



(12) 发明专利

(10) 授权公告号 CN 112333524 B

(45) 授权公告日 2024. 02. 02

(21) 申请号 202011257196.4

(22) 申请日 2016.12.16

(65) 同一申请的已公布的文献号
申请公布号 CN 112333524 A

(43) 申请公布日 2021.02.05

(30) 优先权数据
62/268,410 2015.12.16 US

(62) 分案原申请数据
201680074415.0 2016.12.16

(73) 专利权人 六科股份有限公司
地址 美国加利福尼亚州

(72) 发明人 马库斯·K·克莱莫
彼得·东克尔 史蒂文·D·舍夫
大卫·亨利·皮尔斯
谢尔曼·凌风·李
威尔逊·哈尔隆
康斯坦丁诺斯·安东尼奥斯·迪米
特里乌
杜威·浩·李

(74) 专利代理机构 北京柏杉松知识产权代理事
务所(普通合伙) 11413
专利代理师 罗小晨 刘继富

(51) Int.Cl.

H04N 21/431 (2011.01)

H04N 21/432 (2011.01)

H04N 21/44 (2011.01)

H04N 21/458 (2011.01)

H04N 21/472 (2011.01)

H04N 21/81 (2011.01)

(56) 对比文件

CN 104219559 A, 2014.12.17

EP 2820815 A1, 2015.01.07

US 2004172478 A1, 2004.09.02

US 2012117584 A1, 2012.05.10

CN 103814569 A, 2014.05.21

US 2008127253 A1, 2008.05.29

US 2013031582 A1, 2013.01.31

US 2015040176 A1, 2015.02.05

US 2014196085 A1, 2014.07.10

US 2004189873 A1, 2004.09.30

US 2003149983 A1, 2003.08.07 (续)

审查员 齐欣

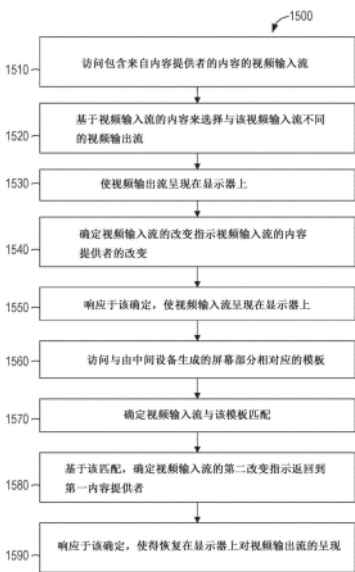
(54) 发明名称

动态视频覆盖

(57) 摘要

本发明涉及动态视频覆盖。客户端设备从中间设备访问视频输入流以进行显示。客户端设备分析视频输入流以确定视频输入流和与屏幕部分相对应的模板匹配。基于视频输入流与模板匹配,生成视频输出流并使其呈现在显示器上。在一些示例实施例中,在客户端设备正在经由中间设备替换从内容源接收的视频内容时执行分析。例如,通过机顶盒从国家内容提供者传输到智能电视的商业广告可以被有目标广告替换。在替换期间,可以检测由机顶盒产生的菜单,并且由智能电视改变替换视频以包括菜单。

权利要求书4页 说明书28页 附图15页



CN 112333524 B

[接上页]

(56) 对比文件

卢选民,史浩山,张原.多媒体监控网络视频

多画面全动态显示的设计与实现.小型微型计算机系统.2000,(第09期),全文.

1. 一种动态提供视频覆盖的方法,所述方法包括:

通过一个或多个处理器访问包括来自第一内容提供者的第一视频内容的视频输入流;

基于所述视频输入流的第一视频内容,选择视频输出流的替换视频内容,其中,所述视频输出流的替换视频内容不同于所述视频输入流的第一视频内容,其中,所述替换视频内容是响应于视频识别系统从客户端设备的查询指纹生成器接收查询指纹并且所述视频识别系统识别到查询指纹与由参考指纹生成器生成的至少一个参考指纹之间的匹配来选择的,其中,查询指纹是基于所述第一视频内容的帧来生成的,并且其中,所述视频识别系统和所述参考指纹生成器相对于所述客户端设备是远程的;

使所述视频输出流的替换视频内容呈现在所述客户端设备的显示器上;

当显示器正在呈现所述视频输出流的替换视频内容时,通过所述一个或多个处理器确定所述视频输入流已经发生第一改变,其中,所述第一改变指示已经发生从所述视频输入流的第一内容提供者到第二内容提供者的改变,其中,所述视频输入流包括在第一改变已经发生后来自第二内容提供者的第二视频内容;

响应于确定已经发生从所述第一内容提供者到第二内容提供者的改变,使所述视频输入流的第二视频内容呈现在所述显示器上并且使所述替换视频内容的呈现停止;

在使所述视频输入流的第二视频内容呈现在所述显示器上之后:

从数据库访问与针对由中间设备生成的第三视频内容的显示器的屏幕部分相对应的模板;

当视频输入流包括来自第一内容提供者的第一视频内容和由中间设备生成的第三视频内容时,通过所述一个或多个处理器分析所述视频输入流以确定所述视频输入流的帧的第三视频内容的至少一部分与所述模板匹配,其中,所述第三视频内容配置为呈现在显示器上以指示第一内容提供者的标识;

基于所述视频输入流的所述帧的第三视频内容的所述至少一部分与所述模板匹配,确定所述视频输入流的第二改变指示返回到所述视频输入流的第一内容提供者;以及

响应于确定已经返回到所述视频输入流的第一内容提供者:

修改视频输出流以包括替换视频内容和第三视频内容,以及

恢复替换视频内容在显示器上的呈现,其中,第三视频内容在所述显示器的屏幕部分被呈现为在替换视频内容上的视频覆盖。

2. 根据权利要求1所述的方法,其中:

选择所述视频输出流包括发送请求到替换内容服务器,所述请求包括要被替换的第一视频内容的指示。

3. 根据权利要求2所述的方法,其中发送请求包括:

通过所述一个或多个处理器生成所述视频输入流的指纹;以及

将所述视频输入流的指纹与要被替换的内容的指纹进行比较。

4. 根据权利要求1所述的方法,其中确定所述视频输入流的改变指示从视频输入流的第一内容提供者到第二内容提供者的改变包括:

根据所述视频输入流生成指纹序列;以及

将所述指纹序列的连续指纹与要被替换的内容的连续指纹进行比较。

5. 根据权利要求2所述的方法, 其中经由第一通信网络接收所述视频输入流, 并且经由第二通信网络接收所述替换视频内容。

6. 根据权利要求1所述的方法, 其中所述第一内容提供者是有线电视台。

7. 根据权利要求1所述的方法, 其中, 分析所述视频输入流以确定所述视频输入流的帧的第三视频内容的至少一部分与所述模板匹配包括:

对所述视频输入流的所述帧进行下采样;

确定下采样的帧与所述模板之间的互相关; 以及

将所述互相关与阈值进行比较。

8. 根据权利要求1所述的方法, 其中, 所述中间设备是机顶盒,

其中, 访问视频输入流包括通过一个或更多个处理器从所述机顶盒接收视频输入流, 以及

其中, 所述第三视频内容配置为响应于视频输入流中从第二内容提供者到第一内容提供者的改变而在显示器上呈现以指示第一内容提供者的标识。

9. 根据权利要求1所述的方法, 其中, 当视频输入流包括来自第一内容提供者的第一视频内容和由中间设备生成的第三视频内容时, 所述第一视频内容配置为在显示器的第一屏幕部分显示, 并且所述第三视频内容配置为在显示器的第二屏幕部分显示, 以及

其中, 修改视频输出流以包括替换视频内容和第三视频内容, 包括: 修改视频输出流以在显示器的第一屏幕部分包括替换视频内容, 并在显示器的第二屏幕部分包括第三视频内容。

10. 一种动态提供视频覆盖的系统, 所述系统包括:

存储器, 其存储指令; 以及

一个或更多个处理器, 其由所述指令配置为执行包括操作, 所述操作包括:

访问包括来自第一内容提供者的第一视频内容的视频输入流;

基于所述视频输入流的第一视频内容, 选择视频输出流的替换视频内容, 其中, 所述视频输出流的替换视频内容不同于所述视频输入流的第一视频内容, 其中, 所述替换视频内容是响应于视频识别系统从客户端设备的查询指纹生成器接收查询指纹并且所述视频识别系统识别到查询指纹与由参考指纹生成器生成的至少一个参考指纹之间的匹配来选择的, 其中, 查询指纹是基于所述第一视频内容的帧来生成的, 并且其中, 所述视频识别系统和所述参考指纹生成器相对于所述客户端设备是远程的;

使所述视频输出流的替换视频内容呈现在所述客户端设备的显示器上;

当显示器正在呈现所述视频输出流的替换视频内容时, 确定所述视频输入流已经发生第一改变, 其中, 所述第一改变指示已经发生从所述视频输入流的第一内容提供者到第二内容提供者的改变, 其中, 所述视频输入流包括在第一改变已经发生后来自第二内容提供者的第二视频内容;

响应于确定已经发生从所述第一内容提供者到第二内容提供者的改变, 使所述视频输入流的第二视频内容呈现在所述显示器上并且使所述替换视频内容的呈现停止;

在使所述视频输入流的第二视频内容呈现在所述显示器上之后:

从数据库访问与针对由中间设备生成的第三视频内容的显示器的屏幕部分相对应的模板;

当视频输入流包括来自第一内容提供者的第一视频内容和由中间设备生成的第三视频内容时,分析所述视频输入流以确定所述视频输入流的帧的第三视频内容的至少一部分与所述模板匹配,其中,所述第三视频内容配置为呈现在显示器上以指示第一内容提供者的标识;

基于所述视频输入流的所述帧的第三视频内容的所述至少一部分与所述模板匹配,确定所述视频输入流的第二改变指示返回到所述视频输入流的第一内容提供者;以及

响应于确定已经返回到所述视频输入流的第一内容提供者:

修改视频输出流以包括替换视频内容和第三视频内容,以及

恢复替换视频内容在显示器上的呈现,其中,第三视频内容在所述显示器的屏幕部分被呈现为在替换视频内容上的视频覆盖。

11. 根据权利要求10所述的系统,其中:

选择所述视频输出流包括发送请求到替换内容服务器,所述请求包括要被替换的第一视频内容的指示。

12. 根据权利要求11所述的系统,其中发送请求包括:

生成所述视频输入流的指纹;以及

将所述视频输入流的指纹与要被替换的内容的指纹进行比较。

13. 根据权利要求10所述的系统,其中确定所述视频输入流的改变指示从所述视频输入流的第一内容提供者到第二内容提供者的改变包括:

根据所述视频输入流生成指纹序列;以及

将所述指纹序列的连续指纹与要被替换的内容的连续指纹进行比较。

14. 根据权利要求11所述的系统,其中经由第一通信网络接收所述视频输入流,并且经由第二通信网络接收所述替换视频内容。

15. 根据权利要求10所述的系统,其中所述第一内容提供者是有线电视台。

16. 根据权利要求10所述的系统,其中,分析视频输入流以确定所述视频输入流的帧的第三视频内容的至少一部分与所述模板匹配包括:

对所述视频输入流的所述帧进行下采样;

确定下采样的帧与所述模板之间的互相关;以及

将所述互相关与阈值进行比较。

17. 一种存储指令的非暂时性机器可读介质,所述指令在由一个或更多个处理器执行时使所述一个或更多个处理器执行包括操作,所述操作包括:

访问包括来自第一内容提供者的第一视频内容的视频输入流;

基于所述视频输入流的第一视频内容,选择视频输出流的替换视频内容,其中,所述视频输出流的替换视频内容不同于所述视频输入流的第一视频内容,其中,所述替换视频内容是响应于视频识别系统从客户端设备的查询指纹生成器接收查询指纹并且所述视频识别系统识别到查询指纹与由参考指纹生成器生成的至少一个参考指纹之间的匹配来选择的,其中,查询指纹是基于所述第一视频内容的帧来生成的,并且其中,所述视频识别系统和所述参考指纹生成器相对于所述客户端设备是远程的;

使所述视频输出流的替换视频内容呈现在所述客户端设备的显示器上;

当显示器正在呈现所述视频输出流的替换视频内容时,确定所述视频输入流已经发生

第一改变,其中,所述第一改变指示已经发生从所述视频输入流的第一内容提供者到第二内容提供者的改变,其中,所述视频输入流包括在第一改变已经发生后来自第二内容提供者的第二视频内容;

响应于确定已经发生从所述第一内容提供者到第二内容提供者的改变,使所述视频输入流的第二视频内容呈现在所述显示器上并且使所述替换视频内容的呈现停止;

在使所述视频输入流的第二视频内容呈现在所述显示器上之后:

从数据库访问与针对由中间设备生成的第三视频内容的显示器的屏幕部分相对应的模板;

当视频输入流包括来自第一内容提供者的第一视频内容和由中间设备生成的第三视频内容时,分析所述视频输入流以确定所述视频输入流的帧的第三视频内容的至少一部分与所述模板匹配,其中,所述第三视频内容配置为呈现在显示器上以指示第一内容提供者的标识;

基于所述视频输入流的所述帧的第三视频内容的所述至少一部分与所述模板匹配,确定所述视频输入流的第二改变指示返回到所述视频输入流的第一内容提供者;以及

响应于确定已经返回到所述视频输入流的第一内容提供者:

修改视频输出流以包括替换视频内容和第三视频内容,以及

恢复替换视频内容在显示器上的呈现,其中,第三视频内容在所述显示器的屏幕部分被呈现为在替换视频内容上的视频覆盖。

18. 根据权利要求17所述的非暂时性机器可读介质,其中:

选择所述视频输出流包括发送请求到替换内容服务器,所述请求包括要被替换的第一视频内容的指示。

19. 根据权利要求18所述的非暂时性机器可读介质,其中发送请求包括:

生成所述视频输入流的指纹;以及

将所述视频输入流的指纹与要被替换的内容的指纹进行比较。

20. 根据权利要求17所述的非暂时性机器可读介质,其中确定所述视频输入流的改变指示从所述视频输入流的第一内容提供者到第二内容提供者的改变包括:

根据所述视频输入流生成指纹序列;以及

将所述指纹序列的连续指纹与要被替换的内容的连续指纹进行比较。

动态视频覆盖

[0001] 本申请是分案申请,其原案申请是申请号为PCT/US2016/067250、申请日为2016年12月16日的PCT申请并且于2018年6月19日进入中国国家阶段,申请号为201680074415.0,名称为“动态视频覆盖”。

[0002] 本申请要求于2015年12月16日提交的题为“动态视频覆盖”的美国临时专利申请62/268,410的优先权,其全部内容通过引用并入本文。

技术领域

[0003] 本文公开的主题总体上涉及视频流的处理。具体而言,本公开涉及呈现覆盖视频的系统和方法。

背景技术

[0004] 通常,人们通过接收来自内容源的传输的设备观看诸如电视节目、广告、电影、视频剪辑等的视频内容。例如,广播公司(例如,HBO[®]或CNN[®])、网络服务器(例如YouTube[®])、对等源(例如另一个设备)或其他内容源将视频内容流送或以其他方式传输到能够呈现视频内容的各种设备,例如电视机和相关联的机顶盒,计算和/或移动设备以及相关联的媒体播放器或浏览器等等。

附图说明

[0005] 一些实施例在附图中以示例而非限制的方式进行说明。

[0006] 图1是描绘示例实施例中的用于动态地覆盖视频的网络环境的框图。

[0007] 图2是示出根据一些示例实施例的视频识别系统和查询指纹生成器的组件的框图。

[0008] 图3是示出根据一些示例实施例的客户端设备和替换内容服务器的组件的框图。

[0009] 图4是示例实施例中的示出了动态视频覆盖的显示图。

[0010] 图5是示例实施例中的示出了动态视频覆盖的显示图。

[0011] 图6是示例实施例中的示出了动态视频覆盖的显示图。

[0012] 图7是示例实施例中的示出了动态视频覆盖的显示图。

[0013] 图8是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0014] 图9是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0015] 图10是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0016] 图11是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0017] 图12是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0018] 图13是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0019] 图14是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0020] 图15是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。

[0021] 图16是示出了一些示例实施例中的用于动态地提供视频覆盖的数据结构的框图。

[0022] 图17是示出根据一些示例实施例的机器的组件的框图,该机器能够从机器可读介质读取指令并且执行本文讨论的任何一种或更多种方法。

具体实施方式

[0023] 描述了用于动态地提供视频覆盖的示例方法和系统。在一些示例实施例中,客户端设备(例如,智能电视(TV))识别呈现覆盖在视频流上的信息的机顶盒。基于识别的机顶盒和从机顶盒接收到的视频数据,选择与所呈现的信息相对应的模板。基于该模板,将替换视频流与信息一起呈现,模仿机顶盒呈现原始视频流的方式。在一些示例实施例中,基于原始视频流来选择替换视频流。例如,客户端设备可以生成原始视频内容的帧的查询指纹。客户端设备使用所生成的查询指纹查询已知参考指纹的数据库,确定视频内容的帧的查询指纹与已知参考指纹匹配,并且基于指纹的匹配来识别视频内容。基于所识别的视频内容,选择替换视频流。例如,可以选择示出特定本地商业广告的替换视频流来替换示出特定国家商业广告的原始视频流。

[0024] 指纹可以是单帧指纹或多帧指纹。多帧指纹是单帧指纹以及帧的相对时序的集合。术语“指纹”用于指代单帧和多帧指纹。术语“组件”用于指代多帧指纹内的单帧指纹,否则可能导致混淆。

[0025] 在一些示例实施例中,客户端设备基于所识别的模板在特定位置以特定尺寸呈现替换视频流。例如,机顶盒可能将原始视频流呈现为缩小尺寸的画中画。因此,为了模仿机顶盒的行为,替换视频流在与原始视频流相同的位置处以与原始视频流相同的尺寸被呈现。

[0026] 在进一步的示例实施例中,客户端设备基于所识别的模板呈现替换视频流的特定部分。例如,机顶盒可能将原始视频流呈现为全屏图像,但用信息栏覆盖流的底部。因此,为了模仿机顶盒的行为,仅呈现替换视频流的上部。

[0027] 在另外的示例实施例中,客户端设备基于所识别的模板来执行替换视频流的特定部分的阿尔法(alpha)(透明度)混合。例如,机顶盒可能呈现叠加在原始视频流上的菜单。客户端设备可以从合成图像提取菜单并使用在模板中定义的阿尔法值将菜单叠加在替换视频流上。

[0028] 在一些示例实施例中,客户端设备基于所识别的模板以特定速度呈现替换视频流。例如,机顶盒可能以快进模式呈现原始视频流。因此,为了模仿机顶盒的行为,替换视频流以快进模式呈现。

