



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 00135539.2

[45] 授权公告日 2004 年 8 月 4 日

[11] 授权公告号 CN 1160667C

[22] 申请日 2000.10.7 [21] 申请号 00135539.2

[30] 优先权

[32] 1999.10.7 [33] JP [31] 287218/1999

[71] 专利权人 精工爱普生株式会社

地址 日本东京都

[72] 发明人 宫坂德章 柳泽重一 大本隆二

审查员 涂洪文

[74] 专利代理机构 中国专利代理(香港)有限公司

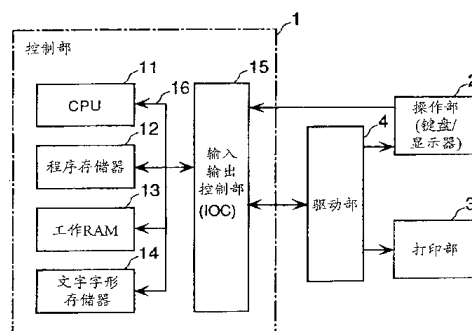
代理人 叶恺东

权利要求书 5 页 说明书 26 页 附图 53 页

[54] 发明名称 轮廓平滑化处理方法及其装置

[57] 摘要

本发明提供以规定某种文字字形的文字作为点阵状的位图数据,根据上述位图数据显示该文字的文字输出装置所使用的方法。上述文字放大时,把该位图数据一次变换成矢量数据。然后,在上述位图数据的倾斜方向上,包括邻接数据列(数组)为不连续的部分在内,从上述变换矢量的始点开始向该矢量方向的终点进行新的引线,把该新的引线所生成的线的内部涂满,再生成位图数据。



1、一种将点阵状的位图数据所规定的位图字形的文字图像的轮廓平滑化的方法，包括下述步骤：

5 在上述文字图像放大时，把上述位图数据一次变换成矢量数据；

 在上述位图数据开始到结束之间，给定数目的矢量是连锁的，根据每一矢量位移的连锁，在新的位图上进行引线，制作上述放大文字图像的轮廓；以及
 用点涂满上述引线生成的轮廓的内部，生成表示上述放大文字图像的新的位图数据，

10 上述进行引线的步骤包括以下步骤：

 根据上述矢量数据求出作为基准的轮廓坐标；

 分别与上述给定数目的矢量连锁中变化的模式对应，参照预先定义的补正数据对上述作为基准的轮廓坐标进行补正，以此作为补正轮廓坐标；以及

 将该补正轮廓坐标的各邻接坐标彼此之间用线连接。

15 2、一种将点阵状的位图数据所规定的位图字形的文字图像的轮廓平滑化的方法，包括下述步骤：

 根据预先规定的条件，参照表示各点矢量化与否的矢量化后的信息，对于上述的位图数据求出对应于变换之前矢量数据的开始的上述位图数据的始点坐标；

20 从上述始点开始，按照给定的顺序确定位于对应的前面点的下一位置的邻接点的方向，根据该确定的方向，把表示上述位图数据的各点变换成上下左右4个方向中的任一方向的矢量，由此，把上述位图数据变换成正交矢量行列，同时更新上述矢量化后的信息；

 为了适用于把上述正交矢量行列的矢量变换成斜线化矢量时的情况，根据
25 上述正交矢量行列的给定数目的矢量的各连锁，参照预先定义上述连锁给定模式的斜线化表，由此，把上述正交矢量行列变换成斜线化矢量行列；

 根据上述始点和上述斜线矢量行列制作位图坐标，从分别对应于上述斜线矢量行列的给定数目矢量的连锁模式开始，参照预先定义点单位补正量的补正数据表，对该位图坐标进行补正，由此，进行引线处理；

30 经过求出上述位图数据的始点坐标的步骤，在找不到新的始点之前，反复

实施上述各步骤，生成表示上述引线处理结果的线数据；以及
基于上述线数据生成新的位图数据。

3、根据权利要求2所述的方法，其特征是，进行上述引线处理的步骤包括：根据规定上述文字图像的轮廓的数据和涂满处理，生成用于规定把内侧用
5 点涂满的线的的数据，

生成上述新的位图数据的步骤包括以下步骤：

进行上述涂满处理；

使上述涂满处理所得到数据与规定上述轮廓的数据重合；以及

根据上述矢量化进行使变粗的上述文字图像的线幅变细的线幅补正处理。

10 4、根据权利要求2所述的方法，其特征是，求出上述始点坐标的步骤包括：沿着给定的方向顺次扫描上述位图数据的点，不能根据上述矢量化后的信息表示矢量化后的情况，并且，把表示之前点的变化的点坐标确定为上述始点坐标。

5、根据权利要求2所述的方法，其特征是，把上述位图数据变换成正交矢
15 量行列的步骤包括以下步骤：

设定与上述始点对应的矢量值，更新上述矢量化后的信息，使上述始点为矢量化后的始点；

把矢量化的点位置移动到矢量方向的下一点处；

查找上述下一点周围的点，确定该下一点的矢量值；

20 在该下一点得到始点候补的场合，更新上述矢量化后的信息，使该点成为矢量化后的点；

在上述矢量化的点位置返回到上述始点之前，反复进行从移动到上述下一点的步骤到更新上述矢量化后的信息的步骤。

6、根据权利要求2所述的方法，其特征是，进行上述引线处理的步骤包括
25 以下步骤：

基于上述始点，根据预先规定的条件确定补正前的基准轮廓始点坐标，之后，把前面的坐标值加上对应于从该坐标斜线化矢量行列的矢量值的移动量所得到的值作为下次的坐标位置，由此设定基准轮廓坐标；

参照上述补正数据表，以点单位补正上述基准轮廓坐标，生成补正的轮廓
30 坐标。

7、根据权利要求6所述的方法，其特征是，进行上述引线处理的步骤还包括以下步骤：

在上述轮廓坐标的各连续的两点之间，根据上述轮廓坐标的坐标系坐标轴的两根正交轴中直线的倾斜度，以该直线表示更大的坐标值的变化的一方为基准轴，在始点、终点及上述各连续的各两点间的直线上设定最接近整数值的坐标上的插入点并进行连接，生成规定上述文字图像的轮廓的数据；

在上述轮廓坐标的上述各连续的两点之间，只以上述两根轴的给定的一方为基准轴，在始点及上述插入点中设定点并进行连接，生成用于规定用点涂满内侧的线的数据。

8、根据权利要求3所述的方法，其特征是，使根据上述涂满处理得到的数据与规定上述轮廓的数据重合的步骤包括：

对规定上述轮廓的数据及通过上述涂满处理得到的数据进行逻辑或的运算。

9、根据权利要求3所述的方法，其特征是，上述进行线幅补正处理的步骤包括：沿纵向扫描通过使上述涂满处理所得到的数据与规定上述轮廓的数据重合所得到的重合位图数据的坐标，在当前坐标点为“黑”、下一坐标点为“白”的场合，看作“白”，并且沿横向扫描上述重合的位图数据的上述坐标，在当前坐标点为“黑”、下一坐标点为“白”的场合，看作“白”，并且从表示纵横各方向上1个点幅部分变粗的上述文字图像的上述重合位图数据开始，通过矢量化分别删除该1个点幅部分。

10、一种轮廓平滑化处理装置，包括：

用于存储规定位图字形的、点阵状的位图数据的存储装置；

字形加工装置，在上述位图字形的文字放大时，把上述位图数据一次变换成矢量数据，在上述位图数据开始到结束之间，给定数目的矢量是连锁的，根据每一矢量位移的连锁，在新的位图上进行引线，制作上述放大文字图像的轮廓，接着，用点涂满上述引线生成的轮廓的内部，生成表示上述放大文字图像的新的位图数据；以及

打印或显示上述制成的位图数据的文字输出装置；

上述字形加工装置包括：

根据所变换的矢量数据求出作为基准的轮廓坐标的基准轮廓运算部；

分别与上述给定数目的矢量连锁中变化的模式对应，参照预先定义的补正数据对上述作为基准的轮廓坐标进行补正，以此作为补正轮廓坐标，然后，将该补正轮廓坐标的各邻接坐标彼此之间用线连接的线段生成部。

11、一种轮廓平滑化处理装置，适用于根据位图数据打印或显示的文字输出装置，包括：

用于存储规定位图字形的、点阵状的位图数据的存储装置；

根据预先规定的条件，参照表示各点矢量化与否的矢量化后的信息，对于上述的位图数据求出对应于变换之前矢量数据的开始的上述位图数据的始点坐标的始点确定部；

10 正交矢量提取部，从上述始点开始，按照给定的顺序确定位于对应的前面点的下一位置的邻接点的方向，根据该确定的方向，把表示上述位图数据的各点变换成上下左右4个方向中的任一方向的矢量，由此，把上述位图数据变换成正交矢量行列，同时更新上述矢量化后的信息；

15 斜线矢量变换部，为了适用于把上述正交矢量行列的矢量变换成斜线化矢量时的情况，根据上述正交矢量行列的给定数目的矢量的各连锁，参照预先定义上述连锁给定模式的斜线化表，由此，把上述正交矢量行列变换成斜线化矢量行列；

引线处理部，根据上述始点和上述斜线矢量行列制作位图坐标，从分别对应于上述斜线矢量行列的给定数目矢量的连锁模式开始，参照预先定义点单位补正量的补正数据表，对该位图坐标进行补正，由此，进行引线处理；以及

20 位图数据生成部，经过求出上述位图数据的始点坐标的步骤，在找不到新的始点之前，反复实施上述各部的处理，由此，生成表示上述引线处理结果的线数据，基于上述线数据生成新的位图数据。

12、根据权利要求11所述的轮廓平滑化处理装置，其特征是，上述引线处理部包括：生成用于规定上述文字图像的轮廓的数据的轮廓数据生成部；生成用于规定通过涂满处理把内侧用点涂满的线的数据的涂满线数据生成部，上述位图数据生成部包括：进行上述涂满处理的涂满部，使上述涂满处理所得数据与规定上述轮廓的数据重合的重合部；以及根据上述矢量化进行使变粗的上述文字图像的线幅变细的线幅补正处理的线幅补正部。

30 13、根据权利要求11所述的轮廓平滑化处理装置，其特征是，上述补正表

是分别预备多个位图字形，根据对该多个位图字形的选择，有选择地使用的补正数据的多个表格的集合。

轮廓平滑化处理方法及其装置

5 本发明涉及适于带式打印装置、字处理器或计算机中的打印装置或显示装置等的具有位图字形的文字输出装置的轮廓平滑化处理方法及其装置。

以往，打印或显示各种尺寸的文字的方法有：

- (1) 全部保持作为对象的文字数据的方法；
- (2) 保持并扩大、缩小位图字形的方法；
- 10 (3) 保持并扩大、缩小轮廓字形的方法。

可是，在(1)的情况下，由于数据量增大而无法实现。在(2)的情况下，由于比(1)时的打印质量低，在克服该缺点时，软件的负担增大，其处理需要时间。

具体来讲，在保持位图字形的打印装置中，作为提高打印质量的处理，有
15 文字放大时的“台阶差”的平滑化处理。这里所谓的“台阶差”是指，在位图数据的倾斜方向上邻接的点列(数组)为不连续的、随着文字的放大所产生的错位部分。

以往，为了实现平滑化，需要检测该“台阶差”，在放大时要追加补正点，弥补该“台阶差”。

20 但是，在这种平滑化处理中，由于补正的点数多，文字轮廓变形，有损于外观，另外，如果减少补正点数，就会残留有凹凸，具有放大率越大，越不能得到补正效果的缺点。

另一方面，在(3)的情况下，与(2)的情况相比，系统规模更大，并且，数据量也增大，例如在带式打印装置的比较小规模打印装置中，无论是
25 处理时间还是价格方面都会得不偿失。

本发明的目的是，提供一种即使放大率变大，也能生成使轮廓平滑的位图字形文字图像的轮廓平滑化处理方法、装置及存储介质。

为了达到上述目的，根据本发明的第一种形式，提供一种把点阵状的位图数据所规定的位图字形的文字图像的轮廓平滑化的方法，包括下述步骤：

30 在上述文字图像放大时，把上述位图数据一次变换成矢量数据；

在上述位图数据开始到结束之间，给定数目的矢量是连锁的，根据每一矢量位移的连锁，在新的位图上进行引线，制作上述放大文字图像的轮廓；以及用点涂满上述引线生成的轮廓的内部，生成表示上述放大文字图像的新的位图数据。

5 最好是，进行上述引线的步骤包括以下步骤：

根据上述矢量数据求出作为基准的轮廓坐标；

分别与上述给定数目的矢量连锁中变化的模式对应，参照预先定义的补正数据对上述作为基准的轮廓坐标进行补正，以此作为补正轮廓坐标；以及

将该补正轮廓坐标的各邻接坐标彼此之间用线连接。

10 为了完成上述目的，根据本发明的第二形式，提供一种把点阵状的位图数据所规定的位图字形的文字图像的轮廓平滑化的方法，包括下述步骤：

根据预先规定的条件，参照表示各点矢量化与否的矢量化后的信息，对于上述的位图数据求出对应于变换之前矢量数据的开始的上述位图数据的始点坐标；

15 从上述始点开始，按照给定的顺序确定位于对应的前面点的下一位置的邻接点的方向，根据该确定的方向，把表示上述位图数据的各点变换成上下左右4个方向中的任一方向的矢量，由此，把上述位图数据变换成正交矢量行列，同时更新上述矢量化后的信息；

20 为了适用于把上述正交矢量行列的矢量变换成斜线化矢量时的情况，根据上述正交矢量行列的给定数目的矢量的各连锁，参照预先定义上述连锁给定模式的斜线化表，由此，把上述正交矢量行列变换成斜线化矢量行列；

根据上述始点和上述斜线矢量行列制作位图坐标，从分别对应于上述斜线矢量行列的给定数目矢量的连锁模式开始，参照预先定义点单位补正量的补正数据表，对该位图坐标进行补正，由此，进行引线处理；

25 经过求出上述位图数据的始点坐标的步骤，在找不到新的始点之前，反复实施上述各步骤，生成表示上述引线处理结果的线数据；以及

基于上述线数据生成新的位图数据。

最好是，进行上述引线处理的步骤包括：根据规定上述文字图像的轮廓的数据和涂满处理，生成用于规定把内侧用点涂满的线的数据，

30 生成上述新的位图数据的步骤包括以下步骤：

进行上述涂满处理;

使上述涂满处理所得到数据与规定上述轮廓的数据重合; 以及

根据上述矢量化进行使变粗的上述文字图像的线幅变细的线幅补正处理。

- 最好是, 求出上述始点坐标的步骤包括: 沿着给定的方向顺次扫描上述位图数据的点, 不能根据上述矢量化后的信息表示矢量化后的情况, 并且, 把表示之前点的变化的点坐标确定为上述始点坐标。

最好是, 把上述位图数据变换成正交矢量行列的步骤包括以下步骤:

设定与上述始点对应的矢量值, 更新上述矢量化后的信息, 使上述始点为矢量化后的始点;

- 10 把矢量化的点位置移动到矢量方向的下一点处;

查找上述下一点周围的点, 确定该下一点的矢量值;

在该下一点得到始点候补的场合, 更新上述矢量化后的信息, 使该点成为矢量化后的点;

- 15 在上述矢量化的点位置返回到上述始点之前, 反复进行从移动到上述下一点的步骤到更新上述矢量化后的信息的步骤。

最好是, 进行上述引线处理的步骤包括以下步骤:

基于上述始点, 根据预先规定的条件确定补正前的基准轮廓始点坐标, 之后, 把前面的坐标值加上对应于从该坐标斜线化矢量行列的矢量值的移动量所得到的值作为下次的坐标位置, 由此设定基准轮廓坐标;

- 20 参照上述补正数据表, 以点单位补正上述基准轮廓坐标, 生成轮廓坐标。

最好是, 进行上述引线处理的步骤还包括以下步骤:

- 25 在上述轮廓坐标的各连续的两点之间, 根据上述轮廓坐标的坐标系坐标轴的两根正交轴中直线的倾斜度, 以该直线表示更大的坐标值的变化的一方为基准轴, 在始点、终点及上述各连续的各两点直线上设定最接近整数值的坐标上的插入点并进行连接, 生成规定上述文字图像的轮廓的数据;

在上述轮廓坐标的上述各连续的两点之间, 只以上述两根轴的给定的一方为基准轴, 在始点及上述插入点中设定点并进行连接, 生成用于规定用点涂满内侧的线的数据。

- 30 最好是, 使根据上述涂满处理得到的数据与规定上述轮廓的数据重合的步骤包括:

对规定上述轮廓的数据及通过上述涂满处理得到的数据进行逻辑或的运算。

更好的是，上述进行线幅补正处理的步骤包括：沿纵向扫描通过使上述涂满处理所得到的数据与规定上述轮廓的数据重合所得到的重合位图数据的坐标，在当前坐标点为“黑”、下一坐标点为“白”的场合，看作“白”，并且沿横向扫描上述重合的位图数据的上述坐标，在当前坐标点为“黑”、下一坐标点为“白”的场合，看作“白”，并且从表示纵横各方向上1个点幅部分变粗的上述文字图像的上述重合位图数据开始，通过矢量化分别删除该1个点幅部分。

10 为了完成上述目的，根据本发明的第三形式，提供一种轮廓平滑化处理装置，包括：

用于存储规定位图字形的、点阵状的位图数据的存储装置；

字形加工装置，在上述位图字形的文字放大时，把上述位图数据一次变换成矢量数据，在上述位图数据开始到结束之间，给定数目的矢量是连锁的，根据每一矢量位移的连锁，在新的位图上进行引线，制作上述放大文字图像的轮廓，接着，用点涂满上述引线生成的轮廓的内部，生成表示上述放大文字图像的新的位图数据；以及

打印或显示上述制成的位图数据的文字输出装置。

最好是，上述字形加工装置包括：

20 根据所变换的矢量数据求出作为基准的轮廓坐标的基准轮廓运算部；

分别与上述给定数目的矢量连锁中变化的模式对应，参照预先定义的补正数据对上述作为基准的轮廓坐标进行补正，以此作为补正轮廓坐标，然后，将该补正轮廓坐标的各邻接坐标彼此之间用线连接的线段生成部。

25 为了完成上述目的，根据本发明的第四形式，提供一种轮廓平滑化处理装置，适用于根据位图数据打印或显示的文字输出装置，包括：

用于存储规定位图字形的、点阵状的位图数据的存储装置；

根据预先规定的条件，参照表示各点矢量化与否的矢量化后的信息，对于上述的位图数据求出对应于变换之前矢量数据的开始的上述位图数据的始点坐标的始点确定部；

30 正交矢量提取部，从上述始点开始，按照给定的顺序确定位于对应的前面

点的下一位置的邻接点的方向，根据该确定的方向，把表示上述位图数据的各点变换成上下左右4个方向中的任一方向的矢量，由此，把上述位图数据变换成正交矢量行列，同时更新上述矢量化后的信息；

斜线矢量变换部，为了适用于把上述正交矢量行列的矢量变换成斜线化矢量时的情况，根据上述正交矢量行列的给定数目的矢量的各连锁，参照预先定义上述连锁给定模式的斜线化表，由此，把上述正交矢量行列变换成斜线化矢量行列；

