



(12) 发明专利申请

(10) 申请公布号 CN 103679785 A

(43) 申请公布日 2014. 03. 26

(21) 申请号 201310401067. 1

G06T 19/00(2011. 01)

(22) 申请日 2013. 09. 05

(30) 优先权数据

61/697, 785 2012. 09. 06 US

(71) 申请人 想象技术有限公司

地址 英国赫特福德郡

(72) 发明人 J·A·梅特卡夫

(74) 专利代理机构 北京市金杜律师事务所

11256

代理人 王茂华

(51) Int. Cl.

G06T 15/00(2011. 01)

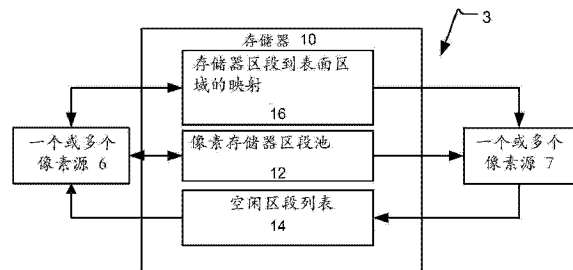
权利要求书3页 说明书10页 附图9页

(54) 发明名称

用于部分帧缓冲区更新的系统和方法

(57) 摘要

本发明的多个方面包括一种像素源,该像素源产生了用于分成多个区域的一个被渲染表面的数据。一种映射识别出多个存储器区段,这些存储器区段存储了用于该表面的每个区域的像素数据。这种映射可以从一个先前被渲染表面中指向在对该渲染处理进行渲染的过程中未被改变的区域识别出存储了像素数据的存储器区段。此类改变的/未改变的状态是在逐区域的基础上进行跟踪的。可以为每个存储器区段维持一个计数器,以便跟踪多少个表面使用了存储在其中的像素数据。可以维持一个空闲存储器区段池。读取一个表面(如为了通过一个显示控制器来显示一个被渲染表面)包括:识别并读取这种映射以便识别存储了用于该表面的多个区域的像素数据的每个存储器区段、对其进行读取、并且更新用于被读取的这些存储器区段的计数器。



1. 一种计算系统,包括:

非瞬态存储器;以及

处理器,该处理器被配置成用于

对用于第一渲染表面的像素数据进行渲染,这些像素数据被组织在共同包括该第一渲染表面的多个区域内;

在该存储器内存储展现该第一渲染表面的数据,这些数据包括

存储器区段,这些存储器区段存储了用于该第一渲染表面的区域的像素数据,以及

在每个区域与到存储用于该区域的像素数据的存储器区段的参引项之间的映射;

在该存储器内存储展现第二渲染表面的数据,这些数据包括

存储器区段,这些存储器区段仅存储了该第二渲染表面的被确定为不同于该第一渲染表面的对应的相应区域的像素数据的像素数据区域,以及

映射,这种映射包括:指向以下存储器区段的那些参引项,这些存储器区段存储了用于该第一渲染表面的具有与该第二渲染表面的对应的相应区域相同的像素数据的区域的像素数据,以及指向以下存储器区段的那些参引项,这些存储器区段存储了用于该第二渲染表面的具有与该第一渲染表面的对应的相应区域不同的像素数据的区域的像素数据。

2. 如权利要求1所述的计算系统,其中,该处理器被进一步配置成用于维持与每个存储器区段相关联的对应计数,该计数跟踪使用存储在存储器区段内的像素数据的表面的数目。

3. 如权利要求1所述的计算系统,进一步包括显示控制器,该显示控制器被配置成用于读取这种映射并使用这种映射来识别存储了用于该第一表面的每个区域的像素数据的对应的存储器区段。

4. 如权利要求3所述的计算系统,其中,该显示控制器被进一步配置成用于将与被识别为存储了用于该第一表面的像素数据的每个存储器区段相关联的计数器减值。

5. 一种用所存储的数据来展现图形渲染输出的机器实施的方法,该方法包括:

提供渲染表面,该渲染表面包括像素数据的多个区域;

确定该多个区域中的哪些区域具有与来自先前的渲染表面的像素数据的对应的相应区域不同的像素数据以及该多个区域中的哪些区域具有与来自该先前的渲染表面的像素数据的对应的相应区域相同的像素数据,其中用于该先前的渲染表面的每个区域的像素数据被存储在非瞬态存储器的对应的存储器区段之中;

