



(12) 发明专利

(10) 授权公告号 CN 113538648 B

(45) 授权公告日 2024.04.30

(21) 申请号 202110853251.4

(22) 申请日 2021.07.27

(65) 同一申请的已公布的文献号
申请公布号 CN 113538648 A

(43) 申请公布日 2021.10.22

(73) 专利权人 歌尔科技有限公司
地址 266104 山东省青岛市崂山区松岭路
500号

(72) 发明人 邱绪东

(74) 专利代理机构 深圳市世纪恒程知识产权代
理事务所 44287
专利代理师 晏波

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

(56) 对比文件
CN 109887065 A, 2019.06.14
CN 112347408 A, 2021.02.09
US 2020279533 A1, 2020.09.03
WO 2015196791 A1, 2015.12.30

WO 2018086295 A1, 2018.05.17

WO 2020060984 A1, 2020.03.26

WO 2020062052 A1, 2020.04.02

WO 2021001883 A1, 2021.01.07

CN 108921951 A, 2018.11.30

CN 111652962 A, 2020.09.11

US 2019043448 A1, 2019.02.07

US 2020105227 A1, 2020.04.02

CN 112230776 A, 2021.01.15

US 2008192060 A1, 2008.08.14

Xiaofen Li et al..Multi-channel Scene Synchronization Strategy in Real Time Rendering .Transaction on Edutainment XV.2019,246-255.

Tobias Ritschel et al..Perceptual Rasterization for Head-mounted Display Image Synthesis.ACM Transactions on Graphics.2019,1-14.

陈学亮 等.基于5G网络的虚拟现实优化方案研究.广东通信技术.2018,1-6. (续)

审查员 杨华

权利要求书2页 说明书13页 附图3页

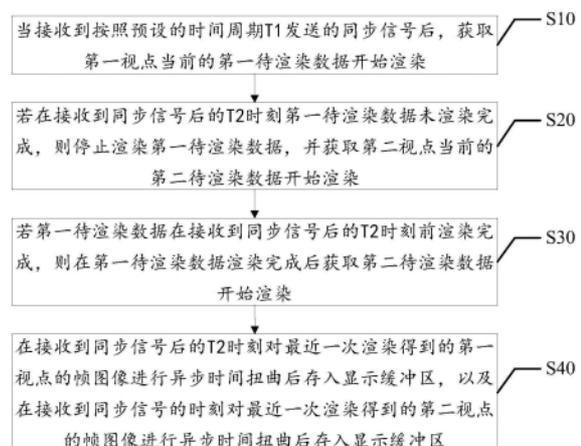
(54) 发明名称

图像渲染方法、装置、设备及计算机可读存储介质

(57) 摘要

本发明公开了一种图像渲染方法、装置、设备及计算机可读存储介质,所述方法包括:当接收到按照时间周期T1发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;若在接收到同步信号后的T2时刻第一待渲染数据未渲染完成,则停止渲染并获取第二视点当前的第二待渲染数据开始渲染;若在接收到同步信号后的T2时刻前渲染完成,则获取第二待渲染数据开始渲染;在接收到同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到同步信号的

进行异步时间扭曲后存入显示缓冲区。本发明避免了GPU资源浪费,并提高了图像渲染使用率。



CN 113538648 B

[接上页]

(56) 对比文件

王涌天 等. 虚拟现实光学显示技术. 中国科

学: 信息科学. 2016, 第46卷 (第12期), 1694-1710.

1. 一种图像渲染方法,其特征在于,所述方法包括以下步骤:

当接收到按照预设的时间周期 $T1$ 发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;

若在接收到所述同步信号后的 $T2$ 时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T2 < T1$;

若所述第一待渲染数据在接收到所述同步信号后的 $T2$ 时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

在接收到所述同步信号后的 $T2$ 时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的時刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区;

最近一次渲染得到的第一视点的帧图像为:对所述第一待渲染数据渲染得到的帧图像,或者,所述第一视点上一次渲染得到的帧图像;

最近一次渲染得到的第二视点的帧图像为:对所述第二待渲染数据渲染得到的帧图像,或者,所述第二视点上一次渲染得到的帧图像。

2. 如权利要求1所述的图像渲染方法,其特征在于,所述当接收到按照预设的时间周期 $T1$ 发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染的步骤包括:

当接收到按照预设的时间周期 $T1$ 发送的同步信号时,检测第二视点是否有正在渲染的第三待渲染数据;

若有,则停止渲染所述第三待渲染数据,并获取第一视点当前的第一待渲染数据开始渲染;

若没有,则获取所述第一待渲染数据开始渲染。

3. 如权利要求1所述的图像渲染方法,其特征在于,所述方法还包括:

在接收到所述同步信号的時刻获取所述显示缓冲区中当前缓存的第一视点的帧图像;

采用所述第一视点帧图像刷新显示设备中当前显示的第一视点的帧图像。

4. 如权利要求1所述的图像渲染方法,其特征在于,所述方法还包括:

在接收到所述同步信号后的 $T2$ 时刻获取所述显示缓冲区中当前缓存的第二视点的帧图像;

采用所述第二视点帧图像刷新显示设备中当前显示的第二视点的帧图像。

5. 如权利要求1至4任一项所述的图像渲染方法,其特征在于,所述方法还包括:

若在接收到所述同步信号后的 $T2$ 时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

当检测到第一视点的丢帧次数达到预设次数时,增大 $T2$,其中,增大后的 $T2$ 小于 $T1$ 。

6. 如权利要求1至4任一项所述的图像渲染方法,其特征在于,所述方法还包括:

若在接收到所述同步信号后的 $T2$ 时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

若在接收到所述同步信号的時刻第二视点有正在渲染的待渲染数据,则对第二视点的丢帧次数累加一次;

当检测到第一视点的丢帧次数与第二视点的丢帧次数的比值大于预设比值时,增大

T2,其中,增大后的T2小于T1;

当检测到第二视点的丢帧次数与第一视点的丢帧次数的比值大于所述预设比值时,减小T2,其中,减小后的T2大于0。

7.如权利要求1至4任一项所述的图像渲染方法,其特征在于,所述方法还包括:

当第一次接收到所述同步信号时,将第一视点和第二视点中其中一个设置为左眼视点,另一个设置为右眼视点;

从第一次接收到所述同步信号起每隔预设的时间周期T3将第一视点和第二视点的左右眼视点设置进行互换。

8.一种图像渲染装置,其特征在于,所述装置包括:

第一渲染模块,用于当接收到按照预设的时间周期T1发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;

第二渲染模块,用于若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T2 < T1$;

所述第二渲染模块还用于若所述第一待渲染数据在接收到所述同步信号后的T2时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

缓存模块,用于在接收到所述同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区;

最近一次渲染得到的第一视点的帧图像为:对所述第一待渲染数据渲染得到的帧图像,或者,所述第一视点上一次渲染得到的帧图像;

