

图形处理器和图形绘制方法

5 本申请要求于 2023 年 2 月 27 日提交国家知识产权局、申请号为 202310231313.7、申请名称为“图形处理器和图形绘制方法”的中国专利申请的优先权,其全部内容通过引用结合在本申请中。
技术领域

本申请实施例涉及芯片技术领域,尤其涉及一种图形处理器和图形绘制方法。

背景技术

10 目前,图形处理器(graphics processing unit, GPU)处理的三维场景由多个物体构成,其中,每个物体又由多个基本图元(例如三角形、线和点等)构成。在处理绘制任务时,物体按照应用程序指定的顺序发送给图形处理器,即每个物体的基本图元按照在物体中的顺序发送至图形处理器,由图形处理器进行一系列处理,以得到最终的像素颜色。其中,片元着色程序通常执行最多的工作负荷。为了减少片元着色程序的工作开销,图形处理器会进行多种算法处理,例如利用物体的深度信息进行深度剔除,以减少片元着色程序任务的数量。

15 随着图形处理器应用场景的不断扩大,除了三维应用程序外,二维用户界面(user interface, UI)应用也大量使用图形处理器进行绘制。在二维用户界面绘制中,可能没有深度信息,且对于存在特殊操作的场景无法执行剔除算法。由此图形处理器仍存在过度绘制问题,造成着色处理器的工作负荷较高。

发明内容

20 本申请实施例提供一种图形处理器和图形绘制方法,解决了现有的图形处理器执行绘制任务时过度绘制的问题。

为达到上述目的,本申请实施例采用如下技术方案。

25 第一方面,本申请实施例提供一种图形处理器,图形处理器包括剔除模块和着色处理器,剔除模块包括寄存器和过度绘制剔除模块,过度绘制剔除模块的输出端和着色处理器的输入端耦合;寄存器,用于存储指示信息,指示信息用于指示绘制任务的图形区域中参与过度绘制剔除操作的子区域;过度绘制剔除模块,用于记录绘制任务的多个片元在图形区域中的位置信息和参与绘制时的顺序信息,并基于位置信息和顺序信息对子区域内的片元进行过度绘制剔除操作,将多个片元中需要绘制的片元发送至着色处理器;着色处理器,用于对多个片元中需要绘制的片元进行着色处理。其中,参与过度绘制剔除操作的子区域可以是进行混合操作、模板比较操作和指令丢弃操作的区域。

30 相较于现有技术中,对于存在混合操作、模板比较操作和指令丢弃操作的场景,图形处理器中的着色处理器无法执行剔除算法的问题,本申请实施例中的图形处理器先通过寄存器存储的指示信息确定出参与过度绘制剔除操作的子区域,然后在着色处理器对片元进行着色处理之前,由着色处理器之外的过度绘制剔除模块对参与过度绘制剔除操作的子区域进行过度绘制剔除操作,35 即使参与过度绘制剔除操作的子区域中存在无深度信息的二维场景中的片元,也可对无深度信息的二维场景中的片元也执行过度绘制剔除操作。这样,不仅可避免着色处理器对于存在混合操作、模板比较操作和指令丢弃操作的场景无法执行剔除算法的问题,也可在剔除子区域中不需要绘制的片元的基础上,将需要绘制的片元发送至着色处理器,可以有效减少过度绘制情况,减少着色处理器的负载,提高图形处理器性能,降低功耗。

40 在一种可能的设计中,过度绘制剔除操作为剔除多个片元中位置信息相同且顺序信息为在前顺序的片元。

这种设计中,当位置信息相同时,片元中在后顺序的片元能够覆盖在前顺序的片元,将在前顺序的片元剔除,可以减少片元的指令开销,达到减少过度绘制的目的。

45 在一种可能的设计中,过度绘制剔除模块还用于基于位置信息和顺序信息将子区域外的片元发送至着色处理器。

这种设计中,对于子区域外的片元,过度绘制剔除模块不对片元进行处理,仅将片元透传至着色处理器,由此,可以减少片元的处理过程,提高图形处理器的效率。

在一种可能的设计中，指示信息包括子区域的区域范围和第一标识，第一标识用于指示子区域的区域范围内部为参与过度绘制剔除操作的区域范围。

这种设计中，子区域的区域范围可以用数学公式描述的封闭曲线，也可以通过输入的图像或纹理指定。通过第一标识，可以灵活设置参与过度绘制剔除操作的区域范围，适用于多种绘制场景。

在一种可能的设计中，指示信息包括子区域的区域范围和第二标识，第二标识用于指示子区域的区域范围外部为参与过度绘制剔除操作的区域范围。

这种设计中，子区域的区域范围可以用数学公式描述的封闭曲线，也可以通过输入的图像或纹理指定。通过第二标识，可以灵活设置参与过度绘制剔除操作的区域范围，适用于多种绘制场景。

