

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G01M 11/02

[12] 发明专利说明书

[21] ZL 专利号 93119892.5

[45] 授权公告日 2001 年 9 月 26 日

[11] 授权公告号 CN 1071893C

[22] 申请日 1993. 12. 21

[21] 申请号 93119892.5

[30] 优先权

[32] 1992. 12. 21 [33] US [31] 993,756

[73] 专利权人 庄臣及庄臣视力产品有限公司

地址 美国佛罗里达州

[72] 发明人 J·A·伊贝尔 P·赛斯

审查员 丁惠玲

[74] 专利代理机构 中国专利代理(香港)有限公司

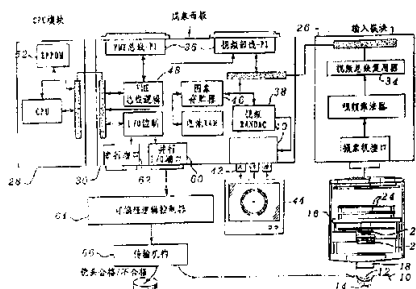
代理人 马铁良 张志醒

权利要求书 4 页 说明书 37 页 附图页数 13 页

[54] 发明名称 眼用透镜的检验方法和设备

[57] 摘要

一种眼用透镜的检验方法和包括摄像机的设备,每个像素光强和位置转换成电量存于存储器。计算机从接收器视场中心至边缘进行光强和位置比较,直至有强度偏差。评价强度变异的像素,获得透镜边缘的轮廓线。建立包围透镜边缘的环形区。所有的像素从绝对光强值变为梯度值,由两个转换边缘所体现。进行特性提取,定位有缺陷的像素、分组。根据其中缺陷像素的数量、性质打分。据此对透镜加权打分,判断合格与否。



I S S N 1 0 0 8 - 4 2 7 4

权 利 要 求 书

1. 一种检验眼用透镜的方法, 包括步骤:

在象素阵列上形成透镜的被照射图象;

产生表示所述象素阵列上被照射图象的强度的数据值;

给每个象素分配一个位置值和一个表示象素上透镜图象强度的图象强度值;

其特征在于,

(i) 比较象素之间的位置值和图象强度值, 以建立象素之间的关系;

(ii) 从所述象素关系识别预定透镜特征被照射图象中的象素组;

(iii) 将被识别象素组中的象素关系与预定建立的关系进行比较, 以确认透镜是否可接受。

2. 按照权利要求 1 的方法, 其中象素之间比较是沿着透镜边缘的轮廓这样一条路径进行的。

3. 按照权利要求 1 的方法, 其中特征的识别包括采集具有形成象素组特性的象素。

4. 按照权利要求 1 的方法, 其中比较是在包括有透镜边缘的象素组之中进行的。

5. 按照权利要求 1 的方法, 其中比较是在包括有透镜内部部分的象素组之间进行的。

6. 按照权利要求 2 的方法, 其中透镜边缘首先被如此定位, 即从接近象素组中心的点起动延伸到象素组的边缘, 直到发现一个具有透镜边缘强度值特征的象素。

7. 按照权利要求 6 的方法, 其中靠近具有透镜边缘强度值特

征象素的其他象素处跟随着具有透镜边缘强度值特征的相连象素组的轮廓，以确定该轮廓是否为透镜的边缘。

8. 按照权利要求 4 的方法，其中处理环被设置在透镜边缘附近，以限制要处理的那些靠近透镜边缘的象素数目。

9. 按照权利要求 4 的方法，其中包括透镜边缘的象素组进一步被分成两个子象素组，以完成所说的比较，一个子象素组包括从透镜内部到透镜边缘的转移，而另一个子象素组包括从透镜外部区域到透镜边缘的转移。

10. 按照权利要求 9 的方法，其中所说比较是在包括从透镜内部到透镜边缘转移的子象素组的象素之间进行的。

11. 按照权利要求 9 的方法，其中所说的比较是在包括从透镜外部区域到透镜边缘转移的子象素组中的象素之间进行的。

12. 按照权利要求 9 的方法，其中所说的比较是在两个子象素组之间进行的，一个子象素组包括从透镜内部到透镜边缘的转移，另一个子象素组包括从透镜外部区域到透镜边缘不转移。

13. 按照权利要求 4 的方法，其中所说被比较的关系在形成透镜边缘的象素处是间断的。

14. 按照权利要求 4 的方法，其中所说的被比较的关系在形成透镜边缘的象素的强度上具有梯度偏差。

15. 按照权利要求 4 的方法，其中所说的被比较的关系在形成透镜边缘的象素的位置上具有径向的偏差。

16. 按照权利要求 4 的方法，其中所说的被比较的关系是在形成透镜边缘的象素的位置上的空间偏差。

17. 一种检验眼用透镜的方法，包括：

- 1) 在象素阵列上形成透镜的被照射图象；
- 2) 产生表示所述象素阵列上的被照射图象的强度的数据值；

3) 将所述数据值之一分配给每个象素组; 其特征在于以下步骤:

用数字数据计算机

4) 选择一个象素阵列作为感兴趣的象素;

5) 从数据值确定该感兴趣的象素是否具有一种特征,

如果感兴趣的象素没有特征:

6a) 将感兴趣的象素换为另一象素阵列中的象素, 其中所述两个象素都在同一给定路径上, 和

6b) 重复步骤(5);

如感兴趣的象素具有特定特征:

7a) 将该感兴趣象素的数据值与邻近象素的数据值进行比较,

7b) 确定一个相对特定特征具有最佳相关性的所述邻近象素,

7c) 将感兴趣象素改为具有所述最佳相关性的所述相邻象素之一,

7d) 重复步骤(7a), (7b)和(7c), 直到感兴趣象素完全代表该特性,

7e) 确定由感兴趣象素所聚集的象素组是否代表了透镜的被照射图象的边缘,

8) 对于那些不代表透镜边缘的象素组, 重复步骤(6a)和(6b), 以及

9) 对于那些代表透镜边缘的象素组, 将象素组之间的关系与预定的关系进行比较, 以确定透镜是否可接受。

18. 按照权利要求 17 的方法, 其中起始象素定位在透镜的中心附近, 而横过透镜边缘的路线是从透镜中心延伸出来的一条射

线。

19. 按照权利要求 18 的方法，其中特定的特征在图象强度方面有变化。

20. 按照权利要求 18 的方法，其中透镜的大概中心是如此确定的，即在具有边缘特征的至少三个点的组中至少取一个组。

21. 一种用于检验与评价眼用透镜的装置，包括：

一个用于照亮透镜的光源；

一个摄象机，该摄象机包括一个接收器，接收器中包括多个象素，用于接收透镜的被照射的图象；和

用于产生表示每个象素上被照射图象的光强度的数据值的装置；其特征在于，

存储装置，用于存储与每个象素相关的图象强度值和与每个象素在接收器上位置相关的位置值。

一个数字计算机，与存储图象强度值和位置值的存储器连接，该计算机包括用于在象素之间比较图象强度值和位置值的装置，并进一步包括用于确定所述特征是否使透镜不可接受的装置。

说明书

眼用透镜的检验方法和设备

本发明涉及一种用于自动检验眼用透镜，特别是其结构由一部分实质性为水的水凝胶隐型眼镜，的方法和设备，这种方法也适用于检验其它小型高精度眼用透镜，例如眼内透镜。与本发明同时提出的系列申请的中国申请的申请号为 No.93119939.5, No.93121292.0, No.9312049.2 和 No.9310497.6。

由于对眼镜的要求极严格的特性所致(用于人眼矫正视力的透镜)，在眼镜工业中，占据极端重要地位并需优先考虑的是对这些眼镜进行检验，以确定它们是否满足需要的条件。这些条件不仅涉及光学性质如放大率，同时也涉及物理性质，例如，尺寸、曲率、边缘完整性和有无气泡，有无杂质和其它缺陷。

以前检验此类透镜的最可靠的方法是由一个观察者观察放大的镜片，以检验镜片是否满足要求的每一特性。然而，随着眼镜工业的发展，这种检验方法对该工业来说是一个很大的人力和财力上的负担，并且要求检验员完成乏味的工作。特别是考虑到周期性频繁更换的隐型眼镜，需要制造和检验的镜片的数量会急剧增加。与检验眼镜相关的一个问题是，镜片本身是光透明的，因此不能显示出检验常规物体时所能发现的明暗特征。

现行的由人工进行的检验方法采用的是本领域熟知的暗场照明纹影 (Schlieren) 法，特别是用于研究透明的液体流动和光学元件的检验，在该方法中，由点光源发出的光被一个透镜准直，再穿过介质 (即，镜片)。然后该光线通过共二个透镜被直接焦聚在一个刀刃上。

由于镜片的折射不均匀所造成的反射光线（虽然透明）都未聚集在刀刃上。被刀刃遮断的反射光线接下来通过一个物镜投射到一个屏幕上，然后在本来应该全黑的投影屏上对应于非均匀性产生一个光点。

对于检验眼镜来说特有的另一个问题是，尽管镜片的尺寸允许在标称尺寸的基础上有所变化，但是一些可允许的缺陷的尺寸，例如边缘的裂痕和中心的水泡的尺寸使比允许的标称尺寸的变化值小两个数量级也是不能接受的。自动检验的通常方法，是将花纹或模板放置在待检的象上然后与象进行对比，但不适用于检验眼镜，因为检验所要找寻的缺陷往往比透镜的标称尺寸允许的变化值的百分之一还要小。

因此，本发明的目的之一是提供一种高速自动检验眼镜的方法和设备，其精度与人工在放大检验情况下的精度在同一水平。

本发明的另一目的是采用现成的视象设备、电子元件和计算机系统完成上述检验。

本发明的另一目的是，当一个透镜在生产线上丢失时，能够被发现。

本发明的另一目的是将图象分析集中在透镜上产生缺陷的最可疑的部位。

本发明的又一目的是在确定透镜边缘之前分析透镜的特征，并确定该特征是不是导致透镜成为废品的一个裂纹。

本发明的又一目的是将在透镜边缘发现的间断点搭接，从而微小的透镜缺陷和图象处理遗失信息不致导致检验步骤对于某一特别的透镜没有作用，进而导致好的透镜被当作废品。

本发明的又一目的是根据类别对特征的偏差进行归类，以把透镜中发现的缺陷类型列成表，为生产工艺提供信息。

本发明的这些和其他目的通过一个电子摄像和摄取被一个点光源照亮的的一个眼镜象来达到。象被投射到摄像机的接收部位；该接收器由一组象素组成。每个象素的位置和投射到其上的光强被转换成一个电信号然后转移储存在一个存储器中。与存储器相接的计算机能够取出存入的位置和光强值。计算机中装有指令，对各象素的光强和位置数值进行比较。

在优选实施例中，上述比较过程包括：从位于接收器中心附近的一个象素开始，朝着象素场边缘的方向继续下去，直至出现强度差值。通过计算围绕中心象素的那些含有强度差值的象素，找出与中心象素最接近的匹配，然后重复上述过程，就得到了特征的概况。如果这些特征的特性并非与透镜的边缘相对应，就判断该特征，以便确定该透镜是否应予废弃。如果该特征对应于透镜的边缘，就要判断边缘的特性，以便决定该边缘亦即该镜片是否能够被接受。

在优选实施例中，是这样实现的：沿边缘位置探测，直至找到相隔 12° 的 30 个数据点。求解三个联立方程式（3 点一组共十组）以发现透镜的平均中心和平均半径。据此，使用上述中心和半径数据，产生一个环。这个环的边界包围真正的透镜边缘。在该环内，所有的象素从绝对光强值转换成一个梯度值，使得透镜边缘可以被两个转换边缘代表，一个从接触透镜边缘的内面，一个从其外面转换而来。然后削薄这两个边缘使得只剩下沿边缘的光强最大的象素。然后这些边缘被从 X Y 坐标系转移到极坐标系，保持半径值、角度和强度梯度不变。然后在这两种数据阵中进行提取特征。提取特征包括搜索半径偏差，定域内的梯度偏差、空间偏差，D I P 定域内的梯度偏差，单向定域内梯度偏差和不连续性。在判断了这些象素的电平之后，每个缺

陷像素被视为是一个缺陷组的成员。在对内外透镜边缘轮廓分别分组之后，那些在透镜边缘两面的同一区域相互重叠的组落入复合缺陷组。当所有的组被确定之后，根据放入该组中的像素的数量、类型和严重程度，每个组被打一个分数。据此，根据这些分数赋予整个透镜一个加权的分数，确定透镜合格或不合格。

图 1 是与本发明的物理元器件的方框图，以及它们之间的连接关系、光源、待检验的透镜。

图 2 是本发明的设备对透镜成象并对这个图象进行处理以检验透镜是否合格时的处理步骤流程图。

图 3 是本发明的算法在图象上对透镜进行定位时采用的技术手段。

图 4 是一种在透镜边缘上每三个为一组采集数据点的代表方法。

图 5 是一种从事先收集的数据点确定要探查的透镜边缘的中心和半径的方法实施例。

图 6 表示出在已获得的理论上的接触透镜边缘周围所建立的处理环。

图 7 表示边缘加强操作的结果，该操作产生两个转换边缘，一个为内边缘，一个为外边缘。

图 8 表示为了使用不同方向的边缘操作透镜被划分成的不同区域。

图 9 是一个示意图，表示本发明的算法所采用的方程式在对不同的像素位置进行表示时所使用的标志法。

图 10 是一个示意图，表示对透镜图象的角度标记的惯用方式。

图 11 是表示在处理环内部使用搜索矢量确定透镜的结果。

图 12 表示加强的透镜图象中出现的不连续点以及探测一个不连续点时进行的像素处理，其中，图 12 a 是在直角坐标系中，图 12

b 是在极坐标系中。

图 1 3 表示对透镜的断裂部分的搭接，其中图 1 3 a 是在直角坐标系中，图 1 3 b 是在极坐标系中。开始的搭接使用的是外推法技术，最后的搭接使用的是跳跃技术。

