



(12) 发明专利申请

(10) 申请公布号 CN 117292039 A

(43) 申请公布日 2023. 12. 26

(21) 申请号 202311585648.5

(22) 申请日 2023.11.27

(71) 申请人 芯瞳半导体技术(山东)有限公司
地址 264006 山东省烟台市自由贸易试验区烟台片区长江路300号业达科技园D栋301室

(72) 发明人 王恒

(74) 专利代理机构 西安维英格知识产权代理事务所(普通合伙) 61253
专利代理师 沈寒西 归莹

(51) Int. Cl.
G06T 15/00 (2011.01)
G06T 1/20 (2006.01)

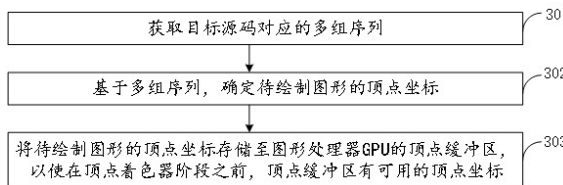
权利要求书2页 说明书10页 附图4页

(54) 发明名称

顶点坐标生成方法、装置、电子设备及计算机存储介质

(57) 摘要

本公开提供了一种顶点坐标生成方法、装置、电子设备及计算机存储介质,属于计算机图形学技术领域,该方法在顶点着色器阶段之前,将生成的顶点坐标存储至顶点缓冲区以供其他接口使用。该方法包括:获取目标源码对应的多组序列,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;基于多组序列,确定待绘制图形的顶点坐标;将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以便在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。



1. 一种顶点坐标生成方法,其特征在于,所述方法包括:

获取目标源码对应的多组序列,所述目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;

基于所述多组序列,确定所述待绘制图形的顶点坐标;

将所述待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,所述顶点缓冲区有可用的顶点坐标。

2. 根据权利要求1所述的方法,其特征在于,所述每组序列包括多个子输入序列和对应的子指令序列,每个子输入序列和对应的子指令序列对应一个顶点的目标坐标,所述目标坐标为横坐标、纵坐标或竖坐标;所述基于所述多组NIR指令序列和输入序列,确定所述待绘制图形的顶点坐标,包括:

将每个所述子输入序列作为目标输入序列,对应的所述子指令序列作为目标指令序列,执行下述步骤S1,直至所述目标指令序列为空,得到一个顶点的目标坐标;

步骤S1包括:

在所述目标指令序列不为空的情况下,对所述目标输入序列中位于前两位的输入,执行所述目标指令序列位于第一位的指令对应的运算,得到运算结果;

删除所述目标输入序列中前两位输入,并将所述运算结果添加到删除后的目标输入序列的第一位,得到更新后的目标输入序列;

删除所述目标指令序列中第一位的指令,得到更新后的目标指令序列;

将所述更新后的目标输入序列作为所述目标输入序列,所述更新后的目标指令序列作为目标指令序列。

3. 根据权利要求1所述的方法,其特征在于,所述将所述待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区之后,所述方法还包括:

绑定所述顶点缓冲区与CPU中的目标缓冲区,以便于在CPU中能够通过所述目标缓冲区获取所述顶点缓冲区中的顶点坐标。

4. 根据权利要求1所述的方法,其特征在于,所述获取目标源码对应的多组序列之前,所述方法还包括:

检测CPU中的目标缓冲区是否为空,所述目标缓冲区对应所述顶点缓冲区;

所述获取目标源码对应的多组序列,包括:

在所述目标缓冲区为空的情况下,获取目标源码对应的多组序列。

5. 根据权利要求1至4任一项所述的方法,其特征在于,所述NIR指令序列中指令对应的运算包括:加,减,乘,除,左移,右移,取余,大于,小于,大于或等于,小于或等于,等于,不等于,与,或,与或。

6. 一种顶点坐标生成装置,其特征在于,所述装置包括:获取部分,确定部分和存储部分;

所述获取部分,用于获取目标源码对应的多组序列,所述目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;

所述确定部分,用于基于所述多组序列,确定所述待绘制图形的顶点坐标;

所述存储部分,用于将所述待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,所述顶点缓冲区有可用的顶点坐标。

7. 根据权利要求6所述的装置,其特征在于,所述每组序列包括多个子输入序列和对应的子指令序列,每个子输入序列和对应的子指令序列对应一个顶点的目标坐标,所述目标坐标为横坐标、纵坐标或竖坐标;

所述确定部分,具体用于将每个所述子输入序列作为目标输入序列,对应的所述子指令序列作为目标指令序列,执行下述步骤S1,直至所述目标指令序列为空,得到一个顶点的目标坐标;

步骤S1包括:

在所述目标指令序列不为空的情况下,对所述目标输入序列中位于前两位的输入,执行所述目标指令序列位于第一位的指令对应的运算,得到运算结果;

删除所述目标输入序列中前两位输入,并将所述运算结果添加到删除后的目标输入序列的第一位,得到更新后的目标输入序列;

删除所述目标指令序列中第一位的指令,得到更新后的目标指令序列;

将所述更新后的目标输入序列作为所述目标输入序列,所述更新后的目标指令序列作为目标指令序列。

8. 根据权利要求6所述的装置,其特征在于,所述装置包括:检测部分;

所述检测部分,用于在所述获取目标源码对应的多组序列之前,检测CPU中的目标缓冲区是否为空,所述目标缓冲区对应所述顶点缓冲区;

所述获取部分,具体用于在所述目标缓冲区为空的情况下,获取目标源码对应的多组序列。

9. 一种电子设备,其特征在于,包括处理器,存储器及存储在所述存储器上并可在所述处理器上运行的程序或指令,所述程序或指令被所述处理器执行时实现如权利要求1至5中任一项所述的顶点坐标生成方法的步骤。

