



(12) 发明专利

(10) 授权公告号 CN 110213636 B

(45) 授权公告日 2023. 01. 10

(21) 申请号 201810398279.1

H04N 21/488 (2011.01)

(22) 申请日 2018.04.28

H04N 21/44 (2011.01)

(65) 同一申请的已公布的文献号

H04N 21/2187 (2011.01)

申请公布号 CN 110213636 A

G06F 9/50 (2006.01)

(43) 申请公布日 2019.09.06

审查员 赵真真

(73) 专利权人 腾讯科技(深圳)有限公司

地址 518057 广东省深圳市南山区高新区

科技中一路腾讯大厦35层

(72) 发明人 向晨宇

(74) 专利代理机构 北京三高永信知识产权代理

有限责任公司 11138

专利代理师 刘映东

(51) Int.Cl.

H04N 21/431 (2011.01)

H04N 21/4788 (2011.01)

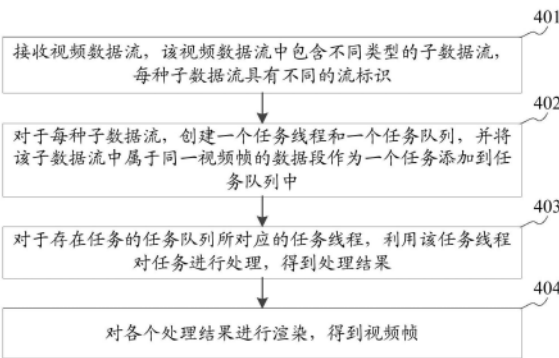
权利要求书4页 说明书19页 附图7页

(54) 发明名称

在线视频的视频帧生成方法、装置、存储介质及设备

(57) 摘要

本申请实施例公开了一种在线视频的视频帧生成方法、装置、存储介质及设备,属于计算机技术领域。所述方法包括:接收视频数据流,所述视频数据流中包含不同类型的子数据流,每种子数据流具有不同的流标识;对于每种子数据流,创建一个任务线程和一个任务队列,并将所述子数据流中属于同一视频帧的数据段作为一个任务添加到所述任务队列中;对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到各个处理结果;对各个处理结果进行渲染,得到视频帧。本申请实施例既可以提高终端性能,也可以提高画面播放的流畅度。



1. 一种在线视频的视频帧生成方法,其特征在于,用于客户端中,所述方法包括:

接收视频数据流,所述视频数据流中包含不同类型的子数据流,每种子数据流具有不同的流标识;所述视频数据流至少包括视频子数据流、弹幕子数据流和广告子数据流;

对于每种子数据流,创建一个任务线程和一个任务队列,并将所述子数据流中属于同一视频帧的数据段作为一个任务添加到所述任务队列中,其中每个任务的数据段是在所述子数据流中按固定时长截取的,并且不同任务的数据段分别对应所述子数据流中的不同时段;

对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到处理结果,所述处理包括:解封装和解码;

利用各个任务线程将各自得到的所述处理结果发送给渲染线程;

利用所述渲染线程将各个处理结果添加到视频缓冲区中;

将所述客户端对应的进程的绘制上下文赋值给所述渲染线程;

利用所述渲染线程按照所述处理结果被添加到所述视频缓冲区中的顺序,依次从所述视频缓冲区中读取一个处理结果;

对于每个处理结果,利用所述渲染线程创建纹理,并将所述纹理添加到所述绘制上下文中,对所述处理结果进行渲染,得到中间数据,将所述中间数据添加到所述纹理中,得到渲染结果;

通知图形处理器GPU对各个渲染结果进行渲染,得到视频帧,

其中当存在至少两个任务线程同时向所述渲染线程发送对应于同一视频帧的处理结果时,所述利用所述渲染线程将各个处理结果添加到视频缓冲区中,包括:

利用所述渲染线程接收对应于同一视频帧的至少两个处理结果;

利用所述渲染线程确定每个处理结果对应的子数据流的处理优先级;

利用所述渲染线程根据所述处理优先级进行轮询,将每次轮询得到的一个处理结果添加到所述视频缓冲区中,从而所述至少两个处理结果按所述处理优先级从高到底的顺序添加到所述视频缓冲区中。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当接收到后台运行指令时,通知所述渲染线程停止轮询。

3. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当接收到后台运行指令时,利用所述渲染线程清除所述绘制上下文中的纹理数据,将所述绘制上下文赋值给所述进程。

4. 根据权利要求1所述的方法,其特征在于,所述通知图形处理器GPU对各个渲染结果进行渲染,包括:

在每次得到一个视频帧对应的各个渲染结果时,向图形处理器GPU发送开始渲染指令,所述开始渲染指令用于指示所述GPU开始渲染所述视频帧对应的各个渲染结果,得到所述视频帧。

5. 根据权利要求1至4任一项所述的方法,其特征在于,所述方法还包括:

当接收到后台运行指令时,向图形处理器GPU发送停止渲染指令,所述停止渲染指令用于指示所述GPU停止对所述渲染结果进行渲染。

6. 一种直播视频的视频帧播放方法,其特征在于,用于直播客户端中,所述方法包括:

获取直播视频流,所述直播视频流包括直播画面数据流和互动数据流,所述互动数据流至少包括礼物数据流和弹幕数据流中的一种,所述直播视频流、所述礼物数据流和所述弹幕数据流具有不同的流标识;

对于每种数据流,创建一个任务线程和一个任务队列,并将所述数据流中属于同一直播视频帧的数据段作为一个任务添加到所述任务队列中,其中每个任务的数据段是在所述数据流中按固定时长截取的,并且不同任务的数据段分别对应所述数据流中的不同时段;

对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到处理结果,所述处理包括:解封装和解码;

利用各个任务线程将各自得到的所述处理结果发送给渲染线程;

利用所述渲染线程将各个处理结果添加到视频缓冲区中;

将所述直播客户端对应的进程的绘制上下文赋值给所述渲染线程;

利用所述渲染线程按照所述处理结果被添加到所述视频缓冲区中的顺序,依次从所述视频缓冲区中读取一个处理结果;

对于每个处理结果,利用所述渲染线程创建纹理,并将所述纹理添加到所述绘制上下文中,对所述处理结果进行渲染,得到中间数据,将所述中间数据添加到所述纹理中,得到渲染结果;

通知图形处理器GPU对各个渲染结果进行渲染得到直播视频帧,所述直播视频帧包括直播画面和互动数据;

播放所述直播视频帧,

其中当存在至少两个任务线程同时向所述渲染线程发送对应于同一直播视频帧的处理结果时,所述利用所述渲染线程将各个处理结果添加到视频缓冲区中,包括:

利用所述渲染线程接收对应于同一直播视频帧的至少两个处理结果;

利用所述渲染线程确定每个处理结果对应的数据流的处理优先级;

利用所述渲染线程根据所述处理优先级进行轮询,将每次轮询得到的一个处理结果添加到所述视频缓冲区中,从而所述至少两个处理结果按所述处理优先级从高到底的顺序添加到所述视频缓冲区中。

7. 一种在线视频的视频帧生成装置,其特征在于,用于客户端中,所述装置包括:

接收模块,用于接收视频数据流,所述视频数据流中包含不同类型的子数据流,每种子数据流具有不同的流标识;所述视频数据流至少包括视频子数据流、弹幕子数据流和广告子数据流;

创建模块,用于对于所述接收模块接收到的每种子数据流,创建一个任务线程和一个任务队列,并将所述子数据流中属于同一视频帧的数据段作为一个任务添加到所述任务队列中,其中每个任务的数据段是在所述子数据流中按固定时长截取的,并且不同任务的数据段分别对应所述子数据流中的不同时段;

处理模块,用于对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到各个处理结果,所述处理包括:解封装和解码;

生成模块,用于利用各个任务线程将各自得到的所述处理结果发送给渲染线程;利用所述渲染线程将各个处理结果添加到视频缓冲区中;将所述客户端对应的进程的绘制上下文赋值给所述渲染线程;利用所述渲染线程按照所述处理结果被添加到所述视频缓冲区中

的顺序,依次从所述视频缓冲区中读取一个处理结果;对于每个处理结果,利用所述渲染线程创建纹理,并将所述纹理添加到所述绘制上下文中,对所述处理结果进行渲染,得到中间数据,将所述中间数据添加到所述纹理中,得到渲染结果;通知图形处理器GPU对各个渲染结果进行渲染,得到视频帧,

其中当存在至少两个任务线程同时向所述渲染线程发送对应于同一视频帧的处理结果时,所述利用所述渲染线程将各个处理结果添加到视频缓冲区中,包括:

利用所述渲染线程接收对应于同一视频帧的至少两个处理结果;

利用所述渲染线程确定每个处理结果对应的子数据流的处理优先级;

利用所述渲染线程根据所述处理优先级进行轮询,将每次轮询得到的一个处理结果添加到所述视频缓冲区中,从而所述至少两个处理结果按所述处理优先级从高到底的顺序添加到所述视频缓冲区中。

8.一种直播视频的视频帧播放装置,其特征在于,用于直播客户端中,所述装置包括:

获取模块,用于获取直播视频流,所述直播视频流包括直播画面数据流和互动数据流,所述互动数据流至少包括礼物数据流和弹幕数据流中的一种,所述直播视频流、所述礼物数据流和所述弹幕数据流具有不同的流标识;

创建模块,用于对于所述获取模块得到的每种数据流,创建一个任务线程和一个任务队列,并将所述数据流中属于同一直播视频帧的数据段作为一个任务添加到所述任务队列中,其中每个任务的数据段是在所述数据流中按固定时长截取的,并且不同任务的数据段分别对应所述数据流中的不同时段;

处理模块,用于对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到处理结果,所述处理包括:解封装和解码;

生成模块,用于利用各个任务线程将各自得到的所述处理结果发送给渲染线程;利用所述渲染线程将各个处理结果添加到视频缓冲区中;将所述直播客户端对应的进程的绘制上下文赋值给所述渲染线程;利用所述渲染线程按照所述处理结果被添加到所述视频缓冲区中的顺序,依次从所述视频缓冲区中读取一个处理结果;对于每个处理结果,利用所述渲染线程创建纹理,并将所述纹理添加到所述绘制上下文中,对所述处理结果进行渲染,得到中间数据,将所述中间数据添加到所述纹理中,得到渲染结果;通知图形处理器GPU对各个渲染结果进行渲染得到直播视频帧,所述直播视频帧包括直播画面和互动数据;

播放模块,用于播放所述生成模块生成的所述直播视频帧,

其中当存在至少两个任务线程同时向所述渲染线程发送对应于同一直播视频帧的处理结果时,所述利用所述渲染线程将各个处理结果添加到视频缓冲区中,包括:

利用所述渲染线程接收对应于同一直播视频帧的至少两个处理结果;

利用所述渲染线程确定每个处理结果对应的数据流的处理优先级;

利用所述渲染线程根据所述处理优先级进行轮询,将每次轮询得到的一个处理结果添加到所述视频缓冲区中,从而所述至少两个处理结果按所述处理优先级从高到底的顺序添加到所述视频缓冲区中。

