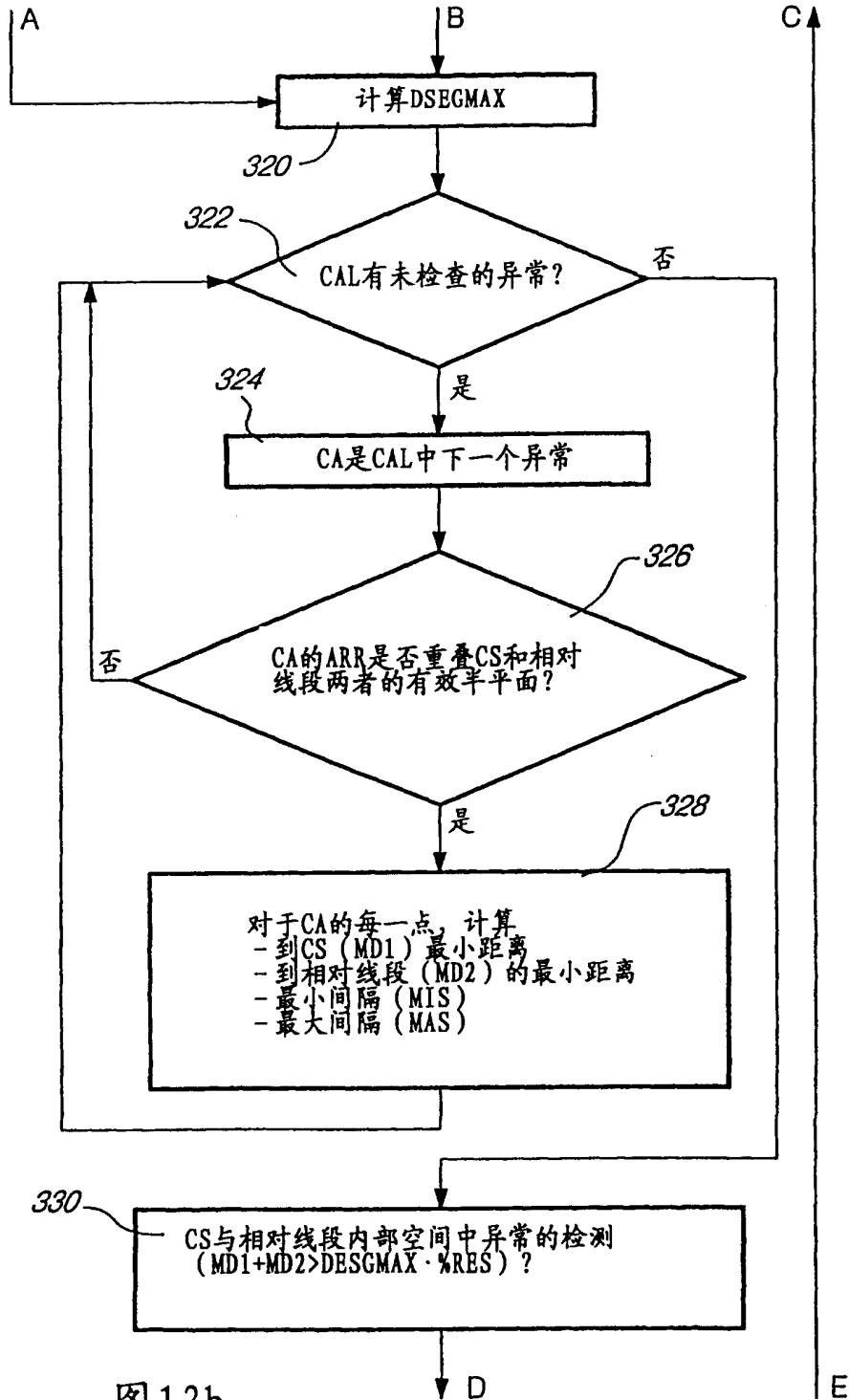


图13a



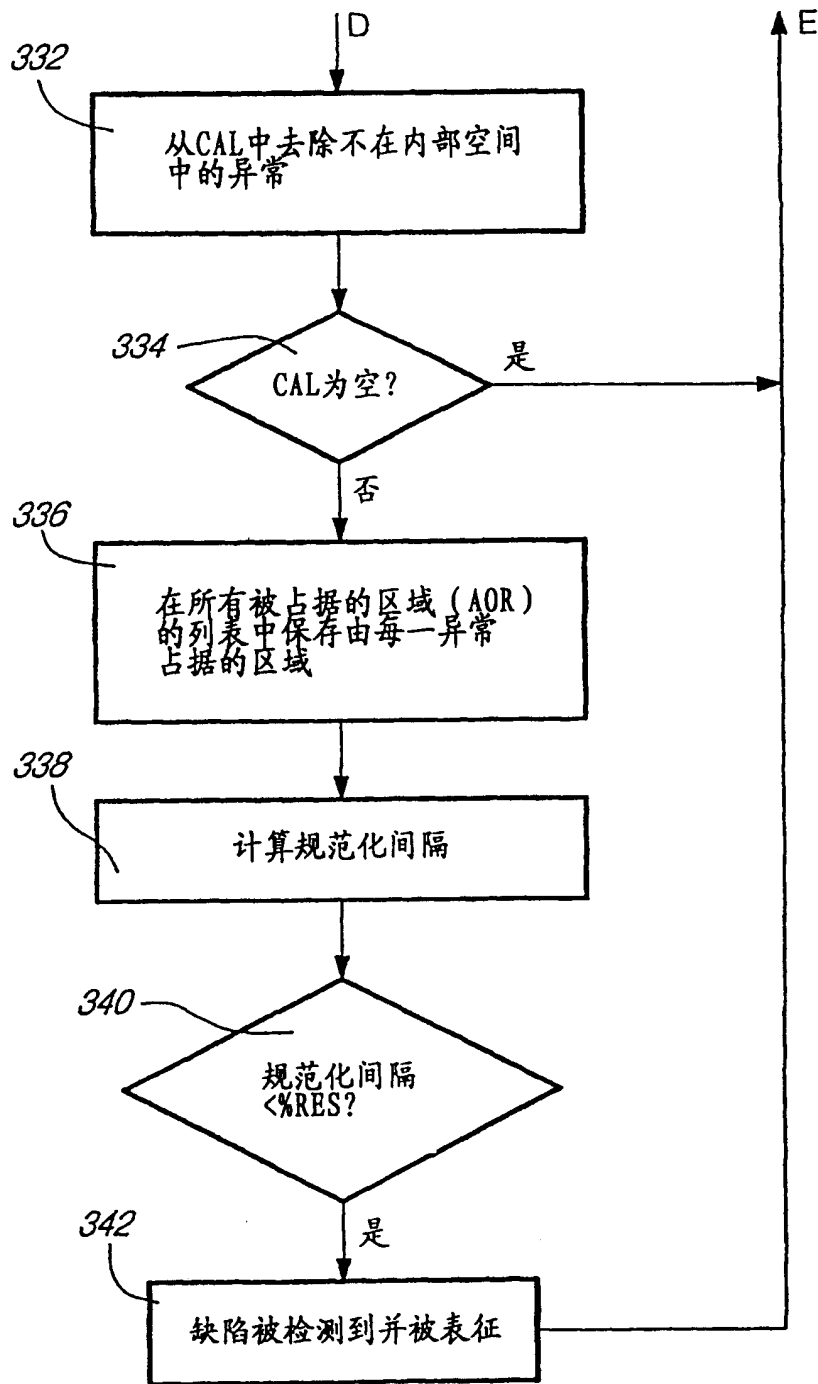


