



[12] 发明专利说明书

[21] ZL 专利号 98805330.6

[45] 授权公告日 2003 年 10 月 1 日

[11] 授权公告号 CN 1122945C

[22] 申请日 1998.5.22 [21] 申请号 98805330.6

[30] 优先权

[32] 1997.5.22 [33] JP [31] 131831/1997

[86] 国际申请 PCT/JP98/02262 1998.5.22

[87] 国际公布 WO98/53425 日 1998.11.26

[85] 进入国家阶段日期 1999.11.22

[71] 专利权人 世雅企业股份有限公司

地址 日本东京

[72] 发明人 安井哲祐 森冈诚介 大久保淳

审查员 孙治国

[74] 专利代理机构 中原信达知识产权代理有限公司

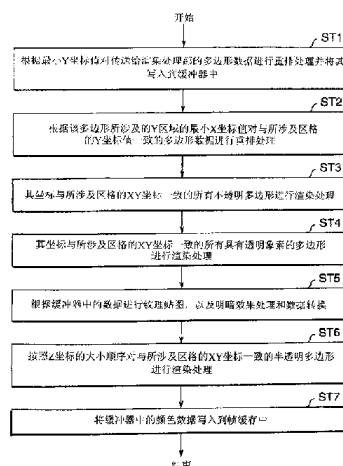
代理人 黄敏

权利要求书 3 页 说明书 18 页 附图 14 页

[54] 发明名称 图象处理器及图象处理方法

[57] 摘要

一种能够高效且快速地对图象元素数据进行处理,并且与常规技术相比能够以较低的成本来产生高质量图象的计算机图形系统。一种用于将屏幕划分成多个预定大小的区域(区格),并对所划分的每个区域(区格)上的数据分别进行处理的图象处理器,其中对图象构成元素上的信息沿相对于扫描线的垂直方向进行重排处理(ST1),和沿平行于扫描线的方向进行重排处理(ST2),并根据重排后的信息对不透明多边形、具有透明像素的多边形和半透明多边形依次进行图象处理(ST3 到 ST6)。



1. 一种图象处理器，将显示器划分成多个预定大小的区域并分别对所划分各区域进行处理，该图象处理器包括：

5 第一重排装置，沿垂直于扫描线的方向对一组图象构成元素上的信息根据最大值或最小值进行重排，该组图像构成元素至少构成了屏幕上描绘的一幅图象；

 第二重排装置，沿相对于扫描线的水平方向对图象构成元素上的信息根据最大值或最小值进行重排；以及

10 图象处理单元，从重排后的多个图象构成元素上的信息中取出与执行该处理的所在区域相对应的图象构成元素上的信息，并进行图象处理。

2. 如权利要求 1 所述的图象处理器，其特征在于：

15 第二重排装置在由第一重排装置所处理的区域的垂直坐标范围内进行重排处理。

3. 如权利要求 1 或 2 所述的图象处理器，其特征在于第一和第二重排装置通过将图象构成元素上的信息相互链接在一起的链接处理，来重排图象构成元素上的信息。

20

4. 如权利要求 3 所述的图象处理器，其特征在于在图象处理单元执行完处理之后，所述第一和第二重排装置确定在接下来所要进行图象处理的区域中是否还包括该图象处理单元所用到的图象构成元素上的信息，并执行链接更新处理，以从链接中去除掉被确定为不再被包含的图象构成元素上的信息。

25

5. 如权利要求 1 到 4 之一所述的图象处理器，其特征在于在对每个所划分区进行图象处理时，图象处理单元不仅将所要处理的目标分成不透明多边形、具有透明象素的多边形和半透明多边形，并且

30

还按照不透明多边形、具有透明像素的多边形和半透明多边形的顺序来进行处理。

5 6. 一种图象处理方法，其将屏幕划分为多个预定大小的区域，并对每个所划分的区域分别进行处理，该方法包括如下步骤：

 第一重排步骤，用于沿垂直于扫描线的方向对构成了屏幕上至少一幅图象的一组图象构成元素上的信息根据最大值或最小值进行重排；

10 第二重排步骤，用于沿相对于扫描线的水平方向对图象构成元素上的信息根据最大值或最小值进行重排；以及

 图象处理步骤，用于从重排后的图象构成元素上的信息中取出与执行该处理的所在区域相对应的图象构成元素上的相关信息，并执行图象处理。

15 7. 如权利要求 6 所述的方法，其特征在于：

 而在所述第二重排步骤中，则是在第一重排步骤中所处理区域的垂直坐标范围内进行重排处理。

20 8. 如权利要求 7 所述的方法，其特征在于在所述第一和第二重排步骤中将执行链接处理，以将重排后的图象构成元素上的信息链接在一起。

25 9. 如权利要求 8 所述的方法，其特征在于在所述第一和第二重排步骤中将执行链接更新处理，以使与图象构成元素相对应各区域中的多余区域变为无效状态。

30 10. 如权利要求 6 到 9 之一所述的方法，其特征在于在图象处理步骤中，是在将处理对象分成不透明多边形、具有透明像素的多边形和半透明多边形的情况下，并按照不透明多边形、具有透明像素的多边形和半透明多边形的顺序依次来对每个所划分区域来进行图象处理

的。

11. 一种图象处理器，其中将屏幕划分成多个预定大小的区域，且分别对各区域进行处理，其特征在于包括：

5 存储缓冲器，用于存储图象构成元素上的信息和链接信息，以在图象构成元素上各信息的存储位置上进行相关图象构成元素之间的链接处理；

 索引缓冲器，对于所划分的每个区域，存储该区域中所包含的图象构成元素上的信息中至少一个图象构成元素上的信息的存储位置；

10 以及

 图象处理单元，用于利用索引缓冲器从存储缓冲器中读出正在处理的区域中所包含的图象构成元素上的信息，并且还执行所需图象处理。

15 12. 如权利要求 11 所述的图象处理器，其特征在于链接信息包括：垂直链接信息，用于将垂直于扫描线的方向的各位置上的相关的图象构成元素相互链接在一起，以及水平链接信息，用于将相对于扫描线的水平方向各位置上的相关的图象构成元素相互链接在一起；以及

 所述索引缓冲器包含有：垂直位置索引缓冲器，其对于所划分的
20 每个区域，分别存储该区域所包含的垂直于扫描线方向上的图象构成元素中的至少一个图象构成元素上的信息的存储位置，以及水平位置索引缓冲器，用于对所划分的每个区域，分别存储该区域所包含的相对于扫描线的水平方向上的图象构成元素中至少一个图象构成元素上的信息的存储位置。

25

 13. 如权利要求 11 或 12 所述的图象处理器，其特征在于所述图象处理单元在执行完图象处理之后，将执行链接更新处理，以确定在接下来所要进行图象处理的区域中是否还包括该图象处理单元所用到的图象构成元素上的相关信息，并从链接中去除掉已确定不再被包含
30 的图象构成元素上的信息。

图象处理器及图象处理方法

5 本发明涉及一种图象处理器，具体涉及一种用于计算机图形的图象处理器和图象处理方法。

 在计算机图形领域，为了提高图象处理的性能，常用一种方法是将屏幕划分为许多微小矩形区域（下文中简称为“区格（fragment）”），
10 并以区格为单位逐区格地来对图象进行处理。

 接下来参照图 14 到 17 对此种常规方法进行说明。其中首先，(1)：利用区格框架来划分各多边形，并利用含有已知各多边形的区段来进行存储。

15 附图 14 中，15 个区格排列成一个 3×5 的矩阵，其上定义有多边形 A，B 和 C。就多边形 A 来说，如图 15 所示，其与 8 个区格相对应。根据其坐标 α ， β ， γ 等将这些区格存储到如图 16 所示的缓冲存储器中。其中图 15 所示的 α_1 和 α_2 被存储在缓冲存储器的存储区 α 中，而 β_1 和 β_2 则被存储在缓冲存储器的存储区 B 中， γ_1 和 γ_2 则被存储在缓冲存储器的存储区 R 中。其余坐标与此类似。

 (2)：接着，将对所有区格中由当前所涉及的多边形所覆盖的那些区格进行计算及着色。例如，如图 17 所示，首先将针对多边形 A（见图 17(b)）对区格 δ_1 进行着色，接着再针对多边形 B（见图 17(c)）对其进行着色，最后则针对多边形 C 进行着色（见图 17(d)）。重复执行这种操作，直到处理完最后一个多边形，或完全覆盖了该区格为止。
25 对所有区格进行上述处理。

30 上述步骤（1）和（2）即为常规图象元素数据处理的操作。

然而，此方法需要将所有多边形分别划分成其各自的目标区格（objective fragment），因此其计算量十分巨大，同时相应地需要较长的处理时间。

5

此外，随之还会带来多种缺陷。例如，其应配置一个容量较大的存储装置来存储所生成的关于所划分多边形的信息。为了使处理速度更快，要求存储操作应更加迅速，从而使存储装置的成本上升。

10

而且，当半透明多边形具有纹理数据时，则常规方法将无法对其进行处理。

15

本发明旨在解决上述问题，其目的是能够在计算机图形系统中高效地且快速地对图象元素数据进行处理，并以低于常规水平的成本获取高质量的图象。实现此目的的一种方法是将屏幕划分为多个微小矩形区域（区格）来进行处理。这样做的目的是使其能够高效地检索出所涉及区域中所包含的多边形。简而言之，本发明提供了一种能够在计算机图形系统中，以低于常规水平的成本来提供更高质量图象的系统和方法。

20

25

在根据本发明的图象处理器中，将屏幕划分为多个预定大小的区域且分别对所划分的各区域进行处理，该图象处理器包含有用于对垂直于扫描线的方向上的图象构成元素上的信息进行重排的第一重排装置，用于对相对于扫描线的水平方向上的图象构成元素上的信息进行重排的第二重排装置，以及用于根据上述图象构成元素上重排后的信息进行图象处理的图象处理单元。

30

在根据本发明的图象处理器中，第一重排装置根据与垂直方向有关的图象构成元素的最小值和最大值进行重排，而第二重排装置则根据由第一重排装置所处理区域的垂直坐标范围内，图象构成元素水平

方向上的最小值和最大值进行重排。

5 以最小值或最大值进行重排均可以。当以最大值作为索引来进行重排时，只是处理顺序的方向会发生改变（即“从左到右”变为“从右到左”，或“从上到下”变为“从下到上”），而没有其它不同之处，也就是说，对于最小值和最大值可以采用类似的处理方法。即使当处理顺序是从垂直到水平或从水平到垂直时，其也可以采用类似的处理。