9.一种计算机可读存储介质,其特征在于,所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集,所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现如权利要求1至5任一所述的在线视频的视频帧生成方法,

或者,所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现如权利要求6所述的直播视频的视频帧播放方法。

10.一种电子设备,其特征在于,所述电子设备包括处理器和存储器,所述存储器中存储有至少一条指令,所述指令由所述处理器加载并执行以实现如权利要求1至5任一所述的在线视频的视频帧生成方法,或者,所述指令由所述处理器加载并执行以实现如权利要求6所述的直播视频的视频帧播放方法。

在线视频的视频帧生成方法、装置、存储介质及设备

技术领域

[0001] 本申请实施例涉及计算机技术领域，特别涉及一种在线视频的视频帧生成方法、装置、存储介质及设备。

背景技术

[0002] 直播间支持多人在线语音通信以及视频通信。每个直播间可以包括至少一个主播用户，在至少一个观众用户进入直播间后，直播间还可以包括至少一个观众用户。在直播期间，主播用户与主播用户之间、主播用户和观众用户之间可以在线互动，互动方式包括但不限于发送弹幕和赠送礼物。此时，直播客户端需要将弹幕子数据流、礼物子数据流和主播拍摄的视频子数据流这三种不同类型的子数据流生成一个视频进行播放。这里的直播客户端包括主播用户的直播客户端和/或观众用户的直播客户端。

[0003] 相关技术中，直播客户端接收视频数据流，该视频数据流中包含不同类型的子数据流，且每个子数据流具有不同的流标识；创建一个任务线程，并针对每种子数据流创建对应的任务队列，将每种子数据流中属于同一视频帧的数据段作为一个任务添加到该子数据流对应的任务队列中；利用该任务线程依次对各个任务队列中的任务进行渲染，得到各个渲染结果；通知硬件对各个渲染结果进行渲染，得到视频帧。

[0004] 任务线程在处理一个任务队列中的任务时，其他任务队列中的任务会来抢占该任务线程的处理资源，这样，该任务线程会中断对当前任务的处理，而去处理其他队列中的任务，使得视频帧中当前任务的画面不能被完全显示，从而导致画面播放不流畅。

发明内容

[0005] 本申请实施例提供了一种在线视频的视频帧生成方法、装置、存储介质及设备，用于解决不同任务队列中的任务抢占同一任务线程的处理资源，导致画面播放不流畅的问题。所述技术方案如下：

[0006] 一方面，提供了一种在线视频的视频帧生成方法，所述方法包括：

[0007] 接收视频数据流，所述视频数据流中包含不同类型的子数据流，每种子数据流具有不同的流标识；

[0008] 对于每种子数据流，创建一个任务线程和一个任务队列，并将所述子数据流中属于同一视频帧的数据段作为一个任务添加到所述任务队列中；

[0009] 对于存在任务的任务队列所对应的任务线程，利用所述任务线程对所述任务进行处理，得到处理结果；

[0010] 对各个处理结果进行渲染，得到视频帧。

[0011] 一方面，提供了一种在线视频的视频帧生成装置，所述装置包括：

[0012] 接收模块，用于接收视频数据流，所述视频数据流中包含不同类型的子数据流，每种子数据流具有不同的流标识；

[0013] 创建模块，用于对于所述接收模块接收到的每种子数据流，创建一个任务线程和

一个任务队列,并将所述子数据流中属于同一视频帧的数据段作为一个任务添加到所述任务队列中;

[0014] 处理模块,用于对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到处理结果;

[0015] 生成模块,用于对所述处理模块得到的各个处理结果进行渲染,得到视频帧。

[0016] 一方面,提供了一种直播视频的视频帧播放方法,用于直播客户端中,所述方法包括:

[0017] 获取直播视频流,所述直播视频流包括直播画面数据流和互动数据流,所述互动数据流至少包括礼物数据流和弹幕数据流中的一种,所述直播视频流、所述礼物数据流和所述弹幕数据流具有不同的流标识;

[0018] 对于每种数据流,创建一个任务线程和一个任务队列,并将所述数据流中属于同一直播视频帧的数据段作为一个任务添加到所述任务队列中;

[0019] 对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到处理结果;

[0020] 对各个处理结果进行渲染得到直播视频帧,所述直播视频帧包括直播画面和互动数据;

[0021] 播放所述直播视频帧。

[0022] 一方面,提供了一种直播视频的视频帧播放装置,用于直播客户端中,所述装置包括:

[0023] 获取模块,用于获取直播视频流,所述直播视频流包括直播画面数据流和互动数据流,所述互动数据流至少包括礼物数据流和弹幕数据流中的一种,所述直播视频流、所述礼物数据流和所述弹幕数据流具有不同的流标识;

[0024] 创建模块,用于对于所述获取模块得到的每种数据流,创建一个任务线程和一个任务队列,并将所述数据流中属于同一视频帧的数据段作为一个任务添加到所述任务队列中;

[0025] 处理模块,用于对于存在任务的任务队列所对应的任务线程,利用所述任务线程对所述任务进行处理,得到处理结果;

[0026] 生成模块,用于对所述处理模块得到的各个处理结果进行渲染得到直播视频帧,所述直播视频帧包括直播画面和互动数据;

[0027] 播放模块,用于播放所述生成模块生成的所述直播视频帧。

[0028] 一方面,提供了一种计算机可读存储介质,所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集,所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现如上所述的在线视频的视频帧生成方法,或者,所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现如上所述的直播视频的视频帧播放方法。

[0029] 一方面,提供了一种电子设备,所述电子设备包括处理器和存储器,所述存储器中存储有至少一条指令,所述指令由所述处理器加载并执行以实现如上所述的在线视频的视频帧生成方法,或者,所述指令由所述处理器加载并执行以实现如上所述的直播视频的视频帧播放方法。

[0030] 本申请实施例提供的技术方案的有益效果至少包括：

[0031] 由于任务抢占的是处理该任务的任务线程的处理资源，所以，在对每个任务线程创建一个任务队列时，可以避免不同任务队列中的任务抢占同一任务线程的处理资源的问题，从而避免了由于抢占处理资源导致终端的性能降低、发热发烫的问题，提高了终端的性能。另外，由于每个任务线程可以处理完每个任务，使得视频帧中每个任务的画面能够被完全渲染，从而提高了画面播放的流畅度。另外，通过多个任务线程并行处理任务，可以加快任务的处理进度，避免了画面播放卡顿的问题，提高了画面播放的流畅度。

附图说明

[0032] 为了更清楚地说明本申请实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0033] 图1是根据部分示例性实施例提供的一种电竞类直播客户端的界面示意图；

[0034] 图2是根据部分示例性实施例提供的一种直播间界面的示意图；

[0035] 图3是根据部分示例性实施例提供的一种视频帧生成系统的结构示意图；

[0036] 图4是本申请一个实施例提供的在线视频的视频帧生成方法的方法流程图；

[0037] 图5是本申请一个实施例提供的runloop和任务线程的交互示意图；

[0038] 图6是本申请另一实施例提供的在线视频的视频帧生成方法的方法流程图；

[0039] 图7是本申请另一实施例提供的任务处理和渲染的流程图；

[0040] 图8是本申请另一实施例提供的退出后台时处理流程的示意图；

[0041] 图9是本申请一个实施例提供的直播视频的视频帧播放方法的方法流程图；

[0042] 图10是本申请一个实施例提供的在线视频的视频帧生成装置的结构框图；

[0043] 图11是本申请一个实施例提供的直播视频的视频帧播放装置的结构框图；

[0044] 图12是本申请再一实施例提供的终端的结构框图。

具体实施方式

[0045] 为使本申请实施例的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。

[0046] 在对本申请实施例进行详细地解释说明之前，先对本申请实施例的应用场景进行说明。

[0047] 本申请实施例应用于将至少两种不同类型的数据生成一个在线播放视频进行播放的场景中，下面以其中两种典型的场景进行说明：

[0048] 1、直播场景

[0049] 直播场景包括直播间直播场景和电视直播场景，下面分别对这两种直播场景进行说明。

[0050] 1) 直播间直播场景

[0051] 主播用户通过主播用户的直播客户端进入直播间、观众用户通过观众用户的直播客户端进入直播间后，主播用户与主播用户之间、主播用户和观众用户之间可以在线互动，

互动方式包括但不限于发送弹幕和赠送礼物。此时,直播客户端需要将弹幕子数据流、礼物子数据流和主播拍摄的视频子数据流这三种不同类型的子数据流生成一个视频进行播放,以体现互动的乐趣。这里的直播客户端包括主播用户的直播客户端和/或观众用户的直播客户端。

[0052] 这里的直播客户端可以是用于直播游戏过程的电竞类直播客户端,也可以是用于直播歌舞、脱口秀等娱乐节目的娱乐类直播客户端,本实施例不对直播客户端的类型作限定。

[0053] 以电竞类直播客户端为例,请参考图1,其示出了“王者”游戏的标签页界面,图1中的“BA无情”和“企鹅”是两个直播间,当用户点击“BA无情”时,进入“BA无情”的直播间,从标签页界面跳转到图2所示的直播间界面;请参考图2,其上半部分的区域正在播放“BA无情”的直播间中正在进行的游戏的视频,中间区域显示有弹幕,下半部分的区域显示有礼物类型。

[0054] 需要说明的是,图2以在界面中三个不同的区域显示三种数据进行举例,在实际实现时,还可以将弹幕、礼物叠加显示在视频画面上,本实施例不对各种类型的数据的显示形式作限定。

[0055] 2) 电视直播场景

[0056] 电视在播放节目时除了播放节目数据,还可以播放节目方联系方式的二维码、短信号码等互动数据,电视应用可以将节目数据、互动数据等不同类型的数​​据生成一个视频进行播放。

[0057] 2、视频应用播放场景

[0058] 视频播放应用除了播放视频数据,还可以播放弹幕、与视频数据相关的广告链接、广告语等互动数据,视频播放应用可以将视频数据、互动数据等不同类型的数​​据生成一个视频进行播放。

[0059] 本实施例中,针对每种类型的子数据流创建一个任务线程和一个任务队列,这样,处理任务时,不同任务队列中的任务不会相互抢占任务线程的处理资源,使得每个任务能够被完全处理,从而保证了画面播放的流畅度。

[0060] 接下来对本发明实施例的系统架构进行介绍。

[0061] 请参考图3,其示出了本申请实施例提供的一种视频帧生成系统的结构示意图。该视频帧生成系统包括至少一个终端310和服务器320。终端310与服务器320通过有线网络或无线网络建立连接。

[0062] 终端310是具有数据收发功能的设备,比如智能手机。终端310中可以安装有上面所述的直播客户端、电视客户端、视频播放客户端等。

[0063] 以终端310中安装有直播客户端为例,则直播客户端可以根据用户的不同而分为:主播用户使用的主播客户端和观众用户使用的观众客户端。其中,主播用户是拥有互动权限和直播视频上传权限的用户,可以将互动数据和本地采集到的视频数据上传给服务器,然后由服务器将该互动数据和视频数据转发给主播客户端和观众客户端,主播客户端可以接收到服务器发送的视频数据和互动数据,然后进行展示;观众用户是拥有互动权限的用户,可以将互动数据上传给服务器,由服务器将该互动数据转发给主播客户端和观众客户端,观众客户端可以接收到服务器发送的视频数据和互动数据,然后进行展示。除此之外,

