(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111052182 A
(43)申请公布日 2020.04.21

(21)申请号 201880041720.9

(22)申请日 2018.04.20

(30)优先权数据
62/488,526 2017.04.21 US
62/634,464 2018.02.23 US
62/640,945 2018.03.09 US
62/644,164 2018.03.16 US

(85)PCT国际申请进入国家阶段日
2019.12.20

(86)PCT国际申请的申请数据
PCT/US2018/028620 2018.04.20

(87)PCT国际申请的公布数据
WO2018/195461 EN 2018.10.25

(71)申请人 泽尼马克斯媒体公司
地址 美国马里兰州

(72)发明人 迈克尔·科皮茨

(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
代理人 郜凤珠 刘继富

(51)Int.Cl.
G06T 7/20(2017.01)
G06T 5/00(2006.01)
G06T 7/00(2017.01)

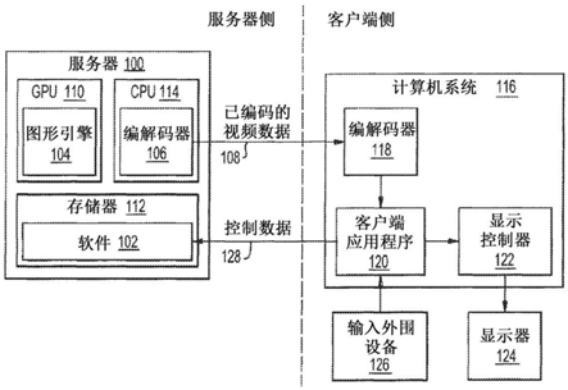
权利要求书4页 说明书17页 附图12页

(54)发明名称

通过预期运动矢量的玩家输入运动补偿

(57)摘要

通过运动估算和补偿来减少延迟的系统和方法结合了客户端设备，该客户端设备使用来自远程服务器的查找表来对运动矢量进行匹配、加标签和求和。当远程服务器将编码的视频帧发送到客户端时，客户端对它们进行解码，然后将求和的运动矢量应用于估算运动。该技术还基于预定的标准生成运动矢量，并且将所述运动矢量以及无效器发送到客户端以进行缓存。服务器指示客户端接收输入并使用该输入以匹配缓存的运动矢量或无效器。该技术还缓存重复运动矢量并且向客户端发送先前生成的运动矢量库。服务器指示客户端计算运动估算结果并指示客户端更新存储的运动矢量库，使得客户端应用存储的库以在接收实际运动矢量数据之前启动运动。



CN 111052182 A

1.一种用于运动估算的由计算机实现的方法,包括以下步骤:

发送包括一个或更多个用户输入以及相关联的一个或更多个运动矢量的查找表,其中所述查找表与缓存所述查找表的指令被发送;

在接收到玩家输入时,发送指令来查询查找表以匹配运动矢量;

发送指令以将唯一标签与来自查找表的匹配的运动矢量关联,并且将加标签的运动矢量添加到队列,其中对所述加标签的运动矢量求和;

发送具有唯一标识符标签的帧,其中所述标签表示帧包括与玩家输入相关联的运动的时间点;

发送指令以从队列中移除已经具有与从服务器接收的加标签的帧相关联的标签的运动矢量;以及

其中,当服务器将编码的视频帧发送给客户端时,客户端被指示在视频输出之前对视频帧进行解码并将求和的运动矢量应用于解码帧以估算运动。

2.根据权利要求1所述的由计算机实现的方法,其中所述编码的视频帧在不进行残差处理的情况下解码。

3.根据权利要求1所述的由计算机实现的方法,还包括指示客户端将一个或更多个平滑功能应用于队列中的求和的加标签的运动矢量的步骤。

4.根据权利要求1所述的由计算机实现的方法,其中与运动矢量相关联的所述唯一标签本质上是按时间顺序的。

5.根据权利要求1所述的由计算机实现的方法,其中所述求和的加标签的运动矢量与相应的宏块相关联。

6.根据权利要求1所述的由计算机实现的方法,其中缓存的查找表设置为基于玩家输入进行修改。

7.根据权利要求1所述的由计算机实现的方法,其中所述求和的加标签的运动矢量设置为应用一次。

8.根据权利要求1所述的由计算机实现的方法,其中所述求和的加标签的运动矢量描述任意复杂度的运动。

9.根据权利要求1所述的由计算机实现的方法,其中所述求和的加标签的运动矢量针对玩家输入估算未来的反馈。

10.根据权利要求1所述的由计算机实现的方法,其中所述求和的加标签的运动矢量遵循H.264编码标准。

11.一种用于运动估算的系统,其中,通过网络,服务器:

在接收到玩家输入时,向客户端发送指令来查询查找表以匹配运动矢量;

向客户端发送指令以将唯一标签与来自查找表的匹配的运动矢量关联,并且将加标签的运动矢量添加到队列,其中对所述加标签的运动矢量求和;

向客户端发送具有唯一标识符标签的帧,其中所述唯一标识符标签表示帧包括与玩家输入相关联的运动的时间点;以及

向客户端发送指令以从队列中移除已经具有与从服务器接收的加标签的帧相关联的标签的运动矢量;以及

其中,当服务器将编码的视频帧发送给客户端时,客户端被指示对视频帧进行解码并

将求和的运动矢量应用于解码帧以估算运动。

12.根据权利要求11所述的系统，其中所述编码的视频帧在不进行残差处理的情况下解码。

13.根据权利要求11所述的系统，所述服务器还指示客户端将一个或更多个平滑功能应用于队列中的求和的加标签的运动矢量。

14.根据权利要求11所述的系统，其中与运动矢量相关联的标签本质上是按时间顺序的。

15.根据权利要求11所述的系统，其中所述求和的加标签的运动矢量与相应的宏块相关联。

16.根据权利要求11所述的系统，其中缓存的查找表设置为基于玩家输入进行修改。

17.根据权利要求11所述的系统，其中所述求和的加标签的运动矢量设置为应用一次。

18.根据权利要求11所述的系统，其中所述求和的加标签的运动矢量描述任意复杂度的运动。

19.根据权利要求11所述的系统，其中所述求和的加标签的运动矢量针对玩家输入估算未来的反馈。

20.根据权利要求11所述的由计算机实现的方法，其中所述求和的加标签的运动矢量遵循H.264编码标准。

21.一种用于缓存运动矢量的由计算机实现的方法，包括：

在服务器处生成一个或更多个运动矢量，其中所述运动矢量基于预定的标准生成；

向客户端发送生成的运动矢量以及一个或更多个无效器，其中所述生成的运动矢量以及无效器缓存在客户端处；

向客户端发送指令以从用户接收输入，其中所述客户端设置为将输入与缓存的运动矢量或无效器匹配；以及

向客户端发送指令以应用匹配的运动矢量或无效器以实现图形界面中的运动补偿。

22.根据权利要求21所述的方法，其中所述运动矢量缓存在查找表中。

23.根据权利要求21所述的方法，其中所述一个或更多个无效器与一个或更多个缓存的运动矢量相关联。

24.根据权利要求23所述的方法，其中所述一个或更多个无效器设置为禁用所述一个或更多个缓存的运动矢量。

25.根据权利要求21所述的方法，还包括以下步骤：如果输入和与一个或更多个运动矢量相关联的缓存的无效器匹配，则指示客户端删除缓存的一个或更多个运动矢量。

26.根据权利要求21所述的方法，其中所述运动矢量是唯一的并且可预测的。

27.根据权利要求21所述的方法，其中所述运动矢量事先生成或在运行期间生成。

28.根据权利要求21所述的方法，其中所述运动矢量存储在服务器处，并且被发送给客户端以按需缓存。

29.根据权利要求21所述的方法，其中所述运动矢量是游戏生成的。

30.根据权利要求29所述的方法，其中所述游戏生成的运动矢量生成为每个像素的运动矢量并转换为每个宏块的运动矢量。

31.一种用于缓存运动矢量的系统，其中，通过网络，服务器：

生成一个或更多个运动矢量,其中所述运动矢量基于预定的标准生成;

向客户端发送生成的运动矢量以及一个或更多个无效器,其中所述生成的运动矢量以及无效器缓存在客户端处;

向客户端发送指令以从用户接收输入,其中所述客户端设置为将输入与缓存的运动矢量或无效器匹配;以及

向客户端发送指令以应用匹配的运动矢量或无效器以实现图形界面中的运动补偿。

32.根据权利要求31所述的系统,其中所述运动矢量缓存在查找表中。

33.根据权利要求31所述的系统,其中所述一个或更多个无效器与一个或更多个缓存的运动矢量相关联。

34.根据权利要求33所述的系统,其中所述一个或更多个无效器设置为禁用所述一个或更多个缓存的运动矢量。

35.根据权利要求31所述的系统,还包括以下步骤:如果输入和与一个或更多个运动矢量相关联的缓存的无效器匹配,则指示客户端删除缓存的一个或更多个运动矢量。

36.根据权利要求31所述的系统,其中所述运动矢量是唯一的并且可预测的。

37.根据权利要求31所述的系统,其中所述运动矢量事先生成或在运行期间生成。

38.根据权利要求31所述的系统,其中所述运动矢量存储在服务器处,并且被发送给客户端以按需缓存。

39.根据权利要求31所述的系统,其中所述运动矢量是游戏生成的。

40.根据权利要求39所述的系统,其中所述游戏生成的运动矢量生成为每个像素的运动矢量并转换为每个宏块的运动矢量。

41.一种用于缓存运动矢量的由计算机实现的方法,包括:

将先前生成的运动矢量库从服务器发送到客户端,其中所述运动矢量库设置为存储在客户端处;

向客户端发送指令以监视来自用户的输入数据;

向客户端发送指令以根据输入数据计算运动估算结果;以及

向客户端发送指令以基于输入数据更新存储的运动矢量库,其中所述客户端设置为应用存储的运动矢量库以在从服务器接收实际运动矢量数据之前启动图形界面中的运动。

42.根据权利要求41所述的由计算机实现的方法,还包括从服务器向客户端发送场景更新以禁用存储的运动矢量库的应用的步骤。

43.根据权利要求41所述的由计算机实现的方法,还包括发送指令以将一个或更多个缩放因子应用于运动矢量库的步骤。

44.根据权利要求3所述的由计算机实现的方法,其中基于以下一般公式计算所述缩放因子:

$$\text{播放速度缩放} = \frac{\text{缓存的动画时间}}{\text{缓存的动画时间+延迟}}$$

。

45.根据权利要求41所述的由计算机实现的方法,其中所述生成的运动矢量库由多个运动矢量组成。

46.根据权利要求45所述的由计算机实现的方法,其中所述运动矢量是游戏生成的。

47.根据权利要求41所述的由计算机实现的方法，其中所述生成的运动矢量库设置为永久存储在客户端上。

48.根据权利要求41所述的由计算机实现的方法，其中所述运动矢量库在构建过程期间生成。

49.根据权利要求41所述的由计算机实现的方法，其中所述生成的运动矢量库与来自用户的输入数据相关联。

50.根据权利要求41所述的由计算机实现的方法，其中所述指令由相关标签组成，其中所述相关标签与来自用户的输入数据相关联。

51.一种用于缓存运动矢量的系统，其中，通过网络，服务器：

将先前生成的运动矢量库发送到客户端，其中所述运动矢量库设置为存储在客户端处；

向客户端发送指令以监视来自用户的输入数据；

向客户端发送指令以根据输入数据计算运动估算结果；以及

向客户端发送指令以基于输入数据更新存储的运动矢量库，其中所述客户端设置为应用存储的运动矢量库以在从服务器接收实际运动矢量数据之前启动图形界面中的运动。

52.根据权利要求51所述的系统，还包括从服务器向客户端发送场景更新以禁用存储的运动矢量库的应用的步骤。

53.根据权利要求51所述的系统，其中所述服务器还发送指令以将一个或更多个缩放因子应用于运动矢量库。

54.根据权利要求53所述的系统，其中基于以下一般公式计算所述缩放因子：

$$\text{播放速度缩放} = \frac{\text{缓存的动画时间}}{\text{缓存的动画时间+延迟}}$$

。

55.根据权利要求51所述的系统，其中所述生成的运动矢量库由多个运动矢量组成。

56.根据权利要求55所述的系统，其中所述运动矢量是游戏生成的。

57.根据权利要求51所述的系统，其中所述生成的运动矢量库设置为永久存储在客户端上。

58.根据权利要求51所述的系统，其中所述运动矢量库在构建过程期间生成。

59.根据权利要求51所述的系统，其中所述生成的运动矢量库与来自用户的输入数据相关联。

60.根据权利要求51所述的系统，其中所述指令由相关标签组成，其中所述相关标签与来自用户的输入数据相关联。

通过预期运动矢量的玩家输入运动补偿

[0001] 相关申请

[0002] 本申请要求以下美国临时申请的利益:2017年4月21日提交的第62/488,526号、2018年2月23日提交的第62/634,464号、2018年3月9日提交的第62/640,945号和2018年3月16日提交的第62/644,164号。

背景技术

[0003] 服务器侧游戏由客户端侧玩家控制的远程游戏应用程序已尝试使用现有或定制化的编码器对来自三维(3D)图形引擎的视频输出进行实时编码。然而,视频游戏的交互性,特别是视频输出和玩家输入之间的玩家反馈环节,使游戏视频流传输比常规视频流传输对延迟更敏感。现有视频编码方法能够为减少编码时间而牺牲计算能力,而几乎别无其它方法。将编码过程集成到视频渲染过程的新方法能够提供显著减少编码时间,同时还降低了计算能力,从而提高编码视频的质量并保持原始比特流数据格式以保持现有硬件设备的互操作性。

[0004] 与常规视频播放不同,视频游戏对于视频反馈环节具有唯一的玩家输入。玩家对输入和视频输出之间的延迟非常敏感。在这样的玩家输入反馈环节中的高延迟对视频游戏流传输在服务器托管的视频游戏实例由远程玩家控制的应用中是巨大的绊脚石。任何能够减少输入和反馈之间的时间的过程都将直接改善用户体验。

