

1. 一种图像处理设备,用于从多个视点绘制共同的物体,所述图像处理设备包括:

存储单元,其存储包括以预定顺序管理的多个虚拟物体的数据的场景数据,其中所述场景数据包括各虚拟物体的几何信息和材质信息;

第一单元,其绘制从第一视点看见的第一图像;以及

第二单元,其绘制从第二视点看见的第二图像,

其中,所述第一单元绘制所述第一图像的上部区域并且所述第二单元并行地绘制所述第二图像的下部区域,然后所述第一单元绘制所述第一图像的下部区域并且所述第二单元并行地绘制所述第二图像的上部区域,以及

所述第一单元在绘制所述第一图像的上部区域时按照所述预定顺序在所述场景数据中搜索能从所述第一视点看见的虚拟物体以获得该虚拟物体的生成所述第一图像的上部区域所用的几何信息和材质信息,并且通过将所述场景数据中的多个虚拟物体的顺序改变成改变后的顺序来生成改变后的场景数据,其中,能从所述第一视点看见的虚拟物体在所述改变后的顺序中位于顶部位置,

在所述第一单元绘制了所述第一图像的上部区域后所述第二单元绘制所述第二图像的上部区域的情况下,所述第二单元通过按照所述改变后的顺序在改变后的场景数据中搜索能从所述第二视点看见的虚拟物体以获得该虚拟物体的生成所述第二图像的上部区域所用的几何信息和材质信息,来绘制所述第二图像的上部区域。

2. 根据权利要求1所述的图像处理设备,其特征在于,还包括将虚拟空间中所包括的各虚拟物体的数据作为场景数据来管理的单元,

其中,所述第一单元和所述第二单元均是绘制单元,该绘制单元按照预定顺序,在所述场景数据中搜索从所设置的视点开始的视野中所包括的虚拟物体的数据,并且基于所搜索到的数据,绘制从所述所设置的视点开始的虚拟空间的图像;以及

所述第一单元基于在生成从第一视点观察到的虚拟空间的图像时执行的所述搜索的处理结果,通过改变所述场景数据中的虚拟物体的数据的管理顺序,更新所述场景数据,并且将更新后的所述场景数据设置为所述第二单元用以生成从第二视点观察到的虚拟空间的图像的场景数据。

3. 根据权利要求2所述的图像处理设备,其特征在于,所述第一单元更新所述场景数据,以将在绘制从所述第一视点观察到的虚拟空间的图像时搜索到的虚拟物体置于所述预定顺序的顶部。

4. 根据权利要求2所述的图像处理设备,其特征在于,所述第一单元更新所述场景数据,以按照搜索频率的降序,从所述预定顺序的顶部开始配置在绘制从所述第一视点观察到的虚拟空间的图像时所搜索到的虚拟物体。

5. 根据权利要求1所述的图像处理设备,其特征在于,还包括将虚拟空间中所包括的各虚拟物体的数据作为场景数据进行管理的单元,

其中,所述第一单元在从第一视点开始的视野中设置多个区域,按照预定顺序在所述场景数据中搜索所设置的各区域中所包括的虚拟物体的数据,基于所搜索到的数据绘制从所述第一视点开始的虚拟空间的图像,通过基于在绘制时所执行的所述搜索的处理结果改变所述场景数据的管理顺序,来更新所述场景数据,并且使用更新后的所述场景数据生成所述视野中的所述多个区域以外的区域中的虚拟空间的图像。

6. 根据权利要求 1 所述的图像处理设备,其特征在于,所述第一单元和所述第二单元均通过将所述绘制处理所获得的深度值转换成自另一视点的坐标系来进行绘制。

7. 根据权利要求 2 所述的图像处理设备,其特征在于,所述第一视点不同于所述第二视点。

8. 根据权利要求 2 所述的图像处理设备,其特征在于,所述第一视点对应于观看者的一只眼,所述第二视点对应于所述观看者的另一只眼。

9. 根据权利要求 1 所述的图像处理设备,其特征在于,所述第一单元和所述第二单元均分割绘制区域而使绘制的区域不重叠。

10. 一种由图像处理设备执行的图像处理方法,所述图像处理设备用于从多个视点绘制共同的物体,所述图像处理方法包括:

存储步骤,用于存储包括以预定顺序管理的多个虚拟物体的数据的场景数据,其中所述场景数据包括各虚拟物体的几何信息和材质信息;

第一步骤,用于绘制从第一视点看见的第一图像;以及

第二步骤,用于绘制从第二视点看见的第二图像,

其中,所述第一步骤绘制所述第一图像的上部区域并且所述第二步骤并行地绘制所述第二图像的下部区域,然后所述第一步骤绘制所述第一图像的下部区域并且所述第二步骤并行地绘制所述第二图像的上部区域,以及

所述第一步骤在绘制所述第一图像的上部区域时按照所述预定顺序在所述场景数据中搜索能从所述第一视点看见的虚拟物体以获得该虚拟物体的生成所述第一图像的上部区域所用的几何信息和材质信息,并且通过将所述场景数据中的多个虚拟物体的顺序改变成改变后的顺序来生成改变后的场景数据,其中,能从所述第一视点看见的虚拟物体在所述改变后的顺序中位于顶部位置,

在所述第一步骤绘制了所述第一图像的上部区域后所述第二步骤绘制所述第二图像的上部区域的情况下,所述第二步骤通过按照所述改变后的顺序在改变后的场景数据中搜索能从所述第二视点看见的虚拟物体以获得该虚拟物体的生成所述第二图像的上部区域所用的几何信息和材质信息,来绘制所述第二图像的上部区域。

图像处理设备和图像处理方法

技术领域

[0001] 本发明涉及一种图像处理设备和图像处理方法。

背景技术

[0002] 随着近年来计算机处理能力的提高,用于向用户提供真实体验的虚拟现实技术的研究得以发展(参考非专利文献1)。通过利用计算机图形表现虚拟空间并将其显示在HMD(HeadMounted Display,头戴显示器)或壁挂式显示器(wall-typedisplay)上,来实现该技术。

[0003] 在该领域中,用于向用户提供高质量体验所必需的要素是图像生成的速度。通常假定在跟随用户的视点移动的同时生成虚拟空间图像所需的处理速度为10~15帧/秒。为了满足该要求,已开发了一种用于在保持较以前更高的表现能力的同时高速生成图像的技术。

[0004] 近年来,成熟的计算机并行化和虚拟空间处理技术已使得先前不可能的实时光线跟踪(ray tracing)方法能够实现(参考非专利文献2)。特别地,将非专利文献2中所公开的光线跟踪方法称为实时光线跟踪,并且已得以广泛研究。该技术允许反射和折射的表现、高速阴影生成以及全局光照的表现,这些对传统光栅化(rasterization)方法来说都是困难的。因此,可以生成高质量图像。

[0005] 随着图像生成处理的表现能力的提高,获得高质量图像所需的计算负荷不断增大。要处理的数据量也在增大以满足在虚拟空间中实时显示物体的要求。因为这些原因,即使当实现实时光线跟踪时,计算负荷减轻技术也是必不可少的,以在维持高表现能力的同时以高帧频输出。

[0006] 专利文献1公开了一种用于通过使用时间序列相关性的光线跟踪来提高动画生成的效率的方法。动画通过更新逐渐变化的画面(帧)来表现运动。逐渐变化的画面包含应该处于视线内的物体的位置关系等的时间序列相关性(连贯性)。

[0007] 通常,在使用光线跟踪的图像生成处理中,光线跟踪中的搜索所需的时间最长。在专利文献1中,使用时间序列图像之间的连贯性,对于在前一帧和当前帧之间没有发生变化的部分,再使用前一帧的结果,从而缩短光线跟踪方法中的搜索时间。

[0008] [非专利文献1]VR世界の構築手法、監修者館すすむ、編者廣瀬通孝、発行所株式会社 培風館、2000年

[0009] [非专利文献2]Ingo Wald, Carsten Benthin, Markus Wagner, and Philipp Slusallek, "Interactive Rendering with CoherentRay-Tracing" in Computer Graphics Forum/Proceedings of theEUROGRAPHICS2001, pp. 153-164, 20(3), Manchester, UnitedKingdom, September3-7, 2001

[0010] [专利文献1]日本专利号2532055

[0011] 上述方法通过光线跟踪提高了图像生成处理的时间序列效率和速度。然而,根本没有考虑一次生成不同视点的多个图像的处理。例如,为了使戴着HMD的用户体验虚拟空

间,必需同时生成左右眼的两个图像,并且将它们作为立体图像呈现给他/她。在立体图像生成中,基于不同的视点位置和方位生成右眼图像和左眼图像,因此光线的反射方向不同。因为这个原因,不能使用时间序列相关性。为了生成要呈现给该用户的立体图像,必须针对左右眼均执行图像生成处理。

发明内容

[0012] 考虑到上述问题做出了本发明,并且本发明的目的是提供一种用于高效生成多个虚拟空间图像的技术。

[0013] 根据本发明的第一方面,一种图像处理设备,用于从多个视点绘制共同的物体,所述图像处理设备包括:

[0014] 第一单元,其绘制第一图像;以及

[0015] 第二单元,其绘制第二图像,

[0016] 其中,所述第一单元和所述第二单元均通过参考由另一所述单元的绘制处理所获得的信息,来绘制未绘制的区域。

[0017] 根据本发明的第二方面,一种由图像处理设备执行的图像处理方法,所述图像处理设备用于从多个视点绘制共同的物体,所述图像处理方法包括:

[0018] 第一步骤,用于绘制第一图像;以及

[0019] 第二步骤,用于绘制第二图像,

[0020] 其中,在所述第一步骤和所述第二步骤中,均通过参考由另一所述步骤中的绘制处理所获得的信息,来绘制未绘制的区域。

[0021] 通过以下参考附图对典型实施例的说明,本发明的其它特征将显而易见。

附图说明

[0022] 图 1 是用于解释根据本发明的第一实施例的图像处理的概况的图;

[0023] 图 2 是示出可适用于根据本发明的第一实施例的图像处理设备的计算机的硬件结构的框图;

[0024] 图 3 是示出为了生成要呈现给用户的左右眼的虚拟空间图像、由具有图 2 所示的结构计算机所执行的处理的流程图;

[0025] 图 4 是用于解释光线跟踪方法和虚拟空间分割的图;

[0026] 图 5 是示出由场景数据 206 所表示的树的结构例子的图;

[0027] 图 6 是示出场景树搜索的图;

[0028] 图 7 是用于解释根据本发明的第二实施例的图像处理的概况的图;

[0029] 图 8 是用于解释根据本发明的第二实施例的图像处理的概况的图;

[0030] 图 9 是示出根据本发明的第二实施例为了生成要呈现给用户的左右眼的虚拟空间图像、由具有图 2 所示的结构计算机所执行的处理的流程图;

[0031] 图 10A 和 10B 是用于解释自适应分割画面的处理的图;

[0032] 图 11 是用于解释根据本发明的第三实施例的图像生成处理的图;

[0033] 图 12 是示出根据本发明的第四实施例为了生成要呈现给用户的左右眼的虚拟空间图像、由具有图 2 所示的结构计算机所执行的处理的流程图;

- [0034] 图 13 是用于解释利用一般光栅化的图像生成处理的序列的图；
- [0035] 图 14 是用于解释利用立体视觉的三维坐标估计方法的图；
- [0036] 图 15 是用于解释根据本发明的第五实施例的图像处理的概况的图；以及
- [0037] 图 16 是示出根据本发明的第五实施例为了生成要呈现给用户的左右眼的虚拟空间图像、由具有图 2 所示的结构计算机所执行的处理的流程图。

具体实施方式

[0038] 现在，将参考附图详细说明本发明的优选实施例。作为权利要求书中所述的本发明的优选配置的例子来说明这些实施例，但是本发明不局限于以下所述的实施例。

[0039] 第一实施例

[0040] 本实施例假定利用光线跟踪方法生成要呈现给用户（观看者）左右眼的虚拟空间图像。更具体地，在本实施例中，生成要呈现给一只眼的虚拟空间图像，并且存储通过生成处理所获得的计算结果。然后，使用该计算结果生成要呈现给另一只眼的虚拟空间图像。这提高了用于生成要呈现给双眼的虚拟空间图像的处理的效率和速度。