最近一次渲染得到的第二视点的帧图像为:对所述第二待渲染数据渲染得到的帧图像,或者,所述第二视点上一次渲染得到的帧图像。

9.一种图像渲染设备,其特征在于,所述图像渲染设备包括:存储器、处理器及存储在所述存储器上并可在所述处理器上运行的图像渲染程序,所述图像渲染程序被所述处理器执行时实现如权利要求1至7中任一项所述的图像渲染方法的步骤。

10.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有图像渲染程序,所述图像渲染程序被处理器执行时实现如权利要求1至7中任一项所述的图像渲染方法的步骤。

图像渲染方法、装置、设备及计算机可读存储介质

技术领域

[0001] 本发明涉及图像处理技术领域,尤其涉及一种图像渲染方法、装置、设备及计算机可读存储介质。

背景技术

[0002] AR/VR通过图像渲染技术,将渲染的虚拟图像刷新到显示设备,用户通过头戴显示设备体验到虚拟显示/增强显示的效果。由于渲染过程需要时间,就会造成实际和感知之间的时间延迟,这种延迟必须控制在一定范围,否则用户会出现眩晕等不适感。为了缓解用户这种不适,出现了ATW(Asynchronous Time warp,异步时间扭曲)技术。

[0003] 现行的智能眼镜假设屏幕刷新周期是 T ,在前 $T/2$ 渲染左眼,左眼渲染完成如果还未到 $T/2$ 时刻,等到 $T/2$ 时刻渲染右眼,固定在 $T/2$ 时刻启动左眼ATW,在 T 时刻启动右眼ATW,如果启动ATW时没做完,就用上一帧来代替。这种渲染机制下,渲染进程可能存在等待空期和做无用的渲染,导致既浪费GPU资源做处理,图像渲染使用率又不高。

发明内容

[0004] 本发明的主要目的在于提供一种图像渲染方法、装置、设备及计算机可读存储介质,旨在解决现有的渲染机制既浪费GPU资源做处理,图像渲染使用率又不高的技术问题。

[0005] 为实现上述目的,本发明提供一种图像渲染方法,所述方法包括以下步骤:

[0006] 当接收到按照预设的时间周期 T_1 发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;

[0007] 若在接收到所述同步信号后的 T_2 时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T_2 < T_1$;

[0008] 若所述第一待渲染数据在接收到所述同步信号后的 T_2 时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

[0009] 在接收到所述同步信号后的 T_2 时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区。

[0010] 可选地,所述当接收到按照预设的时间周期 T_1 发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染的步骤包括:

[0011] 当接收到按照预设的时间周期 T_1 发送的同步信号时,检测第二视点是否有正在渲染的第三待渲染数据;

[0012] 若有,则停止渲染所述第三待渲染数据,并获取第一视点当前的第一待渲染数据开始渲染;

[0013] 若没有,则获取所述第一待渲染数据开始渲染。

[0014] 可选地,所述方法还包括:

[0015] 在接收到所述同步信号的时刻获取所述显示缓冲区中当前缓存的第一视点的第

一视点帧图像；

[0016] 采用所述第一视点帧图像刷新显示设备中当前显示的第一视点的帧图像。

[0017] 可选地,所述方法还包括:

[0018] 在接收到所述同步信号后的T2时刻获取所述显示缓冲区中当前缓存的第二视点的第二视点帧图像;

[0019] 采用所述第二视点帧图像刷新显示设备中当前显示的第二视点的帧图像。

[0020] 可选地,所述方法还包括:

[0021] 若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

[0022] 当检测到第一视点的丢帧次数达到预设次数时,增大T2,其中,增大后的T2小于T1。

[0023] 可选地,所述方法还包括:

[0024] 若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

[0025] 若在接收到所述同步信号的时刻第二视点有正在渲染的待渲染数据,则对第二视点的丢帧次数累加一次;

[0026] 当检测到第一视点的丢帧次数与第二视点的丢帧次数的比值大于预设比值时,增大T2,其中,增大后的T2小于T1;

[0027] 当检测到第二视点的丢帧次数与第一视点的丢帧次数的比值大于所述预设比值时,减小T2,其中,减小后的T2大于0。

[0028] 可选地,所述方法还包括:

[0029] 当第一次接收到所述同步信号时,将第一视点和第二视点中其中一个设置为左眼视点,另一个设置为右眼视点;

[0030] 从第一次接收到所述同步信号起每隔预设的时间周期T3将第一视点和第二视点的左右眼视点设置进行互换。

[0031] 为实现上述目的,本发明还提供一种图像渲染装置,所述装置包括:

[0032] 第一渲染模块,用于当接收到按照预设的时间周期T1发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;

[0033] 第二渲染模块,用于若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T2 < T1$;

[0034] 所述第二渲染模块还用于若所述第一待渲染数据在接收到所述同步信号后的T2时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

[0035] 缓存模块,用于在接收到所述同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区。

[0036] 为实现上述目的,本发明还提供一种图像渲染设备,所述图像渲染设备包括:存储器、处理器及存储在所述存储器上并可在所述处理器上运行的图像渲染程序,所述图像渲

染程序被所述处理器执行时实现如上所述的图像渲染方法的步骤。

[0037] 此外,为实现上述目的,本发明还提出一种计算机可读存储介质,所述计算机可读存储介质上存储有图像渲染程序,所述图像渲染程序被处理器执行时实现如上所述的图像渲染方法的步骤。

[0038] 本发明中,通过在接收到同步信号后,获取第一视点当前的第一待渲染数据开始渲染;若在接收到同步信号后的T2时刻第一待渲染数据未渲染完成,则停止渲染第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染;若在接收到同步信号后的T2时刻前第一待渲染数据渲染完成,则在第一待渲染数据渲染完成后获取第二待渲染数据开始渲染;在接收到同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到同步信号的时时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入显示缓冲区。在本发明中,通过在T2时刻第一待渲染数据还未渲染完成时,停止渲染第一待渲染数据,以避免对第一待渲染数据继续渲染而造成GPU做无用功,避免浪费GPU资源;当第一待渲染数据在T2时刻前就渲染完成时,GPU可以立即获取第二视点当前时刻的第二待渲染数据进行渲染,从而避免GPU因出现等待状态而浪费GPU资源,并且由于提早开始第二待渲染数据的渲染,也提高了第二待渲染数据的渲染完成率,从而进一步降低了GPU做无用功的可能性,也提高了图像渲染使用率。

附图说明

[0039] 图1为本发明实施例方案涉及的硬件运行环境的结构示意图;

[0040] 图2为本发明图像渲染方法第一实施例的流程示意图;

[0041] 图3为本发明图像渲染方法实施例涉及的一种渲染流程示意图;