在一种可能的设计中，剔除模块还包括模板测试模块；模板测试模块用于接收多个片元，并基于多个片元的模板值与模板参考值的比较结果确定是否剔除片元。

这种设计中，模板测试模块可以对片元进行初步剔除处理，剔除不需要绘制的片元，有效减少过度绘制情况，减少着色处理器的负载，提高图形处理器性能，降低功耗。

在一种可能的设计中，剔除模块还包括深度测试模块，多个片元包括有深度片元；深度测试模块，用于基于多个片元中有深度片元的深度值与深度缓冲区的值的比较结果确定是否剔除片元。

这种设计中，深度测试模块可以对片元进行初步剔除处理，剔除不需要绘制的片元，有效减少过度绘制情况，减少着色处理器的负载，提高图形处理器性能，降低功耗。

第二方面，本申请实施例提供一种图形绘制方法，该方法应用于图形处理器，图形处理器包括剔除模块和着色处理器，剔除模块包括寄存器和过度绘制剔除模块，过度绘制剔除模块的输出端和着色处理器的输入端耦合，该方法包括：存储指示信息，指示信息用于指示绘制任务的图形区域中参与过度绘制剔除操作的子区域；记录绘制任务的多个片元在图形区域中的位置信息和参与绘制时的顺序信息，并基于位置信息和顺序信息对子区域内的片元进行过度绘制剔除操作，将多个片元中需要绘制的片元发送至着色处理器；对多个片元中需要绘制的片元进行着色处理。

第二方面的有益效果请参见第一方面的有益效果。

在一种可能的设计中，过度绘制剔除操作为剔除多个片元中位置信息相同且顺序信息为在前顺序的片元。

在一种可能的设计中，对多个片元中需要绘制的片元进行着色处理之前，方法还包括：基于位置信息和顺序信息将子区域外的片元发送至着色处理器。

在一种可能的设计中，指示信息包括子区域的区域范围和第一标识，第一标识用于指示子区域的区域范围内部为参与过度绘制剔除操作的区域范围。

在一种可能的设计中，指示信息包括子区域的区域范围和第二标识，第二标识用于指示子区域的区域范围外部为参与过度绘制剔除操作的区域范围。

在一种可能的设计中，剔除模块还包括模板测试模块，记录绘制任务的多个片元在图形区域中的位置信息和参与绘制时的顺序信息之前，方法还包括：接收多个片元，并基于多个片元的模板值与模板参考值的比较结果确定是否剔除片元。

在一种可能的设计中，剔除模块还包括深度测试模块，多个片元包括有深度片元，方法还包括：基于多个片元中有深度片元的深度值与深度缓冲区的值的比较结果确定是否剔除片元。

第三方面，本申请实施例提供了一种计算机可读存储介质，包括计算机指令，当计算机指令在电子设备上运行时，使得电子设备执行上述任一方面及任一项可能的实现方式中的图形绘制方法。

第四方面，本申请实施例提供了一种计算机程序产品，当计算机程序产品在计算机或处理器上运行时，使得计算机或处理器执行上述任一方面及任一项可能的实现方式中的图形绘制方法。

第五方面，本申请实施例提供了一种芯片系统，该系统可以包括以上任一方面的任一项可能的实现方式中的无线接入设备和至少一个电子设备。该电子设备和无线接入设备可以执行上述任一方面及任一项可能的实现方式中的图形绘制方法。

可以理解的是，上述提供的任一种图形处理器、芯片系统、计算机可读存储介质或计算机程

序产品等均可以应用于上文所提供的对应的方法，因此，其所能达到的有益效果可参考对应的方法中的有益效果，此处不再赘述。

本申请的这些方面或其他方面在以下的描述中会更加简明易懂。

附图说明

- 5 图1为本申请实施例提供的一种执行设备的结构示意图；
图2为本申请实施例提供的一种执行设备的内部架构图；
图3为本申请实施例提供的一种图形处理器的内部架构图；
图4为本申请实施例提供的一种图形处理器的结构示意图；
图5为本申请实施例提供的一种子区域的区域范围的示意图；
10 图6为本申请实施例提供的另一种子区域的区域范围的示意图；
图7为本申请实施例提供的一种剔除模块的结构示意图；
图8为本申请实施例提供的一种图形处理器的处理流程图；
图9为本申请实施例提供的一种图形绘制方法的流程图；
图10为本申请实施例提供的另一种图形绘制方法的流程图；
15 图11为本申请实施例提供的一种图形处理器的处理流程图。