[0005] 游戏流传输环境中的客户端硬件可以具有不同级别的计算性能,但是专用的H.264硬件解码器越来越普遍,甚至是在移动设备和其它低功率设备中。硬件解码器擅长执行少量计算,例如,通常根据H.264编码标准定期执行的运动补偿。无论客户端的整体计算能力如何,都能够利用专用解码硬件的优势以在游戏流传输环境中提供更好的玩家体验。

[0006] 在本地、非流式传输渲染应用中,游戏引擎能够在玩家输入和视频反馈之间添加几帧延迟。在游戏流传输应用中,因为玩家输入必须通过网络传输到远程服务器并且视频输出必须在玩家接收到反馈之前进行编码、传输和解码,所以附加延迟被引入到玩家输入-反馈循环。对于某些玩家输入,客户端能够通过立即执行运动补偿估算视频反馈的结果,从而减少网络延迟。

[0007] 玩家输入运动补偿最基本上是这样一种技术,即移动几个像素组,以便在视频游戏运行在服务器上同时由联网的客户端远程控制的情况下,牺牲一些图像准确度,减少输入-反馈的延迟。该技术擅长减少输入的玩家反馈延迟,从而引起连贯的运动矢量,例如,第一人称游戏中的玩家视图旋转。

[0008] 在视频游戏中,玩家场景定义为当前游戏状态,所述当前游戏状态是先前的玩家动作、输入和决策的结果。游戏流传输系统中的客户端对于玩家场景是简单的,即,客户端仅接收视频输出而不接收会决定特定玩家输入结果的任何游戏状态信息。存在范围广泛的输入,这些输入基于游戏状态信息产生唯一但可预测的运动结果。这些输入将会从玩家-反馈延迟的减少中受益,但是因为客户端不会具有玩家场景信息,所以不能够在客户端上预缓存这些输入以针对常规玩家输入运动补偿。另外,玩家场景置换空间可能是太小以至于

不能够为诸如缓存重复运动矢量的方法预生成运动矢量。这些系统和方法在美国临时申请第62/488,526号、第62/634,464号和第62/640,945号中描述;这三个申请的全部内容都并入本申请中。游戏服务器能够通过生成预期的运动矢量以及在玩家场景改变时更新客户端的输入运动补偿缓存来进行补偿。这允许客户端对一组有限的场景相关输入使用玩家输入运动补偿技术,从而引起输入反馈延迟的减少。

[0009] 美国专利第9,661,351号(“‘351专利”)公开用于从服务器到客户端设备传输期间跳帧的系统和方法,其中,响应于检测到跳过的帧,客户端设备生成在压缩视频流中替换跳过的帧的预测帧,该预测帧基于从由客户端设备解码的一个或更多个先前的帧扩展的增量(Delta)信息来生成。对于该技术,基于一个或更多个先前帧的数据(例如,运动矢量、残差等)在跳过的帧之后进行一个或更多个重构的或预测的帧的客户端侧的帧预测。该技术还优先考虑比特分配和/或子特征编码。编码网络抽象层单元(NALU)可以划分成(1)运动矢量和(2)残差。装置可以按优先级仅发送最小的编码数据来代替实际地跳帧。例如,如果运动已分配优先级,则可能仅发送运动矢量。本发明优于‘351专利的技术,这至少因为‘351专利未公开使用从服务器发送的查找表以将用户输入与运动矢量匹配并对这些运动矢量加标签且求和的客户端设备。‘351专利也未公开将那些求和的运动矢量应用于解码的帧以估算这些帧中的运动。本发明也是具有优势的,这是因为本发明减少了输入-反馈延迟,所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0010] 美国专利第8,678,929号(‘929专利)涉及联网的、交互游戏系统的操作。该公开的方法旨在通过双向混合确定共享全局对象的路线校正后的位置值来减少网络滞后。本专利中讨论的“混合”包括发送本地玩家在本地控制台上姿势的计算结果的网络仿真的步骤。控制台混合本地仿真和网络仿真。该方法还通过使用本地玩家的混合姿势来混合共享全局对象以在本地控制台为共享全局对象确定位置值。本发明也优于‘929专利的技术,这至少因为‘929专利未公开使用从服务器发送的查找表以将用户输入与运动矢量匹配并对这些运动矢量加标签且求和的客户端设备。‘929专利也未公开将那些求和的运动矢量应用于解码帧以估算这些帧中的运动。本发明也是具有优势的,这是因为本发明不需要存在具有极高处理能力的客户端来减少输入-反馈延迟,所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0011] 美国专利第8,069,258号(“‘258专利”)涉及使用本地帧处理来减少多玩家仿真的明显网络滞后。描述的方法包括拦截来自本地用户的输入、从先前的游戏帧确定网络仿真中的远程对象的状态数据以及从联网的仿真的部分的多游戏系统确定不确定对象的交互。该交互数据与状态数据和本地输入一起用于生成视频帧的本地仿真。以这种方式,针对单个帧,本地仿真和网络仿真能够异步运行,使每帧对应于游戏内的单个时间阶段。这允许本地仿真在联网的游戏期间实时地更新,同时保持与网络的(基本)同步。本发明也优于‘258专利的技术,这至少因为‘258专利未公开使用从服务器发送的查找表以将用户输入与运动矢量匹配并对这些运动矢量加标签且求和的客户端设备。‘929专利也未公开将那些求和的运动矢量应用于解码的帧以估算这些帧中的运动。本发明也是具有优势的,这是因为本发明不需要存在具有极高处理能力的客户端来减少输入-反馈延迟,所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0012] 美国专利第9,665,334 B2号(“‘334专利”)公开用于渲染协议的系统和方法,所述

协议应用多个过程以及合成器在显示器上渲染结合的图像。该技术如以下操作:当服务器同时向几个客户端设备提供游戏屏幕时,服务器上的渲染处理的计算负载变繁重,例如,需要快速响应的游戏内容。即,服务器能够向其提供屏幕的客户端设备的数量受限于服务器的渲染性能和所需的响应。相反,当每个客户端设备被控制以执行能够通过一般渲染性能在服务器与客户端设备之间共享渲染过程的处理时,能够向更多个客户端设备提供屏幕。而且,通常,没有应用纹理映射而被渲染的游戏屏幕具有高压缩效率,并且能够经由诸如互联网的网络以更小的带宽被发送。本发明优于‘334专利讨论的技术,这至少因为‘334专利未公开基于预定义标准在服务器处生成运动矢量,并将生成的运动矢量以及一个或更多个无效器(invalidator)发送给客户端,所述客户端缓存这些运动矢量和无效器。‘334专利也未公开服务器指示客户端接收来自用户的输入并使用该输入与缓存的运动矢量或无效器匹配,其中这些矢量或无效器用于运动补偿中。本发明也是具有优势的,这是因为本发明减少了输入-反馈延迟,所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0013] 美国专利第9,736,454号(“‘454专利”)公开用于编码的系统和方法,该系统和方法包括检查与纹理块并置的深度块的可用性,基于并置的深度块的可用性确定针对纹理块的预测方法,以及基于并置的深度块推导出纹理块的第一预测块。本发明也优于‘454专利讨论的技术,这至少因为‘454专利未公开基于预定义标准在服务器处生成运动矢量,并将生成的运动矢量以及一个或更多个无效器发送给客户端,所述客户端缓存这些运动矢量和无效器。‘454专利也未公开服务器指示客户端接收来自用户的输入并使用该输入与缓存的运动矢量或无效器匹配,其中这些矢量或无效器用于运动补偿中。本发明也是具有优势的,这是因为本发明减少了输入-反馈延迟,所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0014] 美国专利第9,705,526号(“‘526专利”)公开用于在媒体和图像应用中进行熵编码的系统和方法。该技术公开一种系统,其中如所指示的,压缩从接收图像和/或视频数据的源开始。然后应用无损压缩方案。预测器/增量计算单元随后接受输入并尝试使用相邻输入元素之间的增量计算来减少输入数据中的冗余。然后,这些值使用熵编码器中的预定义统计模型进行编码以生成压缩图像和/或视频数据。与以上专利类似,本发明优于‘526专利讨论的技术,这至少因为‘526专利未公开基于预定义标准在服务器处生成运动矢量,并将生成的运动矢量以及一个或更多个无效器发送给客户端,所述客户端缓存这些运动矢量和无效器。‘526专利也未公开服务器指示客户端从用户接收输入并使用该输入与缓存的运动矢量或无效器匹配,其中这些矢量或无效器用于运动补偿中。本发明也是具有优势的,这是因为本发明减少了输入-反馈延迟,所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0015] 美国专利第8,873,636B2号(“‘636专利”)涉及向运行游戏本地实例的用户电脑提供编码图像数据的运动图像分发服务器(例如,一个运行在线游戏的服务器)。为了执行该过程,在相关的细节中,用户电脑的CPU指定待参考的区域,以便在先前的帧屏幕中解码与选择的块相关联的运动矢量。该操作通过参考与选择的块相关联的运动矢量(选择的块提供的预处理信息中包括的矢量)完成,并提取该区域的图像作为参考图像。与其它参考文献一样,本发明优于‘636专利讨论的技术,这至少因为‘636专利未公开基于预定义标准在服

务器处生成运动矢量，并将生成的运动矢量以及一个或更多个无效器发送给客户端，所述客户端缓存这些运动矢量和无效器。‘636专利也未公开服务器指示客户端从用户接收输入并使用该输入与缓存的运动矢量或无效器匹配，其中这些矢量或无效器用于运动补偿中。本发明也是具有优势的，这是因为本发明减少了输入-反馈延迟，所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显减少。

[0016] 国际专利公开WO2009138878 A2(“‘878公开”)涉及在集中化流传输应用服务器中处理并流传输多个交互应用，同时控制细节的层次以及各渲染对象的后过滤。在该系统中，集中化交互式应用服务器，在其视频预处理器，在对向客户端设备发送的视听内容的压缩流进行编码之前对帧序列执行空间和时间的过滤，所述客户端设备对压缩流进行解码并显示该内容。集中化交互式应用服务器的GPU命令处理器包括还对于视频编码器中的目标编码帧中的每个宏块计算运动补偿估算结果的模块。仍然，本发明优于‘878公开讨论的技术，这至少因为‘878未公开基于预定义标准在服务器处生成运动矢量，并将生成的运动矢量以及一个或更多个无效器发送给客户端，所述客户端缓存这些运动矢量和无效器。‘878公开也未公开服务器指示客户端从用户接收输入并使用该输入与缓存的运动矢量或无效器匹配，其中这些矢量或无效器用于运动补偿中。本发明也是具有优势的，这是因为本发明减少了输入-反馈延迟，所述输入-反馈延迟通过使用玩家输入运动补偿而不是等待服务器返回输出视频而显著减少。

[0017] 美国专利第9,358,466 B2号(“‘466专利”)涉及通过缓存数据的重复使用来提高视频游戏的性能。该系统公开至少部分地通过识别使用的数字资产并确定识别的数字资产是否存在于缓存中来对不同的生成的视频游戏任务的缓存性能进行评分。能够基于与已存在于缓存中的任务的数字资产的比例相应的缓存重复使用率能够计算缓存得分。其它用于生成缓存得分的技术可能考虑以下因素：例如，已存在于缓存中的任务的组合的数字资产的整体大小和/或不存在于缓存中的任务的组合的数字资产的整体大小。通过以这种方式来缓存数据，该数据以及未缓存的数据请求变得更有效。本发明仍然优于‘466专利讨论的技术，这至少因为‘466专利未公开缓存重复运动矢量、根据输入数据计算运动估算结果或基于输入数据更新存储的运动矢量库，使得客户端能够使用存储的运动矢量库在从服务器接收实际的运动矢量数据之前启动运动。

[0018] 美国专利第6,903,662 B2号(“‘662专利”)涉及可配置计算机输入设备，特别是缓存输入数据以保持更快的响应时间的一种可配置计算机输入设备。该系统通过检查对内部存储器(或缓存)的键按压来确定该特定键是否先前已被识别来进行操作。如果系统在此之前尚未与键通信，该系统可以随后对应于来自系统键存储器的输入检索数据。系统将会随后将键标识和其相应的数据更新到系统的内部存储器(缓存)。系统可以随后向主机计算机发送来自键的输入数据。然而，下次系统遇到处于按压状态的相同键时，系统能够查询系统自身的存储器(缓存)而不是再次从键存储器检索相同的扫描代码。本发明优于‘662专利讨论的技术，这至少因为‘662专利未公开缓存重复运动矢量、根据输入数据计算运动估算结果或基于输入数据更新存储的运动矢量库，使得客户端能够使用存储的运动矢量库在从服务器接收实际的运动矢量数据之前启动运动。

[0019] 日本专利JP6129865B2(“‘865专利”)公开为路径段的子集向游戏客户端发送渲染的游戏内容数据以在本地缓存上缓存的系统和方法，使得在实时游戏播放期间在需要时可

使用该游戏内容数据。再次，本发明优于‘865专利讨论的技术，这至少因为‘865专利未公开缓存重复运动矢量、根据输入数据计算运动估算结果或基于输入数据更新存储的运动矢量库，使得客户端能够使用存储的运动矢量库在从服务器接收实际的运动矢量数据之前启动运动。