引线处理部，根据上述始点和上述斜线矢量行列制作位图坐标，从分别对应于上述斜线矢量行列的给定数目矢量的连锁模式开始，参照预先定义点单位补正量的补正数据表，对该位图坐标进行补正，由此，进行引线处理；以及

位图数据生成部，经过求出上述位图数据的始点坐标的步骤，在找不到新的始点之前，反复实施上述各部的处理，由此，生成表示上述引线处理结果的线数据，基于上述线数据生成新的位图数据。

最好是，上述引线处理部包括：生成用于规定上述文字图像的轮廓的数据的轮廓数据生成部；生成用于规定通过涂满处理把内侧用点涂满的线的数据的涂满线数据生成部，上述位图数据生成部包括：进行上述涂满处理的涂满部，使上述涂满处理所得数据与规定上述轮廓的数据重合的重合部；以及根据上述矢量化进行使变粗的上述文字图像的线幅变细的线幅补正处理的线幅补正部。

最好是，上述补正表是分别预备数个位图字形，根据对该数个位图字形的选择，有选择地使用的补正数据的数个表格的集合。

为了完成上述目的，根据本发明的第五形式，提供一种存储介质，用于实现使点阵状位图数据所规定的位图字形文字图像的轮廓平滑化的方法，并储存计算机可以实施的程序，所述的程序包括：

在上述文字图像放大时，将上述位图数据一次变换成矢量数据的矢量数据生成模块；

在上述位图数据开始到结束之间，给定数目的矢量是连锁的，根据每一矢量位移的连锁，在新的位图上进行引线，制作上述放大文字图像的轮廓的轮廓生成模块；以及

用点涂满上述引线生成的轮廓的内部，生成表示上述放大文字图像的新位

图数据的位图生成模块。

为了完成上述目的，根据本发明的第六形式，提供一种存储介质，用于实现将点阵状位图数据所规定的位图字形文字图像的轮廓平滑化的方法，并储存计算机可以实施的程序，所述的程序包括：

- 5 根据预先规定的条件；参照表示各点矢量化与否的矢量化后的信息，对于上述的位图数据求出对应于变换之前矢量数据的开始的上述位图数据的始点坐标的始点确定模块；

- 从上述始点开始，按照给定的顺序确定位于对应的前面点的下一位置的邻接点的方向，根据该确定的方向，把表示上述位图数据的各点变换成上下左右
10 4个方向中的任一方向的矢量，由此，把上述位图数据变换成正交矢量行列，同时更新上述矢量化后的信息的正交矢量提取模块；

- 为了适用于把上述正交矢量行列的矢量变换成斜线化矢量时的情况，根据上述正交矢量行列的给定数目的矢量的各连锁，参照预先定义上述连锁给定模式的斜线化表，由此，把上述正交矢量行列变换成斜线矢量行列的斜线矢量变
15 换模块；

根据上述始点和上述斜线矢量行列制作位图坐标，从分别对应于上述斜线矢量行列的给定数目矢量的连锁模式开始，参照预先定义点单位补正量的补正数据表，对该位图坐标进行补正，由此，进行引线处理的引线处理模块；

- 经过求出上述位图数据的始点坐标的步骤，在找不到新的始点之前，反复
20 实施上述各步骤，生成表示上述引线处理结果的线数据的线数据生成模块；以及

基于上述线数据生成新的位图数据的位图数据生成模块。

- 根据本发明的轮廓平滑化处理方法、装置及其存储，把字形存储器准备的位图数据一次置换成矢量数据，用矢量数据实现平滑化，由此，即使放大率变
25 大，也能生成可产生使轮廓平滑的文字图像的位图数据。因此，可用8个方向的矢量（上下左右4个方向及倾斜的4个方向）表示位图，从6个矢量的连锁模式中确定出最合适的矢量方向，另外，从6个矢量的连锁模式中，以用矢量不能表现的1个点单位进行补正。

- 根据本发明，即使放大，也能以良好的质量进行打印或显示，除此之外，
30 为了使用位图数据，可进行压缩，根据算术符号化的压缩算法，可依赖于文字

尺寸等，但也可以压缩例如40%的程度。通常，在装载有轮廓字形的机器中，在分辨率低、处理小尺寸的文字的场合，由于打印质量劣化，要用位图数据进行打印，但是，根据本发明，与文字大小无关，能以相同的质量打印文字，另外，可节约存储器的容量。

- 5 另外，根据本发明，无论是用于字处理器或计算机中的打印装置还是显示装置，即使用于带式打印装置等装置规模比较小的机种中，也可在不牺牲打印质量、处理速度、价格任一方面的性能的前提下，在文字放大时，依然可进行平滑的轮廓平滑化处理。

10 本发明的上述及其他目的、特征和优点，通过下文基于附图的详细说明将会更加清楚明了。

图1是表示采用了基于本发明实施例的轮廓平滑化处理方法及其装置的文字输出装置的内部构成的方框图。

图2是表示把图1中控制部的功能展开到存储器上的方框图。

图3是用于说明关于本发明实施例的坐标和位图定义的说明图。

- 15 图3B1是用于说明关于本发明实施例的矢量定义的说明图。

图3B2是把上述矢量的定义用表格形式表示的说明图。

图4是表示上述控制部的位图处理顺序的程序方框图。

图5是表示图4中矢量始点检索处理的具体处理顺序的程序方框图。

图6是表示图4中正交矢量的变换处理的具体处理顺序的程序方框图。

- 20 图7是表示图4中斜线矢量的变换处理的具体处理顺序的程序方框图。

图8是表示图4中引线处理的具体处理顺序的程序方框图。

图9是表示图4中涂满处理的具体处理顺序的程序方框图。

图10是表示图4中线幅补正处理的具体处理顺序的程序方框图。

图11A~11C是表示把确定始点检索的例子在位图上展开的视图。

- 25 图12是用表格（<表1>）形式表示确定矢量始点的条件和矢量值的关系的视图。

图13是用表格（<表2>）形式表示一个前矢量和检索顺序的关系的视图。

- 30 图14是用表格（<表3>）形式表示检索方向模式、条件和矢量的关系的视图。

图15是用表格（<表4>）形式表示矢量和矢量化位置的关系的视图。

图16A和图16B是在位图上将正交矢量化了的例子展开的视图。

图17是用表格（<表5>）形式表示在正交矢量行列中的连续两个矢量要素的变化方向的视图。

5 图18是用表格（<表6>）形式表示正交矢量和变换的斜线矢量的关系的视图。

图19是用表格（<表7>）形式表示矢量变换条件的视图。

图20是用表格（<表8>）形式表示把变换条件作为矢量模式的视图。

图21是表示斜线化判定表内容的视图。

10 图22A和图22B是在位图上将矢量斜线化的例子展开的视图。

图23是用表格（<表9>）形式表示矢量值和坐标移动量的关系的视图。

图24是用表格（<表10>）形式表示矢量值、从上次坐标的移动量和轮廓基准坐标的关系的视图。

图25是用表格（<表11>）形式表示引线坐标的视图。

15 图26是用表格（<表12>）形式表示补正数据和补正方向的关系的视图。

图27是用表格（<表13>）形式表示补正方向和坐标移动量的关系的视图。

图28是用表格（<表14>）形式表示放大倍率和补正点数的关系的视图。

图29是用表格（<表15>）形式表示点补正例子的视图。

20 图30是用表格（<表16>）形式表示点补正例子的视图。

图31是用于说明点插入的例示曲线（<曲线1>）图。

图32是用明细表（<明细表1>）形式表示插入点组运算算法的视图。

图33是用明细表（<明细表2>）形式表示插入点组运算算法的视图。

图34是用明细表（<明细表3>）形式表示插入点组运算算法的视图。

25 图35是用明细表（<明细表4>）形式表示插入点组运算算法的视图。

图36是为说明点插入的例示曲线（<曲线2>）图。

图37是用明细表（<明细表5>）形式表示插入点组运算算法的视图。

图38是为说明点插入的例示曲线（<曲线3>）图。

图39是为说明点插入的例示曲线（<曲线4>）图。

30 图40是用明细表（<明细表6>）形式表示插入点组运算算法的视图。

图41是为说明点插入的例示曲线（＜曲线5＞）图。

图42是用明细表（＜明细表7＞）形式表示笔画线涂满算法的视图。

图43A和43B是为说明采用笔画线涂满的例示曲线（＜曲线6＞）图。

图44A和44B是为说明采用笔画线涂满的例示曲线（＜曲线7＞）图。

5 图45A和45B是为说明采用笔画线涂满的例示曲线（＜曲线8＞）图。

图46A和46B是为说明采用X扫描引线处理涂满的例示曲线（＜曲线9＞）图。

图47是用明细表（＜明细表8＞）形式表示填充线插入点组发生算法的视图。

10 图48A和图48B是用于说明填充线处理的涂满的例示曲线（＜曲线10＞）图。

图49是用明细表（＜明细表9＞）形式表示填充线涂满的算法的视图。

图50是用曲线表示笔画区域和填充区域重合的动作概念图。

图51A～51D是表示在位图上将粗度补正的例子展开的视图。

15 图52A～52D是表示在位图上将粗度补正的例外处理展开的视图。

图53A～53C是表示本发明实施例所使用的矢量定义的更详细例子的视图。

实施例的说明

以下，参照附图，详细说明本发明的一实施例。

20 图1是表示采用本发明的文字输出装置的内部构成的方框图。在本发明的实施例中，用例示的作为文字输出装置的带式打印装置进行说明。带式打印装置，用热方式将基于键入的所期望的文字等作成的打印图像打印在带盒内的带上，同时，将带的打印部分切断并制成标签。

如图1所示，带式打印装置的基本构成包括：把CPU作为控制中枢的控制部1，具有键盘和显示器作为用户接口的操作部2，通过热式打印头在带盒中的带上进行打印的打印部3。操作部2和打印部3通过内置有各种驱动器并驱动各部回路的驱动部4连接在控制部1上。鉴于操作部2、打印部3及驱动部4与本发明的主题没有直接关系，因此，省略其详细说明。

30 控制部1由CPU11、程序存储器12、工作RAM13、字形存储器14及输入输出控制部15构成。各方块（方框）11、12、13、14及15用双向内部总线16共同

连接。CPU11具有作为本发明的轮廓平滑化处理装置的功能，除此之外，根据
储存到程序存储器12内的后述各程序，使用工作RAM13，对带式打印装置整
体进行控制。

工作RAM13用于作业区域，每一次都写入程序的轮廓平滑化处理经过的
5 数据。字形存储器14，以点阵状的位图形式存储有文字等的字形数据，在对文
字等赋予特定的代码数据时，将所对应的字形数据输出。

输入输出控制部（IOC）15由门阵列或定制LSI等构成，并组装有补偿
CPU11功能并处理各种周边电路的接口信号的理论电路。例如，把从键盘输入
的各种指令及输入数据等按原样或进行加工后输入给内部总线16，同时，与
10 CPU11协调工作，通过CPU11，把从内部总线16输出的数据或控制信号按原样
或经过加工后输出给驱动部4。

图2是表示把图1所示控制部1具有的功能展开到存储器上的方框图。如上
述，控制部1具有作为轮廓平滑化处理装置的功能，在该控制部1中，在文字加
工时，一旦将该位图数据变换成矢量数据，在位图数据的倾斜方向上，邻接的
15 数据列含有不连续的部分，从矢量的始点向该矢量方向的终点引线并进行点补
正，而且把内部涂满，再次生成位图，对打印或显示输出进行控制。

如图2所示，本发明的轮廓平滑化处理装置（控制部1）由字形存储器14、
字形加工装置（CPU11）和文字输出部20构成。字形加工装置11在功能上由始
点检索·确定部111、正交矢量提取部112、斜线矢量化变换部113、引线处理
20 部114、涂满处理部115及线幅补正处理部116构成。各处理部111~116根据后述
逻辑并依程序而动作。

始点检索·确定部111相对字形存储器14中的位图数据，根据预先规定的
条件，求出矢量始点的坐标，供给正交矢量提取部112。正交矢量提取部112从
始点检索·确定部111输出的始点坐标按顺序调整点的变化方向，在上下左右
25 四个方向的正交矢量行列中变换的同时，更新表示该点矢量化后的矢量化后信
息（标记存储器131）的内容。

斜线矢量变换部113，从变换的正交矢量行列的6个矢量的连锁模式中，参
照要预先斜线化的模式所定义的斜线化判定表132的内容，变换成斜线矢量行
列，供给引线处理部114。

30 引线处理部114相对放大等加工的位图坐标，从斜线矢量行列的6个矢量的

连锁模式中，参照预先定义的补正数据进行引线并以点单位补正坐标，将补正的轮廓坐标之间用线连接。引线处理部114包括：从变换的矢量行列求出作为基准的轮廓坐标的基准轮廓运算部、通过连续矢量值的模式设定预先定义的补正数据的补正表、参照上述补正表补正作为上述基准的轮廓坐标并把该补正的轮廓坐标之间用线连接的线段生成部。这里，作为轮廓坐标，可生成外框用线和涂满用线2个种类。

涂满处理部115，把先进行轮廓坐标补正所得到的填充线的内侧涂满之后，把展示在正在描绘字形外框的工作存储器13的轮廓用描绘区域和填充线用描绘区域（任何位图区域134）的内容进行逻辑或运算，并进行重合处理。

10 线幅补正处理部116，跟踪由引线处理部114生成的位图数据，在定义的XY两方向分别删除1个点部分，由此，通过矢量化对XY方向的作为1个点部分的粗度的位图线幅进行补正。

文字输出部20，从打印用的工作用位图区域134读取所希望的位图，展开点并通过输入输出控制部15、驱动部4输出给打印部3。

15 下面，用程序方框图详细说明上述各处理部111~116的详细逻辑。

首先，说明本发明的动作。下文，参照图3A及3B说明以下说明的坐标和位图数据的定义。图3A示出了坐标和位图的定义，图3B1及3B2示出了矢量的定义。

如图3A所示，定义坐标X和Y，在以下表示坐标值的场合，用X，Y来标记。为了方便，将X轴的正方向规定为向右，负方向规定为向左，将Y轴的正方向规定为向下，负方向规定为向上。位图数据的白表示为 0×0 （“0”为C语言中的16进制），黑表示为 0×1 （“1”为C语言中的16进制）。赋予位图数据的坐标外（[0, -1]等）为白（ 0×0 ）。

25 矢量的方向和序号用图3B1定义。即，在上下左右的0~7的8个方向上表示矢量，从而，可表示为图3B2所示的形式。另外，由于矢量的1和5、3和7不能用1个文字区别，因此在旁边添上“'”和“.”以示区别。

图53A~53C示出了矢量的定义。如图53A所示，矢量用有方向的线段OP表示，一般来说，矢量用黑体字表示，在这里表示成 $\uparrow a$ 。另外，如图53B所示，X-Y坐标上的矢量 $\uparrow v$ 使用基本的矢量 $\uparrow i$ 、 $\uparrow j$ ，其中 $\uparrow v = x \cdot \uparrow i + y \cdot \uparrow j$ 。这里，只表示为 $\uparrow v$ 。如图53C所示，定义各方向的矢量，并分别置换到各

个行列上。把该各矢量的集合称为矢量空间、矢量组，表示为 $\uparrow \text{vct}$ 。

各方向的矢量以X轴为基准的旋转角度如下。

$$\begin{aligned} \angle(\uparrow a) &= 0, \angle(\uparrow b) = \pi/4, \angle(\uparrow c) = \pi/2, \angle(\uparrow d) = 3\pi/4, \\ \angle(\uparrow e) &= \pi, \angle(\uparrow f) = 5\pi/4, \angle(\uparrow g) = 3\pi/2, \angle(\uparrow h) = 7\pi/4 \end{aligned}$$

5 各矢量间的旋转角度，根据上述各值的减法运算求出

$$\angle(\uparrow f - \uparrow d) = \angle(\uparrow f) - \angle(\uparrow d) = \pi/2。$$

这里，为了使旋转角度的计算等简单易行，把各矢量看作“记号”，分配0~7的数字。即，按以下方式分配。

10 $\uparrow a \rightarrow "0" \quad \uparrow b \rightarrow "1" \quad \uparrow c \rightarrow "2" \quad \uparrow d \rightarrow "3" \quad \uparrow e \rightarrow "4" \quad \uparrow f \rightarrow "5" \quad \uparrow g \rightarrow "6" \quad \uparrow h \rightarrow "7"。$

在这种场合，上述的 $\uparrow \text{vct}$ 是这样的。

$$\begin{aligned} \uparrow \text{vct} &= \{\uparrow a, \uparrow b, \uparrow c, \uparrow d, \uparrow e, \uparrow f, \uparrow g, \uparrow h\} \\ &= \{0, 1, 2, 3, 4, 5, 6, 7\} \end{aligned}$$

矢量组 $\uparrow \text{vct}$ 的第n个矢量，表示成 $\uparrow \text{vct}[n]$ ，例如， $\uparrow \text{vct}[2] = \uparrow c = "2"。$

15 图4是表示本发明实施例的処理和数据の流程的程序方框图。以下，概括说明动作的流程。

首先，始点检索・确定部111对于从文字字形存储器14读出的位图数据，求出作为矢量始点候补的坐标。另外，从该位图和后述的矢量化后的信息求出新的矢量始点坐标，再次把该始点坐标交给正交矢量提取部112（步骤S41，
20 S42）。

正交矢量提取部112，从始点扫描位图，通过置换成上下左右4个方向的正交矢量行列，从始点提取1周部分的正交矢量，把提取的正交矢量行列交给斜线矢量变换部113（步骤S43）。斜线矢量变换部113参照后述的斜线化判定表132，从得到的正交矢量行列的6个矢量的连锁模式中变换成斜线矢量行列，在
25 引线处理部114中进行处理（步骤S44）。这时，在上下左右4个方向添加倾斜的4个方向，构成8个方向的矢量。

引线处理部114一边补正坐标一边制作轮廓，在对于放大加工的位图坐标寻找矢量并引线之际，参照后述的补正表133，从得到的斜线矢量行列的6个矢量的连锁模式中，以一个点单位补正引线的坐标（步骤S45）。在这里，把线
30 作成外框用线和涂满用线两种类型，该线数据在对于全部始点处理结束之前，

保持在工作位图区域134中。由此，把补正加到轮廓坐标上，从斜线矢量进行引线。然后，再次进行矢量始点的检索（步骤S41），反复进行上述的步骤S41～S45的处理，直到没有新的始点为止，交给涂满处理部115进行处理。

涂满处理部115对引线处理部114生成的涂满用线的内部涂满之后，使之与
5 外框用线重合（步骤S46）。线幅补正处理部116，为了使整体线幅变粗，通过矢量化进行使纵横1点线变细的处理（步骤S47）。

以下，参照图5～图10所示的详细程序方框图，详细说明从矢量始点检索（步骤S41）到线幅补正处理（步骤S47）的处理。

首先，说明始点检索・确定部111的始点检索・确定所需要的处理（图4中
10 的步骤S41、S42）。

图5以程序方框图示出了进行始点检索・确定的处理顺序。在这里，首先从坐标[0, 0]开始始点检索（步骤S411）。然后，沿X方向扫描坐标，检查是否符合每1点的始点条件（步骤S412）。