图 1 4 表示使用象素处理法对透镜边缘的径向偏差进行的判断，其中图 1 4 a 是在直角坐标系中，图 1 4 b 是在极坐标系中。

图 1 5 表示被关注的象素与其相邻象素在梯度上的关系。

图 1 6 表示由象素处理所得到的梯度提取，其中图 1 6 a 是在直角坐标系中，图 1 6 b 是在极坐标系中。

图 1 7 表示从透镜边缘提取空间偏差特性的方法，图 1 7 a 是在直角坐标系中，图 1 7 b 是在极坐标系中。

本发明的实现不局限于一种特定的硬件系统，实际上可采用许多种软件系统来完成。但是，下面所给出的实施例，是一种特别好的模式。

参见图 1，本发明系统包括一个支托着一个接触式透镜包装盒 1 0 的结构（未画出），该包装盒中装着放置于去离子水之中的接触式透镜，位于光源 1 4 之上摄像机 1 6 之下。

摄像机 1 6 是一种高分辨率固态摄像仪，例如 Kodak 公司生产的 Videk Megaplu 牌。

这种摄像仪包括一个镜头 1 8，在本实例中该镜头被固定在 1 4 . 5 mm 的视场上。该摄像仪配备了一个 Nikkor 5 5 mm 标准镜头。该镜头设置在 $f / 2 . 8$ ，然后连接到一个与摄像仪镜头的终端相连的中心波长 5 5 0 nm、具有 1 0 nm 全波半高（FMHH）的 Andover 带通滤波器上。该滤波器能够消除色差从而提高总体空间分

分辨率并且使得检验透镜时的光响应保持在与人工检验员的视觉响应相似的水平。该滤波器还能消除CCD探测器的红外光，该红外光会降低整体系统的调制传输功能(MTF)。

在装有盛放在去离子水中的透镜的包装盒之下，在它之下是一个频闪灯14之类的光源。该频闪灯由图象处理系统激励，能够产生5焦耳、10毫秒的光脉冲。通常，在频闪灯每次发光之间，需要一个450毫秒的恢复时间用以充电。

摄象机16的调焦是通过一个精密的导杆驱动机构完成的，该驱动杆推动摄象机沿着它所固定的框架上、下移动。一旦摄象机聚焦，它就保持在静止状态，完成检验程序。

摄象机还包括一个电荷耦合器件传感器(CCD)。它起到光接收器的作用。该CCD接收器由一个矩形阵列(1,320×1,035)的像素阵列构成。

摄象机的接收器电荷耦合器件传感器将光强转换成一个电信号。这个模拟电信号接着又被电路24通过一个8位模/数(A/D)转换器转换成含有256个灰度电平的数字视频输出信号。

摄象机工作在非同步方式，它利用透镜产生的信号和移动到适当位置的包装盒激发频闪灯的发光以及后续的图象转移。

通过摄象机中的电路24，图象被传输到图象处理系统的输入模块26。图象处理系统由三个部分构成，输入模块26，CPU模块28，和图象母板30。

对输入模块26中的数字信号进行处理，以保证每条都有相应的视频同步信号。修正后的信号然后被输送给一个视频复用器34，它

将数字化的信号传输给CPU模块28。

这组代表一个视帧（或者特别描述为摄像机1，048，476个像素）的数据被送入视频主线36，它能够把信号传输到其它处理硬件。同时还使用视频RAMDAC38显示图象，它通过转换器40被转换成一个伪彩色输出，并通过RGB输出42传输到一个视频监视器44上，在这里图象能够显示出来。RGB伪视频彩色输出由三个查阅表（LUTS）提供。

CPU模块28的输入在第三种情况下被传输到约为1兆字节的图象存储器46中。储存在图象存储器46中的数据通过一个图象时间控制器，以通常的方式传输到处理单元48。

数据的处理是由Motorda 68040 CPU完成的。

所需的图象处理编码存储在可擦除、可编程、只读存储器EPROM52。图象处理后的结果被送给图象母板30，以进行适当的输出控制。输出即可通过16位数字并行输入/输出端口60，也可通过RS232或RS422串行口62。

处理器所做出的接受/否定决定被送到传输机构66，它决定透镜12是否进一步进行包装和消毒处理，或被做为废品销毁。

现在参见图2，由图1所示装置实现的检验程序以框图的方式给出，并且详细示出了CPU模块28进行的检验算法。

在第一步骤，由摄像机捕捉到原始图像，并送入图象传输系统。这个图象被转换成一系列数字数据，它包含有校正摄像机中已知缺陷像素的算法规则。

高分辨率摄像机的价格取决于CCD传感器接收器20上所能允许的缺陷像素的数目。这首先包括这样一些像素单元，它对相邻像素

和五个或更少象素单元集簇有 10 % 或更多的光响应。

由于缺陷象素的数目即使在不十分贵的摄象机上也是有限的，而且这种聚集在一起的象素数目也是有限的，所以价格不贵的摄象机仍然能够用于透镜的检验。

Videk Megaplus 摄象机的 CCD 传感器包含数个柱形缺陷。这些缺陷通常限于一个单独的柱形，可以是几个象素长或 50 或更多个象素长。这些缺陷区域导致象素灰色电平在不考虑图象本身的情况下高于或低于相邻的象素。如果这些缺陷位于透镜边缘，它们能够错误引起软件将其当成透镜的不连续或将其当作透镜的缺陷。

探测缺陷区域可以通过人眼观察一个均匀的目标的图象，找出灰色电平值出现异常偏差的区域。由于缺陷局限于一个单独的柱形，在相邻的柱之间插入修改是一种适合的修正。柱形缺陷的插入值恰好是缺陷两侧的柱的灰色电平值的平均值。所用摄象机的传感器也有可能具有聚集型的缺陷，它们的形状为圆形的斑点。这种类型缺陷的调节也可以是将它们保持在一个不影响图象处理的区域或者是使用上述同样的内插技术。

当上述对已知有缺陷的摄象机象素做出修正之后，通过辨认透镜的边缘透镜被安置在图象视场中。当进行了找寻透镜边缘的尝试之后，可确定透镜确实在容器中或者透镜从容器中遗失。如果容器中丢失了透镜，它将被视为一个废品透镜，从而空的包装盒不进行处理而被送到销毁装置处。

当确立了镜头的边缘存在的情况下，将根据围绕透镜边缘的点的位置采集数据。这些数据用于建立多个边缘数据组（三个一组）每组限定一个圆和一个圆心点。距离平均中心最远的中心被丢弃，以去除

特别数据。

在这时特殊的边缘信息还未被采集，仅采集了三个一组的边缘数据组以定位边缘和透镜中心。

为了实际检验边缘，计算出来的中心点被用于建立一个其内包含有实际的接触透镜边缘的处理环。这保证了进一步的详细的数据处理仅仅被集中在包含有所感兴趣边缘部分的环之中。

为了从背景中区分出透镜边缘，在处理环的像素中加入了边缘增强处理操作。这个边缘增强带来两个透镜转换边缘。一个是从透镜的内部到边缘，一个是从透镜的外面到边缘。

虽然这时只有边缘信息留下，两个边缘（实际上是边缘转换）仍多多少少被朝着边缘和背向边缘的光强梯度所模糊。为了更清晰地限定这些转换边缘，对处理环包含的边缘数据进行修正轮廓操作。框架运算保留了灰度电平信息，因为它包含了对于特征提取有用的信息。

算法要进行的下一个步骤是阈值计算的使用，从而消除那些对提取特征来讲多余的那些灰度电平信息。

算法操作的下一步骤是搜索内，外边缘以提取透镜边缘特征。这个搜索过程的开始方式类似于开始定位透镜边缘的方式；但是还有不同之处：在从处理环的内界面出发向外搜索的过程中采用灰度电平临界值定位透镜边缘。当遇到一个符合灰度电平临界值的像素时，要检查与其相邻的一系列像素，以确定它是否是透镜的边缘。如果是，在整个边缘的附近继续找寻，并且位置和相应的像素光强梯度被储存下来。

然后直角坐标系的信息被转换成极坐标系中的半径和角度代替值，同时还有相应的光强梯度值。为了恰当地处理这些数据，不错误地否

定好的透镜，小于某量的像素不连续将被搭接。

采用现在的极坐标系中的信息，将进行五种类型的特征提取。第一种是探测与理想的内和外边缘的径向偏差（ $R D$ ）。接下来一种是定域内的梯度偏差（ $L G D$ ），它考虑的是每个像素相对于其相邻像素的光强度的梯度。然后进行空间偏差（ $S D$ ）特征的提取。空间偏差要测量半径相于角度位移的改变。与径向偏差提取相反，空间偏差提取着重的是首先是边缘半径相对角度位移改变的急剧程度或突然改变。

进行三种最终的特征提取。不连续性缺陷是这样一种情况：无论在透镜的内和外边缘的不连续性如此之大以致于算法规则不能将其视为可以搭接修正的。

相似于定域内梯度偏差特征，下降定域梯度偏差（ $D L G D$ ）着眼点是感兴趣的像素的梯度值与其相邻的定域内的平均值之间的偏差量。区别是所采用的相邻像素数量更多，而且在感兴趣的像素周围的未采用的像素的区间更大。 $D L G D$ 被设计成仅仅对小于其相邻像素的梯度偏差敏感，因此得名“下降”。

$D L G D$ 特征特别能够辨别出其它特征无法辨别的微小边缘缺口。一个根据 $D L G D$ 被归为有缺陷的像素具有包含着剧烈程度的偏差量。

单向定域内梯度偏差（ $A L G D$ ）采用感兴趣的像素某单独一边的邻边像素来计算偏差。在感兴趣的像素的前方取20个相邻像素（除掉4个与感兴趣的像素直接相邻的像素），然后计算平均值。 $A L G D$ 所着重的是即有负的也有正的梯度偏差。

当上述特征提取处理步骤进行的同时，与构成缺陷的剧烈程度成比例地，对所获得的已分辨出来的特征标记出不同的分值。此外，通

过检查内、外边缘的异常象素将缺陷归为不同的小组，以确定它们是否属于应归于同一缺陷组的一个较大的缺陷。然后判断这些组，看它们是否应当互相结合成为较大的组，以及当结合之后，内和外边缘缺陷组是否应当被认为是单独一个缺陷。

最后，根据剧烈程度给每个缺陷和缺陷组打分数，并且根据它对于透镜质量的影响，对每种类型的缺陷进行加权。

对所有这些缺陷的这些结果进行相加，以得一个单独的数字，该数字确定该透镜应当被接受或者必须被否定。

此外，可以以表格的形式列出或显示出加权和打分程序获得的多个数字信息，从而为正在生产的透镜给出一个统计的质量分析，从而引导生产工艺的控制者发现工艺参数的异常变化，以及评价变化对生产工艺产生的影响。

上述算法程序进行的步骤将在下面详细描述。

参照图 3，所示是一个由一系列象素（未示出）构成的接收场 70。在场内是一个接触透镜 72 的象。在这个特定的象中，透镜由边缘 74 和缺陷或噪声 76 构成。在这个特定的透镜象中还可发现边缘 78 处有一个缺口。

透镜边缘的设定首先从视场的中央开始，使用成 45° 角的搜索矢量 80。搜索矢量从视场的中心径向向外逐个象素地移动直至遇到预期中的透镜边缘。算法程序检查沿搜索矢量的每个象素，直至边缘临界值得到满足；将每个象素与一个灰度电平和一个预置的标定临界值进行比较。如果正处在矢量中的某象素的灰度电平低于“*trk - thers*”参数规定的值，则假定边缘上的一个象素为已经被碰到过。

为了验证所碰到的物体就是透镜的边缘，矢量接下来搜索该目标

的轮廓线。在存在搜索矢量 8 0 和 8 2 的情形，运算程序发现该目标不是透镜边缘的一部分，因为其路径的曲率与预计的透镜边缘的曲率或周边不相符合。这个确认技术只需探测轮廓线轨迹在返回时是否在规定的象素之内就穿过起始点的象素，这个目标被确认为足够小从而是噪声或一个透镜缺陷。如果遇到一个噪声目标，接着建立一个大约 11° 的顺时针方向的搜索矢量，进行与原矢量相同的搜索。

图中还示出了另一个搜索矢量 8 4，它沿着一条穿过透镜边缘 7 8 的缺口的路径前行。这个搜索矢量将继续搜索直至到达接收视场 7 0 的边界，在这一点搜索中止，而另一个距离原搜索矢量顺时针方向 11° 的搜索矢量开始搜索。在每一种情形下，逐个象素的搜索都是对紧挨着的象素进行的，或水平方向，垂直方向或以阶梯形式进行的对角线方向。对于上述已遇到的两种透镜缺陷的任一种，位于透镜体 7 6 缺陷或者位于透镜边缘 7 8 处的裂缝，可以使用适当的临界值，使透镜被否定。

