



(12) 发明专利

(10) 授权公告号 CN 116188552 B

(45) 授权公告日 2023.08.11

(21) 申请号 202310444667.X

G06T 7/194 (2017.01)

(22) 申请日 2023.04.24

G06T 7/11 (2017.01)

(65) 同一申请的已公布的文献号

G06T 7/49 (2017.01)

申请公布号 CN 116188552 A

G06T 1/60 (2006.01)

G06T 1/20 (2006.01)

(43) 申请公布日 2023.05.30

(56) 对比文件

(73) 专利权人 摩尔线程智能科技(北京)有限公司
任公司

CN 109101869 A, 2018.12.28

CN 115935897 A, 2023.04.07

地址 100080 北京市海淀区海淀大街31号2
层209

CN 112116519 A, 2020.12.22

CN 103886633 A, 2014.06.25

(72) 发明人 请求不公布姓名

TW 201131509 A, 2011.09.16

US 11127148 B1, 2021.09.21

(74) 专利代理机构 北京派特恩知识产权代理有
限公司 11270

审查员 黄蓉冰

专利代理师 胡亮 张颖玲

(51) Int. Cl.

G06T 7/50 (2017.01)

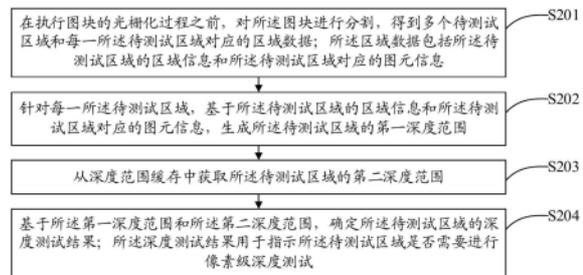
权利要求书3页 说明书20页 附图7页

(54) 发明名称

基于区域的深度测试方法、装置、设备及存储介质

(57) 摘要

本申请实施例公开了一种基于区域的深度测试方法、装置、设备及存储介质,其中,所述方法包括:在执行图块的光栅化过程之前,对所述图块进行分割,得到多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息;针对每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围;从深度范围缓存中获取所述待测试区域的第二深度范围;基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要像素级深度测试。



1. 一种基于区域的深度测试方法,应用于基于图块渲染TBR架构的图形处理器,其特征在于,所述方法包括:

在执行图块的光栅化过程之前,对所述图块进行分割,得到所述图块内多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息;

针对所述图块内每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围;

从深度范围缓存中获取所述待测试区域的第二深度范围;

基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要进一步进行像素级深度测试;

其中,所述方法还包括:

响应于深度信息缓存被分配给所述图块并完成初始化,从所述深度信息缓存中获取所述图块内每一初始区域中每一像素的深度值;所述初始化过程为从内存中加载所述图块内所有像素的深度值并写入到深度信息缓存的过程;

基于所述图块内每一所述初始区域中每一像素的深度值,确定每一所述初始区域对应的第二深度范围;

将每一所述初始区域对应的第二深度范围更新至所述深度范围缓存。

2. 根据权利要求1所述的方法,其特征在于,所述图元信息包括图元的深度面方程;所述基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围,包括:

基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值;

基于所述最小插值深度值和所述最大插值深度值,生成所述待测试区域的第一深度范围。

3. 根据权利要求2所述的方法,其特征在于,所述基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值,包括:

基于所述图元的深度面方程,确定第一斜率和第二斜率;所述第一斜率表征所述图元的深度面在第一方向上的斜率,所述第二斜率为所述图元的深度面在第二方向上的斜率,所述第一方向、所述第二方向与深度方向相互垂直;

基于所述第一斜率、所述第二斜率和所述区域信息,确定最小插值深度值的坐标信息和最大插值深度值的坐标信息;

基于所述最小插值深度值的坐标信息、所述最大插值深度值的坐标信息和所述深度面方程,确定最小插值深度值和最大插值深度值。

4. 根据权利要求1所述的方法,其特征在于,所述从深度范围缓存中获取所述待测试区域的第二深度范围,包括:

获取所述待测试区域在所述图块中的区域位置信息;

基于所述待测试区域的区域位置信息生成所述待测试区域的索引信息;

基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围。

5. 根据权利要求4所述的方法,其特征在于,所述基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围,包括:

基于所述待测试区域的索引信息,在所述深度范围缓存中进行命中测试;

在所述命中测试的命中测试结果表征所述深度范围缓存中存在所述待测试区域的第二深度范围的情况下,获取所述待测试区域的第二深度范围;

在所述命中测试的命中测试结果表征所述深度范围缓存中不存在所述待测试区域的第二深度范围的情况下,从所述深度信息缓存中确定所述待测试区域的第二深度范围并写入所述深度范围缓存。

6. 根据权利要求1所述的方法,其特征在于,所述对所述图块进行分割,得到所述图块内多个待测试区域和每一所述待测试区域对应的区域数据,包括:

基于预设区域尺寸对所述图块进行分割,得到多个初始区域和每一所述初始区域对应的区域信息;

获取所述图块对应的图元的图元信息;所述图元信息包括所述图元的边界框信息;

基于所述图元的边界框信息,对所述多个初始区域进行筛选,得到所述图块对应的多个待测试区域和每一所述待测试区域的区域数据。

7. 根据权利要求1所述的方法,其特征在于,所述基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果,包括:

在所述第一深度范围的最大值小于所述第二深度范围的最小值的情况下,生成第一测试结果;所述第一测试结果用于表征所述待测试区域不需要进行像素级深度测试,并更新所述深度信息缓存中所述待测试区域中各像素的深度值;

在所述第一深度范围和所述第二深度范围存在交集的情况下,生成第二测试结果;所述第二测试结果用于表征所述待测试区域需要进行像素级深度测试;

在所述第一深度范围的最小值大于所述第二深度范围的最大值的情况下,生成第三测试结果;所述第三测试结果用于表征所述待测试区域需要被剔除且不需要进行像素级深度测试。

8. 根据权利要求7所述的方法,其特征在于,所述方法还包括:

响应于所述待测试区域的深度测试结果为所述第一测试结果或所述第二测试结果,利用所述第一深度范围中的最小插值深度值,在所述深度范围缓存中更新所述待测试区域的第二深度范围中的最小缓存深度值。

9. 根据权利要求1至8任一项所述的方法,其特征在于,所述方法还包括:

在前期深度测试和/或后期深度测试过程中,响应于图元的深度值全部写入深度信息缓存,从所述深度信息缓存中读取所述图块对应的每一像素的深度值;

基于所述图块对应的每一像素的深度值,对所述深度范围缓存中每一初始区域对应的第二深度范围进行更新。

10. 一种基于区域的深度测试装置,应用于基于图块渲染TBR架构的图形处理器,其特征在于,所述装置包括:

图块分割模块,用于在执行图块的光栅化过程之前,对所述图块进行分割,得到所述图块内多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息;

区域深度测试模块,用于针对所述图块内每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围;

所述区域深度测试模块,还用于从深度范围缓存中获取所述待测试区域的第二深度范围;

所述区域深度测试模块,还用于基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要进行像素级深度测试;

深度排序模块,用于响应于深度信息缓存被分配给所述图块并完成初始化,从所述深度信息缓存中获取所述图块内每一初始区域中每一像素的深度值;所述初始化过程为从内存中加载所述图块内所有像素的深度值并写入到深度信息缓存的过程;基于所述图块内每一所述初始区域中每一像素的深度值,确定每一所述初始区域对应的第二深度范围;将每一所述初始区域对应的第二深度范围更新至所述深度范围缓存。

11. 一种计算机设备,包括存储器和处理器,所述存储器存储有可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现权利要求1至9任一项所述方法中的步骤。

12. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至9任一项所述方法中的步骤。

基于区域的深度测试方法、装置、设备及存储介质

技术领域

[0001] 本申请涉及但不限于图像处理技术领域,尤其涉及一种基于区域的深度测试方法、装置、设备及存储介质。

背景技术

[0002] 当前在标准的渲染管线中,完成光栅化后直接将有效的像素送入到像素着色(pixel shader)模块中着色。为了提高图形处理器(Graphics Processing Unit, GPU)的性能,GPU厂商在架构上做了许多优化,前期深度测试(Early-Ztest)就其中之一。所谓Early-Ztest,即在执行pixel shader之前就做深度测试,将一些被深度所遮挡的像素提前被剔除,以降低pixel shader的负载。

[0003] 然而,为了实现像素级的前期深度测试,该前期深度测试必须放置在光栅化过程之后,即该前期深度测试需要基于像素光栅化和深度插值等要素,剔除效率不可避免的受到较大的影响。

发明内容

[0004] 有鉴于此,本申请实施例至少提供一种基于区域的深度测试方法、装置、设备、存储介质。

[0005] 本申请实施例的技术方案是这样实现的:

[0006] 一方面,本申请实施例提供一种基于区域的深度测试方法,所述方法包括:在执行图块的光栅化过程之前,对所述图块进行分割,得到多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息;针对每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围;从深度范围缓存中获取所述待测试区域的第二深度范围;基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要像素级深度测试。

[0007] 在一些实施例中,所述图元信息包括图元的深度面方程;所述基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围,包括:基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值;基于所述最小插值深度值和所述最大插值深度值,生成所述待测试区域的第一深度范围。

[0008] 在一些实施例中,所述基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值,包括:基于所述图元的深度面方程,确定第一斜率和第二斜率;所述第一斜率表征所述图元的深度面在第一方向上的斜率,所述第二斜率为所述图元的深度面在第二方向上的斜率,所述第一方向、所述第二方向与深度方向相互垂直;基于所述第一斜率、所述第二斜率和所述区域信息,确定最小插

值深度值的坐标信息和最大插值深度值的坐标信息;基于所述最小插值深度值的坐标信息、所述最大插值深度值的坐标信息和所述深度面方程,确定最小插值深度值和最大插值深度值。

[0009] 在一些实施例中,所述待测试区域由第一区域顶点、第二区域顶点、第三区域顶点和第四区域顶点确定,所述区域信息包括第一区域顶点的顶点坐标、第二区域顶点的顶点坐标、第三区域顶点的顶点坐标和第四区域顶点的顶点坐标;所述基于所述第一斜率、所述第二斜率和所述区域信息,确定最小插值深度值的坐标信息和最大插值深度值的坐标信息,包括:在所述第一斜率大于或等于0、且所述第二斜率大于或等于0的情况下,将所述第一区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第三区域顶点的顶点坐标作为最大插值深度值的坐标信息;在所述第一斜率小于0、且所述第二斜率小于0的情况下,将所述第三区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第一区域顶点的顶点坐标作为最大插值深度值的坐标信息;在所述第一斜率大于或等于0、且所述第二斜率小于0的情况下,将所述第二区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第四区域顶点的顶点坐标作为最大插值深度值的坐标信息;在所述第一斜率小于0、且所述第二斜率大于或等于0的情况下,将所述第四区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第二区域顶点的顶点坐标作为最大插值深度值的坐标信息。

[0010] 在一些实施例中,所述从深度范围缓存中获取所述待测试区域的第二深度范围,包括:获取所述待测试区域在所述图块中的区域位置信息;基于所述待测试区域的区域位置信息生成所述待测试区域的索引信息;基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围。

[0011] 在一些实施例中,所述基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围,包括:基于所述待测试区域的索引信息,在所述深度范围缓存中进行命中测试;在所述命中测试的命中测试结果表征所述深度范围缓存中存在所述待测试区域的第二深度范围的情况下,获取所述待测试区域的第二深度范围;在所述命中测试的命中测试结果表征所述深度范围缓存中不存在所述待测试区域的第二深度范围的情况下,从深度信息缓存中确定所述待测试区域的第二深度范围并写入所述深度范围缓存。

[0012] 在一些实施例中,所述对所述图块进行分割,得到多个待测试区域和每一所述待测试区域对应的区域数据,包括:基于预设区域尺寸对所述图块进行分割,得到多个初始区域和每一所述初始区域对应的区域信息;获取所述图块对应的图元的图元信息;所述图元信息包括所述图元的边界框信息;基于所述图元的边界框信息,对所述多个初始区域进行筛选,得到所述图块对应的多个待测试区域和每一所述待测试区域的区域数据。

