



(12) 发明专利申请

(10) 申请公布号 CN 103891303 A

(43) 申请公布日 2014.06.25

(21) 申请号 201280050754.7

代理人 余刚 吴孟秋

(22) 申请日 2012.05.25

(51) Int. Cl.

(30) 优先权数据

61/524,269 2011.08.16 US

61/557,856 2011.11.09 US

H04N 21/44 (2006.01)

H04N 21/434 (2006.01)

H04N 21/438 (2006.01)

H04N 21/236 (2006.01)

(85) PCT国际申请进入国家阶段日

2014.04.15

(86) PCT国际申请的申请数据

PCT/CA2012/050345 2012.05.25

(87) PCT国际申请的公布数据

W02013/023287 EN 2013.02.21

(71) 申请人 黛斯梯尼软件产品有限公司

地址 加拿大不列颠哥伦比亚省

(72) 发明人 史蒂文·埃里克·韦斯特高 徐志威

涂绍宁

(74) 专利代理机构 北京康信知识产权代理有限

责任公司 11240

权利要求书13页 说明书34页 附图11页

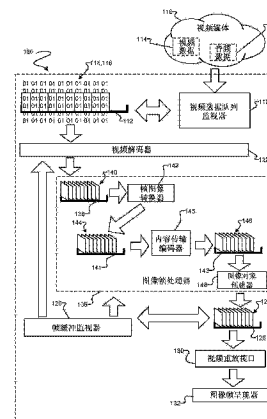
按照条约第19条修改的权利要求书19页

(54) 发明名称

基于脚本的视频呈现

(57) 摘要

提供了用于在用户计算平台上跨平台呈现视频内容的系统和方法,所述用户计算平台是多个不同用户计算平台类型中的一种。将脚本发送给用户计算平台并且所述脚本由被编译以在所述多个用户计算平台类型中的任一个上进行操作的程序解释。发送脚本,使得所述脚本由在用户计算平台上操作的程序解释,以使得所解释的脚本直接将发送给用户计算平台或者可由用户计算平台访问的经编码的视频数据解码成经解码的视频数据,并且进一步使得呈现经解码的视频数据。



1. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

其中,发送所述脚本,使所述脚本由在所述用户计算平台上操作的所述应用程序解释,以使所解释的脚本直接将所述用户计算平台可访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据。

2. 根据权利要求1所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以在所述用户计算平台上本地地显示所述一个或多个帧图像。

3. 根据权利要求1所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以使得所解释的脚本直接将可由所述用户计算平台访问的经编码的视频数据解码成经解码的音频数据。

4. 根据权利要求3所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以:通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据;并且通过在所述用户计算平台上重放所述经解码的音频数据,来呈现所述经解码的音频数据。

5. 根据权利要求3所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以:通过在所述用户计算平台上显示所述一个或多个帧图像,来本地地呈现所述经解码的视频数据;并且通过重放所述经解码的音频数据,来本地地呈现所述经解码的音频数据。

6. 根据权利要求4或5中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,所述一个或多个帧图像包括多个帧图像,并且其中,发送所述脚本使得脚本由所述应用程序解释,以使得所解释的脚本直接同步所述帧图像的显示与所述经解码的音频数据。

7. 根据权利要求4或5中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,其中,所述一个或多个帧图像包括多个帧图像,并且其中,发送所述脚本使得所述脚本由所述应用程序解释,以确定与经解码的音频数据的重放相关联的音频重放时间基准,并且使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述经解码的音频数据的重放。

8. 根据权利要求7所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括以下中的至少一个:根据所述视频数据的呈现比所述音频重放时间基准晚第一阈值时间量这一确定,通过跳过一个或多个帧图像的显示,脱离顺序地提前显示帧图像;并且根据所述视频数据的呈现比所述音频重放时间基准提早第二阈值时间量这一确定,使得显示一个或多个帧图像的时间段比由所述视频数据的帧速率指示的时间段更长。

9. 根据权利要求7所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括:使用所述音频重放时间基准来确定当前帧索引;并且如果准备好显示与所述当前帧索引相关的帧图像,那么显示与所述当前帧索引相关的帧图像。

10. 根据权利要求9所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括确定与所述当前帧索引相关的帧图像在帧图像队列中位于一个或多个其

他帧图像之后,并且如果是这样的话,那么跳过显示所述一个或多个其他帧图像。

11. 根据权利要求 9 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括确定未准备好显示与所述当前帧索引相关的所述帧图像,并且根据这种确定,开始丢帧动作,所述丢帧动作包括以下中的一个或多个:记录未准备好呈现与所述当前帧索引相关的帧图像的指示;使得所述经解码的音频数据的重放暂停或放慢;使得所述经编码的视频数据的解码速率增大;使得切换成具有更低的帧速率的视频数据;通过将所述图像帧的一个子集解码,使得切换成有效下采样视频数据。

12. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代调度和执行包括将所述经编码的视频数据直接解码成所述帧图像的第一模拟线程;并且迭代调度和执行包括通过显示所述帧图像来呈现所述视频数据的第二模拟线程,来模拟多线程执行环境。

13. 根据权利要求 12 所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代调度和执行包括将所述经编码的音频数据直接解码成所述经解码的音频数据的第三模拟线程;并且迭代调度和执行包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的第四模拟线程,来模拟多线程执行环境。

14. 根据权利要求 4 所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代地调度和执行一种方法,来模拟多线程执行环境,所述方法在每个迭代上对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,选择和执行以下中的一个:

第一模拟线程,其包括将所述经编码的视频数据直接解码成所述帧图像;

第二模拟线程,其包括通过显示所述帧图像来呈现所述视频数据;

第三模拟线程,其包括将所述经编码的音频数据直接解码成经解码的音频数据;

第四模拟线程,其包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口。

15. 根据权利要求 12 所述的方法,其中,执行所述第一模拟线程,包括:将第一模拟线程分成多个子部分;在子部分对的性能之间,对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,暂时中断第一模拟线程,以选择和执行以下中的一个的迭代:第二模拟线程;以及包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的另一个模拟线程。

16. 根据权利要求 4 或 5 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据,包括将至少一对帧图像 α 混合在一起。

17. 根据权利要求 4 或 5 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据,包括在显示第一个所述帧图像与显示第二个所述帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像,包括 α 混合所述第一个帧图像和所述第二个帧图像。

18. 根据权利要求 17 所述的方法,其中,在显示所述第一个帧图像与显示所述第二个帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像,包括使用 α 混合系数 α , 在所述先前显示的图像上 α 混合所述第二个帧图像,由

$$\alpha = [\beta + 1 - \text{mod}(f\#, \beta)]^{-1}$$

提供所述 α 混合系数 α ,

其中 :

β 是等于所述一个或多个时间 +1 的数量的下采样因数 ;

$f\#$ 是参照所述一个或多个时间中的当前一个的数量的指数 ;以及

$\text{mod}(x, y)$ 是返回由 x 除以 y 的整数余数的模数运算符。

19. 根据权利要求 1 到 18 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,使得所解释的脚本直接实现视频数据队列监视器的至少一部分,所述部分直接监视视频数据队列,以确定其状态,并且根据所述视频数据队列的所确定的状态,确定是否提取更多经编码的视频数据。

20. 根据权利要求 19 所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以实现所述视频队列监视器的第二部分,所述第二部分使用 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

21. 根据权利要求 19 所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以实现所述视频队列监视器的第二部分,所述第二部分交替地使用多个 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

22. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以直接多路分用所述视频数据和所述音频数据。

23. 根据权利要求 1 到 22 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以使得所解释的脚本直接实现帧缓冲监视器,所述帧缓冲监视器监视帧图像缓冲器,以确定其状态,并且根据帧图像缓冲器的所确定的状态,确定是否将更多所述经编码的视频数据解码。

24. 根据权利要求 1 到 23 中任一项所述的方法,其中,发送脚本,使得所述脚本由应用程序解释,以使得所解释的脚本直接将至少一个所述帧图像的图像文件格式从第一图像文件格式转换成第二图像文件格式,所述第二图像文件格式能够由所述应用程序本地地呈现。

25. 根据权利要求 1 到 24 中任一项所述的方法,包括将所述经编码的视频数据发送给所述用户计算平台。

26. 根据权利要求 25 所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:将至少一个所述帧图像的图像文件格式从第一图像文件格式转换成第二图像文件格式,所述第二图像文件格式能够由在所述用户计算机平台上操作的应用程序本地地呈现。

27. 根据权利要求 25 所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:

对于以第一文件格式设置的至少第一个所述帧图像:分析所述第一个帧图像的内容;根据所述第一个帧图像的内容的分析,在多个二次图像文件格式之中选择一个;并且如果所述第一文件格式与所述二次图像文件格式中所选择的一个格式不同,那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的一个格式;并且

对于以第二文件格式设置的至少第二个所述帧图像:分析所述第二个帧图像的内容;

根据所述第二个帧图像的内容的分析,在多个二次图像文件格式之中选择不同的一个;并且如果所述第二文件格式与所述二次图像文件格式中所选择的不同的一个格式不同,那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的不同的一个格式;

其中,所述多个二次图像文件格式中的每个能够由所述应用程序本地地呈现。

28. 根据权利要求 1 到 27 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以使得所解释的脚本直接将至少一个所述帧图像转换成内容传输格式,所述内容传输格式能够由解释脚本的脚本解释器传递给所述应用程序。

29. 根据权利要求 1 到 27 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以使得所解释的脚本直接将至少一个所述帧图像转换成 base64 内容传输格式。

30. 根据权利要求 25 所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:在内容传输格式中编码至少一个所述帧图像,据此,所述内容传输编码的帧图像可由解释所述脚本的脚本解释器传递给所述应用程序。

31. 根据权利要求 25 所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:对于多个帧图像中的每个:将所述帧图像编码成内容传输格式,据此,所述内容传输编码的帧图像可由解释所述脚本的脚本解释器传递给所述应用程序;并且通过与所述帧图像无关的填充数据,将所述内容传输编码的帧图像填充成特定的尺寸。

32. 根据权利要求 25 所述的方法,包括:在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:以 base64 内容传输格式编码至少一个帧图像。

33. 根据权利要求 1 到 32 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以将至少一个所述帧图像本地地转换成相应的图像对象,所述图像对象能够由所述应用程序本地地呈现。

34. 根据权利要求 1 到 33 中任一项所述的方法,其中,所述应用程序包括网络浏览器。

35. 根据权利要求 1 到 34 中任一项所述的方法,其中,所述脚本嵌入 HTML 文档中。

36. 根据权利要求 3 所述的方法,其中,所述一个或多个帧图像包括多个帧图像,并且其中,发送所述脚本使得所述脚本由所述应用程序解释,以:通过重放所述经解码的音频数据,来呈现所述音频数据;并且通过选择所述多个帧图像中的一个子集和显示所述多个帧图像的所选择的一个子集,来呈现所述视频数据。

37. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,改变包括可由所述用户计算平台访问的经编码的视频数据的文件。

38. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,修改以下一个或多个项中的至少一个特征:将所述经编

码的数据解码以及显示所述一个或多个帧图像。

39. 根据权利要求 4 或 5 中的任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以确定将所述经编码的视频数据发送给用户计算平台的数据速率,并且根据所述确定,改变包括发送给所述用户计算平台的所述经编码的视频数据的文件。

40. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以确定将所述经编码的视频数据发送给所述用户计算平台的数据速率,并且根据所述确定,修改以下一个或多个中的至少一个特征:将所述经编码的数据解码以及显示所述帧图像。

41. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在计算机平台上使用所述应用程序,来解释脚本;

其中,解释所述脚本,使得所解释的脚本直接将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据。

42. 根据权利要求 41 所述的方法,其中,解释所述脚本使得所述应用程序在所述用户计算平台上本地地显示所述一个或多个帧图像。

43. 根据权利要求 41 所述的方法,其中,解释所述脚本使得所解释的脚本直接将可由用户计算平台访问的经编码的视频数据解码成经解码的音频数据。

44. 根据权利要求 43 所述的方法,其中,解释所述脚本,通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据,并且通过在所述用户计算平台上重放所述经解码的音频数据,来呈现所述经解码的音频数据。

45. 根据权利要求 43 所述的方法,其中,解释所述脚本使得所述应用程序:通过在所述用户计算平台上显示所述一个或多个帧图像,来本地地呈现所述经解码的视频数据;并且通过重放所述经解码的音频数据,来本地地呈现所述经解码的音频数据。

46. 根据权利要求 44 或 45 中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,所述一个或多个帧图像包括多个帧图像,并且其中,解释所述脚本,使得所解释的脚本直接同步所述帧图像的显示与所述经解码的音频数据的重放。

47. 根据权利要求 44 或 45 中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,其中,所述一个或多个帧图像包括多个帧图像,并且其中,解释所述脚本,确定与所述经解码的音频数据的重放相关联的音频重放时间基准,并且使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述经解码的音频数据的重放。

48. 根据权利要求 47 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括以下中的至少一个:根据所述视频数据的呈现比所述音频重放时间基准晚第一阈值时间量这一确定,通过跳过一个或多个帧图像的显示,脱离次序地提前显示帧图像;并且根据所述视频数据的呈现比所述音频重放时间基准早第二阈值时间量这一确定,使得显示一个或多个帧图像的时间段比由所述视频数据的帧速率指示的时间段更长。

49. 根据权利要求 47 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数

据的重放同步,包括:使用所述音频重放时间基准来确定当前帧索引;并且如果准备好显示与所述当前帧索引相关的帧图像,那么显示与所述当前帧索引相关的帧图像。

50. 根据权利要求 49 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括:确定与所述当前帧索引相关的帧图像在帧图像队列中位于一个或多个其他帧图像之后,并且如果是这样的话,那么跳过显示所述一个或多个其他帧图像。

51. 根据权利要求 49 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括确定未准备好显示与所述当前帧索引相关的帧图像,并且根据这种确定,开始丢帧动作,所述丢帧动作包括以下中的一个或多个:记录未准备好呈现与所述当前帧索引相关的帧图像的指示;使得所述经解码的音频数据的重放暂停或放慢;使得所述经编码的视频数据的解码速率增大;使得切换成具有更低的帧速率的视频数据;通过将图像帧的一个子集解码,使得切换成有效下采样视频数据。

52. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本,通过:迭代调度和执行包括将所述经编码的视频数据直接解码成帧图像的第一模拟线程;并且迭代调度和执行包括通过显示所述帧图像来呈现所述视频数据的第二模拟线程,来模拟多线程执行环境。

53. 根据权利要求 52 所述的方法,其中,解释所述脚本,通过:迭代调度和执行包括将所述经编码的音频数据直接解码成经解码的音频数据的第三模拟线程;并且迭代调度和执行包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的第四模拟线程,来模拟多线程执行环境。

54. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本,通过:迭代调度和执行一种方法,来模拟多线程执行环境,所述方法在每个迭代上对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,选择和执行以下中的一个:

第一模拟线程,其包括将所述经编码的视频数据直接解码成帧图像;

第二模拟线程,其包括通过显示所述帧图像来呈现所述视频数据;

第三模拟线程,其包括将所述经编码的音频数据直接解码成经解码的音频数据;

第四模拟线程,其包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口。

55. 根据权利要求 52 所述的方法,其中,执行所述第一模拟线程,包括:将所述第一模拟线程分成多个子部分;在所述子部分对的性能之间,对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,暂时中断所述第一模拟线程,以选择和执行以下中的一个的迭代:第二模拟线程;以及包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的另一个模拟线程。

56. 根据权利要求 44 或 45 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据包括将至少一对帧图像 α 混合在一起。

57. 根据权利要求 44 或 45 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据包括在显示第一个所述帧图像与显示第二个所述帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像包括 α 混合第一个帧图像和第二个帧图像。

58. 根据权利要求 57 所述的方法,其中,在显示所述第一个帧图像与显示所述第二个

帧图像之间的一个或多个时间的每个时间,显示所述 α 混合的图像包括使用 α 混合系数 α ,在先前显示的图像上 α 混合第二个所述帧图像,由

$$\alpha = [\beta + 1 - \text{mod}(f\#, \beta)]^{-1}$$

提供所述 α 混合系数 α ,

其中:

β 是等于所述一个或多个时间 +1 的数量的下采样因数;

$f\#$ 是参照所述一个或多个时间中的当前一个的数量的指数;以及

$\text{mod}(x, y)$ 是返回由 x 除以 y 的整数余数的模数运算符。

59. 根据权利要求 41 到 58 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接实现视频数据队列监视器的至少一部分,所述至少一部分直接监视视频数据队列,以确定其状态,并且根据所述视频数据队列的所确定的状态,确定是否提取更多经编码的视频数据。

60. 根据权利要求 59 所述的方法,其中,解释所述脚本实现所述视频队列监视器的第二部分,所述第二部分使用 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

61. 根据权利要求 59 所述的方法,其中,解释所述脚本实现所述视频队列监视器的第二部分,所述第二部分交替地使用多个 AJAXXMLHttpRequest 对象来提取更多经编码的视频数据。

62. 根据权利要求 44 到 45 中任一项所述的方法,其中,解释所述脚本使得所述应用程序本地地多路分用所述视频数据和所述音频数据。

63. 根据权利要求 41 到 62 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接实现帧缓冲监视器,所述帧缓冲监视器监视帧图像缓冲器,以确定其状态,并且根据所述帧图像缓冲器的所确定的状态,确定是否将更多的所述经编码的视频数据解码。

64. 根据权利要求 41 到 63 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接将至少一个所述帧图像的图像文件格式从第一图像文件格式转换成第二图像文件格式,所述第二图像文件格式能够由所述应用程序本地地呈现。

65. 根据权利要求 41 到 64 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接将至少一个所述帧图像转换成内容传输格式,所述内容传输格式能够由解释所述脚本的脚本解释器传递给所述应用程序。

66. 根据权利要求 41 到 64 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接将至少一个所述帧图像转换成 base64 内容传输格式。

67. 根据权利要求 41 到 66 中任一项所述的方法,其中,解释所述脚本使得所述应用程序将至少一个所述帧图像本地地转换成相应的图像对象,所述图像对象能够由应用程序本地地呈现。

68. 根据权利要求 41 到 67 中任一项所述的方法,其中,所述应用程序包括网络浏览器。

69. 根据权利要求 1 到 68 中任一项所述的方法,其中,所述脚本嵌入 HTML 文档中。

70. 根据权利要求 43 所述的方法,其中,所述一个或多个帧图像包括多个帧图像,并且其中,解释所述脚本,通过重放所述经解码的音频数据,来呈现所述音频数据,并且通过选择所述多个帧图像的一个子集和显示所述多个帧图像的所选择的一个子集,来呈现所述视频数据。

71. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,改变包括可由所述用户计算平台访问的经编码的视频数据的文件。

72. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,修改以下的一个或多个中的至少一个特征:将所述经编码的数据解码以及显示所述一个或多个帧图像。

73. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本,确定由所述用户计算平台接收所述经编码的视频数据的数据速率,并且根据所述确定,改变包括可由所述用户计算平台访问的经编码的视频数据的文件。

74. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本确定由所述用户计算平台接收所述经编码的视频数据的数据速率,并且根据所述确定,修改以下一个或多个中的至少一个特征:将所述经编码的视频数据解码以及显示所述一个或多个帧图像。

75. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括:

服务器计算装置,用于将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释;并且

其中,发送所述脚本使得所述脚本由在所述用户计算平台上操作的所述应用程序解释,以使得所解释的脚本直接将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据。

76. 根据权利要求 75 所述的系统,包括根据权利要求 2 到 40 所述的任一个特征或特征的组合。

77. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括处理器,该处理器被配置为提供:

解码部件,用于将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;以及

其中,所述处理器被配置为通过在所述用户计算平台上执行所述应用程序,来提供所述解码部件,所述应用程序解释使得所解释的脚本直接解码所述视频数据的跨平台脚本。

78. 根据权利要求 77 所述的系统,包括根据权利要求 42 到 74 所述的任一个特征或特征的组合。

79. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

其中,发送所述脚本使得所述脚本由在所述用户计算平台上操作的所述应用程序解释,以:重放可由所述用户计算平台访问的音频数据;通过显示构成所述视频数据的帧图像,来呈现可由所述用户计算平台访问的视频数据;确定与所述音频数据的重放相关的音频重放时间基准;并且使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像

的显示与所述音频数据的重放。

80. 根据权利要求 79 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

81. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 79 和 80 所述的任一个特征或特征的组合。

82. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用所述应用程序,来解释所述脚本;

其中,解释所述脚本:重放可由所述用户计算平台访问的音频数据;通过显示构成视频数据的帧图像,来呈现可由所述用户计算平台访问的所述视频数据;确定与所述音频数据的重放相关的音频重放时间基准;并且使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述音频数据的重放。

83. 根据权利要求 82 所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

84. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括根据权利要求 82 和 83 所述的任一个特征或特征的组合。

85. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代调度和执行包括将可由所述用户计算平台访问的经编码的视频数据直接解码成包括一个或多个帧图像的经解码的视频数据的第一模拟线程;并且迭代调度和执行包括通过显示所述一个或多个帧图像来呈现所述视频数据的第二模拟线程,来模拟多线程执行环境。

86. 根据权利要求 86 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

87. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 85 和 86 所述的任一个特征或特征的组合。

88. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用所述应用程序,来解释所述脚本;

其中,解释所述脚本:通过:迭代调度和执行包括将可由所述用户计算平台访问的经编码的视频数据直接解码成包括一个或多个帧图像的经解码的视频数据的第一模拟线程;并且迭代调度和执行包括通过显示所述一个或多个帧图像来呈现所述视频数据的第二模

拟线程,来模拟多线程执行环境。

89. 根据权利要求 88 所述的方法,包括根据权利要求 41 和 74 所述的任一个特征或特征的组合。

90. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括根据权利 88 和 89 所述的任一个特征或特征的组合。

91. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

其中,发送所述脚本,使得所述脚本由所述应用程序解释,以通过显示构成视频数据的帧图像,来呈现可由所述用户计算平台访问的视频数据;以及

其中,通过显示所述帧图像来呈现所述视频数据,包括在显示第一个所述帧图像与显示第二个所述帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像,包括 α 混合所述第一个帧图像和所述第二个帧图像。

92. 根据权利要求 91 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

93. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 91 和 92 所述的任一个特征或特征的组合。

94. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用所述应用程序,来解释所述脚本;

其中,解释所述脚本,通过显示构成所述视频数据的帧图像,来呈现可由所述用户计算平台访问的视频数据;以及

其中,通过显示帧图像来呈现所述视频数据,包括在显示第一个所述帧图像与显示第二个所述帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像,包括 α 混合所述第一个帧图像和所述第二个帧图像。

95. 根据权利要求 94 所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

96. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括根据权利 94 和 95 所述的任一个特征或特征的组合。

97. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

编码一系列帧图像,以提供经编码的视频数据,编码这系列帧图像,包括将所述帧图像

编码成内容传输格式, 据此, 所述内容传输编码的帧图像可由所解释的脚本传递给所述应用程序;

将视频数据发送给所述用户计算平台; 以及

其中, 发送所述脚本, 使得所述脚本由所述应用程序解释, 以将所发送的编码视频数据解码成包括这系列帧图像的经解码的视频数据, 并且显示所述帧图像。

98. 根据权利要求 97 所述的方法, 包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

99. 一种用于在用户计算平台上呈现视频内容的系统, 所述用户计算平台是多个不同用户计算平台类型中的一种, 所述系统包括根据权利要求 97 和 98 所述的任一个特征或特征的组合。

100. 一种由计算机系统执行的方法, 用于在用户计算平台上呈现视频内容, 所述用户计算平台是多个不同用户计算平台类型中的一种, 所述方法包括:

将脚本发送给所述用户计算平台, 所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释, 所述应用程序在所述用户计算平台上进行操作;

编码一系列帧图像, 以提供经编码的视频数据, 其中, 编码这系列帧图像, 包括:

对于以第一文件格式设置的至少第一个所述帧图像: 分析所述第一个帧图像的内容; 根据所述第一个帧图像的内容的分析, 在多个二次图像文件格式之中选择一个; 并且如果所述第一文件格式与所述二次图像文件格式中所选择的一个格式不同, 那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的一个格式; 并且

对于以第二文件格式设置的至少第二个帧图像: 分析所述第二个帧图像的内容; 根据所述第二个帧图像的内容的分析, 在所述多个二次图像文件格式之中选择不同的一个; 并且如果所述第二文件格式与所述二次图像文件格式中所选择的一个格式不同, 那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的一个格式;

其中, 所述多个二次图像文件格式中的每个能够由所述应用程序本地地呈现;

将视频数据发送给所述用户计算平台; 以及

其中, 发送所述脚本使得所述脚本由所述应用程序解释, 以将所发送的编码视频数据解码成包括这系列帧图像的经解码的视频数据, 并且显示所述帧图像。

101. 根据权利要求 100 所述的方法, 包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

102. 一种用于在用户计算平台上呈现视频内容的系统, 所述用户计算平台是多个不同用户计算平台类型中的一种, 所述系统包括根据权利要求 100 和 101 所述的任一个特征或特征的组合。

103. 一种由计算机系统执行的方法, 用于在所述用户计算平台上呈现视频内容, 所述用户计算平台是多个不同用户计算平台类型中的一种, 所述方法包括:

将脚本发送给所述用户计算平台, 所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释, 所述应用程序在用户计算平台上进行操作;

其中,发送所述脚本使得所述脚本由所述应用程序解释,以:将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;通过在所述用户计算平台上显示一个或多个帧图像,来呈现所述经解码的视频数据;查询所述用户计算平台的一个或多个特征;并且根据所述用户计算平台的所述一个或多个查询的特征,修改以下中的至少一个:包括可由所述用户计算平台访问的经编码的视频数据的文件;将所述经编码的数据解码的特征;以及显示所述一个或多个帧图像的特征。

104. 根据权利要求 103 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

105. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 104 和 105 所述的任一个特征或特征的组合。

106. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在是用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用所述应用程序,来解释所述脚本;

其中,解释所述脚本:将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据;查询所述用户计算平台的一个或多个特征;并且根据所述用户计算平台的所述一个或多个查询的特征,修改以下中的至少一个:包括可由所述用户计算平台访问的所述经编码的视频数据的文件;将所述经编码的数据解码的特征;以及显示所述一个或多个帧图像的特征。

107. 根据权利要求 106 所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

108. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括根据权利要求 106 和 107 所述的任一个特征或特征的组合。

109. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

其中,发送所述脚本使得所述脚本由所述应用程序解释,以:将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据;确定将所述经编码的视频数据发送给所述用户计算平台的速率;并且根据所述确定,修改以下中的至少一个:包括发送给所述用户计算平台的所述经编码的视频数据的文件;将所述经编码的数据解码的特征;以及显示所述一个或多个帧图像的特征。

110. 根据权利要求 109 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

111. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同

用户计算平台类型中的一种,所述系统包括根据权利要求 109 和 110 所述的任一个特征或特征的组合。

112. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用应用程序,来解释所述脚本;

其中,解释所述脚本:将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据;确定由所述用户计算平台接收所述经编码的视频数据的速率;并且根据所述确定,修改以下中的至少一个:包括由所述用户计算平台接收的所述经编码的视频数据的文件;将所述经编码的数据解码的特征;以及显示所述一个或多个帧图像的特征。

113. 根据权利要求 112 所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

114. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括根据权利 112 和 113 所述的任一个特征或特征的组合。

115. 一种计算机系统,包括使用数据通信网络有效地连接的一个或多个计算机,所述数据通信网络适合于执行根据权利要求 1 到 40、79、80、85、86、91、92、97、98、100、101、103、104、109 以及 110 所述的任一种方法。

116. 根据权利要求 75、76、81、87、93、99、102、105、111 中任一项所述的系统,其中,所述系统包括使用数据通信网络有效地连接的一个或多个计算机。

117. 一种计算机可读介质,包括计算机可执行代码,所述计算机可执行代码在由包括使用数据通信网络有效地连接的一个计算机或多个计算机的计算机系统执行时,使得所述计算机系统执行根据权利要求 1 到 40、79、80、85、86、91、92、97、98、100、101、103、104、109 以及 110 所述的任一种方法。

基于脚本的视频呈现

技术领域

[0001] 本发明涉及用于使用所解释的脚本来在用户计算装置上跨平台呈现(render)视频内容的方法和系统。特定的实施方式提供用于使用嵌入 HTML 文档内的脚本来跨平台呈现视频内容的方法和系统,在由网络浏览器的脚本解释器解释时,这些方法和系统使得所解释的脚本将经解码的视频数据直接解码成帧图像,并且使得网络浏览器本地地(natively)呈现帧图像。特定的实施方式提供用于将视频内容编码的方法和系统。

背景技术

[0002] 通常使用多个术语,例如,视频内容、视频媒体、视听媒体、视听内容、多媒体、富媒体等来描述数字视频。在该说明书中,视频内容应理解为包括可由视频数据、音频数据或与音频数据结合的视频数据体现或者可包括视频数据、音频数据或与音频数据结合的视频数据的任何这种内容和 / 或媒体。在某些情况下,视频数据和 / 或音频数据可与其他数据构成组,例如,图像数据、元数据等。除非在上下文中另有规定,否则音频内容应理解为可能包括这种图像数据、元数据等。通过非限制性实例,视频内容可通过互联网或某种其他类型的数据网络(例如,通过主机服务器或对等装置)流传输或发送、缓存(由 ISP、代理服务器或某个其他中间装置缓存)、本地缓存(例如,由在用户计算装置上运行的网络浏览器缓存)或本地储存或可本地进入用户计算装置。