[0041] 图 1 是用于解释根据本实施例的图像处理的概况的图。本实施例假定：在用户要体验虚拟空间时，利用光线跟踪方法为左眼和右眼生成不同视点位置和方位的虚拟空间图像。因此，将基于不同视点位置和方位的虚拟空间图像分别显示在画面 101 和画面 102 上，其中，画面 101 用于显示要呈现给用户的左眼的图像，画面 102 用于显示要呈现给用户的右眼的图像。

[0042] 假定显示在画面 101 和 102 上的图像的视点位置和方位如人的两眼间的距离一样，没有太大的不同。因为这个原因，在本实施例中，假定在画面 102 上也看见显示在画面 101 上的虚拟物体。这一假定使得即使在用于生成右眼的虚拟空间图像的处理中也可以使用通过生成左眼的虚拟空间图像所获得的计算结果。

[0043] 图像处理设备 103 利用光线跟踪方法生成被显示在画面 101 和 102 上的虚拟空间图像。

[0044] 画面 101 和 102 的区域 104 在左右眼的视野中不重叠。对于区域 104，不能再使用在生成被显示在一个画面上的虚拟空间图像时所获得的计算结果。因此，在每一图像生成处理中必需进行计算。

[0045] 在使用光线跟踪方法的图像处理计算中，用于搜索与光线交叉的虚拟物体的处理需要最长的处理时间。

[0046] 搜索的难度依赖于虚拟物体的多少，即，场景的复杂度。假定在场景中存在 10000 个虚拟物体。如果场景树中的第 10000 个物体是绘制对象，则每当生成一个帧的虚拟空间图像时必须进行 10000 次搜索处理，其中，场景树使用树状结构管理包括在虚拟空间中的元素（后面将详细说明虚拟物体搜索处理）。

[0047] 在本实施例中，当为了生成要呈现给一只眼的虚拟空间图像而从场景数据中搜索该只眼将看见的虚拟物体时，将该搜索结果应用于用于生成要呈现给另一只眼的虚拟空间图像的处理中。这减少了总计算负荷，加速了图像生成处理。

[0048] 图 2 是示出可适用于根据本实施例的图像处理设备的计算机的硬件结构的框图。计算机的硬件不局限于所示的结构。也可以使用具有任何其它结构的计算机，只要该计算

机主要包括执行处理的执行单元、以及存储程序和数据的存储单元即可。

[0049] 在图 2 中,使用存储在 RAM202 和 ROM203 中的计算机程序和数据,CPU201 对整个计算机进行控制,并且执行后面说明的各处理,作为该计算机的处理。

[0050] RAM202 具有用于临时存储从外部存储装置 204 所装载的处理程序 205(计算机程序)和场景数据 206 的区域,并且还具 CPU201 执行各种类型的处理所使用的工作区域。即,RAM202 可以根据需要提供各种类型的区域。

[0051] ROM203 存储计算机的引导程序和设置数据。

[0052] 外部存储装置 204 是以硬盘驱动器为代表的大容量信息存储装置。外部存储装置 204 存储 OS(操作系统)以及处理程序 205 和场景数据 206。外部存储装置 204 还存储后面说明的已知信息和以下说明中本领域技术人员在实践中所使用的信息。在需要时,在 CPU201 的控制下将存储在外部存储装置 204 中的计算机程序和数据载入 RAM202。CPU201 使用所装载的计算机程序和数据执行各种类型的处理。

[0053] 存储在外部存储装置 204 中的处理程序 205 是使 CPU201 执行后面说明的各种类型的处理作为计算机的处理的计算机程序。

[0054] 场景数据 206 是以树形式(树状结构)管理包括在虚拟空间中的元素的数据。例如,当虚拟物体包括众所周知的多边形时,场景数据 206 包含该多边形的颜色数据和法线矢量数据以及该多边形各顶点的坐标值数据(下文将这些数据称为几何信息)。如果虚拟物体具有纹理映射(texture mapping),则场景数据 206 还包含纹理映射数据。场景数据 206 还包含照射虚拟空间的虚拟光源的类型和亮度的信息。

[0055] 当使用光线跟踪方法绘制虚拟空间图像时,场景数据 206 还包含用于使得易于进行光线和虚拟物体的交叉判断的空间分割信息。在本实施例中,如上所述,利用光线跟踪方法生成虚拟空间图像。

[0056] 输入装置 207 输入观看虚拟空间图像的用户左右的每一个的位置和方位信息。因此,各种装置可用作输入装置 207。

[0057] 例如,可以使用键盘或鼠标作为输入装置 207。在这种情况下,用户使用输入装置 207 手动输入每只眼的位置和方位信息。

[0058] 可以使用位置和方位传感器作为输入装置 207。在这种情况下,将位置和方位传感器戴在用户的头上。位置和方位传感器将测量结果作为数据输入 RAM202。CPU201 使用该测量结果数据获得左右眼的位置和方位、位置和方位传感器与右眼之间的位置和方位关系以及位置和方位传感器与左眼之间的位置和方位关系。

[0059] 如上所述,可以使用各种方法来获取用户的左右眼的位置和方位信息,本发明不局限于一种方法。根据所使用的方法确定用作输入装置 207 的装置。

[0060] 显示装置 208 显示由 CPU201 生成的右眼的虚拟空间图像和左眼的虚拟空间图像,显示装置 208 由例如 CRT 或液晶面板形成。显示装置 208 当然还可以显示任何其它信息。更具体地,显示装置 208 可以使用图像或文本显示 CPU201 的处理结果。

[0061] 总线 209 连接上述单元。

[0062] 接着,将参考图 3 说明为了生成要呈现给用户的右眼和左眼的虚拟空间图像(右眼虚拟空间图像和左眼虚拟空间图像)、由具有图 2 所示的结构计算机执行的处理,图 3 是用于说明该处理的流程图。

[0063] 注意,将用以使 CPU201 执行根据图 3 所示的流程图的处理的计算机程序(包括处理程序 205)和数据(包括场景数据 206)存储在外部存储装置 204 中。如上所述,在需要时,在 CPU201 的控制下,将该计算机程序和数据载入 RAM202。CPU201 使用所装载的计算机程序和数据执行处理。因此,计算机根据图 3 所示的流程图执行该处理。

[0064] 在步骤 S300,对随后的处理执行初始化处理。初始化处理包括从外部存储装置 204 读出处理程序 205 并将其载入 RAM202 的处理。该初始化处理还包括在 RAM202 中分配随后的处理要使用的区域的处理。

[0065] 在步骤 S301,从外部存储装置 204 读出场景数据 206,并且在 RAM202 上顺序展开场景数据 206。此时在 RAM202 上展开的数据包含描述整个场景的树结构的场景树和节点信息。节点信息包含作为场景树的各个元素的虚拟物体的几何信息和材质信息以及虚拟光源信息。

[0066] 在步骤 S302,在 RAM202 中获取虚拟空间中的各视点(右眼和左眼)的位置和方位信息。如上所述,可以通过各种方法进行该获取。在本实施例中,用户使用输入装置 207 手动输入该信息。然而,可以使用预定的固定值作为右眼和左眼的视点位置和方位信息。

[0067] 在步骤 S303,使用在步骤 S301 在 RAM202 中所获取的数据组和步骤 S302 在 RAM202 中所获取的左右眼的位置和方位信息,生成要呈现给一只眼(第一视点)的虚拟空间图像(第一画面)。将所生成的第一画面显示在显示装置 208 的显示画面上。后面将详细说明该步骤的处理。

[0068] 在步骤 S304,使用在步骤 S303 的处理中更新后的场景数据 206,生成要呈现给另一只眼(第二视点)的虚拟空间图像(第二画面)。将所生成的第二画面显示在显示装置 208 的显示画面上。后面将详细说明该步骤的处理。

[0069] 在步骤 S305,判断是否终止该处理。为了终止该处理,例如,用户使用输入装置 207 输入结束指令。可选地,可以预先设置处理结束条件。如果判断出终止该处理,则该处理结束。如果不终止该处理,则处理返回到步骤 S302,以进行用于生成下一帧的右眼虚拟空间图像和左眼虚拟空间图像的处理。

[0070] 接着,将参考图 4 说明光线跟踪方法。图 4 是用于解释光线跟踪方法和虚拟空间的分割的图。

[0071] 在光线跟踪方法中,执行用于将光线从设置在虚拟空间中的视点 401 投影到虚拟画面 402 的各像素的处理。确定通过虚拟画面 402 的各光线和例如八叉树分割的虚拟空间 403 中的虚拟物体之间的交叉。当光线与虚拟物体交叉时,从场景数据 206 中搜索该虚拟物体的信息。计算由搜索到的信息所表示的虚拟物体对来自虚拟光源 405 的光的反射,从而确定虚拟画面 402 上的像素值。对虚拟画面 402 的所有像素进行该处理。

[0072] 对虚拟空间 403 进行八叉树分割以使得易于进行光线和虚拟物体之间的交叉判断。对于光线跟踪方法,提出了许多用于通过分割空间使得易于进行交叉判断的技术。例子有 kd 树分割和层次包围盒(BVH, Boundary Volume Hierarchy)。本实施例不依赖于空间分割算法。因此,可以使用任何空间分割方法。

[0073] 通过在场景数据 206 中搜索场景树,获得与光线交叉的虚拟物体的信息。

[0074] 接着,将说明场景数据 206。图 5 是示出由场景数据 206 所表示的树的结构的例子图。

[0075] “World”501 是与场景（虚拟空间）的基（根）节点相对应的节点。该节点定义场景的绝对坐标。

[0076] “Camera”502 是存储位置和方位以及视点的视角的节点。

[0077] “Object”503 是保持虚拟物体的各种类型的信息的节点。通常，由于场景包括多个虚拟物体，因而在“Object”503 下设置有对场景中的虚拟物体进行分组的“Sub-object”505。

[0078] “Transform”506 是定义相对于“World”501 的绝对坐标的“Object”503 的位置和方位的参数。

[0079] “Sub-object”505 是对作为表示虚拟物体的最小单位的“object1”507、“object2”... 进行分组的节点。在“Sub-object”505 下，将与出现在场景中的虚拟物体一样多的对象节点相互关联。

[0080] “object 1”507 具有“shape”508、“material”509 和“transform”510 的信息。

[0081] “shape”508 具有“object1”507 的多边形各顶点的坐标值数据和法线矢量数据等几何形状信息。

[0082] “material”509 将“object1”507 的纹理信息以及来自光源的光的漫反射信息和镜面反射信息，存储为属性数据。

[0083] “transform”510 表示“object1”507 的位置和方位信息。

[0084] “Light”504 具有照射虚拟空间场景的虚拟光源的信息，并且存储虚拟光源的位置（几何信息）、类型（例如，直接光源（directlight）、点光源（point source）、或聚光灯光源（spot light））和亮度信息（包括色彩信息）的数据。

[0085] 在上述配置中，为了获得与光线交叉的虚拟物体的信息，必须进行次数与光线交叉的数量一样多的、用于搜索图 5 所示的场景树的处理（场景数据中的搜索）。即，当光线在分层结构的深位置处或者在包括许多虚拟物体的场景中，与虚拟物体交叉时，搜索处理的量变得很大。

[0086] 接着，将说明场景数据 206 中的场景树搜索。图 6 是示出场景树搜索的图。

[0087] 场景树 601 是初始状态下的场景树。场景树 601 中的各子节点 602 表示从给定视点（感兴趣的视点）看见的虚拟物体。搜索路径 603 表示用以在光线与虚拟物体（子节点 602）交叉时搜索该虚拟物体的节点的路径。预先设置该路径。

[0088] 通常，在用于生成从感兴趣的视点所看见的虚拟空间图像的处理中，沿着与显示面板上同子节点 602 相对应的虚拟物体的像素一样多的搜索路径 603，搜索子节点 602。在场景树 601 中，由于子节点 602 位于搜索路径 603 的末端，因而每次搜索都很耗时。在本实施例中，为了缩短时间，将子节点 602 的位置移动至搜索路径 603 的顶部，从而更新场景树 601。