[0013] 在一些实施例中,所述基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果,包括:在所述第一深度范围的最大值小于所述第二深度范围的最小值的情况下,生成第一测试结果;所述第一测试结果用于表征所述待测试区域不需要进行像素级深度测试,并更新深度信息缓存中所述待测试区域中各像素的深度值;在所述第一深度范围和所述第二深度范围存在交集的情况下,生成第二测试结果;所述第二测试结果用于表征所述待测试区域需要进行像素级深度测试;在所述第一深度范围的最小值大于所述第二深度范围的最大值的情况下,生成第三测试结果;所述第三测试结果用于表征

所述待测试区域需要被剔除且不需要进行像素级深度测试。

[0014] 在一些实施例中,所述方法还包括:响应于所述待测试区域的深度测试结果 of 所述第一测试结果或所述第二测试结果,利用所述第一深度范围中的最小插值深度值,在所述深度范围缓存中更新所述待测试区域的第二深度范围中的最小缓存深度值。

[0015] 在一些实施例中,所述方法还包括:响应于深度信息缓存被分配给所述图块并完成初始化,从所述深度信息缓存中获取每一所述初始区域中每一像素的深度值;基于每一所述初始区域中每一像素的深度值,确定每一所述初始区域对应的第二深度范围;将每一所述初始区域对应的第二深度范围更新至所述深度范围缓存。

[0016] 在一些实施例中,所述方法还包括:在前期深度测试和/或后期深度测试过程中,响应于图元的深度值全部写入所述深度信息缓存,从所述深度信息缓存中读取所述图块对应的每一像素的深度值;基于所述图块对应的每一像素的深度值,对所述深度范围缓存中每一所述初始区域对应的第二深度范围进行更新。

[0017] 另一方面,本申请实施例提供一种基于区域的深度测试装置,所述装置包括:

[0018] 图块分割模块,用于在执行图块的光栅化过程之前,对所述图块进行分割,得到多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息;

[0019] 区域深度测试模块,用于针对每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围;

[0020] 所述区域深度测试模块,还用于从深度范围缓存中获取所述待测试区域的第二深度范围;

[0021] 所述区域深度测试模块,还用于基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要进一步进行像素级深度测试。

[0022] 再一方面,本申请实施例提供一种计算机设备,包括存储器和处理器,所述存储器存储有可在处理器上运行的计算机程序,所述处理器执行所述程序时实现上述方法中的部分或全部步骤。

[0023] 又一方面,本申请实施例提供一种计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现上述方法中的部分或全部步骤。

[0024] 本申请实施例中,由于在执行图块的光栅化过程之前,设置基于区域的深度测试过程,相较于相关技术中的前期深度测试过程,本申请可以更早的剔除掉不需要渲染的被遮挡物体,不需要进入光栅化阶段,节省了光栅化模块的计算资源;同时,通过设置深度范围缓存,并通过深度范围缓存存储图块中每一区域的第二深度范围,这样,可以直接从该深度范围缓存中进行获取待测试区域的第二深度范围,这样可以提升整体渲染效率。

[0025] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,而非限制本申请的技术方案。

附图说明

[0026] 此处的附图被并入说明书中并构成本说明书的一部分,这些附图示出了符合本申请的实施例,并与说明书一起用于说明本申请的技术方案。

- [0027] 图1为本申请实施例提供的一种典型的TBR管线流程示意图；
- [0028] 图2为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图一；
- [0029] 图3为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图二；
- [0030] 图4为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图三；
- [0031] 图5为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图四；
- [0032] 图6为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图五；
- [0033] 图7为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图六；
- [0034] 图8为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图七；
- [0035] 图9为本申请实施例提供的一种基于区域的深度测试方法的整体架构示意图；
- [0036] 图10为本申请实施例提供的一种区域分割过程示意图；
- [0037] 图11为本申请实施例提供的一种区域深度测试模块的结构示意图；
- [0038] 图12A为本申请实施例提供的一种深度范围比对示意图；
- [0039] 图12B为本申请实施例提供的另一种深度范围比对示意图；
- [0040] 图12C为本申请实施例提供的再一种深度范围比对示意图；
- [0041] 图12D为本申请实施例提供的又一种深度范围比对示意图；
- [0042] 图13为本申请实施例提供的一种基于区域的深度测试装置的组成结构示意图；
- [0043] 图14为本申请实施例提供的一种计算机设备的硬件实体示意图。

具体实施方式

[0044] 为了使本申请的目的、技术方案和优点更加清楚，下面结合附图和实施例对本申请的技术方案进一步详细阐述，所描述的实施例不应视为对本申请的限制，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例，都属于本申请保护的范围内。

[0045] 在以下的描述中，涉及到“一些实施例”，其描述了所有可能实施例的子集，但是可以理解，“一些实施例”可以是所有可能实施例的相同子集或不同子集，并且可以在不冲突的情况下相互结合。所涉及的术语“第一/第二/第三”仅仅是区别类似的对象，不代表针对对象的特定排序，可以理解地，“第一/第二/第三”在允许的情况下可以互换特定的顺序或先后次序，以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。

[0046] 除非另有定义，本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请的目的，不是旨在限制本申请。

[0047] 基于图块渲染(Tile-Based Rendering, TBR)是一种在光学空间中通过规则的网格细分计算机图形图像并分别渲染网格(grid)或图块(Tile)各部分的过程。这种设计的优点在于，与立即绘制整个帧的立即模式渲染系统相比，它减少了对内存和带宽的消耗。这使图块渲染系统在低功耗硬件设备中普遍使用。图块渲染有时也被称为中置排序(sort middle)架构，因为它在绘图管线中间而不是接近结束时进行几何排序。TBR是移动GPU的最常用架构，对降低功耗有显著优势。

[0048] 典型的TBR管线流程如图1所示，TBR管线流程分为前端模块110和后端模块120，其

中,前端模块110包括顶点处理模块111、图形处理模块112和分块(Tiling)模块113;后端模块120包括光栅化模块121、隐藏表面去除(Hidden Surface Removal,HSR)模块122、像素着色模块123和输出合并模块124。

[0049] 前端模块110可以执行顶点和图元的变换(Vertex processing)、图形处理(包括clip/cull等),然后在Tiling阶段完成屏幕的切分、记录覆盖到Tile的图形数据,将上述生成信息写入到系统内存130中。这样,该系统内存130中可以存储图块信息(Primitive List)和顶点信息(Vertex Data),其中,PrimitiveList为长度为Tile的固定长度数组,该数组中,每个元素是一个linked list,存的是和当前Tile相交的所有三角形的指针,指针指向VertexData;Vertex Data存放顶点和顶点属性数据。

[0050] 后端模块120主要执行光栅化(Raster)、Depth Test、像素着色等操作,最后输出到渲染目标(Render target)上。对于每个Tile来说,由于其数据量不大,可以将其所需的深度数据(Depth)、纹理数据(texture)或者色彩数据(color)加载到GPU存储在片上存储SRAM中,即图中的片载内存140中;例如,隐藏表面去除模块122可以将深度数据(Depth)存储到片载内存140中的深度数据缓存(Depth Buffer)中,像素着色模块123可以将纹理数据(texture)存储到片载内存140中的纹理数据缓存中,输出合并模块124可以将色彩数据(color)存储到片载内存140中的色彩数据缓存(color Buffer)中。

[0051] 在渲染流程中,渲染对象(画面)会被分割为多个图块(Tile),这样,上述片载内存140可以容纳这个图块的所有数据。在至少一个绘图指令到达GPU后,前端模块110依次处理每一绘图指令,并将对应的图块信息和顶点信息存储至系统内存130中,直至系统内存130中存储的数据达到预设阈值或者至少一个绘图指令均已处理完毕。后端模块120将以图块为单位从系统内存130中读取对应的顶点信息,并进行后续处理。这样,由于将后端模块120对系统内存130的访问变化为后端模块120对片载内存140的访问,可以提升渲染效率。

[0052] 本申请实施例提供一种基于区域的深度测试方法,该方法可以由计算机设备的处理器执行。其中,计算机设备指的可以是服务器、笔记本电脑、平板电脑、台式计算机、智能电视、机顶盒、移动设备(例如移动电话、便携式视频播放器、个人数字助理、专用消息设备、便携式游戏设备)等具备数据处理能力的设备。

[0053] 图2为本申请实施例提供的一种基于区域的深度测试方法的实现流程示意图一,该方法可以由计算机设备的处理器执行,将结合图2示出的步骤进行说明。

[0054] 步骤S201、在执行图块的光栅化过程之前,对所述图块进行分割,得到多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息。

[0055] 在一些实施例中,在TBR架构中,后端模块是以图块为单位进行后续渲染过程的,相较于相关技术中的深度测试方法,本申请为了更早的进行深度测试,在执行图块的光栅化过程之前,执行本申请的基于区域的深度测试过程。

[0056] 在一些实施例中,可以按照预设分割规则,将该图块分割为相同大小的多个区域,得到的多个区域不存在交集。此时,将存在图元覆盖到的区域作为上述待测试区域,可以得到该多个待测试区域。其中,该待测试区域的区域信息可以包括该待测试区域的区域范围、区域坐标等信息,该待测试区域对应的图元信息为覆盖到该待测试区域的图元的图元形状信息、图元深度信息等。

[0057] 步骤S202、针对每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围。

[0058] 在一些实施例中,该第一深度范围为待测试区域在渲染当前图元的情况下,该待测试区域中各像素的深度值的范围。由此,可以基于该待测试区域的区域范围,和该待测试区域对应的当前图元的图元深度信息和图元形状信息,确定该待测试区域在渲染该图元的情况下,待测试区域内深度值的范围,即第一深度范围。

[0059] 步骤S203、从深度范围缓存中获取所述待测试区域的第二深度范围。

[0060] 在一些实施例中,该深度范围缓存用于实时缓存图块中各个区域的深度值范围,也就是说,该深度范围缓存中缓存有各个待测试区域对应的第二深度范围。其中,该第二深度范围表征待测试区域的深度值的历史范围,可以理解为,针对该待测试区域,在执行当前图元的区域深度测试过程之前,已经完成其他图元的区域深度测试过程,此时,深度范围缓存中存储的待测试区域的第二深度范围为待测试区域在渲染该其他图元的情况下,该待测试区域中各像素的深度值的范围。

[0061] 在一些实施例中,在初始状态下,即深度范围缓存刚被分配该当前图块的情况下,该深度范围缓存中该待测试区域的第二深度范围是基于深度信息缓存中该待测试区域中各像素的深度值确定的。

[0062] 步骤S204、基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要进一步进行像素级深度测试。

[0063] 在一些实施例中,基于该第一深度范围和第二深度范围之间的数值大小关系,可以确定后续渲染过程中,所述待测试区域是否需要渲染当前图元,进而生成用于指示所述待测试区域是否需要进一步进行像素级深度测试的深度测试结果。

[0064] 本申请实施例中,由于在执行图块的光栅化过程之前,设置基于区域的深度测试过程,相较于相关技术中的前期深度测试过程,本申请可以更早的剔除掉不需要渲染的被遮挡物体,不需要进入光栅化阶段,节省了光栅化模块的计算资源;同时,通过设置深度范围缓存,并通过深度范围缓存存储图块中每一区域的第二深度范围,这样,可以直接从该深度范围缓存中进行获取待测试区域的第二深度范围,这样,可以提升整体渲染效率。

[0065] 图3是本申请实施例提供的基于区域的深度测试方法的一个可选的流程示意图二,该方法可以由计算机设备的处理器执行。所述图元信息包括图元的深度面方程;基于图2,图2中的步骤S202可以更新为步骤S301至步骤S302,将结合图3示出的步骤进行说明。

[0066] 步骤S301、基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值。

[0067] 在一些实施例中,该待测试区域的区域信息可以包括该待测试区域中每一像素的位置坐标,结合该图元对应的深度面方程,可以确定该待测试区域中每一像素的深度值,对每一像素的深度值进行排序,将最大深度值作为该最大插值深度值,将最小深度值作为该最小插值深度值。

[0068] 在一些实施例中,可以通过步骤S3011至步骤S3013实现上述基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值。

[0069] 步骤S3011、基于所述图元的深度面方程,确定第一斜率和第二斜率。

[0070] 其中,所述第一斜率表征所述图元的深度面在第一方向上的斜率,所述第二斜率为所述图元的深度面在第二方向上的斜率,所述第一方向、所述第二方向与深度方向相互垂直。