作为始点的条件，当前坐标位置点（ $[x, y]$ ）为黑，1点前（ $[x-1, y]$ ）为
15 白，不对标记存储器113设定的矢量化后的信息对应点进行设置，以所谓的还没有矢量化的点为始点。标记存储器113设定的矢量化后的信息在后述的正交矢量提取处理（步骤S43）中更新。全部清除初始值。

把符合上述始点条件的点作为始点，进入后述的正交矢量提取处理。在这里，如果看不到始点，把 y 更新成+1（ y 到达坐标原点时 x 为+1， $y=0$ ）（步骤
20 S413），反复进行步骤S411以后的处理。

参照图11A～11C所示的位图说明始点确定的例子。图11A、图11B及图11C分别表示第一次、第二次、第三次始点检索时的位图和矢量化后的信息。

在第一次检索中，为了全部清除标记存储器113中的矢量化后的信息，最初为“白→黑”，此时[6, 2]作为始点。在第二次检索的场合，通过相对[6, 2]
25 的正交矢量化（参照图16A），更新“矢量化后的信息”（参照图16B）。变为“白→黑”的点[6, 3] [6, 4] [6, 5] [6, 6]等不是用于调整矢量化后信息的始点。从而，符合条件的最初点[15, 5]成为始点。在第三次检索的场合，通过第一、第二次的正交矢量化，变成“白→黑”的点全部确定成“矢量化后的信息”，用标记存储器113设置。因此，在判断为没有第三次的始点时，进入后
30 述的涂满处理（步骤S46）。

下面，参照图6详细说明正交矢量提取部112的正交矢量变换处理（图4中的步骤S42）。图6中，用程序方框图详细地示出了为进行正交矢量变换处理的顺序。

所谓正交矢量的变换处理，是指把位图变换成正交矢量值（→、↓、←、
5 ↑的四种）的处理，由始点顺次调整点的变化方向，使矢量化后的数据变成为
↑VCT=[0, 0, 2, 4, ……]的行列，没有位置信息。

首先，求出始点的矢量值，设置矢量化信息（步骤S431）。其次，把矢量化
的点位置从始点向矢量方向移动（步骤S432），并且调整移动的点位置的外
围点，确定矢量值（步骤S433）。然后，使矢量化的点位置向矢量方向移动
10 （步骤S434），并在矢量化的点位置得到始点候补的场合（步骤S435），调整
矢量化后的信息（步骤S436），如果不这样，就反复进行点位置的移动以后的
步骤（步骤S433～S435）。

另外，始点矢量值的确定，根据图12所示的<表1>来确定，使点位置从
始点沿着矢量方向移动。这时，相对当前位置的点确定矢量值，从1以前的矢
15 量值确定矢量检索方向模式顺序，根据该顺序调整各矢量检索方向的模式条
件。最初以对应于一致的条件值作为矢量值。图13示出了作为<表2>的1以
前的矢量和检索顺序的关系，图14示出了作为<表3>的检索方向模式、条件
以及矢量值的关系。

检索条件不是排他的，矢量0和4，或者2和6所确定的条件是相容的。从
20 而，在不根据检索顺序进行处理时，不能得到正确的矢量。这里的检索顺序按
照矢量“向左旋转方优先”的方式确定。

但是，在当前位置矢量值为向上方向的“6”的场合，向上方移动的点作
为始点候补。因此，在始点的检索处理中，由于把该点除外是必要的，所以，
用标记存储器13设置“矢量化后的信息”。沿矢量方向的点移动是根据图15所
25 示的<表4>进行的。即是说，根据<表4>进行矢量化，使点坐标位置移动。
此后，反复进行上述的步骤S432～S436的处理（步骤S437），直到矢量化的点
位置可返回到始点为止。

下面，参照图16A和16B说明正交矢量化化的例子。对应于“矢量化后的信
息”，示出了上述始点检索的例子，图16A示出了第一次始点检索时的正交矢
30 量数据、正交矢量变换结束时的“矢量化后的信息”，图16B示出了第二次始

下面，说明矢量运算方法。相对正交矢量提取部112作成的正交矢量行列查找（核查）斜线化条件，输出新的斜线化矢量行列。斜线化的矢量不受下面的斜线化条件影响。

参照图7所示的程序方框图，说明为进行矢量运算的处理流程。从正交矢量行列的先头位置开始矢量运算（步骤S441）。 $V_{n2} \sim V_2$ 的位置成为 $\uparrow VCT=\{V_0, V_1, V_2, \dots V_{n2}, V_{n1}\}$ 。接着，调整 $V_0 \rightarrow V_1$ 的旋转方向（步骤S442）。这时，如果是左旋转，则进入步骤S444以后的处理，在左旋转以外的场合，以当前正交矢量值 V_0 为斜线化矢量值，使检查的正交矢量行列的要素位置移动到下次矢量上（步骤S443）。

例如，把 $V_{n2} \sim V_2$ 的位置从 $\uparrow VCT=\{V_0, V_1, V_2, \dots V_{n2}, V_{n1}\}$ ，按照 $\uparrow VCT=\{V_{n1}, V_0, V_1, V_2, \dots V_{n2}\}$ 的方式移动1个位置。

在步骤S444中，用以下的C语言运算式（1）算出 $V_{n2} \sim V_3$ 的矢量变化方向。

$$dr(V[n-1] \rightarrow V[n]) = (V[n] - V[n-1] + 10) \text{ AND } 0 \times 07 \dots \dots (1)$$

运算的结果意味着“0”为左旋转，“2”为同方向，“4”是右旋转，“6”为反对（不存在）。

其次，把变化方向分为当前矢量的前后，根据以下算式（2）、（3）进行加权处理（步骤S445）。

• 后方向的变化加权值

$$W1 = \{dr(V1 - V2) \times 3 + dr(V2 \rightarrow V3)\} / 2 \dots \dots \dots (2)$$

• 前方向的变化加权值

$$W2 = \{dr(V_{n1} - V_0) \times 3 + dr(V_{n2} \rightarrow V_{n1})\} / 2 \dots \dots (3)$$

然后，相对该 $W1$ 、 $W2$ 查找斜线化判定表132的值（步骤S446）。这里，查找该值（步骤S447）并且在值为“1”的场合，把当前矢量斜线化，代入斜线化矢量行列。根据以下（4）式运算。

$$\text{斜线化矢量值} = (\text{当前矢量值} + 0 \times 07) \text{ AND } 0 \times 07 \dots \dots (4)$$

由此，使“0”变换成“7”、“2”变换成“1”、“4”变换成“3”、“6”变换成“5”。由于成为斜线化对象的正交矢量序列要素的下次要素含有斜线化矢量，所以，使检查的正交矢量行列要素位置移动2个（步骤S448）。

另一方面，当值为“0”时，把当前正交矢量值 V_0 看作斜线化矢量值，使

检查的正交矢量行列的要素位置移动到下一个矢量上（步骤S449）。然后，反复进行上述步骤S441～S449的处理（步骤S450），直到返回到正交矢量行列的先头位置为止。

关于使最终矢量斜线化时的始点的移动处理，在正交矢量行列的最后的矢量和先头的矢量成为斜线化对象的场合（在 $\uparrow VCT=\{V1, V2, \dots, Vn2, Vn1, V0\}$ 中的 $V0-V1$ 斜线化的场合），删除代入斜线化矢量行列的先头的矢量，把从始点坐标向删除的先头矢量方向所移动的坐标看作是新的始点。

图21示出了斜线化判定表132的内容。在斜线化判定表132中，相对上述的加权运算得到的表示W1、W2的值，设定是否进行斜线化的数据。通过变更该表的值，能变更斜线化的模式。

下面，参照图22A、22B说明矢量斜线化的例子。图22A示出了第一次始点确定时的斜线化矢量数据，图22B示出了第二次始点确定时的斜线化矢量数据。

图22A中, 始点为[6, 2]、斜线化矢量数据是 $\uparrow \text{VCT1B}=[0, 0, 0, 0, 0,$
 15 $0, 0, 0, 0, 1, 0, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3,$
 $4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,$
 $6, 6, 6, 6, 6, 6, 6]$ 。

图22B中, 初始始点为[15、5]、斜线化矢量数据 $\uparrow$ VCT2B=[4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 7, 7, 7, 6, 6, 6, 6, 6, 5, 5, 5]。

在这里，使最终矢量和先头矢量斜线化，删除的先头矢量是“4”，从而，始点移动到[-1,0]，新始点成为[14, 5]。

下面，说明引线处理部114对含有点补正的引线处理（图4中的步骤S45）。在这里，引线处理包括补正轮廓坐标所进行的处理和补正所得到的坐标间引线的处理。图8用程序方框图示出进行引线处理的处理顺序。以下，参照程序方框图进行详细说明。

首先，说明轮廓坐标补正的处理。在这里，从基本位图中提取的斜线化矢量行列开始，求出矢量的始点及终点坐标，在X方向、Y方向放大的新位图区域（工作用位图134）上描绘轮廓（线）。矢量的始点、终点坐标，参照斜线

30 化矢量区域的连续的6个要素，以1点单位添加补正。在此，将这种轮廓制成外

框用线和涂满用线两种类型。因此，为放大加工处理的位图区域有必要为2个。

由始点和矢量值确定作为基准的轮廓坐标（步骤S451）。这时，参照补正数据进行运算，求出作为前面基准的轮廓坐标。

5 具体地，在以X，Y方向的倍率分别为BX、BY，原始点为X0、Y0的场合，以 $[BX \times X0, BY \times Y0]$ 求出放大加工后的始点 $[X1, Y1]$ 。以后的基准轮廓坐标的运算是，把前面的坐标（最初的始点坐标）值加上对应于从该坐标斜线化的矢量值的移动量所得到的值作为下次的坐标位置。图23中的<表9>示出了斜线化矢量值和移动量的关系。

10 以下示出了轮廓基准坐标的例子。即，将斜线化矢量变换部113作成的斜线化矢量行列用 $BX=2, BY=3$ 放大的轮廓坐标是，使始点坐标 $[X0, Y0]$ 为 $[6, 2]$ ，斜线化矢量行列 $\uparrow VCT1B=[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 6]$ 。放大后的始点坐标 $[X1, Y1]$ 为 $[12, 6]$ ，始点坐标以后的坐标运算用图24中的<表10>表示。

20 为了从轮廓中得到位图数据，可在图25<表11>所示的坐标间顺序地引线。但是，由于只能在45度方向上作斜线，位图数据不是十分圆滑。因此，如下所述，要考虑参照前后矢量数据，以1个点坐标单位补正各坐标，使其成为更圆滑的位图数据。

即，确定成为基准的轮廓坐标之后，进行步骤S452以后的补正数据的基准坐标的补正处理。在此，相对上述求出的基准轮廓坐标，参照前后的矢量值，求出借助补正数据进行补正的引线用轮廓坐标。补正中有三种类型，把基准坐标看作照原样引线用坐标的“不补正”；该基准坐标不使用引线，包含在下次以后的引线处理中的“跳过”；相对基准坐标以1个点单位移动坐标位置的“点补正”等。并且，补正数据是通过前后矢量值的模式分别确定的值，用补正表133设定该值。

30 以下，说明补正数据的参照方法。首先说明按照以下方式定义当前矢量及其外围矢量。即是说，在斜线化矢量行列 $\uparrow VCTB=\{\dots, VCTB[n-3], VCTB[n-2], VCTB[n-1], VCTB[n], VCTB[n+1], VCTB[n+2], \dots\}$ 中，把

连续的6个矢量要素的VCTB[n-3]定义成Vn3（3以前的矢量），VCTB[n-2]定义成Vn2（2以前的矢量），VCTB[n-1]定义成Vn1（1以前的矢量），VCTB[n]定义成V0（当前矢量），VCTB[n+1]定义成V1（1以前的矢量），VCTB[n+2]定义成V2（2以前的矢量）。

- 5 在这种场合，首先，查找（调查）当前矢量V0是正交矢量（0，2，4，6）还是斜线化矢量（1，3，5，7）。在正交矢量的场合，标记a0作为“0”，在斜线化矢量的场合，标记a0设置成“1”。

然后，通过以下算式（5）算出Vn2→V3的矢量变化方向（步骤S452）。

$$\text{dr} (V (n-1) \rightarrow V (n)) =$$

$$10 \quad (V (n) - V (n-1) + 10) \text{ AND } 0 \times 07 \cdots \cdots (5)$$

这里，dr的运算结果是“0”时为左旋转，“1”时为左旋转45度，“2”时为同方向，“3”时为右旋转45度，“4”时为右旋转，“5”时为右旋转135度（不存在），“6”时为反对（不存在），“7”时为左旋转135度（不存在）。

- 15 接着，通过以下的运算式（6），以对应当前矢量的补正数据求出设定的表132的地址addr（步骤S453）。

$$\text{addr} = 1250 \times \text{dr} (V1 \rightarrow V2) + 250 \times \text{dr} (V0 \rightarrow V1) +$$

$$125 \times a0 + 25 \times \text{dr} (Vn3 \rightarrow Vn2) + 5 \times \text{dr} (Vn1 \rightarrow V0) \cdots \cdots (6)$$

- 然后，求出对应于当前矢量的补正数据，以该补正数据作为补正数据H的
20 addr字节数的数据。这里，补正数据H为“0”～“8”的9个值，在H为“0”的场合，“不补正”（步骤S456），在H为“8”的场合，当前矢量决定的基准坐标不使用引线，而是包含在下次以后的引线处理中的“跳过”（步骤S458），在H为“1”～“7”的场合，进行以后的轮廓坐标补正处理，并以1个单位补正坐标位置。补正数据H和补正方向的关系在图26的<表12>中示
25 出。

下面说明轮廓坐标补正处理方法。补正方向Hdr由以下的运算式（7）运算并决定（步骤S457）。

$$\text{Hdr} = (H + V0 + 7) \text{ AND } 7 \cdots \cdots (7)$$

然后，根据图27所示的<表13>求出XY方向的补正点数（步骤S459）。

- 30 这里，HX、HY是每一放大倍率确定的补正点数，通过把补正后相邻线之间以

不紧密的方式连接起来，使其在放大倍率的1/2以下。图28作为<表14>，示出了每一倍率的补正点数作为参考值的情况。相对于通过当前矢量确定的基准坐标来说，只有补正点数移动的坐标看作引线用坐标（步骤S480）。

此外，上述矢量的计算，既不能求出储存补正数据的表132的地址，而且
5 在此也不能用计算方法求出补正数据，通过连续的矢量值的模式确定各个补正
数据，采用参照该补正数据的集合的补正表132的形式。这样，特别是通过例
如，用别的字形字体置换补正数据，才能进行产生字体特征的轮廓平滑化处
理。

下文，说明点补正例子中的两个示例。

10 如图29作为<表15>所示的例子示出了：使始点坐标[X0, Y0]为[6, 2]，
斜线化矢量行列↑VCTIB=[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 2,
1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 4, 3, 4, 4, 4, 4, 4, 4, 4,
4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6]; 对于X方
向3倍、Y方向4倍的放大文字来说，用X方向补正点值“1”、Y方向补正点值
15 “2”引线的情况。表中，LINE{ (X1, Y1) — (X2, Y2) }表示在坐标
(X1, Y1) — (X2, Y2) 之间的引线处理。另外，LINE SKIP表示跳过引
线处理。

图30中的作为<表16>所示的例子示出了：始点为[14, 5]、斜线化矢量数据↑VCT2B=[4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 7, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 5]; 对于X方向3倍、Y方向4倍的放大加工文字来说，用X方向补正点值“1”、Y方向补正点值“2”引线的情况。

以下，详细说明步骤S480以后的引线。这里所谓的引线处理是指：确保在基本的位图数据尺寸上乘以X、Y方向的放大率所得到的尺寸数据区域为2个，

25 在按照上述确定的引线用坐标的2点之间通过插入作为外框用线及涂满用线的点进行描绘。在这里，把外框用线称作笔画线，外框用描绘区域称作笔画区域，涂满用线称作填充线，涂满用描绘区域称作填充区域，并继续说明。另外，把引线对象的2点看作 $P1 = (X1, Y1)$ 、 $P2 = (X2, Y2)$ 。

首先说明笔画线的处理。在这里，在笔画区域内，在上述确定的引线用坐
30 标的2点之间插入点并描绘，以作为字形外框用线。这时，为了得到光滑的

线，通过线的倾斜，代替基准轴（步骤S482）插入点。

具体地，为了确定引线基准轴，以X、Y变化量大的一方为基准轴，求出2点之间的X、Y的差 $\Delta X = \text{ABS}(X2 - X1)$ 、 $\Delta Y = \text{ABS}(Y2 - Y1)$ ，当 $\Delta X \geq \Delta Y$ 时，以X轴为基准，沿X方向扫描，当 $\Delta X < \Delta Y$ 时，以Y轴为基准，沿Y方向扫描，在这里，ABS是求绝对值的函数。

在X轴基准引线处理（沿X方向的扫描）中，比较X1、X2，从X坐标小的一方顺次描绘点。在这里，当 $X1 \geq X2$ 时，以 $P2 \rightarrow P1$ 的顺序描绘（始点 $PS = P2$ ，终点 $PE = P1$ ），当 $X1 < X2$ 时，以 $P1 \rightarrow P2$ 的顺序描绘（始点 $PS = P1$ ，终点 $PE = P2$ ）。

在Y轴基准引线处理（沿Y方向的扫描）中，比较Y1、Y2，从Y坐标小的一方开始引线。在这里，当 $Y1 \geq Y2$ 时，以 $P2 \rightarrow P1$ 的顺序描绘（始点 $PS = P2$ ，终点 $PE = P1$ ），当 $Y1 < Y2$ 时，以 $P1 \rightarrow P2$ 的顺序描绘（始点 $PS = P1$ ，终点 $PE = P2$ ）。

笔画线的处理是，用始点、终点、全部插入点来描绘点，描绘的点看作笔画区域的数据的逻辑或（OR）的运算（步骤S483）。

下文说明填充线的处理。在这里，在填充区域内，在上述确定的引线用坐标的2点之间插入点并描绘，以作为字形涂满用线。用于插入点的基准轴只有X方向。

具体地，在X轴基准引线处理（沿X方向的扫描）中，比较X1、X2，从X坐标小的一方顺次描绘点（步骤S484）。当 $X1 \geq X2$ 时，以 $P2 \rightarrow P1$ 的顺序描绘（始点 $PS = P2$ ，终点 $PE = P1$ ），当 $X1 < X2$ 时，以 $P1 \rightarrow P2$ 的顺序描绘（始点 $PS = P1$ ，终点 $PE = P2$ ）。填充线的处理是，在始点、插入点中描绘点，终点看作是未处理。另外，描绘的点看作是填充区域的数据的异或逻辑（XOR）的运算（步骤S485）。

参照图31～图41详细说明上述的笔画线、填充线的处理中的2点间描绘直线的算法。

在图31所示的<曲线1>中，在2点之间（图中★符号）引直线的场合，象在曲线上描绘直线一样，要用存储器在带小数点的值坐标上描绘点是不可能的。实际上，如图中的●所示，在2点之间的直线上，在最接近的整数值坐标处插入点。