[0003] 目前,使用各种技术和方案,来编码、发送以及呈现在互联网上(或以其他方式)可利用的数字视频内容。具有与目前可用的视频呈现技术相关的多个缺点。

[0004] 使用典型的先有技术的数字视频技术来在用户计算装置上分布和执行视频内容的一个缺点在于,必须在用户计算装置上下载和安装或者必须由用户计算装置平台(用户计算装置平台包括用户计算装置硬件和操作系统软件)预先安装专用应用程序(通常称为媒体播放器)。在该说明书中,应用程序称为编译的程序(可执行对象模块和 / 或其他编译代码段),该程序能够独立地并且无需另一个应用程序的协助来单独执行应用、功能或操作。应用程序与下面进一步描述的附加组件形成对比,这是因为附加组件取决于主机应用程序,以提供其功能,然而,应用程序能够独立地执行其功能。而且,并非所有独立的程序都是应用程序。例如,虚拟机(例如,Java 虚拟机)并非应用程序,这是因为这种虚拟机仅仅提供虚拟环境(例如,虚拟操作系统和虚拟硬件),并且不执行独立的应用、功能或操作。这种虚拟机要求应用程序(例如,编译的 Java 字节码应用程序)执行任何应用、功能或操作。一些用户不能下载或安装单独的媒体播放器。考虑到通常与下载文件(尤其与可执行应用程序)相关的安全性和 / 或隐私威胁(例如,病毒、恶意软件、本地共享对象等),一些用户不愿意下载单独的媒体播放器。

[0005] 此外,通常下载和安装媒体播放器应用程序,作为编译的对象代码,因此,在市场上可具有每个版本的用户计算装置平台需要一个不同的编译对象。考虑到移动智能手机的发展和多样性以及各种个人电脑平台,必须支持的对象平台的数量繁多,这是因为编译版本的媒体播放器可能不在并非其定向的硬件和 / 或操作系统平台上进行操作。在可与媒

体播放器冲突的其他安装的应用程序可损害媒体播放器与平台的兼容性时,复杂度增大。

[0006] 一些用户计算装置硬件和 / 或软件平台不支持一些媒体播放器应用(反之亦然),并且媒体播放器应用不能用于这种硬件和软件平台。而且,希望下载和安装媒体播放器的用户通常需要其用户计算装置硬件和 / 或软件平台的一些知识,来下载正确的播放器。甚至在已经下载和安装特定的播放器时,除了与下载和安装媒体播放器相关的问题以外,呈现视频内容还需要执行媒体播放器应用,这通常消耗大量计算资源(例如, RAM、CPU 时间等)。

[0007] 一些视频内容呈现技术使用“隐藏的播放器”,这表示在用户计算装置上进行操作的主机应用程序(例如,网络浏览器等)自动下载和安装媒体播放器附加组件。在该说明书中,术语附加组件应理解为包括附加组件、插件、管理单元、扩展部分、小应用程序等。附加组件是编译的程序(即,可执行对象代码模块、其他编译的代码段等),这些编译的程序将特定的功能(例如,视频内容呈现)加入“主机”应用程序中。虽然附加组件可执行一些功能,但是附加组件取决于其主机应用程序的可操作性。可与主机网络浏览器应用程序相结合使用的附加组件的实例包括:媒体播放器附加组件、PDF 阅读器附加组件、Java 虚拟机(JVM)附加组件等。

[0008] 在某些情况下,媒体播放器附加组件(或使得用户计算装置下载和安装媒体播放器附加组件的指令)可与视频内容文件捆绑在一起。在其他情况下,视频分布技术使得用户计算装置的网络浏览器单独下载媒体播放器附加组件和视频内容文件。由于实际上必须在执行之前下载和安装这种媒体播放器附加组件,所以这种媒体播放器附加组件具有与其明显的媒体播放器对应物的缺点相似的缺点。一些应用程序(例如,网络浏览器、抗病毒程序等)或操作系统软件可具有安全性和 / 或隐私设置(由其用户、其网络管理员等设置),该安全性和 / 或隐私设置阻止自动下载这种媒体播放器附加组件。在下载和安装媒体播放器附加组件之后,执行媒体播放器附加组件,消耗大量处理资源。作为编译的模块,必须为可打开的每个可能的对象平台,创造和测试一个不同版本的附加组件。

[0009] 通常需要通过避免在用户计算装置上下载和安装、预先安装或编译单独的媒体播放器或媒体播放器附加组件的方式,来呈现视频内容。通常需要通过与平台无关的或者具有较高度跨平台可操作性的方式,分布和执行视频内容。

[0010] 在流媒体视频内容的背景下,使用当前技术来在用户计算装置上分布和执行内容,具有另一个缺点,其中,当前技术通常使用流媒体内容服务器,每当用户计算装置请求视频内容流时,该服务器通过用户计算装置执行某种“握手”协议。与使用内容服务器和相关的握手协议相关的缺点在于,可防止本地(或下游)缓存视频内容。这种本地或下游缓存可节省带宽(以及相关的成本)并且提高流的可靠性(从而提高视频质量)。

[0011] 在需要更新一个或多个 CODEC 时,使用当前技术来在用户计算装置上分布和执行视频内容,具有另一个缺点。CODEC 是一种方案,用于将音频或视频数据编码,以减小其传输所需要的带宽,然后,用于在传输的另一侧将数据解码。通常,使用 CODEC 包括在传输侧上压缩数据并且在接收侧上将数据解压。解码还可包括在到达接收的计算机装置时解析和拆开数据,和 / 或将数据重新组织成可由接收的计算机装置使用的格式。先有技术的视频呈现技术通常包括使用编译的计算机程序模块,在执行时,该模块执行解码工序。如果在传输侧使用更新的 CODEC 来将视频内容编码,那么通常不能在用户计算装置上将视频内容解

码,除非在用户计算装置上下载和安装用于该 CODEC 的新更新的解码程序。

[0012] 各种更新的网络浏览器应用程序能够使用 HTML5<video>tag 来本地地呈现嵌入网页内的视频。遗憾的是, CODEC 支持目前并非在所有浏览器上均匀。在 HTML5 标准正在进行的发展中,继续争论(如果有的话)应本地地由 HTML5 兼容浏览器可呈现哪些视频格式。虽然可提供多份视频内容,供不同的 CODEC 使用,但是这通常需要将视频内容从一个 CODEC 转码成另一个 CODEC。虽然能够转码视频内容,以便根据不同的 CODEC 进行编码,但是这具有缺点。例如:

[0013] • 转码需要时间和精力;

[0014] • 转码通常是在计算上密集的工序,以便目前在技术上难以提供实时转码和 / 或在计算上比较昂贵地提供实时转码;

[0015] • 在提前转码视频内容,以根据不同的 CODEC 进行解码时,通常必须储存多份视频内容(每个不同的 CODEC 具有一份),对于大量视频内容,这较为繁重;

[0016] • 专有一些 CODEC,并且其合法使用受到限制或者需要支付许可费;以及

[0017] • 为根据不同的 CODEC 解码的视频内容服务,需要使用不同的流媒体服务器。

[0018] 因此,通常需要呈现视频内容的方法和系统,这通过现有视频内容呈现技术克服或减轻了这些和其他缺点中的一些缺点。

[0019] 相关技术以及与其相关的限制的上述实例旨在进行说明,而不排外。在阅读说明书以及研究附图时,对于本领域的技术人员,相关技术的其他限制显而易见。

附图说明

[0020] 在示图的参考图中,说明了示例性实施方式。其目的在于,在本文中公开的实施方式和示图被视为进行说明,而非进行限制。

[0021] 在说明本发明的非限制性实施方式的示图中:

[0022] 图 1 为示出根据一个特定实施方式的脚本的功能元件的示意性方框图,该脚本可嵌入 HTML 文档内并且可由应用程序(例如,网络浏览器)解释,以呈现视频内容;

[0023] 图 2A 为根据一个实例实施方式的视频呈现系统的示意性方框图;

[0024] 图 2B 为根据一个实例实施方式的音频呈现系统的示意性方框图;

[0025] 图 3 为根据一个实例实施方式的视频数据排队和解码方法的流程图;

[0026] 图 4 为根据一个实例实施方式的视频数据呈现方法的流程图;

[0027] 图 5 为根据一个实例实施方式的音频数据排队和解码方法的流程图;

[0028] 图 6 为根据一个特定的实施方式的用于实现音频呈现器接口的方法的流程图;

[0029] 图 7 为根据另一个实例实施方式的用于将音频和视频数据排队和解码、呈现视频数据并且实现音频呈现器接口的方法的流程图;

[0030] 图 8 为根据一个特定的实例实施方式的用于提取视频数据的方法的流程图;

[0031] 图 9A 和 9B 为根据特定的实例实施方式的用于下采样帧插补的技术的示意图;

[0032] 图 9C 为用于制备根据图 9A 和 9B 适用于帧插补的根据一个特定的实施方式的 α 混合(alpha-blend)的图像的方法的方框图;以及

[0033] 图 10 为用于呈现视频内容的方法的方框图,其中,根据用户计算装置的一个或多个特征和 / 或在服务器与用户计算装置之间的网络连接,在解释脚本时,该脚本适合呈现

视频内容。

具体实施方式

[0034] 在以下描述中,提出具体细节,以便为本领域中的技术人员提供更彻底的理解。然而,未详细显示或描述众所周知的部件,以免本公开不必要地晦涩难懂。因此,该说明书和附图具有说明而非限制的意义。

[0035] 在本公开和专利权利要求中,以下技术术语具有在下面提出的相应定义。

[0036] • 应用程序表示编译的程序(可执行对象代码模块、其他编译的代码段等),该程序能够独立地并且无需另一个应用程序的协助来单独执行应用、功能或操作。

[0037] • 附加组件表示编译的程序(即,可执行对象代码模块、其他编译的代码段等),该程序将特定的功能(例如,视频内容呈现)加入“主机”应用程序中。

[0038] • 脚本表示一系列非编译的跨平台指令,这些指令通常(而非必要地)为人类可读并且由应用程序的脚本解释器解释,无需编译。

[0039] • 在脚本由应用程序解释以便“直接”执行功能时,这表示解释包含功能表示的脚本,以执行功能本身,而不依靠由应用程序(或某个其他应用程序或附加组件)提供以便执行功能的预先编译的模块。

[0040] • 在应用程序(例如,网络浏览器)“本地地”执行功能时,这表示应用程序自动执行功能(可能在计算装置硬件和/或操作系统的协助下),但是无需依靠附加组件。通过请求其自身的预先编译的代码模块,而非通过附加组件的预先编译的代码模块,应用程序可本地地执行功能。应用程序可在脚本的指导下本地地执行功能。

[0041] 本发明的特定实施方式提供了用于使用跨平台脚本在用户计算装置上呈现视频内容的方法和系统,该脚本由应用程序(例如,网络浏览器)解释,以本地地呈现体现视频内容的视频和/或音频数据。在解释脚本时,可由脚本直接执行视频呈现工序的特定功能。在本说明书中,术语脚本表示一系列非编译的跨平台指令,这些指令通常(而非必要地)为人类可读并且由应用程序的脚本解释器解释,无需编译。在机器代码内执行之前,编译典型的应用程序和附加组件。由于未编译脚本,所以脚本与传统的编译代码应用程序和附加组件形成对比。脚本也与 Java 字节码不同。在解释之前,部分预先编译字节码,并且在解释时,将字节码编译到机器代码内。与此相反,在解释时,未将脚本编译到机器代码内。在本说明书中,在解释脚本时,这表示应用程序的脚本解释器读取脚本指令并且执行脚本指令,无需将脚本指令编译到机器代码内。

[0042] 通过非限制性实例,在一些实施方式中,可将脚本嵌入超文本标记语言(HTML)文档内,并且在网络浏览器应用程序呈现 HTML 文档,从而显示 HTML 页面时,该脚本可由在用户计算装置上运行的网络浏览器应用程序解释。在用户计算装置上进行操作的浏览器可从在互联网上的位置中接收 HTML 文档,该位置由输入到浏览器内的统一资源定位器(URL)规定。大部分网络浏览器应用程序包括本地地解释脚本的一个或多个本地脚本解释器。通过非限制性实例,脚本可写入跨平台脚本语法(例如,ECMAScript、JavaScript、Jscript、ActionScript)、在 HTML 标准(例如,HTML5)中固有的脚本语法等内。在本说明书中,在应用程序(例如,网络浏览器)本地地执行功能或操作时,这表示应用程序本身执行功能(可能在计算装置硬件和/或操作系统的协助下),但是无需依靠附加组件。通过请求其

自身的预先编译的代码模块,而非通过附加组件的预先编译的代码模块,应用程序可本地地执行功能。

[0043] 本发明的特定方面提供脚本,脚本可由应用程序(例如,网络浏览器)解释,以使得浏览器在用户计算装置上本地地呈现视频数据。在一些实施方式中,用户计算装置可使用一个不同的应用程序(例如,而非网络浏览器),该应用程序可包括本地脚本解释器或者能够解释脚本。在这种实施方式中,脚本可由其他这种应用程序解释,并且在这样进行解释时,可使得其他这种应用程序在用户计算装置上本地地呈现视频数据。为了简单起见并且不失一般性地,该应用的剩余部分可表示由网络浏览器解释的脚本,以使得网络浏览器本地地呈现视频数据。网络浏览器的引用应理解为包括其他应用程序,这些应用程序并非网络浏览器,但是可包括一个或多个脚本解释器,或者能够本地地解释脚本。通过非限制性实例,这种脚本可(但并非必须)位于使用超文本传输协议(HTTP)传达的文档内。

[0044] 在一些实施方式中,应用程序可解释脚本,从而使得所解释的脚本直接执行一个或多个功能(与视频和/或音频呈现相关)。在本说明书中,在脚本由应用程序解释,以直接执行功能时,这表示解释脚本,以执行功能本身。在脚本由应用程序解释,以直接执行功能时,所解释的脚本包含功能表示并且不依靠由应用程序(或某个其他应用程序或附加组件)提供以便执行功能的预先编译的模块。例如,在一些实施方式中,网络浏览器可解释脚本,以使得所解释的脚本直接:执行解码功能,该功能将视频内容从其编码的格式解码成帧图像;将帧图像重新格式化为可由网络浏览器本地地呈现的格式;和/或控制或同步视频和音频呈现的时间。

[0045] 特定的实施方式包括使用嵌入 HTML 文档内的脚本,该脚本由网络浏览器解释,以使得所解释的脚本将视频数据直接解码成一系列单独的帧图像并且在用户计算装置的显示器上本地地显示这系列帧图像,作为一系列位图。一些实施方式同样涉及另外使用脚本,该脚本由网络浏览器解释,以使得所解释的脚本将音频数据直接解码并且在用户计算装置上本地地重放经解码的视频数据。在视频内容包括视频数据和音频数据时,脚本可由网络浏览器解释,以使帧图像的显示与经解码的音频数据的重放同步,从而通过用户期望体验的方式呈现视听视频内容(例如,电影)。

[0046] 特定的实施方式提供用于将视频内容编码的方法和系统,这些方法和系统适用于上述基于脚本的视频呈现技术。

[0047] 计算机和相似的用户计算装置通常理解可在微处理器类型之间变化的机器或对象代码。软件工程师通常通过所谓的“高级”语言(例如,C)来编程,然后使用提前转换高级代码(通常称为源代码)机器或对象代码的编译器。通过高级语言编程,然后将人类可读的源代码编译成经编译的可执行版本,其缺点在于,编译版本的程序不便于携带或者并非“跨平台”。创建编译的代码,专门用于对象用户计算装置(用户计算装置的硬件和操作系统)。如上所述,在某些情况下,这可能成为问题。

[0048] 为了解决该问题,已经提出了跨平台计算机语言,例如,Java。在用户计算装置上接收之前,将该计算机语言部分预先编译成所谓的字节码,并且该计算机语言期望用户计算装置可调用额外的预先编译的程序(称为 Java 虚拟机(JVM)),该程序会理解字节码。作为编译的代码,在用户计算装置上解释字节码的 JVM 特别用于用户计算平台。JVM 可能不能用于特定的用户计算装置平台或者对特定的平台可具有有限的功能。因此,Java (以及

依靠虚拟机的相似的计算机语言)的使用仅仅将跨平台可操作性问题从单独的程序转变成 JVM。即,预先编译的程序不需要以各种特定的用户计算平台为对象,JVM 需要专门以各种用户计算平台为对象。

[0049] 与编译的语言和部分编译的语言(例如,Java)不同,未编译脚本。脚本包括根据各种脚本语法(例如,ECMAScript、JavaScript、Jscript、ActionScript、在 HTML 标准中固有的脚本语法等)编写的一系列指令(例如,脚本标签)。并未编译脚本指令,确切地说,由在用户计算装置上进行操作的应用程序解释脚本指令(并未编译),以使得应用程序执行某些动作。使用脚本(特别是用于处理器密集的任务)带来的困难在于,脚本的解释较慢(例如,在与编译的程序相比时),这是因为在运行时间进行解释。因此,就发明人所知,包括一系列脚本标签的脚本迄今还未用于使得在用户计算装置上进行操作的网络浏览器本地地呈现视频内容。如上所述,先有技术的视频呈现技术依靠已经为特定的硬件和软件平台编译的预先编译的媒体播放器、附加组件、JVM 等。

[0050] 图 1 为示出根据一个特定实施方式的脚本 10 的功能元件的示意性方框图,该脚本用于在用户计算装置(在图 1 中未明确显示)呈现视频内容。脚本 10 可包括在跨平台脚本语法(例如,通过非限制性实例)、ECMAScript、JavaScript、Jscript、ActionScript 等内写入的一系列脚本标签。写入呈现的引擎脚本 10 的脚本语法优选地由主要的操作系统、网络浏览器和 / 或其他应用程序广泛地支持,以便脚本 10 可独立于平台,从而在各种用户计算装置上进行操作(通过非限制性实例,台式电脑、膝上型电脑、平板电脑、智能电话、个人计算装置、专用装置等)。如上所述,用户计算装置可具有各种平台,并且脚本 10 能够在具有各种平台的各种用户计算装置上进行解释。脚本 10 可嵌入 HTML 文档内,以便在脚本 10 由在用户计算装置上进行操作的网络浏览器解释时,脚本 10 使得网络浏览器本地地提供在图 1 中所示的功能元件。在一些实施方式中,在用户计算装置上进行操作的网络浏览器可本地地解释脚本 10,该脚本可反过来使得网络浏览器本地地呈现视频数据,不要求用户计算装置下载、安装或执行用于 CODEC、媒体播放器附加装置或其他媒体播放器应用的任何额外的预先编译的解码器,以与网络浏览器一起或与网络浏览器相结合地运行。

[0051] 在一个特定的实施方式中,HTML 文档包括脚本 10 和视频内容文件的引用。脚本 10 可由在用户计算装置上进行操作的网络浏览器解释,以使得网络浏览器呈现视频内容。由解释脚本 10 呈现的视频内容可:通过网络通信链路(例如,从互联网中或者从能够通过主机服务器或对等装置与用户计算装置连通的某个其他网络中)流传输 12 (或发送);通过合适的网络通信链路从下游网络缓存 14 中获取(例如,由 ISP、代理服务器或某个其他中间装置缓存);本地缓存 15 (例如,由在用户计算装置上运行的网络浏览器缓存在本地缓存内或者缓存在可进入用户计算装置的本地缓存内);或者包含在本地文件 16 (例如,储存在可进入用户计算装置的本地存储器内的文件)内等。

[0052] 在进行解释时,脚本 10 可使得网络浏览器本地地提供数据接收部件 20,该部件接收(流传输或下载 12、从一个或多个下游缓存 14 中获取、本地缓存 15、本地储存 16 等 17) 视频内容,访问经编码的视频内容以及多路分用(或者分离)不同类型的数据(例如,视频数据 22 以及可选地音频数据 24、图像数据 26 以及元数据(未显示)),这些数据最初可包含在单个数据源内。

[0053] 通常,使用某个 CODEC 方案,将由数据接收模块 20 获取的视频数据 22 编码。因此,

在由浏览器解释时,所显示的实施方式的脚本 10 使得网络浏览器提供解码部件 30,该解码部件本地地进行操作,以将视频数据 22 解码,从而获得经解码的(并且通常解压的)视频数据 42。在一些实施方式中,可直接由脚本执行将功能解码的模块 30。由解码部件 30 实现的特定的解码和解压方案当然取决于用于编码和压缩视频数据 22 的编码方案或 CODEC。因此,脚本 10 的开发人员了解编码方案和相应的解码方案的规格。虽然并非必要,但是这些编码和解码方案(尤其是由解码部件 30 实现的解码方案)可包括开源或者公开可用的标准。

[0054] 与视频数据 22 一样,通常使用某个 CODEC 方案,将由数据接收模块 20 获取的音频数据 24 编码。因此,在执行时,所显示的实施方式的脚本 10 使得网络浏览器提供解码部件 32,该解码部件本地地进行操作,以将音频数据 24 解码,从而获得经解码的(并且通常解压的)音频数据 44。在一些实施方式中,可直接由脚本执行将功能解码的模块 32。由解码部件 32 实现的特定的解码和解压方案当然取决于用于编码和压缩音频数据 24 的 CODEC。因此,脚本 10 的开发人员了解编码方案和相应的解码方案的规格。虽然并非必要,但是这些编码和解码方案(尤其是由解码部件 32 实现的解码方案)可包括开源或者公开可用的标准。通过与上述视频数据 22 的方式相似的方式,可将专用格式中编码的压缩视频数据解码成公开可用的格式,以提供音频数据 24,以便通过执行公开可见的脚本 10,可实现解码部件 32。

[0055] 在进行解释时,脚本 10 还使得网络浏览器提供视频显示部件 50,该部件在用户计算装置的显示器 70 上从经解码的视频数据流 42 中本地地显示帧图像。一旦在解码模块 30 中将经编码的视频数据 22 解码,以提供经解码的视频流 42 (例如,包括一系列帧图像),就由显示部件 50 在用户计算装置的显示器上显示经解码的视频流 42。在一些实施方式中,显示部件 50 包括使用所谓的画布部件或画布标记。画布标记是 HTML5 标记,该标记允许在用户计算装置的显示器(例如,显示器 70)上动态地可编写脚本地呈现(具有规定的像素高度和宽度的)2D 形状和位图图像。画布标记可用于以足够快的帧速率来显示被人们视为视频的位图图像。

[0056] 在一些实施方式中,显示部件 50 包括使用画布标记,来显示经解码的视频数据流 42 的帧图像的子集。例如,在可选的同步模块 60 (下面更详细地解释)表示显示模块 50 正在以低于音频数据 44 的重放的速率显示位图图像的情况下和 / 或在可用的处理资源可能可取地显示经解码的视频流 42 的帧图像的子集的其他情况下,可丢弃经解码的视频流 42 的位图帧。在一些实施方式中,可使得帧图像坚持显示用户计算装置,例如,在同步功能 60 表示显示部件 50 正在以高于音频数据 44 的重放的速率显示图像时。在显示部件 50 包括使用画布标记,来显示经解码的视频流 42 的帧图像的子集时,显示部件 50 和 / 或同步模块 60 可包括从经解码的视频流 42 内选择特定的帧图像,用于进行显示。

[0057] 在一些实施方式中,执行脚本 10,使得网络浏览器提供显示模块 50,该显示模块的功能(例如,从经解码的视频流 42 中显示帧图像(例如,位图帧图像)并且可选地从经解码的视频流 42 内选择帧图像,用于进行显示)由网络浏览器本地地实现,而不调用或者执行任何额外的预先编译的应用程序或附加组件。

[0058] 由于继续增大数字装置的处理能力,所以由执行脚本 10 提供的显示部件 50 能够以帧速率显示位图图像,这些帧速率接近使用定制的视频重放应用的传统视频重放的帧速率。在一些实施方式中,显示部件 50 能够以大于 40 帧 / 秒的帧速率来显示 640x480 像素

位图图像。在一些实施方式中,显示部件 50 能够以大于 50 帧 / 秒的帧速率来显示 640x480 像素位图图像。

[0059] 在所显示的实施方式中,视频内容包括可选的图像数据 26。显示部件 50 还可使得在用户计算装置 70 上本地地显示图像数据 26。大部分网络浏览器包括本地功能,用于显示 JPEG 图像数据,在与显示部件 50 的供应相关的脚本 10 部分的指导下,该图像数据可由浏览器调用。

[0060] 在实现时,脚本 10 还提供音频重放部件 52,用于本地地重放经解码的音频流 44。在一些实施方式中,经解码的音频流 44 包括 .WAV 格式音频流。大部分网络浏览器包括本地功能,用于重放 .WAV 格式音频流,在与音频重放部件 52 相关的脚本 10 部分的指导下,这些音频流可由浏览器调用。该本地音频重放功能还可包括或使用与经解码的音频流 44 的重放相关的时间基准信息。可将该重放时间参考信息反馈给在下面描述的同步模块 60。

[0061] 在实现时,脚本 10 可选地使得网络浏览器提供同步模块 60,该同步模块在位于音频重放部件 52 重放经解码的音频流 44 与显示部件 50 显示经解码的视频流 42 的帧图像之间的可接受容差内保持同步。在一些实施方式中,同步模块 60 利用从音频重放模块 52 中获得的参考时间点(例如,开始重放音频数据 44)和时间戳信息(称为音频重放时间基准)。在一些实施方式中,同步模块 60 可选地使得显示部件 50 在选择帧图像用于进行显示之前,检查音频重放模块 52 的音频重放时间基准。然后,同步模块 60 可选地使得显示部件 50 选择与音频重放参考时间对应的帧图像,并且使用画布标记来显示所选择的帧图像,作为位图。如果音频重放部件 52 的音频重放参考时间表示经解码的音频流 44 的重放发生在从解压的视频数据 42 内显示帧图像之前,那么同步模块 60 可使得显示部件 50 跳过一个或多个帧图像。如果音频重放部件 52 的音频重放参考时间表示经解码的音频流 44 的重放发生在从解压的视频数据 42 内显示帧图像之后,那么同步模块 60 可使得显示部件 50 在更新帧图像时造成延迟(例如,使得一些帧图像持续存在)。通过这种方式,由显示部件 50 显示的视频数据可与音频重放部件 52 的音频重放同步。

[0062] 可通过例如图形用户接口为用户(未显示)提供控制面板 80 功能。控制面板 80 可为用户提供控制脚本 10 重放视频数据 42 和 / 或音频数据 44 的能力。通过非限制性实例,控制面板 80 可为用户提供控制重放数据、暂停重放数据、在数据内及时寻找位置、调节重放的音量、调节重放的速度等的功能。在所显示的实施方式中,在脚本 10 的外面提供控制面板 80,但是这不是很必要。在一些实施方式中,脚本 10 可使得网络浏览器提供控制面板 80。

[0063] 图 2A 为根据一个实例实施方式的视频呈现系统 100 的示意性方框图。在一些实施方式中,使用由通过在用户计算装置上进行操作的网络浏览器(或其他应用程序)体现的脚本解释器解释的跨平台脚本,至少部分实现视频呈现系统 100。在一些实施方式中,部分视频呈现系统 100 本地地由解释脚本的网络浏览器实现。在一些实施方式中,脚本直接执行或实现视频呈现系统 100 的一些功能。

[0064] 在所显示的实施方式中,视频呈现系统 100 获得视频内容 116。在一些实施方式中,视频内容 116 位于从网络上可获取的主机服务器、从网络上可获取的缓存或从网络上可获取的对等装置上。在其他实施方式中,视频内容 116 可在本地用于用户计算装置。在本说明书的剩余部分中,假设(不失一般性地)视频内容 116 位于主机服务器上。在特定的

实施方式中,通过将包含 URL 的 HTTP 请求发送给服务器,视频呈现系统 100 检索视频内容 116,该 URL 识别在服务器上托管的特定视频内容 116(例如,通过从服务器中开始“下载”或“流传输”视频内容 116)。如上所述,视频内容 116 可包括视频数据 114 以及其他数据,例如,音频数据 164、图像数据(未显示)、元数据(未显示)等。在所显示的实施方式中,提供给视频呈现系统 100 的视频内容 116 通常可包括任何这种数据,并且由视频呈现系统 100 多路分用,以访问包含在其内的视频数据 114。在其他实施方式中,在视频呈现系统 100 上由合适的多路分用器(未显示)接收,以分离视频数据 114 和视频内容 116 并且使视频呈现系统 100 直接访问多路分用的视频数据 114 之前,可多路分用视频内容 116。使用由在用户计算装置上进行操作的网络浏览器解释的跨平台脚本,还可至少部分实现这种多路分用器。

[0065] 在所显示的实施方式中,视频呈现系统 100 包括视频数据队列监视器 110,该监视器被配置为使得视频内容 116 的视频数据 114 储存在体现视频数据队列 112 的数据结构内并且监视视频数据队列 112。在一些实施方式中,在用户计算装置上进行操作的浏览器内实现视频队列监视器 110。在一些实施方式中,将视频数据 114 储存在视频数据队列 112 内,包括多路分用视频数据 114 和音频数据 164 以及可形成一部分视频内容 116 的其他数据(例如,图像、元数据等)。该多路分用功能可由视频数据队列监视器 110 实现。在其他实施方式中,视频数据队列监视器 110 不需要执行该多路分用功能。如上所述,在一些实施方式中,在由视频呈现系统 100 接收之前,视频数据 114 可由另一个多路分用器(未显示)多路分用。在另外一些实施方式中,储存在视频数据队列 112 内的数据包括(还未多路分用的)视频内容 116,并且视频数据 114 可与音频数据 164 和 / 或其他数据多路分用,同时由解码器 122 解码。

[0066] 系统 100 包括视频解码器 122。视频解码器 122 将视频数据 114 解码,以提供帧图像。在特定的实施方式中,在脚本由网络浏览器解释时,视频解码器 122 直接由脚本实现。系统 100 包括帧缓冲监视器 120。在所显示的实施方式中,帧缓冲监视器 120 被配置为使得解码器 122 将包含在视频数据队列 112 内的视频数据 114 解码成帧图像,并且使得帧图像处理 136 处理那些帧图像,从而产生帧图像 124,这些帧图像储存在体现帧图像缓冲器 126 的数据结构内并且具有适合于由帧图像呈现器 132 显示的格式。帧缓冲监视器 120 可由网络浏览器本地地实现。在一些实施方式中,在脚本由网络浏览器解释时,帧缓冲监视器 120 的功能直接由脚本实现。帧图像 124 包括一系列图像,在依次(例如,以可作为视频数据 114 的帧速率的合适的帧速率)呈现时,这些图像被人们视为视频。帧缓冲监视器 120 可被配置为使得解码器 122 和帧图像处理器 136 继续从视频数据 114 中解码和处理帧图像 124,直到在帧图像缓冲器 126 内至少具有阈值数量的帧图像 124(例如,直到帧图像缓冲器 126 充满)。