10. 一种计算机可读存储介质,其特征在于,所述可读存储介质上存储程序或指令,所述程序或指令被处理器执行时实现如权利要求1至5中任一项所述的顶点坐标生成方法的步骤。

顶点坐标生成方法、装置、电子设备及计算机存储介质

技术领域

[0001] 本公开涉及计算机图形学技术领域,尤其涉及一种顶点坐标生成方法、装置、电子设备及计算机存储介质。

[0002]

背景技术

[0003] 在图形渲染过程中,顶点坐标是图形管线的一个重要数据源。通常,顶点坐标是由用户直接定义,或者是在顶点着色器源码中定义。将顶点坐标存储在图形处理器(Graphics Processing Unit,GPU)的顶点缓冲区中,以在图形管线的顶点着色器阶段将顶点坐标作为输入执行渲染。

[0004] 对于通过顶点着色器源码生成对应的顶点坐标,顶点坐标是在顶点着色器阶段才生成再存入顶点缓冲区。因此,若有接口(如vkCmdDraw函数)在顶点着色器阶段之前需要使用顶点坐标,则会由于无法从顶点缓冲区获取到顶点坐标导致程序运行失败。

[0005]

发明内容

[0006] 本公开提供了一种顶点坐标生成方法、装置、电子设备及计算机存储介质,该方法在顶点着色器阶段之前,将生成的顶点坐标存储至顶点缓冲区以供其他接口使用。

[0007] 第一方面,本公开提供了一种顶点坐标生成方法,该方法包括:获取目标源码对应的多组序列,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;基于多组序列,确定待绘制图形的顶点坐标;将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。

[0008] 第二方面,本公开提供了一种顶点坐标生成装置,该装置包括:获取部分,确定部分和存储部分;获取部分,用于获取目标源码对应的多组序列,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;确定部分,用于基于多组序列,确定待绘制图形的顶点坐标;存储部分,用于将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。

[0009] 第三方面,本公开提供了一种电子设备,该电子设备包括处理器、存储器及存储在该存储器上并可在该处理器上运行的程序或指令,该程序或指令被该处理器执行时实现如第一方面所述的顶点坐标生成方法的步骤。

[0010] 第四方面,本公开提供了一种计算机可读存储介质,该计算机可读存储介质上存储程序或指令,该程序或指令被处理器执行时实现如第一方面所述的顶点坐标生成方法的步骤。

[0011] 第五方面,本公开提供了一种计算机程序产品,其中,该计算机程序产品包括计算

机程序或指令,当该计算机程序产品在处理器上运行时,使得处理器执行该计算机程序或指令,实现如第一方面所述的顶点坐标生成方法的步骤。

[0012] 第六方面,本公开提供了一种芯片,该芯片包括处理器和通信接口,该通信接口和该处理器耦合,该处理器用于运行程序或指令,实现如第一方面所述的顶点坐标生成方法。

[0013] 本公开提供了一种顶点坐标生成方法,包括:获取目标源码对应的多组序列,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;基于多组序列,确定待绘制图形的顶点坐标;将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。如此,可以在渲染管线创建之后的任一阶段使用顶点缓冲区中的顶点坐标,从而更便于图形渲染,提升图形渲染的效率。

[0014]

附图说明

[0015] 图1为本公开提供的一种应用场景示意图;
图2为本公开提供的一种图形管线的结构示意图;
图3为本公开提供的顶点坐标生成方法的流程示意图之一;
图4为本公开提供的顶点坐标生成方法的流程示意图之二;
图5为本公开提供的顶点坐标生成方法的流程示意图之三;
图6为本公开提供的顶点坐标生成方法的流程示意图之四;
图7为本公开提供的一种顶点坐标生成装置的结构框图;
图8为本公开提供的一种电子设备的硬件结构示意图。

[0016]

具体实施方式

[0017] 下面将结合本公开中的附图,对本申请实施例中的技术方案进行清楚地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员获得的所有其他实施例,都属于本申请保护的范围。

[0018] 本申请的说明书中的术语“第一”“第二”等是用于区别类似的对象,而不用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便本公开能够以除了在这里图示或描述的那些以外的顺序实施,且“第一”“第二”等所区分的对象通常为一类,并不限定对象的个数,例如第一对象可以是一个,也可以是多个。此外,说明书中“和/或”表示所连接对象的至少其中之一,字符“/”,一般表示前后关联对象是一种“或”的关系。

[0019] 首先,图1为本公开示出的一种应用场景示意图。如图1所示,当中央处理器(Central Processing Unit,CPU) 101需要渲染对象时,会将渲染对象依赖的渲染数据(如:着色器参数、纹理数据、顶点数据等)存储至主存102中,再通过绑定主存102与显存107,从而使得显存107能够获取到主存102中的渲染数据;CPU 101通过应用程序103向Vulkan 104发送渲染命令,Vulkan 104即为上层应用程序103与底层GPU106的桥梁,显卡驱动105用于

将从Vulkan 104接收到的渲染命令转化为GPU 106可执行的指令,GPU106用于从显存107中提取渲染数据以执行从显卡驱动105接收的指令,GPU 106处理后的渲染数据存储至显存107中。

[0020] 其中,Vulkan是一个跨平台的2D和3D图形程序接口(Application Program Interface,API),在图形学中,通常与GPU对接。

[0021] 显存,也被叫做帧缓存,它的作用是用来存储GPU处理过或者即将提取的渲染数据,现今,高密度运算由GPU在显卡上完成

显卡驱动把Vulkan的函数调用翻译成GPU 的执行命令,也就是说两者中有一个庞大的数据库,该数据库即为Vulkan,Vulkan中既存储了函数同时也存储了GPU的执行命令,以供显卡驱动将函数调用转换为执行命令。

