



(12) 发明专利

(10) 授权公告号 CN 103283250 B

(45) 授权公告日 2015. 01. 21

(21) 申请号 201280002716. 4

(22) 申请日 2012. 12. 13

(85) PCT国际申请进入国家阶段日  
2013. 03. 19

(86) PCT国际申请的申请数据  
PCT/CN2012/086499 2012. 12. 13

(87) PCT国际申请的公布数据  
W02014/089793 ZH 2014. 06. 19

(73) 专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为  
总部办公楼

(72) 发明人 陈普 许天锡

(51) Int. Cl.  
H04N 21/43 (2011. 01)  
H04N 21/4402 (2011. 01)  
H04N 21/443 (2011. 01)  
H04N 21/434 (2011. 01)

(56) 对比文件

CN 101390397 A, 2009. 03. 18, 说明书第 33  
和 34 页 .

CN 1968395 A, 2007. 05. 23, 说明书第 6 页第  
8 行 - 第 7 页第 19 行 .

CN 102547394 A, 2012. 07. 04, 说明书第  
0017-0026 段, 附图 1 和 2 .

CN 1972402 A, 2007. 05. 30, 全文 .

审查员 吴爽

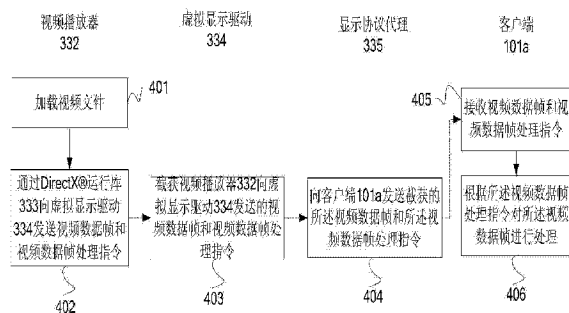
权利要求书3页 说明书17页 附图8页

(54) 发明名称

一种视频重定向的方法、装置、系统

(57) 摘要

本发明实施例公开了视频重定向方法、装置、系统及计算机可读存储介质,所述方法包括:截获视频播放器向虚拟显示驱动发送的视频数据帧和包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令的视频数据帧处理指令,发送视频数据帧和视频数据帧处理指令到客户端,使客户端根据视频数据帧处理指令对视频数据帧进行解码、后处理和显示,保证客户端用户的视频体验;同时只要服务器侧视频播放器支持视频重定向功能即可实现本发明;并且视频播放器向虚拟显示驱动发送的视频数据帧并没有在服务器侧根据视频数据帧处理指令进行处理,发送的仍是编码的视频数据帧,从而可以进一步节约服务器计算资源和网络传输带宽。



1. 一种视频重定向的方法,其特征在于,包括:

截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

2. 如权利要求1所述的方法,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

3. 如权利要求1所述的方法,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器通过 DirectX 运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令,所述 DirectX 运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

4. 如权利要求1所述的方法,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器通过 DirectX 运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述 DirectX 运行库截获所述视频数据帧和所述视频数据帧处理指令。

5. 如权利要求1所述的方法,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器调用 DirectX 运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

6. 如权利要求1至5任一所述的方法,其特征在于,还包括:所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,所述虚拟显示驱动接收所述视频播放器发送的视频硬件加速能力查询请求;所述虚拟显示驱动向所述视频播放器发送查询请求响应,所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

7. 如权利要求6所述的方法,其特征在于,所述虚拟显示驱动向所述视频播放器发送查询请求响应,具体包括:

所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;

所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;

如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

8. 一种服务器,其特征在于,包括中央处理器和存储器,所述中央处理器与所述存储器

通过系统总线连接,当所述服务器运行时,所述中央处理器执行所述存储器存储的计算机执行指令,使所述服务器执行如下步骤:

截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

9. 如权利要求 8 所述的服务器,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

10. 如权利要求 8 所述的服务器,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器通过 DirectX 运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令,所述 DirectX 运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

11. 如权利要求 8 所述的服务器,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器通过 DirectX 运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述 DirectX 运行库截获所述视频数据帧和所述视频数据帧处理指令。

12. 如权利要求 8 所述的服务器,其特征在于,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

当所述视频播放器调用 DirectX 运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

13. 如权利要求 8 至 12 任一所述的服务器,其特征在于,还用于执行:所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,所述虚拟显示驱动接收所述视频播放器发送的视频硬件加速能力查询请求,所述虚拟显示驱动向所述视频播放器发送查询请求响应;所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

14. 如权利要求 13 所述的服务器,其特征在于,所述虚拟显示驱动向所述视频播放器发送查询请求响应,具体包括:

所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;

所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;

如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支

持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

15. 一种视频重定向系统,包括如权利要求8至14任一所述的服务器和所述客户端,其中,所述客户端用于根据所述视频数据帧处理指令对所述视频数据帧进行处理。

16. 一种服务器,其特征在于,包括:

截获单元,用于截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

发送单元,用于向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

17. 如权利要求16所述的服务器,其特征在于,所述截获单元具体为所述虚拟显示驱动;所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

18. 如权利要求16所述的服务器,其特征在于,所述截获单元具体为动态链接库,当所述视频播放器通过DirectX运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令,所述DirectX运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时,所述动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

19. 如权利要求16所述的服务器,其特征在于,所述截获单元具体为DirectX运行库,当所述视频播放器通过所述DirectX运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令时,所述DirectX运行库截获所述视频数据帧和所述视频数据帧处理指令。

20. 如权利要求16所述的服务器,其特征在于,所述截获单元具体为动态链接库,当所述视频播放器调用DirectX运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

21. 如权利要求16至20任一所述的服务器,其特征在于,所述虚拟显示驱动还用于所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,接收所述视频播放器发送的视频硬件加速能力查询请求,所述虚拟显示驱动向所述视频播放器发送查询请求响应;所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

22. 如权利要求21所述的服务器,其特征在于,所述虚拟显示驱动向所述视频播放器发送查询请求响应,具体包括:

所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;

所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;

如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

23. 一种视频重定向系统,包括如权利要求16至22任一所述的服务器和所述客户端,其中,所述客户端用于根据所述视频数据帧处理指令对所述视频数据帧进行处理。

## 一种视频重定向的方法、装置、系统

### 技术领域

[0001] 本发明涉及信息技术领域,特别涉及一种视频重定向的方法、装置、系统及计算机可读介质。

### 背景技术

[0002] 随着计算机技术的不断发展,远程桌面成为一种流行技术。远程桌面技术通过登陆到远程服务器,将该远程服务器屏幕显示的内容传递到本地客户端进行显示。远程桌面应用并不仅指远程的屏幕显示,还可以包含远程视频传送。

[0003] 现有技术中,使用远程桌面中的视频播放器播放视频文件时,为了减轻远程服务器压力,使用基于Microsoft®公司推出的DirectShow®框架的多媒体播放器时,在远程服务器上通过该视频播放器加载视频文件后,可以从DirectShow®中的Filter 截获未解码的视频数据,在客户端的 DirectShow®框架中也插入 Filter,接收远程服务器发送的视频数据进行解码显示,同时将视频数据显示信息发送到客户端,但是该方案依赖特定播放器,并且不能对视频数据进行后处理操作,影响用户体验。

### 发明内容

[0004] 本发明实施例公开了视频重定向的方法、装置、系统和计算机可读存储介质,用于通过视频重定向,在客户端实现视频数据帧后处理,提高客户端用户视频体验。

[0005] 第一方面,本发明实施例公开了一种视频重定向的方法,包括:

[0006] 截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解 复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0007] 向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0008] 结合第一方面实施例,在第一种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0009] 当所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

[0010] 结合第一方面实施例,在第二种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0011] 当所述视频播放器通过DirectX®运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令,所述DirectX®运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0012] 结合第一方面实施例,在第三种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0013] 当所述视频播放器通过DirectX®运行库向所述虚拟显示驱动发送的所述视频

数据帧和所述视频数据帧处理指令时,所述DirectX®运行库截获所述视频数据帧和所述视频数据帧处理指令。

[0014] 结合第一方面,在第四种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0015] 当所述视频播放器调用DirectX®运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0016] 结合第一方面、第一方面的第一种、第二种、第三种或第四种可能的实施方式,在第五种可能的实施方式中,还包括:所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,所述虚拟显示驱动接收所述视频播放器发送的视频硬件加速能力查询请求,所述虚拟显示驱动向所述视频播放器发送查询请求响应;所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

[0017] 结合第一方面的第五种可能的实施方式,在第六种可能的实施方式中,所述虚拟显示驱动向所述视频播放器发送查询请求响应,具体包括:

[0018] 所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;

[0019] 所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;

[0020] 如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

[0021] 第二方面,本发明实施例公开了一种视频重定向的方法,包括:

[0022] 客户端接收视频数据帧和视频数据帧处理指令;其中,所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0023] 根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0024] 结合第二方面实施例,在第一种可能的实施方式中,根据所述视频数据帧处理指令对所述视频数据帧进行处理,具体包括:

[0025] 根据所述视频数据帧解码指令对所述视频数据帧进行解码;

[0026] 根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理;

[0027] 根据所述视频数据帧呈现指令对所述视频后处理的视频数据帧进行显示。

[0028] 结合第二方面的第一种可能的实施方式,在第二种可能的实施方式中,所述根据所述视频数据帧解码指令对所述视频数据帧进行解码;根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理,具体为,在所述客户端启动视频硬件加速条件下,根据所述视频数据帧解码指令对所述视频数据帧进行解码,根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理。

[0029] 第三方面,本发明实施例公开了一种服务器,包括:中央处理器和存储器,所述中央处理器与所述存储器通过系统总线连接,当所述服务器运行时,所述中央处理器执行所述存储器存储的计算机执行指令,使所述服务器执行如下步骤:

[0030] 截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令；其中，所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的；所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令；

[0031] 向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令，以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0032] 结合第三方面实施例，在第一种可能的实施方式中，所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令，具体包括：

[0033] 所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时，所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

[0034] 结合第三方面实施例，在第二种可能的实施方式中，所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令，具体包括：

[0035] 当所述视频播放器通过**DirectX®**运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令，所述**DirectX®**运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时，动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0036] 结合第三方面实施例，在第三种可能的实施方式中，所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令，具体包括：

[0037] 当所述视频播放器通过**DirectX®**运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令时，所述**DirectX®**运行库截获所述视频数据帧和所述视频数据帧处理指令。

