(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111143018 A
(43)申请公布日 2020.05.12

(21)申请号 201911417460.3

(22)申请日 2019.12.31

(71)申请人 北京旷视机器人技术有限公司
地址 100096 北京市海淀区西三旗建材城内1幢一层125号

(72)发明人 夏天晗 姚晓谊 谭家玮

(74)专利代理机构 北京隆源天恒知识产权代理事务所(普通合伙) 11473
代理人 鞠永帅

(51)Int.Cl.
G06F 9/451(2018.01)
G06T 11/00(2006.01)
G06T 15/00(2011.01)

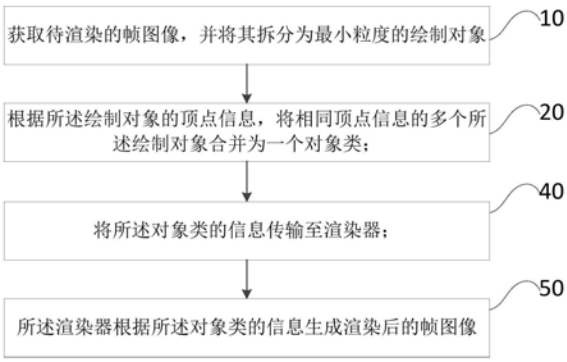
权利要求书1页 说明书9页 附图5页

(54)发明名称

一种前端图像渲染方法、装置和电子设备

(57)摘要

本发明提供了一种前端图像渲染方法、装置和电子设备，所述方法包括：获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象；根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；每个所述对象类的信息中包含一个顶点信息和多个可变信息；将所述对象类的信息传输至渲染器；所述渲染器根据所述对象类的信息生成渲染后的帧图像。这样，将相同的顶点信息进行合并，这样，可以将该帧图像的绘制或渲染过程中JS到WebGL的顶点信息传输次数降低到最少，从而大大增加图像渲染的效率。



CN 111143018 A

1.一种前端图像渲染方法,其特征在于,包括:

获取待渲染的帧图像,并将其拆分为最小粒度的绘制对象;

根据所述绘制对象的顶点信息,将相同顶点信息的多个所述绘制对象合并为一个对象类;每个所述对象类的信息中包含一个顶点信息和多个可变信息,所述多个可变信息与所述对象类中的多个绘制对象一一对应;

将所述对象类的信息传输至渲染器;

所述渲染器根据所述对象类的信息生成渲染后的帧图像。

2.根据权利要求1所述的前端图像渲染方法,其特征在于,所述绘制对象包括顶点信息和可变信息,所述可变信息包括偏移属性、缩放比、颜色信息、材质信息和边长信息中的至少一项。

3.根据权利要求1所述的前端图像渲染方法,其特征在于,在所述根据所述绘制对象的顶点信息,将相同顶点信息的多个所述绘制对象合并为一个对象类之前,所述方法还包括:

开辟存储空间,所述存储空间用于分别存储所述顶点信息和所述可变信息。

4.根据权利要求1-3中任一所述的前端图像渲染方法,其特征在于,还包括:

接收到帧图像的更新指令、平移指令或缩放指令后,根据所述更新指令、平移指令或缩放指令获取更新信息;

将所述更新信息传输至渲染器;

所述渲染器根据所述更新信息生成渲染后的帧图像。

5.根据权利要求4所述的前端图像渲染方法,其特征在于,所述将所述更新信息传输至渲染器,之前还包括:

根据所述更新信息对存储在所述存储空间内的所述对象类的信息进行更新。

6.根据权利要求4所述的前端图像渲染方法,其特征在于,所述对象类的信息还包括视口矩阵,所述视口矩阵与所述顶点信息、所述可变信息相互独立存储。

7.根据权利要求6所述的前端图像渲染方法,其特征在于,所述根据所述更新指令、平移指令或缩放指令获取更新信息中,在接收到帧图像的所述平移指令或缩放指令后,获取的所述更新信息用于更新视口矩阵。

8.一种前端图像渲染装置,其特征在于,包括:

获取单元(1),用于获取待渲染的帧图像,并将其拆分为最小粒度的绘制对象;

合并单元(2),用于根据所述绘制对象的顶点信息,将相同顶点信息的多个所述绘制对象合并为一个对象类;每个所述对象类的信息中包含一个顶点信息和多个可变信息,所述多个可变信息与所述对象类中的多个绘制对象一一对应;

传输单元(3),用于将所述对象类的信息传输至渲染器;

渲染器(4),用于根据所述对象类的信息生成渲染后的帧图像。

9.一种电子设备,包括处理器以及存储器,其特征在于,所述存储器存储有控制程序,所述控制程序被处理器执行时实现如权利要求1-7中任一所述的前端图像渲染方法。

10.一种计算机可读存储介质,存储有指令,其特征在于,所述指令被处理器加载并执行时实现如权利要求1-7中任一所述的前端图像渲染方法。

一种前端图像渲染方法、装置和电子设备

技术领域

[0001] 本发明涉及图像渲染技术领域，具体而言，涉及一种前端图像渲染方法、装置和电子设备。

背景技术

[0002] 在现实图像的具体显示中，现实中的场景和实体用三维形式表示，便于操纵和变换，而图形的显示设备大多是二维的光栅化显示器和点阵化打印机，对三维实体场景进行N维光栅和点阵化的表示就是图像渲染。

[0003] WebGL(Web Graphics Library)是图像渲染的重要工具，是一种3D绘图协议，这种绘图技术标准允许把JavaScript和OpenGL ES 2.0结合在一起，通过增加OpenGL ES 2.0的一个JavaScript绑定，WebGL可以为HTML5 Canvas提供硬件3D加速渲染，这样就可以借助系统显卡来在浏览器里更流畅地展示3D场景和模型了，还能创建复杂的导航和数据视觉化。WebGL技术标准免去了开发网页专用渲染插件的麻烦，可被用于创建具有复杂3D结构的网站页面，给现有的图像渲染带来了极大的便利。

[0004] 但是现有的渲染方法，都存在渲染效果不够高效的问题；在渲染较低数量级的渲染对象时，尚可保持一定的流畅性，但是在同时渲染千量级或更多的2d渲染对象时就会开始严重掉帧，在需要对视口画面整体缩放移动时，大部分框架存在较严重卡顿的问题。

[0005] 因此，迫切需要一种渲染效率高的前端图像渲染方法及装置。

发明内容

[0006] 本发明解决的问题是如何高效地进行前端二维图像的渲染。

[0007] 为解决上述问题，本发明实施例首先提供一种前端图像渲染方法，其包括：

[0008] 获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象；

[0009] 根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；每个所述对象类的信息中包含一个顶点信息和多个可变信息，所述多个可变信息与所述对象类中的多个绘制对象一一对应；

