



# (12)发明专利

(10)授权公告号 CN 107005689 B

(45)授权公告日 2019.06.14

(21)申请号 201480083825.2  
 (22)申请日 2014.12.03  
 (65)同一申请的已公布的文献号  
 申请公布号 CN 107005689 A  
 (43)申请公布日 2017.08.01  
 (85)PCT国际申请进入国家阶段日  
 2017.06.02  
 (86)PCT国际申请的申请数据  
 PCT/FI2014/050946 2014.12.03  
 (87)PCT国际申请的公布数据  
 W02016/087702 EN 2016.06.09  
 (73)专利权人 诺基亚技术有限公司  
 地址 芬兰埃斯波  
 (72)发明人 M·涅梅拉 K·格隆霍姆  
 A·鲍德温  
 (74)专利代理机构 北京市金杜律师事务所  
 11256  
 代理人 鄢迅 黄捷

(51)Int.Cl.  
*H04N 13/15*(2018.01)  
*H04N 13/128*(2018.01)  
*G06T 15/20*(2011.01)  
*G06T 15/40*(2011.01)  
 (56)对比文件  
 CN 102204262 A,2011.09.28,  
 CN 1853200 A,2006.10.25,  
 US 2007126753 A1,2007.06.07,  
 US 2008094395 A1,2008.04.24,  
 US 2008079719 A1,2008.04.03,  
 COHEN-OR,D.et al.《A survey of  
 visibility for walkthrough applications》.  
 《IEEE Trans.on Visualization and Computer  
 Graphics》.2003,第9卷(第3期),第412-431页.

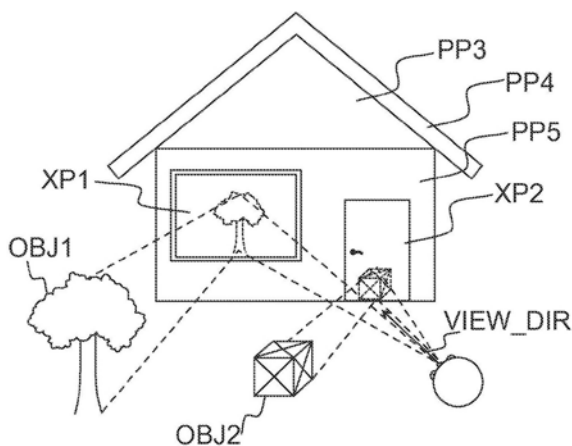
审查员 李雁

权利要求书7页 说明书15页 附图14页

(54)发明名称  
 数字视频渲染

(57)摘要

本发明涉及用于渲染数字图像和视频内容的方法、装置和计算机程序产品。用于渲染第二图形对象的第一图形对象的第一信息和用于渲染第二图形对象的第二图形对象的第二信息被形成或接收。第一图形对象和第二图形对象之间的依从性信息被形成或接收,其中,该依从性信息包括用于根据第一图形对象和第二图形对象在观看方向上的重叠来渲染第二图形对象的信息。通过使用第二图形对象的第二信息和第一图形对象的第一信息来创建第二像素数据来渲染第二图形对象,其中,针对与第一图形对象在观看方向上重叠的第二图形对象的这一部分进行第二图形对象的渲染。



1. 一种用于渲染图形对象的方法,包括:
  - 接收用于渲染第二图形对象的第一图形对象的第一信息,
  - 接收用于渲染所述第二图形对象的所述第二图形对象的第二信息,
  - 接收所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及
    - 通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来渲染所述第二图形对象,其中针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行所述第二图形对象的所述渲染。
2. 根据权利要求1所述的方法,其中渲染所述第二图形对象包括:使用所述第一图形对象的反射信息来确定在通过所述第一图形对象渲染所述第二图形对象时所述第一图形对象的透明度。
3. 根据权利要求1所述的方法,其中所述第二图形对象是第三对象的镜像,并且所述方法包括:
  - 通过相对于所述第一图形对象的镜面对所述第三对象进行镜像来形成所述第二图形对象的所述第二信息。
4. 根据权利要求1至3中的任一项所述的方法,其中所述依从性信息是指示图形对象通过其他图形对象的可见性的依从性图形,并且所述方法包括:
  - 通过使用所述第一图形对象的所述第一信息创建第一像素的第一像素数据来渲染所述第一图形对象,
  - 通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素的第二像素数据来渲染所述第二图形对象,并且所述第二像素是所述第一像素的子集。
5. 根据权利要求4所述的方法,包括:
  - 使用模版缓冲器来标记要被渲染的第二像素,以及
  - 通过创建与所述模版缓冲器的所述标记相对应的像素数据来渲染所述第二图形对象。
6. 根据权利要求1至3中的任一项所述的方法,其中所述第一图形对象包括第一像素,所述第二图形对象包括渲染层,并且所述方法包括:
  - 确定所述第一像素是否是所述第二图形对象通过其可见的门户像素,以及
  - 在所述确定指示所述第一像素是门户像素的情况下,使用所述第一信息和所述第二信息来渲染所述第一像素。
7. 根据权利要求1至3中的任一项所述的方法,其中对象的可见性通过所述依从性信息被限制在其他对象之外,使得一些对象仅通过其他对象可见并且不在其它对象外部,并且所述方法包括:通过混合来自所述第一图形对象的颜色信息和来自所述第二图形对象的颜色信息来形成所述第二像素数据。
8. 根据权利要求1至3中的任一项所述的方法,其中所述依从性图形是依从性树,所述第一图形对象和所述第二图形对象分别与所述依从性树的第一节点和第二节点相对应,使得所述第二节点是所述第一节点的子节点,并且所述依从性树确定所述第二图形对象仅通

过所述第一图形对象而非其他对象可见。

9. 根据权利要求1至3中的任一项所述的方法,其中所述第一信息和所述第二信息包括所述第一图形对象和所述第二图形对象的一个或多个图形片段的信息,并且所述第一图形对象的一个或多个图形片段是所述第二图形对象的片段通过其可见的特殊片段。

10. 根据权利要求1至3中的任一项所述的方法,包括:

-接收包括所述第一图形对象的第一渲染层和包括所述第二图形对象的第二渲染层,所述第一渲染层和所述第二渲染层包括像素,所述第一渲染层包括与从渲染视点观看的所述第一图形对象相对应的像素,并且所述第二渲染层包括与从所述渲染视点观看的所述第二图形对象相对应的像素,其中所述第二图形对象被从所述渲染视点观看的所述第一图形对象部分地遮蔽,

-将所述第一渲染层的像素和所述第二渲染层的像素放置在渲染空间中,

-将深度值与所述像素相关联,以及

-使用所述像素和所述深度值来渲染左眼图像和右眼图像。

11. 一种用于渲染图形对象的方法,包括:

-形成用于渲染第二图形对象的第一图形对象的第一信息,

-形成用于渲染所述第二图形对象的所述第二图形对象的第二信息,

-形成所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来提供要在渲染所述第二图形对象中使用的所述第一信息、所述第二信息以及所述依从性信息,其中所述第二图形对象的所述渲染被提供以针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行。

12. 根据权利要求11所述的方法,其中所述第一图形对象包括用于在通过所述第一图形对象渲染所述第二图形对象时确定所述第一图形对象的透明度的反射信息。

13. 根据权利要求11所述的方法,其中所述第二图形对象是第三对象的镜像,并且所述方法包括:

-通过相对于所述第一图形对象的镜面对所述第三对象进行镜像来形成所述第二图形对象的所述第二信息。

14. 根据权利要求11至13中的任一项所述的方法,其中所述依从性信息是指示图形对象通过其他图形对象的可见性的依从性图形。

15. 根据权利要求14所述的方法,包括:

-提供用于使用模版缓冲器来标记要被渲染的像素的信息,用于通过创建与所述模版缓冲器的所述标记相对应的像素数据来渲染所述第二图形对象。

16. 根据权利要求11至13中的任一项所述的方法,其中所述第一图形对象包括第一像素,并且所述第二图形对象包括渲染层,并且所述方法包括:

-提供所述第一像素是所述第二图形对象通过其可见的门户像素的信息,用于在所述第一像素是门户像素的情况下,使用所述第一信息和所述第二信息来渲染所述第一像素。

17. 根据权利要求11至13中的任一项所述的方法,其中对象的可见性通过所述依从性

信息被限制在其他对象外部,使得一些对象仅通过其他对象可见,并且不在其他对象外部。

18. 根据权利要求11至13中的任一项所述的方法,其中所述依从性图形是依从性树,所述第一图形对象和所述第二图形对象分别与所述依从性树的第一节点和第二节点相对应,使得所述第二节点是所述第一节点的子节点,并且所述依从性树确定所述第二图形对象仅通过所述第一图形对象而非其他对象可见。

19. 根据权利要求11至13中的任一项所述的方法,其中所述第一信息和所述第二信息包括所述第一图形对象和所述第二图形对象的一个或多个图形片段的信息,并且所述第一图形对象的一个或多个图形片段是所述第二图形对象的片段通过其可见的特殊片段。

20. 根据权利要求11至13中的任一项所述的方法,包括:

-形成包括所述第一图形对象的第一渲染层和包括所述第二图形对象的第二渲染层,所述第一渲染层和所述第二渲染层包括像素,所述第一渲染层包括与从渲染视点观看的所述第一图形对象相对应的像素,并且所述第二渲染层包括与从所述渲染视点观看的所述第二图形对象相对应的像素,其中所述第二图形对象被从所述渲染视点观看的所述第一图形对象部分地遮蔽,

-提供映射参数,所述映射参数用于将所述第一渲染层的像素和所述第二渲染层的像素放置在渲染空间中。

21. 一种用于渲染图形对象的装置,包括至少一个处理器、包括计算机程序代码的存储器,所述存储器和所述计算机程序代码被配置成与所述至少一个处理器一起使所述装置至少执行以下操作:

-接收用于渲染第二图形对象的第一图形对象的第一信息,

-接收用于渲染所述第二图形对象的所述第二图形对象的第二信息,

-接收所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来渲染所述第二图形对象,其中针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行所述第二图形对象的所述渲染。

22. 根据权利要求21所述的装置,其中渲染所述第二图形对象被布置成包括:使用所述第一图形对象的反射信息来确定在通过所述第一图形对象渲染所述第二图形对象时所述第一图形对象的透明度。

23. 根据权利要求21所述的装置,其中所述第二图形对象是第三对象的镜像,并且所述装置包括计算机程序代码,以使所述装置:

-通过相对于所述第一图形对象的镜面对所述第三对象进行镜像来形成所述第二图形对象的所述第二信息。

24. 根据权利要求21至23中的任一项所述的装置,其中所述依从性信息是指示图形对象通过其他图形对象的可见性的依从性图形,并且所述装置包括计算机程序代码,以使所述装置:

-通过使用所述第一图形对象的所述第一信息创建第一像素的第一像素数据来渲染所述第一图形对象,

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素的第二像素数据来渲染所述第二图形对象,并且所述第二像素是所述第一像素的子集。

25. 根据权利要求24所述的装置,包括计算机程序代码,用于使所述装置:

-使用模版缓冲器来标记要被渲染的第二像素,以及  
-通过创建与所述模版缓冲器的所述标记相对应的像素数据来渲染所述第二图形对象。

26. 根据权利要求21至23中的任一项所述的装置,其中所述第一图形对象包括第一像素,并且所述第二图形对象包括渲染层,并且所述装置包括计算机程序代码,以使得所述装置:

-确定所述第一像素是否是所述第二图形对象通过其可见的门户像素,以及  
-在所述确定指示所述第一像素是门户像素的情况下,使用所述第一信息和所述第二信息来渲染所述第一像素。

27. 根据权利要求21至23中的任一项所述的装置,其中对象的可见性通过所述依从性信息被布置成被限制在其他对象外部,使得一些对象仅通过其他对象可见,并且不在其它对象外部,并且所述装置包括计算机程序代码,以使所述装置:通过混合来自所述第一图形对象的颜色信息和来自所述第二图形对象的颜色信息来形成所述第二像素数据。

28. 根据权利要求21至23中的任一项所述的装置,其中所述依从性图形是依从性树,所述第一图形对象和所述第二图形对象分别与所述依从性树的第一节点和第二节点相对应,使得所述第二节点是所述第一节点的子节点,并且所述依从性树确定所述第二图形对象仅通过所述第一图形对象而非其他对象可见。

29. 根据权利要求21至23中的任一项所述的装置,其中所述第一信息和所述第二信息包括所述第一图形对象和所述第二图形对象的一个或多个图形片段的信息,并且所述第一图形对象的一个或多个图形片段是所述第二图形对象的片段通过其可见的特殊片段。

30. 根据权利要求21至23中的任一项所述的装置,包括计算机程序代码,以使得所述装置:

-接收包括所述第一图形对象的第一渲染层和包括所述第二图形对象的第二渲染层,所述第一渲染层和所述第二渲染层包括像素,所述第一渲染层包括与从渲染视点观看的所述第一图形对象相对应的像素,并且所述第二渲染层包括与从所述渲染视点观看的所述第二图形对象相对应的像素,其中所述第二图形对象被从所述渲染视点观看的所述第一图形对象部分地遮蔽,

-将所述第一渲染层的像素和所述第二渲染层的像素放置在渲染空间中,

-将深度值与所述像素相关联,以及

-使用所述像素和所述深度值来渲染左眼图像和右眼图像。

31. 一种用于渲染图形对象的装置,包括至少一个处理器、包括计算机程序代码的存储器,所述存储器和所述计算机程序代码被配置成与所述至少一个处理器一起使得所述装置至少执行以下操作:

-形成用于渲染第二图形对象的第一图形对象的第一信息,

-形成用于渲染所述第二图形对象的所述第二图形对象的第二信息,

-形成所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来提供要在渲染所述第二图形对象中使用的所述第一信息、所述第二信息以及所述依从性信息,其中所述第二图形对象的所述渲染被提供以针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行。

32. 根据权利要求31所述的装置,其中所述第一图形对象包括用于在通过所述第一图形对象渲染所述第二图形对象时确定所述第一图形对象的透明度的反射信息。

33. 根据权利要求31所述的装置,其中所述第二图形对象是第三对象的镜像,并且所述装置包括计算机程序代码,以使得所述装置:

-通过相对于所述第一图形对象的镜面对所述第三对象进行镜像来形成所述第二图形对象的所述第二信息。

34. 根据权利要求31至33中的任一项所述的装置,其中所述依从性信息是指示图形对象通过其他图形对象的可见性的依从性图形。

35. 根据权利要求34所述的装置,包括计算机程序代码,以使得所述装置:

-提供用于使用模版缓冲器来标记要被渲染的像素的信息,用于通过创建与所述模版缓冲器的所述标记相对应的像素数据来渲染所述第二图形对象。

36. 根据权利要求31至33中的任一项所述的装置,其中所述第一图形对象包括第一像素,并且所述第二图形对象包括渲染层,并且所述装置包括计算机程序代码,以使得所述装置:

-提供所述第一像素是所述第二图形对象通过其可见的门户像素的信息,用于在所述第一像素是门户像素的情况下,使用所述第一信息和所述第二信息来渲染所述第一像素。

37. 根据权利要求31至33中的任一项所述的装置,其中对象的可见性通过所述依从性信息被限制在其他对象外部,使得一些对象仅通过其他对象可见,并且不在其他对象外部。