主播用户使用的客户端和观众用户使用的客户端基本相同。

[0064] 服务器320可以是一台或者多台服务器,还可以是一个云计算中心。换句话说,服务器320可以由一台服务器实现;也可以由多台服务器组合实现,每台服务器承担相同或者不同的功能,比如有用于注册和登录的服务器、用于保存用户头像的服务器、用于保存频道信息及配置信息的服务器、用于保存图片或视频的服务器等等;服务器320也可以采用云计算中心实现,云计算中心是一整片服务集群所形成的虚拟计算平台。

[0065] 图3中以示出两个终端310和服务器320来举例说明。

[0066] 请参考图4,其示出了本申请一个实施例提供的在线视频的视频帧生成方法的方法流程图,该在线视频的视频帧生成方法可以应用于图3所示的终端中的客户端中。该在线视频的视频帧生成方法,包括:

[0067] 步骤401,接收视频数据流,该视频数据流中包含不同类型的子数据流,每种子数据流具有不同的流标识。

[0068] 视频数据流是服务器向客户端推送的视频的数据流。比如,用户通过客户端进入直播间后,服务器向客户端推送的视频数据流;或者,用户打开电视后,服务器向客户端推送的视频数据流。

[0069] 不同视频的视频数据流包含不同类型的子数据流,每种子数据流用于提供视频中的一种元素且每种子数据流具有不同的流标识,以示区分。比如,直播间的视频数据流至少包括视频子数据流、弹幕子数据流、礼物子数据流;视频播放应用的视频数据流至少包括视频子数据流、弹幕子数据流、广告子数据流。

[0070] 本实施例中,客户端先向服务器发送视频获取请求,服务器根据该视频获取请求确定视频数据流,将该视频数据流发送给客户端。

[0071] 本实施例中,可以将流标识携带在根据视频数据流中的数据所生成的IP (Internet Protocol,互联网协议) 数据包的包头中,这样,客户端在接收到各个IP数据包后,可以对各个IP数据包的包头进行解析,对于解析到的每种流标识,将具有该流标识的各个IP数据包中的数据组成一种子数据流,最终将视频数据流划分成多个子数据流。

[0072] 步骤402,对于每种子数据流,创建一个任务线程和一个任务队列,并将该子数据流中属于同一视频帧的数据段作为一个任务添加到任务队列中。

[0073] 任务线程是用于对任务进行处理的线程。其中,线程被称为轻量级进程,是程序执行流的最小单元。

[0074] 任务队列是用于缓存任务的队列,每个任务是子数据流中属于同一视频帧的数据段。其中,任务队列在直播间未关闭时始终有效,即,在直播间未关闭期间,始终可以向该任务队列中添加任务,并从该任务队列中读取任务。

[0075] 本实施例中,客户端对每种子数据流创建一个任务线程和一个任务队列。比如,视频的数据流包括视频子数据流、弹幕子数据流和礼物子数据流,则客户端可以为视频子数据流创建一个任务线程和一个任务队列,为弹幕子数据流创建一个任务线程和一个任务队列,为礼物子数据流创建一个任务线程和一个任务队列。

[0076] 对于每个子数据流,客户端可以将该子数据流划分成若干个任务添加到该子数据流对应的任务队列中。在将子数据流划分成若干个任务时,一种实现方式是在子数据流中截取固定时长的数据,将得到的数据段作为一个任务,该数据段即为子数据流中属于一个

视频帧的数据。比如,固定时长是1s,则在接收到子数据流后,截取0-1秒内接收到的数据,将得到的数据段划分为一个任务添加到任务队列中;截取1-2s内接收到的数据,将得到的数据段划分为一个任务添加到任务队列中,依此类推。

[0077] 需要说明的是,在客户端接收视频数据流的期间,持续将该视频数据流划分成任务添加到对应的任务队列中。在此期间,各个任务线程执行步骤403来处理对应的任务队列中的任务。

[0078] 步骤403,对于存在任务的任务队列所对应的任务线程,利用该任务线程对任务进行处理,得到处理结果。

[0079] 本实施例中,在生成任务时,客户端还可以将视频帧的时间戳作为任务的时间戳,这样,可以根据时间戳确定任务线程对任务的处理顺序,下面对任务线程处理任务的过程进行介绍。

[0080] 每个任务线程处理任务时循环执行以下步骤:从对应的任务队列中查找其中时间戳最早的一个任务,启动该任务线程对应的一个runloop(运行循环),将该任务线程注册为该runloop的模式中的注册观察者,将任务注册为该runloop(运行循环)的模式中的输入源,并设置runloop的模式中的定时源,在根据定时源运行该runloop的期间,利用该任务线程对该任务进行解封装、解码等处理,得到一个处理结果。

[0081] 其中,runloop包括所要监听的输入源、定时源和要通知的注册观察者,输入源即为注册的任务,定时源即为任务线程处理一个任务的时长,注册观察者即为任务线程。在runloop每次运行时,会指定一个模式,该模式与输入源对应,与该模式对应的输入源在本次runloop运行时被监测,与该模式不对应的输入源则处于暂停状态,等待与其匹配的模式

的runloop启动时才能被监测。换句话说,运行一次runloop只能监测一类感兴趣的输入源。

[0082] 本实施例中,一个runloop对应于一个任务线程,可以在一个任务线程处理完一个任务后,将下一个任务注册到runloop中,从而通过runloop(运行循环)来保证该任务线程的持续运行。请参考图5,其示出了runloop和任务线程的交互示意图,其中,Input Sources是输入源,Port是基于端口的输入源,Custom是自定义输入源,Timer Sources是定时源,handlePort是处理基于端口的输入源,customSrc是处理用户自定义输入源,mySelector是处理Selector的源,timerFired是处理定时源。

[0083] 需要说明的是,不同任务线程之间可以并行处理各自对应的任务队列中的任务,以加快任务的处理进度。

[0084] 步骤404,对各个处理结果进行渲染,得到视频帧。

[0085] 客户端可以先对各个处理结果进行渲染得到渲染结果,再通知硬件对渲染结果进行渲染得到视频帧。其中,对各个处理结果进行渲染的流程详见步骤604-607中的描述,此处不作赘述。

[0086] 这里的硬件可以是GPU(Graphics Processing Unit,图形处理器),也可以是其他处理器,本实施例不作限定。在硬件渲染得到视频帧后,可以按照一定的帧率对视频帧进行播放,即,对视频帧进行上屏。

[0087] 综上所述,本申请实施例提供的在线视频的视频帧生成方法,由于任务抢占的是处理该任务的任务线程的处理资源,所以,在对每个任务线程创建一个任务队列时,可以避免不同任务队列中的任务抢占同一任务线程的处理资源的问题,从而避免了由于抢占处理

资源导致终端的性能降低、发热发烫的问题,提高了终端的性能。另外,由于每个任务线程可以处理完每个任务,使得视频帧中每个任务的画面能够被完全渲染,从而提高了画面播放的流畅度。另外,通过多个任务线程并行处理任务,可以加快任务的处理进度,避免了画面播放卡顿的问题,提高了画面播放的流畅度。

[0088] 请参考图6,其示出了本申请另一实施例提供的在线视频的视频帧生成方法的方法流程图,该在线视频的视频帧生成方法可以应用于图3所示的终端中的客户端中。该在线视频的视频帧生成方法,包括:

[0089] 步骤601,接收视频数据流,该视频数据流中包含不同类型的子数据流,每种子数据流具有不同的流标识。

[0090] 步骤602,对于每种子数据流,创建一个任务线程和一个任务队列,并将该子数据流中属于同一视频帧的数据段作为一个任务添加到任务队列中。

[0091] 步骤603,对于存在任务的任务队列所对应的任务线程,利用该任务线程对任务进行处理,得到处理结果。

[0092] 其中,步骤601-603的实现流程与步骤401-403的实现流程相同,此处不作赘述。

[0093] 步骤604,利用各个任务线程将各自得到的处理结果发送给渲染线程。

[0094] 渲染线程是用于对任务的处理结果进行渲染的线程。

[0095] 对于每个处理任务的任务线程,在每得到一个处理结果后,即将该处理结果发送给渲染线程。

[0096] 步骤605,利用渲染线程将各个处理结果添加到视频缓冲区中。

[0097] 渲染线程按照接收到处理结果的顺序,将各个处理结果添加到视频缓冲区中,下面对添加过程中的两种情况进行说明。

[0098] 1) 在当前时刻渲染线程接收到一个处理结果时,渲染线程将该处理结果直接添加到视频缓冲区中。

[0099] 比如,视频缓冲区中已缓存的列表为:视频任务1的处理结果->视频任务2的处理结果->礼物任务1的处理结果->礼物任务2的处理结果,若当前时刻渲染线程接收到的处理结果是视频任务3的处理结果,则在将该处理结果添加到视频缓冲区中后,该列表为:视频任务1的处理结果->视频任务2的处理结果->礼物任务1的处理结果->礼物任务2的处理结果->视频任务3的处理结果。

[0100] 2) 在当前时刻渲染线程接收到至少两个处理结果时:

[0101] 若确定该至少两个处理结果对应于不同的时间戳,则按照时间戳的由早到晚的顺序,将该至少两个处理结果依次添加到视频缓冲区中。

[0102] 若确定该至少两个处理结果对应于相同的时间戳,即当存在至少两个任务线程同时向渲染线程发送对应于同一视频帧的处理结果时,利用渲染线程将各个处理结果添加到视频缓冲区中,可以包括如下几个步骤:

[0103] 步骤6051,利用渲染线程接收对应于同一视频帧的至少两个处理结果。

[0104] 步骤6052,利用渲染线程确定每个处理结果对应的子数据流的处理优先级。

[0105] 由于渲染线程是按照处理结果添加到视频缓冲区中的顺序来渲染处理结果的,为了保证同一视频帧中重要的任务的处理结果先被渲染,需要按照子数据流的处理优先级来确定各个处理结果添加到视频缓冲区中的顺序。

[0106] 本实施例中,可以根据视频中各种类型的子数据流的重要性对子数据流设置优先级,子数据流的优先级的高低与重要性成正相关关系。即,子数据流越重要,其优先级越高。仍然以视频数据流包括视频子数据流、礼物子数据流和弹幕子数据流为例进行说明,则可以设置视频子数据流的优先级最高,设置弹幕子数据流的优先级最低。

[0107] 步骤6053,利用渲染线程根据处理优先级进行轮询,将每次轮询得到的一个处理结果添加到视频缓冲区中。

[0108] 比如,渲染线程同时接收到视频任务的处理结果和礼物任务的处理结果,且视频子数据流的优先级高于礼物子数据流的优先级,则渲染线程先轮询是否有视频任务的处理结果,当轮询到视频任务的处理结果时,将该视频任务的处理结果添加到视频缓冲区中;渲染线程再轮询是否有礼物任务的处理结果,当轮询到礼物任务的处理结果时,将该礼物任务的处理结果添加到视频缓冲区中。