[0042] 图4为本发明图像渲染装置较佳实施例的功能模块示意图。

[0043] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0044] 应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0045] 如图1所示,图1是本发明实施例方案涉及的硬件运行环境的设备结构示意图。

[0046] 需要说明的是,本发明实施例图像渲染设备可以是智能手机、个人计算机和服务器等设备,在此不做具体限制。

[0047] 如图1所示,该图像渲染设备可以包括:处理器1001,例如CPU,网络接口1004,用户接口1003,存储器1005,通信总线1002。其中,通信总线1002用于实现这些组件之间的连接通信。用户接口1003可以包括显示屏(Display)、输入单元比如键盘(Keyboard),可选用户接口1003还可以包括标准的有线接口、无线接口。网络接口1004可选的可以包括标准的有线接口、无线接口(如WI-FI接口)。存储器1005可以是高速RAM存储器,也可以是稳定的存储器(non-volatile memory),例如磁盘存储器。存储器1005可选的还可以是独立于前述处理器1001的存储装置。

[0048] 本领域技术人员可以理解,图1中示出的设备结构并不构成对图像渲染设备的限定,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。

[0049] 如图1所示,作为一种计算机存储介质的存储器1005中可以包括操作系统、网络通

信模块、用户接口模块以及图像渲染程序。操作系统是管理和控制设备硬件和软件资源的程序,支持图像渲染程序以及其它软件或程序的运行。在图1所示的设备中,用户接口1003主要用于与客户端进行数据通信;网络接口1004主要用于与服务器建立通信连接;而处理器1001可以用于调用存储器1005中存储的图像渲染程序,并执行以下操作:

[0050] 当接收到按照预设的时间周期T1发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;

[0051] 若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T2 < T1$;

[0052] 若所述第一待渲染数据在接收到所述同步信号后的T2时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

[0053] 在接收到所述同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的時刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区。

[0054] 进一步地,所述当接收到按照预设的时间周期T1发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染的包括:

[0055] 当接收到按照预设的时间周期T1发送的同步信号时,检测第二视点是否有正在渲染的第三待渲染数据;

[0056] 若有,则停止渲染所述第三待渲染数据,并获取第一视点当前的第一待渲染数据开始渲染;

[0057] 若没有,则获取所述第一待渲染数据开始渲染。

[0058] 进一步地,处理器1001还可以用于调用存储器1005中存储的图像渲染程序,执行以下操作:

[0059] 在接收到所述同步信号的時刻获取所述显示缓冲区中当前缓存的第一视点的的第一视点帧图像;

[0060] 采用所述第一视点帧图像刷新显示设备中当前显示的第一视点的帧图像。

[0061] 进一步地,处理器1001还可以用于调用存储器1005中存储的图像渲染程序,执行以下操作:

[0062] 在接收到所述同步信号后的T2时刻获取所述显示缓冲区中当前缓存的第二视点的第二视点帧图像;

[0063] 采用所述第二视点帧图像刷新显示设备中当前显示的第二视点的帧图像。

[0064] 进一步地,处理器1001还可以用于调用存储器1005中存储的图像渲染程序,执行以下操作:

[0065] 若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

[0066] 当检测到第一视点的丢帧次数达到预设次数时,增大T2,其中,增大后的T2小于T1。

[0067] 进一步地,处理器1001还可以用于调用存储器1005中存储的图像渲染程序,执行以下操作:

[0068] 若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一

视点的丢帧次数累加一次；

[0069] 若在接收到所述同步信号的时刻第二视点有正在渲染的待渲染数据,则对第二视点的丢帧次数累加一次；

[0070] 当检测到第一视点的丢帧次数与第二视点的丢帧次数的比值大于预设比值时,增大 T_2 ,其中,增大后的 T_2 小于 T_1 ；

[0071] 当检测到第二视点的丢帧次数与第一视点的丢帧次数的比值大于所述预设比值时,减小 T_2 ,其中,减小后的 T_2 大于0。

[0072] 进一步地,处理器1001还可以用于调用存储器1005中存储的图像渲染程序,执行以下操作：

[0073] 当第一次接收到所述同步信号时,将第一视点和第二视点中其中一个设置为左眼视点,另一个设置为右眼视点；

[0074] 从第一次接收到所述同步信号起每隔预设的时间周期 T_3 将第一视点和第二视点的左右眼视点设置进行互换。

[0075] 基于上述的结构,提出图像渲染方法的各个实施例。

[0076] 参照图2,图2为本发明图像渲染方法第一实施例的流程示意图。

[0077] 本发明实施例提供了图像渲染方法的实施例,需要说明的是,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。本发明图像渲染方法应用于可做图像渲染的设备或处理器,例如可应用于图像处理器GPU,以下以GPU为例进行各实施例的阐述。在本实施例中,图像渲染方法包括：

[0078] 步骤S10,当接收到按照预设的时间周期 T_1 发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染；

[0079] 同步信号是控制显示设备刷新频率的信号,在本实施例中,同步信号可以是垂直同步信号(vertical synchronization, vsync),也可以是其他能够控制显示设备刷新频率的信号。一般是由显卡DAC(数字模拟转换器)每完成一帧的扫描后产生一个vsync,预示着一帧结束,新一帧的开始。在本实施例中,将发送同步信号的时间周期称为 T_1 ,也即,每隔 T_1 发送一个同步信号。每当GPU接收到同步信号后,将执行相同的渲染流程,只是每一次处理的数据有变化,以下是以一个同步信号为例进行的具体说明。

[0080] 当GPU接收到一个同步信号后,获取第一视点当前的待渲染数据(为示区别以下称为第一待渲染数据),并开始对第一待渲染数据进行渲染。其中,VR(Virtual Reality,虚拟现实)/AR(Augmented Reality,增强现实)场景中,通过针对左右眼渲染出不同的图像来形成3D视觉效果,在本实施例中,第一视点可以是左眼和右眼中的其中一个,若第一视点是左眼那么第二视点就是右眼,若第一视点是右眼那么第二视点就是左眼。待渲染数据是指需要由GPU进行渲染的数据,例如顶点数据、纹理数据等,待渲染数据可以来自于CPU,或来自于GPU数据处理流程中的前一步输出结果,在此对待渲染数据的来源不做限制;对待渲染数据进行渲染可以包括顶点着色、纹理填充等渲染操作,具体可参考现有的GPU图像渲染原理,在此不进行详细赘述。同一视点的待渲染数据会随着时间更新,更新的周期可能与同步信号的发送时间周期同步,也可能不同步,在本实施例中不做限制;GPU在接收到同步信号后需要获取第一视点的待渲染数据时,是获取第一视点当前时刻的待渲染数据进行渲染。需要说明的是,VR/AR场景中,需要结合用户的头部运动姿态来进行渲染,GPU在对第一视点