[0089] 更具体地，当找到子节点 602 时，在 RAM202 中开始新场景树的创建。首先，在 RAM202 中创建场景树 601 的副本。将所复制的场景树中的各子节点 602 的位置移动至搜索路径 603 的顶部，从而更新所复制的场景树。即，通过改变场景数据中的虚拟物体的数据的管理顺序来更新场景数据。在图 6 中，附图标记 604 表示通过将子节点 602 的位置移动至搜索路径 603 的顶部位置而更新的所复制的场景树。如附图标记 605 所示，用于搜索子节点 602 的路径短于搜索路径 603。子节点 602 中的节点的排列顺序（管理顺序）没有特别

限制。

[0090] 当结束用于生成从感兴趣的视点所看见的虚拟空间图像的处理时,将场景树 601 更新成所复制的场景树。将更新后的场景树 601 设置为用于生成从感兴趣的视点之外的其它视点所看见的虚拟空间图像的场景树。

[0091] 如上所述,使用在用于生成给定视点的虚拟空间图像的处理中更新后的场景树,来生成下一视点的虚拟空间图像。因此,当生成多个视点的虚拟空间图像时,对生成顺序中处于较晚的视点来说,搜索距离变得较短。

[0092] 将在用于生成虚拟空间图像的处理中没有搜索到的节点配置在新场景树(搜索路径 603)的最后端位置。甚至将未搜索到的节点也复制到新场景树。因为这个原因,即使当第一画面和第二画面在视野中具有不同虚拟物体时,也不会出现问题。即,仅存在于第二画面中的虚拟物体不是新场景树中重新配置的对象。然而,由于新场景树存储该虚拟物体的信息,因而可以毫无问题地显示该虚拟物体。

[0093] 如上所述,在生成一个视点的虚拟空间图像时,将要在场景树中搜索的虚拟物体的节点的位置重新配置在搜索路径的顶部位置处,从而实现高效搜索处理。

[0094] 在新场景树创建的节点重新配置操作中,可以对搜索的频率进行计数,并且可以不基于搜索顺序,而是按照搜索频率的顺序,重新配置场景树。更具体地,对各节点的搜索的次数进行计数。当结束一个虚拟空间图像的生成处理时,按照计数值的降序从搜索路径的顶部开始配置节点。

[0095] 另外,如果在第一画面的图像生成的处理中,在视野中没有虚拟物体,则在第二画面中不需要进行场景树搜索处理。因此,当在第一画面的图像处理过程中,将表示在视野中没有虚拟物体的信息添加给新场景树的信息时,能够高速执行第二画面图像生成处理。

[0096] 对于在第一画面的视野中没有虚拟物体时的第二画面图像生成处理,预先准备背景图像作为纹理,并且不利用光线跟踪而利用纹理绘制进行图像生成处理。

[0097] 在某些情况下,仅在第二画面中包括虚拟物体。在这种情况下,不能使用虚拟物体的有无信息。然而,当由于虚拟物体和视点位置之间的极短的距离而造成视差大时,发生仅一只眼可以看见虚拟物体的现象。为了使用虚拟物体的有无信息,进行限制以使得虚拟物体存在于预定深度值或者更远离视点,从而防止视差变大。该限制使得能够使用虚拟物体的有无信息。

[0098] 如果第一画面图像生成中的视点位置和方位与第二画面图像生成中的视点位置和方位相同或差不多相同,则第一画面的搜索结果等同于第二画面的搜索结果。这使得能够对第二画面图像生成处理再使用第一画面的搜索结果。

[0099] 在各种类型的图像生成处理(例如,光栅化方法、体绘制(volume rendering)方法和颗粒绘制(particule rendering)方法)中,场景搜索的问题始终是必然要遇到的问题。因此,即使在改变图像生成处理时,通过场景重建来提高图像生成的效率的方法也是有效的。因此,本实施例通常可适用于各种类型的图像生成处理。

[0100] 在本实施例中,计算机的显示装置 208 显示第一画面和第二画面。然而,任何其它显示装置可以显示第一画面和第二画面。例如,当 HMD 与计算机连接时,HMD 的右眼显示画面可以显示右眼虚拟空间图像,而左眼显示画面可以显示左眼虚拟空间图像。

[0101] 第二实施例

[0102] 在第一实施例中,通过顺序处理来生成第一画面和第二画面。然而,在第二实施例中,并行分割和生成第一画面和第二画面。

[0103] 图 7 是用于解释根据该实施例的图像处理的概况的图。

[0104] 本实施例假定:将第一画面和第二画面都分成上下两个区域,并且通过一个 CPU 处理各区域。

[0105] 参考图 7,附图标记 701 表示第一画面的上部区域(部分区域),而附图标记 703 表示第一画面的下部区域。通过将第一画面分成上下两个部分来获得上部区域 701 和下部区域 703。该上部区域和下部区域不重叠。

[0106] 参考图 7,附图标记 702 表示第二画面的上部区域,而附图标记 704 表示第二画面的下部区域。通过将第二画面分割成上下两个部分来获得上部区域 702 和下部区域 704。该上部区域和下部区域不重叠。

[0107] 在本实施例中,并行生成一个画面的上部区域和另一画面的下部区域。在一个画面的上部区域的生成处理中,复制原始场景数据,并且如在第一实施例中一样,更新所复制的场景树,以将出现在一个画面的上部区域中的虚拟物体的节点移动至搜索路径的顶部。在另一画面的下部区域的生成处理中,复制原始场景数据,并且如在第一实施例中一样,更新所复制的场景树,以将出现在另一画面的下部区域中的虚拟物体的节点移动至搜索路径的顶部。即,在本实施例中,生成一个画面的场景数据(第一场景数据)和另一画面的场景数据(第二场景数据)。

[0108] 接着,并行生成一个画面的下部区域和另一画面的上部区域。在一个画面的下部区域的生成处理中,使用在用于生成另一画面的下部区域的处理中所更新的场景数据。在另一画面的上部区域的生成处理中,使用在用于生成一个画面的上部区域的处理中所更新的场景数据。

[0109] 如上所述,并行执行两个处理。因此,当图像生成完成到画面的一半时,可以高效地开始场景树搜索处理。

[0110] 图 8 是用于解释根据本实施例的图像处理的概况的图。在图 8 中,与图 1 中相同的附图标记表示相同部分,将不再重复对其的说明。在本实施例中,为了并行生成第一画面 101 和第二画面 102,增加了画面区域分割控制单元 801。这使得可以在并行生成第一画面 101 和第二画面 102 的图像时,将各画面分割成区域,并且执行前半部分和后半部分中的图像生成处理。

[0111] 另外,在本实施例中,在图像生成的后半部分中进行信息交换。因此,不同于第一实施例,还将在第二画面 102 的图像生成处理中所获取的信息输出给图像处理设备 103,并且将该信息用于第一画面 101 的图像生成。

[0112] 接着,将参考图 9 说明根据本实施例的处理,该处理由具有图 2 所示的结构 of 的计算机执行,以生成要呈现给用户的左右眼的虚拟空间图像,图 9 是示出该处理的流程图。

[0113] 注意,将使 CPU201 执行根据图 9 所示的流程图的处理的计算机程序(包括处理程序 205)和数据(包括场景数据 206)存储在外部存储装置 204 中。如上所述,在需要时,在 CPU201 的控制下,将该计算机程序和数据载入 RAM202。CPU201 使用所装载的计算机程序和数据执行处理。因此,计算机根据图 9 所示的流程图执行该处理。

[0114] 在步骤 S900,对随后的处理执行初始化处理。该初始化处理包括用于从外部存储

装置 204 读出处理程序 205 并将其载入 RAM202 的处理。该初始化处理还包括用于在 RAM202 中分配随后处理要使用的区域的处理。

[0115] 在步骤 S901, 从外部存储装置 204 读出场景数据 206, 并且在 RAM202 上顺序展开场景数据 206。此时在 RAM202 上所展开的数据包含描述整个场景的树结构的场景树和节点信息。节点信息包含作为场景树的各个元素的虚拟物体的几何信息和材质信息以及虚拟光源信息。

[0116] 在步骤 S902, 将右眼的画面和左眼的画面均分割成上下两个区域。生成表示各分割的区域的区域信息并将其存储在 RAM202 中。区域信息包含例如表示该区域属于哪一画面 (右眼画面或左眼画面) 的信息以及该区域的左上角和右下角的坐标位置。

[0117] 在步骤 S903, 在 RAM202 中获取虚拟空间中各视点 (左右眼) 的位置和方位信息。如上所述, 可以通过各种方法进行位置和方位信息的获取。在本实施例中, 用户使用输入装置 207 手动输入该信息。然而, 可以使用预定的固定值作为右眼的位置和方位信息以及左眼的位置和方位信息。

[0118] 在步骤 S904, 使用在步骤 S901 和 S902 在 RAM202 中所获取的数据组以及在步骤 S903 在 RAM202 中所获取的左右眼的位置和方位信息, 生成要呈现给一只眼的虚拟空间图像 (第一画面) 的上部区域。将所生成的第一画面的上部区域显示在显示装置 208 的显示画面上。

[0119] 与步骤 S904 并行地执行步骤 S905 中的处理。在步骤 S905, 使用在步骤 S901 和 S902 在 RAM202 中所获取的数据组以及在步骤 S903 中在 RAM202 所获取的左右眼的位置和方位信息, 生成要呈现给另一只眼的虚拟空间图像 (第二画面) 的下部区域。将所生成的第二画面的下部区域显示在显示装置 208 的显示画面上。

[0120] 在步骤 S907, 使用在步骤 S905 中更新后的场景数据, 生成第一画面的下部区域。将所生成的第一画面的下部区域显示在显示装置 208 的显示画面上。

[0121] 在与步骤 S907 并行的步骤 S908 中, 使用在步骤 S904 中更新后的场景数据, 生成第二画面的上部区域。将所生成的第二画面的上部区域显示在显示装置 208 的显示画面上。

[0122] 在步骤 S909, 判断是否终止该处理。为了终止该处理, 例如, 用户使用输入装置 207 输入结束指令。可选地, 可以预先设置处理结束条件。如果判断为终止该处理, 则结束该处理。如果不终止该处理, 则处理返回到步骤 S903, 以进行用于生成下一帧的右眼虚拟空间图像和左眼虚拟空间图像的处理。然后, 对下一帧进行随后的处理。

[0123] 本实施例假定: 通过将并行计数设置为 2, 在两个处理中同时进行第一画面的图像生成和第二画面的图像生成。然而, 并行计数不必一定为 2。本实施例还可以处理并行计数为 3 或更多的情况。在这种情况下, 根据并行计数分割画面区域, 并且在各处理中生成用于提高搜索效率的场景树。

[0124] 本实施例假定: 通过将画面分割成上下两个区域来执行图像生成处理。然而, 不是必须将画面分成上下两个区域。可以将画面分成左右两个区域。当并行计数增大时, 可以相应地改变画面区域分割方法。可以根据要构建的系统或要体验的场景选择优选的画面分割方法。无论如何, 在本实施例中, 进行多次图像生成, 以并行生成一个视点的虚拟空间图像和另一视点的虚拟空间图像。

[0125] 变形例

[0126] 这里,将说明自适应画面分割的例子。

[0127] 图 10A 和 10B 是用于解释自适应分割画面的处理的图。

[0128] 参考图 10A,从左上角的位置 1001 开始第一画面 101 中的图像生成处理。从右下角的位置 1004 开始第二画面 102 中的图像生成处理。如图 10B 所示,当在这两个画面中当前正处理的像素的位置到达了相同的位置时,对第一画面使用在生成第二画面时所更新的场景数据,以处理剩余区域。对于第二画面,使用在生成第一画面时所更新的场景数据。在图 10B 中,附图标记 1002 表示当在这两个画面中当前正处理的像素的位置到达了相同位置时的处理过的区域。

[0129] 如上所述,根据本实施例,根据图像生成处理的并行处理能力分割画面,以分开执行图像生成处理。这使得能够高速高效地执行图像生成处理。

[0130] 第三实施例

[0131] 在第一和第二实施例中,对各像素顺序进行图像生成。第三实施例与这两个实施例的很大不同在于:在图像生成处理的前半部分中仅对部分区域进行图像生成,对场景数据进行更新,然后再次更详细地执行图像生成处理。

