



(12)发明专利申请

(10)申请公布号 CN 108234924 A

(43)申请公布日 2018.06.29

(21)申请号 201810108329.8

(22)申请日 2018.02.02

(71)申请人 北京百度网讯科技有限公司
地址 100085 北京市海淀区上地十街10号
百度大厦2层

(72)发明人 杨安宁

(74)专利代理机构 北京市铸成律师事务所
11313
代理人 张臻贤 王珺

(51) Int. Cl.
H04N 7/15(2006.01)

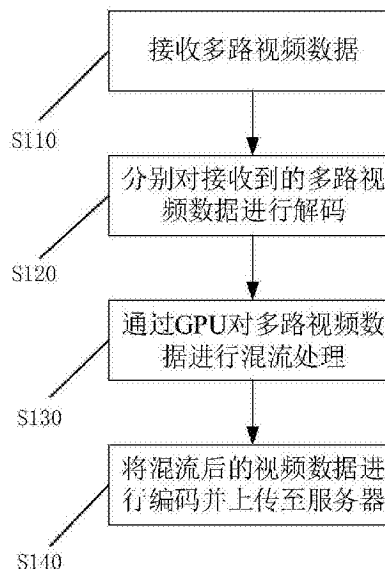
权利要求书2页 说明书7页 附图7页

(54)发明名称

视频混流方法、装置、设备及计算机可读介质

(57)摘要

本发明提出一种视频混流方法,包括:接收多路视频数据;分别对接收到的多路视频数据进行解码;通过GPU对多路视频数据进行混流处理;将混流后的视频数据进行编码并上传至服务器。本发明实施例通过客户端对视频数据进行混流处理,从而可以缓解服务器的压力。另外,在CPU和GPU交换数据时无需拷贝,这个过程无需CPU干预,占用资源少,同时充分利用了GPU的并行计算能力,不给进行混流的客户端造成额外负担,不会造成视频卡断。



1. 一种视频混流方法,其特征在于,包括:
接收多路视频数据;
分别对接收到的多路视频数据进行解码;
通过图形处理器对所述多路视频数据进行混流处理;以及
将混流后的视频数据进行编码并上传至服务器。
2. 根据权利要求1所述视频混流方法,其特征在于,所述接收多路视频数据,包括:
接收本地摄像头的视频数据和服务器下传的视频数据。
3. 根据权利要求2所述视频混流方法,其特征在于,在通过GPU对所述多路视频数据进行混流处理之前,所述视频混流方法还包括:
在图形处理器中为每路视频数据分别创建Texture数据结构;
为所述Texture数据结构创建对应的SurfaceTexture数据结构;
在CPU中创建SurfaceTexture对应的Surface数据结构作为解码缓冲区;
通过所述解码缓冲区接收解码后的视频数据;以及
通过SurfaceTexture数据结构将视频数据同步到Texture数据结构中。
4. 根据权利要求3所述视频混流方法,其特征在于,所述通过图形处理器对多路视频数据进行混流处理包括:
将本地视频数据解码后的纹理数据作为背景,置于最下层;
将服务器接收的视频数据解码后的纹理数据进行覆盖;以及
将本地视频数据和服务器接收的视频数据逐个像素进行缩放和坐标运算,再进行颜色融合,完成混流。
5. 根据权利要求1所述视频混流方法,其特征在于,所述将混流后的视频数据进行编码并上传至服务器的步骤中,包括:
将编码后的数据进行封装,再上传至服务器。
6. 一种视频混流装置,其特征在于,包括:
接收模块,用于接收多路视频数据;
解码模块,用于分别对接收到的多路视频数据进行解码;
混流模块,用于通过图形处理器对多路视频数据进行混流处理;以及
编码模块,用于将混流后的视频数据进行编码并上传至服务器。
7. 根据权利要求6所述视频混流装置,其特征在于,所述接收模块用于接收本地摄像头的视频数据和服务器下传的视频数据。
8. 根据权利要求7所述视频混流装置,其特征在于,还包括存储同步模块,用于对解码后的视频数据进行同步存储至所述图形处理器中;所述存储同步模块包括:
第一创建子模块,用于在图形处理器中为每路视频数据分别创建Texture数据结构;
第二创建子模块,用于为所述Texture数据结构创建对应的SurfaceTexture数据结构;
第三创建子模块,用于在CPU中创建SurfaceTexture对应的Surface数据结构作为解码缓冲区;
接收子模块,用于通过所述解码缓冲区接收解码后的视频数据;以及
同步子模块,用于通过SurfaceTexture数据结构将视频数据同步到Texture数据结构中。

9. 根据权利要求8所述视频混流装置,其特征在于,所述混流模块包括:
置底子模块,用于将本地视频数据解码后的纹理数据作为背景,置于最下层;
覆盖子模块,用于将服务器接收的视频数据解码后的纹理数据进行覆盖;以及
融合子模块,用于将本地视频数据和服务器接收的视频数据逐个像素进行缩放和坐标运算,再进行颜色融合,完成混流。

10. 根据权利要求6所述视频混流装置,其特征在于,所述编码模块用于将编码后的数据进行封装,再上传至服务器。

11. 一种视频混流设备,其特征在于,所述设备包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序;

