

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06T 15/00 (2006.01)

G06T 1/60 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200710088590.8

[43] 公开日 2007年9月19日

[11] 公开号 CN 101038676A

[22] 申请日 2007.3.16

[21] 申请号 200710088590.8

[30] 优先权

[32] 2006.3.17 [33] JP [31] 2006-074719

[71] 申请人 索尼株式会社

地址 日本东京

[72] 发明人 渊江孝明

[74] 专利代理机构 中国国际贸易促进委员会专利商
标事务所
代理人 赵科

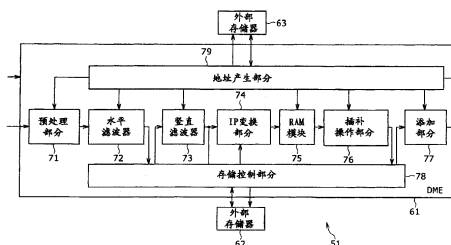
权利要求书 3 页 说明书 32 页 附图 14 页

[54] 发明名称

图像处理装置和图像处理方法

[57] 摘要

本发明提供了一种图像处理装置，包括：产生部分，被配置用于将分别包括预定数量的在第一方向上并置的多边形的多边形组以这些多边形组基本上垂直于第一方向的第二方向上被并置的顺序相继确定为作为产生过程的对象的对象块、并且产生构成多边形单元中对象块的多边形的顶点数据；以及确定部分，被配置用于将与构成相对于所述对象块作为所述对象块的前一对象块的前一紧邻对象块的那些多边形中位于所述前一紧邻对象块的边界部分中的那些多边形相对应的图像数据存储到用于存储数据块单元中的像素数据的存储部分中，以确定构成多边形组并且在第一方向上并置的多边形的数量。



1、一种图像处理装置，包括：

产生部分，被配置用于将分别包括预定数量的在第一方向上并置的多边形的多边形组以所述多边形组在基本垂直于所述第一方向的第二方向上被并置的顺序相继确定为作为产生过程的对象的对象块，并且生成多边形单元的构成对象块的多边形的顶点数据；以及

确定部分，被配置用于将与构成相对于所述对象块作为所述对象块的前一对象块的前一紧邻对象块的那些多边形中位于所述前一紧邻对象块的边界部分中的那些多边形相对应的图像数据存储到用于存储数据块单元的像素数据的存储部分中，以确定构成多边形组并且在第一方向上并置的多边形的数量。

2、根据权利要求1的图像处理装置，其中，所述确定部分基于构成数据块的像素数据的数量确定构成多边形组并且在第一方向上并置的多边形的数量。

3、根据权利要求2的图像处理装置，其中，所述确定部分基于被存储到所述存储部分中的数据块的数量确定构成多边形组且在第一方向上并置的多边形的数量。

4、根据权利要求3的图像处理装置，其中，所述确定部分基于多边形的长度确定构成多边形组且在第一方向上并置的多边形的数量。

5、根据权利要求1的图像处理装置，还包括：

顶点数据变换部分，被配置用于对由所述产生部分所产生的顶点数据进行变换过程，使得由与顶点数据相对应的顶点所界定的多边形可以被放大或缩小；以及

像素数据变换部分，被配置用于进行将由所述顶点数据变换部分所变换的顶点数据变换成像素数据的变换过程。

6、根据权利要求 5 的图像处理装置，其中，所述确定部分基于构成数据块的像素数据的数量确定构成多边形组且在第一方向上并置的多边形的数量。

7、根据权利要求 6 的图像处理装置，其中，所述确定部分基于将被存储到所述存储部分中的数据块的数量确定构成多边形组且在第一方向上并置的多边形的数量。

8、根据权利要求 7 的图像处理装置，其中，所述确定部分基于多边形的长度确定构成多边形组且在第一方向上并置的多边形的数量。

9、根据权利要求 1 的图像处理装置，其中，像素数据包括像素的 z 数据，并且所述图像处理装置还包括：

转换处理部分，被配置用于比较被包括在由所述像素数据变换部分所变换的像素数据中的 z 数据与被包括在与所述像素数据相对应的像素的被存储在所述存储部分中的像素数据中的 z 数据，以执行对由像素构成的图像进行转换的转换过程。

10、一种图像处理方法，包括如下步骤：

将分别包括预定数量的在第一方向上并置的多边形的多边形组以所述多边形组在基本垂直于第一方向的第二方向上被并置的顺序相继确定为作为产生过程的对象的对象块，并且生成多边形单元的构成对象块的多边形的顶点数据；以及

将与构成相对于所述对象块作为所述对象块的前一对象块的前一紧邻对象块的那些多边形中位于所述前一紧邻对象块的边界部分中

的那些多边形相对应的图像数据存储到用于存储数据块单元的像素数据的存储部分中，以确定构成多边形组并且在第一方向上并置的多边形的数量。

图像处理装置和图像处理方法

相关申请的交叉引用

本发明包含与在2006年3月17日向日本专利局提交的日本专利申请 JP 2006-074719 相关的主题，这里通过引用将其全部内容包含在内。

技术领域

本发明涉及图像处理装置和图像处理方法。

背景技术

在相关技术中，在虚拟三维空间中进行图像转换的图像转换装置产生由诸如三角形的多边形的组合所构成的主体形状模型，并进行将图像附着到该模型的纹理贴图以进行图像转换。图像转换装置在例如日本专利公开 No. 2002-83316 中被公开。

如上所述的这种图像转换装置产生纹理地址用于纹理贴图，其中纹理地址代表将被最终输出到将在纹理贴图中被附着的图像上的转换后图像的每个像素在对应于该像素的多边形单元中的位置。然后，为每个像素将纹理地址存储到存储器中。这时，图像转换装置将纹理地址存储到与对应于屏幕上纹理地址的像素位置相对应的存储器的地址中。

参照图 1A、1B、2A 和 2B 描述作为一组多边形集的网格的实例。

图 1 示出了方格形式的网格 1。具体地，网格 1 由多个方形 1C 形成，其中每个方形由多边形 1A 和另一多边形 1B 形成，并且被设置在水平方向和垂直于水平方向的竖直方向上的格子中。网格 1 被用于在涉及固定方向上的转换、诸如翻页或摇动 (page turn or wave) 时的建模。例如，如果对图 1A 中所示的网格 1 进行页面翻转，则网格 1

显示如图 1B 中看到的这种形状。

图 2A 示出了同心网格 2。具体地，网格 2 由多个四边形 2C 形成，其中每个四边形由多边形 2A 和另一多边形 2B 形成，并且被设置在径向和基本垂直于径向的圆周方向上。网格 2 被用于在涉及诸如波纹的同心转换时的建模。例如，如果对图 2A 所示的网格 2 执行波纹，则网格 2 显示如图 2B 中看到的这种形状。

图像转换装置首先利用上述的网格 1 或网格 2 生成转换后的模型，然后基于与形成模型的多边形的顶点相关的数据（这种数据以下被称作顶点数据）执行多边形单元的顶点操作，诸如建模转换，例如平行移动、放大或缩小或旋转、剪贴、用于投影到屏幕上的透视投影转换等。然后，图像转换装置进行光栅化、深度测试、纹理贴图等，以将作为顶点操作结果所得到的多边形单元的顶点数据转换成与像素相关的数据（这种数据以下被称作像素数据），并输出所得到的图像作为转换之后的图像。

应该注意，每个顶点数据包括例如顶点单元数据，诸如例如顶点在建模坐标系上的坐标值、纹理地址和写系数（writing coefficient）。同时，每个像素数据包括代表将屏幕上的像素与虚视点（virtual eye point）相互连接的直线穿过平面的点的 z 坐标的 z 数据、纹理地址、写系数以及代表像素在输出图像上的位置的数据（该数据以下被称作位置数据）。

为了使图像转换装置进行高精确度的转换，必需使用小尺寸的多边形。但是，这导致如下问题，即例如当缓存被用于将在深度测试时包括在像素数据中的 z 数据存储到存储器中时，缓存的命中率被劣化，从而导致对存储器的存取的增加。

这一问题参照图 3 到 5 描述。

应该注意，参照图 3 到 5 给出的下列描述是在假设不执行图像转换的情况下进行的。同时还假设，在图 3 到 5 中，所使用的缓存被配置成使得包括在水平方向和竖直方向上并置的四个像素的 4×4 像素的 z 数据形成作为缓存的存储单元的缓存块。

图3示出模型10,其使用由多边形 11_1 、 11_2 、 11_3 、 11_4 、 11_5 、.....、 11_{n1} 、 11_{n1+1} 、.....、 11_{2n1} 、 11_{2n1+1} 、.....、 11_{3n1}形成的方格。要注意,在没有必要区分各个多边形时,多边形被总称为多边形11。

要注意,在图3中,包括白色或空心圆标记和黑色或实心圆标记的圆形标记中的每一个代表一个像素,并且圆形标记的颜色的白色和黑色代表缓存块之间的边界。具体地,由白色圆形标记所代表的像素和由黑色圆形标记所代表的像素的z数据属于彼此不同的缓存块。此外,彼此不相邻的白色圆形标记的像素的z数据或彼此不相邻的黑色圆形标记的像素的z数据属于互不相同的缓存块。

例如,参照图3,由方形12围住的由黑色圆形标记所代表的 15_1 到 15_4 、 15_{m1+1} 到 15_{m1+4} 、 15_{2m1+1} 到 15_{2m1+4} 和 15_{3m1+1} 到 15_{3m1+4} 这16个像素;由方形13围住的由白色圆形标记所代表的 15_5 到 15_8 、 15_{m1+5} 到 15_{m1+8} 、 15_{2m1+5} 到 15_{2m1+8} 和 15_{3m1+5} 到 15_{3m1+8} 这16个像素的z数据、以及由方形14围住的由黑色圆形标记所代表的 15_9 到 15_{12} 、 15_{m1+9} 到 15_{m1+12} 、 15_{2m1+9} 到 15_{2m1+12} 和 15_{3m1+9} 到 15_{3m1+12} 这16个像素的z数据属于互不相同的缓存块。要注意,在没有必要区分像素时,像素在下文中被总称为像素15。

图像转换装置为构成图3的模型10的多边形11以多边形11被设置在其中的水平方向上的顺序、即以由图3中箭头标记A所指示的方向的顺序产生顶点数据。

具体地,图像转换装置首先为沿着竖直方向上最上面一行并置的 $n1$ 个多边形 11_1 到 11_{n1} 以多边形 11_1 、 11_2 、 11_3 、 11_4 、 11_5 、.....、 11_{n1} 的顺序、即以多边形在水平方向上并置的顺序产生多边形11单元的顶点数据。然后,图像转换装置为沿着竖直方向上上面第二行并置的 $n1$ 个多边形 11_{n1+1} 到 11_{2n1} 以多边形 11_{n1+1} 、.....、 11_{2n1} 的顺序、即以多边形在水平方向上并置的顺序产生多边形11单元的顶点数据。之后,图像转换装置为沿着竖直方向上上面第三行并置的 $n1$ 个多边形 11_{2n1+1} 到 11_{3n1} 以多边形 11_{2n1+1} 、.....、 11_{3n1} 的顺序、即以多边形在水平方向上并置的顺序产生多边形11单元的顶点数据。之后,上面第四

行以及随后行中的多边形 11 的顶点数据被类似地相继产生。

此外，图像转换装置为以上述方式所产生的顶点数据执行顶点操作，并且以顶点数据被获得的顺序对顶点数据进行光栅化以得到像素数据。

换句话说，图像转换装置以多边形 11_1 、 11_2 、 11_3 、 11_4 、 11_5 、……、 11_{n1} 、 11_{n1+1} 、……、 11_{2n1} 、 11_{2n1+1} 、……、 11_{3n1} ……的顺序得到多边形 11 的顶点数据的顶点操作结果，并以这一顺序进行光栅化。因此，以分别对应于多边形 11_1 、 11_2 、 11_3 、 11_4 、 11_5 、……、 11_{n1} 、 11_{n1+1} 、……、 11_{2n1} 、 11_{2n1+1} 、……、 11_{3n1} ……的像素 15_1 、 15_2 、 15_3 、 15_4 、 15_5 、……、 15_{m1} 、 15_{m1+1} 、……、 15_{2m1} 、 15_{2m1+1} 、……、 15_{3m1} ……的顺序得到像素 15 的像素数据。

此外，图像转换装置利用存储在存储器中的 z 数据以获得图像数据的顺序对作为光栅化结果而得到的图像数据进行深度测试。

具体地，图像转换装置首先从存储器中读出由作为缓存块的方形 12 所包围的像素 15_1 到 15_4 、 15_{m1+1} 到 15_{m1+4} 、 15_{2m1+1} 到 15_{2m1+4} 、 15_{3m1+1} 到 15_{3m1+4} 的 z 数据。然后，图像转换装置连续地对像素 15_1 到 15_4 的 z 数据与作为光栅化结果而得到的像素 15_1 到 15_4 的像素数据中所包括的 z 数据按顺序分别进行相互比较，从而进行深度测试。

然后，图像转换装置首先从存储器中读出由作为缓存块的方形 13 所包围的像素 15_5 到 15_8 、 15_{m1+5} 到 15_{m1+8} 、 15_{2m1+5} 到 15_{2m1+8} 和 15_{3m1+5} 到 15_{3m1+8} 的 z 数据。然后，图像转换装置相继地对像素 15_5 到 15_8 的 z 数据与作为光栅化结果而得到的像素 15_5 到 15_8 的像素数据中所包括的 z 数据按顺序分别进行相互比较，从而进行深度测试。之后，像素 15_9 到 15_{m1} 的 z 数据被以类似的方式进行比较以进行深度测试。

这时，因为可以被存储到缓存中的缓存块的数量被限制，所以当要读出新的缓存块时，图像转换装置从作为偶尔需要的最旧的缓存块开始按顺序清除缓存块。

相应地，当图像转换装置在像素 15_9 到 15_{m1} 的像素数据的深度测试之后将要像素 15_{m1+1} 进行深度测试时，包括在当执行像素 15_1 到

15₄ 的深度测试时从存储器中读出作为缓存块的方形 12 中的 16 个像素 15 的 z 数据通常被从缓存中清除。

因此，当要执行像素 15_{m+1} 的深度测试时，图像转换装置必需再次读出包括在方形 12 中的 16 个像素 15 的缓存块。换句话说，图像转换装置必需将每个缓存块读出三次。

图 4 示出了使用由多边形 21 构成的同心网格的模型 20，图 5 示出了使用由多边形 31 构成的同心网格的另一模型 30。

在图 4 中，图像转换装置以多边形 21 被安排的径向上的顺序产生顶点数据并进行构成模型 20 的多边形 21 的顶点操作和光栅化，从而得到像素数据。

具体地，图像转换装置从位于模型 20 中心 O 附近的多边形 21 开始按顺序产生顶点数据并执行在径向 B1 上并置的多边形 21 的顶点操作和光栅化。然后，图像转换装置从位于中心 O 的多边形 21 开始按顺序产生顶点数据并执行位于前面提到的多边形 21 右侧的那些多边形 21 的顶点操作和光栅化。之后，多边形 21 的顶点数据被产生，并且顶点操作和光栅化被类似地执行。

因此，图像转换装置必需在每次对与并置在投影方向 B1 和 B2 上的多边形 21 相对应并且与像素相对应的像素数据进行深度测试时从存储器中读出例如被包括在作为缓存块的方形 22 中的那些像素的 z 数据。