在该非瞬态存储器的对应存储器区段内存储用于该多个区域的每个区域的像素数据,这些区域被确定为具有不同于来自该先前的渲染输出的对应的相应区域的像素数据;以及

产生映射,这种映射包括指向每个存储器区段的参引项,这些存储器区段存储了用于被确定为具有与来自该先前的渲染输出的相应区域不同的像素数据的区域的像素数据,并且为该渲染表面的具有与该先前的渲染表面相同的像素数据的每个区域产生指向存储了用于该先前的渲染表面的这个区域的像素数据的存储器区段的参引项。

6. 如权利要求5所述的机器实施的方法,进一步包括对与每个存储被确定与该渲染表面和该先前的渲染表面中相同的像素数据区域的存储器区段相关联的对应计数进行增值。

7. 如权利要求6所述的机器实施的方法,进一步包括对与每个响应于消耗来自该存储器区段的像素数据的存储器区段相关联的对应计数进行减值。

8. 一种计算机图形系统,包括:

非瞬态存储器;以及

像素数据汇,该像素数据汇用于通过执行以下过程来从该非瞬态存储器读取用于一系列渲染表面的像素数据,该过程包括

读取第一渲染表面的每个区域与指向存储用于该区域的像素数据的存储器区段的参引项之间的映射,

从存储用于该第一渲染表面的相应区域的像素数据的存储器区段读取展现该第一渲染表面的数据,

使用定义该第一渲染表面的数据,

读取第二渲染表面的每个区域与存储用于该第二渲染表面的该区域的像素数据的存储器区段之间的映射,

其中针对该第二渲染表面的与该第一渲染表面的区域完全相同的区域,该映射将相同的那些存储器区段识别为用于该第一渲染表面的映射,并且不同的存储器区段存储用于该第二渲染表面的不同于该第一渲染表面的相应区域的区域的数据,

从存储用于该第二渲染表面的相应区域的像素数据的映射后的存储器区段读取展现该第二渲染表面的数据,以及

使用定义该第一渲染表面的数据。

9. 如权利要求 8 所述的计算机图形系统,进一步包括存储器控制器,该存储器控制器被配置成用于维持存储器区段池,这些存储区段被分配用于存储来自像素源的像素数据,该存储器区段池具有能够同时存储用于具有所确定的分辨率的至少三个渲染表面的像素数据的容量。

10. 如权利要求 9 所述的计算机图形系统,其中,该像素数据源被进一步配置成用于将该第一被渲染表面内的像素数据的每个相应区域与对应的计数器相关联以及如果第二渲染表面参引该区域则对该第一渲染表面的区域的对应计数器进行增值。

11. 如权利要求 10 所述的计算机图形系统,其中,该像素数据汇被进一步配置成用于每次来自该区域的像素数据被用于产生用于显示的帧时对用于该第一被渲染表面内的每个像素数据区域的对应计数器进行减值。

12. 如权利要求 9 所述的计算机图形系统,进一步包括存储单元,该存储单元被配置成用于监控每个计数器以及响应于任何计数器指示该存储器区段不再被参引用于任何渲染表面,将该存储器区段返回至空闲存储器区段列表。

13. 如权利要求 9 所述的计算机图形系统,进一步包括在存储器内维持来自该第一被渲染表面的每个像素数据区域,直到已经消耗了该第二被渲染表面。

14. 如权利要求 8 所述的计算机图形系统,其中,该像素源包括 3D 图形处理器以及视频编码器中的一个或多个。

15. 如权利要求 8 所述的计算机图形系统,其中,该像素源包括显示控制器。

16. 一种用于操作存取像素数据的机器实施的方法,该方法包括:

接收输出像素数据帧的指示符;

响应于该指示符,识别像素数据区域的映射,这些像素数据区域共同定义该帧;

读取通过来自非瞬态存储器的映射识别的那些像素数据区域;

维持跟踪在定义任何其他像素数据帧时是否使用每个像素数据区域的指示符集合；以及

对于定义该被输出帧的以及不用于定义任何其他帧的任何像素数据区域而言，在存储该像素数据区域的非瞬态存储器内解除分配存储器区段。

17. 如权利要求 16 所述的用于操作存取像素数据的机器实施的方法，其中，显示控制器执行该方法并且该接收包括接收将该像素数据帧输出至显示器上的指示符。

18. 如权利要求 16 所述的用于操作存取像素数据的机器实施的方法，进一步包括在产生定义该帧的那些像素数据区域的渲染过程中监控存储像素数据区域的存储器区段的写锁状态。