[0109] 步骤606,利用渲染线程依次对视频缓冲区中的各个处理结果进行渲染,得到各个渲染结果。

[0110] 其中,利用渲染线程依次对视频缓冲区中的各个处理结果进行渲染,得到各个渲染结果,可以包括如下几个步骤:

[0111] 步骤6061,将客户端对应的进程的绘制上下文赋值给渲染线程。

[0112] 绘制上下文用于生成上下文环境,在GPU对渲染结果进行渲染上屏时用到。其中,一个进程对应于一个绘制上下文,这里的进程是客户端对应的进程,即创建任务线程和渲染线程的进程。

[0113] 步骤6062,利用渲染线程按照处理结果被添加到视频缓冲区中的顺序,依次从视频缓冲区中读取一个处理结果。

[0114] 步骤6063,对于每个处理结果,利用渲染线程创建纹理,并将纹理添加到绘制上下文中,对处理结果进行渲染,得到中间数据,将中间数据添加到纹理中,得到渲染结果。

[0115] 计算机图形学中的纹理既包括通常意义上物体表面的纹理,即使物体表面呈现凹凸不平的沟纹,也包括在物体的光滑表面上的彩色图案,换句话说,纹理是指物体表面的花纹。

[0116] 这里的中间数据可以是RGB (Red Green Blue,红绿蓝) 数据。

[0117] 仍然以步骤605中的列表为例,则先将绘制上下文赋值给渲染线程,渲染线程读取视频任务1的处理结果,创建纹理,并将纹理添加到绘制上下文中,对视频任务1的处理结果进行渲染,得到RGB,将RGB数据添加到纹理中,得到视频任务1的渲染结果;渲染线程读取视频任务2的处理结果,创建纹理,并将纹理添加到绘制上下文中,对视频任务2的处理结果进行渲染,得到RGB,将RGB数据添加到纹理中,得到视频任务2的渲染结果;渲染线程读取礼物任务1的处理结果,创建纹理,并将纹理添加到绘制上下文中,对礼物任务1的处理结果进行渲染,得到RGB,将RGB数据添加到纹理中,得到礼物任务1的渲染结果,依此类推。

[0118] 请参考图7,其示出了建立任务队列、对各个任务进行处理、设置用于轮询的处理优先级、轮询处理结果添加到视频缓冲区中,并进行渲染的流程图。

[0119] 步骤607,在每次得到一个视频帧对应的各个渲染结果时,向GPU发送开始渲染指令,开始渲染指令用于指示GPU开始渲染视频帧对应的各个渲染结果,得到视频帧。

[0120] 本实施例中,在得到各个渲染结果后,客户端可以通知GPU立即对渲染结果进行渲

染和播放,由于此时同一视频帧的其他渲染结果可能还没准备好,会导致画面播放不流畅,所以,可选的,还可以在每得到一个视频帧对应的所有渲染结果后,再通知GPU对该视频帧对应的所有渲染结果进行渲染,得到视频帧后再播放。

[0121] 当客户端接收到用户发出的后台运行指令时,根据该后台运行指令退出到后台运行。相关技术中,若客户端不关闭线程,则线程会继续对任务进行处理,并通知GPU生成视频帧进行播放,由于终端的操作系统不允许后台运行的客户端继续播放视频帧,所以,操作系统会造成客户端崩溃来制止客户端播放视频帧。若客户端关闭线程,则用户可能会忘记正在后台运行客户端,减少客户端的使用率。本实施例中,在客户端退出到后台运行时,还执行步骤608,以解决上述问题。

[0122] 步骤608,当接收到后台运行指令时,向GPU发送停止渲染指令,停止渲染指令用于指示GPU停止对渲染结果进行渲染。

[0123] 本实施例中,客户端在后台运行时不关闭线程,且任务线程会继续对任务进行处理,并将处理结果发送给渲染线程,渲染线程将处理结果添加到视频缓冲区中,并指示GPU停止对渲染结果进行渲染。这样,由于GPU不对渲染结果进行渲染,也就不会播放视频帧,操作系统不会造成客户端崩溃。另外,由于客户端不关闭线程,所以,诸如音频处理之类的线程还可以继续输出视频的声音,以提示用户该客户端还在运行,从而提高客户端的使用率。

[0124] 可选的,该方法还包括:当接收到后台运行指令时,通知渲染线程停止轮询。

[0125] 轮询的目的是将同一视频帧中重要的任务的处理结果优先添加到视频缓冲区中,以优先对该处理结果进行渲染得到渲染结果,使得硬件优先渲染并播放该渲染结果。在客户端在后台运行时,由于GPU不需要对渲染结果进行渲染和播放,所以,不需要关注处理结果添加到视频缓冲区中的顺序,即,渲染线程可以随机将同一视频帧对应的至少两个处理结果添加到视频缓冲区中,此时,客户端可以通知渲染线程停止轮询,以节省资源。

[0126] 需要说明的是,渲染线程在同时接收到至少两个处理结果时即开始轮询,不用向渲染线程发送指令以通知其开始轮询。

[0127] 可选的,该方法还包括:当接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据,将绘制上下文赋值给进程。

[0128] 在客户端在后台运行时,还需要清除纹理数据,否则,残留的纹理数据将影响视频缓冲区。

[0129] 在一种可能的实现方式,在客户端接收到后台运行指令时,通知渲染线程停止轮询,渲染线程停止轮询,清理绘制上下文中的纹理数据,将绘制上下文赋值给进程,并向GPU发送停止渲染指令,以指示GPU停止对渲染结果进行渲染,请参考图8所示的流程。

[0130] 在一个实例中,使用本实施例提供的方法后,CPU(Central Processing Unit,中央处理器)的占用率从未使用该方法时测得的38.9%降低到25.6%,且内存从未使用该方法时测得的250MB降低到190MB,大大提升了终端的性能。另外,在未使用本实施例提供的方法之前,客户端退出到后台运行时有0.5%的崩溃率,在使用本实施例提供的方法后,客户端退出到后台运行时不会产生崩溃。

[0131] 综上所述,本申请实施例提供的在线视频的视频帧生成方法,由于任务抢占的是处理该任务的任务线程的处理资源,所以,在对每个任务线程创建一个任务队列时,可以避免不同任务队列中的任务抢占同一任务线程的处理资源的问题,从而避免了由于抢占处理

资源导致终端的性能降低、发热发烫的问题,提高了终端的性能。另外,由于每个任务线程可以处理完每个任务,使得视频帧中每个任务的画面能够被完全渲染,从而提高了画面播放的流畅度。另外,通过多个任务线程并行处理任务,可以加快任务的处理进度,避免了画面播放卡顿的问题,提高了画面播放的流畅度。

[0132] 当接收到后台运行指令时,通知渲染线程停止轮询,以节省资源。

[0133] 当接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据以避免残留的纹理数据对视频缓冲区的影响。

[0134] 请参考图9,其示出了本申请另一实施例提供的直播视频的视频帧播放方法的方法流程图,该直播视频的视频帧播放方法可以应用于图3所示的终端中的直播客户端中。该直播视频的视频帧播放方法,包括:

[0135] 步骤901,获取直播视频流,该直播视频流包括直播画面数据流和互动数据流,该互动数据流至少包括礼物数据流和弹幕数据流中的一种,直播视频流、礼物数据流和弹幕数据流具有不同的流标识。

[0136] 直播视频流是服务器向直播客户端推送的、用于在直播间中播放的视频流。

[0137] 直播视频流除了包括直播画面数据流,还包括主播用户和观众用户在互动时产生的互动数据流。可选的,互动数据流至少包括礼物数据流和弹幕数据流中的一种。其中,礼物数据流是赠送礼物时产生的数据流,弹幕数据流是发送弹幕时产生的数据流。

[0138] 在一种可能的实现方式中,直播客户端先向服务器发送直播视频获取请求,服务器根据该直播视频获取请求确定直播视频流,将该直播视频流发送给直播客户端。

[0139] 步骤902,对于每种数据流,创建一个任务线程和一个任务队列,并将该数据流中属于同一直播视频帧的数据段作为一个任务添加到任务队列中。

[0140] 本实施例中,直播客户端对每种数据流创建一个任务线程和一个任务队列。比如,直播视频流包括直播画面数据流、弹幕数据流和礼物数据流,则直播客户端可以为直播画面数据流创建一个任务线程和一个任务队列,为弹幕数据流创建一个任务线程和一个任务队列,为礼物数据流创建一个任务线程和一个任务队列。

[0141] 对于每个数据流,直播客户端可以将该数据流划分成若干个任务添加到该数据流对应的任务队列中。

[0142] 在将数据流划分成若干个任务时,一种可能的实现方式是在数据流中截取固定时长的数据,将得到的数据段作为一个任务,该数据段即为数据流中属于一个直播视频帧的数据。比如,固定时长是1s,则在接收到数据流后,截取0-1秒内接收到的数据,将得到的数据段划分为一个任务添加到任务队列中;截取1-2s内接收到的数据,将得到的数据段划分为一个任务添加到任务队列中,依此类推。

[0143] 步骤903,对于存在任务的任务队列所对应的任务线程,利用该任务线程对任务进行处理,得到处理结果。

[0144] 其中,步骤903的实现流程与步骤403的实现流程相同,此处不作赘述。

[0145] 步骤904,利用各个任务线程将各自得到的处理结果发送给渲染线程。

[0146] 对于每个处理任务的任务线程,在任务线程每得到一个处理结果后,即将该处理结果发送给渲染线程。

[0147] 步骤905,利用渲染线程将各个处理结果添加到视频缓冲区中。

[0148] 渲染线程按照接收到处理结果的顺序,将各个处理结果添加到视频缓冲区中,下面对添加过程中的两种情况进行说明。

[0149] 1) 在当前时刻渲染线程接收到一个处理结果时,渲染线程将该处理结果直接添加到视频缓冲区中。

[0150] 2) 在当前时刻渲染线程接收到至少两个处理结果时:

[0151] 若确定该至少两个处理结果对应于不同的时间戳,则按照时间戳的由早到晚的顺序,将该至少两个处理结果依次添加到视频缓冲区中。

[0152] 若确定该至少两个处理结果对应于相同的时间戳,即当存在至少两个任务线程同时向渲染线程发送对应于同一直播视频帧的处理结果时,利用渲染线程将各个处理结果添加到视频缓冲区中,可以包括如下几个步骤:

[0153] 步骤9051,利用渲染线程接收对应于同一直播视频帧的至少两个处理结果。

[0154] 步骤9052,利用渲染线程确定每个处理结果对应的数据流的处理优先级。

[0155] 由于渲染线程是按照处理结果添加到视频缓冲区中的顺序来渲染处理结果的,为了保证同一直播视频帧中重要的任务的处理结果先被渲染,可选的,可以按照数据流的处理优先级来确定各个处理结果添加到视频缓冲区中的顺序。

[0156] 本实施例中,可以根据各个数据流的重要性设置优先级,数据流的优先级的高低与重要性成正相关关系。即,数据流越重要,其优先级越高。仍然以直播视频流包括直播画面数据流、礼物数据流和弹幕数据流为例进行说明,则可以设置直播画面数据流的优先级最高,设置弹幕数据流的优先级最低。