[0020] 美国专利第9,762,919号(“‘919专利”)公开用于在块处理管线中缓存参考数据的系统和方法。‘919专利技术公开可以例如在SRAM(静态随机存取存储器)之类的本地(对于管线)存储器中实现的缓存存储器(例如，全相联缓存)，对于该缓存存储器，可以从该存储器预取与为宏块在管线的早期阶段确定的运动矢量相应的色度参考数据的部分(例如，64字节存储器块)。色度缓存逻辑可以保持缓存，并可以在管线的多个阶段扩展。对经过管线的给定宏块的运动矢量的提取可以通过在色度运动补偿阶段之前的一个或更多个阶段的色度缓存逻辑来启动，以提供时间(即，多个管线周期)以在色度运动补偿需要相应存储器块之前将该相应存储器块从存储器读取到缓存中。然而，‘919专利与本发明相比仍然不足。本发明优于‘919专利讨论的技术，这至少因为‘919专利未公开缓存重复运动矢量、根据输入数据计算运动估算结果或基于输入数据更新存储的运动矢量库，使得客户端能够使用存储的运动矢量库在从服务器接收实际的运动矢量数据之前启动运动。

[0021] 从对本技术的现有技术的以上讨论中显而易见的是，在本领域中需要对与实时游戏环境的编码有关的现有计算机技术进行改进。

发明内容

[0022] 因此，本发明的目的是公开通过运动补偿技术来减少延迟的系统和方法，其中客户端设备使用从服务器发送的查找表以将用户输入与运动矢量匹配并对这些运动矢量加标签和求和。当远程服务器将编码的视频帧发送到客户端时，客户端解码这些视频帧并将求和的运动矢量应用于解码的帧以估算这些帧中的运动。

[0023] 本发明的另一目的是公开其中编码的视频帧在不进行残差处理的情况下解码的系统和方法。

[0024] 本发明的又一目的是公开通过运动补偿技术减少延迟的系统和方法，其中客户端对队列中的求和的加标签的运动矢量应用一个或更多个平滑功能。

[0025] 本发明的又一目的是公开通过运动补偿技术减少延迟的系统和方法，其中与在客户端设备处的运动矢量相关联的标签实际上是按时间顺序的。

[0026] 本发明的另一目的是提供系统和方法，所述系统和方法通过基于预定义标准在服务器处生成运动矢量，并将生成的运动矢量以及一个或更多个无效器传输给客户端以减少输入-反馈延迟，所述客户端缓存这些运动矢量和无效器。服务器指示客户端接收来自用户的输入并使用该输入以匹配缓存的运动矢量或无效器。基于比较结果，客户端随后应用匹配的运动矢量或无效器以实现图形界面中的运动补偿。

[0027] 本发明的另一目的是提供通过在查找表中缓存运动矢量来减少输入-反馈延迟的系统和方法。

[0028] 本发明的又一目的是提供通过将无效器与一个或更多个缓存的运动矢量关联来减少输入-反馈延迟的系统和方法。

[0029] 本发明的又一目的是提供系统和方法，所述系统和方法通过在输入和与一个或更

多个运动矢量相关联的缓存的无效器匹配时指示客户端删除一个或更多个缓存的运动矢量来减少输入-反馈延迟。

[0030] 本发明的另一目的是公开通过在向客户端发送先前生成的运动矢量库的服务器处缓存重复运动矢量来减少延迟的系统和方法。客户端存储运动矢量库并监视用户输入数据。服务器指示客户端根据输入数据计算运动估算结果并指示客户端基于输入数据更新存储的运动矢量库，使得客户端应用存储的运动矢量库以在从服务器接收到实际运动矢量库之前在图形界面中启动运动。

[0031] 本发明的另一目的是公开通过缓存重复运动矢量来减少延迟的系统和方法，其中服务器向客户端发送场景更新以禁用存储的运动矢量库的应用。

[0032] 本发明的另一目的是公开通过缓存重复运动矢量来减少延迟的系统和方法，其中将一个或更多个缩放因子应用于运动矢量库。

附图说明

[0033] 当结合附图考虑时，通过参考以下详细描述会更好地理解本发明，并且将容易获得对本发明及其许多伴随优点的更全面的理解，附图中：

[0034] 图1是示例性地示出远程游戏系统的框图，其中视频游戏在服务器上运行，同时受远程客户端处的输入控制；

[0035] 图2是示例性地示出通过对合适的玩家输入应用运动补偿来减少游戏流传输应用中的输入-反馈延迟的流程图；

[0036] 图3是示例性地示出利用玩家输入运动补偿的视频游戏流传输环境的运行期间的示例性时刻的框图；

[0037] 图4是示例性地示出图3的玩家输入运动补偿期间的示例性宏块的图；

[0038] 图5是示例性地示出在客户端上在玩家输入运动补偿期间应用运动矢量的可替代方法的图；

[0039] 图6A、6B和6C示出关于图5中所示的可替代方法进行玩家输入运动补偿和混合的示例性宏块。

[0040] 图7是示出在运行时生成预期的运动矢量的图；

[0041] 图8是示例性地示出以玩家输入运动补偿为目的的预期运动矢量的传输以及客户端侧的存储的流程图；

[0042] 图9是示出使预期运动矢量无效的示例性过程的图；

[0043] 图10是示出生成运动矢量库的示例性方法和用于缓存目的的示例性重复运动矢量库的图；

[0044] 图11是示例地示出缓存、应用以及更新运动矢量库以进行玩家输入运动补偿的过程的流程图；

[0045] 图12是示例性地示出运动矢量映射可以如何更新的图；以及

[0046] 图13是示出在玩家输入运动补偿期间应用多帧运动矢量，特别是在缓存的运动矢量的情况下的示例性修改的图。

具体实施方式

[0047] 在描述附图中所示的本发明的优选实施例时，为了清楚起见，将采用特定的术语。然而，本发明不旨在限于如此选择的特定术语，并且应当理解，每个特定术语包括以相似方式操作以实现相似目的的所有技术等同物。为了说明的目的，描述了本发明的几个优选实施例，应该理解，本发明可以以未在附图中具体示出的其它形式来实施。

[0048] 某些类型的玩家输入对玩家输入运动补偿是更好的选择。两个因素有助于给定输入的适用性：玩家对输入-反馈延迟的敏感性，以及在不引入显著伪影的情况下实施玩家输入运动补偿的难度。将需要对每个输入进行适用性评估。例如，在第一人称射击游戏中，玩家将对基于鼠标的视角旋转非常敏感，玩家输入和视频输出之间的小到如16ms的不到1秒的延迟将会是明显的。然而，在同样的情况下，基于游戏手柄控制器的视角旋转通常较慢，并且玩家可能对输入-反馈延迟不那么敏感。能够通过沿旋转的相反方向移动场景来近似视角旋转，但是在旋转方向上沿着图像的边缘可能会出现不希望的伪影。对于小视角旋转，例如，调节对屏幕上敌人的瞄准，玩家可能甚至不会注意到边缘伪影。在另一示例中，对于玩家输入运动补偿，由于缺少玩家对延迟的敏感性和/或惯性，在驾驶游戏中加速车辆可能是低优先级的，但是转向和制动输入可能是高优先级的，这是因为玩家将会注意到输入-反馈延迟。

[0049] 接收玩家输入和显示运动输出之间的时间是玩家-反馈延迟。通过使用运动补偿，估算的运动几乎能够在等待服务器处理玩家输入时立即提供反馈。通过这种方式，玩家-反馈延迟在游戏流传输应用中显著减少。通过在游戏流传输应用中实施用户输入运动补偿，能够在下一可用帧中提供运动估算。相反，需要几帧来将输入发送到服务器、生成输出帧以及返回。玩家输入运动补偿还可以在游戏引擎和渲染器可能具有几帧的玩家-反馈延迟的常规非流式传输游戏应用中提供一些好处。

[0050] 客户端将不会具有合适的场景以跟踪对象在屏幕四周的移动。玩家输入运动补偿将不适用于客户端不知道特定宏块或视频对象的位置的情况。例如，在2D平台游戏上，角色能够在屏幕四周从左向右移动。当玩家按压跳跃输入时，客户端将不会知道角色位于何处，因此，在这种情况下，玩家输入运动补偿不能够单独使用来减少输入-反馈延迟。

[0051] 通常，用于玩家输入运动补偿的运动矢量应当提前生成。对于诸如玩家摄像头旋转的运动，能够基于游戏如何对输入加权来计算运动矢量。在某些实施例中，运动矢量可以是输入值乘以敏感度权重。对于诸如动画运动的不能够直接计算的运动，可以在开发期间触发动画使得运动矢量可以被直接测量和存储。能够通过在H.264编码期间执行的相同的运动估算技术完成运动矢量的测量。

[0052] 图1示出视频游戏由远程客户端控制的示例性系统。在该系统中，服务器100托管视频游戏软件102和渲染视频输出的图形引擎104。视频在编解码器(也称为编码引擎或编码器)106中被编码并且已编码的视频数据108被发送到远程客户端。服务器架构100可以是能够支持图形引擎104和编解码器106两者功能的硬件或软件的任何组合。在给定的示例中，图形引擎104可以实现为例如由加载到某些计算机可读存储器112中的视频游戏软件102控制的GPU 110，而编解码器106可以实现为运行视频编码软件的CPU 114。

[0053] 远程客户端包括能够运行客户端侧编解码器118以解码发送的编码的视频数据108的计算机系统116和应用玩家输入运动补偿的客户端应用程序120。客户端计算机系统

116还包括显示控制器122以驱动显示硬件124。来自客户端侧的输入外围设备126的输入将由客户端应用程序120转换为控制数据128，并将所述控制数据发送回服务器100上运行的游戏软件102。来自外围设备126的输入还将用于如图2更详细地示出地确定应用什么玩家运动补偿(如果存在)。

[0054] 图2是描述针对单个输入执行玩家输入运动补偿所需的步骤的流程图。在步骤200，在客户端初始化时，服务器发送包括输入和与输入相关联的运动矢量的查找表，然后在步骤202，由客户端缓存所述查找表。在该实施方式中，客户端通用于满足多个流传输游戏的需求。在某些实施例中，游戏专用客户端能够跳过步骤200和202，因为该游戏专用客户端已经包括用于游戏的查找表。在可替代实施方式中，游戏专用客户端可以永久存储运动矢量查找表，而不需要从服务器缓存。

[0055] 当在步骤204客户端从诸如鼠标或游戏手柄控制器的输入设备接收玩家输入时，在步骤206，客户端应用程序将检查缓存的运动矢量查找表以匹配输入。如果没有匹配的玩家输入，则客户端将不进行任何附加动作并将输入发送到服务器而无需进行附加修改。如果缓存中有匹配的玩家输入，则客户端将应用玩家输入运动补偿。可选地，缓存可能能够基于玩家输入更改查找表中的条目。例如，当玩家按下暂停按钮，所有玩家移动输入应当被禁止直到玩家退出暂停屏幕。在一种实施方式中，客户端控制的查找表可以具有两组输入，一组输入在暂停菜单中使用，一组输入在暂停菜单外使用，每当玩家选择暂停输入时，所述两组输入优选地由客户端切换。在可替代的实施方式中，服务器可以切换客户端上缓存的查找表的内容。

[0056] 在步骤208，当客户端应用程序接收用于运动补偿的玩家输入时，客户端将给玩家输入和与玩家输入相关联的运动矢量加标签。在步骤210，向服务器发送加标签的输入。标签是任何能够将玩家输入关联到未来帧的标识符。例如，标签可以是整数，每当客户端接收将被用于执行玩家输入运动补偿的输入时，该整数就会递增。标签能够作为元数据添加到与玩家输入相同的网络数据包中或以使标签信息保持与输入信息同步的相似的消息传递模式发送。在步骤213，客户端确认是否接收到加标签的标签。在步骤212，当对玩家输入加标签并发送该玩家输入时，客户端应用包括在缓存的查找表中的运动矢量。这些运动矢量将被应用于每个传入帧直到从服务器返回相关的标签。图3示出了应用这些运动矢量的示例性方法的详细描述。

[0057] 在步骤214，当服务器接收到加标签的玩家输入，加标签的玩家输入被传递给游戏，该游戏生成输出帧。然后，在步骤216，对视频图像进行编码。在步骤218，在编码帧被发送回客户端之前，给编码帧附加玩家输入标签。这是与先前与玩家输入一起发送的帧相同的帧，并且表示输出帧包括来自玩家输入的实际视频反馈。能够通过将标签作为元数据添加到与编码帧相同的网络数据包来完成给编码帧附加标签。在步骤220，将加标签的编码帧发送回客户端。当客户端接收到具有标签的编码帧时，客户端能够使标签与先前的玩家输入运动补偿关联。然后，在步骤222，客户端停止应用先前的运动补偿。

[0058] 图3是利用玩家输入运动补偿的视频游戏流传输环境的运行期间的示例性时刻的图示。当在步骤300客户端接收到任何玩家输入时，在步骤302，将其与运动矢量查找表进行比较。如果存在匹配玩家输入，则使用相关联的运动矢量来进行玩家输入运动补偿。在步骤306，利用在步骤304的唯一的输入标签对运动矢量加标签。在该示例中，所选的标签是整数

“1003”。在步骤308，将加标签的运动矢量添加到队列，所述队列包括当前正用于玩家输入补偿的任何其它加标签的运动矢量。

[0059] 在步骤320，下一帧到达来自服务器的比特流中。在步骤318，为帧添加具有唯一标识符的标签，在这种情况下是整数“1001”，所述整数表示包括直到并且包括对应于标签“1001”的输入的所有先前玩家输入的合成运动的帧。在步骤322，标签“1001”指示客户端其能够停止为该加标签的输入应用运动补偿。然后，在步骤308，将具有标签“1001”的运动矢量与在先前数据包已经丢失的情况下可以保留在队列中的具有较早标签的任何运动矢量一起从加标签的运动矢量队列中移除。