图13c

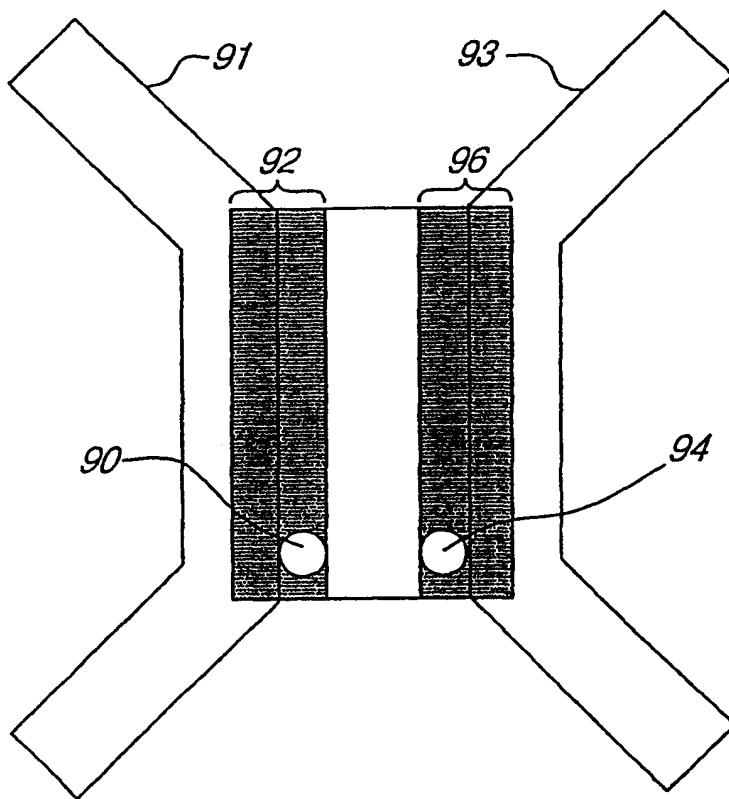


图14

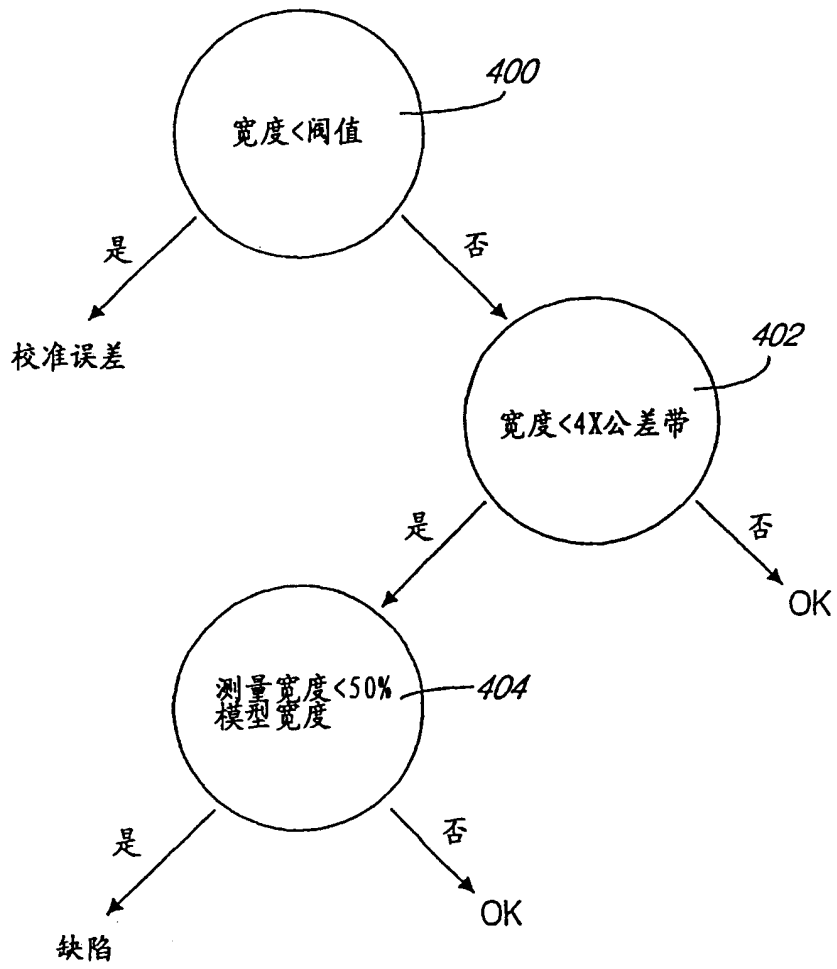