[0157] 步骤9053,利用渲染线程根据处理优先级进行轮询,将每次轮询得到的一个处理结果添加到视频缓冲区中。

[0158] 比如,渲染线程同时接收到直播画面任务的处理结果和礼物任务的处理结果,且直播画面数据流的优先级高于礼物数据流的优先级,则渲染线程先轮询是否有直播画面任务的处理结果,当轮询到直播画面任务的处理结果时,将该直播画面任务的处理结果添加到视频缓冲区中;渲染线程再轮询是否有礼物任务的处理结果,当轮询到礼物任务的处理结果时,将该礼物任务的处理结果添加到视频缓冲区中。

[0159] 步骤906,利用渲染线程依次对视频缓冲区中的各个处理结果进行渲染,得到各个渲染结果。

[0160] 其中,利用渲染线程依次对视频缓冲区中的各个处理结果进行渲染,得到各个渲染结果,可以包括如下几个步骤:

[0161] 步骤9061,将直播客户端对应的进程的绘制上下文赋值给渲染线程。

[0162] 绘制上下文用于生成上下文环境,在GPU对渲染结果进行渲染上屏时用到。其中,一个进程对应于一个绘制上下文,这里的进程是直播客户端对应的进程,即创建任务线程和渲染线程的进程。

[0163] 步骤9062,利用渲染线程按照处理结果被添加到视频缓冲区中的顺序,依次从视频缓冲区中读取一个处理结果。

[0164] 步骤9063,对于每个处理结果,利用渲染线程创建纹理,并将纹理添加到绘制上下文中,对处理结果进行渲染,得到中间数据,将中间数据添加到纹理中,得到渲染结果。

[0165] 步骤907,在每次得到一个直播视频帧对应的各个渲染结果时,向GPU发送开始渲

染指令,开始渲染指令用于指示GPU开始渲染直播视频帧对应的各个渲染结果,得到直播视频帧,该直播视频帧包括直播画面和互动数据。

[0166] 本实施例中,在得到各个渲染结果后,直播客户端可以通知GPU立即对渲染结果进行渲染和播放,由于此时同一直播视频帧的其他渲染结果可能还没准备好,会导致画面播放不流畅,所以,可选的,还可以在每得到一个直播视频帧对应的所有渲染结果后,再通知GPU对该直播视频帧对应的所有渲染结果进行渲染,得到直播视频帧。

[0167] 步骤908,播放直播视频帧。

[0168] 当直播客户端接收到用户发出的后台运行指令时,根据该后台运行指令退出到后台运行。相关技术中,若直播客户端不关闭线程,则任务线程和渲染线程会继续对任务进行处理,并通知GPU生成直播视频帧进行播放,由于终端的操作系统不允许后台运行的直播客户端继续播放直播视频帧,所以,操作系统会造成直播客户端崩溃来制止直播客户端播放直播视频帧。若直播客户端关闭线程,则用户可能会忘记正在后台运行直播客户端,减少直播客户端的使用率。本实施例中,在直播客户端退出到后台运行时,还执行步骤608,以解决上述问题。

[0169] 步骤909,当接收到后台运行指令时,向GPU发送停止渲染指令,停止渲染指令用于指示GPU停止对渲染结果进行渲染。

[0170] 本实施例中,直播客户端在后台运行时不关闭线程,且任务线程会继续对任务进行处理,并将处理结果发送给渲染线程,渲染线程将处理结果添加到视频缓冲区中,并指示GPU停止对渲染结果进行渲染。这样,由于GPU不对渲染结果进行渲染,也就不会播放视频帧,操作系统不会造成直播客户端崩溃。另外,由于直播客户端不关闭线程,所以,诸如音频处理之类的线程还可以继续输出视频的声音,以提示用户该直播客户端还在运行,从而提高直播客户端的使用率。

[0171] 可选的,该方法还包括:当接收到后台运行指令时,通知渲染线程停止轮询。

[0172] 轮询的目的是将同一直播视频帧中重要的任务的处理结果优先添加到视频缓冲区中,以优先对该处理结果进行渲染得到渲染结果,使得GPU优先渲染并播放该渲染结果。在直播客户端在后台运行时,由于GPU不需要对渲染结果进行渲染和播放,所以,不需要关注处理结果添加到视频缓冲区中的顺序,即,渲染线程可以随机将同一直播视频帧对应的至少两个处理结果添加到视频缓冲区中,此时,直播客户端可以通知渲染线程停止轮询,以节省资源。

[0173] 需要说明的是,渲染线程在同时接收到至少两个处理结果时即开始轮询,不用向渲染线程发送指令以通知其开始轮询。

[0174] 可选的,该方法还包括:当接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据,将绘制上下文赋值给进程。

[0175] 在直播客户端在后台运行时,还需要清除纹理数据,否则,残留的纹理数据将影响视频缓冲区。

[0176] 在一种可能的实现方式,在直播客户端接收到后台运行指令时,通知渲染线程停止轮询,渲染线程停止轮询,清理绘制上下文中的纹理数据,将绘制上下文赋值给进程,并向GPU发送停止渲染指令,以指示GPU停止对渲染结果进行渲染。

[0177] 综上所述,本申请实施例提供的直播视频的视频帧播放方法,由于任务抢占的是

处理该任务的任务线程的处理资源,所以,在对每个任务线程创建一个任务队列时,可以避免不同任务队列中的任务抢占同一任务线程的处理资源的问题,从而避免了由于抢占处理资源导致终端的性能降低、发热发烫的问题,提高了终端的性能。另外,由于每个任务线程可以处理完每个任务,使得直播视频帧中每个任务的画面能够被完全渲染,从而提高了直播画面播放的流畅度。另外,通过多个任务线程并行处理任务,可以加快任务的处理进度,避免了直播画面播放卡顿的问题,提高了直播画面播放的流畅度。

[0178] 当接收到后台运行指令时,通知渲染线程停止轮询,以节省资源。

[0179] 当接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据以避免残留的纹理数据对视频缓冲区的影响。

[0180] 请参考图10,其示出了本申请一个实施例提供的在线视频的视频帧生成装置的结构框图,该在线视频的视频帧生成装置可以应用于图3所示的终端中的客户端中。该在线视频的视频帧生成装置,包括:

[0181] 接收模块1010,用于接收视频数据流,视频数据流中包含不同类型的子数据流,每种子数据流具有不同的流标识;

[0182] 创建模块1020,用于对于接收模块1010接收到的每种子数据流,创建一个任务线程和一个任务队列,并将子数据流中属于同一视频帧的数据段作为一个任务添加到任务队列中;

[0183] 处理模块1030,用于对于存在任务的任务队列所对应的任务线程,利用任务线程对任务进行处理,得到处理结果;

[0184] 生成模块1040,用于对处理模块1030得到的各个处理结果进行渲染,得到视频帧。

[0185] 可选的,生成模块1040,还用于:

[0186] 利用各个任务线程将各自得到的处理结果发送给渲染线程;

[0187] 利用渲染线程将各个处理结果添加到视频缓冲区中;

[0188] 利用渲染线程依次对视频缓冲区中的各个处理结果进行渲染,得到各个渲染结果;

[0189] 通知GPU对各个渲染结果进行渲染,得到视频帧。

[0190] 可选的,当存在至少两个任务线程同时向渲染线程发送对应于同一视频帧的处理结果时,生成模块1040,还用于:

[0191] 利用渲染线程接收对应于同一视频帧的至少两个处理结果;

[0192] 利用渲染线程确定每个处理结果对应的子数据流的处理优先级;

[0193] 利用渲染线程根据处理优先级进行轮询,将每次轮询得到的一个处理结果添加到视频缓冲区中。

[0194] 可选的,生成模块1040,还用于在接收到后台运行指令时,通知渲染线程停止轮询。

[0195] 可选的,生成模块1040,还用于:

[0196] 将客户端对应的进程的绘制上下文赋值给渲染线程;

[0197] 利用渲染线程按照处理结果被添加到视频缓冲区中的顺序,依次从视频缓冲区中读取一个处理结果;

[0198] 对于每个处理结果,利用渲染线程创建纹理,并将纹理添加到绘制上下文中,对处

理结果进行渲染,得到中间数据,将中间数据添加到纹理中,得到渲染结果。

[0199] 可选的,该装置还包括:

[0200] 清除模块,用于在接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据,将绘制上下文赋值给进程。

[0201] 可选的,生成模块1040,还用于在每次得到一个视频帧对应的各个渲染结果时,向硬件发送开始渲染指令,开始渲染指令用于指示GPU开始渲染视频帧对应的各个渲染结果,得到视频帧。

[0202] 可选的,该装置还包括:

[0203] 发送模块,用于在接收到后台运行指令时,向GPU发送停止渲染指令,停止渲染指令用于指示GPU停止对渲染结果进行渲染。

[0204] 综上所述,本申请实施例提供的在线视频的视频帧生成装置,由于任务抢占的是处理该任务的任务线程的处理资源,所以,在对每个任务线程创建一个任务队列时,可以避免不同任务队列中的任务抢占同一任务线程的处理资源的问题,从而避免了由于抢占处理资源导致终端的性能降低、发热发烫的问题,提高了终端的性能。另外,由于每个任务线程可以处理完每个任务,使得视频帧中每个任务的画面能够被完全渲染,从而提高了画面播放的流畅度。另外,通过多个任务线程并行处理任务,可以加快任务的处理进度,避免了画面播放卡顿的问题,提高了画面播放的流畅度。

[0205] 当接收到后台运行指令时,通知渲染线程停止轮询,以节省资源。

[0206] 当接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据以避免残留的纹理数据对视频缓冲区的影响。

[0207] 请参考图11,其示出了本申请一个实施例提供的直播视频的视频帧播放装置的结构框图,该直播视频的视频帧播放装置可以应用于图3所示的终端中的直播客户端中。该直播视频的视频帧播放装置,包括:

[0208] 获取模块1110,用于获取直播视频流,直播视频流包括直播画面数据流和互动数据流,互动数据流至少包括礼物数据流和弹幕数据流中的一种,直播视频流、礼物数据流和弹幕数据流具有不同的流标识;

[0209] 创建模块1120,用于对于获取模块1110得到的每种数据流,创建一个任务线程和一个任务队列,并将数据流中属于同一直播视频帧的数据段作为一个任务添加到任务队列中;

[0210] 处理模块1130,用于对于存在任务的任务队列所对应的任务线程,利用任务线程对任务进行处理,得到处理结果;

[0211] 生成模块1140,用于对处理模块1130得到的各个处理结果进行渲染得到直播视频帧,直播视频帧包括直播画面和互动数据;

[0212] 播放模块1150,用于播放生成模块1140生成的直播视频帧。

[0213] 可选的,生成模块1140,还用于:

[0214] 利用各个任务线程将各自得到的处理结果发送给渲染线程;

[0215] 利用渲染线程将各个处理结果添加到视频缓冲区中;

[0216] 利用渲染线程依次对视频缓冲区中的各个处理结果进行渲染,得到各个渲染结果;