矢量 8 6 的情形，搜索是成功的，发现了透镜边缘 7 4。确认所发现的特征确实是透镜边缘的验证在图 4 中画出。

由先前发现的好的数据点开始，软件沿着透镜的轮廓线运行，使用 4 8 联通性追踪，追踪大约 9 0 个象素。根据透镜的实际半径 r ，跟踪距离根据以下公式将随不同的图象而不同：

$$\text{追踪距离} \equiv Y = (1024 / F) \times (2 \pi r / 30)$$

其中，

$$F = 14.5 \text{ mm (视场)}$$

因此，对于处在去离子水中的 12.2 mm 透镜来说， T 一般为 9 0 个象素。如果沿着透镜边缘距离 T 被成功地经过，一个数据点被

记录下来。

因此，透镜边缘的位置被验证了。当间隔 12° 的 30 个数据点被发现以后，对 3 个点一组的 10 组进行三方程联之求解，以求出透镜的平均中心和半径。

在图 4 中，方块，三角和圆形符号代表已收集了数据的点。具有相同内部图案的符号被采集到同一数据组。

然后用 10 个数据组来计算 10 个不同圆的方程，其中，每个圆代表透镜边缘的一个模型。计算出一个平均行象素和一个平均列象素圆心。然后计算出 10 个圆中的每一个的圆心到平均圆心的距离。任一圆心，当它与圆心分布的统计最频圆心之间达到预置的偏差时，将被排除。进行这种排除是为了排除掉那些由于采集了偏离正常透镜边缘的缺陷处的数据而导致的虚假透镜边缘。这被表示在图 5 中，其中，由于透镜边缘 87 的偏差，圆心 88 偏离了其它九个圆心的集中区域，因此被排除。

然后计算其它行和列圆心的标准偏差，并与一个规定阈值相比较。如果行和列标准偏差满足阈值临界值，则认为透镜已被发现。在最终模型中采用的半径是所剩的 9 个半径的平均值。如果两个标准偏差中有一个不满足阈值临界值，则建立一个新的搜索因子。新的矢量的起始角度从原来的矢量处以这样的方式确定：使得数据不在沿透镜边缘的相同点处采集。这种嵌套的反复的找寻边缘和采集数据点的程序最多将连续两个循环。如果在这段时间内不能成功地找到透镜，该透镜被认为丢失了，被自动否定。

见图 6，一个处理环 90 被建立并重叠在透镜边缘 72 周围。因为建立该环时采用先前得到中心和半径值和处理环宽度的内部参数，

处理环的边界必然将透镜边缘包括在其中。所有对图象进行的进一步处理都将仅仅在该环内进行，目的是通过限制所要评价的象素的数量提高速度，减少处理时间。对处理环大小的要求（确实需要使用一个限制性的处理环）取决于能获得的计算机处理能力和有关的花费。环形圆环的圆心是运算程序的先前步骤建立的圆。环的象素宽度取决于参数“ank - width”。

见图 7，画出了运算程序的下一步骤，但前一附图中的处理环未被画出。在接收器视场 7 0 上再次显示的是平均中心 9 2。由于灰度电平信息本身不足以敏感得保证将正常的透镜边缘和其缺陷区域区分开来，因此在原始透镜图象上进行了边缘加强操作。因此，该操作被用于引起包含在透镜边缘图象的内、外两侧的信息的区分。所使用的操作手段是一个改型的 3×3 运算符，它利用透镜边缘附近的不同区域。前一附图中的接触透镜边缘 7 2 在实行边缘加强操作之后已经被消除。边缘加强操作将 2 至 4 个象素宽的原始透镜边缘图象处理成为一个单独的内边缘 9 4 和外边缘 9 6，如图 6 所示。

为了实现边缘加强操作的处理，必须将透镜划分成不同的区域。

区域的准确起始和结束取决于图象中透镜的大小以及透镜在图象中的位置。图 8 表示出透镜采用的五个区之间的关系。建立这些区的目的是区分透镜的水平方向、垂直方向和对角线方向的区域。

图 8 中的每个区使用一个不同方向的边缘操作。对于 1、3、5 区，使用了一个对角线操作。对于 2、4 区，使用了水平/垂直操作。使用不同的方向操作是为补偿透镜的曲率并且无衡透镜上的梯度值。即，在近乎垂直的透镜部分的对角线操作，基本上等同于水平/垂直操作在基本上斜交叉的透镜部分的情况。

为了消除不希望的画象抖动，边缘操作被设计为相对于被处理的边缘是交叉的。正常的原始透镜的边缘的厚度在一定区域内是轻微波动的。在平行于或垂直于原始透镜图象的方向上进行的操作因此有可能拾取上述波动信息，从而错误地保留着边缘的小偏差。特征提取软件视这些小偏差为边缘的缺损。

参见图 9，所示为在边缘加强操作的实现过程中采用的象素标志。如本领域的普通技术人员所能理解的那样，这个标志符号是用于表示矩阵或阵列中元素的标准标记法。

下面的方式程 1 - 6 表示两种边缘处理所用的算法规则。所得的每个象素的梯度值被换算成为落入 0 至 255 范围内的 8 位数字。

$$\text{水平/垂直操作} = \text{abs}(\text{khdif}) + \text{abs}(\text{vdif})$$

其中：

$$\text{hdif} = P_{i-1,j+1} + 2 * P_{i,j+1} + P_{i+1,j+1} - \\ (P_{i-1,j-1} + 2 * P_{i,j-1} + P_{i+1,j-1})$$

$$\text{vdif} = P_{i+1,j+1} + 2 * P_{i+1,j} + P_{i+1,j-1} - \\ (P_{i-1,j+1} + 2 * P_{i-1,j} + P_{i-1,j-1})$$

$$\text{对角线操作} = \text{abs}(\text{d1dif}) + \text{abs}(\text{d2dif})$$

其中：

$$\text{d1dif} = P_{i-1,j} + 2 * P_{i-1,j-1} + P_{i,j-1} - \\ (P_{i,j+1} + 2 * P_{i+1,j+1} + P_{i+1,j})$$

$$\text{d2dif} = P_{i-1,j} + 2 * P_{i-1,j+1} + P_{i,j+1} - \\ (P_{i,j-1} + 2 * P_{i+1,j-1} + P_{i+1,j})$$

当在 1、3、5 区内进行了对角线式边缘加强操作，在 2、4 区进行了水平/垂直边缘加强操作之后，对所获得的最终内、外边缘进

行轮廓操作。所获得的边缘典型地为 1 个象素宽，并包含仅仅来自从边缘的剖面处看，边缘最强部分的信息。在这个步骤中，无论如何，包含在这些象素中的灰度电平信号仍被保留。这种方式的边缘加强是沿着与用于多个环区的梯度操作方向相匹配的方向进行的。该操作仅仅在以前的步骤中得到的梯度信息的基础上进行，它从感兴趣的象素出发沿四个方向搜索峰值。如果它发现一个峰值，就用这个值取代感兴趣的象素。如果未发现则该象素变成 0，最终出现如图 7 所示的图象。

算法运算的下一步骤是使用阈值机理对刚刚被加强的和进行了轮廓处理的透镜内、外边缘进行定位和追踪。当边缘被搜索时，阈值仅仅沿着透镜轮廓线发挥作用。

算法运算的下一步骤是对超过规定的象素光强值的选定象素施加阈值限定。施加阈值操作的目的在于消除环区内所有不再是感兴趣的边缘部分而成为噪声的象素。用于阈值限定的灰度电平值对于内、外边缘分别是参数“inner - thr”和“outer - thr”。这些是用于追踪外轮廓线的算法中的阈值。阈值操作的实施如下面所示：

如果 $(P_i, j) \geq \text{阈值}$ 并且是被处理的透镜边缘上的象素) 则

$P_i, j = \text{轮廓线象素}$

图 9 所示为用于对在传感器视场 70 中透镜图象的角度标示法。为了找到透镜边以开始追踪，使用了一种与开始用于定位透镜所使用的相类似的搜索矢量。在这种情况下，搜索矢量仅仅使用灰度电平值作为搜索下一个象素的判断依据。

参照图 10，矢量在零度处从处理环内开始搜索，并且沿着象素的行进行处理，直至遇到透镜边缘或者到达处理环的另一边。

参照图 1 1，该图表示了用于搜索内边缘的一种可能情况。

该图示出了一部分放大的加强透镜边缘，其中有内边缘 9 4，外边缘 9 6 和外边界 1 0 0。

以上作为实施例描述的是第一搜索矢量 1 0 2。在第一搜索矢量的这一实施例中，它遇到了一个小缺陷 1 0 4，因为这个噪声或缺陷 1 0 4 具有特异的灰度电平值。搜索矢量追踪了它的边界，但是根据遇到开始那个象素之前追踪的象素的数目，判断出它的曲率与透镜边缘不相符合，因此算法程序否定了该目标。

在定位边缘的此次尝试失败之后，在距离第一个搜索矢量 2 0 行象素的地方又建立起第二搜索矢量。作为一种例子，第二搜索矢量从处理环 9 8 的内边界开始向外边界环 1 0 0 前进以试图找到透镜内边缘 9 4。在该例子中，搜索矢量 1 0 6 没有遇到特异于透镜边缘的象素，并且通过一个缺口 1 0 8 以及内边缘 9 4 和外边缘 9 6。一旦搜索矢量到达处理环 1 0 0 的外边界，搜索即中止。

然后，在距离第二搜索矢量 2 0 行象素的地方又建立起第三个搜索矢量 1 1 0。在该第三搜索矢量 1 1 0 的实施例中，发现内透镜边缘 9 4 的努力成功了，接着算法程序将追查集中在内和外透镜边缘 9 4 和 9 6。

只要前一矢量由于遇到小缺陷或遇到透镜边缘上的缺口没有能够成功，就重复建立新搜索矢量的过程，直至的找到透镜边缘，但该过程最多只重复十五次。这个过程对于透镜内边缘 9 4 和透镜外边缘 9 6 是分别进行的。

一旦一个边缘的位置被探测到，就进行 8 联缀轮廓程序。矢量以零度角开始，用 8 联缀法追踪透镜边缘的内、外轮廓线。8 联缀法可

以保证任何属于边缘的象素将被包括在最终的轮廓线中。为了确定某一象素是否属于边缘的一部分，使用一个灰度阈值，对于内边缘，采用“inner - thr”参数，对于外边缘，采用“outer -thr”参数。

如果刚刚进入的象素是一个边缘象素，运算程序采取右手转弯，如果刚刚进入象素不是一个边缘，运算程序采取左手转弯。当遇到特别的周边时，检查对角线的象素。由于追踪内、外边缘所用的编码是同一个，因此在内边缘追踪时采用顺时针方向，对于外边缘追踪采用逆时针方向。

当完成之后，透镜边缘由大约 6 0 0 0 个象素构成，3 0 0 0 个在内边缘，3 0 0 0 个在外边缘。如果象素的数目不在限定的范围之内，算法程序就决定，某透镜未被发现，然后或者重复先前的过程，或者否定该包装。

对于边缘上的每一个象素，其一组信息被存储在一个阵列结构中。这些信息包括其径向和角度位置，梯度灰度水平，缺陷类型和严重程度。这时在算法程序中不是包括了所有的透镜阵列信息，但是为了将来的使用存储工作要配合进来。如果一个象素被发现是边缘的一部分，对其进行从直角坐标到极坐标系的转换。直角坐标系和极坐标系的原点是开始定位透镜的圆心。下面的方程式表示出完成转换的方式，其中的 θ 是角度， r 是半径。

$$\theta = \arctan \left[\frac{(\text{透镜圆心行数} - \text{象素行数})}{(\text{象素列数} - \text{透镜圆心列数})} \right]$$

$$R = [(\text{象素列数} - \text{透镜中心列数})^2 + (\text{透镜中心行数} - \text{象素行数})^2]^{+2}$$

θ 从位于 0.0 到 360.0 度的一个浮动带小数点的值被转换成位于 0 到 8191 范围内的一个整数，可用 13 比特表示， 2^{13} 。
R 在开始也被计算成一个浮动小数点的值，被舍位成为一个整数。
然后，每个象素的半径和角度位移被存入一个大的阵列结构。更经济起见，进一步的处理将仅仅针对在这个大阵列中发现的 6000 个左右的象素。

在下面的附图中，透镜的边缘是以图画方式表示的，然后运算是由算法程序在数字域中进行的。

见图 1 2，所示为加强的透镜边缘的图象，图 1 2 a 是在直角坐标系中，图 1 2 b 是在极坐标系中。由于边缘已经被发现并且被追踪，该图中删去了处理环。图 1 2 a 和图 1 2 b 给出了透镜内边缘 94 和透镜外边缘 96。在图 1 2 画出的操作中，边缘的不连续性由边缘的缺陷、薄边缘或者边缘加强操作引起的变异而导致。不考虑引起的原因是什么，需要探测和修补这些不连续，以便能够处理边缘剩下的部分。

探测不连续性是这样进行的：保持已经被追踪的象素中最远的那个象素的角度位移，将它与正在被处理的象素的角度位移进行比较，如图 1 2 b 所示，如果最远的象素 112 和本象素之间产生的角度与追踪的方向反方向，而且它大于“bktrk - degs”参数规定的角度，则探测到了一个不连续处。这一点如图 1 2 b 的 114 点所示。

当探测到一个不连续之后，算法程序利用最远的那个象素作为参照物进行搭接。进行不连续性搭接的开始步骤是使用外推法技术，它能够搭接 1 到 3 个象素的缺口。外推法采用到达不连续处之前正在运行的方向。