[0071] 步骤S3012、基于所述第一斜率、所述第二斜率和所述区域信息,确定最小插值深度值的坐标信息和最大插值深度值的坐标信息。

[0072] 其中,所述待测试区域由第一区域顶点、第二区域顶点、第三区域顶点和第四区域顶点确定,所述区域信息包括第一区域顶点的顶点坐标、第二区域顶点的顶点坐标、第三区域顶点的顶点坐标和第四区域顶点的顶点坐标。上述基于所述第一斜率、所述第二斜率和所述区域信息,确定最小插值深度值的坐标信息和最大插值深度值的坐标信息,可以通过以下方式实现:

[0073] 在一些实施例中,在所述第一斜率大于或等于0、且所述第二斜率大于或等于0的情况下,将所述第一区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第三区域顶点的顶点坐标作为最大插值深度值的坐标信息。

[0074] 在一些实施例中,在所述第一斜率小于0、且所述第二斜率小于0的情况下,将所述第三区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第一区域顶点的顶点坐标作为最大插值深度值的坐标信息。

[0075] 在一些实施例中,在所述第一斜率大于或等于0、且所述第二斜率小于0的情况下,将所述第二区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第四区域顶点的顶点坐标作为最大插值深度值的坐标信息。

[0076] 在一些实施例中,在所述第一斜率小于0、且所述第二斜率大于或等于0的情况下,将所述第四区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第二区域顶点的顶点坐标作为最大插值深度值的坐标信息。

[0077] 步骤S3013、基于所述最小插值深度值的坐标信息、所述最大插值深度值的坐标信息和所述深度面方程,确定最小插值深度值和最大插值深度值。

[0078] 其中,将上述最小插值深度值的坐标信息代入该深度面方程,可以得到最小插值深度值,将上述最大插值深度值的坐标信息代入该深度面方程,可以得到最大插值深度值。

[0079] 步骤S302、基于所述最小插值深度值和所述最大插值深度值,生成所述待测试区域的第一深度范围。

[0080] 本申请实施例中,通过待测试区域中图元的深度面方程确定该待测试区域的第一深度范围,进而可以实现后续的低精度深度测试,可提早地剔除掉被遮挡物体,可以降低后续渲染模块的工作负载,提升渲染管线的性能。

[0081] 图4是本申请实施例提供的基于区域的深度测试方法的一个可选的流程示意图三,该方法可以由计算机设备的处理器执行。基于图2,图2中的步骤S203可以更新为步骤S401至步骤S403,将结合图4示出的步骤进行说明。

[0082] 步骤S401、获取所述待测试区域在所述图块中的区域位置信息。

[0083] 在一些实施例中,该区域位置信息用于表征该待测试区域在图块中的相对位置。例如,在图块尺寸为 32×32 ,预设区域尺寸为 16×16 的情况下,可以将图块分割为4个初始区域,这4个初始区域的区域位置信息分别为 $(0,0)$ 、 $(1,0)$ 、 $(0,1)$ 、 $(1,1)$ 。

[0084] 步骤S402、基于所述待测试区域的区域位置信息生成所述待测试区域的索引信息。

[0085] 在一些实施例中,上述深度范围缓存中存储有该图块中每一初始区域的第二深度范围,且每一初始区域的第二深度范围在该深度范围缓存中的存储地址与每一初始区域的区域位置信息相关。基于此,可以预设一个索引转换算法,该索引转换算法可以将待测试区域的区域位置信息转换为待测试区域的索引信息,进而可以基于该索引信息确定该待测试区域的存储地址,之后从深度范围缓存中找到该待测试区域的第二深度范围。

[0086] 步骤S403、基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围。

[0087] 在一些实施例中,可以通过步骤S4031至步骤S4033实现上述基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围。

[0088] 步骤S4031、基于所述待测试区域的索引信息,在所述深度范围缓存中进行命中测试。

[0089] 其中,该命中测试用于确定深度范围缓存中是否存在该待测试区域对应的索引信息,若存在该索引信息,则可以直接基于该索引信息找到存储地址,进而获取对应的第二深度范围;若不存在该索引信息,则表示当前深度范围缓存中不存在所述待测试区域的第二深度范围。

[0090] 步骤S4032、在所述命中测试的命中测试结果表征所述深度范围缓存中存在所述待测试区域的第二深度范围的情况下,获取所述待测试区域的第二深度范围。

[0091] 此时,直接基于该索引信息找到存储地址,进而获取对应的第二深度范围。

[0092] 步骤S4033、在所述命中测试的命中测试结果表征所述深度范围缓存中不存在所述待测试区域的第二深度范围的情况下,从深度信息缓存中确定所述待测试区域的第二深度范围并写入所述深度范围缓存。

[0093] 此时,当前深度范围缓存中不存在所述待测试区域的第二深度范围,此时,需要从深度信息缓存中获取该待测试区域中每一像素的深度值,对位于该待测试区域内的每一像素的深度值进行排序,确定待测试区域对应的最大深度值和最小深度值,以得到该第二深度范围并写入所述深度范围缓存。

[0094] 本申请实施例中,通过深度范围缓存存储图块中每一区域的第二深度范围,这样,在获取待测试区域的第二深度范围的过程中,不需要从深度信息缓存中进行获取,直接从该深度范围缓存中进行获取,这样,可以提升数据查询效率,进而提升深度测试效率,从整体上提高了渲染速度。

[0095] 图5是本申请实施例提供的基于区域的深度测试方法的一个可选的流程示意图四,该方法可以由计算机设备的处理器执行。基于上述任一实施例,以基于图2为例,图2中的步骤S201可以更新为步骤S501至步骤S503,将结合图5示出的步骤进行说明。

[0096] 步骤S501、基于预设区域尺寸对所述图块进行分割,得到多个初始区域和每一所述初始区域对应的区域信息。

[0097] 在一些实施例中,该预设区域尺寸可以但不限于是以下任意之一:16×16,8×8和4×4。

[0098] 在一些实施例中,该区域信息可以包括该初始区域在图块中的区域位置信息和初

始区域的顶点信息。

[0099] 其中,该区域位置信息表征初始区域在图块中的相对位置,例如,在图块尺寸为 32×32 ,预设区域尺寸为 16×16 的情况下,可以将图块分割为4个初始区域,这4个初始区域的区域位置信息分别为 $(0,0)$ 、 $(1,0)$ 、 $(0,1)$ 、 $(1,1)$ 。

[0100] 其中,该初始区域的顶点信息为该初始区域的4个区域顶点的坐标信息。

[0101] 步骤S502、获取所述图块对应的图元的图元信息;所述图元信息包括所述图元的边界框信息。

[0102] 在一些实施例中,该图块对应的图元为落入该图块的图元。图元的边界框信息为该图元所在的矩形框的边界框信息,示例性的,该边界框信息可以由 $BBoxMinX$, $BBoxMaxX$, $BBoxMinY$, $BBoxMaxY$ 确定,也就是说,图元的边界框信息用于表示该图元在X方向上的坐标范围和在Y方向上的坐标范围。

[0103] 在一些实施例中,该图元信息还可以包括以下至少之一:图元的边的方程、图元的面的方程。

[0104] 步骤S503、基于所述图元的边界框信息,对所述多个初始区域进行筛选,得到所述图块对应的多个待测试区域和每一所述待测试区域的区域数据。

[0105] 在一些实施例中,基于该图元的边界框信息和每一初始区域的区域信息,可以在该多个初始区域中确定与该图元所在矩形框存在交集的初始区域,并将与矩形框存在交集的初始区域作为该待测试区域。

[0106] 在一些实施例中,针对每一待测试区域,该待测试区域的区域信息为该待测试区域对应的初始区域的区域信息,该待测试区域的图元信息为上述图块对应的图元的图元信息。

[0107] 本申请实施例中,通过将图块分割为区域,实现了以区域为粒度的低精度深度测试,进而可以降低后续渲染模块的工作负载,提升渲染管线的性能。

[0108] 图6是本申请实施例提供的基于区域的深度测试方法的一个可选的流程示意图五,该方法可以由计算机设备的处理器执行。基于上述任一实施例,以基于图2为例,图2中的步骤S204可以更新为步骤S601至步骤S603,将结合图6示出的步骤进行说明。

[0109] 步骤S601、在所述第一深度范围的最大值小于所述第二深度范围的最小值的情况下,生成第一测试结果;所述第一测试结果用于表征所述待测试区域不需要进行像素级深度测试,并更新深度信息缓存中所述待测试区域中各像素的深度值。

[0110] 在一些实施例中,该第一深度范围由最大插值深度值和最小插值深度值确定,该第二深度范围由最大缓存深度值和最小缓存深度值确定。

[0111] 在一些实施例中,在该最大插值深度值小于该最小缓存深度值的情况下,此时,所述第一深度范围和所述第二深度范围不存在交集,同时,第一深度范围整体位于第二深度范围之前,因此,可以生成用于表征所述待测试区域不需要进行像素级深度测试,并更新深度信息缓存中所述待测试区域中各像素的深度值。

[0112] 其中,由于该第一深度范围和第二深度范围不存在交集,且第一深度范围整体位于第二深度范围之前,因此,可以直接基于当前图元在该待测试区域中各像素的深度值,更新深度信息缓存中对应位置的像素的深度值。

[0113] 步骤S602、在所述第一深度范围和所述第二深度范围存在交集的情况下,生成第

二测试结果;所述第二测试结果用于表征所述待测试区域需要进行像素级深度测试。

[0114] 在一些实施例中,在最小插值深度值小于或等于最小缓存深度值,且最小缓存深度值小于或等于最大插值深度值的情况下,所述第一深度范围和所述第二深度范围存在交集。

[0115] 在另一些实施例中,在最小插值深度值小于或等于最大缓存深度值,且最大缓存深度值小于或等于最大插值深度值的情况下,所述第一深度范围和所述第二深度范围存在交集。

[0116] 在上述两种情况中的任意一种情况下,可以生成表征所述待测试区域需要进行像素级深度测试的第二测试结果。

[0117] 其中,在得到该第二测试结果,并对该待测试区域进行像素级深度测试之后,基于像素级深度测试的测试结果,更新深度信息缓存中对应位置的像素的深度值。即,针对任一像素,在该像素在当前图元中对应的深度值小于该像素在深度信息缓存中的深度值的情况下,基于该像素在当前图元中对应的深度值对深度信息缓存中像素的深度值进行更新;在该像素在当前图元中对应的深度值大于或等于该像素在深度信息缓存中的深度值的情况下,保持该像素在深度信息缓存中的深度值不变。

[0118] 步骤S603、在所述第一深度范围的最小值大于所述第二深度范围的最大值的情况下,生成第三测试结果;所述第三测试结果用于表征所述待测试区域需要被剔除且不需要进行像素级深度测试。

[0119] 在一些实施例中,在该最小插值深度值大于该最大缓存深度值的情况下,此时,所述第一深度范围和所述第二深度范围不存在交集,生成用于表征所述待测试区域需要被剔除且不需要进行像素级深度测试的第三测试结果。

[0120] 其中,在得到该第三测试结果之后,由于剔除当前图元在该待测试区域的深度值,不需要进行像素级深度测试。因此,相比于第一测试结果和第二测试结果,在当前情况下,维持深度信息缓存中该待测试区域的各像素的深度值不变。

[0121] 在一些实施例中,所述方法还可以包括步骤S604。

[0122] 步骤S604、响应于所述待测试区域的深度测试结果为所述第一测试结果或所述第二测试结果,利用所述第一深度范围中的最小插值深度值,在所述深度范围缓存中更新所述待测试区域的第二深度范围中的最小缓存深度值。

[0123] 在一些实施例中,在待测试区域的深度测试结果为所述第一测试结果或所述第二测试结果的情况下,表示当前进行深度测试的待测试区域的最小插值深度值已经小于该深度范围缓存中所述待测试区域的第二深度范围中的最小缓存深度值。此时,需要将利用该最小插值深度值更新最小缓存深度值,即将深度范围缓存中待测试区域的第二深度范围中的最小缓存深度值减少至该最小插值深度值。

[0124] 本申请实施例中,针对每一待测试区域,通过比较通过插值得到的该待测试区域的第一深度范围和深度范围缓存中该待测试区域的第二深度范围,可以快速提早地剔除掉被遮挡物体,不必进入光栅化阶段。这样,可以降低工作负载,提升渲染管线的性能。

[0125] 图7是本申请实施例提供的基于区域的深度测试方法的一个可选的流程示意图六,该方法可以由计算机设备的处理器执行。基于上述任一实施例,以基于图2为例,所述方法还可以包括步骤S701至步骤S703,将结合图7示出的步骤进行说明。