- [0217] 通知GPU对各个渲染结果进行渲染,得到直播视频帧。
- [0218] 可选的,当存在至少两个任务线程同时向渲染线程发送对应于同一直播视频帧的处理结果时,生成模块1040,还用于:
- [0219] 利用渲染线程接收对应于同一直播视频帧的至少两个处理结果;
- [0220] 利用渲染线程确定每个处理结果对应的子数据流的处理优先级;
- [0221] 利用渲染线程根据处理优先级进行轮询,将每次轮询得到的一个处理结果添加到视频缓冲区中。
- [0222] 可选的,生成模块1040,还用于在接收到后台运行指令时,通知渲染线程停止轮询。
- [0223] 可选的,生成模块1040,还用于:
- [0224] 将直播客户端对应的进程的绘制上下文赋值给渲染线程;
- [0225] 利用渲染线程按照处理结果被添加到视频缓冲区中的顺序,依次从视频缓冲区中读取一个处理结果;
- [0226] 对于每个处理结果,利用渲染线程创建纹理,并将纹理添加到绘制上下文中,对处理结果进行渲染,得到中间数据,将中间数据添加到纹理中,得到渲染结果。
- [0227] 可选的,该装置还包括:
- [0228] 清除模块,用于在接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据,将绘制上下文赋值给进程。
- [0229] 可选的,生成模块1040,还用于在每次得到一个直播视频帧对应的各个渲染结果时,向GPU发送开始渲染指令,开始渲染指令用于指示GPU开始渲染直播视频帧对应的各个渲染结果,得到直播视频帧。
- [0230] 可选的,该装置还包括:
- [0231] 发送模块,用于在接收到后台运行指令时,向GPU发送停止渲染指令,停止渲染指令用于指示GPU停止对渲染结果进行渲染。
- [0232] 综上所述,本申请实施例提供的直播视频的视频帧播放装置,由于任务抢占的是处理该任务的任务线程的处理资源,所以,在对每个任务线程创建一个任务队列时,可以避免不同任务队列中的任务抢占同一任务线程的处理资源的问题,从而避免了由于抢占处理资源导致终端的性能降低、发热发烫的问题,提高了终端的性能。另外,由于每个任务线程可以处理完每个任务,使得视频帧中每个任务的画面能够被完全渲染,从而提高了画面播放的流畅度。另外,通过多个任务线程并行处理任务,可以加快任务的处理进度,避免了画面播放卡顿的问题,提高了画面播放的流畅度。
- [0233] 当接收到后台运行指令时,通知渲染线程停止轮询,以节省资源。
- [0234] 当接收到后台运行指令时,利用渲染线程清除绘制上下文中的纹理数据以避免残留的纹理数据对视频缓冲区的影响。
- [0235] 图12示出了本申请一个示例性实施例提供的终端1200的结构框图。该终端1200可以是便携式移动终端,比如:智能手机、平板电脑、MP3播放器(Moving Picture Experts Group Audio Layer III,动态影像专家压缩标准音频层面3)、MP4(Moving Picture Experts Group Audio Layer IV,动态影像专家压缩标准音频层面4)播放器、笔记本电脑或台式电脑。终端1200还可能被称为用户设备、便携式终端、膝上型终端、台式终端等其他

名称。

[0236] 通常,终端1200包括有:处理器1201和存储器1202。

[0237] 处理器1201可以包括一个或多个处理核心,比如4核心处理器、8核心处理器等。处理器1201可以采用DSP (Digital Signal Processing, 数字信号处理)、FPGA (Field-Programmable Gate Array, 现场可编程门阵列)、PLA (Programmable Logic Array, 可编程逻辑阵列) 中的至少一种硬件形式来实现。处理器1201也可以包括主处理器和协处理器,主处理器是用于对在唤醒状态下的数据进行处理的处理单元,也称CPU (Central Processing Unit, 中央处理器);协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中,处理器1201可以在集成有GPU (Graphics Processing Unit, 图像处理器), GPU用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中,处理器1201还可以包括AI (Artificial Intelligence, 人工智能) 处理器,该AI处理器用于处理有关机器学习的计算操作。

[0238] 存储器1202可以包括一个或多个计算机可读存储介质,该计算机可读存储介质可以是非暂态的。存储器1202还可包括高速随机存取存储器,以及非易失性存储器,比如一个或多个磁盘存储设备、闪存存储设备。在一些实施例中,存储器1202中的非暂态的计算机可读存储介质用于存储至少一个指令,该至少一个指令用于被处理器1201所执行以实现本申请中方法实施例提供的在线视频的视频帧生成方法和/或直播视频的视频帧播放方法。

[0239] 在一些实施例中,终端1200还可选包括有:外围设备接口1203和至少一个外围设备。处理器1201、存储器1202和外围设备接口1203之间可以通过总线或信号线相连。各个外围设备可以通过总线、信号线或电路板与外围设备接口1203相连。具体地,外围设备包括:射频电路1204、触摸显示屏1205、摄像头1206、音频电路1207、定位组件1208和电源1209中的至少一种。

[0240] 外围设备接口1203可被用于将I/O (Input/Output, 输入/输出) 相关的至少一个外围设备连接到处理器1201和存储器1202。在一些实施例中,处理器1201、存储器1202和外围设备接口1203被集成在同一芯片或电路板上;在一些其他实施例中,处理器1201、存储器1202和外围设备接口1203中的任意一个或两个可以在单独的芯片或电路板上实现,本实施例对此不加以限定。

[0241] 射频电路1204用于接收和发射RF (Radio Frequency, 射频) 信号,也称电磁信号。射频电路1204通过电磁信号与通信网络以及其他通信设备进行通信。射频电路1204将电信号转换为电磁信号进行发送,或者,将接收到的电磁信号转换为电信号。可选地,射频电路1204包括:天线系统、RF收发器、一个或多个放大器、调谐器、振荡器、数字信号处理器、编解码芯片组、用户身份模块卡等等。射频电路1204可以通过至少一种无线通信协议来与其它终端进行通信。该无线通信协议包括但不限于:城域网、各代移动通信网络 (2G、3G、4G及5G)、无线局域网和/或WiFi (Wireless Fidelity, 无线保真) 网络。在一些实施例中,射频电路1204还可以包括NFC (Near Field Communication, 近距离无线通信) 有关的电路,本申请对此不加以限定。

[0242] 显示屏1205用于显示UI (User Interface, 用户界面)。该UI可以包括图形、文本、图标、视频及其它们的任意组合。当显示屏1205是触摸显示屏时,显示屏1205还具有采集在显示屏1205的表面或表面上方的触摸信号的能力。该触摸信号可以作为控制信号输入至处

理器1201进行处理。此时,显示屏1205还可以用于提供虚拟按钮和/或虚拟键盘,也称软按钮和/或软键盘。在一些实施例中,显示屏1205可以为一个,设置终端1200的前面板;在另一些实施例中,显示屏1205可以为至少两个,分别设置在终端1200的不同表面或呈折叠设计;在再一些实施例中,显示屏1205可以是柔性显示屏,设置在终端1200的弯曲表面上或折叠面上。甚至,显示屏1205还可以设置成非矩形的不规则图形,也即异形屏。显示屏1205可以采用LCD(Liquid Crystal Display,液晶显示屏)、OLED(Organic Light-Emitting Diode,有机发光二极管)等材质制备。

[0243] 摄像头组件1206用于采集图像或视频。可选地,摄像头组件1206包括前置摄像头和后置摄像头。通常,前置摄像头设置在终端的前面板,后置摄像头设置在终端的背面。在一些实施例中,后置摄像头为至少两个,分别为主摄像头、景深摄像头、广角摄像头、长焦摄像头中的任意一种,以实现主摄像头和景深摄像头融合实现背景虚化功能、主摄像头和广角摄像头融合实现全景拍摄以及VR(Virtual Reality,虚拟现实)拍摄功能或者其它融合拍摄功能。在一些实施例中,摄像头组件1206还可以包括闪光灯。闪光灯可以是单色温闪光灯,也可以是双色温闪光灯。双色温闪光灯是指暖光闪光灯和冷光闪光灯的组合,可以用于不同色温下的光线补偿。

[0244] 音频电路1207可以包括麦克风和扬声器。麦克风用于采集用户及环境的声波,并将声波转换为电信号输入至处理器1201进行处理,或者输入至射频电路1204以实现语音通信。出于立体声采集或降噪的目的,麦克风可以为多个,分别设置在终端1200的不同部位。麦克风还可以是阵列麦克风或全向采集型麦克风。扬声器则用于将来自处理器1201或射频电路1204的电信号转换为声波。扬声器可以是传统的薄膜扬声器,也可以是压电陶瓷扬声器。当扬声器是压电陶瓷扬声器时,不仅可以将电信号转换为人类可听见的声波,也可以将电信号转换为人类听不见的声波以进行测距等用途。在一些实施例中,音频电路1207还可以包括耳机插孔。

[0245] 定位组件1208用于定位终端1200的当前地理位置,以实现导航或LBS(Location Based Service,基于位置的服务)。定位组件1208可以是基于美国的GPS(Global Positioning System,全球定位系统)、中国的北斗系统、俄罗斯的格雷纳斯系统或欧盟的伽利略系统的定位组件。

[0246] 电源1209用于为终端1200中的各个组件进行供电。电源1209可以是交流电、直流电、一次性电池或可充电电池。当电源1209包括可充电电池时,该可充电电池可以支持有线充电或无线充电。该可充电电池还可以用于支持快充技术。

[0247] 在一些实施例中,终端1200还包括有一个或多个传感器1210。该一个或多个传感器1210包括但不限于:加速度传感器1211、陀螺仪传感器1212、压力传感器1213、指纹传感器1214、光学传感器1215以及接近传感器1216。

[0248] 加速度传感器1211可以检测以终端1200建立的坐标系的三个坐标轴上的加速度大小。比如,加速度传感器1211可以用于检测重力加速度在三个坐标轴上的分量。处理器1201可以根据加速度传感器1211采集的重力加速度信号,控制触摸显示屏1205以横向视图或纵向视图进行用户界面的显示。加速度传感器1211还可以用于游戏或者用户的运动数据的采集。

[0249] 陀螺仪传感器1212可以检测终端1200的机体方向及转动角度,陀螺仪传感器1212

可以与加速度传感器1211协同采集用户对终端1200的3D动作。处理器1201根据陀螺仪传感器1212采集的数据,可以实现如下功能:动作感应(比如根据用户的倾斜操作来改变UI)、拍摄时的图像稳定、游戏控制以及惯性导航。

[0250] 压力传感器1213可以设置在终端1200的侧边框和/或触摸显示屏1205的下层。当压力传感器1213设置在终端1200的侧边框时,可以检测用户对终端1200的握持信号,由处理器1201根据压力传感器1213采集的握持信号进行左右手识别或快捷操作。当压力传感器1213设置在触摸显示屏1205的下层时,由处理器1201根据用户对触摸显示屏1205的压力操作,实现对UI界面上的可操作性控件进行控制。可操作性控件包括按钮控件、滚动条控件、图标控件、菜单控件中的至少一种。