图15

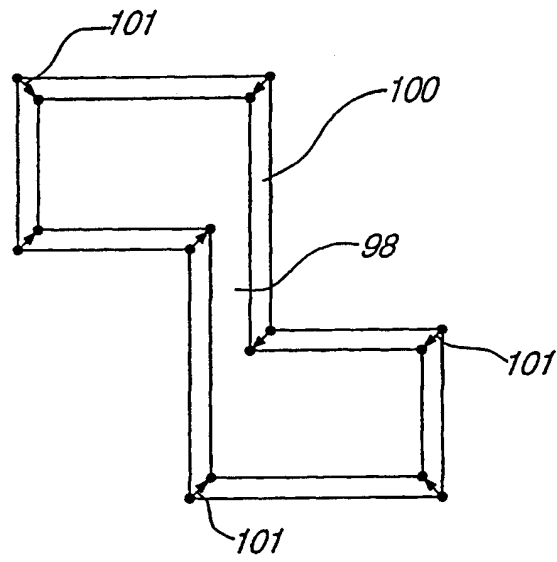


图16

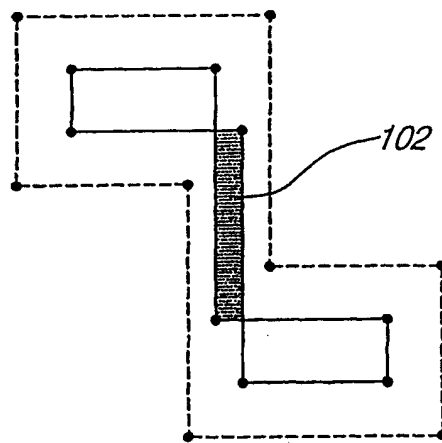


图17