[0029] 在以下描述中,出于解释的目的,阐述了许多具体细节以提供对示例实施例的透彻理解。然而,对于本领域技术人员而言显而易见的是,可以在没有这些具体细节的情况下实践本主题。

[0030] 图1是示出根据一些示例实施例的适用于动态视频覆盖的网络环境100的网络图。网络环境100可以包括观看站120,该观看站120从诸如广播公司、网络服务器、有线电视台等的内容源110接收视频和其他多媒体内容。例如,内容源110可以是诸如电视台或电视网络之类的广播公司,其通过电视频道将媒体流送或传输到观看站120;和/或诸如网站之类的网络服务,其通过网络170将媒体流送或传输到观看站120,等等。观看站120包括生成从内容源110接收的视频内容的参考指纹的参考指纹生成器130。

[0031] 一个或多个机顶盒140还可以例如经由广播信道和/或通过网络170从内容源110接收视频和其他多媒体内容。机顶盒140可以在将内容发送给客户端设备150之前修改所接收的内容。客户端设备150是能够接收和呈现视频流和/或其它多媒体内容流的任何设备(例如,电视、第二机顶盒、膝上型电脑或其他个人计算机(PC)、平板电脑或其他移动设备、数字录像机(DVR)或游戏设备)。

[0032] 在一些示例实施例中,机顶盒140包括调谐器,调谐器被配置为接收视频内容的输入流并且通过处理输入流来生成视频内容的输出流。机顶盒140可以是配备有调谐器、解码器等等的设备,以通过视频内容分发网络(例如,由多频道视频节目分发商采用的地面广播、电缆和/或卫星广播网络)访问视频内容。另外地或可选地,机顶盒140可以是配备有网络适配器和解码器等等的设备,以通过由用于访问互联网视频流服务的设备采用的广域网(例如,互联网)来访问视频内容。视频内容的输出流可以由多个分量组成。例如,视频内容的输出流可以包括输出流的第一区域中的对应于输入流的第一分量和输出流的第二区域中的对应于由机顶盒140生成的视频数据的第二分量。第二分量可以包括用于机顶盒140的菜单、来自机顶盒140的信息消息、或由其他设备(例如,电话或安全警报器)生成并且被发送给机顶盒140用于呈现的信息消息。机顶盒140输出可由客户端设备150的显示器使用的信号(例如,数字或模拟信号)以将输出视频内容呈现给与客户端设备150相关联的用户。从客户端设备150的角度来看,机顶盒140输出的信号是视频输入流。

[0033] 客户端设备150还可以包括被配置为显示经处理的视频内容流的显示器或其他用户界面。显示器可以是平板屏幕、等离子屏幕、发光二极管(LED)屏幕、阴极射线管(CRT)、液晶显示器(LCD)、投影仪等。在一些示例实施例中,机顶盒140和客户端设备150被集成到单个设备中。

[0034] 网络170可以是能够实现设备之间通信的任何网络,例如有线网络、无线网络(例如,移动网络)等。网络170可以包括构成专用网络(例如,有线电视网络或卫星电视网络)、公共网络(例如,无线广播信道或因特网)等的一个或多个部分。

[0035] 在一些示例实施例中,视频识别系统190通过网络170与观看站120和客户端设备150通信。视频识别系统190可以接收由客户端设备150的查询指纹生成器160生成的查询指纹,例如视频内容内的帧或帧块的指纹,并且查询由观看站120的参考指纹生成器130生成的已知参考指纹的索引,以便通过将查询指纹与一个或多个参考指纹进行匹配来识别视频内容。

[0036] 在识别视频内容时,视频识别系统190可以将与视频内容相关联的替换内容(例如,替换节目,替换商业广告等)的标识符返回给客户端设备150。替换内容可以被存储在替换内容服务器180中。使用该标识符,客户端设备150可以访问来自替换内容服务器180的替换内容并且用从机顶盒140接收到的视频内容覆盖该替换内容。例如,客户端设备150可以访问和呈现来自替换内容服务器180的替换内容,例如广播信道的替换广告。

[0037] 图1中所示的任何机器、数据库或设备可以在由软件修改(例如,配置或编程)为专用计算机的通用计算机中实现,以执行本文描述的用于该机器的功能。例如,以下参考图16讨论能够实现本文描述的方法中的任何一个或多个的计算机系统。如本文所使用的,“数据库”是数据存储资源并且可以存储被构建为文本文件、表格、电子表格、关系数据库、文档存储、键-值存储、三元组存储或其任何合适组合的数据。此外,图1中所示的任何两个或更

多个机器可以组合成单个机器,并且本文中针对任何单个机器描述的功能可以细分在多个机器中。

[0038] 此外,任何模块、系统和/或发生器可以位于图1所示的任何机器、数据库或设备中。例如,视频识别系统190可以包括查询指纹生成器160和来自客户端设备150的视频内容的帧,并且可以使用所包括的查询指纹生成器160以及其他配置来生成查询指纹。

[0039] 图2是示出根据一些示例实施例的查询指纹生成器160、参考指纹生成器130和视频识别系统190的组件的框图200。

[0040] 客户端设备150的查询指纹生成器160包括补丁(patch)选择模块210和值计算模块220,它们被配置为彼此通信(例如,经由总线、共享存储器或交换机)。视频识别系统190包括索引模块230、指纹匹配模块240和识别模块250,它们被配置为彼此通信(例如,经由总线、共享存储器或交换机)。观看站120的参考指纹生成器130包括补丁选择模块260和值计算模块270,它们被配置为彼此通信(例如,经由总线、共享存储器或交换机)。这里描述的一个或更多个模块可以使用硬件(例如,机器的处理器,现场可编程门阵列(FPGA)或专用集成电路(ASIC))或者硬件和软件的组合(例如,由软件配置的处理器)来实现。而且,这些模块中的任何两个或更多个可以组合成单个模块,并且本文中针对单个模块描述的功能可以细分在多个模块中。

[0041] 在一些示例实施例中,查询指纹生成器160被配置和/或编程为生成在客户端设备150处捕获的视频内容的一个或更多个帧的查询指纹。例如,查询指纹生成器160可以计算补丁的值,诸如视频内容内的一个或更多个帧的类Haar特征、区域、部分和/或其他方面。例如,补丁可以是具有各种不同几何形状、类Haar特征等的帧的一部分。在一些示例实施例中,一些或所有捕获的补丁中的每个可以具有不同的尺度(scale)并且位于帧内的不同位置处。通过报告接收到的视频内容的补丁的指纹,查询指纹生成器160可以确定每个补丁的内容。因此,如果接收到的视频包括来自不同源的多个分量(例如,来自广播公司的视频输入流和机顶盒140生成的覆盖视频流),则可以识别与多个分量相对应的区域。

[0042] 查询指纹生成器160(和参考指纹生成器130)生成和/或创建用于从视频内容中的帧中识别该视频内容的指纹。通常,由客户端设备150接收的视频内容将采用不同的格式和采样率,并且,针对视频内容的帧中的一些或全部,查询指纹生成器160为每个帧创建查询指纹,这是尺度无关的并对不同的压缩失真具有鲁棒性。在一些示例实施例中,查询指纹生成器160可以组合每个帧的查询指纹以生成视频内容的帧块(例如,多个帧)的查询指纹。视频识别系统190可以响应来自数百万查询指纹发生器160的视频识别请求。

[0043] 查询指纹生成器160可以包括补丁选择模块210,其被配置和/或编程为选择视频内容的多个补丁,例如与视频内容内的一个或更多个帧的显示区域相关联的补丁。类似地,参考指纹生成器130可以包括补丁选择模块260,其被配置和/或编程为选择视频内容的多个补丁,例如与视频内容内的一个或更多个帧的显示区域相关联的补丁。

[0044] 可以通过将帧分成网格(例如,2x2网格,4x3网格或4x4网格)来创建补丁。补丁可以重叠。例如,可以使用20个补丁:对应于帧的象限的四个大补丁,以及将该帧分割成4x4网格所得到的16个小补丁。作为另一个例子,可以使用五个补丁:对应于帧的象限的四个大补丁,位于该帧的中心处的同等尺寸的第五个补丁。在一些示例中,可以通过将帧的一部分(例如不包括帧的外边界的帧的中心部分)划分为网格来创建补丁。因此,这些补丁的累积

范围可以小于整个帧。

[0045] 查询指纹生成器160还可以包括值计算模块220,其被配置或编程为使用积分图像技术来计算所选多个补丁中的每个的值,所述积分图像技术例如是使用生成区域的矩形区域中的值的总和的求和区域表格或其他数据结构来计算值的技术。类似地,参考指纹生成器130可以包括值计算模块270,其被配置或编程为使用积分图像技术来计算所选多个补丁中的每个的值,所述积分图像技术例如是使用生成区域的矩形区域中的值的总和的求和区域表格或其他数据结构来计算值的技术。

[0046] 例如,补丁选择模块210、260可以选择一个或更多个帧的区域的补丁,例如在对象检测中常用的类Haar特征。值计算模块220、270可以利用类Haar特征为帧中的对象(例如帧的视觉图像中的对象)生成或计算相同值,而不管对象的相对尺寸如何。例如,值计算模块220、270可以通过使用框形滤波(例如,帧的区域的平均值)近似高斯滤波器及其衍生物来生成值,其中高斯滤波器的衍生物是通过找出框形滤波中的差异而创建的。

[0047] 经由值计算模块220,查询指纹生成器160可以通过计算不同尺度的和位于视频内容的帧的显示区域的不同位置处的类Haar特征或补丁的值来生成查询指纹。经由值计算模块270,参考指纹生成器130可以通过计算不同尺度的和位于视频内容的帧的显示区域的不同位置处的类Haar特征或补丁的值来生成参考指纹。

[0048] 在一些示例中,补丁选择模块210、260可以从划分一个或更多个帧的区域的网格中选择补丁。然后,值计算模块220、270可以计算每个补丁的特征值,例如每个补片内的像素值的总和值、中值等。而且,值计算模块220、270可以计算这些特征值之间的差和/或这些特征值的线性组合。一些情况下,所得到的差和/或线性组合然后可以确定二进制值(例如,0或1),例如通过与阈值进行比较。因此,经由计算模块220、260,指纹生成器160、130可以通过计算基于视频内容的一个或更多个帧的多个补丁的特征值的差和/或线性组合的二进制值来生成指纹。

[0049] 视频识别系统190包括索引模块230、指纹匹配模块240和识别模块250以及其他模块,它们被配置和/或编程为将查询指纹与已知参考指纹进行匹配。

[0050] 在一些示例实施例中,索引模块230被配置和/或编程为查询在诸如观看站120的参考设备处捕获的视频内容的已知参考指纹的数据库。例如,索引模块230可以查询存储在观看站120的数据库内的索引、存储在视频识别系统190内的索引等等。

[0051] 例如,索引模块230可以被配置为查询已知参考指纹的量化补丁值的索引。索引模块230可以查询与参考指纹的单个帧相关联的1位、8位、16位和/或24位数的索引。索引模块230可以通过量化参考指纹的一个或更多个补丁值来得到以上数。例如,索引模块230可以通过它们的绝对和、通过对数归一化等来对指纹值进行归一化。

[0052] 索引模块230可以使用帧的较大区域的最佳或较好相关值来对参考指纹的一些或全部帧做索引。索引模块230可以使用全帧补丁的高度相关值来创建索引,因为与最佳相关值相关联的特征可以表示帧的剩余补丁等等。高度相关值可以使用样本数据集预先确定。

[0053] 例如,当帧的三个区域是最佳相关值时,索引模块230将三个值中的每一个量化为8位,例如通过将它们放置在具有最大和最小限制的均匀间隔的直方图中,从而创建24位数。索引模块230然后可以利用24位数的参考指纹索引来快速查找和/或识别帧。在一些示例实施例中,执行模糊搜索。模糊搜索可以找到不精确的匹配。例如,当匹配值与对应值中

的一个完全匹配或在对应值中的一个内时,可以找到三个8位值的模糊匹配。在这个示例中,当尝试匹配指纹时,每个索引有27种排列供解析,三个字节中的每个有三个匹配值。类似地,当将四个8位值用于具有相同精度的模糊匹配(即,每个值的最大距离为1)时,每个索引有81种排列。更高的精度和/或更多的值增加了可能的排列的数量。

[0054] 作为另一示例,索引模块230可以将三个值中的每一个量化为1位,诸如通过将每个值与阈值进行比较,从而生成3位数。索引模块230然后可以利用3位数的参考指纹索引来快速查找和/或识别帧,因为当尝试匹配指纹时,每个索引只有8个值要比较。

[0055] 在另一个示例实施例中,使用32个区域并且每个区域由单个位表示。可以将32位值散列化以生成密钥。例如,如果使用256个散列值,则32位值的前8位可以用作散列。为了查找指纹,可以迭代并比较具有相同散列值的潜在匹配指纹。

[0056] 在一些示例实施例中,指纹匹配模块240被配置和/或编程为确定查询指纹与至少一个已知参考指纹匹配。例如,指纹匹配模块240可以通过确定查询指纹与至少一个已知参考指纹之间的相似度满足与Tanimoto距离测量、Manhattan距离测量和/或与匹配图像或其他基于视觉的内容相关联的其他距离测量相关联的预定阈值,来确定查询指纹与至少一个已知参考指纹匹配。

[0057] 例如,指纹匹配模块240可以使用Tanimoto或Manhattan距离测量来将查询指纹与一个或多个参考指纹进行比较,并且当比较表明距离测量满足预定阈值(例如,在一定的距离阈值内)时,确定指纹匹配。当然,指纹匹配模块240可以利用其他匹配技术来确定查询指纹是否与参考指纹匹配,例如欧几里得、余弦、KL散度和/或Itakura距离测量技术以及其它距离测量技术。

[0058] 在一些示例实施例中,当将查询指纹与参考指纹进行匹配时,视频识别系统190可以利用指纹的各种不同的块尺寸(例如,帧数或帧率)。例如,参考指纹可以被设置为5fps,并且在给定指纹的8位值(32字节/帧)的情况下占据大约560KB/h的运行时间,并且可以包括偏移误差的查询指纹可以被设置为15fps或更高。在该示例中,索引的查询可涉及查询参考指纹索引的多个参考帧(例如,参考指纹的三个帧)。作为另一示例,参考指纹可以被设置为15fps,而查询指纹被设置为5fps。在该示例中,将查询指纹与参考指纹进行比较可能涉及将每个查询指纹帧与参考指纹索引的多个参考帧进行比较。

[0059] 因此,在一些示例实施例中,视频识别系统190可以通过修改指纹的块尺寸来优化将查询指纹与参考指纹进行匹配的匹配率。例如,指纹匹配模块240可以将查询指纹与1秒的参考视频内容、0.5秒的参考视频内容、一分钟的参考视频内容等进行匹配。在一些情况下,匹配的准确时间可能无法确定。例如,如果视频包括静止图像,则显示静止图像的视频的任何部分都将与静止图像片段的整个长度进行匹配。

[0060] 如上所述,一些补丁可能具有与其他补丁不同的尺度。大补丁在识别匹配方面可能比小补丁更重要。例如,在所有位均等对待的情况下,使用1位值,可以基于低于阈值的不匹配位的数量来识别匹配。作为另一个示例,可以基于低于第一阈值的不匹配的低位数量和低于第二阈值的不匹配的高位的数量来识别匹配。第二个阈值可以为零。

[0061] 在一些示例实施例中,识别模块250被配置和/或编程为基于确定查询指纹匹配至少一个参考指纹来识别在客户端设备150处捕获的视频内容。例如,识别模块250可以识别视频内容的名称或标题、当前由客户端设备150呈现的视频内容内的位置、提供视频内容的

频道或广播公司等。

[0062] 图3是示出根据一些示例实施例的客户端设备150和替换内容服务器180的组件的框图300。客户端设备150包括被配置为彼此通信的机顶盒标识符模块310、视频源标识符模块320和视频覆盖模块330。替换内容服务器180包括视频提供模块340和模板提供模块350。

[0063] 机顶盒标识符模块310识别连接到客户端设备150的机顶盒140。例如,当机顶盒140和客户端设备150通过高清晰度多媒体接口(HDMI)电缆连接时,机顶盒140的扩展显示识别数据(EDID)可用于识别机顶盒140。

[0064] 作为另一示例,机顶盒标识符模块310可以在客户端设备150上呈现用户界面,用户可以通过该用户界面来识别机顶盒140。这可以通过让用户通过从下拉列表中选择品牌和型号或者将信息输入到文本框来手动识别机顶盒140而完成。用户交互式识别还可以通过指示用户激活机顶盒140的特定菜单选项来完成。客户端设备150访问来自不同机顶盒的已知菜单的数据库,并将其与在用户激活机顶盒140的菜单之后接收到的视频数据进行比较。当找到匹配时,机顶盒140就被识别了。在一些示例实施例中,需要对机顶盒140的多个用户动作来唯一地识别机顶盒140。例如,第一菜单可以用于将可能的匹配减少80%,第二菜单可以用于将可能的匹配进一步减少80%(即,减小到原始可能的匹配集的4%),并且可能需要第三菜单来唯一地识别机顶盒140。

[0065] 视频源标识符模块320识别提供给客户端设备150的视频源。在一些示例实施例中,视频源标识符模块320由查询指纹生成器160实现。

[0066] 视频覆盖模块330基于由机顶盒标识符模块310识别的特定机顶盒来将从替换内容服务器180接收到的视频数据与从机顶盒140接收到的视频数据进行组合。例如,视频提供模块340可以基于视频源标识符模块320对本地视频源的识别来提供本地商业广告。本地商业广告可以显示为覆盖从机顶盒140接收到的视频数据的一部分的画中画。可以基于机顶盒标识符模块310对机顶盒140的识别来确定要覆盖的部分。

[0067] 替换内容服务器180的视频提供模块340基于所识别的视频源向客户端设备150提供替换(或替代)视频内容。例如,视频提供模块340可以访问将针对多个地点的本地商业广告映射到全国的特定广播公司的国家商业广告的数据库。以这种方式,取决于向机顶盒140提供信号的本地广播公司,国家广播公司的特定国家商业广告可能会被不同的本地广告替换。

[0068] 替换内容服务器180的模板提供模块350基于所识别的机顶盒向客户端设备150提供模板。例如,一家制造商的机顶盒与另一家制造商的机顶盒可能有不同的菜单模板集。同样,来自同一制造商的不同型号可能有不同的菜单模板集。此外,同一型号的不同固件版本可能有不同的菜单模板集。相应地,在替换内容服务器180上维护将机顶盒标识符映射到模板集的数据库,并且根据请求全部或部分地将该数据库发送到客户端设备150。