38. 根据权利要求31至33中的任一项所述的装置,其中所述依从性图形是依从性树,所述第一图形对象和所述第二图形对象分别与所述依从性树的第一节点和第二节点相对应,使得所述第二节点是所述第一节点的子节点,并且所述依从性树确定所述第二图形对象仅通过所述第一图形对象而非其他对象可见。

39. 根据权利要求31至33中的任一项所述的装置,其中所述第一信息和所述第二信息包括所述第一图形对象和所述第二图形对象的一个或多个图形片段的信息,并且所述第一图形对象的一个或多个图形片段是所述第二图形对象的片段通过其可见的特殊片段。

40. 根据权利要求31至33中的任一项所述的装置,包括计算机程序代码,以使得所述装置:

-形成包括所述第一图形对象的第一渲染层和包括所述第二图形对象的第二渲染层,所述第一渲染层和所述第二渲染层包括像素,所述第一渲染层包括与从渲染视点观看的所述第一图形对象相对应的像素,并且所述第二渲染层包括与从所述渲染视点观看的所述第二图形对象相对应的像素,其中所述第二图形对象被从所述渲染视点观看的所述第一图形对象部分地遮蔽,

-提供映射参数,所述映射参数用于将所述第一渲染层的像素和所述第二渲染层的像素放置在渲染空间中。

41.一种用于渲染图形对象的系统,包括至少一个处理器、包括计算机程序代码的存储器,所述存储器和所述计算机程序代码被配置成与所述至少一个处理器一起使得所述系统至少执行以下操作:

- 接收用于渲染第二图形对象的第一图形对象的第一信息,
- 接收用于渲染所述第二图形对象的所述第二图形对象的第二信息,
- 接收所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来渲染所述第二图形对象,其中针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行所述第二图形对象的所述渲染。

42.一种用于渲染图形对象的系统,包括至少一个处理器、包括计算机程序代码的存储器,所述存储器和所述计算机程序代码被配置成与所述至少一个处理器一起使得所述系统至少执行以下操作:

- 形成用于渲染第二图形对象的第一图形对象的第一信息,
- 形成用于渲染所述第二图形对象的所述第二图形对象的第二信息,
- 形成所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来提供要在渲染所述第二图形对象中使用的所述第一信息、所述第二信息以及所述依从性信息,其中所述第二图形对象的所述渲染被提供以针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行。

43.一种用于渲染图形对象的装置,包括:

- 用于接收用于渲染第二图形对象的第一图形对象的第一信息的器件;
- 用于接收用于渲染所述第二图形对象的所述第二图形对象的第二信息的器件;
- 用于接收所述第一图形对象和所述第二图形对象之间的依从性信息的器件,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息;以及

-用于通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来渲染所述第二图形对象的器件,其中针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行所述第二图形对象的所述渲染。

44.一种用于渲染图形对象的装置,包括:

- 用于形成用于渲染第二图形对象的第一图形对象的第一信息的器件;
- 用于形成用于渲染所述第二图形对象的所述第二图形对象的第二信息的器件;
- 用于形成所述第一图形对象和所述第二图形对象之间的依从性信息的器件,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲

染所述第二图形对象的信息,以及

-用于通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来提供要在渲染所述第二图形对象中使用的所述第一信息、所述第二信息以及所述依从性信息的器件,其中所述第二图形对象的所述渲染被提供以针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行。

45. 一种非暂态计算机可读介质,包括其上存储的计算机程序代码,所述计算机程序代码被配置成当在至少一个处理器上被执行时,使得装置或系统:

- 接收用于渲染第二图形对象的第一图形对象的第一信息,
- 接收用于渲染所述第二图形对象的所述第二图形对象的第二信息,
- 接收所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来渲染所述第二图形对象,其中针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行所述第二图形对象的所述渲染。

46. 一种非暂态计算机可读介质,包括其上存储的计算机程序代码,所述计算机程序代码被配置成当在至少一个处理器上被执行时,使得装置或系统:

- 形成用于渲染第二图形对象的第一图形对象的第一信息,
- 形成用于渲染所述第二图形对象的所述第二图形对象的第二信息,
- 形成所述第一图形对象和所述第二图形对象之间的依从性信息,所述依从性信息包括用于根据所述第一图形对象和所述第二图形对象在观看方向上的重叠来渲染所述第二图形对象的信息,以及

-通过使用所述第二图形对象的所述第二信息和所述第一图形对象的所述第一信息创建第二像素数据来提供要在渲染所述第二图形对象中使用的所述第一信息、所述第二信息以及所述依从性信息,其中所述第二图形对象的所述渲染被提供以针对在所述观看方向上与所述第一图形对象重叠的所述第二图形对象的这样的部分进行。

## 数字视频渲染

### 背景技术

[0001] 静止图像和运动图像的数字立体观看已经变得司空见惯,用于观看3D(三维)电影的设备更广泛地可用。剧院正在基于使用特殊眼镜观看电影来提供3D电影,该特殊眼睛确保看到电影的每帧的左眼和右眼的不同的图像。相同的方法已经被引入到与3D功能玩家和电视一起使用的家用场合。在实践中,电影由相同场景的两幅视图组成,一个是左眼视图,而一个是右眼视图。这些视图已经通过使用直接创建适合于立体观看的该内容的特殊立体相机拍摄电影得以创建。当视图被呈现给两只眼睛时,人类视觉系统创建了场景的3D视图。该技术的缺点在于观看区域(电影屏幕或电视)仅占据视野的一部分,因此,3D视图的体验受到限制。

[0002] 为了更逼真的体验,占据整个视野的较大观看区域的设备已经被创建。存在可用的特殊立体观看眼镜,其意指被戴在头上,使得它们覆盖眼睛,并且用小屏幕和镜头布置显示左眼和右眼的画面。与通常用于3D观看的相当大的电视机相比,这种技术的优点还在于它可以在很小的空间中使用,甚至在移动的同时使用。

[0003] 因此,存在需要使得能够记录并且渲染数字图像/视频以用于观看3D视频的解决方案。

### 发明内容

[0004] 现在,已经发明了一种改进的方法,和实现该方法的技术设备,通过它们缓解了上述问题。本发明的各个方面包括一种方法、装置、服务器、渲染器、数据结构和包括其中存储的计算机程序的计算机可读介质,其特征在于独立权利要求中所述的内容。在从属权利要求中公开了本发明的各种实施例。

[0005] 本描述公开了用于渲染数字图像和视频内容的方法、装置和计算机程序产品。形成或接收用于渲染第二图形对象的第一图形对象的第一信息和用于渲染第二图形对象的第二图形对象的第二信息。形成或接收第一图形对象和第二图形对象之间的依从性信息,其中,该依从性信息包括用于根据所述第一图形对象和第二图形对象在观看方向上的重叠来渲染第二图形对象的信息。通过使用第二图形对象的第二信息和第一图形对象的第一信息来创建第二像素数据来渲染第二图形对象,其中,对于与第一图形对象在观看方向上重叠的第二图形对象的这一部分进行第二图形对象的渲染。

[0006] 换句话说,提供了方法、装置和计算机程序产品,其中,当通过其他对象看到场景的计算机模型中的一些对象而非其他情况时,被渲染到包含像素的计算机图像。例如,可以通过窗户而非通过墙壁看到房屋内灯具的模型。另外,如果房屋的模型只包含前面(正面)墙壁,那么即使当从侧面以倾斜角度观看房屋时灯具在墙壁的一侧上是可见的,也防止它被渲染。还有,当另一对象在当前观看方向上在反射镜中可能被看到时,像镜子那样的透明对象还可以通过镜子中渲染另一对象的镜像来渲染。这简化了渲染,因为可以镜像的对象被预先计算。然而,其他对象的镜像当它可能在镜子的侧面上可见时将不会被渲染,因为这样会在没有反射的地方错误地示出反射。对通过其他对象的对象的渲染可以被控制,使

得对象仅当它们与前景对象重叠并且可以通过前景对象被看到时才被渲染。模版缓冲器可以用于这种条件渲染,即,如果该画面缓冲器像素的模版缓冲器像素被适当地设置,则像素数据被输出到画面缓冲器。

### 附图说明

- [0007] 在下文中,将参照附图对本发明的各种实施例进行更详细地描述,其中
- [0008] 在下文中,将参照附图对本发明的各种实施例进行更详细地描述,其中
- [0009] 图1a、图1b、图1c和图1d示出了向用户形成立体图像的设置;
- [0010] 图2a示出了用于立体观看的系统 and 装置;
- [0011] 图2b示出了用于立体观看的立体相机设备;
- [0012] 图2c示出用于立体观看的头戴式显示器;
- [0013] 图2d示出了相机;
- [0014] 图3a和图3b图示了形成渲染层并且形成用于存储的图像数据;
- [0015] 图4图示了使用渲染层渲染图像;
- [0016] 图5a和图5b描绘了包括用于渲染图像的渲染层的数据结构;
- [0017] 图6a示出了具有部分反射的对象的场景;
- [0018] 图6b示出了用于渲染反射对象和半透明对象的依从性图形;
- [0019] 图6c示出了渲染反射对象;
- [0020] 图6d图示了使用模版缓冲器来渲染反射对象;以及
- [0021] 图7a和图7b示出了用于形成和渲染具有反射对象的数字场景的流程图。

### 具体实施方式

[0022] 在下文中,将在立体观看的上下文中对本发明的几个实施例进行了描述。然而,应当指出,本发明不限于任何特定目的。事实上,不同的实施例具有在需要观看具有反射表面的数字场景的任何环境中的应用(例如,电影和电视)。

[0023] 图1a、图1b、图1c和图1d示出了向用户形成立体图像的设置。在图1a中,示出了人正在使用双眼E1和E2观察两个球体A1和A2的情形。球体A1比球体A2更靠近观看者,距第一眼E1的相应距离为LE1,A1和LE1,A2。不同的对象在空间中存在于由坐标系SZ,SY和SZ定义的它们相应的坐标(x,y,z)处。人的眼睛之间的距离d12可以平均约为62mm至64mm,并且因人而异,在55mm和74mm之间变化。该距离被称为视差,人的视觉的具有立体感的视图基于该视差。观看方向(光轴)DIR1和DIR2通常基本上是平行的,其可能具有与平行的小偏差,并且限定了眼睛的视野。用户的头部具有相对于周围环境的取向(头部取向),其当眼睛在直视前方时最容易由眼睛的共同方向限定。也就是说,头部取向告知关于用户所在场景的坐标系的头部的偏航、俯仰和滚转。

[0024] 在图1a的设置中,球体A1和A2在双眼的视野中。球体和眼睛之间的中心点O12位于同一条线上。也就是说,从中心点,球体A2在球体A1的后面。然而,每只眼睛从A1后面看到球体A2的部分,因为球体与任何一只眼睛都不在同一视线上。

[0025] 在图1b中,示出了其中眼睛已经被相机C1和C2代替的设置,其位于图1a中眼睛所在的位置。在其他方面,设置的距离和方向相同。自然地,图1b的设置目的是能够拍摄球体

A1和A2的立体图像。由图像捕捉产生的两个图像是FC1和FC2。“左眼”图像FC1示出了在球体A1的图像SA1的左侧部分可见的球体A2的图像SA2。“右眼”图像FC2示出了在球体A1的图像SA1的右侧部分可见的球体A2的图像SA2。右图像和左图像之间的差异被称为双眼像差异 (disparity),而这种双眼像差异是人类视觉系统确定深度信息并且创建场景的3D视图的基本机制,可以用于创建3D图像的错觉。

[0026] 在图1b的这个设置中,其中眼睛间距离与图1a中的眼睛的眼睛间距离相对应,相机对C1和C2具有自然视差(parallax),即,它具有在相机的两个图像中创建自然双眼像差异的特性。自然双眼像差异可以被理解为即使形成立体相机对的两个相机之间的距离稍微小于或大于人眼之间的正常距离(视差),例如,基本上介于40mm和100mm之间,甚至介于30mm和120mm之间,也被创建。

[0027] 在图1c中,示出了这种3D错觉的创建。由相机C1和C2捕捉的图像FC1和FC2分别使用显示器D1和D2显示给眼睛E1和E2。图像之间的双眼像差异通过人类视觉系统处理,从而产生对深度的理解。也就是说,当左眼看到球体A2在球体A1的图像SA1的左侧的图像SA2并且右眼分别看到右侧的A2的图像时,人类视觉系统产生了以下认识:在三维世界中的球体V1后面存在球体V2。这里,需要理解的是,图像FC1和FC2也可以是合成的,即,由计算机创建的。如果它们携带双眼像差异信息,则合成图像也将被人类视觉系统视为三维的。也就是说,可以形成一对计算机生成图像,使得它们可以用作立体图像。

[0028] 图1d图示了如何将立体图像显示给眼睛的原理可以用于创建具有三维错觉的3D电影或虚拟现实场景。图像FX1和FX2用立体相机捕捉或从模型中计算,以使图像具有适当的双眼像差异。通过使用显示器D1和D2每秒钟向双眼显示大量(例如,30个)帧,使得左眼和右眼之间的图像具有双眼像差异,人类视觉系统将产生对移动的三维图像的认知。当相机转动或者计算合成图像的观看方向改变时,图像的改变产生观看方向改变的错觉,即,观看者的头部转动的错觉。该观看方向,即头部取向,可以例如通过安装在头部上的取向检测器被确定为头部的真实取向,或者通过诸如操纵杆或鼠标之类的控制设备被确定为虚拟取向,该控制设备可以用于操纵观看方向,而不用用户实际上移动其头部。也就是说,术语“头部取向”可以用于是指用户的头部的实际物理取向和相同的改变,或者可以用于是指由计算机程序或计算机输入设备确定的用户观看的虚拟方向。

[0029] 图2a示出了用于立体观看,即,用于3D视频和3D音频数字捕捉和重放的系统和装置。该系统的任务是从特定位置捕捉足够的视觉信息和听觉信息,使得能够通过物理上位于不同位置并且可选地在未来的某个时间的一个或多个观看者来实现在该位置的体验或存在的令人信服的再现。这样的再现需要比由单个相机或麦克风捕捉的更多的信息,以便观看者可以使用他们的眼睛和他们的耳朵来确定场景内的对象的距离和位置。如在图1a和图1d的上下文中所解释的,为了创建一对具有双眼像差异的图像,使用两个相机源。以类似的方法,为了能够感测声音方向的人类听觉系统,使用至少两个麦克风(公知的立体声通过记录两个音频通道来产生)。人类听觉系统可以检测例如在音频信号的定时差异中的线索以检测声音的方向。

[0030] 图2a的系统可以由三个主要部分组成:图像源、服务器和渲染设备。视频捕捉设备SRC1包括具有重叠视场的多个(例如,8个)相机CAM1,CAM2,⋯,CAMN,使得从至少两个相机捕捉视频捕捉设备周围的观看区域。设备SRC1可以包括多个麦克风,以捕捉源自不同方向