[0038] 结合第三方面实施例，在第四种可能的实施方式中，所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令，具体包括：

[0039] 当所述视频播放器调用**DirectX®**运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时，动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0040] 结合本发明第三方面、第三方面的第一种、第二种、第三种或第四种可能的实施方式，在第五种可能的实施方式中，还用于执行：所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前，所述虚拟显示驱动接收所述视频播放器发送的视频硬件加速能力查询请求，所述虚拟显示驱动向所述视频播放器发送查询请求响应；所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

[0041] 结合本发明第三方面的第五种可能的实施方式，在第六种可能的实施方式中，所述虚拟显示驱动向所述视频播放器发送查询请求响应，具体包括：

[0042] 所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息；

[0043] 所述虚拟显示驱动根据所述视频播放器的信息，与支持视频重定向的视频播放器列表进行匹配；

[0044] 如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配，则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应；其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

[0045] 第四方面，本发明实施例公开了一种客户端，包括中央处理器和存储器，所述中央

处理器与所述存储器通过系统总线连接,当所述客户端运行时,所述中央处理器执行所述存储器存储的计算机执行指令,使所述客户端执行如下步骤:

[0046] 客户端接收视频数据帧和视频数据帧处理指令;其中,所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0047] 根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0048] 结合本发明第四方面实施例,在第一种可能的实施方式中,根据所述视频数据帧处理指令对所述视频数据帧进行处理,具体包括:

[0049] 根据所述视频数据帧解码指令对所述视频数据帧进行解码;

[0050] 根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理;

[0051] 根据所述视频数据帧呈现指令对所述视频后处理的视频数据帧进行显示。

[0052] 结合本发明第四方面的第一种可能的实施方式,在第二种可能的实施方式中,所述根据所述视频数据帧解码指令对所述视频数据帧进行解码;根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理,具体为,在所述客户端启动视频硬件加速条件下,根据所述视频数据帧解码指令对所述视频数据帧进行解码,根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理。

[0053] 本发明第五方面,本发明实施例公开了一种视频重定向系统,包括本发明第三方面任一所述的服务器和本发明第四方面任一所述的客户端。

[0054] 第六方面,本发明实施例公开了一种服务器,包括:

[0055] 截获单元,用于截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0056] 发送单元,用于向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0057] 结合本发明第六方面实施例,在第一种可能的实施方式中,所述截获单元具体为所述虚拟显示驱动;所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

[0058] 结合本发明第六方面实施例,在第二种可能的实施方式中,所述截获单元具体为动态链接库,当所述视频播放器通过DirectX®运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令,所述DirectX®运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时,所述动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0059] 结合本发明第六方面实施例,在第三种可能的实施方式中,所述截获单元具体为DirectX®运行库,当所述视频播放器通过所述DirectX®运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令时,所述DirectX®运行库截获所述视频数据帧和所述视频数据帧处理指令。

[0060] 结合本发明第六方面实施例,在第四种可能的实施方式中,所述截获单元具体为动态链接库,当所述视频播放器调用DirectX®运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述动态链接库截获所述视频数据帧和所述视频数



据帧处理指令。

[0061] 结合本发明第六方面、第六方面的第一种、第二种、第三种或第四种可能的实施方式,在第五种可能的实施方式中,所述虚拟显示驱动还用于所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,接收所述视频播放器发送的视频硬件加速能力查询请求,所述虚拟显示驱动向所述视频播放器发送查询请求响应;所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

[0062] 结合本发明第六方面的第五种可能的实施方式,在第六种可能的实施方式中,所述虚拟显示驱动向所述视频播放器发送查询请求响应,具体包括:

[0063] 所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;

[0064] 所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;

[0065] 如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

[0066] 第七方面,本发明实施例公开了一种客户端,包括:接收单元,用于接收视频数据帧和视频数据帧处理指令;其中,所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0067] 处理单元,用于根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0068] 结合本发明第七方面实施例,在第一种可能的实施方式中,所述处理单元,具体用于:

[0069] 根据所述视频数据帧解码指令对所述视频数据帧进行解码;

[0070] 根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理;

[0071] 根据所述视频数据帧呈现指令对所述视频后处理的视频数据帧进行显示。

[0072] 第八方面,本发明实施例公开了一种视频重定向系统,包括本发明第六方面所述的任一服务器和本发明第七方面任一所述的客户端。

[0073] 第九方面,本发明实施例公开了一种计算机可读存储介质,当计算机执行所述计算机可读存储介质时,所述计算机执行如下步骤:

[0074] 截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0075] 向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0076] 结合本发明第九方面实施例,在第一种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0077] 所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,所述虚拟显示驱动截获所述视频数据帧和所述视频数据帧处理指令。

[0078] 结合本发明第九方面实施例,在第二种可能的实施方式中,所述截获视频播放器

向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0079] 当所述视频播放器通过**DirectX®**运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令,所述**DirectX®**运行库调用所述虚拟显示驱动对所述视频数据帧进行处理时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0080] 结合本发明第九方面实施例,在第三种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0081] 当所述视频播放器通过**DirectX®**运行库向所述虚拟显示驱动发送的所述视频数据帧和所述视频数据帧处理指令时,所述**DirectX®**运行库截获所述视频数据帧和所述视频数据帧处理指令。

[0082] 结合本发明第九方面实施例,在第四种可能的实施方式中,所述截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令,具体包括:

[0083] 当所述视频播放器调用**DirectX®**运行库向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令时,动态链接库截获所述视频数据帧和所述视频数据帧处理指令。

[0084] 结合本发明第九方面、第九方面第一种、第二种、第三种或第四种可能的实施方式,在第五种可能的实施方式中,还包括:所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,所述虚拟显示驱动接收所述视频播放器发送的视频硬件加速能力查询请求,所述虚拟显示驱动向所述视频播放器发送查询请求响应;所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。

[0085] 结合本发明第九方面的第五种可能的实施方式,在第六种可能的实施方式中,所述虚拟显示驱动向所述视频播放器发送查询请求响应,具体包括:

[0086] 所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;

[0087] 所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;

[0088] 如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。

[0089] 第十方面,本发明实施例公开了一种计算机可读存储介质,包括,当计算机执行所述计算机可读存储介质时,所述计算机执行如下步骤:

[0090] 客户端接收视频数据帧和视频数据帧处理指令;其中,所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;

[0091] 根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0092] 结合本发明第十方面的实施例,在第一种可能的实施方式中,根据所述视频数据帧处理指令对所述视频数据帧进行处理,具体包括:

[0093] 根据所述视频数据帧解码指令对所述视频数据帧进行解码;

[0094] 根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理;

[0095] 根据所述视频数据帧呈现指令对所述视频后处理的视频数据帧进行显示。

[0096] 结合本发明第十方面的第一种可能的实施方式,在第二种可能的实施方式中,所述根据所述视频数据帧解码指令对所述视频数据帧进行解码;根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理,具体为,在所述客户端启动视频硬件加速条件下,根据所述视频数据帧解码指令对所述视频数据帧进行解码,根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理。

[0097] 本发明实施例公开的视频重定向方法、装置、系统及计算机可读存储介质,截获视频播放器向虚拟显示驱动发送的视频数据帧和包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令的视频数据帧处理指令,发送视频数据帧和视频数据帧处理指令到客户端,使客户端根据视频数据帧处理指令对视频数据帧进行解码、后处理和呈现,保证客户端用户的视频体验。同时只要服务器侧视频播放器支持视频重定向功能即可实现本发明,不依赖某一种视频播放器。

### 附图说明

[0098] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例或现有技术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发明的一些实施例,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。

- [0099] 图 1 为本发明实施例应用场景系统示意图;
- [0100] 图 2 为本发明实施例远程桌面服务器结构示意图;
- [0101] 图 3a、3b、3c、3d 为本发明实施例虚拟桌面结构示意图;
- [0102] 图 4 为本发明实施例视频重定向流程图;
- [0103] 图 5 为本发明实施例客户端结构示意图;
- [0104] 图 6 为本发明实施例视频数据帧后处理流程图;
- [0105] 图 7a 和图 7b 为本发明实施例视频数据帧后处理效果图;
- [0106] 图 8 为本发明实施例视频数据帧呈现效果图;
- [0107] 图 9 为本发明实施例服务器结构图;
- [0108] 图 10 为本发明实施例客户端结构示意图;
- [0109] 图 11 为本发明实施例系统结构示意图。

### 具体实施方式

[0110] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述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,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0111] 远程桌面环境中,客户端发起视频播放请求,服务器根据客户端的视频播放请求,通过视频播放器加载视频文件,将视频文件的视频流传输到客户端,这一过程称为视频重定向。本发明实施例中的服务器可以包括普通个人计算机或其他终端,也可以为商业服务器等,在这里并不具体限定,本发明实施例以向客户端提供基于 Windows® 操作系统平台的远程桌面服务为例对发明方案进行详细描述。但是本发明方案并不仅限于 Windows®

操作系统,根据本发明方案,同样可以扩展到其他常见的系统平台,如 **Linux®**操作系统等。

[0112] 本发明实施例公开了视频重定向的方法,应用于远程桌面场景。远程桌面场景通常由本地计算机和远程计算机组成。本地计算机通过访问远程计算机,将远程计算机桌面传送到本地计算机。本地计算机,即本发明实施例中的客户端;远程计算机,即本发明实施例中的服务器。本发明实施例中的视频重定向是指将服务器上的视频文件的视频数据流传送到客户端。其中一种远程桌面场景为虚拟桌面基础架构场景,但是本发明并不局限于图 1 所示架构,本发明方案可以应用于其他用于传送远程视频数据到客户端的场景。虚拟桌面基础架构场景如图 1 所示,本实施场景仅是一种示例性说明,只是给出了与本发明实施例相关的组成部分,但是图 1 并不是对本发明应用的具体限定。图 1 中左侧是虚拟桌面的客户端,通常称为瘦客户端,客户端的设备形式可以是普通计算机 101a、平板电脑 101b、智能手机 101c 等。它们通过网络 102 使用远程桌面协议 103 访问远程桌面服务。服务器 204a...204n 提供了远程桌面的载体,用户的虚拟桌面 (Virtual Desktop) 以虚拟机 205a、205b...205n 的形式存在于服务器上。虚拟桌面管理系统 106,用于提供用户的客户端与虚拟机的映射等功能。客户端首先连接到虚拟桌面管理系统 106,获取用户的虚拟机地址,进而连接到虚拟机,虚拟桌面管理系统 106 可以为服务器,也可以为普通个人计算机等,本发明对此不作具体限定。用户通过客户端访问服务器上分配给该用户的虚拟桌面 (即虚拟机),该虚拟桌面将用户访问的内容传输到用户的客户端进行显示。