当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器实现如权利要求1-5中任一所述的视频混流方法。

12. 一种计算机可读介质,其存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1-5中任一所述的视频混流方法。

视频混流方法、装置、设备及计算机可读介质

技术领域

[0001] 本发明涉及视频混流技术领域,尤其涉及一种在终端进行视频混流方法及装置、设备和计算机可读介质。

背景技术

[0002] 当进行多人视频会议或多人互动直播时,为提高互动程度,每位参会者或每位主播所在终端均需要显示其他参会者或主播的图像,这就需要进行视频混流处理。

[0003] 而在传统技术中,视频混流通常在服务器端(MCU)进行。如图1所示,A、B、C三位主播参与的直播时(A为大主播),服务器端首先需要接收来自A、B、C三位主播的视频流,再将三路视频流解码,将图像提取出来并进行混流处理,再做编码处理,最后分发给三位主播。

[0004] 目前常用于视频混流的是MCU服务器,但该服务器价格常在几十到几百万不等,而且单一服务器最多只支持10个左右的会议/直播同时进行,会议/直播数量过多容易造成卡顿。

发明内容

[0005] 本发明实施例提供一种视频混流方法、装置、设备及计算机可读介质,以解决或缓解现有技术中的以上技术问题。

[0006] 第一方面,本发明实施例提供了一种视频混流方法,包括:

[0007] 接收多路视频数据;

[0008] 分别对接收到的多路视频数据进行解码;

[0009] 通过图形处理器对所述多路视频数据进行混流处理;

[0010] 将混流后的视频数据进行编码并上传至服务器。

[0011] 结合第一方面,本发明在第一方面的第一种实现方式中,所述接收多路视频数据的步骤中,包括:接收本地摄像头的视频数据和服务器下传的视频数据。

[0012] 结合第一方面的第一种实现方式,本发明在第一方面的第二种实现方式中,

[0013] 在通过GPU对所述多路视频数据进行混流处理之前,所述视频混流方法还包括:

[0014] 在图形处理器中为每路视频数据分别创建Texture数据结构;

[0015] 为所述Texture数据结构创建对应的SurfaceTexture数据结构;

[0016] 在CPU中创建SurfaceTexture对应的Surface数据结构作为解码缓冲区;

[0017] 通过所述解码缓冲区接收解码后的视频数据;以及

[0018] 通过SurfaceTexture数据结构将视频数据同步到Texture数据结构中。

[0019] 结合第一方面的第二种实现方式,本发明在第一方面的第三种实现方式中,所述通过图形处理器对多路视频数据进行混流处理包括:

[0020] 将本地视频数据解码后的纹理数据作为背景,置于最下层;

[0021] 将服务器接收的视频数据解码后的纹理数据进行覆盖;

[0022] 将本地视频数据和服务器接收的视频数据逐个像素进行缩放和坐标运算,再进行

颜色融合,完成混流。

[0023] 结合第一方面,本发明在第一方面的第四种实现方式中,所述将混流后的视频数据进行编码并上传至服务器的步骤中,包括:将编码后的数据进行封装,再上传至服务器。

[0024] 第二方面,本发明实施例提供了一种视频混流装置,包括:

[0025] 接收模块,用于接收多路视频数据;

[0026] 解码模块,用于分别对接收到的多路视频数据进行解码;

[0027] 混流模块,用于通过图形处理器对多路视频数据进行混流处理;

[0028] 编码模块,用于将混流后的视频数据进行编码并上传至服务器。

[0029] 结合第二方面,本发明在第二方面第一种实现方式中,所述接收模块用于接收本地摄像头的视频数据和服务器下传的视频数据。

[0030] 结合第二方面的第一种实现方式,本发明在第二方面第二种实现方式中,

[0031] 还包括存储同步模块,所述存储同步模块包括:

[0032] 第一创建子模块,用于在图形处理器中为每路视频数据分别创建Texture数据结构;

[0033] 第二创建子模块,用于为所述Texture数据结构创建对应的SurfaceTexture数据结构;

[0034] 第三创建子模块,用于在CPU中创建SurfaceTexture对应的Surface数据结构作为解码缓冲区;

[0035] 接收子模块,用于通过所述解码缓冲区接收解码后的视频数据;以及

[0036] 同步子模块,用于通过SurfaceTexture数据结构将视频数据同步到Texture数据结构中。

[0037] 结合第二方面的第二种实现方式,本发明在第二方面第三种实现方式中,所述混流模块包括:

[0038] 置底子模块,用于将本地视频数据解码后的纹理数据作为背景,置于最下层;

[0039] 覆盖子模块,用于将服务器接收的视频数据解码后的纹理数据进行覆盖;

[0040] 融合子模块,用于将本地视频数据和服务器接收的视频数据逐个像素进行缩放和坐标运算,再进行颜色融合,完成混流。

[0041] 结合第二方面,本发明在第二方面的第四种实现方式中,所述编码模块用于将编码后的数据进行封装,再上传至服务器。

[0042] 所述装置的功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的模块。

[0043] 在一个可能的设计中,视频混流装置的结构中包括处理器和存储器,所述存储器用于存储支持视频混流装置执行上述第一方面中视频混流方法的程序,所述处理器被配置为用于执行所述存储器中存储的程序。所述视频混流装置还可以包括通信接口,用于视频混流装置与其他设备或通信网络通信。