19. 如权利要求 16 所述的用于操作存取像素数据的机器实施的方法，其中，每个像素数据区域对应于由图形处理元件产生的一片像素数据。

用于部分帧缓冲区更新的系统和方法

背景技术

发明领域：

[0001] 以下内容涉及图形处理、图形处理系统，如图形处理器单元（GPU），并且在示例具体方面涉及影像合成和图像生产技术。

[0002] 相关领域：

[0003] 用于存储等待被输出至显示器的数据的缓冲区通常被称为“帧缓冲区”，引发了显示一个帧时间序列中的一帧的概念，其中“帧”为一个完整的像素数据集合。在一些计算架构中，帧缓冲区还可以用作有待如被 GPU 进一步处理的数据源。

[0004] 在这种情况下，GPU 可以从帧缓冲区（例如，存储图像数据的存储器中的一个位置，如位图）上读取数据，并使用该数据和其他数据一起形成一张合成图像。然后该 GPU 将展现整个合成图像的数据写出至输出帧缓冲区。显示控制器可以从该输出帧缓冲区内上读取该合成图像并引起显示该合成图像。因此，存储整个像素数据集以展现每个可显示帧。

[0005] 作为一个更加具体的示例，可以设置一个常规渲染和显示系统以执行三倍缓冲方法，其中，预留了存储器的三个部分，每个部分能够保存一个全分辨率可显示帧（例如，缓冲区 0、1 和 2），并且被安排在一条翻转链（flip chain）内。例如，图形处理器可以将有待显示的连续帧写到缓冲区 0、1 和 2 并从 0 重新开始。类似地，在再次返回读取缓冲区 0 之前，显示控制器读取缓冲区 0、1 和 2。当缓冲区 0、1 和 2 中的任一个或多个之间像素数据相同时，在该像素数据出现的所有缓冲区内重复该像素数据。

[0006] 概述

[0007] 在根据本披露的实现方式中，展现多个渲染表面的数据（例如，有待显示的像素数据帧）括用于该渲染表面的表面区域和存储器区段之间的一个对应的映射集合。每个渲染表面可以被展现作为来自先前的表面的多个区域以及被确定为不同于该先前的表面的像素数据的像素数据区域的像素数据的组合。

[0008] 作为示例，一种用于渲染的方法包括存储展现一个第一被渲染表面的数据，该第一被渲染表面包括多个存储用于该第一被渲染表面的对应区域的像素数据的存储器区段，以及这些区域与这些对应的存储器区段之间的映射。该方法还包括跟踪使用各存储器区段中的像素数据的多个被渲染表面。为了存储展现一个后续被渲染表面（该表面具有具有对应于该第一被渲染表面内的像素数据区域的像素数据区域），该方法包括确定该后续被渲染表面内哪些区域具有未从该第一参引表面改变的像素数据以及哪些区域具有不同的像素数据。对于带有相同像素数据的区域，用于该后续被渲染表面的映射被创建成包括一个指向与该第一表面相同的存储器区段的参引项。对于带有不同像素数据的区域，此类像素数据被写至从空闲区段池中获得的存储器区段，并且在该映射中提供了指向这些存储器区段的参引项。维持了存储用于该第二表面的像素数据的所有存储器表面的计数，这些计数可以包括对用于每个表面的计时器进行增值。

[0009] 为了读取根据上述方法存储的表面，一种装置存取用于该表面的映射，以确定用

于该表面的每个像素数据区域被存储在哪儿(例如,在通过具体的存储地址识别的存储器区段内)。该装置存取该像素数据并更新从那些存储器区段使用像素数据的那些表面的跟踪。带有像素数据的不再被任何表面使用的存储器区段可以被释放返回可用的空闲存储器区段池以接收用于进一步的被渲染表面的像素数据。

[0010] 附图简要说明

[0011] 本披露的各种特征和方面将从以下详细说明中变得更加明显,将结合附图阅读以下详细说明,其中:

[0012] 图 1 描绘了系统的功能元件的概观,在该系统中可以实践所披露的方法和结构方面;