的第一待渲染数据进行渲染时,还可获取当前时刻的姿态数据,基于姿态数据来对待渲染数据进行渲染。

[0081] 需要说明的是,GPU在接收到同步信号时,可以立即获取第一视点当前的第一待渲染数据开始渲染,也可以是在一定时长后才开始渲染,对此本实施例并不做限制。例如,若GPU在接收到同步信号时,有正在渲染的工作没完成,则在一实施方式中,GPU可以将当前渲染工作完成后再获取第一视点当前的第一待渲染数据开始渲染,在另一实施方式中,GPU也可以停止正在渲染的工作,立即获取第一视点当前的第一待渲染数据开始渲染。

[0082] 步骤S20,若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T2 < T1$;

[0083] 在接收到同步信号后,GPU可以检测在接收到同步信号后的T2时刻前,第一待渲染数据是否渲染完成。若GPU在接收到同步信号后的T2时刻第一待渲染数据还未渲染完成,则可以停止渲染第一待渲染数据,并获取第二视点当前的待渲染数据(为示区别以下称为第二待渲染数据),并开始对第二待渲染数据进行渲染。其中,T2可以是预先根据需要设置的一个时长,T2大于0且小于T1,具体可以根据第一视点和第二视点的待渲染数据的平均渲染时长来设置,例如第一视点和第二视点的待渲染数据的平均渲染时长差不多,则可以将T2设置为T1的一半。当在接收到同步信号后的T2时刻第一待渲染数据还未渲染完成,说明第一待渲染数据的渲染耗费了较长时间;而在T2时刻GPU要对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲,此时,由于第一待渲染数据还未渲染完成,GPU进行异步时间扭曲的帧图像是第一视点上一个待渲染数据渲染得到的帧图像,因此,对第一待渲染数据继续进行渲染只是让GPU做无用功;对此,在本实施例中,在T2时刻第一待渲染数据还未渲染完成时,停止渲染第一待渲染数据,以避免GPU做无用的渲染工作。

[0084] 具体地,GPU中可以设置一个计时器,从接收到同步信号后开始计时,检测在计时器的计时时长到达T2之前,是否已完成第一待渲染数据的渲染。

[0085] 步骤S30,若所述第一待渲染数据在接收到所述同步信号后的T2时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

[0086] 若第一待渲染数据在GPU接收到同步信号后的T2时刻前渲染完成,则GPU可以在第一待渲染数据渲染完成后,获取第二待渲染数据开始渲染。当第一待渲染数据在T2时刻前渲染完成时,说明第一待渲染数据的渲染耗费的较短的时长;若等到T2时刻才对第二视点的第二待渲染数据进行渲染,从第一待渲染数据渲染完成到T2时刻的这段时间GPU的渲染进程处于暂停等待状态,显然浪费了GPU的计算资源;并且,很可能由于第二待渲染数据需要耗费较长时间,导致第二待渲染数据在下一同步信号到时还未渲染完成,从而导致GPU对第二待渲染数据的渲染处理也变成无用功;对此,在本实施例中,当第一待渲染数据在T2时刻前就渲染完成时,GPU可以立即获取第二视点当前时刻的第二待渲染数据进行渲染,从而避免GPU因出现等待状态而浪费GPU资源,并且由于提早开始第二待渲染数据的渲染,也提高了第二待渲染数据的渲染完成率,从而进一步降低了GPU做无用功的可能性,也即提高了图像渲染使用率。图像渲染使用率即GPU渲染出的帧图像被使用的次数占GPU渲染总次数(包括渲染完成和未渲染完成)的比例。

[0087] 步骤S40,在接收到所述同步信号后的T2时刻对最近一次渲染得到的第一视点的

帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区。

[0088] GPU在接收到同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲,将异步时间扭曲后得到的帧图像存入显示缓冲区;在接收到同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲,将异步时间扭曲后得到的帧图像存入显示缓冲区。其中,异步时间扭曲是指GPU通过另一个线程(以下称为ATW线程)来对帧图像进行时间扭曲,ATW线程与渲染待渲染数据的渲染进程是可以并行运行的,例如在T2时刻GPU通过渲染线程对第二待渲染数据进行渲染的同时,ATW线程也开始对帧图像进行时间扭曲,两者可以并行执行。显示缓冲区是一个用于存放可以供显示器显示的帧图像的缓冲区,也即,显示器显示的帧图像将从显示缓冲区中获取;预先可以设置第一视点和第二视点对应的帧图像存放于显示缓冲区中的位置,例如,将显示缓冲区中用于存放第一视点对应的帧图像的位置称为第一视点缓冲区,将用于存放第二视点对应的帧图像的位置称为第二视点缓冲区。需要说明的是,GPU对待渲染数据进行渲染完成后,将得到一帧图像(以下称为“帧图像”),待渲染数据随着时间更新,GPU也会不断渲染得到新的帧图像,GPU在需要获取帧图像进行异步时间扭曲时,获取的是最近一次渲染得到的帧图像。GPU可将帧图像存放于一个特定的存储区域,例如一个专门用于存放渲染得到的帧图像的缓冲区,如纹理缓冲区,ATW线程从该特定的存储区域获取最近一次渲染得到的帧图像。

[0089] 可以理解的是,如果第一待渲染数据在接收到同步信号后的T2时刻前就渲染完成了,那么在T2时刻进行异步时间扭曲时,第一视点最近一次渲染得到的帧图像就是对该第一待渲染数据渲染得到的帧图像;如果第一待渲染数据在接收到同步信号后的T2时刻还未渲染完成,那么在T2时刻进行异步时间扭曲的帧图像就是第一视点上一次渲染得到的帧图像。如果第二待渲染数据在接收到下一个同步信号之前就渲染完成了,那么在接收到下一个同步信号的时刻进行异步时间扭曲时,第二视点最近一次渲染得到的帧图像就是对该第二待渲染数据渲染得到的帧图像;如果第二待渲染数据在接收到下一个同步信号的时刻还未渲染完成,那么在接收到下一个同步信号的时刻进行异步时间扭曲的帧图像就是第二视点上一次渲染得到的帧图像。

[0090] 在本实施例中,通过在接收到同步信号后,获取第一视点当前的第一待渲染数据开始渲染;若在接收到同步信号后的T2时刻第一待渲染数据未渲染完成,则停止渲染第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染;若在接收到同步信号后的T2时刻前第一待渲染数据渲染完成,则在第一待渲染数据渲染完成后获取第二待渲染数据开始渲染;在接收到同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入显示缓冲区。在本实施例中,通过在T2时刻第一待渲染数据还未渲染完成时,停止渲染第一待渲染数据,以避免对第一待渲染数据继续渲染而造成GPU做无用功,避免浪费GPU资源;当第一待渲染数据在T2时刻前就渲染完成时,GPU可以立即获取第二视点当前时刻的第二待渲染数据进行渲染,从而避免GPU因出现等待状态而浪费GPU资源,并且由于提早开始第二待渲染数据的渲染,也提高了第二待渲染数据的渲染完成率,从而进一步降低了GPU做无用功的可能性,也提高了图像渲染使用率。