[0010] 将所述对象类的信息传输至渲染器；

[0011] 所述渲染器根据所述对象类的信息生成渲染后的帧图像。

[0012] 这样，将相同的顶点信息进行合并，这样，可以将该帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数降低到最少(原本十万量级的次数降低到10量级)，从而大大增加图像渲染的效率(信息传输的最小化带来的是绘制效率的提升)。

[0013] 可选的，所述绘制对象包括顶点信息和可变信息，所述可变信息包括偏移属性、缩放比、颜色信息、材质信息和边长信息中的至少一项。

[0014] 可选的，在所述根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类之前，所述方法还包括：

[0015] 开辟存储空间，所述存储空间用于分别存储所述顶点信息和所述可变信息；

[0016] 通过开辟存储空间，可以使得绘制多个图形的过程中不会重复发生多次内存(存储空间)的申请及销毁，大幅提升绘制效率，降低绘制过程中内存回收导致的卡顿问题。

[0017] 可选的，所述存储空间的存储量大于所述绘制对象的数据量。这样可以便于后期进行数据的更新操作。

[0018] 可选的，所述前端图像渲染方法还包括：

[0019] 接收到帧图像的更新指令、平移指令或缩放指令后，根据所述更新指令、平移指令或缩放指令获取更新信息；

[0020] 将所述更新信息传输至渲染器；

[0021] 所述渲染器根据所述更新信息生成渲染后的帧图像。

[0022] 这样，通过更新信息，可以使得在程序执行环境中仅传输需要更新的对象类的信息到渲染器，未发生改变的对象类的信息则不再传输，从而大大减少需要传输的数据量，避免多量级的帧图像渲染处理时出现掉帧和卡顿现象。

[0023] 可选的，所述将所述更新信息传输至渲染器，之前还包括：

[0024] 根据所述更新信息对存储在所述存储空间内的所述对象类的信息进行更新。

[0025] 通过更新信息，直接对存储在所述存储空间内的所述对象类的信息进行更新，可以最大化利用存储空间，避免重复发生多次内存(存储空间)的申请及销毁，大幅提升绘制效率，降低绘制过程中内存回收导致的卡顿问题。

[0026] 可选的，所述根据所述更新指令、平移指令或缩放指令获取更新信息中，在接收到帧图像的所述平移指令或缩放指令后，获取的所述更新信息用于更新视口矩阵。

[0027] 可选的，所述对象类的信息还包括视口矩阵，所述视口矩阵与所述顶点信息、所述可变信息相互独立存储。

[0028] 可选的，所述根据所述更新信息对存储在所述存储空间内的所述对象类的信息进行更新中，在所述更新信息用于更新视口矩阵时，根据所述更新信息对所述对象类的信息中的视口矩阵进行更新。

[0029] 通过增加视口矩阵属性，可以描述可见内容(帧图像)的平移或缩放，从而单独传输更新的视口矩阵，即可完成对平移指令或缩放指令的数据传输操作，大大减少了需要传输的数据量，从而将帧图像的绘制(或渲染)过程中JS到WebGL的传输数据进一步降低，增加图像渲染的效率。

[0030] 其次，本发明实施例还提供了一种前端图像渲染装置，其包括：

[0031] 获取单元，用于获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象；

[0032] 合并单元，用于根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；每个所述对象类的信息中包含一个顶点信息和多个可变信息，所述多个可变信息与所述对象类中的多个绘制对象一一对应。

[0033] 传输单元，用于将所述对象类的信息传输至渲染器；

[0034] 渲染器，用于所述渲染器根据所述对象类的信息生成渲染后的帧图像。

[0035] 这样，将相同的顶点信息进行合并，这样，可以将该帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数降低到最少(原本十万量级的次数降低到10量级)，从而大大增加图像渲染的效率(信息传输的最小化带来的是绘制效率的提升)。

[0036] 另外，本发明实施例还提供了一种电子设备，包括处理器以及存储器，所述存储器

存储有控制程序,所述控制程序被处理器执行时实现上述所述的前端图像渲染方法。

[0037] 另外,本发明实施例还提供了一种计算机可读存储介质,存储有指令,所述指令被处理器加载并执行时实现上述所述的前端图像渲染方法。

附图说明

[0038] 图1为根据本发明一个实施例的前端图像渲染方法的流程图;

[0039] 图2为根据本发明另一实施例的前端图像渲染方法的流程图;

[0040] 图3为根据本发明再一实施例的前端图像渲染方法的流程图;

[0041] 图4为根据本发明又一实施例的前端图像渲染方法的流程图;

[0042] 图5为根据本发明一个实施例的前端图像渲染装置的结构框图;

[0043] 图6为根据本发明另一实施例的前端图像渲染装置的结构框图;

[0044] 图7为根据本发明实施例的一种电子设备的结构框图;

[0045] 图8为根据本发明实施例的另一种电子设备的框图。

[0046] 附图标记说明:

[0047] 1-获取单元,2-合并单元,3-传输单元,4-渲染器,5-存储单元,12-电子设备,14-外部设备,16-处理单元,18-总线,20-网络适配器,22-输入/输出(I/O)接口,24-显示器,28-系统存储器,30-随机存取存储器,32-高速缓存存储器,34-存储系统,40-实用工具,42-程序模块。

具体实施方式

[0048] 为使本发明的上述目的、特征和优点能够更为明显易懂,下面结合附图对本发明的具体实施例做详细的说明。

[0049] 显然,所说明的实施例是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域技术人员在没有做出创造性劳动的前提下获得的所有其他实施例,都属于本发明保护的范围。

[0050] 为了便于理解,在本发明中,需要对其中的技术问题进行详细阐述。

[0051] 在现有的WebGL中,一般设置有attribute变量、uniform变量、uniform变量;attribute变量代表的是与顶点相关的数据,其存储了从外部传输进顶点着色器的数据。在着色器中定义好attribute变量之后,还需要通过JS与shader进行交互。通过WebGL的渲染上下文变量,可以获取attribute变量的存储地址,获取地址之后,就可以由JS向attribute变量传送绘制点的顶点位置。uniform变量,表示的是JavaScript程序向顶点着色器和片元着色器传输的一致的(不变的)数据;也就是是说这种变量既可以用在顶点着色器也可以用于片元着色器。与attribute变量类似,uniform变量也是先获取其地址,然后向其传值。varying变量,其表示的是从顶点着色器流向片元着色器可变的变量。

[0052] 在此,需要对绘制对象进行说明,绘制对象是JS层抽象的最小粒度的单元,其包含的信息包括:用于描述图形形状信息的顶点信息、用于描述图形位置信息的偏移属性、缩放比、用于描述图形三维信息从数据到具体呈现过程中的三维变换的矩阵的视口矩阵、颜色信息、材质信息、边长信息等。