[0126] 步骤S701、响应于深度信息缓存被分配给所述图块并完成初始化,从所述深度信息缓存中获取每一所述初始区域中每一像素的深度值。

[0127] 在本实施例中,上述初始化过程包括:从内存中加载该图块所有像素的深度值并写入到深度信息缓存(depthbuffer)中。此时,深度信息缓存中缓存的深度值不是无穷大,是0至1之间的值。相应地,深度范围缓存也基于该深度信息缓存中缓存的各像素的深度值被初始化,已完成初始化过程。

[0128] 其中,由于本申请实施例提供的基于区域的深度测试方法应用于基于TBR架构的图像处理器,因此,该TBR架构的后端模块是以图块为单位处理渲染任务的。

[0129] 在一些实施例中,在后端模块开始处理一个图块对应的渲染任务的情况下,需要将深度信息缓存分配给该图块,并完成初始化;之后,将从所述深度信息缓存中获取每一所述初始区域中每一像素的深度值。

[0130] 步骤S702、基于每一所述初始区域中每一像素的深度值,确定每一所述初始区域对应的第二深度范围。

[0131] 在一些实施例中,针对每一所述初始区域,对位于该初始区域内的每一像素的深度值进行排序,确定初始区域对应的最大深度值和最小深度值。其中,该第二深度范围由最大缓存深度值和最小缓存深度值确定。进而可以将该最大深度值作为初始区域的最大缓存深度值,将该最小深度值作为初始区域的最小缓存深度值。

[0132] 步骤S703、将每一所述初始区域对应的第二深度范围更新至所述深度范围缓存。

[0133] 在一些实施例中,在该深度范围缓存中,将得到的每一初始区域对应的最大缓存深度值和最小缓存深度值,作为每一初始区域对应的第二深度范围。

[0134] 本申请实施例中,在将深度信息缓存分配给当前图块的情况下,基于深度信息缓存中该图块的各个像素的深度值,对深度范围缓存中存储的各初始区域的第二深度范围进行更新。这样,可以使深度范围缓存更加准确的接近深度信息缓存,实现了深度范围缓存的实时更新。

[0135] 图8是本申请实施例提供的基于区域的深度测试方法的一个可选的流程示意图七,该方法可以由计算机设备的处理器执行。基于上述任一实施例,以基于图2为例,所述方法还可以包括步骤S801至步骤S802,将结合图8示出的步骤进行说明。

[0136] 步骤S801、在前期深度测试和/或后期深度测试过程中,响应于图元的深度值全部写入所述深度信息缓存,从所述深度信息缓存中读取所述图块对应的每一像素的深度值。

[0137] 其中,该前期深度测试为相关技术中的Early-Ztest,也可以称为Early Depth Test。该前期深度测试可以让深度测试发生在像素着色模块之前,这样没有通过测试的像素就不必再进行Pixel Shader计算,从而提升性能。该后期深度测试为相关技术中位于Pixel Shader之后的深度测试(Depth Test)。

[0138] 在一些实施例中,由于TBR架构以图块为单元进行渲染的,在前期深度测试和/或后期深度测试过程中,响应于图元的深度值全部写入所述深度信息缓存,即在将图元的最后一笔深度值写入到深度信息缓存的情况下,从所述深度信息缓存中读取所述图块对应的每一像素的深度值。

[0139] 步骤S802、基于所述图块对应的每一像素的深度值,对所述深度范围缓存中每一所述初始区域对应的第二深度范围进行更新。

[0140] 在一些实施例中,在得到该图块中每一像素的深度值之后,可以按照该图块中每一初始区域的区域范围,对区域范围内的每一像素的深度值进行排序,得到每一初始区域对应的最大深度值和最小深度值,进而可以将该最大深度值作为初始区域的最大缓存深度值,将该最小深度值作为初始区域的最小缓存深度值;之后,在该深度范围缓存中,将得到的每一初始区域对应的最大缓存深度值和最小缓存深度值,作为每一初始区域对应的第二深度范围。

[0141] 在一些实施例中,由于前期深度测试和/或后期深度测试过程对图块内每一像素的深度值进行深度测试并更新为更小的深度值,因此,上述对第二深度范围进行更新主要是为了更新该第二深度范围中的最大缓存深度值,这样,可以增加第三测试结果命中范围,提升区域剔除率。

[0142] 本申请实施例中,由于在前期深度测试和/或后期深度测试过程中,会进行像素级的深度测试,测试完成之后,会在深度信息缓存中对图块内每一像素的深度值进行更新,此时,利用该深度信息缓存中每一像素的深度值对深度范围缓存中的第二深度范围进行更新,这样,通过不断对该第二深度范围进行更新,可以提高对区域的剔除效率。

[0143] 下面说明本申请实施例提供的基于区域的深度测试方法在实际场景中的应用。

[0144] 本申请以相关技术中的TBR深度测试(Ztest)进行改进,提出一种基于TBR的区域深度测试(Block Z)算法,该算法设计主要在TBR架构上提升前期深度测试(Early-Ztest)的执行效率,并从以下几个方面弥补的传统Ztest算法缺陷:(1)为尽快尽早地将被遮挡物体消除,提升现有Early-Ztest的效率,在像素级Early-Ztest之前增加区域深度测试模块。(2)以区域为单位执行前期深度测试(Early-Ztest),可以实现在光栅化之前就完成遮挡测试,以降低光栅化模块和像素着色模块的负载。(3)充分利用TBR架构下深度信息缓存(Depth Buffer)缓存当前图块的深度信息(Depth value)的优势,以二级深度范围信息读取机制,简化区域深度测试所需的深度范围信息的读写复杂度。

[0145] 在一些实施例中,本申请基于TBR架构,通过在光栅化阶段(光栅化模块)之前增加区域深度测试(Block z)模块,利用现有的深度信息缓存(Depth Buffer)快速执行区域深度测试,从而提升深度测试(Ztest)的整体性能,并在GPU硬件上实施。整体设计思路如下:

[0146] 通过三角形初始化(Triangle setup)模块构建三角形的边和属性的方程;

[0147] 通过图块分割(Tile splitter)模块对图块按照区域(Block)进行划分。其中,根据性能验证的结果,区域的尺寸(size)可以配置为 16×16 , 8×8 和 4×4 三种。图块分割模块用于提取区域信息,主要包括区域坐标,区域大小以及区域对应的三角形的数据(包括边的方程和深度面的方程),并打包送到区域深度测试(Block z)模块;

[0148] 区域深度测试模块以输入的区域信息为基础,一方面依据Z(面)的方程参数确定该区域的最小插值深度值(bmin_z)的位置(min_z_x,min_z_y),和最大插值深度值(bmax_z)的位置(max_z_x,max_z_y),然后通过内部的网络算术逻辑单元(Arithmetic Logic Unit,ALU)插值出该区域的最小插值深度值和该区域的最大插值深度值;另一方面会向深度排序(Zsorter)模块请求区域在深度信息缓存(Depth Buffer)中对应最小缓存深度值(rmin_z)和最大缓存深度值(rmax_z)。之后,区域深度测试的比较单元以“最小插值深度值、最大插值深度值”和“最小缓存深度值、最大缓存深度值”作为输入,根据当前drawZtest的比较模式执行Block Ztest,比较模式可以包括以下任意之一:LESS模式、GREATER模式;

区域深度测试过程的测试结果会输出给光栅化模块以及HSR模块,HSR模块包括像素级前期深度测试(Early-Ztest)和后期深度测试(late Ztest)。HSR模块会根据深度测试过程的测试结果执行像素级Ztest和深度信息缓存的更新。

[0149] 在一些实施例中,本申请通过深度范围缓存(Zrange Buffer)存储图块的所有区域的深度范围信息Zrange(包括最小缓存深度值,最大缓存深度值),深度范围缓存内存储的区域信息的数量和深度信息缓存内存储的区域信息的数量相同,也就是说,同一个区域的深度范围信息一个深度范围缓存和一个深度信息缓存相绑定。该深度范围缓存会被区域深度测试模块和深度排序模块写入数据。其中深度排序模块向深度范围缓存中写入初始的深度范围信息,而区域深度测试模块在图像渲染(draw)的过程中不断更新深度范围缓存中的深度范围信息。

[0150] 在一些实施例中,深度排序模块可以为深度范围缓存写入初始值。深度排序模块会首先为区域的所有像素(sample)向深度信息缓存中加载深度值(Depth value),然后从这些深度值中比较得到最小值和最大值,即(最小缓存深度值,最大缓存深度值),将其写入到深度范围缓存中对应位置。在另一些实施例中,在前期深度测试模块或后期深度测试模块对图元的像素完成ztest处理,并写入深度信息缓存之后,深度排序模块会基于深度信息缓存内的最新深度信息主动更新Zrange,以提高Zrange数值的实时性。Block Z会为新的Block向Zsorter发起Zrange请求。

[0151] 请参阅图9,其示出了一种基于区域的深度测试方法的整体架构示意图。

[0152] 在一些实施例中,针对每一个图块,三角形初始化模块901可以为该图块中的每一个图元(primitive)计算图元信息。以图元为三角形为例,图元信息可以成为三角形数据,该三角形数据包括三角形的边的方程、深度面方程和三角形标识框的范围信息。

[0153] 示例性的,三角形的边的方程可以以公式(1)至公式(3)体现。

[0154] EDGE0: $A0*x+B0*y+C0=0$ 公式(1);

[0155] EDGE1: $A1*x+B1*y+C1=0$ 公式(2);

[0156] EDGE2: $A2*x+B2*y+C2=0$ 公式(3);

[0157] 上述公式(1)至公式(3)可以用于光栅化过程。

[0158] 示例性的,Z的深度面方程可以以公式(4)体现。

[0159] ZFace: $ZA*x+ZB*y+ZC =0$ 公式(4);

[0160] 其中,ZA可以认为是三角形深度(Z)面在x方向上的斜率,ZB可以认为是三角形深度面在y方向上的斜率。

[0161] 示例性的,三角形标识框的范围信息可以通过BBoxMinX,BBoxMaxX,BBoxMinY,BBoxMaxY体现,即,三角形标识框在X方向上的范围和在Y方向上的范围。

[0162] 在一些实施例中,针对每一个图块,图块分割模块902可以以区域为粒度对图块进行拆分。请参阅图10,其示出了一种区域分割过程示意图。在图10中,图块1010的尺寸为 $32*32$,在区域1020尺寸为 $8*8$ 的情况下,可以将该图块1010分割为B0至B15共16个 $8*8$ 的区域,结合上述三角形标识框的范围信息1030,图块分割模块902最终可以得到12个 $8*8$ 的Block,如图中的B1、B2、B3、B5、B6、B7、B9、B10、B11、B13、B14、B15。

[0163] 在一些实施例中,图块分割模块902会将预设数量个区域的区域信息和对应的图元信息打包发送至区域深度测试模块903。该预设数量与区域深度测试模块903中的ALU数

量相关。示例性的,以区域深度测试模块903中包括4个ALU为例,图块分割模块902会依次将4个区域的区域信息和对应的图元信息打包发送至区域深度测试模块903。

[0164] 请参阅图11,其示出了一种区域深度测试模块903的结构示意图。

[0165] 区域深度测试模块903在接收到4个区域的信息后,由深度范围抓取(Zrange prefetcher)单元1101根据各区域在图块中的相对位置(包括Offset_x和Offset_y),确定各区域在深度范围缓存(Zrange Buffer)中的索引信息(Block index),通过该索引信息可以在深度范围缓存中抓取各区域的深度范围信息。在一些实施例中,深度范围缓存会根据索引信息在深度范围缓存中进行hit-miss test测试,目的是检测区域对应的深度范围信息是否已经存在于深度范围缓存,在区域对应的深度范围信息已经存在于深度范围缓存的情况下,直接取出返回给该区域对应的比较(Tester)单元1103。在区域对应的深度范围信息不存在于深度范围缓存的情况下,那么深度范围抓取单元1101需要通知深度排序模块去深度信息缓存(Depth)中取该区域对应的深度范围信息,然后再写入到深度范围缓存中。

[0166] 如图11所示,区域深度测试模块903内部包括4个独立的网络算术逻辑单元1102和比较单元1103。

[0167] 其中,网络算术逻辑单元用于计算最小插值深度值的坐标信息和最大插值深度值的坐标信息,并通过插值确定最小插值深度值(bmin_z)和最大插值深度值(bmax_z)。比较单元用于进行深度测试Ztest和深度范围缓存的更新。