这种插入点组可以通过图32中的作为<明细表1>所示的算法顺次求出。在这里，插入点组用 $(x+x1, y)$ 表示。即是说，对每一次步进计数器所算的 x 进行运算（round函数），把带小数点的坐标值圆整成最接近的整数。在如图32所示的例子中，虽然使用了圆整成整数的函数round，但是，在将其置换成通常的四则运算的场合，也能得到如图33的<明细表2>所示的算法。这里，插入点组是 (x, y) 。

实际上，在用程序实施图33所示的运算式的场合，作为另一种情况，在用硬件实现的场合，包括 dy/dx 的除法、带小数点的变量 1 、 $1/2$ 的比较等都是难关。因此，在图34或图35中，使分别作为<明细表3>、<明细表4>所示的算法变形。由于图32~图35中的用明细表所示的算法与本发明的目的没有直接关系，故省略其说明。

到此为止的说明，已经描述了 $dx \geq dy$ 的情况。这里的 x 是用于判断每次步进计数器增量为 1 时“ y 步进计数器的增量是否为 1 ”的情况，所以，增量 dy/dx 不能适用于步进计数器 1 以上的场合。如果试图在 $dx < dy$ 的场合实施，则会出现图36中的曲线（<曲线2>）所示的情况（Y坐标正方向=向下方向）。

为了避免上述问题，要反复地将 1 与 dx 进行比较并进行处理和变更。这种算法在图37的<明细表5>中示出。通过这种算法，即使在增量 dy/dx 大于 1 的场合，也能在图38的曲线（<曲线3>）所表示的直线上得到最近的坐标点组。

而在 $dx < dy$ 的场合的处理，也不会发生问题，用上述算法可生成直线。不过，在 dx 相对 dy 为更大的值（急剧倾斜的场合）的情况下，按照图39的曲线（<曲线4>）所表示的方式插入的点会特别少。

实际上，作为涂满用的算法是非常有用的。可是，在作为用于得到字形外框的引线场合，如图39所示，点的加密部分和点的稀疏部分相混，不能变成保持漂亮外形的字形。

插入点组变密时，不依赖于线的倾斜程度，根据 dx ， dy 的大小可进行分类。当 $dx < dy$ 时，正好替换X轴和Y轴进行运算处理。即是说，每当 y 步进计数器的增量是 1 时，不进行 x 步进计数器的增量是否是 1 的判断，把增量 dy/dx 或 dx/dy 正规化到 1 以下。这时，算法变成图40中的作为<明细表6>所示的形式。根据该算法，结果在图41所示的曲线（<曲线5>）中表示，以加密的方式描绘点。

归纳上文所表述的情况，笔画线（外框用引线）用算法成为如下形式。即是说，当 $dx \geq dy$ 时，每当x步进计数器的增量是1时，进行判断y是增量与否的处理（X扫描引线处理）。 $dx < dy$ 时，每当y步进计数器增量是1时，进行判断x是增量与否的处理（y扫描引线处理），对始点、终点和插入点进行笔画区域
5 数据和逻辑或（OR）的运算。

下面，说明用笔画线的涂满处理。涂满算法如图42中的作为<明细表7>所示，是通过进行Y方向的1个在前的点的异或逻辑（XOR）的运算实现的。这样，这在用图43A所示的图形（<曲线6>）的例子实施的场合，变成如图43B所示的形式，而且特别不会产生问题。

10 接着，在借助于使用笔画线处理所描绘图44A的例子（<曲线7>）实施的场合，结果变成图44B所示的情况，不能顺利地进行涂满。同样地，在使用图45A所示的长方形（<曲线8>）实施的场合，结果变成图45B所示的形式，不能顺利地进行边a-d，边b-c之间的涂满。

鉴于这种原因，对于以使用涂满算法在1个点到1个点之间进行弥补处理为目的的情况来说，可以例如用1个点以上的点，只从作为涂满方向的Y方向上描绘
15 线段A-C、B-C（图44A）、a-d、b-c（图45A）。

对线段A-C，B-C来说用1个点以上的点的描绘，是在进行Y方向扫描引线处理中描绘的，为了解决此问题，与线段的倾斜程度无关，最好进行X方向的扫描引线处理。

20 图46A（<曲线9>）例示了单纯在坐标点间进行X方向扫描的引线处理的例子，这时试图只进行涂满。在这种场合，如图46B所示，与并用X扫描、Y扫描的引线处理的例子相比，虽然有所改善，但是由于顶点A、B仅是一个点，不存在涂满终点，引线已经结束。

下面说明填充线（涂满用引线）的处理。在图46B中，点A、B是图形的顶点，很难描绘出涂满的终点。因此，在用顶点A、B进行引线处理中，可以不
25 描绘黑点，只描绘白点，但是在单纯引线处理的始点、终点为白点的处理中，即使到顶点C为白点，这一次也可以在点C以后进行引线。

因此，引线处理的始点、终点可与上次或下次的引线重合，利用两次的处理，在从始点到终点的1个前插入点组中，通过在终点未处理填充区域的数据
30 的异或逻辑运算，可以使顶点变成白点。填充线的插入点组生成算法用图47的

明细表形式表示。

根据填充线的处理，图48A的（<曲线10>）△形（顶点A、B、C）遵循以下的顺序。即是说，在图48A中的填充线A-B的处理中，A点是黑点，B点没有处理（白点），在填充线C-B的处理中，C点为黑点，B点没有处理（白点），在填充线A-C的处理中，A点是异或逻辑运算的白点，C点是未处理的黑点。图48A所示的△形中，实施涂满处理变为图48B所示的形式。

填充线（涂满用引线）的算法归纳为以下形式。即，在2点间的坐标中，以X坐标小的一方作为始点。然后，每当x步进计数器增量为1时，判断y为增量与否（X扫描引线处理）。用始点、插入点组进行填充区域数据的异或逻辑运算，终点为未处理。

其次，说明通过涂满处理部115的涂满处理（图4中的步骤S46）。图9以程序方框图示出了处理顺序。以下，参照图9所示的程序方框图进行说明。

利用涂满处理部115，把通过引线处理部114制成的填充线的内侧涂满。填充线的涂满是通过实施Y方向的1个前点的异或逻辑运算XOR（步骤S461）实现的。图49以明细表（<明细表9>）的形式示出了算法。

下面说明笔画区域、填充区域的重合。在这里，通过逻辑或运算，使描绘字形外框的笔画区域和描绘涂满字形的填充区域的内容重合（步骤S462）。该样式表示在图50中的作为动作概念图之中。

最后，说明线幅补正处理部116进行的粗度补正处理（图4中的步骤S47）。图10以程序方框图示出了处理顺序。

在此，通过引线处理部114制成的位图，由于经过矢量化，在X方向、Y方向同时以1个点部分变粗，因此，可实施在X、Y两个方向上把位图分别删除1个点部分的功能。

下面，具体说明处理的方法。首先，在Y方向上扫描坐标（步骤S471），查找点删除的条件（步骤S472），把以下所示的条件的点变成“白”（步骤S473）。在这里，所谓点删除的条件是指，当前坐标点为“黑”，下一坐标点（=Y-1）是“白”，坐标端点为未处理。

接着，在X方向上扫描坐标（步骤S472），查找点删除的条件（步骤S472），把以下所示的条件的点变成“白”（步骤S473）。在这里，所谓点删除的条件是指，当前坐标点为“黑”，下一坐标点（=X+1）是“白”，坐标

端点为未处理。

下面，参照图51A~51D所示的位图说明粗度补正的例子。

在图51A所示的位图例子中，提取轮廓并放大。在这里，矢量数据 $\uparrow$ VCT1=[0, 0, 2, 2, 0, 0, 2, 2, 4, 4, 4, 4, 6, 6, 6, 6]。并且，以X方向的2倍、Y方向的2倍，进行引线、涂满、重合处理，其结果为图51B所示的情况。

由于图51B示出的位图是在X方向放大2倍、在Y方向放大2倍进行描绘时的位图，因此，图形的大小本应是 4×2 的8点部分，但是，在矢量化的场合，如图所示，可能会以+1点部分变粗。这里，+1点并不限于此例，也不是放大率的值，是均匀的。LINE{ (X1, Y1) — (X2, Y2) }表示的是从坐标 (X1, Y1) 向坐标 (X2, Y2) 引线段的处理。

这样，虽然进行了粗度补正处理，但是，在这里，首先在Y方向扫描坐标，在 $F[x, y]=0 \times 1$ （黑）、 $F[x, y+1]=0 \times 0$ （白）的场合，置换成 $F[x, y]=0 \times 0$ （白）。由此，生成图51C所示的位图。具体地，补正用□表示的点。

接着，在X方向扫描坐标，在 $F[x, y]=0 \times 1$ （黑）、 $F[x, y+1]=0 \times 0$ （白）的场合，置换成 $F[x, y]=0 \times 0$ （白）。由此，生成图51D所示的位图。具体地，补正用◇表示的点。进行该粗度补正的结果，使在XY方向变成2倍的图形的宽度成为在XY的各个方向上的本来的 4×2 的8点。

下面，参照图52A~52D所示的位图，说明粗度补正的例外处理（步骤S474）。基本的位图数据如图52A所示，在 24×24 点字形尺寸的端部相接的场合，进行粗度补正。

在X方向、Y方向同时为2倍时，实施引线、涂满、重合处理，变成图52B所示的位图。在引线处理中，由于不能描绘出超过终点坐标放大后的位图尺寸的坐标，因此，图形的大小存在不能以1点部分变大的位置的位置。

单纯进行粗度补正处理，变成如图52C所示的位图。在该场合，补正处理后的图形在XY方向上以1个点列的部分变小。在此，在放大后的图形中，如果在位图坐标端的点为未处理，可得到图52D所示的位图。在图52C、52D中，□表示补正的点。

另外，把存储介质提供给系统或装置，该存储介质用于记录实现上述实施例的轮廓平滑化方法的软件程序代码，该系统或装置的计算机（或CPU或

MPU) 读出储存在存储介质中的程序代码, 并进行实施, 不言而喻, 这样也能实现本发明的目的。

在这种场合, 从存储介质读出的程序代码本身能实现本发明新的功能, 存储该程序代码的存储介质构成本发明。

- 5 作为供给程序代码的存储介质, 可使用例如软盘、硬盘、光盘、光磁盘、CD-ROM、CD-R、磁带、非易失性存储器插件、ROM等。另外, 通过通讯网络从服务器供给程序代码也是可行的。