10 在本发明的图象处理器中，第一和第二重排装置执行链接处理，以将重排后的图象构成元素相关信息的相互链接在一起。

15 在根据本发明的图象处理器中，第一和第二重排装置执行链接更新处理，以将与图象构成元素相对应区域中的多余图象构成元素变为无效状态。

20 在本发明的图象处理器中，在对每个所划分区域进行图象处理时，图象处理单元不仅将所要处理的对象分成不透明多边形、具有透明像素的多边形、以及半透明多边形，并且还将按照不透明多边形、具有透明像素的多边形和半透明多边形的顺序来依次进行处理。

25 根据本发明的图象处理方法可将屏幕划分为多个预定大小的区域，并可对所划分的每个区域分别进行处理，其具有第一重排步骤：对相对于扫描线垂直方向上的图象构成元素的信息进行重排；第二重排步骤：对相对于扫描线水平方向上的图象构成元素的信息进行重排；以及图象处理步骤：根据重排后的图象构成元素信息进行图象处理。

30 根据本发明的图象处理方法，在第一重排步骤中，是根据图象构成元素在垂直方向上的最小值来进行重排的，而在第二重排步骤中，

则是在第一重排步骤中所处理区域的垂直坐标的范围内，根据图象构成元素在水平方向上的最小值来进行重排的。

5 根据本发明的图象处理方法，在第一重排步骤中将执行链接处理，以将图象构成元素上重排后的信息相互链接在一起。

根据本发明的图象处理方法，在第一和第二重排步骤中，将执行链接更新处理，以将与图象构成元素相对应区域中的不需要的图象构成元素变为无效状态。

10

根据本发明的图象处理方法，在图象处理步骤中，在将所要处理的对象分组成不透明多边形、具有透明象素的多边形和半透明多边形的情况下，按照不透明多边形、具有透明象素的多边形和半透明多边形的顺序对所划分各区域分别进行图象处理。

15

图 1 所示为根据本发明实施例 1 的图象处理器的功能方框图；

图 2 所示为根据本发明实施例 1 的图象处理器的几何处理器的功能方框图；

20 图 3 所示为根据本发明实施例 1 的图象处理器的填充处理器的功能方框图；

图 4 所示为根据本发明实施例 1 的图象处理器的纹理处理器的功能方框图；

图 5 所示为根据本发明实施例 1 的图象处理器的阴影（shading）处理器的功能方框图；

25 图 6 所示为本发明实施例 1 的整体处理流程图；

图 7 所示为本发明实施例 1 中的 Y 索引缓冲器和 Y 排序缓冲器的相关处理的示意图；

图 8 所示为本发明实施例 1 中的 Y 索引缓冲器和 Y 排序缓冲器的相关处理的示意图；

30 图 9 所示为本发明实施例 1 中的 Y 索引缓冲器和 Y 排序缓冲器

的相关处理的示意图；

图 10 所示为本发明实施例 1 中的 Y 索引缓冲器和 Y 排序缓冲器相关处理的示意图；

5 图 11 所示为本发明实施例 1 中，对于每个多边形最先变为有效状态的区格的示意图；

图 12 所示为本发明实施例 1 中的链接更新处理的示意图；

图 13 所示为本发明实施例 1 中的链接更新处理的示意图；

图 14 所示为常规处理方法中区格和多边形 A，B 和 C 之间关系的示意图；

10 图 15 所示为常规处理方法中与多边形 A 相对应的区格的示意图。

图 16 所示为常规处理方法中缓冲存储器的存储内容的示意图；

图 17 所示为根据常规处理方法的着色处理的示意图。

实施本发明的优选实施例：

15 本发明的实施例 1：

接下来将对根据本发明的实施例 1 的装置和方法进行说明。

图 1 所示为根据本发明实施例 1 的功能方框图。其中，参考标号 1 表示一个 CPU（中央处理单元），用于对虚拟空间中的物体进行操作以获得关于其操作的信息，以及进行多种控制。参考标号 2 表示用于以高速进行诸如多边形坐标变换、剪裁处理、三维计算机图形中的透视变换和/或亮度计算等类型的几何变换（即，纹理压缩计算）的几何处理器。参考标号 2a 表示一个用于存储对几何处理器 2 正进行操作的一帧图象有效的多边形数据、材质数据以及光线数据的多边形/材质/光线缓冲器存储器（多边形/材质/光线缓冲器 RAM）。这些多边形构成虚拟空间中各种多面体。缓冲存储器 2a 中所存储数据的内容如下：

20

25

链接信息，坐标信息和关于多边形其它属性的信息，即包括 LINK X，LINK Y，X，Y，iz，Tx，Ty，Nx，Ny，SIGN N，Alpha（透明

30

度)，光线 ID 以及材质 ID。

5 材质信息包括“是否有深度”(Depth Enable)，深度函数(Depth function)，深度密度(Depth density)，有无纹理(Texture enable)，是否雾化(Fog enable)，是否半透明(translucency enable)，纹理类型(texture type)，纹理函数(texture function)，偏移(offset) x, y，尺寸(size) x, y，重复(repeat) x, y，镜向(mirror) x, y，颜色(color) id，高亮光(Shininess)，材质反射光斑(Material specular)，材质发射光(Material emission)，多边形颜色(Polygon color)，纹理模式(Texture mode)，混合模式(blend mode)；以及

10 光线信息，其包括光线位置(Light position)，光照方向(Light Direction)，光线类型(Light Type)，衰减截止点(Attenuation Cutoff)，光点曝光表(Spotexp)，光颜色(Light Color)，以及光线环境(Light Ambient)。

15 参考标号 3 表示负责进行“消隐面去除处理”的填充处理器。填充处理器 3 对存在于某区域之内的多边形进行着色，以逐像素地获取位于前景位置上的各多边形的相关信息。

20 参考标号 4 表示纹理处理器。纹理处理器 4 将纹理贴(map)到某区域中的各像素上。纹理贴图是为了产生图象而将因物体形状而异的不同纹理贴到有确定形状的物体表面上的处理。参考标号 4a 表示其中保存有纹理处理器 4 所进行的处理中用到的各种纹理贴图的纹理存储器(纹理 RAM)。

25 参考标号 5 所示为阴影处理器。阴影是一种在综合考虑多边形的法向矢量、光源位置和颜色、视点位置、视线方向以及其它因素的情况下，用于由多个多边形组成的物体上表现阴影或诸如此类效果的技术。阴影处理器 5 在进行处理时需计算某区域内各像素的亮度。参考

30 标号 5a 所示为其中存储有一屏图象数据的帧缓存。从帧缓存 5a 中顺

序地读出数据，随后在将其提供给诸如 CRT、液晶显示器、等离子显示装置等类型的显示器之前（图中未示出）先将其从数字数据转化为模拟信号。

5 参考标号 6 表示存储有多条给图象处理器的命令（诸如多边形数据库和显示列表）以及到 CPU 1 的多种程序的工作程序/多边形缓冲存储器（工作程序/多边形缓冲 RAM）。缓冲存储器 6 同时还用作 CPU 1 的工作存储器。

10 填充处理器 3，纹理处理器 4 以及阴影处理器 5 用于进行被称为渲染(rendering)的处理。在各区域中，从其屏幕的左上角位置开始顺序地进行渲染处理。实际上，利用几何原理可以将物体定位在虚拟空间坐标系中，并将其透视变换到屏幕上。利用渲染处理，可以在定义于屏幕坐标系中的数据上产生相应的图象。循环进行渲染处理，其次
15 数与总区域数相同。

接下来将参照图 2 到 5 中所示的功能方框图，对根据本发明实施例 1 的图象处理器进行说明。

20 图 2 所示为几何处理器 2 的功能方框图。该图中，参考标号 21 表示一个不仅要从缓冲存储器 6 中读出命令，而且还要对其进行分析，以及据分析结果对矢量引擎(engine)22 和剪裁（clipping）引擎 24 进行控制，并向排序引擎 27 提供处理所得数据的数据分配器。

25 参考标号 22 所示为用于执行矢量计算的矢量引擎。由该引擎所处理的矢量被存储在矢量寄存器 23 中。

参考标号 23 表示一个用于保存矢量引擎 22 所要计算的矢量数据的矢量寄存器。

30

参考标号 24 表示一个用于进行剪裁处理的剪裁引擎。

参考标号 25 表示一个保存有由排序引擎 27 所执行的 Y 排序处理中所用到的 Y 索引的 Y 排序索引 (Y-sort INDEX)。

5

参考标号 26 表示一个保存有由排序引擎 27 所执行的 X 排序处理中所用到的 X 索引的 X 排序索引 (X-sort INDEX)。

10 参考标号 27 所示为一个用于执行 X 排序和 Y 排序、以从缓冲器 6 中检索出落在相关区格内的多边形的排序引擎。所检索出的多边形不仅被存储到缓冲存储器 2a 中，同时还被发送给填充处理器 3 以进行渲染处理。排序引擎 27 另外还负责对多边形 TAG 28 和多边形高速缓存 34 进行控制。

15 参考标号 28 表示一个用于保存多边形高速缓存 34 的 TAG 的多边形 TAG。

20 图 3 所示为填充处理器 3 的功能方框图。该图中，参考标号 31 表示用于对下文中所要说明的材质信息高速缓存 42，45，51b，52a 和 53a 以及光线高速缓存 51a 进行控制的高速缓存控制器。

参考标号 32 表示用于保存下文中所要说明的材质信息高速缓存 42，45，51b，52a 和 53a 以及光线高速缓存 51a 的 TAG 的材质 TAG。

25 参考标号 33 所示为光线 TAG，其是一个保存有下文中所要说明的光线高速缓存 51a 的 TAG 的缓冲器。

参考标号 34 表示一个用作多边形数据的高速缓冲存储器的多边形高速缓存。

30

参考标号 35 所示为用于求出 DDA 初值的初始参数计算器。

5 参考标号 36 所示为一个 Z 值比较器阵列，其负责在多个多边形之间进行 Z 值比较，以进行“消隐面去除”处理，以及对多边形 ID 和值为 t_0 , t_1 和 t_2 的分割比 (division ratio) 进行处理。Z 值比较器阵列 36 由 64 (8×8) 个 Z 值比较器构成。这些比较器以并行方式进行操作，因此其能够同时对 64 像素进行处理。一个 Z 值比较器保存一个多边形的数据。这些数据可以包括，例如多边形 ID, iz , t_0 , t_1 , t_2 , 窗口, 样板 (stencil) 和阴影。

10

参考标号 37 是一个用于保存关于各多边形各顶点的数据的顶点参数缓冲器。与 Z 值比较器阵列 36 相对应，该缓冲器能够存储 64 个多边形的数据。

15

参考标号 38 是一个插值器，用于通过根据由 Z 值比较器阵列 36 计算所得的结果 t_0 , t_1 , t_2 和 iz 、以及顶点参数缓冲器 37 的存储内容进行插值，来计算像素参数。