[0091] 进一步地,在一实施方式中,所述步骤S10包括:

[0092] 步骤S101,当接收到按照预设的时间周期T1发送的同步信号时,检测第二视点是否有正在渲染的第三待渲染数据;

[0093] 当GPU接收到同步信号时,可检测第二视点是否有正在渲染的待渲染数据(为示区别以下称为第三待渲染数据)。可以理解的是,第三待渲染数据也就相当于是上一个同步信号阶段中的第二待渲染数据,也即,相当于是GPU检测在接收到下一个同步信号前第二待渲染数据是否渲染完成。

[0094] 步骤S102,若有,则停止渲染所述第三待渲染数据,并获取第一视点当前的第一待渲染数据开始渲染;

[0095] 步骤S103,若没有,则获取所述第一待渲染数据开始渲染。

[0096] 若没有正在渲染的第三待渲染数据,则GPU可以直接获取第一待渲染数据开始渲染。

[0097] 若有正在渲染的第三待渲染数据,则GPU可以停止渲染该第三待渲染数据,并获取第一视点当前的第一待渲染数据开始渲染。当在接收到同步信号时第三待渲染数据还未渲染完成,说明第三待渲染数据的渲染耗费了较长时间;而在接收到同步信号的时刻GPU要对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲,此时,由于第三待渲染数据还未渲染完成,GPU进行异步时间扭曲的帧图像是第二视点上一个待渲染数据渲染得到的帧图像,因此,对第三待渲染数据继续进行渲染只是让GPU做无用功,并且还会占用第一待渲染数据的渲染时间,导致第一待渲染数据可能无法在T2时刻渲染完成,从而导致GPU对第一待渲染数据的渲染处理也变成无用功;对此,在本实施例中,当在接收到同步信号的时刻第三待渲染数据还未完成时,停止渲染第三待渲染数据,以避免GPU做无用的渲染工作,也避免第三渲染数据的渲染占用第一渲染数据的渲染时间,提高了第一待渲染数据的渲染完成率,从而进一步降低了GPU做无用功的可能性,也即提高了图像渲染使用率。

[0098] 进一步地,基于上述第一实施例,提出本发明图像渲染方法的第二实施例,在本实施例中,所述方法还包括:

[0099] 步骤S50,在接收到所述同步信号的时刻获取所述显示缓冲区中当前缓存的第一视点的帧图像;

[0100] 在GPU接收到同步信号的时刻,GPU获取显示缓冲区中当前缓存的第一视点的帧图像(以下称为第一视点帧图像以示区分)。也即,GPU在接收到同步信号后的T2时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲,将异步时间扭曲后得到的帧图像存入显示缓冲区,该帧图像会在GPU下一次接收到同步信号时被作为第一视点帧图像获取。

[0101] 步骤S60,采用所述第一视点帧图像刷新显示设备中当前显示的第一视点的帧图像。

[0102] 在获取到第一视点帧图像后,GPU采用第一视点帧图像刷新显示设备中当前显示的第一视点的帧图像。具体地,GPU可以将第一视点帧图像发送给显示设备,由显示设备采用第一视点帧图像刷新当前正在显示的第一视点的帧图像。在一实施方式中,GPU发送第一视点帧图像可以基于MIPI(Mobile Industry Processor Interface,移动行业处理器接口)协议发送。

[0103] 进一步地,在一实施方式中,所述方法还包括:

[0104] 步骤S70,在接收到所述同步信号后的T2时刻获取所述显示缓冲区中当前缓存的

第二视点的第二视点帧图像；

[0105] 在GPU接收到同步信号后的T2时刻, GPU获取显示缓冲区中当前缓存的第二视点的帧图像(以下称为第二视点帧图像以示区分)。也即, GPU在接收到同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲, 将异步时间扭曲后得到的帧图像存入显示缓冲区, 该帧图像会在GPU接收到该同步信号后的T2时刻被作为第二视点帧图像获取。

[0106] 步骤S80, 采用所述第二视点帧图像刷新显示设备中当前显示的第二视点的帧图像。

[0107] 在获取到第二视点帧图像后, GPU采用第二视点帧图像刷新显示设备中当前显示的第二视点的帧图像。具体地, GPU可以将第二视点帧图像发送给显示设备, 由显示设备采用第二视点帧图像刷新当前正在显示的第二视点的帧图像。在一实施方式中, GPU发送第二视点帧图像可以基于MIPI协议发送。

[0108] 在一实施方式中, 如图3所示, 示出了两个周期T1内渲染线程、ATM线程的处理流程。

[0109] 进一步地, 基于上述第一和/或第二实施例, 提出本发明图像渲染方法的第三实施例, 在本实施例中, 所述方法还包括:

[0110] 步骤A10, 若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成, 则对第一视点的丢帧次数累加一次;

[0111] 在本实施例中, GPU可以对T2的大小进行更改, 以调整一个周期T1内可供渲染第一视点和第二视点的待渲染数据的时长。

[0112] 具体地, GPU可以对第一视点的丢帧次数进行统计, 在渲染进程开始时, 可以对第一视点的丢帧次数初始化为0。若GPU在接收到同步信号后的T2时刻第一待渲染数据未渲染完成, 则对第一视点的丢帧次数累加一次, 也即, 在原丢帧次数上加上1。例如, 第一次接收到同步信号后的T2时刻第一待渲染数据(这里的第一待渲染数据是相对于第一次接收到的该同步信号而言的)已渲染完成, 则第一视点的丢帧次数仍然为0; 第二次接收到同步信号后的T2时刻第一待渲染数据(这里的第一待渲染数据是相对于第二次接收到的该同步信号而言的)未渲染完成, 则第一视点的丢帧次数累加一次, 变成1; 第三次接收到同步信号后的T2时刻第一待渲染数据(这里的第一待渲染数据是相对于第三次接收到的该同步信号而言的)已渲染完成, 则第一视点的丢帧次数仍然为1; 第四次接收到同步信号后的T2时刻第一待渲染数据(这里的第一待渲染数据是相对于第四次接收到的该同步信号而言的)未渲染完成, 则第一视点的丢帧次数累加一次, 变成2; 依次类推。

[0113] 步骤A20, 当检测到第一视点的丢帧次数达到预设次数时, 增大T2, 其中, 增大后的T2小于T1。

[0114] 当第一视点的丢帧次数累加达到预设次数时, GPU可以增大T2, 但增大后的T2仍然小于T1。其中, 预设次数可以是预先根据需要设置的, 在此不做限制, 例如, 预设次数为10次, 则当第一视点的丢帧次数等于10时, GPU增大T2。增大T2可以是在T2的基础上加上一个预设的值, 例如, 假设原始值是10, 预设值是2, 则增大后的T2为12; 也可以是直接将T2设置为一个比原始值更大的值, 例如原始值为10, 直接将T2设置为12。

