(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116524096 A
(43)申请公布日 2023.08.01

(21)申请号 202310805324.1

(22)申请日 2023.06.30

(71)申请人 南京砺算科技有限公司
地址 210031 江苏省南京市(江苏)自由贸易试验区南京片区团结路99号孵鹰大厦2794室

(72)发明人 阙恒 朱康挺 张祖英 丛亮亮

(74)专利代理机构 北京集佳知识产权代理有限公司 11227
专利代理师 李笑笑

(51)Int.Cl.
G06T 15/00(2011.01)
G06F 9/54(2006.01)

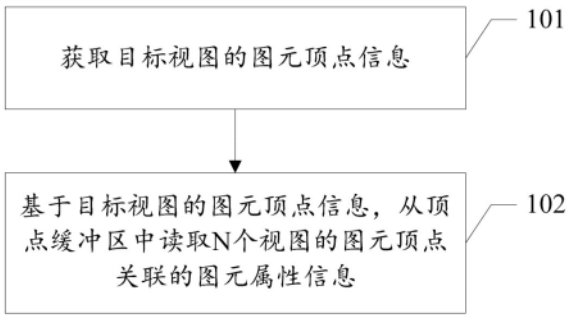
权利要求书1页 说明书5页 附图1页

(54)发明名称

图形处理方法及图形处理器、存储介质

(57)摘要

一种图形处理方法及图形处理器、存储介质，所述图形处理方法包括：获取目标视图的图元顶点信息；基于所述目标视图的图元顶点信息，从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据；所述目标视图为所述N个视图中的一个，N个视图对应的图元个数均相等，N为正整数且N≥2。采用上述方案，能够提高数据传输效率，降低GPU功耗。



CN 116524096 A

1.一种图形处理方法,其特征在于,包括:

获取目标视图的图元顶点信息;

基于所述目标视图的图元顶点信息,从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据;所述目标视图为所述N个视图中的一个,N个视图对应的图元个数均相等,N为正整数且N≥2。

2.如权利要求1所述的图形处理方法,其特征在于,所述基于所述目标视图的图元顶点信息,从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据,包括:

基于视图个数N,对所述目标视图的图元顶点信息依视图进行展开,获取所述N个视图的图元顶点;

从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据。

3.如权利要求2所述的图形处理方法,其特征在于,所述基于视图个数N,对所述目标视图的图元顶点信息依视图进行展开,包括:

获取图元展开粒度,所述图元展开粒度为M个图元;M为正整数;

分多个轮次获取所述N个视图的图元顶点,在每一轮次中,依次获取每一个视图在当前轮次的图元顶点,相邻轮次之间的步长为M个图元。

4.如权利要求3所述的图形处理方法,其特征在于,所述从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据,包括:

按照所述每一个视图对应的图元顶点的排列顺序,从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据。

5.如权利要求4所述的图形处理方法,其特征在于,还包括:

读取第i轮次第一个图元顶点关联的图元属性数据之前,则清除属性命中测试缓存中存储的第i-1轮次读取的图元属性数据,其中,i为正整数,且1≤i≤K,K为轮次总数。

6.如权利要求1所述的图形处理方法,其特征在于,所述获取目标视图的图元顶点信息,包括:

获取图元生成器发送的所述目标视图的图元顶点信息。

7.如权利要求1所述的图形处理方法,其特征在于,所述顶点缓冲区中存储所述N个视图的图元顶点关联的图元属性数据,不同视图的图元顶点关联的图元属性数据存储在不同区域。

8.一种图形处理器,其特征在于,包括:图元生成器以及多核心工作收集器,其中:

所述图元生成器,用于输出目标视图的图元顶点信息;

所述多核心工作收集器,用于基于所述目标视图的图元顶点信息,从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据;所述目标视图为所述N个视图中的一个,N个视图对应的图元个数均相等,N为正整数且N≥2。

9.如权利要求8所述的图形处理器,其特征在于,所述多核心工作收集器,用于基于视图个数N,对所述目标视图的图元顶点信息依视图进行展开,获取所述N个视图的图元顶点;从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据。

10.一种计算机可读存储介质,所述计算机可读存储介质为非易失性存储介质或非瞬态存储介质,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运行时执行权利要求1~7任一项所述的图形处理方法的步骤。