[0132] 图 11 是用于解释根据本实施例的图像生成处理的图。包括在从视点开始的视野中的虚拟空间图像(画面)1104 对应于第一或第二画面。在图像 1104 中设置特定区域 1101。在图 11 中,等间隔地离散设置特定区域 1101。在本实施例中,各特定区域 1101 对应于图像 1104 的一个像素。然而,特定区域 1101 的大小没有特别限制。另外,特定区域不必一定具有所示的布局。

[0133] 在本实施例中,以两个步骤生成一个虚拟空间图像。在第一步骤的生成(第一生成)中,生成特定区域中的图像。在第二步骤的生成(第二生成)中,以与第一实施例中相同的方法,使用在第一生成中所更新的场景数据,生成剩余区域(除特定区域 1101 以外的区域)中的图像。

[0134] 在本实施例中,如上所述,离散设置光线所投影到的区域。这使得可以高速重建整个场景的场景数据,而无需计算画面上的所有像素。

[0135] 对除特定区域 1101 以外的区域,通过在场景重建之后再次执行详细的图像生成处理,可以高效地进行图像生成。本实施例的方法主要对于在确定画面上的像素值时计算各像素的光线跟踪等图像生成处理非常有效。

[0136] 可以将特定区域中的场景重建应用于第一画面 101 和第二画面 102。可选地,通过将场景重建应用于这两个画面的其中一个所获得的结果可应用于第一画面和第二画面两者中的图像生成处理。

[0137] 第四实施例

[0138] 在第一、第二和第三实施例中,说明了用于通过交换在各图像生成处理中所输出的场景数据来高效生成图像的处理。第四实施例与这些实施例的不同在于:在利用光栅化的图像生成处理中,对左右眼各自的图像生成处理中所输出的深度值执行视点坐标变换,并且使用处理后的深度值。

[0139] 根据本实施例的图像处理的概况与第二实施例中所述的概况有许多相同点,将仅说明不同点。

[0140] 在本实施例中,首先,通过普通绘制方法,计算通过图 7 所示的区域分割所获得的上部区域 701 和下部区域 704 的不重叠的部分区域的亮度值。将此时在计算亮度值的处理中所获得的深度值(Z 缓存的值)以及对象物体的材质信息存储在 RAM202 中。

[0141] 将存储在 RAM202 中的深度值变换成另一视点的坐标系中的深度值。如果可以获得通过该变换所获得的深度值和对象物体的材质信息,则可以计算亮度值。

[0142] 接着,将参考图 12 说明根据本实施例的处理,该处理由具有图 2 所示的结构计算机执行,以生成要呈现给用户的左右眼的虚拟空间图像,其中,图 12 是示出该处理的流程图。图 12 中与图 9 中相同的步骤编号表示执行相同处理的步骤。

[0143] 在步骤 S904 和 S905,除第二实施例中所述的处理外,还进行用于获得各分割的区域的每一像素的亮度值的处理。

[0144] 在步骤 S1201,将在步骤 S904 和 S905 的处理中所获得的深度值(Z 缓存的值)、对象物体的各顶点的法线信息以及对象物体的材质信息存储在 RAM202 中。

[0145] 在步骤 S1202,对在步骤 S1201 中存储在 RAM202 中的深度值进行视点坐标变换。该坐标变换是用于将左眼的深度值变换成右眼的深度值、并将右眼的深度值变换成左眼的深度值的处理。

[0146] 最后,在步骤 S907 和 S908,除第二实施例中所述的处理外,还对在步骤 S1202 中变换后的深度值进行下面的处理:基于在步骤 S1201 中所存储的材质信息,计算各分割的区域的每一像素的亮度值。

[0147] 接着,将说明本实施例的绘制处理中的亮度值确定方法。

[0148] 图 13 是用于解释利用普通光栅化进行图像生成处理的序列的图。

[0149] 在模型变换 1301 中,将存储在外部存储装置 204 中的场景数据的信息(三维坐标)载入 RAM202,并将其变换成全局坐标。即,在模型变换 1301 中,在三维空间中转动并变形虚拟物体。该变换还包括在局部对象空间上的放大/缩小和转动等基本坐标变换。在处理模型变换 1301 时所获得的数据不依赖于视点位置和方位。因此,左右眼可以共享该数据。

[0150] 在视点变换 1302 中,基于虚拟照相机的位置和方位,将在全局坐标上所定义的虚拟物体的位置和方位变换成局部坐标系上的位置和方位。更具体地,预先获得视点变换的矩阵,并且对虚拟物体的各顶点使用矩阵运算进行视点变换。结果,将原始三维场景变换成从虚拟照相机观察的坐标系上的场景。

[0151] 在进行视点变换 1302 处理时所获得的数据依赖于各个视点。因此,左右眼不能共享该数据。

[0152] 在投影变换 1303 中,进行从利用虚拟照相机所定义的三维坐标系向二维坐标系的变换。通过投影变换 1303,将虚拟空间映射为从虚拟照相机观察的平面(虚拟画面)上的二维信息。

[0153] 在光栅化 1304 中,在执行裁切处理和隐藏表面去除之后,计算投影在虚拟画面上的场景的二维图像的每一像素的亮度值。

[0154] 在裁切处理中,丢弃视野外的虚拟物体的多边形,并且仅剪切视野中的多边形。在隐藏表面去除中,丢弃未指向视点的多边形,即,理论上从视点不可见的多边形。此时,按照与视点的距离的降序,将从该视点应该可见的多边形写入 Z 缓存中。当顺序覆盖这些值时,

计算与各像素值相对应的深度值,并且仅选择从该视点可见的多边形。

[0155] 在光栅化 1304 中,在隐藏表面去除之后,从阴影处理 (shading processing) 的场景数据提取各顶点的法线信息和虚拟物体的材质信息。在需要时,还提取纹理信息。如果左右眼看见同一物体,则可以共享该物体的材质信息。基于所提取的数据、以及虚拟视点的位置和方位信息,计算反射光。然后,计算虚拟画面上的每一像素的亮度值。然而,根据物体的材质信息所计算出的阴影处理的结果根据视点的位置和方位而变化,因此,不能共享该结果。

[0156] 在显示 1305 中,将最终着色的像素显示在监视器或其它显示装置上。

[0157] 当在一般光栅化处理中从不同视点观察共同的虚拟物体时,可以共享通过模型变换 1301 所获得的数据和物体的材质信息,而无需变换处理。然而,还可以通过视点坐标变换使用在光栅化 1304 的处理中所获得的深度值。然而,根据视点和物体之间的位置关系可能发生阻挡,不可能精确地计算深度值。在这种情况下,通过参考前一帧的相应像素值来确定各亮度值。

[0158] 接着,将说明用于根据利用视点坐标变换所获得的深度值计算各像素的亮度值的方法。

[0159] 图 14 是用于解释利用立体视觉的三维坐标估计方法的图。

[0160] 如图 14 所示,在三维空间中定义 xyz 绝对坐标系 1401。配置左照相机镜头和右照相机镜头,使得将其中心的绝对坐标设置为以距离 d 间隔开的 $0_L = (0, 0, 0)$ 和 $0_R = (d, 0, 0)$ 。假定镜头的焦距为 f ,即,从镜头中心到左图像平面和右图像平面中的每一个的距离为 f 。从这样设置的虚拟视点观察虚拟物体 1402。将其上投影有所观察的图像的右眼画面和左眼画面分别定义为虚拟画面 1403R 和 1403L。

[0161] 当右眼观察虚拟物体 1402 上的点 P 时,将点 P 投影到虚拟画面 1403R 上的点 $P_R(x_R, y_R)$ 。当左眼观察虚拟物体 1402 上的点 P 时,将点 P 投影到虚拟画面 1403L 上的点 $P_L(x_L, y_L)$ 。点 P_L 和 P_R 的坐标分别是基于设置为虚拟画面 1403L 和 1403R 的中心的原点的相对坐标。

[0162] 此时,通过使用由测量点和这两个照相机的中心所形成的三角形的三角测量,可以获得对象物体的表面上的点 $P(x_p, y_p, z_p)$ 。

[0163] 如果已知点 P_L 和 P_R 以及各种参数,则可以计算对象的三维坐标。这是使用计算机视觉中的立体视觉的一般深度估计方法。

[0164] 在本实施例中,当已知左右眼的其中一个的画面上的像素值、以及与该像素相对应的对象的三维坐标时,使用基于立体视觉的深度估计方法确定另一只眼的画面上的相应像素的像素值。例如,当将点 $P_R(x_R, y_R)$ 和对象物体的表面上的点 $P(x_p, y_p, z_p)$ 作为输入时,计算点 $P_L(x_L, y_L)$ 。

[0165] 因此,如果已知视点位置和方位,并且可以使用通过视点坐标变换所获得的深度值计算对象物体的表面上的点 P ,则可以计算虚拟画面上的相应点。

[0166] 在步骤 S907 和 S908,除第二实施例中所述的处理外,还进行下面的处理:与通过计算所获得的虚拟画面上的点对应地,装载存储在 RAM202 中的物体的材质信息。通过光栅化 1304 的处理执行阴影处理和纹理映射处理,从而计算各个亮度值。重复该处理,直到计算出所有像素为止。利用光栅化的图像生成方法是使用用于进行一般图形处理的硬件所实

现的公知技术。

[0167] 在上述处理中,共享通过用于计算一个画面上的分割区域的亮度值的处理所获得的信息,从而计算另一画面上的相应区域的亮度值。

[0168] 第五实施例

[0169] 在第一~第四实施例中,说明了用于通过交换在生成立体图像时的图像生成处理中所获得的信息来高效生成图像的处理。第五实施例假定使用两或多眼的照相机系统进行图像生成,其与上述实施例的不同在于:将画面区域分成两个或多个区域,并且对各区域应用图像生成处理。

[0170] 图 15 是用于解释根据本实施例的图像处理的概况的图。

[0171] 将三个视点的绘制结果分别显示在第一画面 1501、第二画面 1502 和第三画面 1503 上。在本实施例中,将每个画面分割成三个区域,并且在各区域中进行绘制。

[0172] 区域 1504 ~ 1506 是各个画面中首先开始处理的绘制区域。在本实施例中,将区域 1504 ~ 1506 设置为不重叠。当完成了画面的区域 1504 ~ 1506 中的绘制时,通过参考区域 1504 ~ 1506 的计算结果开始对未计算的区域 1507 ~ 1509 进行计算。当完成了对未计算的区域 1507 ~ 1509 的计算时,进行对剩余区域的计算。

[0173] 接着,将参考图 16 说明根据本实施例的处理,该处理由具有图 2 所示的结构 of 计算机执行以生成要呈现给三只眼的虚拟空间图像,其中,图 16 是示出该处理的流程图。图 16 中与图 9 相同的步骤编号表示用于执行相同处理的步骤。

[0174] 在步骤 S902,将三个画面(第一画面、第二画面和第三画面)都分割成上、中和下三段(对应于照相机的数量)。不特别限制该分割形式。可以将各画面垂直地分割成相等的部分。可选地,可以根据场景的复杂度改变分割的区域。

[0175] 在步骤 S1601a,进行第一画面的上段的绘制处理。在步骤 S1601b,进行第二画面的中段的绘制处理。在步骤 S1601c,进行第三画面的下段的绘制处理。以与其它实施例相同的方法执行步骤 S1601a、S1601b 和 S1601c 中的每一个的绘制处理。

[0176] 在步骤 S1602,将步骤 S1601a、S1601b 和 S1601c 的绘制结果存储在 RAM202 中。

[0177] 在步骤 S1603a,通过参考存储在 RAM202 中的、第一画面的上段的绘制结果,确定第一画面的中段的区域中的亮度值,并且执行该中段的绘制处理。在步骤 S1603b,通过参考存储在 RAM202 中的、第二画面的中段的绘制结果,确定第二画面的下段的区域中的亮度值,并且执行该下段的绘制处理。在步骤 S1603c,通过参考存储在 RAM202 中的、第三画面的下段的绘制结果,确定第三画面的上段的区域中的亮度值,并且执行该上段的绘制处理。然而,在步骤 S1603a、S1603b 和 S1603c 进行绘制的区域可以是任意区域,该区域不必仅为计算时未计算的区域。