在某些情况下，边缘上的缺口大于 3 个像素，因此不能用外推法搭接。参见图 1 3，在这种情况下，对外推法不能够闭合如 1 1 b 这样的断开之处，可使用跳跃技术。跳跃技术取不连续处的角度位置，沿着搜索的方向旋转“gap-angle”参数规定的度数，并且建立一个搜索矢量，以寻找不连续处的另一侧。

搜索矢量从处理环的内部开始搜索，并根据不连续处的角度位置，分别沿行或列搜索。搜索一直进行下去，直至遇到一个边缘像素，或者到达处理环的外边缘。如果在搜索过程中没有发现边缘像素，该透镜将被认为严重扭曲变形而被否定。一个不连续处无法用外推法进行搭接的事实指示出一个缺陷的存在，一个“跳跃搭接”被识别为一个特征。从最远的像素开始被处理直至发现不连续处的所有像素都被从轮廓线阵列中被消除，因为它们在边缘追踪时代表退回。

有时透镜边缘的某一部分断裂成这样的片断以至于对某一不连续点的搭接点使得整个追踪处理程序被置于边缘的某一独立的片段上进行，如图 1 3 a 中的 1 1 8。在这种情况下，一般探查不连续性的方法不再有效，因为追踪程序不可能掉头折回所需要的量。为了克服这种异常，使用了一种特殊的探查技术。这种技术将不连续处被搭接后马上把插入的像素的行和列数保持下来。如果进一步的追踪四次经过该插入像素，就探查到一个小的、分离的边缘部分。在这个断裂段上发现的最远的像素，被用来作为进行另一次搭接的位置。在图 13b 中表示出一种情形，其涉及复合搭接 1 2 0 和接着进行的如前面所讨论的“跳跃搭接” 1 1 6。

对复合的分离的各段进行搭接是一项重复过程，需要多少次，就进行多少次，以便使断裂的各段连结上。每次重复首先尝试外推法搭

接然后是跳跃搭接。

当边缘已经被追踪之后，并且所有的缺口被搭接之后，算法程序从每个边缘轮廓线上所发现的象素中提取六个不同的特征。这些特征是这样的：

径向偏差 (RD)

定域内梯度偏差 (LGD)

空间偏差 (SD)

不连续性 (D)

下降定域内梯度偏差 (DLGD)

单向定域内梯度偏差 (ALGD)

最后两个特征与定域内梯度偏差有关，增加它们是为了辨别一些非此无法探测到的特定类型的缺陷。

对每一个特征计算出的数值与阈值进行比较。所有的阈值作为用户参数都是可得到的。如果某一个特征值符合阈值标准，则该象素根据该特征被归为有缺陷的。有可能同一象素根据多项特征被归为有缺陷的。

参见图 1 4，所示为一种加强的边缘，图 1 4 a 是位于平面坐标系，图 1 4 b 位于极坐标系，该边缘具有可归入径向偏差一类的特征。径向偏差是指感兴趣象素的半径偏离开标称半径的距离。如果偏差等于或大于参数 “rad - dev - thr” 规定的值，则该象素被认为有缺陷。标称半径被定义为感兴趣的象素之前的 250 个和之后的 250 个轮廓线象素的平均半径值。如果某一象素因为径向偏差被归为有缺陷，则该偏差值被保留下来作为严重程度的指标。图 1 4 a 和图 14b 给出了透镜内边缘 9 4 和透镜外边缘 9 6。此外图 1 4 b 还示出了按

上述方法计算出来的内、外边缘的理想半径。图 1 4 b 还示出了三个径向异常特征的例子，1 2 4、1 2 6 和 1 2 8。用于完成径向偏差特征提取的方程式如下所述：

$$RD = R_i - \left(\sum_{n=i-250}^{i-1} R_n + \sum_{m=i+10}^{i+250} R_m \right) / 500$$

$$n=i-250 \quad m=i+10$$

R = 半径值，

i 、 n 、 m = 外轮廓线索引数。

如果 ($RD \geq \text{阈值}$ 或 $RD \leq -\text{阈值}$)

则感兴趣的像素是有缺陷的。

下一个要提取的特征是定域内梯度偏差。LGD 所看的是感兴趣的像素的梯度值与其定域内相邻像素的平均值的偏离值。相邻像素是指那些在边缘轮廓线上的并最靠近感兴趣的像素的那些像素。参见图 1 5，感兴趣的像素用 i 代表。根据 LGD 判断一个像素是否有缺陷的阈值是参数 “ $\text{grd} - \text{dev} - \text{thr}$ ”。下面的方程表示该特征实现的实际形式：

$$LGD = \frac{G_{i-1} + G_i + G_{i+1}}{3} - \left(\sum_{n=i-11}^{i-2} G_n + \sum_{m=i+1}^{i+11} G_m \right) / 20$$

$$n=i-11 \quad m=i+11$$

其中，

G = 梯度值

i 、 n 、 m = 外轮廓线索引数。

如果 ($LGD \geq \text{阈值}$) 则感兴趣的像素为有缺陷的。

这个提取定域内梯度偏差的过程由图 1 6 示出。图 1 6 a 是示意性示出加强的透镜边缘, 图 1 6 b 是极坐标系中的梯度信息。如同图 1 6 a 所示定域内梯度偏差 1 3 4 仅仅发生在加强的透镜边缘图象的边缘, 当被追踪并在极坐标系中显示时, 它呈现为不成对的不规则性 1 3 6。如果某一像素被根据 LGD 归为有缺陷的, 则偏差量被保持下来作为严重程度的指标。

特征提取的下一特征是不连续性。如前面所讨论的, 不连续性缺陷是由利用跳跃搭接法搭接边缘引起的。在特征提取中它的识别是通过对看一个像素与其下一个像素之间的角度位移差。不连续性不包含任何严重程度的指标信息, 仅仅指出发现了一个跳跃。不连续点两端的起始像素都被认为是有缺陷的。

下一个要提取的特征是下降定域内梯度偏差。下降定域内梯度偏差类似于已提取的定域内梯度偏差特征。类似于 LGD 、 $DLGD$ 看的是感兴趣的像素的光强梯度值与其定域内的相邻像素的平均值之间的偏离量。其区别是, 所采用的相邻像素数更多, 而且在感兴趣的像素周围的一个较大距离内的像素不被采用。 $DLGD$ 被设计使得只对光强梯度小于其相邻像素的偏差值敏感, 因此被称为“下降”定域内梯度偏差。

用于测定某一像素是否因为 $DLGD$ 而有缺陷的阈值取自参数 “ $dip - lgd - thr$ ”。下面的方程式表示这个特征的实际实现方式:

$$DLGD = \frac{G_{i-1} + G_i + G_{i+1}}{3} - \left(\sum_{n=i-35}^{i-5} G_n + \sum_{m=j+5}^{i+35} G_m \right) / 60$$

其中，

G = 梯度值

i 、 n 、 m = 外轮廓线索引值。

如果 ($DLGD \leq \text{阈值}$) 则感兴趣的象素为有缺陷的。

$DLGD$ 特征被特别用于识别其它特性无法识别的边缘小裂口。

如果某一象素根据 $DLGD$ 被归入有缺陷的一类，则其偏差值被储存下来作为严重程度的指标。

另一个要被提取的特性是单向定域内梯度偏差 ($ALGD$)。在提取该特征时，用于计算偏差的相邻象素取自感兴趣象素的同一边。感兴趣象素之前的 20 个象素被用于算出相邻象素的平均值。但是前面四个紧挨着感兴趣象素的象素不予采用。 $ALGD$ 看的不仅有正的梯度偏差也有负的梯度偏差。

用于进行比较的存储在参数 “aux - lgd - low” 和 “aux - lgd - up” 中。下列方程式表示出 $ALGD$ 特征完成：

$$ALGD = \frac{G_{i-1} + G_i + G_{i+1}}{3} - \left(\sum_{n=i-25}^{i-5} G_n \right) / 20$$

其中，

G = 梯度值

i 、 n 、 m = 外轮廓线索引值。

如果 $ALGD = \text{上阈值或下阈值}$ 或 $ALGD \leq \text{阈值}$ 则感兴趣的像素是有缺陷的。

最后一个提取的特性是空间偏差 (SD)。空间偏差所测量的是半径相对于角度位移的变化。如果在一个小的角度范围内半径发生急剧变化, 则有可能出现了一个缺陷。图 1 7 a 表示出一个平面坐标系中的加强的透镜边缘, 图 1 7 b 被转换成了极坐标, 表示出内透镜边缘 9 4 和外透镜边缘 9 6 的理想半径 1 2 2。

图 1 7 b 中画出了空间偏差特性 1 3 6, 以及它的半径 1 3 8 相对于角度 1 4 0 的变化。

空间偏差的完成由下列方程给出:

$$\Delta R = (\text{轮廓像素}_{i+2} \text{ 的半径} - \text{轮廓像素}_{i-2} \text{ 的半径})$$

其中, $\Delta R = \text{半径的变化}$

$i = \text{感兴趣的像素的索引}$

$$\Delta \theta = (\text{轮廓像素}_{i-2} \text{ 的角度位移值} - \text{轮廓像素}_{i+2} \text{ 的角度位移})$$

其中

$\Delta \theta = \text{角度位移的改变}$

$i = \text{感兴趣的像素的轮廓线索引}$

$$SD = \Delta R / \Delta \theta$$

如果 ($SD \geq \text{正阈值}$ 或 $SD < 0$)

则感兴趣的像素是有缺陷的。

如果某一像素根据以上方程的结果被归为有缺陷的, 不保留任何严重程度指标信息。算法程序的进一步处理仅根据这样的事实: 即像素已根据 SD 被视为有缺陷。

将缺陷分组包括三个步骤。前两个步骤分别单独在内外边缘上进行，最后一个步骤将来自内外边缘的信息结合起来。在内外边缘的分组处理完成以后，对所得的相近组进行比较，以便观察来自内边缘的组中间是否应与来自外边缘的组相结合。如果出现这样的合并，就形成了一个混合缺陷组。

开始步骤是逐个像素地观测每个缺陷像素，并判断它是否属于一个较大缺陷的一部分。如果一个像素被确认是一个较大缺陷的一部分，则它将被放入一个叫作缺陷组的结构中。

第二步确认这些缺陷组中是否有一些应当被相互结合以形成较大的组。最后一个步骤是将内外缺陷组互相比较，以判断它们是否应当相结合。其结果是，可能出现的最大的那个缺陷组代表透镜的可能缺陷。这一点又能够最精确代表真正的缺陷程度。更清楚地说，复合缺陷比单个的缺陷更为严重，并且仅仅出现在较严重的边缘缺陷中。

如前所述，该过程的开始是逐个像素的分组。第一个遇到缺陷像素被自动放在一个单独的像素缺陷组以开始该过程。后续的缺陷像素的角度位移与目前正进行处理的缺陷中最远的像素进行比较。如果像素处于参数“prox - zone”规定的角度位移之内，它被放在该组之内，该组的最远角被加以修正。如果缺陷像素未落入该正在被处理的组中，则认为遇到了一个新的缺陷。结果是，一个新的仅仅包含有现行缺陷像素的缺陷组被建立起来，并成为目前进行处理的组。该过程一直持续下去，直到所有的边缘上的缺陷像素被检查出来为止。

如果轮廓线上的非缺陷像素被发现处于被放入同一个缺陷组的缺陷像素之间，它们也要被包括该缺陷组之中，并且被归入分组类型的缺陷像素而不再是非缺陷的像素。

分组过程的第二个步骤是如下所述。同一个缺陷有可能被二个以上的缺陷组表示。为了排除这种混乱状况，为所有边缘上发现的缺陷组制订了一个处理数据体的循环。进行两个比较。一个比较针对一组的起始角度位移和另一组的结束角度位移进行比较检验。第二个比较将同一组的结束角度位移与另一组的起始角度位移进行比较检验。如果两个比较中有一个的结果即角度位移的改变小于“prox - angle”规定的数量，这两组被合并。一组的起始角度如果与另一组的结束角度相当接近，则前一组被包括在该组之内。被包括的组将其信息进行转换然后被无效掉。

最后，在角度上与内外边缘缺陷相对应的缺陷组被组合在一起。这个分组与单独在内外边缘上进行的分组是类似的。对各组的起始和结束位置进行对比。再额外进行一个对比来确定某一组是否完全被另一组所包围。如果任何这些比较导致一个合并，则建立一个包含被合并的两个缺陷组的信息的单独结构，这两个原来的组被无效掉。

在缺陷象素被识别出来并且上述分组操作已完成以后，对每个缺陷组赋以严重程度指标分数。严重程度指标分数是对该组内所有象素所赋以的分数的总和。当同一个象素被两个以上的缺陷类型归入有缺陷的一类时，其结果是那个特别的象素具有几倍的分数的。

每一种缺陷被赋予一个权重，它可以允许不同的缺陷相对而言具有不同的严重程度。所有权重的数值可以由用户手中的参数进行控制。RD、LGD、SD、D、DLGD和ALGD的权重可以分别在下列参数中找到：“rd - weight”，“lgd - weight”，“SD - weight”，“disc - weight”，“dip - lgd - wgt”和“aux - lgd - wgt”。

与其它三种缺陷不同，RD、LGD、和DLGD对每个给定的象素保留了严重程度的信息。对这些严重程度信息进行规范处理，并乘以被赋给缺陷象素的缺陷类型权重。进行规范化处理是因为不同特性的数值的范围各不相同因而是不可相比的。在规范化处理之后，每个权重后的分数将落在1.0到2.0的范围之内。规范化的范围是通过所使用的阈值的数值和该特性所能获得的最小或最大理论值求得的。