具体实施方式

为了便于理解，示例的给出了部分与本申请实施例相关概念的说明以供参考。如下所示：

- 20 图形处理器（graphics processing unit, GPU），又称显示核心、视觉处理器和显示芯片等，是一种专门在个人电脑、工作站、游戏机和一些移动设备（如平板电脑和智能手机等）上图像运算工作的微处理器。图形处理器将计算机系统所需要的显示信息进行转换驱动，并向显示器提供扫描信号，以控制显示器的正确显示。

混合（blend）操作，在两个原始路径之间生成很多特殊的路径的操作。混合操作能将一种形状渐变到另一种形状，从而得到一种三维立体的效果。

- 25 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述。其中，在本申请实施例的描述中，除非另有说明，“/”表示或的意思，例如，A/B可以表示A或B；本文中的“和/或”仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。另外，在本申请实施例的描述中，“多个”是指两个或多于两个。

- 30 以下，术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。在本实施例的描述中，除非另有说明，“多个”的含义是两个或两个以上。

目前，为了降低片元着色器的工作负荷，现有技术提供一种马里（mali）GPU，该 mali

- 35 GPU 可以通过前向像素消除（forward pixel kill, FPK）技术解决过度绘制问题。其中，在 mali GPU 中，经过深度测试和模板测试的片元在进入着色处理器（shader processor）中执行指令前，先进入到一个着色处理器的先入先出队列（first in first out, FIFO）中进行排队。其中，在新的片元进入到队列后，队列中会检查队列前面的片元，如果后进入队列的片元，在屏幕位置上与前面的片元重合，并且在深度或者顺序上能够覆盖前面的片元，则用新的片元替换前面的片元。通过此种方式，减少了前面的片元的指令开销，达到减少过度绘制的目的。

- 40 但是，FPK 技术对应用场景存在严格要求，在存在混合（blend）、模板比较和指令丢弃（shader discard）等操作的场景中，FPK 技术无法适用，否则图形处理器会出现功能错误。

由此，考虑到现有技术中存在的过度绘制问题，即在所绘制的物体包括混合操作、模板比较操作和指令丢弃等特殊操作的情况下无法进行剔除处理的问题，由此造成片元着色器的工作负荷较高。

- 45 因此，本申请实施例对图形处理器进行了改进，新增过度绘制剔除模块，由过度绘制剔除模块在存在混合操作、模板比较操作和指令丢弃等特殊操作的子区域内执行过度绘制剔除操作，同时也对无深度信息的片元执行过度绘制剔除操作，从而减少着色处理器的负载，提高图形处理器性能，降低功耗。

本申请实施例提供的图形处理器可以应用于不同的设备中,如应用于图 1 所示的执行设备 10, 图 1 为本申请实施例提供的一种执行设备的结构示意图。该执行设备 10 可以是终端,如服务器 11、手机终端 12、平板电脑 13、笔记本电脑 14、增强现实技术(augmented reality, AR)设备(图 1 中未示出)、虚拟现实技术(virtual reality, VR)设备(图 1 中未示出)或车载终端(图 1 中未示出)等。

如图 2 所示,图 2 为本申请实施例提供的一种执行设备的内部架构图。该执行设备 10 可以包括图形处理器、中央处理器(central processing unit, CPU)和存储器等。其中,存储器可以包括:只读存储器(read only memory, ROM)和随机存取存储器(random access memory, RAM)。该执行设备 10 还可以配置有输入/输出(input/output, I/O)接口(图 2 中未示出),用于与外部设备进行数据交互,例如用户可以通过外部设备向 I/O 接口输入数据,该输入数据在本申请实施例中可以包括:需绘制的图像,可以是执行设备 10 通过数据采集设备采集的图像,还可以是执行设备 10 数据库中的图像,还可以是来自客户设备的图像等。

如图 3 所示,图 3 为本申请实施例提供的一种图形处理器的结构框图。该图形处理器可以包括:任务解析和调度器、多个着色任务创建器(图 3 中示例的任务创建器 0~任务创建器 N, N 为正整数)、图形处理集群、总线以及缓存(例如 L2 缓存)等。

在一些实施例中,图形驱动在 CPU 对应的芯片中,CPU 可以将绘制任务发送给 GPU。由 GPU 的任务解析和调度器进行任务的解析和根据任务优先级调度。通常,图形学中包含多种类型的着色器,例如顶点着色器(vertex shader)、几何着色器(geometry shader)以及像素着色器(fragment shader)等。当任务通过任务解析和调度器下发到任务创建器时,任务创建器可以用于创建任务,将任务发送到图形处理集群。其中,图形处理集群包含了 M 个图形处理核心(graphics processing core)单元, M 为正整数。例如图形处理核心单元的数量为 4 个或者 8 个或者 16 个等。M 个图形处理核心单元可以并行的执行不同着色器程序指令。