[0178] 在步骤 S1604a,通过参考第一画面的中段的绘制结果,确定第一画面的下段的区域中的亮度值,并且执行该下段的绘制处理。在步骤 S1604b,通过参考第二画面的下段的绘制结果,确定第二画面的上段的区域中的亮度值,并且执行该上段的绘制处理。在步骤 S1604c,通过参考第三画面的上段的绘制结果,确定第三画面的中段的区域中的亮度值,并且执行该中段的绘制处理。

[0179] 在本实施例中,不将步骤 S1603a、S1603b 和 S1603c 的计算结果存储在 RAM202 中。然而,根据所构建的系统,可以利用在步骤 S1603a、S1603b 和 S1603c 计算出的多条信息覆

盖在步骤 S1602 中所存储的多条信息。

[0180] 假定：将视点配置在水平方向，通过按照视点的配置顺序配置对应于视点的画面所获得的结果为第一画面、第二画面和第三画面。在这种情况下，通过使用具有更近视点的第二画面的绘制处理结果，可以比通过使用第三画面的结果更高效地生成第一画面的图像。因此，为了参考所获得的信息，可以根据所构建系统选择合适的部件。

[0181] 如上所述，即使当使用两个或多个虚拟视点来观察共同的虚拟物体时，也能够通过将各画面分成任意数量的区域、并共享各个区域的计算结果，来高速高效地执行绘制。

[0182] 第六实施例

[0183] 在区域分割中，当第一画面和第二画面之间区域不重叠时，效率可以最大化。然而，如果区域完全不重叠，则边界部分的边缘变得明显。可以通过在各边界边缘附近提供若干像素的重叠区域，并且合成并平滑通过计算所获得的图像，使得边缘变得不明显。可以通过根据所构建的系统使用合适的部件，来确定区域分割时的重叠区域的大小。

[0184] 其它实施例

[0185] 本发明的目的还通过下面的方法实现。向系统或设备提供记录用以实现上述实施例的功能的软件程序代码的记录介质（或存储介质）。该存储介质当然为计算机可读存储介质。该系统或设备的计算机（或 CPU 或 MPU）读出并执行存储在该记录介质中的程序代码。在这种情况下，从记录介质读出的程序代码本身实现上述实施例的功能。记录该程序代码的记录介质构成本发明。

[0186] 计算机执行所读出的程序代码，运行在计算机上的操作系统（OS）基于该程序代码的指令执行部分或全部实际处理，从而实现上述实施例的功能。

[0187] 假定将从该记录介质读出的程序代码写入插入计算机中的功能扩展板或与计算机连接的功能扩展单元的存储器中。该功能扩展板或功能扩展单元的 CPU 基于该程序代码的指令执行部分或全部实际处理，从而实现上述实施例的功能。

[0188] 应用本发明的记录介质存储与上述流程图相对应的程序代码。

[0189] 尽管参考典型实施例说明了本发明，但是应该理解，本发明不局限于所公开的典型实施例。所附权利要求书的范围符合最宽的解释，以包含所有这类修改、等同结构和功能。