20

图 4 所示为纹理处理器 4 的功能方框图。该图中，参考标号 41 表示一个用于计算雾化和深度排队 (queuing) 所需的混合率的密度计算器。

参考标号 42 所示为一个其中存储有与深度信息有关的数据的材质信息高速缓存。

25

参考标号 43 表示用于保存各窗口的相关信息的窗口寄存器。这些信息是以，例如 kz , cz , 雾化函数 (fog function), 雾化密度 (fog density), 和雾化结束点 (fog end) z 的形式来表示的。

30

参考标号 44 表示一个用于利用坐标 T_x , T_y 和 LOD 技术来计算

纹理贴图的地址的地址发生器。

参考标号 45 所示为其中保存有与材质有关的数据的材质信息高速缓存。

5

参考标号 46 表示一个用于执行三维插值的常用方法 — “三线性 MIP 贴图插值处理”的 TLMMI 计算器 (TLMMI: 三线性 MIP 贴图插值处理)。“MIP 贴图处理”是一种用于进行边缘平滑,即消除纹理在贴图过程中所产生的锯齿的技术。此技术原理如下。理论上,一个物体投影到一个像素上的表面的颜色(亮度)应该是其相应各映射区上的颜色的均值。如果不是,则锯齿现象将十分明显,从而使纹理的品质变差。另一方面,为求得每个均值所需的处理将使计算量极大地增加,其结果是处理将需要更长的时间,或者是需要速度更快的处理方法。而 MIP 贴图便可以解决此问题。根据 MIP 贴图技术,为了简化与单个像素相关的映射区颜色(强度)的采集操作,将事先准备一组其宽度为 2 的整次幂的贴图数据。与一个像素相关的所有贴图区的尺寸将变为,存在于任何通过将其乘以 2 的整次幂所产生的两个数据之间的值(例如,如果所需贴图的大小为 192×192 ,则可以用 2 大小分别为 256×256 和 128×128 的贴图进行内插计算,求得最合适的贴图)。通过对这两个数据进行比较,可以得到对应贴图区域的颜色。例如,如果有一个单倍(one-time)屏幕 A 和半倍屏幕 B,则将能够得到与 1/1.5 倍屏幕 C 相对应的屏幕 A 和 B 的每个像素。本例中,屏幕 C 各像素的颜色等于两个屏幕 A 和 B 的像素之间的中间颜色。

25 参考标号 47 表示一个用于在 4 位纹理单元(texel)处理中变换颜色的颜色变换器。

参考标号 48 所示为其中颜色信息是在 4 位 texel 处理中进行保存的颜色调色板。调色板 48 被用来存储用于“图形绘制”的各种颜色。根据调色板 48 的内容,可以确定每个像素可用的颜色。

30

图 5 所示为阴影处理器 5 的功能方框图。该图中，参考标号 51 表示一个用于计算已进行完纹理贴图处理的各多边形的亮度的图象处理器。

5

参考标号 51a 表示一个用于存储与光线有关的信息的光线信息高速缓存。

参考标号 51b 所示为用于存储与材质有关的信息的材质信息高速缓存。此信息包括高亮光(Shinies)，材质反射光斑(Material specular)，以及材质发射光(material emission)等信息。

10

参考标号 51c 表示一个用于保存与窗口有关的信息的窗口寄存器。这些信息由屏幕中心(Screen center)，焦点(Focus)，场景环境(Scene ambient)以及其它数据组成。

15

参考标号 52 表示用于对多边形颜色和纹理颜色进行关联，以及亮度调制和雾化处理的调制处理器。

20

参考标号 52a 表示一个其中存储有与材质有关的信息的材质信息高速缓存。此信息包括多边形颜色和纹理模式。

参考标号 52b 表示一个用于保存与窗口有关的信息的窗口寄存器。此信息包括雾化颜色。

25

参考标号 53 表示用于将存储在颜色缓冲器 54 中的数据进行混合并将其结果写入到颜色缓冲器 54 中的混合处理器。混合处理器 53 根据混合率寄存器的值对当前像素的颜色与帧缓存的像素颜色进行混合，随后将混合结果写入到由存储体写入(write bank)寄存器所表示的存储体帧缓存中。

30

参考标号 53a 所示为用于存储与材质有关的信息的材质信息高速缓存。此信息包括“混合模式”。

5 参考标号 54 所示为一个大小与区格相同的 8×8 的双存储体 (double-bank) 结构的颜色缓冲器。

参考标号 55 表示一个用于将颜色缓冲器 54 中的数据写入到帧缓存 5a 中的绘图处理器。

10

参考标号 56 所示为用于执行位图处理的位图处理器。

参考标号 57 表示用于从帧缓存 5a 中读出数据,并将其提供给 DAC (数字-模拟转换器), 以将其显示于显示器上 (未示出) 的显示控制
15 器。

同时在本发明的实施例 1 中,不再是将多边形划分成多个区格,而是同时在相对于扫描线的垂直方向 (下文中简称为 Y 方向) 和水平方向 (下文中简称为 X 方向) 对多边形进行重排,从而能够检索出每个区格中所包含的所有多边形。利用所检索出的信息,便能够针对当前所涉及的各区格来对多边形进行处理,由此可以在不进行上述多边形划分操作的情况下来进行处理。因此,处理时间将大大缩短,同时
20 其对处理装置中存储装置容量的要求也将大为降低。

25 在渲染处理的过程中,将把多边形分成不透明多边形、半透明多边形、以及具有透明像素的多边形几类分别进行处理,同时到最后步骤才对半透明多边形进行处理,由此使得半透明多边形也能够得到处理。

30 接下来将参照图 6 对根据本发明实施例 1 的处理方法的流程进行

说明。

先按照各多边形的最小 Y 坐标值对所要传送给渲染处理部的多边形数据进行重排，而后将其写入到缓冲器中（ST1）。

5

再按照各多边形所涉及的 Y 区的最小 X 坐标值对与当前所涉及区格的 Y 坐标值一致的多边形数据进行重排（ST2）。

10 对所有其坐标与当前所涉及区格的 X 和 Y 坐标一致的不透明多边形进行渲染处理（ST3）。

对所有其坐标与当前所涉及区格的 X 和 Y 坐标一致的透明多边形进行渲染处理（ST4）。

15 根据缓冲器中的数据，先进行纹理着色处理，随后再通过将其转换为颜色数据来完成阴影处理（ST5）。

步骤 ST4 和 ST5 的执行顺序可以相反。实际上，更多情况下是按照先步骤 ST5 再 ST4 的顺序来执行的。

20

随后，按照 Z 坐标值的大小顺序对与当前所涉及区格的 X 和 Y 坐标一致的半透明多边形进行渲染处理（ST6）。

而后将缓冲器中的颜色数据写入到帧缓存中（ST7）。

25

如图 6 所示，多边形数据首先按照 Y 进行重排，随后在继续对每条显示线进行处理之前再按照区域 Y 中 X 的最小值对重排所得的上述数据进行重排。其结果是，可以找出某多边形所应最先被处理的区格的位置。因此，其首先将找出是哪个区格包含有该多边形。此外，在对第一区格所包含的相关多边形进行完处理之后，将对各多边形进行

30

链接处理（详见下文），通过其依次地清除掉对于所处理区格已不再需要的多边形，由此在下一区格中所包含的多边形相关信息中将只包括必要的多边形。因此，通过追溯（track back）迄今为止所作出的所有链接，可以使所有多边形落在所正检索的每个区格中。在不需要为产生新多边形以及对其进行处理而将多边形划分成区格的情况下，可以同时读取并处理各区格中各多边形的相关信息。

链接更新处理是一种用于在对所涉及的区格进行完多边形处理之后，检查在接下来所涉及的区格中是否还需要再对多边形进行处理（即是否变为无效状态），并当所检查的多边形变为无效状态时将其从链接中去除的技术。

接下来将对用于渲染的处理单元进行说明。

（1）：首先将多边形分成不透明多边形、具有透明象素的多边形和半透明多边形三种类型。

（2）：接着先对不透明多边形进行处理。其中将独立地重写纹理数据，从而产生最终所余下的有效象素。随后则只需快速地进行 Z 值比较处理。

（3）：在进行完上述处理（1）和（2）时，区格缓冲器中的所有数据均将变为有效状态。此时将根据帧缓存中的数据读出纹理数据，随后再对其颜色数据进行转换。

（4）：而后将对具有透明象素的多边形进行渲染。为了实现此步骤，其需要根据其纹理单元（texel）是否透明来确定所要描绘的象素。因此，在渲染过程中，着色是在读取关于透明纹理单元的信息的同时执行的。其它步骤与不透明多边形的处理步骤相同。因为其很难在渲染过程中读取关于透明纹理的信息，所以优选情况下将采用先步骤（4）后步骤（3）的处理顺序。

（5）：随后将对半透明多边形进行处理，其中经过 Z 值比较处理的所有象素的数据均为有效状态。因而，其将读出每个象素的纹理

数据，并通过到颜色数据的变换（混合）来进行着色处理。由于多边形描绘的顺序对此种混合处理有很大的影响，因此需要事先以任何适宜方式来进行 Z 值排序处理。

5 接下来将参照图 7 到图 13 对根据本发明实施例 1 的区格 CG 系统进行详细说明。

10 首先将先对沿 Y 方向的重排处理进行说明。图 7 所示为 Y 索引缓冲器和 Y 排序缓冲器的结构示意图。该种 Y 索引缓冲器中存储了含有区格行（其中分别写入有 Y 坐标最小值 Ymin）的多边形列表链接的顶端地址。而 Y 排序缓冲器中则按照数据输入的顺序存储了多边形数据。同时将属于同一扫描线的邻接各多边形的地址存储为 Y 排序缓冲器中的 LINK Y 参数。

15 因此，如果想要知道属于某特定行的多边形时，其完全可以追溯出从相应 Y 索引地址到 LINK Y 结束点的多边形链接。

20 接下来将对图 7 所示的示例进行说明。当 Y 索引缓冲器中最上端区格行的地址为 0，即其内容为“空”(EMPTY)时，其表示没有多边形。第二行同样如此。但 Y 排序缓冲器中显示用于表示关于第三行的信息的地址 2 的内容为“ADR8”以及“空”。因此，访问排序缓冲器的 ADR8 将得到内容为“ADR5”的 LINK Y。因而，下一次将对 ADR5 进行访问。此步骤之后，将依次对 ADR3 和 ADR1 进行类似的访问。由于 ADR1 的 LINK Y 内容为“END”，所以链接就此中止。对所有
25 区格行均将以上述方式进行相同的处理。