另一方面，在图 5 中，图像转换装置在设置的多边形 31 上的同心圆上按顺序产生顶点数据并执行构成图 5 的模型 30 的多边形 31 的顶点操作和光栅化，从而得到像素数据。

具体地，图像转换装置以多边形 31 被设置在其上的同心圆的顺序产生顶点数据并执行在圆周方向 C1 上同心并置的多边形 31 的顶点操作和光栅化。然后，图像转换装置以多边形 31 在箭头标记 C2 方向上并置的顺序产生顶点数据并进行在多边形 31 的外侧相邻并且以箭头标记 C2 同心并置的多边形 31 的顶点操作和光栅化。之后，多边形 31 的顶点数据被产生，并且顶点操作和光栅化被类似地执行。

因此，图像转换装置在每次对与在对应于像素的圆周方向 C1 到 C4 上并置的多边形 31 相对应的图像数据进行深度测试时必需从存储器中读出例如包括在方形 32 中的像素的 z 数据。换句话说，图像转换装置必需将每个缓存块读出四次。

发明内容

如果在不同于与上述顶点数据的产生顺序的方向相对应的方向上并置小的且与缓存块的 z 数据相对应的多个多边形，则通常由于清除的发生而必需将每个缓存块读出多次。换句话说，缓存的命中效率低。

因此，在多边形尺寸被减小以实现高精确度的转换的情况下，存储器的带宽必须被增加，从而导致功耗的增加或生产成本的增加。

因此，期望提供一种对存储器的存取效率增加并且能够高精确度且低功耗地转换图像的图像处理装置和图像处理方法。

根据本发明的一个实施例，提供了一种图像处理装置，其包括：产生部分，被配置用于将分别包括预定数量的在第一方向上并置的多边形所构成的多边形组以这些多边形组在基本上垂直于第一方向的第二方向上被并置的顺序相继确定为作为产生过程的对象的对象块、并且生成多边形单元的构成对象块的多边形的顶点数据；以及确定部分，被配置用于将与构成相对于该对象块的作为该对象块前一对象块的紧邻的前一对象块的那些多边形中位于该紧邻的前一对象块的边界部分中的那些多边形相对应的图像数据存储到用于存储数据块单元的像素数据的存储部分中，以确定构成多边形组并且在第一方向上并置的多边形的数量。

确定部分可以基于构成数据块的像素数据的数量确定构成多边形组并且在第一方向上并置的多边形的数量。在这种情况下，确定部分可以基于要被存储到存储部分中的数据块的数量确定构成多边形组并且在第一方向上并置的多边形的数量。此外，确定部分基于多边形的长度确定构成多边形组并且在第一方向上并置的多边形的数量。

图像处理装置还可以包括：顶点数据变换部分，被配置用于对由产生部分所产生的顶点数据进行变换过程，使得由与顶点数据相对应的顶点所限定的多边形可以被放大或缩小；以及像素数据变换部分，被配置用于进行将由顶点数据变换部分所变换的顶点数据变换成像素数据的变换过程。

在这种情况下，确定部分可以基于构成数据块的像素数据的数量确定构成多边形组且在第一方向上并置的多边形的数量。此外，确定部分可以基于要被存储到存储部分中的数据块的数量确定构成多边形组且在第一方向上并置的多边形的数量。而且，确定部分可以基于多边形的长度确定构成多边形组且在第一方向上并置的多边形的数量。

图像处理装置可以被配置成使得像素数据包括像素的 z 数据，并且图像处理装置还包括：转换处理部分，被配置用于比较包括在由像素数据变换部分所变换的像素数据中的 z 数据与包括在与该像素数据相对应的像素的被存储在存储部分中的像素数据中的 z 数据，以进行对由像素构成的图像进行转换的转换过程。

根据本发明的另一实施例，提供了一种图像处理方法，包括：产生步骤，将分别由预定数量的在第一方向上并置的多边形构成的多边形组以这些多边形组在基本上垂直于第一方向的第二方向上被并置的顺序相继确定为作为产生过程的对象的对象块、并且生成多边形单元的构成对象块的多边形的顶点数据；以及确定步骤，将与构成相对于该对象块的作为该对象块前一对象块的紧邻的前一对象块的那些多边形中位于该紧邻的前一对象块的边界部分中的那些多边形相对应的图像数据存储到用于存储数据块单元的像素数据的存储部分中，以确定构成多边形组并且在第一方向上并置的多边形的数量。

在图像处理装置和图像处理方法中，分别由预定数量的在第一方向上并置的多边形所构成的多边形组被以这些多边形组在基本上垂直于第一方向的第二方向上被并置的顺序相继确定为作为产生过程对象的对象块。然后，生成多边形单元的构成对象块的多边形的顶点数据。然后，将与构成相对于该对象块的作为该对象块前一对象块的紧邻的

前一对象块的那些多边形中位于该紧邻的前一对象块的边界部分中的那些多边形相对应的图像数据存储在存储部分中，以确定构成多边形组并且在第一方向上并置的多边形的数量。

通过图像处理装置和图像处理方法，图像可以被转换。

此外，通过图像处理装置和图像处理方法，提高了对存储部分的存取效率，并且因此，可以低功耗地实现具有高精确度的图像转换。

结合附图，根据下面的描述和所附权利要求，本发明的上述和其它特征以及优点将变得很清楚，在附图中，相同的部分或单元由相同的标号指代。

附图说明

图 1A 和 1B 是示出方格形式的网格实例的示意图；

图 2A 和 2B 是示出同心圆形式的网格实例的示意图；

图 3、4 和 5 是图示顶点数据的传统产生顺序的不同实例的示意图；

图 6 是示出本发明被应用到其中的图像处理系统的配置实例的框图；

图 7 是图示图像转换过程的流程图；

图 8 是示出在图 6 中示出的外部存储器的具体配置和地址产生过程实例的框图；

图 9 是图示地址产生过程的流程图；

图 10、11 和 12 是图示图像处理系统产生顶点数据的不同顺序的示意图；

图 13 是图示图像处理系统的输出过程的流程图；

图 14 是示出图像处理装置的配置实例的框图；以及

图 15 是示出记录和再现装置的配置实例的框图。

具体实施方式

图 6 示出了本发明被应用到其中的图像处理系统的配置实例。

现在参照图 6, 所示的图像处理系统 51 包括数据图像特殊效果装置 (数字多效装置: Digital Multi Effects apparatus) (以下简称为 DME) 61、外部存储器 62 和另一外部存储器 63。图像处理系统 51 进行作为被输入到其中的场单元图像的输入图像的缩小、放大、类型改变、旋转、向左或向右翻转、反转或移动, 或者对输入图像应用特殊效果以转换输入图像, 并输出所得到的帧单元图像。特殊效果可以是例如在水平方向或竖直方向上的镶嵌效果 (mosaic effect)、均一效果 (postalization)、负/正翻转效果或散焦效果。

应该注意, 输入图像由像素的代表亮度信号、色差信号和用于键控的键 (key) 信号的像素值形成。此外, 图像处理系统 51 的部件根据存储在外存储器 63 中的程序执行各种过程。

DME 61 由例如 IC (集成电路)、LST 等形成。DME 61 包括预处理部分 71、水平滤波器 72、竖直滤波器 73、IP (隔行逐行: Interlace Progressive) 变换部分 74、RAM (随机存取存储器) 模块 75、插补 (interpolation) 操作部分 76、添加部分 77、存储控制部分 78 和地址产生部分 79。

DME 61 接收作为在纹理贴图时要被应用的图像的输入图像以及被提供给其的定时信号。输入图像被提供给预处理部分 71。定时信号被提供给 DME 61 的部件, 使得这些部件可以响应于定时信号而执行相应过程。

预处理部分 71 响应于从地址产生部分 79 提供给其的指令信号对输入图像应用诸如镶嵌、均一和正/负翻转效果的特殊效果。具体地, 预处理部分 71 对像素单元中形成输入图像的像素中预定的一些像素进行滤波过程, 以对输入图像应用镶嵌效果。此外, 预处理部分 71 改变形成输入图像的像素的像素值的等级数, 以对输入图像应用均一化。而且, 预处理部分 71 使形成输入图像的像素的像素值的等级反转, 从而对输入图像应用正/负翻转。预处理部分 71 将通过应用特殊效果而得到的场单元图像提供给水平滤波器 72。

水平滤波器 72 接收从地址产生部分 79 提供给其的水平方向上的

缩小比例。然后，为了消除在图像被缩小时所出现的水平方向上的重叠部分，水平滤波器 72 在水平方向上对从预处理部分 71 所接收的场单元图像执行与所接收的缩小比例相对应的滤波过程。此外，水平滤波器 72 响应于从地址产生部分 79 提供给其的指令信号对来自预处理部分 71 的场单元图像在水平方向上应用散焦作为特殊效果。水平滤波器 72 将通过在水平方向上应用滤波过程或散焦过程而得到的场单元图像通过存储控制部分 78 提供给外部存储器 62，使得图像被存储到外部存储器 62 中。

竖直滤波器 73 接收从地址产生部分 79 提供给其的竖直方向上的缩小比例。此外，为了消除在图像被缩小时所出现的竖直方向上的重叠部分，竖直滤波器 73 在竖直方向上对从存储控制部分 78 提供给其的并在竖直方向上从外部存储器 62 中读出的场单元图像进行与所接收的缩小比例相对应的滤波过程。此外，竖直滤波器 73 响应于从地址产生部分 79 提供给其的指令信号对来自存储控制部分 71 的场单元图像在竖直方向上应用散焦作为特殊效果。竖直滤波器 73 将通过在竖直方向上应用滤波过程或散焦过程而得到的场单元图像提供给 IP 变换部分 74。竖直滤波器 73 还通过存储控制部分 78 将图像提供给外部存储器 62，使得图像被存储到外部存储器 62 中。

IP 变换部分 74 通过参照另一位于该图像紧邻前面的场单元图像以及该紧邻的前一图像前面的另一场单元图像来对从竖直滤波器 73 提供给其的场单元图像（隔行图像：interlaced image）进行 IP 变换。这两个前面的图像被从存储控制部分 78 提供到 IP 变换部分 74。IP 变换部分 74 将通过 IP 变换而得到的帧单元图像（逐行图像：progressive image）提供给 RAM 模块 75。

RAM 模块 75 存储来自 IP 变换部分 74 的帧单元图像。此外，RAM 模块 75 基于从地址产生部分 79 所提供的纹理地址的整数部分、即输入图像上作为插补操作部分 76 的插补操作的对象的那些像素（以下称作插补像素）的坐标的水平方向上坐标值的整数部分和竖直方向上坐标值的整数部分来进行读出操作。RAM 模块 75 从构成已被存储在

RAM 模块 75 中的帧单元图像的像素的像素值中读出将被用于插补像素的插补的不同像素的多个像素值作为像素值组。例如，RAM 模块 75 读出插补像素周围的不同像素的 64 个像素值作为像素值组。RAM 模块 75 将所读出的像素值组提供给插补操作部分 76。

应该注意，对于输入图像上的坐标系，构成输入图像的像素在水平方向和竖直方向上的坐标值都是整数。

插补操作部分 76 基于纹理地址的小数部分 - 即从地址产生部分 79 所提供的输入图像上插补像素的坐标的水平方向和竖直方向上的坐标值的小数部分 - 以及从 RAM 模块 75 所提供的像素值组执行插补操作，以对插补像素的像素值进行插补来执行纹理贴图。插补操作部分 76 通过存储控制部分 78 将插补后的帧单元图像提供给外部存储器 62，以被存储到外部存储器 62 中。

添加部分 77 利用写系数为从地址产生部分 79 提供给其的每个插补像素应用浓淡处理 (shading)。添加部分 77 输出添加后的图像作为转换后的图像。

存储控制部分 78 控制到和从外部存储器 62 的写入和读出。具体地，存储控制部分 78 将用于控制写入到外部存储器 62 的控制信号提供给外部存储器 62，并且将从水平滤波器 72、竖直滤波器 73 或插补操作部分 76 所提供的图像提供给外部存储器 62，使得图像被写入外部存储器 62 中。

此外，存储控制部分 78 将用于控制从外部存储器 62 读出图像的控制信号提供给外部存储器 62 以控制从外部存储器 62 的读出。而且，存储控制部分 78 提供从外部存储器 62 所读出的图像作为对竖直滤波器 73、IP 变换部分 74 和添加部分 77 的控制的结果。

地址产生部分 79 响应于来自用户的转换输入图像指令而控制图像处理系统 51 的部件。此外，地址产生部分 79 控制到和从外部存储器 63 的写入以及读出。

具体地，地址产生部分 79 将处理的中间结果或最终结果与用于控制写入到外部存储器 63 以被存储到外部存储器 63 中的控制信号一

起提供给外部存储器 63，并且使得处理的中间或最终结果被写入到外部存储器 63 中。此外，地址产生部分 79 将用于控制从外部存储器 63 读出处理的中间结果或最终结果的控制信号提供给外部存储器 63，以控制从外部存储器 63 的读出，从而从外部存储器 63 读出处理的中间或最终结果。地址产生部分 79 的细节在下面参照图 8 描述。

现在，参照图 7 描述由图 6 的图像处理系统 51 所执行的用以转换输入图像的图像转换过程。例如在将输入图像输入到图像处理系统 51 中时开始图像转换过程。

在步骤 S1，预处理部分 71 响应于从地址产生部分 79 提供给其的指令信号而对输入图像执行诸如镶嵌、均一化或正/负翻转的过程。然后，预处理部分 71 将通过该过程所得到的场单元图像提供给水平滤波器 72，之后，过程前进到步骤 S2。要注意，如果没有从地址产生部分 79 提供指令信号，则处理跳过步骤 S1 并前进到步骤 S2。

在步骤 S2，水平滤波器 72 响应于从地址产生部分 79 提供给其的水平方向上的缩小比例而对来自预处理部分 71 的场单元图像执行对应于水平方向上缩小比例的滤波过程。此外，水平滤波器 72 响应于从地址产生部分 79 所提供的指令信号而对图像执行水平方向上的散焦过程作为用于特殊效果。然后，水平滤波器 72 将通过对该图像执行水平方向上的滤波过程和/或散焦过程而得到的场单元图像提供给存储控制部分 78。

在步骤 S2 的过程之后，处理前进到步骤 S3，在该步骤，存储控制部分 78 将从水平滤波器 72 提供给其的场单元图像连同用于控制写入到外部存储器 62 的控制信号一起提供给外部存储器 62，使得图像可以被存储到外部存储器 62 中。在步骤 S3 的过程之后，处理前进到步骤 S4，在该步骤，存储控制部分 78 将用于控制从外部存储器 62 读出图像的控制信号提供给外部存储器 62。因此，在步骤 S3 被存储的场单元图像在竖直方向上被从外部存储器 62 中读出，并被提供到竖直滤波器 73。

在步骤 S4 的过程之后，处理前进到步骤 S5，在该步骤，竖直滤

波器 73 响应于从地址产生部分 79 所提供的竖直方向上的缩小比例而对从存储控制部分 78 所提供的场单元图像执行对应于竖直方向上缩小比例的滤波过程。此外，竖直滤波器 73 响应于从地址产生部分 79 所提供的指令信号而对场单元图像执行竖直方向上的散焦过程作为特殊效果。然后，竖直滤波器 73 将通过滤波过程和/或散焦过程所得到的场单元图像提供给 IP 变换部分 74，并且还提供给存储控制部分 78。