[0168] 其中,针对一个区域,它的深度值理应是一个范围,由于Z面是线性变化的,所以区域最小深度值和区域最大深度值必然处于区域边界的4个点上,由此,区域最小深度值和区域最大深度值与Z面的参数ZA、ZB之间的关系如表1:

[0169] 表1

ZA	ZB	区域最小深度值	区域最大深度值
≥ 0	≥ 0	上-左顶点	下-右顶点
< 0	< 0	下-右顶点	上-左顶点
≥ 0	< 0	下-左顶点	上-右顶点
< 0	≥ 0	上-右顶点	下-左顶点

[0171] 结合上表和区域的四个顶点的坐标信息,即可确定最小插值深度值的坐标信息和最大插值深度值的坐标信息。在得到最小插值深度值的坐标信息和最大插值深度值的坐标信息之后,区域的最小插值深度值和最大插值深度值便可以由Z面的方程确定,如公式(5)和公式(6):

[0172] $bmin_z = ZA * bmin_z.Offset_x + ZB * bmin_z.Offset_y + ZC$ 公式(5);

[0173] $bmax_z = ZA * bmax_z.Offset_x + ZB * bmax_z.Offset_y + ZC$ 公式(6);

[0174] 其中,bmin_z为最小插值深度值,bmax_z为最大插值深度值。

[0175] 比较单元用于对基于插值得到的“最小插值深度值和最大插值深度值”和从深度范围缓存中抓取的“最小缓存深度值和最大缓存深度值”进行比较,以完成该区域的深度测试过程。深度测试过程共有三种结果,以下将分别进行说明。

[0176] 在一些实施例中,在最大插值深度值小于或等于最小缓存深度值($bmax_z \leq rmin_z$)的情况下,该深度测试过程得到第一测试结果(可以设置为PASS)。第一测试结果的区域

将进入到像素级的前期深度测试(Early-Ztest),但可以避免做Ztest,只做Depth Buffer的更新。如图12A示出的深度范围比对示意图。

[0177] 在一些实施例中,在最小插值深度值小于或等于最小缓存深度值($bmin_z \leq rmin_z$),且最小缓存深度值小于或等于最大插值深度值($rmin_z \leq bmax_z$)的情况下,或,在最小插值深度值小于或等于最大缓存深度值($bmin_z \leq rmin_z$),且最大缓存深度值小于或等于最大插值深度值($rmin_z \leq bmax_z$)的情况下,该深度测试过程得到第二测试结果(可以设置为UNDONE)。第二测试结果的区域执行像素级的前期深度测试(Early-Ztest)。如图12B和图12C示出的深度范围比对示意图。

[0178] 在一些实施例中,在最大缓存深度值小于或等于最小插值深度值($rmax_z \leq bmin_z$)的情况下,该深度测试过程得到第三测试结果(可以设置为FAIL)。第三测试结果的区域被立即剔除,不会进入到像素级的前期深度测试(Early-Ztest)。如图12D示出的深度范围比对示意图。

[0179] 在一些实施例中,区域深度测试过程的测试结果会输出给光栅化模块905、前期深度测试模块906和后期深度测试模块907。

[0180] 在一些实施例中,深度范围缓存可以被上述比较单元和深度排序模块更新,如下述实施例,深度范围缓存会在以下情况中被更新。

[0181] 第一种情况为,比较单元根据深度测试过程得到测试结果更新深度范围信息的最小缓存深度值($rmin_z$),即,针对测试结果为第一测试结果和第二测试结果的区域,将该区域的最小插值深度值直接更新至最小缓存深度值,这样,可以使得到第一测试结果的区域插值深度值不断缩小,提高得到第一测试结果的准确度。

[0182] 第二种情况为,深度信息缓存被分配给当前的图块并且被初始化之后,深度排序模块会向深度信息缓存读取区域中所有像素(pixel)的深度值,然后对所有的深度值做排序获取最小缓存深度值和最大缓存深度值,然后将其更新到深度范围缓存中。

[0183] 第三种情况为,为进一步增强深度测试的效率,在深度排序模块中增加主动更新深度范围缓存的方法。这种方式要在前期深度测试(Early-Ztest)或者后期深度测试(later Ztest)中,将图元的最后一笔深度值写入到深度信息缓存后,立即通知深度排序模块从深度信息缓存读取这个图块的所有像素的深度值,做完排序后向深度范围缓存更新。这一步其实是不断更新最大缓存深度值,进而提高区域剔除效率。

[0184] 基于上述实施例,本申请以前期深度测试(Early-Ztest)为主题,通过在光栅化模块904之前增加区域深度测试模块903,一方面完实现了深度测试过程,提升了前期深度测试(Early-Ztest)的效率,降低了光栅化模块904和像素着色模块的负载,另一方面通过两种深度范围缓存的更新机制,使深度范围缓存更加准确地接近深度信息缓存,基本实现了深度范围缓存的即时更新,提升了深度测试的效率,从而整体上增强了GPU的渲染性能。

[0185] 本申请实施例通过基于分块渲染架构下的前期深度测试的实现方案,可以充分利用TBR架构的片载深度信息缓存资源,实现区域深度测试逻辑,可快速提早地剔除掉被遮挡物体,不必进入光栅化阶段,节省了计算资源;同时,通过更高效的深度范围缓存的更新方法,既保证了低精度(coarse)深度测试的准确进行,又保证了深度范围缓存一定的实时性,提高区域剔除的效率;同时,本申请提出的基于区域的深度测试方法,适用于TBR的架构,有效降低像素着色模块的工作负载,提升现有光栅化管线的性能。

[0186] 基于前述的实施例,本申请实施例提供一种基于区域的深度测试装置,该装置包括所包括的各单元、以及各单元所包括的各模块,可以通过计算机设备中的处理器来实现;当然也可通过具体的逻辑电路实现。

[0187] 图13为本申请实施例提供的一种基于区域的深度测试装置的组成结构示意图,如图13所示,基于区域的深度测试装置1300包括:图块分割模块1310、区域深度测试模块1320,其中:

[0188] 图块分割模块1310,用于在执行图块的光栅化过程之前,对所述图块进行分割,得到多个待测试区域和每一所述待测试区域对应的区域数据;所述区域数据包括所述待测试区域的区域信息和所述待测试区域对应的图元信息;

[0189] 区域深度测试模块1320,用于针对每一所述待测试区域,基于所述待测试区域的区域信息和所述待测试区域对应的图元信息,生成所述待测试区域的第一深度范围;

[0190] 所述区域深度测试模块1320,还用于从深度范围缓存中获取所述待测试区域的第二深度范围;

[0191] 所述区域深度测试模块1320,还用于基于所述第一深度范围和所述第二深度范围,确定所述待测试区域的深度测试结果;所述深度测试结果用于指示所述待测试区域是否需要进行像素级深度测试。

[0192] 在一些实施例中,所述图元信息包括图元的深度面方程;所述区域深度测试模块1320,还用于:基于所述图元的深度面方程和所述待测试区域的区域信息,确定所述待测试区域中的最小插值深度值和最大插值深度值;基于所述最小插值深度值和所述最大插值深度值,生成所述待测试区域的第一深度范围。

[0193] 在一些实施例中,所述区域深度测试模块1320,还用于:基于所述图元的深度面方程,确定第一斜率和第二斜率;所述第一斜率表征所述图元的深度面在第一方向上的斜率,所述第二斜率为所述图元的深度面在第二方向上的斜率,所述第一方向、所述第二方向与深度方向相互垂直;基于所述第一斜率、所述第二斜率和所述区域信息,确定最小插值深度值的坐标信息和最大插值深度值的坐标信息;基于所述最小插值深度值的坐标信息、所述最大插值深度值的坐标信息和所述深度面方程,确定最小插值深度值和最大插值深度值。

[0194] 在一些实施例中,所述待测试区域由第一区域顶点、第二区域顶点、第三区域顶点和第四区域顶点确定,所述区域信息包括第一区域顶点的顶点坐标、第二区域顶点的顶点坐标、第三区域顶点的顶点坐标和第四区域顶点的顶点坐标;所述区域深度测试模块1320,还用于:在所述第一斜率大于或等于0、且所述第二斜率大于或等于0的情况下,将所述第一区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第三区域顶点的顶点坐标作为最大插值深度值的坐标信息;在所述第一斜率小于0、且所述第二斜率小于0的情况下,将所述第三区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第一区域顶点的顶点坐标作为最大插值深度值的坐标信息;在所述第一斜率大于或等于0、且所述第二斜率小于0的情况下,将所述第二区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第四区域顶点的顶点坐标作为最大插值深度值的坐标信息;在所述第一斜率小于0、且所述第二斜率大于或等于0的情况下,将所述第四区域顶点的顶点坐标作为最小插值深度值的坐标信息,并将所述第二区域顶点的顶点坐标作为最大插值深度值的坐标信息。

[0195] 在一些实施例中,所述区域深度测试模块1320,还用于:获取所述待测试区域在所

述图块中的区域位置信息;基于所述待测试区域的区域位置信息生成所述待测试区域的索引信息;基于所述待测试区域的索引信息在所述深度范围缓存中获取所述待测试区域的第二深度范围。

[0196] 在一些实施例中,所述区域深度测试模块1320,还用于:基于所述待测试区域的索引信息,在所述深度范围缓存中进行命中测试;在所述命中测试的命中测试结果表征所述深度范围缓存中存在所述待测试区域的第二深度范围的情况下,获取所述待测试区域的第二深度范围;在所述命中测试的命中测试结果表征所述深度范围缓存中不存在所述待测试区域的第二深度范围的情况下,从深度信息缓存中确定所述待测试区域的第二深度范围并写入所述深度范围缓存。

[0197] 在一些实施例中,所述图块分割模块1310,还用于:基于预设区域尺寸对所述图块进行分割,得到多个初始区域和每一所述初始区域对应的区域信息;获取所述图块对应的图元的图元信息;所述图元信息包括所述图元的边界框信息;基于所述图元的边界框信息,对所述多个初始区域进行筛选,得到所述图块对应的多个待测试区域和每一所述待测试区域的区域数据。

[0198] 在一些实施例中,所述区域深度测试模块1320,还用于:在所述第一深度范围的最大值小于所述第二深度范围的最小值的情况下,生成第一测试结果;所述第一测试结果用于表征所述待测试区域不需要进行像素级深度测试,并更新深度信息缓存中所述待测试区域中各像素的深度值;在所述第一深度范围和所述第二深度范围存在交集的情况下,生成第二测试结果;所述第二测试结果用于表征所述待测试区域需要进行像素级深度测试;在所述第一深度范围的最小值大于所述第二深度范围的最大值的情况下,生成第三测试结果;所述第三测试结果用于表征所述待测试区域需要被剔除且不需要进行像素级深度测试。

[0199] 在一些实施例中,所述区域深度测试模块1320,还用于:响应于所述待测试区域的深度测试结果为所述第一测试结果或所述第二测试结果,利用所述第一深度范围中的最小插值深度值,在所述深度范围缓存中更新所述待测试区域的第二深度范围中的最小缓存深度值。

[0200] 在一些实施例中,基于区域的深度测试装置还包括深度排序模块,所述深度排序模块,用于:响应于深度信息缓存被分配给所述图块并完成初始化,从所述深度信息缓存中获取每一所述初始区域中每一像素的深度值;基于每一所述初始区域中每一像素的深度值,确定每一所述初始区域对应的第二深度范围;将每一所述初始区域对应的第二深度范围更新至所述深度范围缓存。

[0201] 在一些实施例中,所述深度排序模块,还用于:在前期深度测试和/或后期深度测试过程中,响应于图元的深度值全部写入所述深度信息缓存,从所述深度信息缓存中读取所述图块对应的每一像素的深度值;基于所述图块对应的每一像素的深度值,对所述深度范围缓存中每一所述初始区域对应的第二深度范围进行更新。

[0202] 以上装置实施例的描述,与上述方法实施例的描述是类似的,具有同方法实施例相似的有益效果。在一些实施例中,本申请实施例提供的装置具有的功能或包含的模块可以用于执行上述方法实施例描述的方法,对于本申请装置实施例中未披露的技术细节,请参照本申请方法实施例的描述而理解。