下面对多边形数据的存储步骤进行说明。

30 假设在图 7 所示的状态下，输入的是满足其 Ymin 的 Y INDEX = 5 的多边形数据。由于 Y 索引缓冲器中第 5 索引的内容为“EMPTY”，

所以该多边形数据的 LINK Y 将变为“END”。随后将此多边形数据存储到 Y 排序缓冲器的 ADR10 中；另一方面，将刚存储的多边形数据的地址“ADR10”写入到 Y 索引缓冲器的第 5 行中。更新后的状态如图 8 所示。

5

下面将对在图 8 所示的状态下,输入的是满足 Ymin 的 Y INDEX=2 的多边形数据的情况进行说明, 由于在 Y 索引缓冲器的第二行中存储了某多边形数据的地址(图 8 中的“ADR8”), 因此从现在起每写入一个多边形便要存入相应多边形数据。因此, 将该多边形数据的 LINK Y 设置为其原始 Y 索引值“ADR8”; 另一方面, 将所述存储的多边形数据的地址“ADR11”写入到 Y 索引缓冲器的第二行中。更新后的状态如图 9 所示。

10

图 10 所示为沿 X 方向进行的重排处理的示意图。X 方向的处理步骤与 Y 方向的步骤相似, 其中先求出 Xmin, 随后再利用所求出的 Xmin 来进行重排处理。

15

如上所述, 通过分别在 X 和 Y 方向上进行重排处理, 将能够找出多边形最先变为有效状态时所在区格的位置(即, 使区格最先为有效的多边形)。例如, 如图 11 所示, 在各多边形的最小 Y 坐标值中选出其 X 坐标值最小的区格。

20

此外, 关于重排结果, 从 0 到相关显示线(行)的所有显示线均被视为有效状态, 因此, 如果要显示某个多边形, 则使该多边形生效的区域将如图 12 所示, 因而该多边形的有效区域将包括不必要的区格。因此, 在此情况下将需要以如下方式来更新链接。

25

首先, 在对该多边形进行完处理之后, 将把那些在后面所处理的区格中处于无效状态的多边形从此链接上去除掉。例如, 在图 13 (a) 所示的示例中, 渲染处理后将 X LINK 上去除掉区格 F1 到 F7。同

30

时在渲染处理之后还将从 Y LINK 上去除掉区格 F7。通过执行链接更新处理所得的多边形有效区如图 11 (b) 所例示。

接下来将参照图 13 对 X 方向链接处理的步骤进行说明。

5

(1)：在多边形边缘与沿 X 方向延伸的各区格的水平边界之间所形成的各交集中，将对其中 X 坐标值稍大 (P2) 的一个交集进行检查。

10

(2)：将上述步骤 (1) 中所检查的交集的 X 坐标值，分别和从与当前所涉及的显示线中的区格 X 坐标轴垂直的边界中选出的其 X 坐标值较大的边界 (边界 1 到 3) 的 X 坐标值进行比较。

(3)：如果比较结果是“小 (边界 1 和 2)”，则将对该多边形进行链接处理，使其对下一区格也有效。

15

(4)：如果比较结果为“大于”，则在渲染处理之后，将把该多边形从链接上去除掉，使得在下一区格 (边界 3) 的处理中不再对其进行处理。

对于 Y 方向也将使用相同的步骤来进行 Y 方向链接处理。

20

如上所述，在本发明的实施例 1 中，其不再是逐区格地对多边形进行划分，而是同时在相对于扫描线的垂直和水平方向上对多边形进行重排处理，由此而能够检索出各区格中所包含的所有多边形。由于其利用了所检索出的信息，同时是针对当前所涉及的每个区格来划分多边形，所以其能够读出及处理每个区格中所包含的多边形，同时却不会产生新的多边形。因此，与常规技术不同，其不再需要针对所要处理 (to be objective) 的每个区格来划分所有多边形，从而使原先的巨大计算量大为减少，同时其也不再需要配置大容量的存储装置来存储由划分处理所产生的多边形的相关信息。其结果是，处理时间将大大缩短，而同时系统只需配置很小容量的存储装置。换句话说，其可以同时实现低成本和高性能的系统。

30

另外，将半透明特性多边形的处理分离为分别对不透明多边形，具有透明像素的多边形和半透明多边形进行处理，且同时到最后才处理半透明多边形的处理方法对系统性能的提高也有很大贡献。

5

根据本发明，在其中将屏幕划分成多个预定大小的区域，且同时以区域为单位逐区域地进行处理的图象处理器中，由于不仅在相对于扫描线的垂直方向上，而且同时在相对于扫描线的水平方向上均对图象构成元素的相关信息进行了重排处理，进而根据重排后的图象构成元素相关信息来进行图象处理，所以其能够检索出每个区域中的所有多边形。因此，利用所检索出的信息对相关各区域分别来划分多边形，能够读出并处理绘图区域中所包含的所有多边形，而同时却不会产生新的多边形。其结果是，处理时间的长度将大为缩短，而存储装置所需的容量也将大为降低。

10

因此，根据本发明，在对每个所划分区域进行图象处理时，将把所要处理的对象分成不透明多边形、具有透明像素的多边形和半透明多边形三种类型，随后再以不透明多边形，具有透明像素的多边形和半透明多边形的顺序依次进行处理，从而即使当多边形具有纹理数据时，其也能够对半透明多边形进行处理。

15

20

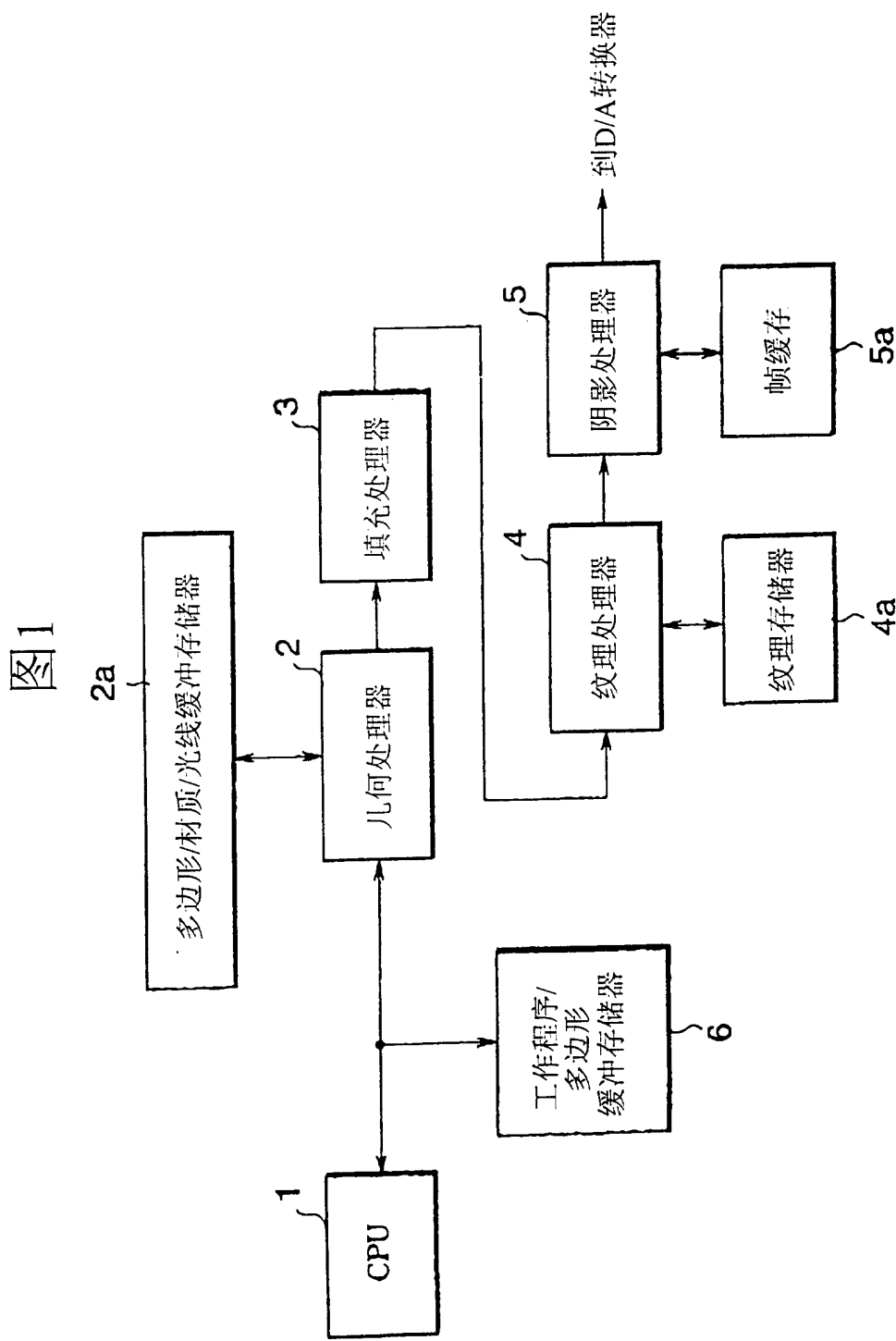
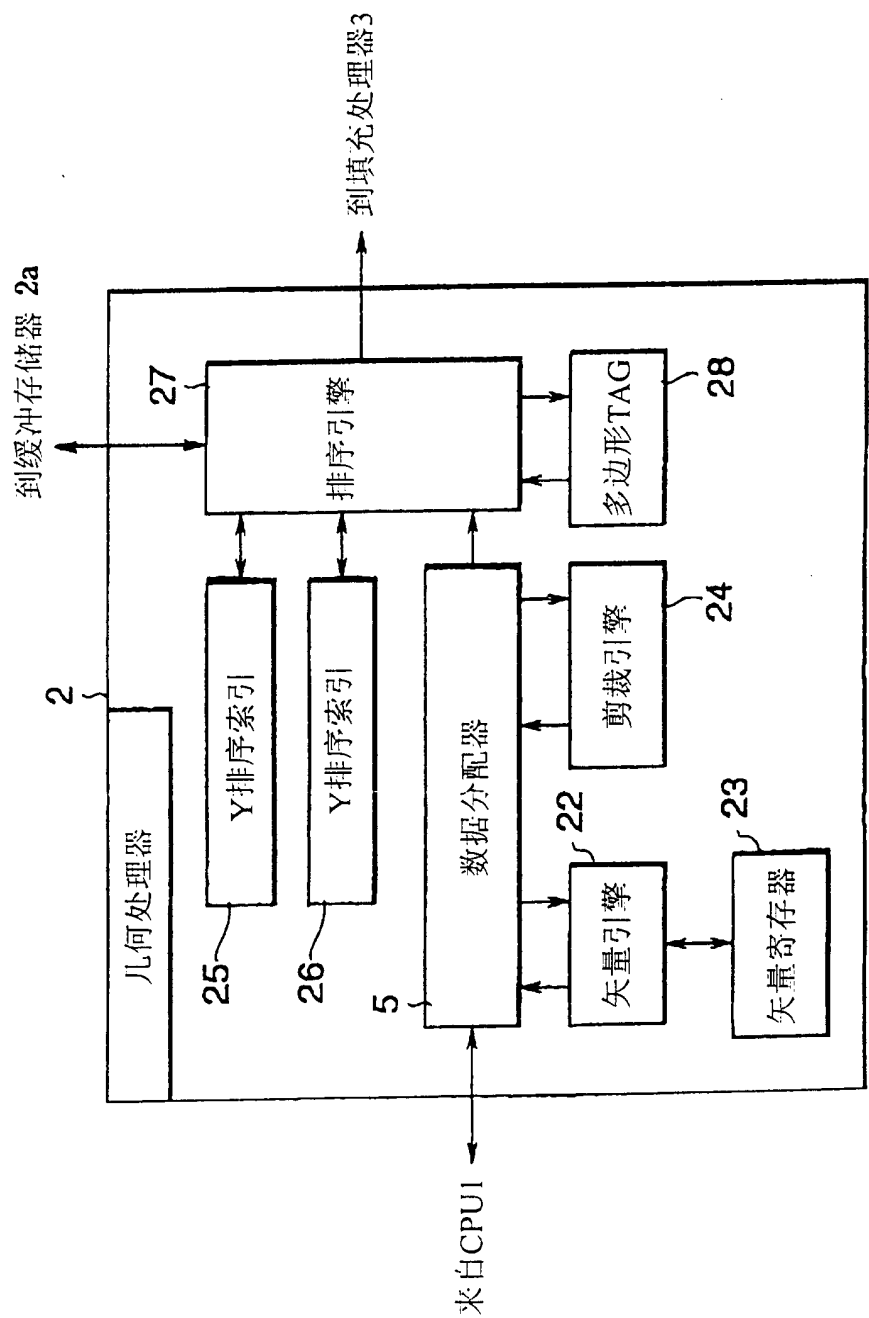