[0067] 在所显示的实施方式中,可选的帧图像处理器 136 通过视频解码器 122 接收帧图像 140(例如,从视频数据 114 中解码的帧图像 140),并且处理这种帧图像 140,以在适合于供帧图像呈现器 132 使用的格式中生成帧图像 124。在特定的实施方式中,在脚本由网络浏览器解释时,帧图像处理器 136 的一些功能直接由脚本实现。从解码器 122 中输出的帧图像 140 可包括任何合适图形文件格式的图像文件,包括通过非限制性实例:JPEG、GIF、PNG、TIFF、RAW、BMP 等。一些网络浏览器能够本地地显示可能的图像文件格式的有限子集。例如,几乎每个网络浏览器都能够本地地显示 JPEG 和 GIF 图像文件格式,但是很多网络浏览

器不能本地地显示具有 TIFF 文件格式的图像。在这种情况下,帧图像处理器 136 可包括帧图像转换器 142,该转换器将帧图像 140 转换成具有可由网络浏览器本地地显示的一个或多个图像文件格式(例如, GIF 和 / 或 JPEG)的浏览器可呈现的帧图像 144。在脚本由网络浏览器解释时,帧图像转换器 142 的功能可由脚本直接实现。要理解的是,在帧图像 140 已经具有可由在用户计算装置上进行操作的网络浏览器本地地显示的文件格式的情况下,不需要或者可绕过帧图像转换器 142。

[0068] 通常在所谓的二进制数据格式中表示帧图像 144。在从脚本解释器中传输或传送具有二进制数据格式的图像时,一些网络浏览器不能接受这种图像。在这种情况下,帧图像 144 可进一步由帧图像处理器 136 处理成适合于从脚本解释器中传送给网络浏览器的格式。在所显示的实施方式中,该处理功能由内容传输编码器 145 执行,该编码器处理帧图像 144,以生成内容传输编码的帧图像 146。在脚本由网络浏览器解释时,内容传输编码器 145 的功能可直接由脚本实现。在一个特定的实施方式中,内容传输编码器 145 包括 base64 编码器,该编码器处理帧图像 144,以生成 base64 编码的帧图像 146,这不是很必要。在其他实施方式中,其他内容传输编码技术可用于代替 base64 编码,并且可根据某种其他合适的内容传输编码格式,将帧图像 146 编码。

[0069] 然后,帧图像处理器 136 可使用内容传输编码的帧图像 146,来使用图像对象创建器 148 创建图像对象。在一些实施方式中,图像对象创建器 148 包括使得脚本解释器调用由网络浏览器本地地实现的画布部件等,以使用内容传输编码的帧图像 146,来创建具有 CanvasPixelArray 图像对象的形式帧图像 124,每个帧图像包含大量 RGB 像素值。在其他实施方式中,图像对象创建器 148 可使用内容传输编码的帧图像 146,来创建具有其他图像对象格式的帧图像 124。在一些实施方式中,可由图像对象创建器 148 (或者)使用具有以下形式的脚本指令来将内容传输编码的帧图像 146 传送给浏览器。

[0070] `image.src="data:image/jpg;base64,"+jpeg_encoded_in_content-transfer-format;`

[0071] 如上所述,帧图像 124 可具有适合于由帧图像呈现器 132 显示的格式,并且可储存在体现帧图像缓冲器 126 的数据结构内。

[0072] 在所显示的实施方式中,经解码的帧图像 140、浏览器可呈现的帧图像 144 以及内容传输编码的帧图像 146 显示为分别储存在经解码的帧图像缓冲器 138、浏览器可呈现的帧图像缓冲器 141 以及内容传输编码的帧图像缓冲器 143 内。在一些实施方式中,尤其(但并非限制性地)在视频解码器 122、帧图像转换器 142、内容传输编码器 145 和 / 或图像对象创建器 148 中的任一个的功能在离散的迭代中彼此独立地执行时(下面更详细地进行解释),可使用这种缓冲器 138、141、143。在其他实施方式中,这种缓冲器 138、141、143 并不必要,即,在每个迭代中,视频解码器 122 可创建单个解码的帧图像 140,帧图像转换器 142 反过来可将该单个解码的帧图像转换成单个相应的浏览器可呈现的帧图像 144, base64 编码器反过来可将该浏览器可呈现的帧图像编码成单个相应内容传输编码的帧图像 146,图像对象创建器 148 反过来可使用该内容传输编码的帧图像来创建单个帧图像对象 124,以加入帧图像缓冲器 126 中。

[0073] 所显示的实施方式的帧图像处理器 136 可选。在其他实施方式中,帧图像处理器 136 的任何功能可包含在解码器 122 内和 / 或视频重放接口 130 内(下面进一步描述)。通

过非限制性实例,帧图像转换器 142 的功能可包含在解码器 122 内,并且图像对象创建器的功能可包含在视频重放接口 130 内。在另一个非限制性实例中,所有帧图像处理器 136 的功能可包含在解码器 122 内。

[0074] 视频数据队列监视器 110 可被配置为在视频数据 114 已经由解码器 122 解码之后,从视频数据队列 112 中清除视频数据 114 (或视频内容 116)。在一些实施方式中,视频数据队列监视器 110 被配置为根据在已经由解码器 122 解码的视频数据队列 112 中的视频数据 114 的量,从视频内容源(例如,远程服务器)中请求视频内容 116 (和 / 或包含在其内的视频数据 114)。例如,每当视频数据队列 112 包含比已经由解码器 122 解码的视频数据 114 的阈值量(例如,与适当的阈值数量的帧图像 124 对应的视频数据的阈值量)更少的量时,视频数据队列监视器 110 可被配置为请求视频内容 116 和 / 或视频数据 114。在视频数据 114 包括一部分已知长度的视频段(例如,预先录制的节目、短影片、电影等)的情况下,或者在可辨别视频数据 114 的结尾时,视频数据队列监视器 110 可被配置为在已经将构成视频数据 114 的结尾的一部分视频数据 114 放入视频数据队列 112 内之后,停止请求视频内容 116 和 / 或视频数据 114。

[0075] 系统 100 包括视频重放接口 130。在脚本由网络浏览器解释时,视频重放接口 130 的功能可直接由脚本实现。视频重放接口 130 被配置为与帧呈现器 132 的帧图像 124 的显示(例如,音频数据的重放)同步,或者使得帧呈现器 132 在适当的时间显示帧图像 124。帧呈现器 132 可完全或部分由本地地位于在例如用户计算装置上进行操作的网络浏览器上的图像呈现引擎实现。帧呈现器 132 可包括调用 drawImage() 方法和 / 或将包括帧图像 124 的 Image Data 对象作为其(其中的一个)争论的 put Image Data 方法。

[0076] 在一些实施方式中,视频重放接口 130 被配置为使得帧呈现器 132 以大约等于视频数据 114 的帧速率的速率显示帧图像 124。在一些实施方式中,视频重放接口 130 被配置为使得帧呈现器 132 以小于视频数据 114 的帧速率的速率显示帧图像 124,并且可选地显示在帧图像 124 的显示之间插补的图像。在一些实施方式中,视频重放接口 130 可使得帧呈现器 132 以用户可配置的速率显示帧图像 124。视频重放接口 130 可被配置为在某个合适的容差内,使得帧呈现器 132 至少大致同步地呈现帧图像 124 以及呈现在视频内容 116 中伴随视频数据 114 的音频数据 164。由于视频重放接口 130 可控制帧呈现器 132 呈现帧图像 124 的时间,所以视频重放接口 130 可称为重放时间监视器 130。

[0077] 图 2B 为根据一个实例实施方式的音频呈现系统 150 的示意性方框图。在一些实施方式中,使用由通过用户在用户计算装置上进行操作的网络浏览器解释器解释的跨平台脚本,至少部分实现音频呈现系统 150。在一些实施方式中,部分音频呈现系统 150 本地地由解释脚本的网络浏览器实现。在一些实施方式中,脚本直接执行或实现音频呈现系统 150 的一些功能。在一些实施方式中,所有或部分视频呈现系统 100 (图 2A)和音频呈现系统 150 (图 2B)可彼此结合。在一些实施方式中,例如,在图 2A 和 2B 中所示的示例性实施方式中,可分别实现视频呈现系统 100 (图 2A)和音频呈现系统 150 (图 2B)。在某些方面,音频呈现系统 150 与上述视频呈现系统 100 相似。音频呈现系统 150 可通过与视频呈现系统 100 相同的方式检索视频内容 116。由视频呈现系统 100 检索的视频内容 116 可由音频呈现系统 150 获得,反之亦然。

[0078] 如上所述,提供给视频呈现系统 100 的视频内容 116 通常可包括视频数据 114、音

频数据 164、图像数据(未显示)、元数据(未显示)等。所显示的实施方式的音频呈现系统 150 包括音频数据队列监视器 160,该监视器被配置为使得视频内容 116 的音频数据 164 储存在体现音频数据队列 162 的数据结构内并且监视音频数据队列 162。音频数据队列监视器 160 可与视频呈现系统 100 的视频数据队列监视器 110 相似。在一些实施方式中,在用户计算装置上进行操作的网络浏览器内实现音频数据队列监视器 160。在一些实施方式中,将音频数据 164 储存在音频数据队列 162 内,包括多路分用音频数据 164 和视频数据 114 以及可形成一部分视频内容 116 的其他数据(例如,图像数据、元数据等)。该多路分用功能可由音频数据队列监视器 160 实现。然而,通常,音频数据队列监视器 160 不需要执行该多路分用功能。如上所述,在一些实施方式中,在由音频呈现系统 150 接收之前,音频数据 164 可由另一个多路分用器(未显示)多路分用。在另外一些实施方式中,储存在音频数据队列 162 内的数据还未多路分用,并且音频数据 164 可与视频数据 114 和 / 或其他数据多路分用,同时由解码器 172 解码。在这种实施方式中,视频数据队列 112 和音频数据队列 162 均可由可储存视频内容 116 的相同视频内容队列(未显示)实现。

[0079] 音频呈现系统 150 包括音频解码器 172。音频解码器 172 将音频数据 164 解码,以提供经解码的音频数据 174。在特定的实施方式中,在脚本由网络浏览器解释时,音频解码器 172 直接由脚本实现。所显示的实施方式的音频呈现系统 150 包括经解码的音频缓冲监视器 170。经解码的音频缓冲监视器 170 被配置为使得解码器 172 将包含在音频数据队列 162 内的音频数据 164 解码,以产生储存在体现经解码的音频缓冲器 176 的数据结构内的经解码的音频数据 174。经解码的音频缓冲监视器 170 可由网络浏览器本地地实现。在一些实施方式中,在脚本由网络浏览器解释时,经解码的音频缓冲监视器 170 的功能直接由脚本实现。经解码的音频缓冲监视器 170 可被配置为使得解码器 172 继续将音频数据 164 解码并且生成经解码的音频数据 174,直到在经解码的音频缓冲器 176 内至少具有阈值数量的经解码的音频数据 174(例如,直到经解码的音频缓冲器 176 充满)。经解码的音频数据 174 可包括具有可由在用户计算装置上进行操作的网络浏览器(例如,使用浏览器的本地音频重放功能)本地地呈现的格式的音频数据。在一些实施方式中,音频呈现系统 150 可包括音频数据处理器(未显示),该处理器包括一个或多个功能,以使用本地音频重放功能,将音频数据 164 转换成适合于供在用户计算装置上进行操作的网络浏览器使用的经解码的格式 174。在一些实施方式中,可由解码器 172 和 / 或音频呈现器接口 180 执行这种功能。

[0080] 音频数据队列监视器 160 可被配置为在音频数据 164 已经由解码器 172 解码之后,从音频数据队列 162 中清除音频数据 164(或视频内容 116)。在一些实施方式中,音频数据队列监视器 160 被配置为根据在已经由解码器 172 解码的音频数据队列 162 中的音频数据 164 的量,从视频内容源(例如,远程服务器)中请求视频内容 116(和 / 或包含在其内的音频数据 164)。例如,每当音频数据队列 162 包含比已经由解码器 172 解码的音频数据 164 的阈值量更少的量时,音频数据队列监视器 160 可被配置为请求视频内容 116 和 / 或音频数据 164。在可辨别音频数据 164 的结尾时,音频数据队列监视器 160 可被配置为在已经将构成音频数据 164 的结尾的一部分音频数据 164 放入音频数据队列 162 内之后,停止请求视频内容 116 和 / 或音频数据 164。

[0081] 音频呈现系统 150 包括音频呈现器接口 180。音频呈现器接口 180 被配置为将经解码的音频数据 174 提供给(例如,载入)具有适合于供音频呈现器 182 使用的格式的音频

呈现器 182。音频呈现器 182 可包括音频重放引擎,例如,该引擎本地地位于在用户计算装置上进行操作的网络浏览器内。在一些实施方式中,音频呈现器 182 可使用专用音频呈现硬件,并且可在与在主要的用户计算装置处理器上运行的工序平行的一个或多个工序中,呈现(例如,重放)音频数据。在一些实施方式中,音频呈现器 182 可使用主要的用户计算装置处理器来实现,但是依然可在与负责视频呈现系统 100 和音频呈现系统 150 的线程平行运行的一个或多个处理线程中实现。在一些实施方式中,音频呈现器接口 180 可选地被配置为确定与由音频呈现器 182 呈现经解码的音频数据 174 相关的重放时间信息(例如,音频重放时间基准)。由音频呈现器接口 180 获得的重放时间信息可由网络浏览器本地地获得。音频呈现器接口 180 的其他功能可直接由所解释的脚本执行。该重放时间信息可由帧图像呈现器 132 和 / 或视频重放接口 130 (图 2A)使用,以使得帧图像 124 由帧呈现器 132 以与由音频呈现器 182 呈现经解码的音频数据 174 大致同步的速率呈现。

[0082] 在一些实施方式中,能够多路分用音频数据 164 的整个流和视频内容 116,并且缓冲在音频数据队列 162 内的音频数据 164 的整个流。在这种实施方式中,音频数据队列监视器 160 不需要积极地监视音频数据队列 162 的状态,以确定检索更多的音频数据 164 的时间。相反,音频数据队列监视器 160 可对音频数据 164 做出单个请求。同样,在这种实施方式中,音频数据 164 的整个流可由解码器 172 解码,并且作为经解码的音频数据 174 储存在经解码的音频缓冲器 176 内。在这种实施方式中,经解码的音频数据监视器 170 不需要积极地监视经解码的音频缓冲器 176 的状态,而是对由解码器 172 解码的并且放入经解码的音频缓冲器 176 内的音频数据 164 的流做出单个请求。

[0083] 一些网络浏览器包括本地功能,该功能允许这些浏览器本地地呈现经编码的音频数据 164。通过非限制性实例,Firefox™ 可本地地处理 Vorbis 编码的音频格式,并且 Safari™ 和 Internet Explorer™ 可本地地处理 mp3 编码的音频格式。在这种实施方式中,不需要解码器 172、经解码的音频缓冲器 176 以及经解码的音频数据监视器 170,并且音频呈现器接口 180 可将经编码的音频数据 164 从音频数据队列 162 中直接提供给音频呈现器 182。

[0084] 在一些实施方式中,通过使用嵌入 HTML 文档内的跨平台脚本实现的一种或多种方法,至少部分实现视频呈现系统 100 和 / 或音频呈现系统 150 的功能。这种脚本可由脚本解释器解释,该脚本解释器体现为在用户计算装置上进行操作的应用程序的一部分。例如,这种脚本解释器可为网络浏览器的一部分。由脚本解释器解释这种脚本,可造成执行逻辑和 / 或算术运算以及例示和 / 或操纵数据结构,这些数据结构提供视频呈现系统 100 和 / 或音频呈现系统 150 的功能。

[0085] 在一些实施方式中,至少部分由实现两种方法的脚本解释器来实现视频呈现系统 100 的功能:一种方法提供系统 100 的视频数据排队、解码以及制备功能(例如,通过实现视频数据队列监视器 110、帧缓冲监视器 120、解码器 122 以及帧图像处理器 136),另一种方法提供系统 100 的帧图像同步和呈现功能(例如,通过实现视频重放接口 130 以及可选地帧图像呈现器 132)。图 3 示意性描述了根据一个特定的实施方式的用于使视频数据排队、解码以及制备视频数据的该第一种方法的实现方式。图 4 示意性描述了根据一个特定的实施方式的用于同步和呈现视频数据的帧图像的该第二种方法的实现方式。可使得脚本解释器通过异步的方式反复执行这两种方法,以递增地使视频数据排队、解码、制备、同步和呈现视

频数据。视频数据的这种递增的排队、解码、制备、同步和呈现可具有优点,其中,在由单线程解释器解释的脚本中实现这些方法。

[0086] 图 3 为根据一个实例实施方式的方法 200 的流程图。在浏览器解释跨平台脚本(或部分脚本)时,方法 200 可由在用户计算装置上进行操作的网络浏览器本地地实现。例如,方法 200 可作为一组脚本指令(例如,一组 Javascript 指令)来实现。在解释脚本时,方法 200 的特定功能可直接由脚本实现。在所显示的实施方式中,方法 200 在方框 204 中开始,该方框涉及查询是否至少使视频数据的阈值量排队,用于进行解码。在一些实施方式中,方框 204 包括确定视频数据队列(例如,系统 100 的视频数据队列 112 (图 2A))是否包含已经解码的视频数据的至少一个帧。

[0087] 如果方框 204 查询确定小于排队的视频数据的阈值量(方框 204,否支路),那么方法 200 继续进入方框 206。方框 206 包括查询是否已经使所有可用的视频数据排队。方框 206 可包括例如确定是否已经将视频段的尾部(例如,已知长度的视频段的尾部或者视频段的可辨别的尾部)放入视频数据队列内。如果方框 206 查询确定已经使所有可用的视频数据排队(方框 206,是支路),那么方法 200 结束。

[0088] 如果方框 206 查询确定依然可利用未排队的视频数据(方框 206,否支路),那么方法 200 继续进入方框 208。方框 208 包括使得将更多的视频数据放入视频数据队列内。通过非限制性说明,在特定的实例实施方式中,方法 200 作为一组 Javascript 指令来实现,并且方框 208 包括调用与在服务器上托管的视频数据对应的 AJAX XML Http Request 对象的 open() 方法。该方框 208 功能要求网络浏览器在可由 Javascript 解释的格式中检索一些视频数据。在用户计算装置内可能的情况下,检索视频数据的该方框 208 功能可由网络浏览器执行,作为与方法 200 的剩余部分的线程平行的线程(在多线程环境下)或者作为与方法 200 的剩余部分的工序平行的工序(在多工序环境下)。例如,网络浏览器的脚本解释器可执行一个平行的线程/工序,以实现方法 200,而网络浏览器实现第二平行的线程/工序,以实现方框 208。

[0089] 在一些实施方式中,方框 208 包括仅仅获得数量递增的视频数据(例如,通过规定将视频数据的特定部分或特定量放入视频数据队列内,例如,通过在 HTTP 请求的 Range 标题字段内指出字节范围)。在这种实施方式中,方框 208 可包括更新跟踪已经在视频数据队列内请求和/或放入哪个视频数据的指数、范围等。

[0090] 在方框 208 之后,方法 200 继续进入方框 216,下面进行描述。在一些实施方式中,方法 200 可在方框 208 之后继续进入方框 212。在一些实施方式中,省略方框 206 (例如,如果在方框 204 中确定排队的视频数据至少具有不止一个帧(方框 204,否支路),那么方法 200 可尝试使得视频数据排队(例如,执行方框 208),而不确定可利用的视频数据是否排队)。

[0091] 返回方框 204,如果方框 204 查询确定排队的视频数据具有至少一个帧(方框 204,是支路),那么方法 200 继续进入方框 212。方框 212 包括确定在帧图像缓冲器(例如,系统 100 的帧图像缓冲器 126 (图 2A))内是否具有充足的自由空间(例如,足以储存经解码的帧图像的自由空间)。如果方框 212 查询确定在帧图像缓冲器内没有自由空间(方框 212,否支路),那么方法 200 继续进入方框 216。如果方框 212 查询确定在帧图像缓冲器内具有自由空间(方框 212,是支路),那么方法 200 继续进入方框 204。

[0092] 在所显示的实施方式中,方框 214 包括将排队的视频数据的一个帧制备成帧图像,该帧图像适合于呈现(例如,帧图像 124 (图 2A))制备的帧图像并且将该帧图像放入帧图像缓冲器(例如,帧图像缓冲器 126 (图 2A))内。方框 214 可包括在例如一系列帧图像内制备与下一个未解码的帧图像对应的(或者包括该未解码的帧图像)视频数据。在一些实施方式中,在方框 214 中制备的帧图像包括具有一种格式的图像,该格式可由在用户计算装置上进行操作的网络浏览器本地地呈现,该用户计算装置解释实现所有或部分方法 200 的脚本。通过非限制性说明,在一个特定的实例实施方式中,方法 200 作为一组脚本指令(例如,一组 Javascript 指令)来实现,并且方框 214 包括将视频数据制备成 Image Data 对象,该对象可包括包含大量 RGB 像素值的 Canvas Pixel Array 对象。在另一个实例实施方式中,方框 214 包括将视频数据制备成具有一种格式(例如, PNG、JPEG、GIF 等)的帧图像,该格式可由在用户计算装置上进行操作的网络浏览器本地地显示。

[0093] 方框 214 通常可包括以下实例操作中的任一个或任一个组合:

[0094] • 执行算术和 / 或逻辑运算;

[0095] • 创建、删除和 / 或修改数据结构和 / 或储存在其内的数据;

[0096] • 执行一组或多组脚本指令,访问脚本解释器可本地地利用的可执行库等;

[0097] • 使得浏览器调用本地浏览器功能(例如,在排队的视频数据包括(或者可转换成) JPEG 图像数据时,方框 214 可涉及使得浏览器调用在用户计算装置上进行操作的浏览器的本地 JPEG 呈现功能,以获得大量 RGB 像素值);和 / 或

[0098] • 等类似操作。

[0099] 在所显示的实施方式的特定情况下,在方框 214 中制备帧图像包括:

[0100] • 在方框 214A 中,将视频数据(例如,视频数据缓冲器 112 的视频数据 114)解码,以生成经解码的帧图像(例如,经解码的帧图像 140);

[0101] • 在方框 214B 中,可选地将经解码的帧图像转换成可由网络浏览器本地地呈现的文件格式,已生成浏览器可呈现的帧图像(例如,浏览器可呈现的帧图像 144);

[0102] • 在方框 214C 中,将浏览器可呈现的帧图像可选地编码成内容传输格式,以生成内容传输编码的帧图像(例如,内容传输编码的帧图像 146);以及

[0103] • 在方框 214D 中,可选地创建帧图像的帧图像对象(例如,帧图像对象 124)。

[0104] 方框 214 还可选地包括经解码的帧图像的图像相关处理,例如,调节对比度、亮度、清晰度等。

[0105] 在方框 214 之后,方法 200 继续进入方框 216,该方框包括在未来的时间调度方法 200 的连续迭代。通过非限制性说明,在一个特定的实例实施方式中,方法 200 作为一组脚本指令(例如,一组 Javascript 指令)来实现,并且方框 216 包括调用 setTimeout() 方法,以在经过规定的时间量之后,调度实现方法 200 的这组 Javascript 指令。在另一个特定的实例实施方式中,方法 200 作为一组脚本指令(例如,一组 Javascript 指令)来实现,并且 setInterval() 方法可用于在经过规定的时间量之后,调度实现方法 200 的这组 Javascript 指令。在这种实施方式中,对于方法 200 的每个迭代,并非明确需要方框 216。在特定的实施方式中,以小于呈现视频数据的帧间隔(即, (帧速率)⁻¹) 的间隔,可调度方法 200 的连续迭代。在一些实施方式中,调度的方法 200 间隔小于呈现视频数据的帧间隔的 50%。在一些实施方式中,调度的方法 200 间隔小于呈现视频数据的帧间隔的 20%。在一些

实施方式中,调度的方法 200 间隔小于呈现视频数据的帧间隔的 10%。在方框 216 中调度使方法 200 排队和解码的下一个迭代之后,方法 200 结束,并且将控制传送回网络浏览器及其脚本解释器。

[0106] 在所显示的实例实施方式中,在涉及方框 214 的方法 200 的每个迭代中,方框 214 包括将视频数据的帧从其在视频数据缓冲器(例如,视频数据缓冲器 112)中的编码格式一直制备成适合于显示的格式(例如,在帧图像缓冲器 126 中的帧图像对象 124)。该帧图像处理可涉及多个步骤,这些步骤包括:解码(方框 214A)、图像格式转换(方框 214B)、内容传输编码(方框 214C)以及图像对象制备(方框 214D)。在某些情况下(例如,在用户计算装置处理资源比较有限或较慢时),在执行所有这些步骤时涉及的处理量不可取地较大。因此,在一些实施方式中,可修改方法 200,以便为到达方框 214 的方法 200 的每个迭代执行方框 214A、214B、214C、214D 的步骤的某个子集。例如,到达方框 214 的方法 200 的每个迭代可涉及执行仅仅一个方框 214A、214B、214C、214D。这就减少了与方法 200 的每个迭代相关的处理时间。可限定合适的逻辑,以在每个迭代中在方框 214A、214B、214C、214D 之间进行选择。通过非限制性实例,这种逻辑可确定哪个队列(例如,从队列 138、141、143、126 之中(图 2A))具有最少数量的帧图像,并且在此基础上可选择执行 214A、214B、214C、214D 中的哪一个。

[0107] 如上所述,在浏览器解释跨平台脚本(或部分脚本)时,方法 200 可由在用户计算装置上操作的网络浏览器本地地实现。在解释脚本时,方法 200 的特定功能可直接由脚本实现。例如,在特定的实施方式中,在解释脚本时,任何方框 204、206、212、214A、214B、214C 的功能可直接由脚本执行。

[0108] 图 4 为根据一个实例实施方式的用于同步和呈现排队的帧图像的方法 220 的流程图。在浏览器解释跨平台脚本(或部分脚本)时,方法 220 可由在用户计算装置上操作的网络浏览器本地地实现。例如,方法 200 可由一组脚本指令(例如,一组 Javascript 指令)实现。在解释脚本时,方法 220 的特定功能可直接由脚本实现。

[0109] 方法 220 的方框 224 包括确定呈现当前帧图像。方框 224 可包括例如根据呈现视频数据的帧速率以及当前时间值来确定当前视频帧索引。在可选的方框 224A 和 224B 中显示用于确定适用于方框 224 中的当前视频帧索引的方法的一个特定的非限制性实施方式。方框 224A 涉及从合适的源中获得当前时间值。在所显示的实施方式中,通过呈现在视频内容内伴随视频数据的音频数据,获得方框 224A 的当前时间值。在一个非限制性实施方式中,通过从视频内容 116 内呈现视频数据 164,获得方框 224A 的当前时间值(参照图 2B)。如上所述,在特定的实施方式中,音频呈现系统 150 的音频呈现器接口 180 (图 2B)可监视视频呈现器 182 进行音频重放的时间(例如,相对于参考点,例如,音频重放的开始等),并且可在方框 224A 中查询,用于当前视频重放时间基准。在某些这种实施方式中,音频呈现器接口 180 可由脚本直接实现,以获得视频重放时间基准。在其他实施方式中,音频呈现器接口 180 可利用本地浏览器功能,以获得视频重放时间基准。在另一个非限制性实例实施方式中,实现方法 220,作为一组脚本指令(例如,一组 Javascript 指令),并且方框 224A 可涉及使得浏览器在相关的视频对象上调用 `audio.buffered.end` 方法,以获得表示相关视频数据的最后呈现(或即将呈现)的区段的时间的值。该示例性实施方式可由浏览器本地地实现,并且不需要依靠音频呈现器接口 180。

[0110] 在所显示的实例实施方式中,可由脚本直接实现的方框 224B 涉及使方框 224A 的当前时间值(例如,当前音频重放时间基准)乘以视频数据帧速率,以获得期望方框 224 的当前视频帧索引。在一些实现方式中,方框 224A 的当前时间值与视频数据帧速率的乘积可四舍五入为最近的整数,以提供期望方框 224 的当前视频帧索引。在基于 Javascript 的实现方式中,在不能向上取整时,通过在乘积中加入 0.5 并且向下取整为最近的整数,可执行该方框 224B 的取整运算。

[0111] 在一些网络浏览器中,可能是以下情况:以比呈现音频数据的帧速率更小的频率(例如,在音频缓冲信息的连续更新之间可经过两个或多个帧周期),更新音频缓冲信息。为了改善由此引起的同步问题,方框 224 可包括确定自从执行最后的时间方框 224 (或方法 200 的另一个方框)以来经过的时间量,并且根据以下组合,确定当前方框 224 的视频帧索引:该经过时间;以及方框 224 的视频帧索引确定执行最后的时间方框 224。可通过系统时钟等,测量或确定该经过时间。

[0112] 可由脚本直接实现的方框 226 包括确定是否已经呈现由当前方框 224 的视频帧索引引用的帧图像。在单个视频帧的周期内将方法 220 执行两次的情况下,方框 226 可避免重复呈现相同的帧。在一些实施方式中,方框 226 包括比较当前方框 224 的视频帧索引和在方法 220 的前一个迭代(例如,恰好前一个迭代)中确定的方框 224 的视频帧索引。