[0053] 这样,现有的WebGL中,会先将绘制对象信息存储在临时开辟的存储空间内,然后

该存储空间将存储的信息传输给着色器;发明人发现,现有技术中,同样形状的绘制对象或绘制单元(需要明确的是,同样形状的绘制对象,其顶点信息是相同的),分别存储各自的顶点信息,并在渲染时分别将各自的顶点信息传输到渲染器。该传输方式,在需要处理的帧图像或渲染对象较少时,对整个渲染方式的影响并不大,但是在同时渲染千量级或更多的2d渲染对象时,该种传输方式会使得需要传输的数据量大大增加,从而导致传输不及时,造成掉帧和卡顿的问题。

[0054] 本公开实施例提供了一种前端图像渲染方法,该方法可以由前端图像渲染装置来执行,该前端图像渲染装置可以集成在手机、笔记本、服务器、PAD等电子设备中。如图1所示,其为根据本发明实施例的前端图像渲染方法的流程图;其中,所述前端图像渲染方法,包括:

[0055] 步骤10,获取待渲染的帧图像,并将其拆分为最小粒度的绘制对象;

[0056] 其中,所述待渲染的帧图像,至少包括图像模型和渲染信息,也即是说,该待渲染的帧图像,其具体信息包括生成真正渲染图像所需的必要信息,也即是说在获取该待渲染的帧图像后,通过拆分、变换、处理后可以直接由渲染器生成渲染图像,而不需要再额外获取其他的必要信息。该待渲染的帧图像,其存储形式可以为任何形式,如数据矩阵、灰白图像、或数据流等任何形式。

[0057] 本步骤中,该最小粒度,是指在JS层抽象的最小粒度。拆分出的绘制对象,至少包括顶点信息、偏移属性、缩放比、颜色信息、材质信息和边长信息中的至少一项。

[0058] 步骤20,根据所述绘制对象的顶点信息,将相同顶点信息的多个所述绘制对象合并为一个对象类;每个所述对象类的信息中包含一个顶点信息和多个可变信息,所述多个可变信息与所述对象类中的多个绘制对象一一对应;

[0059] 所述绘制对象的顶点信息,用于描述绘制对象的图形形状信息;相同形状的绘制对象,其顶点信息相同。

[0060] 所述绘制对象包括顶点信息和可变信息,所述可变信息包括偏移属性、缩放比、颜色信息、材质信息和边长信息中的至少一项。这样,将绘制对象的信息划分为顶点信息和可变信息,通过顶点信息将相同形状的绘制对象合并为一个对象类,这样,每个对象类的信息(对象类的信息)中包含一个顶点信息(也即是将所有形状相同的绘制对象的顶点信息合并为一个)和多个可变信息(可变信息的数量与合并的绘制对象的数量相同);这样,用于表征N个绘制对象的信息,就由N个顶点信息和N个可变信息,转换成了一个顶点信息和N个可变信息。

[0061] 步骤40,将所述对象类的信息传输至渲染器;

[0062] 所述对象类的信息,即为合并后的对象类包含的信息;通过将绘制对象合并为对象类,用于表征N个绘制对象的信息,就由N个顶点信息和N个可变信息,转换成了一个顶点信息和N个可变信息;这样,通过将对象类的信息传输至渲染器,就由传输N个顶点信息和N个可变信息,转换成了传输一个顶点信息和N个可变信息,从而大大减少了传输的信息量。

[0063] 其中,该传输是由JS代码环境向WebGL渲染器进行信息传输。

[0064] 步骤50,所述渲染器根据所述对象类的信息生成渲染后的帧图像。

[0065] 在此,需要说明的是,webgl是一个浏览器标准的渲染引擎,本步骤中的渲染器可以为webgl渲染器。Webgl渲染引擎由于其本身的设计问题,其中顶点信息的传输,对该渲染

引擎的渲染效率有很大的影响;也即是说,需要传输不同量级的顶点信息,对渲染引擎的渲染效率的影响与顶点信息的量级并不成正比,而是呈指数性增长。

[0066] 通过步骤10-50,将相同的顶点信息进行合并,这样,可以将该帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数降低到最少(原本十万量级的次数降低到10量级),从而大大增加图像渲染的效率(信息传输的最小化带来的是绘制效率的提升)。

[0067] 可选的,如图2所示,在所述步骤20,根据所述绘制对象的顶点信息,将相同顶点信息的多个所述绘制对象合并为一个对象类之前,所述方法还包括:

[0068] 步骤30,开辟存储空间,所述存储空间用于分别存储所述顶点信息和所述可变信息;

[0069] 该存储空间,可以为预先开辟的,也可以合并对象类后开辟的,该存储空间,是在JS代码环境(代码环境中的内存)中开辟的。

[0070] 需要说明的是,现有的渲染方法中,每绘制一次图形,就会向代码环境中的内存请求一次存储空间,每当一个绘制图形被删掉时,就会将该图形占用的存储空间销毁并回收。

[0071] 通过开辟存储空间,可以使得绘制多个图形的过程中不会重复发生多次内存(存储空间)的申请及销毁,大幅提升绘制效率,降低绘制过程中内存回收导致的卡顿问题。

[0072] 其中,所述分别存储所述顶点信息和所述可变信息,是指将绘制对象(对象类)的顶点信息和可变信息分开存储。

[0073] 需要说明的是,现有的渲染方法中,是将绘制对象的信息存储在一起,一起存储,一起传输;这样,一旦存储在一起的数据产生变化,就需要重新传输;本步骤中,是将顶点信息和可变信息分开存储,分开传输;这样,由于顶点信息一般不改变,从而可以避免可变数据产生变化导致的顶点信息重新传输,从而将帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数进一步降低,增加图像渲染的效率。

[0074] 其中,所述存储空间的存储量大于所述绘制对象的数据量。这样可以便于后期进行数据的更新操作。

[0075] 可选的,如图3所示,所述前端图像渲染方法还包括:

[0076] 步骤60,接收到帧图像的更新指令、平移指令或缩放指令后,根据所述更新指令、平移指令或缩放指令获取更新信息;

[0077] 所述更新指令,是指可见内容(帧图像)中各个绘制对象的可变信息变更产生的指令,也即是表达可见内容(帧图像)的各个绘制对象的可变信息变更的指令;所述平移指令,是指可见内容(帧图像)的平移产生的指令,也即是表达可见内容(帧图像)的平移的指令;所述缩放指令,是指可见内容(帧图像)的缩小或放大产生的指令,也即是表达可见内容(帧图像)的缩小或放大的指令;需要说明的是,对于待渲染的帧图像中的各个绘制对象而言,对帧图像进行平移,意味着对该帧图像中的各个绘制对象进行同样的平移操作,对帧图像进行缩放,意味着对该帧图像中的各个绘制对象进行同样的缩小或放大操作。

[0078] 可选的,所述对象类的信息还包括视口矩阵,所述视口矩阵与所述顶点信息、所述可变信息相互独立存储。

[0079] 其中,所述视口矩阵为用于描述图形三维信息从数据到具体呈现过程中的三维变换的矩阵。这样,通过偏移属性、缩放比和视口矩阵,可以对绘制对象进行定位(在结合顶点信息的情况下)。