[0069] 模板可以手动生成或以编程方式生成。对于手动生成,一个人与客户端设备和机顶盒一起工作,并且通过操纵机顶盒,使得呈现可以由机顶盒生成的不同菜单中的每个。通过分析客户端设备的屏幕上的菜单呈现,这个人为每个不同的菜单格式生成模板。模板存储在替换内容服务器180的数据库中以供以后访问。

[0070] 对于编程生成模板,在使用时监视(例如,通过客户端设备向替换内容服务器180报告状态)一个或更多个客户端设备(例如,客户端设备150)和对应的机顶盒(例如,机顶盒

140)。在一些示例实施例中,客户端设备或替换内容服务器180使用对区域(例如,屏幕区域的象限或十六分之一)的指纹识别确定从机顶盒接收的视频何时包含一部分已知视频数据(例如,识别出的有线节目)和一部分未知视频数据。未知视频数据被假定为机顶盒提供的附加视频内容。相应地,替换内容服务器180对接收到的视频数据执行像素级分析,以确定已知视频数据占用的区域、未知视频数据占据的区域,以及已知和未知视频数据的混合(例如,通过阿尔法混合)占据的区域。从区域生成模板,并且如果模板尚未存储在数据库中,则将其存储在替换内容服务器180的数据库中以供以后访问。

[0071] 在一些示例实施例中,模板的编程生成涉及监视由客户端设备从机顶盒接收到的视频数据的多个帧。通过比较视频数据的连续帧,可以将视频数据快速变化的区域与静态或经历相对很少变化的区域区分开。从这个分析可以推断出,快速变化的区域正在显示视频节目,而静态区域正在显示机顶盒提供的信息。类似地,快速变化但平均颜色与视频数据的其余部分的平均颜色明显不同的区域可以指示正在发生动态视频与静态覆盖的混合。因此,混合区域也可以通过多帧分析来识别。从区域生成模板,并且如果该模板尚未存储在数据库中,则将其存储在替换内容服务器180的数据库中以供以后访问。

[0072] 图4是示例实施例中的显示图400,示出了包括客户端设备150的显示区域中的区域410和区域420的动态视频覆盖。使从机顶盒140接收到的视频通过以显示在区域410中。替换视频数据在区域420中呈现。如在图4的示例中可以看出的,区域420是比客户端设备150的显示区域小的矩形区域。区域420与显示区域也没有共同的边缘。在其他示例实施例中,区域420具有其他形状、尺寸和位置。在一些示例实施例中,机顶盒的菜单占据区域410,并且从内容源110接收到的原始视频数据由机顶盒140缩放(例如缩小)和重新定位,并从机顶盒140发送给客户端设备150(在区域420中)。相应地,当替换视频数据被呈现在区域420中时,从内容源110接收到的视频数据被替换,而区域410中所示的菜单对用户保持可见。

[0073] 图5是示例实施例中的显示图500,示出了包括客户端设备150的显示区域中的区域510和区域520的动态视频覆盖。使从机顶盒140接收到的视频通过以显示在区域510中。从替换内容服务器180接收到的替换视频数据被裁剪到并呈现在区域520中。在一些示例实施例中,机顶盒140的菜单占据区域510,并且从内容源110接收到的原始视频数据的一部分被机顶盒140裁剪到区域520,并从机顶盒140发送到客户端设备150(在区域520中)。相应地,当替换视频数据的部分被呈现在区域520中时,从内容源110接收的视频数据被替换,而区域510中所示的菜单对用户保持可见。如图5所示,区域520的上边缘与显示区域的上边缘相同,区域520的左边缘和右边缘与显示区域的左边缘和右边缘重合,但区域520不占据显示器的整个高度。

[0074] 图6是示例实施例中的显示图600,示出了包括客户端设备150的显示区域中的区域610和区域620的动态视频覆盖。使从机顶盒140接收到的视频通过以显示在区域610中。从替换内容服务器180接收到的替换视频数据(不含有对应于区域610的部分)呈现在区域620中。在一些示例实施例中,机顶盒的菜单占据区域610,并且从内容源110接收到的原始视频数据的对应于区域620的部分从机顶盒140被发送到客户端设备150(在区域620中)。相应地,当替换视频数据的部分被呈现在区域620中时,从内容源110接收到的视频数据被替换,而区域610中所示的菜单对用户保持可见。

[0075] 图7是示例实施例中的显示图700,示出了包括客户端设备150的显示区域中的区

域710、区域720和区域730的动态视频覆盖。使从机顶盒140接收到的视频通过以显示在区域710中。从替换内容服务器180接收到的替换视频数据(不含有对应于区域710的部分)呈现在区域720中。从机顶盒140接收到的视频与替换视频数据混合并呈现在区域730中。例如,可以应用梯度,使得邻接区域710的区域730的部分完全(或几乎完全)由从机顶盒140接收到的视频组成,并且邻接区域720的区域730的部分完全(或几乎完全)由替换视频组成。

[0076] 在一些示例实施例中,在区域730中应用进一步的图像处理以反转机顶盒140应用的图像处理。例如,机顶盒140可以已经将由机顶盒140提供的信息与视频输入流混合。因此,该混合信息可以被认为是第三视频分量,其与对应于到机顶盒140的输入视频的第一视频分量和对应于由机顶盒140提供的覆盖信息的第二视频分量不同。为了完全从视频覆盖模块330的输出中消除视频输入流,对第三视频分量的区域(例如,区域730)应用进一步的滤波。

[0077] 在一些示例实施例中,所应用的滤波取决于亮度。超过预定亮度值的区域730的部分被确定为处于前景并且包含应该被传递给用户的信息。低于预定亮度值的区域730的部分被确定为处于背景中,并且因此应该过滤掉所包含的任何信息。在其他示例实施例中,为了滤除背景信息,每个背景像素的颜色被改变为所有背景像素的平均颜色。例如,叠加在足球比赛视频上的淡蓝色背景一旦平均,将保持淡蓝色,并且不传达关于足球比赛的任何信息。在其他示例实施例中,背景颜色被存储在模板中。因此,前景像素的颜色被保持,并且其余像素被替换为背景颜色。

[0078] 在进一步的示例实施例中,图像处理使用视频输入流反转由机顶盒140执行的图像处理。例如,通过识别视频输入流(例如,通过指纹识别),可以访问原始视频输入流(例如,从视频识别系统190)。通过应用与机顶盒140应用的相反的修改,恢复由机顶盒140覆盖的信息。例如,如果机顶盒140将占80%权重的覆盖信息和占20%权重的输入视频信息混合来生成输出视频,则客户端设备150可以通过减去20%权重的输入视频并将剩余部分乘以1.25来反转该过程。然后,通过应用与先前由机顶盒140应用的相同的混合来将恢复的覆盖应用于替换视频内容。在该示例中,这意味着通过利用80%权重的覆盖与20%权替换视频内容来生成用于显示的像素。

[0079] 图8是示出了一些示例实施例中的用于动态地提供视频覆盖的方法800的流程图。方法800包括操作810、820、830、840和850。作为示例而非限制,操作810-850被描述为由图1的组件110-190,图2的模块210-250以及图3的模块310-350执行。

[0080] 在操作810中,机顶盒标识符模块310识别向客户端设备(例如,客户端设备150)提供视频数据的机顶盒(例如,机顶盒140)。例如,当机顶盒140和客户端设备150通过HDMI电缆连接时,机顶盒140的EDID可用于识别机顶盒140。作为另一示例,机顶盒标识符模块310可以在客户端设备150上呈现用户界面,用户可以通过该界面来识别机顶盒140(例如,通过从选项列表中选择)。作为第三示例,机顶盒标识符模块310可以将其互联网协议(IP)地址发送到替换内容服务器180(或另一服务器)以供查找。在这个示例中,接收服务器维护数据库,该数据库将IP地址映射到有线电视公司的服务区域,并将有线电视公司映射到它们偏好的机顶盒。因此,客户端设备150的IP地址可以用于识别最有可能是机顶盒140的设备类型。

[0081] 一旦机顶盒140被机顶盒标识符模块310识别,机顶盒140的标识符被存储在客户

端设备150中以稍后访问。例如,可以在用户首次使用客户端设备150时提示用户识别机顶盒140,并且由于可以从存储器访问信息,因而在未来使用时不提示。

[0082] 视频源标识符模块320识别接收到的视频数据的一部分(操作820)。例如,接收到的视频数据可以包括覆盖有由机顶盒140生成的菜单的广播电视信号。使用补丁选择模块210和值计算模块220来生成所接收到的视频数据的一个或多个指纹,客户端设备150将指纹发送到视频识别系统190。作为响应,客户端设备150接收对应于广播电视信号的标识符,该标识符用于标识接收到的视频数据的包含广播电视信号的部分。

[0083] 可替换地,操作820可以通过分析特定模式的视频数据来执行,而无需指纹识别。例如,由机顶盒140或客户端设备150提供的特定视频覆盖可以在屏幕上的特定位置包括品牌名称。相应地,屏幕的该部分可以与捕获的品牌名称的图像进行比较以确定何时存在特定视频覆盖。作为另一个示例,特定视频覆盖可能在屏幕上的特定位置或通知中包括图标。为了说明,电话图标可以显示在关于来电话的通知中。相应地,将屏幕的各个部分与存储的电话图标的图像进行比较,以确定何时存在通知。下面的伪代码表示一些示例实施例:

```
templateID templateMatch(videoInStream, deviceID) {  
    //遍历中间设备的可用模板  
    //直到找到输入流的匹配  
    //返回匹配的流的 templateID, 如果没有则返回 0。  
    matchFound = 0;  
    do {  
        currentTemplate = getNextTemplate(deviceID);  
[0084]        if (singleTemplateMatch(videoInStream, currentTemplate))  
            matchFound = currentTemplate;  
    } until (matchFound != 0)  
    return matchFound;  
}  
  
Boolean singleTemplateMatch(videoInStream, template) {  
    //比较模板和输入流  
    //如果模板匹配则返回 TRUE, 否则返回 FALSE
```



```
        region = template.region;

        count = 0;

        for (x = region.x; x < region.width; x++)
            for (y = region.y; y < region.height; y++)
                if ((videoInStream.currentFrame(x,y) - template(x,y)) ** 2) <=
PIXEL_MATCH_DIFFERENCE)
                    count++;

[0085]         //对于精确的像素匹配，PIXEL_MATCH_DIFFERENCE = 0

        maxCount = x*y;

        threshold = PERCENT_THRESHOLD * maxCount;

        if (count >= threshold)

            return TRUE;

        else

            return FALSE;

    }
```

[0086] 上面的singleTemplateMatch()的示例实现使用相应像素之间的差的平方作为距离测量。其他示例实施例使用差的绝对值、差的归一化平方、相关系数、归一化相关系数、互相关、归一化互相关或其任何合适的组合。

[0087] 在操作830中,客户端设备150或替换内容服务器180基于所识别的视频数据和/或与客户端设备150相关联的简档信息来确定替换视频。例如,可以基于对特定的R级内容的识别来选择R级内容的PG级替换版本。作为另一个示例,可以选择当地商业广告来替换特定的国家商业广告。

[0088] 在操作840中,视频覆盖模块330基于所识别的机顶盒和视频数据来确定覆盖掩模。在一些示例实施例中,将视频数据与一组已知视频帧进行比较,以确定机顶盒140是否正在显示特定菜单或其他信息元素。例如,机顶盒140可能在屏幕的顶部呈现信息栏,其中某些像素是黑色的,并且其中某个区域具有(110,110,110) - (160,160,160)范围内的平均像素颜色(以红-绿-蓝(RGB)颜色元组表示)。因此,当所接收到的视频数据的一个帧或一组连续帧具有信息栏的特性时,视频覆盖模块330识别出对应的覆盖掩模,其中通过部分对应于栏所占据的区域,并且替换部分对应于由机顶盒140接收的原始视频内容占据的区域。

[0089] 模板(例如与图4-7中呈现的示例相对应的那些)可以以几种不同方式指示覆盖的区域或区块。例如,覆盖区域的边界的坐标可以由模板提供。或者,整个显示器的位图可以指示覆盖区域(例如,值1指示对应像素是覆盖的部分,并且值0指示对应像素不是覆盖的部分)。

[0090] 在上面的示例中,选择掩模以允许用户看到由机顶盒生成的覆盖信息,以及替换视频内容。在一些示例实施例中,选择掩模来抑制由机顶盒生成的覆盖信息。例如,可以对覆盖信息执行光学字符识别(OCR)以识别其中包含的任何文本。基于文本,可以执行数据库查找以确定覆盖是应该通过还是抑制。当覆盖要被抑制时,选择在覆盖区域中呈现替换视频内容的掩模。这两种方法也可以混合使用,允许向用户呈现一些覆盖内容,并且抑制其它覆盖内容。例如,如果覆盖呈现与输入视频内容相关联的文本信息,例如正在显示的产品的描述,则文本信息不可能与替换视频内容相关。因此,抑制文本信息改善了用户观看替换视频内容的体验。

[0091] 在操作850中,视频覆盖模块330使用掩模来显示视频数据的一部分和替换视频的一部分。继续上面的示例,与掩模的通过部分对应的包含信息栏的视频数据与替换视频的与掩模的替换部分对应的一部分一起显示。在一些示例实施例中,对替换视频的该部分的显示包括重新定位替换视频或调整替换视频的尺寸。例如,如果使用图4的掩模,则替换视频的尺寸减小并且重新定位以占据区域420。

[0092] 在一些示例实施例中,掩模随时间变化。例如,由机顶盒140呈现的覆盖信息可以是动态的并且随时间改变位置。相应地,其中呈现接收到的视频数据和替换视频数据的区域的位置可以随着时间而改变。在一些示例中,可以通过重复操作840和850来执行随时间改变掩模和/或替换视频的尺寸/位置。例如,操作840和850可以以迭代方式重复,直到视频覆盖模块330不再确定应该使用覆盖掩模(例如,当输入视频不再匹配具有相应掩模的模板时)。在一些示例中,可以基于与特定掩模和/或模板相关联的预定定时例程(例如,其中掩模的一些方面以可预测的方式随时间变化的动画例程)来执行随时间改变掩模和/或替换视频的尺寸/位置。

[0093] 还可以稍微修改地执行方法800以补偿由客户端设备150提供的视频元素。可以通过分析视频数据来检测由客户端设备150提供的视频元素(方式与分析由机顶盒140提供的视频元素相同)。可替换地,客户端设备150可以将用户交互报告给视频覆盖模块330。例如,当客户端设备150接收到显示菜单的用户命令时,可以将该信息提供给视频覆盖模块330。基于正在显示的特定菜单,视频覆盖模块330识别要应用的掩模。

[0094] 图9是示出了一些示例实施例中的用于动态地提供视频覆盖的方法900的流程图。方法900包括操作810-850和910-950。以上参照图8描述了操作810-850。作为示例而非限制,操作910-950被描述为由图1的组件110-190、图2的模块210-250以及图3的模块310-350执行。

[0095] 在操作910中,客户端设备150检测由机顶盒140提供的视频数据到第二视频数据的改变。例如,机顶盒140的用户可能已经改变了正在显示的频道。

[0096] 响应于检测到改变,视频覆盖模块330停止应用掩模并且允许显示从机顶盒140接收到的视频的全部(操作920)。因此,机顶盒140的用户将看到频道改变的效果。

[0097] 在操作930和940中,客户端设备150检测由机顶盒140提供的视频数据中的第二改变。第二改变将接收到的视频数据返回到之前的状态。例如,机顶盒140的用户可能已经返回到正在显示的原始视频频道。检测可以通过从视频识别系统190接收一组预期指纹并将预期指纹与由查询指纹生成器160生成的指纹进行比较来完成。因此,当用户改变频道以离开识别的内容时,指纹将停止匹配,并且当用户将频道改变回所识别的内容时,指纹将恢复

匹配。

[0098] 在操作950中,响应于确定第二改变是返回到第一视频数据,视频覆盖模块330重新开始使用掩模以显示第一视频数据的一部分和替换视频的一部分。因此,客户端设备150的用户将感知到对在操作850中呈现的替换视频的返回。

[0099] 图10是示出了一些示例实施例中的用于动态地提供视频覆盖的方法1000的流程图。方法1000包括操作810、820、830、840、1010和1020。以上参照图8描述了操作810-840。作为示例而非限制,操作1010和1020被描述为由图1的组件110-190、图2的模块210-250以及图3的模块310-350执行。

[0100] 在操作1010中,视频源标识符模块320确定所识别的视频数据的播放速率。例如,视频识别系统190可以提供针对视频数据的一组预期指纹,该组指纹与针对接收到的视频数据生成的指纹进行比较。因此,当针对接收到的视频数据的连续帧的指纹匹配非连续的预期指纹(例如,每二或四个指纹)时,确定所接收的视频以与由视频识别系统190使用的参考视频不同的速率(例如,两倍或四倍速率)播放。

[0101] 在操作1020中,视频覆盖模块330基于掩模和播放速率来显示从机顶盒140接收到的视频数据的一部分以及替换视频的一部分。例如,如在操作850中那样,使用掩模来确定从机顶盒140接收到的视频的哪一部分通过并且哪一部分被替换视频替换。此外,播放速率用于控制替换视频的播放速率。以这种方式,客户端设备150和机顶盒140的用户感觉到以与对原始显示的视频进行播放控制相同的方式对替换视频进行播放控制,例如快进、后退和暂停等。

[0102] 图11是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。方法1100包括操作1110-1150。作为示例而非限制,操作1110-1150被描述为由图1的组件110-190、图2的模块210-250以及图3的模块310-350执行。

[0103] 在操作1110中,客户端设备150的机顶盒标识符模块310访问包含来自内容提供者(例如,内容源110)和中间设备(例如,机顶盒140)的内容的视频输入流。例如,机顶盒140可能已经经由电缆连接从内容源110接收了视频流,修改了视频流,并且经由HDMI连接将修改的视频流发送到客户端设备150。客户端设备150可以在视频输入流被接收时访问视频输入流或者从数据存储(例如,从集成到客户端设备150中的固态驱动器)访问视频输入流。

[0104] 在操作1120中,机顶盒标识符模块310访问与由中间设备生成的屏幕部分相对应的模板。例如,中间设备执行的某些图像操纵对于客户端设备150可能是已知的。图像操纵可以对应于屏幕的特定部分。为了说明,机顶盒140可以通过在屏幕的顶部或底部布置菜单来修改从内容源110接收到的视频流。由机顶盒标识符模块310访问的模板对应于包含由中间设备生成的视频的屏幕部分。