[0251] 指纹传感器1214用于采集用户的指纹,由处理器1201根据指纹传感器1214采集到的指纹识别用户的身份,或者,由指纹传感器1214根据采集到的指纹识别用户的身份。在识别出用户的身份为可信身份时,由处理器1201授权该用户执行相关的敏感操作,该敏感操作包括解锁屏幕、查看加密信息、下载软件、支付及更改设置等。指纹传感器1214可以被设置终端1200的正面、背面或侧面。当终端1200上设置有物理按键或厂商Logo时,指纹传感器1214可以与物理按键或厂商Logo集成在一起。

[0252] 光学传感器1215用于采集环境光强度。在一个实施例中,处理器1201可以根据光学传感器1215采集的环境光强度,控制触摸显示屏1205的显示亮度。具体地,当环境光强度较高时,调高触摸显示屏1205的显示亮度;当环境光强度较低时,调低触摸显示屏1205的显示亮度。在另一个实施例中,处理器1201还可以根据光学传感器1215采集的环境光强度,动态调整摄像头组件1206的拍摄参数。

[0253] 接近传感器1216,也称距离传感器,通常设置在终端1200的前面板。接近传感器1216用于采集用户与终端1200的正面之间的距离。在一个实施例中,当接近传感器1216检测到用户与终端1200的正面之间的距离逐渐变小时,由处理器1201控制触摸显示屏1205从亮屏状态切换为息屏状态;当接近传感器1216检测到用户与终端1200的正面之间的距离逐渐变大时,由处理器1201控制触摸显示屏1205从息屏状态切换为亮屏状态。

[0254] 本领域技术人员可以理解,图12中示出的结构并不构成对终端1200的限定,可以包括比图示更多或更少的组件,或者组合某些组件,或者采用不同的组件布置。

[0255] 本申请一个实施例提供了一种计算机可读存储介质,所述存储介质中存储有至少一条指令、至少一段程序、代码集或指令集,所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现如上所述的在线视频的视频帧生成方法,或者,所述至少一条指令、所述至少一段程序、所述代码集或指令集由所述处理器加载并执行以实现如上所述的直播视频的视频帧播放方法。

[0256] 本申请一个实施例提供了一种电子设备,所述电子设备包括处理器和存储器,所述存储器中存储有至少一条指令,所述指令由所述处理器加载并执行以实现如上所述的在线视频的视频帧生成方法,或者,所述指令由所述处理器加载并执行以实现如上所述的直播视频的视频帧播放方法。

[0257] 需要说明的是:上述实施例提供的在线视频的视频帧生成装置在进行视频帧生成、直播视频的视频帧播放装置在播放视频帧时,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,即将在线视频的

视频帧生成装置的内部结构划分成不同的功能模块,将直播视频的视频帧播放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。另外,上述实施例提供的在线视频的视频帧生成装置与在线视频的视频帧生成方法实施例属于同一构思,直播视频的视频帧播放装置与直播视频的视频帧播放方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0258] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,所述的程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0259] 以上所述并不用以限制本申请实施例,凡在本申请实施例的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请实施例的保护范围之内。