[0080] 需要说明的是，由于平移和缩放操作对帧图像中的各个绘制对象而言，会产生同样的操作，反应到所述对象类的信息中，即是仅改变视口矩阵就可以描述该同样的操作；也即是说，在对象类的信息还包括视口矩阵的情况下，若发生帧图像的平移或缩放，则仅仅需要更新对象类的信息中的视口矩阵即可。

[0081] 现有技术中对视口处理(即可见内容的缩放移动)时，需要批量修改画面中所有绘制对象的位置属性、缩放比属性，然后进行重绘，需要进行大量的数据操作；本步骤中，是将所述视口矩阵与所述顶点信息、所述可变信息分开存储，分开传输；这样，对于平移或缩放操作，由于顶点信息和可变信息一般不改变，从而可以避免视口矩阵产生变化导致的顶点信息和可变信息重新传输，从而将帧图像的绘制(或渲染)过程中JS到WebGL的传输数据进一步降低，增加图像渲染的效率。

[0082] 其中，根据接收到的指令的不同，获取的所述更新信息也不相同；在接收到帧图像的所述更新指令后，获取的所述更新信息用于更新可变信息；在接收到帧图像的所述平移指令或缩放指令后，获取的所述更新信息用于更新视口矩阵。

[0083] 步骤80，将所述更新信息传输至渲染器；

[0084] 所述更新信息，是与原有的对象类的信息相比较，发生改变的信息；仅传输更新信息，即是仅传输发生改变的对象类的信息，未发生改变的对象类的信息则不再传输。

[0085] 现有的渲染方法中，一旦绘制对象的可变信息或其他信息发生改变，就会重新传输整个绘制对象的顶点信息和可变信息(以及绘制对象除顶点信息和可变信息之外的其他信息)，这就需要传输大量的数据，导致多帧处理时出现掉帧和卡顿现象。

[0086] 步骤90，所述渲染器根据所述更新信息生成渲染后的帧图像。

[0087] 这样，通过更新信息，可以使得在程序执行环境中仅传输需要更新的对象类的信息到渲染器，未发生改变的对象类的信息则不再传输，从而大大减少需要传输的数据量，避免多量级的帧图像渲染处理时出现掉帧和卡顿现象。

[0088] 另外，通过增加视口矩阵属性，可以描述可见内容(帧图像)的平移或缩放，从而单独传输更新的视口矩阵，即可完成对平移指令或缩放指令的数据传输操作，大大减少了需要传输的数据量，从而将帧图像的绘制(或渲染)过程中JS到WebGL的传输数据进一步降低，增加图像渲染的效率。

[0089] 可选的，如图4所示，所述步骤80，将所述更新信息传输至渲染器，之前还包括：

[0090] 步骤70，根据所述更新信息对存储在所述存储空间内的所述对象类的信息进行更新。

[0091] 在所述更新信息用于更新可变信息时，根据所述更新信息对所述对象类的信息中的可变信息进行更新；在所述更新信息用于更新视口矩阵时，根据所述更新信息对所述对象类的信息中的视口矩阵进行更新。

[0092] 其中，对可变信息进行更新时，可以根据实际操作需求仅更新可变信息(偏移属性、缩放比、颜色信息、材质信息和边长信息)中的其中一项，也可以是更新整个可变信息。

[0093] 该存储空间，可以为预先开辟的，也可以合并对象类后开辟的，该存储空间，是在JS代码环境(代码环境中的内存)中开辟的。

[0094] 通过更新信息，直接对存储在所述存储空间内的所述对象类的信息进行更新，可以最大化利用存储空间，避免重复发生多次内存(存储空间)的申请及销毁，大幅提升绘制

效率,降低绘制过程中内存回收导致的卡顿问题。

[0095] 通过上述前端图像渲染方法,可以最终在8GB内存、1.5GB显存集显的计算机中实现十万级别2D图形对象的60fps绘制,视口缩放移动均不会发生掉帧。

[0096] 本公开实施例提供了一种前端图像渲染装置,用于执行本发明上述内容所述的前端图像渲染方法,以下对所述前端图像渲染装置进行详细描述。

[0097] 如图5所示,一种前端图像渲染装置,包括:

[0098] 获取单元1,用于获取待渲染的帧图像,并将其拆分为最小粒度的绘制对象;

[0099] 合并单元2,用于根据所述绘制对象的顶点信息,将相同顶点信息的多个所述绘制对象合并为一个对象类;每个所述对象类的信息中包含一个顶点信息和多个可变信息,所述多个可变信息与所述对象类中的多个绘制对象一一对应。

[0100] 传输单元3,用于将所述对象类的信息传输至渲染器;

[0101] 渲染器4,用于根据所述对象类的信息生成渲染后的帧图像。

[0102] 这样,将相同的顶点信息进行合并,这样,可以将该帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数降低到最少(原本十万量级的次数降低到10量级),从而大大增加图像渲染的效率(信息传输的最小化带来的是绘制效率的提升)。

[0103] 可选的,所述绘制对象包括顶点信息和可变信息,所述可变信息包括偏移属性、缩放比、颜色信息、材质信息和边长信息中的至少一项。

[0104] 可选的,如图6所示,所述前端图像渲染装置还包括:

[0105] 存储单元5,用于开辟存储空间,所述存储空间用于分别存储所述顶点信息和所述可变信息。

[0106] 可选的,所述存储空间的存储量大于所述绘制对象的数据量。

[0107] 可选的,所述获取单元1还用于:接收到帧图像的更新指令、平移指令或缩放指令后,根据所述更新指令、平移指令或缩放指令获取更新信息;

[0108] 所述传输单元3还用于:将所述更新信息传输至渲染器;

[0109] 所述渲染器4根据所述更新信息生成渲染后的帧图像。

[0110] 可选的,所述存储单元5还用于:根据所述更新信息对存储在所述存储空间内的所述对象类的信息进行更新。

[0111] 可选的,所述获取单元1还用于:在接收到帧图像的所述平移指令或缩放指令后,获取的所述更新信息用于更新视口矩阵。

[0112] 可选的,所述对象类的信息还包括视口矩阵,所述视口矩阵与所述顶点信息、所述可变信息相互独立存储。

[0113] 可选的,所述存储单元5还用于:在所述更新信息用于更新视口矩阵时,根据所述更新信息对所述对象类的信息中的视口矩阵进行更新。

[0114] 以上描述了前端图像渲染装置的内部功能和结构,如图7所示,实际中,该前端图像渲染装置可实现为电子设备,包括:处理器以及存储器,所述存储器存储有控制程序,所述控制程序被处理器执行时实现上述所述的前端图像渲染方法。

[0115] 这样,将相同的顶点信息进行合并,这样,可以将该帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数降低到最少(原本十万量级的次数降低到10量级),从而大大增加图像渲染的效率(信息传输的最小化带来的是绘制效率的提升)。