[0013] 图 2 描绘了此处根据所披露的实施例的可以由一个或多个渲染表面构成的可显示帧的时间流程;

[0014] 图 3 描绘了定义被分成多个区域的第一表面的数据的展示;

[0015] 图 4 描绘了像素源以及像素汇的多个示例方面,该像素源渲染表面并存储存储器区段内的被渲染表面的区域以及产生区域到存储器区段的映射,该像素汇消耗该被渲染表面的区域;

[0016] 图 5 描绘了一个第二表面的展示,包括与该第一表面一样的区域;

[0017] 图 6 描绘了展现使用展现该第一表面的数据的区域的第二表面以及修改后的区域到存储器区段的映射的方面;

[0018] 图 7 描绘了存储器区段中存储器表面区域的进一步的方面;

[0019] 图 8 基于展现该第一表面的数据描绘了一种示例方法,该方法用于确定展现该第二表面的数据集合以及存储用于具有不同于该第一表面的像素数据的第二表面的区域的数据的存储器区段;

[0020] 图 9 描绘了根据本披露的多个方面存取像素数据表面的示例过程,如用于产生可显示帧;

[0021] 图 10 描绘了执行图 9 的过程中的示例数据流

[0022] 图 11 描绘了在其中可以实践所披露的方面的系统的多个方面;以及

[0023] 图 12 描绘了可以配置成执行本披露的多个方面的示例处理器的多个方面。

[0024] 详细说明

[0025] 许多常见的计算装置(如台式计算机、膝上型计算机、平板计算机或移动电话)包括一个可以产生用于显示的图形数据的图形子系统,如液晶显示器(LCD)或其他合适的显示技术。该显示器具有一个分辨率,该分辨率通常特征在于多个水平和垂直像素,如 1080p 显示器,其具有 1920 个水平像素(列)和 1080 个垂直像素(排)。刷新速率确定该显示器上显示的图像多长时间变化一次。该显示器上显示的每张独特的图像可以被认为是一个可显示帧。

[0026] 在将图像数据存储于存储器内的背景下,将存储器表面概念化为一个平面,该平面具有根据一种图案分布在该平面上的多个像素位置。每个像素位置将具有被分配用于存储展现该像素的数据的存储量。因此,对于 1080p 显示器而言,该存储器表面将包括 1920×1080 像素,每个像素带有一定量被分配的存储空间,如 4 个字节。然而,此类存储器表面的物理实现方式可以在实现方式之间不同并且不必遵循该概念模型。在本披露的背景

下,渲染表面被分成多个区域,并且每个区域在该表面内包含这些像素中的一些。例如,一个区域可以是一个 8×8 的正方形像素块、一个矩形、一个竖直或水平像素带等。更通常地,一个区域可以是任何形状,尽管希望该形状将是一个矩形,并且高度和宽度各自为一个或多个像素,并且典型地至少一个尺寸为多个像素。区域可以由不连续的像素形成,但为了简单性,本说明书描述了在像素区域为连续的假设下的示例实现方式。

[0027] 可以基于系统架构考虑选定一个区域内的多个像素和一个区域的长度 VS 宽度。例如,设计目标可以是使存储页面大小与存储一个像素区域所需的存储量相匹配。然而,没有要求一个像素区域存储在存储器中的一个连续部分内。相反,每个像素区域指向存储器表面内的其他像素区域将具有一个已知的相对位置,从而使得可以通过根据那些相对位置安排这些区域的像素来展现该存储器表面内的所有像素。同样地,普通技术人员将认识到一个表面内的那些像素区域各自为该表面内建立的像素的具体分组的一个逻辑结构。

[0028] 在一些情况下,一个渲染表面全等于一个可显示帧,而在其他情况下,渲染表面可以是一个可显示帧的一部分,可以包括一个可显示帧和“屏幕外”像素,或者可以是有待被渲染和存储在存储器内的图像数据,或者两者都是。例如,像素数据可以被渲染至一个表面并且被存储用于用作纹理以及被显示。