[0022] Vulkan与OpenGL或DNIRectX pre-11不同,在Vulkan中,所有GPU命令(GPU Commands)都必须通过命令缓冲区(VkCommandBuffer),命令缓冲区从命令池(VkCommandPool)分配,并在队列(VkQueue)上执行。具体的流程为:从命令池分配一块命令缓冲区,使用VkCmd函数将命令记录到命令缓冲区中,调用vkQueueSubmit函数将命令缓冲区提交到一个队列(VkQueue, GPU 的“执行端口”)中,GPU即可执行显卡驱动转换后的队列中的命令。

[0023] 具体地,如图2所示,为一种图形管线的结构示意图,图形管线中的各个阶段串行执行,即上一阶段的输出作为下一阶段的输入。在图2示出的图形管线中,采用方框示意的阶段为固定功能阶段,固定功能阶段仅允许通过调整参数来改变操作,但其工作流程是预先定义好的;采用圆角框示意的阶段为可编程阶段,该阶段允许将相关代码上传至显卡,以实现需要的操作,如:使用片段着色器实现光追、光照的操作;上述图形管线所包括的各渲染阶段依次为:

输入装配阶段21,为固定功能阶段,从顶点缓冲区获取顶点数据(顶点数据包括顶点坐标及顶点着色信息);

顶点着色阶段22,为可编程阶段,在输入装配阶段21中的顶点缓冲区为空的情况下,用于生成顶点坐标并存储至顶点缓冲区中,以及将顶点坐标转换为屏幕坐标;输入装配阶段21中的顶点缓冲区不为空的情况下,用于将顶点坐标转换为屏幕坐标;

细分着色阶段23,为可编程阶段,接收顶点着色阶段22的输出作为输入,并将几何图形细分为更多、更小的三角形使得渲染表面和边缘更加光滑,输出细分后的顶点数据;

几何着色阶段24,为可编程阶段,接收细分着色阶段23的输出作为输入,通过高效的几何运算来增删顶点,并输出顶点数据;

光栅化阶段25,为固定功能阶段,负责准备片段着色阶段26的图元。举例来说,光栅化阶段25可产生若干碎片以供片段着色阶段26进行阴影处理;

片段着色阶段26,为固定功能阶段,负责接收光栅化阶段25输出的碎片且产生例如颜色等像素数据。此外,片段着色阶段26还可执行例如纹理混合和照明模型计算等每像素处理。

[0024] 颜色混合阶段27,为固定功能阶段,通常负责对像素数据实施多种操作,例如实施透明测试(Alpha test)、模板测试(stencil test),以及将像素数据与对应于与该像素相关的其他片段的其他像素数据混合等操作。当颜色混合阶段27已经完成处理像素数据(即

输出数据)时,可以将处理完成的以图像为例的像素数据写入到以显存为例的渲染目标,以产生最终结果。

[0025] 相关技术中,若顶点缓冲区为空,则在队列提交之后,在上述图2所示的顶点着色器阶段22才会通过目标源码(目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码)计算得到顶点坐标。也就是说,在队列提交之前是无法使用顶点缓冲区中的顶点坐标,若在顶点着色器阶段之前使用顶点坐标,则会导致应用程序运行失败。如:在vkQueueSubmit函数之前的绘制命令准备阶段(图形管线流程开始之前),需要调用渲染函数(如vkCmdDraw函数、vkCmdDrawIndexed函数等)以告知GPU在CPU侧已准备的数据以及依据已准备的数据如何绘制,但若渲染函数依赖的顶点缓冲区(VertexBuffer)为空,则应用程序在执行至该函数时会报错(如vkCmdDraw函数的输入参数为顶点缓冲区,即CPU告知GPU顶点缓冲区中的顶点坐标为已准备好的数据,但在顶点缓冲区为空时,该函数获取不到顶点坐标,导致该函数执行失败),导致应用程序执行失败。

[0026] 因此,本公开提供了一种顶点坐标生成方法,以在图形管线创建后(顶点着色器阶段之前),基于顶点着色器源码对应的NIR指令序列和输入序列确定待绘制图形的顶点坐标,并将顶点坐标存储至GPU可以访问的顶点缓冲区(位于显存中)中以供其他接口使用。

[0027] 下面结合附图,通过具体的实施例对本公开提供的顶点坐标生成方法进行详细地说明。

[0028] 如图3所示,本公开的提供一种顶点坐标方法,下面以执行主体为电子设备为例,对本公开提供的顶点坐标生成方法进行示例性的说明。该方法可以包括下述的步骤301至步骤303。

[0029] 301、获取目标源码对应的多组序列。

[0030] 其中,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标。

[0031] 在Mesa架构中,在创建图形管线(顶点着色器阶段之前)时,需要加入顶点着色器和片段着色器,在创建图形管线的同时已经将顶点着色器源码通过中间表示(New Intermediate Representation,NIR)指令转换为GPU可以识别的NIR指令序列(如mul指令表示乘法,add表示加法,shl表示左移,shr表示右移)以及输入序列。

[0032] 可选地,NIR指令序列中指令对应的运算包括:加,减,乘,除,左移,右移,取余,大于,小于,大于或等于,小于或等于,等于,不等于,与,或,与或。

[0033] 本实施例中示出的NIR指令序列中指令对应的运算仅为示例,NIR指令序列中指令对应的运算还可以包括除上述运算外的其他运算,本公开不做具体限定。

[0034] 302、基于多组序列,确定待绘制图形的顶点坐标。

[0035] 具体地,基于多组序列,确定待绘制图形的顶点坐标,可以是建立模板函数,在创建图形管线之后调用模板函数,即可根据模板函数确定待绘制图形的顶点坐标。

[0036] 可选地,为一种可实施的模板函数的具体实现流程,上述步骤302具体可以通过下述步骤302a实现。其中,每组序列包括多个子输入序列和对应的子指令序列,每个子输入序列和对应的子指令序列对应一个顶点的目标坐标,目标坐标为横坐标、纵坐标或竖坐标。