[0116] 图8是根据本发明实施例示出的另一种电子设备的框图。图8显示的电子设备12仅仅是一个示例，不应对本申请实施例的功能和使用范围带来任何限制。

[0117] 如图8所示，电子设备12可以通用电子设备的形式实现。电子设备12的组件可以包括但不限于：一个或者多个处理器或者处理单元16，系统存储器28，连接不同系统组件(包括系统存储器28和处理单元16)的总线18。

[0118] 总线18表示几类总线结构中的一种或多种，包括存储器总线或者存储器控制器，外围总线，图形加速端口，处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说，这些体系结构包括但不限于工业标准体系结构(Industry Standard Architecture；以下简称：ISA)总线，微通道体系结构(Micro Channel Architecture；以下简称：MAC)总线，增强型ISA总线、视频电子标准协会(Video Electronics Standards Association；以下简称：VESA)局域总线以及外围组件互连(Peripheral Component Interconnection；以下简称：PCI)总线。

[0119] 电子设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被电子设备12访问的可用介质，包括易失性和非易失性介质，可移动的和不可移动的介质。

[0120] 存储器28可以包括易失性存储器形式的计算机系统可读介质，例如随机存取存储器(Random Access Memory；以下简称：RAM)30和/或高速缓存存储器32。电子设备12可以进一步包括其它可移动/不可移动的、易失性/非易失性的计算机可读存储介质。仅作为举例，存储系统34可以用于读写不可移动的、非易失性磁介质(图中未显示，通常称为“硬盘驱动器”)。尽管图8中未示出，可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器，以及对可移动非易失性光盘(例如：光盘只读存储器(Compact Disc Read Only Memory；以下简称：CD-ROM)、数字多功能只读光盘(Digital Video Disc Read Only Memory；以下简称：DVD-ROM)或者其它光介质)读写的光盘驱动器。在这些情况下，每个驱动器可以通过一个或者多个数据介质接口与总线18相连。存储器28可以包括至少一个程序产品，该程序产品具有一组(例如至少一个)程序模块，这些程序模块被配置以执行本申请各实施例的功能。

[0121] 具有一组(至少一个)程序模块42的程序/实用工具40，可以存储在例如存储器28中，这样的程序模块42包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本申请所描述的实施例中的功能和/或方法。

[0122] 电子设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信，还可与一个或者多个使得用户能与该计算机系统/服务器12交互的设备通信，和/或与使得该计算机系统/服务器12能与一个或多个其它电子设备进行通信的任何设备(例如网卡，调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口22进行。并且，电子设备12还可以通过网络适配器20与一个或者多个网络(例如局域网(Local Area Network；以下简称：LAN)，广域网(Wide Area Network；以下简称：WAN)和/或公共网络，例如因特网)通信。如图所示，网络适配器20通过总线18与电子设备12的其它模块通信。要说明的是，尽管图中未示出，可以结合电子设备12使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0123] 处理单元16通过运行存储在系统存储器28中的程序，从而执行各种功能应用以及数据处理，例如实现前述实施例中提及的方法。

[0124] 本发明的电子设备可以是服务器，也可以有限算力的终端设备，本发明的轻量级网络结构尤其适用于后者。所述终端设备的基体实现包括但不限于：智能移动通信终端、无人机、机器人、便携式图像处理设备、安防设备等等。本公开实施例提供了一种计算机可读存储介质，存储有指令，所述指令被处理器加载并执行时实现上述所述的前端图像渲染方法。

[0125] 这样，将相同的顶点信息进行合并，这样，可以将该帧图像的绘制(或渲染)过程中JS到WebGL的顶点信息传输次数降低到最少(原本十万量级的次数降低到10量级)，从而大大增加图像渲染的效率(信息传输的最小化带来的是绘制效率的提升)。

[0126] 本发明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)或processor(处理器)执行本发明实施例所述方法的全部或部分步骤S。而前述的存储介质包括：U盘、移动硬盘、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0127] 虽然本公开披露如上，但本公开的保护范围并非仅限于此。本领域技术人员在不脱离本公开的精神和范围的前提下，可进行各种变更与修改，这些变更与修改均将落入本发明的保护范围。

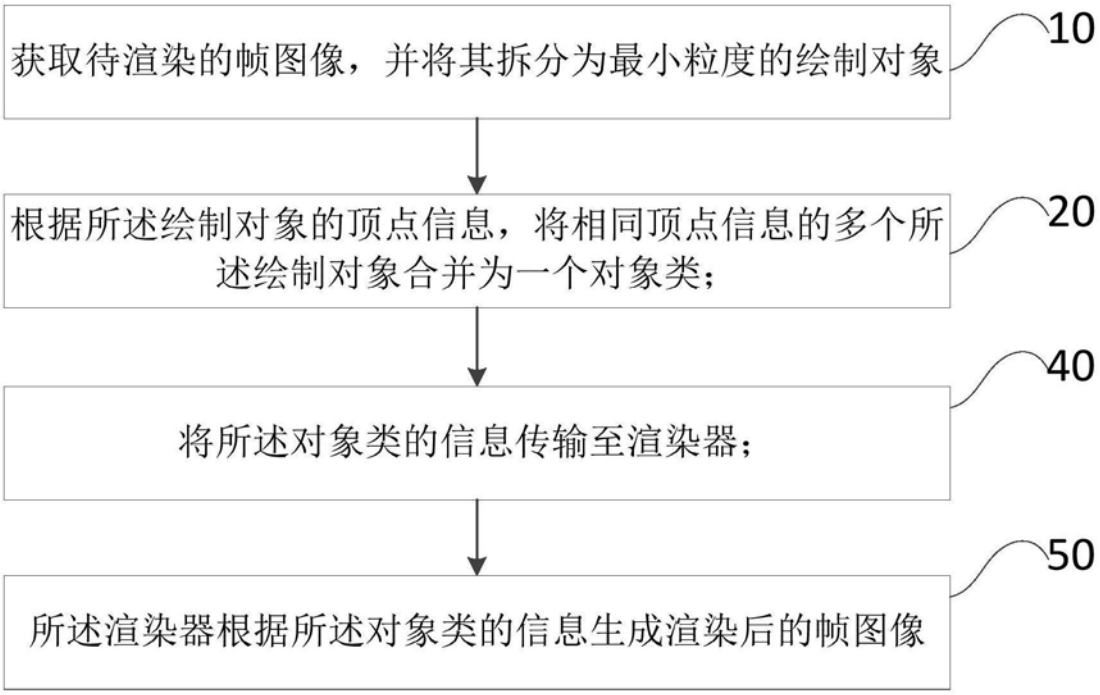

获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象
10
根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；
20
将所述对象类的信息传输至渲染器；
40
所述渲染器根据所述对象类的信息生成渲染后的帧图像
50