[0113] 以服务器 204a 为例,硬件结构和软件结构是虚拟桌面基础架构中最重要的组件,它的典型结构如图 2 所示。服务器 204a 中的典型结构包括 3 个部分,分别为硬件平台 300、虚拟化平台 (也可以称为 Hypervisor)301 和虚拟机执行空间 302。硬件平台 300 中可以包括服务器物理结构中的硬盘 310、中央处理器 311、网络接口卡 312 和存储器 313。Hypervisor301 和虚拟机执行空间 302 分别提供了虚拟机的硬件平台和软件平台。

[0114] Hypervisor 包含了虚拟机监视器 320a、320b...320n。Hypervisor 是物理硬件和虚拟机之间的中间层,负责协调各个虚拟机对硬件平台的访问。虚拟机监视器包括实现虚拟机的虚拟硬件平台。图 2 中,虚拟机监视器 320a 中包含虚拟硬件平台 321a,虚拟硬件平台中包含虚拟出来的存储器 322,中央处理器 323,硬盘 324,显卡 325 和网络接口卡 326 等。虚拟机执行空间 302 中运行了多个虚拟机 205a、205b...205n。虚拟机 205a 的软件系统运行在虚拟机监视器 320a 的虚拟硬件平台 321a 上,其他虚拟机的软件系统也运行在对应的虚拟机监视器的虚拟硬件平台上。客户操作系统 330 运行在虚拟机 205a 中,客户操作系统 330 包含了一个设备驱动层 331,设备驱动层 331 安装了虚拟显示驱动 334、网络接口卡驱动 336 等驱动程序。虚拟机 205a 中包括视频播放器 332。其中,客户操作系统 330 可以为微软®公司提供的**Windows®**操作系统平台。

[0115] 如图 2 所示,用户使用客户端 101a 登陆到虚拟机 205a 时,用户观看视频时,比如观看优酷®网络视频,或者虚拟机 205a 本地视频文件时,通过视频播放器 332,加载视频文件。本发明实施例以观看本地视频为例,通过显示协议代理 335 接收到来自客户端 101a 的视频播放请求,视频播放器 332 根据该视频播放请求,加载视频文件。视频播放器 332 对加载的视频文件解复用获得视频数据帧。视频播放器 332 在初始化时需要确定虚拟机 205a 是否支持视频硬件加速功能,视频播放器 332 发送向虚拟显示驱动 334 发送视频硬件加速

能力查询请求。视频硬件加速是指利用计算机硬件代替视频播放器的软件算法对视频数据帧进行处理,具体可以包括对视频数据帧进行解码、后处理和呈现;其中关于视频数据帧后处理和呈现将后面的进行详细描述。在本发明实施例中,即利用虚拟机 205a 的硬件代替视频播放器 332 自身的软件算法对视频数据帧进行处理,这里虚拟机的 205a 硬件,本领域技术人员可以理解,并不是虚拟机的 205a 真实硬件,而是通过虚拟机监视器 320a 提供的虚拟硬件。其中一种实现方式为,视频播放器 332 在启动查询虚拟显示驱动 334 的视频硬件加速能力,可以通过调用 GetCaps 函数进行查询,或者,可以通过调用 CreateDevice 或 CreateDecodeDevice 函数创建硬件加速设备,确定虚拟显示驱动 334 是否支持视频硬件加速功能。当视频播放器 332 启动查询虚拟显示驱动 334 的视频硬件加速能力时,虚拟显示驱动 334 向视频播放器 332 返回支持视频硬件加速能力的查询请求响应,视频播放器 332 根据该查询请求响应确定虚拟机 205a 支持视频硬件加速能力。本发明实现过程中,发现一些视频播放器会从客户端获取客户端已经解码的视频数据帧,从而影响正常的视频处理,因此视频播放器 332 向虚拟显示驱动 334 查询视频硬件加速能力的同时,本发明实施例还可以确定视频播放器 332 是否支持视频重定向功能,确定视频播放器 332 是否支持视频重定向的一种方式虚拟机 205a 通过网络从虚拟桌面管理系统 106 中获取支持视频重定向的视频播放器的列表(该支持视频重定向的视频播放器的列表也可以存储在远程服务器 204a 本地),该列表中包括了支持视频重定向的视频播放器的信息,支持视频重定向的视频播放器的信息包括支持视频重定向的视频播放器的名称、进程名和版本号。其中,视频播放器 332 启动时,视频播放器 332 发送向虚拟显示驱动 334 发送视频硬件加速能力查询请求,会调用 GetCaps 函数查询视频硬件加速能力,或者调用 CreateDevice 或 CreateDecodeDevice 函数创建视频硬件加速。虚拟显示驱动 334 在收到视频播放器 332 的 GetCaps、或者 CreateDevice 或 CreateDecodeDevice 调用时,检查发起调用的视频播放器 332 名称和视频播放器 332 进程名,与从虚拟桌面管理系统 106 获得的视频播放器列表中的视频播放器的信息进行匹配。若发起调用视频播放器 332 名称和视频播放器 332 进程名在该列表中,则通过客户操作系统 330 的应用程序接口 API 定位到该视频播放器 332 的进程的可执行文件(exe),并从可执行文件中提取版本号,与从虚拟桌面管理系统 106 获取的视频播放器列表中的视频播放器的版本号匹配,若匹配则确定视频播放器 332 支持视频重定向,并响应视频播放器 332 的 GetCaps、或者 CreateDevice 或 CreateDecodeDevice 函数调用,虚拟显示驱动 334 向所述视频播放器 332 发送查询请求响应;所述查询请求响应表明虚拟显示驱动 334 支持视频硬件加速,即在视频播放器 332 向虚拟显示驱动 334 查询是否支持视频硬件加速的同时,虚拟显示驱动判断视频播放器 332 是否支持视频重定向,只有当虚拟显示驱动 334 支持视频硬件加速并且视频播放器 332 支持视频重定向时,虚拟显示驱动 334 才向视频播放器 332 返回查询请求响应,视频硬件加速查询请求响应表明虚拟显示驱动 334 支持视频硬件加速。如果视频播放器 332 不支持视频重定向,则不响应查询请求。本发明实施例关于确定视频播放器 332 是否支持重定向的方法只是一种示例性说明,也即视频播放器 332 的信息和支持视频重定向的视频播放器列表的信息不仅仅局限于视频播放器的名称、进程名和版本号,其他可以唯一确定视频播放器的信息也可以适用于本发明。同时其他可以判断视频播放器 332 是否支持视频重定向的方法也适用于本案。

[0116] 视频播放器 332 向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令,如

图 3a 所示,一种实现方式为虚拟显示驱动 334 截获视频播放器 332 发送的视频数据帧和视频数据帧处理指令。

[0117] 支持视频重定向功能的视频播放器 332 加载视频文件后,启动视频硬件加速功能,通过**DirectX®**运行库 333 向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频硬件加速 (**DirectX®** Video Acceleration, 简称为 DXVA), 是微软®公司专门定制的视频加速规范,它共有两个版本,分别是 DXVA1.0 和 DXVA2.0。视频硬件加速规范不仅仅是在视频播放过程中的视频数据帧解码计算规范,其还与视频数据帧后期处理相关。**DirectX®**运行库是由微软®公司创建的多媒体编程接口。由 C++ 编程语言实现,被广泛使用于**Microsoft® Windows®**、**Microsoft® Xbox®**和 **Microsoft® Xbox 360®** 电子游戏开发,本发明实施例及后面发明实施例描述的**DirectX®**运行库与此相同。

[0118] 其中,视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令。通常,视频数据帧发送到虚拟显示驱动 334,根据视频数据帧解码指令,对该视频数据帧进行解码,然后再根据视频数据帧后处理指令对该解码的视频数据帧进行后处理;通常, **DirectX®**运行库 333 调用虚拟显示驱动 334 提供的视频硬件加速接口中用于视频后处理的接口,通知虚拟显示驱动 334 完成视频数据帧的后处理。视频数据帧后处理指令类型包括对视频数据帧进行亮度、色调、饱和度和缩放等操作中的至少一种,后面将详细描述视频数据帧后处理。将进行后处理的视频数据帧根据呈现指令发送至显卡 325 (即虚拟显卡),由显卡 325 发送至物理显卡 (本发明实施例附图 2 中省略了物理显卡) 通过显示装置进行呈现。本发明实施例中,视频数据帧及视频数据帧处理指令发送至虚拟显示驱动 334 时,虚拟显示驱动 334 截获该视频数据帧和视频数据帧处理指令,并将截获的该视频数据帧和视频数据帧处理指令发送到显示协议代理 335,显示协议代理 335 通过虚拟网络接口卡 326 再经物理网络接口卡 312,将截获的该视频数据帧和视频数据帧处理指令发送到客户端 101a。虚拟显卡驱动 334 实际上并没有根据视频数据帧处理指令对视频数据帧进行处理,而是将视频数据帧和视频数据帧处理指令截获,并通过显示协议代理 335 发送至客户端 101a 进行处理。因此,本发明实施例中截获视频数据帧和视频数据帧处理指令,使得视频播放器 332 向虚拟显示驱动 334 发送的视频数据帧并没有在服务器 204a 侧根据视频数据帧处理指令进行处理,而是发送到客户端进行处理,从而客户端能够根据视频数据帧处理指令对视频数据帧进行解码、后处理和呈现,保证客户端用户的视频体验;同时只要服务器侧视频播放器支持视频重定向功能即可实现本发明,不依赖某一种视频播放器,如不依赖于**Microsoft®**公司推出的 **DirectShow®**框架的 **Media Player®**播放器。并且视频播放器向虚拟显示驱动发送的视频数据帧并没有在服务器侧根据视频数据帧处理指令进行处理,发送的仍是编码的视频数据帧,从而可以进一步节约服务器计算资源和网络传输带宽。