在步骤 S5 的过程之后，处理前进到步骤 S6，在该步骤，存储控制部分 78 将从存储控制部分 78 所提供的场单元图像连同用于控制写入到外部存储器 62 的控制信号一起提供给外部存储器 62，使得图像被存储到外部存储器 62 中。在步骤 S6 的过程之后，处理前进到步骤 S7。在步骤 S7，存储控制部分 78 将用于控制从外部存储器 62 读出图像的控制信号提供给外部存储器 62，以从外部存储器 62 读出位于该场单元图像紧邻前面的且在紧邻的前一操作循环中被步骤 S6 的过程存储的场单元图像以及紧邻的前一场单元图像，并将读出的图像提供给 IP 变换部分 74。

在步骤 S7 的过程之后，处理前进到步骤 S8，在该步骤，IP 转换部分 74 参照在步骤 S7 从存储控制部分 78 提供给它的两个图像，以对在步骤 S5 从竖直滤波器 73 所提供的场单元图像进行 IP 变换。然后，IP 变换部分 74 将通过 IP 变换所得到的帧单元图像提供给 RAM 模块 75。

在步骤 S9，RAM 模块 75 存储来自 IP 变换部分 74 的帧单元图像，之后，处理前进到步骤 S10。

在步骤 S10，RAM 模块 75 基于从地址产生部分 79 所提供的纹理地址的整数部分，从构成已被存储的帧单元图像的像素的像素值中读出将被用于插补像素的插补的不同像素的多个像素值作为像素值组。像素值组被提供给插补操作部分 76。

在步骤 S10 的过程之后，处理前进到步骤 S11，在该步骤，插补操作部分 76 基于从地址产生部分 79 所提供的纹理地址的小数部分以及从 RAM 模块 75 所提供的像素值组进行插补操作，以对插补像素的

像素值进行插补。然后，插补操作部分 76 将插补后的图像提供给存储控制部分 78。

在步骤 S11 的过程之后，处理前进到步骤 S12，在该步骤，存储控制部分 78 将来自插补操作部分 76 的图像连同用于控制到外部存储器 62 的写入的控制信号一起提供给外部存储器 62，使得图像被存储到外部存储器 62 中。在步骤 S12 的过程之后，处理前进到步骤 S13，在该步骤，存储控制部分 78 将用于控制从外部存储器 62 读出图像的控制信号提供给外部存储器 62，以读出在步骤 S12 所存储的图像。之后，处理前进到步骤 S14。

在步骤 S14，添加部分 77 利用从地址产生部分 79 提供给它的写系数给图像添加浓淡处理。然后，添加部分 77 输出添加后的图像作为变换后的图像，从而结束处理。

要注意，前面描述的图 7 的图像转换过程的步骤 S1 到 S9 的过程在插补前的像素单元中执行，而步骤 S10 到 S14 的过程在插补像素单元中执行。

图 8 示出了图 6 所示外部存储器 63 和地址产生部分 79 的详细配置实例。

现在参照图 8，所示的外部存储器 63 包括存储器 101、z 数据存储单元 102、另一 z 数据存储单元 103 和地址存储器 104。

同时，地址产生部分 79 包括 CPU（中央处理单元）121、顶点操作部分 122、DDA（数字微分分析器）设置部分 123、DDA 124 和深度测试部分 125、z 数据缓存 126、切换部分 127、地址缓存 128、存储控制部分 129 和输出部分 130。

CPU 121 根据存储在外存储器 63 的存储器 101 中的程序控制图像处理系统 51 的部件。

例如，CPU 121 响应于用户的指令而将指示镶嵌效果、均一效果或负/正翻转效果的指令信号提供给预处理部分 71。此外，CPU 121 响应于用户的指令将指示水平方向散焦的指令信号提供给水平滤波器 72，或者将指示垂直方向散焦的指令信号提供给垂直滤波器 73。

而且，CPU 121 响应于用户的指令将水平方向上的缩小比例提供给水平滤波器 72，并将垂直方向上的缩小比例提供给垂直滤波器 73。此外，CPU 121 响应于用户的建模操作而在多边形单元中以预定顺序产生关于构成与转换后的帧单元图像相对应的模型的多边形的顶点的顶点数据（以下称作顶点数据）。然后，CPU 121 将所产生的顶点数据提供给顶点操作部分 122。下面参照图 10 到 13 描述顶点数据的产生顺序。

此外，CPU 121 控制到和从存储器 101 的写入以及读出。例如，CPU 121 将处理的中间结果或最终结果连同用于控制到存储器 101 的写入的控制信号一起提供给存储器 101，使得处理的中间结果或最终结果被存储到存储器 101 中。此外，CPU 121 将用于控制从存储器 101 读出的控制信号提供给存储器 101，以从存储器 101 中读出处理的中间结果或最终结果，并执行如上所述的各种过程。

顶点操作部分 122 基于从 CPU 121 所提供的顶点数据而在多边形单元中执行顶点操作，并将顶点操作在多边形单元中所获得的顶点数据提供给 DDA 设置部分 123。DDA 设置部分 123 基于来自顶点操作部分 122 的多边形单元的顶点数据计算 DDA 124 所必需的参数，诸如由屏幕上位置的改变而引起的纹理地址的变化。DDA 设置部分 123 将多边形单元的顶点数据和参数提供给 DDA 124。

DDA 124 基于从 DDA 设置部分 123 所提供的参数将多边形单元的顶点数据变换成像素数据。DDA 124 将变换所得到的像素数据提供给深度测试部分 125。深度测试部分 125 使用来自 DDA 124 的像素数据和存储在 z 数据缓存 126 中的对应于该像素数据的像素的像素数据，以执行阴影面擦除过程。

具体地，深度测试部分 125 向 z 数据缓存 126 发出读出与包括在来自 DDA 124 的像素数据中的位置数据相关联的像素单元的地址（以下称作 z 数据像素单元地址）相关联存储的 z 数据的请求。然后，深度测试部分 125 从 z 数据缓存 126 获得响应于该请求所读出的 z 数据。

然后，深度测试部分 125 比较该 z 数据与包括在来自 DDA 124

的像素数据中的 z 数据，以执行判断与来自 DDA 124 的像素数据相对应的多边形（以下被适当地称作新多边形）相对于与已被存储在 z 数据缓存 126 中的 z 数据相对应的多边形（以下被适当地称作旧多边形）是否位于虚视点侧的深度测试。

深度测试部分 125 响应于深度测试的结果，将包括在相对于旧多边形位于虚视点侧的新多边形的像素数据中的写系数和 z 数据连同与该像素元素相对应的 z 数据像素单元地址一起提供给 z 数据缓存 126。因此， z 数据缓存 126 用新多边形的 z 数据和写系数来更新与来自深度测试部分 125 的 z 数据像素单元地址相关联存储的旧多边形的 z 数据和写系数。

此外，深度测试部分 125 将包括在相对于旧多边形位于虚视点侧的新多边形的像素数据中的纹理地址连同与包括在像素数据中的位置数据相对应的像素单元地址（该地址以下称作纹理像素单元地址）一起输出到地址缓存 128。

z 数据缓存 126 将缓存块单元的 z 数据和写数据作为与存储 z 数据和写系数的 z 数据存储单元 102 或 103 中缓存块单元的地址（以下被称为 z 数据缓存块单元地址）相关联的缓存块临时存储到缓存块区域中。

而且， z 数据缓存 126 基于从深度测试部分 125 所提供的 z 数据像素单元地址在已被存储在其中的 z 数据中搜索与深度测试部分 125 要求读出的 z 数据像素单元地址相对应的 z 数据。具体地， z 数据缓存 126 搜索包括与来自深度测试部分 125 的 z 数据像素单元地址相对应的 z 数据缓存块单元地址相关联存储的请求的目标的 z 数据的缓存块，并将缓存块中的请求的目标的 z 数据确定为搜索结果。

如果搜索到请求目标的 z 数据，则 z 数据缓存 126 将搜索结果的 z 数据和相应的写系数提供给深度测试部分 125。如果没有搜索到请求目标的 z 数据，即如果没有存储包括请求目标的 z 数据的缓存块，则 z 数据缓存 126 如下发送。 z 数据缓存 126 将与从深度测试部分 125 所提供的 z 数据像素单元地址相对应的 z 数据缓存块单元地址连同 z 数

据一起发送到切换部分 127，以请求切换部分 127 读出请求对象的 z 数据和相应的写系数。因此，z 数据和写系数在来自 z 数据存储单元 102 或 103 的块单元中被读出。

z 数据缓存 126 还将响应于请求而从切换部分 127 所提供的缓存块单元的 z 数据和写系数暂时作为缓存块而与在请求时被发送到切换部分 127 的 z 数据缓存块单元地址相关联地存储到缓存块区域中。然后，z 数据缓存 126 将 z 数据提供到深度测试部分 125。

此外，z 数据缓存 126 在其没有缓存块区域可以存储时执行缓存块的清除。具体地，z 数据缓存 126 决定作为清除对象的缓存块，读出该缓存块和与该缓存块相关联的 z 数据缓存块单元地址，并将缓存块和 z 数据缓存块单元地址连同清除请求一起提供给切换部分 127。

切换部分 127 控制到和从 z 数据存储单元 102 和 103 的写入和读出。具体地，切换部分 127 响应于输出部分 130 或 z 数据缓存 126 的请求而将待控制的 z 数据存储单元从外部存储器 63 的 z 数据存储单元 102 和 103 中的一个切换到另一个。例如，切换部分 127 响应于来自 z 数据缓存 126 的读出请求而从 z 数据存储单元 102 和 103 的一个中读出存储在来自 z 数据缓存 126 的 z 数据缓存块单元地址中的缓存块单元的 z 数据和写系数。切换部分 127 将所读出的 z 数据和写系数提供给 z 数据缓存 126。

此外，切换部分 127 响应于来自 z 数据缓存 126 的清除请求而将被提供给它的 z 数据和写系数作为缓存块存储到从 z 数据存储单元 102 和 103 之一中 z 数据缓存 126 所提供的 z 数据缓存块单元地址中。

此外，切换部分 127 响应于来自输出部分 130 的请求而从 z 数据存储单元 102 和 103 中另一个中读出 z 数据和写系数，并将 z 数据和写系数提供给输出部分 130。之后，切换部分 127 清除存储在 z 数据存储单元 102 和 103 中另一个的 z 数据和写系数。

如上所述，z 数据存储单元 102 和 z 数据存储单元 103 之一存储用于深度测试部分 125 比较的 z 数据和相应的写系数，而 z 数据存储单元 102 和 103 中的另一个存储通过输出部分 130 所输出的深度测试结束后的

z 数据和写系数。

换句话说，切换部分 127 在每次与一个帧的模型相对应的像素数据被提供给深度测试部分 125 时都进行切换。

地址缓存 128 将缓存块单元的纹理地址作为缓存块而与存储缓存块的地址存储器 104 中的缓存块单元的地址（以下称作纹理缓存块单元地址）相关联地暂时存储在缓存块区域中。地址缓存 128 基于来自深度测试部分 125 的纹理像素单元地址，将包括与纹理像素单元地址一起所输出的纹理地址的缓存块单元的纹理地址作为缓存块而与存储缓存块地址存储器 104 中缓存块单元的地址（以下称作纹理缓存块单元地址）相关联地暂时存储在缓存块区域中。

此外，地址缓存 128 基于从深度测试部分 125 所提供的纹理像素单元地址而从已被存储在其中的纹理地址中搜索与纹理像素单元地址相对应的纹理。具体地，地址缓存 128 搜索包括与对应于纹理像素单元地址的纹理缓存块单元地址相关联地存储的对应于来自深度测试部分 125 的纹理像素单元地址的纹理地址的缓存块。然后，地址缓存 128 将这样搜索到的缓存块确定为搜索结果。

如果搜索到纹理地址，则地址缓存 128 将搜索结果的纹理地址更新为来自深度测试部分 125 的纹理地址。如果没有搜索到纹理地址，则地址缓存 128 将与从深度测试部分 125 所提供的纹理像素单元地址相对应的纹理缓存块单元地址连同纹理一起发送到存储控制部分 129，从而请求存储控制部分 129 读出与纹理像素单元地址相对应的纹理地址。因此，纹理缓存地址在缓存块单元中被从地址存储器 104 中读出。

此外，地址缓存 128 将响应于请求而从存储控制部分 129 提供给它的缓存块单元中的纹理地址作为缓存块而与在请求时被发送到存储控制部分 129 的纹理缓存块单元地址相关联地暂时存储到缓存块区域中。

此外，当地址缓存 128 没有缓存块区域能够存储时，地址缓存 128 执行缓存块的清除。具体地，地址缓存 128 确定清除对象的缓存块，读出该缓存块和与该缓存块相关联的纹理缓存块单元地址等，并将这

样读出的纹理缓存块单元地址等连同清除请求一起提供给存储控制部分 129。

存储控制部分 129 控制可以由 SDRAM 等构成的地址存储器 104 的写入和读出。具体地，存储控制部分 129 响应于来自地址缓存 128 的读出请求而从地址存储器中读出被存储在来自地址缓存 128 的纹理缓存块单元地址的缓存块单元的纹理地址。然后，存储控制部分 129 将这样读出的纹理地址提供给地址缓存 128。

此外，存储控制部分 129 响应于来自地址缓存 128 的请求，将从缓存块单元中地址缓存 128 作为缓存块而提供的缓存块单元的纹理地址存储到连同缓存块一起提供的地址存储器 104 的纹理缓存块单元地址中，以校正被存储在地址存储器 104 中的纹理地址。

此外，存储控制部分 129 响应于来自输出部分 130 的请求而从地址存储器 104 读出纹理地址，并将所读出的纹理地址提供给输出部分 130。之后，存储控制部分 129 清除存储在地址存储器 104 中的被读出纹理地址。

输出部分 130 向存储控制部分 129 发出以像素顺序读出从添加部分 77 所输出的转换后像素的 z 数据和写系数的请求，并发出读出纹理地址的请求。输出部分 130 对于从响应于请求而从切换部分 127 提供给它的 z 数据和写系数，将 z 数据输出到另一 DME 等，并将写系数输出到添加部分 77。此外，输出部分 130 将响应于请求而从存储控制部分 129 提供给它的纹理地址的整数部分提供给 RAM 模块 75，并将纹理地址的小数部分提供给插补操作部分 76。

现在，参照图 9 描述通过地址产生部分 79 的产生纹理地址的地址产生过程。该地址产生过程例如在用户执行建模操作时开始。要注意，为与转换后的帧单元图像相对应的每一个模型执行图 9 的地址产生过程。

在步骤 S31，CPU 121 响应于用户的建模操作而以预定顺序产生构成与转换后图像相对应的模型的多边形的顶点数据，并将所产生的顶点数据提供给顶点操作部分 122。

在步骤 S31 的过程之后，处理前进到步骤 S32，在该步骤，顶点操作部分 122 基于从 CPU 121 所提供的顶点数据对每个多边形进行顶点操作。顶点操作部分 122 将通过顶点操作所得到的多边形单元的顶点数据提供给 DDA 设置部分 123。