的音频的定时差异和相位差异。该装置可以包括高分辨率取向传感器,使得可以检测并且记录多个相机的取向(观看方向)。设备SRC1包括或在功能上连接至计算机处理器PROC1和存储器MEM1,该存储器包括用于控制捕捉设备的计算机程序PROGR1代码。由设备捕捉的图像流可以存储在存储器设备MEM2上用于另一设备(例如,观看者),和/或使用通信接口COMM1被发送到服务器。

[0031] 需要理解的是,尽管这里描述了8相机立方设置作为系统的一部分,但是另一相机设备可以用来代替系统的一部分。

[0032] 可替代地或除了产生图像流的视频捕捉设备SRC1(或者多个该设备)之外,一个或多个合成图像源SRC2可以存在于系统中。这种合成图像源可以使用虚拟世界的计算机模型来计算其传送的各种图像流。例如,源SRC2可以计算与位于虚拟观看位置处的N个虚拟相机相对应的N个视频流。如之前对于图1d所解释的,当这样合成视频流集合用于观看时,观看者可以看到三维虚拟世界。设备SRC2包括或者在功能上连接至计算机处理器PROC2和存储器MEM2,该存储器包括用于控制合成源设备SRC2的计算机程序PROGR2代码。由设备捕捉的图像流可以存储在存储器设备MEM5(例如,存储卡CARD1)上用于另一设备(例如,观看者),或使用通信接口COMM2被传送到服务器或观看者。

[0033] 除了捕捉设备SRC1之外,还可以存在存储、处理和数据流服务网络。例如,可以存在存储来自捕捉设备SRC1或计算设备SRC2的输出的服务器SERV或多个服务器。该设备包括或者在功能上连接至计算机处理器PROC3和存储器MEM3,该存储器包括用于控制服务器的计算机程序PROGR3代码。服务器可以通过有线网络连接或无线网络连接或二者连接至源SRC1和/或SRC2以及通过通信接口COMM3连接至观看者设备VIEWER1和VIEWER2。

[0034] 为了观看所捕捉或创建的视频内容,可能存在一个或多个观看器设备VIEWER1和VIEWER2。这些设备可以具有渲染模块和显示模块,或者这些功能性可以组合在单个设备中。这些设备可以包括或者在功能上连接至计算机处理器PROC4和存储器MEM4,该存储器包括用于控制观看设备的计算机程序PROGR4代码。观看者(重放)设备可以由用于从服务器接收视频数据流并且用于解码视频数据流的数据流接收器组成。数据流可以通过通信接口COMM4通过网络连接,或者从如存储卡CARD2之类的存储设备MEM6接收。如参照图1c和图1d所描述的,观看者设备可以具有用于将数据处理为用于观看的合适格式的图形处理单元。观看者VIEWER1包括用于观看所渲染的立体视频序列的高分辨率立体图像头戴式显示器。头戴式设备可以具有取向传感器DET1和立体声音频耳机。观看者VIEWER2包括赋予了3D技术的显示器(用于显示立体视频),并且渲染设备可以具有与其连接的头部取向检测器DET2。任何设备(SRC1、SRC2、服务器、渲染器、VIEWER1、VIEWER2)可以是计算机或便携式计算设备,或者连接至这些设备。这样的渲染设备可以具有用于进行根据本文中描述的各种示例的方法的计算机程序代码。

[0035] 图2b示出了用于立体观看的相机设备。相机包括三个或更多个相机,其被配置成用于创建左眼图像和右眼图像的相机对,或者可以被布置成这样的对。相机之间的距离可以与人眼之间的通常距离相对应。相机可以被布置成使得它们在它们的视野中具有显著的重叠。例如,可以使用180度或以上的广角镜头,并且可以存在3个、4个、5个、6个、7个、8个、9个、10个、12个、16个或20个相机。相机可以在整个观看球体内规则地或不规则地间隔开,或者它们可以仅覆盖整个球体的一部分。例如,可以存在三个相机,其被布置成三角形并且具

有朝向三角形的一侧的不同观看方向,使得所有三个相机覆盖观看方向中间的重叠区域。作为另一示例,8个相机具有广角镜头并且规则地被布置在虚拟立方体的角部处并且覆盖整个球体,使得整个或基本上整个球体在所有方向上被至少3个或4个相机覆盖。在图2b中,示出了三个立体相机对。

[0036] 可以使用具有其他类型相机布局的相机设备。例如,可以使用一个半球中具有所有相机的相机设备。相机的数目可以是例如3个、4个、6个、8个、12个或更多个。相机可以被放置成创建其中立体图像可以由两个或更多个相机的图像数据形成的中央视场,以及其中一个相机覆盖场景并且仅可以形成正常的非立体图像的外围(偏远)视场。在本说明书中稍后描述可以在系统中使用的不同相机设备的示例。

[0037] 图2c示出了用于立体观看的头戴式显示器。头戴式显示器包含两个屏幕段或两个屏幕DISP1和DISP2,用于显示左眼图像和右眼图像。显示器靠近眼睛,因此镜头用于使图像容易观看并且用于将图像扩展到尽可能多地覆盖眼睛的视场。该设备附接到用户的头部,使得即使用户转动他的头部,该设备仍然保持其位置。该设备可以具有用于确定头部移动和头部方向的取向检测模块ORDET1。这里,要指出的是,在这种类型的设备中,可以跟踪头部移动,但是由于显示器覆盖视场的大面积,所以不需要检测眼睛的移动。头部取向可能与用户头部的真实物理取向有关,并且可以由传感器跟踪用于确定用户头部的真实取向。可替代地或另外,头部取向可以与用户的观看方向的虚拟取向有关,通过计算机程序控制或者通过诸如操纵杆之类的计算机输入设备控制。也就是说,用户可能能够用输入设备改变所确定的头部取向,或者计算机程序可以改变观看方向(例如,在游戏中,游戏程序可以控制所确定的头部取向,代替真实的头部取向或者附加真实的头部取向)。

[0038] 图2d图示了相机CAM1。相机具有相机检测器CAMDET1,其包括用于感测撞击传感器元件的光的强度的多个传感器元件。相机具有镜头OBJ1(或多个镜头的镜头布置),镜头被定位成使得撞击传感器元件的光行进通过镜头到达传感器元件。相机检测器CAMDET1具有标称中心点CP1,其是多个传感器元件的中点,例如,对于矩形传感器,是对角线的交叉点。镜头也具有标称中心点PP1,其例如位于镜头的对称轴上。相机的取向方向由通过相机传感器的中心点CP1和镜头的中心点PP1的线定义。相机的方向是沿着该线从相机传感器到镜头的方向的向量。相机的光轴被理解为这条线CP1-PP1。

[0039] 上文所描述的功能如下。首先用捕捉设备记录时间同步的视频、音频和取向数据。这可以由如上文所描述的多个并发视频流和音频流组成。然后,这些视频流和音频流立即或稍后传送到存储和处理网络以进行处理并且转换成适合于随后发送到回放设备的格式。该转换可以包含音频数据和视频数据的后处理步骤,以便提高质量和/或减少数据量,同时将质量保持在期望水平。最后,每个回放设备从网络接收数据流,并且将其渲染为用户用头戴式显示器和耳机可以体验的原始位置的立体观看再现。

[0040] 通过如下文所描述的创建用于观看的立体图像的新颖方式,用户可能能够在多个方向上转动他们的头部,并且回放设备能够创建与该特定取向相对应的场景的高频(例如,每秒60帧)立体视频和音频视图,例如,如从原始记录的位置可能出现的一样。也可以使用由相机数据创建用于观看的立体图像的其他方法。

[0041] 图3a图示了形成渲染层并且形成用于存储或传输的图像数据。记录用于存储到文件中或通过创建多个像素集合(即,渲染层)进行传输的场景,其中层中的每个数据点至少

包括来自公共原点和颜色数据的向量。可以使用已知的2D图像或视频序列压缩技术来压缩每个数据集。

[0042] 可以形成图3a中的若干个点 $P_1, \dots, P_N$ 和 $PX_1, PX_2$ ,每个点具有空间中的颜色和位置。点 $PX_1$ 和 $PX_2$ 被隐藏在像素 $P_1, P_2$ 和 $P_3$ 后面。然后,这些点被转换成渲染层,使得当视点VIEWPNT观看时,第一渲染层RENDER LAYER 1由直接可见点创建,并且一个或多个渲染层RENDER LAYER 2至少部分地由隐藏在第一渲染层后面的点创建。每个点的位置向量可以以不同的方式存储或压缩。它可以简单地被表达为每个点的三个独立参数,即,一对角度和距参考向量(由视点和观看方向定义的向量)的距离、或正交轴方向上的3个距离。可替代地,参数化映射函数可以用于基于点的索引来将来自原点的空间中的每个点的位置向量更紧凑地编码成点序列,其被解释为具有已知整数宽度和高度的二维常规布局(图像),包括渲染层像素 $RP_1, RP_2, RP_3$ 和 $RPX_1, RPX_2$ 。这与图3a中的渲染层RENDER LAYER 1和RENDER LAYER 2相对应。这可以例如将 $x$ 和 $y$ 坐标直接映射到偏航坐标和俯仰坐标,允许将整个球体编码为矩形结构。每个(偏航、俯仰)像素的像素颜色值可以通过从现有点值的内插形成。可替代地,可以使用圆形映射函数,诸如等立体角映射(equisolid mapping)(半径 $=2 * \text{焦距} * \sin(\text{与光轴夹角}/2)$ ),以将半球或更多球体映射到圆形图像。

[0043] 可替代地,圆形映射函数可以用于将球面坐标映射为2d笛卡尔坐标。这些映射函数创建产生圆形图像,其中每个 $x$ 和 $y$ 值对可以映射回到球面坐标。函数将与光轴的夹角(theta)映射为点与图像圆心( $r$ )的距离。对于每个点,围绕光轴的角度(phi)在球面坐标和映射图像圆中保持相同。 $x$ 和 $y$ 坐标与映射图像圆中的 $r$ 和 $\phi$ 之间的关系如下: $x = x_0 + r * \cos(\phi)$ ,  $y = y_0 + r * \sin(\phi)$ ,其中点 $(x_0, y_0)$ 是图像圆的中心。

[0044] 这种映射函数的示例是在鱼镜头中通常使用的等立体角。等立体角映射取决于镜头的焦距( $f$ ),并且如下所示: $r = 2 * f * \sin(\text{theta}/2)$ 。因此,对于位于光轴中心(theta为0)的点, $r$ 变为零,因此映射点也在图像圆的中心。对于在垂直于光轴的向量上的点(theta为90度), $r$ 变为 $1.41 * f$ ,并且图像圆中的点可以如下计算: $x = x_0 + 1.41 * f * \cos(\phi)$ ,  $y = y_0 + 1.41 * f * \sin(\phi)$ 。可以使用常数乘数对 $x$ 和 $y$ 进行缩放,以将坐标转换为目标分辨率中的像素。其他映射函数可以是球面投影( $r = 2 * f * \tan(\text{theta}/2)$ )、等距投影( $r = f * \text{theta}$ )和正交投影( $r = f * \sin(\text{theta})$ )。

[0045] 每层可以完全(即,没有漏洞,以连续方式)覆盖相机周围的空间(诸如图3a中的渲染层RENDER LAYER 1)。或者它可能是稀疏覆盖空间(诸如图3a中的渲染层RENDER LAYER 2),其中未覆盖的部分完全不使用映射参数,或被编码为较大尺寸的可被高度压缩的零值。可以被可视化的所有对象被记录在层中的一个层中。每层都提供了所需的映射参数,用于将层的二维图像数据映射到渲染空间中。所有层可以最终被打包成与必要的映射元数据一起提供的单个数据结构以对它们进行解码。可替代地,不同的层可以以不同的文件或流或不同的数据结构提供。

[0046] 另外,层的编码可以允许对渲染复杂度的缩放,或减少所递送的数据量,同时仍然给出场景的良好再现。实现这样的途径是将所有层打包成2D图像,其中随着子层距离增加,沿着轴(例如,沿着增加的 $y$ 轴方向(向下))更远地定位。当需要较少的渲染时,较低的数据根本不被递送或不被解码/处理,而渲染只通过顶层和可能的子层的有限子集来进行。

[0047] 本发明可以允许除了具有渲染完全合成场景的大数据处理能力之外还具有先前

不可能的物理上现实行为水平的复杂3D环境的记录、分发和再现。这可以通过由于使用渲染层结构而大大减少需要为特定图像分辨率递送的数据量,来改善基于来自不同视点的多个图像的早期再现技术。

[0048] 在图3b中,图示了使用两个相机CAMR和CAML形成两个渲染层RENDER LAYER 1和RENDER LAYER 2。因为对象REAROBJ被隐藏在另一对象FRONTOBJ的后面,所以不同相机“看到”对象REAROBJ的不同部分。左侧相机CAML能够从左侧捕捉对象REAROBJ的更多图像信息,并且右侧相机CAMR从右侧捕捉对象REAROBJ的更多图像信息。当渲染层例如通过将视点VIEWPNT保持为视点来创建时,FRONTOBJ对象隐藏具有图像信息的对象REAROBJ的部分,以及没有图像信息的部分。从而,第一渲染层RENDER LAYER 1包括表示第一对象FRONTOBJ的像素AREA1和表示第二对象REAROBJ的可见部分的像素AREA2。第二渲染层包括与第二对象REAROBJ的隐藏部分的图像信息相对应的像素AREA3。AREA3外的像素可能为空,或是虚拟像素。如之前所解释的,渲染层的深度信息可以被创建。

[0049] 图4图示了使用渲染层渲染图像。为了渲染立体图像或立体视频序列,如之前所解释的,形成左眼和右眼的图像帧。为了渲染图像帧,所有层RENDER LAYER 1、RENDER LAYER 2的内容投影到一个新的渲染相机空间中,并且按深度进行排序以渲染正确场景。例如,通过传统的图形处理单元,每个渲染层点RP1,RP2,⋯,RPN和RPX1,RPX2,⋯可以被视为“粒子”,并且使用顶点着色器程序进行变换,而且被转换为3D渲染空间中具有单个像素的“点子画面”,其包括相对于渲染视点的深度值。将重叠投影粒子的深度值与正确的混合函数以正确次序进行比较和绘制。这由与点RP1、RP2、RP3、RPX1、RPX2相对应的虚线矩形来图示。以这种方式,可以使像素位于与它们在真实空间中的相应的源图像点位置相对应的位置处。不透明内容被渲染成使得与渲染相机最近的点被示出。非不透明内容可以用正确混合在其后面可见的内容来渲染。

[0050] 这里需要注意的是,渲染层的像素在渲染空间中可以表示对象的不同大小。与更接近视点的像素相比,远离视点(具有大深度值)的像素可以表示更大的对象。这是因为渲染层像素可以原始地表示某个空间“锥”以及该“锥”中的图像内容。取决于锥的底部有多远,像素表示空间中的点的不同大小。渲染层可以被对准,用于以如下方式被渲染:当从渲染视点观看时,像素网格基本上在彼此的顶部上对齐。

[0051] 为了将渲染层转换为渲染空间,它们可能需要被转动。围绕x轴的坐标的转动变换 $R_x$ 的角度 $\gamma$ (也被称为俯仰角)的示例通过转动矩阵来定义:

$$[0052] \quad R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}.$$

[0053] 以类似的方式,可以形成围绕不同轴的转动 $R_y$ (用于偏航)和 $R_z$ (用于滚转)。作为一般转动,可以基于 $R=R_x R_y R_z$ 形成三个转动的矩阵乘法。然后,该转动矩阵可以用于根据 $v_2=R v_1$ 乘以第一坐标系中的任何向量,以获得目标坐标系中的向量。