以上, 说明了本发明的最佳实施形式, 但是, 可以理解, 对于本领域的熟练技术人员来说, 在不脱离本发明的精神和范围的前提下, 可作出各种变更。

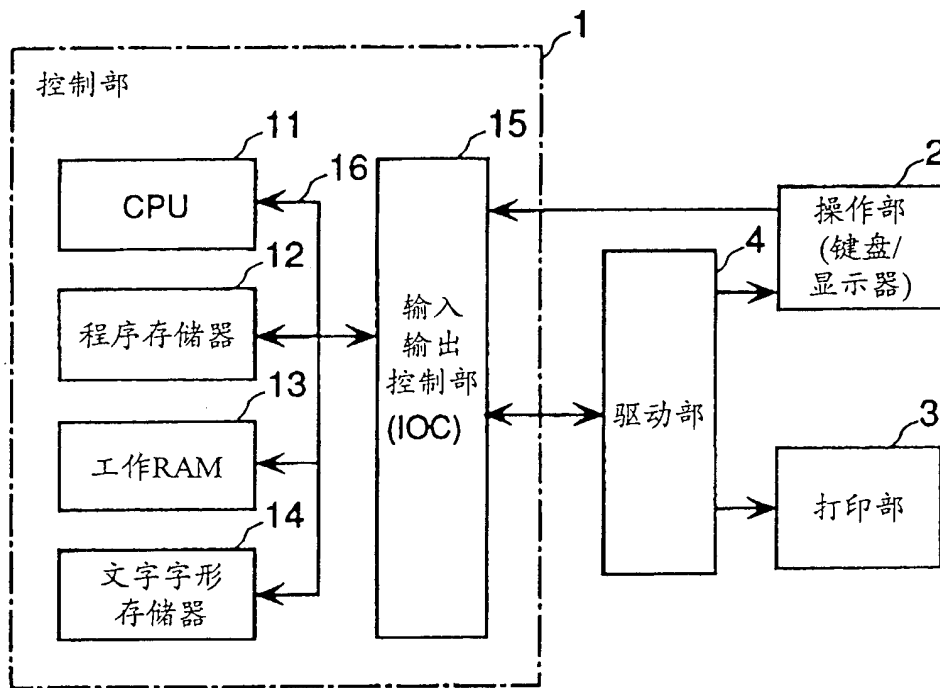


图 1

控制部 1

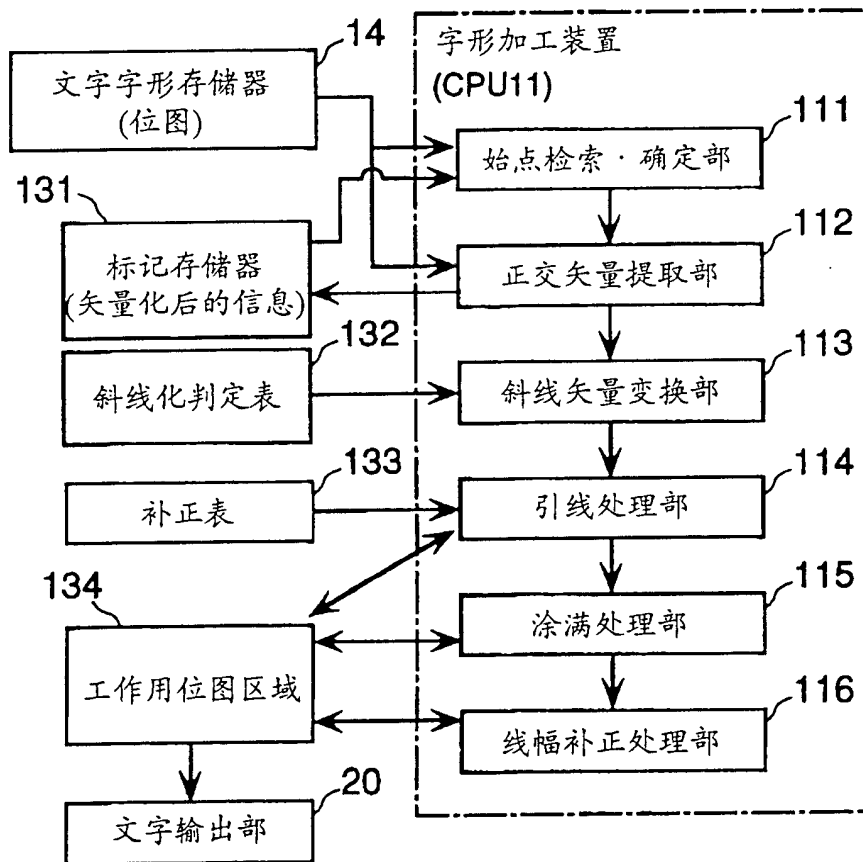


图 2

图 3A

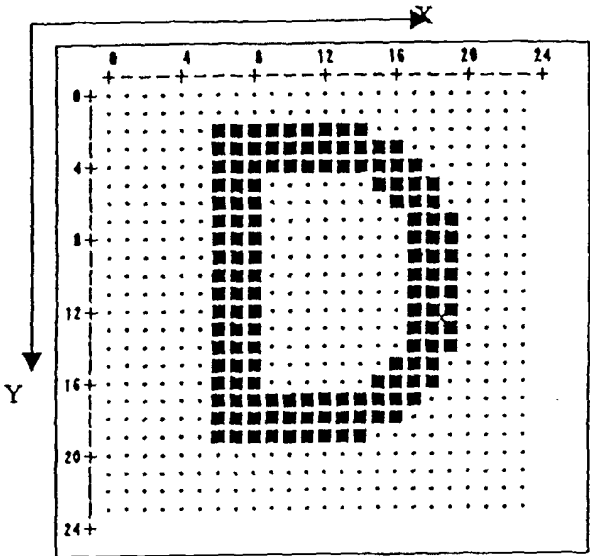


图 3B1

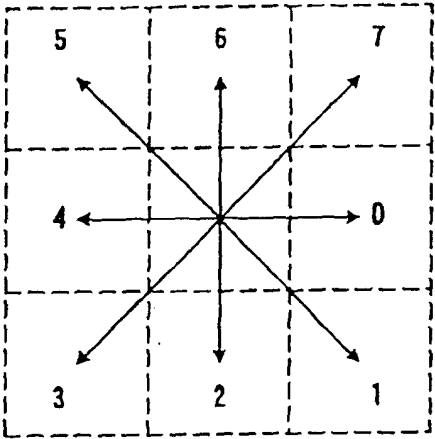


图 3B2

矢量序号	0	1	2	3	4	5	6	7
标记	→	↘	↓	↙	←	↖	↑	↗

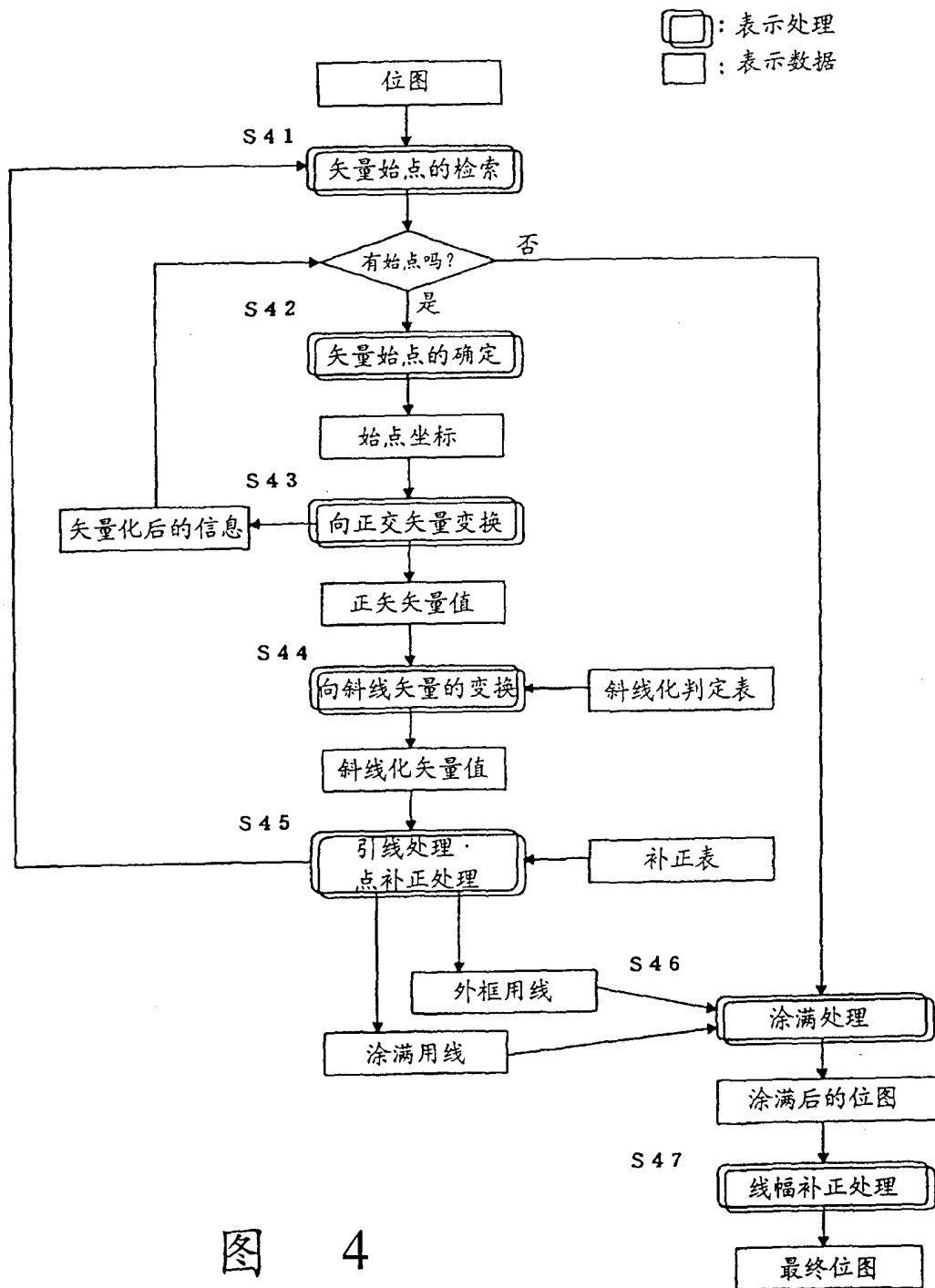


图 4

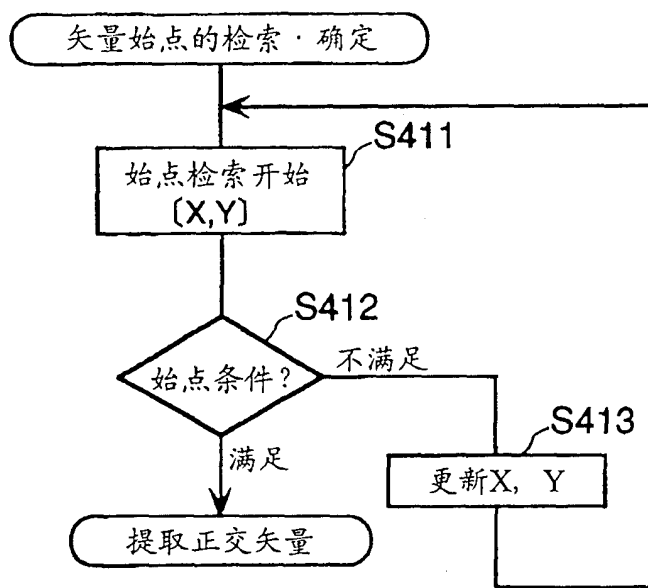


图 5

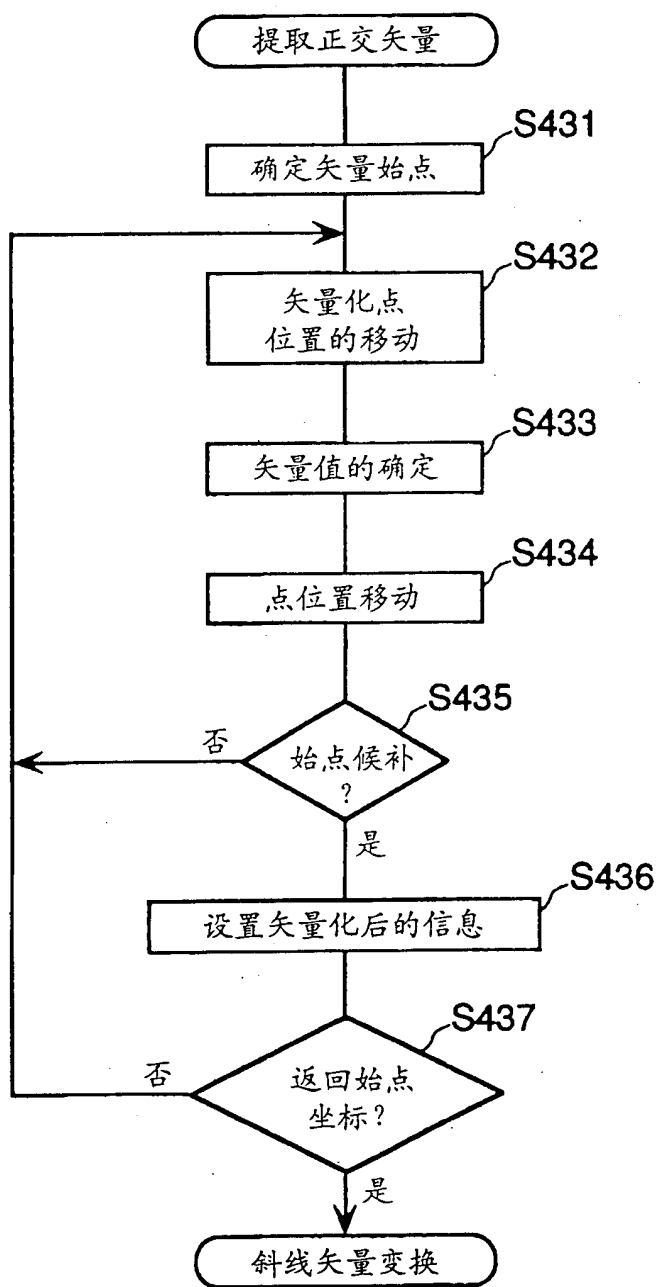


图 6

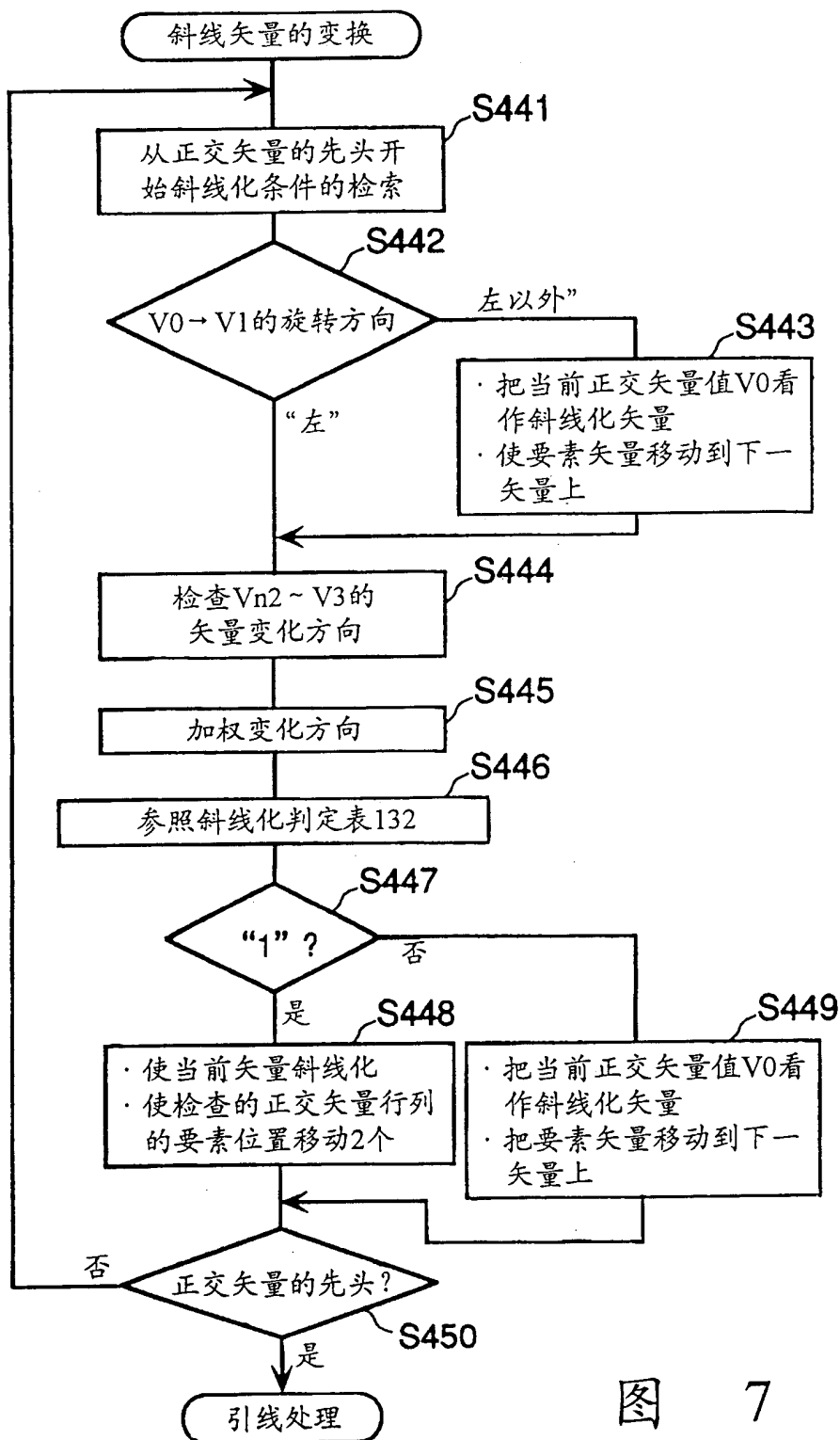


图 7

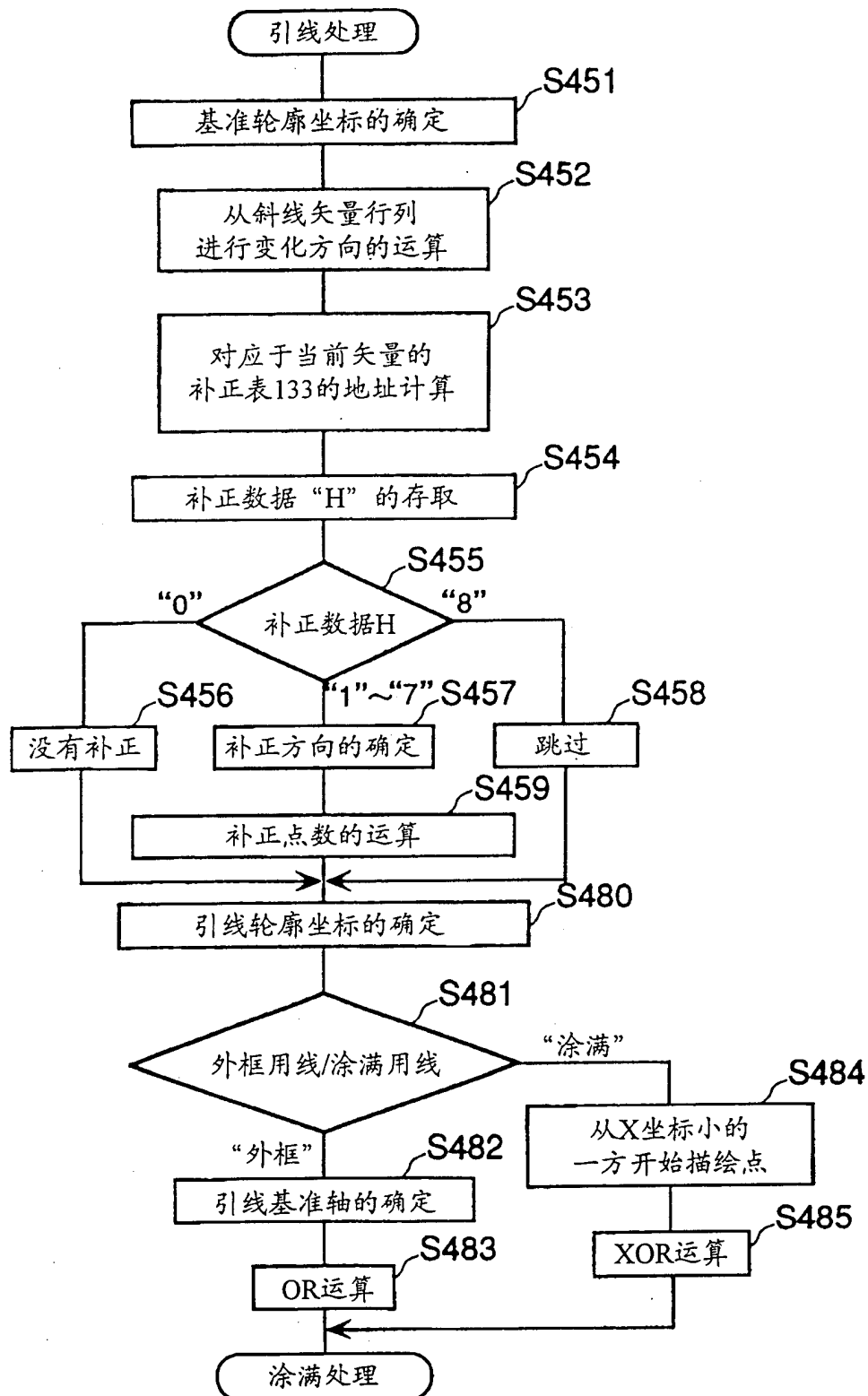


图 8

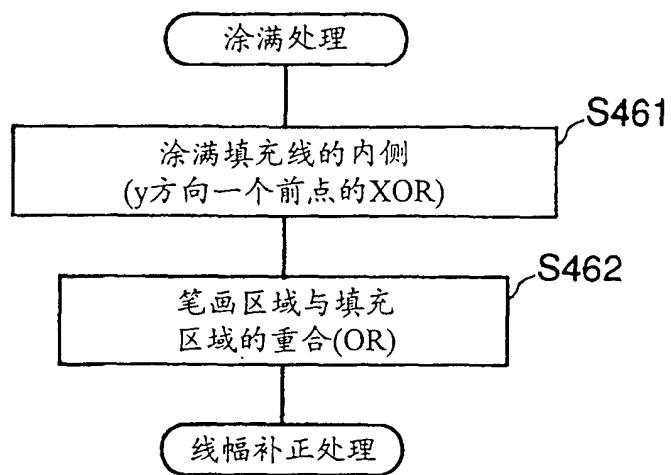


图 9

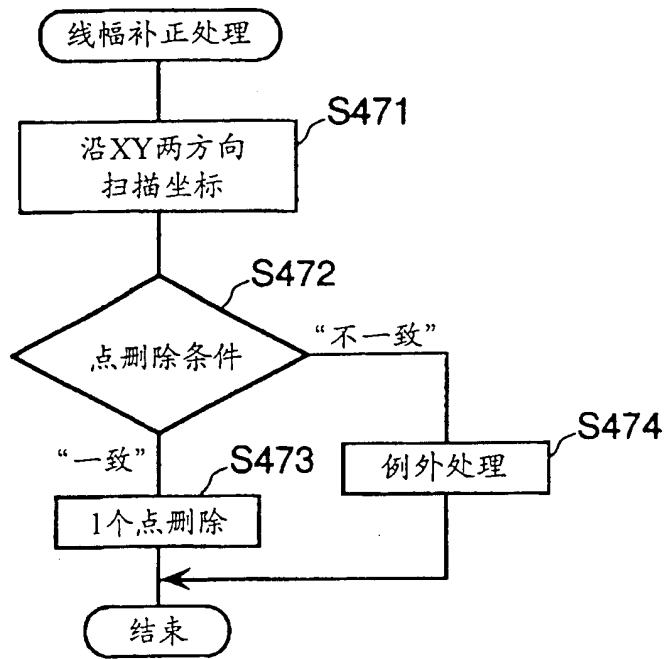


图 10

图 11A

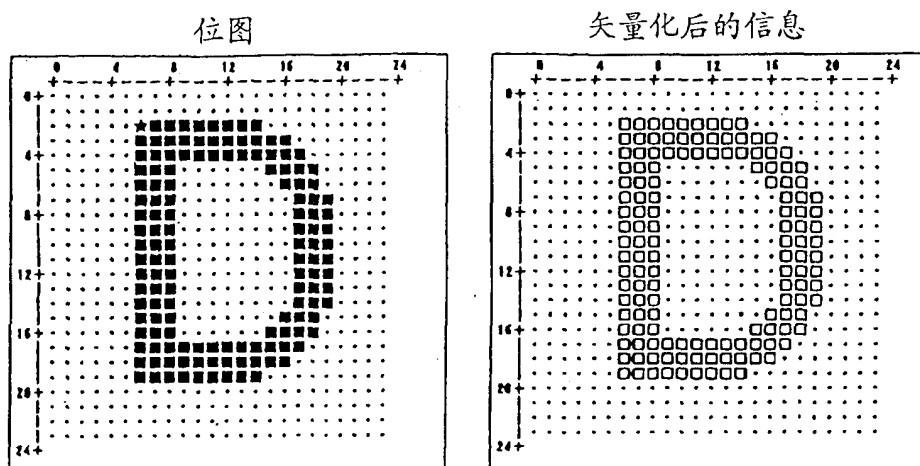


图 11B

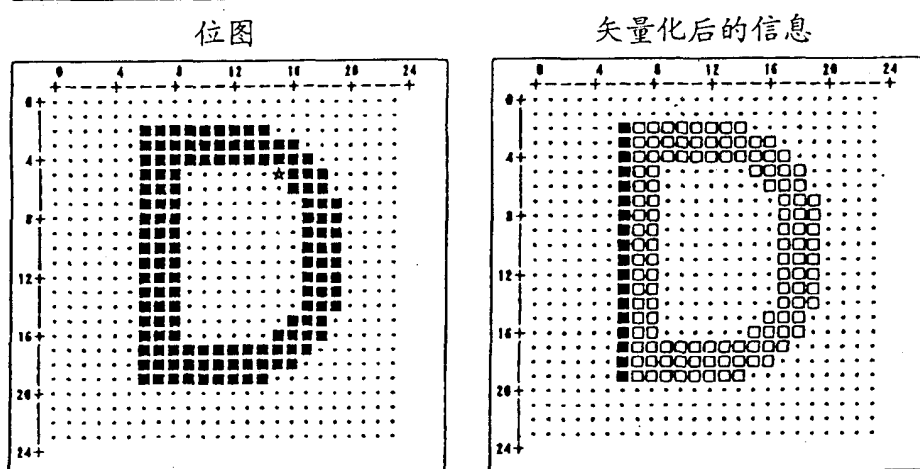
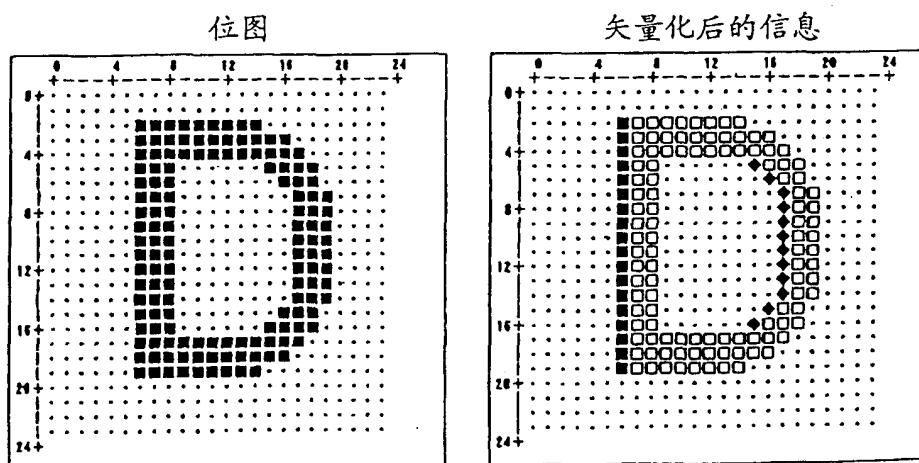