[0037] 302a、将每个子输入序列作为目标输入序列,对应的子指令序列作为目标指令序

列,执行下述步骤S1,直至目标指令序列为空,得到一个顶点的目标坐标。

[0038] 在待绘制图形为二维图形的情况下,每组序列包括:确定顶点的横坐标需要的子输入序列和对应的子指令序列,以及确定顶点的纵坐标需要的子输入序列和对应的子指令序列;在待绘制图形为三维图形的情况下,在二维图形的基础上每组序列还包括:确定顶点的竖坐标需要的子输入序列和对应的子指令序列;一组序列确定一个顶点坐标。

[0039] 其中,如图4所示,步骤S1包括下述步骤S11至步骤S16。

[0040] S11、检测目标指令序列是否为空。

[0041] 是则执行步骤S12,否则执行步骤S13至步骤S16。

[0042] S12、确定运算结果为目标坐标。

[0043] S13、对目标输入序列中位于前两位的输入,执行目标指令序列位于第一位的指令对应的运算,得到运算结果。

[0044] S14、删除目标输入序列中前两位输入,并将运算结果添加到删除后的目标输入序列的第一位,得到更新后的目标输入序列。

[0045] S15、删除目标指令序列中第一位的指令,得到更新后的目标指令序列。

[0046] S16、将更新后的目标输入序列作为目标输入序列,更新后的目标指令序列作为目标指令序列。

[0047] 返回执行步骤S11。

[0048] 示例性地,顶点着色器伪源码为:

```
int segCount = 3;
float a = 2;
float b = 5;
for (int i = 0; i < segCount; i++) {
    float x = a * (a+i);
    float y = b * (b-i);
    vertext[i] = (Vertex){x, y};
    printf("%f , %f\n", x, y);}

```

根据源码得到的第一组序列包括:子指令序列(add,mul)和子输入序列(2,0,2),子指令序列(sub,mul)和子输入序列(5,0,5);第二组序列包括:子指令序列(add,mul)和子输入序列(2,1,2),子指令序列(sub,mul)和子输入序列(5,1,5);第三组序列包括:子指令序列(add,mul)和子输入序列(2,2,2),子指令序列(sub,mul)和子输入序列(5,2,5)。

[0049] 将第一组序列中的子指令序列(add,mul)和子输入序列(2,0,2)作为目标指令序列个目标输入序列,目标指令序列不为空,则对目标输入序列的前两位2和0执行目标指令序列第一位的add对应的运算,即:2+0=2,得到的运算结果为2;删除目标指令序列中第一位的指令,得到更新后的目标指令序列为(mu1),删除目标输入序列中前两位的输入,并将运算结果添加到目标输入序列中,得到更新后的目标输入序列为(2,2),将更新后的目标指令序列作为目标指令序列,将更新后的目标输入序列作为目标输入序列;继续检测目标指令序列是否为空,确定不为空,则对目标输入序列的前两位2和2执行目标指令序列第一位的mul对应的运算,即 $2 \times 2 = 4$,得到的运算结果为4;删除目标指令序列中第一位的指令,得到更新后的目标指令序列为(),删除目标输入序列中前两位的输入,并将运算结果添加到目

标输入序列中,得到更新后的目标输入序列为(4),将更新后的目标指令序列和目标输入序列作为目标指令序列和目标输入序列,目标指令序列为空,则确定运算结果4为顶点的一个坐标。

[0050] 将第一组序列中的子指令序列(sub,mul)和子输入序列(5,0,5)作为目标指令序列和目标输入序列,重复执行步骤S11至步骤S12,确定的运算结果为: $5-0=5$, $5\times 5=25$,即顶点的另一个坐标为25;由此得到第一组序列对应的顶点坐标为(4,25)。

[0051] 将第二组序列中的子指令序列和子输入序列分别作为目标指令序列和目标输入序列重复执行步骤S11至步骤S12,一个子指令序列和子输入序列确定的运算结果为: $2+1=3$, $3\times 2=6$,另一个子指令序列和子输入序列确定的运算结果为: $5-1=4$, $4\times 5=20$;由此得到第二组序列对应的顶点坐标为(6,20)。

[0052] 将第三组序列中的子指令序列和子输入序列分别作为目标指令序列和目标输入序列重复执行步骤S11至步骤S12,一个子指令序列和子输入序列确定的运算结果为: $2+2=4$, $4\times 2=8$,另一个子指令序列和子输入序列确定的运算结果为: $5-2=3$, $3\times 5=15$;由此得到第二组序列对应的顶点坐标为(8,15)。

[0053] 本实施例中,对于一个子输入序列和对应的子指令序列,执行步骤S11至步骤S16,得到对应的一个顶点的坐标,将多组序列中每组序列包括的每个子输入序列和对应的子指令序列作为目标输入序列和目标指令序列,执行步骤S11至步骤S16,以最终得到多组序列对应的顶点坐标,即待绘制图形包括的全部顶点坐标。如此,对于任意的子输入序列和子指令序列,通过通用的模板函数(即上述步骤S11至步骤S16实现的功能)即可得到顶点的一个坐标值,顶点坐标生成的过程简便,从而提升了整个渲染过程的效率。

[0054] 需要说明的是,模板函数实现的功能是根据多组输入序列和NIR指令序列,确定出待绘制图形的顶点坐标,但具体模板函数可以根据实际需要确定,本申请实施例不做限定。如:在上述步骤S11至步骤S16对应的模板函数的基础上,可以不删除目标指令序列中的指令,按目标指令序列的顺序执行直至执行到目标指令序列中的最后一个指令,即可得到一个顶点的坐标值。