在一些实施例中,在一个图形处理集群中,还可以包括着色器单元、纹理单元(texture unit)、装载存储(loader store)单元和特殊函数单元(special function unit)。常规的指令可以在图形处理核心单元中执行。较为复杂的纹理指令、内存加载和存储指令以及特殊函数指令由独立的协处理器(包括纹理单元、装载存储单元和特殊函数单元)执行。一个图形处理集群可以包含 N 个纹理单元,装载存储单元和特殊函数单元。

例如,着色器单元用于执行着色程序,其中纹理相关的指令发送到纹理单元处理,内存装载和存储指令送到装载存储单元处理,特殊函数发送到特殊函数单元处理。常规的算术指令、跳转指令和逻辑操作指令等,由着色器单元内部的算术逻辑单元(arithmetic and logic unit, ALU)运算单元处理。纹理单元获取纹素,完成滤波后,将滤波结果发送回着色器单元,由着色器单元继续完成剩余指令操作。

在一些实施例中, GPU 可以包含 S 个图形处理集群,例如 S 可以是 8 个、16 个等。纹理单元和装载存储单元都可向内存加载数据或者存储数据。当对 GPU 的访问请求通过总线发送到 L2 缓存,如果访问请求在 L2 缓存中没有命中,访问请求再通过内存接口被发送到内存(DDR)进行数据读取或者存储。当访问请求为读请求时,读取的结果将再返回给图形处理集群相应的模块。

本申请实施例对图形处理器进行了改进,在图形处理集群中,新增过度绘制剔除模块,在对片元进行着色之前,过度绘制剔除模块对片元中不需要绘制的片元进行剔除,只将需要绘制的片元送至着色处理器中处理。

本申请实施例提供的图形处理器接收应用程序(application, APP)的绘制任务,并执行计算操作。图形处理器和应用程序配合,可以完成各种实时绘制任务,如游戏和应用程序用户界面等。其中,应用程序通过应用程序接口(application program interface, API)设置绘制任务的图形区域中参与过度绘制剔除操作的子区域,即进行混合操作、模板比较操作和指令丢弃操作等操作的区域。应用程序将子区域的指示信息通过 API 发送至图形处理器,图形处理器基于该指示信息对绘制任务进行处理。其中,在手机或平板等设备上,应用程序使用 API 使能图形处理器的功能,可以提升性能,降低负载与功耗,提高续航时间。

下面对本申请实施例提供的图形处理器进行介绍。

本申请实施例提供一种图形处理器 40, 如图 4 所示, 图 4 为本申请实施例提供的一种图形处理器的结构示意图。图形处理器 40 包括剔除模块 41 和着色处理器 42, 剔除模块 41 包括寄存器 411 和过度绘制剔除模块 412, 过度绘制剔除模块 412 的输出端 (例如图 4 中的输出端 a) 和着色处理器 42 的输入端 (例如图 4 中的输入端 b) 耦合。

5 其中, 寄存器 411 用于存储指示信息, 指示信息用于指示绘制任务的图形区域中参与过度绘制剔除操作的子区域。

示例性的, 寄存器 411 可以是接口寄存器, 寄存器 411 可以同时被内部电路和外部电路或应用程序访问。在一个示例中, 寄存器 411 可以暂存应用程序发送的指示信息。

10 示例性的, 指示信息是由应用程序基于绘制任务生成的。对于车载屏幕、手机、平板或电脑等设备上常见的 UI 应用的显示界面, 例如购物平台的商品展示页面或相册浏览页面等, 应用程序调用 API 设置特殊操作的区域, 即进行混合操作、模板比较操作和指令丢弃操作等操作的区域, 并生成指示信息发送至寄存器 411。

另外, 绘制任务的图形区域可以理解为显示界面的区域, 例如可以是手机或显示器的显示 (display) 子系统等。

15 其中, 过度绘制剔除模块 412 用于记录绘制任务的多个片元在图形区域中的位置信息和参与绘制时的顺序信息, 并基于位置信息和顺序信息对子区域内的片元进行过度绘制剔除操作, 将多个片元中需要绘制的片元发送至着色处理器 42。

20 示例性的, 片元是二维图像中包括颜色、深度信息和纹理数据的一个点。其中, 多个片元可以包括有深度片元和无深度片元。其中, 有深度片元即有深度信息的片元, 无深度片元即无深度信息的片元, 深度信息可以是空间里的各个点相对于摄像头的距离。