图形处理方法及图形处理器、存储介质

技术领域

[0001] 本发明涉及图形处理技术领域，尤其涉及一种图形处理方法及图形处理器、存储介质。

背景技术

[0002] 虚拟现实(Virtual Reality,VR)应用需要从不同的视角对同一场景绘制两遍，这需要中央处理器(Central Processing Unit,CPU)向图形处理器(Graphics Processing Unit,GPU)发送两个近乎完全一致的绘制命令(draw call)。相应地，GPU的工作负载被增加(顶点属性读取，执行线程数目等)。且行业标准规定的视图(view)数目最低要求是6，这就意味着上述的GPU工作负载会被增至6倍。

[0003] 多视图渲染(Multiview Rendering)技术能够实现通过一次绘制命令将图形绘制到多个视图上，降低GPU工作负载。

[0004] 现有的多视图渲染技术中，由图元生成器读取多视图的图元(primitive)属性(attribute)数据，并将读取到的每一视图的图元属性数据输出至多核心工作收集器。但是，现有的多视图渲染技术，数据传输量较大，数据传输效率较低，GPU功耗较大。

发明内容

[0005] 本发明解决的是使用多视图渲染技术的过程中，存在数据传输量较大、数据传输效率较低、GPU功耗较大的技术问题。

[0006] 为解决上述技术问题，本发明提供一种图形处理方法，包括：获取目标视图的图元顶点信息；基于所述目标视图的图元顶点信息，从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据；所述目标视图为所述N个视图中的一个，N个视图对应的图元个数均相等，N为正整数且N≥2。

[0007] 可选的，所述基于所述目标视图的图元顶点信息，从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据，包括：基于视图个数N，对所述目标视图的图元顶点信息依视图进行展开，获取所述N个视图的图元顶点；从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据。

[0008] 可选的，所述基于视图个数N，对所述目标视图的图元顶点信息依视图进行展开，包括：获取图元展开粒度，所述图元展开粒度为M个图元；M为正整数；分多个轮次获取所述N个视图的图元顶点，在每一轮次中，依次获取每一个视图在当前轮次的图元顶点，相邻轮次之间的步长为M个图元。

[0009] 可选的，所述从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据，包括：按照所述每一个视图对应的图元顶点的排列顺序，从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据。

[0010] 可选的，所述图形处理方法还包括：读取第i轮次第一个图元顶点关联的图元属性数据之前，则清除属性命中测试缓存中存储的第i-1轮次读取的图元属性数据，其中，i为正

整数，且1≤i≤K，K为轮次总数。

[0011] 可选的，所述获取目标视图的图元顶点信息，包括：获取图元生成器发送的所述目标视图的图元顶点信息。

[0012] 可选的，所述顶点缓冲区中存储所述N个视图的图元顶点关联的图元属性数据，不同视图的图元顶点关联的图元属性数据存储在不同区域。

[0013] 本发明还提供了一种图形处理器，包括：图元生成器以及多核心工作收集器，其中：所述图元生成器，用于输出目标视图的图元顶点信息；所述多核心工作收集器，用于基于所述目标视图的图元顶点信息，从顶点缓冲区中读取N个视图的图元顶点关联的图元属性数据；所述目标视图为所述N个视图中的一个，N个视图对应的图元个数均相等，N为正整数且N≥2。

[0014] 可选的，所述多核心工作收集器，用于基于视图个数N，对所述目标视图的图元顶点信息依视图进行展开，获取所述N个视图的图元顶点；从所述顶点缓冲区中读取所述N个视图的图元顶点关联的图元属性数据。

[0015] 本发明还提供了一种计算机可读存储介质，所述计算机可读存储介质为非易失性存储介质或非瞬态存储介质，其上存储有计算机程序，所述计算机程序被处理器运行时执行上述任一种所述的图形处理方法的步骤。

[0016] 与现有技术相比，本发明的技术方案具有以下有益效果：

多核心工作收集器获取N个视图中的目标视图的图元顶点信息，基于目标视图的图元顶点信息，从顶点缓冲区读取N个视图的图元顶点关联的图元属性数据。由多核心工作收集器进行N个视图的图元顶点的展开，采用一个目标视图的图元顶点信息获取N个视图的图元顶点，可以有效提高数据传输效率，降低GPU功耗。

[0017] 进一步，分多个轮次获取N个视图的图元顶点，在每一轮次中，依次获取每一个视图在当前轮次的图元顶点，可以提高数据命中率。

[0018] 此外，在读取第i轮次第一个图元顶点关联的图元属性数据之前，清除属性命中测试缓存中存储的第i-1轮次读取的图元属性数据，可以避免图元属性数据读取错误。