[0055] 303、将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。

[0056] 本实施例中,在图形管线创建之后,即执行步骤301至步骤303,从而将得到的顶点坐标存储至GPU的顶点缓冲区。

[0057] 本公开实施例中,获取目标源码对应的多组序列,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;基于多组序列,确定待绘制图形的顶点坐标;将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。如此,可以在图形管线创建之后的任一阶段使用顶点缓冲区中的顶点坐标,从而更便于图形渲染,提升图形渲染的效率。

[0058] 应用程序在CPU侧执行,但CPU无法直接访问GPU侧顶点缓冲区的顶点坐标,为了能够在CPU侧访问顶点缓冲区的顶点坐标,本公开一些实施例中,结合图3,如图5所示,上述步骤303之后,该顶点坐标生成方法还包括下述步骤304。

[0059] 304、绑定顶点缓冲区与CPU中的目标缓冲区,以便于在CPU中能够通过目标缓冲区

获取顶点缓冲区中的顶点坐标。

[0060] 本实施例中,通过vkCmdBindVertexBuffers函数将顶点缓冲区与目标缓冲区绑定,即GPU的顶点缓冲区映射到CPU的目标缓冲区,访问目标缓冲区即相当于访问顶点缓冲区。

[0061] 具体地,在Vulkan中,在确定顶点坐标之后,调用vkCreateBuffer函数在GPU侧创建顶点缓冲区,调用vkMapMemory函数将CPU侧计算出来的顶点坐标存储到GPU侧的顶点缓冲区,最后调用vkCmdBindVertexBuffers函数将GPU侧的顶点缓冲区与CPU侧的command buffer(目标缓冲区)绑定在一起,以便于在CPU侧,从command buffer能直接获取到GPU侧顶点缓冲区的顶点坐标。顶点缓冲区与commandbuffer绑定之后,访问CPU侧的command buffer,对应访问的是GPU侧的顶点缓冲区,后续vkCmdDraw函数可以通过command buffer(command buffer作为vkCmdDraw函数的参数)使用顶点缓冲区中的顶点坐标,在vkQueueSubmit函数提交绘制命令到GPU后,即可输出待绘制图像。

[0062] 本公开一些实施例中,结合图3,如图6所示,上述步骤301之前,该顶点坐标生成方法还包括下述步骤305,上述步骤301具体可以通过下述步骤301a实现。

[0063] 305、检测CPU中的目标缓冲区是否为空。

[0064] 其中,目标缓冲区对应顶点缓冲区。由于应用程序在CPU侧执行,无法直接访问GPU侧的顶点缓冲区域,因此,若顶点缓冲区中存储有顶点坐标,则会同步将目标缓冲区和顶点缓冲区绑定,从而可以通过检测目标缓冲区来确定GPU侧的顶点缓冲区是否为空。

[0065] 301a、在目标缓冲区为空的情况下,获取目标源码对应的多组序列。

[0066] 若目标缓冲区为空,则确定顶点缓冲区中未存储顶点坐标,通过执行上述步骤301至步骤303,向顶点缓冲区存入顶点坐标;若目标缓冲区不为空,则确定在创建图形管线时,已将用户自定义的顶点坐标存储至顶点缓冲区,因此,可以继续执行后续的渲染流程,无需执行步骤301至步骤303。

[0067] 本公开实施例中,在确定顶点缓冲区为空的情况下,确定顶点坐标并存储至顶点缓冲区;在确定顶点缓冲区不为空的情况下,继续执行正常的渲染流程;如此,对于顶点缓冲区不为空的情况,不会增加渲染流程,对于顶点缓冲区为空的情况,在创建图形管线后即可将确定的顶点坐标存储至顶点缓冲区,方便在创建图形管线后的任一阶段使用顶点缓冲区的顶点坐标。

[0068] 图7为本公开示出的一种顶点坐标生成装置的结构框图,如图7所示,包括:获取部分701,确定部分702和存储部分703;该获取部分701,用于获取目标源码对应的多组序列,目标源码为顶点着色器源码中用于在顶点着色器阶段生成待绘制图形的顶点坐标的源码,每组序列包括输入序列和对应的中间表示NIR指令序列,每组序列对应一个顶点坐标;该确定部分702,用于基于多组序列,确定待绘制图形的顶点坐标;该存储部分703,用于将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区,以使在顶点着色器阶段之前,顶点缓冲区有可用的顶点坐标。

[0069] 本公开一些实施例中,每组序列包括多个子输入序列和对应的子指令序列,每个子输入序列和对应的子指令序列对应一个顶点的目标坐标,目标坐标为横坐标、纵坐标或竖坐标;确定部分702,具体用于将每个子输入序列作为目标输入序列,对应的子指令序列作为目标指令序列,执行下述步骤S1,直至目标指令序列为空,得到一个顶点的目标坐标;

步骤S1包括:在目标指令序列不为空的情况下,对目标输入序列中位于前两位的输入,执行目标指令序列位于第一位的指令对应的运算,得到运算结果;删除目标输入序列中前两位输入,并将运算结果添加到删除后的目标输入序列的第一位,得到更新后的目标输入序列;删除目标指令序列中第一位的指令,得到更新后的目标指令序列;将更新后的目标输入序列作为目标输入序列,更新后的目标指令序列作为目标指令序列。

[0070] 本公开一些实施例中,该顶点坐标生成装置还包括:绑定部分704,该绑定部分704用于在将待绘制图形的顶点坐标存储至图形处理器GPU的顶点缓冲区之后,绑定顶点缓冲区与CPU中的目标缓冲区,以便于在CPU中能够通过目标缓冲区获取顶点缓冲区中的顶点坐标。