[0105] 在操作1130中,机顶盒标识符模块310分析视频输入流以确定视频输入流的帧与模板匹配。例如,当视频输入流包含由机顶盒140生成的菜单时,模板可以指示特定颜色的某些像素。可以将每个进入视频帧与模板进行比较以确定屏幕部分是否可能包含菜单。可以不分析某些视频帧。例如,可以每隔一个帧或每隔四个帧与模板进行比较,从而减少处理器花费在比较上的时间量。在一些示例实施例中,操作1130仅在呈现替换内容时执行,由此减少了产生误报的机会。

[0106] 可以使用视频输入流的帧的下采样版本与模板的下采样版本的逐像素比较来执

行模板匹配。可以计算(下采样的)帧和模板之间的像素距离。例如,可以使用不匹配像素的数量作为像素差。可替换地,可以计算每个像素的颜色距离(例如,在RGB;亮度和色度(YUV);色调、饱和度和亮度(HSL);或色相、饱和度和值(HSV)的颜色空间中)以及将颜色距离的绝对值之和作为像素距离。将确定的像素距离与阈值进行比较以确定模板是否与视频输入流匹配。在一些示例实施例中,模板和视频输入流是互相关的。互相关允许当一个序列落后于另一个序列时,通过确定帧序列的相似性测量来处理多帧模板。

[0107] 在一些示例实施例中,将多个模板与视频输入流进行比较。例如,图16的表1640可以存储针对多个机顶盒中的每一个的模板。表1640中的每个模板包括指示视频输入流的哪些像素将与模板的图像的像素进行比较的位图。即使当机顶盒140已经被识别时,多种类型的多个模板也可用于与视频输入流进行比较。在一些示例实施例中,通过计算复杂度来对模板进行排序,使得首先测试需要较少计算的模板。可以使用第一个匹配模板,停止比较过程。可替换地,可以比较所有模板并使用最佳匹配(例如,与视频输入流帧具有最低像素距离的模板)。

[0108] 在操作1140中,视频覆盖模块330基于确定视频输入流的帧与模板匹配来生成视频输出流。例如,如果来自内容源110的内容正被客户端设备150替换为替换内容,并且确定机顶盒140正在提供补充内容(例如,菜单),则视频输出流可以被修改以在检测到菜单时包括该菜单,例如通过应用合适的覆盖掩模以允许菜单出现在视频输出流中。

[0109] 在操作1150中,视频覆盖模块330使得视频输出流被呈现在显示器(例如,客户端设备150的显示器)上。一些客户端设备150提供应用程序编程接口(API),其允许视频覆盖模块330指定在其中呈现视频输入流的显示器的一部分以及在其中呈现替换视频内容的显示器的另一部分。例如,可以通过使用逐像素地指示应该使用视频输入流像素还是替换像素的1位透明度掩模、或者通过使用逐像素地指示视频输入流像素和替换像素之间的混合程度的阿尔法掩模,来使用矩形区域的角坐标指定这些部分。对于这些客户端设备150,可以使用API。否则,由视频覆盖模块330执行任何部分混合或替换。

[0110] 图12是示出了一些示例实施例中的用于动态地提供视频覆盖的方法1200的流程图。方法1200包括操作1210-1240。作为示例而非限制,操作1210-1240被描述为由图2的模块210-250以及图3的模块310-350执行。

[0111] 在操作1210中,客户端设备150的机顶盒标识符模块310访问视频输入流,该视频输入流包括第一内容和至少部分覆盖在覆盖区域中的第一内容上的半透明覆盖。例如,第一内容可以是来自内容提供者(例如,内容源110)的内容,并且半透明覆盖可以由中间设备(例如,机顶盒140)提供。例如,机顶盒140可以已经经由电缆连接从内容源110接收了视频流,修改了该视频流,并且经由HDMI连接将修改的视频流传输给客户端设备150。客户端设备150可以在视频流被接收时对其访问或者从数据存储器(例如,从集成到客户端设备150中的固态驱动器)访问视频流。

[0112] 在操作1220中,机顶盒标识符模块310访问包括视频输入流中的覆盖区域的指示的模板。例如,中间设备执行的某些图像操纵对于客户端设备150可以是已知的。图像操纵可以对应于屏幕的特定部分。为了说明,机顶盒140可以通过在屏幕的顶部或底部布置半透明菜单来修改从内容源110接收到的视频流。由机顶盒标识符模块310访问的模板对应于包含由中间设备(例如,机顶盒140)生成的半透明覆盖的屏幕部分。

[0113] 视频源标识符模块320可以使用指纹识别来识别内容。例如,接收到的视频数据可以包括覆盖有由机顶盒140生成的半透明菜单的广播电视信号。使用补丁选择模块210和值计算模块220来生成接收到的视频数据的一个或多个指纹,客户端设备150将指纹发送给视频识别系统190。作为响应,客户端设备150接收对应于广播电视信号的标识符,该标识符用于标识接收到的视频数据的包含广播电视信号的部分。

[0114] 在操作1230中,视频覆盖模块330生成包括覆盖在替换内容上的经修改的半透明覆盖的视频输出流。例如,如果来自内容源110的内容正被客户端设备150替换成替换内容,并且确定机顶盒140已经向来自内容源110的内容添加了半透明覆盖,则视频输出流包括替换内容和半透明覆盖的修改版本。在操作1240中,视频覆盖模块330使视频输出流被呈现在显示器(例如,客户端设备150的显示器)上。

[0115] 下面的伪代码方法表示一些替换实施例:

```
semitransparentOverlayTransfer(inputVideoStream,  
identifiedVideoContent, overlayRegion) {  
  
    // inputVideoStream 是接收到的视频流  
  
    // identifyVideoContent 是从内容提供者传输来的内容,而没有中间设备执行的修改
```

[0116]

```
    // overlayRegion 标识由检测到的模板定义的区域  
  
    //在调用此方法之前, outputVideoStream 包含替换内容,但不包含半透明覆盖  
  
    for (each pixel in overlayRegion) {  
  
        //将由中间设备执行的修改传送给替换内容
```

```
        delta = inputVideoStream(pixel) - identifiedVideoContent(pixel);
        outputVideoStream(pixel) += delta;
    }
}

semitransparentOverlayTransfer(inputVideoStream,
identifiedVideoContent, overlayRegion) {
    // inputVideoStream 是接收到的视频流

    // identifyVideoContent 是从内容提供者传输来的内容，而没有中间
    设备执行的修改

    // overlayRegion 标识由检测到的模板定义的区域

    //在调用此方法之前， outputVideoStream 包含替换内容，但不包含
    半透明覆盖
[0117]    for (each pixel in overlayRegion) {
        //如果像素已被中间设备显著改变，
        //则使用改变的像素
        //而不使用替换内容的对应像素

        delta = inputVideoStream(pixel) - identifiedVideoContent(pixel);
        if (delta > PIXEL_OVERLAY_THRESHOLD)
            outputVideoStream(pixel) = inputVideoStream(pixel);
    }
}

semitransparentOverlayTransfer(inputVideoStream,
identifiedVideoContent, overlayRegion, transparency) {
    // inputVideoStream 是接收到的视频流
```

// identifyVideoContent 是从内容提供者传输来的内容，而没有中间设备执行的修改

// overlayRegion 标识由检测到的模板定义的区域

// transparency 是介于 0 和 1 之间的值，指示由中间设备使用的半透明覆盖的透明度（0 表示完全不透明，1 表示完全透明）

//在调用此方法之前，outputVideoStream 包含替换内容，但不包含半透明覆盖

```
for (each pixel in overlayRegion) {  
    //中间设备的结果是 contentVideoStream *  
    //(transparency) + overlayPixel * (1 - transparency)  
    //求解 overlayPixel 得到：  
    overlayPixel=(inputVideoStream(pixel)-  
[0118] identifiedVideoContent(pixel) * transparency) / (1 - transparency);  
    //现在将原始覆盖应用到我们的输出  
    outputVideoStream(pixel) *= transparency;  
    outputVideoStream(pixel) += overlayPixel * (1 - transparency);  
}  
}
```

semitransparentOverlayTransfer(inputVideoStream,overlayRegion,
transparency) {

// inputVideoStream 是接收到的视频流

// overlayRegion 标识由检测到的模板定义的区域

// transparency 是介于 0 和 1 之间的值，指示在输出中使用的半透明覆盖的透明度（0 表示完全不透明，1 表示完全透明）。

// transparency 可以，但不必与中间设备使用的相同

//在调用此方法之前，outputVideoStream 包含替换内容，但没有半透明覆盖

//标识半透明覆盖的颜色作为覆盖区域的平均颜色（例如，随机未修改的像素应平均为中性灰度，例如 RGB（128,128,128），由半透明覆盖修改的像素应该平均为覆盖的颜色）

//可替代地，overlayColor 可以被包括作为模板的数据，而不是从inputVideoStream 计算出来

overlayColor = average(pixels in overlay region);

//覆盖区域中的像素是背景像素或前景像素

//背景像素是颜色值在预定的覆盖颜色范围内的那些像素；

//前景像素是在预定范围之外的那些像素

[0119] for (each pixel in overlayRegion) {

// delta 的计算表示为简单的减法，但可以实现为颜色空间中的欧几里德距离

delta = inputVideoStream(pixel) – overlayColor;

//不加改变地复制前景像素，但对背景像素应用透明度

if (delta > PIXEL_OVERLAY_THRESHOLD)

outputVideoStream(pixel) = inputVideoStream(pixel);

else {

outputVideoStream(pixel) *= transparency;

outputVideoStream(pixel)+=overlayColor * (1 – transparency);

}

}


```
}
```

```
    semitransparentOverlayTransfer(inputVideoStream,overlayRegion,  
transparency, textColor) {
```

```
    // inputVideoStream 是接收到的视频流
```

```
    // overlayRegion 标识由检测到的模板定义的区域
```

```
    // transparency 是介于 0 和 1 之间的值，指示在输出中使用的半透明  
覆盖的透明度（0 表示完全不透明，1 表示完全透明）。
```

```
    // transparency 可以，但不一定与中间设备使用的相同
```

```
    //在调用此方法之前，outputVideoStream 包含替换内容，但没有半透  
明覆盖
```

[0120] //识别半透明覆盖的颜色作为覆盖区域的平均颜色（例如，随机未修
改的像素应平均为中性灰度，例如 RGB（128,128,128），由半透明覆盖修
改的像素应该平均为覆盖的颜色）

```
    //可替代地，overlayColor 可以被包括作为模板的数据，而不是从  
inputVideoStream 计算出来
```

```
    //textColor 是颜色值，该颜色值由中间设备用于文本
```

```
    overlayColor = average(pixels in overlay region);
```

```
    for (each pixel in overlayRegion) {
```

```
        //复制文本而不更改，但对背景像素应用透明度
```

```
        //包含输入文本的像素在总体上是 inputVideoStream 的文本部分
```

```
        if (inputVideoStream(pixel) == textColor)
```

```
            outputVideoStream(pixel) = inputVideoStream(pixel);
```

```
        else {
```

```
            outputVideoStream(pixel) *= transparency;
```

```
        outputVideoStream(pixel)+= overlayColor * (1 - transparency);  
    }  
}  
}
```

```
semitransparentOverlayTransfer(inputVideoStream,overlayRegion,  
transparency, messageType) {
```

```
    // inputVideoStream 是接收到的视频流
```

```
    // overlayRegion 标识由检测到的模板定义的区域
```

```
    // transparency 是介于 0 和 1 之间的值，指示在输出中使用的半透明  
    覆盖的透明度（0 表示完全不透明，1 表示完全透明）。
```

```
    // transparency 可以，但不必与中间设备使用的相同
```

[0121]

```
    //在调用此方法之前，outputVideoStream 包含替换内容，但不包含半  
    透明覆盖
```

```
    //识别半透明覆盖的颜色作为覆盖区域的平均颜色（例如，随机未修  
    改的像素应平均为中性灰度，例如 RGB（128,128,128），由半透明覆盖修  
    改的像素应该平均为覆盖的颜色）
```

```
    //可替代地，overlayColor 可以被包括作为模板的数据，而不是从  
    inputVideoStream 计算出来
```

```
    // messageType 指示要显示的文本的信息类型
```

```
    //例如，MESSAGE_TIME =当前时间，
```

```
    // MESSAGE_CHANNEL =当前频道
```

```
    overlayColor = average(pixels in overlay region);
```

```
    for (each pixel in the overlay region) {
```

```
        // 将覆盖色应用到区域
```

```
        outputVideoStream(pixel) *= transparency;

        outputVideoStream(pixel) += overlayColor * (1 - transparency);

    }

    //将消息设置为包含适当信息的字符串
[0122]    message = getMessage(messageType);

    //将消息的文本添加到屏幕上，供用户查看

    printMessage(message, overlayRegion);

}
```

[0123] 图13是示出了一些示例实施例中的用于动态地提供视频覆盖的方法1300的流程图。方法1300包括操作1310-1360。作为示例而非限制，操作1310-1360被描述为由图2的模块210-250以及图3的模块310-350执行。

[0124] 在操作1310中，客户端设备150使视频输出流呈现在显示器上。例如，查询指纹生成器160可以已经根据从机顶盒140接收到的视频输入流生成了查询指纹。基于生成的查询指纹，可以已经从替换内容服务器180接收到替换视频内容，并在客户端设备150的显示器上作为视频输出流呈现。

[0125] 在操作1320中，客户端设备150访问包含来自内容提供者和中间设备的内容的视频输入流。例如，视频输入流可以包括来自内容源110的内容，该内容已经被修改以包括来自机顶盒140的附加内容。

[0126] 在操作1330中，客户端设备150访问与由中间设备生成的屏幕部分相对应的模板。例如，视频覆盖模块330可以访问图16的数据库模式中的记录，其包含机顶盒140的设备标识符的数据。

[0127] 在操作1340中，视频覆盖模块330分析视频输入流以确定视频输入流的帧的至少一部分内容与模板匹配。在一些示例实施例中，多个这样的模板被迭代以确定用于中间设备的任何模板是否对应于从中间设备接收到的视频输入流的帧的内容。模板可以对应于由中间设备添加到视频输入流的图标（例如，电话图标或邮件图标）、由中间设备添加到视频输入流的预定文本（例如，“电话”或“邮件”）、由中间设备添加到视频输入流的特定位置中的窗口或其任何合适的组合。

[0128] 下面的伪代码方法表示一些示例实施例：

```
findMatchingTemplate(inputVideoStream, deviceId) {  
    // inputVideoStream 是接收到的视频流  
    // 设备 Id 是中间设备的标识符  
    matchingTemplate = NULL;  
    // matchingTemplate 将存储已识别的匹配，或者，在没有匹配时为  
    NULL  
    while( tmatchingTemplate==NULL&&template=getNexTemplate(deviceId)) {  
        //遍历针对中间设备的所有模板，直到我们查看了所有模板或找到匹配  
        [0129] //将 inputVideoStream 的当前帧与模板的位图进行按位与操作，这  
        将清除帧中不用于确定此模板是否匹配的部分  
        frame = inputVideoStream.currentFrame() & template.bitmap;  
        //确定帧和模板之间的差异  
        //这可能是针对每个比较像素的 RGB 空间中距离的绝对值的总和  
        delta = compareImage(frame, image);  
        //如果模板与帧匹配，则这就是匹配的模板  
        if (delta < TEMPLATE_MATCH_THRESHOLD)  
            matchingTemplate = template;  
    }  
}
```

[0130] 在操作1350中，视频覆盖模块330基于模板与视频输入流的帧的至少一部分内容之间的匹配，修改视频输出流以包括由中间设备生成的屏幕部分。例如，如果机顶盒140正在向从内容源110接收到的内容添加通知消息，则视频输入流的一个或更多个帧将与用于通知的模板匹配。视频覆盖模块330识别包含通知的视频输入流的帧的部分，并将通知复制到视频输出流。在一些示例实施例中，复制通知包括将由模板标识的通知区域中的视频输入流的每个像素复制到视频输出流的对应像素。在其它示例实施例中，复制通知包括确定通知的内容（例如，通过OCR或图像识别）并且再生视频输出流中的内容。以这种方式，当来自替换内容服务器180的替换内容正被显示时，来自机顶盒140的通知被呈现给客户端设备150的观看者。在操作1360中，视频覆盖模块330使经修改的视频输出流被呈现在显示器（例如，客户端设备150的显示器）上。

[0131] 在一些示例实施例中,当视频输入流不再匹配模板时,视频输出流的修改停止。在其它示例实施例中,当视频输入流匹配与修改结束相关联的第二模板时,视频输出流的修改停止。例如,来自机顶盒140的通知可以具有由第二模板匹配的关闭通知的动画中的签名帧。因此,当第二模板匹配时,视频输出流不再被修改为包括通知。

[0132] 图14是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。方法1400包括操作1310-1340、1360和1450-1460。以上参照图13描述了操作1310-1340和1360。作为示例而非限制,操作1450-1460被描述为由图2的模块210-250以及图3的模块310-350执行。

[0133] 在操作1450中,视频覆盖模块330基于匹配来确定视频输入流的播放速度已经从默认速度被修改。例如,模板可以指示视频输入流的特定部分中的朝右的双三角形。基于这个双三角形的存在(由视频输入流和模板之间的匹配确定),视频覆盖模块330确定播放速度已经增加。类似地,模板可以指示朝左的三角形或者暂停符号,并且视频覆盖模块330可以确定视频输入流的播放速度已经被反转或者暂停。

[0134] 在一些示例实施例中,该模板还指示播放的速率。例如,可能会出现“x2”或“x4”,指示播放速度增加到二倍或四倍。作为另一示例,两个三角形可以指示两倍速度,三个三角形可以指示四倍速度,等等。特定模板的特定含义可以取决于由机顶盒标识符模块310识别的机顶盒140。

[0135] 在其他示例实施例中,模板不指示播放的速率。播放的速率可以通过比较视频输入流的连续帧的指纹与参考流的连续帧的指纹来确定。如果视频输入流的连续帧的指纹与参考流的指纹隔一匹配,则以双倍速度进行播放。如果指纹的顺序相反,播放会反向进行。如果视频输入流的连续帧的指纹加倍,则播放以半速进行。

[0136] 视频覆盖模块330在操作1460中修改视频输出流的播放速度。在一些示例实施例中,视频输出流的播放速度被修改为匹配视频输入流的播放速度。因此,如果视频输入流的速度相对于默认速度加倍,则视频输出流的速度也会加倍。