[0203] 需要说明的是,本申请实施例中,如果以软件功能模块的形式实现上述的基于区域的深度测试方法,并作为独立的产品销售或使用,也可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请实施例的技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来,该软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机、服务器、或者网络设备等)执行本申请各个实施例所述方法的全部或部分。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read Only Memory,ROM)、磁碟或者光盘等各种可以存储程序代码的介质。这样,本申请实施例不限制于任何特定的硬件、软件或固件,或者硬件、软件、固件三者之间的任意结合。

[0204] 本申请实施例提供一种计算机设备,包括存储器和处理器,所述存储器存储有可在处理器上运行的计算机程序,所述处理器执行所述程序时实现上述方法中的部分或全部步骤。

[0205] 本申请实施例提供一种计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现上述方法中的部分或全部步骤。所述计算机可读存储介质可以是瞬时性的,也可以是非瞬时性的。

[0206] 本申请实施例提供一种计算机程序,包括计算机可读代码,在所述计算机可读代码在计算机设备中运行的情况下,所述计算机设备中的处理器执行用于实现上述方法中的部分或全部步骤。

[0207] 本申请实施例提供一种计算机程序产品,所述计算机程序产品包括存储了计算机程序的非瞬时性计算机可读存储介质,所述计算机程序被计算机读取并执行时,实现上述方法中的部分或全部步骤。该计算机程序产品可以具体通过硬件、软件或其结合的方式实现。在一些实施例中,所述计算机程序产品具体体现为计算机存储介质,在另一些实施例中,计算机程序产品具体体现为软件产品,例如软件开发包(Software Development Kit, SDK)等等。

[0208] 这里需要指出的是:上文对各个实施例的描述倾向于强调各个实施例之间的不同之处,其相同或相似之处可以互相参考。以上设备、存储介质、计算机程序及计算机程序产品实施例的描述,与上述方法实施例的描述是类似的,具有同方法实施例相似的有益效果。对于本申请设备、存储介质、计算机程序及计算机程序产品实施例中未披露的技术细节,请参照本申请方法实施例的描述而理解。

[0209] 图14为本申请实施例提供的一种计算机设备的硬件实体示意图,如图14所示,该计算机设备1400的硬件实体包括:处理器1401和存储器1402,其中,存储器1402存储有可在处理器1401上运行的计算机程序,处理器1401执行程序时实现上述任一实施例的方法中的步骤。

[0210] 存储器1402存储有可在处理器上运行的计算机程序,存储器1402配置为存储由处理器1401可执行的指令和应用,还可以缓存待处理器1401以及计算机设备1400中各模块待处理或已经处理的数据(例如,图像数据、音频数据、语音通信数据和视频通信数据),可以通过闪存(FLASH)或随机访问存储器(Random Access Memory,RAM)实现。

[0211] 处理器1401执行程序时实现上述任一项的基于区域的深度测试方法的步骤。处理器1401通常控制计算机设备1400的总体操作。

[0212] 本申请实施例提供一种计算机存储介质,计算机存储介质存储有一个或者多个程

序,该一个或者多个程序可被一个或者多个处理器执行,以实现如上任一实施例的基于区域的深度测试方法的步骤。

[0213] 这里需要指出的是:以上存储介质和设备实施例的描述,与上述方法实施例的描述是类似的,具有同方法实施例相似的有益效果。对于本申请存储介质和设备实施例中未披露的技术细节,请参照本申请方法实施例的描述而理解。

[0214] 上述处理器可以为目标用途集成电路(Application Specific Integrated Circuit,ASIC)、数字信号处理器(Digital Signal Processor,DSP)、数字信号处理装置(Digital Signal Processing Device,DSPD)、可编程逻辑装置(Programmable Logic Device,PLD)、现场可编程门阵列(Field Programmable Gate Array,FPGA)、中央处理器(Central Processing Unit,CPU)、控制器、微控制器、微处理器中的至少一种。可以理解地,实现上述处理器功能的电子器件还可以为其它,本申请实施例不作具体限定。

[0215] 上述计算机存储介质/存储器可以是只读存储器(Read Only Memory,ROM)、可编程只读存储器(Programmable Read-Only Memory,PROM)、可擦除可编程只读存储器(Erasable Programmable Read-Only Memory,EPRM)、电可擦除可编程只读存储器(Electrically Erasable Programmable Read-Only Memory,EEPROM)、磁性随机存取存储器(Ferromagnetic Random Access Memory,FRAM)、快闪存储器(Flash Memory)、磁表面存储器、光盘、或只读光盘(Compact Disc Read-Only Memory,CD-ROM)等存储器;也可以是包括上述存储器之一或任意组合的各种终端,如移动电话、计算机、平板设备、个人数字助理等。

[0216] 应理解,说明书通篇中提到的“一个实施例”或“一实施例”意味着与实施例有关的特定特征、结构或特性包括在本申请的至少一个实施例中。因此,在整个说明书各处出现的“在一个实施例中”或“在一实施例中”未必一定指相同的实施例。此外,这些特定的特征、结构或特性可以任意适合的方式结合在一个或多个实施例中。应理解,在本申请的各种实施例中,上述各步骤/过程的序号的大小并不意味着执行顺序的先后,各步骤/过程的执行顺序应以其功能和内在逻辑确定,而不对本申请实施例的实施过程构成任何限定。上述本申请实施例序号仅仅为了描述,不代表实施例的优劣。

[0217] 需要说明的是,在本文中,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0218] 在本申请所提供的几个实施例中,应该理解到,所揭露的设备和方法,可以通过其它的方式实现。以上所描述的设备实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,如:多个单元或组件可以结合,或可以集成到另一个系统,或一些特征可以忽略,或不执行。另外,所显示或讨论的各组成部分相互之间的耦合、或直接耦合、或通信连接可以是通过一些接口,设备或单元的间接耦合或通信连接,可以是电性的、机械的或其它形式的。

[0219] 上述作为分离部件说明的单元可以是、或也可以不是物理上分开的,作为单元显示的部件可以是、或也可以不是物理单元;既可以位于一个地方,也可以分布到多个网络单

元上;可以根据实际的需要选择其中的部分或全部单元来实现本实施例方案的目的。

[0220] 另外,在本申请各实施例中的各功能单元可以全部集成在一个处理单元中,也可以是各单元分别单独作为一个单元,也可以两个或两个以上单元集成在一个单元中;上述集成的单元既可以采用硬件的形式实现,也可以采用硬件加软件功能单元的形式实现。本领域普通技术人员可以理解:实现上述方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成,前述的程序可以存储于计算机可读取存储介质中,该程序在执行时,执行包括上述方法实施例的步骤;而前述的存储介质包括:移动存储设备、只读存储器(Read Only Memory,ROM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0221] 或者,本申请上述集成的单元如果以软件功能模块的形式实现并作为独立的产品销售或使用时,也可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机、服务器、或者网络设备等)执行本申请各个实施例所述方法的全部或部分。而前述的存储介质包括:移动存储设备、ROM、磁碟或者光盘等各种可以存储程序代码的介质。