在步骤 S32 的过程之后，处理前进到步骤 S33，在该步骤 S33，DDA 设置部分 123 基于来自顶点操作部分 122 的多边形单元的顶点数据计算 DDA 124 所必需的参数。DDA 设置部分 123 将多边形单元的顶点数据和参数提供给 DDA 124。

在步骤 S33 的过程之后，处理前进到步骤 S34，在该步骤，DDA 124 基于从 DDA 设置部分 123 所提供的参数将多边形单元的顶点数据转换成像素数据，并将像素数据提供给深度测试部分 125。

在步骤 S34 的过程之后，过程前进到步骤 S35，在该步骤，深度测试部分 125 向 z 数据缓存 126 发出读出与来自 DDA 124 的像素数据相对应的像素的 z 数据的请求。然后，深度测试部分 125 获得响应于请求所读出的 z 数据。

在步骤 S35 的过程之后，过程前进到步骤 S36，在该步骤，深度测试部分 125 比较在步骤 S35 所获得的 z 数据与包括在来自 DDA 124 的像素数据中的 z 数据，以执行判断新多边形相对于旧多边形是否位于虚视点侧的深度测试。

在步骤 S36 的过程之后，过程前进到步骤 S37，在该步骤，深度测试部分 125 判断新多边形相对于旧多边形是否位于虚视点侧。

如果在步骤 S37 判断新多边形相对于旧多边形位于虚视点侧，则深度测试部分 125 将包括在与新多边形相对应的像素数据中的 z 数据和写系数以及 z 数据像素单元地址提供给 z 数据缓存 126。此外，深度测试部分 125 将纹理地址和纹理像素单元地址提供给地址缓存 128。

之后，在步骤 S38，z 数据缓存 126 与 z 数据像素单元地址相关联地存储来自深度测试部分 125 的 z 数据和写系数，以用新多边形的 z 数据和写系数更新旧多边形的 z 数据和写系数。

在步骤 S38 的过程之后，过程前进到步骤 S39，在该步骤，地址

缓存 128 与纹理像素单元地址相关联地存储纹理地址。

当在步骤 S37 判断新多边形相对于旧多边形不位于虚视点侧时，即，当旧多边形相对于新多边形位于虚视点侧时，或者在步骤 S39 的过程之后，过程前进到步骤 S40。在步骤 S40，深度测试部分 125 判断是否已经为所有与一帧的模型相对应且从 DDA 124 所提供的 z 数据执行了深度测试。如果判断还没有为所有 z 数据执行深度测试，则过程返回到步骤 S35，使得类似地执行从步骤 S35 开始的步骤的过程。

另一方面，如果在步骤 S40 判断已经为所有 z 数据执行了深度测试，则过程前进到步骤 S41，在该步骤，z 数据缓存 126 和地址缓存 128 执行缓存块的清除。

具体地，z 数据缓存 126 通过切换部分 127 将存储在缓存块区域中的缓存块和与缓存块相关联地存储的 z 数据缓存块单元地址提供给 z 数据存储单元 102 或 103。

此外，地址缓存 128 将存储在缓存块区域中的缓存块和与缓存块相关联地存储的 z 数据缓存块单元地址通过存储控制部分 129 提供给地址存储器 104。

要注意，这时，z 数据缓存 126 和地址缓存 128 可以执行所有缓存块的清除或者可以执行已经对其进行了更新的缓存块的清除。

在图 9 的地址产生过程之后，输出部分 130 以从添加部分 77 输出像素的顺序从切换部分 127 获得像素的 z 数据和写系数。然后，输出部分 130 将 z 数据输出到另一 DME 等，并将写系数提供给添加部分 77。此外，输出部分 130 从存储控制部分 129 获得像素的纹理地址，并将纹理地址的整数部分提供给 RAM 模块 75，同时它将纹理地址的小数部分提供给插补操作部分 76。

现在，参照图 10 到 12 描述图 8 所示的 CPU 121 产生多边形顶点数据的顺序。

图 10 示出使用与图 3 所示的相同的方格形式的网格的模型 10。

现在参照图 10，缓存块由 4×4 像素的 z 数据和写系数构成。这类似地也适用于以下参照图 11 描述的另一模型。此外，在图 10 中，所

示的模型 10 不处于图像转换后状态。

在图 10 中，CPU 121 将分别包括 12 个在竖直方向上并置的多边形的多边形组以这些多边形组在水平方向上并置的顺序相继地确定为用于产生顶点数据的产生过程的对象块（这种块以下称作处理对象块）。然后，CPU 121 以多边形在竖直方向上并置的顺序产生多边形单元的构成处理对象块的多边形的顶点数据。

要注意，在下面的描述中，与将被确定为处理对象块的多边形组的顺序相对应的方向线（在图 10 的实例中，水平方向）被适当的称作基本移动线。此外，与处理对象块中被相继确定为产生过程对象的多边形顺序相对应的方向（在图 10 的实例中，竖直方向）以下称作移动方向。

这里，形成处理对象块的多边形的数量（该数量以下称作移动量）被确定为使得，当为处理对象块执行产生过程时，位于当前对象处理块与紧邻的前一处理对象块之间的边界上并且对应于该紧邻的前面对象块中多边形的像素数据可以保留在 z 数据缓存 126 中，即可以不被从 z 数据缓存 126 中清除。

具体地，例如，如果缓存块包括总共 $x \times x$ 个像素的 z 数据和写系数，其中， x 个像素在水平方向和竖直方向上并置，并且 N 个缓存块可以被存储到 z 数据缓存 126 中，则在建模坐标系上像素的水平方向和竖直方向上的长度是 1 的情况下，移动量的最大值由下式 (1) 表示：

$$M = \text{floor} \left((x \times N - \text{Margin}) / \text{Mesh_Size} \right) \quad (1)$$

其中，函数 $\text{floor}()$ 代表不超过括号中的值的最大整数，并且 Mesh_Size 是模型 10 在竖直方向上的距离，即多边形 11 在竖直方向上的长度。此外， Margin 是预先确定的值，使得即使在执行图像的转换的情况下，与位于当前对象处理块与紧邻的前一处理对象块之间边界的那些多边形相对应的像素数据可以不被从 z 数据缓存 126 中清除。

具体地，在上述式子 (1) 中，基于其中不执行图像转换的情况确定最大值 M ，并且在考虑其他其中执行图像转换的情况下，减去由 Margin 所给出的量。

等于或低于以上述方式所确定的最大值 M 的数被确定为移动量。要注意，随着移动量增加，缓存的命中效率增加。

在图 10 中，如上所述，最大值 M 是 12，移动量是 12。

在这种情况下，CPU 121 首先将总共包括 12 个在竖直方向上并置的多边形 11_1 、 11_2 、 11_{n1+1} 、 11_{n1+2} 、 11_{2n1+1} 、 11_{2n1+2} 、 11_{3n1+1} 、 11_{3n1+2} 、 11_{4n1+1} 、 11_{4n1+2} 、 11_{5n1+1} 、 11_{5n1+2} 的多边形组确定为处理对象块 150。然后，CPU 121 以所提到的多边形顺序在多边形 11 的单元中进行产生过程。

这里，顶点操作部分 122、DDA 设置部分 123 和 DDA 124 以流水线方式处理通过从 CPU 121 所输出的过程产生的顶点数据。因此，与多边形 11 相对应的图像数据被以多边形 11_1 、 11_2 、 11_{n1+1} 、 11_{n1+2} 、 11_{2n1+1} 、 11_{2n1+2} 、 11_{3n1+1} 、 11_{3n1+2} 、 11_{4n1+1} 、 11_{4n1+2} 、 11_{5n1+1} 、 11_{5n1+2} 的顺序提供给深度测试部分 125。

换句话说，像素 15_1 、 15_2 、 15_{m1+1} 、 15_{m1+2} 、 15_{2m1+1} 、 15_{2m1+2} 、 15_{3m1+1} 、 15_{3m1+2} 、 15_{4m1+1} 、 15_{4m1+2} 、 15_{5m1+1} 、 15_{5m1+2} 、 15_{6m1+1} 、 15_{6m1+2} 、 15_{7m1+1} 、 15_{7m1+2} 的图像数据以该顺序被提供给深度测试部分 125。因此，深度测试部分 125 以这个顺序从 z 数据缓存 126 中读出像素 15 的 z 数据，并对 z 数据进行深度测试。

在这种情况下， z 数据缓存 126 首先通过切换部分 127 从 z 数据存储单元 102 或 103 中读出由作为缓存块的方形 12 包围的像素 15_1 到 15_4 、 15_{m1+1} 和 15_{m1+4} 、 15_{2m1+1} 到 15_{2m1+4} 和 15_{3m1+1} 到 15_{3m1+4} 的 z 数据，并存储所读出的 z 数据。然后，深度测试部分 125 相继地分别比较像素 15_1 、 15_2 、 15_{m1+1} 、 15_{m1+2} 、 15_{2m1+1} 、 15_{2m1+2} 、 15_{3m1+1} 、 15_{3m1+2} 的 z 数据与包括在从 DDA 124 所提供的像素 15_1 、 15_2 、 15_{m1+1} 、 15_{m1+2} 、 15_{2m1+1} 、 15_{2m1+2} 、 15_{3m1+1} 、 15_{3m1+2} 中的 z 数据，以进行深度测试。

然后， z 数据缓存 126 从 z 数据存储单元 102 或 103 中读出由作为缓存块的方形 161 包围的像素 15_{4m1+1} 到 15_{4m1+4} 、 15_{5m1+1} 到 15_{5m1+4} 、 15_{6m1+1} 到 15_{6m1+4} 、 15_{7m1+1} 到 15_{7m1+4} 的 z 数据，并存储所读出的 z 数据。然后，深度测试部分 125 相继地将像素 15_{4m1+1} 、 15_{4m1+2} 、 15_{5m1+1} 、

15_{5m1+2} 、 15_{6m1+1} 、 15_{6m1+2} 、 15_{7m1+1} 、 15_{7m1+2} 的 z 数据与包括在从 DDA 124 所提供的像素 15_{4m1+1} 、 15_{4m1+2} 、 15_{5m1+1} 、 15_{5m1+2} 、 15_{6m1+1} 、 15_{6m1+2} 、 15_{7m1+1} 、 15_{7m1+2} 的像素数据中的 z 数据分别进行比较, 以进行深度测试。

之后, 深度测试部分 125 利用与通过产生过程所产生的像素数据相对应的像素数据执行深度测试, 其中将在水平方向上与处理对象块 150 相邻的处理对象块确定为处理对象。但是, 移动量被确定为使得与位于当前处理对象块 151 与紧邻的前一处理对象块 150 之间边界上的多边形 11_2 、 11_{n1+2} 、 11_{2n1+2} 、 11_{3n1+2} 、 11_{4n1+2} 、 11_{5n1+2} 相对应的像素数据可以不被从 z 数据缓存 126 中清除, 即与方形 12 和 161 相对应的缓存块可以如上所述地不被从 z 数据缓存 126 中清除。因此, z 数据缓存 126 只需利用两个缓存块执行深度测试, 而不必读出新的缓存块。

此外, 深度测试部分 125 随后利用与由产生过程所产生的顶点数据相对应的像素数据进行深度测试, 其中在水平方向上与处理对象块 151 相邻的处理对象块 152 被确定为处理对象。但是, 因为移动量被确定为使得与位于当前处理对象块 152 和紧邻的前一处理对象块 151 之间边界上的多边形 11 相对应的像素数据可以不如上所述地被从 z 数据缓存 126 中清除, z 数据缓存 126 只需重新读出由作为缓存块的方形 13 和 162 包围的像素 15 的 z 数据和写系数, 而不必读出由作为缓存块的方形 12 和 161 包围的像素 15 的 z 数据和写系数。

如上所述, CPU 121 将分别包括 12 个在竖直方向上并置的多边形的多边形组按照这些多边形组在水平方向上并置的顺序相继确定为处理对象块, 然后按照这些多边形在竖直方向上并置的顺序产生多边形单元的构成处理对象块的多边形的顶点数据。因此, z 数据缓存 126 不需要将每个缓存块读出多次。换句话说, z 数据缓存 126 的命中效率被增强。

因此, 同样在多边形的大小被减小以进行高精确度的转换时, 图像处理系统 51 不需要增加 z 数据存储 102 和 103 的带宽, 并且可以实现高精确度的带宽, 同时功耗或生成成本的增加被抑制。

图 11 示出了表示在执行以顺时针方向旋转由图 10 的模型 10 所表示的图像的转换时的转换后图像的模型 170。

在图 11 中, CPU 121 将分别包括 12 个在由图 11 中的箭头标记 b 所指示的方向上并置的多边形的多边形组以这些多边形组在垂直于箭头标记 b 的方向的、由另一箭头标记 a 指示的方向上并置的顺序相继确定为处理对象块。然后, CPU 121 以多边形在箭头标记 b 的方向上并置的顺序产生多边形单元的构成处理对象块的多边形的顶点数据。换句话说, 在图 11 中, 基本移动线是箭头标记 a 的方向上的线, 而移动方向是箭头标记 b 的方向。

这里假设, 图 11 中的移动量是根据上面给出的式子 (1) 所确定的最大值 M。要注意, 在这种情况下, 式子 (1) 的 Mesh_Size 是在箭头标记 b 的方向上模型 170 的间距, 即多边形 171 在箭头标记 b 的方向上的长度。

在这种情况下, CPU 121 首先将在竖直方向上并置的总共 12 个多边形 171_1 、 171_2 、 171_{n6+1} 、 171_{n6+2} 、 171_{2n6+1} 、 171_{2n6+2} 、 171_{3n6+1} 、 171_{3n6+2} 、 171_{4n6+1} 、 171_{4n6+2} 、 171_{5n6+1} 、 171_{5n6+2} 确定为处理对象块 180, 并以这个顺序执行多边形 171 的单元中的产生过程。但是, 因为顶点操作部分 122 执行剪贴, 所以与多边形 171 相对应的顶点数据被以多边形 171_{n6+2} 、 171_{2n6+1} 、 171_{2n6+2} 、 171_{3n6+1} 、 171_{3n6+2} 、 171_{4n6+1} 、 171_{4n6+2} 、 171_{5n6+1} 、 171_{5n6+2} 的顺序提供给 DDA 124。

相应地, DDA 124 将多边形 171 以多边形 171_{n6+2} 、 171_{2n6+1} 、 171_{2n6+2} 、 171_{3n6+1} 、 171_{3n6+2} 、 171_{4n6+1} 、 171_{4n6+2} 、 171_{5n6+1} 、 171_{5n6+2} 的顺序分别变换为相应像素 172_3 、 172_4 、 172_{m2+3} 、 172_{2m2+2} 、 172_{2m2+3} 、 172_{3m2+2} 、 172_{3m2+3} 、 172_{4m2+2} 、 172_{5m2+1} 、 172_{5m2+2} 的像素数据。然后, DDA 124 将所得到的像素数据提供给深度测试部分 125。

深度测试部分 125 从 z 数据缓存 126 中以从 DDA 124 将 z 数据提供给它的顺序读出像素 172 的 z 数据, 并执行深度测试。

这时, z 数据缓存 126 首先通过切换部分 127 从 z 数据存储器 102 或 103 中读出由作为缓存块的方形 173 所包围的像素 172_1 到 172_4 、