[0113] 如果方框 226 查询确定已经呈现与当前方框 224 的视频帧索引对应的视频帧(方框 226,是支路),那么方法 220 继续进入方框 228。可由脚本直接实现的方框 228 包括确定当前方框 224 的视频帧索引是否引用所呈现的视频数据的最后帧(例如,已知长度的视频段的最后帧)。如果在方框 228 中确定当前方框 224 的视频帧索引引用在所呈现的视频数据中的最后帧(方框 228,是支路),那么方法 220 结束。如果方框 228 查询确定当前方框 224 的视频帧索引未引用在所呈现的视频数据中的最后帧(方框 228,否支路),那么方法 220 继续进入方框 230,下面进行描述。

[0114] 返回方框 226,如果方框 226 查询确定还未呈现与当前方框 224 的视频帧索引对应的视频帧(方框 226,否支路),那么方法 220 继续进入方框 232。可由脚本直接实现的方框 232 涉及查询与当前方框 224 的视频帧索引对应的帧图像(为了方便起见,“当前帧图像”)是否排队,用于进行呈现。例如,方框 232 可包括确定在由方法 200 构成的帧图像缓冲器内是否具有当前帧图像(图 3)。如果方框 232 查询确定当前帧图像未排队,用于进行呈现(方框 232,否支路),那么方法 220 继续进入方框 230。在继续进入方框 230 之前,方法 220 可在方框 234 中开始和/或执行一个或多个可选的丢帧动作。通过非限制性实例,方框 234 的丢帧动作可涉及记录当前帧图像不能用于进行呈现这一事实(例如,为了统计和/或分析的目的)、暂停重放相关音频轨道(例如,允许排队的帧赶上由音频重放确定的当前帧)、指示(例如,向另一个工序、方法等)帧图像应更快速地排队(例如,通过以更快的速率获得视频数据、更频繁地将帧解码、切换成更低的比特率或更低的帧速率视频源等)等。方框 234 丢帧动作可由脚本直接指出。执行一些可能的方框 234 丢帧动作,还可由脚本直接实现。执行其他可能的方框 234 丢帧动作,可涉及使用一个或多个本地浏览器功能。在一些实施方式中,可选的方框 234 可包括可用于确定是否应开始丢帧动作的某个逻辑。在一些实施方式中,在已知丢帧的其他情况下,例如,在方框 224 的当前帧索引比表示已经在方法 220 的连续迭代之间丢弃一个或多个帧的最后方框 224 帧索引(来自方法 220 的前一个迭代)大 1

个数目以上的情况下,可采取与方框 234 丢帧动作相似的动作。

[0115] 如果方框 232 查询确定当前帧图像排队,用于进行呈现(方框 232,是支路),那么方法 220 继续进入方框 236。方框 236 包括呈现当前帧图像。在使用由在用户计算装置上操作的网络浏览器解释的脚本实现方法 220 的实施方式中,方框 236 可包括例如使用网络浏览器的本机图像呈现功能来呈现当前帧图像。通过非限制性说明,在一个特定的实例实施方式中,实现方法 220,作为一组脚本指令(例如,一组 Javascript 指令),并且方框 236 包括调用将 ImageData 对象作为其(其中的一个)争论的 putImage Data 方法,该对象包括包含表示要呈现的帧图像(例如,帧图像 124 (图 2A))的大量 RGB 像素值的 Canvas Pixel Array 对象。在另一个实例实施方式中,方框 236 包括调用将图像对象(例如,帧图像 124 (图 2A))作为其(其中的一个)争论的 drawImage 方法。

[0116] 在方框 236 之后,方法 220 继续进入方框 230。方框 230 涉及在未来的时间调度方法 220 的连续迭代。在一些实施方式中,方框 230 可包括使用一个或多个本地浏览器功能。通过非限制性说明,在一个特定的实例实施方式中,方法 220 作为一组脚本指令(例如,一组 Javascript 指令)来实现,并且方框 230 包括调用 setTimeout() 方法,以在经过规定的时间量之后,调度实现方法 220 的这组 Javascript 指令。在另一个特定的实例实施方式中,方法 220 作为一组脚本指令(例如,一组 Javascript 指令)来实现,并且 setInterval() 方法可用于在经过规定的时间量之后,调度实现方法 220 的这组 Javascript 指令。在这种实施方式中,对于方法 220 的每个迭代,并非明确需要方框 230。在特定的实施方式中,以小于呈现视频数据的帧间隔(即,(帧速率)⁻¹)的间隔,可调度方法 220 的连续迭代。在一些实施方式中,预定的方法 220 间隔小于呈现视频数据的帧间隔的 50%。在一些实施方式中,预定的方法 220 间隔小于呈现视频数据的帧间隔的 20%。在一些实施方式中,预定的方法 220 间隔小于呈现视频数据的帧间隔的 10%。在方框 230 中调度视频呈现方法 220 的下一个迭代之后,方法 220 结束,并且将控制传送回脚本解释器。

[0117] 如上所述,在浏览器解释跨平台脚本(或部分脚本)时,方法 220 可由在用户计算装置上操作的网络浏览器本地地实现。在解释脚本时,方法 220 的特定功能可直接由脚本实现。例如,在特定的实施方式中,在解释脚本时,任何方框 224B、226、228、232、234 的功能可直接由脚本执行。

[0118] 在所显示的实施方式中,每当执行时,方法 200 制备视频数据的仅仅一个帧,并且每当执行时,方法 220 呈现仅仅一个帧图像。该实现方式可具有优点,其中,方法 200 和 220 由在单个执行线程中执行的应用程序的相同脚本解释器(例如,一些典型网络浏览器的 Javascript 解释器)重复实现并且异步解释。例如,这种实现方式可允许方法 220 呈现在方法 200 的连续迭代之间由方法 200 解码的帧图像。在其他实施方式中,在方法 200 的每个迭代中,可获得和 / 或制备视频数据的多个帧,和 / 或在方法 220 的每个迭代中,可呈现多个帧图像。而且,如上所述,在一些实施方式中,在方法 200 的每个迭代中,可部分制备视频数据的一个帧图像(例如,方框 214A、214B、214C、214D 的一个子集)。

[0119] 在一些实施方式中,方法 200 和 220 被配置为定期调度其自身的连续迭代(例如,在连续迭代之间预定的周期在迭代上可恒定)。这些预定的间隔可比在方法 200 和 220 的连续迭代之间可经过的最大时间更短,依然允许以视频数据的帧速率来使视频数据排队、制备、同步以及呈现视频数据。在方法 200 和 220 由脚本实现的实施方式中,这较为有利,

该脚本由应用程序的脚本解释器解释,该应用程序不能保证方法 200 和 220 在其预定的时间内发生未来迭代(例如,由于解释器的操作由在相同的执行线程上运行的其他工序作为解释器来优先占用,由于方法 200 和 220 的未来迭代由解释器的其他预定的和 / 或进行的操作优先占用等)。如果方法 200 和 220 在专用线程上运行,那么比严格上所需要的更频繁地调度的连续迭代可提供帮助确保及时执行方法 200 和 220 的余地,以提供令人类观察员满意的视频显示。

[0120] 方法 200 和 220 可被配置为具有相同的重复间周期或不同的重复间周期。在一些实施方式中,方法 200 被配置为比方法 220 更频繁地执行。通过非限制性说明,在一个特定的实例实施方式中,方法 200 被配置为在 1ms 与 10ms 之间的预定重复间周期内调度连续的迭代,并且方法 220 被配置为在 5ms 与 25ms 之间的预定重复间周期内调度连续的迭代。在其他实施方式中,方法 220 可被配置为在 1ms 与 10ms 之间的预定重复间周期内调度连续的迭代。

[0121] 可动态地确定方法 200 和 / 或 220 的重复间周期。例如,根据在连续的迭代之间实际上经过的时间,方法 200 和 / 或 220 可动态地确定其重复间周期。再如,可协调方法 200 和 220 的重复间周期(例如,方法 200 可决定方法 200 的重复间周期,反之亦然)。在一些实施方式中,可重复例示方法 200 和 220 中的一者或两者,以便具有一种或这两种方法中的多个方法,连续地调度其自身的连续迭代(例如,在相同的重复间周期或在不同的重复间周期)。

[0122] 现在转向音频呈现系统 150 (图 2B),在一些实施方式中,至少部分由实现两种方法的脚本解释器来实现音频呈现系统 150 的功能:一种方法提供系统 150 的音频数据排队和解码功能(例如,通过实现音频数据队列监视器 160、经解码的音频缓冲监视器 170 以及解码器 172),另一种方法提供音频呈现器接口(例如,音频呈现器接口 180),用于与音频呈现器 182 接合。图 5 示意性描述了根据一个特定的实施方式的用于使音频数据排队和解码的该第一种方法的实现方式。图 6 示意性描述了根据一个特定的实施方式的用于实现音频呈现器接口的该第二种方法的实现方式。可使得脚本解释器通过异步的方式反复执行这两种方法,以递增地使音频数据排队、解码以及制备视频数据并且与音频呈现器 182 接合。音频数据的这种递增的排队、解码以及与音频呈现器 182 的接合可具有优点,其中,在由单线程解释器解释的脚本中实现这些方法。

[0123] 图 5 为根据一个实例实施方式的用于获得、解码以及制备音频数据的方法 250 的流程图。在浏览器解释跨平台脚本(或部分脚本)时,方法 250 可由在用户计算装置上进行操作的网络浏览器本地地实现。例如,方法 250 可作为一组脚本指令(例如,一组 Javascript 指令)来实现。在解释脚本时,方法 250 的特定功能可直接由脚本实现。在很多方面,方法 205 可与图 3 的视频数据排队、解码以及制备方法 200 相似。在所显示的实施方式中,方法 250 在方框 254 中开始,该方框涉及查询是否至少使音频数据的阈值量排队,用于进行解码。在一些实施方式中,方框 254 包括确定音频数据队列(例如,系统 150 的音频数据队列 162 (图 2B))是否包含已经解码的音频数据的至少一个区段。在方法 250 中(以及在下面描述的方法 270 中)使用的音频段的长度可包括可配置的常数(例如,可取决于在用户计算装置上可利用的处理资源)或者可包括在动态上确定的变量,每当使用音频数据的区段时,或者每当呈现一块音频数据 164 时,确定该变量。在一些实施方式中,虽然这并非必须,但

是在方法 250、270 中使用的每个音频段可包括视频数据的适当数量(例如,1 个)的帧。在一些实施方式中,在方法 250、270 中使用的每个音频段可包括重放时间的适当长度(例如,1 秒)。在一些实施方式中,在方法 250、270 中使用的每个音频段可包括适当数量(例如,44,000)的音频样品。

[0124] 如果方框 254 查询确定小于排队的音频数据的阈值量(方框 254,否支路),那么方法 250 继续进入方框 256,该方框包括查询是否已经使所有可用的音频数据排队。方框 256 可包括例如确定是否已经将音频数据的尾部储存在音频数据队列内。如果方框 256 查询确定已经使所有可用的音频数据排队(方框 256,是支路),那么方法 200 结束。如果方框 256 查询确定依然可利用未排队的音频数据(方框 256,否支路),那么方法 250 继续进入方框 258,该方框包括使得将更多的音频数据放入音频数据队列内。方框 258 可与方法 200 的方框 208 (图 3)相似,除了获得音频数据(而非视频数据)以及将音频数据加入音频数据队列(而非视频数据队列)内以外。

[0125] 在方框 258 之后,方法 250 继续进入方框 266,下面进行描述。在一些实施方式中,方法 250 可在方框 258 之后继续进入方框 262。在一些实施方式中,省略方框 256(例如,如果在方框 254 中确定排队的视频数据具有不止一个区段(方框 254,否支路),那么方法 250 可尝试使得音频数据排队(例如,执行方框 258),而不确定可利用的音频数据是否排队)。

[0126] 返回方框 254,如果方框 254 查询确定排队的音频数据具有至少一个区段(方框 254,是支路),那么方法 250 继续进入方框 262,该方框包括确定在经解码的音频缓冲器(例如,系统 150 的经解码的音频缓冲器 176 (图 2B))内是否具有充足的自由空间(例如,足以储存音频数据的经解码的区段的自由空间)。如果方框 262 查询确定在经解码的音频缓冲器内没有自由空间(方框 262,否支路),那么方法 250 继续进入方框 266。如果方框 262 查询确定在经解码的音频缓冲器内具有自由空间(方框 262,是支路),那么方法 250 继续进入方框 264。

[0127] 方框 264 包括将排队的音频数据的一个区段(例如,来自音频数据队列 162)解码,并且在经解码的音频缓冲器(例如,经解码的音频缓冲器 176)内储存经解码的音频段。例如,方框 264 可包括与在音频轨道中的下一个未解码的区段对应(或者包括该区段)的解码音频数据。在一些实施方式中,在可由在实现本地音频呈现器 182 的用户计算装置上操作的网络浏览器本地地呈现的格式中,提供在方框 264 中解码的音频数据。通过非限制性说明,在一个特定的实例实施方式中,方框 264 包括将音频数据解码成 .WAV 文件格式等,该文件格式可由在用户计算装置上操作的网络浏览器本地地呈现。方框 264 可包括执行任何合适的解码算法的步骤,包括(而非限制性地)上面结合方框 214 描述的那些功能。如上所述,音频呈现系统 150 (图 2B)可包括与帧图像处理器 136 (图 2A)的功能相似的音频处理功能,该处理器可用于将经解码的音频数据放入适合于进行呈现的格式内。在这种实施方式中,方框 264 可选地中断,以通过与方框 214 的可选方框 214A、214B、214C、214D 相似的方式提供这种功能(图 3)。引用在方法 250 中的解码音频数据、解码方框 264 以及解码器 172 (图 2B),应理解为可选地包括与将视频数据制备成适合于进行呈现的格式相关的这些功能中的任一个或多个。方框 264 可选地包括经解码的音频数据的音频相关的处理,例如,过滤、音量调节、压缩等。

[0128] 在方框 264 之后,方法 250 继续进入方框 266,该方法涉及在未来的时间调度方法

250 的下一个迭代。方框 266 可与上述方框 216 相似。以比在方法 250、270 中使用的音频段的重放持续时间更小的间距,调度方法 250 的连续迭代。例如,如果在方法 250、270 中使用的音频段具有重放持续时间 τ ,那么方框 266 可涉及以小于 0.5τ 的间距调度方法 250 的连续迭代。在一些实施方式中,预定的方法 250 间距小于 0.2τ 。在一些实施方式中,预定的方法 250 间距小于 0.1τ 。在方框 266 中调度使方法 250 排队和解码的下一个迭代之后,方法 250 结束,并且将控制传送回脚本解释器。

[0129] 如上所述,在浏览器解释跨平台脚本(或部分脚本)时,方法 250 可由在用户计算装置上操作的网络浏览器本地地实现。在解释脚本时,方法 250 的特定功能可直接由脚本实现。例如,在特定的实施方式中,在解释脚本时,任何方框 254、256、262、264 的功能可直接由脚本执行。

[0130] 图 6 为根据一个特定的实施方式的用于实现音频呈现器接口的方法 270 的流程图。在浏览器解释跨平台脚本(或部分脚本)时,方法 270 可由在用户计算装置上进行操作的网络浏览器本地地实现。例如,方法 270 可作为一组脚本指令(例如,一组 Javascript 指令)来实现。在解释脚本时,方法 270 的特定功能可直接由脚本实现。方法 270 在方框 272 中开始,该方框涉及查询音频数据的最后区段是否已经从经解码的音频缓冲器(例如,经解码的音频缓冲器 176 (图 2B))中装载到音频呈现器(例如,音频呈现器 182 (图 2B))中。如果模块 272 查询肯定(即,已经将所有经解码的音频数据提供给音频呈现器(方框 272,是支路)),那么方法 270 通过下面描述的可选方框 274 继续进入方框 276。如果在另一方面,方框 272 查询确定在经解码的音频缓冲器内依然具有经解码的音频数据,以提供给音频呈现器(方框 272,否支路),那么方法 270 继续进入方框 280。

[0131] 方框 280 涉及查询音频呈现器是否可接受更多的音频数据(例如,在音频呈现器的内部缓冲器内是否具有接受音频数据的另一个区段的空间)。如果方框 280 查询确定音频呈现器不能接受更多的音频数据(方框 280,否支路),那么方法 270 通过下面描述的可选方框 274 继续进入方框 276。另外,如果音频呈现器可接受更多的音频数据(方框 280,是支路),那么方法 270 继续进入方框 282。方框 282 涉及查询是否具有在经解码的音频缓冲器内可利用的经解码的音频数据,以装载到音频呈现器内。如果方框 282 查询确定在经解码的音频缓冲器内可利用充足的经解码的音频数据(方框 282,是支路),那么方法 270 继续进入方框 284,该方框涉及将经解码的音频数据的区段从经解码的音频缓冲器中传输到音频呈现器内。该方框 284 数据传输可涉及在适合于供音频呈现器使用的格式中将音频数据格式化。在方框 284 中将音频数据装载到音频呈现器内之后,方法 270 通过可选的方框 274 继续进入方框 276。

[0132] 返回方框 282,如果方框 282 查询确定在经解码的音频缓冲器内不能利用充足的经解码的音频数据(方框 282,否支路),那么方法 270 通过可选的方框 286 和 274 继续进入方框 276。可选的方框 286 可涉及一个或多个丢弃音频动作,例如:记录以下事实:音频呈现器具有接受更多的经解码的视频数据的能力,但是在经解码的视频缓冲器内不能利用这种经解码的视频数据;音频呈现器暂停或减速重放音频数据(例如,允许音频数据的解码和/或排队赶上音频呈现器)、指示(例如,向另一个工序、方法等)音频数据应更快速地解码和/或排队(例如,通过以更快的速率获得视频数据、更频繁地将音频数据解码、切换成更低的比特率音频源等)等。

[0133] 方法 270 可通过各种路径在可选的方框 274 处结束。可选的方框 274 涉及从音频呈现器中获得当前时间值。该当前时间值可包括与音频数据的当前区段的重放相关的时间基准。通过非限制性说明,在一个特定的实例实施方式中,方法 270 作为一组脚本指令(例如,一组 Javascript 指令)来实现,并且可选的方框 274 可涉及在相关的音频对象上调用 `audio.buffered.end` 方法,以获得表示相关的音频数据的最后呈现(或即将呈现)的区段的时间的值。在可选的方框 274 中获得的时间值可用于方法 220 的方框 224A (图 4)。在一些实施方式中,由于在方法 220 的方框 224A 中获得时间值,所以不需要方框 274。

[0134] 方法 270 最后进入方框 276,该方框涉及查询音频呈现器是否完成音频数据的重放。如果方框 276 查询肯定(即,音频重放完成),那么方法 270 结束。如果在另一方面,音频重放未完成(方框 276,否支路),那么方法 270 继续进入方框 278,该方框涉及在未来的时间调度方法 270 的下一个迭代。方框 278 可与上述方框 230(图 4)相似。以比在方法 250、270 中使用的音频段的重放持续时间更小的间距,调度方法 270 的连续迭代。例如,如果在方法 250、270 中使用的音频段具有重放持续时间 τ ,那么方框 278 可涉及以小于 0.5τ 的间距调度方法 270 的连续迭代。在一些实施方式中,预定的方法 270 间距小于 0.2τ 。在一些实施方式中,预定的方法 270 间距小于 0.1τ 。在方框 278 中调度方法 270 的下一个迭代之后,方法 270 结束,并且将控制传送回脚本解释器。

[0135] 如上所述,在浏览器解释跨平台脚本(或部分脚本)时,方法 270 可由在用户计算装置上操作的网络浏览器本地地实现。在解释脚本时,方法 270 的特定功能可直接由脚本实现。例如,在特定的实施方式中,在解释脚本时,任何方框 272、280、282、286、276 的功能可直接由脚本执行。

[0136] 在所显示的实施方式中,每当执行时,方法 250 获得并且解码(以及另外制备)音频数据的仅仅一个区段,并且每当执行时,方法 270 将音频数据的仅仅一个区段载入音频呈现器内。如上所述,这种区段的长度可为可配置的参数,该参数可例如取决于在用户计算装置上可利用的处理资源。在一些实施方式中,每个音频段包括音频数据的帧。该实现方式可具有优点,其中,方法 250 和 270 由在单个执行线程中执行的应用程序的相同脚本解释器(例如,一些主要的浏览器的 Javascript 解释器)重复实现并且异步解释。例如,这种实现方式可允许方法 270 在方法 250 的连续迭代之间将音频数据(由方法 250 解码)载入音频呈现器内。在其他实施方式中,在方法 250 的每个迭代中,可获得和/或解码(或者制备)音频数据的多个区段,和/或在方法 270 的每个迭代中,可呈现多个区段。而且,如上所述,在一些实施方式中,在方法 250 的每个迭代中,可部分制备音频数据的一个区段(例如,与方框 214A、214B、214C、214D 相似的方框的一个子集)。

[0137] 方法 250、270 的调度可具有与上述方法 200、220 的调度的特征相似的特征。预定间隔可恒定、动态、协调、足够短,以允许使音频数据以音频数据的采样率或者以适合于听者的速率排队、解码以及呈现。方法 250、270 可被配置为具有相同或不同的重复间周期。在一些实施方式中,方法 250 被配置为比方法 270 更频繁地执行。在一些实施方式中,可重复例示方法 250、270 中的一者或两者,以便具有一种或这两种方法中的多个方法,连续地调度其自身的连续迭代(例如,在相同的重复间周期或在不同的重复间周期)。

[0138] 如上所述,在一些实施方式中,网络浏览器可本地地呈现经编码的音频数据。在这种实施方式中,可对方法 250、270 进行适当的修改,以便不需要与使音频数据解码和使经

解码的音频数据排队相关的步骤。

[0139] 在上面结合图 3 到 6 描述的实施方式中,通过调度反过来由单线程解释器执行的方法 200、220、250、270 的连续迭代,模拟多线程处理。图 7 为根据另一个实例实施方式的用于将音频和视频数据排队、解码以及制备这些数据、呈现视频数据并且实现音频呈现器接口的方法 300 的流程图。使用方法 300,涉及迭代调度和解释单个方法 300 代码段的连续迭代,但是将逻辑并入方法 300 代码段内,以确定在每个迭代中执行哪个程序。

[0140] 方法 300 包括逻辑方框 302,在所显示的实施方式中,该逻辑方块包括多个逻辑查询 304、308、312、316。在一些实施方式中,在解释脚本时,逻辑方框 302 或逻辑方框 302 的特定功能可直接由脚本实现。逻辑方框 302 可规定在方法 300 中执行哪个程序。例如,在所显示的实施方式中,逻辑方框 302 在方框 304 中开始,方框 304 涉及查询条件是否适合于执行音频呈现器接口程序。方框 304 条件通常可包括基于可本地地或直接用于脚本的任何信息的条件。例如,方框 304 条件可基于比较在音频呈现器的缓冲器内的音频数据的量与某个阈值。如果确定在音频呈现器缓冲器内的音频数据的量小于某个阈值量,那么可取地执行音频呈现器接口程序,以将更多的音频数据载入音频呈现器内。方框 304 条件不限于通常与音频呈现器接口或者与音频呈现直接相关的条件。在一些实施方式中,方框 304 条件可涉及评估是否更迫切地需要某种其他功能。例如,即使在音频呈现器缓冲器内的音频数据的量小于阈值量,方框 304 条件也可涉及评估执行视频解码程序是否更重要(在方法 300 的当前迭代中)。方框 304 条件可基于在丢弃音频方框 286 (图 6)和 / 或丢帧方框 234 (图 4) 中生成的信息。

[0141] 如果方框 304 条件的评估确定条件适合于执行音频呈现器接口程序(方框 304,是支路),那么方法 300 继续进入方框 306。方框 306 涉及执行音频呈现器接口程序。这种音频呈现器接口程序可涉及与方法 270 (图 6)相似的方法,除了不需要方框 278 调度未来的迭代以外。在从音频呈现器接口程序中返回之后,方法 300 继续进入方框 320,该方框包括在未来的时间调度方法 300 的连续迭代。方框 320 调度可与上面为方法 200、220、250、270 的调度方框描述的调度相似。例如,在方法 300 作为一组脚本指令(例如,一组 Javascript 指令)来实现时,方框 320 可包括使用 `setTimeout()` 和 / 或 `setInterval()` 方法。方框 320 预定间隔可为静态或动态。方框 320 预定间隔可比可经过的最大时间更短,依然允许以适当的速率来使视频和音频数据排队、解码和 / 或制备和呈现视频和音频数据。在一些实施方式中,方框 320 预定间隔在 0.25ms-25ms 的范围内。在一些实施方式中,在完成方法 300 的当前迭代之后,尽快调度方框 320 预定间隔。

[0142] 如果方框 304 条件的评估确定条件不适合于执行音频呈现器接口程序(方框 304,否支路),那么方法 300 继续进入方框 308。方框 308 涉及查询条件是否适合于执行视频呈现器程序。方框 308 查询可与方框 304 查询相似,并且可基于可本地地或直接用于脚本的任何信息。通过非限制性实例,方框 308 查询可基于评估当前帧索引(例如,通过与上面在方框 224 中描述的方式相似的方式获得的(图 4))以及确定当前帧索引是否大于呈现的最后视频帧的帧索引。如果当前帧索引大于呈现的最后视频帧,那么可取地执行视频呈现器程序,以呈现视频数据的另一个帧。与上述方框 308 条件一样,上述方框 308 条件可涉及评估是否更迫切地需要某种其他功能。方框 304 条件可基于在丢弃音频方框 286 (图 6)和 / 或丢帧方框 234 (图 4)中生成的信息。如果方框 308 条件的评估确定条件适合于执行视频

呈现程序(方框 308,是支路),那么方法 300 继续进入方框 310。方框 310 涉及执行视频呈现程序。这种视频呈现程序可涉及与方法 220 (图 4)相似的方法,除了不需要方框 230 调度未来的迭代以外。在从视频呈现器程序中返回之后,方法 300 继续在方框 320 中调度另一个迭代。

[0143] 如果方框 308 条件的评估确定条件不适合于执行视频呈现器程序(方框 308,否支路),那么方法 300 继续进入方框 312。方框 312 涉及查询条件是否适合于执行视频排队/解码程序。方框 312 查询可与方框 304、308 的查询相似,并且可基于可本地地或直接用于脚本的任何信息。通过非限制性实例,方框 312 查询可基于比较在帧图像缓冲器(例如,帧图像缓冲器 126 (图 2A))内制备的视频数据的量与合适的阈值和/或比较在视频数据队列中的视频数据的量(例如,视频数据队列 112 (图 2A))与合适的阈值。与方框 304、308 的条件一样,方框 312 条件可涉及评估是否更迫切地需要某种其他功能。方框 312 条件可基于在丢弃音频方框 286 (图 6)和/或丢帧方框 234 (图 4)中生成的信息。如果方框 312 条件的评估确定条件适合于执行视频排队/解码程序(方框 312,是支路),那么方法 300 继续进入方框 314。方框 314 涉及执行视频排队/解码程序。这种视频排队/解码程序可涉及与方法 200 (图 3)相似的方法,除了不需要方框 216 调度未来的迭代以外。在从视频排队/解码程序中返回之后,方法 300 继续在方框 320 中调度另一个迭代。

[0144] 如果方框 312 条件的评估确定条件不适合于执行视频排队/解码程序(方框 312,否支路),那么方法 300 继续进入方框 316。方框 316 涉及查询条件是否适合于执行音频排队/解码程序。方框 316 查询可与方框 304、308、312 的查询相似,并且可基于可本地地或直接用于脚本的任何信息。通过非限制性实例,方框 316 查询可基于比较在经解码的音频缓冲器(例如,经解码的音频缓冲器 176 (图 2B))内的经解码的音频数据的量与合适的阈值和/或比较在音频数据队列中的音频数据的量(例如,音频数据队列 162 (图 2B))与合适的阈值。与方框 304、308、312 的条件一样,方框 316 条件可涉及评估是否更迫切地需要某种其他功能。方框 316 条件可基于在丢弃音频方框 286 (图 6)和/或丢帧方框 234 (图 4)中生成的信息。如果方框 316 条件的评估确定条件适合于执行音频排队/解码程序(方框 316,是支路),那么方法 300 继续进入方框 318。方框 318 涉及执行音频排队/解码程序。这种音频排队/解码程序可涉及与方法 250 (图 5)相似的方法,除了不需要方框 266 调度未来的迭代以外。在从音频排队/解码程序中返回之后,方法 300 继续在方框 320 中调度另一个迭代。

[0145] 在所显示的实施方式中,如果方框 316 条件的评估确定条件不适合于执行视频排队/解码程序(方框 316,否支路),那么方法 300 继续沿着两个可选路径中的一个。在某些情况下,方框 316 的否支路可使得方法 300 环回方框 304。在其他情况下,方框 316 的否支路可使得方法 300 继续在方框 320 中调度另一个迭代。在其他实施方式中,方框 316 的否支路可将方法 300 引入另一个功能方框(未显示),该功能方框可涉及选择一个程序(例如,方框 306、310、314、318)。这种选择可基于方框 304、308、312、316 的条件中的哪一个最靠近正面结果。在另外一些实施方式中,不需要方框 316,并且方框 312 的否支路可直接引入方框 318。在这种实施方式中,方框 318 的视频排队/解码程序是默认程序。在其他实施方式中,可构成逻辑方框 302 的逻辑,以便一个其他程序(例如,一个方框 306、310、314)是默认程序。

[0146] 在某些情况下,方法 200、220、250、270、300 的一个特定的功能块可能需要可造成呈现假象的过长的时间段。例如,方法 200 (图 3)的处理器密集的功能块是方框 214,该方框涉及制备视频数据的帧。如果方框 214A、214B、214C、214D 的所有功能集中到方框 214 内,那么方框 214 尤其是处理器密集的。在某些情况下,尤其在具有较小的处理容量的装置上,在方框 214 中制备视频数据的帧所需要的时间可比相关视频数据的帧间间距(即,(帧速率)⁻¹)更长。在这种情况下,在可能另外呈现视频数据的帧时,由于处理器致力于在方框 214 中将视频数据的帧解码,所以可能略过或丢失视频帧,造成视频假象。