图1



图2

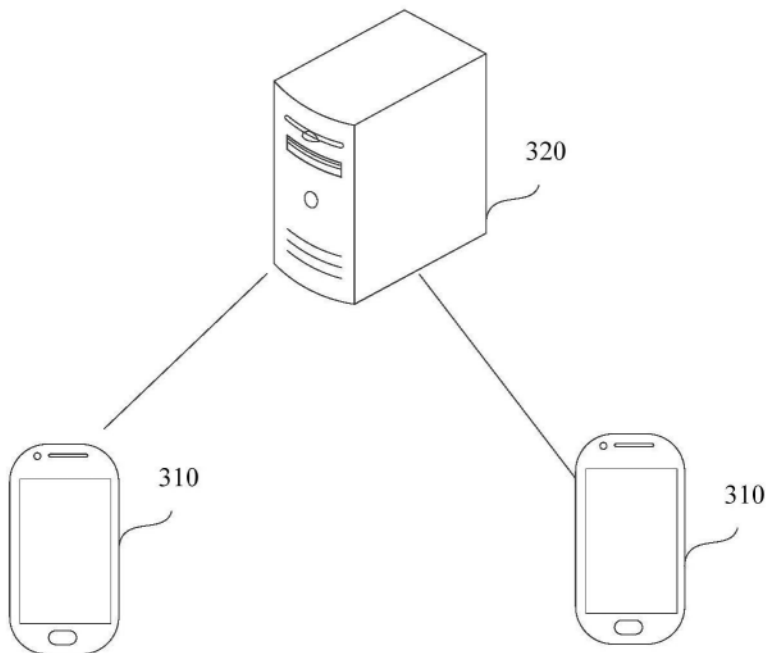


图3

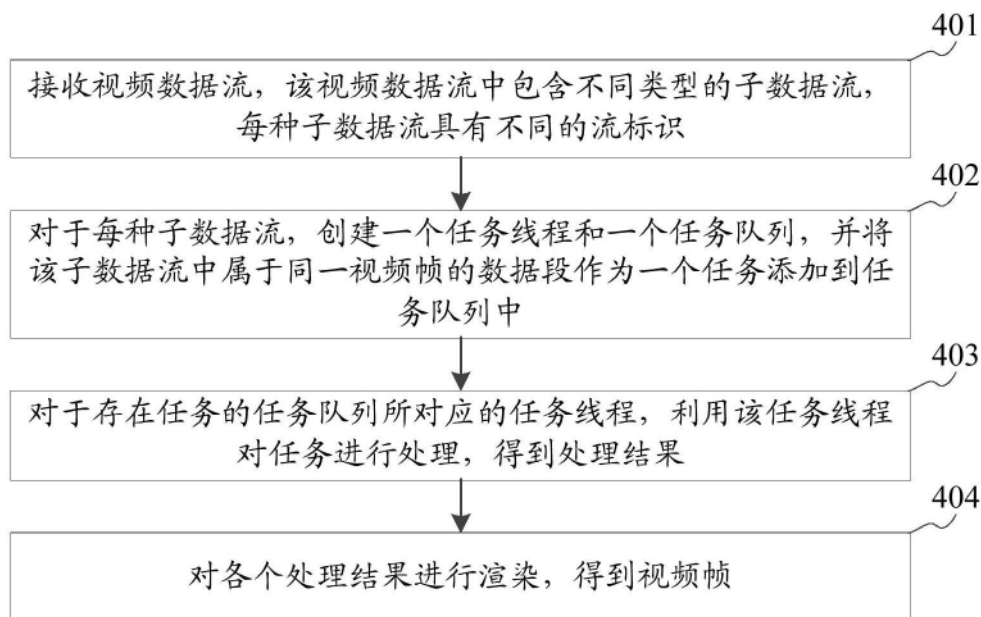


图4

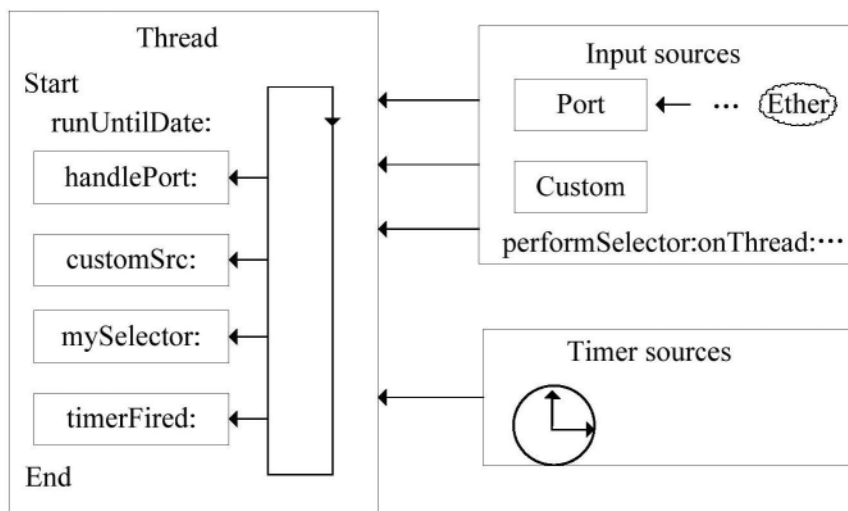


图5

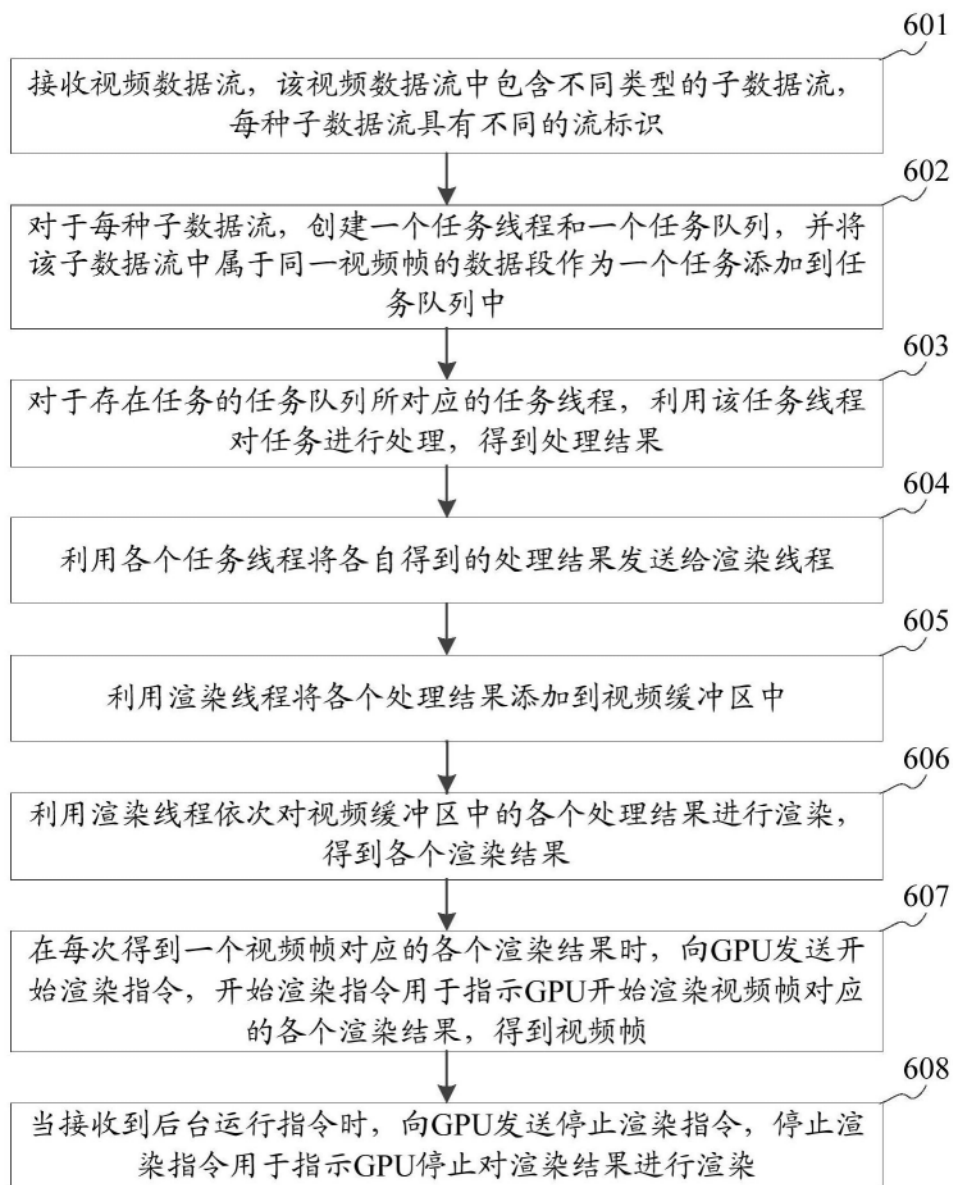


图6

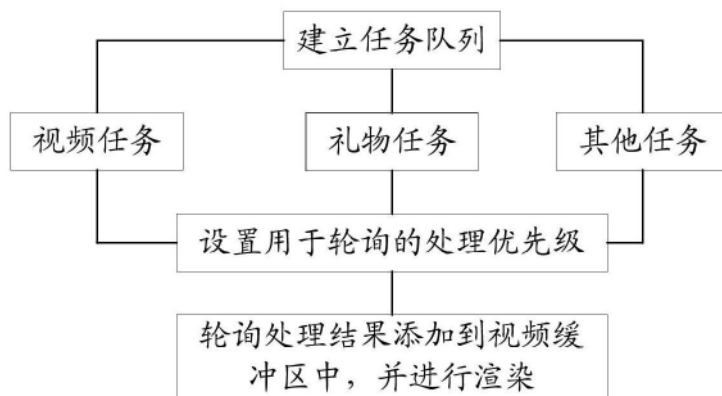


图7

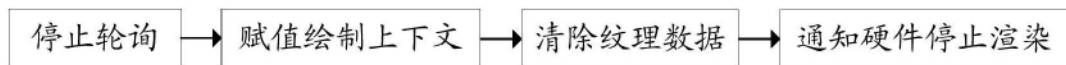


图8

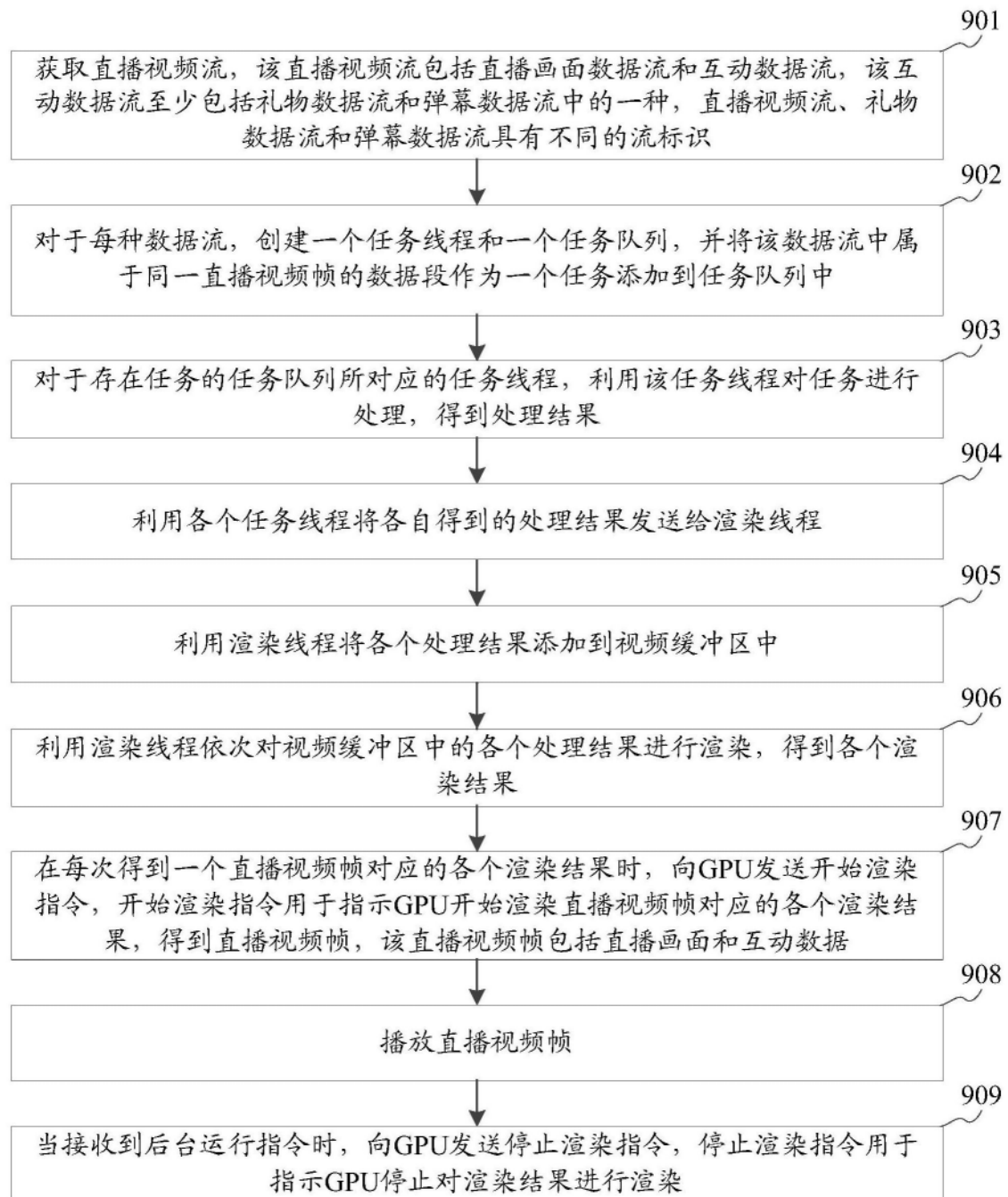


图9

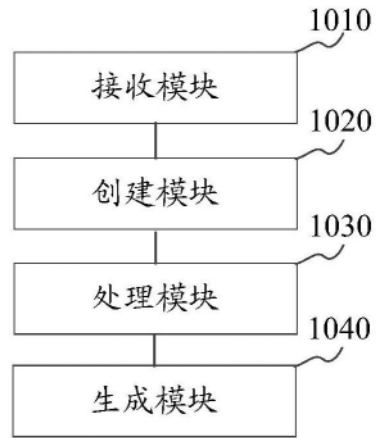


图10

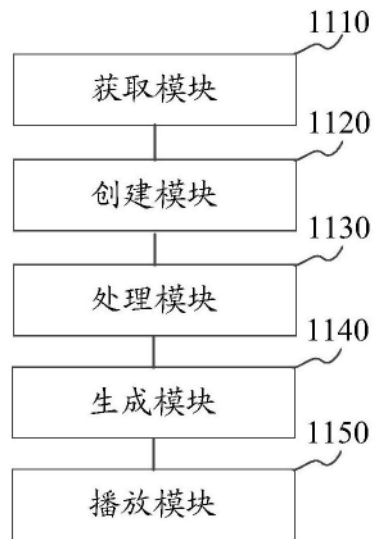


图11

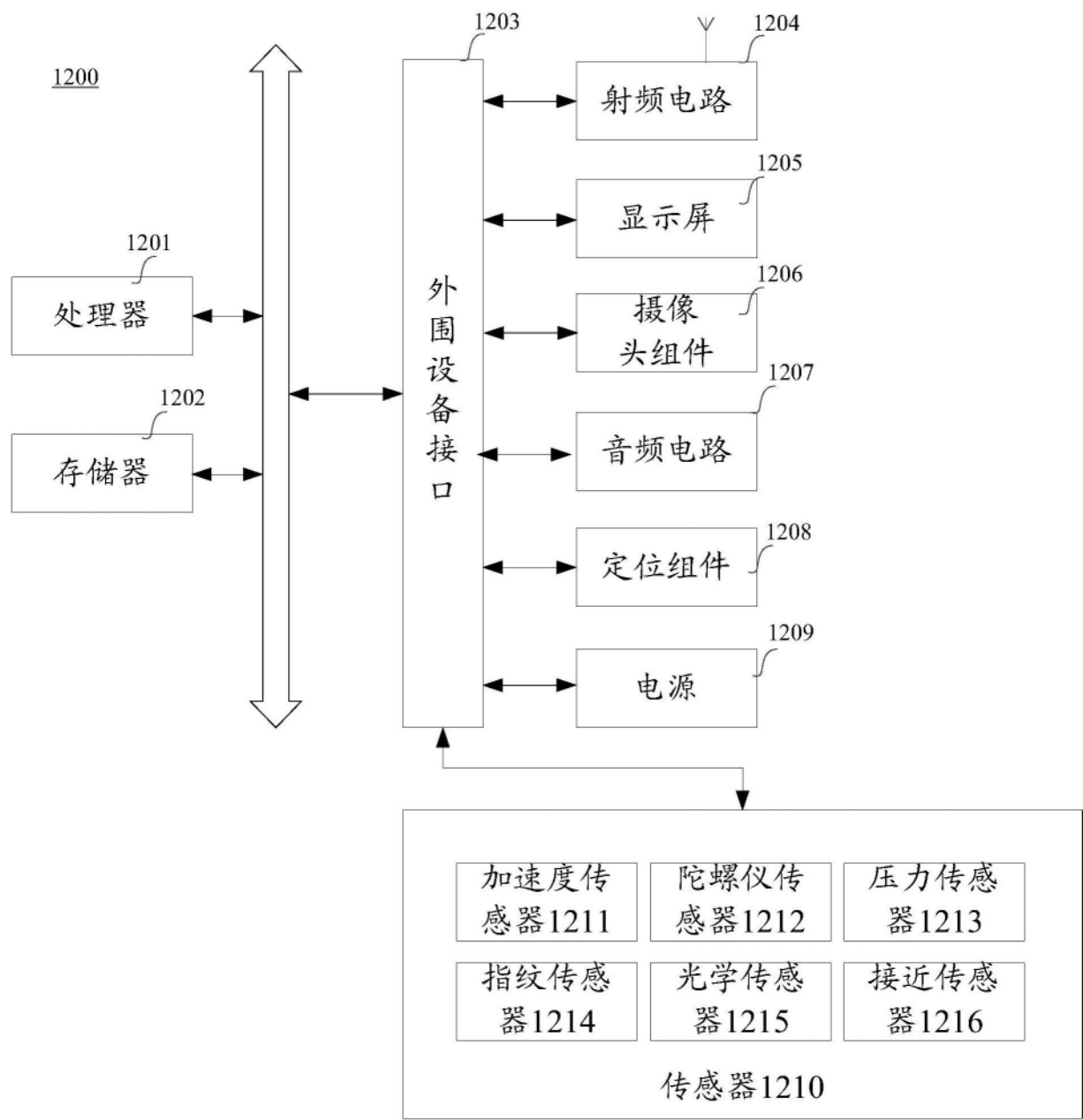


图12