[0115] 进一步地, 在一实施方式中, GPU可以在每次增大T2后, 重置第一视点的丢帧次数, 并开始重新计数, 当丢帧次数再次达到预设次数时, 对T2进一步增加。例如, 第一次达到丢

帧次数时增大T2到12,第二次达到丢帧次数时增大T2到14。可以设置一个上限,保证增大后的T2是小于T1的。

[0116] 在本实施例中,通过对第一视点的丢帧次数进行累计,当丢帧次数大于预设次数时,增大T2,以延长可供渲染第一视点的待渲染数据的时长,使得第一视点的待渲染数据的渲染成功率提高,从而提高第一视点的图像渲染使用率。

[0117] 进一步地,基于上述第一和/或第二实施例,提出本发明图像渲染方法的第四实施例,在本实施例中,所述方法还包括:

[0118] 步骤A30,若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

[0119] 步骤A40,若在接收到所述同步信号的时刻第二视点有正在渲染的待渲染数据,则对第二视点的丢帧次数累加一次;

[0120] 在本实施例中,GPU可以对T2的大小进行更改,以调整一个周期T1内可供渲染第一视点和第二视点的待渲染数据的时长。

[0121] 具体地,GPU可以对第一视点和第二视点的丢帧次数分别进行统计,在渲染进程开始时,可以对第一视点和第二视点的丢帧次数初始化为0。第一视点和第二视点的丢帧次数统计方式可参照上述第三实施例步骤A10中第一视点的丢帧次数统计方法,在此不做重复赘述。

[0122] 步骤A50,当检测到第一视点的丢帧次数与第二视点的丢帧次数的比值大于预设比值时,增大T2,其中,增大后的T2小于T1;

[0123] 当GPU检测到第一视点的丢帧次数与第二视点的丢帧次数的比值大于预设比值时,可以增大T2,但增大后的T2仍然小于T1。其中,第一视点的丢帧次数与第二视点的丢帧次数的比值是指第一视点的丢帧次数除以第二视点的丢帧次数得到的结果。预设比例可以是预先根据需要设置的,在此不做限制,例如设置为0.5。增大T2的方式可参照上述第三实施例步骤A20中增大T2的方式,在此不做重复赘述。

[0124] 步骤A60,当检测到第二视点的丢帧次数与第一视点的丢帧次数的比值大于所述预设比值时,减小T2,其中,减小后的T2大于0。

[0125] 当GPU检测到第二视点的丢帧次数与第一视点的丢帧次数的比值大于预设比值时,可以减小T2,但减小后的T2要大于0。其中,第二视点的丢帧次数与第一视点的丢帧次数的比值是指第二视点的丢帧次数除以第一视点的丢帧次数得到的结果。减小T2可以是在T2的基础上减去一个预设的值,例如,假设原始值是10,预设值是2,则减小后的T2为8;也可以是直接将T2设置为一个比原始值更小的值,例如原始值为10,直接将T2设置为8。可以设置一个上限,保证增大后的T2是小于T1的,设置一个下限,保证减小后的T2是大于0的。

[0126] 在一实施方式中,每次增大和减小T2后,重置第一视点和第二视点的丢帧次数,并开始重新计数,当比值再次大于预设比值时,对应地对T2进一步增加或减少。

[0127] 在本实施例中,通过对第一视点和第二视点的丢帧次数进行累计,当第一视点的丢帧次数与第二视点的丢帧次数比值大于预设阈值时,增大T2,以延长可供渲染第一视点的待渲染数据的时长,当第二视点的丢帧次数与第一视点的丢帧次数比值大于预设阈值时,减小T2,以延长可供渲染第二视点的待渲染数据的时长,使得第一视点和第二视点的渲染成功率相当,从而平衡第一视点和第二视点的图像渲染使用率。

[0128] 进一步地,基于上述第一、第二、第三和/或第四实施例,提出本发明图像渲染方法的第五实施例,在本实施例中,所述方法还包括:

[0129] 步骤A70,当第一次接收到所述同步信号时,将第一视点和第二视点中其中一个设置为左眼视点,另一个设置为右眼视点;

[0130] 当第一次接收到同步信号时,GPU可以将第一视点和第二视点中的其中一个设置为左眼视点,另一个设置为右眼视点。例如,可以将第一视点设置为左眼视点,将第二视点设置为右眼视点,或者,将第一视点设置为右眼视点,将第二视点设置为左眼视点。

[0131] 步骤A80,从第一次接收到所述同步信号起每隔预设的时间周期 T_3 将第一视点和第二视点的左右眼视点设置进行互换。

[0132] GPU可以从第一次接收到同步信号起每隔预设的时间周期 T_3 将第一视点和第二视点的左右眼视点设置进行互换,也即,如果第一视点是左眼视点第二视点是右眼视点,则互换后第一视点是右眼视点第二视点是左眼视点,如果第一视点是右眼视点第二视点是左眼视点,则互换后第一视点是左眼视点第二视点是右眼视点。其中, T_3 可以根据具体的需要进行设置;由于在一个时间周期 T_1 内,对用户而言其看到的画面是先刷新一只眼的画面,再刷新另一只眼的画面,因此,互换第一视点和第二视点的左右眼视点设置,相当于左右眼画面刷新的顺序变化了,为避免刷新顺序频繁变化导致用户看到的画面出现卡顿, T_3 可以设置得远大于 T_1 。由于在一个时间周期 T_1 内,是先渲染第一视点的待渲染数据,再渲染第二视点的待渲染数据,两个视点的待渲染数据的渲染时长可能不同,但可供渲染两个视点的待渲染数据的时长是通过 T_2 相对固定的,从而可能导致两个视点的渲染成功率不平衡,一个渲染成功率较低,另一个渲染成功率较高,对用户而言就是一只眼画面较流畅,另一只眼画面较卡顿;本实施例中,为避免这个问题,通过定期的互换第一视点和第二视点的左右眼视点设置,使得左眼视点和右眼视点的渲染成功率随着时间的推移逐渐平衡,从而使得用户双眼的感受更加平衡和和谐,提高用户体验。

[0133] 此外,本发明实施例还提出一种图像渲染装置,参照图4,所述装置包括:

[0134] 第一渲染模块10,用于当接收到按照预设的时间周期 T_1 发送的同步信号后,获取第一视点当前的第一待渲染数据开始渲染;

[0135] 第二渲染模块20,用于若在接收到所述同步信号后的 T_2 时刻所述第一待渲染数据未渲染完成,则停止渲染所述第一待渲染数据,并获取第二视点当前的第二待渲染数据开始渲染,其中, $0 < T_2 < T_1$;