[0147] 在一些实施方式中,方法 200、220、250、270、300 可解析或细分成子元件(例如,功能子模块或子步骤),并且在这种子元件之间,可执行与视频和 / 或音频呈现相关的功能(例如,视频呈现方法 220 和 / 或音频呈现方法 270)。继续视频制备方框 214 的实例,视频制备方框 214 可细分成多个子元件,这些子元件可包括子元件 214A、214B、214C、214D 或者可包括其他(显示的或未显示的)子元件。在一些实施方式中,子元件 214A、214B、214C、214D 本身可细分成进一步的子元件。在方框 214 的特定子元件之后,可评估条件,以确定是否应中断方框 214,以执行与视频和 / 或音频呈现相关的程序(例如,视频呈现方法 220 和 / 或音频呈现方法 270)。这种中断条件可与上面为方框 304、308、312、316 描述的那些条件相似,并且通常可基于可本地地或直接用于脚本的任何信息。例如,与执行视频呈现程序相关的中断条件可与方框 308 的中断条件相似,并且与执行音频呈现器接口程序相关的中断条件可与方框 304 的中断条件相似。如果判断这种中断条件确定,那么可中断视频解码方框 214,并且可执行合适的视频呈现程序和 / 或音频呈现器接口程序。在执行该程序之后,可重新开始视频解码方框 214。在视频解码方框 214 的子元件之间执行这种功能,可有助于减少呈现假象。

[0148] 细分成子元件并且在这种子元件之间执行与视频和 / 或音频呈现相关的功能的这种方法 200、220、250、270、300 不限于上述方框 214 的实例。方法 200、220、250、270、300 (和 / 或方法 200、220、250、270、300 的任何合适的显示的或未显示的方框或步骤)可通过任何合适的方式细分成任何合适的子元件。

[0149] 如上所述,在特定的非限制性实例实施方式中,方法 200 (图 3)的方框 208 和方法 250 的方框 258 (图 5)可涉及调用与在服务器上托管的视频内容对应的 AJAX XMLHttpRequest 对象的 open() 方法。该方框 258 功能要求网络浏览器检索具有可由 Javascript 解释的格式的一些视频和 / 或音频数据。在一些实施方式中,方框 208 和 / 或 258 (及其相应的方法 200、250)可利用多个 AJAX XMLHttpRequest (后文中称为 AJAX) 对象,可反复使用这些对象,以提高方框 208 和 / 或 258 的效率。

[0150] 图 8 示出了可用于实现提取方框 208 的功能的视频数据的方法 350 的流程图。可通过相似的方式实现方框 258。在所显示的实施方式中,方法 350 涉及使用两个 AJAX 对象。然而,要理解的是,方法 350 可扩展为使用两个以上的 AJAX 对象。方法 350 在方框 352 中开始,该方法涉及查询在方法 350 的最后的迭代中是否使用第一个 AJAX 对象(AJAX_1)。如果方框 352 查询肯定(方框 352,是支路),那么方法 350 继续进入方框 354。在方框 354 中,第二个 AJAX 对象(AJAX_2)用于提取下一组数据。由 AJAX_2 检索的数据储存在由提供 AJAX_2 对象的网络浏览器(或其他应用程序)本地地管理的数据结构内。如果方框 352 查询否定(方框 352,否支路),那么方法 350 继续进入方框 356。在方框 356 中,AJAX_1 用于提取下一

组数据。由 AJAX_1 检索的数据储存在由提供 AJAX_1 对象的网络浏览器(或其他应用程序)本地地管理的数据结构内。在实现方框 354 或方框 356 之后,方法 350 结束。

[0151] 在一些实施方式中,方框 208 (包括方法 350) 的数据检索功能可由网络浏览器在脚本的指导下本地地执行。在一些实施方式中,可执行方框 208 (包括方法 350) 的数据检索功能,作为与方法 200 的剩余部分的线程平行的线程(在多线程环境下)或者作为与方法 200 的剩余部分的工序平行的工序(在多工序环境下)。例如,在一些实施方式中,网络浏览器可执行一个平行的线程 / 工序(例如,涉及脚本解释器),以实现方法 200,并且可执行第二平行的线程 / 工序(例如,涉及 AJAX 对象),以实现方法 350。在方法 350 中使用多个 AJAX 对象,也可利用能够实现多线程和 / 或多工序环境的用户计算装置。例如,可将线程或工序分配给多个 AJAX 对象中的每个。

[0152] 在网络浏览器的脚本解释器解释脚本时,其内存管理程序可相对无效。对于所谓的“垃圾回收”,尤其如此,垃圾回收用于腾出不再使用的先前分配的内存。垃圾回收可特别无效,可消耗处理资源并且可发生在不可取的时间,例如,在需要呈现帧图像或下一个音频段的时间。使用方法 350,多个 AJAX 对象可用作一种“循环”缓冲器,该缓冲器可避免需要中间缓冲使用由网络浏览器的脚本解释器分配的储存器的经编码的视频数据。在仅仅使用一个 AJAX 对象时,需要或可取地将所提取的数据串从 AJAX 对象复制成由网络浏览器的脚本解释器管理的中间缓冲器。这种中间缓冲需要或可取地腾出单个 AJAX 对象,以便可用于提取更多的数据。使用多个 AJAX 对象,可避免这种中间缓冲,这是因为一个 AJAX 对象可用于提取数据,而任何其他 AJAX 对象的数据可用于提供脚本实现方法 200 的剩余部分所需要的数据。除了使用直接从多个 AJAX 对象中提取的数据以外,负责实现 AJAX 对象的浏览器的功能可管理数据的创建 / 破坏,这可比浏览器的脚本解释器对数据的创建 / 破坏更有效。

[0153] 而且,在某些情况下,通过互联网或者通过某种其他网络获取的数据可具有每个连接的带宽限制。在这种情况下,使用多个 AJAX 对象,能够使数据更快速地流传输。

[0154] 在除了 Javascript 脚本中实现的实施方式中,相似的方法和相似的对象可用于提取视频和 / 或音频数据。在这种实施方式中,可通过与在本文中所描述的多个 AJAX 对象的使用相似的方式,使用多个数据提取对象。

[0155] 在一些实施方式中,可使用可有助于使用一个合适的脚本来有效地呈现视频内容的技术,来将视频内容编码。

[0156] 在从脚本解释器中传输或者由脚本解释器传递具有特定的数据格式(称为内容传输格式)的数据时,一些网络浏览器仅仅接受这种数据。例如,在由 Javascript 解释器将这种数据传输或者传递给浏览器时,一些浏览器仅仅读取所谓的 base64 编码的数据格式。如上所述,音频呈现系统 100 (图 2A)的内容传输编码器 145 在方法 200 的可选方框 214C (图 3)中可选地将帧图像转换成 base64 (或其他内容传输)编码的格式。内容传输编码是在用户计算装置上在计算上比较昂贵的工序。例如,在 base64 编码的特定情况下,这种内容传输编码涉及检查每组的 3 个字节并且将每组的 3 个字节转换成 4 个字符。这种计算成本可造成呈现假象,例如,在这种内容传输编码可能延迟呈现视频帧的周期比由期望的帧速率规定的帧间周期更长。

[0157] 根据一些实施方式,例如,在提供给用户计算装置(或者由用户计算装置检索)之前,视频数据的帧图像(和 / 或音频数据的区段(例如,帧))可在 base64 (或某个其他合适

的内容传输格式)中预先编码。下面的描述表示将视频数据的帧图像预先编码成 base64 或某个其他合适的内容传输格式,不失一般性地,这种预先编码还可适用于音频数据。在可由网络浏览器本地地呈现的格式中表示视频数据的基础帧图像时(例如,在 JPEG 或 GIF 格式中表示视频数据的基础帧图像时),将帧图像这样预先编码成 base64(或其他合适的内容传输)格式,尤其有用。在这种情况下,由于帧图像(一旦在方框 214A 中解码)已经具有适合于传输给网络浏览器的以及由网络浏览器呈现的格式,所以可跳过帧图像转换和内容传输编码(方法 200 的方框 214B、214C(图 3))的工序。在这种情况下,同样可省略视频呈现系统 100(图 2A)的帧图像转换器 142 和内容传输编码器 145。在一些实施方式中,可将视频内容的基础帧图像转码成可由网络浏览器本地地呈现的格式(例如, JPEG 或 GIF 格式),然后,可将转码的帧图像预先编码成一种合适的内容传输格式。虽然帧图像这样预先编码成 base64(或其他内容传输)格式,可增大编码的视频内容文件的尺寸,但是由于 base64(或其他内容传输)编码的数据的压缩性,所以文件尺寸的任何增大对编码的文件可流传输到用户计算装置中的速度可具有较小的影响。由 HTTP1.1(例如, gzip、deflate 等)支持的很多数据压缩技术可利用 base64(或其他内容传输)编码的数据使用有限的字符组(例如,在 base64 的情况下,具有 64 个字符和多个控制字符)这一事实。

[0158] 预先编码视频数据的帧图像(和/或音频数据的帧)的这种技术可在用户计算装置上减少消耗处理器资源。更具体而言,如果将帧图像预先编码成 base64 或其它合适的内容传输格式,那么可在用户计算装置上将帧图像直接传递给网络浏览器,而不由脚本在用户计算装置上在 base64 格式中转换或编码。

[0159] 在一些实施方式中,在位于视频内容下面的帧图像预先编码成 base64 格式或某个其他合适的内容传输格式时,填补与图像对应的数据串,以便与每个帧图像对应的数据串具有相同的长度。例如,在 base64 中,可通过一个合适的字符(例如,换行字符 <LF>)填补与帧图像对应的数据串。由于填补串相对可压缩,所以这种填补未明显增大传输速度。然而,帧图像的这种填补可使得网络浏览器的脚本解释器重新使用存储器,无需重新分配以及相应的“垃圾回收”。

[0160] 更具体而言,无需填补,每个帧图像具有一个不同的长度。浏览器的脚本解释器通常在用户计算装置上分配某个存储器,用于储存帧图像。通过非限制性实例,这种帧图像储存可用于实现任何帧图像队列(例如,队列 126、143、141、138(图 2A))。在重新使用一些分配的存储器来处理新帧时,如果新帧的尺寸与要覆盖的前一个帧的尺寸不同,那么脚本解释器可决定需要破坏前一个分配并且重新分配存储器,以容纳新帧。先前分配的存储器的破坏与重新分配通常称为“垃圾回收”。先前分配的存储器的这种破坏与重新分配可消耗处理器资源并且造成假象。相反,如果所有帧具有相同的尺寸,那么脚本可由位于先前分配的存储器内部的新帧覆盖前一个帧,而不破坏先前的内存分配或者造成内存重新分配。这可减少与内存重新分配相关的处理器资源并且减少呈现假象的发生。

[0161] 目前,所有关注的网络浏览器本地地支持至少两个图像编码文件格式: JPEG 和 GIF。在特定的实施方式中,可使用具有多个不同的编码文件格式的视频内容的帧图像,来将视频内容编码。可将视频内容编码,其中,单独的帧图像可编码成两个或多个不同的图像编码文件格式(例如, JPEG 和 GIF)。更具体而言,在编码期间,可根据帧图像的视觉特征或内容,选择每个帧图像的编码文件格式。例如, GIF 可取地用于编码帧图像,其中,视觉特征

可包括清晰的边缘和细节,例如,显示文本信息的图像(记录屏幕、标题屏幕等)、动画图像等。JPEG 可更适用于包含更自然的视觉内容的帧图像(例如,没有这种清晰的边缘)。

[0162] 在一些实施方式中,可处理在特定的视频内容部件内容内的单独帧图像,以确定在 JPEG 文件格式还是 GIF 文件格式中可能更好地编码。例如,可比较与每个帧的单独像素相关的值(例如,光度和 / 或色度值)和与其相邻像素相关的值,以确定一个或多个梯度度量。在一个实施方式中,虽然在其他实施方式中可使用其他组的相邻像素,但是用于一个特定的非边缘像素的这组相邻像素可包括其 8 个相邻的像素。在一个实例实施方式中,梯度度量可基于像素值的差值。然后,可分析在一个特定的帧图像中的所有像素的梯度度量,以确定应在 JPEG 还是 GIF 文件格式中编码该特定的帧图像。例如,如果具有较大数量的较高梯度度量,那么这可表示应在 GIF 格式中编码的具有清晰的边缘的帧图像,并且另一方面,如果具有较大数量的较低梯度度量,那么这可表示应在 JPEG 格式中编码的帧图像。在一个特定的实施方式中,通过将梯度度量装箱(以形成概念直方图等),然后比较在每个箱子内的梯度度量的数量与一个或多个合适的阈值,梯度度量可实现这种分析。

[0163] 在某些情况下,可确定关注的用户计算装置的网络浏览器能够本地地呈现额外的图像编码的文件格式。在这种情况下,这种额外的图像编码的文件格式可并入编码工序内。即,可编码视频内容,其中,在 3 个或多个不同的图像编码的文件格式中,编码在特定的视频内容部件内的单独帧图像。

[0164] 使用具有不同的编码文件格式的单独帧图像来编码视频内容,可允许提高压缩(相对于在单个文件模式中编码帧图像),这可反过来引起在用户计算装置上更有效地获得和 / 或解码的视频内容以及相应更少的呈现假象。

[0165] 在特定的实施方式中,根据每个帧图像的视觉内容,可逐个帧地操纵用于将帧图像编码成 JPEG 格式的量化表(例如,光度和色度量化表)。在一些实施方式中,可比较每个帧图像的像素值与其在时间上相邻的帧图像中的一者或两者的像素值(例如,逐个像素),以确定一个或多个不同的度量,并且可根据这种不同的度量,确定用于将帧图像编码的量化表。例如,如果确定一个特定的图像帧的色度与其相邻的帧的色度相差不大(即,色差度量低),那么这可表示较慢移动的场景,允许使用与用于该特定的帧图像的较高程度的压缩对应的色度量化表。相反,如果用于特定像素的色差度量较高,那么这可表示较快移动的场景,建议使用与用于该特定的帧图像的较低程度的压缩对应的色度量化表。光度量化表可能具有相似的调节。

[0166] 在编码侧操纵用于编码帧图像的 JPEG 量化表,可有助于实现高压缩,而不过度牺牲人们可察觉的细节。帧图像的更大压缩可造成使用更少的计算资源来在用户计算装置上解码和呈现帧图像,这反过来可产生更少的呈现假象。

[0167] 一些用户计算平台不支持提取部分视频内容文件,仅仅能够下载(或获取)完整的视频内容文件。通过非限制性实例,运行 Android™ 操作系统的一些移动电话装置目前具有该限制。在一些实施方式中,可将单个视频内容文件解析成一系列数据块,并且可编码每个数据块,犹如完整的视频内容文件。这允许用户计算装置单独下载(或获得)每个数据块,并且然后,在用户计算装置上解释的脚本可被配置为在用户计算装置上将数据块按照顺序组织在一起,以呈现完整的视频内容文件。在用户计算装置上可使用列表,以按照顺序组织更小的视频内容文件。

[0168] 通常,虽然没有限制性,但是以 24 帧 / 秒或 33 帧 / 秒的帧速率,将视频数据编码。在某些情况下,预期在用户计算装置上可利用的处理资源不充足,以能够使在本文中所述的脚本以这些或其他期望的帧速率支持呈现视频数据。因此,在一些实施方式中,以更低的帧速率,可下采样并且重新编码视频数据,并且在用户计算装置上解码下采样的视频数据时,通常可显示可利用的帧图像,并且在可利用的帧图像之间的时间,可使用 α 混合技术,提取插补的帧图像。这种内插技术不限于在编码之前下采样视频数据的情况。在一些实施方式中,在用户计算装置上可确定以期望的帧速率呈现视频数据的处理资源不充足,在这种情况下,通过选择解码和显示可利用的图像帧的子集,可有效地下采样视频数据,并且在所选择的帧图像之间的时间,可使用 α 混合技术,提取插补的帧图像。在图 9A 和 9B 中示意性显示这种工序的实例。

[0169] 图 9A 和 9B 示出了原始视频数据 400 具有帧速率 $1/t_0$ 的一个实例,其中, t_0 表示帧间距,并且 $f_0, f_1, f_2, f_3 \dots$ 表示原始帧图像(例如,在帧图像内的单独像素的像素值)。在图 9A 和 9B 的每个中,以比原始视频数据 400 更低的帧速率,下采样原始视频数据 400。如上所述,在使视频数据可用于用户计算装置之前,可进行这种下采样;或者,通过从视频数据内选择帧图像的一个特定子集,以进行解码和处理,可在用户计算装置上发生这种下采样。在图 9A 实例中,通过采用每隔一个帧($f_0, f_2, f_4, f_6 \dots$),下采样原始视频数据 400,以便重新编码的视频数据 402 具有帧速率 $1/t_A$ 该帧速率是原始视频数据 400 的帧速率 $1/t_0$ 的一半。在图 9B 实例中,通过采用每隔两个帧($f_0, f_3, f_6, f_9 \dots$),下采样原始视频数据 400,以便重新编码的视频数据 404 具有帧速率 $1/t_B$,该帧速率是原始视频数据 400 的帧速率 $1/t_0$ 的 $1/3$ 。本领域的技术人员要理解的是,图 9A 和 9B 的实例仅仅进行解释,并且通常通过采用原始帧图像 $f_0, f_1, f_2, f_3 \dots$ 的任何合适的部分,可下采样原始视频数据 400。

[0170] 在图 9A 和 9B 的实施方式中的下采样的视频数据 402、404 的一个优点在于,用户计算装置仅仅需要解码和制备图像数据的相应地更少的帧。例如,在图 9A 的情况下,下采样的视频数据 402、视频解码器 122 以及帧图像处理器 136 (见图 2A)可仅仅需要处理在方框 214A、214B、214C、214D (见图 3)中的帧图像的一半。帧图像解码和 / 或制备的这种减少的量可减少用户计算装置处理资源的负担。然而,在一些实施方式中,可取地以原始帧速率 $1/t_0$ 呈现帧图像(例如,由在方框 236 (图 4)中的帧图像呈现器 132 (图 2A)呈现),因为这会提供流传输移动的视频的外观。在这种实施方式中,在下采样的视频数据 402、404 的图像帧之间的时间,可取地呈现内插的帧图像。例如,在图 9A 的下采样的视频数据 402 的情况下,即使下采样的视频数据 402 不包括与时间 $t_1, t_3, t_5 \dots$ 对应的经解码的帧图像,在这些时间,也可取地呈现内插的帧图像。同样,在图 9B 的下采样的视频数据 404 的情况下,即使下采样的视频数据 404 不包括与时间 $t_1, t_2, t_4, t_5 \dots$ 对应的经解码的帧图像,在这些时间,也可取地呈现内插的帧图像。

[0171] 在一些实施方式中,通过使用 α 混合来创建图像对象,可提供这种插补。例如,在方框 148 (图 2A)和方框 214D (图 3)中,可创建这种 α 混合的图像对象。在一个特定的实施方式的情况下,例如,所创建的图像对象可为画布标记的争论。图 9C 示出了根据一个特定的实施方式的用于创建这种 α 混合的图像对象的方法 420。方法 420 在方框 422 中开始,该方框涉及查询确切的帧是否可用于当前的帧索引(例如,在方框 224 中确定的帧索引(图 4))。如果可利用确切的帧(方框 422,是支路),那么方法 420 继续进入方框 424,该方框

涉及根据确切的帧创建图像对象。在下采样的数据 402 (图 9A) 的实例实施方式的情况下, 在时间 t_0 、 t_2 、 t_4 、 t_6 ... 方法 420 能够使用帧 f_0 、 f_2 、 f_4 、 f_6 ... 的确切的帧来创建图像对象, 并且在下采样的数据 404 (图 9B) 的实例实施方式的情况下, 在时间 t_0 、 t_3 、 t_6 ... 方法 420 能够使用帧 f_0 、 f_3 、 f_6 ... 的确切的帧来创建图像对象。这在图 9A 中示意性显示为要提取的数据 406, 并且在图 9B 中示意性显示为要提取的数据 408。在方框 424 中创建图像对象之后, 方法 420 结束。

[0172] 返回方框 422, 如果方框 422 查询确定没有可用于当前的帧索引的确切的帧 (方框 422, 否支路), 那么方法 420 继续进入方框 426, 该方框涉及确定 α 混合系数。通常, 可由

$$[0173] \quad \alpha = [\beta + 1 - \text{mod}(f\#, \beta)]^{-1}$$

[0174] 提供方框 426 的 α 混合系数 α ,

[0175] 其中:

[0176] α 是期望的 α 混合系数;

[0177] β 是有效的下采样因数或等同于在可利用的图像帧 +1 之间的呈现次数 (例如, 对于图 9A, $\beta = 2$, 并且对于图 9B, $\beta = 3$);

[0178] $f\#$ 是当前的帧索引; 以及

[0179] $\text{mod}(x, y)$ 是返回由 x 除以 y 的整数余数的模数运算符。

[0180] 将该公式用于图 9A 的实例, 对于 f_1 、 f_3 、 f_5 ... , $\alpha = 0.5$, 并且用于图 9B 的实例, f_1 、 f_4 、 f_7 ... , $\alpha = 0.33$, 并且对于 f_2 、 f_5 、 f_8 ... , $\alpha = 0.5$ 。通常, 以上等式适合于确定任何下采样因数 β 的 α 混合系数。在方框 426 之后, 方法 420 继续进入方框 428, 该方框涉及通过在前一个图像帧的图像对象上 α 混合下一个可用的图像帧的一部分, 来创建图像对象。下一个可用的图像帧的这部分由方框 426 的 α 混合系数确定。因此, 例如, 对于在图 9A 的实例中的帧 f_1 , 其中, $\alpha = 0.5$, 下一个可用的图像帧是 f_2 , 并且前一个帧的图像对象是 f_0 , 方框 428 涉及通过在前一个图像对象 (f_0) 上 α 混合 $0.5f_2$, 来创建图像对象, 其创立 $f_1 = 0.5f_2 + 0.5f_0$ 。同样, 对于在图 9A 的实例中的帧 f_3 , 其中, $\alpha = 0.5$, 下一个可用的图像帧是 f_4 , 并且前一个帧的图像对象是 f_2 , 方框 428 涉及通过在前一个图像对象 (f_2) 上 α 混合 $0.5f_4$, 来创建图像对象, 其创立 $f_3 = 0.5f_4 + 0.5f_2$ 。这在图 9A 中示意性显示为要提取的数据 406。

[0181] 对于在图 9B 的实例中的帧 f_1 , 其中, $\alpha = 0.33$, 下一个可用的图像帧是 f_3 , 并且前一个帧的图像对象是 f_0 , 方框 428 涉及通过在前一个图像对象 (f_0) 上 α 混合 $0.33f_3$, 来创建图像对象, 其创立 $f_1 = 0.33f_3 + 0.67f_0$ 。对于在图 9B 的实例中的帧 f_2 , 其中, $\alpha = 0.5$, 下一个可用的图像帧是 f_3 , 并且前一个帧的图像对象是 $f_1 = 0.33f_3 + 0.67f_0$, 方框 428 涉及通过在前一个图像对象 ($f_1 = 0.33f_3 + 0.67f_0$) 上 α 混合 $0.5f_3$, 来创建图像对象, 其创立 $f_2 = 0.5f_3 + 0.5(0.33f_3 + 0.67f_0) = 0.67f_3 + 0.33f_0$ 。对于在图 9B 的实例中的帧 f_4 , 其中, $\alpha = 0.33$, 下一个可用的图像帧是 f_6 , 并且前一个帧的图像对象是 f_3 , 方框 428 涉及通过在前一个图像对象 (f_3) 上 α 混合 $0.33f_6$, 来创建图像对象, 其创立 $f_4 = 0.33f_6 + 0.67f_3$ 。对于在图 9B 的实例中的帧 f_5 , 其中, $\alpha = 0.5$, 下一个可用的图像帧是 f_6 , 并且前一个帧的图像对象是 $f_4 = 0.33f_6 + 0.67f_3$, 方框 428 涉及通过在前一个图像对象 ($f_4 = 0.33f_6 + 0.67f_3$) 上 α 混合 $0.5f_6$, 来创建图像对象, 其创立 $f_5 = 0.5f_6 + 0.5(0.33f_6 + 0.67f_3) = 0.67f_6 + 0.33f_3$ 。这在图 9B 中示意性显示为要提取的数据 408。

[0182] 在方框 428 中创建每个图像对象之后, 方法 420 结束。在方法 200 的每个迭代 (图

3)中,可实现方法 420 (图 9C)的工序,例如,在方框 214 中,以插补帧图像,其中,由于在编码器侧进行下采样或者在用户计算装置上有效地进行下采样,所以不能使用帧图像。这可允许用户计算装置以可取的帧速率呈现帧图像,同时在用户计算装置上减少在处理器资源上的负荷,并且可能产生更少的视频假象。

[0183] 在特定的实施方式中,在解释脚本时,脚本使得用户计算装置使用 HTTP 协议来从服务器等中获取视频内容。与可利用较复杂的协议来使视频内容适合于特定的用户计算装置、特定的媒体播放器应用和 / 或特定的流媒体网络连接速度(比特率)的流媒体不同,使用 HTTP 协议来传输数据的 HTTP 服务器通常具有比较有限的功能,用于与用户计算装置相互作用。在一些实施方式中,在解释脚本时,脚本确定用户计算装置、网络浏览器和 / 或在服务器与用户计算装置之间的网络连接的属性,并且采取适当的行动,来适合呈现视频内容。

[0184] 图 10 示出了用于呈现视频内容的方法 450,其中,根据用户计算装置的一个或多个属性和 / 或在服务器与用户计算装置之间的网络连接,在解释脚本时,该脚本适合呈现视频内容。方法 450 或其部分可直接由脚本实现。方法 450 在方框 452 中开始,该方框涉及查询初始装置属性。这种装置属性可包括在装置上操作的装置硬件和 / 或软件(包括网络浏览器和 / 或操作系统软件)的属性。装置属性的非限制性实例包括:处理器速度和 / 或处理能力;显示器属性(例如,显示器分辨率、显示器方向、纵横比、屏幕尺寸等);具有专门硬件(例如,专门的音频呈现硬件和 / 或视频呈现硬件);网络浏览器或解释脚本的其他应用程序的软件属性(例如,网络浏览器本地地呈现特定的音频格式和 / 或本地地呈现特定的图像文件格式的能力);装置的操作系统的软件属性;文件处理能力(例如,该装置、该装置的操作系统和 / 或该装置的网络浏览器是否可作为单个文件容纳整个视频内容,或者视频内容是否必须解析成多个更小的文件);等。

[0185] 一旦已经确定这些装置属性,就在解释脚本时,脚本使用在方框 452 中获得的装置属性来在方框 454 中初始化特定装置的呈现属性。在方框 454 中初始化特定的呈现属性,可涉及从服务器中选择下载合适的文件。例如,在确定特定的用户计算装置需要将视频内容解析成更小的文件时,在用户计算装置或其浏览器的本机视频呈现能力建议使用特定的视频内容文件时,在用户计算装置或其浏览器的本机图像呈现能力建议使用特定的视频内容文件时,在用户装置的显示器的分辨率和 / 或屏幕尺寸建议使用特定的视频内容文件时,在用户计算装置处理器的知识指示应使用下采样的视频内容文件时,诸如此类,可能如此。在方框 454 中初始化其他呈现属性,可涉及记录特定的信息,用于随后进行呈现。例如,呈现属性(例如,屏幕尺寸、显示器方向(纵横比)等)可能如此。

[0186] 然后,方法 450 继续进入方框 456,该方框涉及根据上述任何方法开始呈现工序。在方框 456 中,在解释脚本时,脚本可尝试根据方框 454 的初始化呈现属性,获得合适的视频内容文件。另外 / 或者,在方框 456 中,脚本使得根据方框 454 的初始化呈现属性呈现任何获得的视频内容文件的视频内容。通过非限制性实例,在用户计算的显示器具有某个分辨率或屏幕尺寸的情况下,脚本缩放帧图像,以最佳地适合于用户计算装置的显示器;在用户计算装置上接收之前来下采样视频数据的情况下,脚本插在已知的图像帧之间;在以特定的帧速率接收视频数据,但是用户计算装置的处理器的速度不能足够快地解码和制备图像帧的情况下,通过选择解码和制备图像帧的子集,脚本在用户计算装置上有效地下采样视频数据,并且插在所选择的图像帧之间;诸如此类。

[0187] 在解释脚本时,脚本时常在方框 458 中查询进行中的装置特征。该方框 458 查询可与方框 452 的初始查询相似,除了方框 458 会查询方框 452 的初始查询是否具有某种变化之外。通过非限制性实例,潜在的变化可涉及装置显示器的方向以及要显示的图像帧的相应纵横比的变化。在某些情况下,方框 548 可涉及在方框 234 (图 4) 中从丢弃视频动作和 / 或在方框 286 (图 5) 中从丢弃音频动作中获取信息。例如,如果方框 458 查询确定已经丢弃了阈值数量以上的图像帧,那么可得出以下结论:用户计算装置的容量不足以处理全帧速率视频数据,并且合适的视频内容文件可能应变成下采样的文件,或者应通过选择全速率视频帧的一个合适的子集,来在用户计算装置上有效地下采样视频数据。

[0188] 而且,在解释脚本时,脚本时常在方框 460 中查询在服务器与用户计算装置之间的进行中的网络特征。如果在服务器与用户计算装置之间的可利用的带宽或比特率太低,不能确保视频内容及时到达用户计算装置,以进行解码和呈现,那么可取地切换成更低质量的视频文件,例如,具有下采样的视频数据的文件、具有更低的分辨率的文件等。