[0119] 下面具体描述虚拟显示驱动 334 截获视频数据帧和视频数据帧处理指令的过程。具体地,虚拟机 205a 中的视频播放器 332 根据对应的客户端 101a 发送的视频播放请求,加载视频文件之后,开启视频硬件加速,向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频播放器 332 加载视频文件后,对加载后的视频文件,即对视频数据流解复用,获得其中的视频数据帧。具体地,就视频播放而言,在**DirectX®**运行库 333 中定义了视频硬件加速接口,通过视频播放器 332 调用**DirectX®**运行库 333 启动视频硬件加速,

通知虚拟显示驱动 334 根据视频数据帧处理指令对视频播放器 332 发送的视频数据帧进行处理。当视频数据帧和视频数据帧处理指令发送至虚拟显示驱动 334 时,虚拟显示驱动 334 截获视频数据帧和视频数据帧处理指令。如图 3a 所示,在 **Windows®** 操作系统平台,虚拟显示驱动为 **Windows®** Display Driver Model (简称 WDDM 显示驱动模型) 或者为 **Windows®** XP Display Driver Model (简称 XPDM 显示驱动模型)。具体地, WDDM 或 XPDM 显示驱动模型均包括 Display Driver 和 Miniport Driver 两部分模块。本发明实施例中,一种实现方案为虚拟显示驱动的 Display Driver 模块截获视频数据帧和视频数据帧处理指令。本发明实施例中虚拟显示驱动的 Display Driver 模块将截获的视频数据帧和视频数据帧处理指令,发送到显示协议代理 335,显示协议代理 335 将截获的视频数据帧和视频数据帧处理指令发送到客户端 101a 进行处理。具体地,一种实现方案为 WDDM 或 XPDM 显示驱动模型在 Display Driver 模块中定义视频硬件加速接口,当视频数据帧和视频数据帧处理指令通过该接口时,截获该视频数据帧和视频数据帧处理指令。另一种实现方案是 Display Driver 模块截获该视频数据帧和视频数据帧处理指令,经 Miniport Driver 模块,将截获的视频数据帧和视频数据帧处理指令发送到显示协议代理 335,显示协议代理 335 向客户端 101a 发送截获的视频数据帧和视频数据帧处理指令。

[0120] 本发明实施例截获视频数据帧和视频数据帧处理指令的另一种实施方式,如图 3b 所示,虚拟机 205a 中的视频播放器 332 根据对应的客户端 101a 发送的视频播放请求,加载视频文件之后,开启视频硬件加速,向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频播放器 332 加载视频文件后,对加载后的视频文件,即对视频数据流解复用,获得其中的视频数据帧。具体地,就视频播放而言,在 **DirectX®** 运行库 333 中定义了视频硬件加速接口,通过视频播放器 332 调用 **DirectX®** 运行库 333 启动视频硬件加速,通知虚拟显示驱动 334 根据视频数据帧解码指令对视频播放器 332 发送的视频数据帧进行解码。**DirectX®** 运行库 333 调用所述虚拟显示驱动 334 对所述视频数据帧进行处理时,即在视频数据帧和视频数据帧处理指令通过 **DirectX®** 运行库 333 后,到达虚拟显示驱动 334 之前,动态链接库截获该视频数据帧和视频数据帧处理指令。具体地,其中一种动态链接库实现方式为采用 Hook 技术截获 **DirectX®** 运行库 333 对虚拟显示驱动 334 的调用,需要实现一个对虚拟显示驱动 334 Hook 的动态链接库文件 (dll 文件),并在注册表中将其设为最先加载,以此拦截 **DirectX®** 运行库 333 对虚拟显示驱动 334 的调用,从而实现视频数据帧和视频数据帧处理指令的截获。将截获的视频数据帧和视频数据帧处理指令发送到显示协议代理 335,显示协议代理 335 将截获的视频数据帧和视频数据帧处理指令发送到客户端 101a。仍以 **Windows®** 操作系统平台,虚拟显示驱动为 **Windows®** Display Driver Model (简称 WDDM 显示驱动模型) 或者为 **Windows®** XP Display Driver Model (简称 XPDM 显示驱动模型),这里以 WDDM 显示驱动模型为例,进行说明。具体地, WDDM 显示驱动模型包括 Display Driver 和 Miniport Driver 两部分模块。当 **DirectX®** 运行库 333 调用虚拟显示驱动 334 的 Display Driver 时,动态链接库截获视频数据帧和视频数据帧处理指令,动态链接库将截获的视频数据帧和视频数据帧处理指令,发送到显示协议代理 335,显示协议代理 335 将截获的视频数据帧和视频数据帧处理指令发送到客户端 101a 进行处理。

[0121] 本发明实施例截获视频数据帧和视频数据帧处理指令的另一种实施方式,如图 3c

所示,虚拟机 205a 中的视频播放器 332 根据对应的客户端 101a 发送的视频播放请求,加载视频文件之后,开启视频硬件加速,向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频播放器 332 加载视频文件后,对加载后的视频文件,即对视频数据流解复用,获得其中的视频数据帧。具体地,就视频播放而言,在DirectX®运行库 333 中定义了视频硬件加速接口,通过视频播放器 332 调用DirectX®运行库 333 向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令,启动视频硬件加速,通知虚拟显示驱动 334 对视频播放器 332 发送的视频数据帧进行处理。在视频数据帧和视频数据帧处理指令通过DirectX®运行库 333 时, DirectX®运行库截获该视频数据帧和视频数据帧处理指令。

[0122] 本发明实施例截获视频数据帧和视频数据帧处理指令的另一种实施方式,如图 3d 所示,虚拟机 205a 中的视频播放器 332 根据对应的客户端 101a 发送的视频播放请求,加载视频文件之后,开启视频硬件加速,向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频播放器 332 加载视频文件后,对加载后的视频文件,即对视频数据流解复用,获得其中的视频数据帧。具体地,就视频播放而言,在DirectX®运行库 333 中定义了视频硬件加速接口,视频播放器 332 调用DirectX®运行库 333 启动视频硬件加速,发送视频数据帧和视频数据帧处理指令时,即视频数据帧和视频数据帧处理指令到达DirectX®运行库 333 之前,动态链接库截获视频数据帧和视频数据帧处理指令。为此,需要实现一个对DirectX®运行库 333Hook 的动态链接库文件(dll 文件),并在注册表中将其设为最先加载,以此截获视频播放器 332 对DirectX®运行库 333 的调用,从而实现视频数据帧和视频数据帧处理指令的截获。

[0123] 本发明实施例结合图 3a,具体流程描述如图 4 所示,服务器 204a 上的虚拟机 205a 通过显示协议代理 335 接收来自客户端 101a 发送的视频播放请求。步骤 401:视频播放器 332 加载视频文件。位于虚拟机 205a 中的视频播放器 332 根据来自客户端 101a 的视频播放请求加载视频文件。步骤 402:通过DirectX®运行库 333 向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频播放器 332 加载视频文件后,对加载的视频文件解复用获得视频数据帧,启动视频硬件加速,视频播放器 332 通过 DirectX®运行库 333 向虚拟显示驱动 334 发送视频数据帧和视频数据帧处理指令。视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令。步骤 403:虚拟显示驱动 334 截获视频播放器 332 向虚拟显示驱动 334 发送的视频数据帧和视频数据帧处理指令。步骤 404:显示协议代理 335 向客户端 101a 发送截获的所述视频数据帧和所述视频数据帧处理指令。具体地,显示协议代理 335 通过虚拟网络接口卡 326,然后经物理网络接口卡 312,将截获的所述视频数据帧和所述视频数据帧处理指令发送至客户端 101a。步骤 405:客户端 101a 接收视频数据帧和视频数据帧处理指令。步骤 406:根据所述视频数据帧处理指令对所述视频数据帧进行处理。

[0124] 如图 5 所示,客户端 101a 通过远程协议接收模块 501 接收虚拟机 205a 通过显示协议代理 335 发送的视频数据帧和视频数据帧处理指令。具体地,图 5,作为客户端的一种示例性说明,并不是对本发明中使用的客户端的具体限定,同时附图中只是给出了一些组件的说明,对于实现客户端正常功能的一些其他组件此处并没有说明,但是本领域技术人员根据具体实现,可以从附图中获得这些组件。包括存储器 500、中央处理器 510、网络接口卡 520、显卡 530、硬盘 540 和串行接口(简称串口)550,串口可以连接鼠标和键盘等外设。其



中,显卡 530 连接显示器 560,在另一种客户端中,显示器 560 可以与客户端连接在一起,而不必是两个独立的物理装置。客户端运行时,存储器 500 中加载计算机程序,图 5 中,存储器 500 中包括远程桌面协议接收模块 501、视频播放器 502 和操作系统 503。网络接口卡 520 用于接收虚拟机 205a 通过显示协议代理 335 发送的视频数据帧和视频数据帧处理指令,并通过系统总线传递到远程桌面协议接收模块 501。远程桌面协议接收模块 501 用于接收该视频数据帧和视频数据帧处理指令。视频播放器 502 对远程桌面协议接收模块 501 接收的视频数据帧和视频数据帧处理指令进行处理。操作系统 503 为客户端提供系统运行环境,可以为 Windows®操作系统、Linux®操作系统及其他操作系统,本发明对此不作具体限定。视频播放器 502 根据接收的视频数据帧处理指令对视频数据帧进行处理。针对某一帧视频数据,视频数据帧处理指令依次包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令。当客户端 101a 的视频播放器 502 使用视频播放器自身的视频处理能力来处理视频数据帧时,即利用视频播放器对视频数据帧进行软件解码,然后根据视频数据帧后处理指令对解码的视频数据帧进行后处理。将进行后处理的视频数据帧根据视频数据帧呈现指令发送到显卡,通过客户端 101a 连接的显示器 560 显示视频数据图像。

[0125] 当客户端 101a 支持视频硬件加速时,视频播放器 502 调用视频硬件加速应用程序接口 (API),启动视频硬件加速,将视频数据帧和视频数据帧处理指令通过视频硬件加速 API 发送至相应的硬件(如 Nvidia®、AMD®或 ATI®等公司的显卡芯片,或者 Intel® Atom CPU 芯片中的核芯显卡),进行视频硬件加速。视频播放器 502 在启动时会检测客户端硬件是否支持视频硬件加速,对于查询是否支持视频硬件加速,如前面描述视频播放器 332 进行视频硬件加速能力查询的方式类似,对此不再赘述。如果支持视频硬件加速,则远程协议接收模块 501 接收虚拟机 205a 通过显示协议代理 335 发送的视频数据帧和视频数据帧处理指令,视频播放器 502 通过调用视频硬件加速 API,启动视频硬件加速。客户端 101 启动视频硬件加速功能,可以节约客户端中央处理器的计算资源。接下来,将详细描述客户端 101a 对解码的视频数据帧根据视频数据帧后处理指令和视频数据帧呈现指令进行相应处理的过程。