图 11C



<表 1>

条件		矢量值()是序号	备注
始点的左上点	有	← (4)	以后, 使位图的内周矢量化
[X-1, Y-1]	无	→ (0)	以后, 使位图的外周矢量化

图 12

<表 2>

一个前的矢量	第1检索顺序	第2检索顺序	第3检索顺序
向右方向→(0)	向上方向矢量模式	向右方向矢量模式	向下方向矢量模式
向下方向↓(2)	向右方向矢量模式	向下方向矢量模式	向左方向矢量模式
向左方向←(4)	向下方向矢量模式	向左方向矢量模式	向上方向矢量模式
向上方向↑(6)	向左方向矢量模式	向上方向矢量模式	向右方向矢量模式

图 13

<表 3>

检索方向模式	条件	条件图	矢量
向右方向矢量模式	当前位置点 = 黑 上位置点 = 白		→ (0)
向下方向矢量模式	当前位置点 = 白 左位置点 = 黑		↓ (2)
向左方向矢量模式	左位置点 = 白 左上位置点 = 黑		← (4)
向上方向矢量模式	上位置点 = 黑 左上位置点 = 白		↑ (6)

图 14

<表 4>

矢量	矢量化位置 坐标移动
向右方向→ (0)	X方向 + 1、Y方向 + 0
向下方向↓ (2)	X方向 + 0、Y方向 + 1
向左方向← (4)	X方向 - 1、Y方向 + 0
向上方向↑ (6)	X方向 + 0、Y方向 - 1

图 15

矢量变化方向	VCT[n]	VCT[n+1]
左	→	↑
	↓	→
	←	↓
	↑	←
右	→	↓
	↓	←
	←	↑
	↑	→

图 17

矢量变化方向	VCT[n]	VCT[n+1]	变换的斜线矢量
左	→	↑	右上矢量' /
	↓	→	右下矢量' \
	←	↓	左下矢量' /
	↑	←	左上矢量' \

图 18

<表 7>

		V1→V2						左					
		右			同			右			左		
		V2→V3			V2→V3			V2→V3			V2→V3		
		右	同	左	右	同	左	右	同	左	右	同	左
Vn1 ↓ V0	右	○	○	○	○	○	○	○	○	○	○	○	○
	同	○	○	○	○	○	○	○	○	○	○	○	○
	左	○	○	○	○	○	○	○	○	○	○	○	○
同	右	○	○	○	○	○	○	○	○	○	○	○	○
	同	○	○	○	○	○	○	○	○	○	○	○	○
	左	○	○	○	○	○	○	○	○	○	○	○	○
左	右	○	○	○	○	○	○	○	○	○	○	○	○
	同	○	○	○	○	○	○	○	○	○	○	○	○
	左	○	○	○	○	○	○	○	○	○	○	○	○

图 19

<表 8>

		V1→V2					
		右			同		
		V2→V3			V2→V3		
		右	同	左	右	同	左
Vn1 ↓ V0	右	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓
	同	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓
	左	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓
Vn2 ↓ Vn1	右	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓
	同	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓
	左	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓	→ ↑ V0 → ↑ ↓

图 20

斜线化判定表132										
		w 1								
		8	7	6	5	4	3	2	1	0
w 2	8	1	1	1	1	1	1	0	0	0
	7	1	1	1	1	1	1	0	0	0
	6	1	1	1	1	1	1	0	0	0
	5	1	1	1	1	0	0	0	0	0
	4	1	1	1	0	0	0	0	0	0
	3	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

图 21

<表 9>

向量值	→	↖	↓	↙	←	↘	↑	↗
	0	1	2	3	4	5	6	7
X方向移动量	BX	BX	0	-BX	-BX	-BX	0	BX
Y	0	BY	BY	BY	0	-BY	-BY	-BY

图 23

<表 10> 始点坐标以后的坐标运算

矢量值	从前次坐标的移动量	轮廓基准坐标
↑VCT1B[0]=0	(+BX, 0)	(1 4, 6)
↑VCT1B[1]=0	(+BX, 0)	(1 6, 6)
:	:	:
↑VCT1B[9]=1	(+BX, +BY)	(3 2, 9)
↑VCT1B[10]=0	(+BX, 0)	(3 4, 9)
↑VCT1B[11]=1	(+BX, +BY)	(3 6, 1 2)
:	:	:
↑VCT1B[54]=6	(0, -BY)	(3 2, 9)
↑VCT1B[55]=6	(0, -BY)	(1 2, 6)

图 24

<表 11>

始点坐标 - 终点坐标
(1 2, 6) - (1 4, 6)
(1 4, 6) - (1 6, 6)
:
(3 2, 9) - (3 4, 9)
(3 4, 9) - (3 6, 1 2)
:
(3 2, 9) - (1 2, 6)

图 25

<表 12>

补正数据H	补正方向
0	不补正
1	在与当前矢量相同的方向上补正
2	在从当前矢量向右45度的方向上补正
3	在从当前矢量向右90度的方向上补正
4	在从当前矢量向右135度的方向上补正
5	在从当前矢量相反的方向上补正
6	在从当前矢量向左135度的方向上补正
7	在从当前矢量向左90度的方向上补正
8	在从当前矢量向左45度的方向上补正

图 26

<表 3>

Hdr值	0	1	2	3	4	5	6	7
X方向移动量	HX	HX	0	-HX	-HX	-HX	0	HX
Y	0	HY	HY	HY	0	-HY	-HY	-HY

图 27

<表 14>

倍率 (BX, BY)	补正点数 (HX, HY)
1	0
2	0
3	1
4	2
5	2
6	3
7	3
8	4
9	4
1 0	5
2 0	1 0

图 28

<表 15> 点补正的例子1

处理No.	Vn3	Vn2	Vn1	V0	V1	V2	N	Hdr	基准坐标	补正坐标	处理
1	6	6	6	0	0	0	0	-	(18, 8)	←	始点坐标的补正
2	6	6	6	0	0	0	9	-	(21, 8)	←	LINE SKIP
3	6	6	6	0	0	0	0	-	(24, 8)	←	LINE((18, 8)-(24, 8))
4	0	0	0	0	0	0	0	-	(27, 8)	←	LINE((24, 8)-(27, 8))
5	0	0	0	0	0	0	0	-	(30, 8)	←	LINE((27, 8)-(30, 8))
6	0	0	0	0	0	0	0	-	(33, 8)	←	LINE((30, 8)-(33, 8))
7	0	0	0	0	0	0	0	-	(36, 8)	←	LINE((33, 8)-(36, 8))
8	0	0	0	0	0	1	9	-	(39, 8)	←	LINE SKIP
9	0	0	0	0	1	0	9	-	(42, 8)	←	LINE SKIP
10	0	0	0	1	0	1	2	2	(45, 8)	(45, 10)	LINE((36, 8)-(45, 10))
11	0	0	1	0	1	1	9	-	(48, 12)	←	LINE SKIP
12	0	1	0	1	1	2	2	2	(51, 12)	(51, 14)	LINE((45, 10)-(51, 14))
13	1	0	1	1	2	1	9	-	(54, 16)	←	LINE SKIP
14	0	1	1	2	1	2	3	4	(57, 20)	(58, 20)	LINE((51, 14)-(58, 20))
15	1	1	2	1	2	2	9	-	(57, 24)	←	LINE SKIP
16	1	2	1	2	2	2	3	4	(60, 28)	(59, 28)	LINE((58, 20)-(59, 28))
17	2	1	2	2	2	2	9	-	(60, 32)	←	LINE SKIP
18	1	2	2	2	2	2	9	-	(60, 36)	←	LINE SKIP
19	2	2	2	2	2	2	0	-	(60, 40)	←	LINE((59, 28)-(60, 40))
20	2	2	2	2	2	2	0	-	(60, 44)	←	LINE((60, 40)-(60, 44))
21	2	2	2	2	2	2	0	-	(60, 48)	←	LINE((60, 44)-(60, 48))
22	2	2	2	2	2	3	9	-	(60, 52)	←	LINE SKIP
23	2	2	2	2	2	3	9	-	(60, 56)	←	LINE SKIP
24	2	2	2	3	2	3	2	4	(60, 60)	(59, 60)	LINE((60, 48)-(59, 60))
25	2	2	3	2	3	3	9	-	(57, 64)	←	LINE SKIP
26	2	3	2	3	3	4	2	4	(57, 68)	(56, 68)	LINE((59, 60)-(56, 68))
27	3	2	3	3	4	3	9	-	(54, 72)	←	LINE SKIP
28	2	3	3	4	3	4	3	6	(51, 76)	(51, 74)	LINE((56, 68)-(51, 74))
29	3	3	4	3	4	4	9	-	(48, 76)	←	LINE SKIP
30	3	4	3	4	4	4	3	6	(45, 80)	(45, 78)	LINE((51, 74)-(45, 78))
31	4	3	4	4	4	4	9	-	(42, 80)	←	LINE SKIP
32	3	4	4	4	4	4	9	-	(39, 80)	←	LINE SKIP
33	4	4	4	4	4	4	0	-	(36, 80)	←	LINE((48, 78)-(36, 80))
34	4	4	4	4	4	4	0	-	(33, 80)	←	LINE((38, 80)-(33, 80))
35	4	4	4	4	4	4	0	-	(30, 80)	←	LINE((33, 80)-(30, 80))
36	4	4	4	4	4	4	0	-	(27, 80)	←	LINE((30, 80)-(27, 80))
37	4	4	4	4	4	6	0	-	(24, 80)	←	LINE((27, 80)-(24, 80))
38	4	4	4	4	6	6	9	-	(21, 80)	←	LINE SKIP
39	4	4	4	6	6	6	0	-	(18, 80)	←	LINE((24, 80)-(18, 80))
40	4	4	6	6	6	6	9	-	(18, 76)	←	LINE SKIP
41	4	6	6	6	6	6	0	-	(18, 72)	←	LINE((18, 80)-(18, 72))
42	6	6	6	6	6	6	0	-	(18, 68)	←	LINE((18, 72)-(18, 68))
43	6	6	6	6	6	6	0	-	(18, 64)	←	LINE((18, 68)-(18, 64))
44	6	6	6	6	6	6	0	-	(18, 60)	←	LINE((18, 64)-(18, 60))
45	6	6	6	6	6	6	0	-	(18, 56)	←	LINE((18, 60)-(18, 56))
46	6	6	6	6	6	6	0	-	(18, 52)	←	LINE((18, 56)-(18, 52))
47	6	6	6	6	6	6	0	-	(18, 48)	←	LINE((18, 52)-(18, 48))
48	6	6	6	6	6	6	0	-	(18, 44)	←	LINE((18, 48)-(18, 44))
49	6	6	6	6	6	6	0	-	(18, 40)	←	LINE((18, 44)-(18, 40))
50	6	6	6	6	6	6	0	-	(18, 36)	←	LINE((18, 40)-(18, 36))
51	6	6	6	6	6	6	0	-	(18, 32)	←	LINE((18, 36)-(18, 32))
52	6	6	6	6	6	6	0	-	(18, 28)	←	LINE((18, 32)-(18, 28))
53	6	6	6	6	6	6	0	-	(18, 24)	←	LINE((18, 28)-(18, 24))
54	6	6	6	6	6	6	0	-	(18, 20)	←	LINE((18, 24)-(18, 20))
55	6	6	6	6	6	6	0	-	(18, 16)	←	LINE((18, 20)-(18, 16))
56	6	6	6	6	6	0	9	-	(18, 12)	←	LINE((18, 16)-(18, 12))
57	6	6	6	0	0	0	-	-	(18, 8)	←	基准坐标返回始点
58											始点-终点 引线 LINE((18, 12)-(18, 8)) * 始点坐标使用通过 处理No.1 正的值

图 29

<表 16> 点补正的例子2

处理No.	Vn3	Vn2	Vn1	V0	V1	V2	H	Hdr	基准坐标	补正坐标	处理
1	5	5	5	4	4	4	7	2	(42, 20)	(42, 22)	始点坐标的补正
2	5	5	4	4	4	4	0	-	(39, 20)	←	LINE ((42,22)-(39,20))
3	5	4	4	4	4	4	9	-	(36, 20)	←	LINE SKIP
4	4	4	4	4	4	2	0	-	(33, 20)	←	LINE ((39,20)-(33,20))
5	4	4	4	4	2	2	0	-	(30, 20)	←	LINE ((33,20)-(30,20))
6	4	4	4	2	2	2	0	-	(27, 20)	←	LINE ((30,20)-(27,20))
7	4	4	2	2	2	2	0	-	(27, 24)	←	LINE ((27,20)-(27,24))
8	4	2	2	2	2	2	0	-	(27, 28)	←	LINE ((27,24)-(27,28))
9	2	2	2	2	2	2	0	-	(27, 32)	←	LINE ((27,28)-(27,32))
10	2	2	2	2	2	2	0	-	(27, 36)	←	LINE ((27,32)-(27,36))
11	2	2	2	2	2	2	0	-	(27, 40)	←	LINE ((27,36)-(27,40))
12	2	2	2	2	2	2	0	-	(27, 44)	←	LINE ((27,40)-(27,44))
13	2	2	2	2	2	2	0	-	(27, 48)	←	LINE ((27,44)-(27,48))
14	2	2	2	2	2	2	0	-	(27, 52)	←	LINE ((27,48)-(27,52))
15	2	2	2	2	2	2	0	-	(27, 56)	←	LINE ((27,52)-(27,56))
16	2	2	2	2	2	0	0	-	(27, 60)	←	LINE ((27,56)-(27,60))
17	2	2	2	2	0	0	0	-	(27, 64)	←	LINE ((27,60)-(27,64))
18	2	2	0	0	0	0	0	-	(27, 68)	←	LINE ((27,64)-(27,68))
19	2	2	0	0	0	0	0	-	(30, 68)	←	LINE ((27,68)-(30,68))
20	2	0	0	0	0	0	0	-	(33, 68)	←	LINE ((30,68)-(33,68))
21	0	0	0	0	0	7	9	-	(36, 68)	←	LINE SKIP
22	0	0	0	0	7	7	0	-	(39, 68)	←	LINE ((33,68)-(39,68))
23	0	0	0	7	7	7	8	6	(42, 68)	(42, 66)	LINE ((39,68)-(42,66))
24	0	0	7	7	7	6	0	-	(45, 64)	←	LINE ((42,66)-(45,64))
25	0	7	7	7	6	6	0	-	(48, 60)	←	LINE ((45,64)-(48,60))
26	7	7	7	6	6	6	7	4	(51, 56)	(50, 56)	LINE ((48,60)-(50,56))
27	7	7	6	6	6	6	0	-	(51, 52)	←	LINE ((50,56)-(51,52))
28	7	6	6	6	6	6	9	-	(51, 48)	←	LINE SKIP
29	6	6	6	6	6	6	0	-	(51, 44)	←	LINE ((51,52)-(51,44))
30	6	6	6	6	6	5	9	-	(51, 40)	←	LINE SKIP
31	6	6	6	6	5	5	0	-	(51, 36)	←	LINE ((51,44)-(51,36))
32	6	6	6	5	5	5	8	4	(51, 32)	(50, 32)	LINE ((51,36)-(50,32))
33	6	6	5	5	5	4	0	-	(48, 28)	←	LINE ((50,32)-(48,28))
34	6	5	5	5	4	4	0	-	(45, 24)	←	LINE ((48,28)-(45,24))
35	5	5	5	4	4	4	-	-	(42, 20)	←	基准坐标
36											始点-终点 引线 LINE ((45,24)-(42,22)) *始点坐标使用由 处理No.1补正的值

图 30

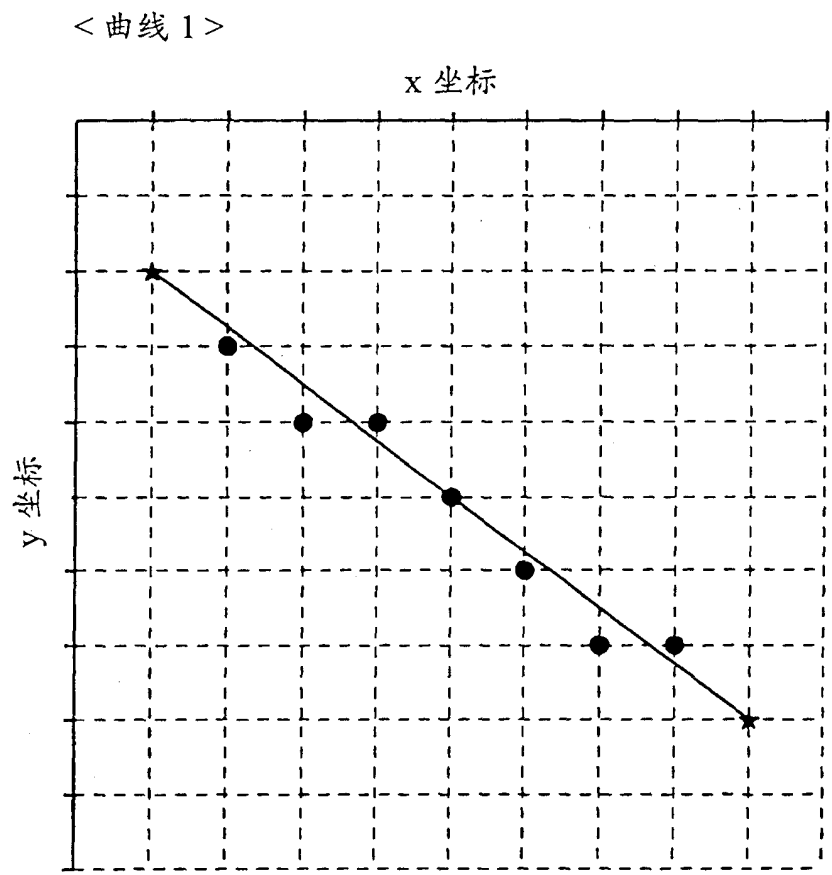


图 31

< 明细表 > 1

```
dx = x2 - x1;           // 看作 x2 > x1
dy = y2 - y1;           // 看作 dx >= dy
l = 0;
x = 0;

while( x <= x2 ){
    //round -> 向最接近的整数圆整
    y = round( x1 + x * dy / dx );
    F[y][x+x1] = F[y][x+x1] | 0x01;    // 逻辑或
    x = x + 1;
}
```