作为一种实施例子，如果某个象素的特征与标称值相同，它的权重后的分数为1.0。相反，如果某个象素的特征与可能的极小值或极大值相同，则权重后的严重程度分数为2.0。径向偏差和定域内梯度偏差的最大理论值分别由“max - rd”和“max - lgd”中发现的参数值确定。下降(Dip)定域内梯度的最小理论值可在参数“min - dip - lgd”中找到。

SD、ALGD、组和D类型缺陷不进任何形式的规范化。“不连续性”和“组”是布尔型缺陷，其数值为1或0。空间偏差和ALGD不包含值得保存的足够的严重程度信息。

对六种缺陷类型的每一种，下面给出了象素缺陷严重程度方程式，以及任何适当的规范化处理和权重。

RD分数 = $(1.0 + (\text{象素RD的绝对值} - \text{RD阈值}) /$

$(\text{RD的最大理论值} - \text{RD阈值})) * \text{rd-weight}$

其中，RD分数 = RD分类赋予某一象素的全部分数，

象素RD值 = 感兴趣象素的RD特性值，

RD阈值 = 用于是否存在一个RD缺陷的阈值

RD最大理论值 = RD特征的最大可能值

rd - weight = 与 R D 缺陷类型相关的权重。

$$\text{LGD 分数} = (1.0 + (\text{像素的 LGD 值} - \text{LGD 阈值}) / (\text{LGD 的最大理论值} - \text{LGD 阈值})) * \text{lgd - weight}$$

其中，

LGD 分数 = LGD 分类赋予某一像素的全部分数，

像素 LGD 值 = 感兴趣像素的 LGD 特征值，

LGD 阈值 = 用于判断是否存在一个 LGD 缺陷的阈值，

LGD 最大理论值 = LGD 特征的最大可能值，

lgd - weight = 与 LGD 缺陷类型有关的权重。

$$\text{DLGD 分数} = (1.0 + (\text{像素 DLGD 值} - \text{DLGD 阈值}) / (\text{DLGD 最大理论值} - \text{DLGD 阈值})) * \text{dip -lgd - weight}$$

其中，

DLGD 分数 = DLGD 分类赋予一个像素的全部分数，

像素 DLGD 值 = 感兴趣的 DLGD 特征值，

DLGD 阈值 = 用于判断是否存在 DLGD 缺陷的阈值，

DLGD 最大理论值 = DLGD 特征的最大可能值，

dip -lgd - weight = 与 DLGD 缺陷类型相关的权重。

$$\text{SD 分数} = \text{Sd - weight}$$

其中，

SD 分数 = SD 分类赋予一个像素的全部分数，

Sd - weight = 与 SD 缺陷类型相关的权重。

$$\text{组分数} = \text{grp - weight}$$

其中，组分数 = 组分类赋予一个像素的全部分数，

grp - weight = 与组缺陷相关的权重，

$$D i s c \text{ 分数} = disc - weight$$

其中，

disc 分数 = 不连续性分类赋予一个象素的全部分数

disc - weight = 与 Disc 缺陷有关的权重。

$$A L G D \text{ 分数} = aux - lgd - wgt$$

其中，

组分数 = 组分类赋予一个象素的全部分数，

aux - lgd - wgt = 与 A L G D 缺陷有关的权重。

如上所述，当象素电平判断已完成以及有缺陷的象素被放入缺陷组中之后（这包括：合并重叠的缺陷，将相近的缺陷分组，对内外边缘上的位于同一角度位移的缺陷分组），一个缺陷组严重程度分数被计算出来。这个缺陷组严重程度分数表现出所有缺陷类型所赋予的全部严重程度分数，并且是通过以下方程计算的：

“缺陷组”严重程度分数 =

$$\begin{aligned} & \sum_{\Omega} R D \text{ 分数} + \sum_{\Omega} L G D \text{ 分数} + \sum_{\Omega} D L G D \text{ 分数} + \sum_{\Omega} S D \text{ 分数} + \\ & \sum_{\Omega} D i s c \text{ 分数} + \sum_{\Omega} \text{组分数} + \sum_{\Omega} A L G D \text{ 分数} \end{aligned}$$

其中：

“缺陷组”严重程度分数 = 所有在该组内的缺陷象素赋予“缺陷组”的所有分数，

Ω = 包括在一个给定“缺陷组”之中的所有象素的总和

当上述计算做完之后，通过一个可由操作者进行定义的抛物线函数对每一个缺陷组进行加权。抛物线函数对于较大的缺陷给予一个相应大的严重程度分数。例如，一个比两个较小缺陷大两倍的缺陷其最

终的严重程度大于两个较小缺陷之和。

加权函数记载在以下方程式中：

加权的“缺陷组”

$$\begin{aligned} \text{严重程度分数} = & a_系数 * (\text{“缺陷组”严重程度分数}) \\ & + b_系数 * (\text{“缺陷组”严重程度分数}) \end{aligned}$$

其中，

a_系数 = 定义抛物线加权函数所用的计算参数

b_系数 = 定义抛物线加权函数所用的计算参数，

所得的加权后的分数随后被换算，以便它落入 0 到 999 的范围。

换算系数是由下列方程求得的：

$$\text{换算系数} = 999.0 / \text{最大加权分数}$$

其中，

$$\begin{aligned} \text{最大加权分数} = & a_系数 * (\text{最大_分数})^2 + \\ & b_系数 * (\text{最大_分数}) \end{aligned}$$

其中，

最大_分数 = 可由操作者根据经验定义的参数值

缺陷组的分数大于 999 的部分被舍掉。所有缺陷组分数的总和是一个给定透镜的最后得分。如果该分数大于或等于阈值（作为运算参数可得到）则该透镜被废弃。否则，透镜是合格的。

虽然透镜分数的最终结果是透镜通过检验或通不过检验，但是所有的中间数据、计算值和分数都是可得到的，以便为考虑透镜的质量和在透镜上观察到的特性的类型提供信息。

对于一个本领域普通技术人员来说很清楚，该信息可以作为按透镜数的统计信息随透镜给出，或者作为可视输出在计算机监视器上给

出。

上述算法是在上述设备上完成的，并且眼用透镜得以检验。这些透镜包含 5.8 % 的水，是 Johnson & Johnson Vision Products AcuvueTM 的软质水凝胶隐形眼镜。在第一次检验中涉及 128 只透镜。

首先，由熟练的生产线透镜检查员使用图象放大系统对这些置于去离子水中的透镜进行检验。检查员把每只透镜分成合格和不合格，并且对每个缺陷辨别出其缺陷种类。

当使用本文描述的自动检验系统对透镜进行检验时，每个透镜被人工放置在检验包装盒的中央，以避免出现照明问题。

当自动检验系统摄取了全部 128 个透镜的图象之后，与人检验员相比发现有 2.5 分的偏差。机器检验和人检验透镜之间的比较结果在表 1 中给出。

表 1 机器 / 人工检验汇总 1

	数量	百分比
透镜的总数	1 2 8	1 0 0 . 0
一致	1 0 3	8 0 . 4
不一致	2 5	1 9 . 5
机器过于否定：		
由于光照	7	5 . 5
人眼不曾发现	<u>5</u>	<u>3 . 9</u>
机器过于否定的总数	1 2	9 . 4
机器过于肯定：		
由于光照	6	4 . 7
被冲洗掉的缺陷	4	3 . 1
未被机器发现的小缺陷	<u>3</u>	<u>2 . 3</u>
机器过于肯定的总数	1 3	1 0 . 1

虽然结果暗示，机器检验的不正确率为 1 9 . 5 %，其时有 1 2 例（9 . 4 %）机器过于严格而且否定了检验员已通过的透镜。即：机器拒绝了。还有 1 3 例（1 0 . 1 %）机器过于肯定，而通过了检验员已经认为是坏透镜的透镜（即，机器认可了。）

在机器过于肯定的情形，可以看出机器检验系统的透镜光照没有充分调节，但能够被调整。在机器过于否定的情形，可以看出机器参

数的设定过于敏感，需要调整。然而，没有一个大缺陷逃脱探测，而且大多数漏掉的缺陷的尺寸都是位于边缘的，小于 50 微米。没有一个半拉透镜或缺少的透镜逃脱检查。

在十二个机器过于否定的情形中，由于光照的问题，造成七个图象边缘的薄弱，有五种情形是缺陷真实存在，但没有被检验人员看到，但是被机器也被第二个检验人员看到了。在十三种机器过于肯定的情形，四个透镜不再有缺陷，或者是由于外来的因素将其冲洗掉了。六个图象由于照明的问题具有弱的边缘，三个透镜上具有的缺陷太小，以至检验人员看不见。

因此，在被检查的 128 个透镜中，大约 20% 与人工检验不相符合。在这些错误中，56% 属于照明引起，36% 属于检验人员的错误或者在处理过程中透镜情况的改变引起，12% 是不正确的判断。这 12% 相当于仅有 2.3% 的总体不正确判断。

由于不一致判断的大多数是由于照明问题引起，故进行了一个调查，发现光源的不均匀，特别是过度光照，导致透镜边缘被冲淡缺陷不可见。

导致人和机器检验之间不一致结果的另一原因是透镜在人工检验和机器检验过程中都没有被搅动，从而水中的颗粒和透镜上的缺陷之间的区别不容易很明显。

通过采用一个更均匀和漫射的照明光源可以改进透镜的照明。当照明改进之后，机器检验了 128 个另外的透镜。在这个检验方案中，

表 2 机器/人工检验汇总 2

	数量	百分比
透镜总数	1 2 8	1 0 0 . 0
正确	9 8	7 6 . 6
不正确	3 0	2 3 . 4
机器过于否定		
未聚焦的图象	1 0	7 . 8
由于照明	5	3 . 9
透镜上污物	3	2 . 3
污染的透镜	1	0 . 8
不连续性	1	0 . 8
未知的	<u>2</u>	<u>1 . 6</u>
错误不否定总数目	2 2	1 7 . 2
不包括未聚焦情况的总数	1 2	9 . 4
机器过于肯定：		
由于照明	0	0
小缺陷	7	5 . 5
冲洗掉的缺陷	1	0 . 8
错误肯定的总数	8	6 . 3

每个透镜有两个图象被机器摄取，并与检验人员的检验报告进行对比。机器检验结果和人工检验的结果在表 II 中给出。

如同可从表 2 中给出的数据中所看到，一个否定的新种类，“未聚焦的图象”被发现。这是由于透镜不适当地放置在照相机下，导致其一部分处于焦距之外。作为系统性能的测试，未聚焦的图象不是可靠性的指标，而是一种操作失误，那些数据点可以适当地去除。

不考虑错误放置透镜引起的焦距问题，检验人员和机器不符合的透镜百分率只有 15.6%。这相对于第一次的 128 个透镜来说是一个 3.9% 的提高。

在第三次对比实验中，192 只透镜先由人工检验，然后用机器检验再次。其结果与前面的实验相似。在总共 384 个图象中，317 只，82.6% 与人工检验相一致。作为对处理算法和所得结果分数一致性的测定，机器摄取的两个图象都被机器处理，其中有 84% 的情形，第二次运算操作的分数与第一次的相同。

虽然检验被设计成首先用于检验透镜的边缘，但因为用于定位透镜边缘的搜索矢量，缺失的透镜也能够被发现。因为透镜边缘搜索程序进行数次，有孔洞的透镜能被探测到，多余的碎片能被发现，有边缘裂口的透镜也能被发现。