[0126] 本实施例中以 Microsoft®的 DXVA 为例,对视频数据帧后处理和视频数据帧呈现进行描述。DXVA 不仅仅提供视频数据帧解码处理,还包括视频数据帧后处理能力,比如图像缩放、色彩空间转换和反交错等。如图 6 所示,DXVA 为视频数据帧后处理流程图。DXVA 可以对主数据流和 15 路分数据流进行处理,将主数据流进行噪声过滤、反交错处理、细节过滤和色彩空间转换后,与经过色彩空间转换后的分数据流进行混合,将混合后的视频数据流进行图像调整,其中图像调整包括亮度、对比度、饱和度调整等。将经过图像调整后的视频数据流进行图像缩放,从而可以调整图像大小。将经图像缩放处理后的数据进行色彩空间转换,从而使视频数据流进行显示。其中,颜色空间转换,包括由 RGB 色彩空间转换至 YUV 色彩空间,或者由 YUV 色彩空间转换至 RGB 色彩空间,其中 RGB 分别表示红、绿和蓝色,YUV 分别表示明亮度、色度和浓度,本发明实施例在此不再赘述。

[0127] 图 6 描述了视频后处理的过程。实际应用中不一定所有的功能都会用到。实际测试中发现,多数播放器如 Microsoft® Media Player,暴风影音®,QQ®影音播放器等默认只使用了图像缩放功能,用以将解码后的视频数据帧从原始大小缩放到播放窗口大小。本发明实施例中客户端 101a 进行视频数据帧后处理以图像缩放为例,将解码后的视频数

据帧进行图像缩放,以将原始图像大小调整到适合在客户端连接的显示器 560 上显示。具体效果如图 7a 所示。视频数据帧后处理还可以包括透明度混合,如图 7b 所示。本发明实施例中视频数据帧后处理还可以包括亮度、对比度和饱和度调整。从而使客户端呈现给用户更好的视频体验效果。

[0128] 将解码的视频数据帧进行视频数据帧后处理,根据视频数据帧呈现指令,将经视频数据帧后处理的视频数据帧发送到显卡 530,通过显示器 560 进行显示。其中,视频数据帧呈现指令携带显示参数,显示参数包括要显示的视频数据帧和视频数据帧的显示区域。如图 8 所示,图中经过视频数据帧后处理之后形成的图像,包括需要显示的 4 个区域,分别表示为①、②、③和④,同时远程桌面图像,即虚拟机 205a 的桌面图像通过远程桌面协议传递到客户端 101a,客户端 101a 通过 2D 渲染显示功能,在客户端 101 显示器 560 上显示,其中,远程桌面图像有一处区域用于对用户呈现“显示设定、音频设定和字幕设定”选项,在呈现视频时,同时需要呈现“显示设定、音频设定和字幕设定”选项区域,而该显示设定、音频设定和字幕设定”选项区域与①、②、③和④四个显示区域围成的中间区域 对应。因此,①、②、③和④四个显示区域围成的中间区域对应的视频数据帧不需要显示,构成①、②、③和④四个显示区域围成的中间区域的视频数据帧中包含的视频数据帧显示参数用于确认显示区域,显示参数用于定义不显示该视频数据帧。从而使视频画面与远程桌面融合为一体,为客户端用户提供更好的视频观看体验。

[0129] 本发明实施例中如果客户端 101a 为非Microsoft®操作系统,则可以使用其他视频硬件加速功能根据视频数据帧处理对视频数据帧进行处理。

[0130] 基于本发明上述实施例的描述,本发明实施例还公开了如图 9 所示的服务器 90,包括截获单元 901 和发送单元 902。其中,截获单元 901 用于截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;发送单元 902,用于向客户端发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端根据所述视频数据帧处理指令对所述视频数据帧进行处理。本实施例中的截获单元 901 将视频播放器发送到虚拟显示驱动的视频数据帧和视频数据帧处理指令截获,从而服务器 90 并未根据视频数据帧处理指令对视频数据帧进行处理,通过发送单元 902 发送至客户端。一种实现方案为服务器 90 的截获单元 901 将视频播放器发送到虚拟显示驱动的视频数据帧和视频数据帧处理指令截获,从而服务器 90 并未根据视频数据帧处理指令对视频数据帧进行处理,而是直接由虚拟显示驱动截获通过发送单元 902 发送至客户端,从而客户端能够根据视频数据帧处理指令对视频数据帧进行解码、后处理和呈现,保护客户端用户的视频体验;同时只要服务器侧视频播放器支持视频重定向功能即可实现本发明,不依赖某一种视频播放器,如不依赖于Microsoft®公司推出的DirectShow®框架的多媒体播放器。并且视频播放器向虚拟显示驱动发送的视频数据帧并没有在服务器侧根据视频数据帧处理指令进行处理,发送的仍是编码的视频数据帧,从而可以进一步节约服务器计算资源和网络传输带宽。

[0131] 另一种实施方案,截获单元 901 为动态连接库,视频播放器 332 调用 DirectX®运行库 333 启动视频硬件加速,通知虚拟显示驱动 334 对视频播放器 332 发送的视频数据帧进行解码。DirectX®运行库 333 调用所述虚拟显示驱动 334 根据所述视频数据

帧处理指令对所述视频数据帧进行处理时,即在视频数据帧和视频数据帧解码指令通过 **DirectX®** 运行库 333 后,到达虚拟显示驱动 334 之前,动态链接库截获该视频数据帧和视频数据帧处理指令。具体地,其中一种动态链接库实现方式为采用 Hook 技术截获 **DirectX®** 运行库 333 对虚拟显示驱动 334 的调用,需要实现一个对虚拟显示驱动 334 Hook 的动态链接库文件(dll 文件),并在注册表中将其设为最先加载,以此拦截 **DirectX®** 运行库 333 对虚拟显示驱动 334 的调用,从而实现视频数据帧和视频数据帧处理指令的截获。发送单元 902 将截获的视频数据帧和视频数据帧处理指令发送到客户端 101a。仍以 **Windows®** 操作系统平台,虚拟显示驱动为 **Windows®** Display Driver Model(简称 WDDM 显示驱动模型)或者为 **Windows®** XP Display Driver Model(简称 XPDM 显示驱动模型),这里 WDDM 显示驱动模型以为例,进行说明。具体地,WDDM 显示驱动模型包括 Display Driver 和 Miniport Driver 两部分模块。当 **DirectX®** 运行库 333 调用虚拟显示驱动 334 的 Display Driver 时,动态链接库截获视频数据帧和视频数据帧处理指令,发送单元 902 将截获的视频数据帧和视频数据帧处理指令发送到客户端 101a 进行处理。

[0132] 另一种实现方式,截获单元 901 作为动态链接库,在视频播放器 332 调用 **DirectX®** 运行库 333 启动视频硬件加速时截获视频数据帧和视频数据帧处理指令,即在视频数据帧和视频数据帧处理指令到达 **DirectX®** 运行库 333 之前截获视频数据帧和视频数据帧处理指令。具体实现原理同在视频数据帧和视频数据帧处理指令通过 **DirectX®** 运行库 333,到达虚拟显示驱动 334 之前,动态链接库截获该视频数据帧和视频数据帧处理指令的实现方式。

[0133] 另一种实现方式,截获单元 901 为 **DirectX®** 运行库 333,在视频数据帧和视频数据帧处理指令通过 **DirectX®** 运行库 333 时, **DirectX®** 运行库截获该视频数据帧和视频数据帧处理指令。

[0134] 本实施例中截获视频数据帧和视频数据帧处理指令的方案可以参见前面实施例相应部分的描述,在此不再赘述。

[0135] 本发明实施例中,虚拟显示驱动用于在所述视频播放器向所述虚拟显示驱动发送所述视频数据帧和所述视频数据帧处理指令之前,接收所述视频播放器发送的视频硬件加速能力查询请求,所述虚拟显示驱动向所述视频播放器发送查询请求响应;所述查询请求响应表明所述虚拟显示驱动支持视频硬件加速。在本发明实现中,发现一些视频播放器会从客户端获取客户端已经解码的视频数据帧,从而影响视频播放,此类视频播放器并不真正支持视频重定向。因此,还可以判断视频播放器是否支持视频重定向,具体实施方式可以为视频播放器向所述虚拟显示驱动发送视频硬件加速能力查询请求时,所述虚拟显示驱动根据所述视频硬件加速能力查询请求获取所述视频播放器的信息;所述虚拟显示驱动根据所述视频播放器的信息,与支持视频重定向的视频播放器列表进行匹配;如果所述视频播放器的信息与所述支持视频重定向的视频播放器列表中视频播放器的信息匹配,则所述虚拟显示驱动向所述视频播放器发送所述查询请求响应;其中所述支持视频重定向的视频播放器列表包含支持视频重定向的视频播放器的信息。虚拟显示驱动通过确认视频播放器支持视频重定向,才向视频播放器返回查询响应,以表明虚拟显示驱动支持视频硬件加速。即只有当视频播放器支持视频重定向时,虚拟显示驱动才向视频播放器返回查询请求响应,表明虚拟显示驱动支持视频硬件加速。以便于当视频播放器查询到虚拟显示驱动支持视频

硬件加速,向虚拟显示驱动发送视频数据帧和包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令的视频数据帧处理指令,截获视频数据帧和视频数据帧处理指令并发送至客户端处理,从而节约服务器计算机资源和网络传输带宽,保证客户端用户的视频体验。同时只要服务器侧视频播放器支持视频重定向功能即可实现本发明,不依赖某一种视频播放器。关于本发明实施例提供的服务器的描述可以参见前述实施例的描述。

[0136] 与服务器相对应,本发明实施例还公开了如图 10 所示的客户端 100,包括接收单元 1001 和处理单元 1002。其中,接收单元 1001 用于接收视频数据帧和视频数据帧处理指令;其中,所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令;处理单元 1002,用于根据所述视频数据帧处理指令对所述视频数据帧进行处理。其中,处理单元 1002 具体用于:根据所述视频数据帧解码指令对所述视频数据帧进行解码;根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理;根据所述视频数据帧呈现指令对所述视频后处理的视频数据帧进行显示。