图2



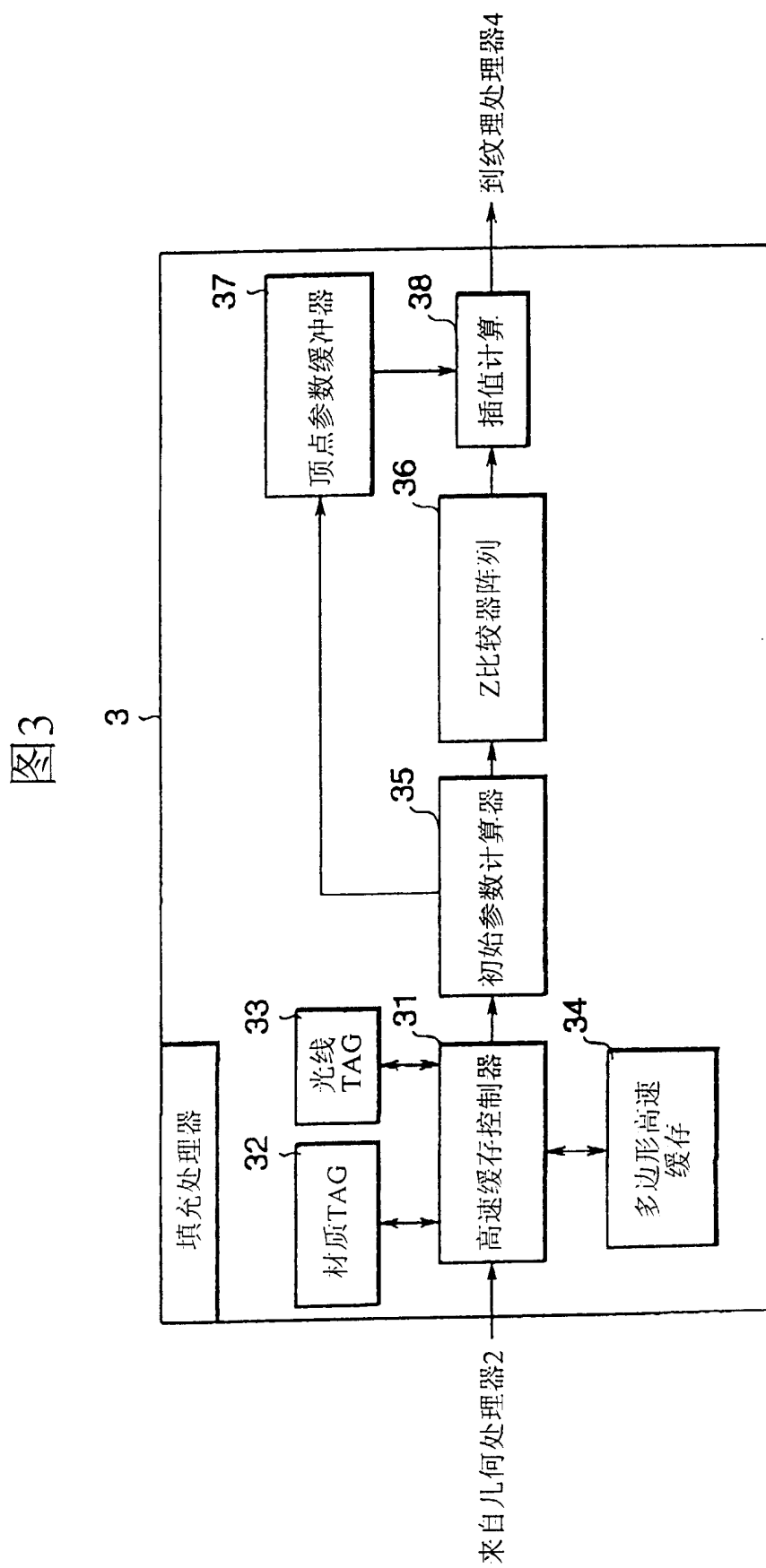
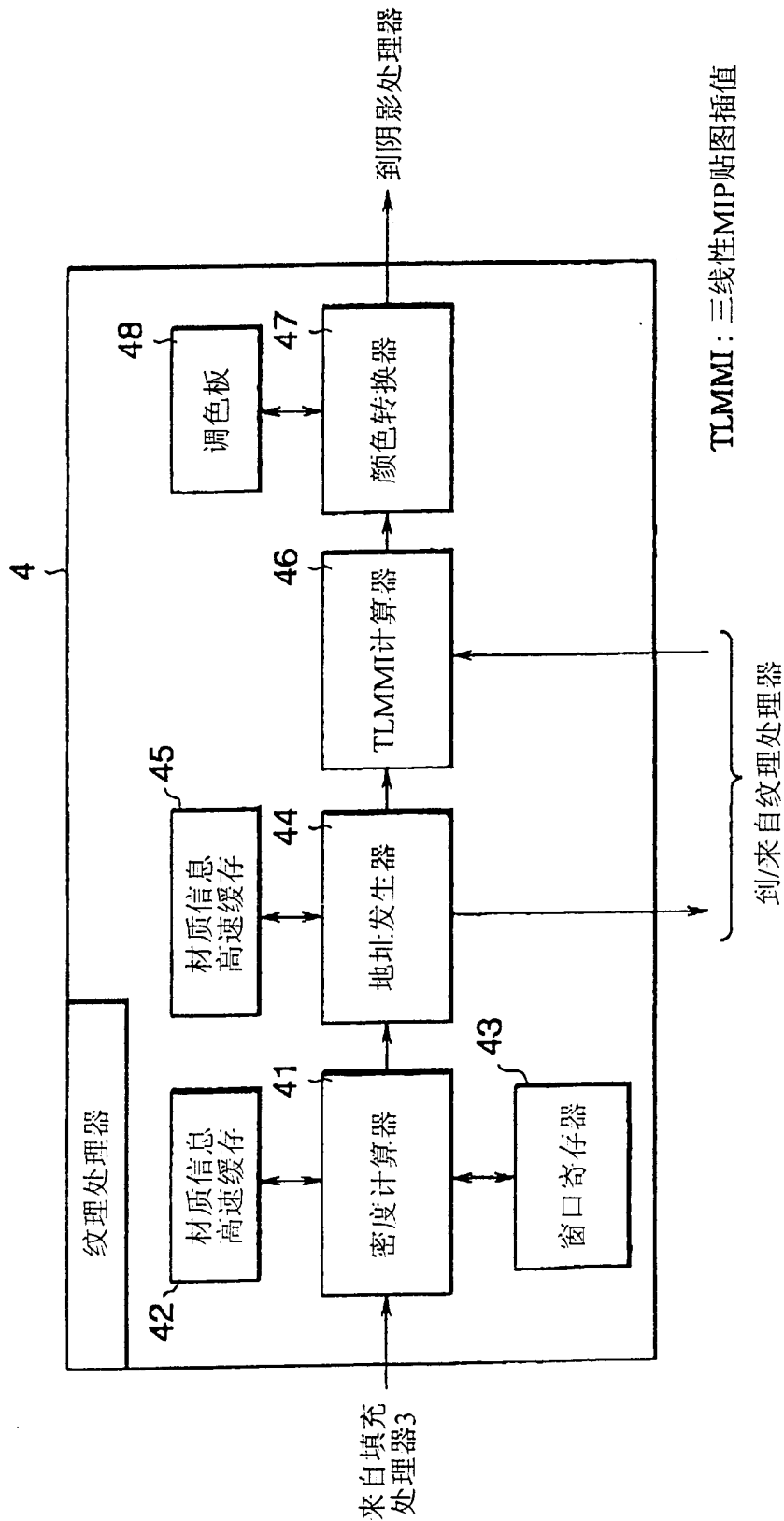