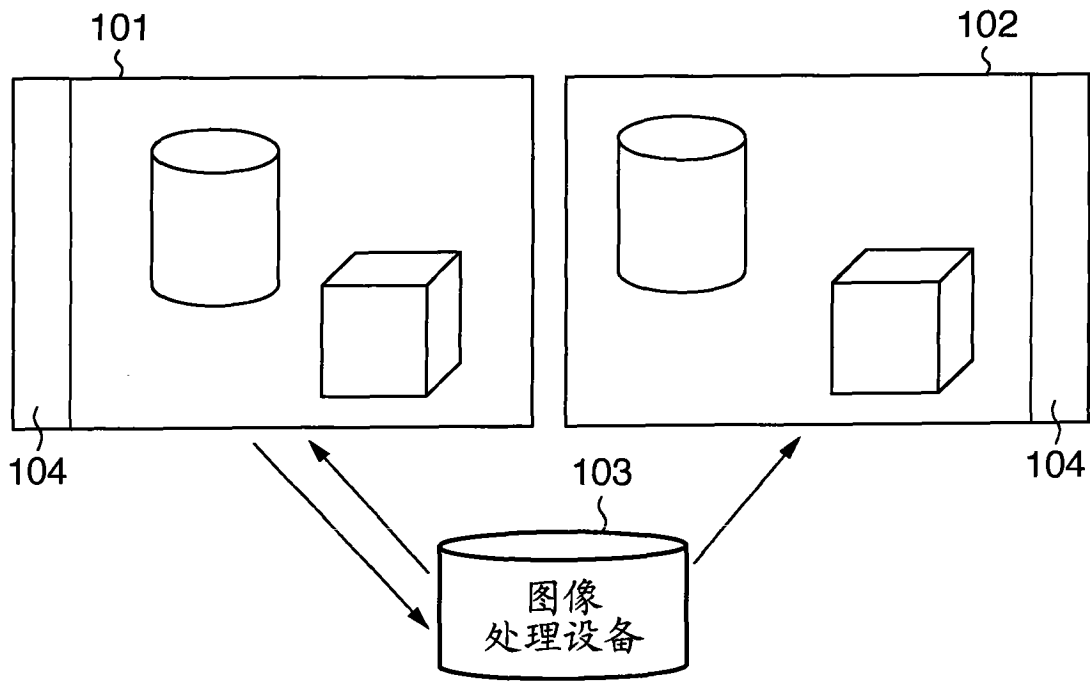


图 1

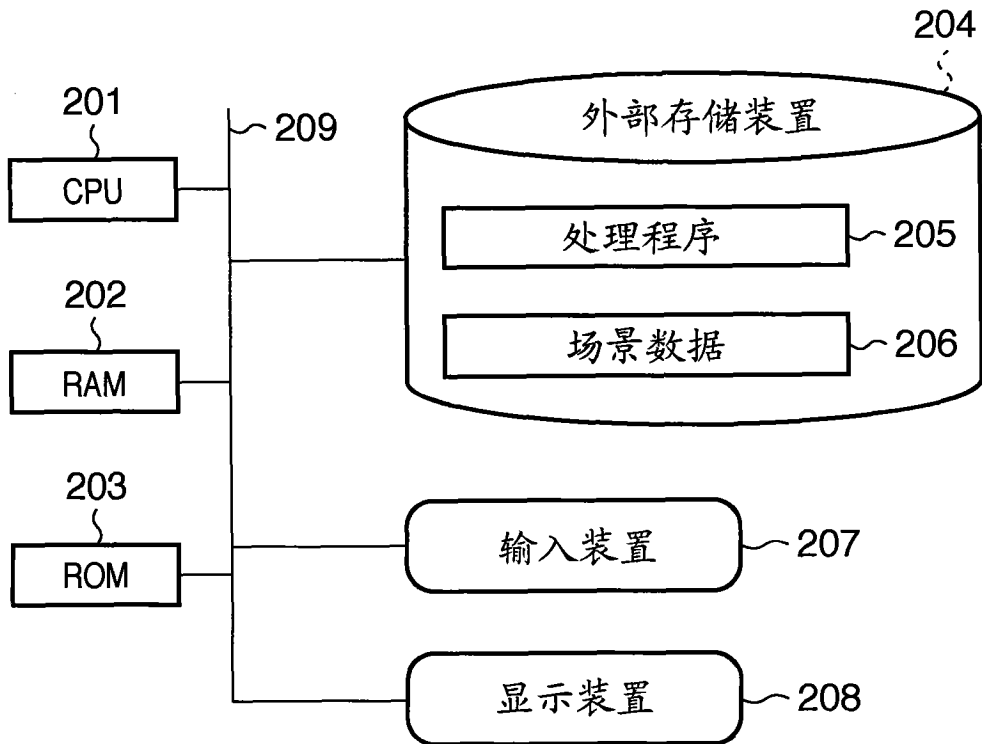


图 2

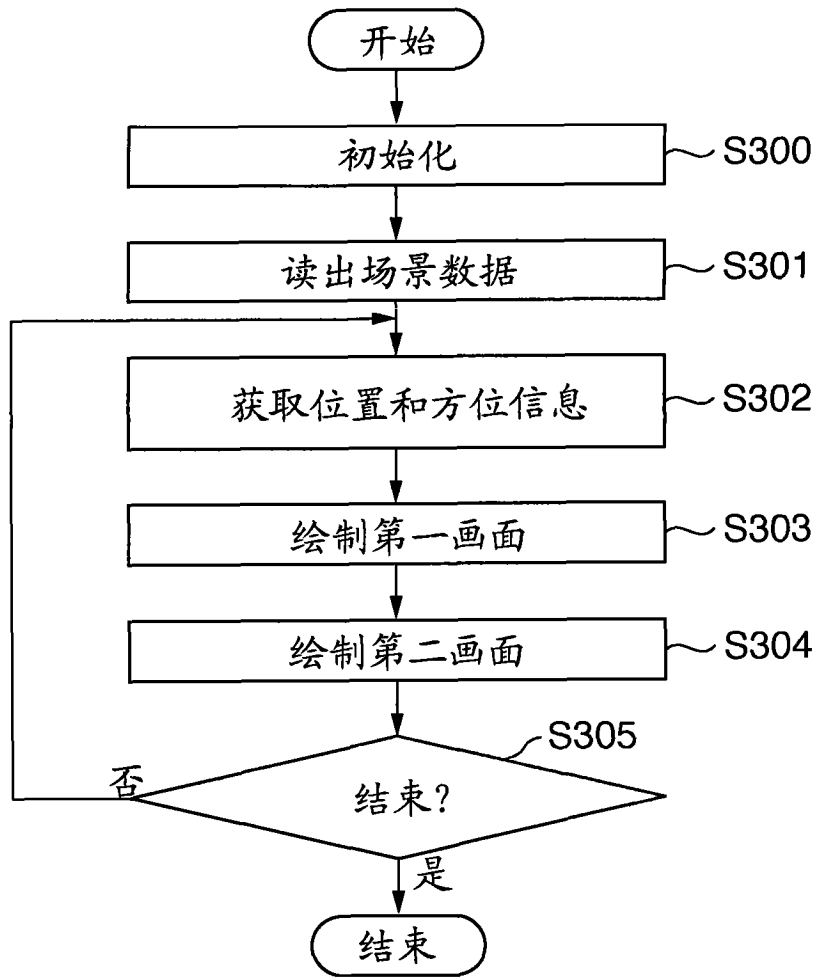


图 3

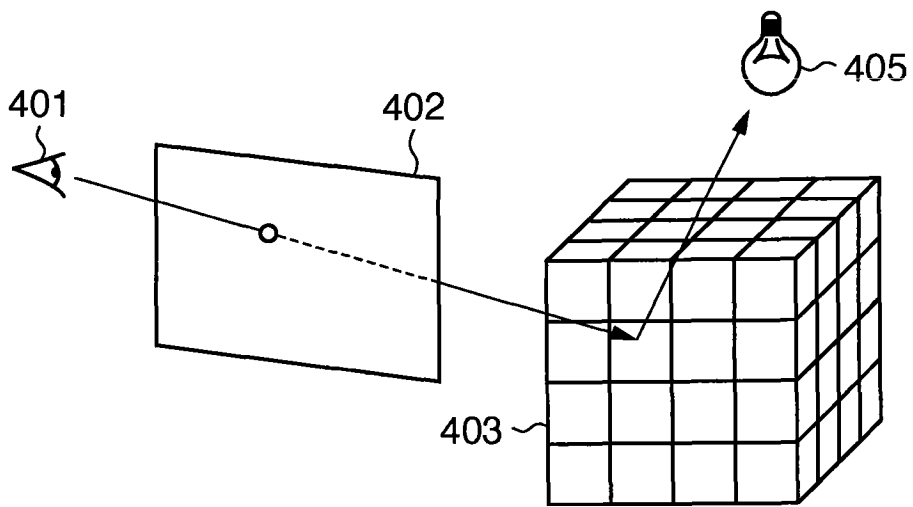


图 4

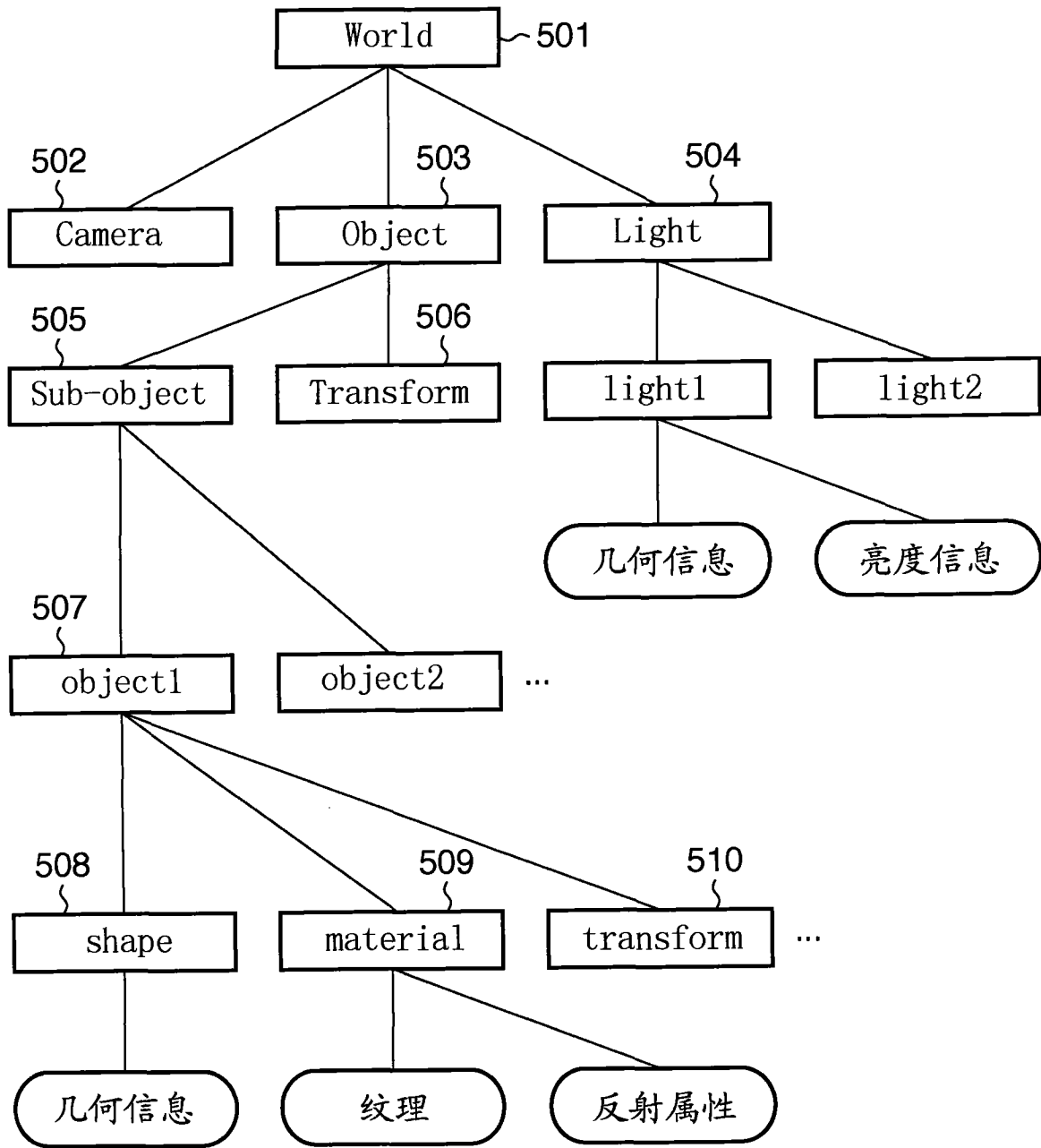


图 5

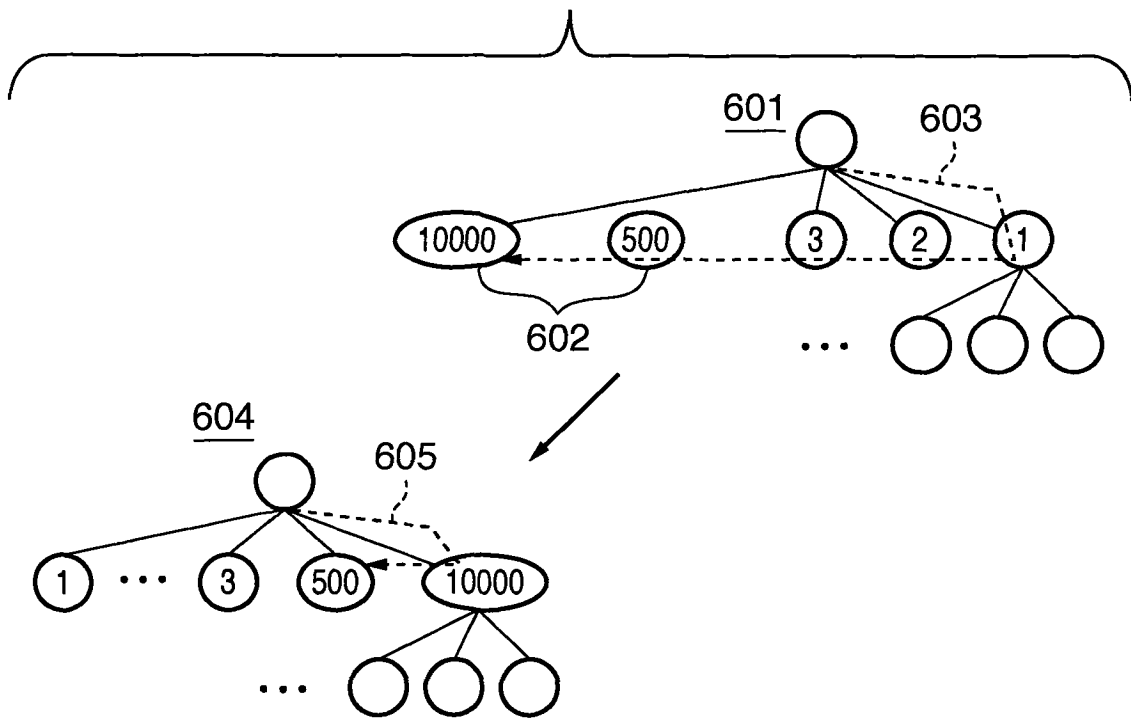