[0054] 作为转动的示例,当用户转动头部(存在由俯仰值、偏转值和滚转值表示的转动)时,可以确定用户的头部取向以获得新的头部取向。这可能发生,例如,使得在头戴式显示器中存在头部运动检测器。当确定了新头部取向时,可以重新计算观看取向和虚拟眼睛的位置,使得所渲染的图像与新头部取向匹配。

[0055] 作为另一示例,对头戴式相机取向的校正进行了解释。这里使用的技术是记录捕

捉设备的取向并且使用取向信息来校正呈现给用户的观看方向,有效地消除了重放期间捕捉设备的转动,使得用户控制观看方向,而非捕捉设备。如果观看者反而想要体验捕捉设备的原始运动,则校正可以被禁用。如果观看者希望体验原始运动的不那么极端的版本,则可以使用过滤器动态地应用校正,使得跟随原始运动,但是速度较为缓慢或与正常方向的偏差较小。

[0056] 对于要显示的帧,层可以在多个渲染遍次 (pass) 中被渲染,从不透明层开始,并且在包含半透明区域的层结束。最后,如果需要,则可以进行单独的后处理渲染遍次来插值空像素的值。

[0057] 在渲染期间,使得图形处理 (诸如OpenGL) 深度测试能够丢弃所遮蔽的片段,并且使能深度缓冲器以进行写入。如果渲染层包含半透明区域,则在渲染期间启用Alpha混合,否则禁用它。场景几何包含大量未连接的顶点 (GL\_POINT), 其各自与所存储的渲染层数据中的一个像素相对应。根据层存储格式,顶点可以具有不同数目个属性。顶点属性是例如指向实际层图像数据的位置 (x,y,z)、颜色或纹理坐标。

[0058] 接下来,作为示例,解释OpenGL顶点和片段处理。其他渲染技术还可以以类似的方式使用。

[0059] 对于不同的层存储格式,顶点和片段处理可能略有不同。处理以未压缩列表格式存储的层的步骤可以如下 (每个顶点):

[0060] 1. 最初,所有顶点都被分配并且传递到顶点处理阶段,其属性包括相对于共同原点 (渲染视点) 的观看角度、颜色和深度。如果经处理的层具有半透明内容,则顶点必须根据其深度值进行排序。

[0061] 2. 顶点的 (偏航、俯仰、深度) 表示被转换为3D笛卡尔向量 (x,y,z)。

[0062] 3. 相机和世界变换通过将其与对应矩阵相乘而应用于顶点。

[0063] 4. 顶点颜色属性被传递到片段处理阶段。

[0064] 5. 最终顶点坐标被写入输出变量 (gl\_Position)。

[0065] 6. 在片段处理阶段,从顶点处理接收到的颜色数据被直接写入输出变量 (gl\_FragColor) 中。

[0066] 处理以压缩图像格式存储的层,即,包括具有像素颜色数据和深度值的像素的渲染层的步骤可以如下 (每个顶点):

[0067] 1. 最初,所有顶点均匀分部在具有相同深度值的场景周围。

[0068] 2. 如果顶点不在观看者的当前视场内部,则应用变换函数以便将其置于当前视场内部。该变换的目的是将所有可用的顶点初始集中到当前可见的区域。否则,由该顶点表示的像素数据可能在片段处理阶段的渲染期间被剪裁掉。在这种情况下避免剪裁可以提高渲染质量。位置变换可以以将视场外的顶点均匀分布在视场内部的方式完成。例如,如果视场在水平上从0度到90度,则最初水平位于91度方向的顶点然后可能被变换成1度的水平位置。类似地,从91度到180度的水平位置的顶点可能被水平地变换为1度到90度的范围。垂直位置可以以相同的方式计算。为了避免经变换的顶点进入与已经在视场内部的其他顶点精确相同的位置,小常数部分 (例如,在该示例情况下为0.25个像素) 可以被添加到顶点新位置值。

[0069] 3. 顶点颜色数据的纹理坐标是根据所转换的顶点位置计算的,并且被传递到片段

处理阶段。

[0070] 4. 顶点的深度值使用来自纹理中的纹理查找来为顶点获取。

[0071] 5. 顶点的观看角度使用映射函数来计算。

[0072] 6. 顶点的(偏航、俯仰、深度)深度表示被转换为笛卡尔三维向量(x,y,z)。

[0073] 7. 相机和世界变换通过将其与对应的矩阵相乘而应用于顶点。

[0074] 8. 像素分辨率在最终顶点位置中引起小舍入误差,这可以通过计算(子像素)舍入误差并且将其传递到片段处理阶段来考虑。

[0075] 9. 最终顶点坐标被写入着色器输出变量(gl\_Position)。

[0076] 10. 在片段处理阶段,使用所接收到的纹理坐标并且考虑到子像素舍入误差值而从颜色纹理中取回颜色数据,以便使用周围点来插值更合适的颜色值(这在未压缩列表格式中是不可能的)。然后,颜色值被写入输出变量(gl\_FragColor)。

[0077] 源像素可以在渲染期间以以下方式对准:以来自第一渲染层的第一像素和来自第二渲染层的第二像素通过以子像素量调整其在空间中的位置来在彼此顶部上配准。取决于渲染层的存储格式,顶点(像素)可以首先与一种虚拟网格对准(步骤1和2,以“压缩的”图像格式),或不与其对准。最后,顶点可能在获取正确深度并且变换和映射坐标之后应用相机和世界变换的步骤中进行对准/定位(步骤7)。需要理解的是,对准也可能发生在另一阶段,或者作为其自身的单独步骤。

[0078] 图5a描绘了包括用于渲染图像的渲染层的数据结构。在非压缩的列表型格式中,各种场景点由点数据结构表示,其各自具有颜色值(3个值,例如,红色、绿色、蓝色)、透明度(例如,Alpha通道)、位置(3个值,例如,偏航坐标、俯仰坐标、深度坐标)和可能的其他属性。

[0079] 在图5b中,在图像数据格式中,第一渲染层中的场景点的颜色值由一个编码图像表示,该图像包括场景点的颜色值作为渲染层像素RP1,RP2,RP3或该图像包含可以用于例如通过纹理映射计算场景点的颜色值的颜色值。以类似的方式,第一渲染层的其他属性可以被表示为图像,例如,包括渲染层像素的深度值RPD1,RPD2,RPD3的深度值图像。第二渲染层中的场景点的颜色值由一个编码图像表示,该图像包括场景点的颜色值作为渲染层像素RPX1,RPX2或该图像包含可以用于例如通过纹理映射计算场景点的颜色值的颜色值。深度值RPDX1,RPDX2位于对应的深度图像中。

[0080] 不同的渲染层可以具有它们自己的图像数据结构,或者渲染层可以被一起被组合为一个或多个图像。例如,图像可以具有用于第一渲染层数据的数据段、第二渲染层数据的另一数据段等等。图像可以使用常规图像压缩技术来压缩。

[0081] 另外,层的编码可以允许渲染复杂度的缩放或减少所递送的数据量,同时仍然给出场景的良好再现。实现这样的途径是将所有层包装成2D图像,其中随着子层距离增加,沿着轴(例如,沿着增加的y轴方向(向下))方向被更远地定位。当需要较少的渲染时,较低的数据根本不被传递或不被解码/处理,并且渲染仅通过顶层和子层的可能的有限子集来进行。

[0082] 简单的静态或动态合成环境可以使用常见的3D图形处理硬件和技术来实时渲染,以便产生交互式三维场景。还有,几何复杂的动态3D场景或模型可以使用已知的“2D加深度”技术来记录或生成并且利用自由观看方向来实时观看。然而,这里已经注意到,这些技术在反射或半透明表面和对象的现实再现中可能是不可行的。即,已知技术依赖于存储若

干个像素的颜色值和三维位置,以便能够以对那些像素的自由观看角度再现视图。这使得可以在三维空间中四处移动虚拟相机并且在几何上准确地渲染像素。这里已经注意到,以这种方式再现现实环境的问题是,当观看角度改变时,如果所存储的像素是反射或半透明的,则所存储的颜色值不会反映现实,其中真实世界中的表面中的很多表面或多或少是反射的或半透明的。在真实世界中,当观察反射或半透明的表面并且移动或转动头部时,表面的颜色根据观看角度以及对象后面的物品或反射来自的地方而改变。这里已经注意到,存储每个在空间中的像素的单个颜色值并且渲染那些颜色而不考虑观看方向,不能精确地再现颜色。

[0083] 图6a示出了具有部分反射的对象的场景。在这种环境下,提出了一种用于紧凑编码和有效地再现高复杂度的3D场景的新技术。这可以通过使用多个2D层(如之前所描述的)和例如层上的像素或片段的透明度和反射率的描述符来实现。

[0084] 如之前所描述的,像图6a所示的场景可以通过创建多个集合或层来记录,其中层中的每个数据点包括来自公共原点的向量和颜色数据。除了向量和颜色之外,数据点(像素或片段)还可以包括主层树透明度、反射层树透明度和反射层子树标识符中的任一个。这些数据集合(颜色、透明度、反射率、子树标识符)中的每个数据集合可以使用公知的2D图像或视频序列压缩技术来独立地压缩。

[0085] 在图6a中,示出了若干个图形对象XP1,XP2,PP3,PP4,PP5,OBJ1和OBJ2。这些图形对象可以与真实世界对象或合成场景的对象直接相对应。如之前所描述的,图形对象可以通过对真实世界对象进行建模来形成。图形对象还可以与渲染层相对应。图形对象还可以存在于一个或多个渲染层上,以便于在观看方向上在彼此后面的对象的真实渲染。以这种方式,渲染层可以被认为包括图6a中的图形对象。又进一步地,图形对象可以是单个像素或像素组。例如,图形对象XP1和XP2可以是图6a的窗户和门中的单个像素或像素组。如所描述的,图形对象可以是渲染层或其部分、图形对象的模型或其部分、和/或像素或像素组。在下文中,为了简单起见,图形对象已经被描述为与各个真实世界的图形对象相对应,并且需要理解的是,还可以使用任何其他形式的图形对象。

[0086] 在图6a中,在VIEW\_DIR的观看方向上,存在两个反射对象XP1和XP2以及一个半透明对象XP1。由于反射率和观看方向,对象OBJ1被反射离开表面XP1(玻璃窗),而对象OBJ2被反射离开表面XP2(金属门)。因此,这些图形对象具有镜面,即,窗户和门是平面对象,并且与这样的表面相一致的平面可以被理解为这些对象的镜面。另外,尽管未示出,但是房屋内部的对象可以通过窗户可见。窗户也可以具有颜色,例如,绿色色调。换句话说,在如这里描述的渲染方法中,可以存在像窗户XP1那样的第一图形对象(其片段中的一个或多个片段)的信息和第二图形对象(被反射离开窗户的树OBJ1的片段)的信息,使得组成对象中的一个对象(例如,窗户)的图形片段是其他图形对象的片段通过其可见的特殊片段(树的图像和房屋内部的任何对象)。

[0087] 对于渲染,来自所有层的内容投影到一个新的渲染相机空间中,并且按深度进行排序以渲染正确的场景。例如,通过常规图形处理单元,每个点可以被视为“粒子”(一种片段的类型),并且使用顶点着色器程序进行变换,并且被变换为3D空间中的具有单个像素的“点子画面”,其包括相对于渲染相机的深度值。将重叠投影粒子的深度值与正确的混合函数以正确次序进行比较和绘制。不透明内容被渲染成使得仅最接近渲染相机的点被示出。

这对于所有不透明或反射的对象都是足够的。

[0088] 对于渲染层,以及一般而言的任何图形对象,映射信息可以用于图形对象放入渲染空间中。也就是说,存在关于渲染层的X和Y坐标如何被变换为三维渲染空间而给出的信息(参见图4)。例如,如在图4的上下文中所描述的,可以使用不同的映射函数。还有,剪裁信息可以用作映射信息的一部分,使得在映射的范围上给出信息。以这种方式,例如,整个球体可能不需要被编码,因为图形对象可以通过剪裁来限制,并且在剪裁区域之外不需要信息。每个时刻都可以使用新层和映射参数集合进行编码,以允许3D环境改变的基于时间的重放。在每个帧中,新层数据和映射信息被投入使用。可替代地,基于时间的重放可以被暂停,并且单个帧可以被使用并且从不同位置进行渲染。

[0089] 图6b示出了用于渲染反射对象和半透明对象的依从性图形。需要利用混合通过它可见的内容来渲染半透明内容。对于简单的半透明对象(诸如有色窗户),依从性图形的主要水平的透明度值用于混合。通过透明对象可见的任何对象都在透明对象的透明度子树中。反射层树透明度值可以用于与反射子树混合,而非(仅用于渲染反射,例如,非透明镜子)或者除了(反射加透明度,例如,窗户)层的主树之外。反射率和/或半透明度的信息可以与每个像素的颜色信息一起例如存储在渲染层数据结构中。可替代地或另外,影响依从对象的反射率和/或半透明度的信息可以存储在图形的数据结构中,例如,使得图形中的两个对象之间的链路包含反射率信息和半透明信息。例如,反射率和半透明度(透明度)可以使用用于透明度的第一alpha值和反射率的第二alpha值来给出。

[0090] 在图6b中,场景以依从性图形数据结构表示。依从性图形可以例如是树结构。场景的常规图形对象PP3,PP4,PP5,OBJ1,OBJ2具有直接取决于场景根节点的节点。如之前已经描述的,这样的对象也可以驻留在渲染层的树结构中,用于在基于层的渲染方法中进行渲染。例如,图形对象可以存在于或可以驻留在渲染层上,并且渲染层可以在树结构中彼此依赖。也就是说,依从性图形可以包括渲染层(作为图形的节点)和渲染层之间的依从性链路。透明和/或反射对象XP1,XP2的节点也取决于主场景根节点。仅通过这些透明和/或反射对象可见的对象XOBJ1,XOBJ2的节点取决于通过该对象它们可见的节点。对于通过透明对象可见的常规对象,诸如图6a所示房屋内部的对象,对象可以具有直接对应的节点,其是透明对象的子节点。被反射离开反射表面的对象可以具有镜面对象XOBJ1,XOBJ2(参见图6c的镜面对象),其具有取决于反射对象的节点。在图6b中,对象OBJ1被反射离开窗户XP1,因此与常规对象OBJ1相对应的镜面对象XOBJ1具有取决于窗户XP1的节点的节点。同样地,对象OBJ2被反射离开门XP2,因此与常规对象OBJ2相对应的镜面对象XOBJ2具有取决于门XP2的节点的节点。