图1

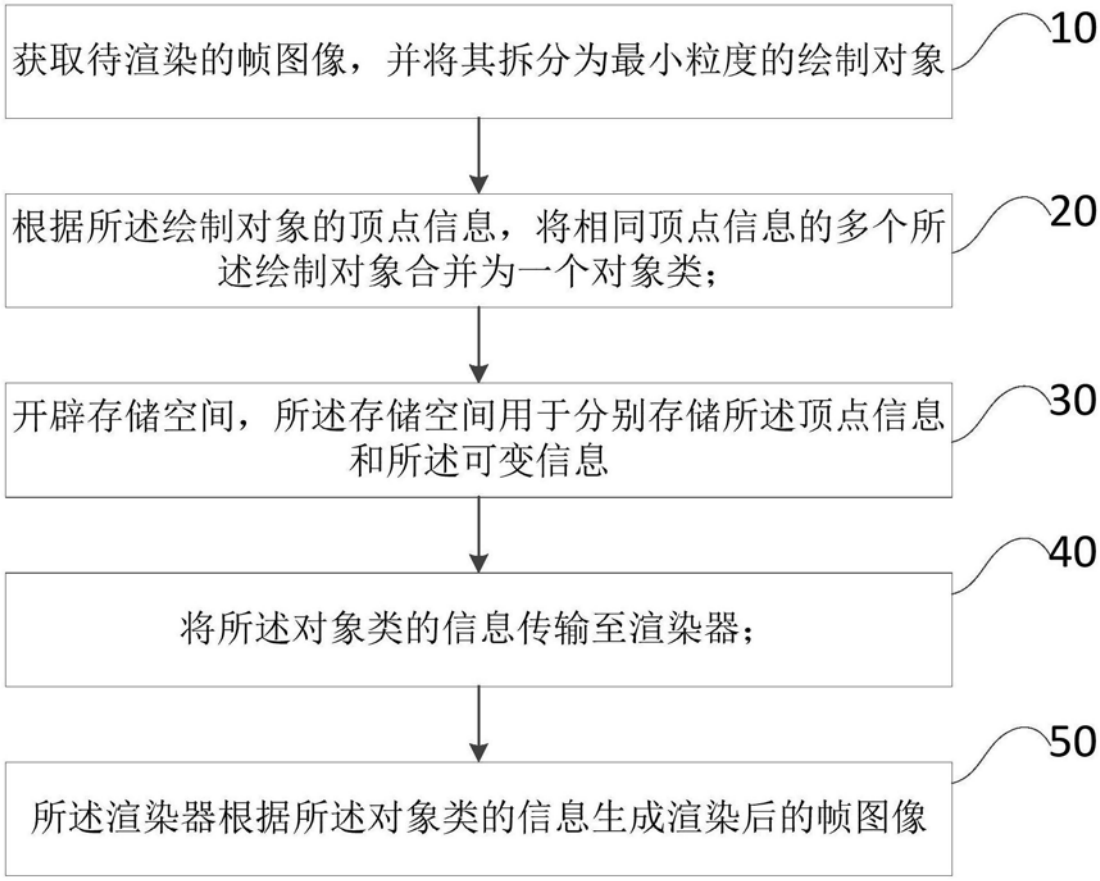

获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象
10
根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；
20
开辟存储空间，所述存储空间用于分别存储所述顶点信息和所述可变信息
30
将所述对象类的信息传输至渲染器；
40
所述渲染器根据所述对象类的信息生成渲染后的帧图像
50

图2

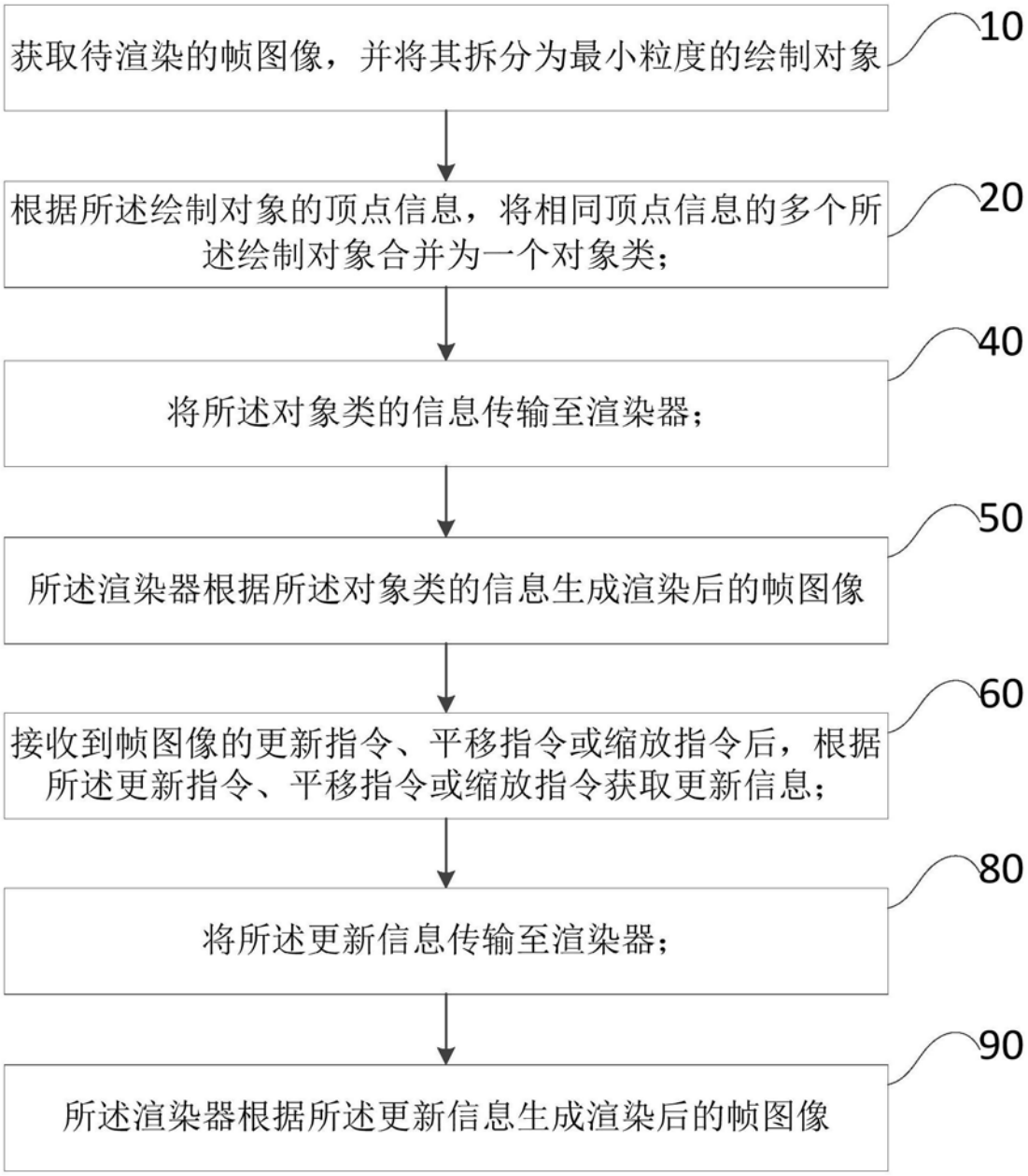

获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象
10
根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；
20
将所述对象类的信息传输至渲染器；
40
所述渲染器根据所述对象类的信息生成渲染后的帧图像
50
接收到帧图像的更新指令、平移指令或缩放指令后，根据所述更新指令、平移指令或缩放指令获取更新信息；
60
将所述更新信息传输至渲染器；
80
所述渲染器根据所述更新信息生成渲染后的帧图像
90