图 6

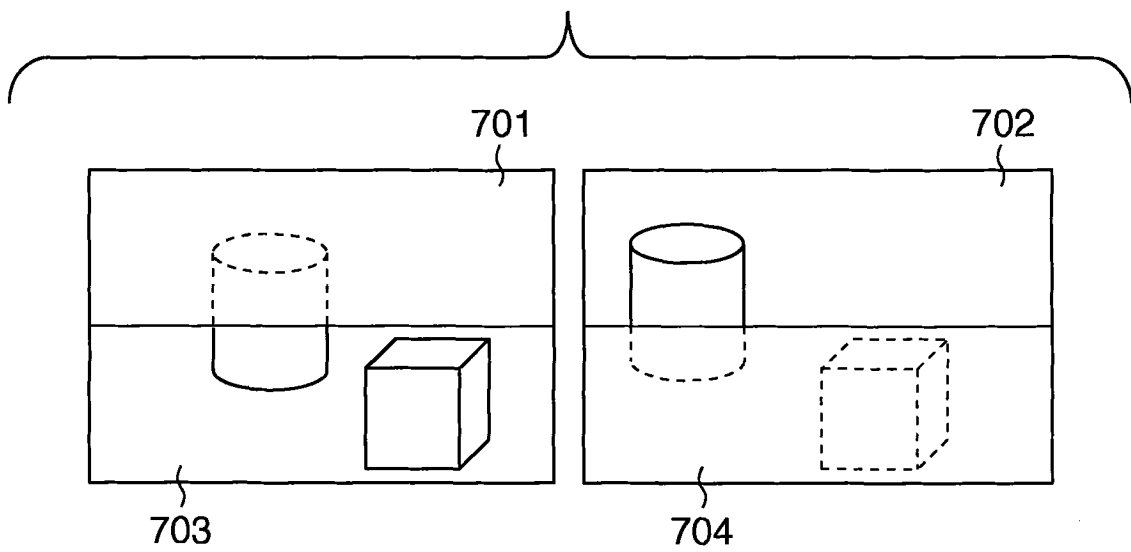


图 7

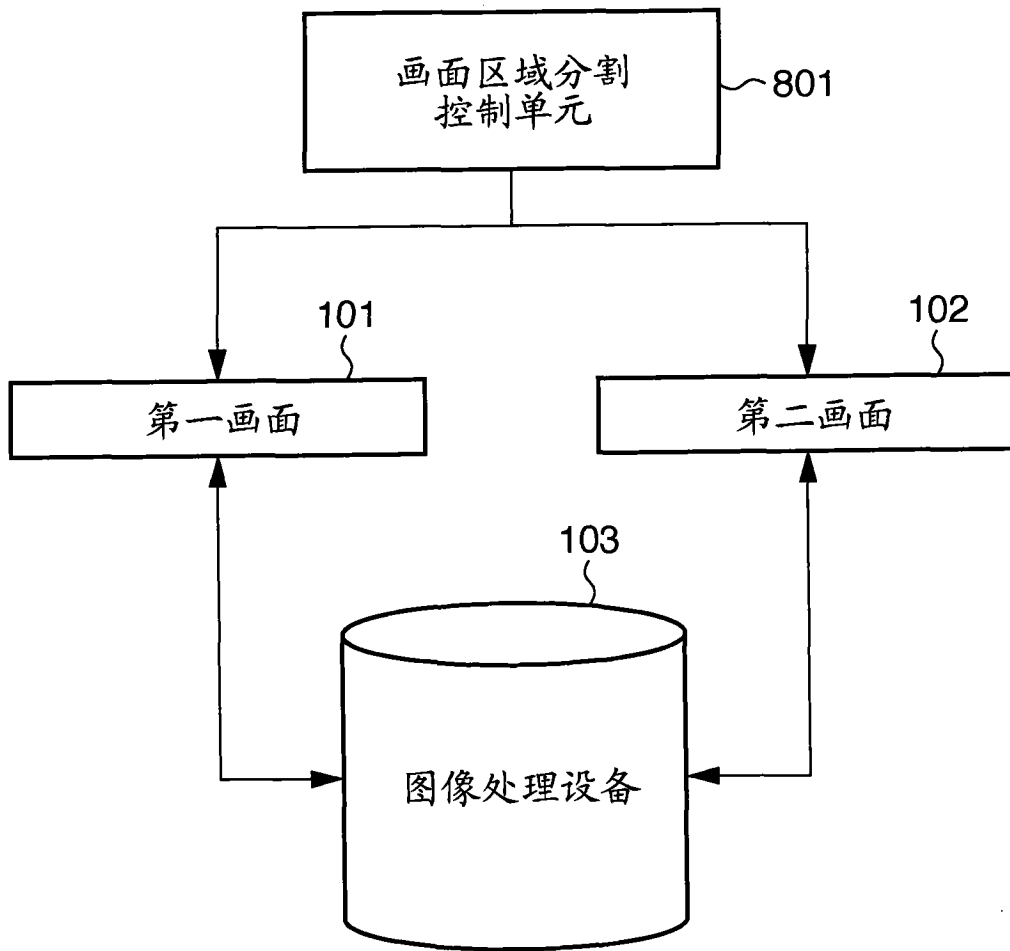


图 8

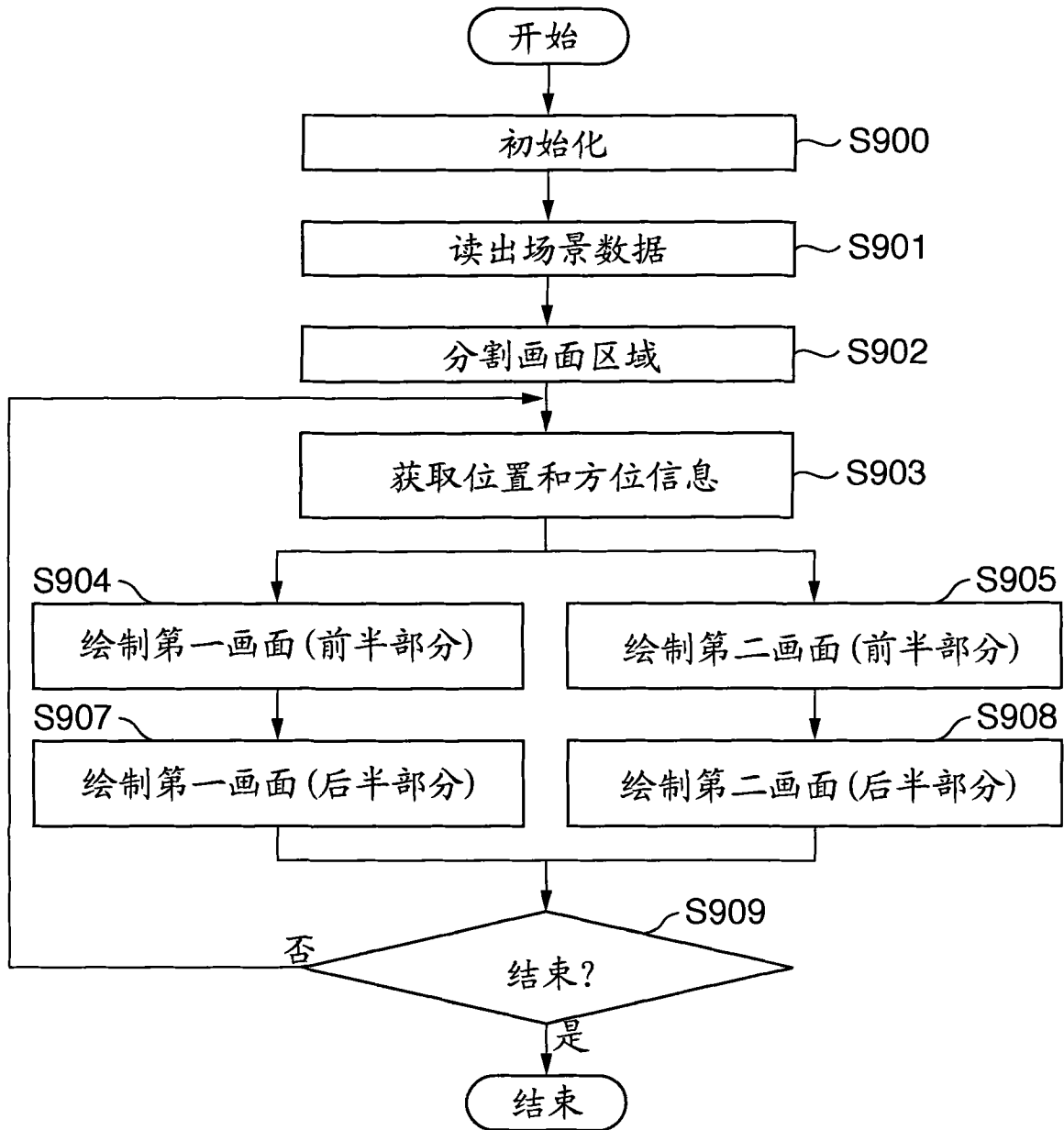


图 9

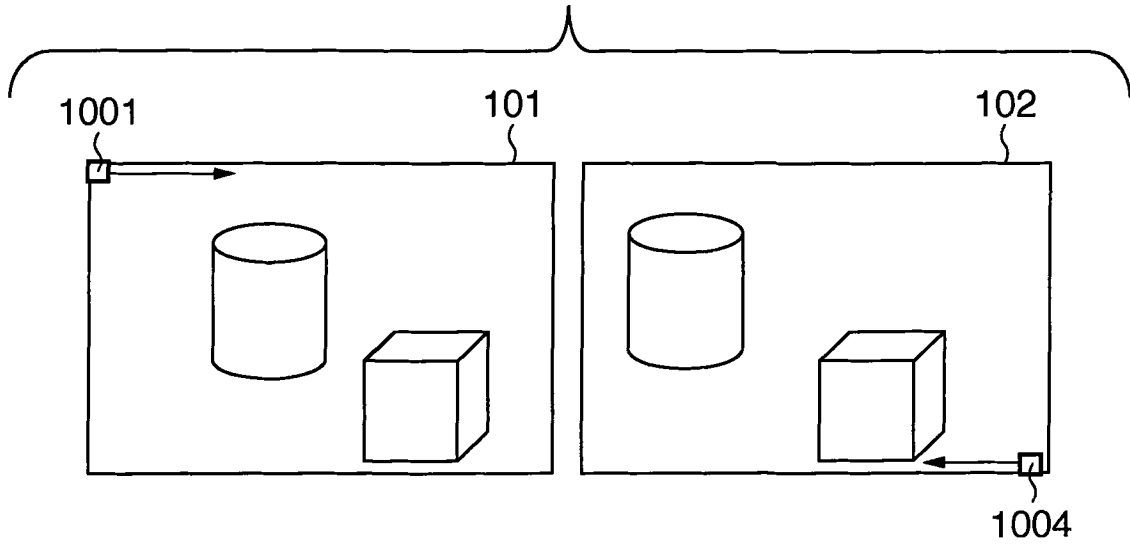


图 10A

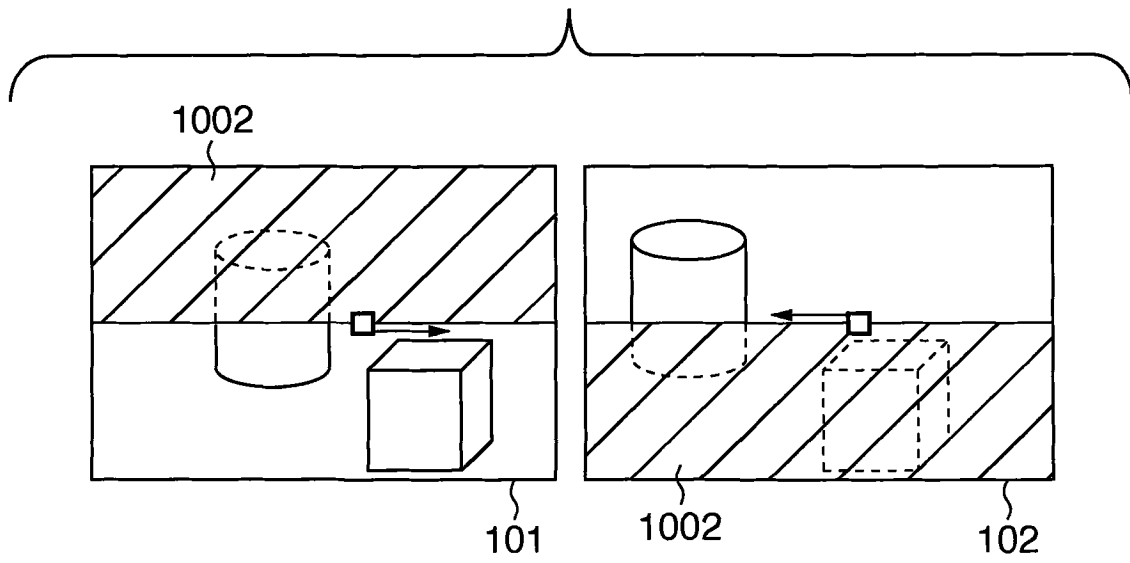


图 10B