其中, 过度绘制剔除模块 412 可以执行两个流程, 分别是片元相互关系生成流程和过度绘制测试流程。片元相互关系生成流程可以包括: 过度绘制剔除模块 412 记录有深度片元和无深度片元的在图形区域中的位置信息和参与绘制时的顺序信息, 并比对片元在图形区域中的位置信息是否位于参与过度绘制剔除操作的子区域内。过度绘制测试流程可以包括: 过度绘制剔除模块 412 25 执行过度绘制剔除操作, 剔除子区域中不需要绘制的片元, 将需要绘制的片元发送至着色处理器 42, 其中, 不需要绘制的片元可以理解为对最终显示界面中像素颜色没有贡献的片元。例如, 过度绘制剔除模块 412 可以去除图层叠加过程中背景 UI 元素或中间无效图层对应的片元, 剔除对最终像素颜色没有贡献的片元, 只将有贡献的片元送给着色处理器 42 进行处理。由此, 可以有效减少过度绘制情况, 减少着色处理器的负载, 提高图形处理器性能, 降低功耗。

30 其中, 着色处理器 42 用于对多个片元中需要绘制的片元进行着色处理。

示例性的, 着色处理器 42 执行着色指令操作, 对多个片元中需要绘制的片元进行着色处理。着色处理器 42 将最终的计算结果经过处理操作写入帧缓存, 最终将帧缓存的结果显示到显示界面上。

35 由此, 相较于 FPK 技术中, 在着色处理器的 FIFO 队列中执行剔除操作。本申请实施例提供的图形处理器 40 在着色处理器 42 之前的过度绘制剔除模块 412 中执行过度绘制剔除操作。其中, 过度绘制剔除模块 412 可以理解为缓存 (buffer), 过度绘制剔除模块 412 较 FIFO 队列可以处理更多片元, 可以有效减少过度绘制情况, 减少着色处理器 42 的负载。

可选的, 过度绘制剔除操作为剔除多个片元中位置信息相同且顺序信息为在前顺序的片元。

40 示例性的, 对于图层叠加过程中背景 UI 元素和中间无效图层等, 假设存在片元 1 和片元 2, 其中, 片元 1 和片元 2 在图形区域中的位置信息相同, 片元 1 的顺序信息为在前顺序, 片元 2 的顺序信息为在后顺序。由此, 片元 2 可以理解为能够覆盖片元 1, 所以片元 1 为不需要绘制的片元, 将片元 1 剔除, 可以减少片元 1 的指令开销, 达到减少过度绘制的目的。

可以理解的是, 该示例中仅示出了两个片元, 但图层叠加的层数可以是两层及以上, 位置信息相同的片元也可以是两个及以上。

45 可选的, 过度绘制剔除模块 412 还用于基于位置信息和顺序信息将子区域外的片元发送至着色处理器 42。

示例性的, 子区域外可以理解为图形区域中不执行特殊操作的区域, 即在子区域外不执行混

合操作、模板比较操作和指令丢弃操作等操作。此时，过度绘制剔除模块 412 直接将片元发送至着色处理器 42，即过度绘制剔除模块 412 不对片元进行处理，仅将片元透传至着色处理器 42。由此，可以减少片元的处理过程，提高图形处理器 40 的效率。

5 可选的，指示信息包括子区域的区域范围和第一标识，第一标识用于指示子区域的区域范围内部为参与过度绘制剔除操作的区域范围。

如图 5 所示，图 5 为本申请实施例提供的一种子区域的区域范围的示意图。其中，子区域的区域范围可以为矩形、三角形、圆或椭圆等可以用数学公式描述的封闭曲线。在一个绘制任务的图形区域中，可以同时设置多个子区域。

其中，第一标识可以指示封闭曲线内部为参与过度绘制剔除操作的区域范围。在一个示例中，
10 第一标识可以为 true。在应用程序中，对于混合操作的定义为：
*SetBlendRegions(Region*regions,bool exclusive)，其中，*regions 为子区域的区域范围，exclusive 为第一标识时，表示过度绘制剔除操作在子区域的区域范围内部生效。另外，对于模板比较操作的定义为：
*SetStencilRegions(Region*regions,bool exclusive)。对于指令丢弃操作的定义为：
15 *SetShaderDiscardRegions(Region*regions,bool exclusive)。对于特殊操作的定义为：
*SetSpecialOpRegions(Region*regions,bool exclusive)。

可以理解的是，对于子区域的区域范围的描述，也可以用输入的图像或者纹理来指定。通过此种方式可以表达一般书序公式无法描述的区域。如图 6 所示，图 6 为本申请实施例提供的另一种子区域的区域范围的示意图。其中，图像的像素对应屏幕上的一块区域，如 1x2、2x1、2x2、2x4 和 4x4 等，从而进行更灵活的控制子区域的区域范围。