[0071] 本公开一些实施例中,该装置还包括:检测部分705;该检测部分705,用于在获取目标源码对应的多组序列之前,检测CPU中的目标缓冲区是否为空,目标缓冲区对应顶点缓冲区;获取部分701,具体用于在目标缓冲区为空的情况下,获取目标源码对应的多组序列。

[0072] 本公开一些实施例中,NIR指令序列中指令对应的运算包括:加,减,乘,除,左移,右移,取余,大于,小于,大于或等于,小于或等于,等于,不等于,与,或,与或。

[0073] 需要说明的是:如图7所示顶点坐标生成装置700中一定包括的部分用实线框示意,如获取部分701,确定部分702和存储部分703;该顶点坐标生成装置700中可以包括也可以不包括的部分用虚线框示意,如绑定部分704和检测部分705。

[0074] 需要说明的是,上述顶点坐标生成装置可以为本申请上述方法实施例中的电子设备,也可以是该电子设备中能够实现该装置实施例功能的功能模块和/或功能实体,本申请实施例不做限定。

[0075] 本申请实施例中,各模块可以实现上述方法实施例提供的顶点坐标生成方法,且能达到相同的技术效果,为避免重复,这里不再赘述。

[0076] 请参考图8,其示出了本公开一个示例性实施例提供的电子设备的结构框图。在一些示例中,电子设备可以为智能手机、智能手表、台式电脑、手提电脑、虚拟现实终端、增强现实终端、无线终端和膝上型便携计算机等设备中的至少一种。电子设备具有通信功能,可以接入有线网络或无线网络。电子设备可以泛指多个终端中的一个,本领域技术人员可以知晓,上述终端的数量可以更多或更少。可以理解地,电子设备承担本公开技术方案的计算及处理工作,本公开对此不作限定。

[0077] 如图8所示,本公开中的电子设备可以包括一个或多个如下部件:处理器810和存储器820。

[0078] 可选的,处理器810利用各种接口和线路连接整个电子设备内的各个部分,通过运行或执行存储在存储器820内的指令、程序、代码集或指令集,以及调用存储在存储器820内的数据,执行电子设备的各种功能和处理数据。可选地,处理器810可以采用数字信号处理(Digital Signal Processing,DSP)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)、可编程逻辑阵列(Programmable Logic Array,PLA)中的至少一种硬件形式来实现。处理器810可集成中央处理器(Central Processing Unit,CPU)、图像处理器(Graphics Processing Unit,GPU)、神经网络处理器(Neural-network Processing Unit,NPU)和基带芯片等中的一种或几种的组合。其中,CPU主要处理操作系统、用户界面和应用程序等;GPU用于负责触摸显示屏所需要显示的内容的渲染和绘制;NPU用于实现人工智能

(Artificial Intelligence, AI) 功能; 基带芯片用于处理无线通信。可以理解的是, 上述基带芯片也可以不集成到处理器810中, 单独通过一块芯片进行实现。

[0079] 存储器820可以包括随机存储器(Random Access Memory, RAM), 也可以包括只存储器(Read-Only Memory, ROM)。可选地, 该存储器820包括非瞬时性计算机可读介质(non-transitory computer-readable storage medium)。存储器820可用于存储指令、程序、代码、代码集或指令集。存储器820可包括存储程序区和存储数据区, 其中, 存储程序区可存储用于实现操作系统的指令、用于至少一个功能的指令(比如触控功能、声音播放功能、图像播放功能等)、用于实现以上各个方法实施例的指令等; 存储数据区可存储根据电子设备的使用所创建的数据等。

[0080] 除此之外, 本领域技术人员可以理解, 上述附图所示出的电子设备的结构并不构成对电子设备的限定, 电子设备可以包括比图示更多或更少的部件, 或者组合某些部件, 或者不同的部件布置。比如, 电子设备中还包括显示屏、摄像组件、麦克风、扬声器、射频电路、输入单元、传感器(比如加速度传感器、角速度传感器、光线传感器等等)、音频电路、WiFi模块、电源、蓝牙模块等部件, 在此不再赘述。

[0081] 本公开还提供了一种计算机可读存储介质, 该计算机可读存储介质存储有至少一条指令, 所述至少一条指令用于被处理器执行以实现如上各个实施例所述的顶点坐标生成方法。

[0082] 本公开还提供了一种计算机程序产品, 该计算机程序产品包括计算机指令, 该计算机指令存储在计算机可读存储介质中; 电子设备的处理器从计算机可读存储介质读取该计算机指令, 处理器执行该计算机指令, 使得该电子设备执行以实现上述各个实施例所述的顶点坐标生成方法。

[0083] 本申请实施例另提供了一种芯片, 所述芯片包括处理器和通信接口, 所述通信接口和所述处理器耦合, 所述处理器用于运行程序或指令, 实现上述顶点坐标生成方法实施例的各个过程, 且能达到相同的技术效果, 为避免重复, 这里不再赘述。

[0084] 应理解, 本申请实施例提到的芯片还可以称为系统级芯片、系统芯片、芯片系统或片上系统芯片等。

[0085] 在本公开所提供的几个实施例中, 应该理解到, 所揭露的系统、装置、服务器和方法, 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系统, 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。

[0086] 本领域技术人员应该可以意识到, 在上述一个或多个示例中, 本公开所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时, 可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质, 其中通信介质包括便于从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0087] 需要说明的是: 本公开所记载的技术方案之间, 在不冲突的情况下, 可以任意组

合。

[0088] 以上所述,仅为本发明的具体实施方式,但本发明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发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。