[0222] 以上所述,仅为本申请的实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。

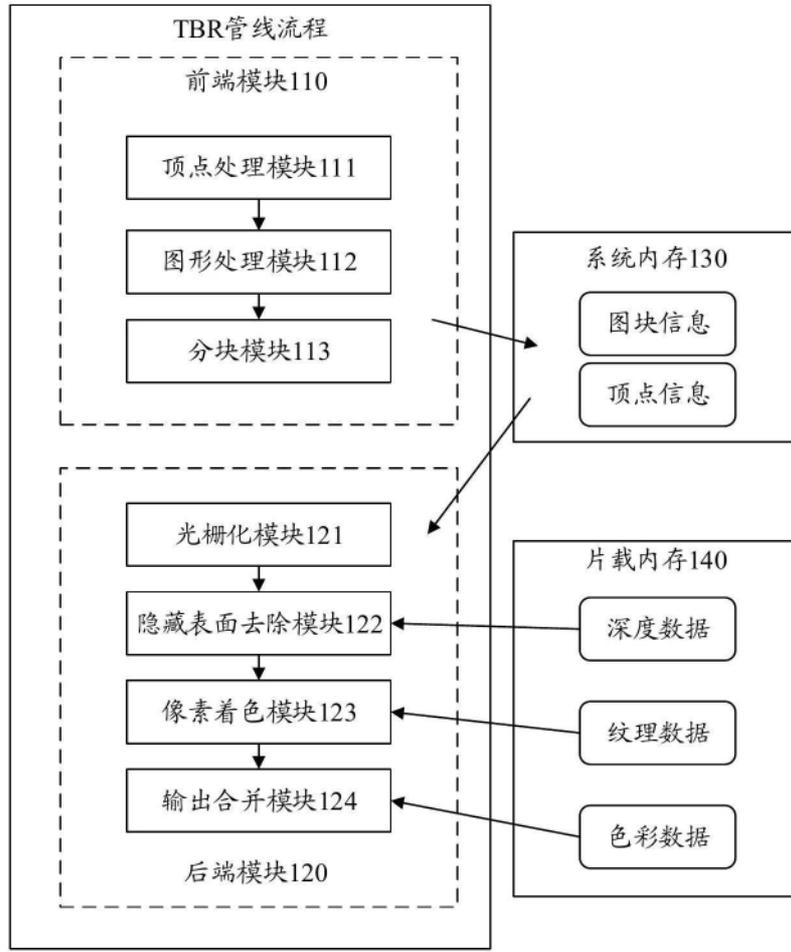


图 1

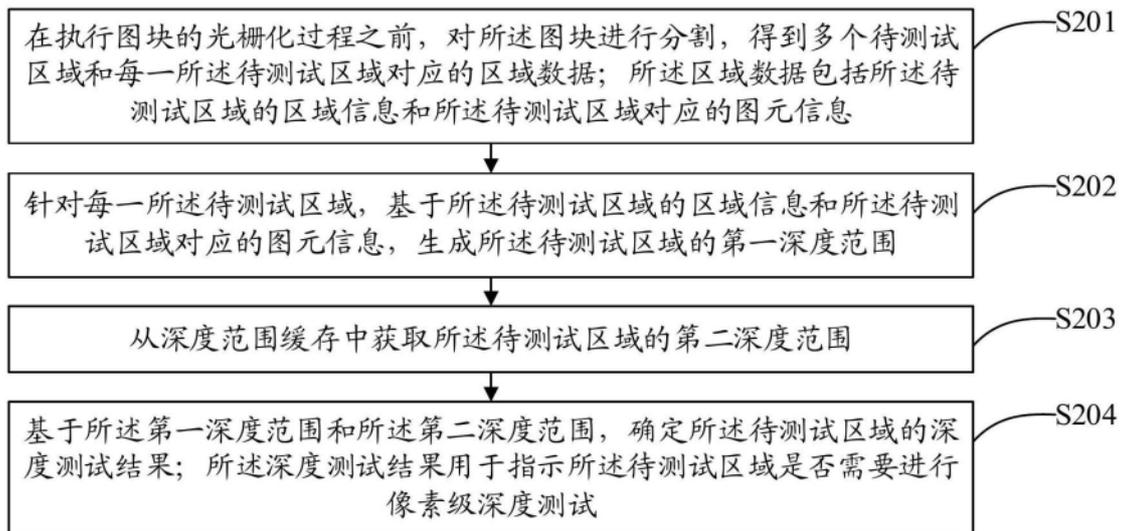


图 2

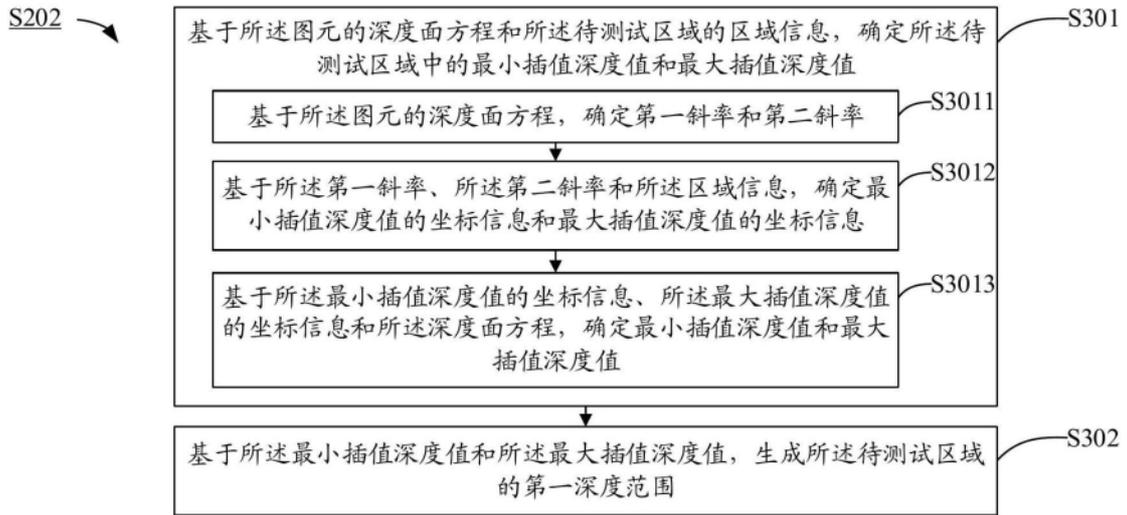


图 3

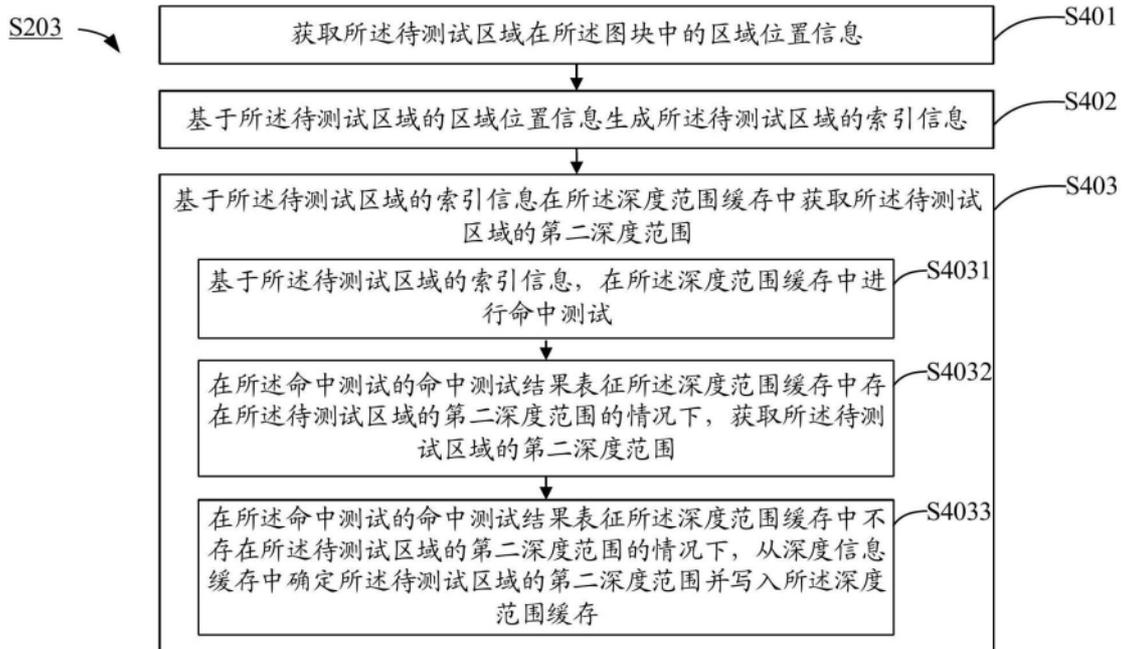


图 4

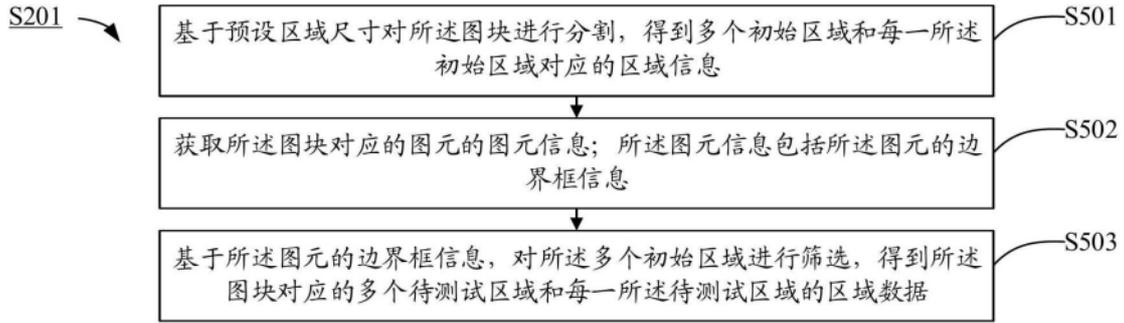


图 5

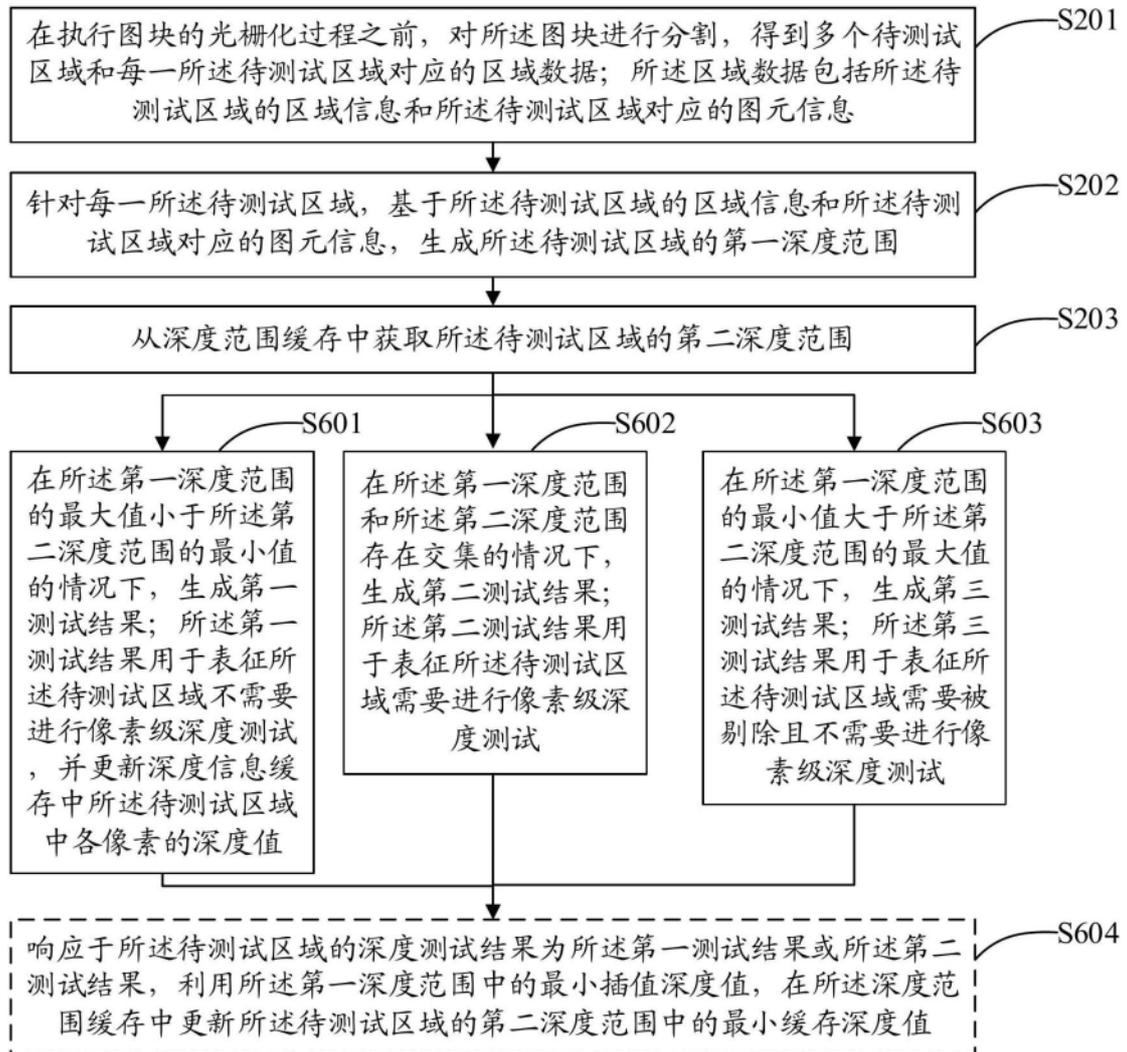


图 6

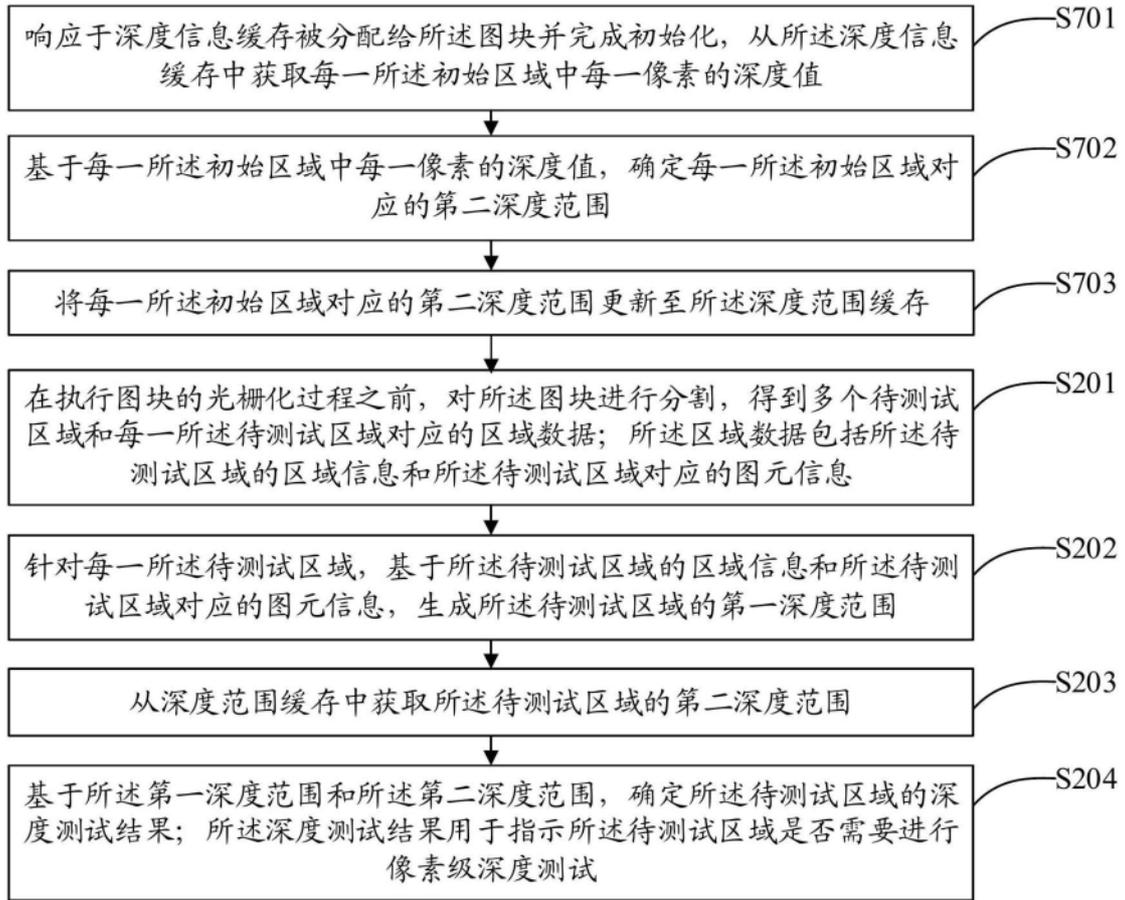


图 7

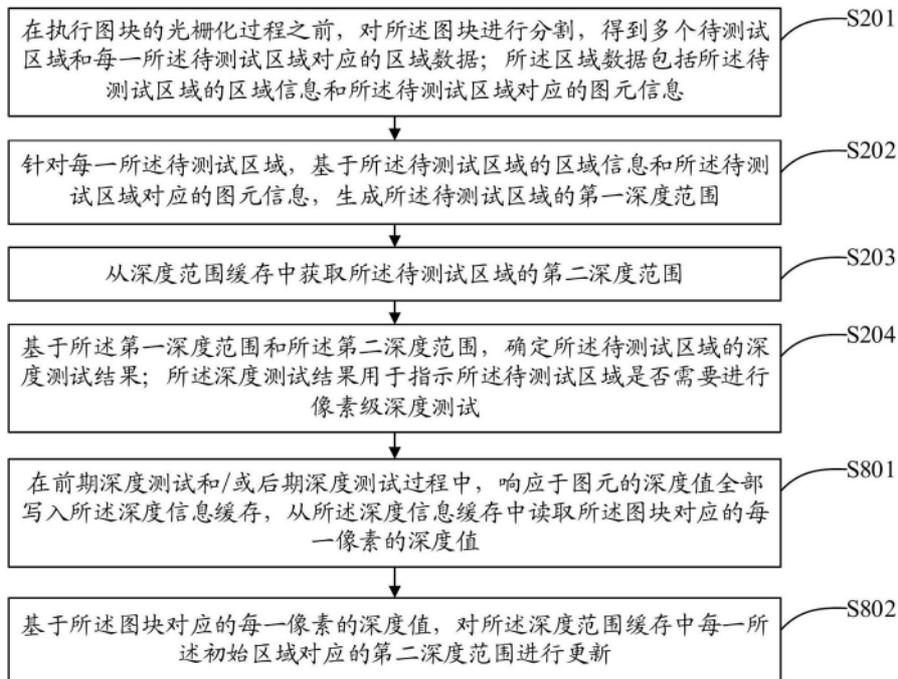