[0189] 方框 462 涉及根据从方框 458 和 460 中获得的信息来更新呈现特征。例如,在第一迭代上,方框 462 可涉及根据从方框 458 和 460 中获得的信息来更新方框 454 的初始化呈现特征。在随后的迭代中,方框 462 可涉及更新先前的方框 462 呈现特征。然后,方法 450 继续进入方框 464,该方框涉及继续呈现视频内容。方框 464 可与上述方框 456 相似,除了方框 464 利用在方框 462 中更新的呈现特征以外。如上所述,更新的方框 462 呈现特征(根据方框 458 和 460 的查询)通常可引起为进行呈现而获得的文件(例如,质量更高或更低的视频文件)或者视频内容的呈现(例如,进行缩放,以容纳显示器方向)发生变化。在方框 464 中可实现这些变化。

[0190] 然后,方法 450 继续进入方框 466,该方框涉及查询视频内容是否完成呈现。如果是的话(方框 466,是支路),那么方法 450 结束。如果具有更多要呈现的视频内容(方框 466,否支路),那么方法 450 环回方框 458,以时常重复查询装置特征和更新呈现特征的工序。

[0191] 本发明的实施方式包括在本文中描述的各种操作。除非另有规定,这些操作可由硬件元件、软件、固件或其组合执行。

[0192] 某些实施方式可作为计算机程序产品来实现,该计算机程序产品可包括储存在机器可读媒体上的指令。这些指令可用于将通用或专用处理器编程,以执行所描述的操作。机器可读媒体可包括任何机构,用于储存具有可由机器(例如,计算机)读取的形式(例如,软件、处理应用)的信息。机器可读媒体可包括但不限于磁储存介质(例如,软盘);光储存介质(例如,CD-ROM);磁光储存介质;只读存储器(ROM);随机存取存储器(RAM);可擦可编程存储器(例如,EPR0M 和 EEPROM);闪速存储器;或者适合于储存电子指令的另一种介质。

[0193] 此外,可在分布的计算环境中,实践一些实施方式,其中,机器可读媒体储存在不止一个计算机系统上和 / 或由计算机系统执行。此外,在连接计算机系统的通信介质上,可拉动或推动在计算机系统之间传输的信息。表示为一系列工序步骤的方法可具有在相应的一个或多个计算机处理器上执行的步骤的一个或多个子集,然后,这些处理器使用例如进程间通信在计算机处理器之间传送中间结果或结果的可用性。可在位于 CPU 附近的高速缓冲存储器内,或者通过内存管理硬件或者甚至通过使得数据储存在要载入计算机主存储器内的外部储存装置上的这种内存管理硬件,来解决计算机主存储器,从而储存数据值。

[0194] 在本发明的各种实施方式的实现方式中使用的计算机处理元件包括一个或多个

通用处理装置,例如,微处理器或中央处理单元、控制器、图形处理单元(GPU)、移动电脑等。或者,这种数字处理元件可包括一个或多个专用处理装置,例如,数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)等。在特定的实施方式中,例如,数字处理装置可为具有包括一个核心单元的多个处理器和多个微引擎的网络处理器。此外,数字处理装置可包括通用处理装置和专用处理装置的任何组合。

[0195] 虽然按照特定的顺序显示和描述了在本文中的方法的操作,但是每个方法的操作顺序可改变,以便可逆序执行某些操作,或者以便可至少部分与其他操作同时执行某些操作。在另一个实施方式中,不同操作的指令或子操作可具有间断和 / 或交替的方式。

[0196] 在上面引用元件(例如,软件模块、处理器、组件、装置、电路等)时,除非另有说明,否则引用该元件(包括引用“装置”),应理解为包括执行所描述的元件的功能的任何元件(与该引用元件等效)(即,在功能上等效),包括在结构上与在本发明的所显示的示例性实施方式中执行功能的所公开的结构不等效的元件。

[0197] 虽然上面已经描述了多个示例性方面和实施方式,但是本领域的技术人员会认识到其某些修改、置换、添加以及子组合。例如:

[0198] • 可反复例示方法 300,以便具有多种方法,连续地调度其自身的连续迭代(例如,在相同的重复间周期或者在不同的重复间周期)。

[0199] • 已知一些用户计算装置具有多个和 / 或专用的图像 / 音频呈现引擎。这种图像 / 音频呈现引擎可包括合适的硬件和 / 或软件。通过非限制性实例,苹果的 iPhone™ 和 iPad™ 装置包括专用 3D 图像 / 音频呈现引擎。一些实施方式可涉及触发用户计算装置的网络浏览器激活这种多个和 / 或专用的呈现引擎。例如,即使视频内容不包括 3D 视频数据,一些实施方式也可涉及触发 iPhone™ 或 iPad™ 装置的 Safari™ 网络浏览器使用该装置的 3D 呈现引擎,以帮助呈现视频内容。在一个特定的非限制性实施方式中,在网页中的以下 CSS 代码触发 Safari™ 使用 3D 呈现引擎:

[0200]

```
<style type="text/css">
body { margin: 0px; overflow: hidden; width : 100%;
// need the following 2 lines to trick safari to use the 3D accelerator
```

[0201]

```
-webkit-transform: translateZ(-1000px);
-webkit-transform-style: preserve-3d;
}
.container { position: absolute; display: block;
// need the following 2 lines to trick safari to use the 3D accelerator
-webkit-transform-style: preserve-3d;
-webkit-transform : translateZ(0px);
overflow: hidden;
pointer-events: none;
}
</style>
```

[0202] • 在一些实施方式中，网络浏览器是为特定的用户计算装置平台编译的应用程序。在其他实施方式中，浏览器功能是操作系统的一部分，或者在某些情况下，是嵌入装置硬件内的功能。