附图说明

[0019] 图1是本发明实施例中的一种图形处理方法的流程图；

图2是本发明实施例中的一种图形处理器的结构示意图；

图3是本发明实施例中的一种图元示意图。

具体实施方式

[0020] 现有的多视图渲染技术中，由图元生成器读取多视图的图元属性数据。图元生成器依据配置的视图数量，展开多视图的图元顶点标识，而后将每个图元顶点标识逐视图地发送至多核心工作收集器。多核心工作收集器实际上需要接收所有视图对应的图元顶点标识。由此，现有的多视图渲染技术，数据传输量较大，数据传输效率较低，图形处理器功耗较大。

[0021] 在本发明实施例中，由多核心工作收集器进行N个视图的图元顶点的展开，采用一个目标视图的图元顶点信息获取N个视图的图元顶点，故可以有效降低数据传输量，提高数

据传输效率,降低图形处理器功耗。

[0022] 为使本发明的上述目的、特征和有益效果能够更为明显易懂,下面结合附图对本发明的具体实施例做详细的说明。

[0023] 本发明提供了一种图形处理方法,参照图1,以下通过具体步骤进行说明。参照图2,给出了本发明实施例中的一种图形处理器的结构示意图。以下结合图1以及图2进行说明。

[0024] 参照图2,在具体实施中,图形处理器可以包括线程构造单元21、调度执行核心22、顶点缓冲区23、图元生成器24以及多核心工作收集器25。

[0025] 步骤101,获取目标视图的图元顶点信息。

[0026] 步骤102,基于目标视图的图元顶点信息,从顶点缓冲区23中读取N个视图的图元顶点关联的图元属性数据。

[0027] 在具体实施中,可以存在N个视图,N为正整数且N≥2。目标视图可以为N个视图中的一个。具体地,N个视图对应的图元个数均相等。由于N个视图对应的图元个数均相等,且N个视图对应的图元的类型相同,故每个视图对应的图元顶点个数相同,图元顶点信息(如图元顶点标识)相同。

[0028] 例如,每个视图对应的图元顶点标识均为0~15。

[0029] 在本实施例中,目标视图可以为N个视图中的第一个视图。通过获取N个视图中的第一个视图的图元顶点信息,确定N个视图对应的图元顶点,进而从顶点缓冲区23中获取N个视图的图元顶点关联的图元属性数据。

[0030] 在具体实施中,编译器(compiler)可以在指令层面,识别出渲染(shader)指令中哪些属性(attributes)数据依赖视图(view)标识,哪些属性数据可以视为能够被不同视图的公用(也即不同视图对应的这些属性数据相同)。

[0031] 对于依赖视图标识的属性数据,编译器可以复制(duplicate)其关联的指令;对于不依赖视图标识的属性数据,其关联的指令不需要进行复制,也即保持一份。

[0032] 编译器可以将依赖视图标识的属性数据以及不依赖视图标识的属性数据输出至调度执行核心22,调度执行核心22可以将依赖视图标识的属性数据以及不依赖视图标识的属性数据输出至顶点缓冲区23(vertex buffer),由顶点缓冲区23缓存上述的属性数据。不同视图对应的属性数据可以存储在顶点缓冲区23的不同区域。

[0033] 目标视图的顶点标识(id)可以由线程构造单元21(warp constructor)传送至图元生成器24。图元生成器24可以基于目标视图的顶点标识,生成目标视图的图元顶点信息。图元顶点信息可以包括图元顶点标识(id)。

[0034] 在本实施例中,由于图元生成器24接收到的顶点标识为目标视图对应的顶点标识,故图元生成器24生成的图元顶点信息,是指目标视图对应的图元顶点信息。换言之,图元生成器24生成的图元顶点信息,是指一个视图的图元顶点信息。

[0035] 在具体实施中,多核心工作收集器25可以包括先入先出(First Input First Output,FIFO)单元,先入先出单元可以接收图元生成器24输出的目标视图的图元顶点标识。多核心工作收集器25采用配置的图元展开粒度(prim_granularity),对N个视图对应的图元顶点标识进行展开,并根据展开顺序从顶点缓冲区23读取相应图元的属性数据。