[0044] 第三方面,本发明实施例提供了一种计算机可读介质,用于存储视频混流装置所用的计算机软件指令,其包括用于执行上述第一方面的视频混流方法所涉及的程序。

[0045] 上述技术方案中的一个技术方案具有如下优点或有益效果:本发明实施例通过客户端对视频数据进行混流处理,从而可以缓解服务器的压力。另外,在CPU和图形处理器

(GPU) 交换数据时无需拷贝, 这个过程无需CPU干预, 占用资源少, 同时充分利用了GPU的并行计算能力, 不给进行混流的客户端造成额外负担, 不会造成视频卡顿。

[0046] 上述概述仅仅是为了说明书的目的, 并不意图以任何方式进行限制。除上述描述的示意性的方面、实施方式和特征之外, 通过参考附图和以下的详细描述, 本发明进一步的方面、实施方式和特征将会是容易明白的。

附图说明

[0047] 在附图中, 除非另外规定, 否则贯穿多个附图相同的附图标记表示相同或相似的部件或元素。这些附图不一定是按照比例绘制的。应该理解, 这些附图仅描绘了根据本发明公开的一些实施方式, 而不应将其视为是对本发明范围的限制。

[0048] 图1为现有的视频混流方法示意图。

[0049] 图2为实施例一的视频混流方法的步骤流程图;

[0050] 图3为实施例一的视频混流方法的示意图。

[0051] 图4为实施例一的步骤S120的具体步骤流程图;

[0052] 图5为实施例一的步骤S130的具体步骤流程图;

[0053] 图6为实施例二的视频混流方法的步骤流程图;

[0054] 图7为实施例三的视频混流装置的连接框图;

[0055] 图8为实施例四的视频混流装置的连接框图;

[0056] 图9为实施例五的视频混流设备连接框图。

具体实施方式

[0057] 在下文中, 仅简单地描述了某些示例性实施例。正如本领域技术人员可认识到的那样, 在不脱离本发明的精神或范围的情况下, 可通过各种不同方式修改所描述的实施例。因此, 附图和描述被认为本质上是示例性的而非限制性的。

[0058] 本发明实施例旨在解决现有技术通过服务器进行视频混流, 容易使服务器负载过重而导致视频卡顿的技术问题, 本发明实施例主要提供了一种通过在客户端进行视频混流方法及装置, 下面分别通过以下实施例进行技术方案的展开描述。

[0059] 实施例一

[0060] 请参阅图2和图3, 其分别为本发明实施例一的视频混流方法的步骤流程图和示意图。本实施例一提供了一种视频混流方法, 在本实施例一中, 将所述视频混流方法应用于移动端直播中, 所述视频混流方法包括以下步骤:

[0061] S110: 接收多路视频数据。

[0062] 在本步骤S110中, 所接收的视频数据包括本地摄像头的视频数据和服务器下传的视频数据。例如, 若当前有A、B、C三位主播, 在A主播的移动终端中接收该移动终端接收的本地视频数据, 以及从服务器中接收另外两位主播B和C的视频数据。

[0063] S120: 分别对接收到的多路视频数据进行解码。

[0064] 如图4所示, 在进行解码过程中, 还对解码后的视频数据进行同步存储至GPU中, 具体步骤流程如下:

[0065] S121: 在GPU中为每路视频数据分别创建Texture数据结构。

- [0066] S122:为所述Texture数据结构创建对应的SurfaceTexture数据结构。
- [0067] S123:在CPU中创建SurfaceTexture对应的Surface数据结构作为解码缓冲区。
- [0068] S124:通过所述解码缓冲区接收解码后的视频数据。
- [0069] S125:通过SurfaceTexture数据结构将解码后的视频数据同步到Texture数据结构中。
- [0070] 在上述解码过程中,在GPU环境中利用OpenGL的glGenTextures接口调用创建纹理数据结构,并利用系统接口为该Texture创建对应的SurfaceTexture数据结构。然后创建SurfaceTexture对应的Surface并作为解码器(MediaCodec)或者播放器(MediaPlayer)解码后的输出缓冲区。在每一帧数据完成解码后,利用SurfaceTexture数据结构的updateTexImage给系统发信号将解码后的数据同步到GPU的Texture中。
- [0071] S130:通过GPU对多路视频数据进行混流处理。
- [0072] 如图5所示,所述步骤S130包括:
- [0073] S131:将本地视频数据解码后的纹理数据作为背景,置于最下层。
- [0074] S132:将服务器接收的视频数据解码后的纹理数据进行覆盖。
- [0075] S133:将本地视频数据和服务器接收的视频数据逐个像素进行缩放和坐标运算,再进行颜色融合,完成混流。
- [0076] 在GPU上进行数据混流时,同样利用SurfaceTexture接收视频的缓冲数据并同步到Texture,然后使用Shader Language这种着色器脚本语言将所有纹理进行融合(blending)操作。其中,以本地预览纹理为背景,置于最下层,上面覆盖其他纹理,逐像素进行缩放和坐标运算,然后再进行颜色融合,完成混流。这里所述本地纹理为移动终端的摄像头接收的视频数据,所述其他纹理为其他主播发送的视频数据。
- [0077] S140:将混流后的视频数据进行编码并上传至服务器。
- [0078] 所述步骤S140中包括将编码后的数据进行封装,再上传至服务器。在进行数据封装时,可以利用MediaCodec对编码流程进行硬件加速,从MediaCodec中获取用来接收输入数据的Surface数据结构,并利用OpenGL的eglCreateWindowSurface方法将该Surface封装成OpenGL的EGLSurface,并设置成OpenGL的渲染目标(Render Target)。
- [0079] 实施例二
- [0080] 与实施例一的区别在于:本实施例二将视频混流方法应用在多人视频会议上,具体的方案如下:
- [0081] 如图6所示,本实施例二提供了一种视频混流方法,包括以下步骤:
- [0082] S210:接收多路会议视频数据。
- [0083] 在本步骤S210中,所接收的多路会议视频数据包括本地摄像头的会议视频数据和服务器下传的会议视频数据。比如,若当前有A、B、C三个会议室,以A会议室为例,在A会议室中接收本地视频数据,然后再从服务器中接收另外两个会议室的视频数据。
- [0084] S220:分别对接收到的多路会议视频数据进行解码。
- [0085] S230:对解码后的视频数据进行同步存储至GPU中。
- [0086] S240:通过GPU对多路会议视频数据进行混流处理。
- [0087] 在进行混流处理中,若A会议室为主讲会议室,则以A会议室的会议视频解码后的纹理数据作为背景,置于最下层。将服务器接收的B、C会议室的视频数据解码后的纹理数据