图 32

< 明细表 2 >

```
dx = x2 - x1;          // 看作 x2 > x1, dx >= dy

if( y2 >= y1 ){         // dy = abs(y2-y1)
    dy = y2 - y1;
    step = 1;
}
else{
    dy = y1 - y2;
    step = -1;
}

l = 0;    x = x1;    y = y1;

while ( x <= x2 ) {
    if( l >= 1/2 ){
        l = l - 1;
        y = y + step;
    }
    F[y][x] = F[y][x] | 0x01;    // 逻辑和
    l = l + dy/dx;
    x = x + 1;
}
```

图 33

< 明细表 3 >

```
dx = x2 - x1;          // 看作 x2 > x1, dx >= dy

if( y2 >= y1 ){        // dy = abs(y2-y1)
    dy = y2 - y1;
    step = 1;
}
else{
    dy = y1 - y2;
    step = -1;
}

l = dx;    x = x1;    y = y1;

while( x <= x2 )
    if( l >= ( 2 * dx ) ){
        l = l - 2 * dx;
        y = y + step;
    }
    F[y][x] = F[y][x] | 0x01;    // 逻辑和
    l = l + 2 * dy;
    x = x + 1;
}
```

图 34

<明细表 4>

```
dx = x2 - x1;          // 看作 x2 > x1, dx >= dy

if( y2 >= y1 ){         // dy = abs(y2-y1)
    dy = y2 - y1;
    step = 1;
}
else{
    dy = y1 - y2;
    step = -1;
}

l = 0;    x = x1;    y = y1;

while( x <= x2 )
    if( ( 2 * l ) >= dx ){
        l = l - dx;
        y = y + 1;
    }
    F[y][x] = F[y][x] | 0x01;    // 逻辑和
    l = l + dy;
    x = x + 1;
}
```

图 35

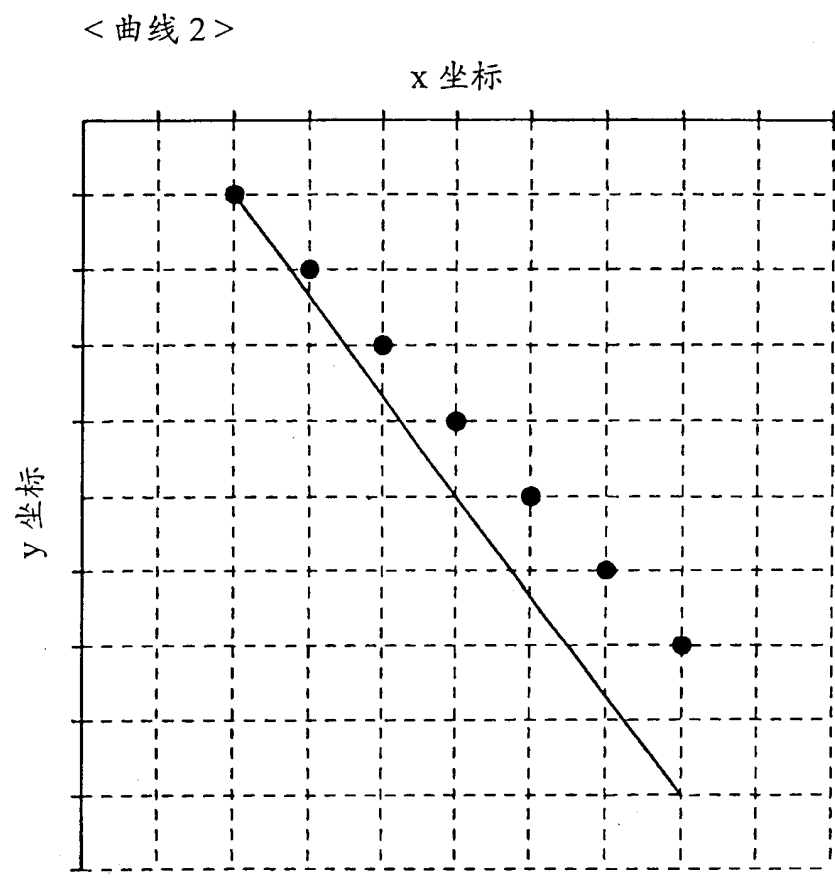


图 36

< 明细表 5 >

```
dx = x2 - x1;          // 看作 x2 > x1, dx >= dy

if( y2 >= y1 ){          // dy = abs(y2-y1)
    dy = y2 - y1;
    step = 1;
}
else{
    dy = y1 - y2;
    step = -1;
}

l = 0;    x = x1;    y = y1;

while( x <= x2 )
    while( ( 2 * l ) >= dx ){
        l = l - dx;
        y = y + step;
    }
    F[y][x] = F[y][x] | 0x01;      // 逻辑和
    l = l + dy;
    x = x + 1;
}
```

图 37

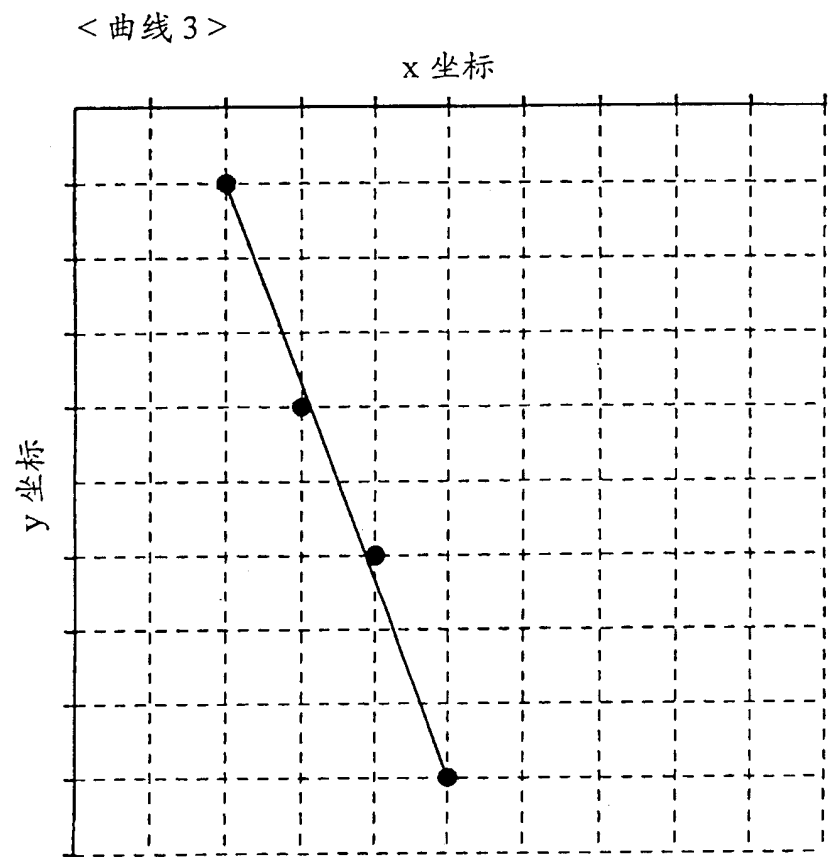


图 38

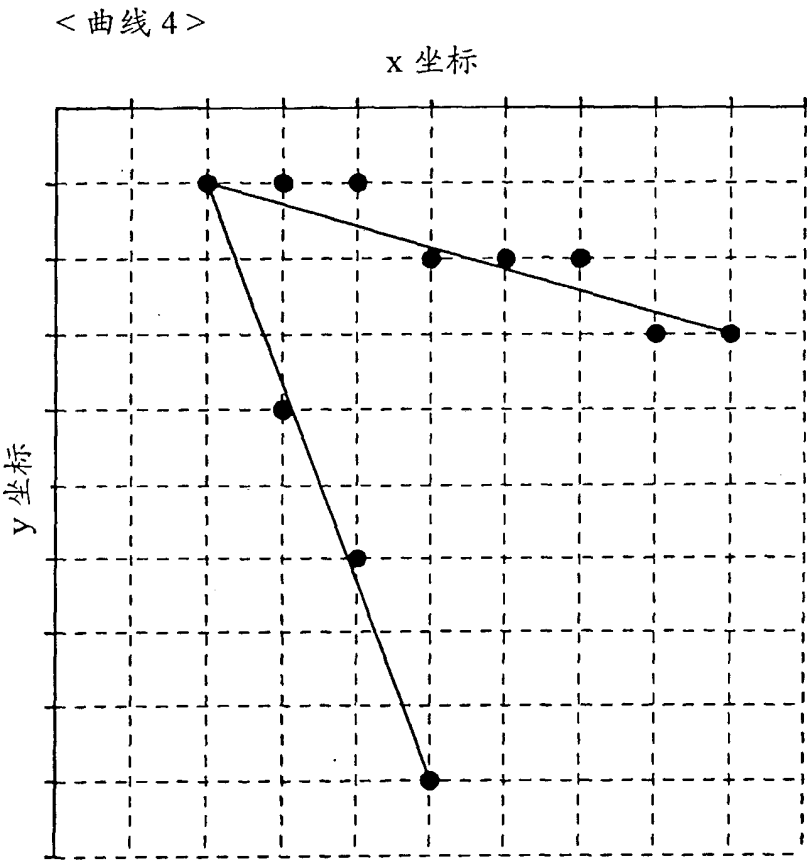


图 39

< 明细表 6 >

```
dy = y2 - y1;          // 看作 y2 > y1, dx < dy

if( x2 >= x1 ){          // dx = abs(x2-x1)
    dx = x2 - x1;
    step = 1;
}
else{
    dx = x1 - x2;
    step = -1;
}

l = 0;    x = x1;    y = y1;

while( y <= y2 )
    if( ( 2 * l ) >= dy ){
        l = l - dy;
        x = x + step;
    }
    F[y][x] = F[y][x] | 0x01;    // 逻辑和
    l = l + dx;
    y = y + 1;
}
```

图 40

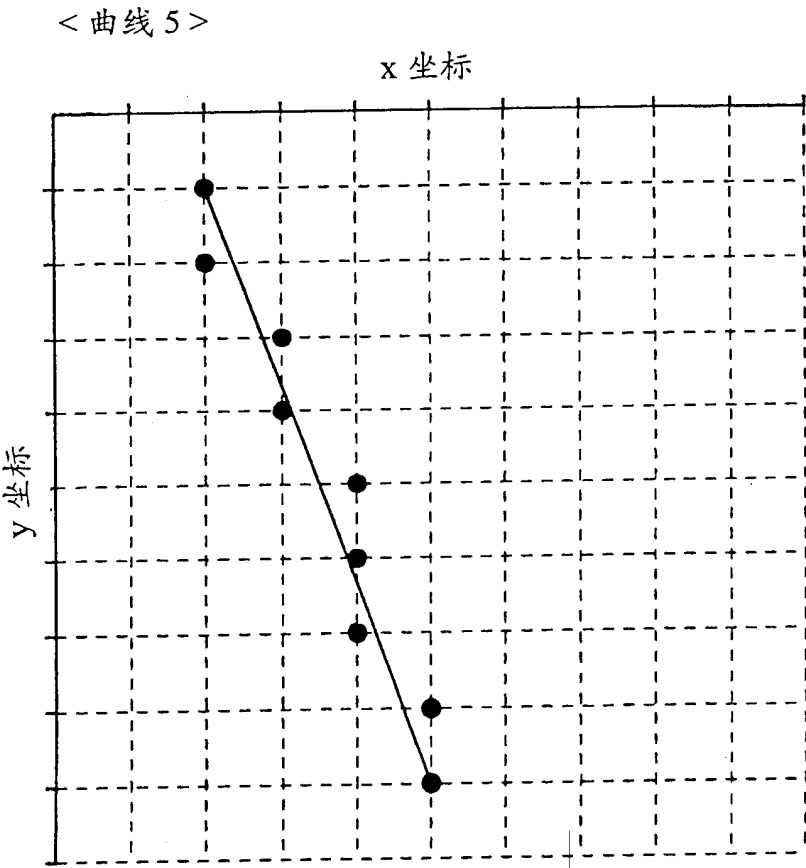


图 41

< 明细表 7 >

```
for( x = 0 ; x < XSIZE ; x++ ){  
    for( y = 0 ; y < YSIZE-1 ; y++ ){  
        F[y+1][x] = F[y+1][x] ^ F[y][x];    // XOR  
    }  
}
```

图 42

图 43A

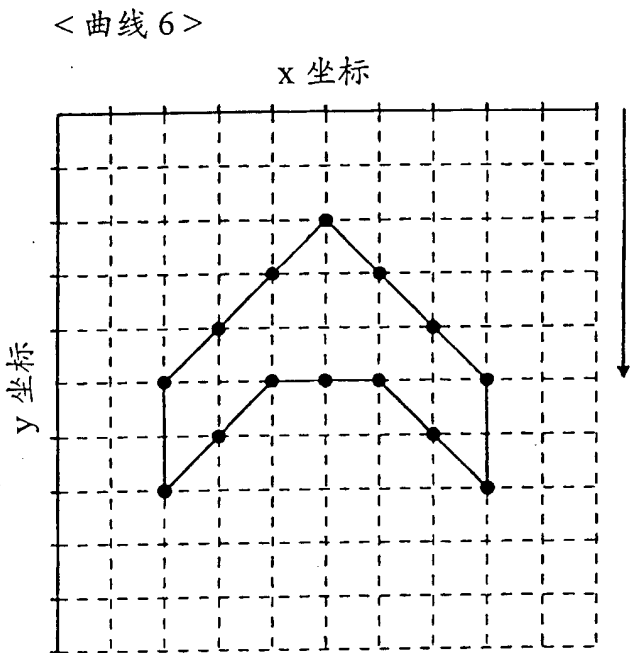


图 43B

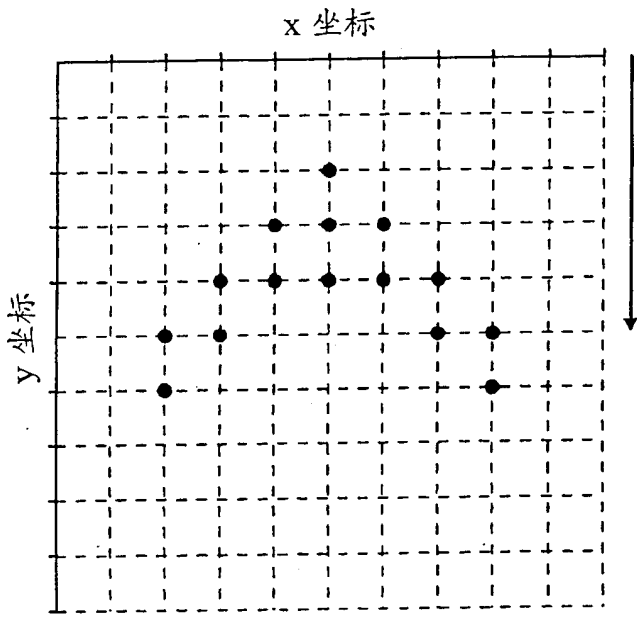


图 44A

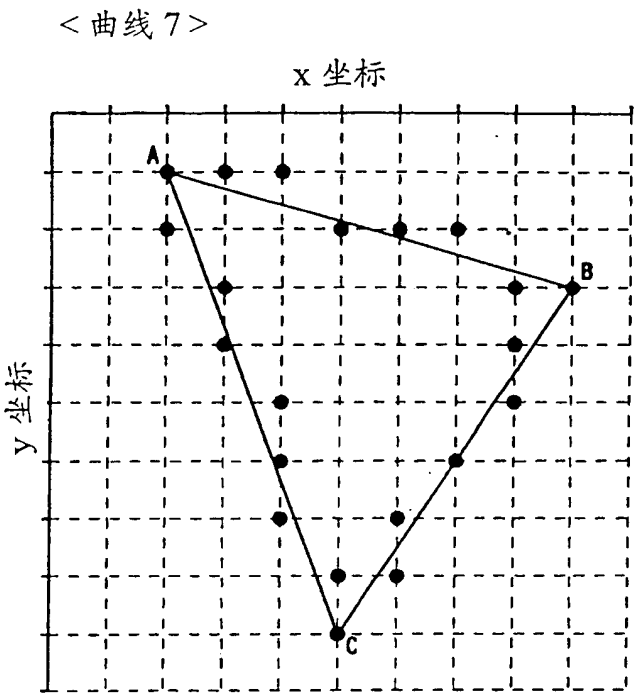
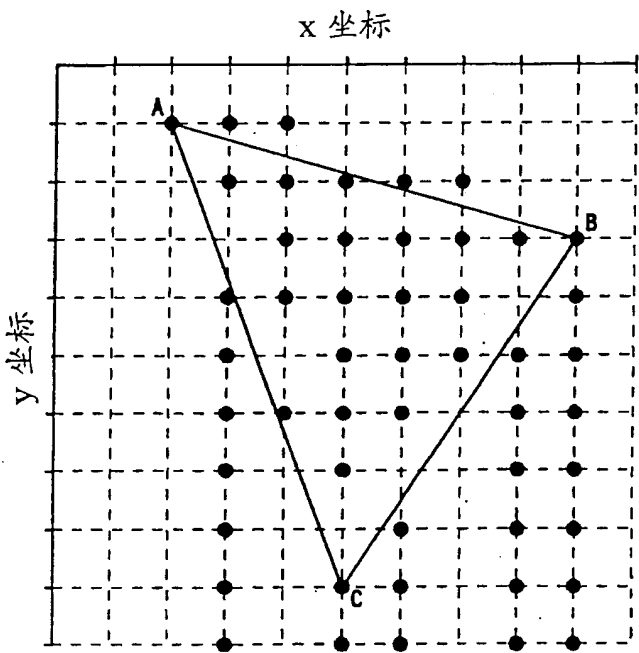