172_{m2+1} 到 172_{m2+4} 、 172_{2m2+1} 到 172_{2m2+4} 和 172_{3m2+1} 到 172_{3m2+4} 的 z 数据，并存储所读出的 z 数据。然后，深度测试部分 125 相继地将像素 172_3 、 172_4 、 172_{m2+3} 、 172_{2m2+2} 、 172_{2m2+3} 、 172_{3m2+2} 、 172_{3m2+3} 与包括在从 DDA 124 所提供的像素 172_3 、 172_4 、 172_{m2+3} 、 172_{2m2+2} 、 172_{2m2+3} 、 172_{3m2+2} 、 172_{3m2+3} 的像素的像素数据中的 z 数据分别进行比较，以进行深度测试。

然后， z 数据缓存 126 从 z 数据存储器 102 或 103 中读出由作为缓存块的方形 174 所包围的像素 172_{4m2+1} 到 172_{4m2+4} 、 172_{5m2+1} 到 172_{5m2+4} 、 172_{6m2+1} 到 172_{6m2+4} 和 172_{7m2+1} 到 172_{7m2+4} 的 z 数据，并存储所读出的 z 数据。然后，深度测试部分 125 相继地将像素 172_{4m2+4} 、 172_{5m2+1} 、 172_{5m2+2} 的 z 数据与包括在从 DDA 124 所提供的像素 172_{4m2+2} 、 172_{5m2+1} 、 172_{5m2+2} 的像素数据中的 z 数据分别进行比较，以进行深度测试。

之后，深度测试部分 125 利用与通过产生过程所产生的顶点数据相对应的像素数据进行深度测试，其中，在由箭头标记 a 所指示的方向上与处理对象块 180 相邻的处理对象块 181 被确定为处理对象。但是，因为移动量被确定为使得与位于当前处理对象块 181 与紧邻的前一处理对象块 182 之间边界上的多边形 171 相对应的像素数据可以不被从 z 数据缓存 126 中清除，所以 z 数据缓存 126 只需读出重新被作为缓存块的方形 175 和 176 所包围的像素 172 的 z 数据和写系数，而不必读出由作为缓存块的方形 173 和 174 所包围的像素 172 的 z 数据和写系数。

如上所述，同样在图像处理系统 51 执行图像转换的情况下，CPU 121 将分别包括 12 个在箭头标记 b 所指示的方向上并置的多边形的多边形组以这些多边形组在由箭头标记 a 所指示的方向上被并置的顺序相继确定为处理对象块。然后，CPU 121 以多边形在箭头标记 b 的方向上并置的顺序产生多边形单元的构成处理对象块的多边形的顶点数据。从而， z 数据缓存 126 的命中效率被增强。

图 12 示出使用由多边形 201 所构成的同心网格的模型 200。

现在参照图 12, CPU 121 将分别包括 5 个在径向上并置的多边形 201 的多边形组以这些多边形组在圆周方向上并置的顺序相继确定为处理对象块。然后, CPU 121 以多边形在径向上并置的顺序产生多边形单元的构成处理对象块的多边形的顶点数据。换句话说, 在图 12 中, 基本移动线是在圆周方向上延伸的线, 移动方向是径向。

这里假设, 图 12 中的移动量是根据上面给出的式子 (1) 所确定的最大值 M 。要注意, 在这种情况下, 式子 (1) 的 Mesh_Size 是投影方向上模型 200 的间距, 即多边形 201 在多边形的投影方向上的长度。

CPU 121 首先将分别包括 5 个在投影方向上并置的多边形 201 的多边形组确定为处理对象块 202, 并以多边形 201 在投影方向上并置的顺序在多边形 201 的单元中进行产生过程。

因此, 与构成处理对象块 202 的多边形 201 相对应的像素数据被以这些多边形 201 在投影方向被并置的顺序提供给深度测试部分 125。深度测试部分 125 以从 DDA 124 提供 z 数据的顺序从 z 数据缓存 126 中读出包括在像素数据中的 z 数据, 以进行深度测试。

这时, z 数据缓存 126 首先通过切换部分 127 从 z 数据存储器 102 或 103 中读出由作为缓存块的方形 210 所包围的像素的 z 数据, 并存储所读出的 z 数据。然后, 深度测试部分 125 相继地将与从 DDA 124 所提供的像素数据相对应的像素的 z 数据与包括在从 DDA 124 所提供的像素数据中的 z 数据分别进行比较, 以进行深度测试。

然后, z 数据缓存 126 从 z 数据存储器 102 或 103 中读出由作为缓存块的方形 211 所包围的像素的 z 数据, 并存储所读出的 z 数据。然后, 深度测试部分 125 相继地将与从 DDA 124 所提供的像素数据相对应的像素的 z 数据与包括在从 DDA 124 所提供的像素数据中的 z 数据分别进行比较, 以进行深度测试。

之后, 深度测试部分 125 利用与通过产生过程所产生的顶点数据相对应的像素数据执行深度测试, 其中, 在圆周方向上与处理对象块 202 相邻的处理对象块 203 被确定为处理对象。但是, 因为移动量被

确定为使得与位于当前处理对象块 203 和紧邻的前一处理对象块 202 之间边界上的多边形 201 相对应的像素数据可以不被从 z 数据缓存 126 中清除, 所以 z 数据缓存 126 不必读出重新由作为缓存块的方形 210 和 211 所包围的像素的 z 数据和写系数。

如上所述, 同样在模型由同心网格模型 200 构成的情况下, CPU 121 将分别包括 5 个在投影方向上并置的多边形的多边形组以这些多边形组在圆周方向上并置的顺序相继确定为处理对象块。然后, CPU 121 以这些多边形在投影方向上并置的顺序产生多边形单元的构成处理对象块的多边形的像素数据。因此, z 数据缓存 126 的命中效率被增强。

要注意, 虽然以上参照图 10 到 12 描述了在深度测试中用于 z 数据的 z 数据缓存 126 的命中效率, 但是类似地增强了存储纹理地址中地址缓存 128 的命中效率。

此外, 通过上面给出的式子 (1), 基于其中上述没有执行图像转换的情况确定最大值 M。因此, 当通过顶点操作部分 122 在建模变换中执行放大或缩小时, 必需进行修正。具体地, 在执行放大的情况下, 必需减少移动量, 但是在执行缩小的情况下, 必需增加移动量。

相应地, 如果放大/缩小比例由 S 表示, 则移动量的最大值 M 可以通过将其乘以放大/缩小比例 S 的倒数而通过下式 (2) 来表示:

$$M = \text{floor} \left((x \times N - \text{Margin}) / \text{Mesh_Size} / S \right) \quad (2)$$

要注意, 如果在式子 (2) 中不涉及放大或缩小, 则因为放大/缩小比例 S 为 1, 所以式子 (2) 等同于式子 (1)。因此, 同样在不涉及放大或缩小时, 最大值 M 满足式子 (2)。换句话说, 不管是否执行放大或缩小, 最大值 M 都可以由式子 (2) 表示。

现在, 参照图 13 描述 CPU 121 所执行的将顶点数据输出到顶点操作部分 122 的输出过程。该输出过程对应于图 9 的步骤 S31。

在步骤 S51, CPU 121 将第一基本移动线上的第一多边形组确定为处理对象块。然后, 过程前进到步骤 S52。

在步骤 S52, CPU 121 从处理对象块中的多边形中将移动方向上

的第一多边形确定为处理对象。然后，过程前进到步骤 S53。

在步骤 S53，CPU 121 执行产生多边形单元中处理对象的多边形的顶点数据的产生过程，并将所得到的顶点数据提供给顶点操作部分 122。顶点数据被用于图 9 的步骤 S32 的顶点操作。

在步骤 S53 的过程之后，过程前进到步骤 S54，在该步骤，CPU 121 判断当前处理对象块的产生过程是否完成，即是否产生当前处理对象块的所有多边形的顶点数据。然后，如果判断当前处理对象块的产生过程还没有完成，则过程前进到步骤 S55。

在步骤 S55，CPU 121 从当前处理对象块中的多边形中将当前处理对象多边形在移动方向上的下一个多边形确定为新的处理对象。然后，过程返回步骤 S53，使得如上描述的过程被重复。

另一方面，如果在步骤 S54 判断当前处理对象块的产生过程完成，即如果当前处理对象是处理对象块中移动方向上的最后一个多边形，则过程前进到步骤 S56。在步骤 S56，CPU 121 判断当前处理对象块是否是当前基本移动线上的最后一个处理对象块。

如果在步骤 S56 判断当前处理对象块不是当前基本移动线上的最后一个处理对象块，则过程前进到步骤 S57。在步骤 S57，CPU 121 将当前基本移动线上的下一个多边形组判断为处理对象块。之后，过程返回到步骤 S53。

另一方面，如果在步骤 S56 判断当前处理对象块是当前基本移动线上的最后一个处理对象块，则过程前进到步骤 S58。在步骤 S58，CPU 121 判断当前基本移动线是否是最后一个基本移动线。

如果在步骤 S58 判断当前基本移动线不是最后一个基本移动线，则过程前进到步骤 S59，在该步骤，CPU 121 将下一基本移动线的第一多边形组确定为处理对象块。

另一方面，如果在步骤 S58 判断当前基本移动线是最后一个基本移动线，则过程结束。

现在，参照图 14 描述包括 DME 11 的图像处理装置 250。

图像处理装置 250 包括根据存储在 ROM（只读存储器）252 中

或者记录在记录部分 258 中的程序执行各种过程的 CPU (中央处理单元) 251 和 DME 11。将由 CPU 121 执行的程序、将由 CPU 251 使用的数据等被适当地存储到 RAM (随机存取存储器) 253 中。DME 11、CPU 251、ROM 252 和 RAM 253 通过总线 254 相互连接。

此外,输入/输出接口 255 通过总线 254 被连接到 CPU 251。包括键盘、鼠标、麦克风、用于接收从未示出的远程控制器所发送的指令的接收部分等的输入部分 256 以及包括显示单元、扬声器等的输出部分 257 被连接到输入/输出接口 255。CPU 251 响应于从输入部分 256 所输入的指令执行各种过程。然后, CPU 251 将过程的结果输出到输出部分 257。

例如, CPU 251 响应于从输入部分 256 所输入的指令控制 DME 11 执行输入图像的缩小、放大、类型改变、旋转、向左和向右翻转、倒转或移动或者对输入图像应用特殊效果。然后, CPU 251 控制输出部分 257 以基于从 DME 11 所输出的图像显示图像。

连接到输入/输出接口 255 的记录部分 258 包括例如硬盘,并存储将由 CPU 251 执行的程序和各種数据。通信部分 259 通过诸如互联网或局域网的网络与外部装置通信。要注意,可以通过通信部分 259 获得记录在记录部分 258 中的程序。

连接到输入/输出接口 255 的驱动 260 在诸如磁盘、光盘、磁光盘或半导体存储器的可移动介质 261 被加载到其中时驱动可移动介质 261,以获得记录在可移动介质 261 上的程序或数据。所获得的程序或数据被传递给记录部分 258 并在必要时被其记录。

现在,参照图 15 描述包括 DME 11 的记录和再现装置 300 的实例。

记录和再现装置 300 包括根据存储在 ROM 306 中或记录在记录部分 305 上的程序执行各种过程的 CPU 301 和 DME 11。将由 CPU 301 执行的程序、数据等被适当地存储到 RAM 307 中。DME 11、CPU 301、ROM 306 和 RAM 307 通过总线相互连接。

此外,输入 I/F (接口) 309 和输出控制部分 310 通过总线被连接

到 CPU 301。输入部分 308 被连接到输入 I/F 309，并且包括键盘、鼠标、麦克风、用于接收从未示出的远程控制器等所发送的指令的接收部分、用于拾取图像拾取对象的图像的图像拾取部分等。同时，输出部分 311 被连接到输出控制部分 310，并且包括显示单元、扬声器等。CPU 301 响应于从输入部分 308 通过输入 I/F 309 输入给它的指令而执行各种过程。CPU 301 通过输出控制部分 310 将过程的结果输出到输出部分 311。

例如，CPU 301 响应于被从输入部分 308 输入给它的指令控制 DME 11 以执行输入图像的缩小、放大、类型改变、旋转、向左和向右翻转、倒转或移动，或者对输入图像应用特殊效果。而且，CPU 301 通过输出控制部分 310 控制输出部分 311 以基于从 DME 11 所输出的图像显示图像。

此外，编码/解码电路 302 以及记录和再现控制部分 304 通过总线被连接到 CPU 301。编码/解码电路 302 在必要时将例如通过输入部分 308 的图像拾取而得到的图像保留到缓冲存储器 303 中，并在 CPU 301 的控制下根据诸如 JPEG（联合图像专家组）或 MPEG（运动图像专家组）系统的预定编码系统对图像进行编码。然后，编码/解码电路 302 通过记录和再现控制部分 304 将由编码所得到的图像记录到记录部分 305 中。

记录和再现控制部分 304 在 CPU 301 的控制下控制记录部分 305 的记录和再现。具体地，记录和再现控制部分 304 控制记录部分 305 以记录从编码/解码电路 302 所提供的图像或者将从记录部分 305 所读出的图像提供给编码/解码电路 302。编码/解码电路 302 在 CPU 301 的控制下对来自记录和再现控制部分 304 的图像进行解码，并将通过解码所得到的图像例如作为输入图像而提供给 DME 11。

本发明可以被应用于例如 GPU（图形处理单元）。

要注意，在本说明书中，描述被记录在程序记录介质中的程序的步骤在顺序上可以、但不是必须以所述的顺序在时间上串行地被处理，并且包括被并行或单独执行而不是在时间上串行执行的过程。