进行覆盖。最后,再将所有视频数据逐个像素进行缩放和坐标运算,再进行颜色融合,完成混流。

[0088] S250:将混流后的会议视频数据进行编码并上传至服务器。

[0089] 本实施例二中的步骤S210-S250的实现原理与实施例一相同,故不再赘述。

[0090] 实施例三

[0091] 本实施例三对应于实施例一,提供了一种视频混流装置。请参阅图7,其为本实施例三的视频混流装置的连接框图。

[0092] 本实施例三提供了一种视频混流装置,包括:

[0093] 接收模块110,用于接收多路视频数据。所述接收模块110用于接收本地摄像头的视频数据和服务器下传的视频数据。

[0094] 解码模块120,用于分别对接收到的多路视频数据进行解码。

[0095] 存储同步模块130,用于对解码后的视频数据进行同步存储至所述图形处理器中;

[0096] 所述存储同步模块130包括:

[0097] 第一创建子模块131,用于在GPU中为每路视频数据分别创建Texture数据结构。

[0098] 第二创建子模块132,用于为所述Texture数据结构创建对应的SurfaceTexture数据结构。

[0099] 第三创建子模块133,用于在CPU中创建SurfaceTexture对应的Surface数据结构作为解码缓冲区,接收解码后的视频数据。

[0100] 同步子模块134,用于当所述视频数据完成解码后,通过SurfaceTexture数据结构将视频数据同步到Texture数据结构中。

[0101] 混流模块140,用于通过GPU对多路视频数据进行混流处理。

[0102] 所述混流模块140包括:

[0103] 置底子模块141,用于将本地视频数据解码后的纹理数据作为背景,置于最下层。

[0104] 覆盖子模块142,用于将服务器接收的视频数据解码后的纹理数据进行覆盖。

[0105] 融合子模块143,用于将本地视频数据和服务器接收的视频数据逐个像素进行缩放和坐标运算,再进行颜色融合,完成混流。

[0106] 编码模块150,用于将混流后的视频数据进行编码并上传至服务器。所述编码模块150用于将编码后的数据进行封装,再上传至服务器。

[0107] 本实施例三与实施例一的原理相同,故不再赘述。

[0108] 实施例四

[0109] 如图8所示,为本实施例四的视频混流装置的连接框图。本实施例四与实施例二对应,提供了一种视频混流装置,具体如下:

[0110] 本实施例四提供了一种视频混流装置,包括:

[0111] 会议视频接收模块210,用于接收多路会议视频数据。

[0112] 会议视频解码模块220,用于分别对接收到的多路会议视频数据进行解码。

[0113] 会议视频存储同步模块230,用于对解码后的视频数据进行同步存储至所述GPU中。

[0114] 会议视频混流模块240,用于通过GPU对多路会议视频数据进行混流处理。

[0115] 会议视频编码模块250,用于将混流后的会议视频数据进行编码并上传至服务器。

[0116] 本实施例四的应用方式与原理与实施例二相同,故不再赘述。

[0117] 实施例五

[0118] 本发明实施例五提供一种视频混流设备,如图9所示,该设备包括:存储器310和处理器320,存储器310内存储有可在处理器320上运行的计算机程序。所述处理器320执行所述计算机程序时实现上述实施例中的视频混流方法。所述存储器310和处理器320的数量可以是一个或多个。

[0119] 该设备还包括:

[0120] 通信接口330,用于与外界设备进行通信,进行数据交互传输。

[0121] 存储器310可能包含高速RAM存储器,也可能还包括非易失性存储器(non-volatile memory),例如至少一个磁盘存储器。

[0122] 如果存储器310、处理器320和通信接口330独立实现,则存储器310、处理器320和通信接口330可以通过总线相互连接并完成相互间的通信。所述总线可以是工业标准体系结构(ISA,Industry Standard Architecture)总线、外部设备互连(PCI,Peripheral Component)总线或扩展工业标准体系结构(EISA,Extended Industry Standard Component)总线等。所述总线可以分为地址总线、数据总线、控制总线等。为便于表示,图8中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0123] 可选的,在具体实现上,如果存储器310、处理器320及通信接口330集成在一块芯片上,则存储器310、处理器320及通信接口330可以通过内部接口完成相互间的通信。