[0091] 在图6b中,需要指出的是,如果观看角度适合于反射,则对象OBJ1也可以被反射离开对象XP2,而对象OBJ2可以被反射离开对象XP1。因此,XOBJ1也可能是XP2的子节点,而XOBJ2可能是XP1的子节点。换句话说,依从性树可以是网络,更具体地说是\*\*有向网络\*\*。可替代地,有向网络可以扩展到例如图6b中的树结构,使得XOBJ1和XOBJ2取决于XP1,XOBJ1和XOBJ2的副本取决于XP2。在图6b中,依从性图形是依从性方向从上到下的有向网络结构。该图形也可以是有向(无环路)网络,其允许每个反射和/或透明对象确定通过它们可见的对象。当对象被渲染时,渲染可能发生,使得首先,第二图形对象(子节点)通过使用第二图形对象(例如,颜色)的信息以及第一图形对象(颜色、透明度、反射率)的信息创建对应的渲染

像素的像素数据来渲染。然后,例如在另一渲染遍次中,第一(母节点)图形对象首先通过使用诸如颜色信息之类的第一图形对象的信息创建渲染像素的像素数据来渲染。在这种布置中,第二对象的像素形成第一对象的像素的子集(该上下文中的像素子集意味着第一对象的像素中的没有一个像素、一些像素或全部像素)。与第二对象相对应的像素不会被渲染在第一对象之外,除非与第一对象相比,第二对象通过另一对象也可见。换句话说,对象的可见性通过依从性信息限制在其他对象之外,使得一些对象仅通过其他对象可见,而不在其他对象之外。换句话说,依从性图形可以是依从性树,并且第一图形对象和第二图形对象与依从性树的第一节点和第二节点分别相对应,使得第二节点是第一节点的子节点,并且依从性树确定第二对象仅通过第一对象而非其他对象可见。

[0092] 图6c图示了渲染反射对象。如之前所解释的,对象(例如,渲染层)的数据可以使用潜在的大量处理由复杂的合成模型(诸如CGI电影内容模型)直接生成,该处理将世界从单个视点渲染为合适格式以进行渲染,例如,之前所描述的被存储在辅助层和层树中的具有任意数目个遮蔽像素的层格式。可替代地,场景数据可以使用不同位置处的多个相机从真实场景捕捉,其中相机对用于创建在两个图像中匹配的每个点的深度估计。点估计被映射到公共原点和取向,并且复制通过比较它们的颜色值和位置值来移除的条目。以这种方式,渲染层可以被获得,并且这样的渲染层可以在渲染中用作图形对象。

[0093] 在图6c中,图示了用于树对象OBJ1和盒子对象OBJ2的用于渲染的图像对象的形成。树对象OBJ1被反射离开窗户对象XP1,因此树对象OBJ1的图像对象XOBJ1通过相对于窗户对象的镜面来对树对象OBJ1进行镜像而形成。以这种方式,镜面对象XOBJ1被放置在房屋的“内部”,并且仅通过反射窗户XP1的区域可见。类似地,盒子的镜面对象XOBJ2通过使用门XP2的平面作为镜面来形成。当渲染时,这些镜面对象XOBJ1和XOBJ2分别与其他对象XP1和XP2的依从性被采用以通过使用依从性图形和父对象的透明度/反射率信息来渲染镜面对象。在渲染中,根据前景对象的透明度/反射率,前景对象的颜色信息与后面对象的颜色信息进行混合。

[0094] 来自真实世界图像的透明图像和反射对象的数据可以通过形成每个前景对象以及通过它可见的反射层树对象的表面颜色和半透明度(alpha)的估计来进行。这可以通过比较来自多个视点的数来区分不同项目的距离以及它们的颜色来实现。例如,在图6c中,镜像对象XOBJ1可以通过比较两个或更多个图像并且将树OBJ1的反射与窗户隔离来直接形成。

[0095] 接下来,描述用于渲染根据上述的数字图像数据的计算机化方法。在该方法中,第一图形对象和第二图形对象的信息被接收以用于渲染第二图形对象。并且,第一图形对象和第二图形对象之间的依从性信息被接收,其中依从性信息包括用于根据第一图形对象和第二图形对象在观看方向上的重叠来渲染第二图形对象的信息。然后,第二图形对象通过使用第二图形对象的信息和第一图形对象的信息来创建像素数据来渲染,其中对在观看方向上与第一图形对象重叠的第二图形对象的这部分执行渲染。换句话说,在该方法中,第二图形对象(例如,镜像图形对象)的可见性依从性可以用于渲染第一图形对象的边界内的第二图形对象。第二图形对象的渲染可以包括:使用第一图形对象的反射信息来确定用于通过第一图形对象渲染第二图形对象的第一图形对象的透明度。也就是说,当镜像对象用于渲染反射时,用于反射第二对象的图像的对象的反射率用于确定第一对象的透明度。

第二图形对象可以是第三对象的镜像,并且第二图形对象的信息可以通过相对于第一图形的镜面来对第三对象(从表面反射的真实对象)进行镜像而形成。镜面可以是接近第一对象表面的任何平面。

[0096] 在上文中,第一图形对象可以例如是第一渲染层,而第二图形对象(通过第一图形对象可见)可以是第二渲染层。可替代地,第一图形对象可以是像素或像素组(例如,第一渲染层)。第二图形对象可以是像素或像素组(例如,第二渲染层)。以这种方式,渲染过程可以包括:确定通过第一渲染层的像素的第二渲染层的哪个部分可见。如果需要,则可以自然地第一渲染层的许多像素进行重复渲染。例如,第一图形对象可以包括第一像素,而第二图形对象可以包括渲染层,并且可以确定第一像素是否是通过其第二图形对象可见的门户像素,然后第一像素可以在确定指示所述第一像素是门户像素的情况下,使用第一图形对象的第一信息和第二图形对象的第二信息来渲染。门户对象或门户像素是这样的图形对象,其包括第二对象通过其可见的特殊区域,使得第二对象在门户对象之外不可见(除非存在其他门户对象使第二对象通过其门户可见)。

[0097] 由于反射的存在,所描述的方法和设备可以便于具有反射和半透明表面的复杂3D环境的更真实的记录、分布和再现。对于3D观看,反射被生成使得它们根据视差出现在左眼图像和右眼图像的不同位置中,并且这将真实感增加到3D图像。

[0098] 图6d图示了使用模版缓冲器来渲染反射对象。每个像素的用于渲染反射对象和透明对象的子树标识符和反射率透明度值可以被存储,然后使用与模版操作组合的附加遍次来渲染,以管理视图中子树的可见性。这可以确保镜子内的内容仅当通过镜子观察时而非在它们后面观察时可见,如在真实的物理环境中一样。

[0099] 通过两个遍次在图6d中通过示出了模版缓冲器(左列)和画面缓冲器(右列)的示例性内容作为渲染进展来说明该过程。最初,模版缓冲器和画面缓冲器为空(无像素数据或零)。要渲染的像素可以通过确定它们所属的子树标识符来标识,该子树标识符取决于图6b所示的依从性信息。例如,在图6d中图6a的常规对象中的“墙壁”像素被示出为“窗户”像素XP1和“门”像素XP2之外的像素PP5,该“墙壁”像素是非反射的。

[0100] 还可以渲染第一遍次中的所有像素,然后标记要重新渲染的模版缓冲器中的像素。这可以确保画面中没有漏洞出现。落在反射/透明对象内的像素用它们相应的子树标识符(例如,“2”和“3”)标记。确定哪些像素落在哪个对象内可以作为常规图形处理器操作来进行。

[0101] 在第一遍次中,落在反射/透明对象XP1和XP2内的像素首先被标记到模版缓冲器。然后,对象XOBJ1的像素(“树”的镜像与“窗户”XP1重叠)和XOBJ2(“盒子”的镜像与“门”XP2重叠)的像素被渲染。如果存在依赖于对象XOBJ1和XOBJ2的对象,则依从对象被首先渲染。在渲染中使用模版缓冲器,使得只有标记为在模版缓冲器中渲染的画面缓冲器的那些像素被渲染(计算并且存储在画面缓冲器中)其,在这种情况下,像素与XP1和XP2重叠。因此,只有落在对象XP1和XP2内部的像素在该遍次中被渲染,并且因而在观看方向上落在对象XP1和XP2之外的对象XOBJ1和XOBJ2的任何部分不被渲染。也就是说,例如,树OBJ1仅从窗户而非墙壁反射,并且除了通过窗户XP1之外,树OBJ1在其他任何地方不能被看到。换句话说,在用于渲染的方法中,要渲染的像素可以使用模版缓冲器来标记,并且第二图形对象可以通过创建与标记所述模版缓冲器相对应的像素数据来渲染。

[0102] 在第二遍次中,在图6d中对标有“1”的像素进行处理以渲染对象PP5、XP1和XP2的像素。如之前已经描述的,渲染PP5的非反射像素和非透明像素可以通过使用图形模型(例如,渲染层)确定每个像素的颜色值来进行。反射对象和透明对象通过使用对象本身和通过它可见的另一对象的信息来渲染。例如,通过对象XP1的镜像对象XOBJ1的渲染可以作为XOBJ1和XP1的像素颜色值的常规混合来进行。也就是说,像素数据可以通过混合来自第一图形对象(例如,“窗户”XP1的有色玻璃)的颜色信息和来自第二图形对象的颜色信息(“树”XOBJ1的镜像的颜色)来形成。在第二遍次中,通过第一图形对象可见的对象的信息已经被计算到画面缓冲器中,因此混合可以通过从画面缓冲器中获取像素颜色值、第一图形对象的所计算的像素颜色值并且适当地混合它们(取像素颜色值的加权平均值)来进行。

[0103] 在上文中,可以如下进行模版缓冲器的设置和使用。用于渲染通过其他对象可见的对象而进行的那些渲染遍次中,可以看透的前景图形对象的那些像素(反射像素和/或透明像素)被标记到模版缓冲器。也就是说,前景对象的不透明像素可能直到最后遍次才会被渲染。然后,模版缓冲器通过将值写入模版缓冲器像素来简单地设置。写入可能会通过重写、通过加法或通过逻辑运算来发生。以这种方式,像素的值的某些比特可以用作一个参数,并且其他比特可以用于其他目的。模版缓冲器的写入可以以并行方式发生,使得对于每个像素执行简单的测试(例如,比较),并且取决于测试的结果写入值。模版缓冲器可以用在渲染中,使得如果模版缓冲器像素满足某个逻辑准则,例如,“等于1”或“大于4”,对应的像素被渲染到画面缓冲器。

[0104] 上文中需要理解的是,反射对象可以通过使用多级依从性(图6c)并且进行多个遍次来在其它反射对象内渲染,其中附加的反射在附加的遍次中被添加到像素值。

[0105] 如之前所描述的,3D图像的渲染可能会发生,使得对象被渲染为渲染层。在这种情况下,该方法可以包括:接收包括第一图形对象的第一渲染层和包括第二图形对象的第二渲染层,第一渲染层和第二渲染层包括像素。第一渲染层包括与从渲染视点观看的第一图形对象相对应的像素,并且第二渲染层包括与从所述渲染视点观看的第二图形对象相对应的像素,其中从渲染视点观察到,第二图形对象被第一对象至少部分地遮蔽。第一渲染层的像素和第二渲染层的像素被放置在渲染空间中,深度值与像素相关联,并且左眼图像和右眼图像使用像素和深度值来渲染。

[0106] 图7a和图7b示出了用于形成和渲染具有反射对象的数字场景的流程图。在阶段710中,用于渲染的第一图形对象被形成。图形对象通过按照惯例描述其几何形状(例如,以顶点、三角形和表面法线的形式)并且给出纹理属性(如颜色信息)来形成。另外,第一对象的透明度和反射率被给出,使得例如其他对象的反射可以被确定。在阶段720中,第二图形对象被形成。该第二图形对象可以是与场景中的真实图形对象相对应的镜像图形对象。如之前所描述的,该镜像可以通过相对于与第一对象相对应的镜面对真实对象进行镜像来形成。在阶段730中,第一对象和第二对象之间的依从性信息被形成。如之前所描述的,这可以以依从性图形(诸如依从性树)的形式发生,其中指示第二图形对象仅通过第一对象可见。在阶段740中,输出图形对象及其依从性的数据被输出到例如文件或传输通道以被传送到渲染设备。

[0107] 在图7b中,从文件或从传输通道接收图形对象的信息和依从性。术语“接收”在这里被理解为意味着从存储器读取信息或通过通信连接接收信息,然后形成在渲染设备的工

作存储器中以便进行渲染。在阶段750中,第一图形对象和第二图形对象的信息(几何形状、颜色、透明度、反射率)被接收。在阶段760中,第一图形和第二图形对象之间的依从性信息被接收。第一图形对象可以基于第一图形对象上的信息来渲染。在阶段770中,对由第一图形对象覆盖的部分的第二图形对象进行渲染,使得第二图形对象在第一图形对象之外不可见。在该阶段中,第一图形对象和第二图形对象的颜色信息可以例如通过alpha混合或其他混合算法混合来形成输出像素。在阶段780中,所得的像素数据被输出以便存储在文件中或用于在屏幕上显示。像素可以以图像格式(或视频格式)压缩。像素数据可以是实际像素的形式,或者像素数据可以以图形图元和纹理信息的形式输出。

[0108] 本发明的各种实施例可以借助驻留在存储器中的计算机程序代码来实现,并且使得相关装置来进行本发明。例如,设备可以包括用于处理、接收和传输数据的电路和电子装置、存储器中的计算机程序代码以及处理器,该处理器当运行计算机程序代码时,使得设备执行实施例的特征。又进一步地,如服务器那样的网络设备可以包括用于处理、接收和传送数据的电路和电子装置、存储器中的计算机程序代码以及处理器,该处理器当运行计算机程序代码时,使得网络设备执行实施例的特征。