[0137] 本发明实施例提供的客户端 100,通过接收服务器发送的视频数据帧和包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令的视频数据帧处理指令,根据视频数据帧处理指令对视频数据帧进行处理,从而保证客户端用户的视频体验,同时充分利用了客户端的计算资源。

[0138] 本发明实施例公开了一种视频重定向系统,如图 11 所示,包括服务器 90 和客户端 100。服务器 90 截获视频播放器向虚拟显示驱动发送的视频数据帧和视频数据帧处理指令;其中,所述视频数据帧是由所述视频播放器对加载的视频文件解复用获得的;所述视频数据帧处理指令包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令,并向客户端 100 发送截获的所述视频数据帧和所述视频数据帧处理指令,以使所述客户端 100 根据所述视频数据帧处理指令对所述视频数据帧进行处理。客户端 100 接收所述视频数据帧和所述视频数据帧处理指令;并根据所述视频数据帧处理指令对所述视频数据帧进行处理。根据所述视频数据帧处理指令对所述视频数据帧进行处理,具体包括:根据所述视频数据帧解码指令对所述视频数据帧进行解码;根据所述视频数据帧后处理指令对所述解码的视频数据帧进行视频数据帧后处理;根据所述视频数据帧呈现指令对所述视频后处理的视频数据帧进行显示。

[0139] 关于本实施例提供的系统的细节描述可参见服务器 90 和客户端 100 的描述。本发明实施例中提供的服务器 90 和客户端 100 具体可以为普通的计算机、移动终端、工作站或专用服务器等,本发明不作具体限定,具有存储器和至少一个中央处理器,所述存储器与所述至少一个中央处理器通过总线相连,存储器存储计算机指令,所述至少一个中央处理器用于执行存储器中的计算机指令,同时也包含通用的外部接口等实现服务器功能的一些必要组件。其中,存储器中存储的计算机指令的算法可以参照本发明实施例中对应的方法描述。

[0140] 本发明实施例提供的视频重定向系统,在服务器侧截获视频播放器向虚拟显示驱动发送视频数据帧和包括视频数据帧解码指令、视频数据帧后处理指令和视频数据帧呈现指令的视频数据帧处理指令,并发送到客户端,客户端接收所述视频数据帧和所述视频数据帧处理指令,根据所述视频数据帧处理指令对所述视频数据帧进行解码、视频后处理和呈现,保证客户端用户的视频体验。同时只要服务器侧视频播放器支持视频重定

向功能即可实现本发明,不依赖某一种视频播放器,如不依赖于Microsoft®公司推出的DirectShow®框架的Media Player®播放器。并且视频播放器向虚拟显示驱动发送的视频数据帧并没有在服务器侧根据视频数据帧处理指令进行处理,发送的仍是编码的视频数据帧,从而可以进一步节约服务器计算资源和网络传输带宽。。

[0141] 本发明实施例提供的基于虚拟桌面的远程桌面架构,本领域技术人员可以意识到,本发明并不限于本实施例的所描述的虚拟桌面架构,对于基于其他类型的虚拟桌面的远程桌面本发明同样适用。另外,本发明并不仅仅适用于基于虚拟桌面的远程桌面架构,同样也适用于其他向远程客户端提供视频数据的应用场景。

[0142] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0143] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0144] 在本申请所提供的几个实施例中,应该理解到,所公开的系统、装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0145] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0146] 另外,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单元中,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单元中。

[0147] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。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或者该技术方案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)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:NAS(Network Attached Storage)、U盘、移动硬盘、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0148] 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,可轻易想到变化或替换,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所述以权利要求的保护范围为准。