图4



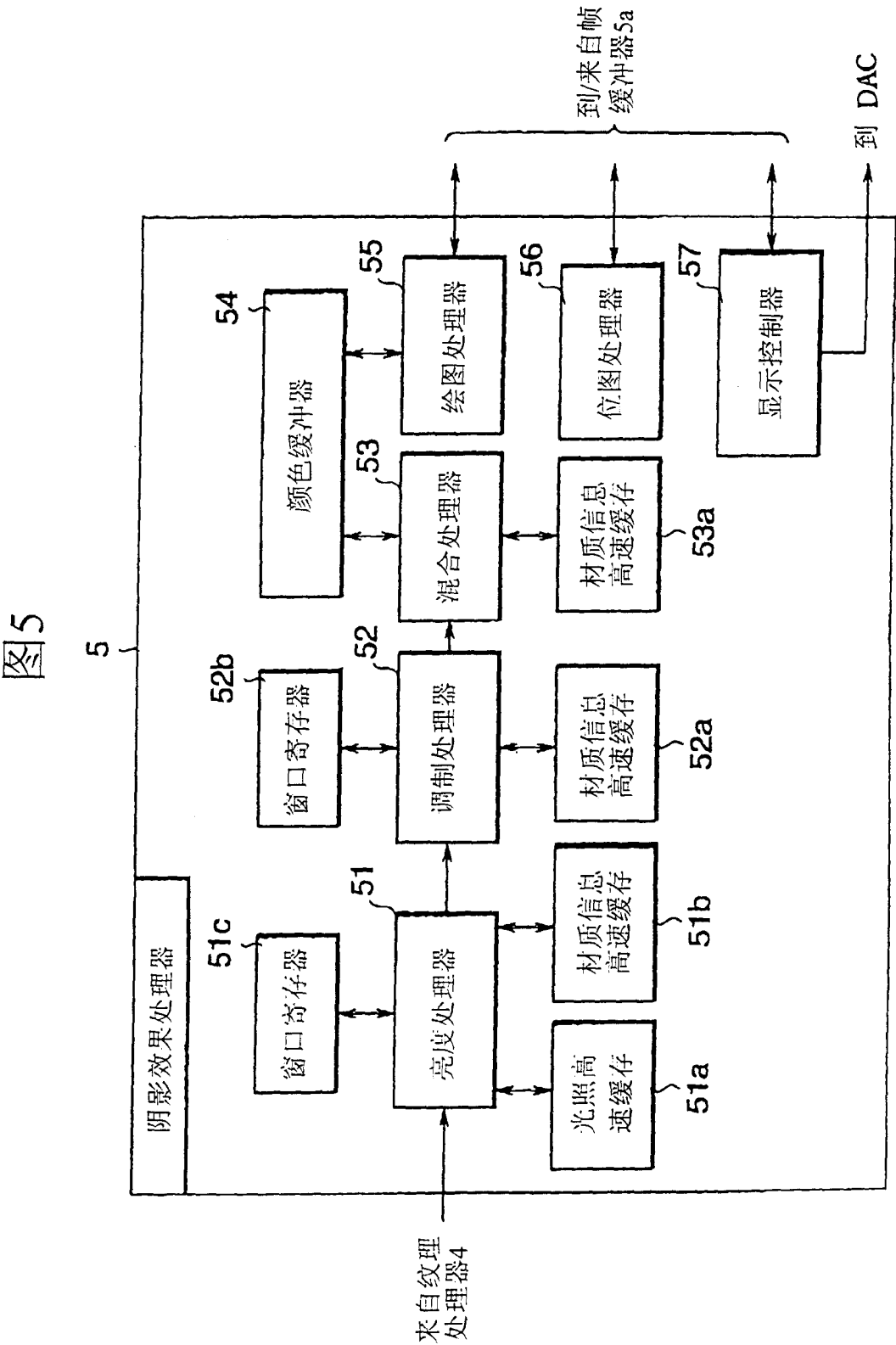


图6

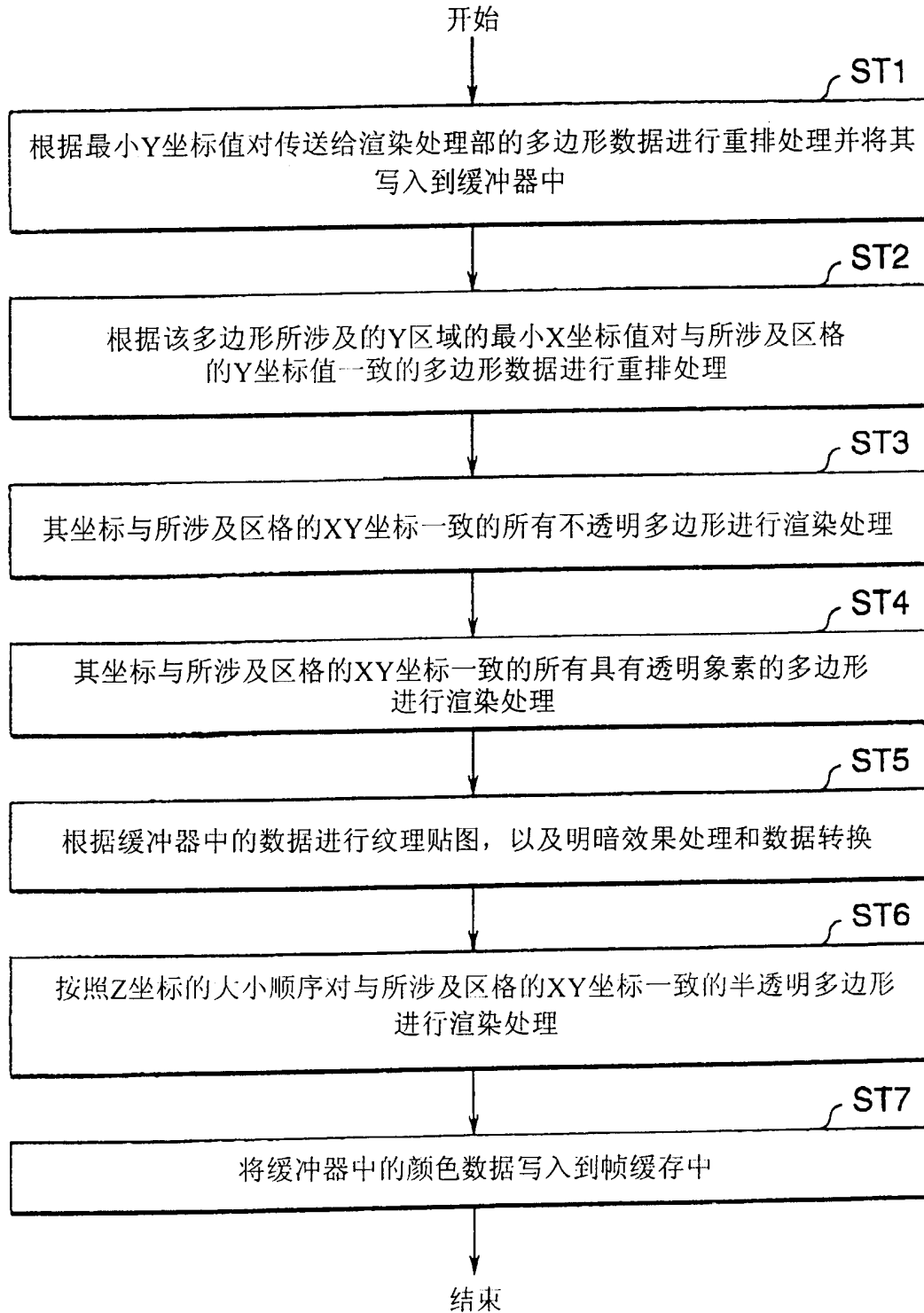


图7

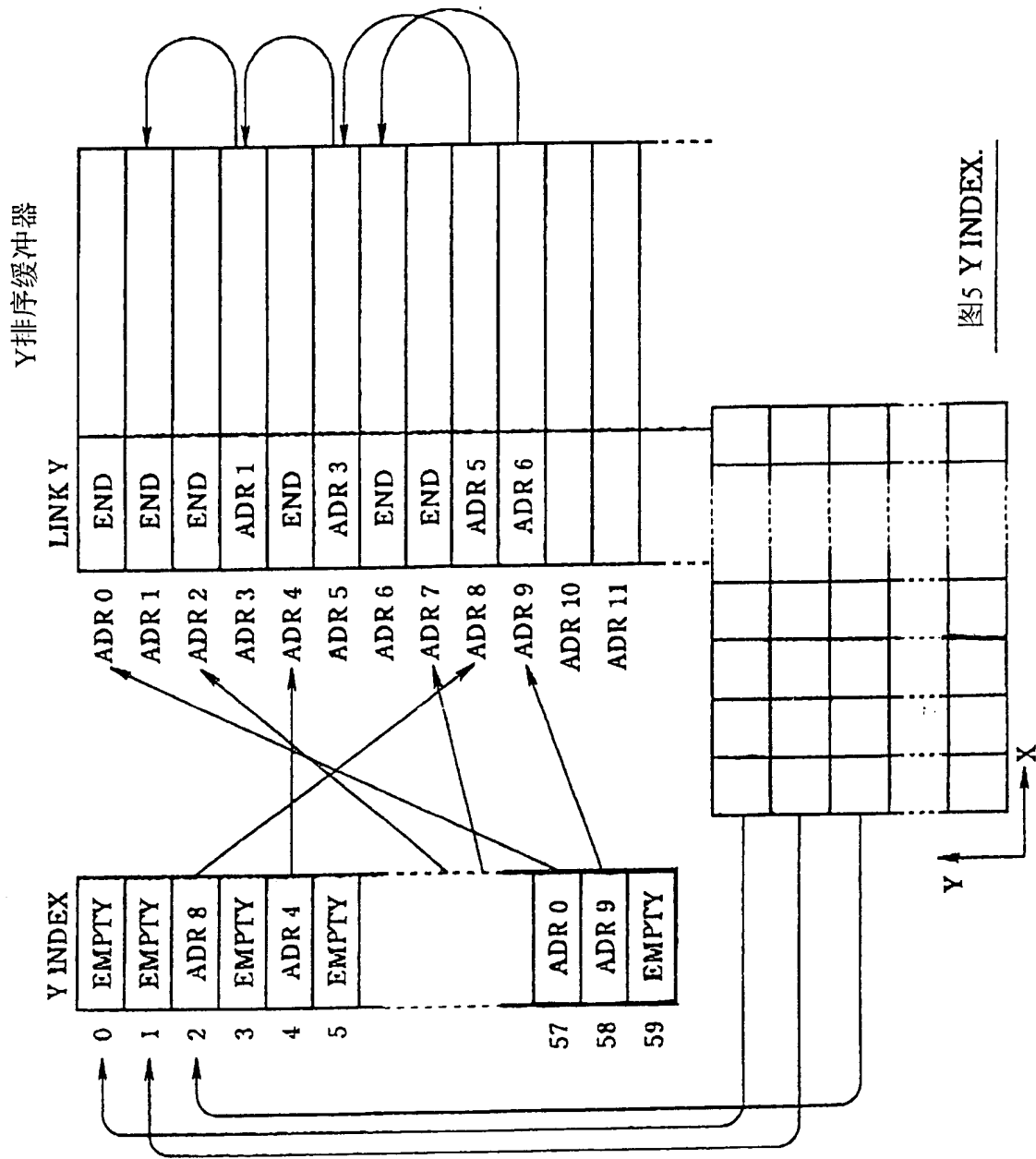


图5 Y INDEX.