[0109] 显然,本发明不是仅限于上文所呈现的实施例,而是可以在所附权利要求的范围内进行修改。

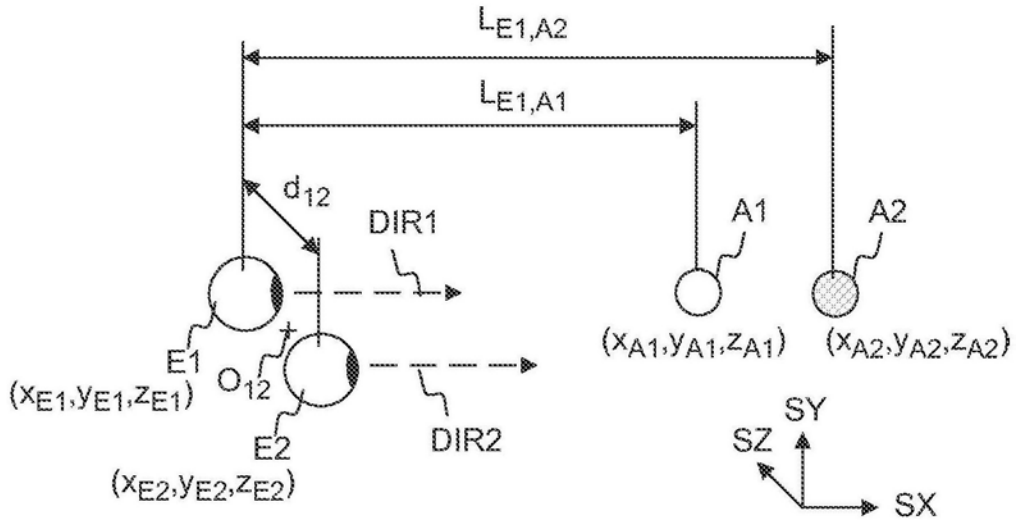


图1a

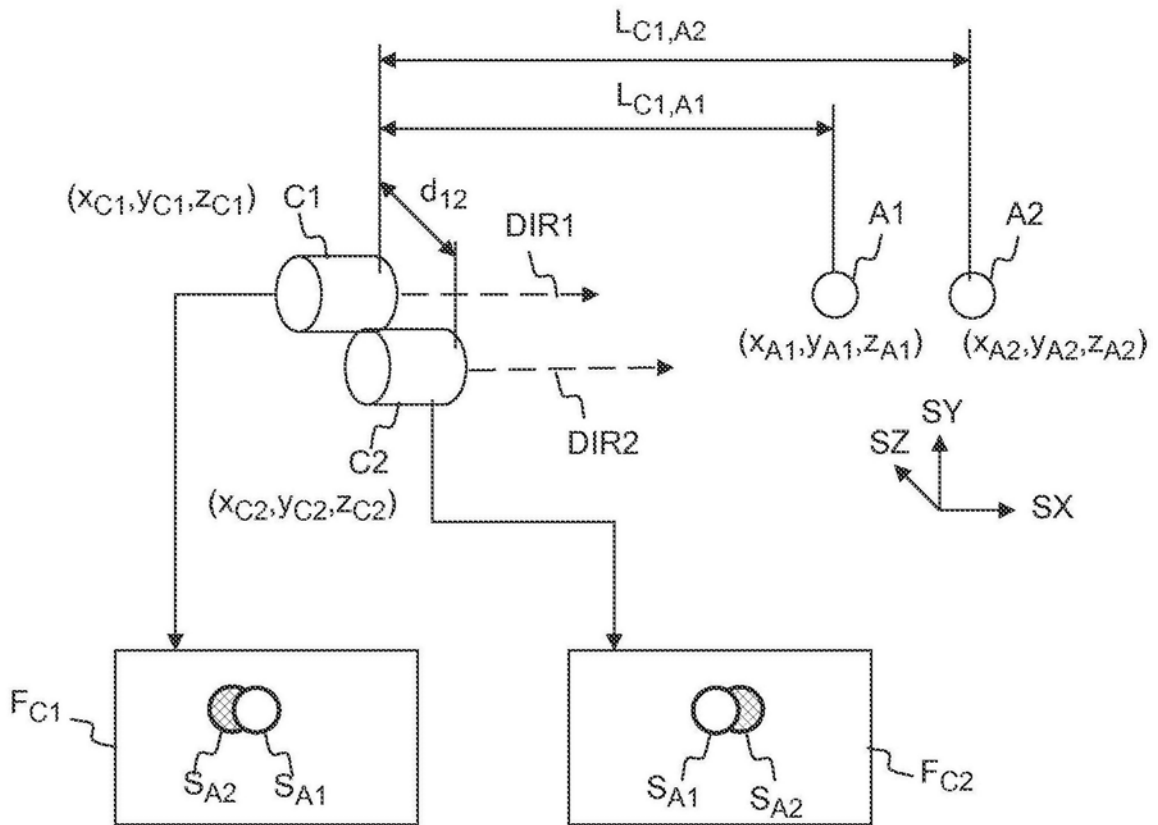


图1b

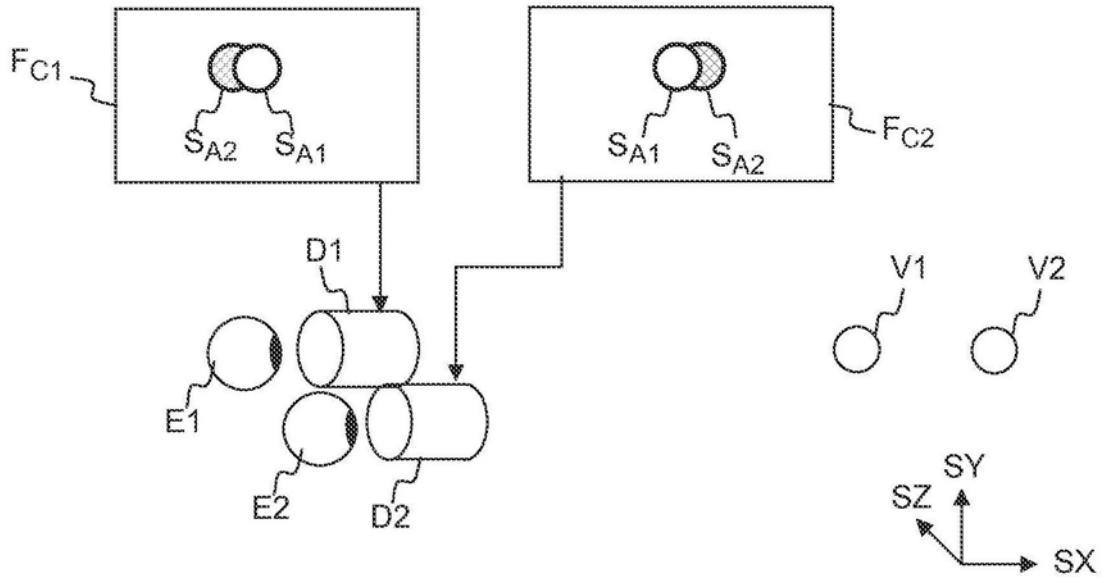


图1c

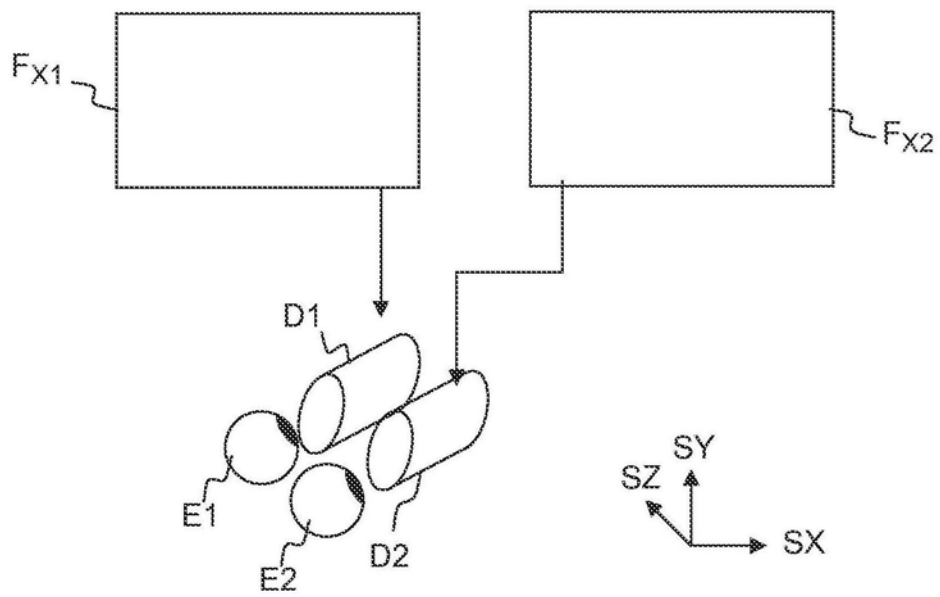


图1d

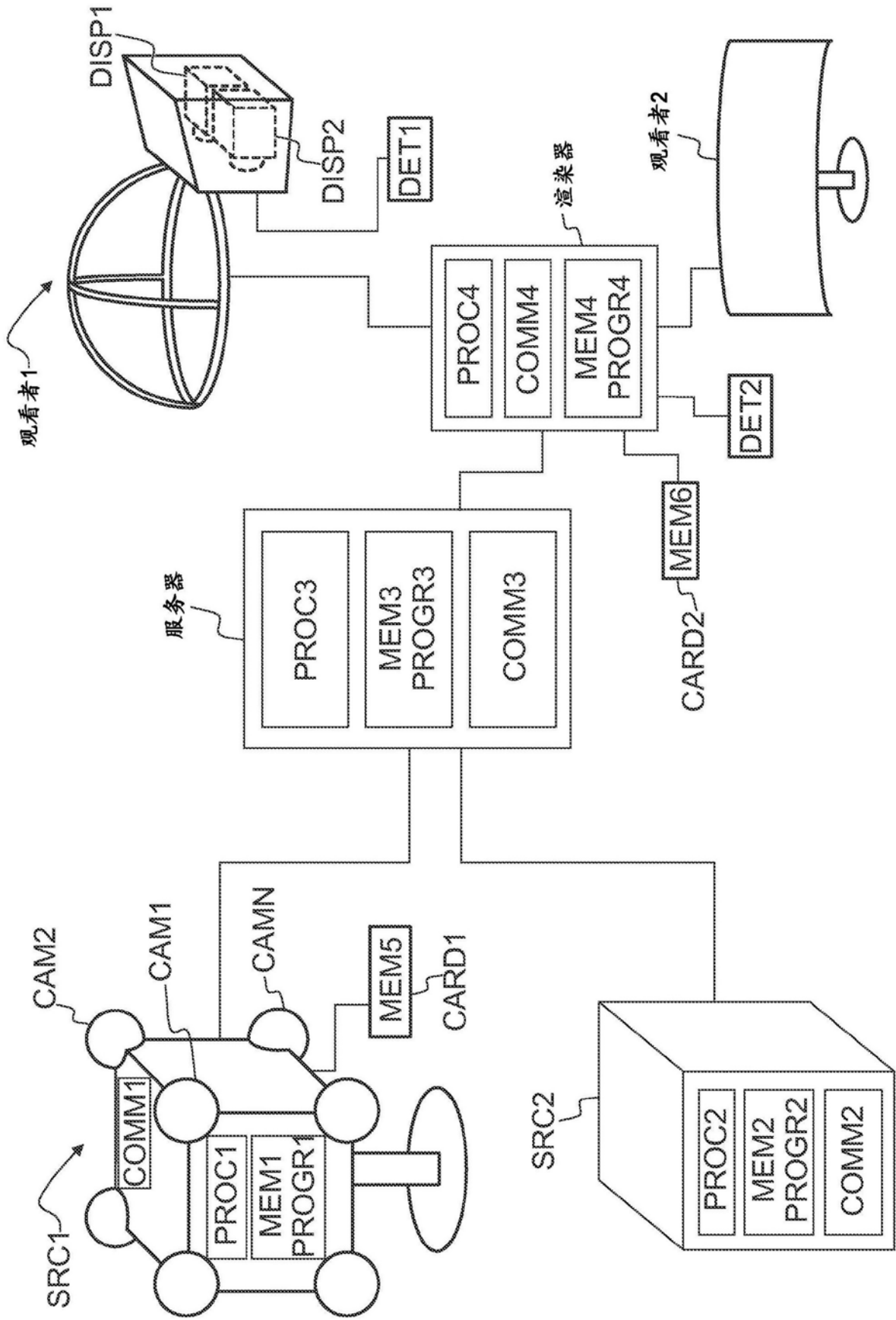