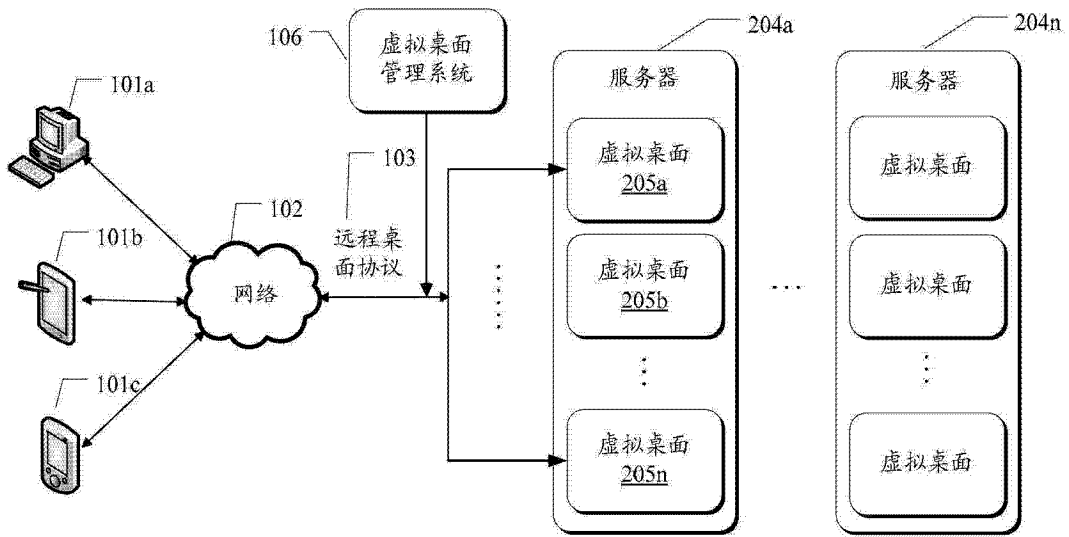


图 1

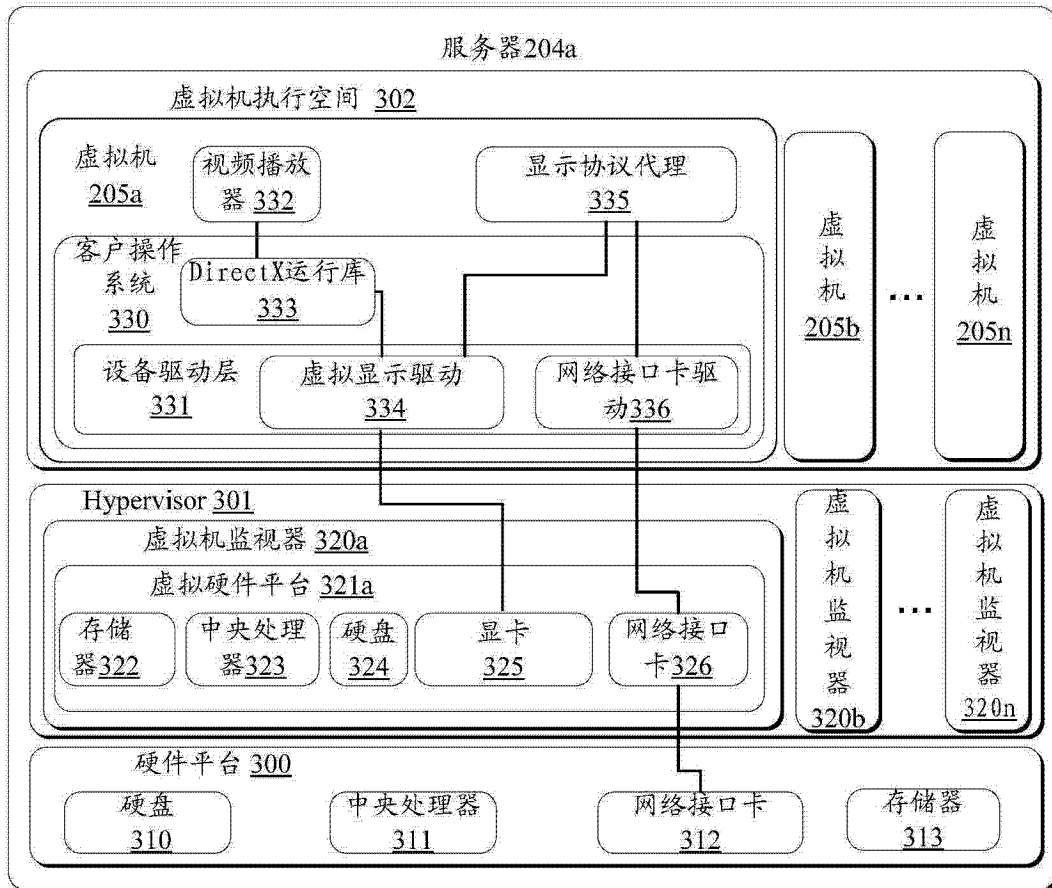


图 2

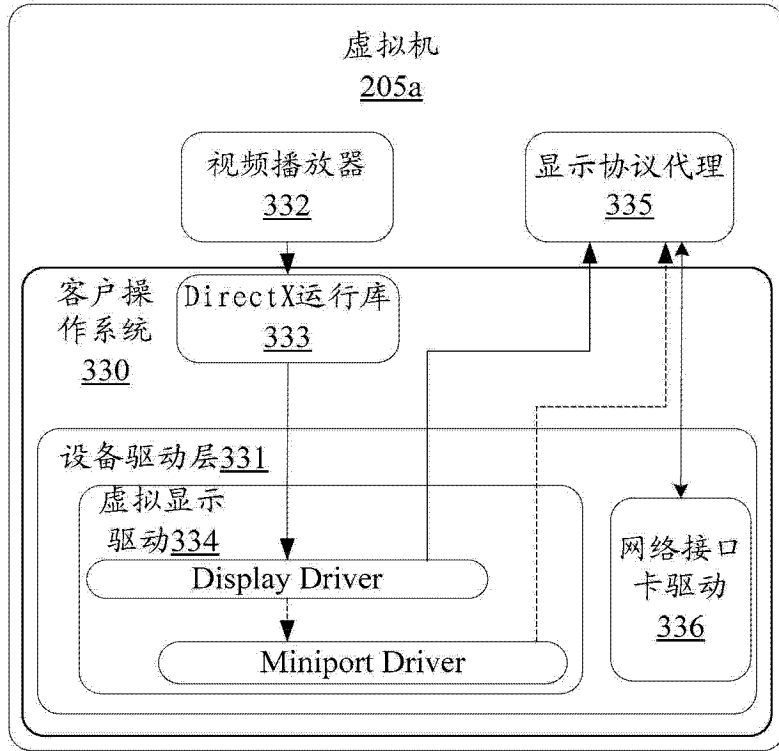


图 3a

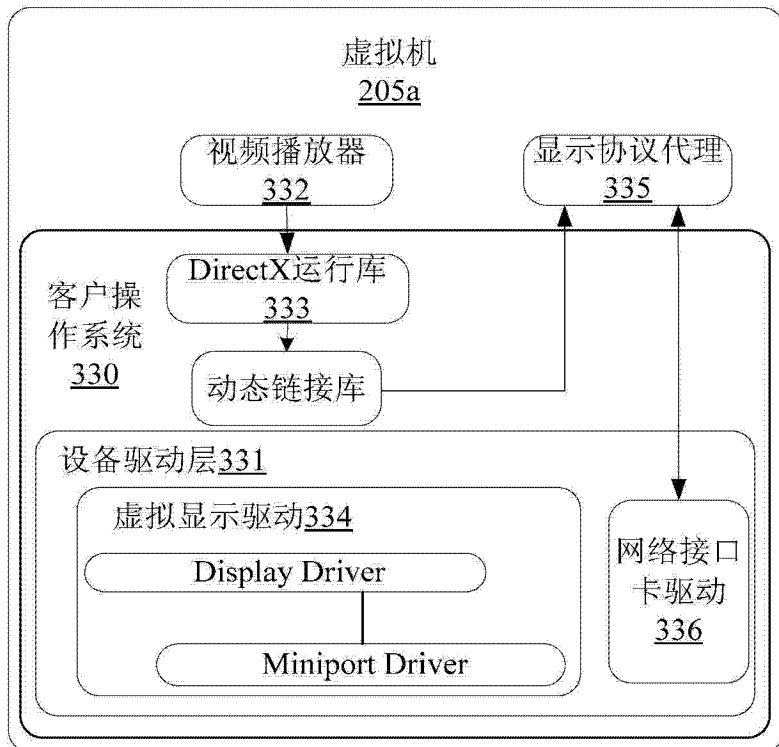


图 3b

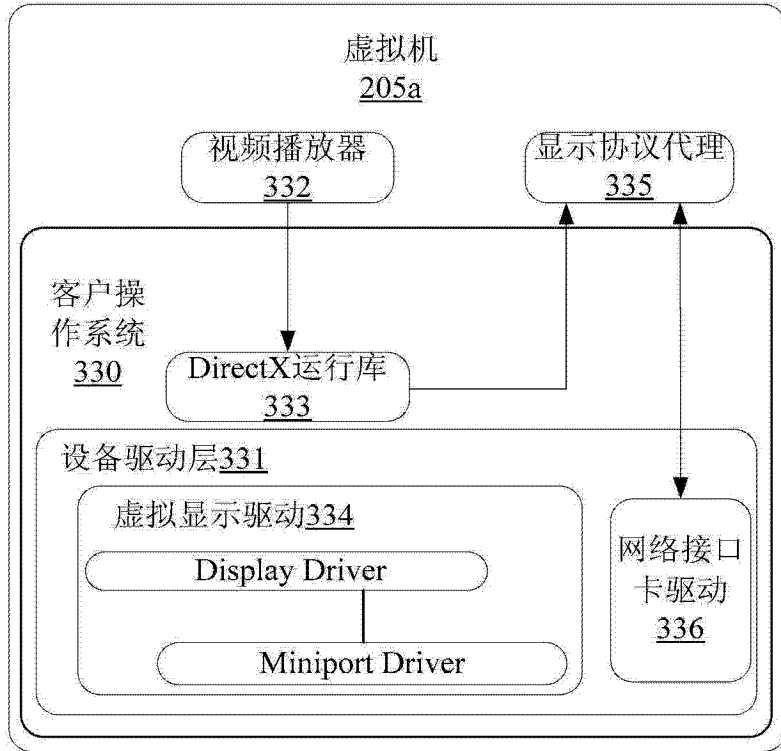


图 3c

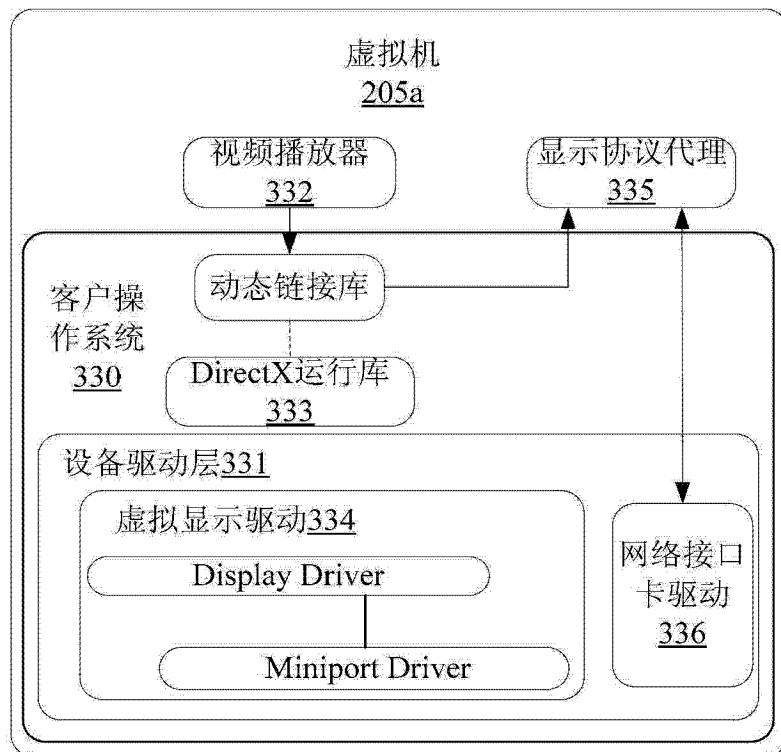


图 3d



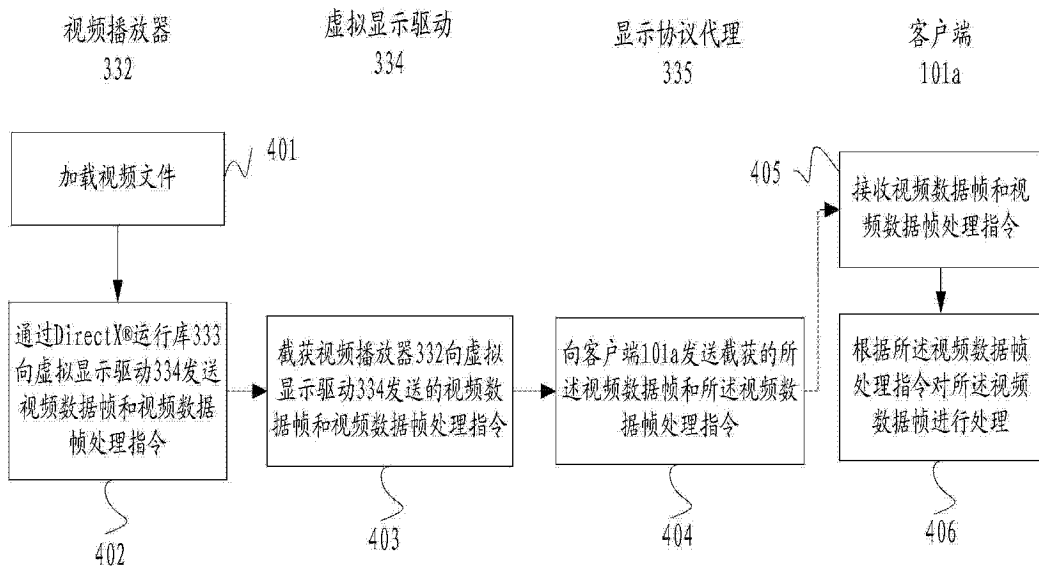


图 4

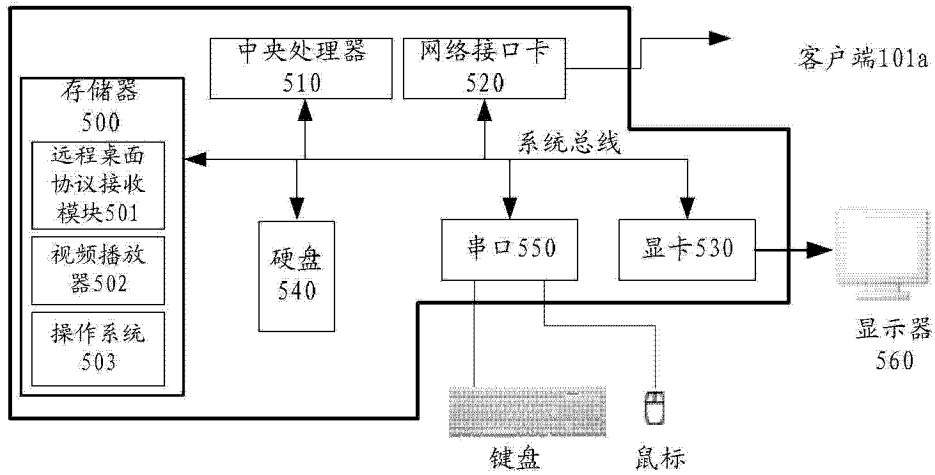