[0137] 本文公开的各种方法可以被组合。例如,将方法1300与方法1400相结合产生当视频输入流具有修改的流时具有修改的速度的视频输出流,其中视频输出流还包括由中间设备生成的屏幕部分(例如,快进指示符)。

[0138] 图15是示出了一些示例实施例中的用于动态地提供视频覆盖的方法的流程图。方法1500包括操作1510-1590。作为示例而非限制,操作1510-1590被描述为由图2的模块210-250以及图3的模块310-350执行。

[0139] 在操作1510中,客户端设备150访问包含来自内容提供者的内容的视频输入流。例如,客户端设备150的视频输入设备1714(如图17所示)可以连接到机顶盒140的视频输出端。视频输入流可以从机顶盒140发送到客户端设备150。视频输入流可以包括来自内容源110的内容和由机顶盒140生成的内容。例如,机顶盒140可以提供叠加在从内容源110接收到的视频流的部分上的信息覆盖。

[0140] 在操作1520中,客户端设备150基于视频输入流的内容来选择视频输出流。例如,查询指纹生成器160可以基于视频输入流的一个或更多个帧的至少一部分来生成查询指纹。基于指纹,选择替换视频内容(例如,通过数据库查找,通过与替换内容服务器180的通信,或两者)用作视频输出流。

[0141] 在操作1530中,客户端设备150使选择的视频输出流被呈现在显示器上。例如,视频输出流可以呈现在客户端设备150的显示器上。可替换地,视频输出流可以由客户端设备150的视频输出端发送到单独的显示设备。

[0142] 在操作1540中,视频源标识符模块320确定视频输入流中的改变指示视频输入流的内容提供者的改变。例如,查询指纹生成器160可以访问与视频输入流的原始内容匹配的一系列指纹。当由查询指纹生成器160生成的指纹不再与原始内容的一系列指纹匹配时,可以确定视频输入流的内容已经改变,这可以指示视频输入流的内容提供者也已经改变。例如,中间设备的用户可能已经改变了频道或改变了输入(例如,从有线电视源到广播电视源)。无论哪种情况,内容提供者都已更改。

[0143] 响应于确定视频输入流的内容提供者已经改变,客户端设备150使视频输入流被呈现在显示器上(操作1550)。因此,替换视频输出流被呈现为对在操作1510中访问的原始视频输入流的替换,但不是对在操作1540中检测到的改变的视频输入流的替换。

[0144] 在操作1560中,客户端设备150访问与由中间设备生成的屏幕部分相对应的模板。例如,视频覆盖模块330可以访问图16的数据库模式中的记录,其包含针对机顶盒140的设备标识符的数据。

[0145] 在操作1570中,视频覆盖模块330分析视频输入流以确定视频输入流的帧的至少一部分内容与模板匹配。在一些示例实施例中,多个这样的模板被迭代以确定中间设备的任何模板是否对应于从中间设备接收到的视频输入流的帧的内容。模板可以对应于由中间设备添加到视频输入流的图标(例如,有线图标或电视图标)、由中间设备添加到视频输入流的预定文本(例如,指示频道的数字或字母数字序列)、由中间设备添加到视频输入流的特定位置中的窗口或其任何合适的组合。

[0146] 在操作1580和1590中,基于匹配,客户端设备150确定视频输入流中的第二改变来指示返回到第一内容提供者,并且响应于该确定,恢复视频输出流的呈现。例如,如果原始视频输入流包括来自频道15的内容(其被识别为要被替换的匹配内容),则在操作1530中呈现替换视频内容。在频道改变之后,在操作1550停止呈现替换视频内容。在检测到返回到原始频道(例如,通过与指示当前频道为15的由中间设备提供的覆盖的模板匹配)之后,恢复视频输出流中的替换视频内容的呈现(操作1590)。因此,通过使用方法1500,可以提供对一个内容提供者的替换内容,同时允许用户查看其他提供者的内容并且不会通过被切换离开和返回到被替换的内容提供者而中断。

[0147] 在一些示例实施例中,视频输出流被修改以包括与模板匹配的视频输入流的一部分。例如,如果图6的拓扑形式的模板与包括由机顶盒140生成的频道信息的区域610匹配,那么区域610可以从视频输入流被复制到视频输出流,允许在呈现视频输出流的同时呈现由中间设备生成的信息。

[0148] 图16是示出了在一些示例实施例中的用于动态地提供视频覆盖的数据结构的框图1600。图16示出了画中画(PictureInPicture)表1610、菜单覆盖(MenuOverlay)表1620、菜单混合(MenuBlend)表1630和模板(Template)表1640。

[0149] PictureInPicture表1610包括用于设备标识符(DeviceID)、模板标识符(TemplateID)和窗口坐标(WindowCoord)的字段。在一些示例实施例中,PictureInPicture表1610包含用于每个机顶盒的每个画中画模板的一行。因此,一旦特定机顶盒通过其

DeviceID被识别并且特定模板通过其TemplateID被识别,则可以检索到替换内容的画中画显示的尺寸和位置的坐标。PictureInPicture表1610可以用于支持拓扑上与图4的模板类似的模板。

[0150] MenuOverlay表1620包括用于设备标识符、模板标识符和菜单坐标(MenuCoords)的字段。在一些示例实施例中,MenuOverlay表1620包含用于每个机顶盒的每个画中画模板的一行。因此,一旦特定的机顶盒通过其DeviceID被识别并且特定的模板通过其TemplateID被识别,则可以根据MenuCoord确定由菜单覆盖占据的视频的部分的尺寸和位置的坐标。MenuOverlay表1620可以用于支持拓扑上与图5和6的模板类似的模板。

[0151] MenuBlend表1630包括用于设备标识符、模板标识符、菜单坐标和混合坐标(BlendCoords)的字段。在一些示例实施例中,MenuBlend表1630包含用于每个机顶盒的每个菜单混合模板的一行。因此,一旦特定的机顶盒通过其DeviceID被识别并且特定的模板通过其TemplateID被识别,则可以根据MenuCoords确定由菜单覆盖占据的视频的部分的尺寸和位置的坐标,并且可以根据BlendCoords确定,通过将替换视频与来自机顶盒的视频混合来生成的屏幕的部分的尺寸和位置的坐标。MenuBlend表1630可以用于支持拓扑上与图7的模板类似的模板。在一些示例实施例中,MenuOverlay表1620被包含在MenuBlend表1630中,因为当BlendCoords定义零面积的区域时,MenuBlend表1630可以用于支持拓扑上与图5和6的模板类似的模板。

[0152] Template表1640包括用于设备标识符、模板标识符、类型、位图和图像的字段。位图的尺寸与图像和视频输入流的帧的尺寸相同。位图中的每个值指示视频输入流和图像中的对应像素是否用于确定模板是否匹配视频输入流。将与位图的“1”值相对应的图像的部分与视频输入流进行比较。与位图的“0”值相对应的图像的部分是黑色的。一旦找到匹配,该类型指示使用的模板的类型。例如,PICTUREINPICTURE的类型指示模板用于画中画显示,并且可以从PictureInPicture表1610中检索用于嵌入画面的窗口坐标。

[0153] 根据各种示例实施例,本文描述的一种或更多种方法可以促进动态视频覆盖。此外,本文描述的一种或更多种方法可以减少执行提供动态视频覆盖的方法的客户端设备上的计算时间、客户端设备和服务器之间的传输时间、对执行动态视频覆盖的方法的服务器的存储器存储要求、以及执行动态视频覆盖的方法的服务器上的计算时间。

[0154] 当综合考虑这些效果时,本文描述的一种或更多种方法可以避免动态视频覆盖中本来会涉及的某些努力或资源的需求。用户在动态视频覆盖中花费的努力可以通过本文所述的一种或多种方法来减少。由一个或多个机器、数据库或设备(例如,在网络环境100内)使用的计算资源可以类似地被减少。这种计算资源的例子包括处理器周期、网络流量、存储器使用、数据存储容量、功耗和冷却能力。

[0155] 图17是示出根据一些示例实施例的机器1700的组件的框图,该机器1700能够从机器可读介质(例如,机器可读存储介质、计算机可读存储介质或其任何适当的组合)读取指令,并且全部或部分地执行本文讨论的任何一种或更多种方法。具体而言,图17示出了计算机系统的示例形式的机器1700的图形表示,并且其中的用于使机器1700执行本文讨论的任何一种或更多种方法的指令1724(例如,软件、程序、应用程序、小程序,app或其他可执行代码)可以被部分或全部执行。在替换实施例中,机器1700作为独立设备操作或可以连接(例如,联网)到其他机器。在联网部署中,机器1700可以在服务器-客户端网络环境中以服务器

机器或客户端机器操作,或者在分布式(例如,对等)网络环境中作为对等机器操作。机器1700可以是服务器计算机、客户端计算机、PC、平板电脑、膝上型计算机、上网本、机顶盒、智能电视、个人数字助理(PDA)、蜂窝电话、智能电话、网络设备、网络路由器、网络交换机、网桥或能够顺序地或以其他方式执行指定要由机器执行的动作的指令1724的任何机器。此外,虽然仅示出单个机器,但术语“机器”也应被理解为包括单独或共同执行指令1724以执行本文所讨论的一个或更多个的方法的全部或部分的机器集合。

[0156] 机器1700包括处理器1702(例如,中央处理单元(CPU)、图形处理单元(GPU)、数字信号处理器(DSP)、ASIC、射频集成电路(RFIC)或其任何合适的组合);主存储器1704和静态存储器1706,其被配置为经由总线1708相互通信。机器1700还可以包括图形显示器1710(例如,等离子显示面板(PDP)、LED显示器、LCD、投影仪或CRT)。机器1700还可以包括字母数字输入设备1712(例如,键盘)、视频输入设备1714(例如,分量视频输入、复合视频输入或高清多媒体接口(HDMI))、存储器单元1716、信号生成设备1718(例如,扬声器)和网络接口设备1720。

[0157] 存储单元1716包括机器可读介质1722,在机器可读介质1722上存储有实现本文描述的任何一种或更多种方法或功能的指令1724。指令1724还可以在机器1700执行期间完全或至少部分地驻留在主存储器1704内,处理器1702内(例如,处理器的高速缓冲存储器内)或两者内。因此,主存储器1704和处理器1702可以被认为是机器可读介质。指令1724可以经由网络接口设备1720通过网络1726(例如,图1的网络170)发送或接收。

[0158] 如本文所使用的,术语“存储器”是指能够暂时或永久存储数据的机器可读介质,并且可以被认为包括但不限于随机存取存储器(RAM)、只读存储器(ROM)、缓冲存储器、闪存和高速缓冲存储器。虽然机器可读介质1722在示例实施例中示出为单个介质,但术语“机器可读介质”应被理解为包括能够存储指令的单个介质或多个介质(例如,集中式或分布式数据库、或者关联的缓存和服务)。术语“机器可读介质”还应被理解为包括能够存储由机器(例如,机器1700)执行的指令的任何介质或多个介质的组合,使得指令在由机器的一个或更多个处理器(例如,处理器1702)执行时,使机器执行本文描述的任何一个或更多个方法。因此,“机器可读介质”是指单个存储装置或设备,以及包括多个存储装置或设备的“基于云”的存储系统或存储网络。因此,术语“机器可读介质”应被理解为包括但不限于以固态存储器、光学介质、磁性介质或其任何适当组合的形式的一个或更多个数据存储库。“非暂时性机器可读介质”是指除暂态信号之外的所有机器可读介质。

[0159] 在整个说明书中,多个实例可以实现被描述为单个实例的组件、操作或结构。尽管一种或更多种方法的各个操作被图示和描述为单独的操作,但是各个操作中的一个或更多个可以同时执行,并且不要求操作以所示的顺序被执行。在示例配置中作为独立组件呈现的结构和功能可以实现为组合结构或组件。类似地,呈现为单个组件的结构和功能可以实现为分离的组件。这些和其他变化、修改、添加和改进落入本文主题的范围。

[0160] 本文将某些实施例描述为包括逻辑或多个组件、模块或机构。模块可以构成硬件模块。“硬件模块”是能够执行某些操作的有形单元并且可以以某种物理方式配置或布置。在各种示例实施例中,一个或更多个计算机系统(例如,独立计算机系统、客户端计算机系统或服务器计算机系统)或计算机系统的一个或更多个硬件模块(例如,处理器或一组处理器)可以由软件(例如,应用程序或应用程序的部分)配置为执行本文所述的某些操作的硬

件模块。

[0161] 在一些实施例中,硬件模块可以机械地、电气地或其任何合适的组合来实现。例如,硬件模块可以包括被永久配置为执行某些操作的专用电路或逻辑。例如,硬件模块可以是专用处理器,例如FPGA或ASIC。硬件模块还可以包括由软件暂时配置以执行某些操作的可编程逻辑或电路。例如,硬件模块可以包括包含在通用处理器或其他可编程处理器内的软件。应该理解,在专用和永久配置的电路中,或者在暂时配置的电路中(例如由软件配置的),机械地实现硬件模块的决定可能是由成本和时间考虑驱动的。

[0162] 因此,术语“硬件模块”应该被理解为包含有形实体,即物理构建、永久配置(例如,硬连线)或暂时配置(例如,编程)以某种方式操作或执行本文描述的某些操作的实体。如本文所使用的,“硬件实现的模块”是指硬件模块。考虑到其中硬件模块被暂时配置(例如,编程)的实施例,每个硬件模块不需要在任何一个时刻被配置或实例化。例如,在硬件模块包括由软件配置成专用处理器的通用处理器的情况下,通用处理器可以在不同时间分别被配置成不同的专用处理器(例如,包括不同的硬件模块)。软件可以相应地将处理器配置为,例如在一个时刻构成特定的硬件模块并且在不同的时刻构成不同的硬件模块。

[0163] 硬件模块可以向其他硬件模块提供信息并从其他硬件模块接收信息。因此,所描述的硬件模块可以被视为通信地耦合。在同时存在多个硬件模块的情况下,可以通过两个或更多个硬件模块之间的信号传输(例如,通过适当的电路和总线)来实现通信。在其中多个硬件模块在不同时间被配置或实例化的实施例中,可以例如通过将信息存储在多个硬件模块可访问的存储器结构中和从这样的存储器结构中检索信息来实现这些硬件模块之间的通信。例如,一个硬件模块可以执行操作并将该操作的输出存储在其通信地耦合到的存储器设备中。之后,另一个硬件模块可以在稍后访问存储器设备以检索和处理存储的输出。硬件模块也可以启动与输入或输出设备的通信,并且可以对资源(例如,一组信息)进行操作。

[0164] 本文描述的示例方法的各种操作可以至少部分地由暂时配置(例如,通过软件)或永久配置为执行相关操作的一个或更多个处理器来执行。无论是暂时配置还是永久配置,这样的处理器都可以构成处理器实现的模块,其操作来执行本文描述的一个或更多个操作或功能。如本文所使用的,“处理器实现的模块”是指使用一个或更多个处理器实现的硬件模块。

[0165] 类似地,本文描述的方法可以至少部分地由处理器实现,处理器是硬件的示例。例如,方法的至少一些操作可以由一个或更多个处理器或处理器实现的模块执行。此外,一个或更多个处理器还可以操作以支持执行“云计算”环境中的相关操作或作为“软件即服务”(SaaS)。例如,至少一些操作可以由一组计算机(作为包括处理器的机器的示例)执行,这些操作可以经由网络(例如,因特网)并且经由一个或更多个适当的接口(例如,API)访问。

[0166] 某些操作的执行可以分布在一个或更多个处理器中,不仅驻留在单个机器中,而且部署在多个机器上。在一些示例实施例中,一个或更多个处理器或处理器实现的模块可以位于单个地理位置(例如,在家庭环境、办公室环境或服务器场内)。在其他示例实施例中,一个或更多个处理器或处理器实现的模块可以分布在多个地理位置。

[0167] 本文讨论的主题的一些部分可以呈现为数据操作的算法或符号表示,数据在机器存储器(例如,计算机存储器)内存储为位或二进制数字信号。这样的算法或符号表示是数

据处理领域的普通技术人员用来将他们的工作的实质传达给本领域其他技术人员的技术的示例。如本文所使用的,“算法”是产生期望结果的操作或类似处理的自洽序列。在这种情况下,算法和操作涉及物理量的物理操纵。通常但不一定地,这样的物理量可以采取能够被机器存储、访问、传输、组合、比较或以其他方式操纵的电、磁或光信号的形式。主要出于常用的原因,有时使用诸如“数据”、“内容”、“位”、“值”、“元素”、“符号”、“字符”、“项”、“数字”、“数值”等词语来指代这样的信号是很方便的。然而,这些词只是便签,并且与适当的物理量相关联。

[0168] 除非另有特别说明,否则本文中使用的诸如“处理”、“计算”、“运算”、“确定”、“呈现”、“显示”等词语的讨论可以指机器的动作或过程(例如,电脑),其操纵或转换表示为在一个或多个存储器(例如,易失性存储器、非易失性存储器或其任何适当的组合)、寄存器或接收、存储、传输或显示信息的其他机器组件内的物理(例如,电、磁或光)量的数据。此外,除非另有特别说明,否则在此使用术语“一”或“一个”,如在专利文件中常见的那样,其包括一个或多个实例。最后,如本文所使用的,连词“或”是指非排他性的“或”,除非另有特别说明。