图8

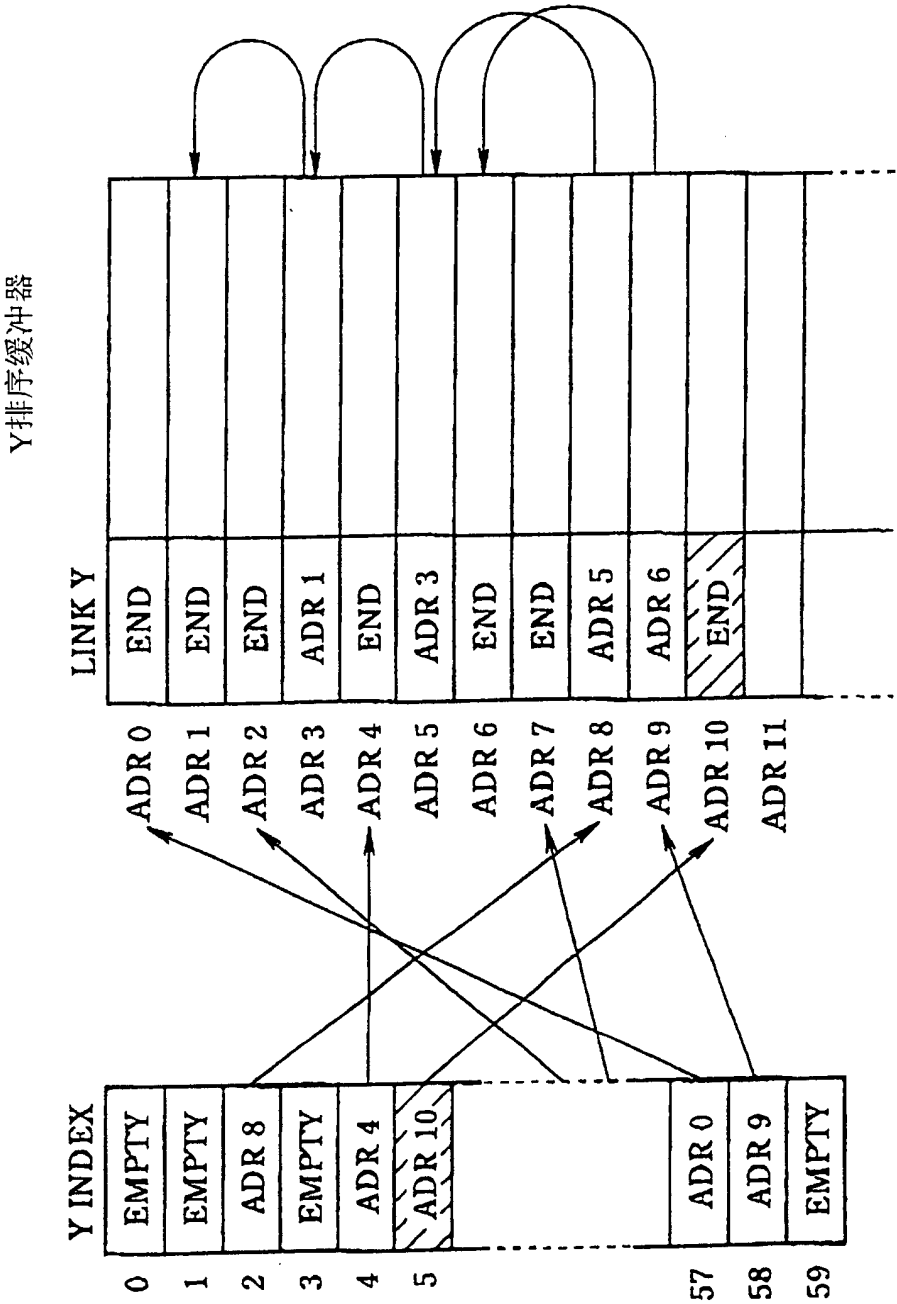


图9

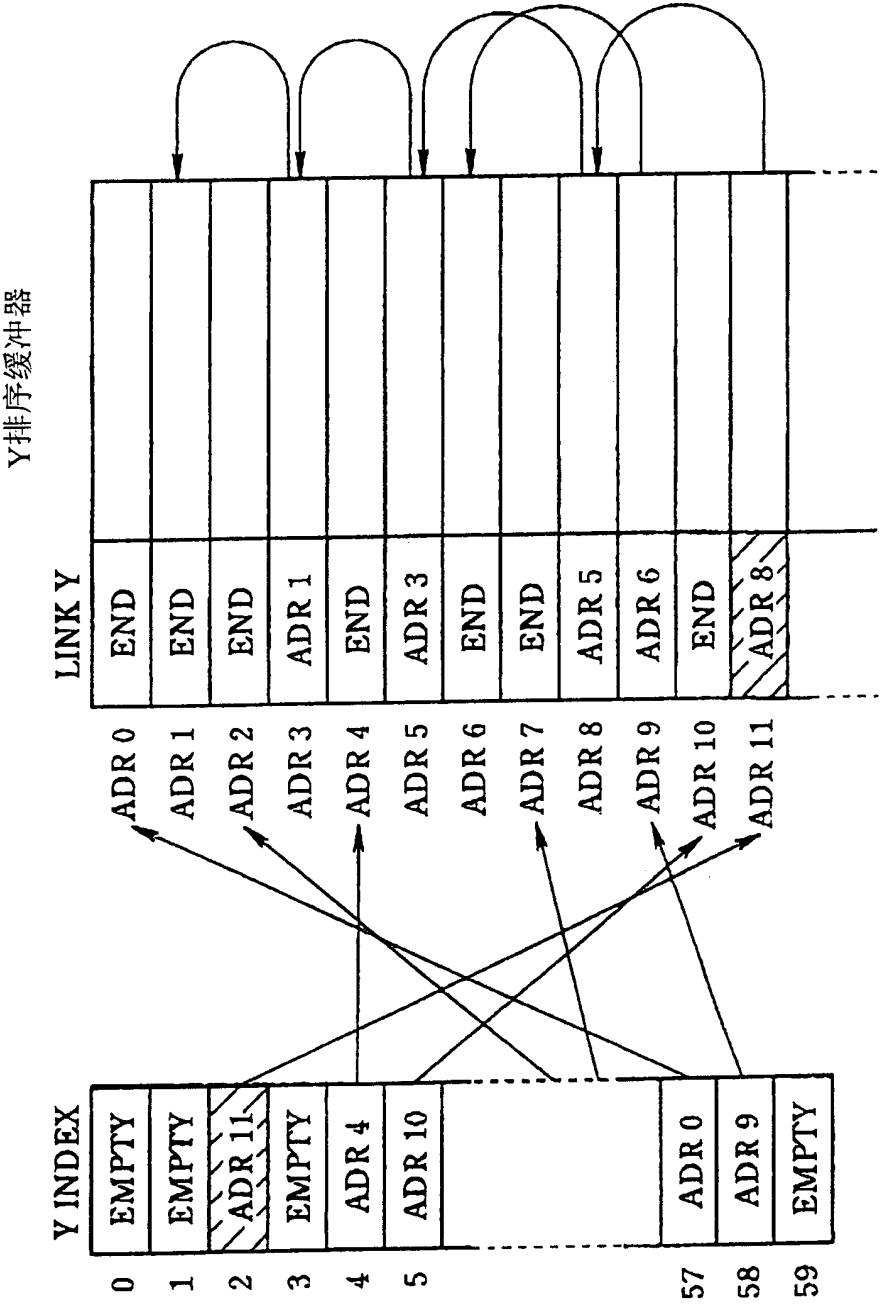
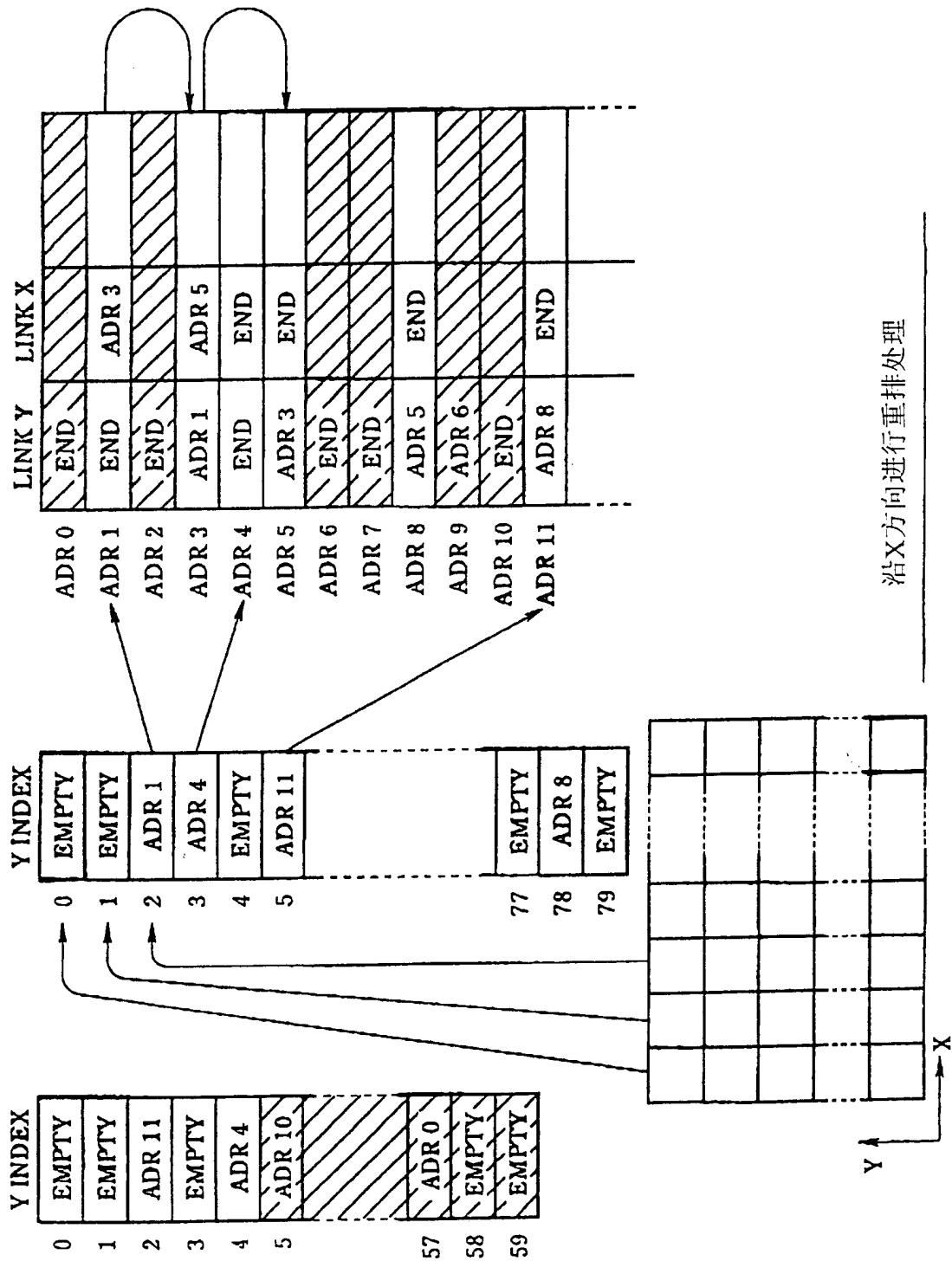