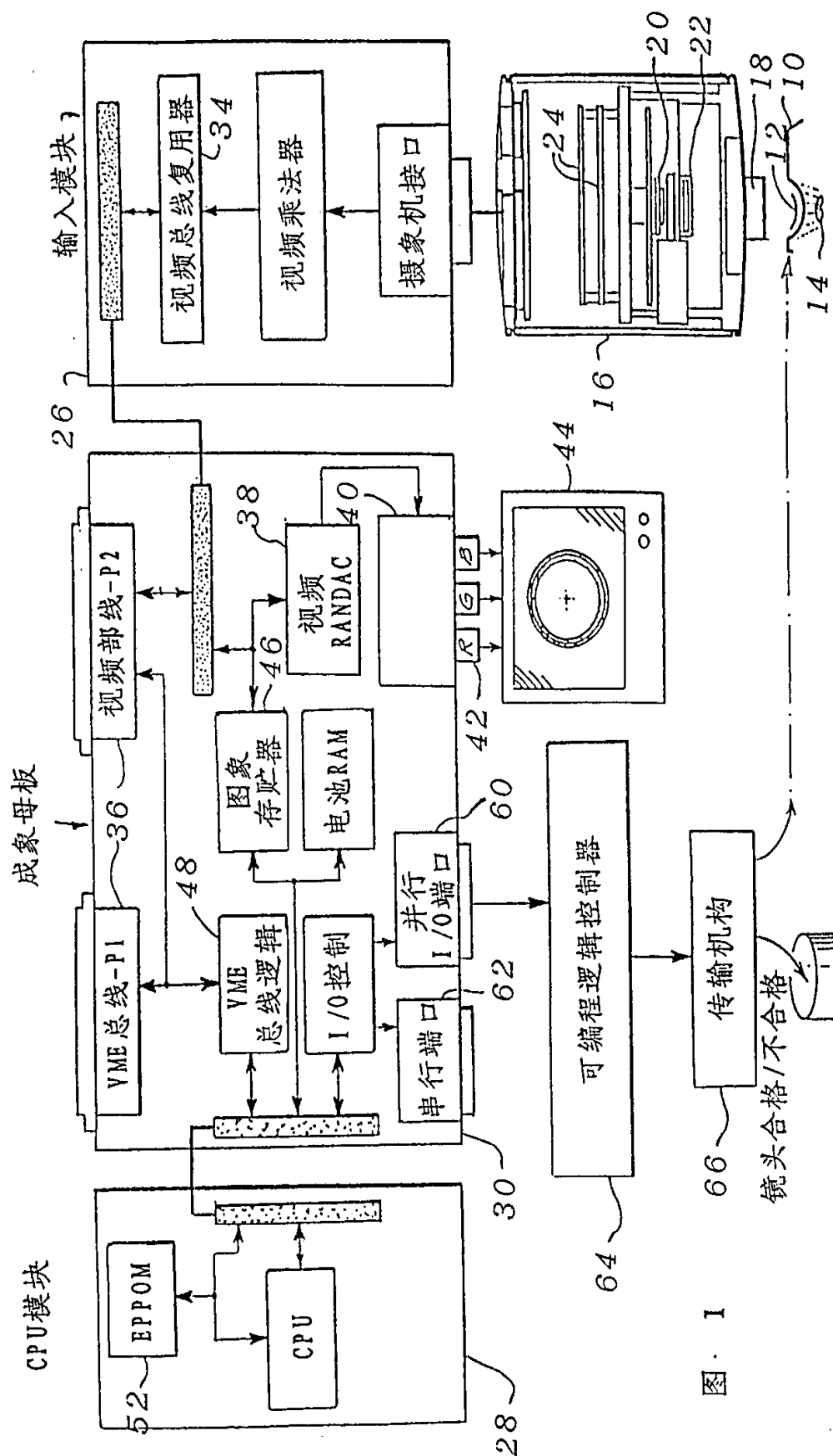
在下面的表 3 中，给出了第三次检验的结果，其中分成了“机器过于否定”、“机器过于肯定”和正确的数目。机器检验中仅有 8.1% 的错误否定和 9.4% 的错误肯定。

表 3

托盘	机器过	机器过	不相符的	正确的	正确率
1	7	9	1 6	4 8	7 5
2	9	1 1	2 0	4 4	6 9
3	2	4	6	5 8	9 1
4	8	3	1 1	5 3	8 3
5	4	2	6	5 8	9 1
6	1	7	8	5 6	8 8
总计	3 1	3 6	6 7	3 1 7	8 3

前两个托盘的结果比后面四个的要坏，这是因为水中发现已经有尘埃污染。因此，这不是系统性能的表现。

总之，检验人员和机器相符合 3 1 7 次，不符合 6 7 次。考虑到从检验的观点来看，一致性和机器的否定判断是可以接受的，透镜处理具有 9 0 . 6 % 的精度。