图3

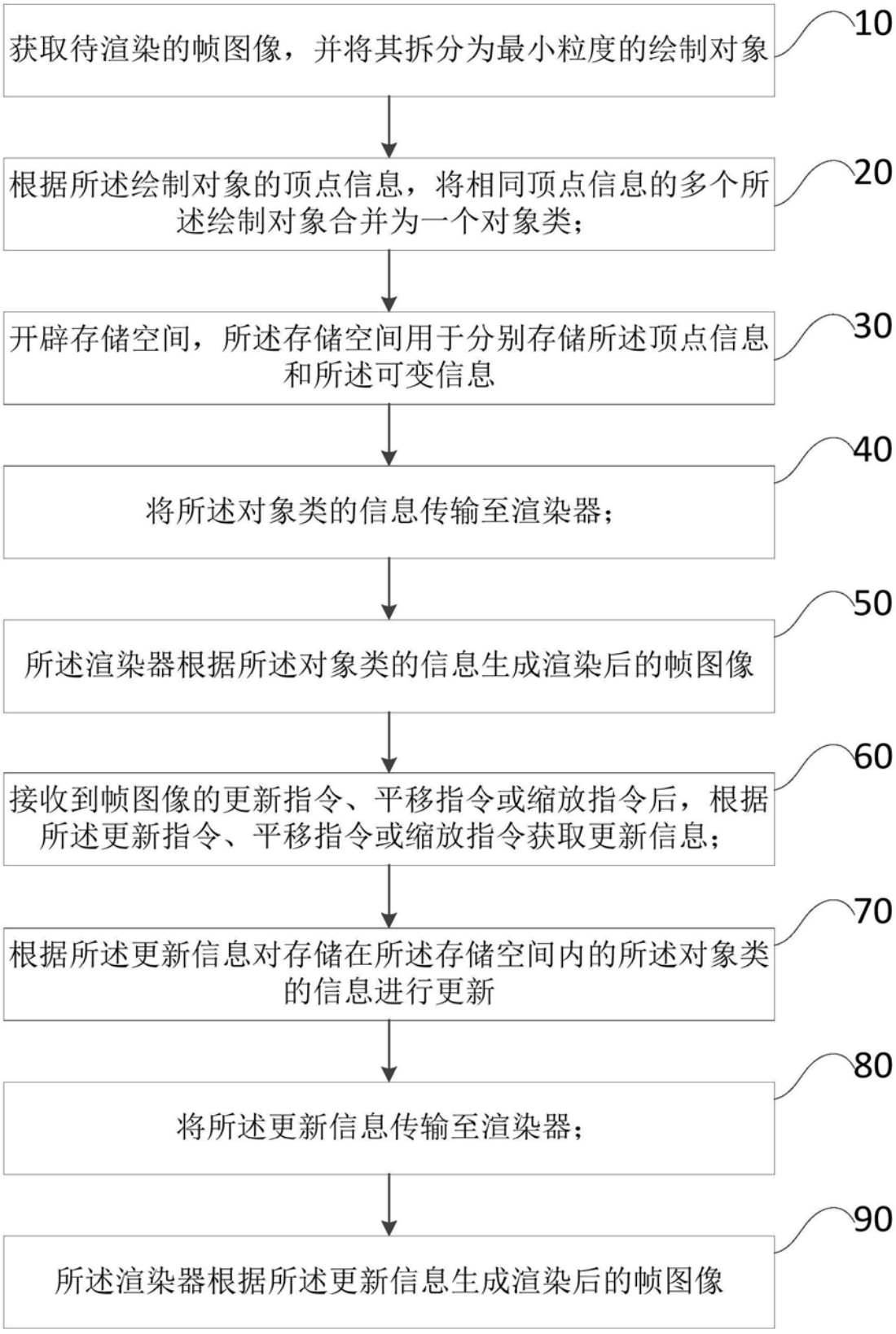

获取待渲染的帧图像，并将其拆分为最小粒度的绘制对象
10
根据所述绘制对象的顶点信息，将相同顶点信息的多个所述绘制对象合并为一个对象类；
20
开辟存储空间，所述存储空间用于分别存储所述顶点信息和所述可变信息
30
将所述对象类的信息传输至渲染器；
40
所述渲染器根据所述对象类的信息生成渲染后的帧图像
50
接收到帧图像的更新指令、平移指令或缩放指令后，根据所述更新指令、平移指令或缩放指令获取更新信息；
60
根据所述更新信息对存储在所述存储空间内的所述对象类的信息进行更新
70
将所述更新信息传输至渲染器；
80
所述渲染器根据所述更新信息生成渲染后的帧图像
90