[0136] 所述第二渲染模块20还用于若所述第一待渲染数据在接收到所述同步信号后的 T_2 时刻前渲染完成,则在所述第一待渲染数据渲染完成后获取所述第二待渲染数据开始渲染;

[0137] 缓存模块30,用于在接收到所述同步信号后的 T_2 时刻对最近一次渲染得到的第一视点的帧图像进行异步时间扭曲后存入显示缓冲区,以及在接收到所述同步信号的时刻对最近一次渲染得到的第二视点的帧图像进行异步时间扭曲后存入所述显示缓冲区。

[0138] 进一步地,所述第一渲染模块10包括:

[0139] 检测单元,用于当接收到按照预设的时间周期 T_1 发送的同步信号时,检测第二视点是否有正在渲染的第三待渲染数据;

[0140] 第一渲染单元,用于若有,则停止渲染所述第三待渲染数据,并获取第一视点当前

的第一待渲染数据开始渲染；

[0141] 第二渲染单元,用于若没有,则获取所述第一待渲染数据开始渲染。

[0142] 进一步地,所述装置还包括:

[0143] 第一获取模块,用于在接收到所述同步信号的时刻获取所述显示缓冲区中当前缓存的第一视点的第一视点帧图像;

[0144] 第一刷新模块,用于采用所述第一视点帧图像刷新显示设备中当前显示的第一视点的帧图像。

[0145] 进一步地,所述装置还包括:

[0146] 第二获取模块,用于在接收到所述同步信号后的T2时刻获取所述显示缓冲区中当前缓存的第二视点的第二视点帧图像;

[0147] 第二刷新模块,用于采用所述第二视点帧图像刷新显示设备中当前显示的第二视点的帧图像。

[0148] 进一步地,所述装置还包括:

[0149] 第一计数模块,用于若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

[0150] 第一调整模块,用于当检测到第一视点的丢帧次数达到预设次数时,增大T2,其中,增大后的T2小于T1。

[0151] 进一步地,所述装置还包括:

[0152] 第二计数模块,用于若在接收到所述同步信号后的T2时刻所述第一待渲染数据未渲染完成,则对第一视点的丢帧次数累加一次;

[0153] 第三计数模块,用于若在接收到所述同步信号的时刻第二视点有正在渲染的待渲染数据,则对第二视点的丢帧次数累加一次;

[0154] 第二调整模块,用于当检测到第一视点的丢帧次数与第二视点的丢帧次数的比值大于预设比值时,增大T2,其中,增大后的T2小于T1;

[0155] 第三调整模块,用于当检测到第二视点的丢帧次数与第一视点的丢帧次数的比值大于所述预设比值时,减小T2,其中,减小后的T2大于0。

[0156] 进一步地,所述装置还包括:

[0157] 设置模块,用于当第一次接收到所述同步信号时,将第一视点和第二视点中其中一个设置为左眼视点,另一个设置为右眼视点;

[0158] 互换模块,用于从第一次接收到所述同步信号起每隔预设的时间周期T3将第一视点和第二视点的左右眼视点设置进行互换。

[0159] 本发明图像渲染装置的具体实施方式的拓展内容与上述图像渲染方法各实施例基本相同,在此不做赘述。

[0160] 此外,本发明实施例还提出一种计算机可读存储介质,所述存储介质上存储有图像渲染程序,所述图像渲染程序被处理器执行时实现如下所述的图像渲染方法的步骤。

[0161] 本发明图像渲染设备和计算机可读存储介质的各实施例,均可参照本发明图像渲染方法各个实施例,此处不再赘述。

[0162] 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。

[0163] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0164] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有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(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,空调器,或者网络设备等)执行本发明各个实施例所述的方法。