图 8

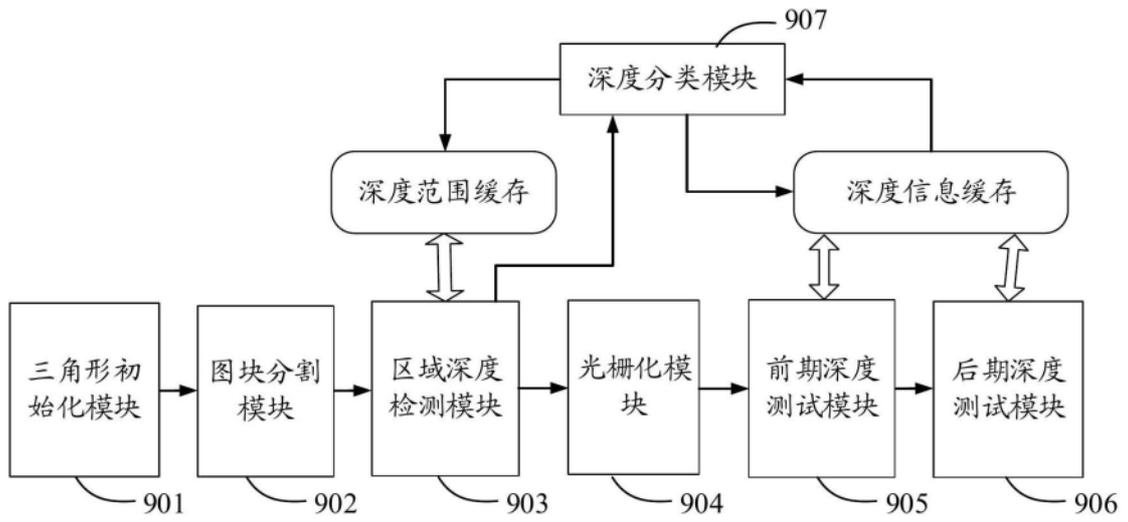


图 9

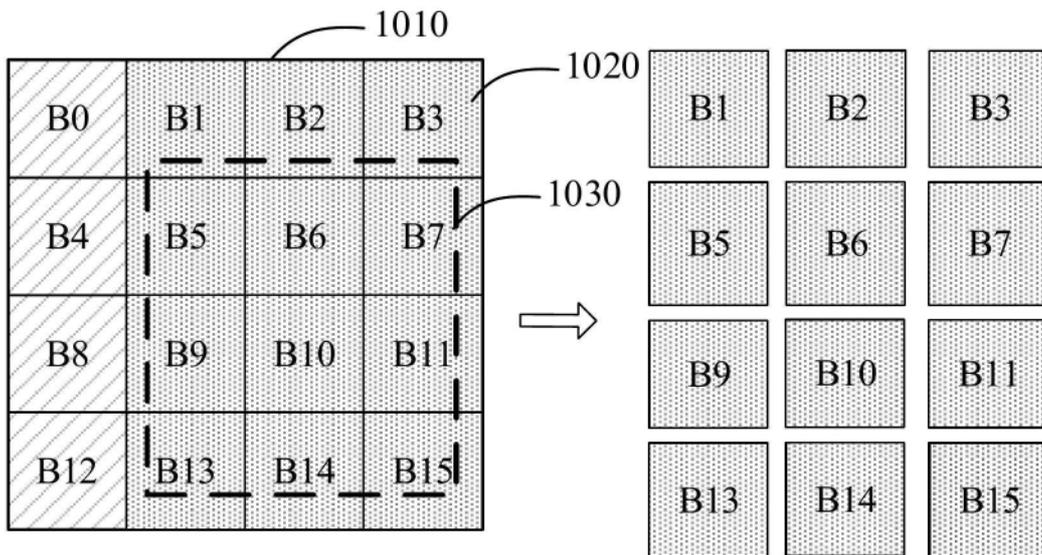


图 10

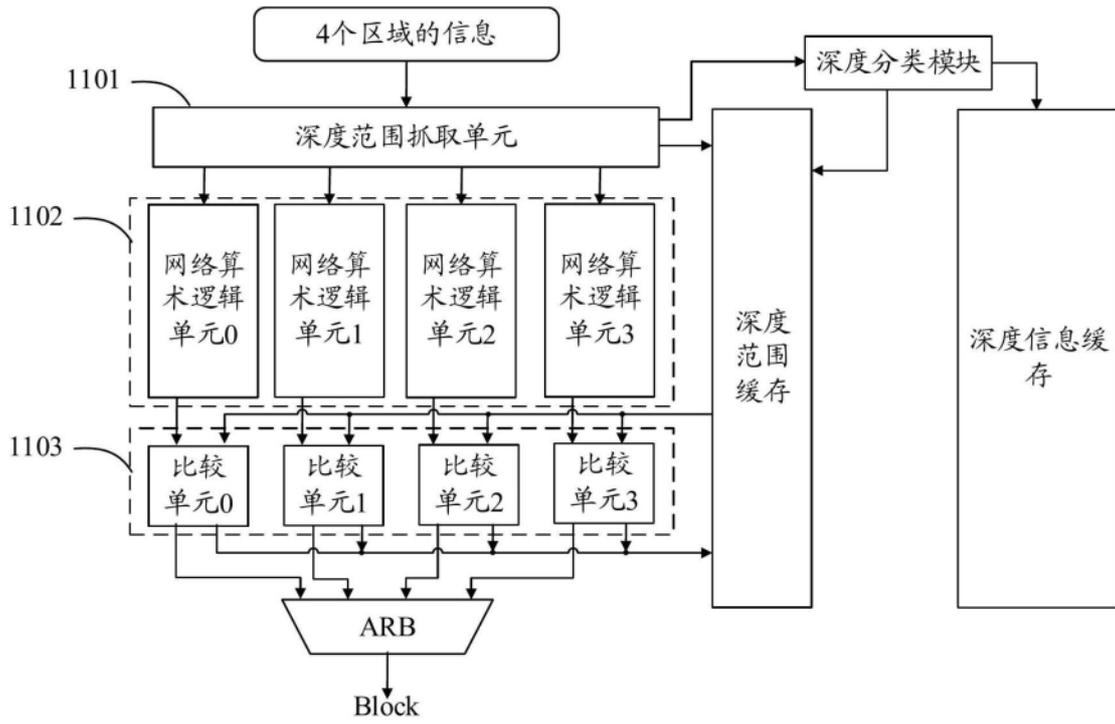


图 11

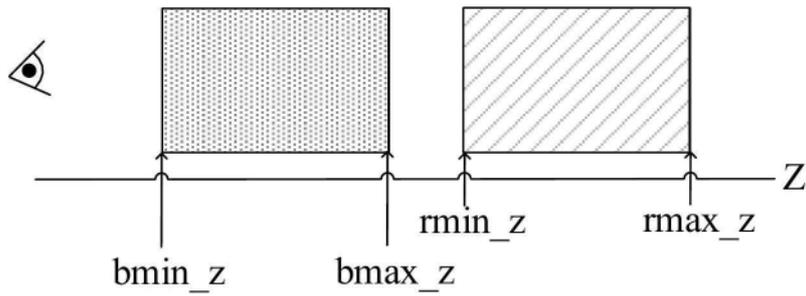


图 12A

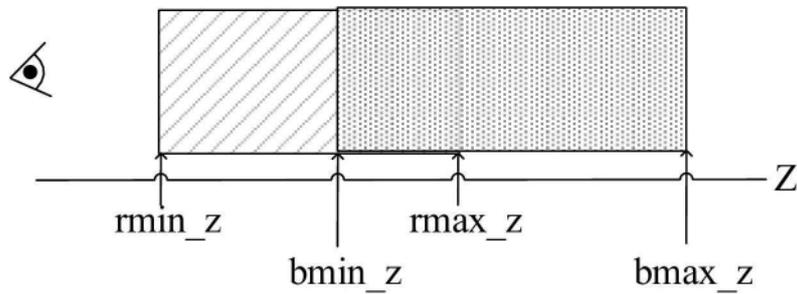


图 12B

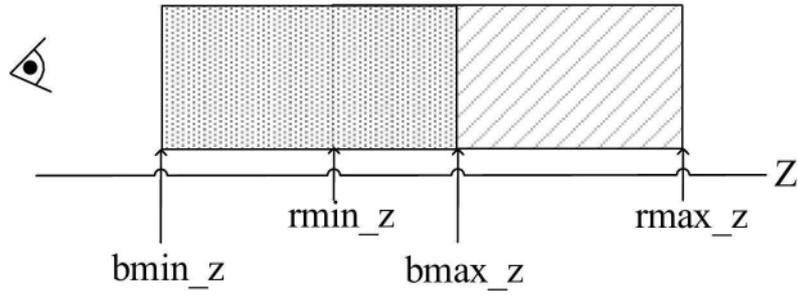


图 12C

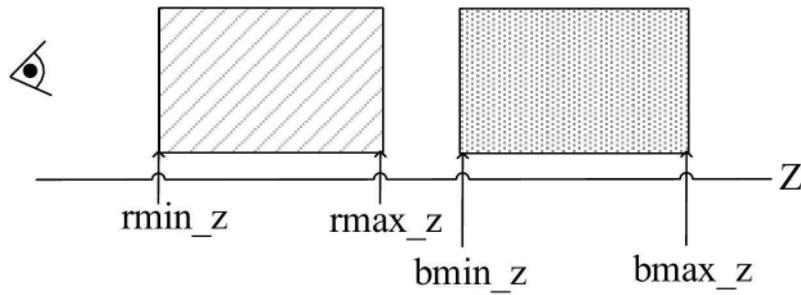


图 12D

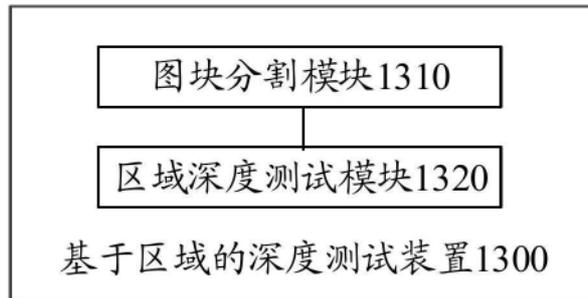


图 13

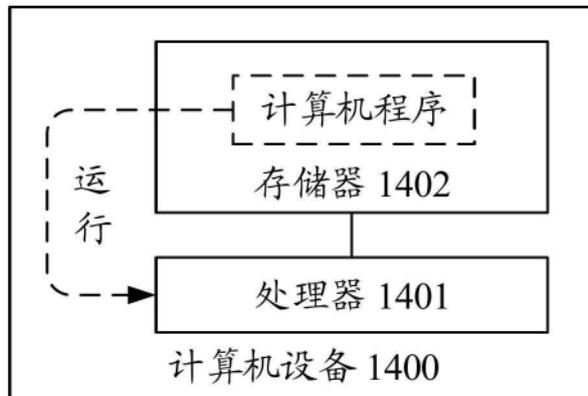


图 14