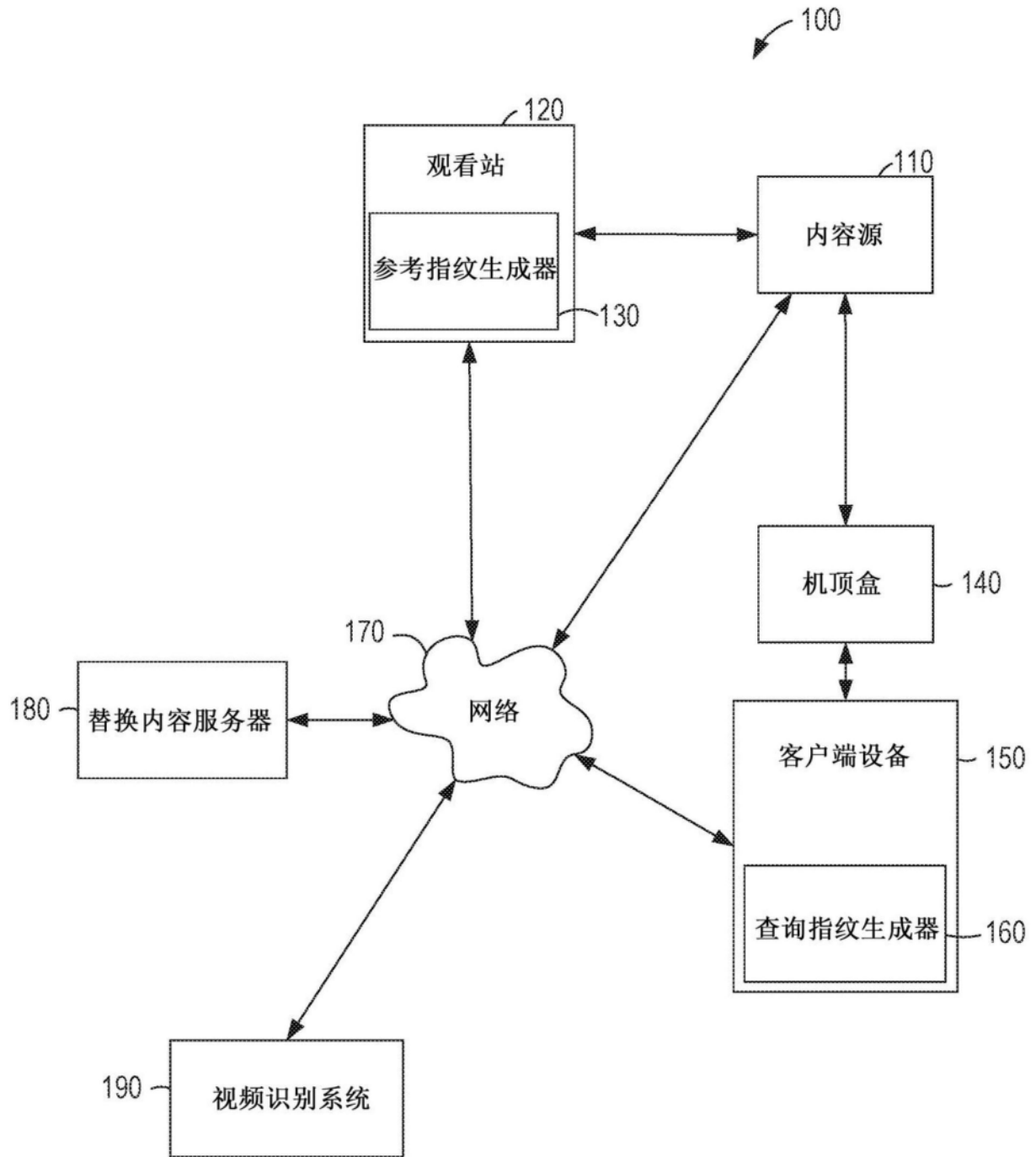


图1

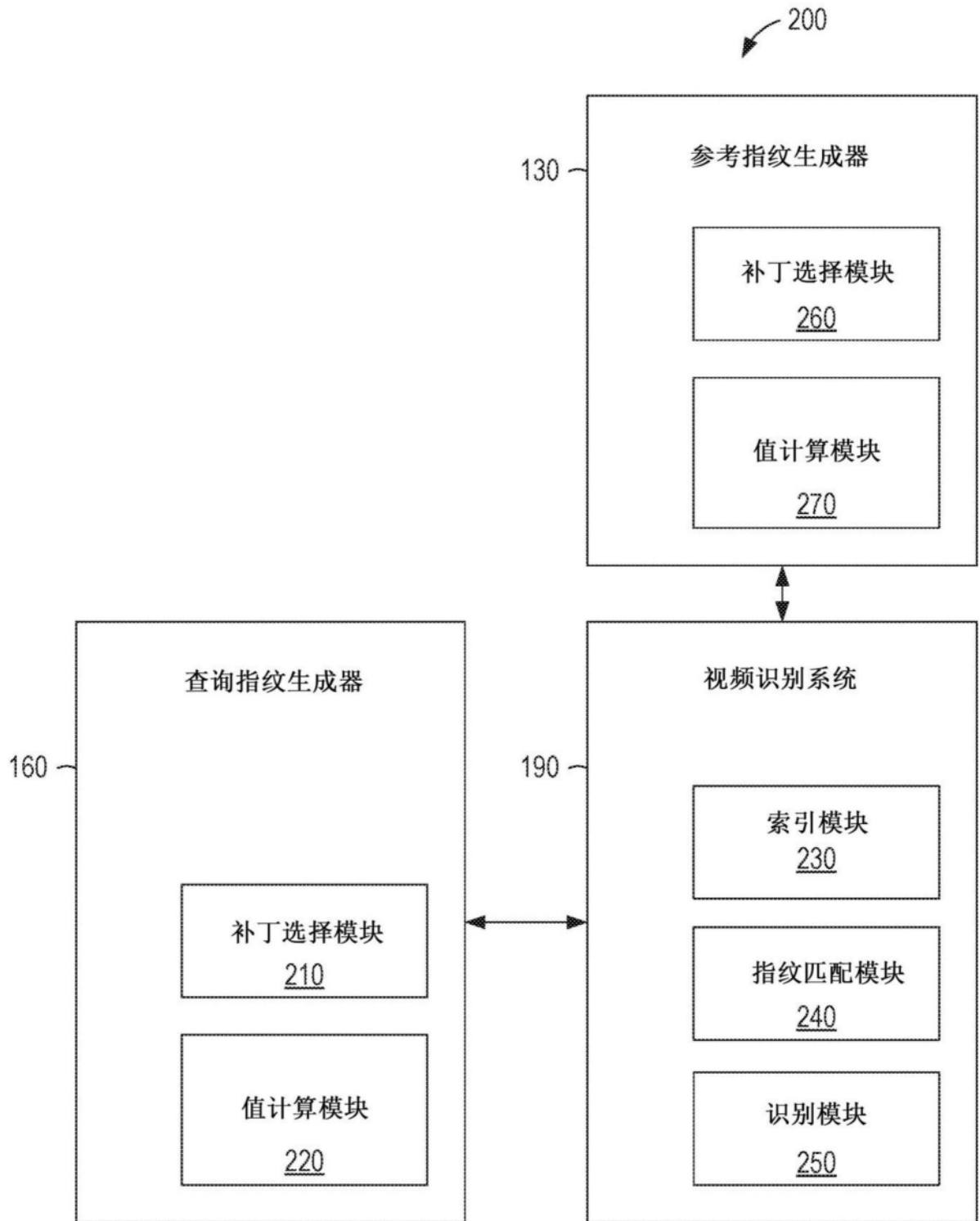


图2

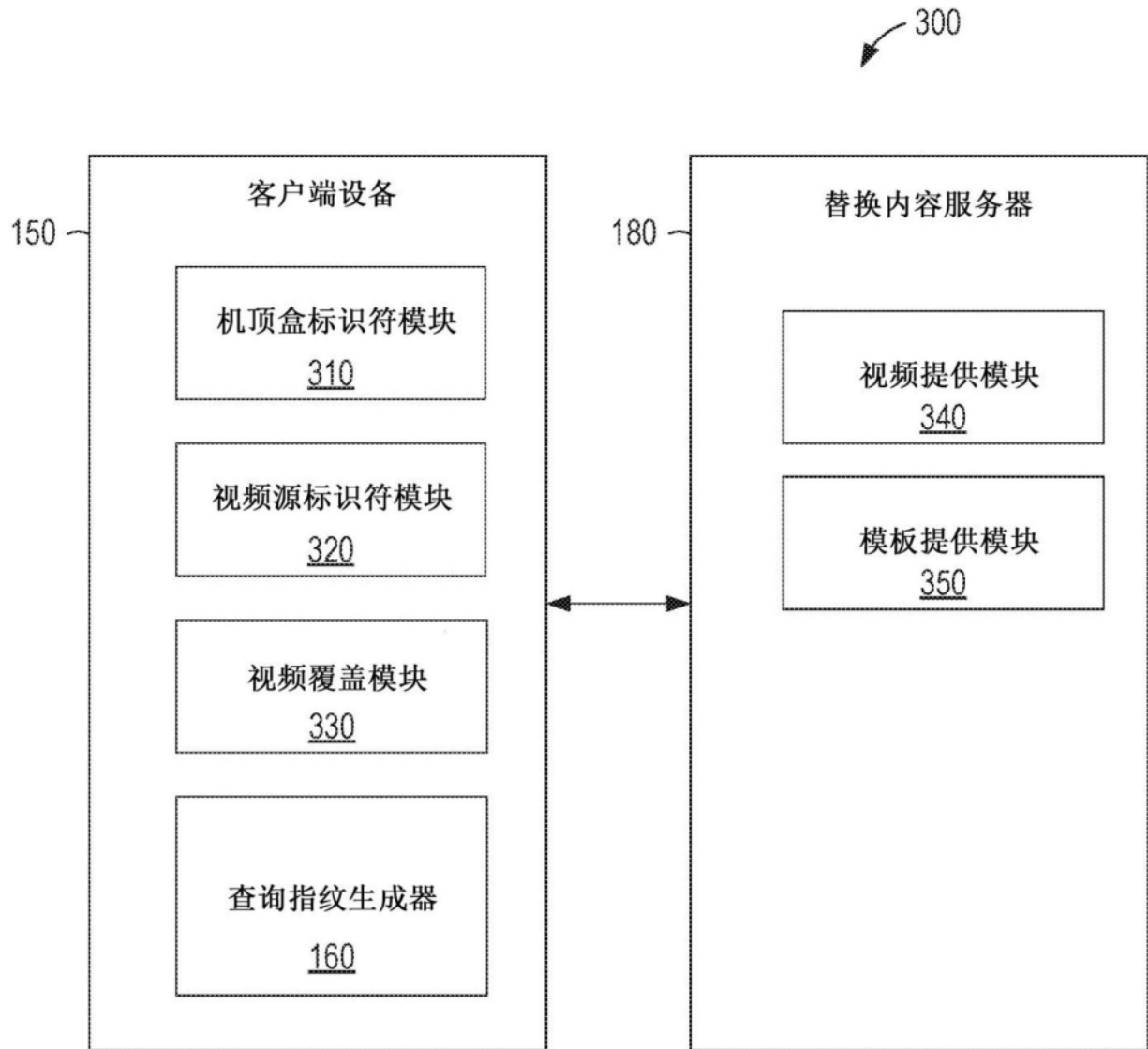


图3

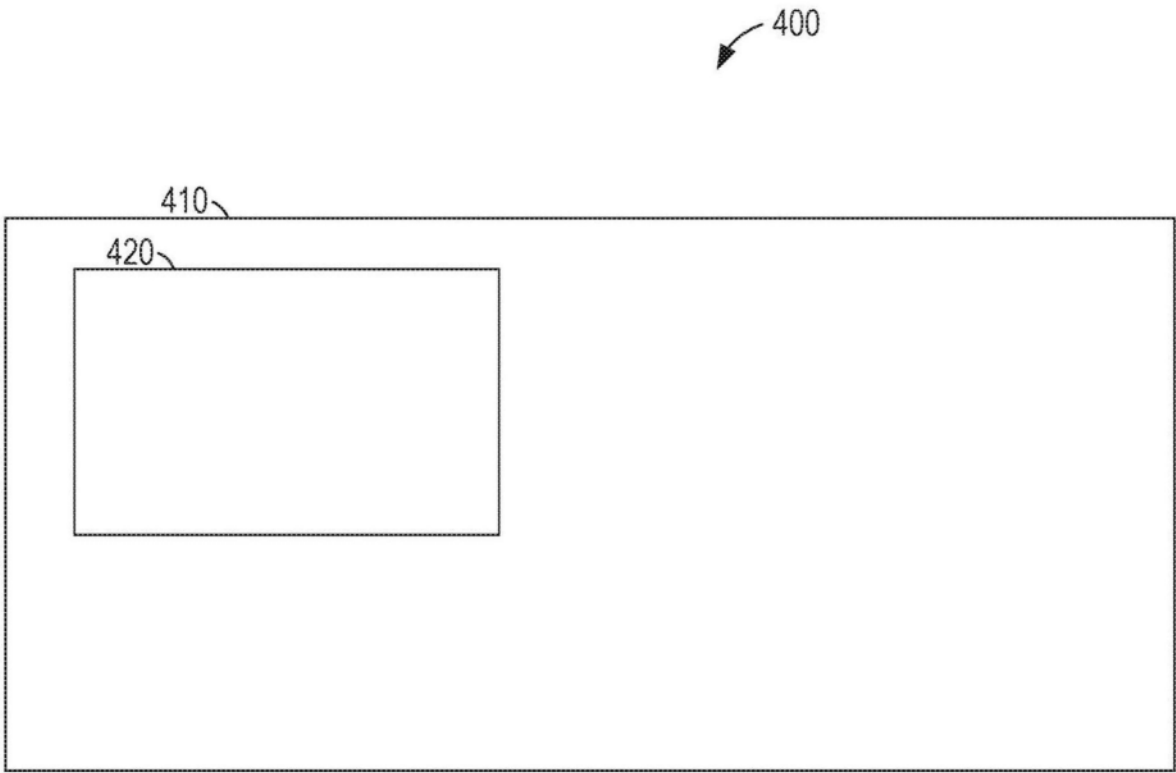


图4

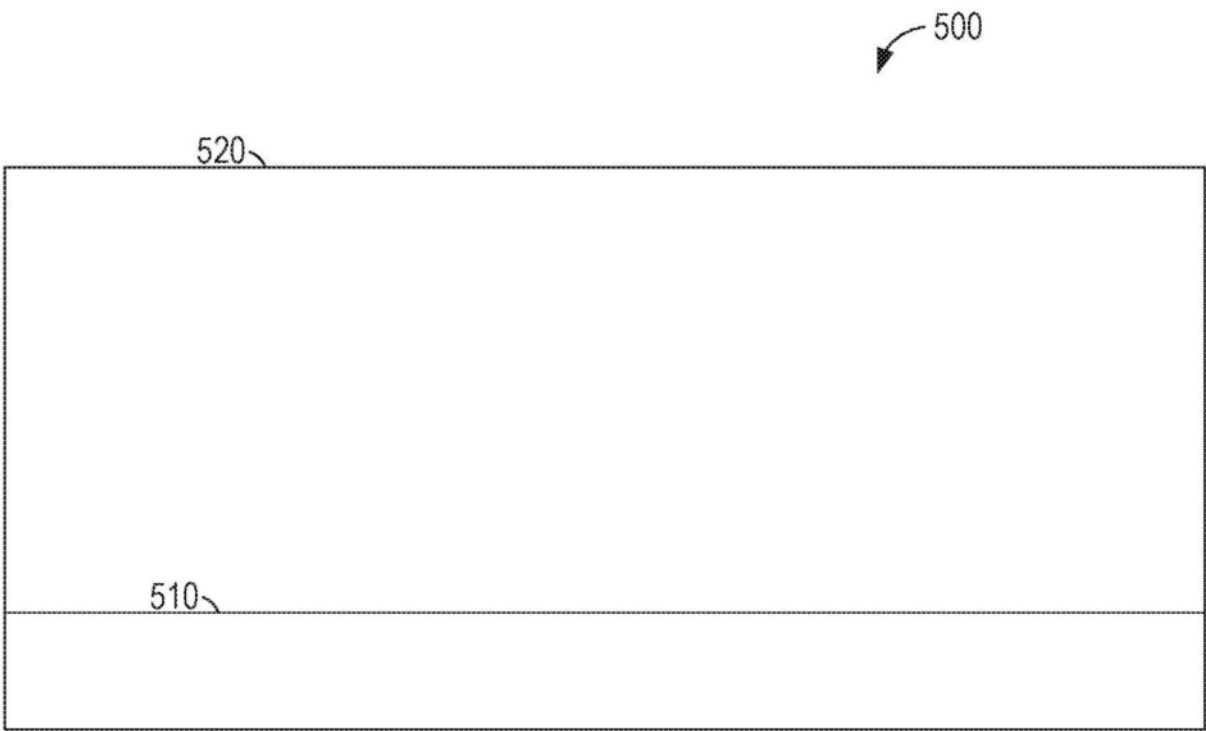


图5

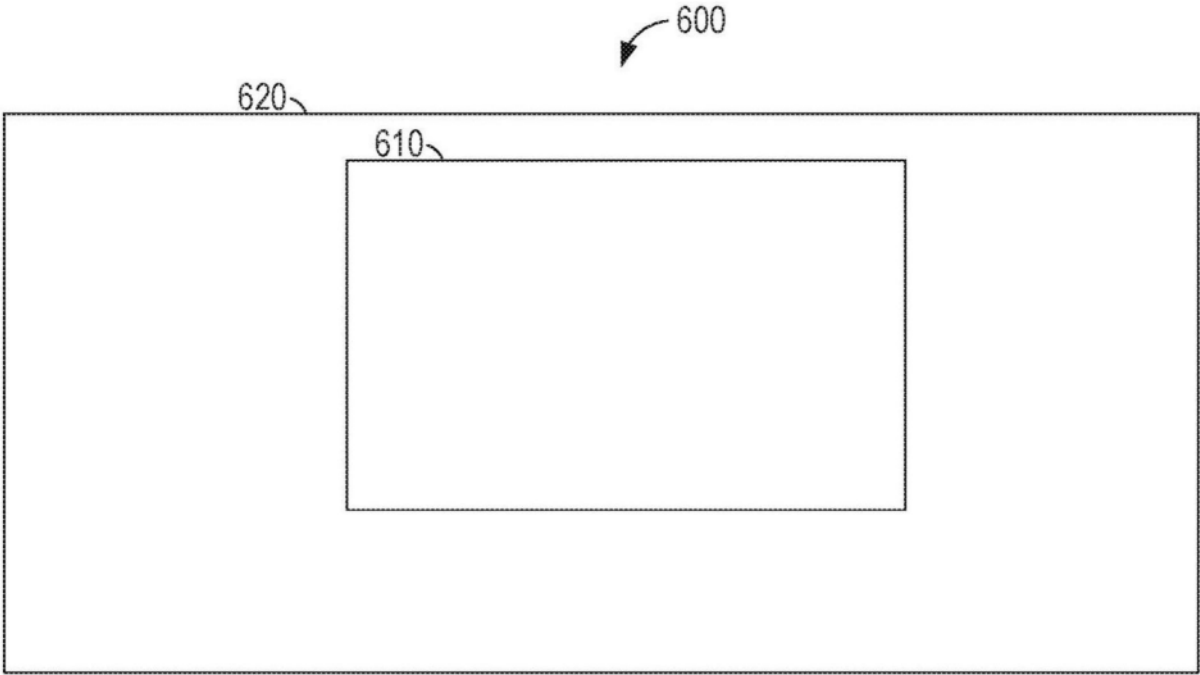