此外，在本说明书中，术语“系统”被用于代表包括多个设备或装置的整个装置。

虽然已经使用具体术语描述了本发明的优选实施例，但是这种描述只是为了说明的目的，并且应该理解，可以进行改变和变化而不脱离下面权利要求的精神或范围。

图 1A

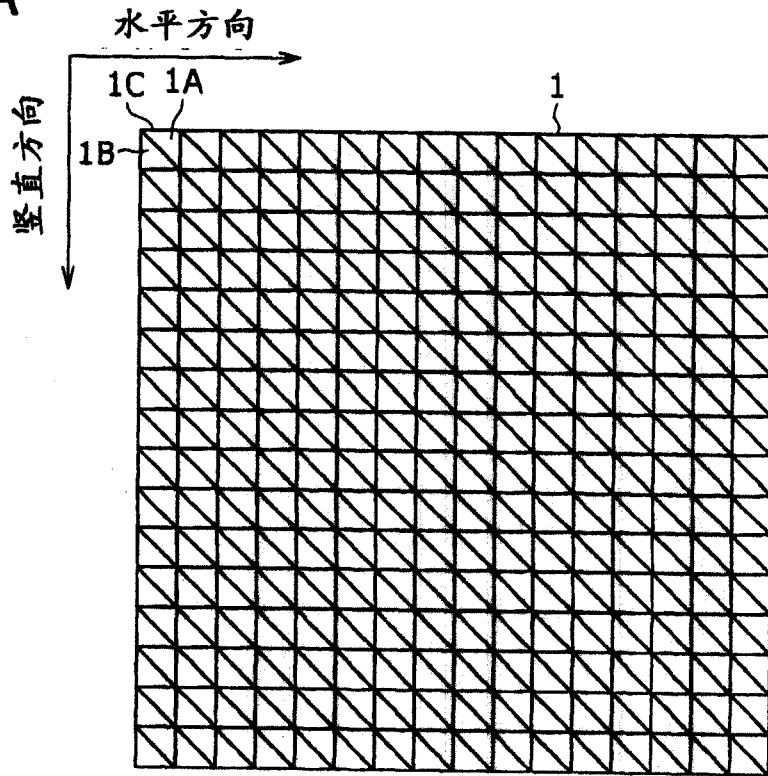


图 1B

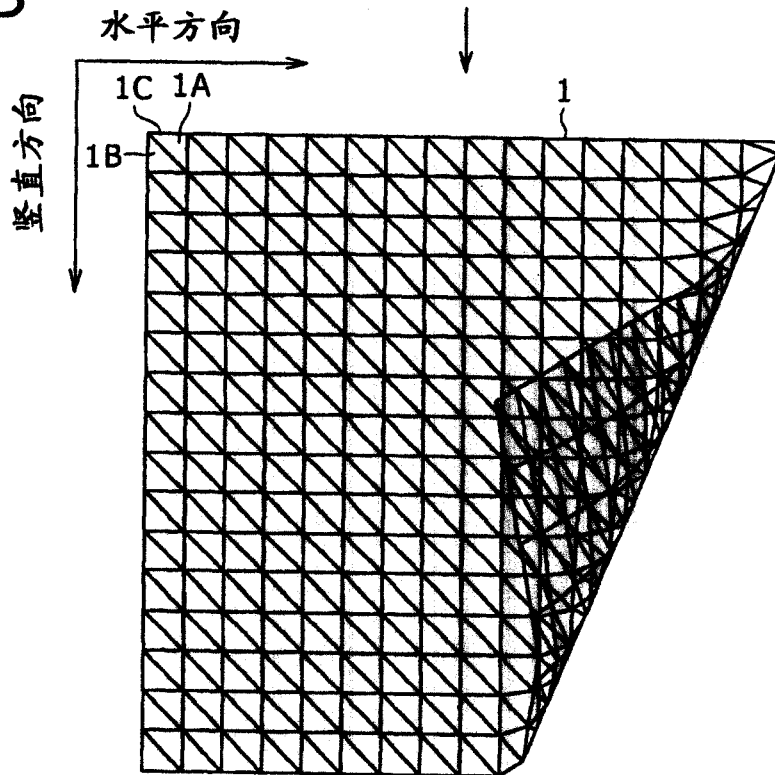


图 2A

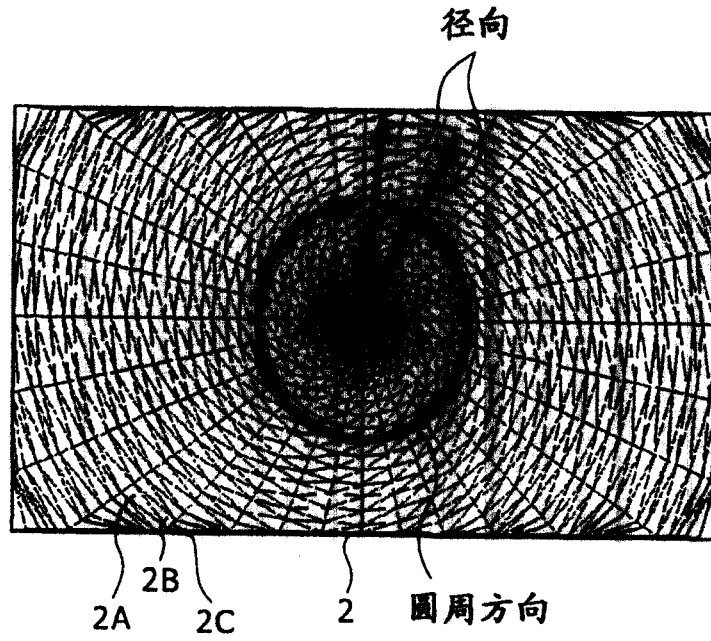
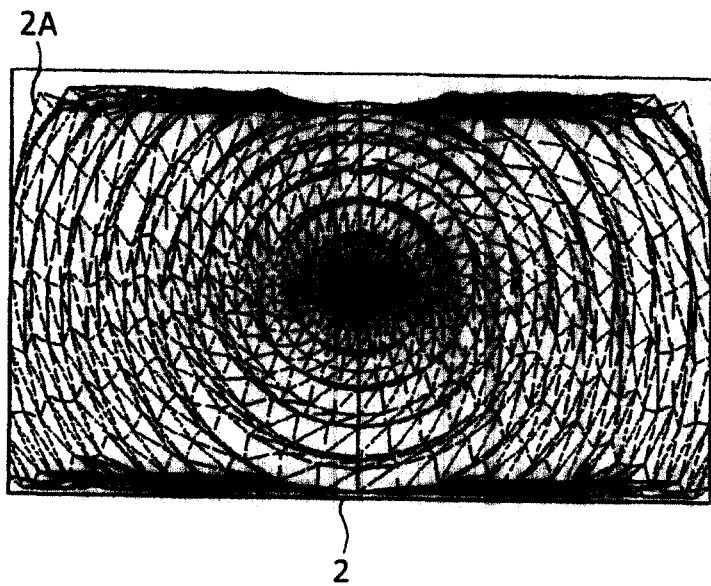