[0060] 在步骤324，对编码的视频316进行解码。同时，在步骤310，对在步骤308的运动矢量队列中的其余运动矢量求和。运动矢量通常是具有针对图像中的每个宏块的矢量的矢量场。为了对运动矢量求和，对矢量逐元素求和，使得结果是具有针对每个宏块的矢量的矢量场。两个矢量场的总和是该场中每个点的矢量的总和，因此两组运动矢量的总和是图像中每个宏块的矢量总和。两个矢量的总和定义为两个矢量的分量的逐分量求和，可以表示为：{u1,u2}+{v1,v2}＝{u1+v1,u2+v2}。在该示例中，具有标签“1002”和“1003”的两组运动矢量包括在队列中；对这两组运动矢量求和。标签本质上是按时间顺序的，这允许客户端知道先前加标签的玩家输入的顺序。这让客户端丢弃直到并且包括传入帧中的返回标签的加标签的运动矢量。另外，上述加标签在计算上比更复杂的方法便宜。在这个时刻能够应用可选的平滑功能以阻止钳制(clamping)伪影或减轻类似引入的伪影。

[0061] 当宏块移出屏幕并表现为垂直于屏幕边缘的像素颜色模糊时，会引入钳制伪影。随着指向外的矢量越来越接近图像的边缘，示例性平滑功能会减少这些矢量的强度。只有向外的分量需要被弱化，指向外的矢量的y分量朝向图像的顶部边缘和底部边缘，指向外的矢量的x分量朝向图像的左侧边缘和右侧边缘。对于指向边界的矢量，矢量分量可以乘以与边界的距离的平方，使得当与边界的距离接近零时，矢量分量将接近零。在该示例中，朝向图像右侧边缘的指向外的矢量将从{x,y}变换为{x*d*d,y}，其中d是与边缘的距离。这将减轻钳制伪影，但换来图像边界附近的一些轻微的图像失真。对于玩家，这种失真远不及钳制伪影明显。

[0062] 当在步骤324对编码视频帧的解码过程完成之后，在步骤312使用求和的运动矢量进行运动补偿。在步骤314，输出结果视频。该输出包括玩家输入运动补偿数据并将会将该输出显示在客户端上。该输出的帧是包括针对具有相关标签“1003”的输入的运动估算的第一帧。该输出的帧还包括针对具有相关标签“1002”的先前输入的运动估算，对于该运动估算，客户端仍然在等待服务器返回实际运动矢量。该输出的帧还是包括具有相关标签“1001”的先前输入的实际运动矢量的第一帧，客户端先前已对其估算运动。结果，在这个时间在该方法中存在三种运动估算状态：针对新输入的一个新的估算状态；等待实际结果的一个继续的估算状态；以及由于实际结果已经到达客户端而停止的另一估算状态。

[0063] 图4是示出图3的玩家输入运动补偿步骤期间的示例性宏块的图。玩家输入运动补偿步骤在程序上类似于在H.264解码期间执行的过程，在H.264解码中，解码的运动矢量用于确定当前帧中的宏块移自先前帧中的何处。对于玩家输入运动补偿，应用的运动矢量将通过在向玩家显示输出视频之前移动当前帧中的宏块来针对玩家输入估算未来的反馈。图4A示出在步骤400的队列中的两组加标签的运动矢量，所述两组加标签的运动矢量来自图3

中在步骤308所示的示例性加标签的运动矢量。通过针对图像中的每个宏块进行矢量求和以对所述两组求和，以创建图4B中在步骤402的一组运动矢量。在该组中的运动矢量中的每一个代表针对当前帧中的每个宏块的估算的运动。针对在步骤402的求和的运动矢量中的每个矢量，将移动图4C的示例性图像中的相应宏块。在步骤404，图4C中所示的一个示例性宏块以其对应的玩家输入运动矢量移位。在示例性图像中的每个宏块将以这种方式移位。图4C中的示例性图像仅包括50个宏块，而高分辨率图像将包括上千宏块。图4中所示的示例性运动矢量示出一致的刚性运动，但是所述的玩家输入运动补偿技术能够与任意复杂度的运动矢量一起使用以描述本领域已知的屏幕空间中的旋转、振动或其它复杂运动。

[0064] 图5是在客户端上在玩家输入运动补偿期间应用运动矢量的可替代方法的图示；在该方法的某些实施例中，不必在视频的编码和解码期间处理残差。当在查找表中发现匹配的玩家输入之后，在步骤500对相关联的运动矢量加标签，并将其用于图2中的步骤208和212中所示的玩家输入运动补偿。在对下一帧的解码过程期间，将玩家输入运动矢量添加到解码的运动矢量，并且注入到H.264编码标准中定义的在步骤502的运动补偿步骤中。在步骤504应用由H.264编码标准定义的解块过滤器，但是应当仅应用于还未被玩家输入运动补偿修改的块。在步骤506示出所产生的输出，该输出包括玩家输入运动补偿并将会显示在客户端上。随后在步骤506和步骤518将输出缓存并且该输出变成对下一帧的运动补偿中使用的先前图像。与图3中的实施方式不同，在该实施方式中，加标签的玩家输入矢量仅应用一次。这意味着加标签的玩家输入运动矢量不需要与队列中的所有加标签的运动矢量一起求和，因为在步骤518所示的先前图像将包括先前的加标签的运动矢量的总和。接收输入和显示视频输出之间的时间是输入-反馈延迟，通过使用玩家输入运动补偿而不是等待服务器返回输出视频，所述输入-反馈延迟将显著地减少。

[0065] 在与步骤500同时，客户端将向服务器发送相应的加标签的玩家输入，如图2中步骤208和210中所示。如图2中的步骤220中所示的一样，客户端最终将在步骤512从服务器接收比特流中具有相同标签510的如步骤508所示的编码视频，如图2中的步骤220所示。与视频相关联的标签510表示由玩家输入运动补偿先前估算的实际运动被示出于如在步骤508所示的编码视频中。如通过H.264编码标准定义的，编码的视频通过在步骤514的熵解码以及在步骤516的逆量化和逆变换传递。

[0066] 已经在先前的段落中描述的步骤508、510、512、514和516之前执行的先前的玩家输入运动补偿步骤已经对宏块进行移位，即来自编码帧的实际运动矢量正尝试移位。因此，直接应用实际运动矢量将移位不正确的宏块。比特流将包括两个相关的数据：编码的视频帧和相关标签。编码帧被发送以在步骤514进行熵解码，同时标签先前被发送以在步骤520进行混合过程。为了补偿，在步骤520的混合过程通过计算传入的实际运动矢量和先前使用的具有匹配的相关标签的玩家输入运动矢量500之差来提供校正运动矢量。该差能够通过将针对相关标签的逆运动矢量和实际运动矢量相加来计算，并且结合图6更详细地进行描述。在步骤502，使用校正矢量代替解码的运动矢量进行运动补偿。通过在步骤504的解块过滤器、在步骤506的下一输出视频帧以及在步骤518的对输出的缓存，解码管线的其余部分照常继续。针对每帧这些步骤可以无限地继续重复。通常，步骤520的混合过程发生在逆量化(步骤514)和逆变换(步骤516)之后，这是因为实际的运动矢量直到此刻才可用。一旦实际运动矢量可用，混合过程将使用所述运动矢量计算实际的运动矢量和估算的运动矢量之

差。

[0067] 图6示出经历玩家输入运动补偿和混合的示例性宏块。在0ms的时间，玩家按压输入以向左旋转摄像头视角。在下一帧向所有宏块应用来自查找表的相关联的玩家输入运动补偿运动矢量(如“PIMC”600所示)。图6A示出正在向右移的示例性宏块。玩家输入运动补偿的结果出现在下一解码的帧中，对于以每秒60帧运行的视频引起16ms(一帧的长度)的最大输入-反馈延迟。在其它的实施例中，对于以每秒30帧运行的视频，最大输入-反馈延迟可以是33ms。因此可以想到通过将最大输入-反馈延迟限制到一帧的长度，本发明跨越各种帧速率进行操作。如图6B所示，当加标签的视频从服务器返回时，该加标签的视频包括由服务器编码的实际的运动矢量602。对于这个示例，编码的视频在时间100ms返回，但是这将很大程度上取决于网络延迟。实际矢量602是指在玩家输入运动补偿600期间已经移位的宏块，因此实际矢量602不能直接应用于现有帧。相反，需要通过找到实际矢量602和玩家输入运动矢量600之差来计算校正矢量604。该差能够通过将逆玩家输入运动矢量与实际运动矢量相加来计算。找到这些矢量差是指在图5中的步骤524处的混合。校正矢量604越小，玩家输入运动补偿方法在估算实际运动矢量方面越成功。图6C中所示的针对示例性宏块的结果运动608与实际运动矢量602相同。该示例示出玩家输入运动补偿能够如何针对玩家输入估算视频反馈并显示出时间为16ms到100ms的结果，而未经修改的系统在接收到玩家输入之后的116ms内不会显示任何玩家-反馈。

[0068] 在开发期间，游戏开发人员将需要决定哪些运动和动画将在运行期间发送预期的运动矢量。唯一但可预期的运动矢量是预期运动矢量的最佳选择。明确的示例将包括由引擎适应性地更改的动画，例如，使用运动学方程式计算关节角度的动画、经过时间扭曲的动画或以其它方式拉伸或压缩的动画。例如，当玩家进入定义的壁范围内并跳跃时，爬壁(ledge grab)动画就会播放。爬壁动画拉伸，使得玩家的手在壁上，但是仍然附接到玩家的身体。动画以定义的帧数播出，将玩家置于壁之上。通过一些可接受的位置和定向，该示例的起始点是可变的。因为确切的动画事先未知，但是通过游戏引擎按需可编程地生成，所以对于生成预期的运动矢量，动画是理想选择。因为客户端不具有任何关于游戏流传输环境中的玩家位置的场景信息，所以特别地客户端不可能知道针对该动画的运动矢量。

[0069] 预期的运动矢量将仅在有限的场景或时间范围可用，例如，特定的摄像头位置、小的时间窗口，或一些其它的特定玩家场景。对于每个预期的运动矢量的集合，需要生成相应的无效器。无效器可以在客户端上使用以防止预期的运动矢量在有效之后再应用。在某些实施例中，无效器可以是将改变游戏场景使得播放预期运动矢量不再可行的任何玩家输入的集合。在其它实施例中，无效器可以是时间窗口，对于该时间窗口，预期的运动矢量可以是有效的。在其它的实施例中，无效器可以是无效输入和时间窗口的组合。例如，针对爬壁动画生成的预期运动矢量仅在有限的玩家位置和定向内有效，因此，无效器将必要地包括任何平移或旋转移动输入。在预期的运动矢量特征的开发期间将需要设计和实现无效器。如以下结合图10至图13所述，预期的运动矢量也可以由从服务器发送的事件或消息来禁用或更新，该图10至图13描述了使用缓存的重复运动矢量进行运动补偿。

[0070] 预期的运动矢量可以提前生成，或可以在运行期间按需生成。例如，对于具有有限数量的置换的动画，能够通过触发每个置换并记录运动矢量离线生成预期的运动矢量。在某些实施例中，对于通用客户端，运动矢量将存储在服务器侧并随后发送到客户端以按需

缓存。当运动矢量被发送到客户端时，将该运动矢量缓存在查找表中。预生成的预期的运动矢量可以存储在服务器并作为游戏可读文件格式可用，该游戏可读文件格式允许在游戏运行期间服务器发送预生成运动矢量以缓存在客户端上的查找表中。因为可能不存在离散数量的可能的动画置换，所以不能够预生成在运行期间生成的动画，例如，通过逆运动学计算的动画。逆运动学是在实时渲染中使用以使动画适合于边界条件集合内的常用方法。例如，视频游戏中的玩家角色想要抓住附近的壁，边界条件将通过玩家的手碰到壁的位置来定义，并且爬壁动画将通过逆运动学相应地改变。对于诸如这些的适应性更改的动画，在运行期间，游戏可以推测性地将可能的动画渲染成屏幕外的运动矢量图像并按需记录预期的运动矢量。例如，如果玩家在可抓住的壁附近，游戏可以预期玩家将很快抓住壁，并且游戏可以推测性地渲染爬壁动画以生成预期的运动矢量。需要在运行期间生成预期运动矢量的适应性更改的动画将需要开发人员提前识别。

[0071] 描述诸如玩家位置追踪、脚本系统、触发器音量或寻路系统之类的玩家场景的现有游戏系统能够用于生成事件，当游戏需要推测性地渲染动画时，该事件将发送信号。例如，游戏可以追踪玩家对可抓住的壁的接近度，并向游戏发送信号以推测性地渲染爬壁动画并记录预期的运动矢量。像捡起武器、拉动操纵杆或按下按钮的某些动画可以基于玩家对交互的接近度和玩家定向来拉伸或调节。这些动画具有太多置换以至于使预生成不可行，但是还是能够在运行期间生成(如图7示例性地示出)。能够离线生成并记录每次以相同方式播放的运动。这些运动通常是每次被触发时以相同速率发生在相同屏幕空间中的运动。能够通过触发动画的所有可能的置换并记录游戏生成的运动矢量或通过诸如在H.264编解码器中使用的更常规的运动估算技术生成运动矢量来离线记录这些动画的运动矢量。在优选的实施例中，如上结合图1至图6所述的游戏生成的运动矢量用于确保高质量的运动估算。此过程能够在开发期间的任何时刻进行，但是优选将此过程添加为构建过程或某些其它现有资产调节过程的一个阶段，例如预生成的纹理映射和多细节层次(“LOD”)。资产调节可以包括将游戏资产从人类可读资源格式编译成机器可读格式的任何过程。例如，可以通过将艺术家创建的纹理文件转换成包含多种分辨率的游戏可用格式来事先生成纹理映射。类似地，可以通过将艺术家创建的模型文件转换成包含多细节层次的游戏可用格式来事先生成LOD。能够将运动矢量生成添加到现有的资产调节过程中，该过程将艺术家生成的动画格式转换成游戏可用格式。