图6

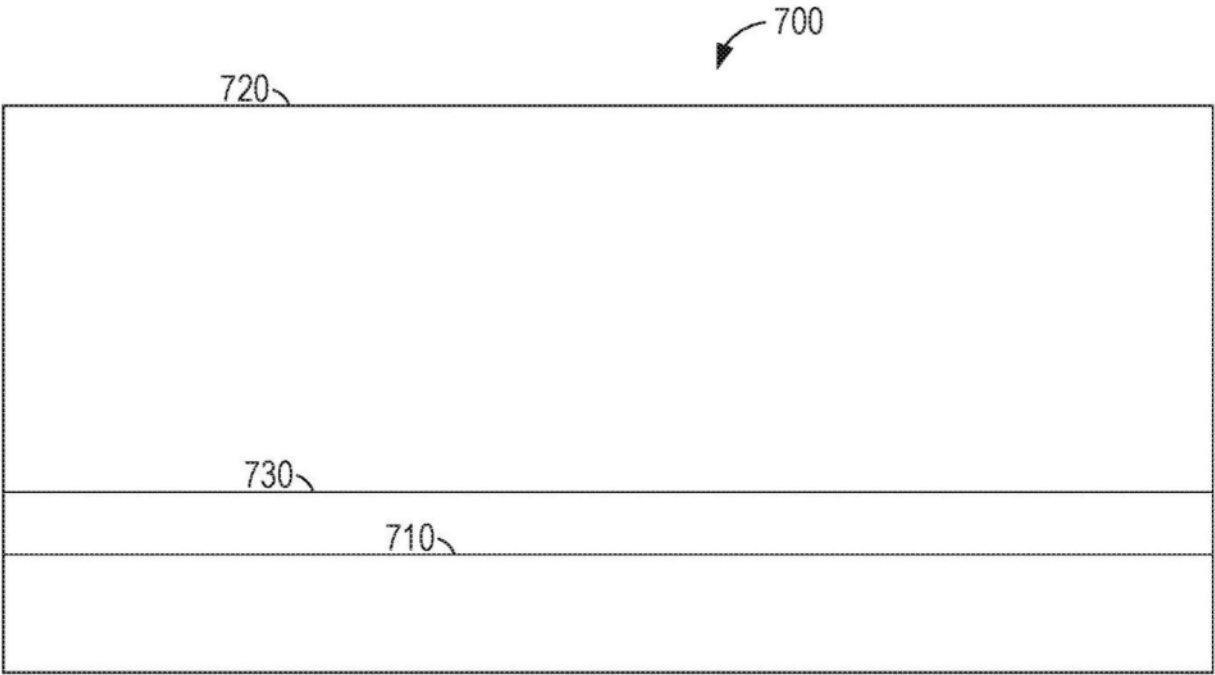


图7

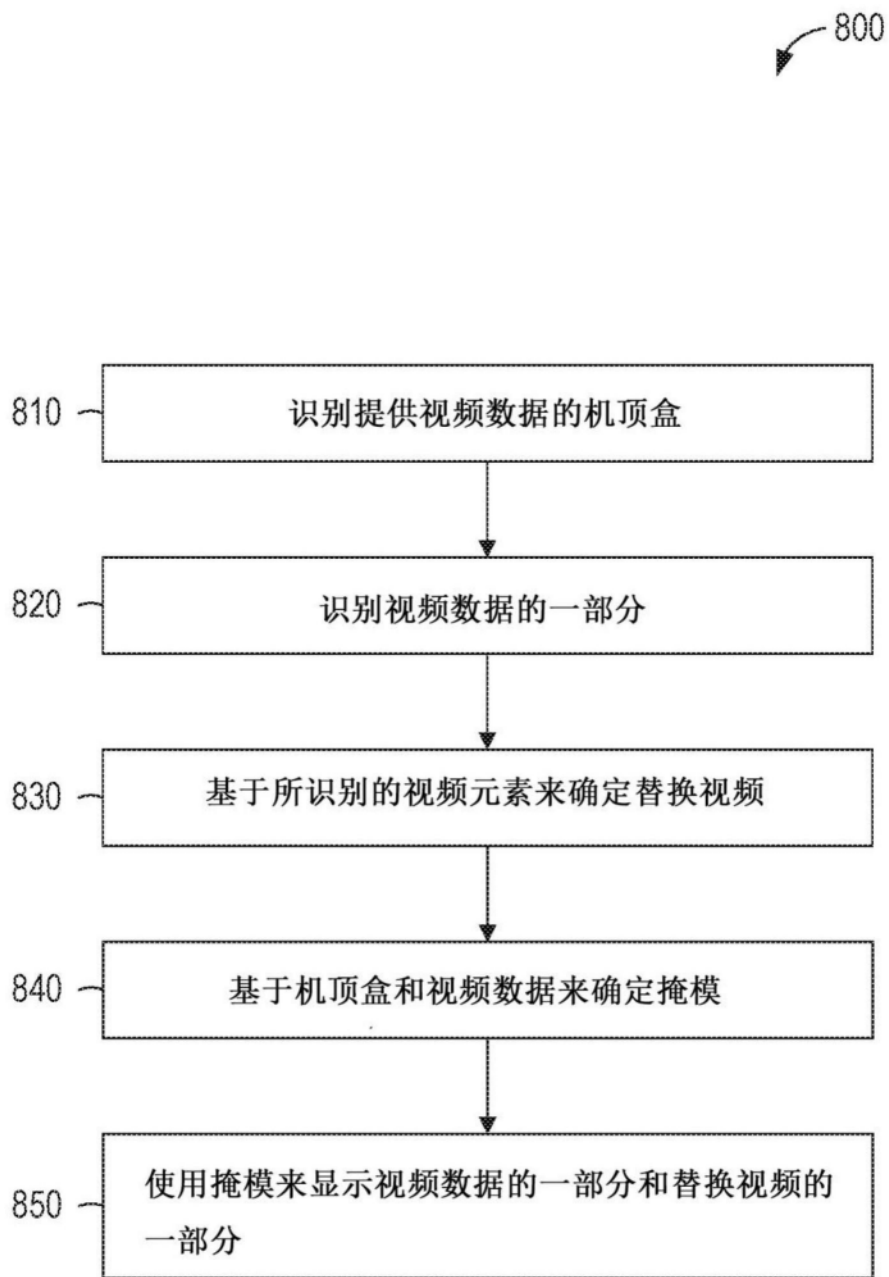


图8

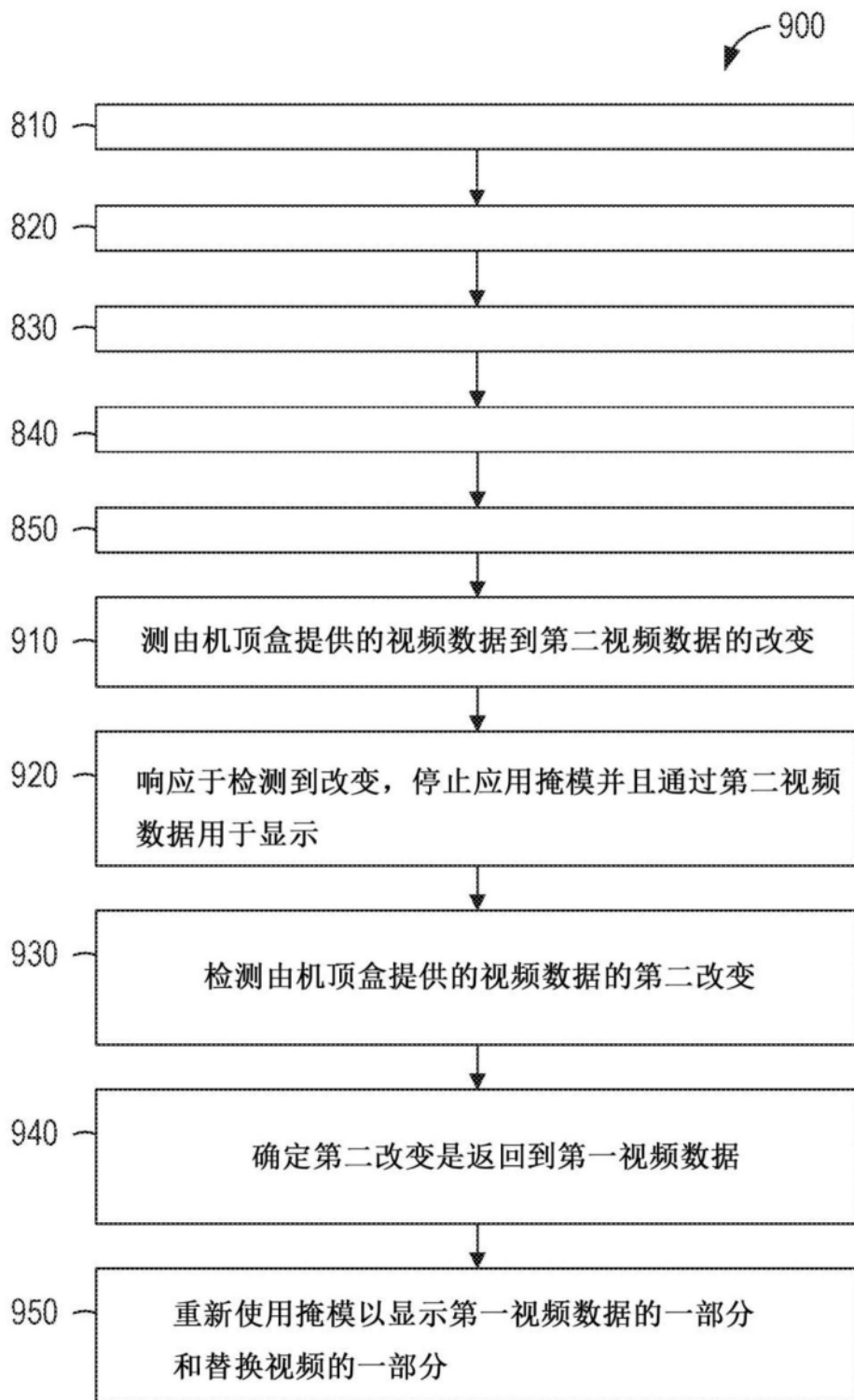


图9

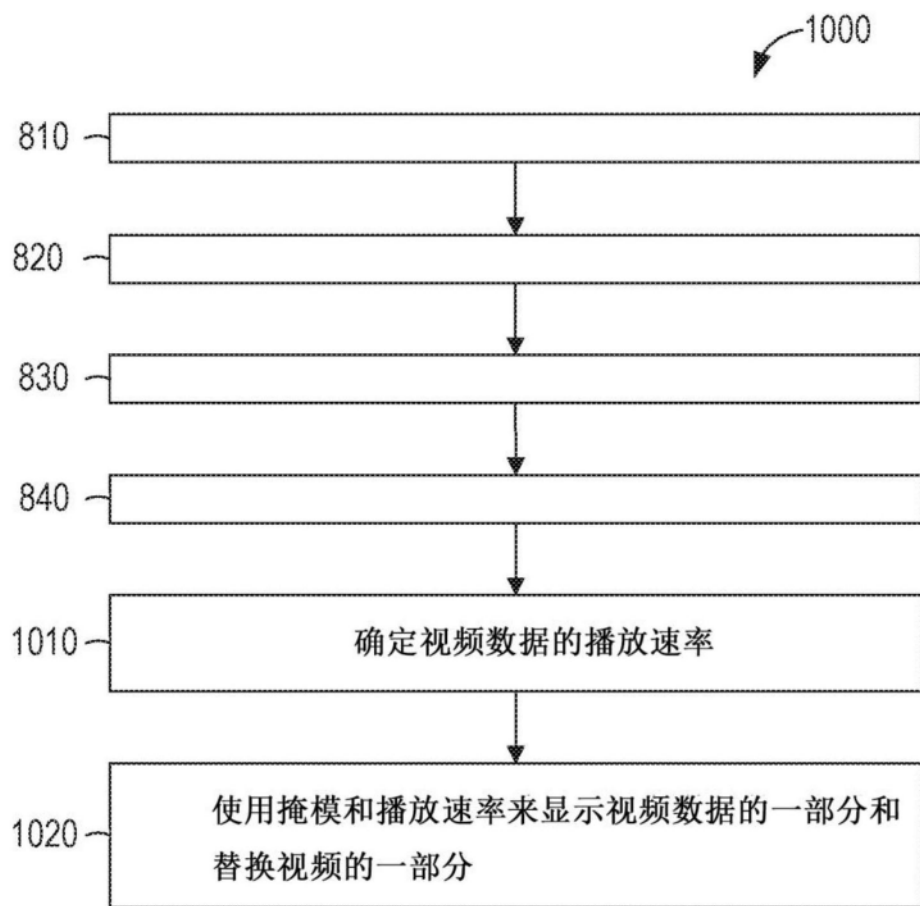


图10

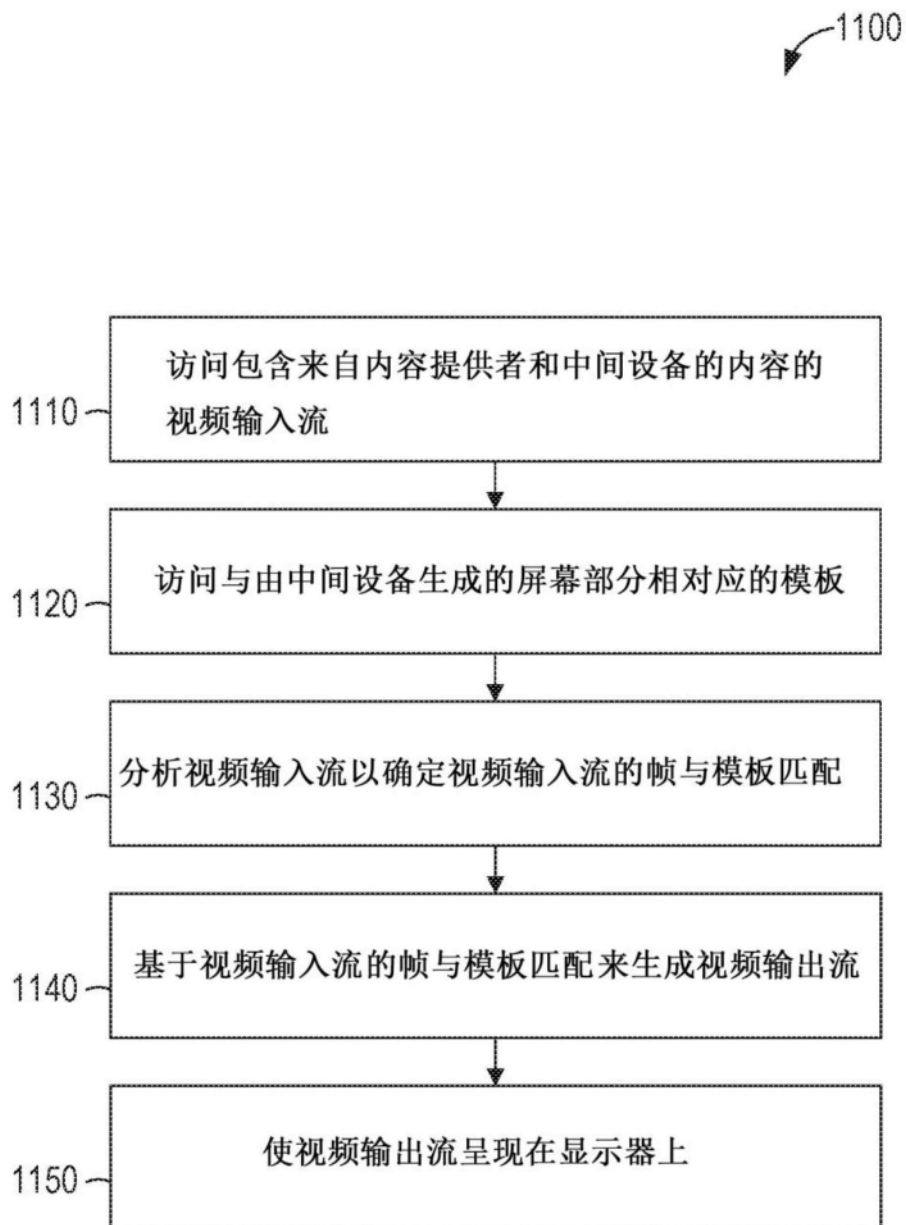


图11