[0124] 在本说明书的描述中,参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。而且,描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外,在不相互矛盾的情况下,本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

[0125] 此外,术语“第一”、“第二”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或隐含地包括至少一个该特征。在本发明的描述中,“多个”的含义是两个或两个以上,除非另有明确具体的限定。

[0126] 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为,表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分,并且本发明的优选实施方式的范围包括另外的实现,其中可以不按所示出或讨论的顺序,包括根据所涉及的功能按基本同时的方式或按相反的顺序,来执行功能,这应被本发明的实施例所属技术领域的技术人员所理解。

[0127] 在流程图中表示或在此以其他方式描述的逻辑和/或步骤,例如,可以被认为用于实现逻辑功能的可执行指令的定序列列表,可以具体实现在任何计算机可读介质中,以供指令执行系统、装置或设备(如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统)使用,或结合这些指令执行系统、装置或设备而使用。就本说明书而言,“计算机可读介质”可以是任何可以包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装

置。

[0128] 本发明实施例所述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质的更具体的示例至少(非穷尽性列表)包括以下:具有一个或多个布线的电连接部(电子装置),便携式计算机盘盒(磁装置),随机存取存储器(RAM),只读存储器(ROM),可擦除可编程只读存储器(EPROM或闪速存储器),光纤装置,以及便携式只读存储器(CDROM)。另外,计算机可读存储介质甚至可以是可在其上打印所述程序的纸或其他合适的介质,因为可以例如通过对纸或其他介质进行光学扫描,接着进行编辑、解译或必要时以其他合适方式进行处理来以电子方式获得所述程序,然后将其存储在计算机存储器中。

[0129] 在本发明实施例中,计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于指令执行系统、输入法或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、射频(Radio Frequency, RF)等等,或者上述的任意合适的组合。

[0130] 应当理解,本发明的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中,多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。例如,如果用硬件来实现,和在另一实施方式中一样,可用本领域公知的下列技术中的任一项或他们的组合来实现:具有用于对数据信号实现逻辑功能的逻辑门电路的离散逻辑电路,具有合适的组合逻辑门电路的专用集成电路,可编程门阵列(PGA),现场可编程门阵列(FPGA)等。

[0131] 本技术领域的普通技术人员可以理解实现上述实施例方法携带的全部或部分步骤是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,该程序在执行时,包括方法实施例的步骤之一或其组合。

[0132] 此外,在本发明各个实施例中的各功能单元可以集成在一个处理模块中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用,也可以存储在一个计算机可读存储介质中。所述存储介质可以是只读存储器,磁盘或光盘等。

[0133] 综上所述,本发明实施例通过客户端对视频数据进行混流处理,从而可以缓解服务器的压力。另外,在CPU和GPU交换数据时无需拷贝,这个过程无需CPU干预,占用资源少,同时充分利用了GPU的并行计算能力,不给进行混流的客户端造成额外负担,不会造成视频卡断。

[0134] 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,可轻易想到其各种变化或替换,这些都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以所述权利要求的保护范围为准。