20 可选的，指示信息包括子区域的区域范围和第二标识，第二标识用于指示子区域的区域范围外部为参与过度绘制剔除操作的区域范围。

示例性的，第二标识可以指示封闭曲线外部为参与过度绘制剔除操作的区域范围。继续以应用程序中对于操作的定义为例，exclusive 为第二标识时，表示过度绘制剔除操作在子区域的区域范围外部生效。

25 由此，通过设置第一标识和第二标识，可以灵活设置子区域的区域范围，适用于多种绘制场景。

可选的，剔除模块 41 还包括模板测试模块 413，模板测试模块 413 用于接收多个片元，并基于多个片元的模板值与模板参考值的比较结果确定是否剔除片元。

30 示例性的，模板测试模块 413 可以对片元进行模板测试 (stencil test)。其中，模板测试的流程为：图形处理器 40 先读取模板缓冲区中该片元的屏幕坐标位置的模板值，然后将该模板值与片元的模板参考值进行比较，其中，比较函数可以由技术人员指定，例如该模板值大于或等于模板参考值时剔除该片元，或是该模板值小于或等于模板参考值时剔除该片元。

由此，模板测试模块 413 可以对片元进行初步剔除处理，剔除不需要绘制的片元，有效减少过度绘制情况，减少着色处理器 42 的负载，提高图形处理器 40 性能，降低功耗。

35 可选的，剔除模块 41 还包括深度测试模块 414，多个片元包括有深度片元，深度测试模块 414 用于基于多个片元中有深度片元的深度值与深度缓冲区的值的比较结果确定是否剔除片元。

40 示例性的，深度测试模块 414 可以对片元进行深度测试 (depth test)，其中，深度测试的流程为：图形处理器 40 将该片元的深度值和已经存在于深度缓冲中的深度值进行比较，其中，比较函数可以由技术人员指定，例如该深度值大于或等于深度缓冲中的深度值时舍弃该片元，或是该深度值小于或等于深度缓冲中的深度值时舍弃该片元。

由此，深度测试模块 414 可以对片元进行初步剔除处理，剔除不需要绘制的片元，有效减少过度绘制情况，减少着色处理器 42 的负载，提高图形处理器 40 性能，降低功耗。其中，对于片元的处理，可以先进行模板测试，再进行深度测试。其中，对于无深度片元，深度测试模块 414 不进行剔除处理，将无深度片元透传至过度绘制剔除模块 412。

45 由此，剔除模块 41 的结构示意图如图 7 所示，剔除模块 41 包括寄存器 411、过度绘制剔除模块 412、模板测试模块 413 和深度测试模块 414。其中，剔除模块 41 的处理流程为：片元先经过模板测试模块 413 进行模板测试，再经过深度测试模块 414 进行深度测试，最后过度绘制剔除

模块 412 综合模板测试模块 413 和深度测试模块 414 的剔除结果执行过度绘制剔除操作, 剔除不需要绘制的片元, 只将需要绘制的片元发送至着色处理器 42。

除此之外, 图形处理器 40 还可以包括光栅化模块 43 (rast unit), 光栅化模块 43 用于将绘制任务的顶点数据转换为多个片元。其中, 片元中的每一个元素对应于帧缓冲区中的一个像素。光栅化实质上是把物体的数学描述以及物体相关的颜色信息转换为屏幕上用于对应位置的像素及用于填充像素的颜色的过程。

总的来说, 如图 8 所示, 图 8 为本申请实施例提供的一种图形处理器的处理流程图。图形处理器的流程图如下: 绘制任务中的顶点数据通过光栅化模块 43 转换为多个片元, 剔除模块 41 中的模板测试模块 413 和深度测试模块 414 进行初步剔除处理, 过度绘制剔除模块 412 综合模板测试模块 413 和深度测试模块 414 的剔除结果, 记录下所有片元在图形区域中的位置信息和参与绘制时的顺序信息, 剔除对像素颜色无贡献的片元, 识别最终影响像素颜色的片元并发送至着色处理器 42 进行着色处理。

应用于图 4 中所示的图形处理器, 本申请实施例提供一种图形绘制方法, 如图 9 所示, 图 9 为本申请实施例提供的一种图形绘制方法的流程图。该方法包括如下流程。

步骤 901、图形处理器存储指示信息, 指示信息用于指示绘制任务的图形区域中参与过度绘制剔除操作的子区域。

步骤 902、图形处理器记录所述绘制任务的多个片元在所述图形区域中的位置信息和参与绘制时的顺序信息, 并基于所述位置信息和所述顺序信息对所述子区域内的片元进行所述过度绘制剔除操作, 将所述多个片元中需要绘制的片元发送至所述着色处理器。