一
•
四

镜头合格/不合格

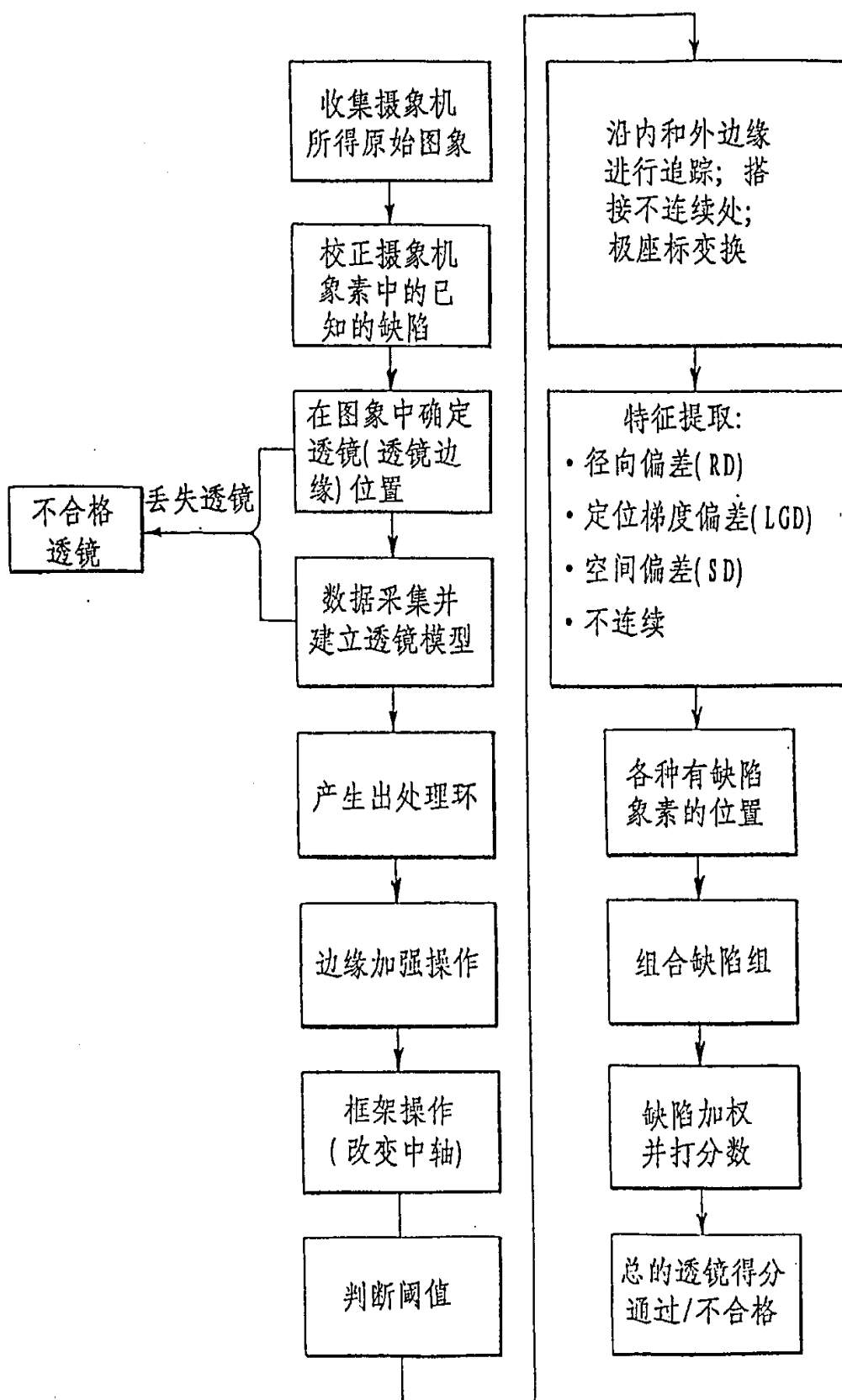


图 2

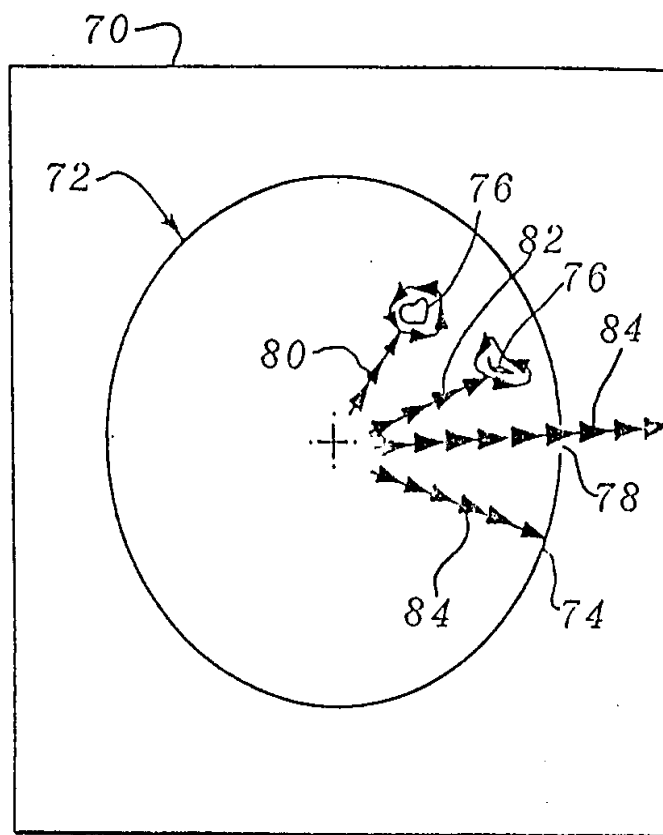


图 3

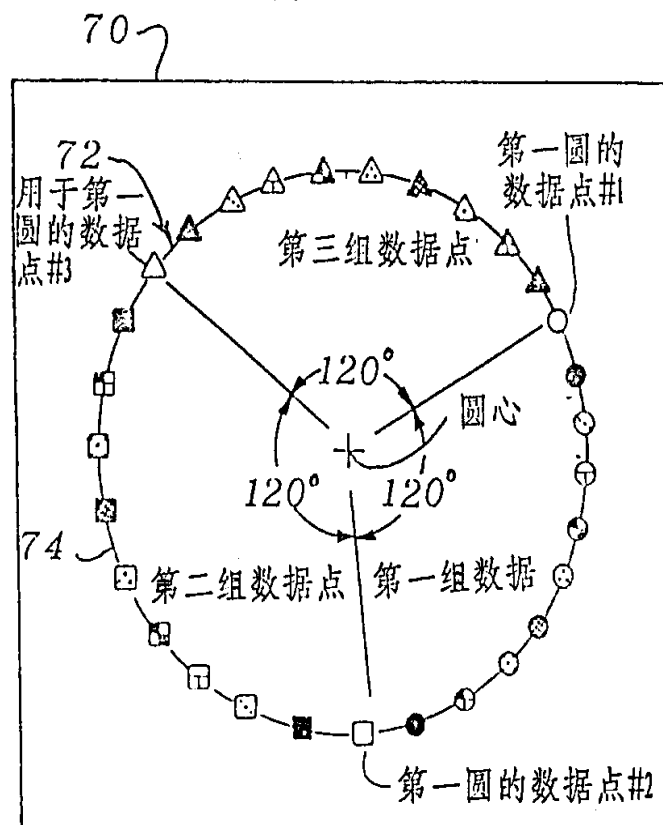


图 4

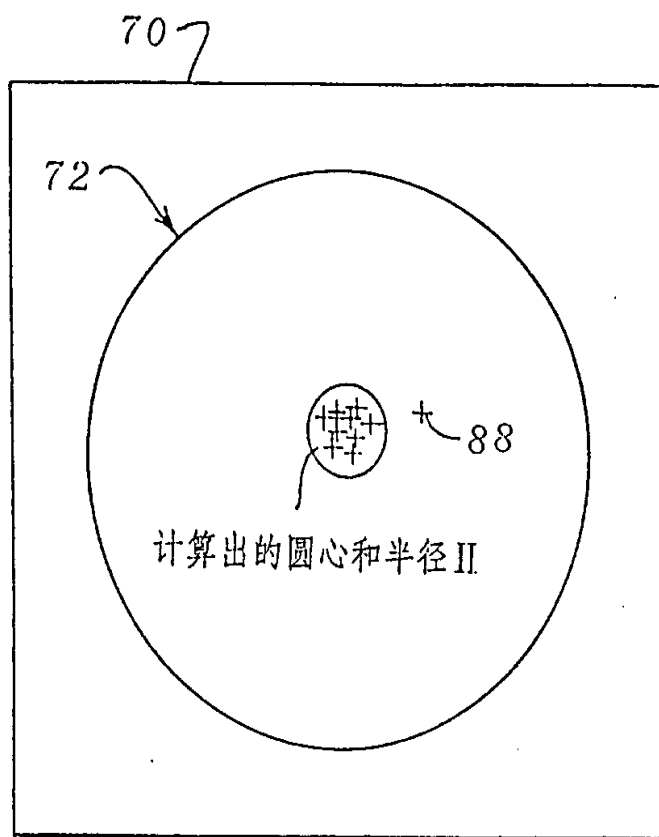


图 5

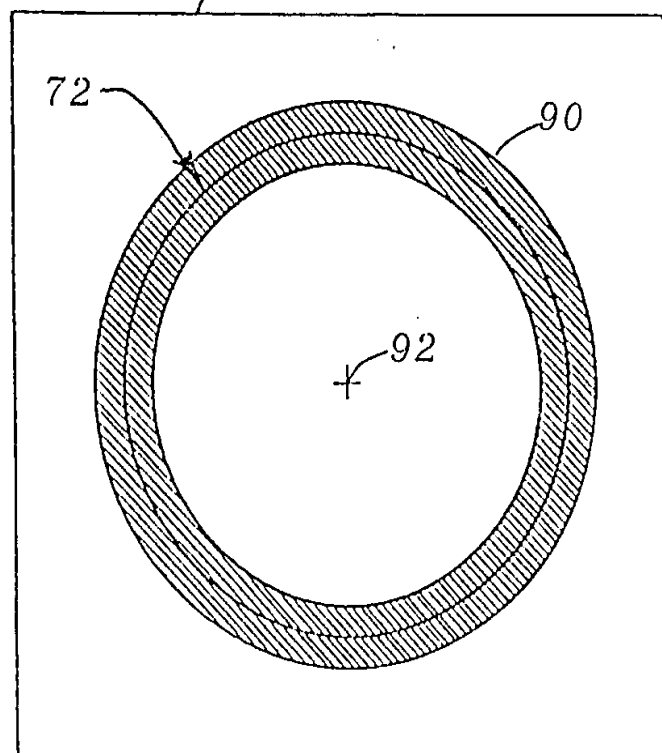


图 6

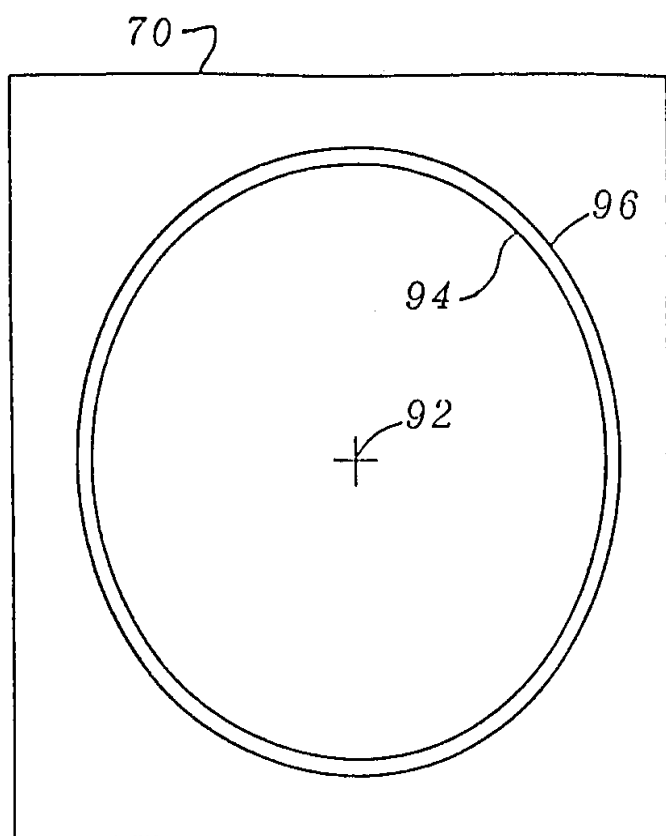


图 7

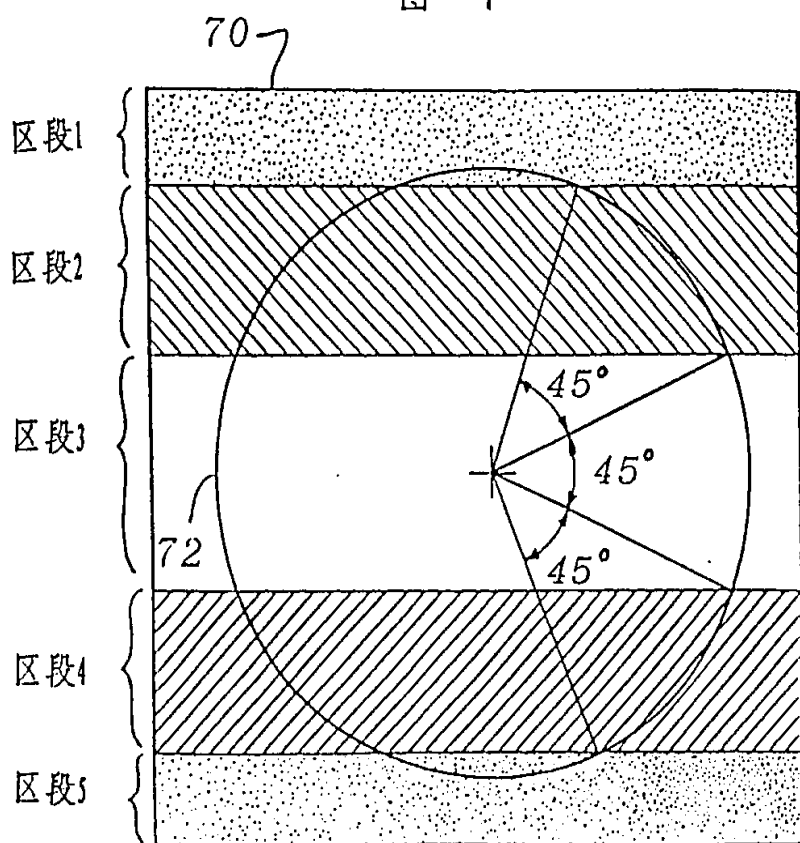


图 8

$P_{i-1,j-1}$	$P_{i-1,j}$	$P_{i-1,j+1}$
$P_{i,j-1}$	$P_{i,j}$	$P_{i,j+1}$
$P_{i+1,j-1}$	$P_{i+1,j}$	$P_{i+1,j+1}$

图 9

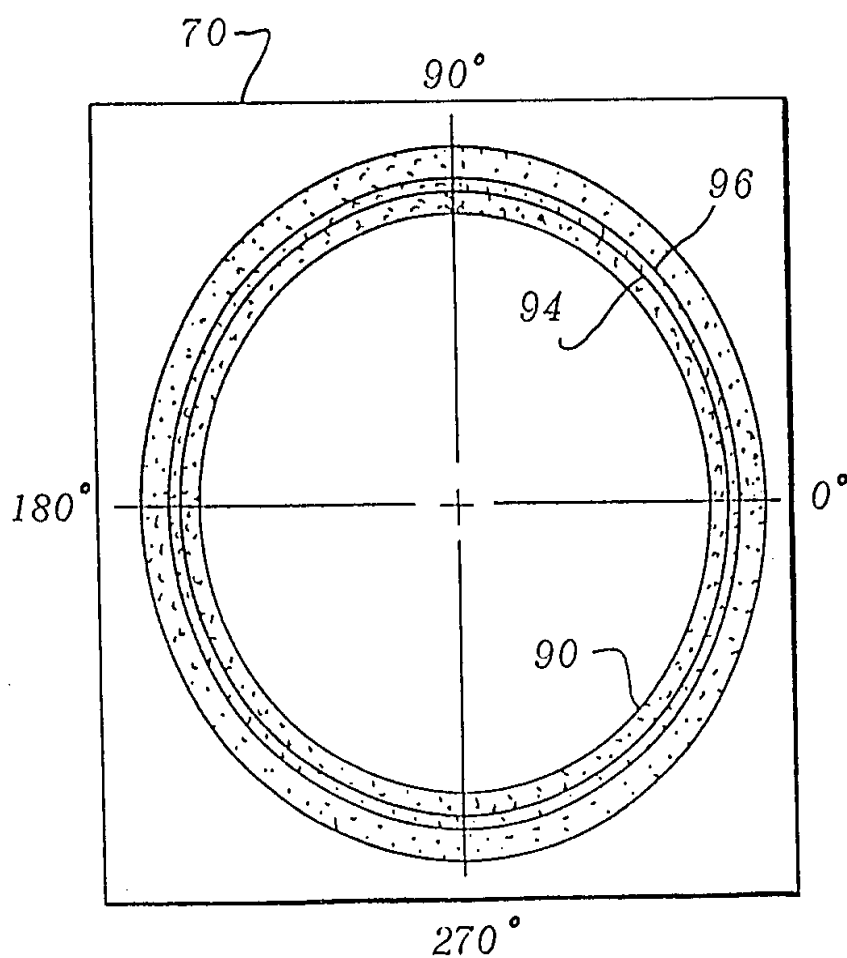
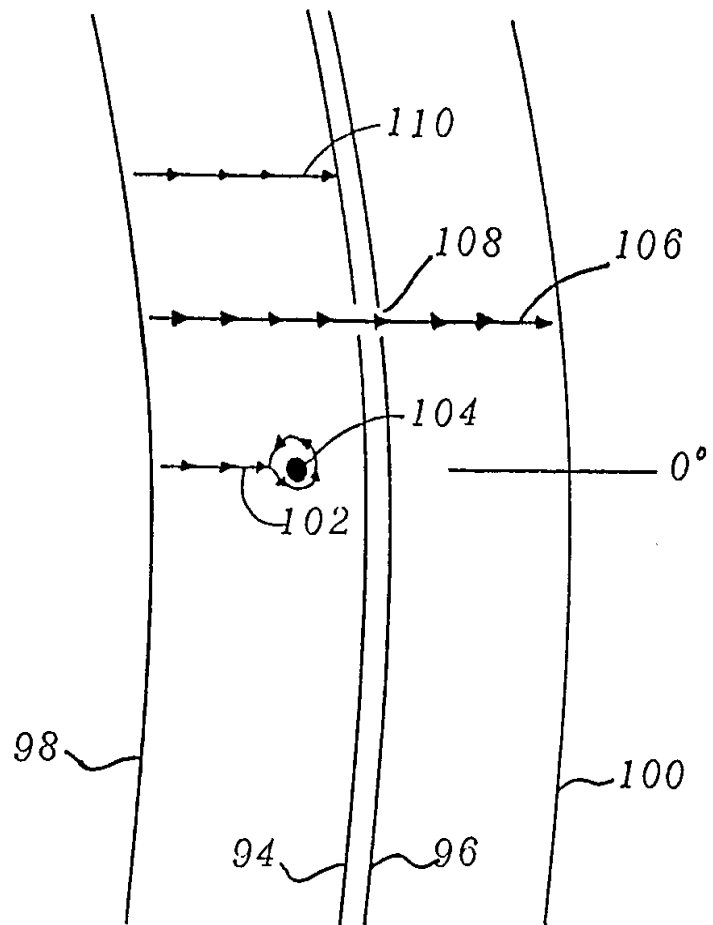