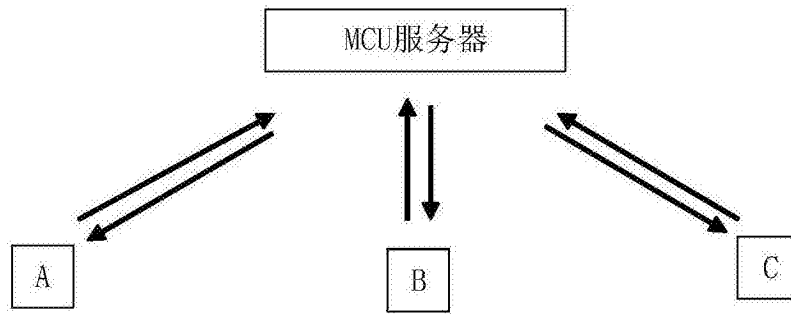


图1

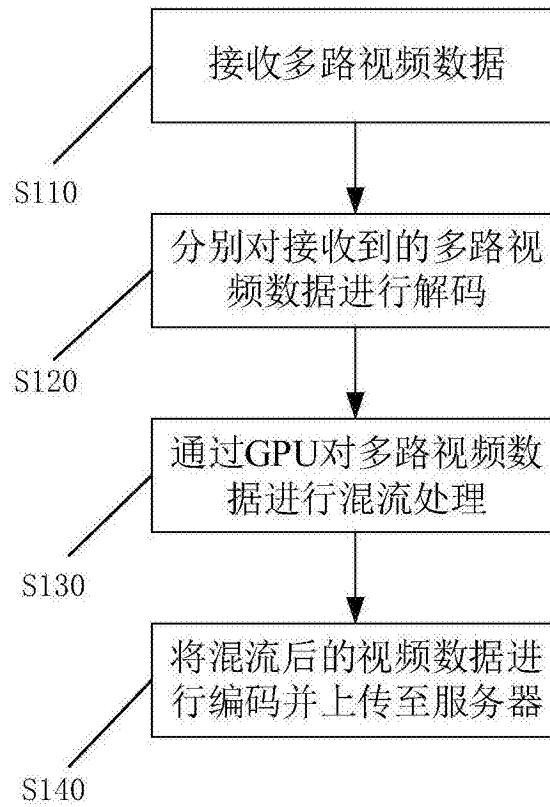


图2

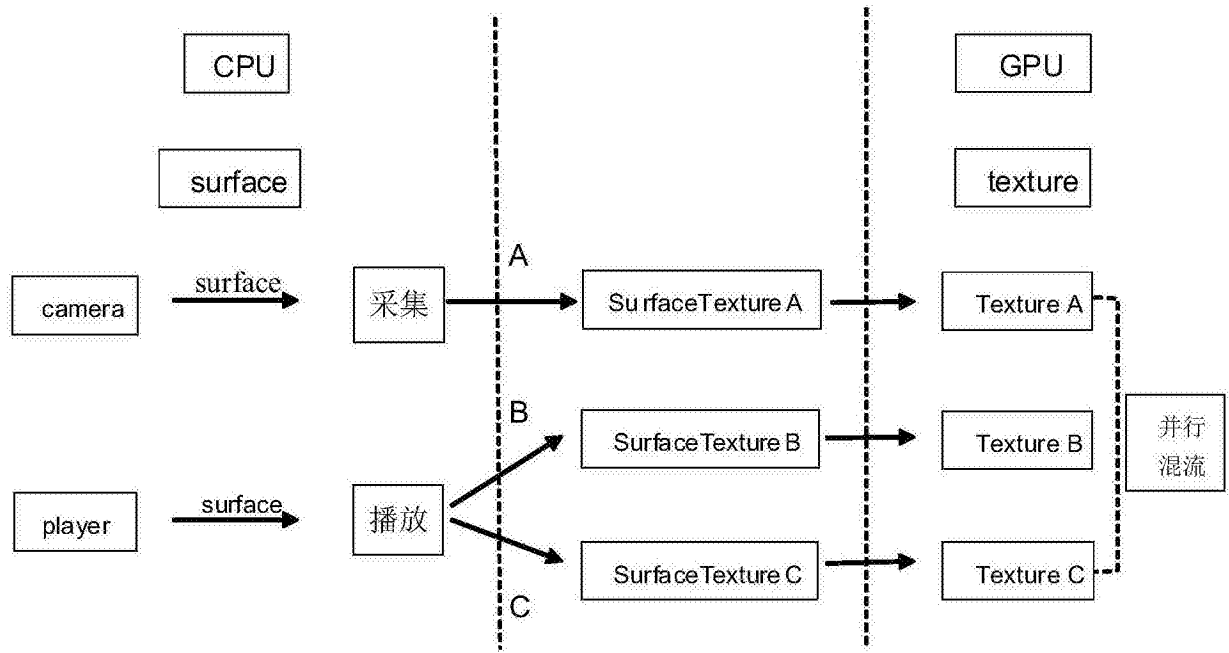


图3