图 44B



< 曲线 8 >

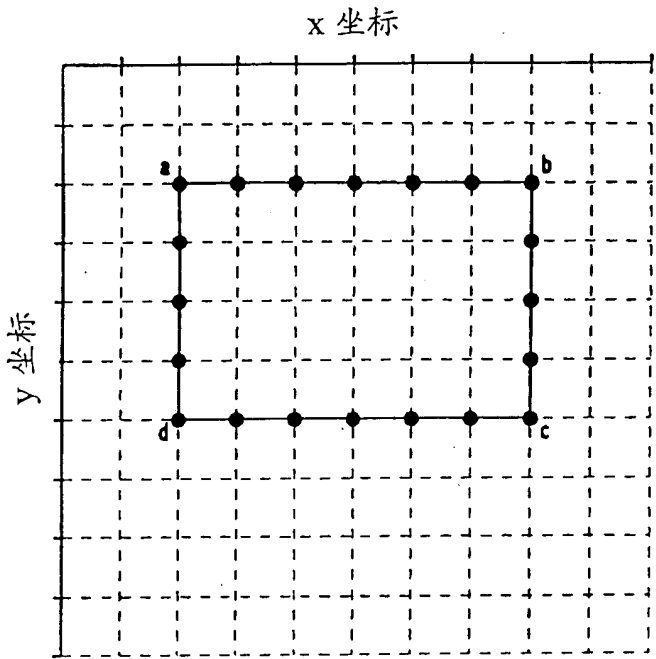


图 45A

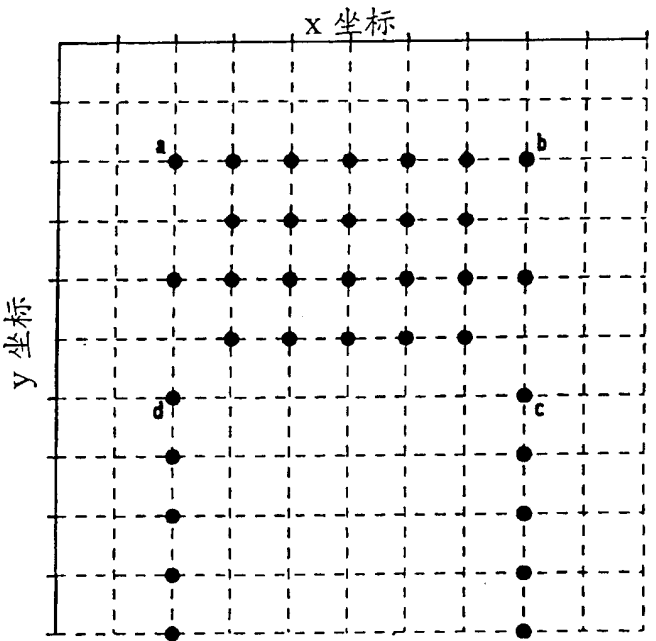


图 45B

图 46A

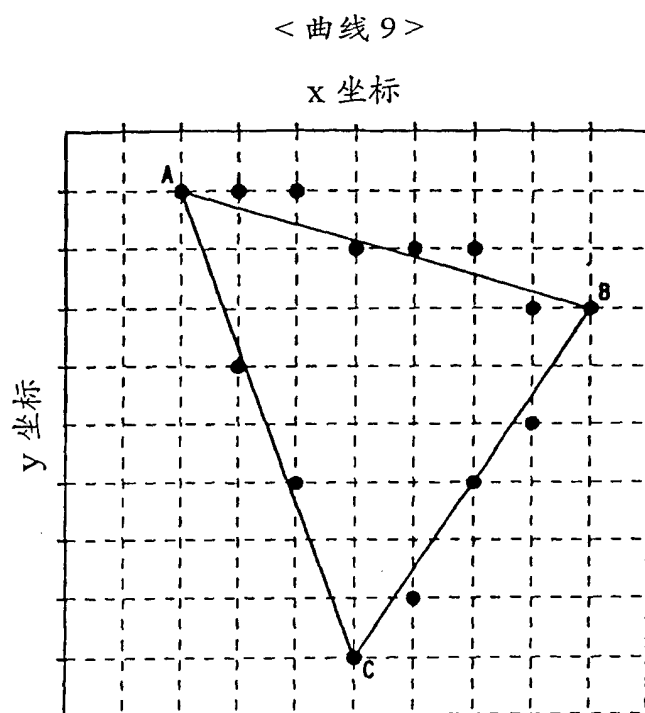
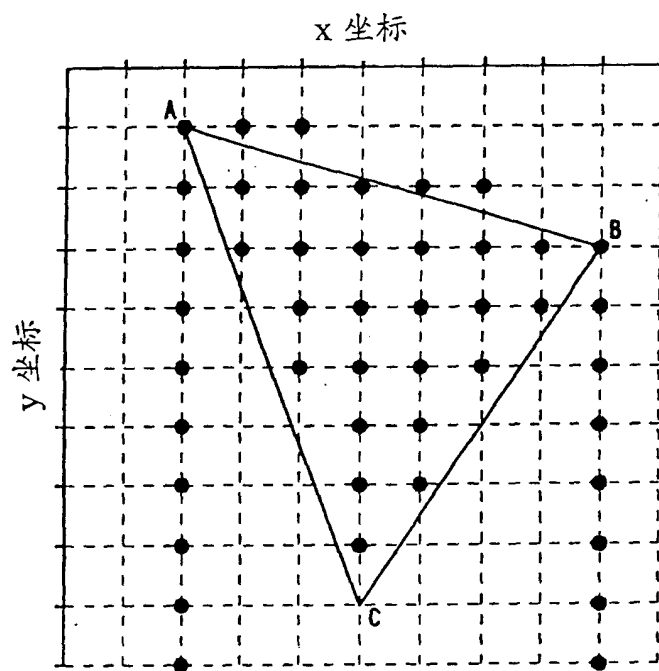


图 46B



<明细表 8>

```
dx = x2 - x1;          // 看作 x2 > x1, dx >= dy

if( y2 >= y1 ){        // dy = abs(y2-y1)
    dy = y2 - y1;
    step = 1;
}
else{
    dy = y1 - y2;
    step = -1;
}

l = 0;    x = x1;    y = y1;

while( x < x2 )        // 终点未处理
    while( ( 2 * l ) >= dx ){
        l = l - dx;
        y = y + step;
    }
    F[y][x] = F[y][x] ^ 0x01;    // 异或逻辑
    l = l + dy;
    x = x + 1;
}
```

图 47

图 48A

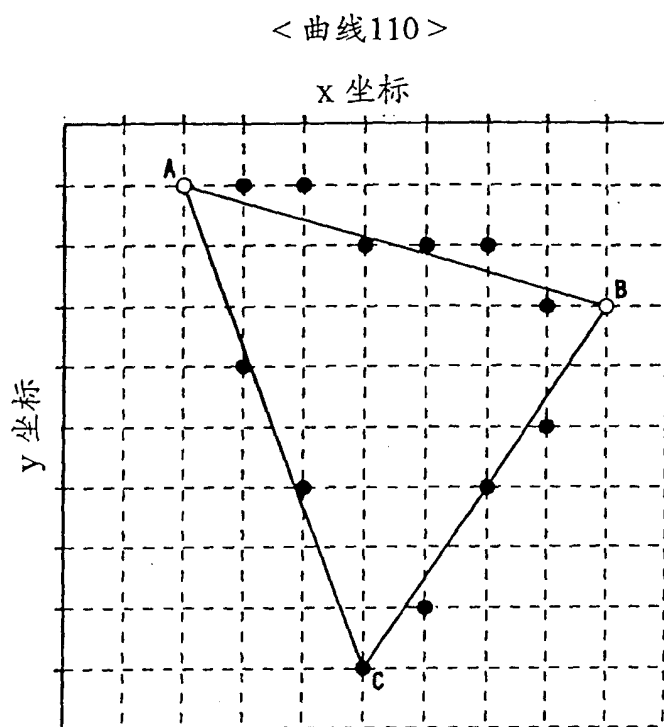
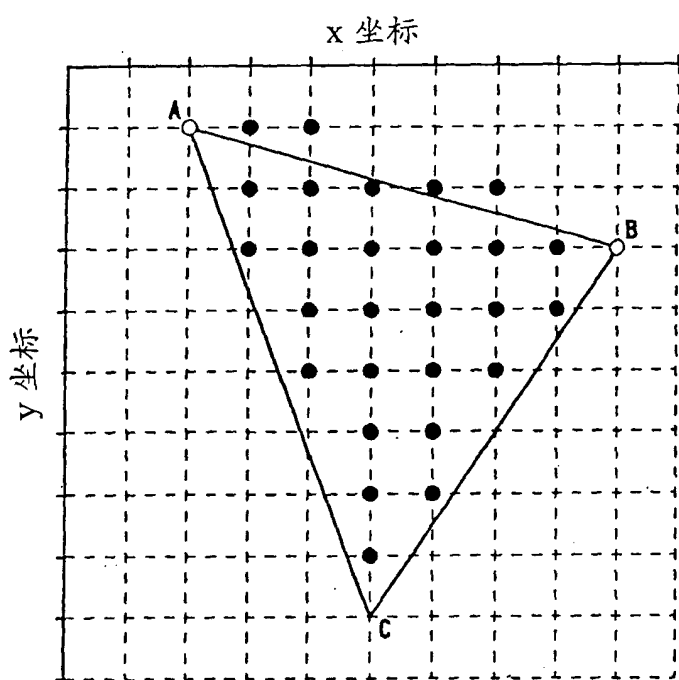


图 48B



< 明细表 9 >

```
for( x = 0 ; x < XSIZE ; x++ ){  
    for( y = 0 ; y < YSIZE-1 ; y++ ){  
        F[y+1][x] = F[y+1][x] ^ F[y][x];    // XOR  
    }  
}
```

图 49

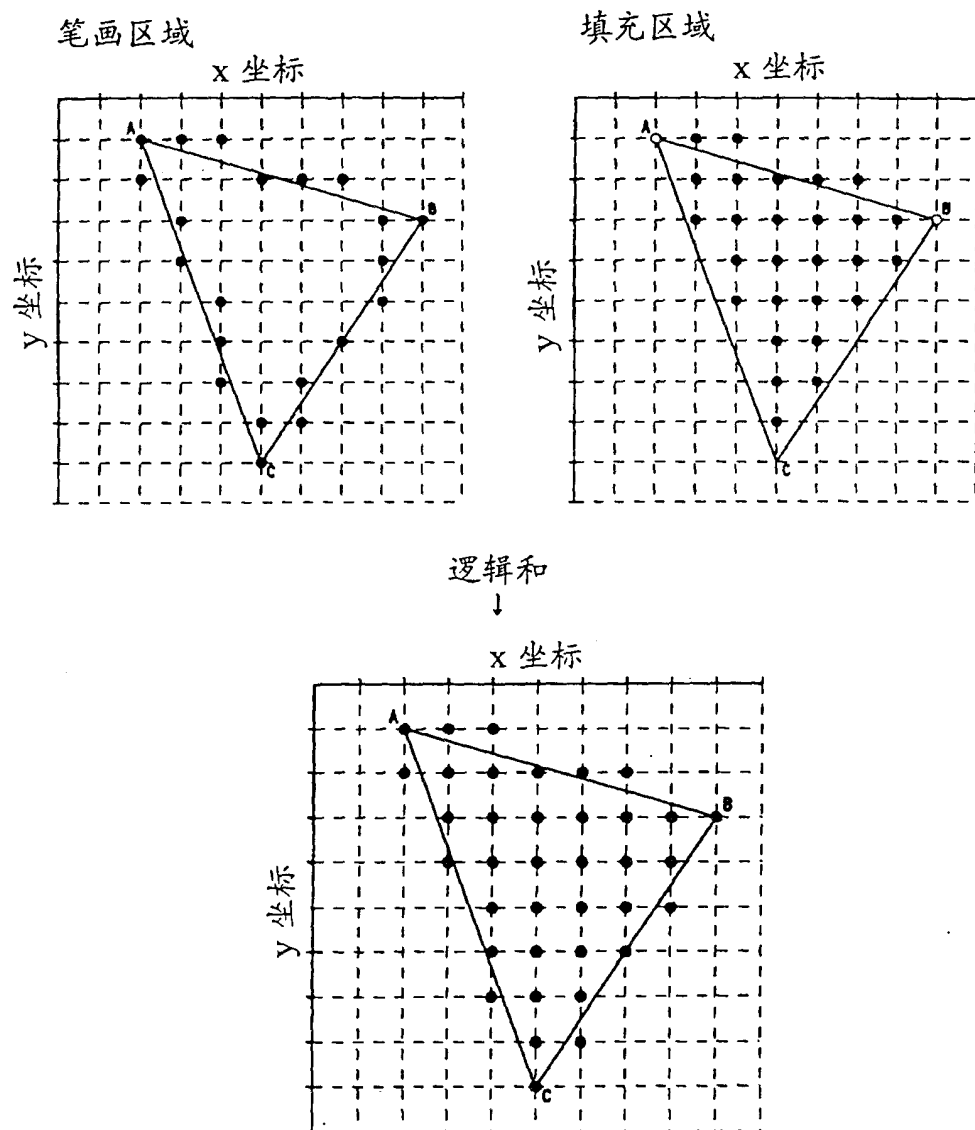


图 50

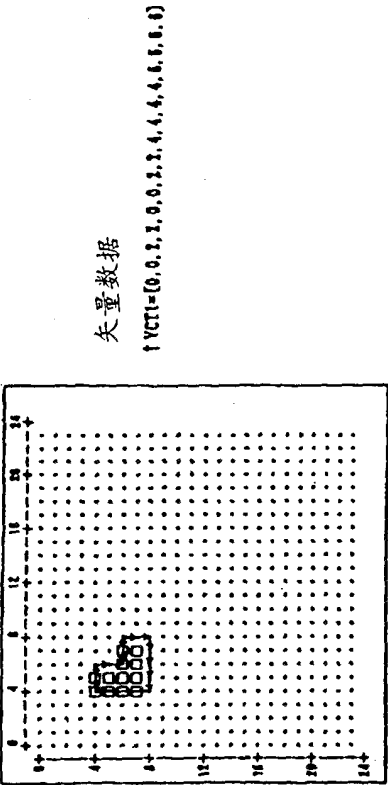


图 51A

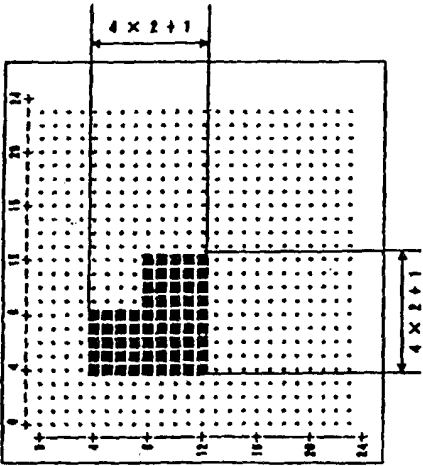


图 51B

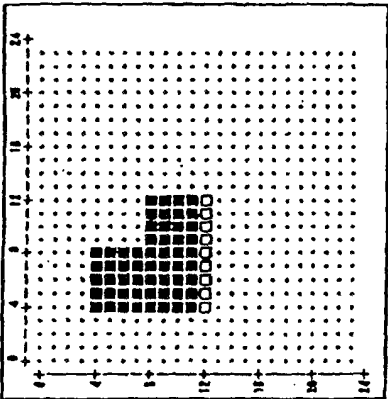


图 51C

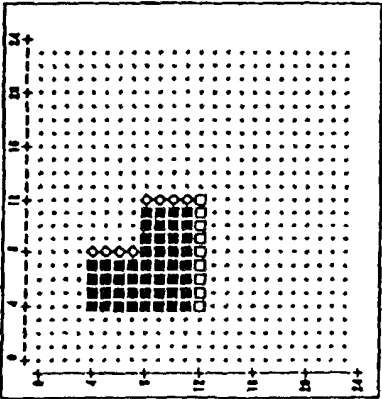


图 51D

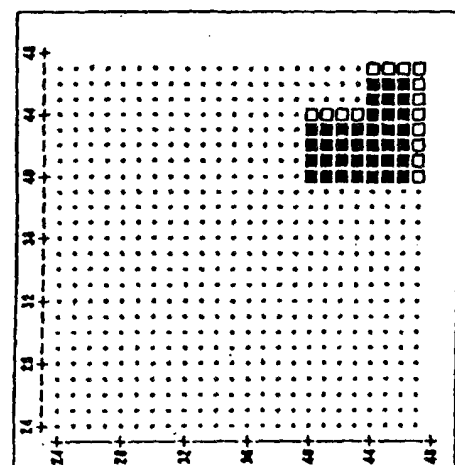


图 52C

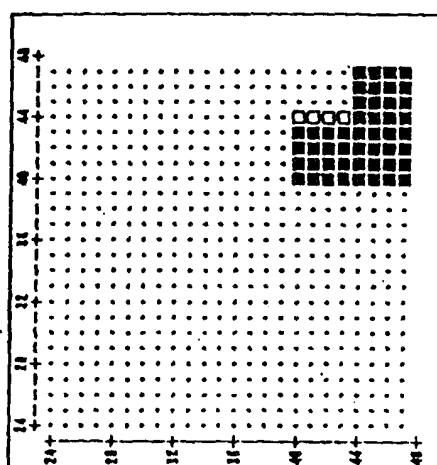


图 52D

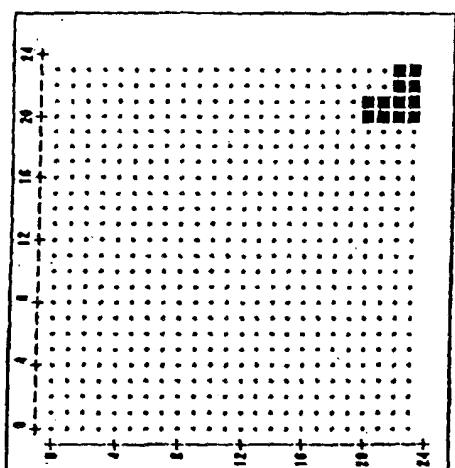


图 52A

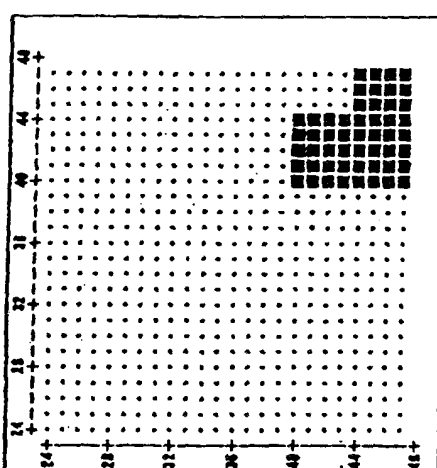


图 52B

图 53A

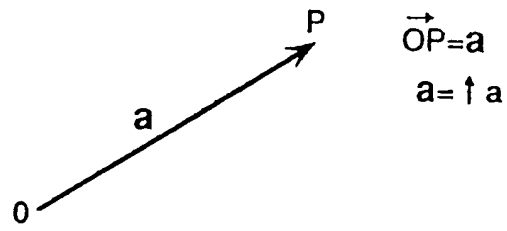


图 53B

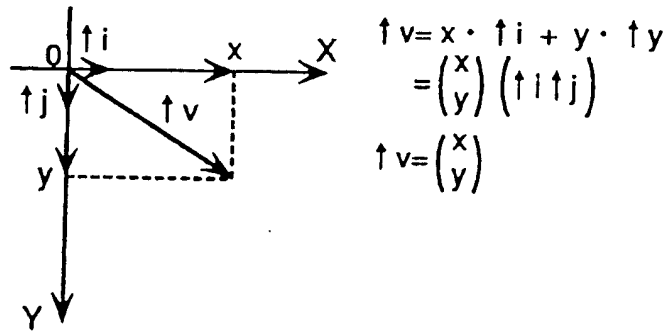
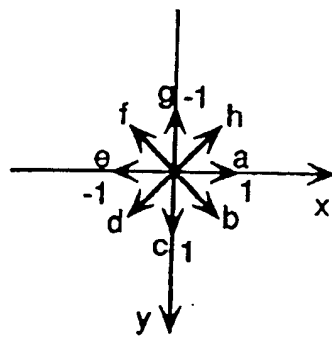


图 53C



$$\uparrow a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \uparrow b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \uparrow c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \uparrow d = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$\uparrow e = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \uparrow f = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \uparrow g = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \uparrow h = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\uparrow vct = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \dots, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

$$= \left\{ \uparrow a, \uparrow b, \uparrow c, \dots, \uparrow h \right\}$$