[0072] 图7示出离线或运行情况下生成运动矢量的示例性方法。在步骤700，能够将动画渲染为屏幕外的表面/图像并且能够记录运动矢量以供立即使用。仅需要渲染屏幕上移动的部分，而能够忽略场景中的其它对象。在步骤702示出的虚线对象和在步骤704示出的实线对象分别表示动画对象在先前帧和当前帧中的位置。在步骤706，将以在步骤708所示的运动矢量的形式捕获从先前帧到当前帧的移动。运动矢量可以从游戏生成的运动矢量捕获或通过诸如H.264编解码器中使用的更常规的运动估算技术来捕获。在优选的实施例中，使用如上文结合图1至图6所述的游戏生成运动矢量来确保高质量的运动矢量。通过重复步骤700至708所示的过程，能够针对给定动画快速计算出几个帧的有价值的预期的运动矢量，直到为所有所需的帧生成运动矢量为止。并不是动画中的所有帧都需要生成，仅需要在视频流赶上的同时生成足以在客户端上播放的帧。生成的帧的最小数量将取决于发送玩家输入和在客户端上接收产生的视频流之间的延迟；并且动画的所生成部分的长度应当至少与

延迟一样长。如下文结合图10至13所述，预期的运动矢量帧可以在播放期间进行速率缩放，所述图10至13讨论运动估算中的缓存的重复运动矢量的使用。如果预期的运动矢量帧的播放进行了速率缩放，针对等于延迟的动画部分生成预期的运动矢量将导致速率缩放为50％的播放。针对更长的动画部分生成运动矢量将导致速率缩放不太显著的播放。

[0073] 当重新编码运动矢量时应当考虑用于视频编码的宏块尺寸，并且针对每个宏块应当具有运动矢量。在优选实施例中，游戏生成的运动矢量被生成为每个像素的运动矢量，并且通过找到每个像素的运动矢量的每个宏块组的算数平均值转换成每个宏块的运动矢量。

[0074] 图8示出以玩家输入运动补偿为目的的预期的运动矢量的发送以及客户端侧的存储。在视频游戏软件的开发期间，需要配置事件以发出即将到来的输入驱动的对场景敏感的动画。例如，游戏开发人员希望发送预期的运动矢量，使得该运动矢量在玩家执行爬壁动画时可用。开发人员实施每当玩家面对可抓住壁并在该可抓住壁的范围内时会触发的事件。在该示例中，在玩家在玩游戏时接近可抓住的壁的情况下，在步骤800“在游戏运行期间接收事件”接收该示例性事件。事件类型在适应性更改动画的情况下将描述是否需要生成预期的运动矢量，并且在从不更改但却很少播放或取决于玩家场景的动画的情况下将描述是否已经离线预生成预期的运动矢量。在以上示例中，基于玩家与壁的距离对爬壁动画进行拉伸，这意味着将需要在运行时在步骤802“生成预期的运动矢量”生成运动矢量。在另一种情况下，可能已经离线生成运动矢量并从步骤804“读取预生成的运动矢量”读取该运动矢量。

[0075] 预生成的运动矢量的一个示例可以是在大型武器库中在武器之间作切换。可能的武器切换置换的数量能够变得很大，使得缓存整个所生成运动矢量的集合是不切实际的。通常，如果运动矢量在有限的缓存中占用过多的空间而并没有被足够频繁地使用，这些运动矢量对于预缓存不是最佳选择。在步骤806“发送预期的运动矢量和无效器”，将预期的运动矢量发送到客户端。在步骤808“缓存预期的运动矢量和无效器”，将预期的运动矢量添加到运动矢量查找表。在一个实施例中，无效系统在功能上类似于触发查找表中的运动矢量的应用但禁用运动矢量的系统。当在步骤808“缓存预期的运动矢量和无效器”接收到运动矢量的集合和无效器时，将需要通过无效系统注册无效器。

[0076] 以上结合图1至6所述的玩家输入运动补偿方法将所有玩家输入与运动矢量查找表中的条目进行比较。在上文的示例中，当玩家输入所需的输入以启动爬壁动画时，在步骤810“匹配接收到的玩家输入”，所述输入将匹配先前缓存在查找表中的输入。如果缓存的预期运动矢量还未无效，则将在步骤812“应用玩家输入运动补偿”应用预期的运动矢量。如果匹配的玩家输入将使预期的运动矢量无效，或如果预期的运动矢量在预定时间后过期，或如果预期的运动矢量在被应用一次之后过期，则在步骤814“无效”从查找表移除预期的运动矢量并不再应用该预期的运动矢量。

[0077] 图9示出可以使预期的运动矢量集合无效的信号的示例性方法。无效是对查找表的一种更新，其中将预期的运动矢量的集合从查找表移除，该查找表优选地缓存在客户端处。除了响应于从服务器接收到的更新事件之外，更新机制还可以监视玩家输入并包括无效倒数计时器。当预期的运动矢量的集合缓存在查找表中时，它的无效器将可能使用更新功能/机制来注册。无效器是对查找表触发无效更新的任何数据信号。在步骤900示出的示例性查找表“查找表”包括三个预期的运动矢量集合：在步骤902“预期的门动画”示出的预

期的门动画，在步骤904“预期的爬壁”示出的预期的爬壁动画，以及在步骤906“预期的射杀动画”示出的预期的射杀动画。预期的运动矢量可以在被应用一次之后无效。在该示例中，当玩家在步骤908“开门输入”按压输入以打开附近的门时，应用在步骤902“预期的门动画”示出的预期的门动画运动矢量。同时，注销在步骤908“开门输入”的开门输入，并且可以在步骤910“使用后无效”将在步骤902“预期的门动画”示出的预期的门动画从查找表移除。类似地，其它输入可以使预期的运动矢量在应用前无效。在示例中，在步骤904“预期的爬壁”示出的爬壁动画的预期运动矢量的集合仅在距壁有限的距离内有效。

[0078] 如果在玩家跳跃以抓住壁(通过使用在步骤914“跳跃输入”示出的跳跃输入)之前，玩家通过使用在步骤912“移动输入”示出的移动输入从壁移开，然后在步骤904“预期的爬壁”示出的预期的爬壁运动矢量将在步骤916“输入无效”变成无效。使输入无效的其它示例可以是以下情况：玩家具有将会使预期的运动矢量无效的抓钩或一些其它基于移动的武器；以及玩家按下按钮以激活特定能力。因为无效输入是用于特定场景的，基于特定的实施方式，其它可能是明显的。预期的运动矢量还可以随着时间推移而过期。在示例中，仅在三秒的窗口内向玩家显示射杀机会。如果在三秒的窗口内，玩家不按压在步骤918“混战输入”的混战输入，则在步骤906“预期的射杀动画”示出的预期的射杀动画的运动矢量将在步骤920“计时器超时无效”变成无效。预期的运动矢量可以具有多个无效器，反之亦然。例如，如果接收到移动输入或在爬壁运动矢量使用之后，不管哪个在先，预期的爬壁都可以变成无效。

[0079] 图10示出用于生成运动矢量库的示例性方法和以缓存为目的的示例性重复运动矢量库。由于所选的运动本质上是高度重复的，所以该运动每次触发时将以相同方式播放。这使得能够事先生成运动矢量并将其组织到库中。运动矢量库生成能够在开发期间的任何时刻进行，但在将此过程添加为构建过程或某些其它资产调节过程期间的一个阶段是有意义的。在该示例中，将针对第一人称射击中的每个可用武器生成运动矢量库。当在步骤1000“库生成”开始针对第一武器进行库生成时，在步骤1002“触发动画”触发第一武器动画并在步骤1004“记录运动矢量”记录运动矢量。运动矢量可以是游戏生成的或通过在H.264编解码器中使用的更常规的运动估算技术生成的。在优选的实施例中，使用如美国临时申请第62/488,256号和第62/634,464号所述的全部内容并入本申请中的游戏生成的运动矢量来确保高质量的运动估算。如果记录的运动矢量不正确或不正确地量化，则这些运动矢量将在玩家输入运动补偿期间被使用时引入伪影。步骤1002和1004重复直到为库中的每个高度重复的动画记录了运动矢量。库生成再次从步骤1000开始直到生成所有库。

[0080] 在步骤1006“运动矢量库”示出的示例是用于第一人称射击游戏中的等离子步枪武器重复运动矢量库的非常简化的版本。该示例被简化为包括两个简单动画：在步骤1008示出的，用于在玩家发射等离子步枪时播放的反馈动画的一个2帧的动画，以及在步骤1010示出的，当玩家向前走动时发生的步枪摆动运动的一个4帧的动画。在典型、真实的环境中，武器可能具有更多动画并且动画可以比四帧长很多。

[0081] 图11示出缓存、应用和更新运动矢量库以进行玩家输入运动补偿的过程。当游戏初始化时，在步骤1100“在启动时，发送重复运动矢量库”，服务器将向客户端发送预生成的运动矢量库。在步骤1102“缓存重复运动矢量”，客户端通常以查找表的形式将运动矢量库存储在存储器中。在该实施方式中，客户端通用于满足多个流传输游戏的要求。在可替代实

施方式中，游戏专用客户端可以永久存储运动矢量库，而不需要从服务器缓存重复运动矢量。

[0082] 此时，客户端能够开始监视玩家输入并将传入的输入与玩家输入运动补偿查找表中的条目进行比较。在步骤1104“匹配接收的玩家输入”，当传入的输入在查找表中具有匹配的条目时，在步骤1106“应用玩家输入运动补偿”，将使用与玩家输入相关联的运动矢量来提供如结合图1至10示例性地描述的运动估算。可能存在某些情况，其中接收特定输入可能需要在步骤1112“更新缓存的重复运动矢量映射”更改查找表。暂停按钮是会更改查找表的输入的示例，该暂停按钮可以禁用诸如角色移动、武器发射动画或其它游戏相关的运动的玩家输入。在应用运动矢量并可选地更新查找表之后，客户端将继续监视玩家输入。当在步骤1112“更新缓存的重复运动矢量映射”更新查找表之后，传入的玩家输入将与查找表中更新的条目进行比较。

[0083] 在游戏的运行期间的任何时刻，对玩家的场景改变可能需要更改查找表。例如，玩家可以切换武器，要求缓存的运动库从先前持有的武器切换到新的武器。示例性的实施方式可以看出针对特定游戏事件的服务器侧的游戏监视。这些事件将在游戏开发期间通过本领域已知的方法进行配置，并可以通过游戏现有的消息传递或事件系统发送。当在步骤1108“在运行期间接收事件”运行的游戏实例接收这些事件之一时，将生成消息并在步骤1110“发送场景更新”将该消息发送到客户端。可以针对改变玩家输入与查找表中包括的运动矢量之间的关系的任何游戏事件发送场景更新。场景更新可以从触发运动矢量的集合中启用或禁用玩家输入，改变与给定输入相关联的运动矢量，或以其他方式添加或移除玩家输入与用于玩家输入运动补偿的运动矢量之间的关联。当消息到达客户端时，在步骤1112“更新缓存的重复运动矢量映射”，根据改变的场景来改变查找表。

[0084] 图12是示出可以如何更新运动矢量映射的图。存储在客户端缓存1202中的查找表1200被客户端用来查找哪些运动矢量与给定玩家输入相关联。在游戏运行期间，有时场景的改变会改变玩家输入的行为。例如，如果玩家向前移动但碰到例如墙的移动障碍，将需要停止应用对于向前移动的预测运动矢量。当这种情况发生时，如图11中的步骤1110所示，服务器将向客户端发送场景更新。接收到该移动阻碍事件1204，这向客户端发送信号以禁用向前移动输入1206和相应的运动矢量1208之间的联系。当玩家使用向前移动输入直到来自服务器的另一事件重新启用向前移动输入与向前移动运动矢量之间的联系时，向前移动运动矢量将停止应用。对于该示例，当玩家的移动变畅通并且向客户端发送场景更新以重新建立向前输入与查找表中的向前移动运动矢量之间的联系。

[0085] 在另一示例中，玩家持有等离子步枪但切换为猎枪。等离子步枪运动矢量库1210与其发射运动矢量1209一起存储在缓存中，所述发射运动矢量对应于特定的发射输入1211。缓存还存储针对包括猎枪1212和手枪1214的其它武器的运动矢量库。当客户端从服务器接收武器切换事件1216时，将等离子步枪运动矢量库1210从用于猎枪运动矢量库1212的查找表1200切换出来。为了避免玩家输入运动补偿在武器切换期间不当地进行，两个事件能够相继使用，以在播放武器切换动画时首先禁用等离子步枪运动矢量库1210，然后在武器切换动画完成之后切换两个运动矢量库。

[0086] 对于更长的多帧运动矢量，能够扩展该多帧运动矢量的应用，使得在从服务器接收实际视频的最后一帧时应用运动矢量的最后一帧。这将允许服务器在客户端用尽缓存的

运动矢量时赶上估算的运动。运动矢量的缩放因子被定义为播放速度缩放，其示例性地计算如下。

[0087] $$播放速度缩放 = \frac{缓存的动画时间}{缓存的动画时间+延迟}$$

[0088] 延迟定义为从初始玩家输入事件与客户端上接收到实际视频之间的时间。延迟包括将输入经由网络发送到服务器所花费的时间，在服务器上处理包括游戏逻辑、渲染逻辑、GPU渲染的时间以及编码的时间，还有将视频返回给玩家的网络时间。应当已经持续地在任何流传输环境中测量延迟。如美国临时申请第62/488,256号和第62/634,464号所述，玩家输入运动补偿的优选实施例使用相关标签，以将玩家输入与实际视频关联。在将传入的玩家输入发送给服务器之前，附加作为唯一标识符的相关标签。当具有相关标签的视频帧从服务器返回时，客户端将唯一标识与先前输入进行匹配。向客户端发送信号停止为相关输入进行运动估算或通过混合技术撤销先前的运动估算。能够通过将缓存动画中的帧数乘以每帧的长度来计算动画的缓存部分的长度或缓存的动画时间。

[0089] 在图13中，在以60帧/秒并具有100ms延迟运行的游戏中，使用包含10帧的示例性动画进行玩家输入运动补偿。当玩家输入触发玩家输入运动补偿时，根据缓存的运动矢量计算出播放速度缩放，如下所示。