[0203] 因此，其目的在于，以下所附权利要求以及此后介绍的权利要求被解释为包括在其真实的精神和范围内的所有这种修改、置换、添加以及子组合。

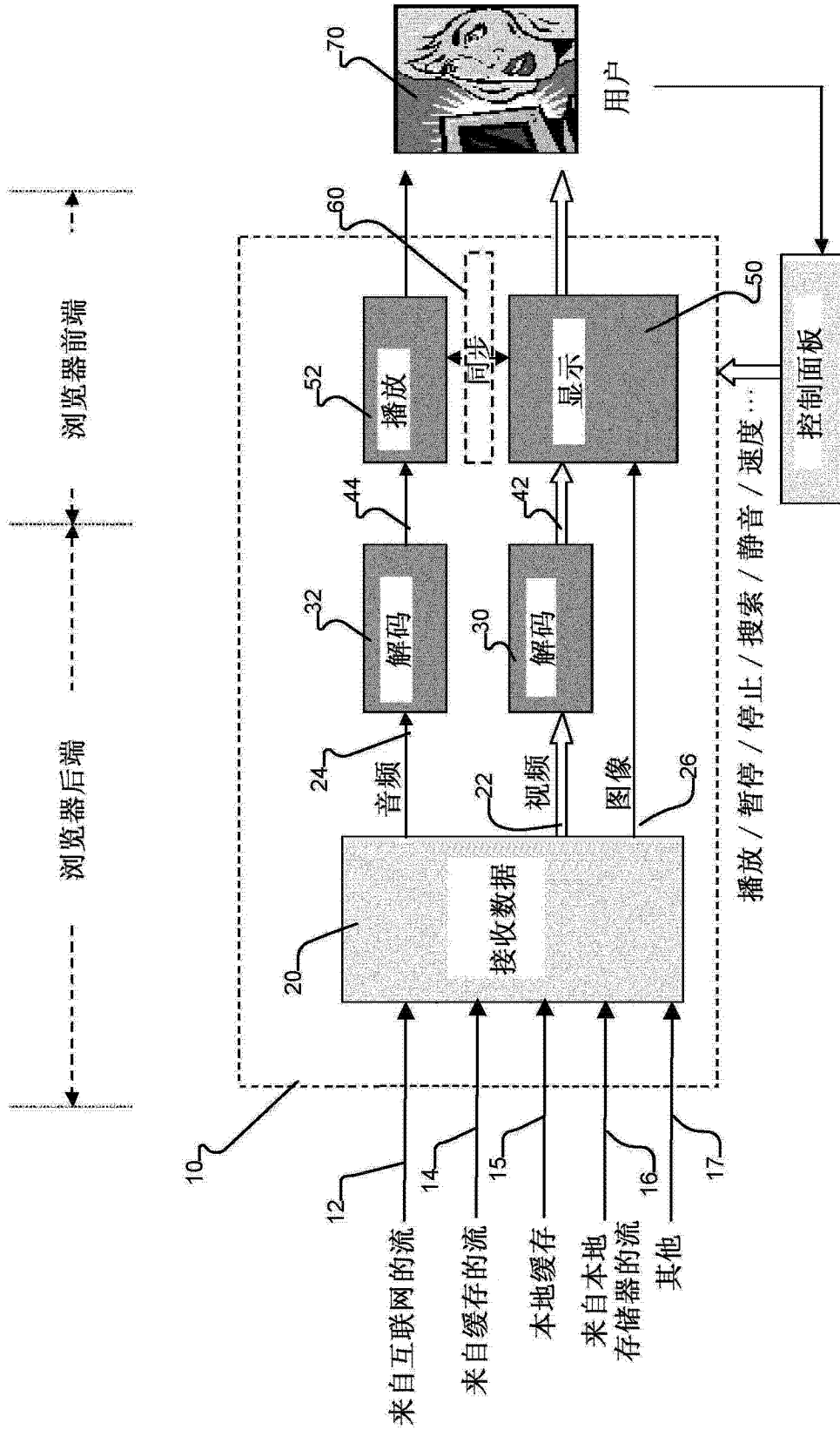


图 1

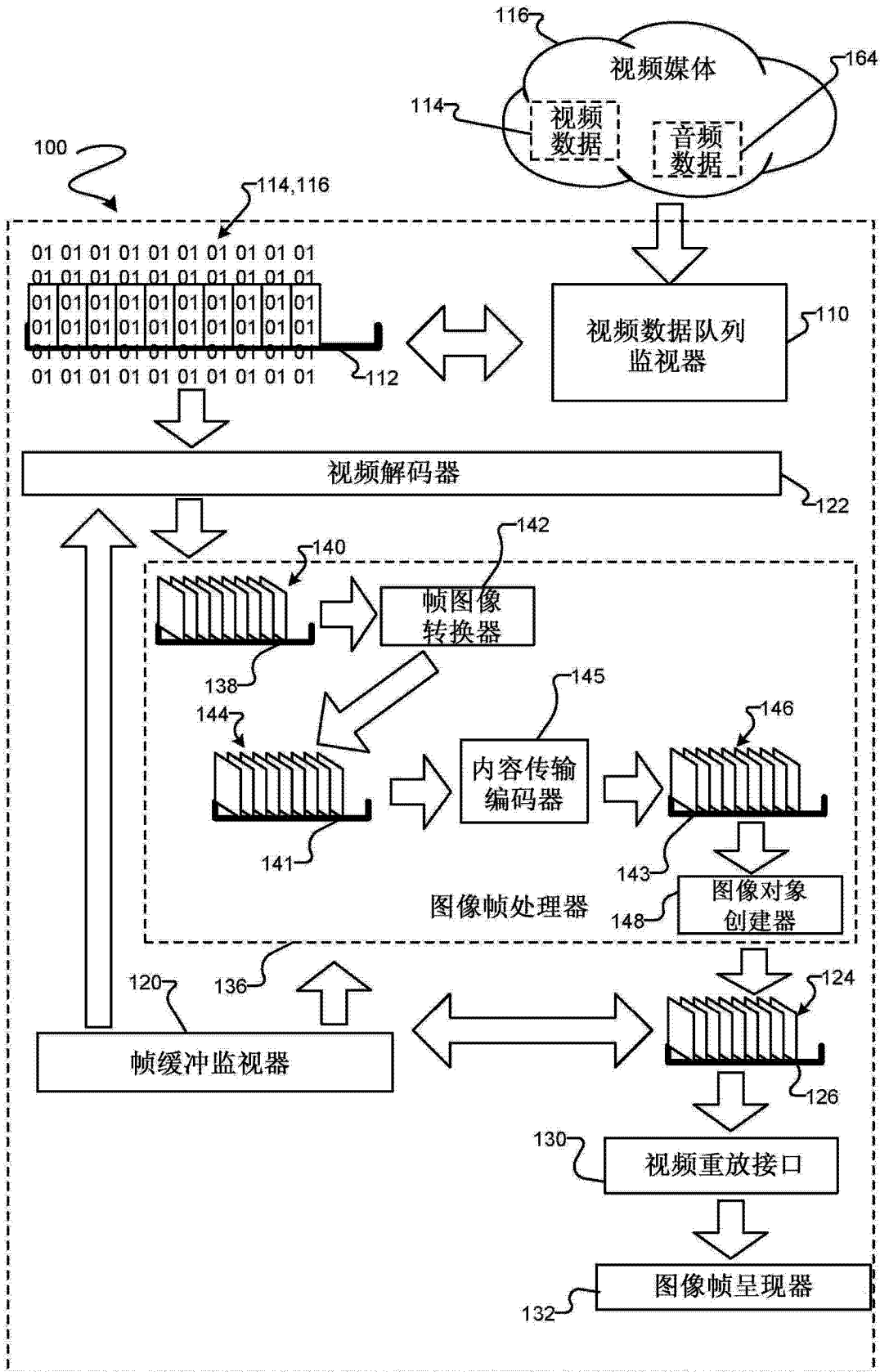


图 2A

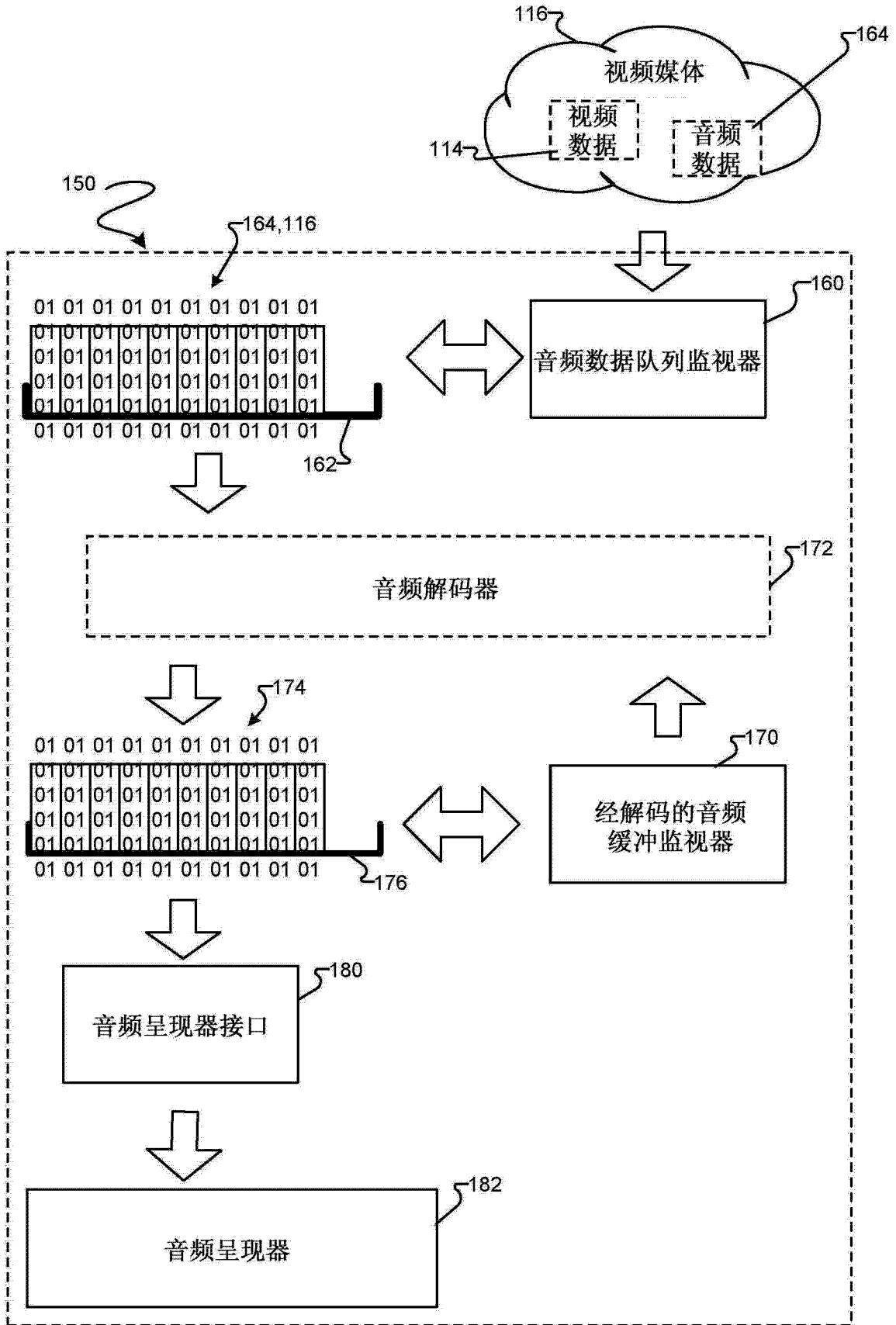


图 2B

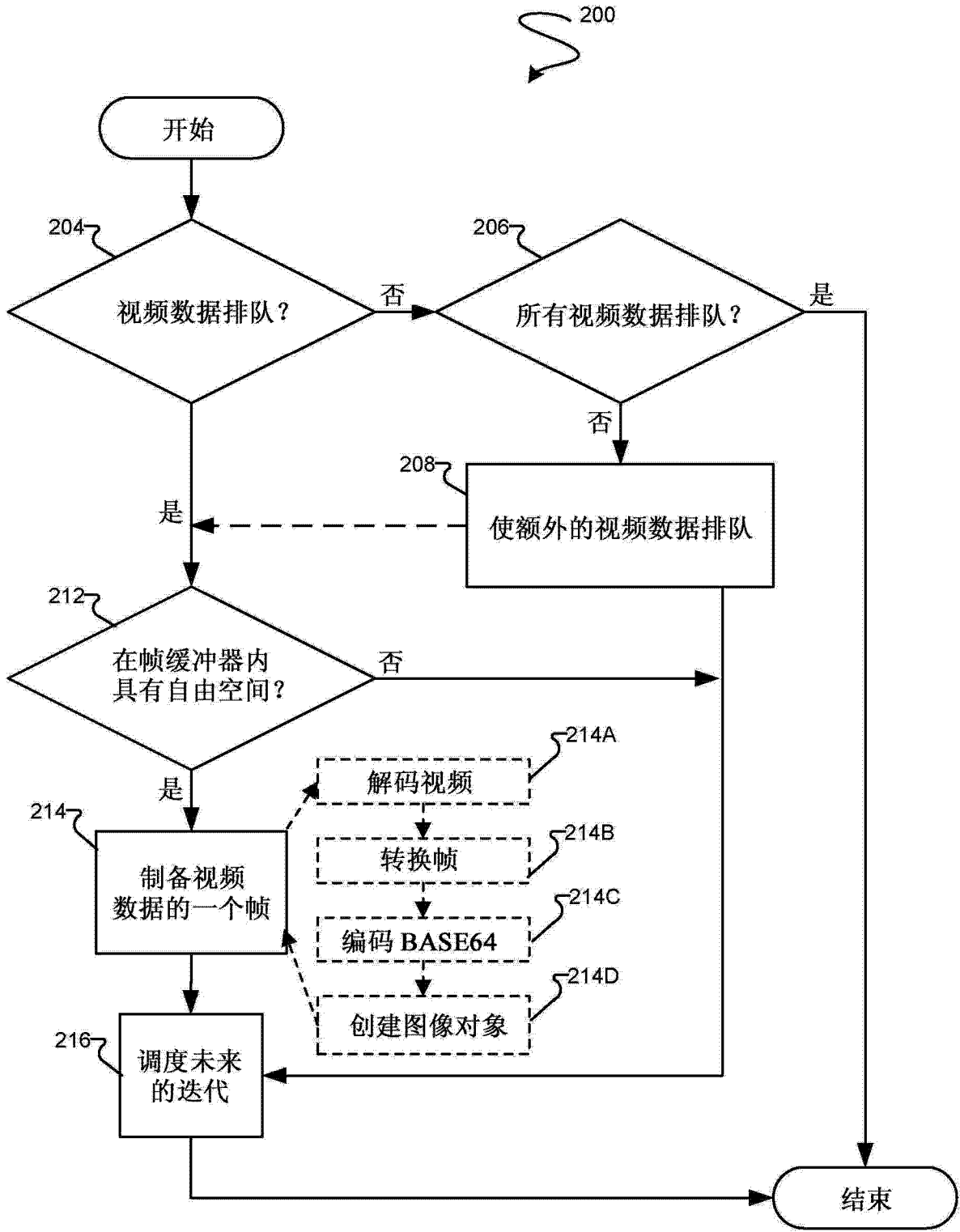


图 3

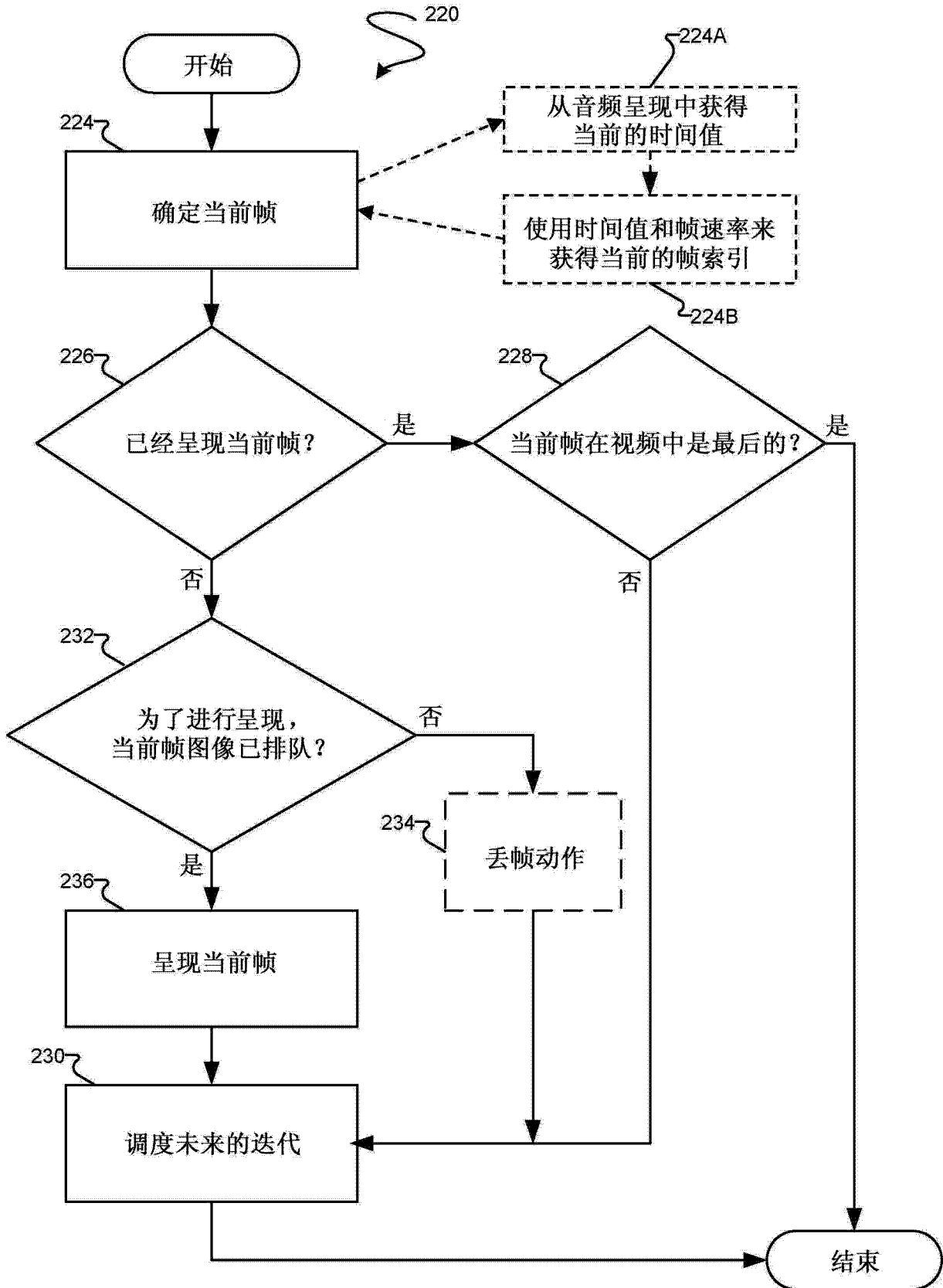


图 4

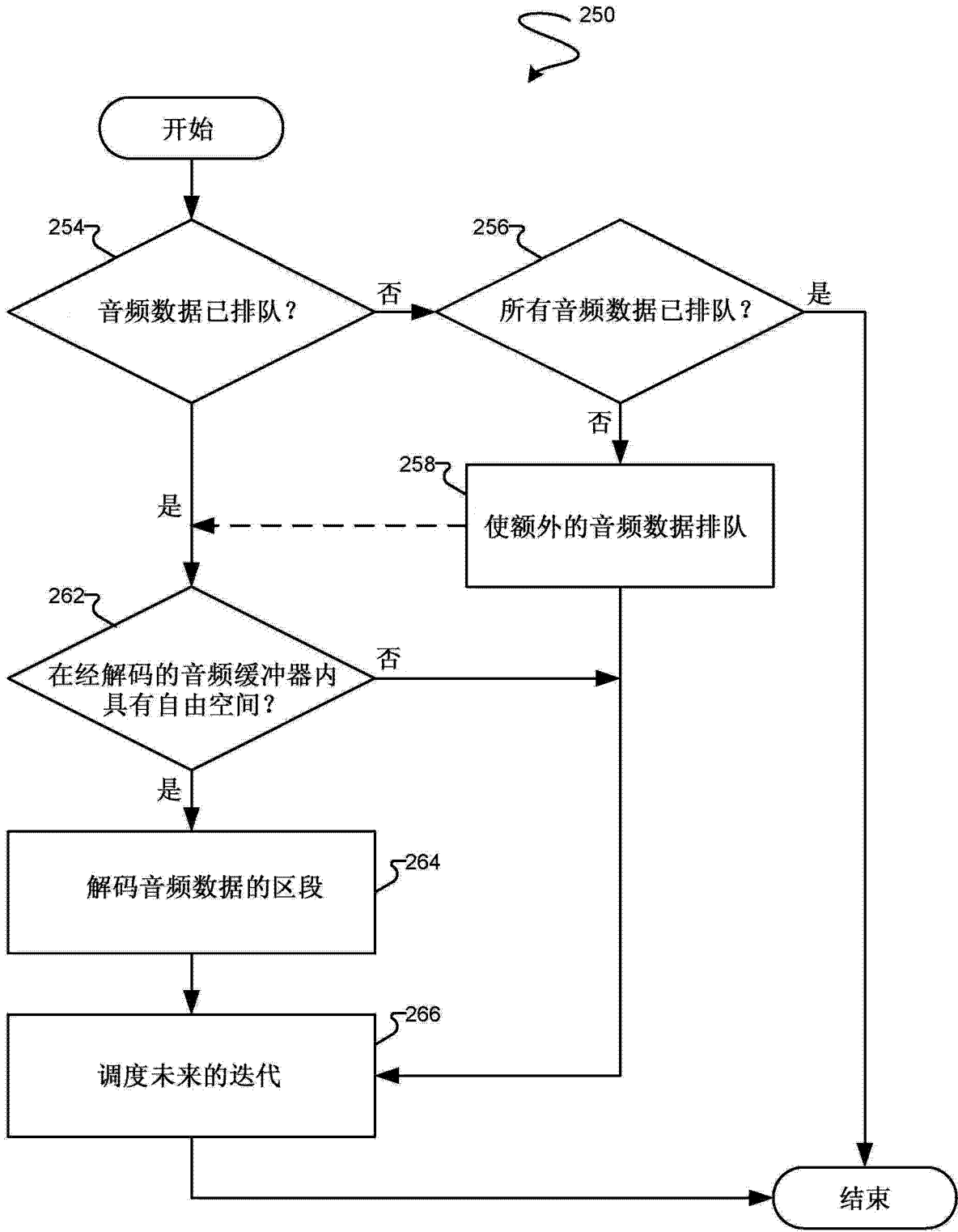


图 5

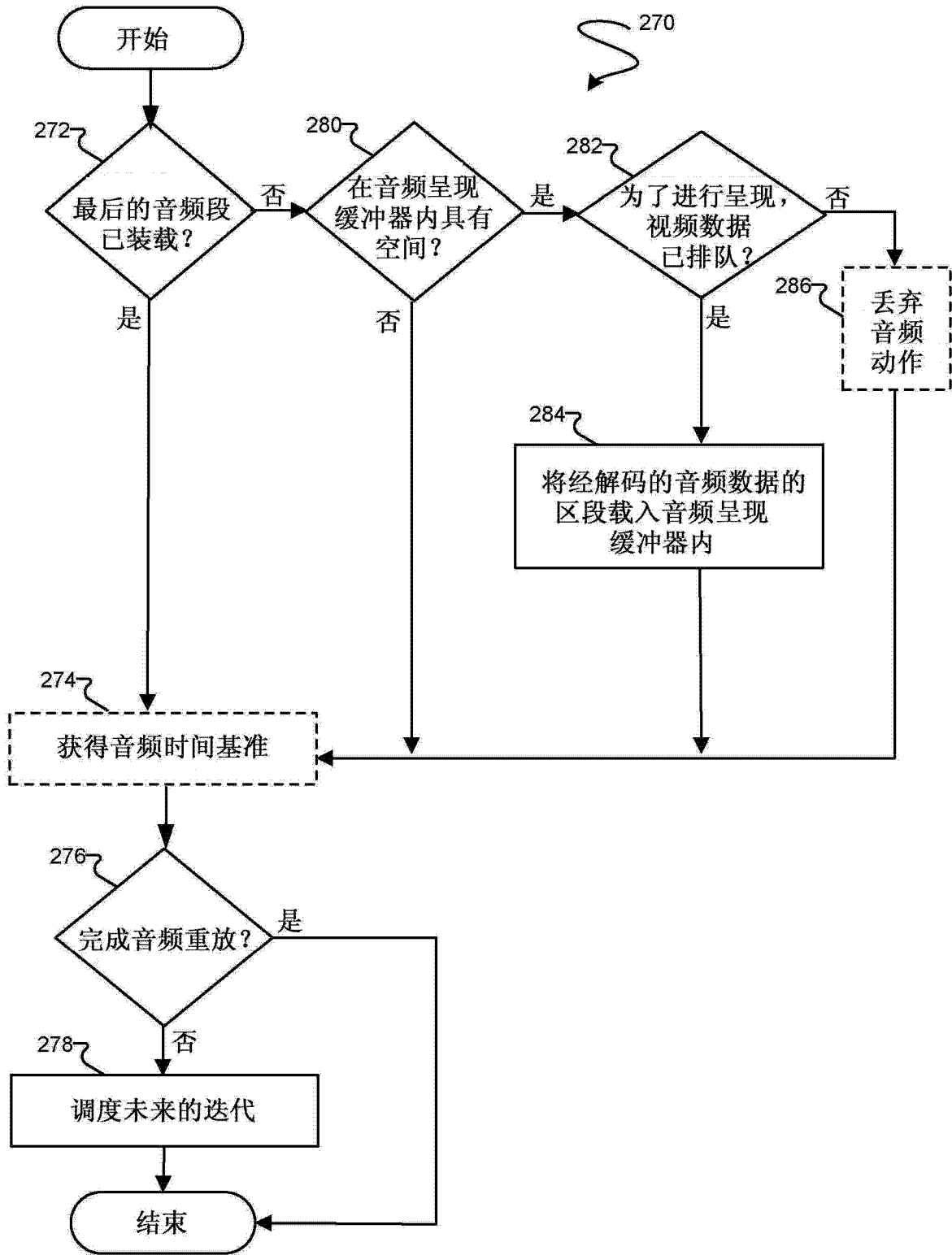


图 6

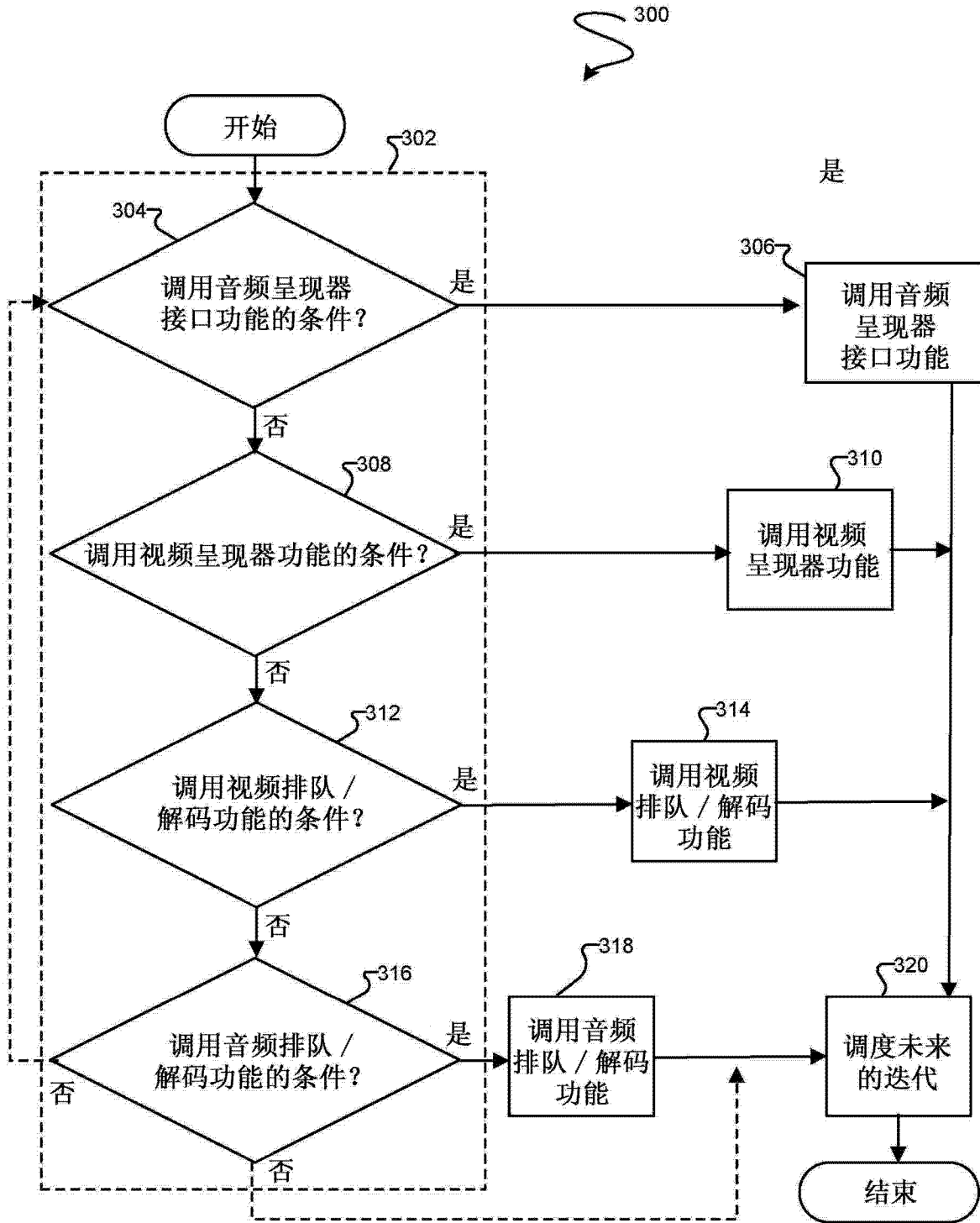


图 7

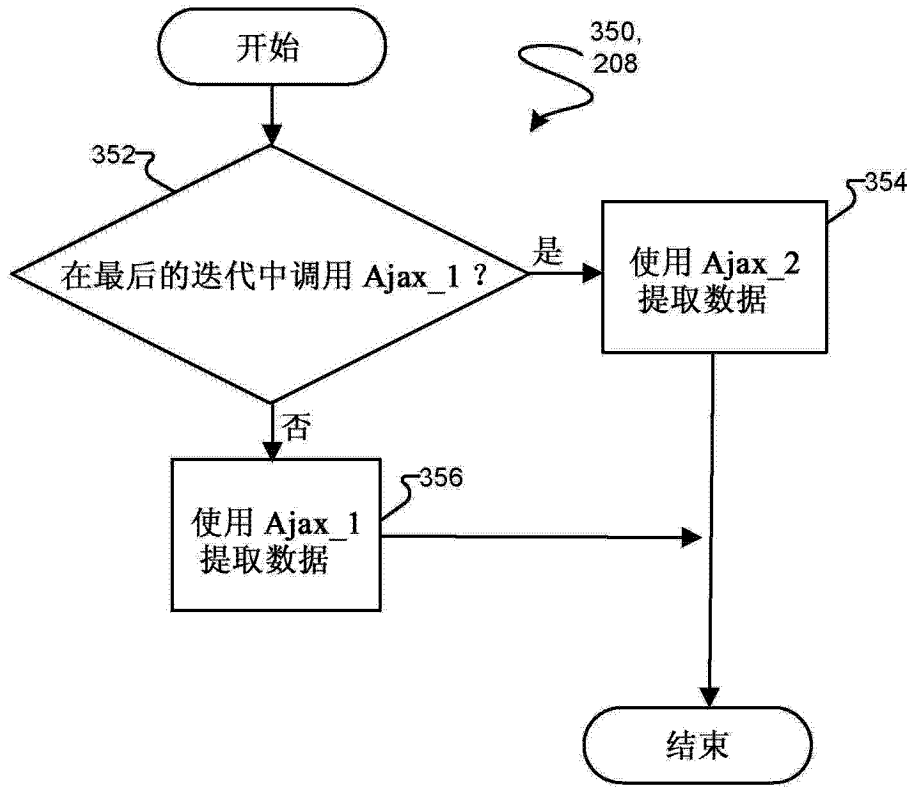


图 8

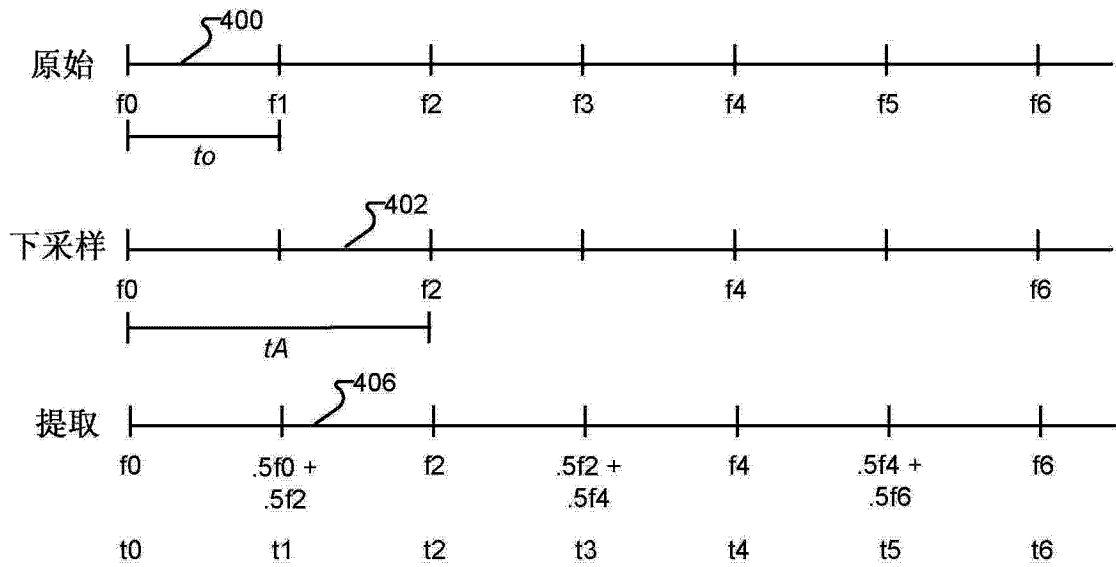


图 9A

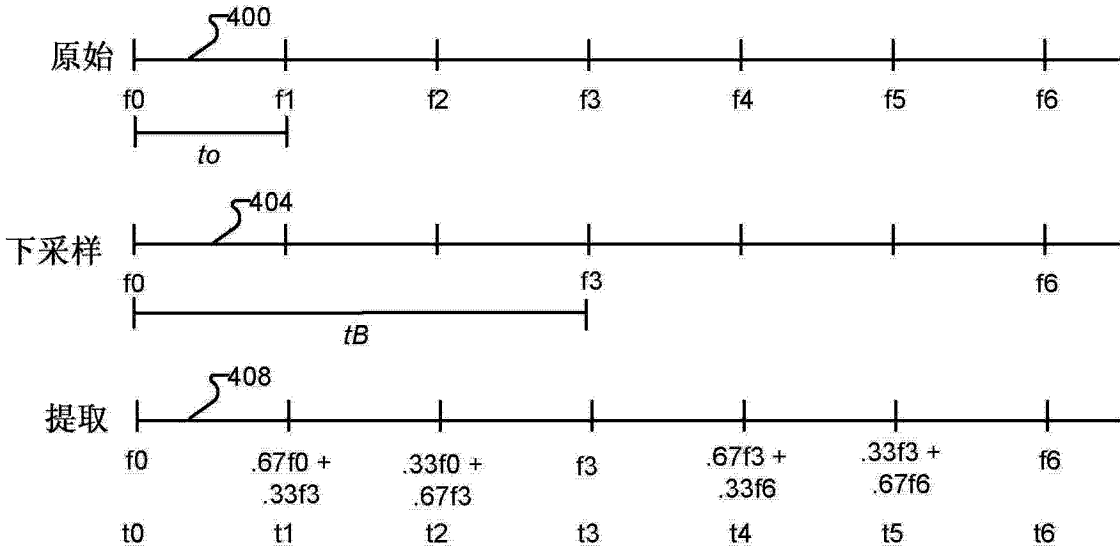


图 9B

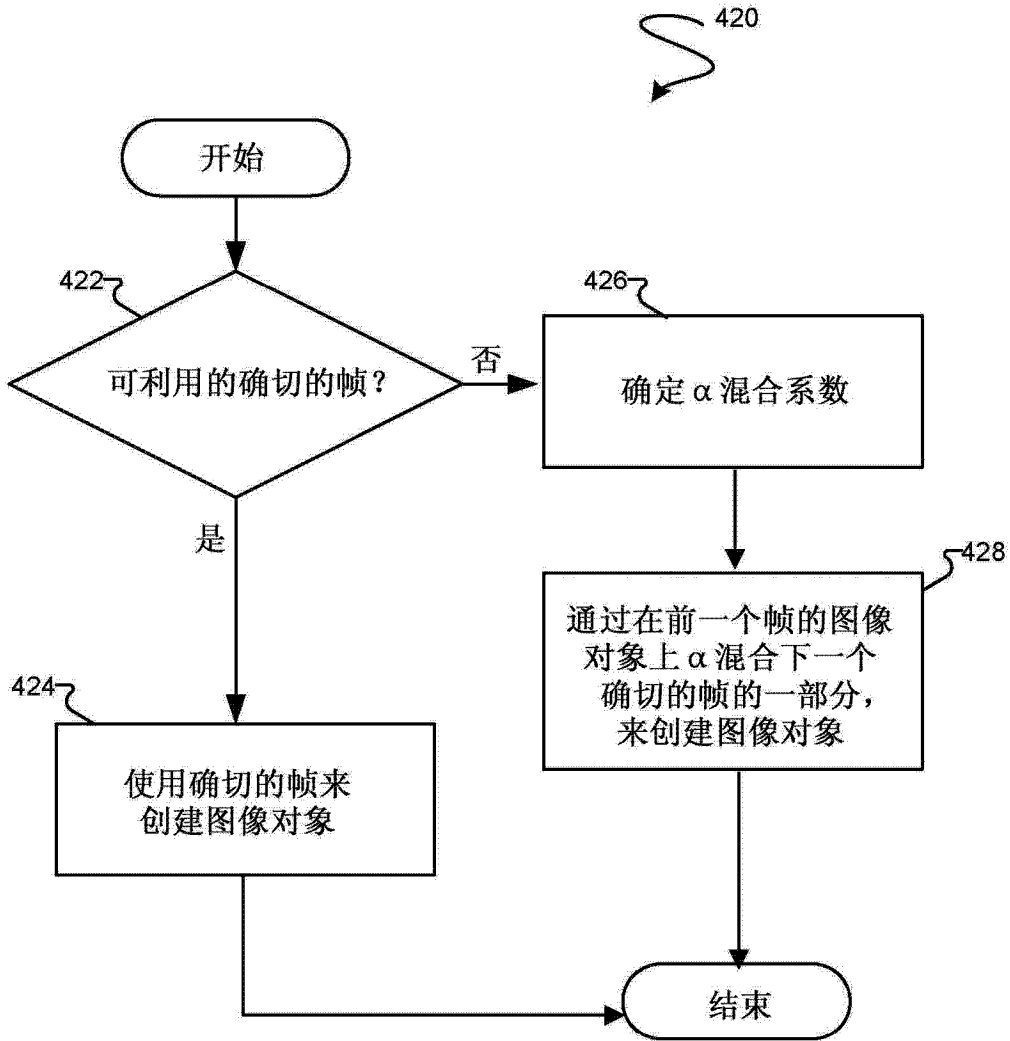


图 9C

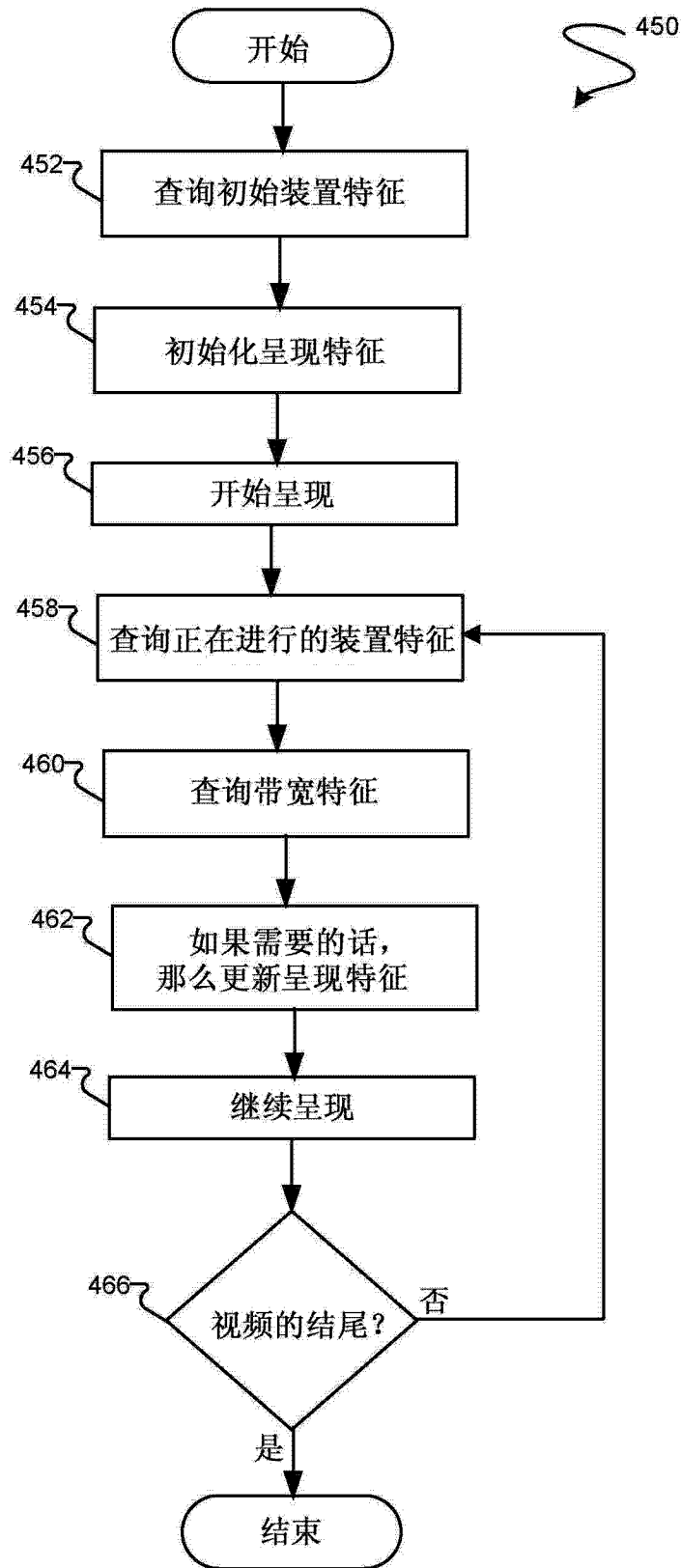


图 10

1. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

其中,发送所述脚本,使所述脚本由在所述用户计算平台上操作的所述应用程序解释,以使所解释的脚本直接将所述用户计算平台可访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据。

2. 根据权利要求1所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以在所述用户计算平台上本地地显示所述一个或多个帧图像。

3. 根据权利要求1所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以使得所解释的脚本直接将可由所述用户计算平台访问的经编码的视频数据解码成经解码的音频数据。

4. 根据权利要求3所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以:通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据;并且通过在所述用户计算平台上重放所述经解码的音频数据,来呈现所述经解码的音频数据。

5. 根据权利要求3所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以:通过在所述用户计算平台上显示所述一个或多个帧图像,来本地地呈现所述经解码的视频数据;并且通过重放所述经解码的音频数据,来本地地呈现所述经解码的音频数据。

6. 根据权利要求4或5中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,所述一个或多个帧图像包括多个帧图像,并且其中,发送所述脚本使得脚本由所述应用程序解释,以使得所解释的脚本直接同步所述帧图像的显示与所述经解码的音频数据。

7. 根据权利要求4或5中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,其中,所述一个或多个帧图像包括多个帧图像,并且其中,发送所述脚本使得所述脚本由所述应用程序解释,以确定与经解码的音频数据的重放相关联的音频重放时间基准,并且使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述经解码的音频数据的重放。

8. 根据权利要求7所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括以下中的至少一个:根据所述视频数据的呈现比所述音频重放时间基准晚第一阈值时间量这一确定,通过跳过一个或多个帧图像的显示,脱离顺序地提前显示帧图像;并且根据所述视频数据的呈现比所述音频重放时间基准提早第二阈值时间量这一确定,使得显示一个或多个帧图像的时间段比由所述视频数据的帧速率指示的时间段更长。

9. 根据权利要求7所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括:使用所述音频重放时间基准来确定当前帧索引;并且如果准备好显示与所述当前帧索引相关的帧图像,那么显示与所述当前帧索引相关的帧图像。

10. 根据权利要求9所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括确定与所述当前帧索引相关的帧图像在帧图像队列中位于一个或多个其

他帧图像之后,并且如果是这样的话,那么跳过显示所述一个或多个其他帧图像。

11. 根据权利要求 9 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括确定未准备好显示与所述当前帧索引相关的所述帧图像,并且根据这种确定,开始丢帧动作,所述丢帧动作包括以下中的一个或多个:记录未准备好呈现与所述当前帧索引相关的帧图像的指示;使得所述经解码的音频数据的重放暂停或放慢;使得所述经编码的视频数据的解码速率增大;使得切换成具有更低的帧速率的视频数据;通过将所述图像帧的一个子集解码,使得切换成有效下采样视频数据。

12. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代调度和执行包括将所述经编码的视频数据直接解码成所述帧图像的第一模拟线程;并且迭代调度和执行包括通过显示所述帧图像来呈现所述视频数据的第二模拟线程,来模拟多线程执行环境。

13. 根据权利要求 12 所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代调度和执行包括将所述经编码的音频数据直接解码成所述经解码的音频数据的第三模拟线程;并且迭代调度和执行包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的第四模拟线程,来模拟多线程执行环境。

14. 根据权利要求 4 所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以通过:迭代地调度和执行一种方法,来模拟多线程执行环境,所述方法在每个迭代上对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,选择和执行以下中的一个:

第一模拟线程,其包括将所述经编码的视频数据直接解码成所述帧图像;

第二模拟线程,其包括通过显示所述帧图像来呈现所述视频数据;

第三模拟线程,其包括将所述经编码的音频数据直接解码成经解码的音频数据;

第四模拟线程,其包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口。

15. 根据权利要求 12 所述的方法,其中,执行所述第一模拟线程,包括:将第一模拟线程分成多个子部分;在子部分对的性能之间,对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,暂时中断第一模拟线程,以选择和执行以下中的一个的迭代:第二模拟线程;以及包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的另一个模拟线程。

16. 根据权利要求 4 或 5 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据,包括将至少一对帧图像 α 混合在一起。

17. 根据权利要求 4 或 5 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据,包括在显示第一个所述帧图像与显示第二个所述帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像,包括 α 混合所述第一个帧图像和所述第二个帧图像。

18. 根据权利要求 17 所述的方法,其中,在显示所述第一个帧图像与显示所述第二个帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像,包括使用 α 混合系数 α , 在所述先前显示的图像上 α 混合所述第二个帧图像,由

$$\alpha = [\beta + 1 - \text{mod}(f\#, \beta)]^{-1}$$

提供所述 α 混合系数 α ,

其中 :

β 是等于所述一个或多个时间 +1 的数量的下采样因数 ;

$f\#$ 是参照所述一个或多个时间中的当前一个的数量的指数 ;以及

$\text{mod}(x, y)$ 是返回由 x 除以 y 的整数余数的模数运算符。

19. 根据权利要求 1 到 18 中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,使得所解释的脚本直接实现视频数据队列监视器的至少一部分,所述部分直接监视视频数据队列,以确定其状态,并且根据所述视频数据队列的所确定的状态,确定是否提取更多经编码的视频数据。

20. 根据权利要求 19 所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以实现所述视频队列监视器的第二部分,所述第二部分使用 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

21. 根据权利要求 19 所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以实现所述视频队列监视器的第二部分,所述第二部分交替地使用多个 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

22. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以直接多路分用所述视频数据和所述音频数据。

23. 根据权利要求 1 到 22 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以使得所解释的脚本直接实现帧缓冲监视器,所述帧缓冲监视器监视帧图像缓冲器,以确定其状态,并且根据帧图像缓冲器的所确定的状态,确定是否将更多所述经编码的视频数据解码。

24. 根据权利要求 1 到 23 中任一项所述的方法,其中,发送脚本,使得所述脚本由应用程序解释,以使得所解释的脚本直接将至少一个所述帧图像的图像文件格式从第一图像文件格式转换成第二图像文件格式,所述第二图像文件格式能够由所述应用程序本地地呈现。

25. 根据权利要求 1 到 24 中任一项所述的方法,包括将所述经编码的视频数据发送给所述用户计算平台。

26. 根据权利要求 25 所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:将至少一个所述帧图像的图像文件格式从第一图像文件格式转换成第二图像文件格式,所述第二图像文件格式能够由在所述用户计算机平台上操作的应用程序本地地呈现。

27. 根据权利要求 25 所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:

对于以第一文件格式设置的至少第一个所述帧图像:分析所述第一个帧图像的内容;根据所述第一个帧图像的内容的分析,在多个二次图像文件格式之中选择一个;并且如果所述第一文件格式与所述二次图像文件格式中所选择的一个格式不同,那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的一个格式;并且

对于以第二文件格式设置的至少第二个所述帧图像:分析所述第二个帧图像的内容;

根据所述第二个帧图像的内容的分析,在多个二次图像文件格式之中选择不同的一个;并且如果所述第二文件格式与所述二次图像文件格式中所选择的一个格式不同,那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的一个格式;

其中,所述多个二次图像文件格式中的每个能够由所述应用程序本地地呈现。

28. 根据权利要求1到27中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以使得所解释的脚本直接将至少一个所述帧图像转换成内容传输格式,所述内容传输格式能够由解释脚本的脚本解释器传递给所述应用程序。

29. 根据权利要求1到27中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以使得所解释的脚本直接将至少一个所述帧图像转换成 base64 内容传输格式。

30. 根据权利要求25所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:在内容传输格式中编码至少一个所述帧图像,据此,所述内容传输编码的帧图像可由解释所述脚本的脚本解释器传递给所述应用程序。

31. 根据权利要求25所述的方法,包括在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:对于多个帧图像中的每个:将所述帧图像编码成内容传输格式,据此,所述内容传输编码的帧图像可由解释所述脚本的脚本解释器传递给所述应用程序;并且通过与所述帧图像无关的填充数据,将所述内容传输编码的帧图像填充成特定的尺寸。

32. 根据权利要求25所述的方法,包括:在将所述经编码的视频数据发送给所述用户计算平台之前,将所述视频内容编码成经编码的视频数据,并且其中,编码所述视频内容包括:以 base64 内容传输格式编码至少一个帧图像。

33. 根据权利要求1到32中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以将至少一个所述帧图像本地地转换成相应的图像对象,所述图像对象能够由所述应用程序本地地呈现。

34. 根据权利要求1到33中任一项所述的方法,其中,所述应用程序包括网络浏览器。

35. 根据权利要求1到34中任一项所述的方法,其中,所述脚本嵌入 HTML 文档中。

36. 根据权利要求3所述的方法,其中,所述一个或多个帧图像包括多个帧图像,并且其中,发送所述脚本使得所述脚本由所述应用程序解释,以:通过重放所述经解码的音频数据,来呈现所述音频数据;并且通过选择所述多个帧图像中的一个子集和显示所述多个帧图像的所选择的一个子集,来呈现所述视频数据。

37. 根据权利要求4或5中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,改变包括可由所述用户计算平台访问的经编码的视频数据的文件。

38. 根据权利要求4或5中任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,修改以下一个或多个项中的至少一个特征:将所述经编

码的数据解码以及显示所述一个或多个帧图像。

39. 根据权利要求 4 或 5 中的任一项所述的方法,其中,发送所述脚本使得所述脚本由所述应用程序解释,以确定将所述经编码的视频数据发送给用户计算平台的数据速率,并且根据所述确定,改变包括发送给所述用户计算平台的所述经编码的视频数据的文件。

40. 根据权利要求 4 或 5 中任一项所述的方法,其中,发送所述脚本,使得所述脚本由所述应用程序解释,以确定将所述经编码的视频数据发送给所述用户计算平台的数据速率,并且根据所述确定,修改以下一个或多个中的至少一个特征:将所述经编码的数据解码以及显示所述帧图像。

41. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在计算机平台上使用所述应用程序,来解释脚本;

其中,解释所述脚本,使得所解释的脚本直接将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据。

42. 根据权利要求 41 所述的方法,其中,解释所述脚本使得所述应用程序在所述用户计算平台上本地地显示所述一个或多个帧图像。

43. 根据权利要求 41 所述的方法,其中,解释所述脚本使得所解释的脚本直接将可由用户计算平台访问的经编码的视频数据解码成经解码的音频数据。

44. 根据权利要求 43 所述的方法,其中,解释所述脚本,通过在所述用户计算平台上显示所述一个或多个帧图像,来呈现所述经解码的视频数据,并且通过在所述用户计算平台上重放所述经解码的音频数据,来呈现所述经解码的音频数据。

45. 根据权利要求 43 所述的方法,其中,解释所述脚本使得所述应用程序:通过在所述用户计算平台上显示所述一个或多个帧图像,来本地地呈现所述经解码的视频数据;并且通过重放所述经解码的音频数据,来本地地呈现所述经解码的音频数据。

46. 根据权利要求 44 或 45 中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,所述一个或多个帧图像包括多个帧图像,并且其中,解释所述脚本,使得所解释的脚本直接同步所述帧图像的显示与所述经解码的音频数据的重放。