[0036] 具体地,图元展开粒度可以根据具体的应用场景设定,设定的图元展开粒度可以

存储在预定的存储器中。在进行视图展开之前，多核心工作收集器25可以从该存储器中读取图元展开粒度。

[0037] 在具体实施中，图元展开粒度可以包括1个或者2个以上的图元。作为一优选示例，图元展开粒度可以包括至少两个图元。

[0038] 在依据图元展开粒度对N个视图对应的图元顶点标识进行展开时，可以分多个轮次获取N个视图的图元顶点。在每一轮次中，依次获取每一个视图在当前轮次的图元顶点，相邻轮次之间的步长为M个图元。

[0039] 在具体实施中，可以通过K个轮次，获取N个视图的图元顶点。K=L/M，L为每个视图对应的图元总个数。

[0040] 例如，设定N=2，也即存在两个视图，分别为view0和view1，两个视图对应的图元个数均为8，view0对应的图元的顶点标识信息依次为prim0~prim7，view1对应的图元的顶点标识信息依次为prim0~prim7。图元展开粒度配置为4，也即一个图元展开粒度包括4个图元，需要两个轮次获取2个视图的图元顶点。

[0041] 在对两个视图进行展开时，以如下顺序展开：

view0.prim0，view0.prim1，view0.prim2，view0.prim3，view1.prim0，view1.prim1，view1.prim2，view1.prim3，view0.prim4，view0.prim5，view0.prim6，view0.prim7，view1.prim4，view1.prim5，view1.prim6，view1.prim7。

[0042] 上述示例中，view0.prim0，表征为view0中顶点标识信息为0的图元，以此类推。

[0043] 可见，本实施例中，由多核心工作收集器25按照“每一个视图读取图元展开粒度(prim_granularity)对应的图元”的顺序，依次交替对N个视图的图元顶点标识进行展开。之后，按照图元顶点标识的展开顺序，从顶点缓冲区23中读取N个视图中每一个图元顶点关联的图元属性数据。

[0044] 而在现有技术中，则是由图元生成器24对N个视图进行展开，展开顺序如下：prim0.view0，prim0.view1，……prim0.view1N，prim1.view0，prim1.view1，……prim0.view1N……。

[0045] 也就是说，现有技术中，由图元生成器24按照“每一个图元(per primitive)每一个视图(per view)”的方式展开，从顶点缓冲区23读取图元对应的属性数据时，图元属性缓存(primitive attributes cache hit test)命中率较低。

[0046] 在本实施例中，由多核心工作收集器25按照“每一个视图读取图元展开粒度(prim_granularity)对应个数的图元”的顺序，依次交替对N个视图的图元顶点标识进行展开，在从顶点缓冲区23读取图元对应的属性数据时，能够有效提高图元属性缓存命中率。

[0047] 在具体实施中，由于是以图元展开粒度为单位，对N个视图的图元顶点进行依次交替展开，则可能会出现如下场景：在第i-1轮次以图元展开粒度为单元，对N个视图的图元顶点进行一次交替展开后，在第i轮次对N个视图的图元顶点进行交替展开时，第i-1轮次展开的第N个视图的图元顶点与第i轮次展开的第1个视图的图元存在重叠。此时，会存在属性数据读取错误的情况出现。

[0048] 参照图3，给出了本发明实施例中的一种图元示意图。设定图元为三角形图元。图元生成器24获取到目标视图的顶点标识为v0、v1、v2、v3、v4、v5、v6、v7、v8。图元生成器24输出的三角形列表为v0v1v2，v1v3v2，v2v3v4，v3v5v4，v4v5v6、v5v7v6、v6v7v8。

[0049] 设定图元展开粒度配置为4,存在2个视图分别为view0和view1,第1次以图元展开粒度为单元,对2个视图的图元顶点进行第一次交替展开,得到如下展开顺序:view0.prim0,view0.prim1,view0.prim2,view0.prim3,view1.prim0,view1.prim1,view1.prim2,view1.prim3。对2个视图的图元顶点进行第二次交替展开,得到如下展开顺序:view0.prim4,view0.prim5,view0.prim6,view0.prim7,view1.prim4,view1.prim5,view1.prim6,view1.prim7。

[0050] 第1次展开,第2个视图对应的最后一个图元为view1.prim3,该图元对应的三角形顶点为v3v5v4;第2次展开,第1个视图对应的第一个图元为view0.prim4,该图元对应的三角形顶点为v4v5v6。可见,图元view1.prim3与图元view0.prim4的顶点存在重叠,该重叠顶点为v4。