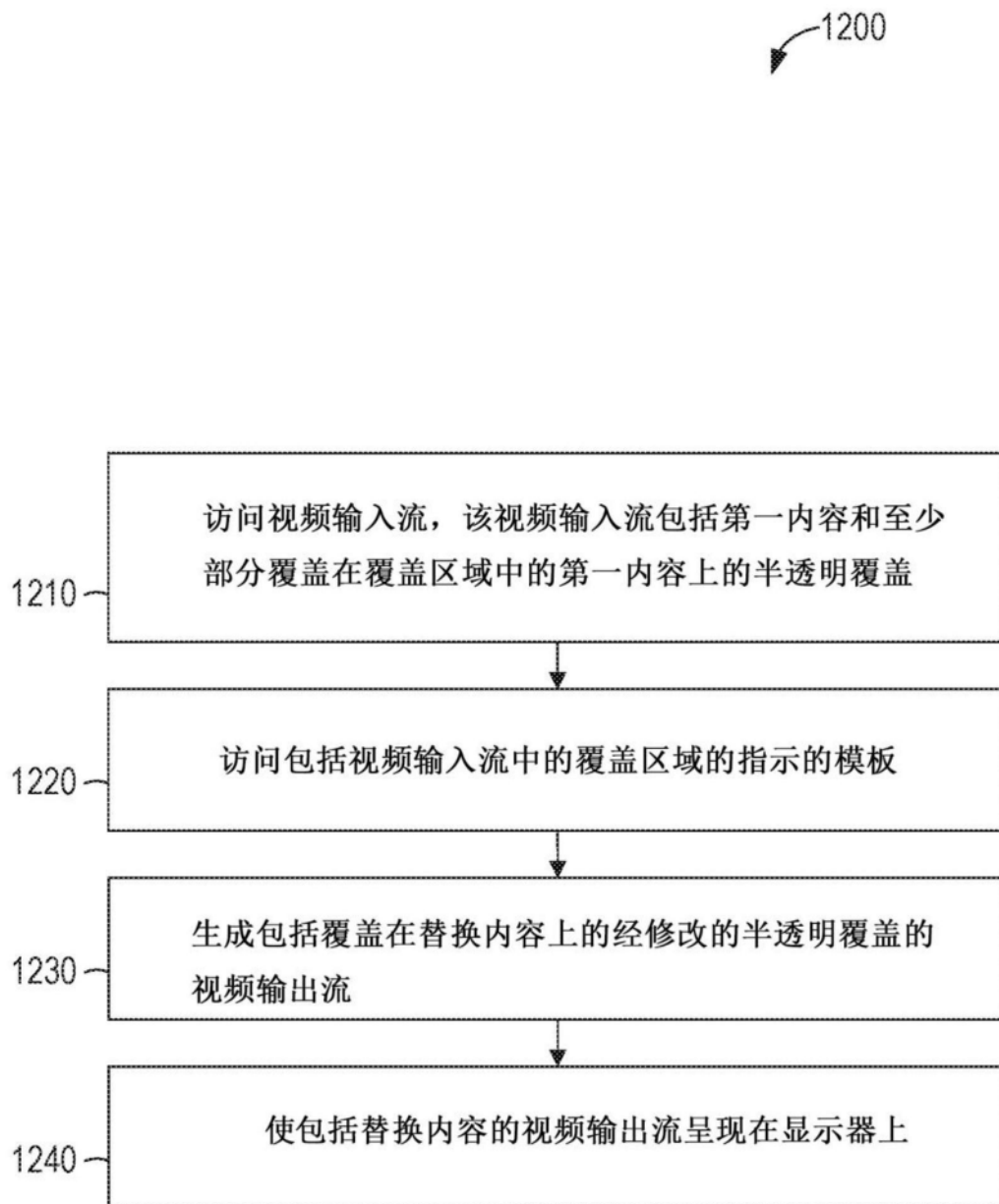


图12

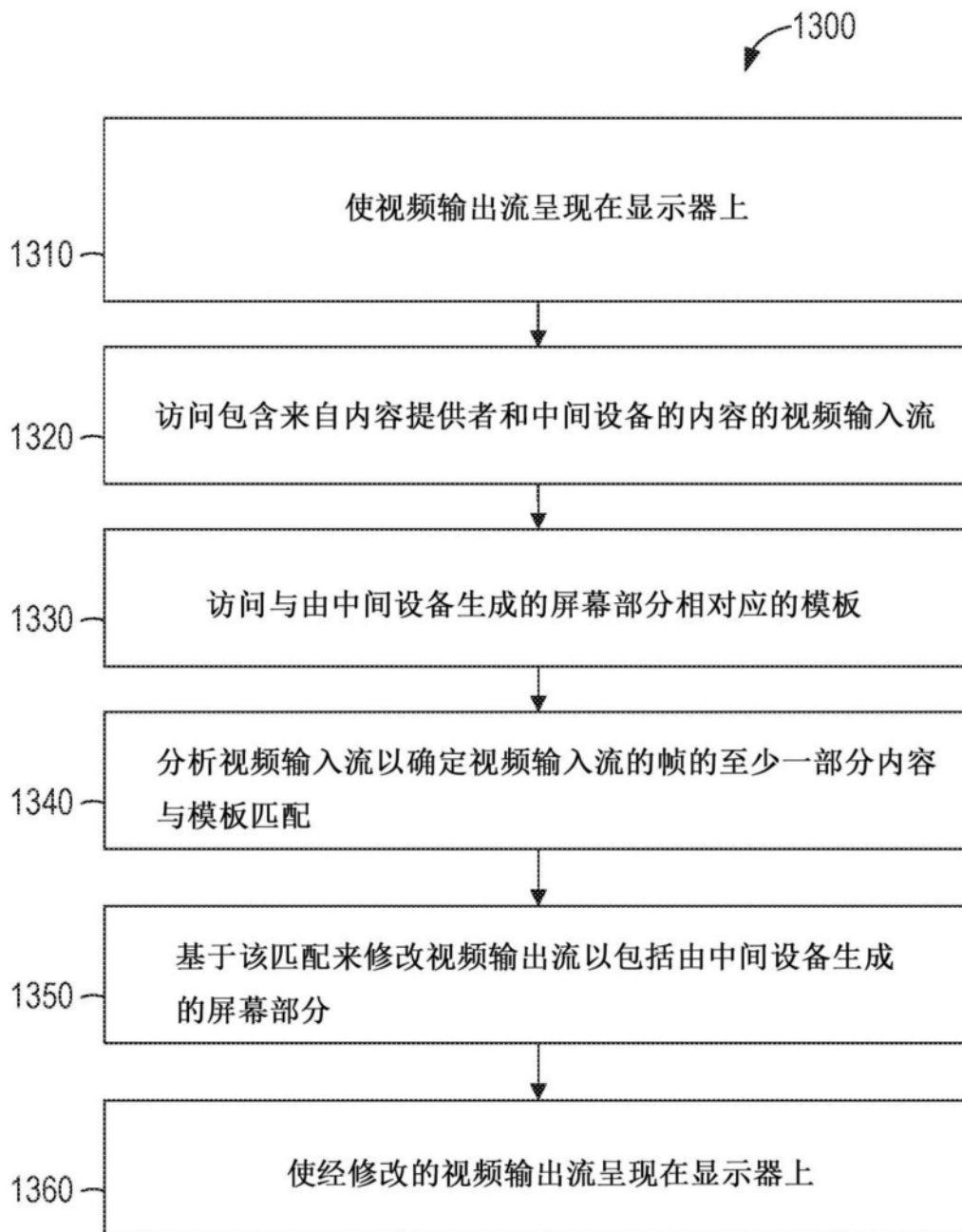


图13

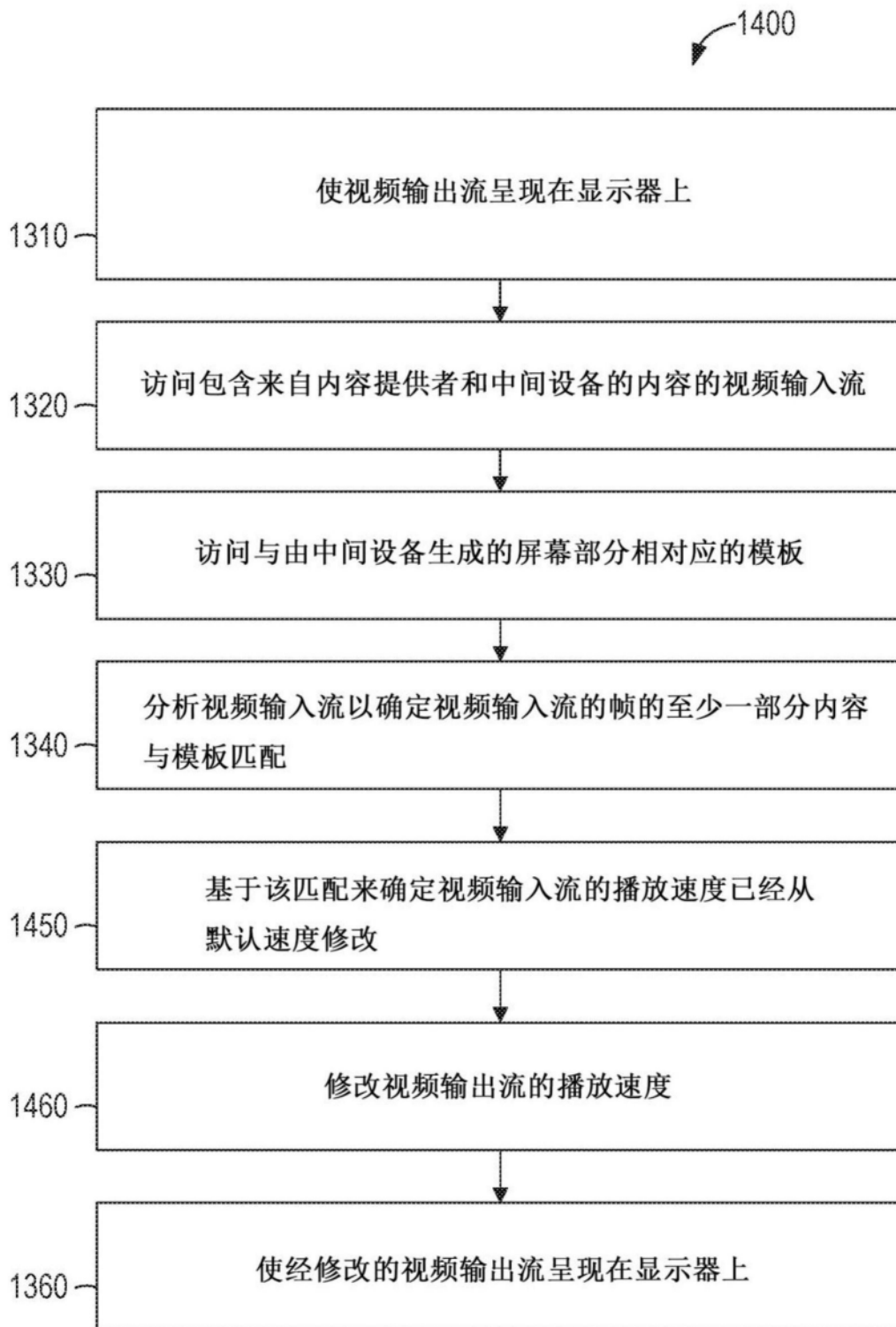


图14

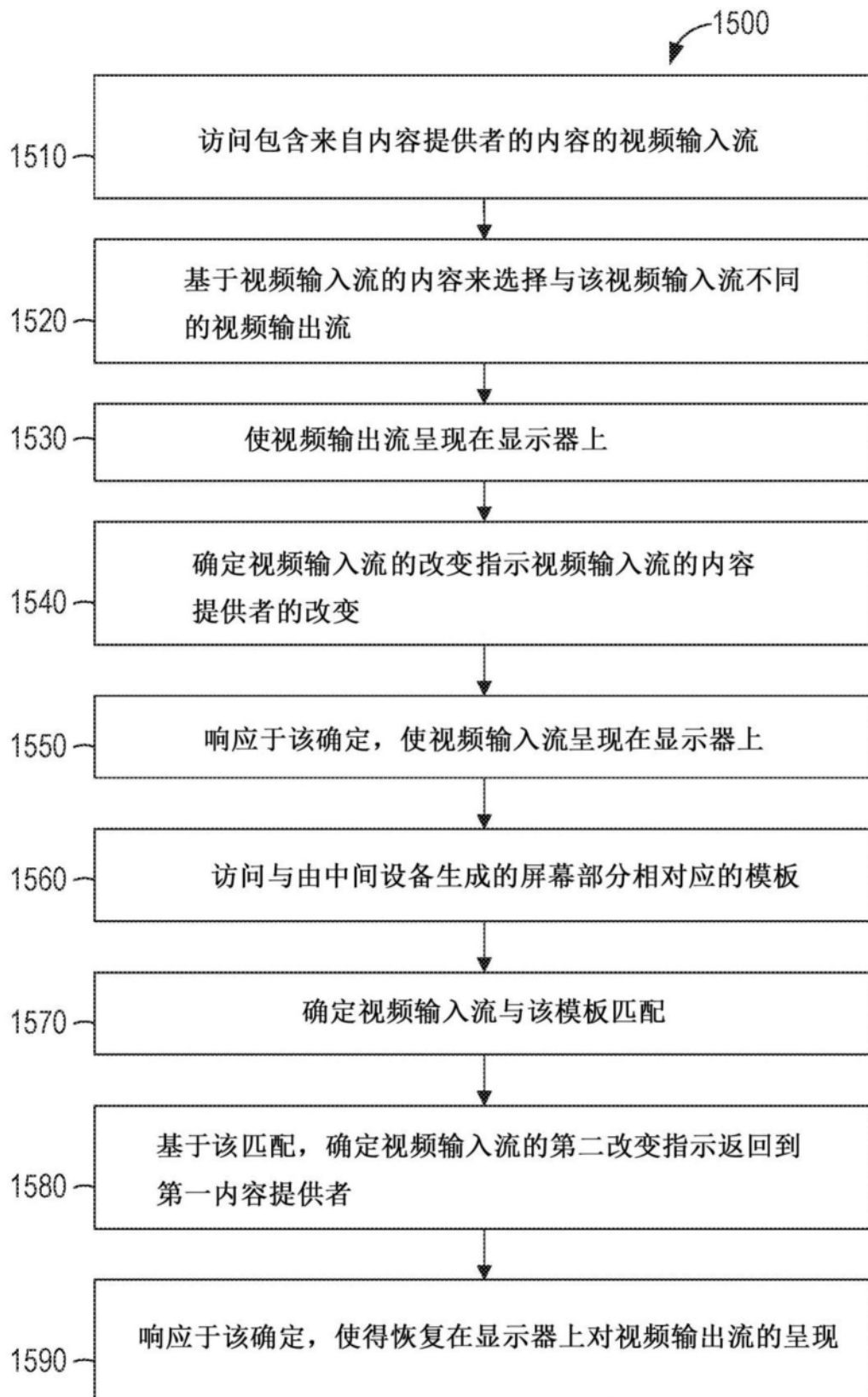


图15

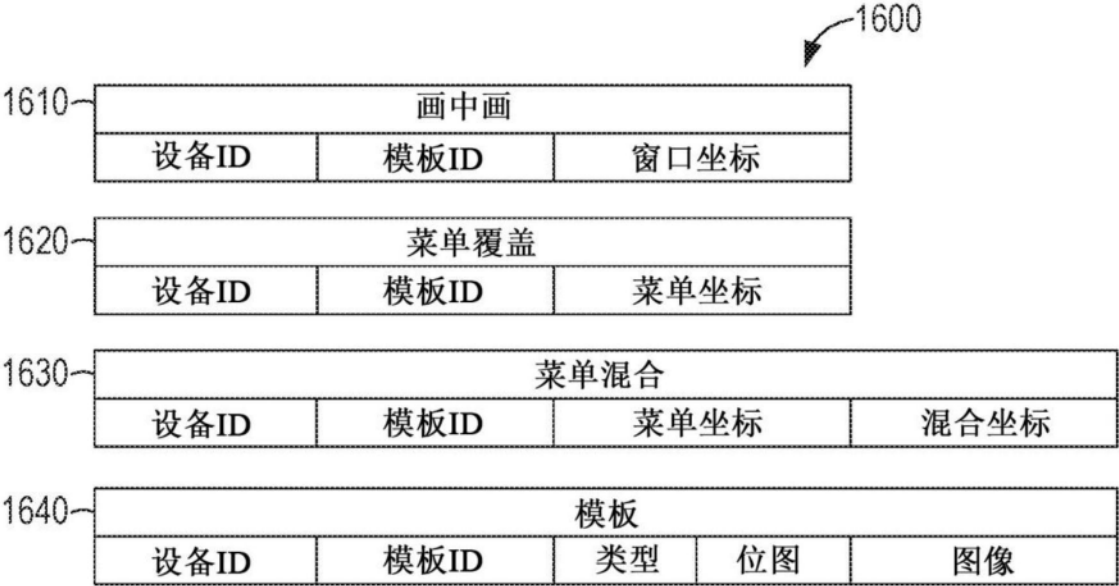


图16

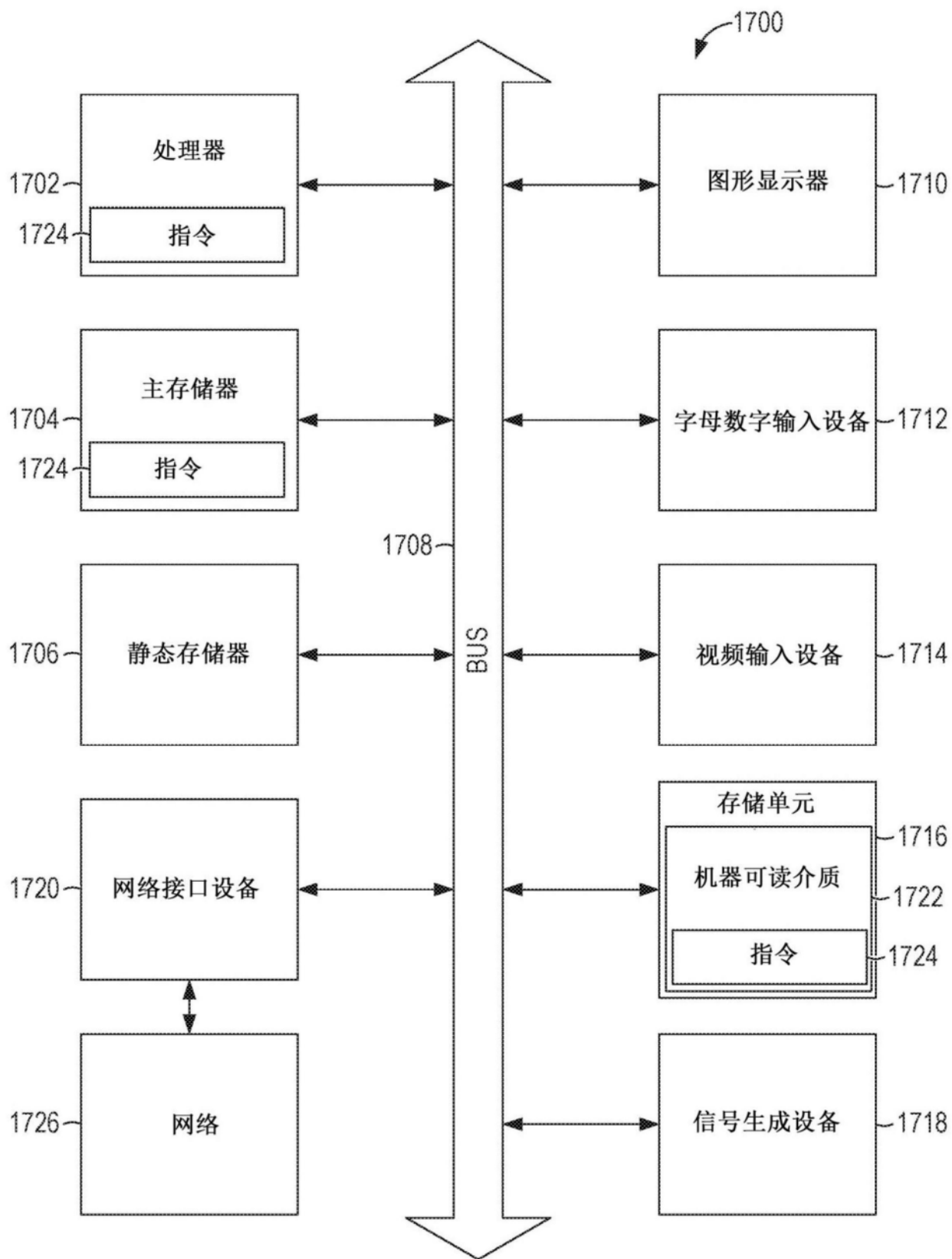


图17