图 5

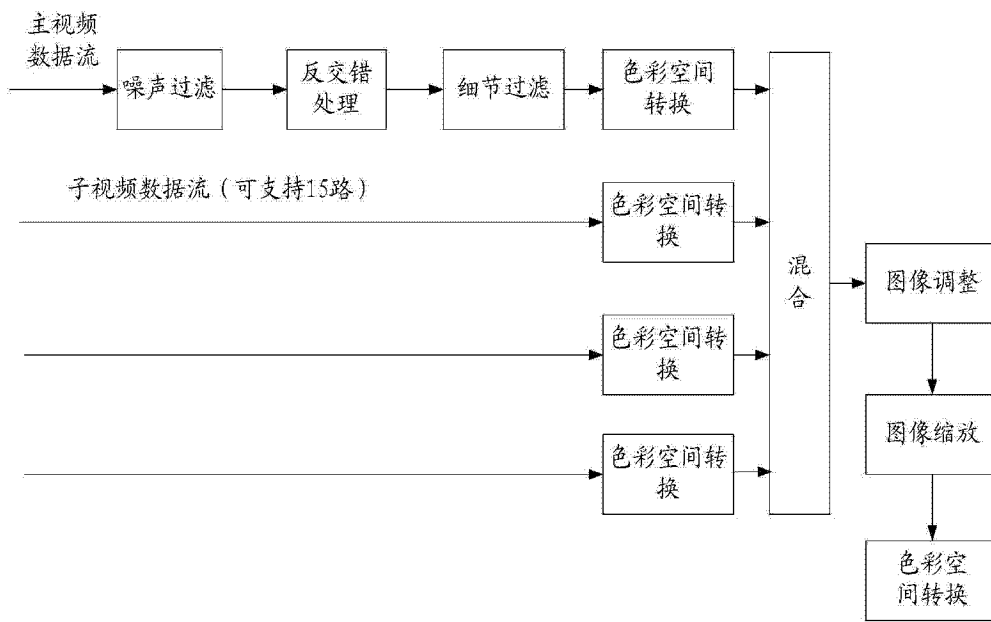


图 6

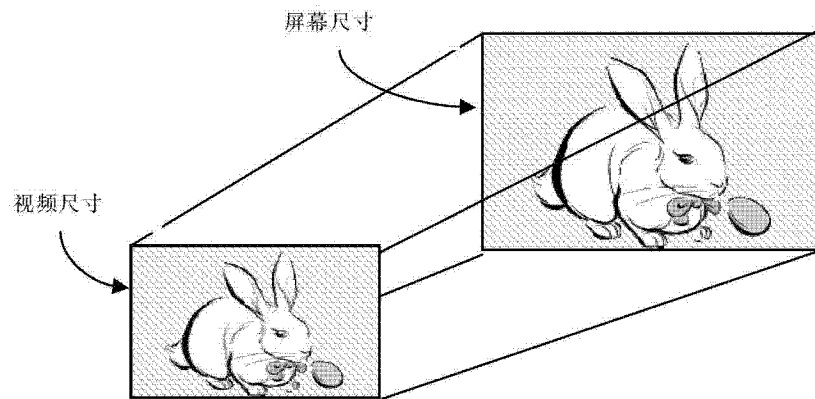


图 7a

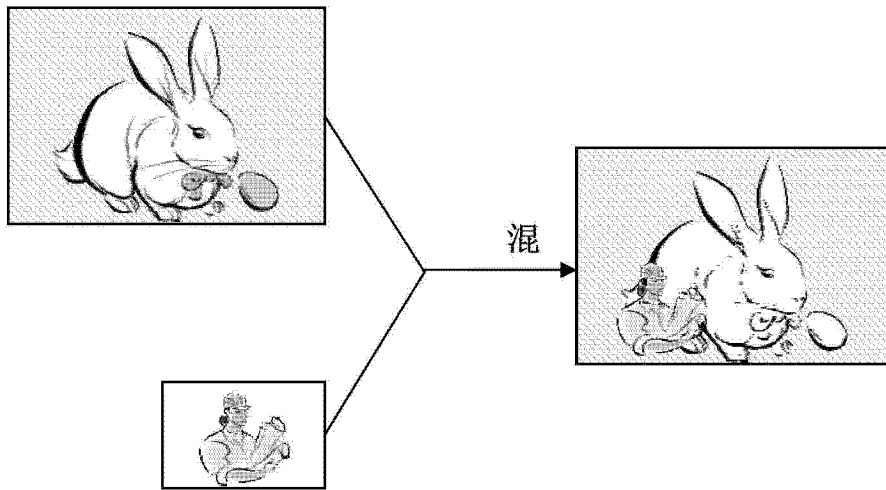


图 7b

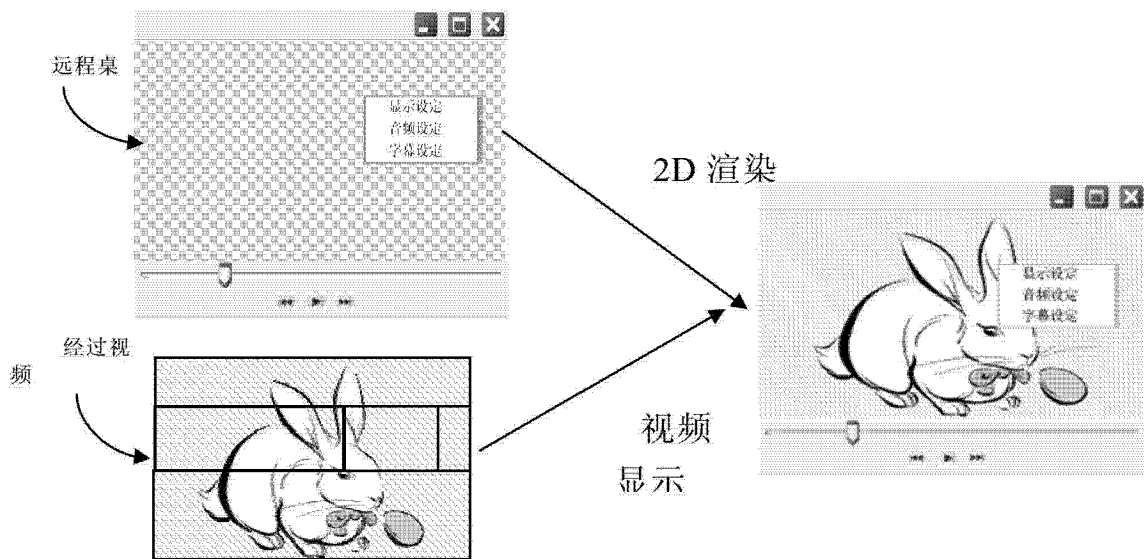


图 8

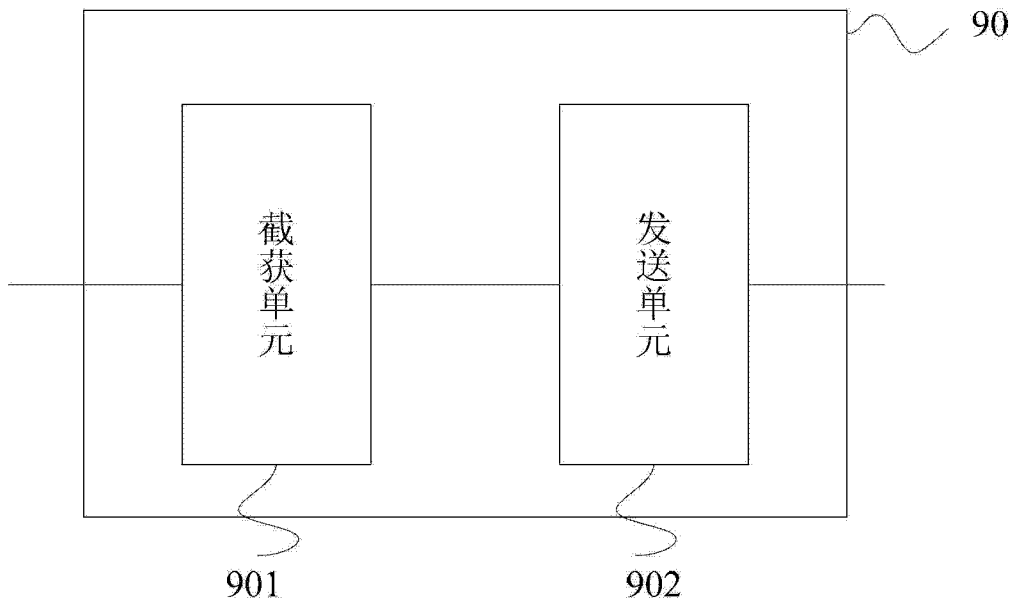


图 9

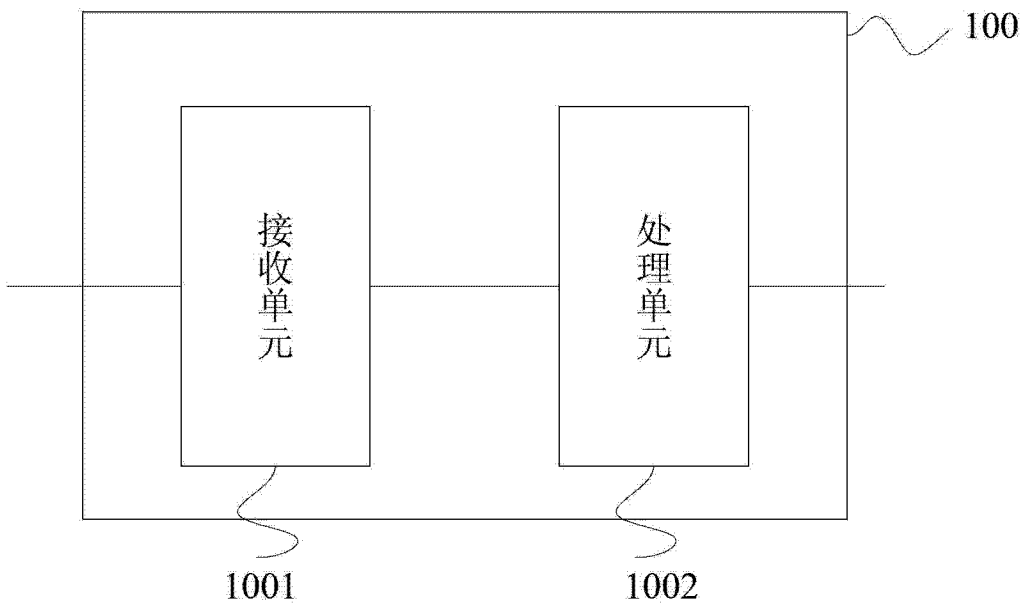


图 10

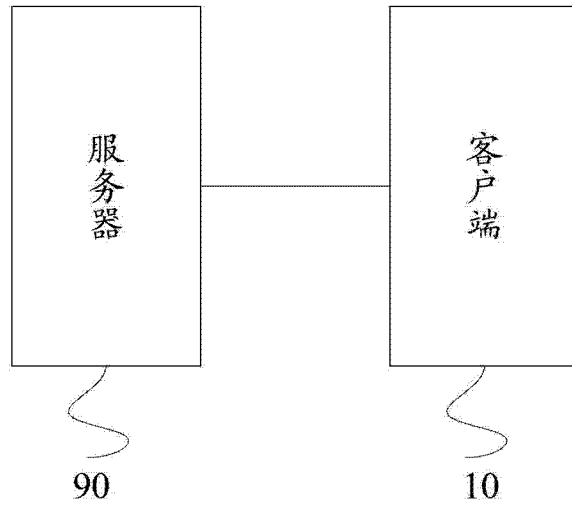


图 11