图10



沿X方向进行重排处理

图11

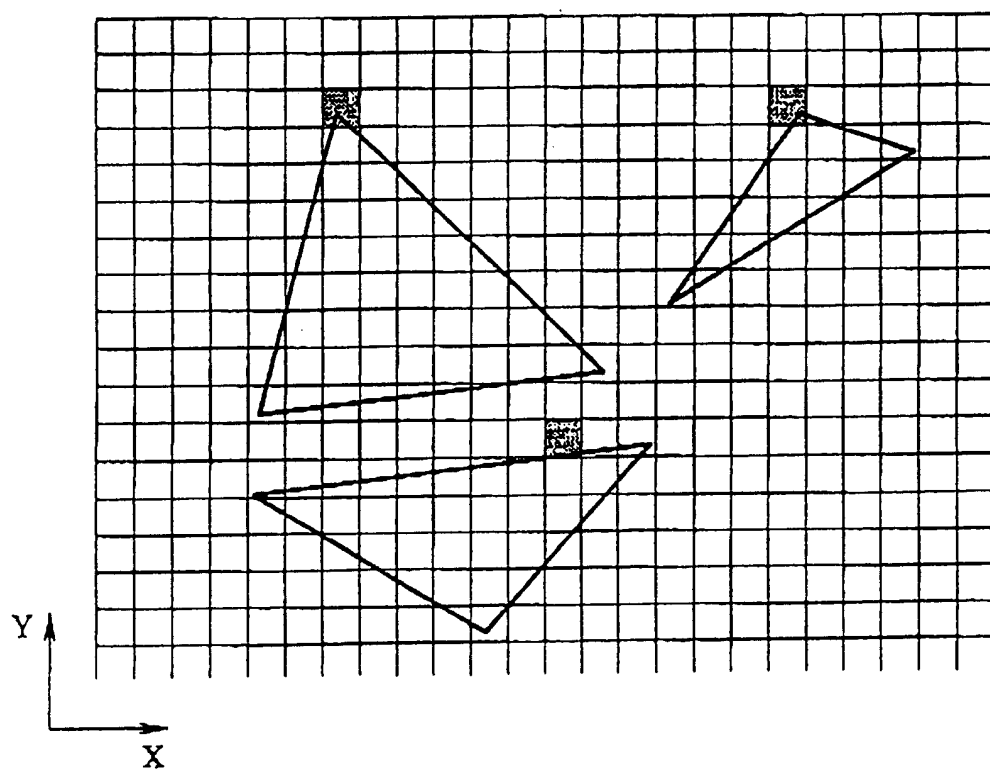


图12

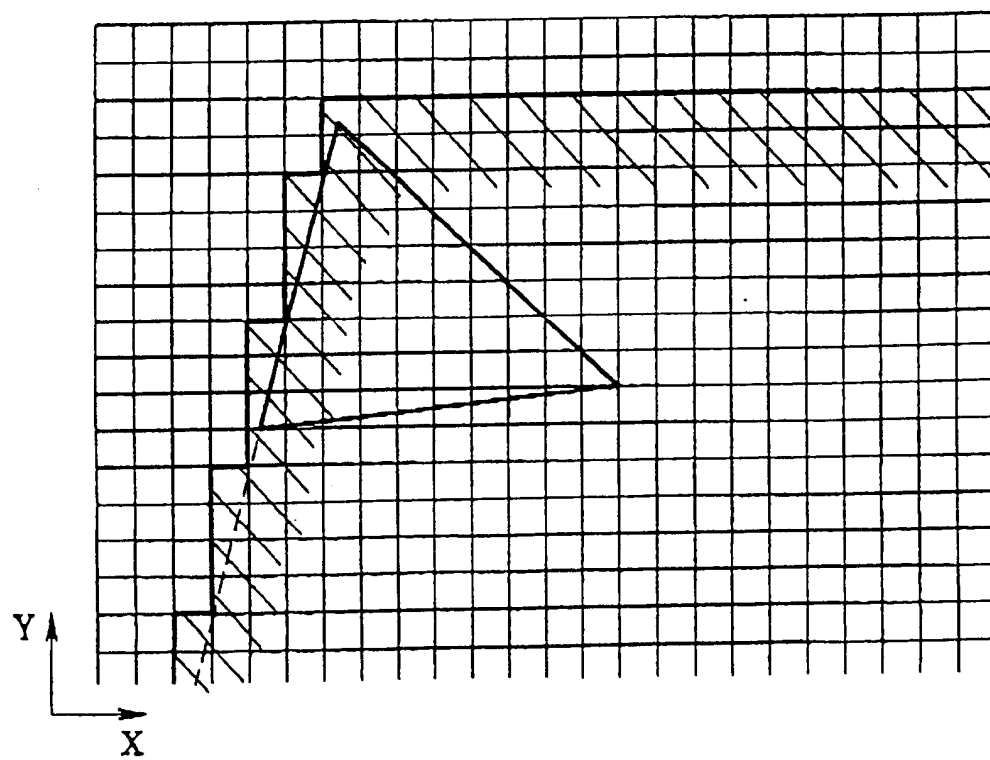
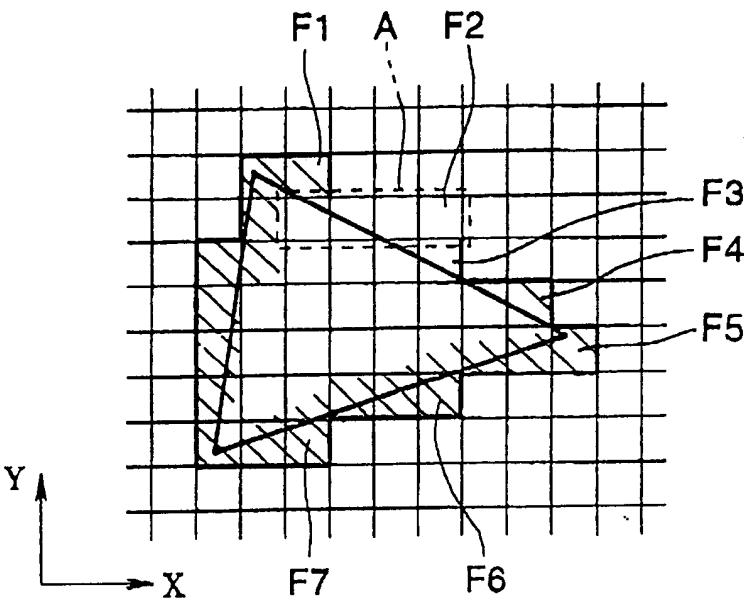
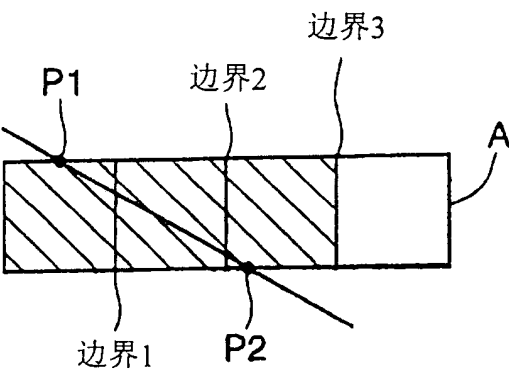


图13



(a)



(b)

图14

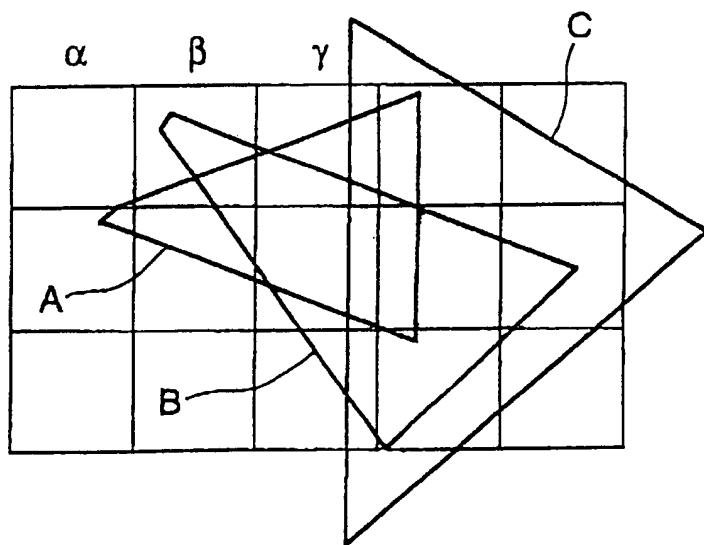


图15

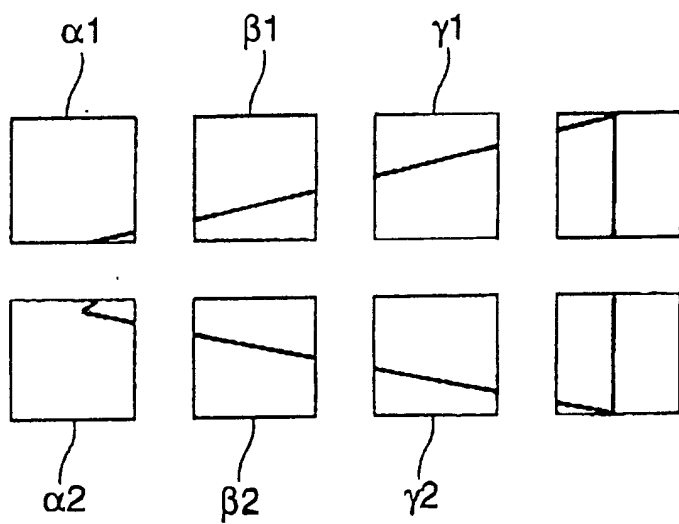


图16

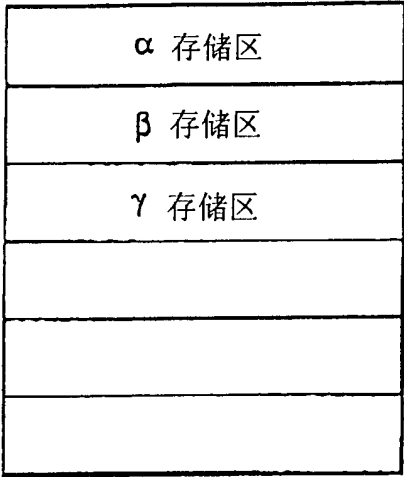


图17