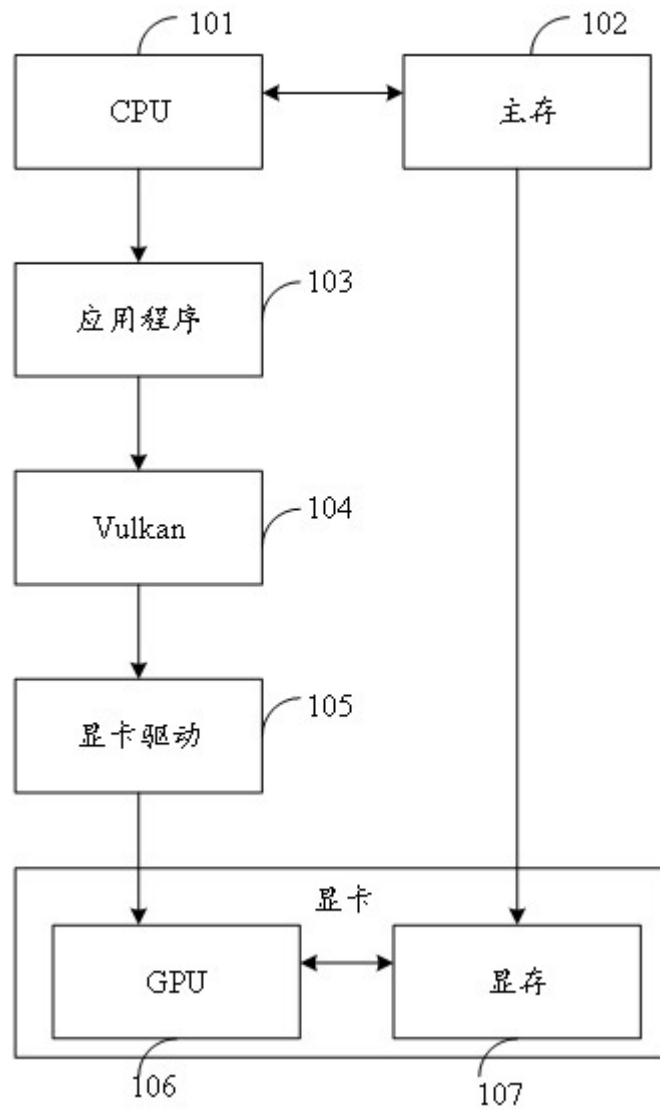


图1



图2

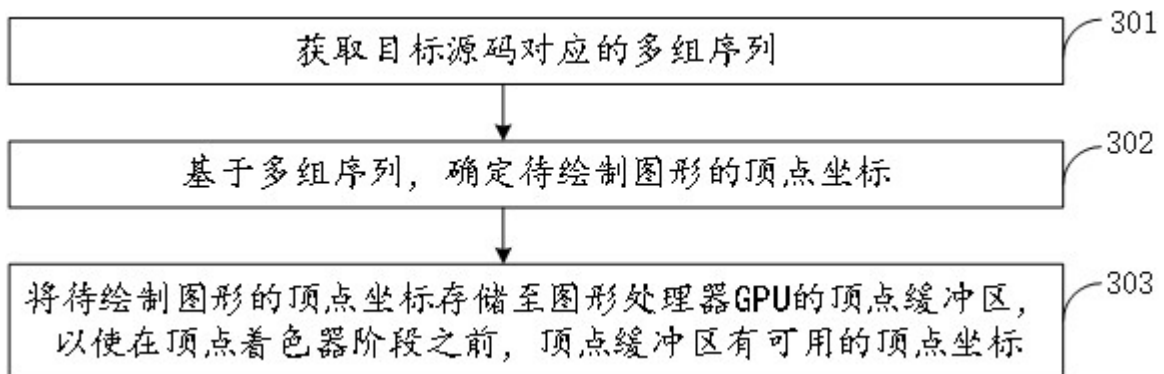


图3

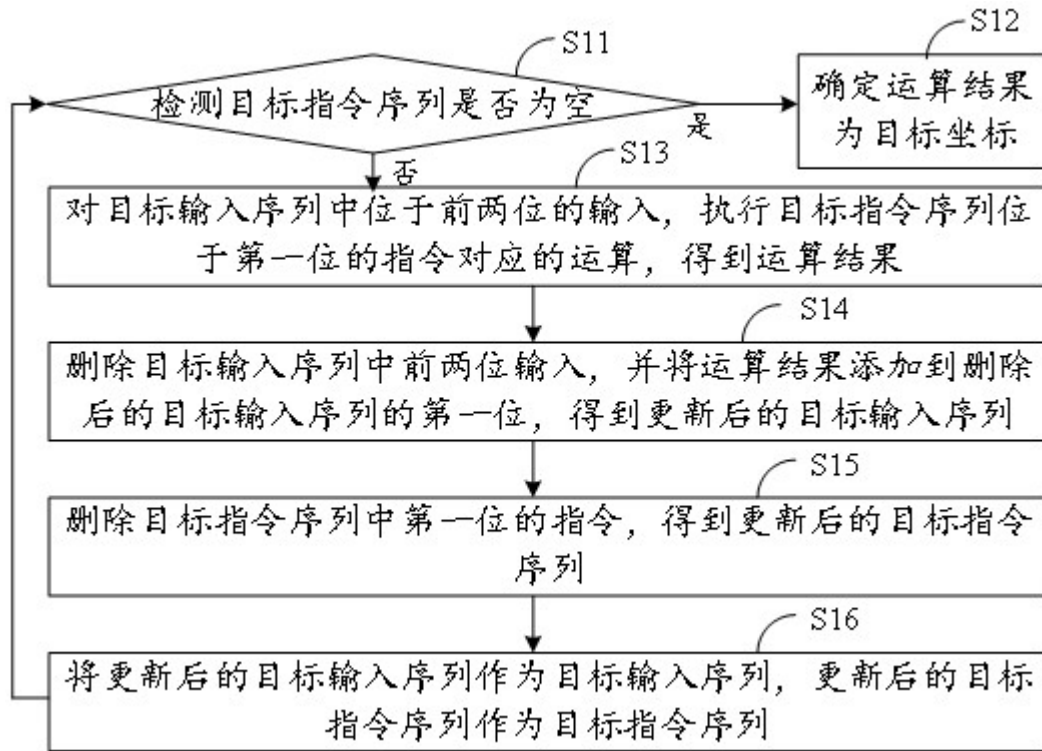


图4