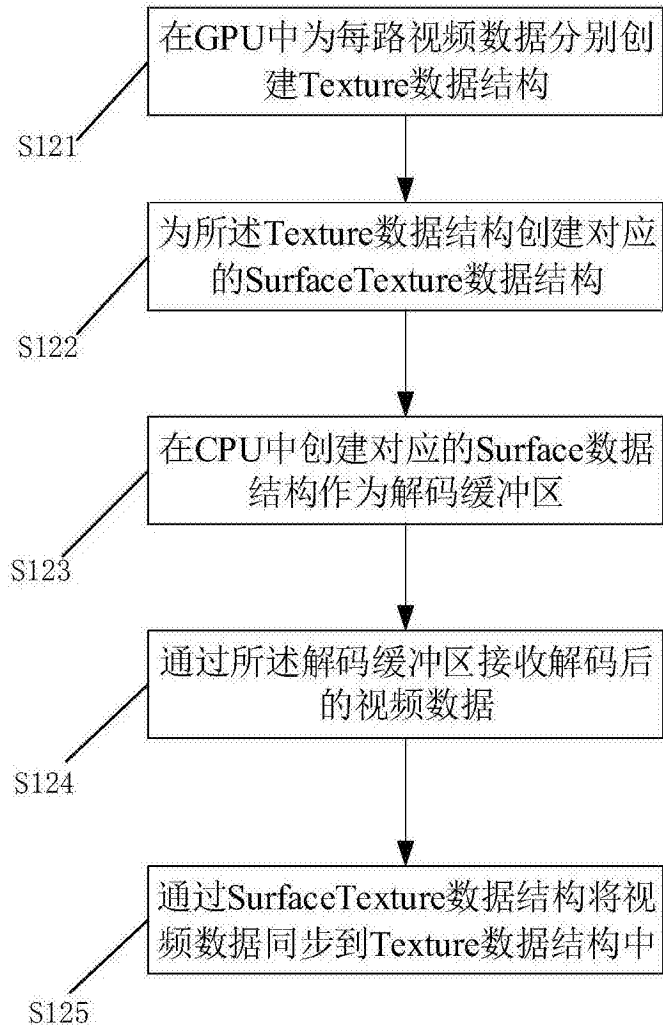


图4

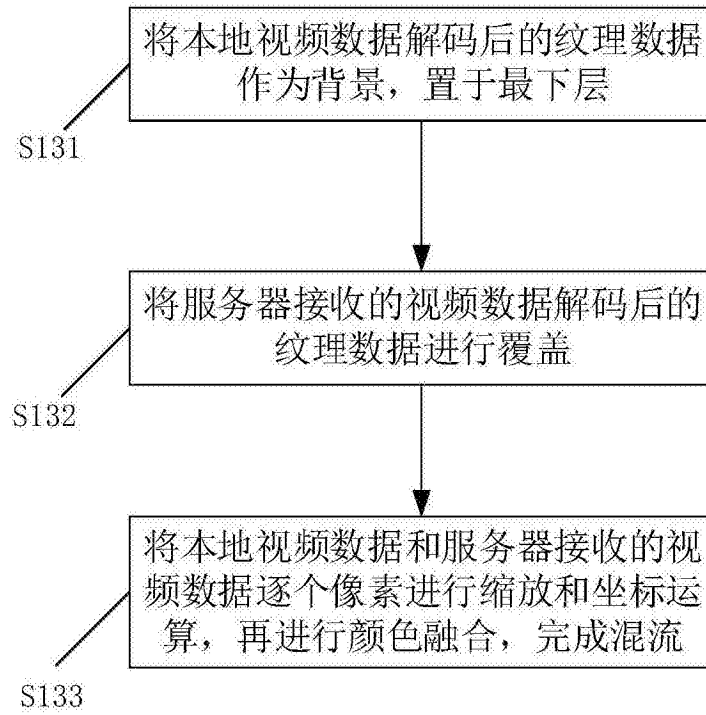


图5

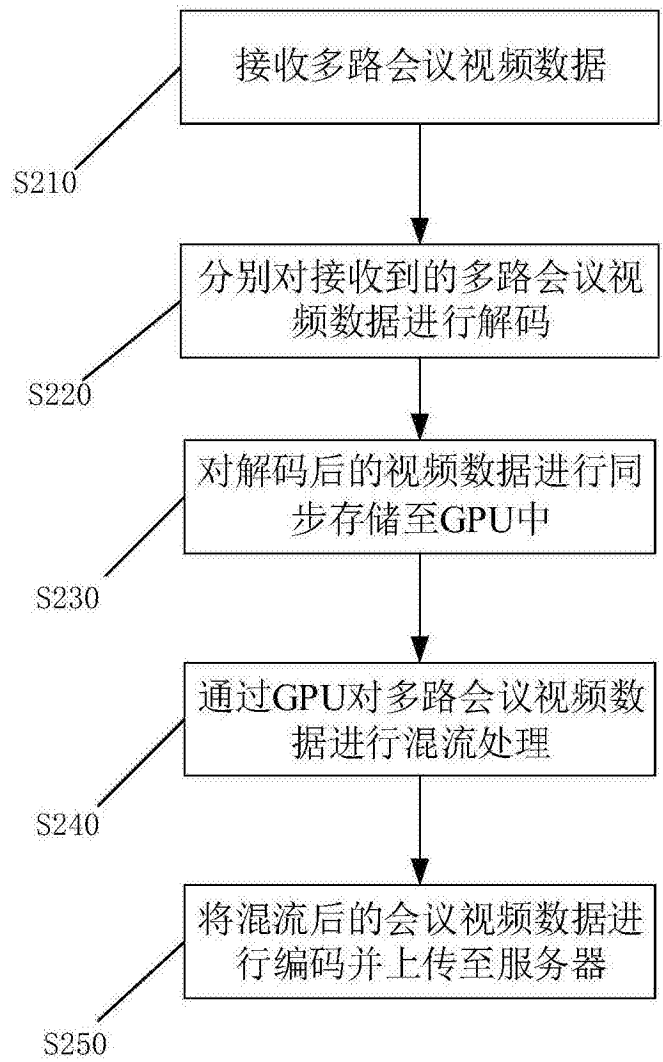


图6