[0165] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

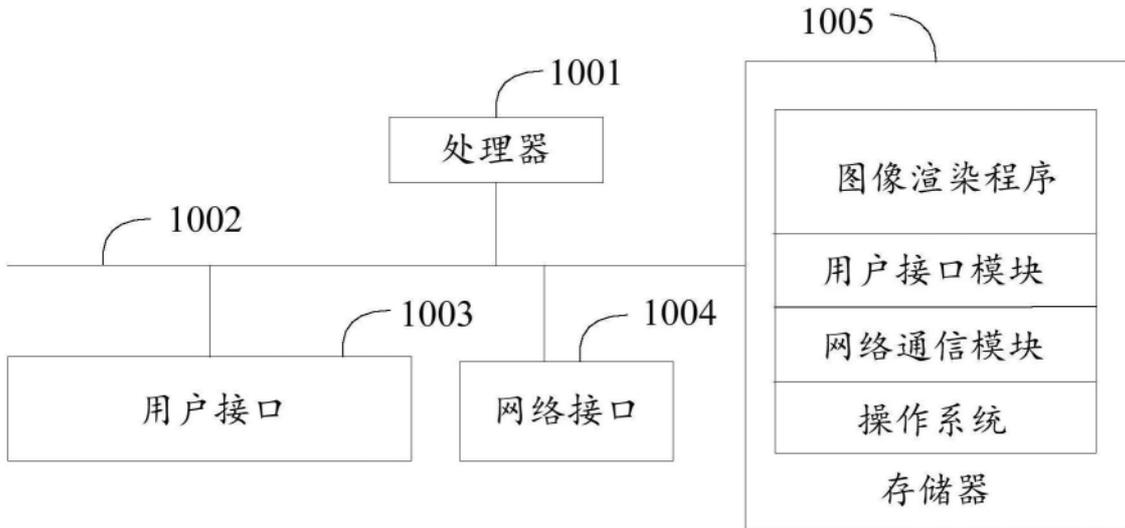


图1

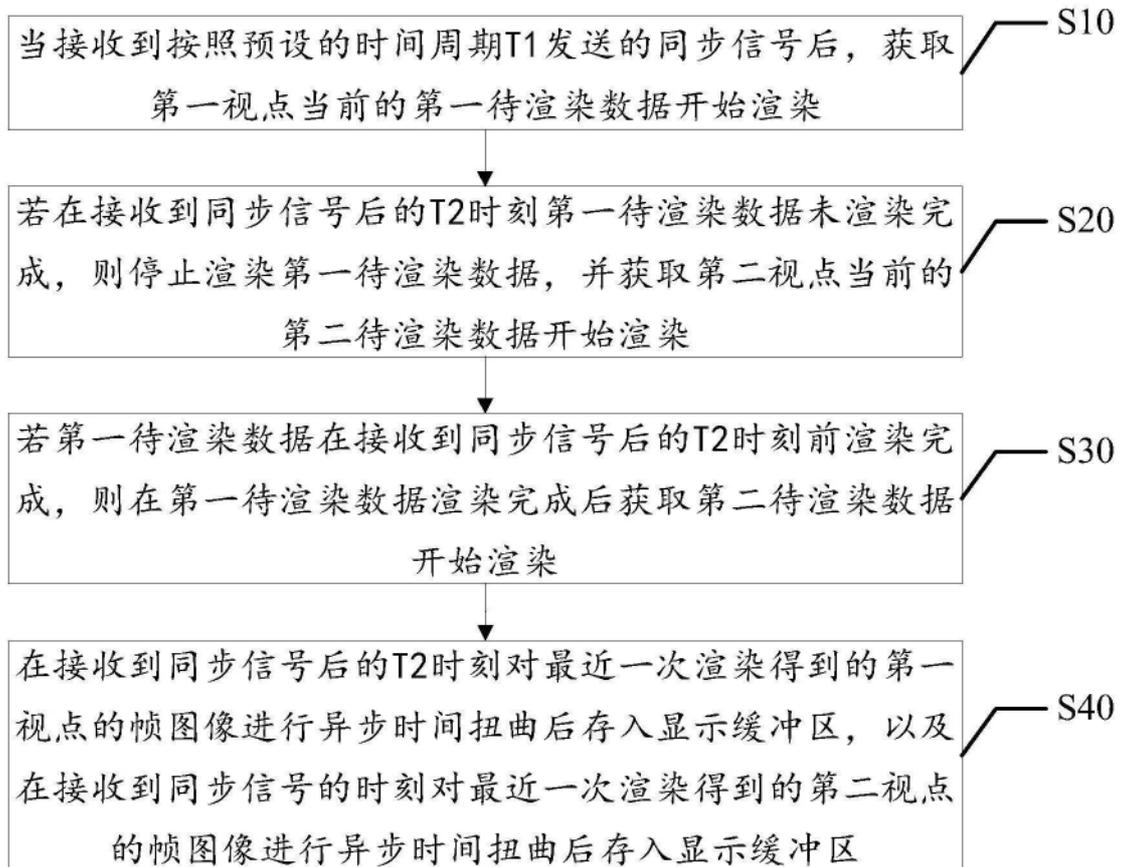


图2

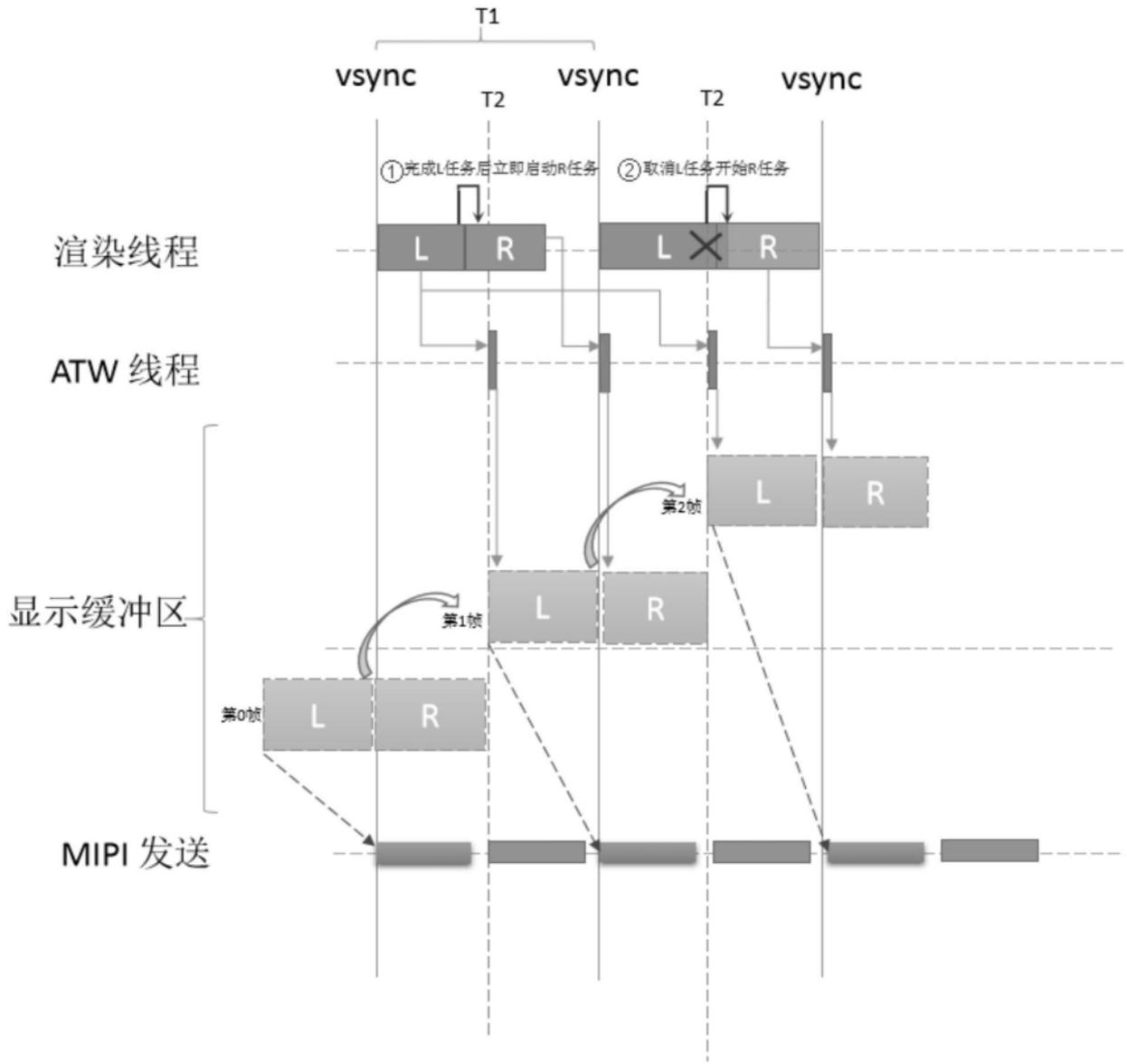


图3

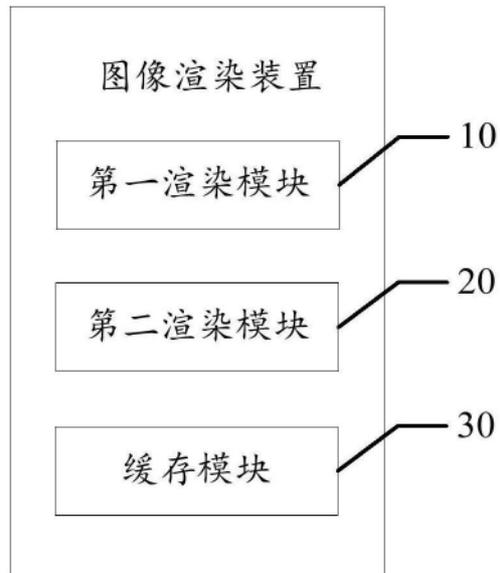


图4