步骤 903、图形处理器对多个片元中需要绘制的片元进行着色处理。

对于存在混合操作、模板比较操作和指令丢弃操作的场景, 图形处理器中的着色处理器无法执行剔除算法的问题, 本申请实施例中的图形处理器先通过寄存器存储的指示信息确定出参与过度绘制剔除操作的子区域, 然后在着色处理器对片元进行着色处理之前, 由着色处理器之外的过度绘制剔除模块对参与过度绘制剔除操作的子区域进行过度绘制剔除操作, 即使参与过度绘制剔除操作的子区域中存在无深度信息的二维场景中的片元, 也可对无深度信息的二维场景中的片元也执行过度绘制剔除操作。这样, 不仅可避免着色处理器对于存在混合操作、模板比较操作和指令丢弃操作的场景无法执行剔除算法的问题, 也可在剔除子区域中不需要绘制的片元的基础上, 将需要绘制的片元发送至着色处理器, 可以有效减少过度绘制情况, 减少着色处理器的负载, 提高图形处理器性能, 降低功耗。

可选的, 如图 10 所示, 图 10 为本申请实施例提供的另一种图形绘制方法的流程图。其中, 步骤 903 之前, 该方法还可以包括:

步骤 904、图形处理器基于位置信息和顺序信息将子区域外的片元发送至着色处理器。

示例性的, 对于子区域外的片元, 过度绘制剔除模块不对片元进行处理, 仅将片元透传至着色处理器, 由此, 可以减少片元的处理过程, 提高图形处理器的效率。步骤 904 的实现方式可以参考上述实施例对图形处理器的介绍。

可选的, 步骤 902 之前, 该方法还可以包括:

步骤 905、图形处理器接收多个片元, 并基于多个片元的模板值与模板参考值的比较结果确定是否剔除片元。

示例性的, 模板测试模块可以对片元进行初步剔除处理, 剔除不需要绘制的片元, 有效减少过度绘制情况, 减少着色处理器的负载, 提高图形处理器性能, 降低功耗。步骤 905 的实现方式可以参考上述实施例对图形处理器的介绍。

可选的, 在步骤 905 之后, 步骤 902 之前, 该方法还可以包括:

步骤 906、图形处理器基于多个片元中有深度片元的深度值与深度缓冲区的值的比较结果确定是否剔除片元。

示例性的, 深度测试模块可以对片元进行初步剔除处理, 剔除不需要绘制的片元, 有效减少过度绘制情况, 减少着色处理器的负载, 提高图形处理器性能, 降低功耗。步骤 906 的实现方式可以参考上述实施例对图形处理器的介绍。

总的来说,如图 11 所示,如 11 为本申请实施例提供的一种图形处理器的处理流程图。其中,包括如下流程。

- (a) 寄存器存储指示信息;
- (b) 应用程序发送绘制任务至图形处理器;
- 5 (c) 图形处理器处理图元;
- (d) 光栅化模块处理图元,得到片元;
- (e) 剔除模块处理片元;
- (f) 着色处理器执行着色指令操作。

10 本申请的实施例还提供一种计算机存储介质,该计算机存储介质中存储有计算机指令,当该计算机指令在电子设备上运行时,使得电子设备执行上述相关方法步骤实现上述实施例中的图形绘制方法。

本申请的实施例还提供了一种计算机程序产品,当该计算机程序产品在计算机上运行时,使得计算机执行上述相关步骤,以实现上述实施例中电子设备执行的图形绘制方法。

15 另外,本申请的实施例还提供一种装置,这个装置具体可以是芯片,组件或模块,该装置可包括相连的处理器和存储器:其中,存储器用于存储计算机执行指令,当装置运行时,处理器可执行存储器存储的计算机执行指令,以使芯片执行上述各方法实施例中电子设备执行的图形绘制方法。

20 其中,本实施例提供的图形处理器、计算机存储介质、计算机程序产品或芯片均用于执行上文所提供的对应的方法,因此,其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果,此处不再赘述。

通过以上实施方式的描述,所属领域的技术人员可以了解到,为描述的方便和简洁,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。

25 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述模块或单元的划分,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,例如多个单元或组件可以结合或者可以集成到另一个装置,或一些特征可以忽略,或不执行。另一点,所显示或讨论的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,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

30 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是一个物理单元或多个物理单元,即可以位于一个地方,或者也可以分布到多个不同地方。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

35 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,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,也可以采用软件功能单元的形式实现。

所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个可读存储介质中。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现有技术做出贡献的部分或者该技术方案的全部或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个设备(可以是单片机,芯片等)或处理器(processor) 40 执行本申请各个实施例所述方法的全部或部分步骤。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)、随机存取存储器(random access memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