[0090] $$播放速度缩放 = \frac{缓存的动画时间}{缓存的动画时间+延迟}$$

[0091] $$播放速度缩放 = \frac{10帧 * \frac{1000毫秒}{60帧}}{(10帧 * \frac{1000毫秒}{60帧}) + 100毫秒}$$

[0092] $$播放速度缩放 = \frac{166.66毫秒}{166.66毫秒+100毫秒} = 0.625$$

[0093] 在时间0ms接收玩家收入。通过计算的播放速度缩放将运动矢量的播放速率缩放1300。运动矢量的第一帧应用于来自服务器的视频流中的下一可用帧。缩放的运动矢量帧将被插值以保持动画平滑。因为运动矢量帧在多个帧上缩放，插值是一种可以用来计算在任何给定帧上应用“多少”运动矢量的方法。示例性实施方式可以基于计算的播放速度缩放使用线性的插值。对于我们的示例，计算的播放速度缩放是0.625，其将在1.6显示帧上拉伸运动矢量的集合。插值是一种用来计算在给定帧上应用多少运动矢量的方法。即，当运动矢量的集合在多个显示帧上拉伸时，插值计算将宏块沿运动矢量向下移动多远。在17ms时，仅第一缩放运动矢量的一部分应当应用于第一显示帧，其等于播放速度缩放0.625。在33ms时的第二显示帧上，应用第一缩放运动矢量的其余部分，计算为1-0.625＝0.375，然后应用第二缩放运动矢量的第一部分，计算为播放速度缩放减去第一缩放运动矢量的其余部分，或0.62-0.375＝0.25。在50ms时的第三显示帧上，继续应用缩放运动矢量的第二集合，将宏块沿运动矢量向下移动下一个62.5％。在67ms时的第四显示帧上，应用第二缩放运动矢量的其余部分，计算为1-0.25-0.625＝0.125，并且应用第三压缩运动矢量的第一部分，计算为播放速度缩放减去第二缩放运动矢量的其余部分0.625-0.125＝0.5。随着缩放运动矢量的

应用,线性插值继续。

[0094] 多帧运动矢量可以针对缓存的动画的每帧发送相关标签以使估算的运动的每帧与未来的实际视频关联。

[0095] 延迟将极大地取决于客户端和服务器之间的网络路径和架构。该示例使用100ms的延迟,但是延迟可能会在数十到数百毫秒之间变化。更短的延迟将提供更好的玩家体验,但是玩家输入运动补偿技术能够有助于掩饰某些情况下的高延迟时间的影响。在延迟1304之后,接收实际视频1302。对于位于边缘的服务器或物理上靠近用户的服务器,延迟时间可以低到30ms。对于更典型的服务器位置,更可能是100ms。实际视频保持原始动画的长度1306,因为其未被缩放。根据玩家输入运动补偿技术应用在实际视频。

[0096] 如果客户端不能够完美地混合先前的运动补偿,则H.264编码标准提供能够校正任何时间传播错误的冗余片特征。基于H.264配置文件设置,每个片将被编码为内部片(I-片)并且以特定频率按轮换时间表进行发送。因为内部片将不包含运动矢量,所以仅当实际运动矢量到达p片时,才应当应用混合运动矢量。这将避免在加标签的帧从服务器返回之前在已经存在于I-片中的宏块上应用混合运动矢量。

[0097] 前面的描述和附图应被认为仅是本发明原理的说明。本发明无意于受优选实施例的限制,并且可以以本领域普通技术人员显而易见的各种方式来实施。本领域技术人员将容易想到本发明的许多应用。因此,不希望将本发明限制于所公开的具体示例或所示出和描述的确切构造和操作。相反,在本发明的范围内,可以采用所有合适的修改和等同方案。

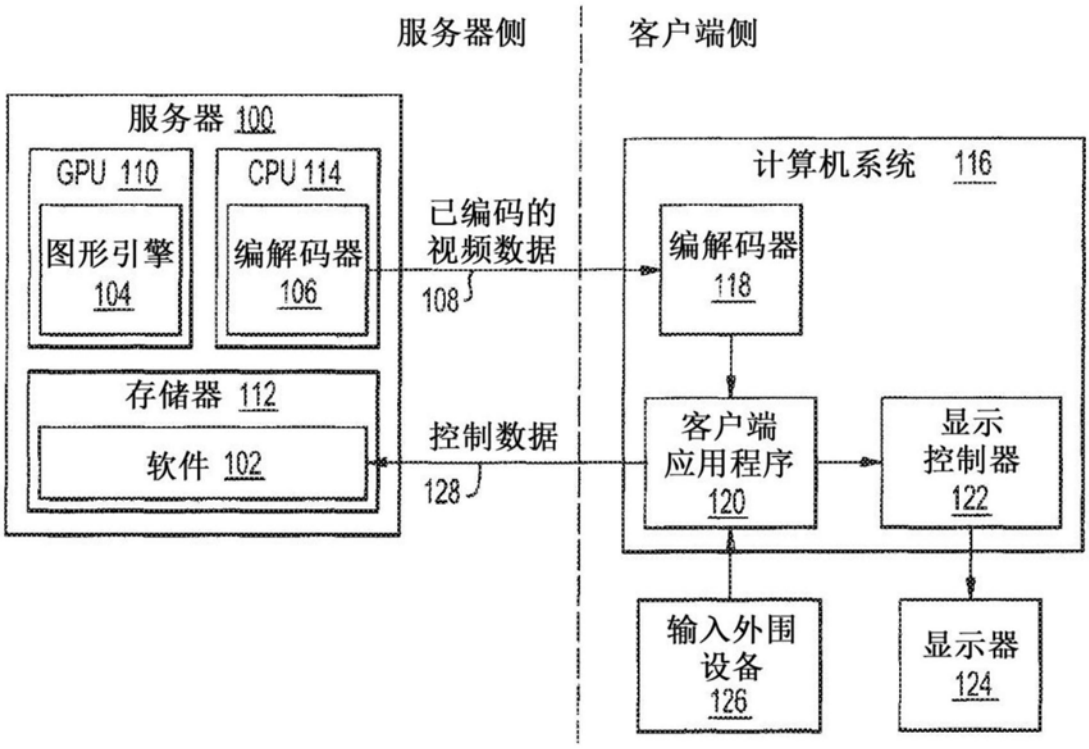

服务器侧
客户端侧
服务器 100
GPU 110
图形引擎 104
CPU 114
编解码器 106
存储器 112
软件 102
已编码的视频数据
108
控制数据
128
计算机系统 116
编解码器 118
客户端应用程序 120
显示控制器 122
输入外围设备 126
显示器 124

图1

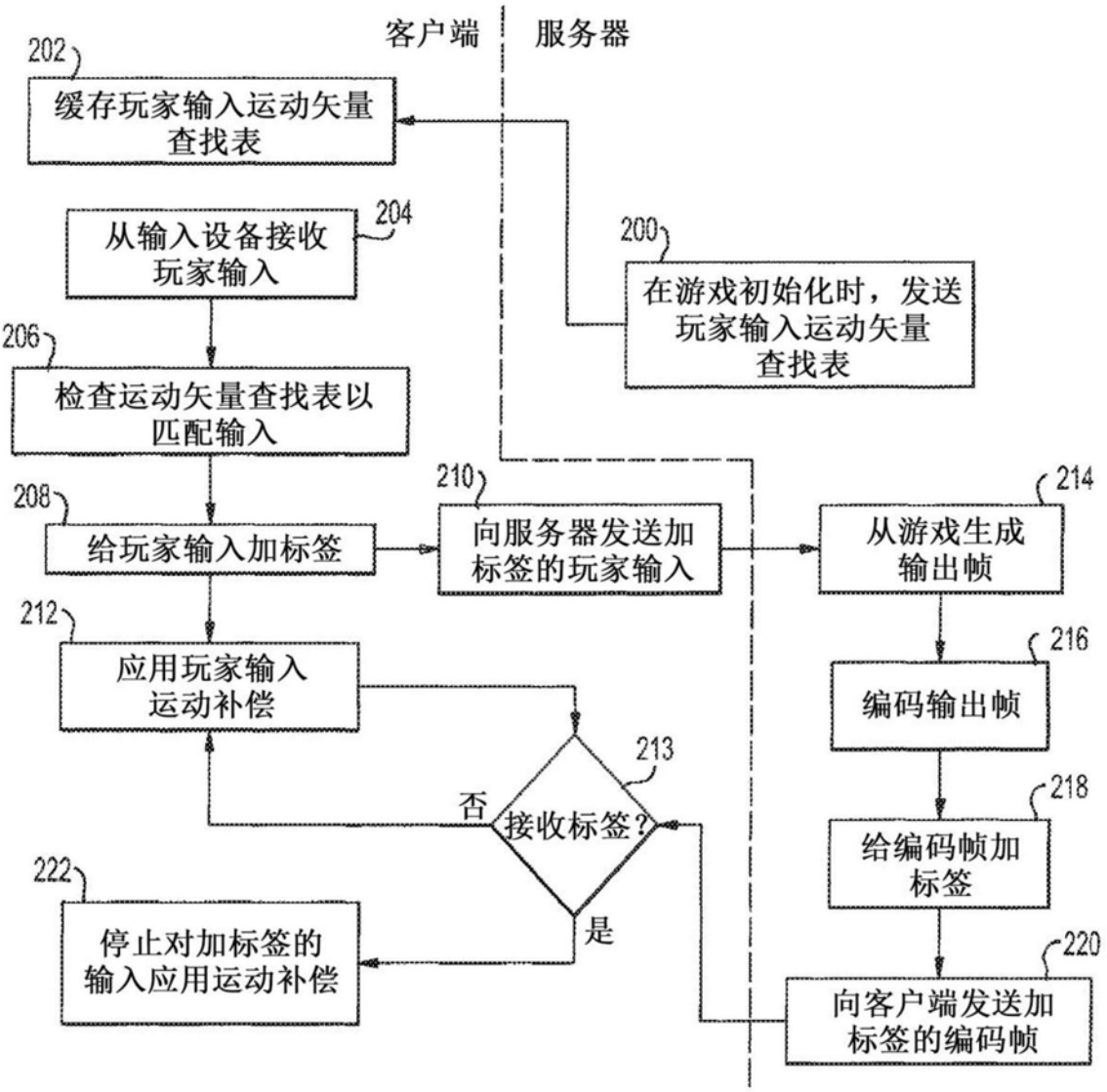

客户端
服务器
202
缓存玩家输入运动矢量查找表
200
在游戏初始化时，发送玩家输入运动矢量查找表
204
从输入设备接收玩家输入
206
检查运动矢量查找表以匹配输入
208
给玩家输入加标签
210
向服务器发送加标签的玩家输入
214
从游戏生成输出帧
212
应用玩家输入运动补偿
216
编码输出帧
213
否
接收标签?
218
给编码帧加标签
222
停止对加标签的输入应用运动补偿
是
220
向客户端发送加标签的编码帧

图2

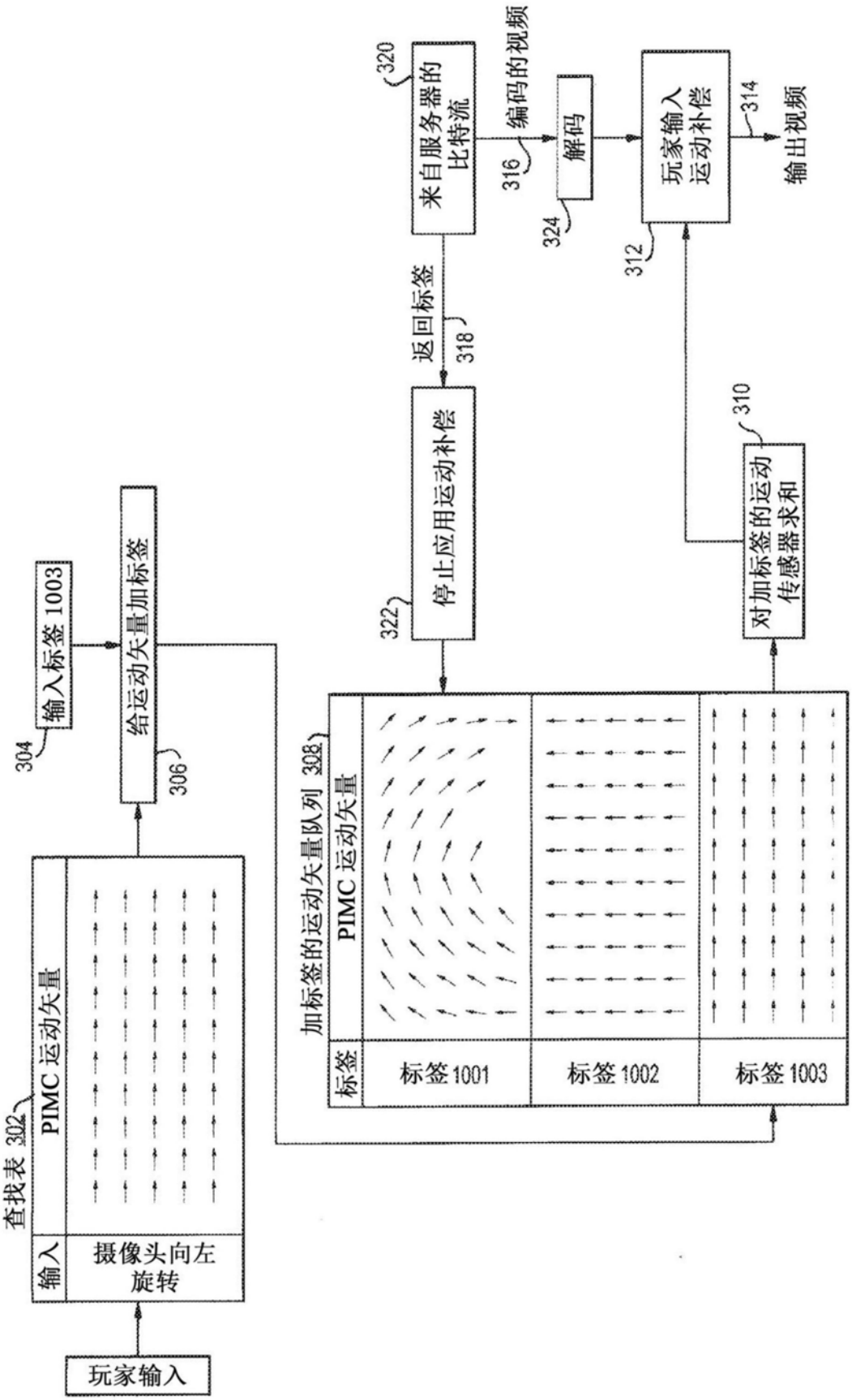

玩家输入
查找表 302
输入
PIMC 运动矢量
摄像头向左旋转
304
输入标签 1003
306
给运动矢量加标签
加标签的运动矢量队列 308
标签
PIMC 运动矢量
标签 1001
标签 1002
标签 1003
322
停止应用运动补偿
320
来自服务器的比特流
316
编码的视频
324
解码
312
玩家输入运动补偿
314
输出视频
返回标签
318
310
对加标签的运动传感器求和