[0051] 在本实施例中,为避免图元顶点的图元属性数据读取错误,可以在第i轮次以图元展开粒度对N个视图的图元顶点进行一次交替展开之前,对属性命中测试缓存(attributes hit test cache)进行一次无效化操作,清除第i-1轮次读取的图元数据数据。

[0052] 具体地,对属性命中测试缓存进行无效化操作,可以是指对属性命中测试缓存进行清除操作。通过对属性命中测试缓存进行无效化操作,可以避免数据读取错误的情况出现。

[0053] 继续上述示例,在读取图元view0.prim4对应的图元属性数据之前,对属性命中测试缓存进行无效化操作,属性命中测试缓存中存储的数据被清除。在完成无效化操作之后,再读取图元view0.prim4对应的图元属性数据。

[0054] 本发明还提供了一种计算机可读存储介质,所述计算机可读存储介质为非易失性存储介质或非瞬态存储介质,其上存储有计算机程序,所述计算机程序被处理器运行时执行上述任一种所述的图形处理方法的步骤。

[0055] 本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通过程序来指示相关的硬件来完成,该程序可以存储于一计算机可读存储介质中,存储介质可以包括:ROM、RAM、磁盘或光盘等。

[0056] 虽然本发明披露如上,但本发明并非限定于此。任何本领域技术人员,在不脱离本发明的精神和范围内,均可作各种更动与修改,因此本发明的保护范围应当以权利要求所限定的范围为准。

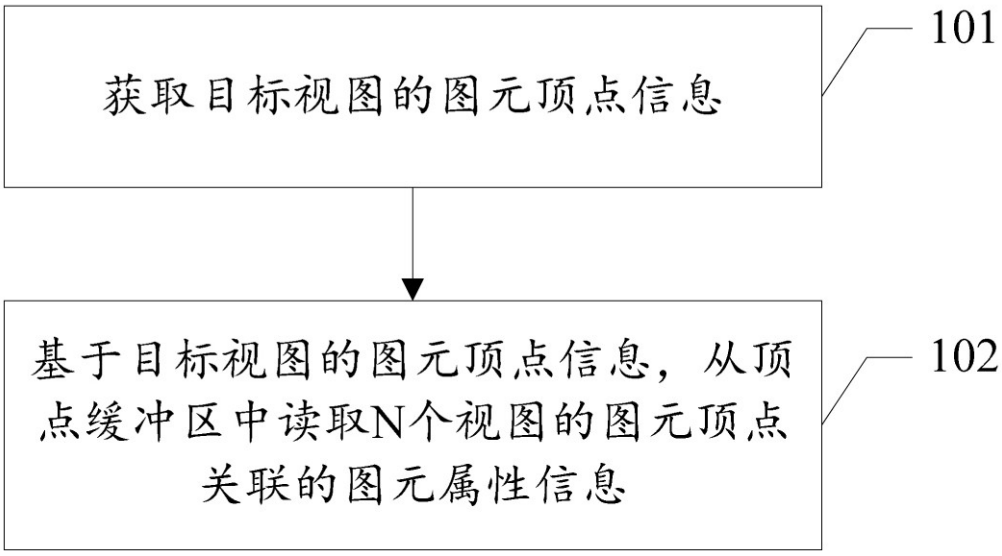

获取目标视图的图元顶点信息
101
基于目标视图的图元顶点信息，从顶点缓冲区中读取N个视图的图元顶点关联的图元属性信息
102

图 1

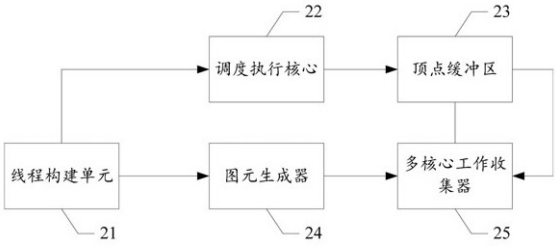

22
23
调度执行核心
顶点缓冲区
线程构建单元
图元生成器
多核心工作收集器
21
24
25

图 2

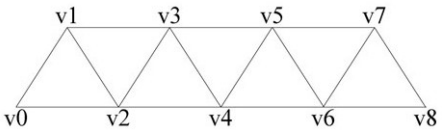

v1
v3
v5
v7
v0
v2
v4
v6
v8

图 3