45 以上内容,仅为本申请的具体实施方式,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。因此,本申请的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1.一种图形处理器，其特征在于，所述图形处理器包括剔除模块和着色处理器，所述剔除模块包括寄存器和过度绘制剔除模块，所述过度绘制剔除模块的输出端和所述着色处理器的输入端耦合；

5 所述寄存器，用于存储指示信息，所述指示信息用于指示绘制任务的图形区域中参与过度绘制剔除操作的子区域；

 所述过度绘制剔除模块，用于记录所述绘制任务的多个片元在所述图形区域中的位置信息和参与绘制时的顺序信息，并基于所述位置信息和所述顺序信息对所述子区域内的片元进行所述过度绘制剔除操作，将所述多个片元中需要绘制的片元发送至所述着色处理器；

10 所述着色处理器，用于对所述多个片元中需要绘制的片元进行着色处理。

 2.根据权利要求 1 所述的图形处理器，其特征在于，所述过度绘制剔除操作为剔除所述多个片元中所述位置信息相同且所述顺序信息为在前顺序的片元。

 3.根据权利要求 1 或 2 所述的图形处理器，其特征在于，所述过度绘制剔除模块，还用于基于所述位置信息和所述顺序信息将所述子区域外的片元发送至所述着色处理器。

15 4.根据权利要求 1 所述的图形处理器，其特征在于，所述指示信息包括子区域的区域范围和第一标识，所述第一标识用于指示所述子区域的区域范围内为参与过度绘制剔除操作的区域范围。

 5.根据权利要求 1 所述的图形处理器，其特征在于，所述指示信息包括子区域的区域范围和第二标识，所述第二标识用于指示所述子区域的区域范围外部为参与过度绘制剔除操作的区域范围。

20 6.根据权利要求 1 所述的图形处理器，其特征在于，所述剔除模块还包括模板测试模块；

 所述模板测试模块，用于接收多个片元，并基于所述多个片元的模板值与模板参考值的比较结果确定是否剔除所述片元。

25 7.根据权利要求 6 所述的图形处理器，其特征在于，所述剔除模块还包括深度测试模块，所述多个片元包括有深度片元；

 所述深度测试模块，用于基于所述多个片元中有深度片元的深度值与深度缓冲区的值的比较结果确定是否剔除所述片元。

 8.一种图形绘制方法，其特征在于，所述方法应用于图形处理器，所述图形处理器包括剔除模块和着色处理器，所述剔除模块包括寄存器和过度绘制剔除模块，所述过度绘制剔除模块的输出端和所述着色处理器的输入端耦合，所述方法包括：

30 存储指示信息，所述指示信息用于指示绘制任务的图形区域中参与过度绘制剔除操作的子区域；

 记录所述绘制任务的多个片元在所述图形区域中的位置信息和参与绘制时的顺序信息，并基于所述位置信息和所述顺序信息对所述子区域内的片元进行所述过度绘制剔除操作，将所述多个片元中需要绘制的片元发送至所述着色处理器；

35 对所述多个片元中需要绘制的片元进行着色处理。

 9.根据权利要求 8 所述的方法，其特征在于，所述过度绘制剔除操作为剔除所述多个片元中所述位置信息相同且所述顺序信息为在前顺序的片元。

 10.根据权利要求 8 或 9 所述的方法，其特征在于，所述对所述多个片元中需要绘制的片元进行着色处理之前，所述方法还包括：

40 基于所述位置信息和所述顺序信息将所述子区域外的片元发送至所述着色处理器。

 11.根据权利要求 8 所述的方法，其特征在于，所述指示信息包括子区域的区域范围和第一标识，所述第一标识用于指示所述子区域的区域范围内为参与过度绘制剔除操作的区域范围。

 12.根据权利要求 8 所述的方法，其特征在于，所述指示信息包括子区域的区域范围和第二标识，所述第二标识用于指示所述子区域的区域范围外部为参与过度绘制剔除操作的区域范围。

45 13.根据权利要求 8 所述的方法，其特征在于，所述剔除模块还包括模板测试模块，所述记录所述绘制任务的多个片元在所述图形区域中的位置信息和参与绘制时的顺序信息之前，所述方法

还包括:

接收多个片元, 并基于所述多个片元的模板值与模板参考值的比较结果确定是否剔除所述片元。

- 5 14. 根据权利要求 9 所述的方法, 其特征在于, 所述剔除模块还包括深度测试模块, 所述多个片元包括有深度片元, 所述方法还包括:

基于所述多个片元中有深度片元的深度值与深度缓冲区的值的比较结果确定是否剔除所述片元。

15. 一种计算机可读存储介质, 其特征在于, 包括计算机指令, 当计算机指令在电子设备上运行时, 使得电子设备执行上述权利要求 8-14 中的任一项所述的方法。