图2a

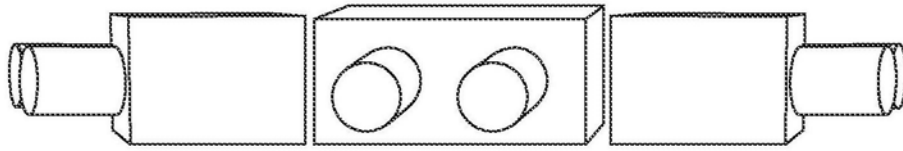


图2b

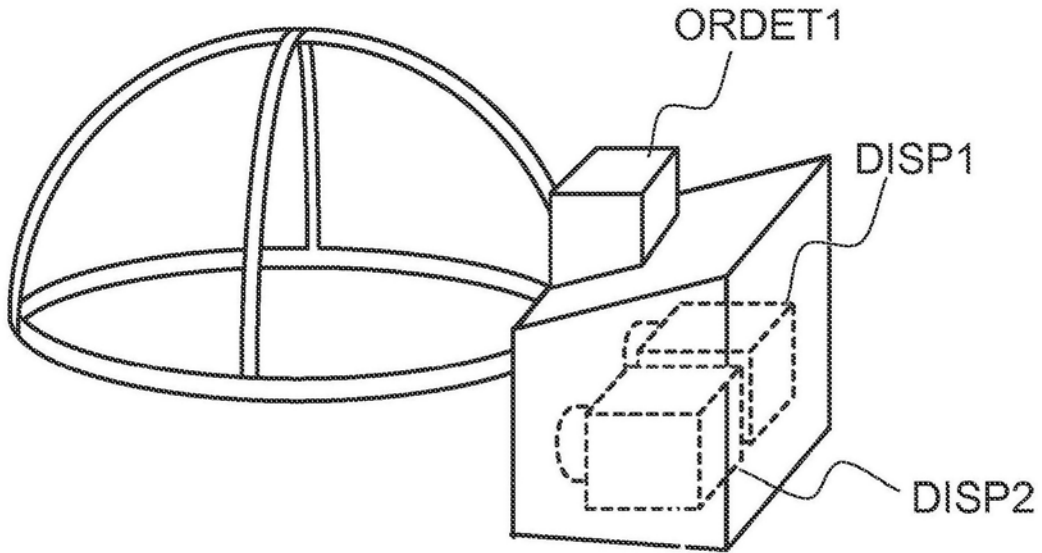


图2c

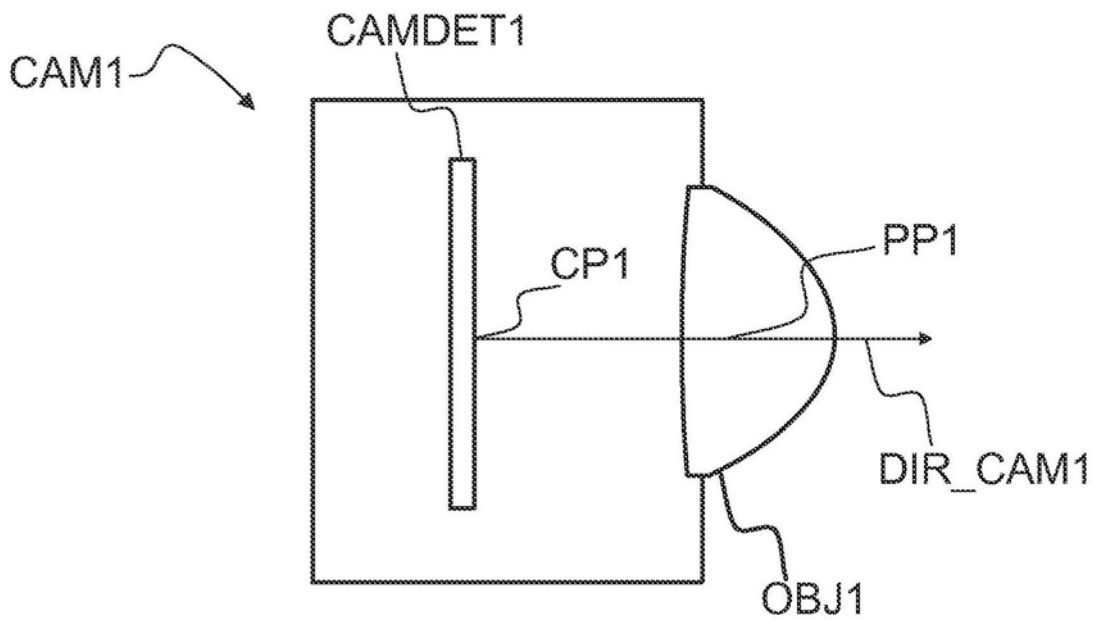


图2d

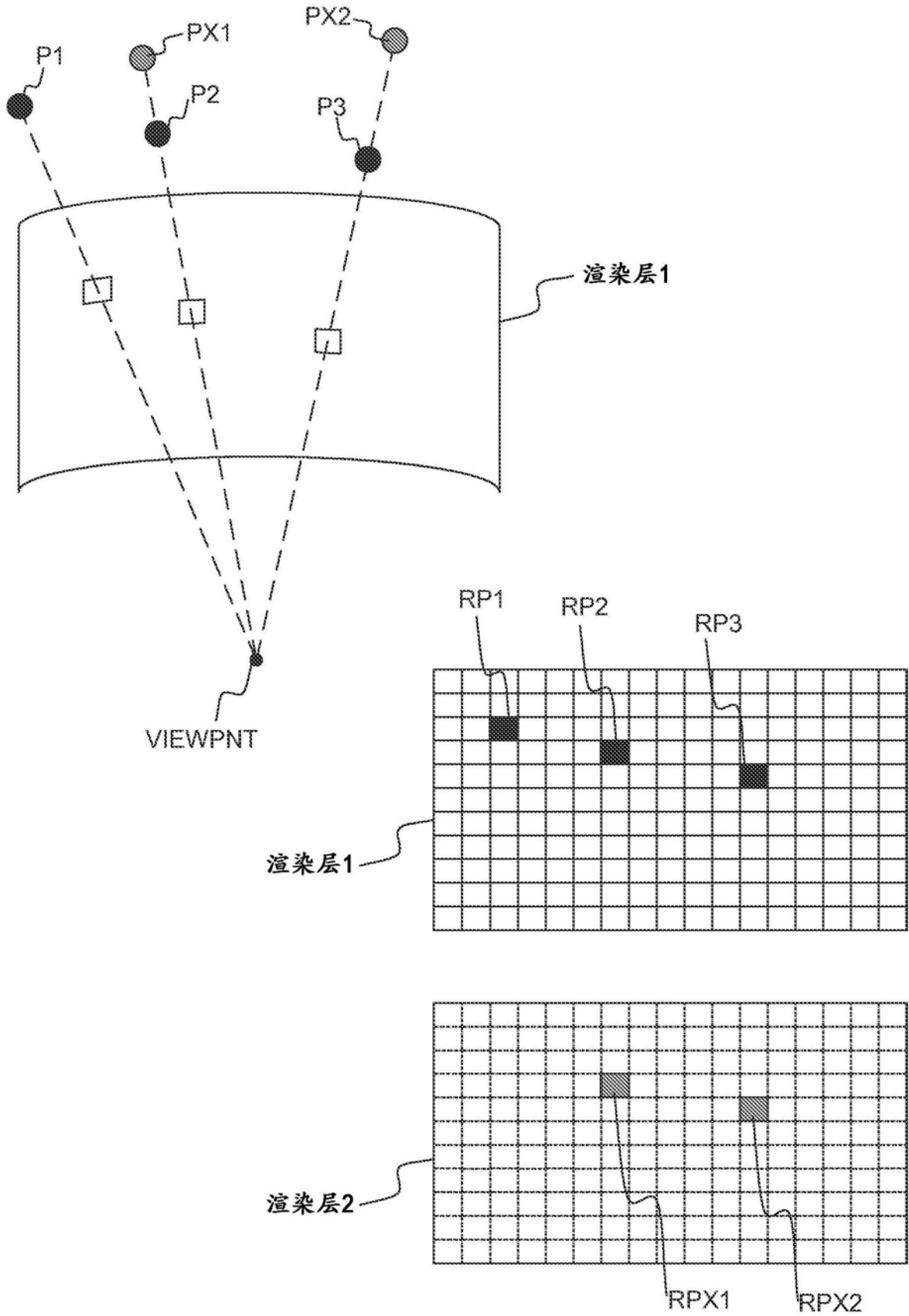


图3a

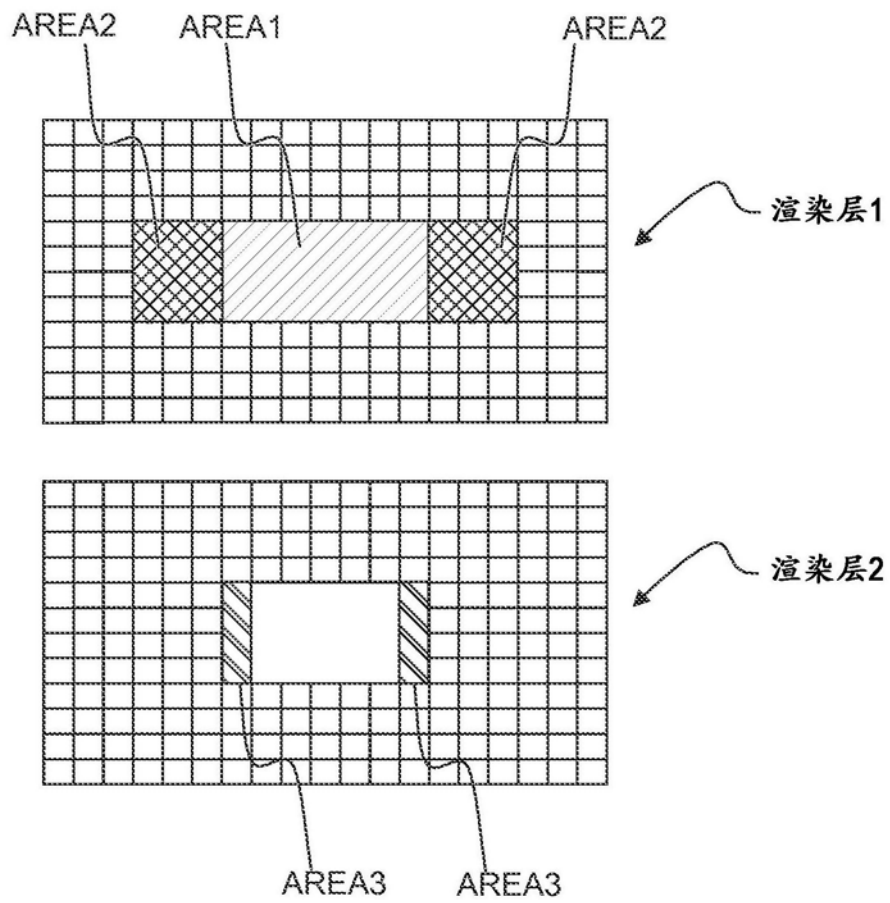
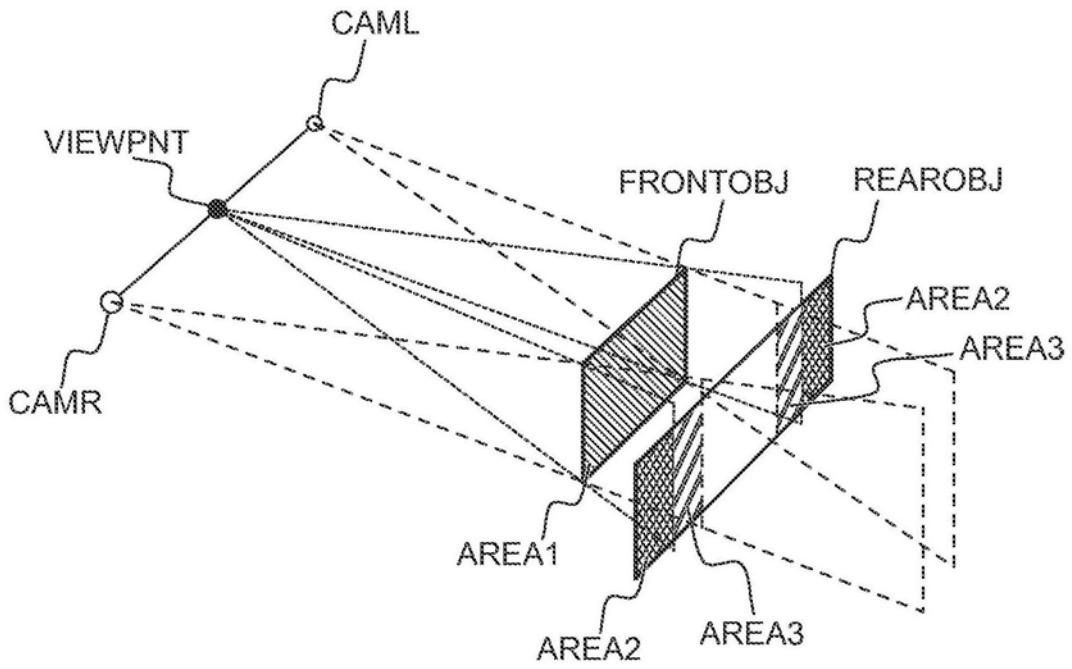