图4

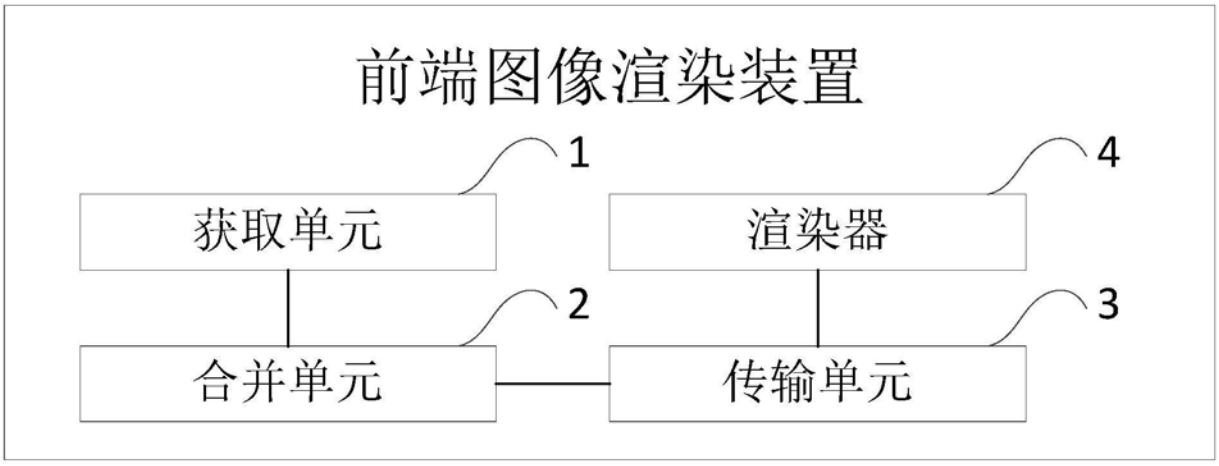

前端图像渲染装置
1
获取单元
4
渲染器
2
合并单元
3
传输单元

图5

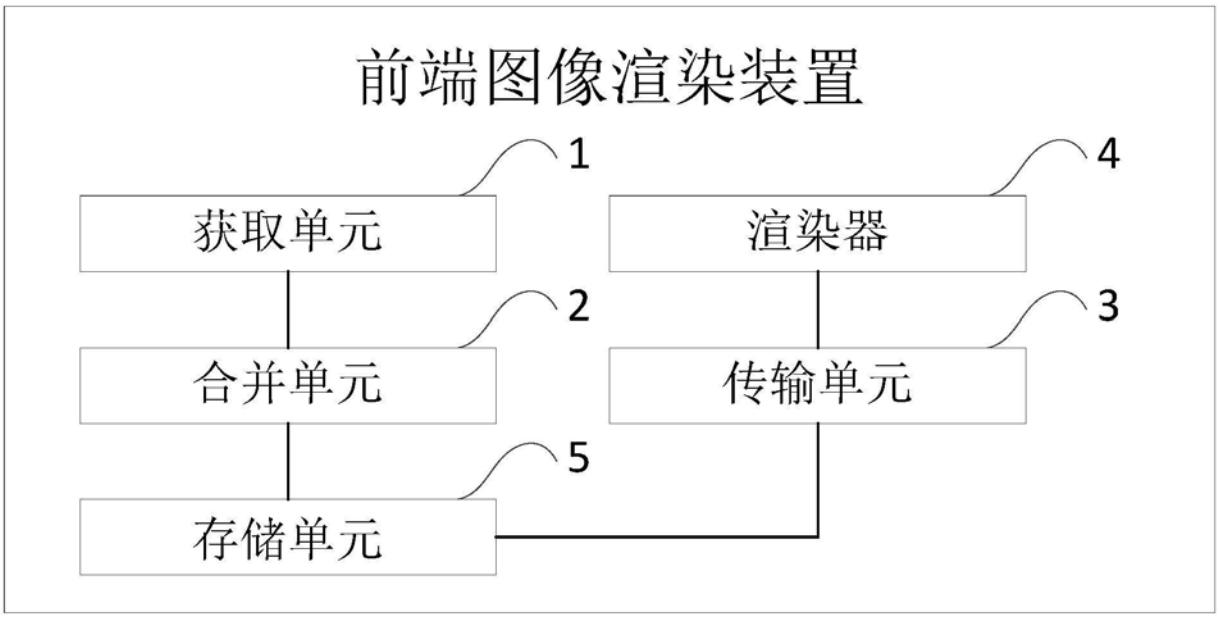

前端图像渲染装置
1
获取单元
4
渲染器
2
合并单元
3
传输单元
5
存储单元

图6


处理器
存储器

图7

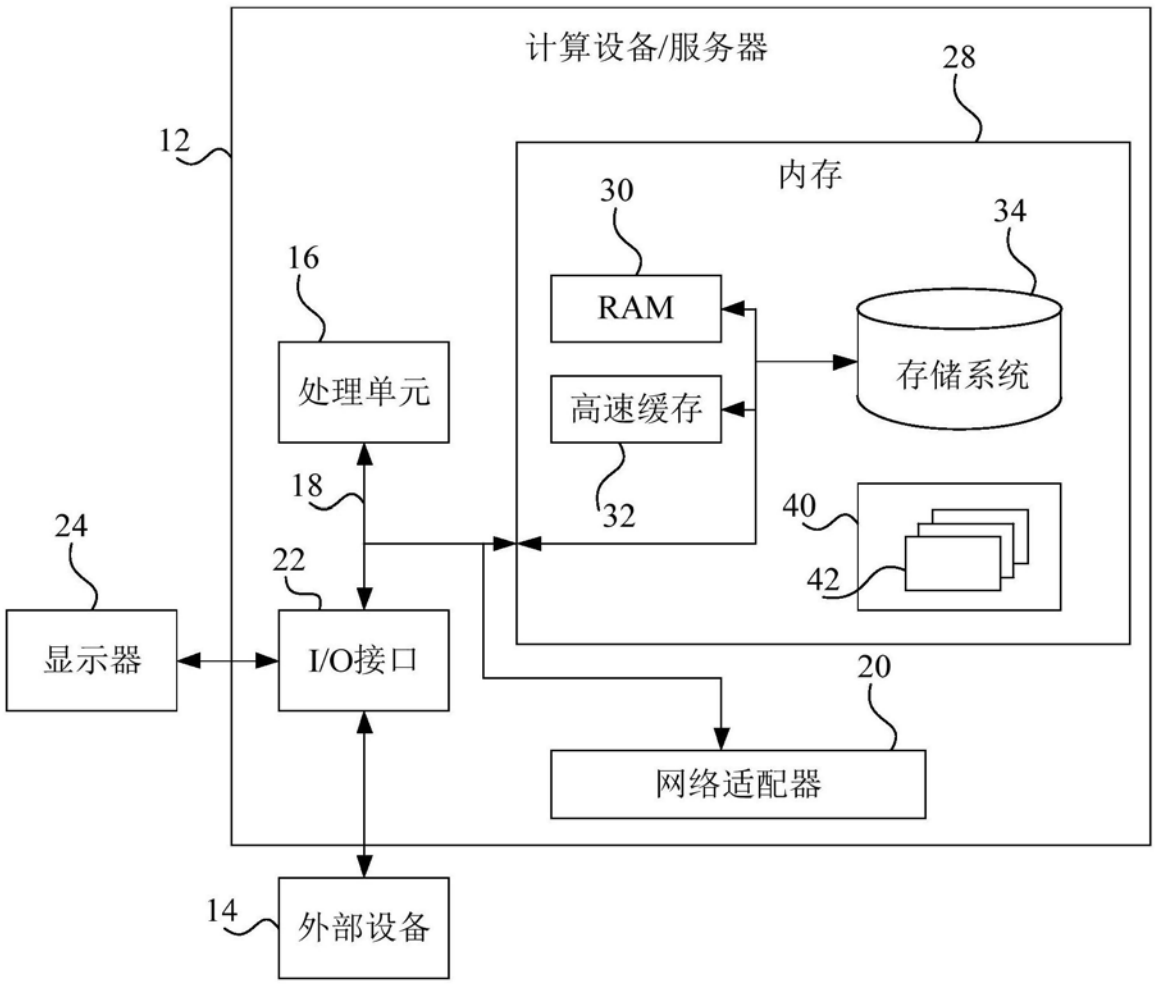

计算设备/服务器
28
12
内存
30
34
16
RAM
处理单元
存储系统
高速缓存
18
32
40
24
42
22
显示器
I/O接口
20
网络适配器
14
外部设备

图8