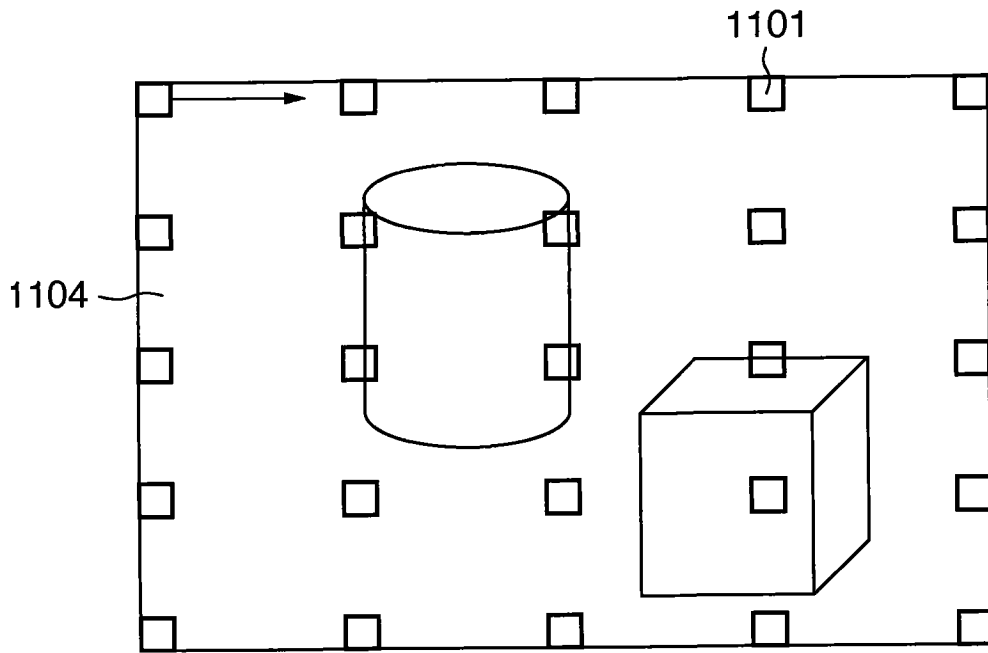


图 11

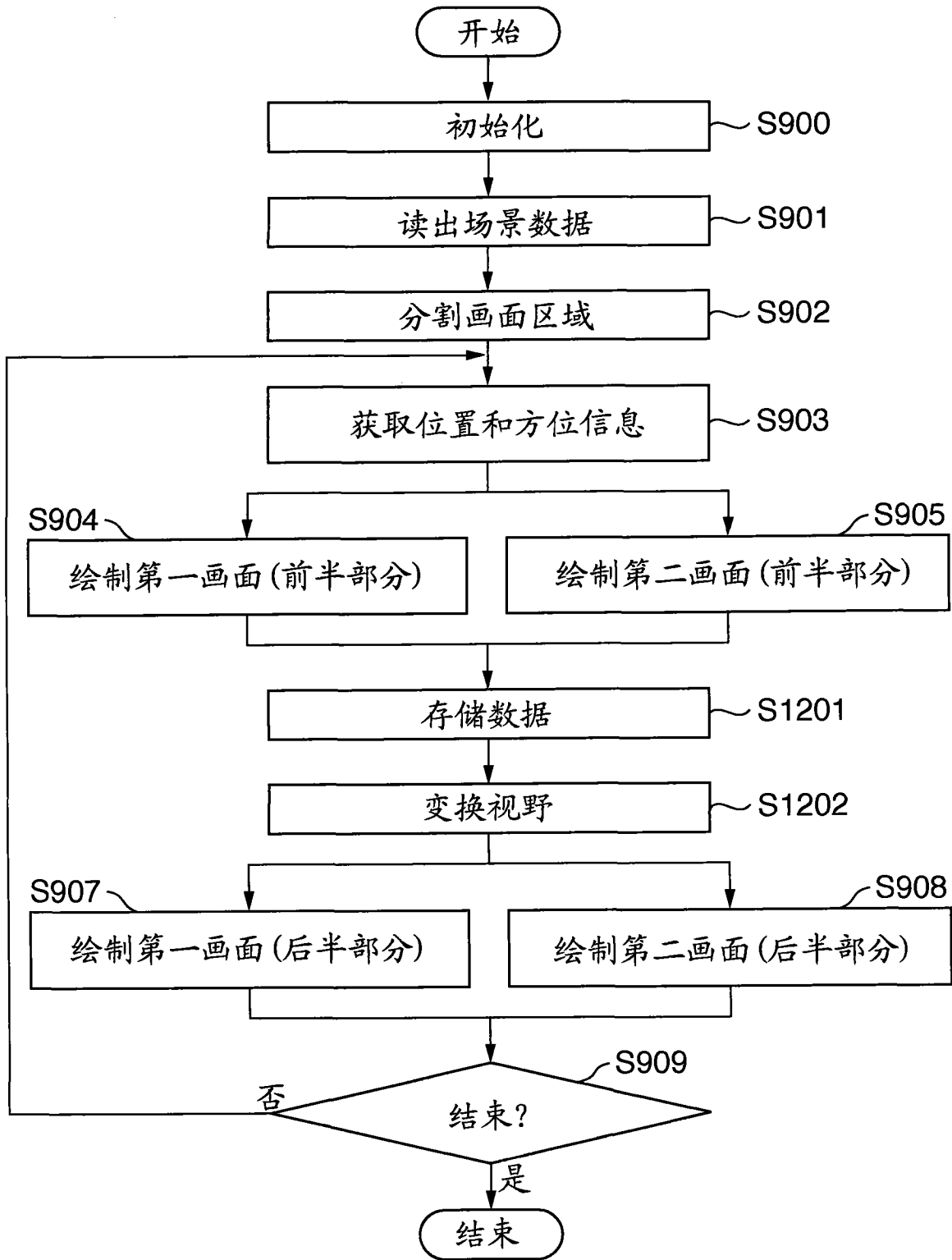


图 12

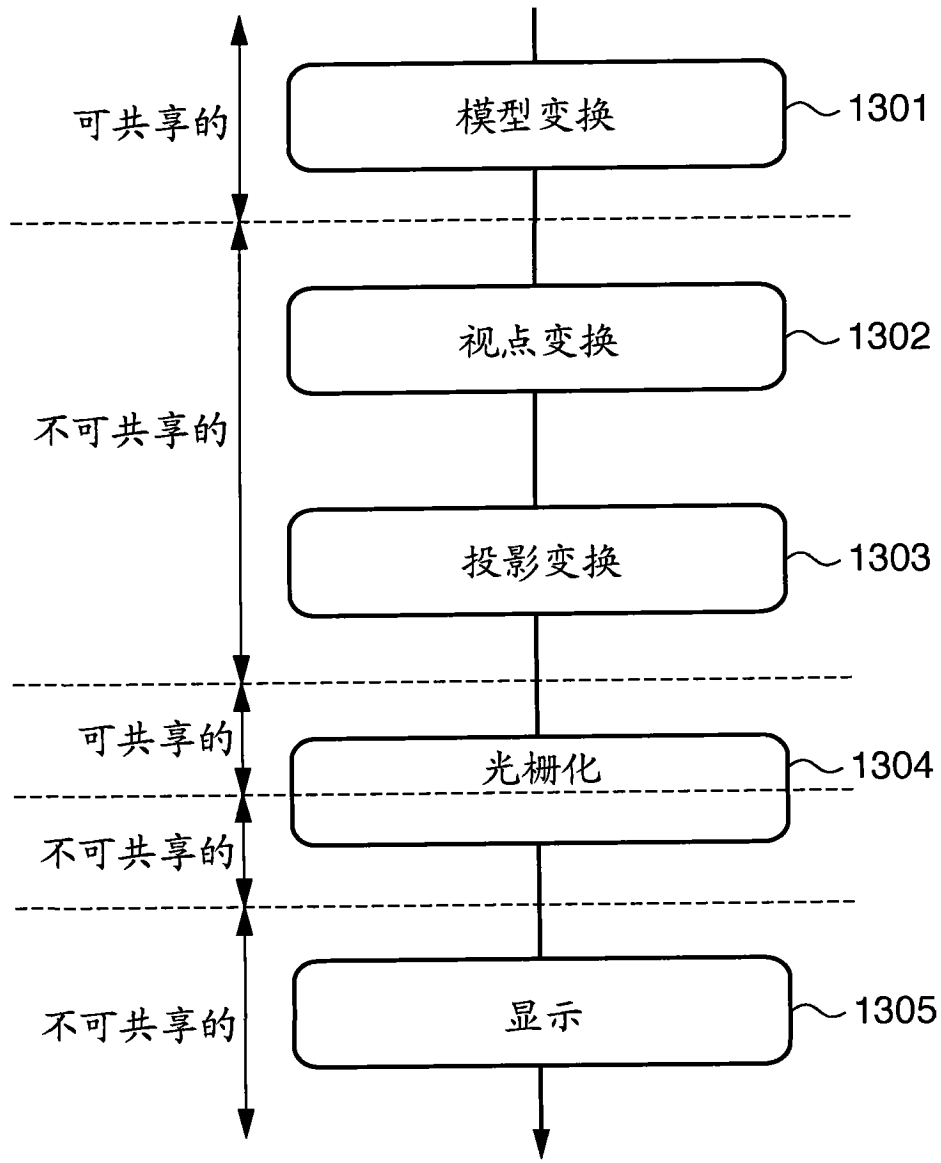


图 13

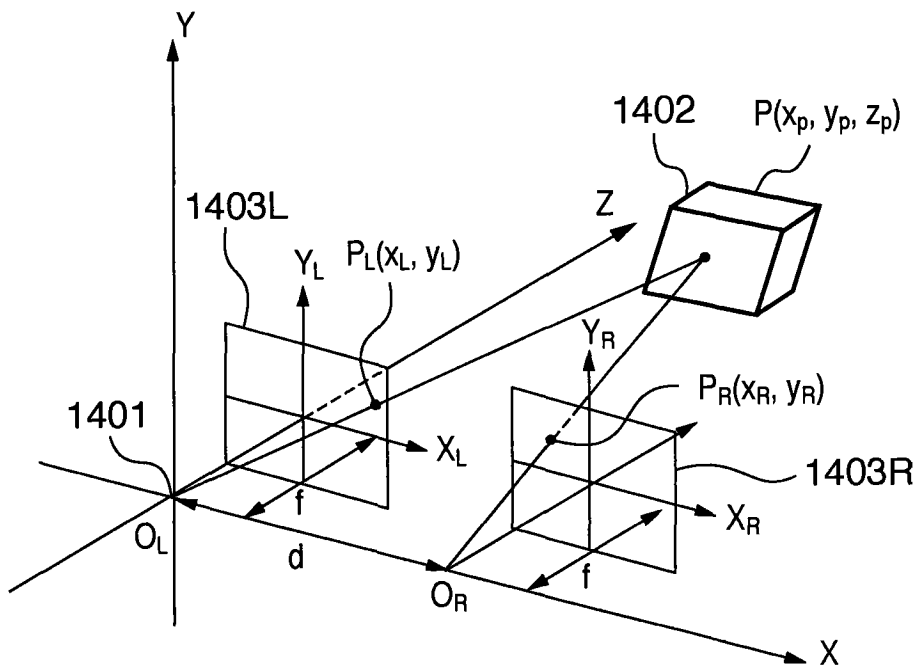


图 14

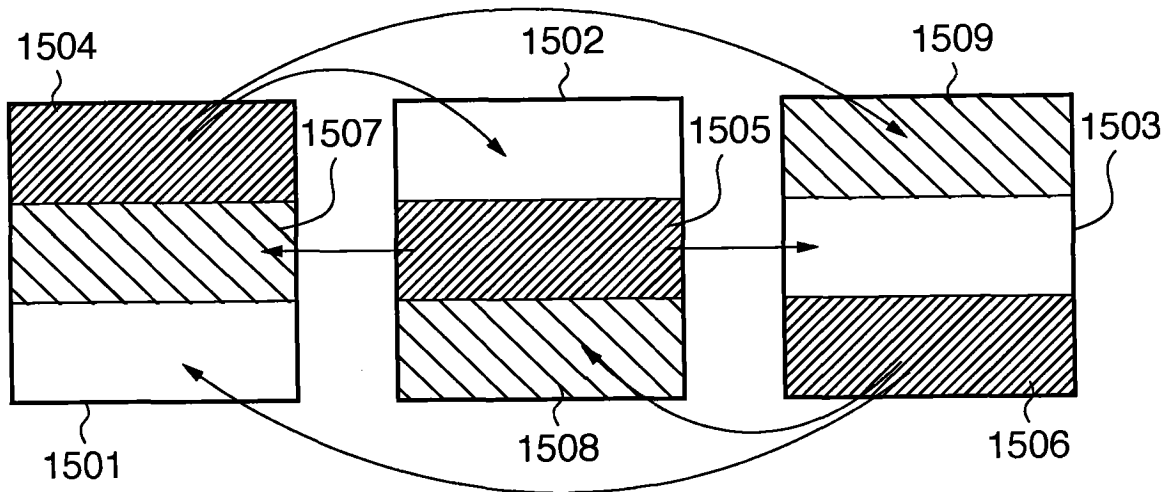


图 15

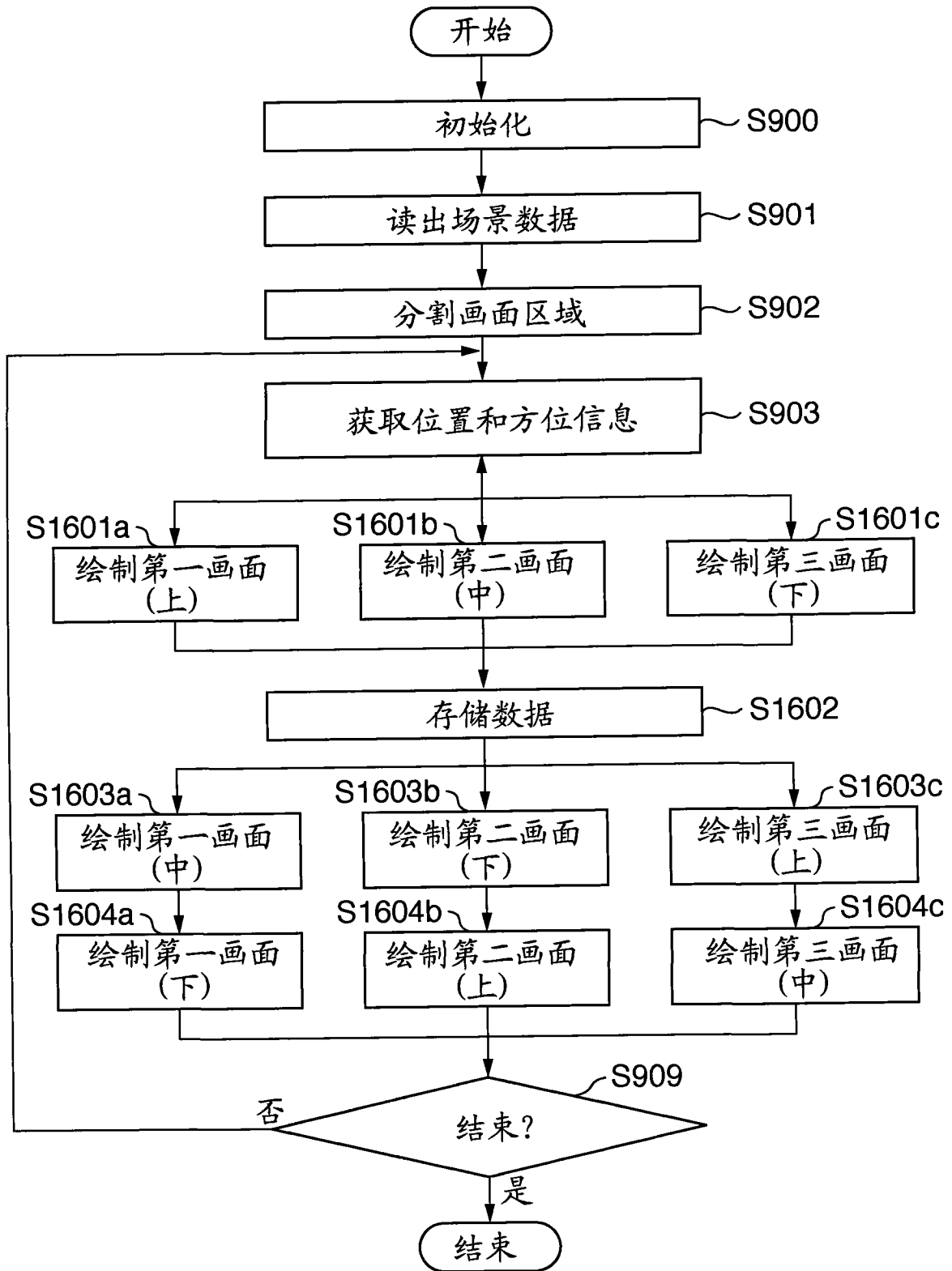


图 16