图3b

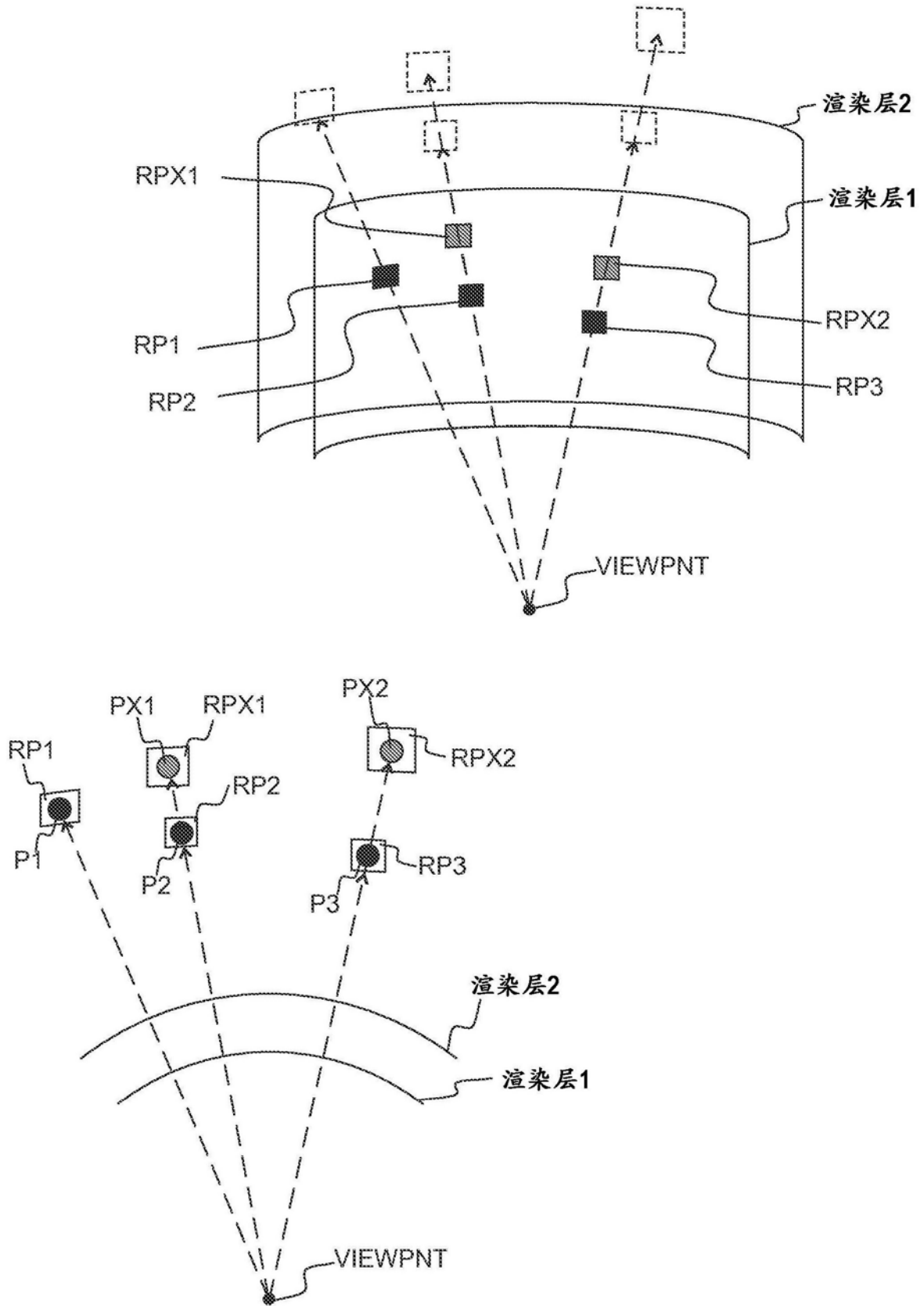


图4

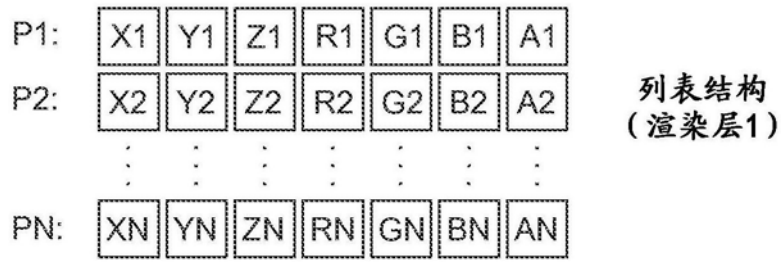


图5a

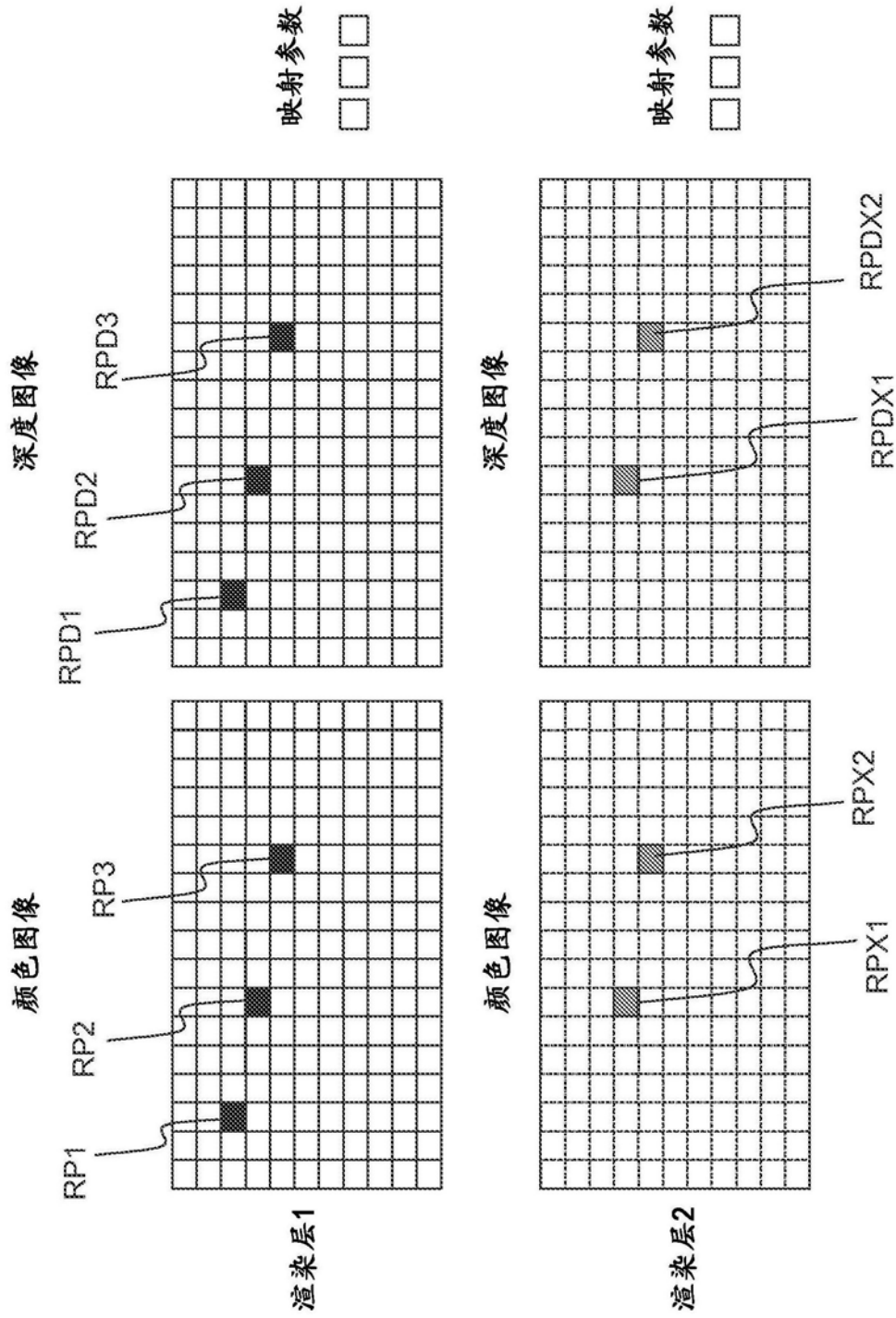


图5b

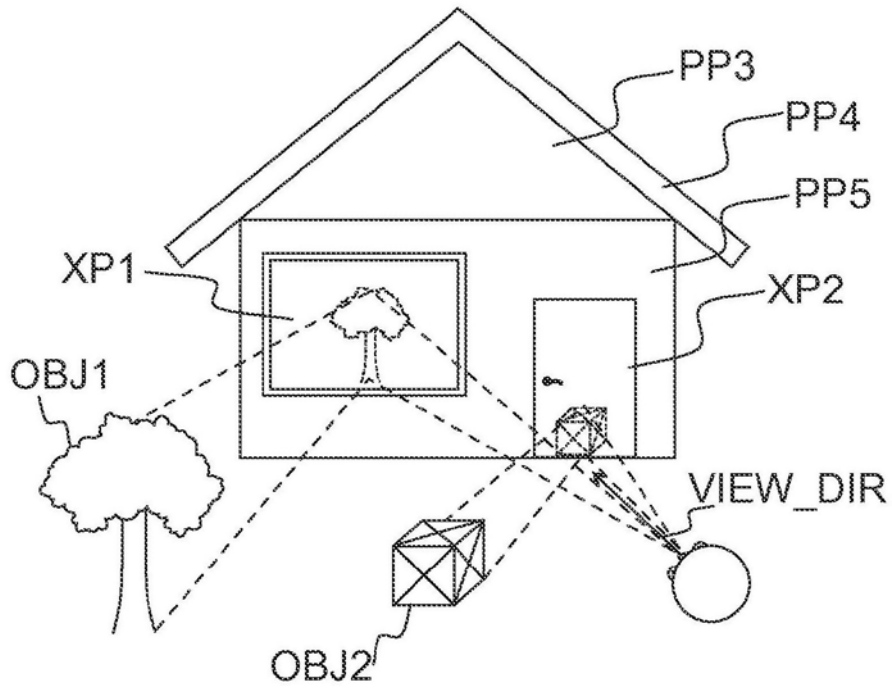


图6a

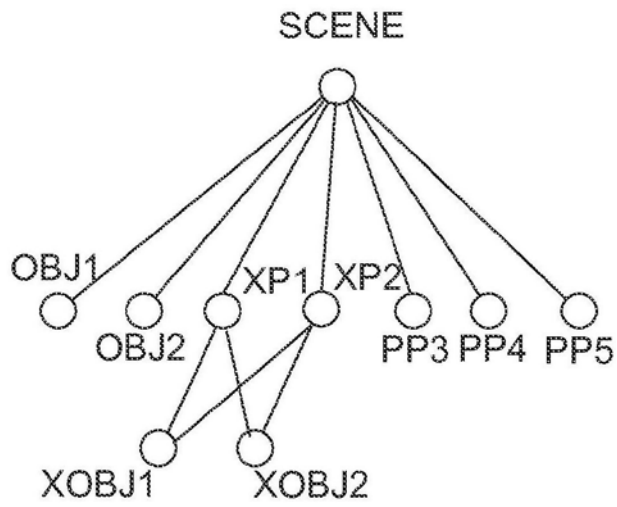


图6b

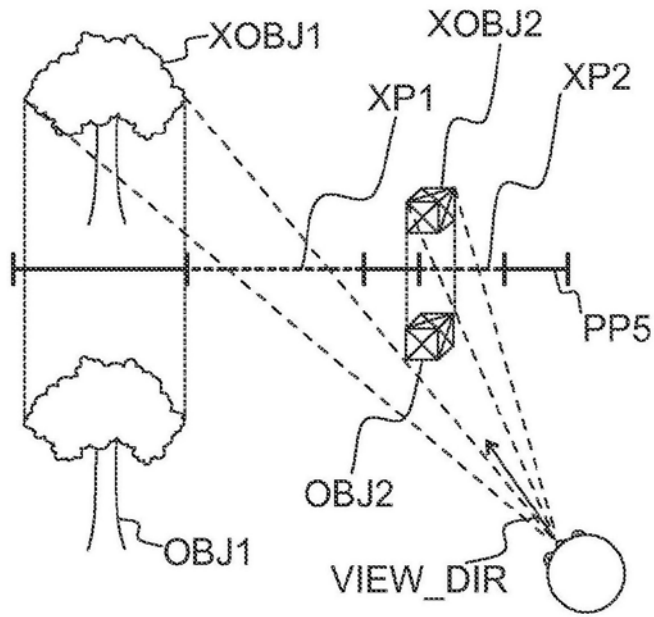


图6c

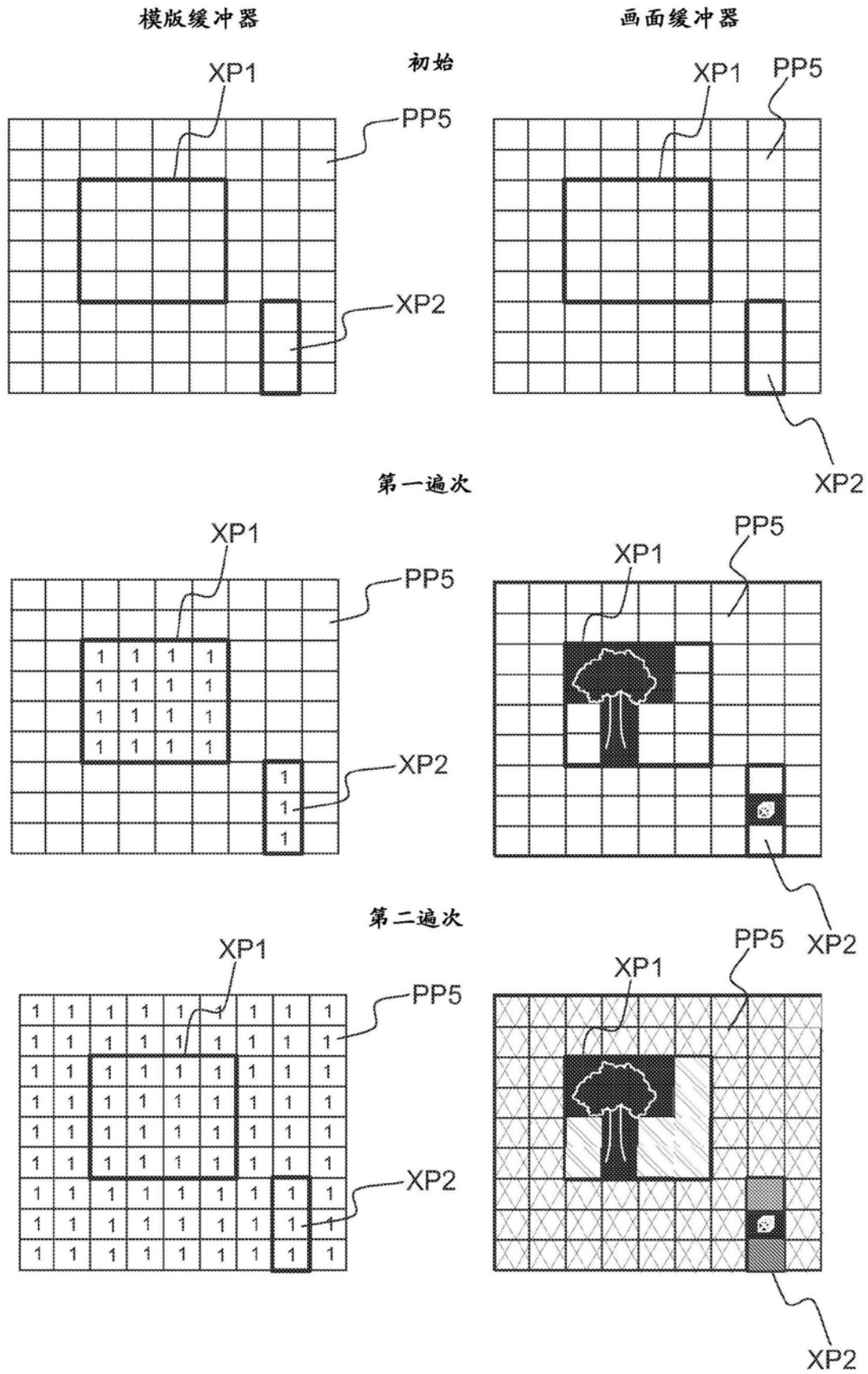


图6d

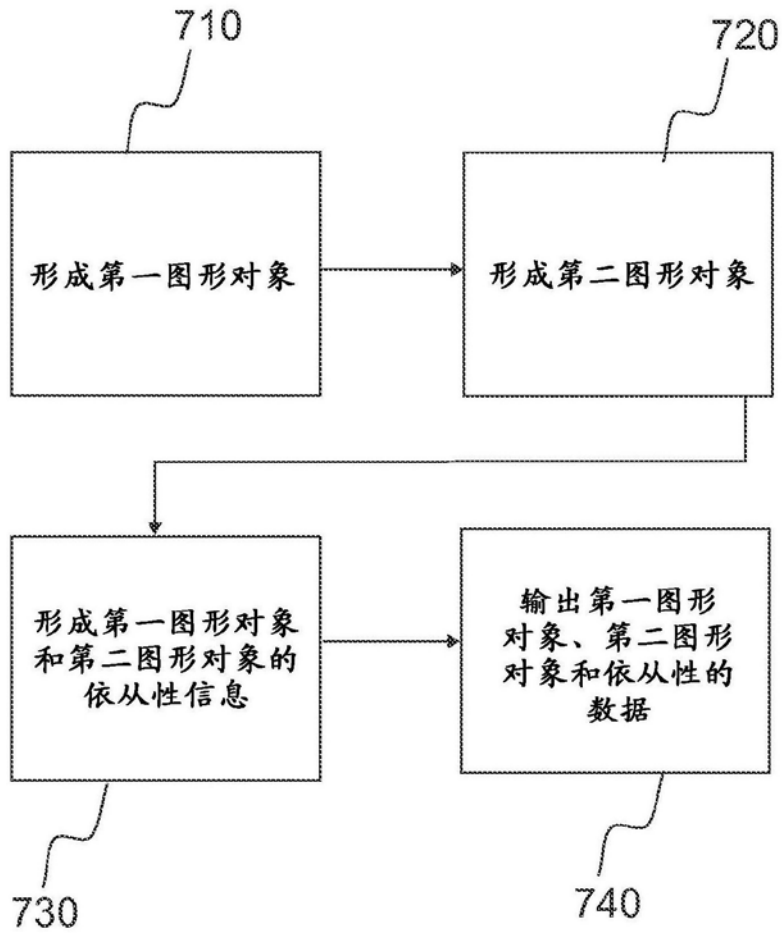


图7a

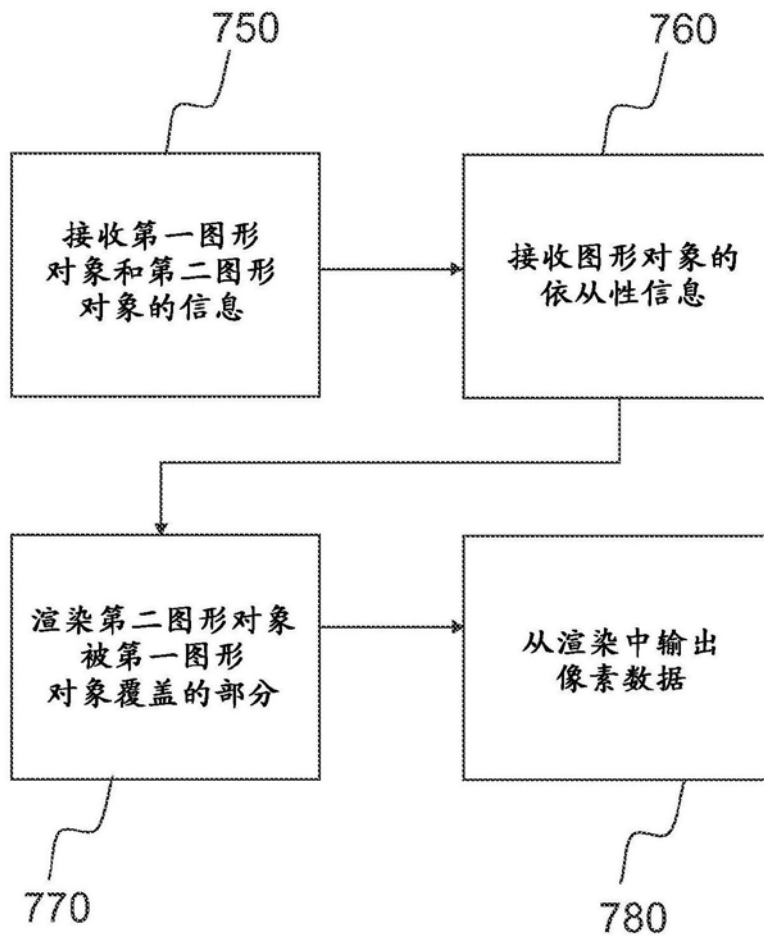


图7b