图 2B



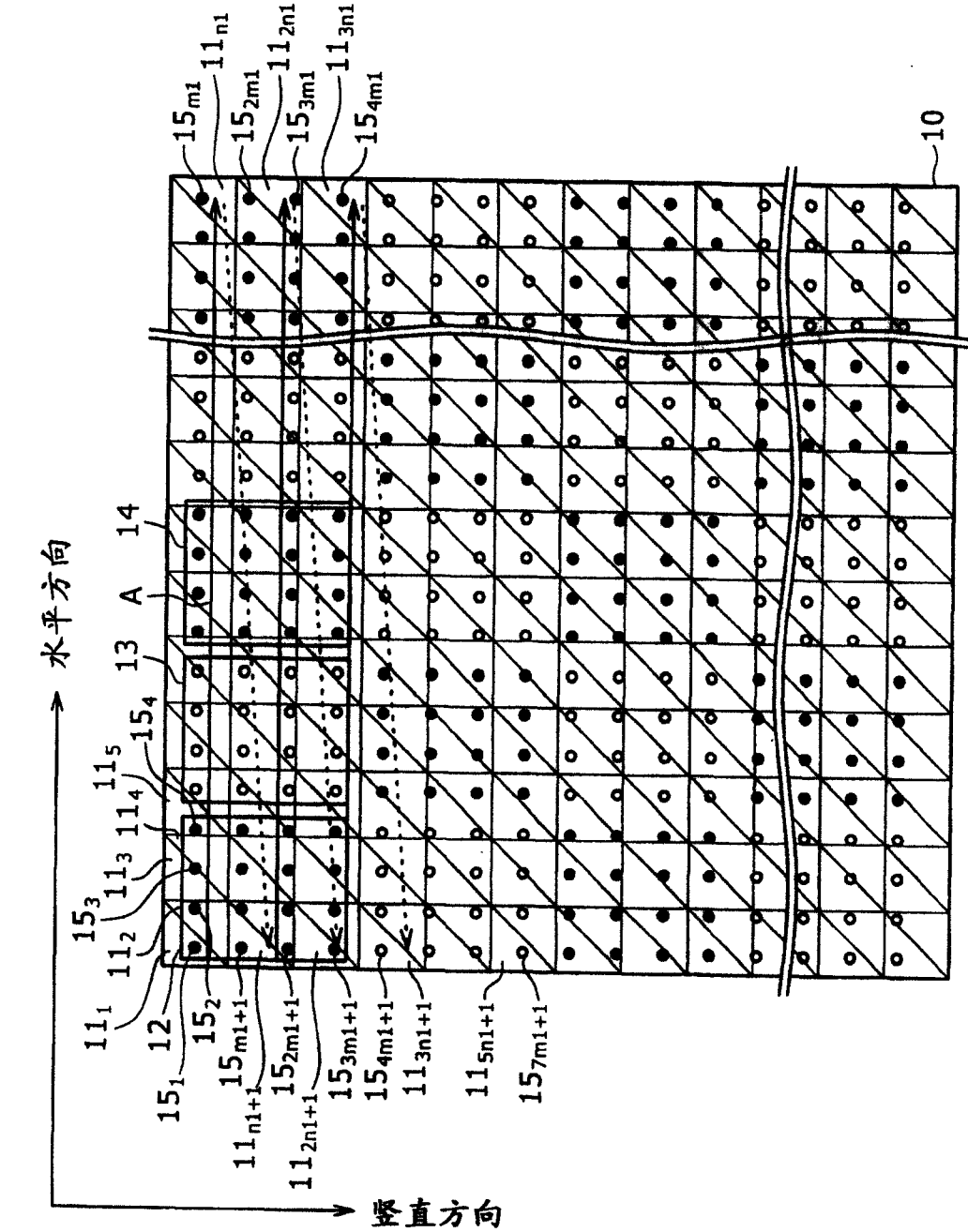


图 3

图 4

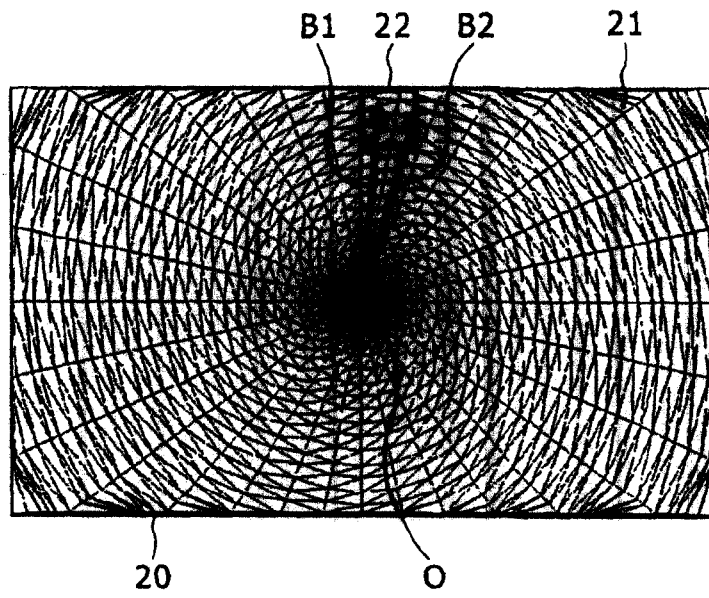


图 5

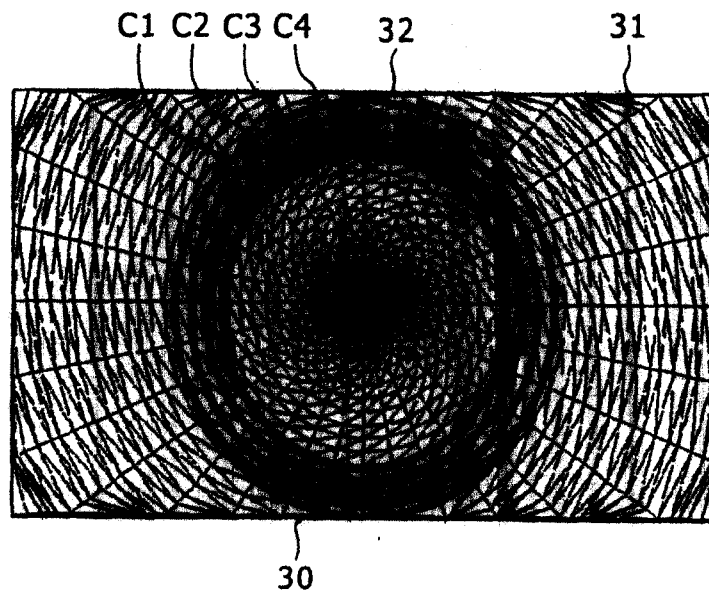


图 6

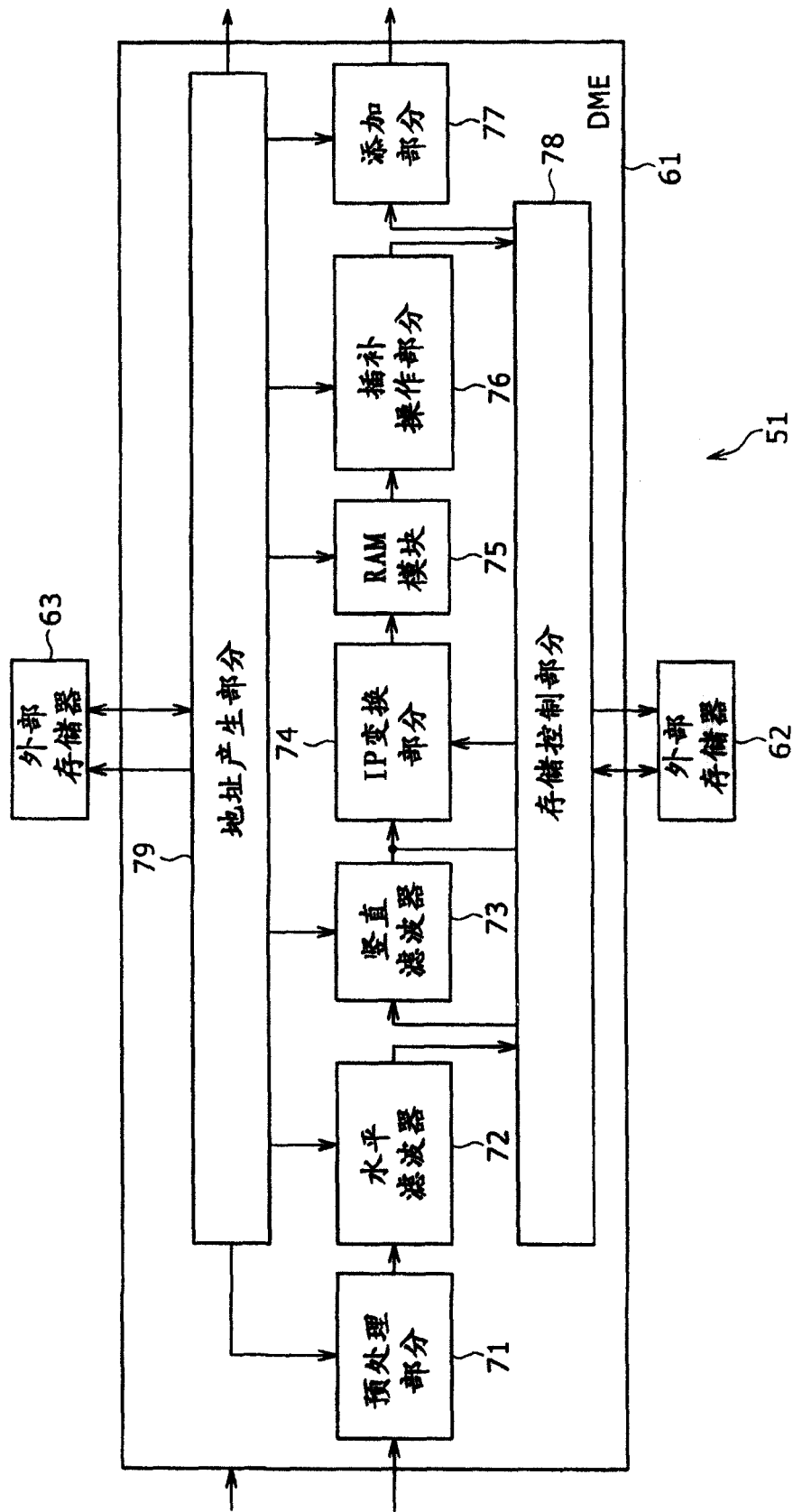


图 7

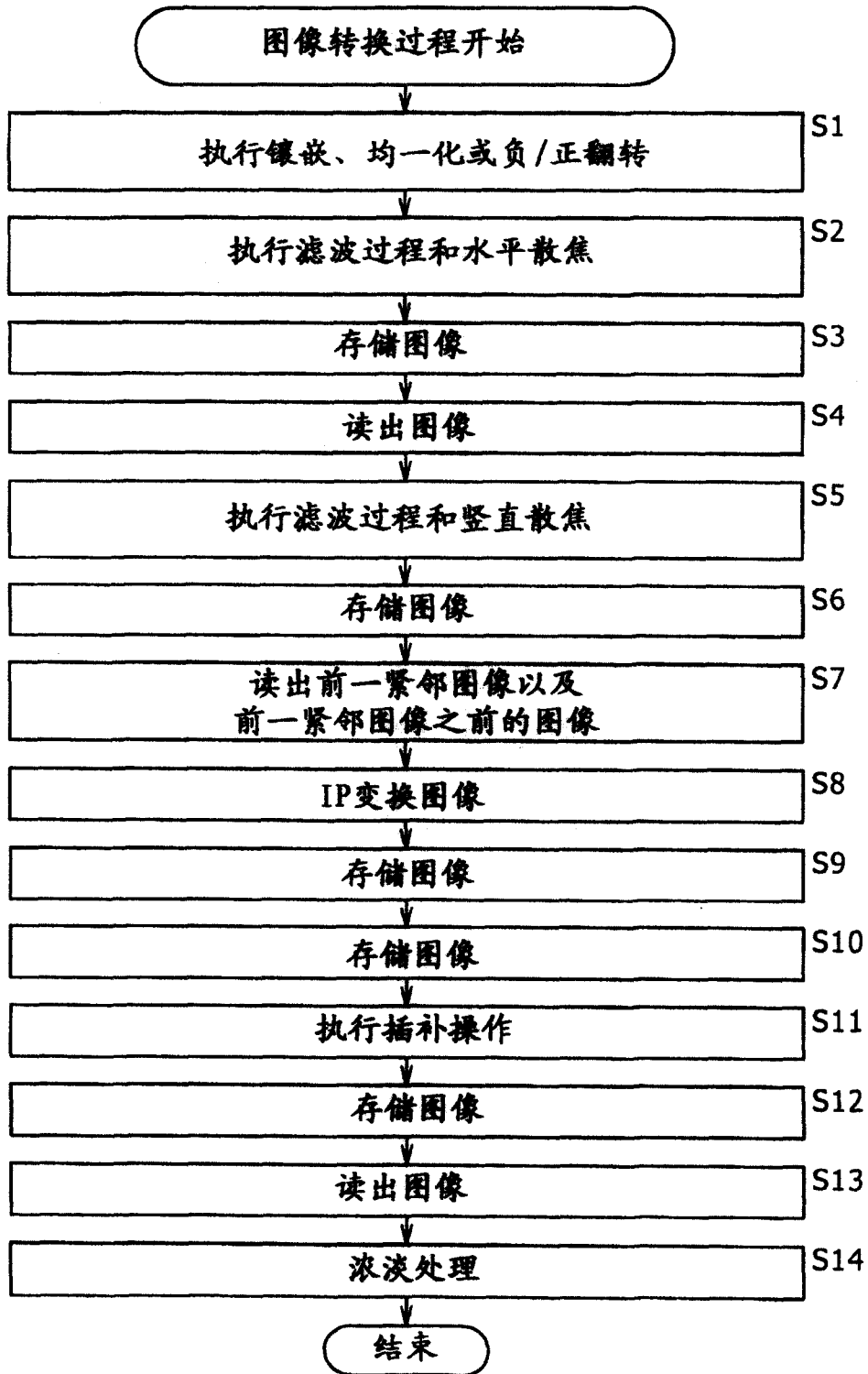


图 8

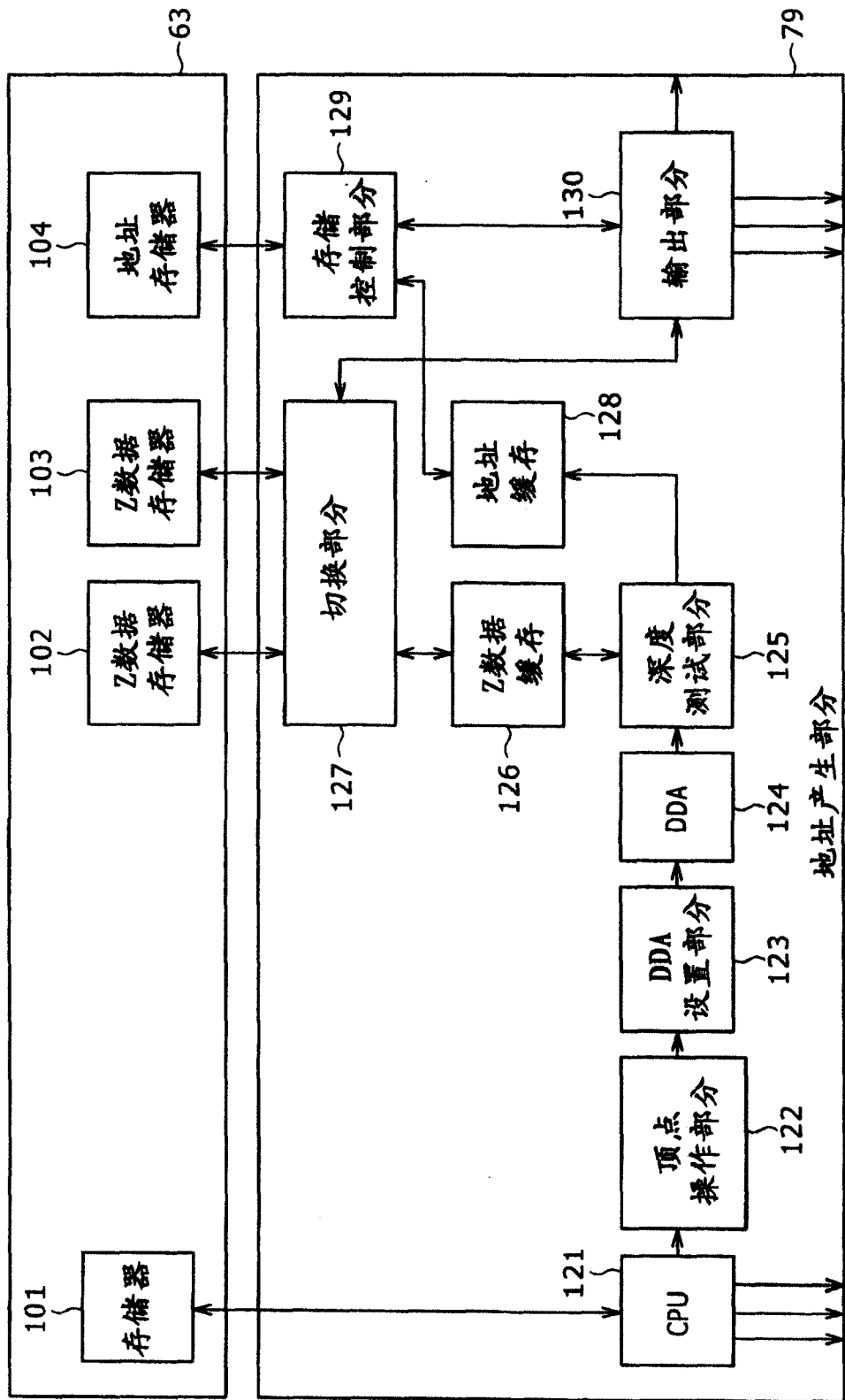


图 9

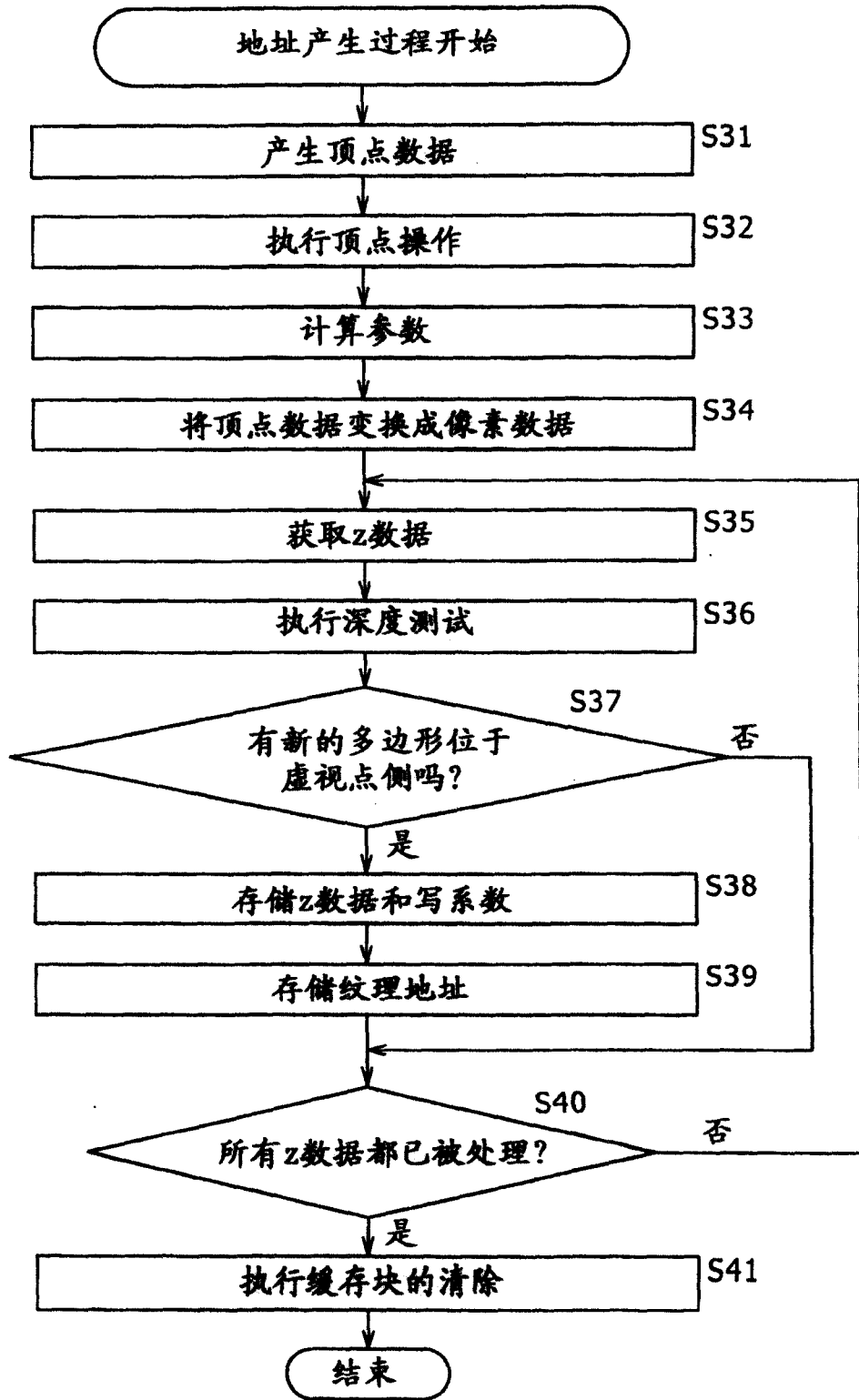
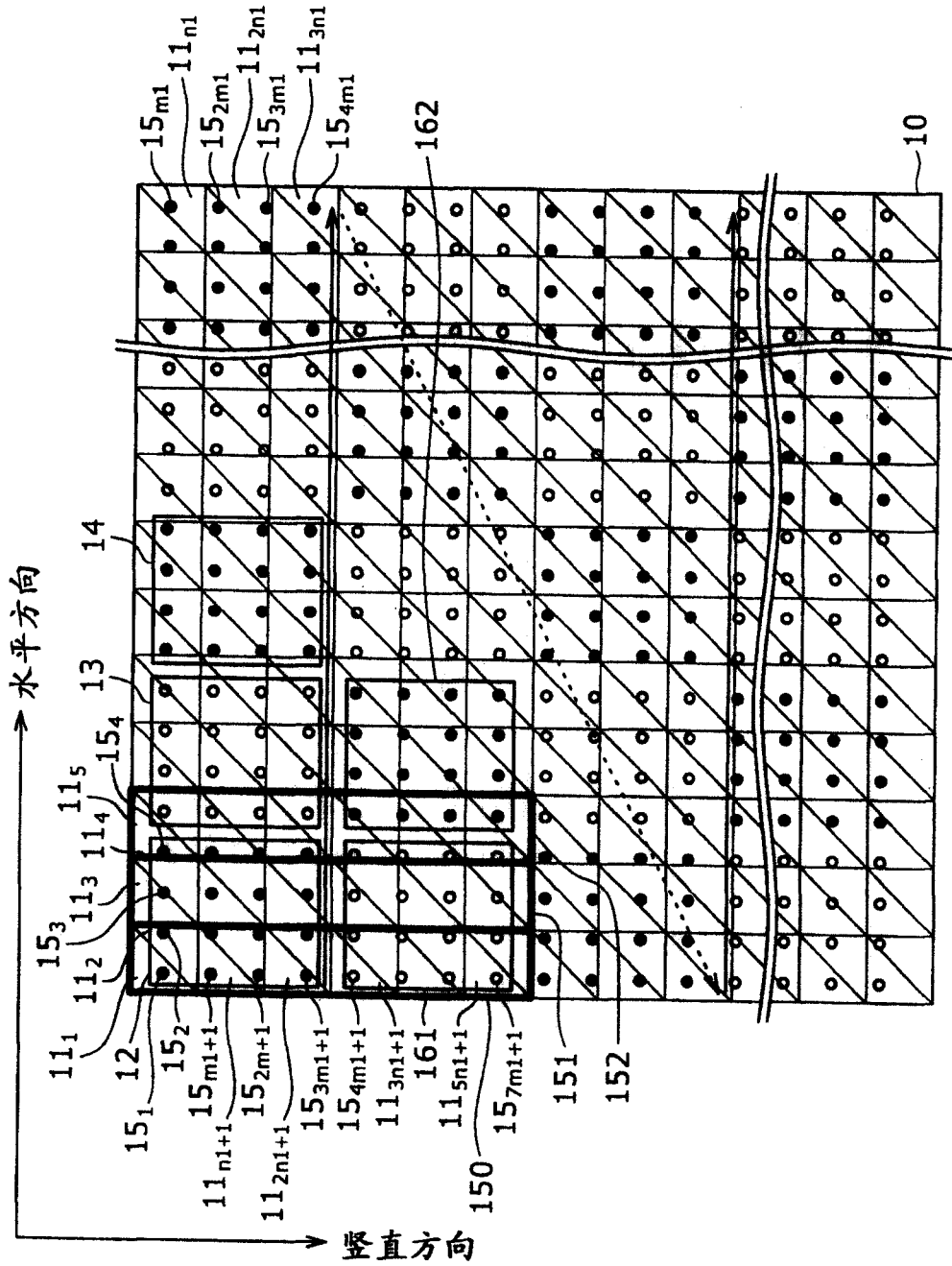