图3

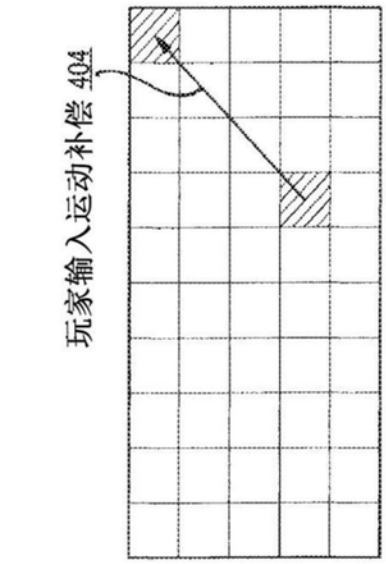

玩家输入运动补偿 404

图4C

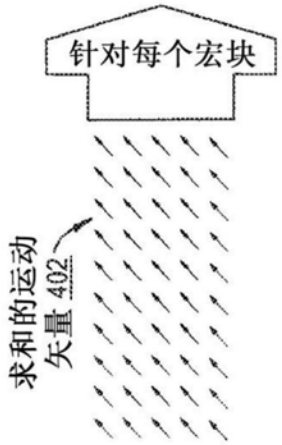

针对每个宏块
求和的运动
矢量 402

图4B

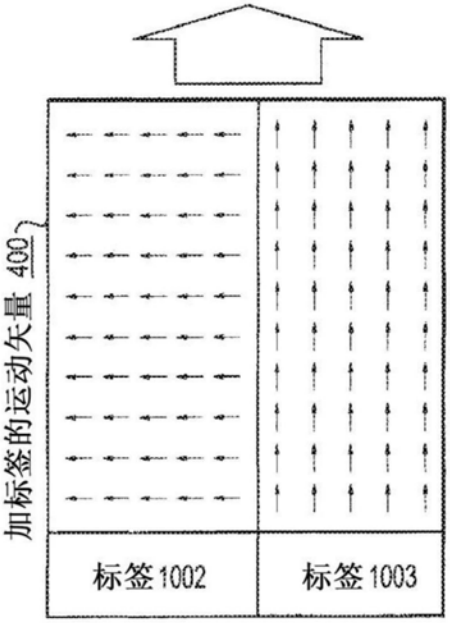

加标签的运动矢量 400
标签1002
标签1003

图4A

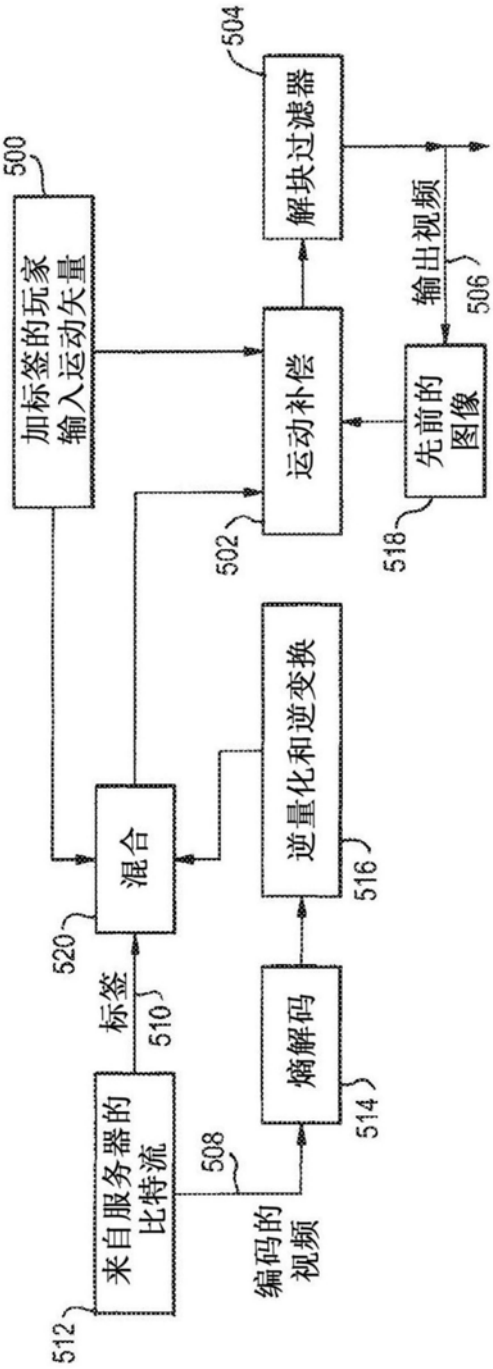

500
加标签的玩家输入运动矢量
504
解块过滤器
输出视频
506
运动补偿
502
先前的图像
518
逆量化和逆变换
516
混合
520
标签
510
熵解码
514
来自服务器的比特流
512
508
编码的视频

图5

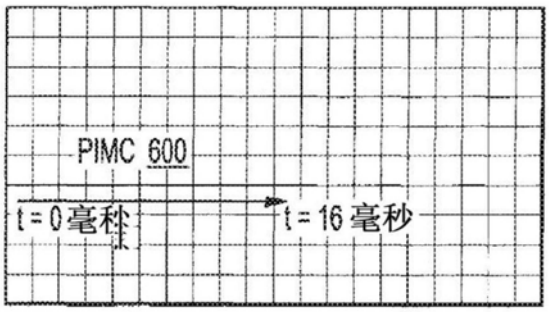

PIMC 600
t = 0 毫秒
t = 16 毫秒

图6A

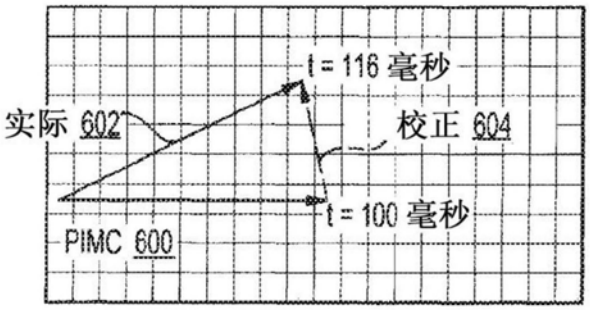

t = 116 毫秒
实际 602
校正 604
t = 100 毫秒
PIMC 600

图6B

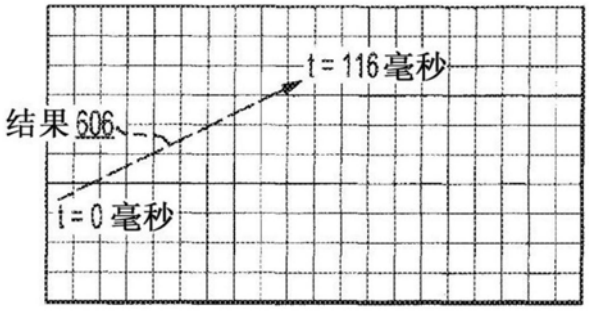

t = 116 毫秒
结果 606
t = 0 毫秒

图6C

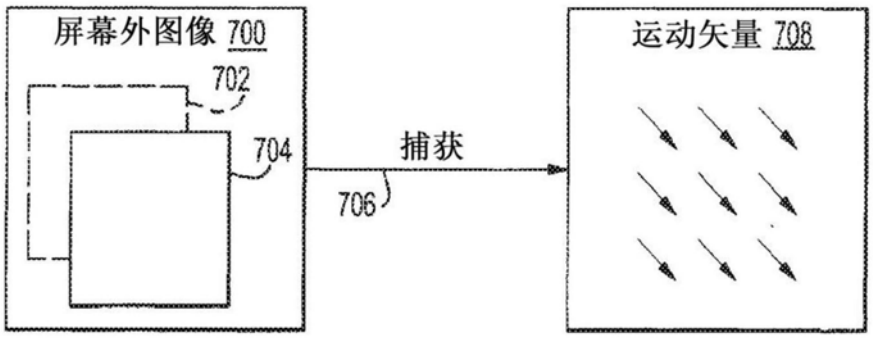

屏幕外图像 700
702
704
捕获
706
运动矢量 708

图7

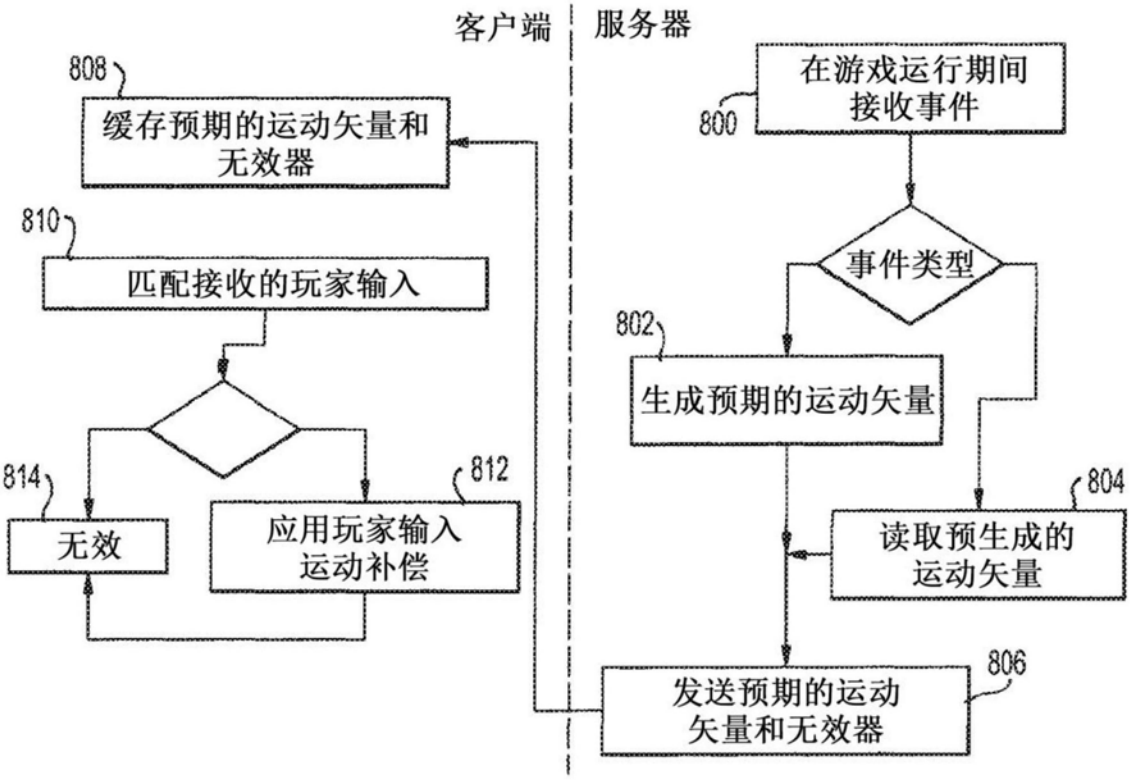

客户端
服务器
808
缓存预期的运动矢量和无效器
810
匹配接收的玩家输入
814
无效
812
应用玩家输入运动补偿
在游戏运行期间接收事件
800
事件类型
802
生成预期的运动矢量
804
读取预生成的运动矢量
806
发送预期的运动矢量和无效器

图8

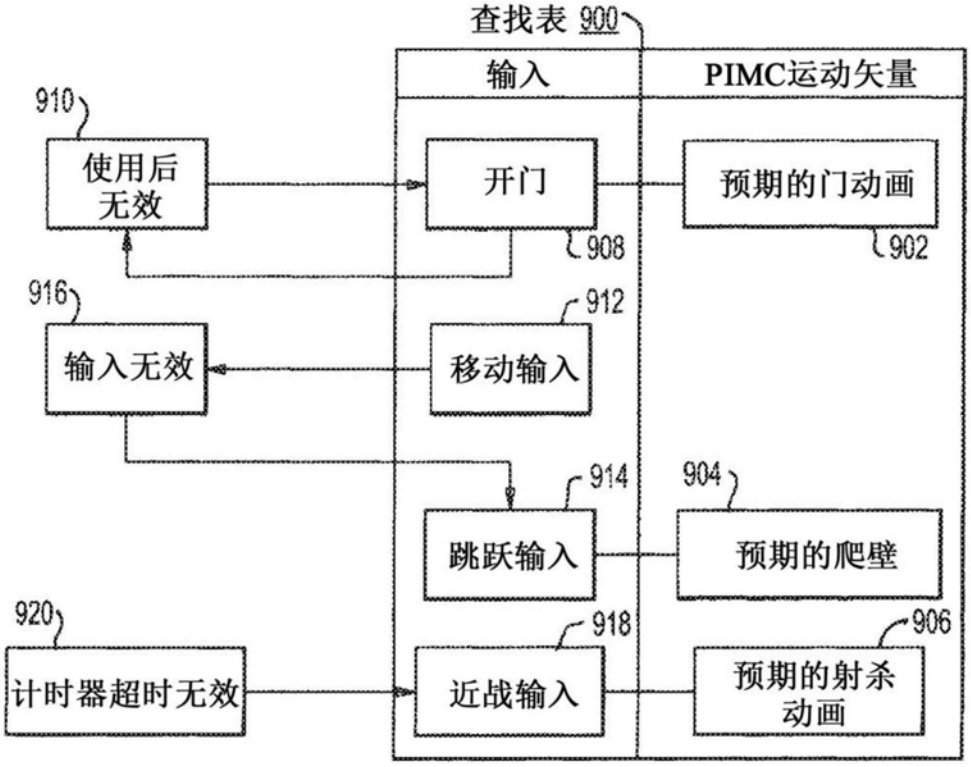

查找表 900
输入
PIMC运动矢量
910
使用后无效
开门
908
预期的门动画
902
916
输入无效
912
移动输入
914
跳跃输入
904
预期的爬壁
920
计时器超时无效
918
近战输入
906
预期的射杀动画