47. 根据权利要求 44 或 45 中任一项所述的方法,其中,所述视频数据和所述音频数据作为部分视频内容项目彼此相关联,其中,所述一个或多个帧图像包括多个帧图像,并且其中,解释所述脚本,确定与所述经解码的音频数据的重放相关联的音频重放时间基准,并且使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述经解码的音频数据的重放。

48. 根据权利要求 47 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括以下中的至少一个:根据所述视频数据的呈现比所述音频重放时间基准晚第一阈值时间量这一确定,通过跳过一个或多个帧图像的显示,脱离次序地提前显示帧图像;并且根据所述视频数据的呈现比所述音频重放时间基准早第二阈值时间量这一确定,使得显示一个或多个帧图像的时间段比由所述视频数据的帧速率指示的时间段更长。

49. 根据权利要求 47 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数

据的重放同步,包括:使用所述音频重放时间基准来确定当前帧索引;并且如果准备好显示与所述当前帧索引相关的帧图像,那么显示与所述当前帧索引相关的帧图像。

50. 根据权利要求 49 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括:确定与所述当前帧索引相关的帧图像在帧图像队列中位于一个或多个其他帧图像之后,并且如果是这样的话,那么跳过显示所述一个或多个其他帧图像。

51. 根据权利要求 49 所述的方法,其中,使所述帧图像的显示与所述经解码的音频数据的重放同步,包括确定未准备好显示与所述当前帧索引相关的帧图像,并且根据这种确定,开始丢帧动作,所述丢帧动作包括以下中的一个或多个:记录未准备好呈现与所述当前帧索引相关的帧图像的指示;使得所述经解码的音频数据的重放暂停或放慢;使得所述经编码的视频数据的解码速率增大;使得切换成具有更低的帧速率的视频数据;通过将图像帧的一个子集解码,使得切换成有效下采样视频数据。

52. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本,通过:迭代调度和执行包括将所述经编码的视频数据直接解码成帧图像的第一模拟线程;并且迭代调度和执行包括通过显示所述帧图像来呈现所述视频数据的第二模拟线程,来模拟多线程执行环境。

53. 根据权利要求 52 所述的方法,其中,解释所述脚本,通过:迭代调度和执行包括将所述经编码的音频数据直接解码成经解码的音频数据的第三模拟线程;并且迭代调度和执行包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的第四模拟线程,来模拟多线程执行环境。

54. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本,通过:迭代调度和执行一种方法,来模拟多线程执行环境,所述方法在每个迭代上对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,选择和执行以下中的一个:

第一模拟线程,其包括将所述经编码的视频数据直接解码成帧图像;

第二模拟线程,其包括通过显示所述帧图像来呈现所述视频数据;

第三模拟线程,其包括将所述经编码的音频数据直接解码成经解码的音频数据;

第四模拟线程,其包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口。

55. 根据权利要求 52 所述的方法,其中,执行所述第一模拟线程,包括:将所述第一模拟线程分成多个子部分;在所述子部分对的性能之间,对一个或多个相应的条件进行一个或多个逻辑查询,并且根据所述一个或多个相应的条件,暂时中断所述第一模拟线程,以选择和执行以下中的一个的迭代:第二模拟线程;以及包括实现将所述经解码的音频数据提供给音频呈现器的音频呈现器接口的另一个模拟线程。

56. 根据权利要求 44 或 45 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据包括将至少一对帧图像 α 混合在一起。

57. 根据权利要求 44 或 45 中任一项所述的方法,其中,通过显示所述一个或多个帧图像来呈现所述视频数据包括在显示第一个所述帧图像与显示第二个所述帧图像之间的一个或多个时间的每个时间,显示 α 混合的图像包括 α 混合第一个帧图像和第二个帧图像。

58. 根据权利要求 57 所述的方法,其中,在显示所述第一个帧图像与显示所述第二个

帧图像之间的一个或多个时间的每个时间,显示所述 α 混合的图像包括使用 α 混合系数 α ,在先前显示的图像上 α 混合第二个所述帧图像,由

$$\alpha = [\beta + 1 - \text{mod}(f\#, \beta)]^{-1}$$

提供所述 α 混合系数 α ,

其中:

β 是等于所述一个或多个时间 +1 的数量的下采样因数;

$f\#$ 是参照所述一个或多个时间中的当前一个的数量的指数;以及

$\text{mod}(x, y)$ 是返回由 x 除以 y 的整数余数的模数运算符。

59. 根据权利要求 41 到 58 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接实现视频数据队列监视器的至少一部分,所述至少一部分直接监视视频数据队列,以确定其状态,并且根据所述视频数据队列的所确定的状态,确定是否提取更多经编码的视频数据。

60. 根据权利要求 59 所述的方法,其中,解释所述脚本实现所述视频队列监视器的第二部分,所述第二部分使用 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

61. 根据权利要求 59 所述的方法,其中,解释所述脚本实现所述视频队列监视器的第二部分,所述第二部分交替地使用多个 AJAX XMLHttpRequest 对象来提取更多经编码的视频数据。

62. 根据权利要求 44 到 45 中任一项所述的方法,其中,解释所述脚本使得所述应用程序本地地多路分用所述视频数据和所述音频数据。

63. 根据权利要求 41 到 62 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接实现帧缓冲监视器,所述帧缓冲监视器监视帧图像缓冲器,以确定其状态,并且根据所述帧图像缓冲器的所确定的状态,确定是否将更多的所述经编码的视频数据解码。

64. 根据权利要求 41 到 63 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接将至少一个所述帧图像的图像文件格式从第一图像文件格式转换成第二图像文件格式,所述第二图像文件格式能够由所述应用程序本地地呈现。

65. 根据权利要求 41 到 64 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接将至少一个所述帧图像转换成内容传输格式,所述内容传输格式能够由解释所述脚本的脚本解释器传递给所述应用程序。

66. 根据权利要求 41 到 64 中任一项所述的方法,其中,解释所述脚本使得所解释的脚本直接将至少一个所述帧图像转换成 base64 内容传输格式。

67. 根据权利要求 41 到 66 中任一项所述的方法,其中,解释所述脚本使得所述应用程序将至少一个所述帧图像本地地转换成相应的图像对象,所述图像对象能够由应用程序本地地呈现。

68. 根据权利要求 41 到 67 中任一项所述的方法,其中,所述应用程序包括网络浏览器。

69. 根据权利要求 1 到 68 中任一项所述的方法,其中,所述脚本嵌入 HTML 文档中。

70. 根据权利要求 43 所述的方法,其中,所述一个或多个帧图像包括多个帧图像,并且其中,解释所述脚本,通过重放所述经解码的音频数据,来呈现所述音频数据,并且通过选择所述多个帧图像的一个子集和显示所述多个帧图像的所选择的一个子集,来呈现所述视频数据。

71. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,改变包括可由所述用户计算平台访问的经编码的视频数据的文件。

72. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本查询所述用户计算平台的一个或多个特征,并且根据所述用户计算平台的所述一个或多个查询的特征,修改以下的一个或多个中的至少一个特征:将所述经编码的数据解码以及显示所述一个或多个帧图像。

73. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本,确定由所述用户计算平台接收所述经编码的视频数据的数据速率,并且根据所述确定,改变包括可由所述用户计算平台访问的经编码的视频数据的文件。

74. 根据权利要求 44 或 45 中任一项所述的方法,其中,解释所述脚本确定由所述用户计算平台接收所述经编码的视频数据的数据速率,并且根据所述确定,修改以下一个或多个中的至少一个特征:将所述经编码的视频数据解码以及显示所述一个或多个帧图像。

75. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括:

服务器计算装置,用于将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释;并且

其中,发送所述脚本使得所述脚本由在所述用户计算平台上操作的所述应用程序解释,以使得所解释的脚本直接将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据。

76. 根据权利要求 75 所述的系统,包括根据权利要求 2 到 40 所述的任一个特征或特征的组合。

77. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括处理器,该处理器被配置为提供:

解码部件,用于将可由所述用户计算平台访问的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;以及

其中,所述处理器被配置为通过在所述用户计算平台上执行所述应用程序,来提供所述解码部件,所述应用程序解释使得所解释的脚本直接解码所述视频数据的跨平台脚本。

78. 根据权利要求 77 所述的系统,包括根据权利要求 42 到 74 所述的任一个特征或特征的组合。

79. 一种由计算机系统执行的方法,用于呈现通过由用户计算平台接收的视频数据以及相关的音频数据体现的视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以:

通过显示构成所述视频数据的帧图像,来呈现所述视频数据;

重放所述相关的音频数据;

确定与所述相关的音频数据的重放相关的音频重放时间基准;并且

使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述相关的音频数据的重放。

80. 根据权利要求 79 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

81. 一种用于呈现通过由用户计算平台接收的视频数据以及相关的音频数据体现的视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括:

服务器计算装置,用于将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以:

通过显示构成所述视频数据的帧图像,来呈现所述视频数据;

重放所述相关的音频数据;

确定与所述相关的音频数据的重放相关的音频重放时间基准;并且

使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述相关的音频数据的重放。

82. 根据权利要求 81 所述的系统,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

83. 一种在用户计算平台上执行的方法,用于呈现通过由用户计算平台接收的视频数据以及相关的音频数据体现的视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用所述应用程序,来解释所述脚本,以:

通过显示构成所述视频数据的帧图像,来呈现所述视频数据;

重放所述相关的音频数据;

确定与所述相关的音频数据的重放相关的音频重放时间基准;并且

使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述相关的音频数据的重放。

84. 根据权利要求 83 所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

85. 一种用于呈现通过由用户计算平台接收的视频数据以及相关的音频数据体现的视频内容的系统,所述系统包括处理器,所述处理器被配置为在所述用户计算平台上执行应用程序,所述应用程序解释跨平台脚本,以:

通过显示构成所述视频数据的帧图像,来呈现所述视频数据;

重放所述相关的音频数据;

确定与所述相关的音频数据的重放相关的音频重放时间基准;并且

使得所解释的脚本根据所述音频重放时间基准直接同步所述帧图像的显示与所述相关的音频数据的重放。

86. 根据权利要求 85 所述的系统,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

87. 一种由计算机系统执行的方法,用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由被编译以在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以通过:

迭代调度和执行包括将所述经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据的第一模拟线程;并且

迭代调度和执行包括通过显示所述一个或多个帧图像来呈现所述经解码的视频数据的第二模拟线程,来模拟多线程执行环境。

88. 根据权利要求87所述的方法,包括根据权利要求1到40所述的任一个特征或特征的组合。

89. 一种用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括:

服务器计算装置,用于将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以通过:

迭代调度和执行包括将所述经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据的第一模拟线程;并且

迭代调度和执行包括通过显示所述一个或多个帧图像来呈现所述经解码的视频数据的第二模拟线程,来模拟多线程执行环境。

90. 根据权利要求89所述的系统,包括根据权利要求1到40所述的任一个特征或特征的组合。

91. 一种在用户计算平台上执行的方法,用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

在所述计算机平台上使用所述应用程序,来解释所述脚本;

其中,解释所述脚本,通过:

迭代调度和执行包括将所述经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据的第一模拟线程;并且

迭代调度和执行包括通过显示所述一个或多个帧图像来呈现所述经解码的视频数据的第二模拟线程,来模拟多线程执行环境。

92. 根据权利要求91所述的方法,包括根据权利要求41到74所述的任一个特征或特征的组合。

93. 一种用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容的系统,所述系统包括处理器,所述处理器被配置为在所述用户计算平台上执行应用程序,所述应用程序解释跨平台脚本,以通过:

迭代调度和执行包括将所述经编码的视频数据解码成包括一个或多个帧图像的经解

码的视频数据的第一模拟线程；并且

迭代调度和执行包括通过显示所述一个或多个帧图像来呈现所述经解码的视频数据的第二模拟线程，来提供模拟的多线程执行环境。

94. 根据权利要求 93 所述的系统，包括根据权利要求 41 到 74 所述的任何一个特征或特征的组合。

95. 一种由计算机系统执行的方法，用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容，所述用户计算平台是多个不同用户计算平台类型中的一种，所述方法包括：

将脚本发送给所述用户计算平台，所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释，并且进一步被配置为使得所述脚本由所述应用程序解释，以：

将所述经编码的视频数据解码成包括多个帧图像的经解码的视频数据；以及
通过显示所述帧图像来呈现所述经解码的视频数据；

其中，通过显示所述帧图像来呈现所述视频数据，包括将至少一对帧图像 α 混合在一起。

96. 根据权利要求 95 所述的方法，包括根据权利要求 1 到 40 所述的任何一个特征或特征的组合。

97. 一种用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容的系统，所述用户计算平台是多个不同用户计算平台类型中的一种，所述系统包括：

服务器计算装置，用于将脚本发送给所述用户计算平台，所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释，并且进一步被配置为使得所述脚本由所述应用程序解释，以：

将所述经编码的视频数据解码成包括多个帧图像的经解码的视频数据；以及
通过显示所述帧图像来呈现所述经解码的视频数据；

其中，通过显示所述帧图像来呈现所述视频数据，包括将至少一对帧图像 α 混合在一起。

98. 根据权利要求 97 所述的系统，包括根据权利要求 1 到 40 所述的任何一个特征或特征的组合。

99. 一种在用户计算平台上执行的方法，用于呈现通过由所述用户计算平台接收的经编码的视频数据体现的视频内容，所述方法包括：

在所述用户计算平台上接收脚本，所述脚本被配置为由在所述用户计算平台上操作的应用程序解释；

使用所述应用程序，来解释所述脚本；

其中，解释所述脚本：

将所述经编码的视频数据解码成包括多个帧图像的经解码的视频数据；以及
通过显示所述帧图像来呈现所述经解码的视频数据；并且

其中，通过显示所述帧图像来呈现所述视频数据，包括将至少一对帧图像 α 混合在一起。

100. 根据权利要求 99 所述的方法，包括根据权利要求 41 到 74 所述的任何一个特征或特

征的组合。

101. 一种用于呈现通过由用户计算平台接收的经编码的视频数据体现的视频内容的系统,所述系统包括处理器,所述处理器被配置为在所述用户计算平台上执行应用程序,所述应用程序解释跨平台脚本,以:

将所述经编码的视频数据解码成包括多个帧图像的经解码的视频数据;以及
通过显示所述帧图像来呈现所述经解码的视频数据;并且

其中,通过显示所述帧图像来呈现所述视频数据,包括将至少一对帧图像 α 混合在一起。

102. 根据权利要求 101 所述的系统,包括根据权利要求 1 到 74 所述的任一个特征或特征的组合。

103. 一种由计算机系统执行的方法,用于在用户计算平台上呈现由经编码的视频数据体现的视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

编码一系列帧图像,以提供经编码的视频数据,其中,编码这系列帧图像,包括将所述帧图像编码成内容传输格式,据此,所述内容传输编码的帧图像可由解释所述脚本的脚本解释器传递给所述应用程序;并且

将所述经编码的视频数据发送给所述用户计算平台;

其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以:

将所发送的编码视频数据解码成包括所述内容传输编码的帧图像的经解码的视频数据;并且

通过显示所述帧图像来呈现所述经解码的视频数据。

104. 根据权利要求 103 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

105. 一种用于在用户计算平台上呈现由经编码的视频数据体现的视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括:

服务器计算装置,被配置为:

将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释;

编码一系列帧图像,以提供经编码的视频数据,其中,编码这系列帧图像,包括将所述帧图像编码成内容传输格式,据此,所述内容传输编码的帧图像可由解释所述脚本的脚本解释器传递给所述应用程序;并且

将所述经编码的视频数据发送给所述用户计算平台;

其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以:

将所发送的编码视频数据解码成包括所述内容传输编码的帧图像的经解码的视频数据;并且

通过显示所述帧图像来呈现所述经解码的视频数据。

106. 根据权利要求 105 所述的系统,包括根据权利要求 1 到 40 所述的任一个特征或特

征的组合。

107. 一种由计算机系统执行的方法,用于在用户计算平台上呈现由经编码的视频数据体现的视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,所述应用程序在所述用户计算平台上进行操作;

编码一系列帧图像,以提供所述经编码的视频数据,其中,编码所述帧图像,包括:

对于以第一文件格式设置的至少第一个所述帧图像:

分析所述第一个帧图像的内容;

根据所述第一个帧图像的内容的分析,在多个二次图像文件格式之中选择一个;并且

如果所述第一文件格式与所述二次图像文件格式中所选择的一个格式不同,那么将所述第一个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的一个格式;并且

对于以所述第二文件格式设置的至少第二个所述帧图像:

分析所述第二个帧图像的内容;

根据所述第二个帧图像的内容的分析,在所述多个二次图像文件格式之中选择不同的一个;并且

如果所述第二文件格式与所述二次图像文件格式中所选择的不同的一个格式不同,那么将所述第二个帧图像的图像文件格式转换成所述二次图像文件格式中所选择的不同的一个格式;

其中,所述多个二次图像文件格式中的每个能够由所述应用程序本地地呈现;以及

将所述视频数据发送给所述用户计算平台;

其中,所述脚本进一步被配置为使得脚本由所述应用程序解释,以:

将所发送的编码视频数据解码成包括所述帧图像的经解码的视频数据;并且

通过显示所述帧图像来呈现所述经解码的视频数据。

108. 根据权利要求 107 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

109. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 107 到 108 所述的任一个特征或特征的组合。

110. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

编码一系列帧图像,以提供经编码的视频数据,其中,编码所述帧图像,包括将所述帧图像转换成内容传输格式;

将所述经编码的视频数据发送给所述用户计算平台,其中,发送所述经编码的数据,使得在所述用户计算平台上进行操作的应用程序的脚本解释器解释所述脚本,以:将所发送的编码视频数据解码成具有所述内容传输格式的经解码的帧图像数据;并且将具有所述内容传输格式的经解码的帧图像数据传递给所述应用程序,以便呈现所述经解码的帧图像数据。

111. 根据权利要求 110 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特

征的组合。

112. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 110 到 111 所述的任一个特征或特征的组合。

113. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:编码一系列帧图像,以提供经编码的视频数据,其中,编码所述帧图像,包括将所述帧图像转换成内容传输格式,据此,当脚本解释器解释所述脚本,以将所述经编码的视频数据解码成包括所述内容传输编码的帧图像的经解码的视频数据并且通过显示所述帧图像来呈现所述经解码的视频数据时,可将内容传输编码的帧图像从在用户计算平台上进行操作的应用程序的脚本解释器中传递给应用程序。

114. 根据权利要求 113 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

115. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 113 到 114 所述的任一个特征或特征的组合。

116. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以:

查询所述用户计算平台的一个或多个特征;以及

发送请求,以接收从体现所述视频内容的多种经编码的视频数据中选择的体现所述视频内容的第一种经编码的视频数据,根据所述用户计算平台的一个或多个查询的特征,来进行所述选择。

117. 根据权利要求 116 所述的方法,其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以:

在所述用户计算平台上接收所述第一种经编码的视频数据时,将所述第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据;以及

通过显示所述一个或多个第一帧图像,来呈现所述第一经解码的视频数据。

118. 根据权利要求 116 所述的方法,其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以:

将来自所述多种经解码的视频数据中并且在所述用户计算平台上接收的第二种经编码的视频数据解码成包括一个或多个第二帧图像的第二经解码的视频数据;以及

通过显示所述一个或多个第二帧图像,来呈现所述第二经解码的视频数据;

其中,在发送接收所述第一种经编码的视频数据的请求之前,解码所述第二种经编码的视频数据并且呈现所述第二经解码的视频数据。

119. 根据权利要求 118 所述的方法,其中,所述第二种经编码的视频数据与所述第一种经编码的视频数据不同。

120. 根据权利要求 117 所述的方法,其中,所述多种经编码的视频数据包括至少两种经编码的视频数据,至少两种所述经编码的视频数据中的每个包括以相应的图像格式中表示的相应的一系列经编码的帧图像,并且其中,所述至少两种经编码的视频数据的相应的图像格式彼此不同。

121. 根据权利要求 116 到 120 中任一项所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

122. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括:

服务器计算装置,用于将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以:

查询所述用户计算平台的一个或多个特征;以及

发送请求,以接收从体现所述视频内容的多种经编码的视频数据中选择地体现所述视频内容的第一种经编码的视频数据,根据所述用户计算平台的一个或多个查询的特征,来进行所述选择。

123. 根据权利要求 122 所述的系统,其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以:

在所述用户计算平台上接收第一种经编码的视频数据时,将所述第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据;以及

通过显示所述一个或多个第一帧图像,来呈现所述第一经解码的视频数据。

124. 根据权利要求 122 所述的系统,其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以:

将来自所述多种经解码的视频数据中并且在所述用户计算平台上接收的第二种经编码的视频数据解码成包括一个或多个第二帧图像的第二经解码的视频数据;以及

通过显示所述一个或多个第二帧图像,来呈现所述第二经解码的视频数据;

其中,在发送接收所述第一种经编码的视频数据的请求之前,解码所述第二种经编码的视频数据并且呈现所述第二经解码的视频数据。

125. 根据权利要求 124 所述的系统,其中,所述第二种经编码的视频数据与所述第一种经编码的视频数据不同。

126. 根据权利要求 123 所述的系统,其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以执行以下中的至少一个:

配置解码所述第一种经编码的视频数据的解码特征,根据所述用户计算平台的一个或多个查询的特征,来进行所述解码特征的配置;以及

配置显示所述一个或多个第一帧图像的显示特征,根据所述用户计算平台的一个或多个查询的特征,来进行所述显示特征的配置。

127. 根据权利要求 122 到 126 中任一项所述的系统,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

128. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由在多个用户计算平台类型的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以:

将体现所述视频内容并且在所述用户计算平台上接收的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据;以及

通过显示所述一个或多个帧图像,来呈现所述经解码的视频数据;

查询所述用户计算平台的一个或多个特征;以及

执行以下中的至少一个:

配置解码所述经编码的视频数据的解码特征,根据所述用户计算平台的一个或多个查询的特征,来进行所述解码特征的配置;以及

配置显示所述一个或多个帧图像的显示特征,根据所述用户计算平台的一个或多个查询的特征,来进行所述显示特征的配置。

129. 根据权利要求 128 所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

130. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 128 到 129 所述的任一个特征或特征的组合。

131. 一种在用户计算平台上执行的方法,用于在所述用户计算平台上呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上操作的应用程序解释;

使用所述应用程序,来解释所述脚本;

其中,解释所述脚本:

查询所述用户计算平台的一个或多个特征;以及

发送请求,以接收从体现所述视频内容的多种经编码的视频数据中选择的体现所述视频内容的第一种经编码的视频数据,根据所述用户计算平台的一个或多个查询的特征,来进行所述选择。

132. 根据权利要求 131 所述的方法,其中,解释所述脚本:

在所述用户计算平台上接收所述第一种经编码的视频数据时,将所述第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据;以及

通过显示所述一个或多个第一帧图像,来呈现所述第一经解码的视频数据。

133. 根据权利要求 131 到 132 中任一项所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

134. 一种用于在用户计算平台上呈现视频内容的系统,所述用户计算平台是多个不同用户计算平台类型中的一种,所述系统包括根据权利要求 131 到 133 所述的任一个特征或特征的组合。

135. 一种在用户计算平台上执行的方法,用于在用户计算平台上呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上进行操

作的应用程序解释；

使用所述应用程序，来解释所述脚本；

其中，解释所述脚本；

将体现所述视频内容并且在用户计算平台上接收的经编码的视频数据解码成包括一个或多个帧图像的经解码的视频数据；

通过显示所述一个或多个帧图像，来呈现所述经解码的视频数据；

查询所述用户计算平台的一个或多个特征；以及

执行以下中的至少一个：

配置解码所述经编码的视频数据的解码特征，根据所述用户计算平台的一个或多个查询的特征，来进行所述解码特征的配置；以及

配置显示所述一个或多个帧图像的显示特征，根据所述用户计算平台的一个或多个查询的特征，来进行所述显示特征的配置。

136. 根据权利要求 135 所述的方法，包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

137. 一种用于在用户计算平台上呈现视频内容的系统，所述系统包括根据权利要求 135 到 136 所述的任一个特征或特征的组合。

138. 一种由计算机系统执行的方法，用于在用户计算平台上呈现视频内容，所述用户计算平台是多个不同用户计算平台类型中的一种，所述方法包括：

将脚本发送给所述用户计算平台，所述脚本被配置为由在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释，并且进一步被配置为使得所述脚本由所述应用程序解释，以；

将体现所述视频内容并且在所述用户计算平台上接收的第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据；

通过显示所述一个或多个第一帧图像，来呈现所述第一经解码的视频数据；

确定由所述用户计算平台接收所述第一种经编码的视频数据的速率并且执行以下中的至少一个：

根据所确定的速率，发送请求，以接收体现所述视频内容的第二种经编码的视频数据；

根据所确定的速率，修改解码所述第一种经编码的视频数据的特征；以及

根据所确定的速率，修改显示所述一个或多个第一帧图像的特征。

139. 根据权利要求 138 所述的方法，包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

140. 一种用于在用户计算平台上呈现视频内容的系统，所述用户计算平台是多个不同用户计算平台类型中的一种，所述系统包括：

服务器计算装置，用于将脚本发送给所述用户计算平台，所述脚本被配置为由在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释，并且进一步被配置为使得所述脚本由所述应用程序解释，以；

将体现所述视频内容并且在所述用户计算平台上接收的第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据；

通过显示所述一个或多个第一帧图像,来呈现所述第一经解码的视频数据;

确定由所述用户计算平台接收所述第一种经编码的视频数据的速率并且执行以下中的至少一个:

根据所确定的速率,发送请求,以接收体现所述视频内容的第二种经编码的视频数据;

根据所确定的速率,修改解码所述第一种经编码的视频数据的特征;以及

根据所确定的速率,修改显示所述一个或多个第一帧图像的特征。

141. 根据权利要求 140 所述的系统,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

142. 一种在用户计算平台上执行的方法,用于呈现视频内容,所述方法包括:

在所述用户计算平台上接收脚本,所述脚本被配置为由在所述用户计算平台上进行操作的应用程序解释;

使用所述应用程序,来解释所述脚本,其中,解释所述脚本:

将体现所述视频内容并且在所述用户计算平台上接收的第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据;

通过显示所述一个或多个第一帧图像,来呈现所述第一经解码的视频数据;

确定由所述用户计算平台接收所述第一种经编码的视频数据的速率并且执行以下中的至少一个:

根据所确定的速率,发送请求,以接收体现所述视频内容的第二种经编码的视频数据;

根据所确定的速率,修改解码所述第一种经编码的视频数据的特征;以及

根据所确定的速率,修改显示所述一个或多个第一帧图像的特征。

143. 根据权利要求 142 所述的方法,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

144. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括处理器,所述处理器被配置为在所述用户计算平台上执行应用程序,所述应用程序解释跨平台脚本,以:

将体现所述视频内容并且在所述用户计算平台上接收的第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据;

通过显示所述一个或多个第一帧图像,来呈现第一经解码的视频数据;

确定由所述用户计算平台接收所述第一种经编码的视频数据的速率并且执行以下中的至少一个:

根据所确定的速率,发送请求,以接收体现所述视频内容的第二种经编码的视频数据;

根据所确定的速率,修改解码所述第一种经编码的视频数据的特征;以及

根据所确定的速率,修改显示所述一个或多个第一帧图像的特征。

145. 根据权利要求 144 所述的系统,包括根据权利要求 41 到 74 所述的任一个特征或特征的组合。

146. 一种由计算机系统执行的方法,用于在用户计算平台上呈现视频内容,所述用户计算平台是多个不同用户计算平台类型中的一种,所述方法包括:

将脚本发送给所述用户计算平台,所述脚本被配置为由在所述多个用户计算平台类型中的任一个上进行操作的应用程序解释,并且进一步被配置为使得所述脚本由所述应用程序解释,以;

将体现所述视频内容并且在所述用户计算平台上接收的第一种经编码的视频数据解码成包括一个或多个第一帧图像的第一经解码的视频数据;

通过显示所述一个或多个第一帧图像,来呈现所述第一经解码的视频数据;

确定由所述用户计算平台接收所述第一种经编码的视频数据的速率;并且

根据所确定的速率,发送请求,以接收体现所述视频内容的第二种经编码的视频数据;

147. 根据权利要求 146 所述的方法,其中,所述脚本进一步被配置为使得所述脚本由所述应用程序解释,以在确定所述速率并且发送所述请求以接收所述第二种经编码的视频数据之后;

将所述第二种经编码的视频数据解码成包括一个或多个第二帧图像的第二经解码的视频数据;

通过显示所述一个或多个第二帧图像,来呈现所述第二经解码的视频数据。

148. 根据权利要求 147 所述的方法,其中,所述第一种经编码的视频数据与所述第二种经编码的视频数据不同。

149. 根据权利要求 146 到 148 中任一项所述的方法,包括根据权利要求 1 到 40 所述的任一个特征或特征的组合。

150. 一种用于在用户计算平台上呈现视频内容的系统,所述系统包括根据权利要求 146 到 149 所述的任一个特征或特征的组合。

151. 一种计算机系统,包括使用数据通信网络有效地连接的一个或多个计算机,所述数据通信网络适合于执行根据权利要求 1 到 40、79、80、87、88、95、96、103、104、107、108、110、111、113、114、116 到 121、128 以及 129 所述的任一种方法。

152. 根据权利要求 75、76、81、82、89、90、97、98、105、106、109、112、115、122 到 127 以及 130 中任一项所述的系统,其中,所述系统包括使用数据通信网络有效地连接的一个或多个计算机。

153. 一种计算机可读介质,包括计算机可执行代码,所述计算机可执行代码在由包括使用数据通信网络有效地连接的一个计算机或多个计算机的计算机系统执行时,使得所述计算机系统执行根据权利要求 1 到 40、79、80、87、88、95、96、103、104、107、108、110、111、113、114、116 到 121、128 以及 129 所述的任一种方法。