图 10



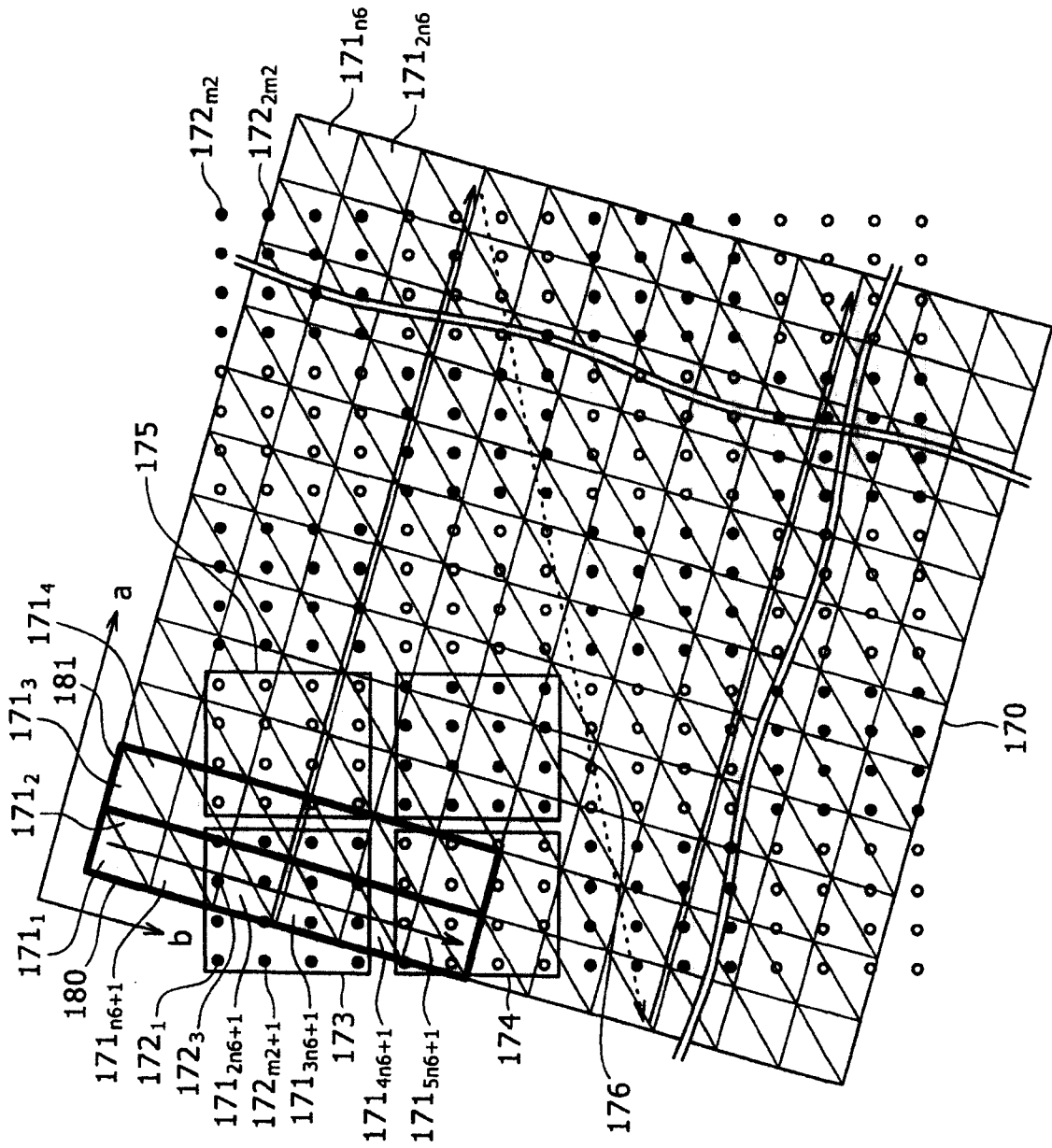


图 11

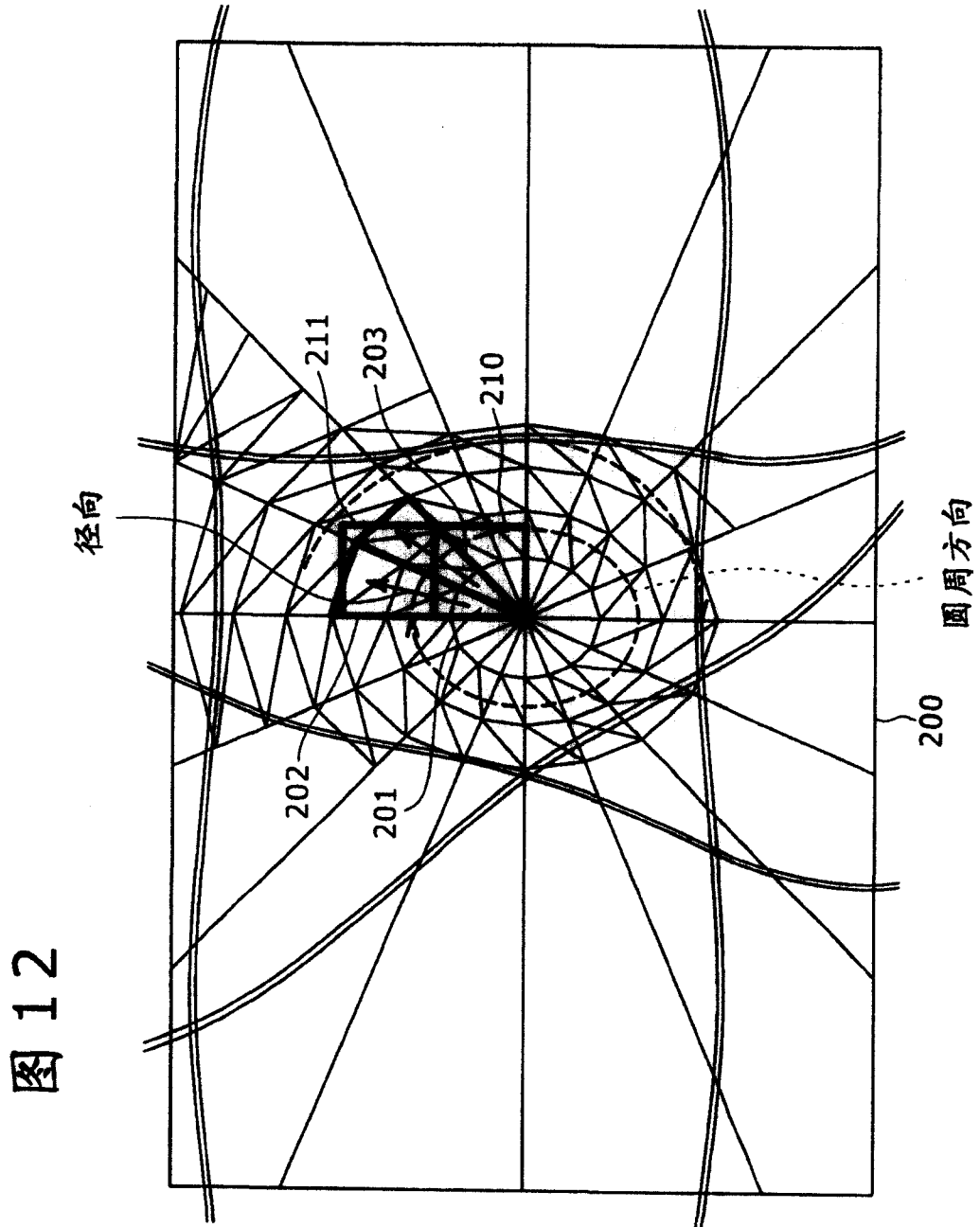


图 12

图 13

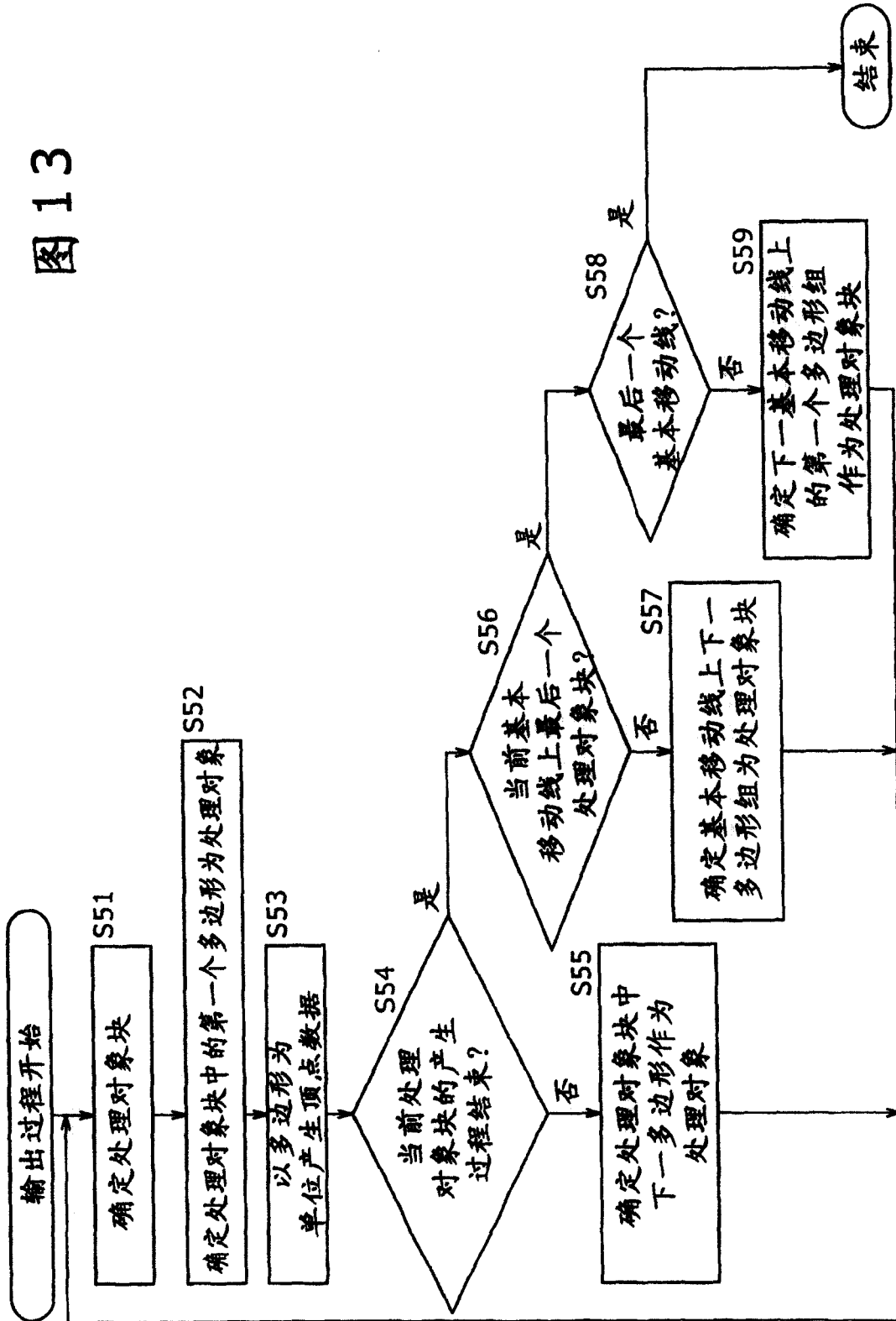


图 14

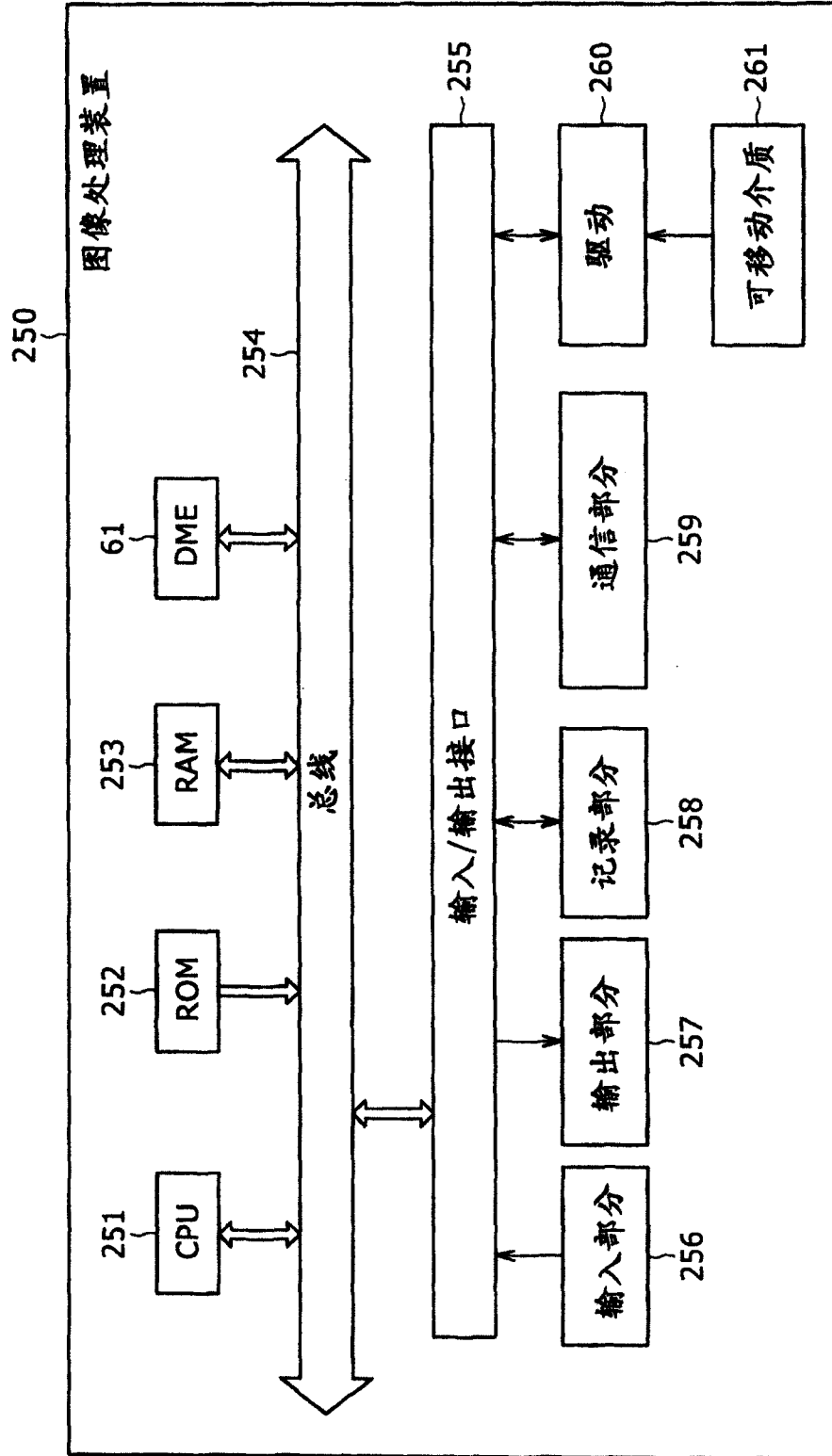


图 15