图 10
6

图 11



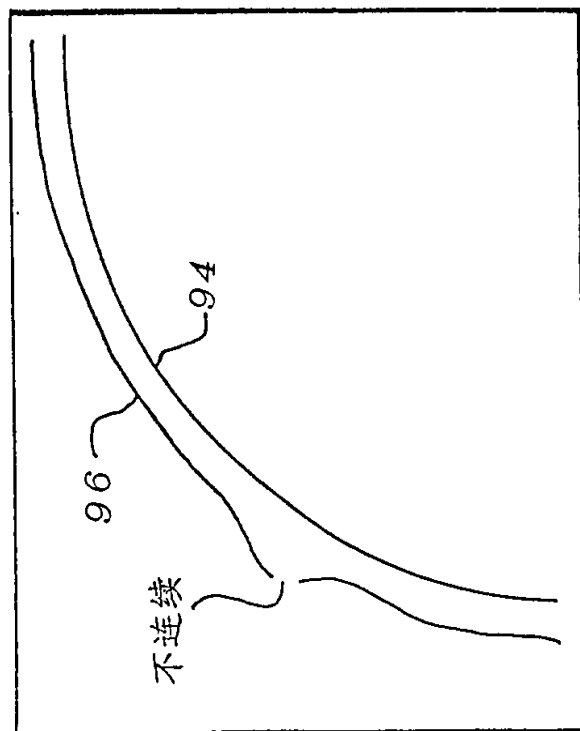


图 12a

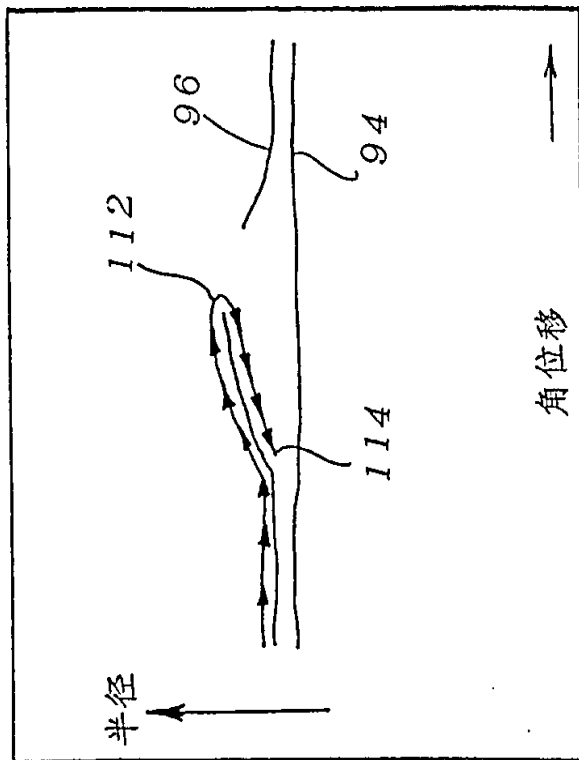


图 12b

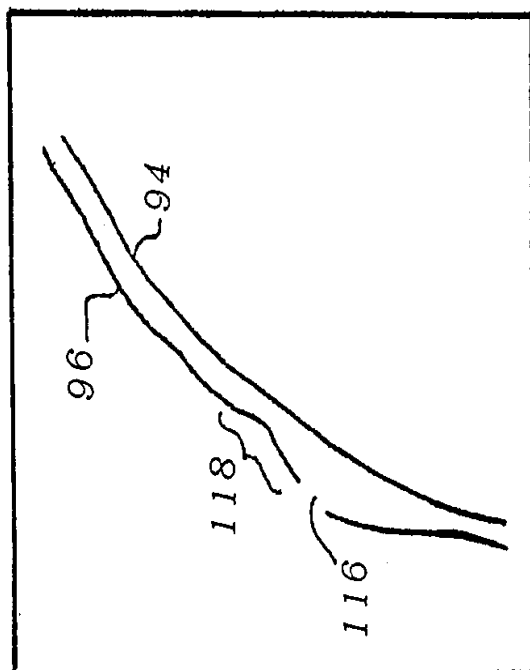


图 13a

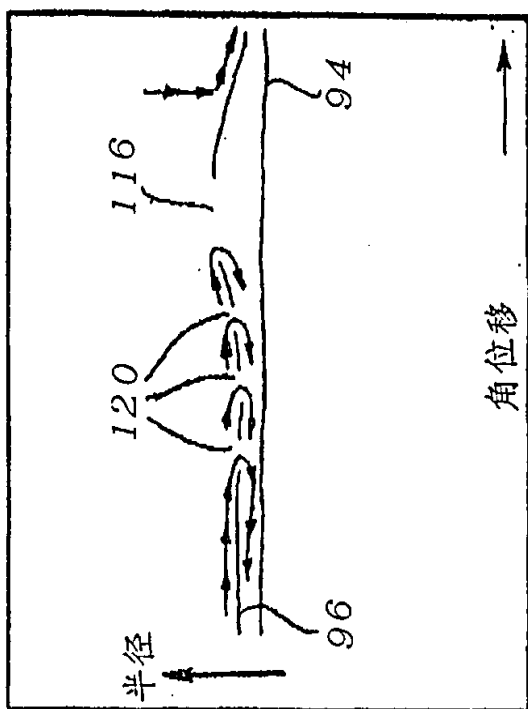


图 13b

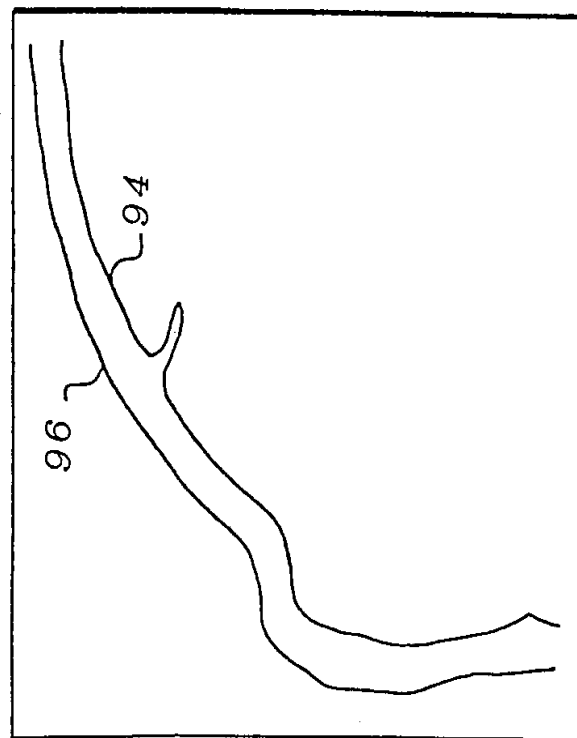


图 . 14a

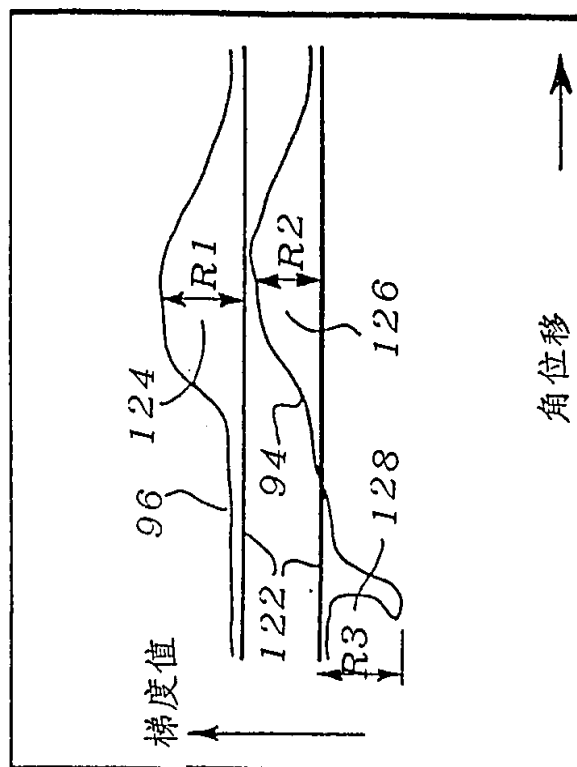


图 14b

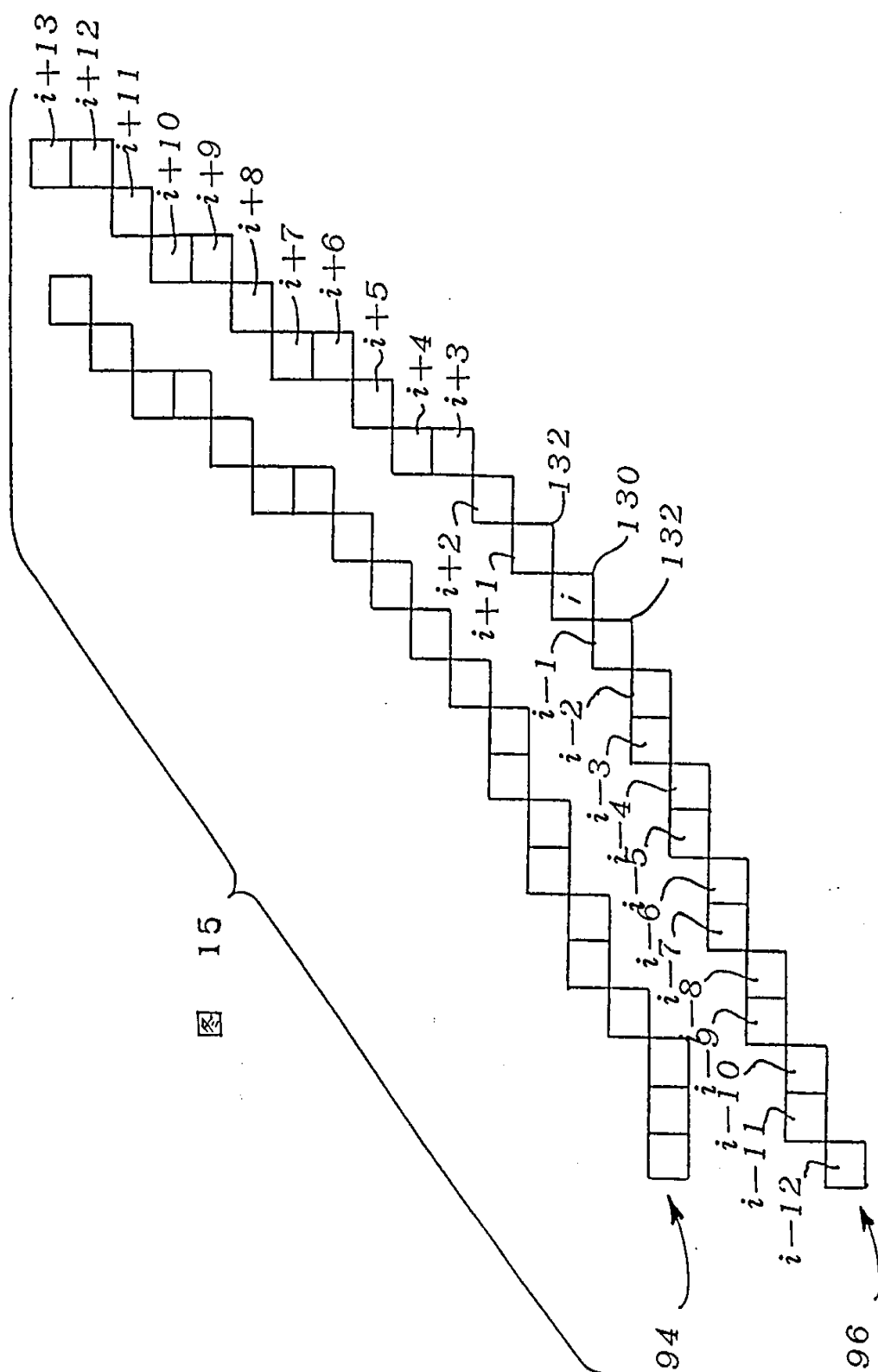


图 15

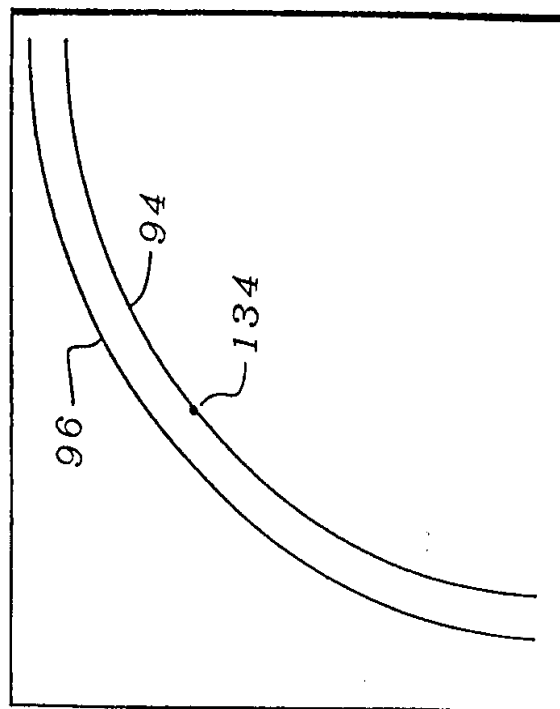


图 16a

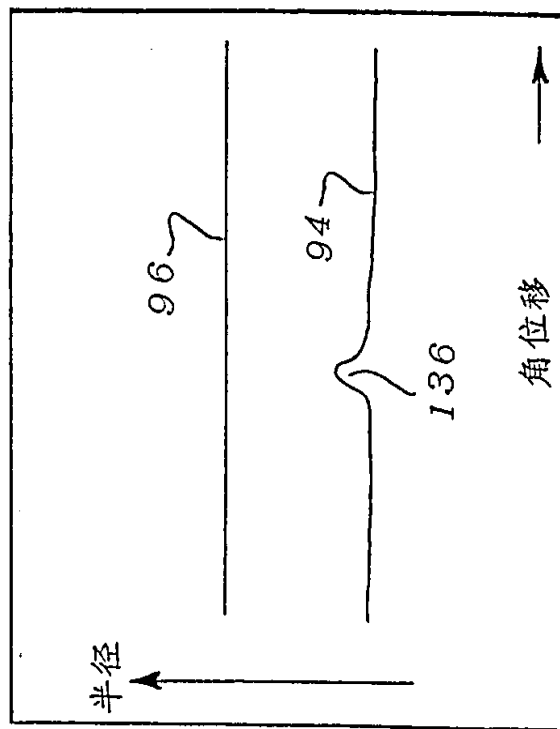


图 16b

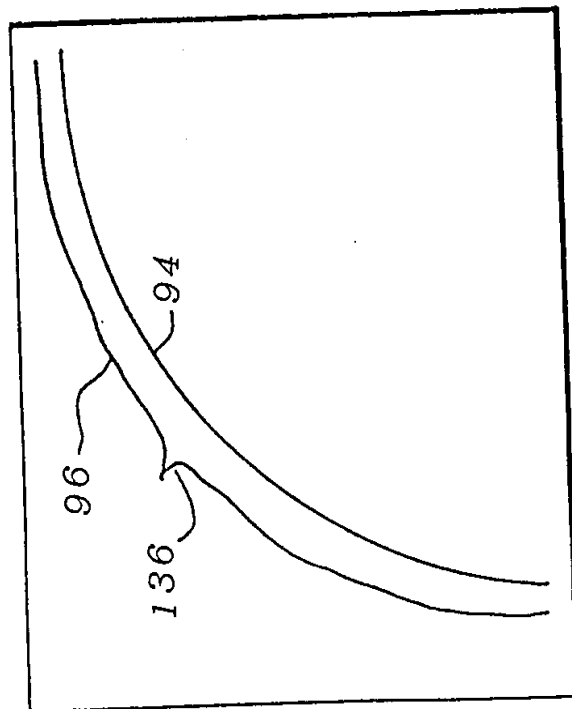


图 17a

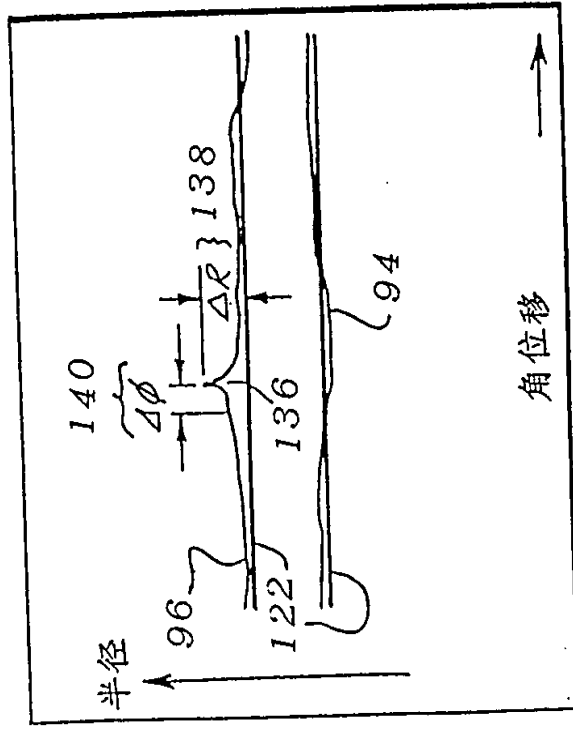


图 17b