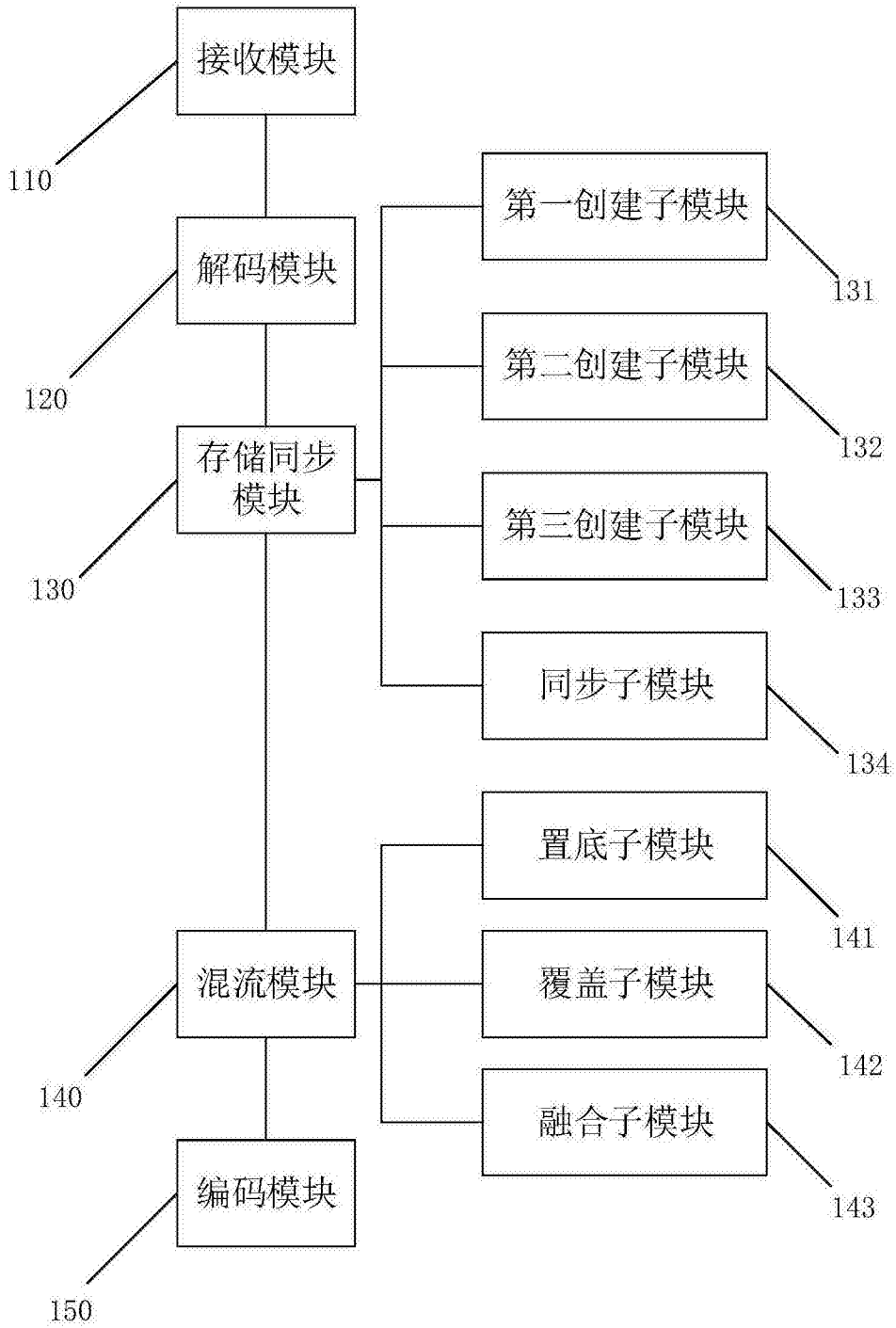


图7

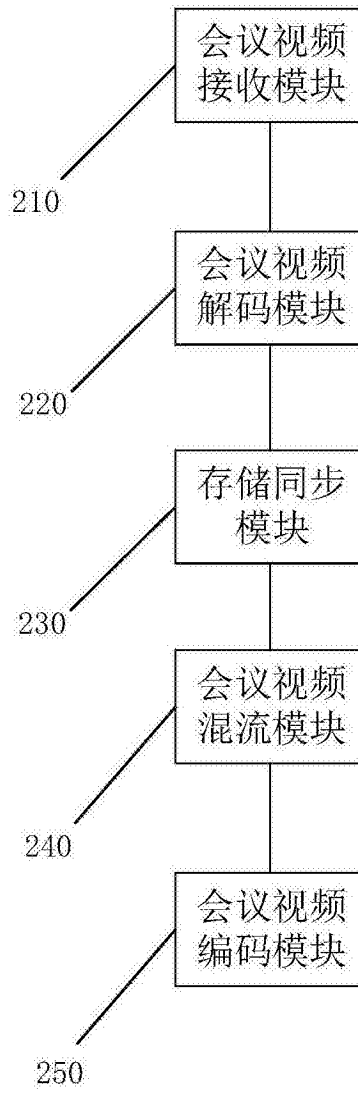


图8

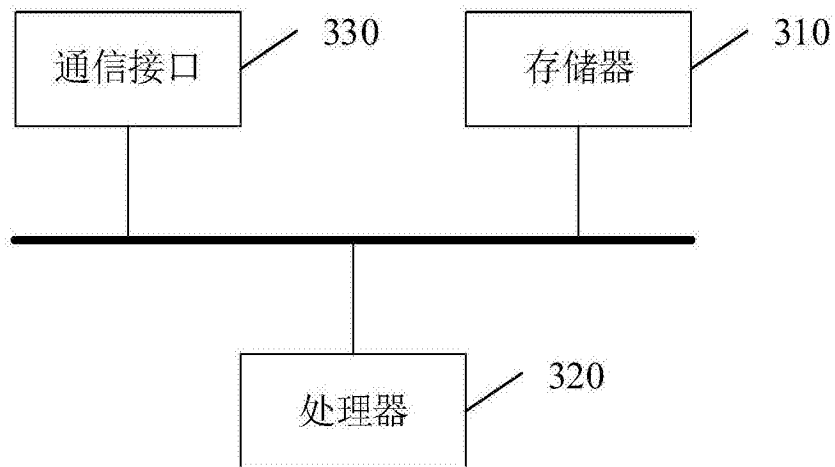


图9