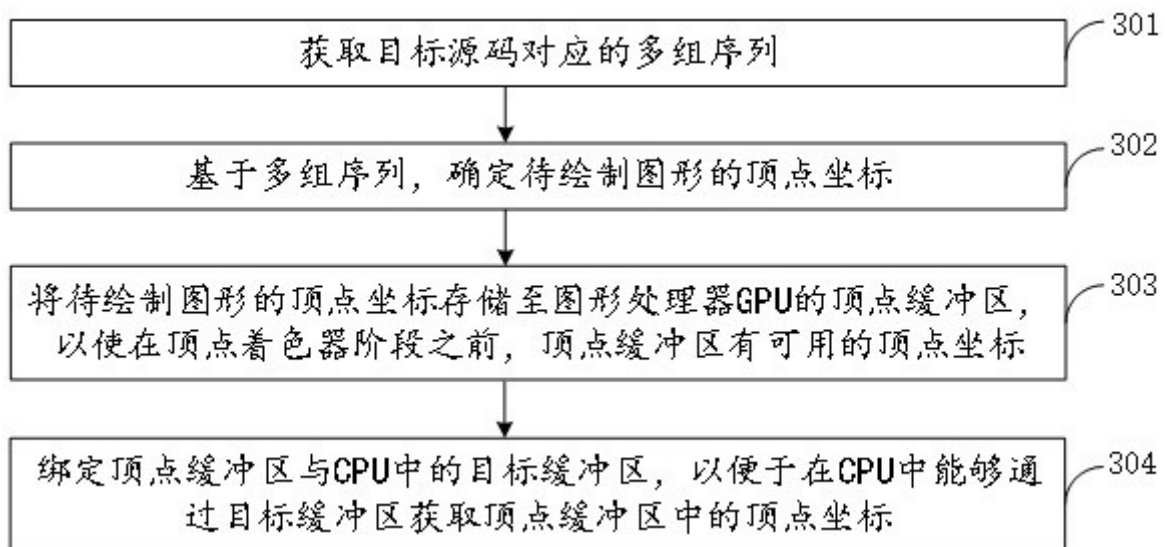


图5

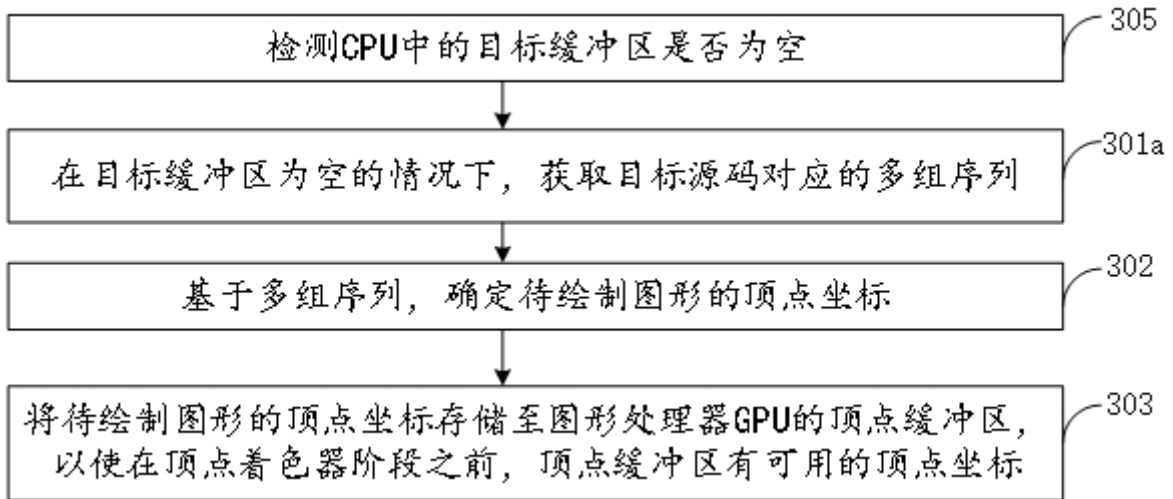


图6

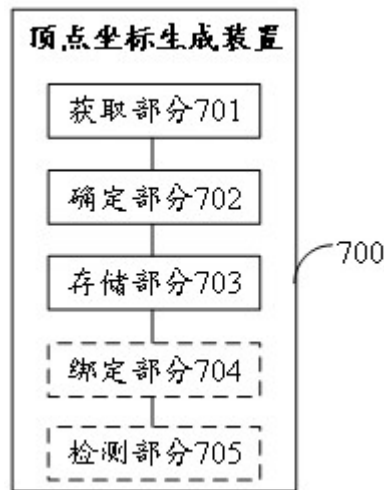


图7

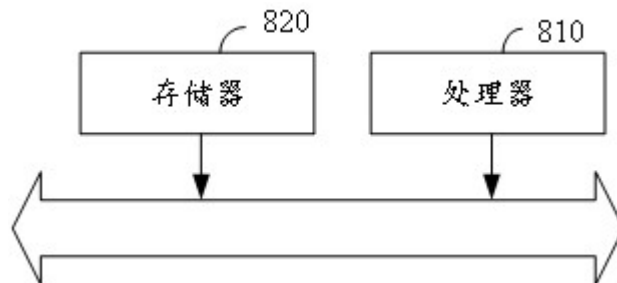


图8