图9

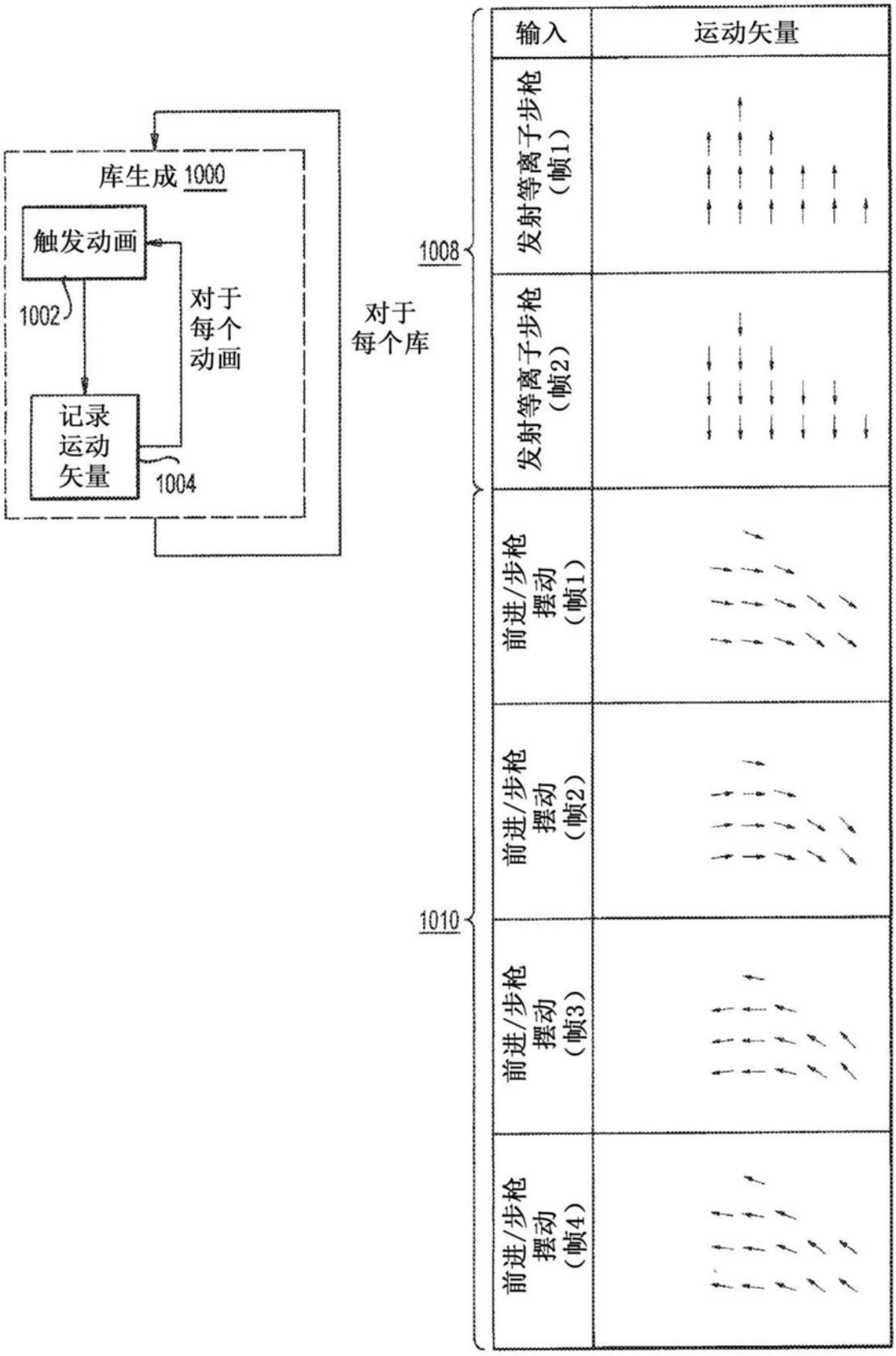

库生成 1000
触发动画
1002
对于每个动画
记录运动矢量
1004
对于每个库
输入
运动矢量
发射等离子步枪（帧1）
发射等离子步枪（帧2）
前进/步枪摆动（帧1）
前进/步枪摆动（帧2）
前进/步枪摆动（帧3）
前进/步枪摆动（帧4）
1008
1010

图10

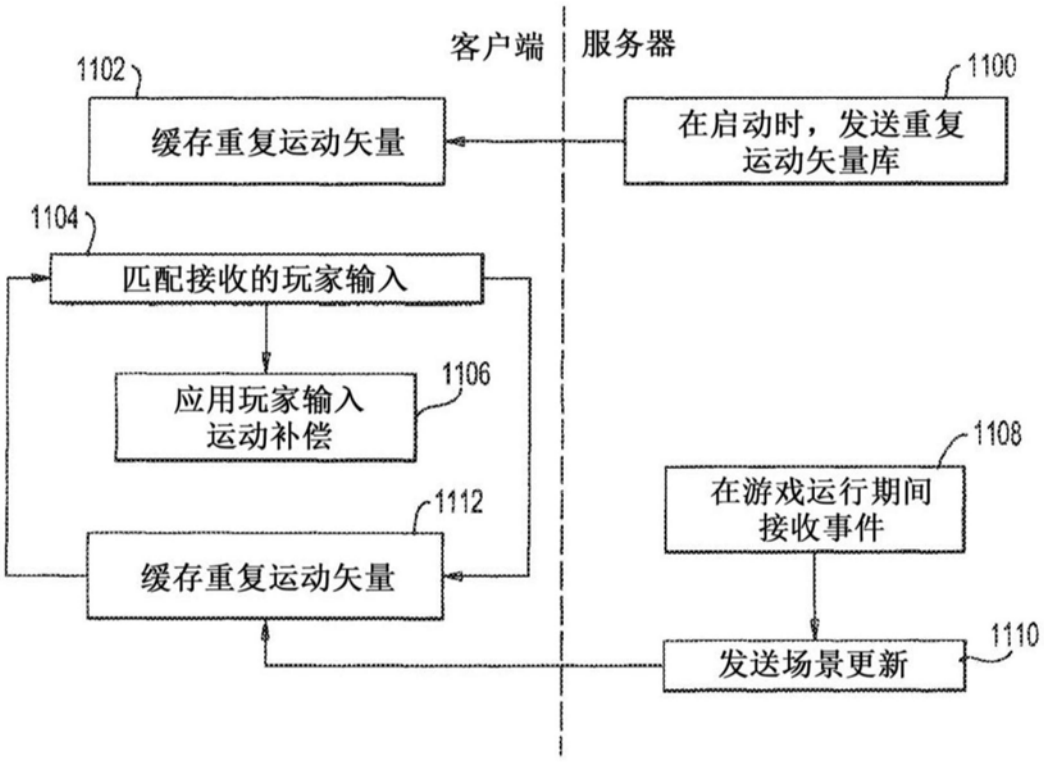

客户端
服务器
1102
缓存重复运动矢量
1100
在启动时，发送重复运动矢量库
1104
匹配接收的玩家输入
1106
应用玩家输入运动补偿
1108
在游戏运行期间接收事件
1112
缓存重复运动矢量
1110
发送场景更新

图11

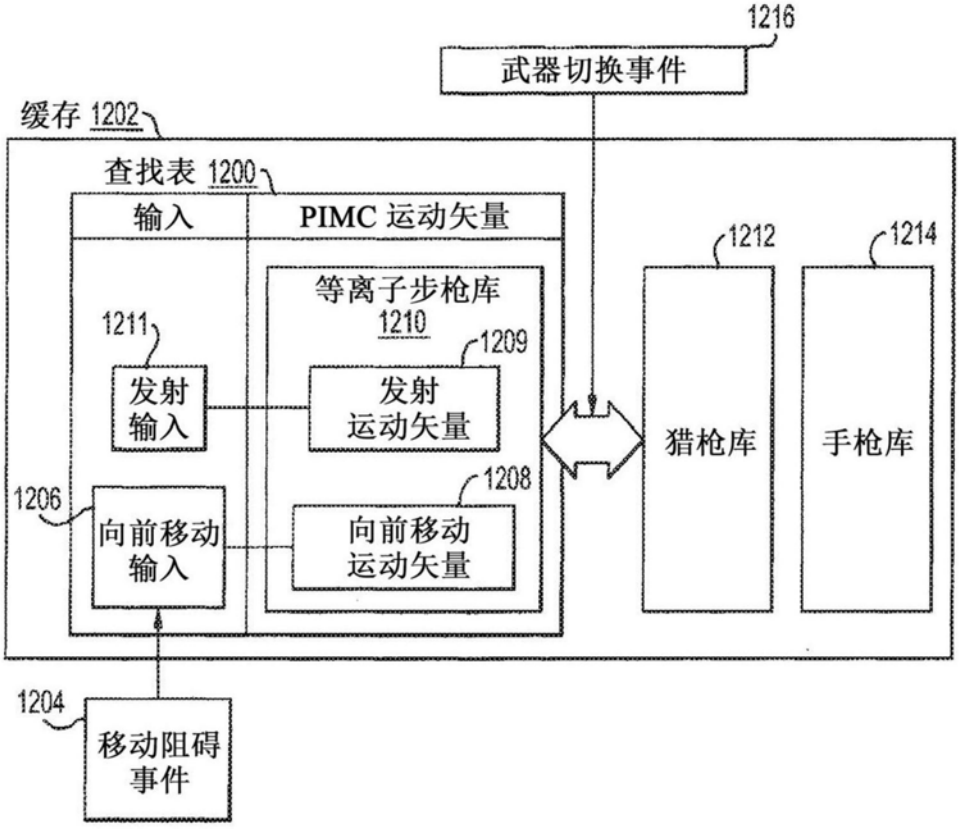

1216
武器切换事件
缓存 1202
查找表 1200
输入
PIMC 运动矢量
1212
1214
等离子步枪库
1210
1211
1209
发射
输入
发射
运动矢量
猎枪库
手枪库
1208
1206
向前移动
输入
向前移动
运动矢量
1204
移动阻碍
事件

图12

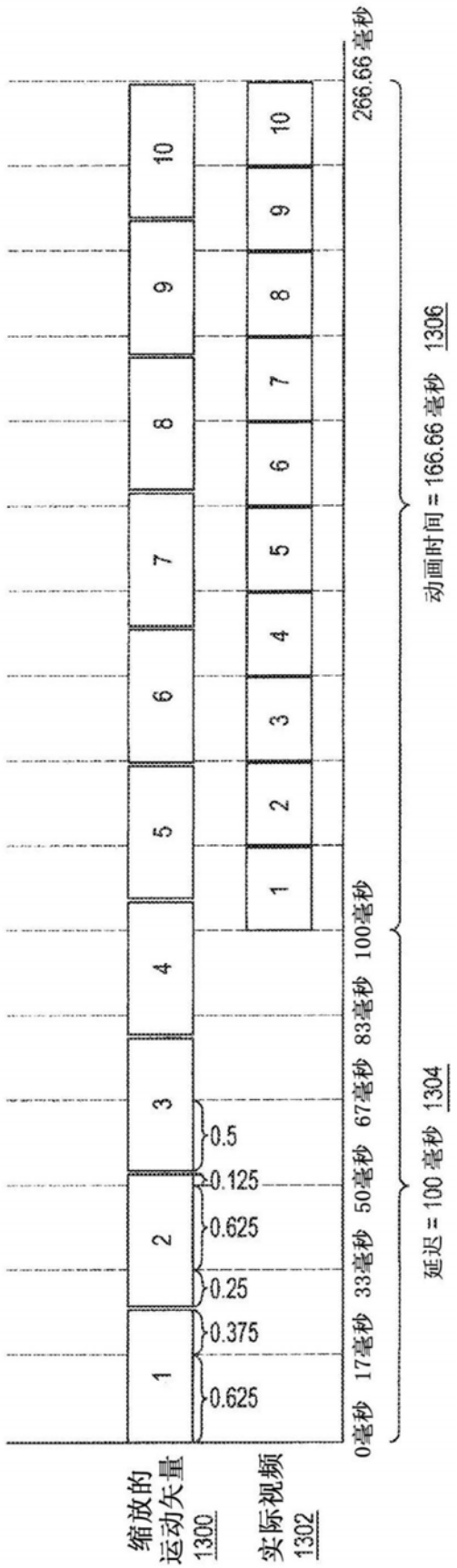

缩放的
运动矢量
1300
实际视频
1302
1
2
3
4
5
6
7
8
9
10
0.625
0.375
0.25
0.625
0.125
0.5
0毫秒
17毫秒
33毫秒
50毫秒
67毫秒
83毫秒
100毫秒
266.66毫秒
延迟 = 100 毫秒 1304
动画时间 = 166.66 毫秒 1306

图13