[0029] 图 1 描绘了根据本披露的系统 3 的实现方式的概念图。系统 3 包括一个像素源 6 和一个像素汇 7,其分别起到生成和消耗像素数据的作用。像素源 6 可以从存储器 10 读取并写至存储器 10。像素源 6 可以从存储器区段池 12 中读取并写入像素存储器区段(例如,如以上介绍的,写入和读取用于存储器表面的区域的数据)。池 12 可以用专用存储器实现,如嵌入在图形处理器中的存储器。这种存储器可以专用于这个目的,或者可以在需要时对其进行分配。池 12 还可以使用图形处理器私有的存储器或者在图形处理器和其他功能单元之间共享的存储器实现。存储器常驻控制元件(如池 12 中的空闲区域列表)还可以在嵌入式存储器中实现。实现方式可以嵌入池 12 的一部分以及存储器常驻控制元件。在示例中,区段池 12 可以是足够大的存储器以便以当前操作的显示分辨率存储像素数据的至少三个全帧,以便实现三倍缓冲。像素源 6 还可以从空闲区段列表 14 上读取,该空闲区段列表包括用于存储器区段的标识符,这些存储器区段可用于存储用于在被像素源 6 在任何表面上操作的像素数据区域。这样,如果像素源 6 使用来自空闲区段列表 14 的存储器区段,像素源 6 将用于该存储器区段的标识符从空闲区段列表 14 上清除。像素源 6 还可以从存储器 10 上读取和写入该存储器,以便更新存储器区段到表面区域的映射 16。通过示例,可以用用于一个具体存储器表面的一个区域列表以及识别用于存储用于每个所列区域的像素数据的一个或多个存储器区段的信息来实现映射 16。多个此类映射 16 可以存储在存储器 10 内,其中,每个映射与一个对应的表面相关联。

[0030] 如以下解释的,当要使用或显示表面时,存取与该具体表面相关联的映射,以检索将像素数据存储在该表面内的存储器区段列表。如以下将要解释的,像素汇 7 使用来自映射 16 的一个选定的映射来识别有待存取的存储器区段,以便输出用于显示的一个具体表面(例如,以便产生一个可显示帧)。每个映射可以参引来自任何一个或多个不同表面的存储器区段,从而使得可以从不同的并且不连续的存储位置读取存储器区段。

[0031] 在示例应用中,如图 2 中所描绘的,像素汇 7 包括一个输出一个可显示帧时间序列,其中,可显示帧 20、21 和 21 被形成为被输出用于在显示器 25 上显示。此操作与显示控

制器的惯例操作相反,该操作可以被给予一个用于有待被显示的帧的起始点的位置,并且连续地从该起始点读取数据以输出帧用于显示。显示控制器还可以支持分散/集中操作。显示控制器还可以具有一个支持存储器的虚拟化的存储器管理单元(提供通过物理存储器的不同部分实现的虚拟存储器的一个连续区段)。通过进一步的对比,在本披露的实现方式中,存储在单个存储器区段内的像素数据可以被像素汇 7 用于形成多个用于显示的帧。例如,存储在给定的存储器区段的像素数据区域可以用于两个或三个连续的帧。

[0032] 图 3 描绘了被分成多个区域 110 至 114 的表面 105。每个区域具有定义数量的像素排和列,从而使得每个区域具有定义数量的像素。在水平和垂直方向上的多个区域由屏幕分辨率(用于固定区域大小)确定。可以为具体目标类型的显示器选定区域大小,并且基于以下考虑,如像素源 6 和像素汇 7 的特征、存储器接口特征以及其他特定实现方式标准。在一些实现方式中,表面 105 展现一系列表面中的一个表面,每个表面具有相同的分辨率和区域细分,从而使得该表面 105 内的每个像素(或像素区域)可以据说在该系列表面中的其他表面内具有相对应的像素(区域)。

[0033] 图 4 描绘了图 1 的系统 3 的操作的多个方面,以便产生用于图 3 的表面 105 的像素数据。图 4 描绘了接收用于渲染像素表面的信息的像素源 6,此类信息可以包括数据、设置和命令信息,如图元、纹理、照明信息、着色器等。像素源 6 的输出包括表面 105 的区域(例如,区域 110 至 114)与存储器区段标识符(“参引项”)之间的映射 127,存储这些输出以便产生用于表面 105 的所有区域的映射 120。

[0034] 生成用于多个区域的像素数据 128 并将其存储在来自池 12 的存储器区段内。从空闲区段列表 14 中识别从池 12 中被选定存储用于区域 110 至 114 的像素的区段。例如,存储地址可以被用作存储地址的区段标识符或部分。例如,可以使用 32 位或 64 位的存储地址。一个 1080p 帧具有 207.36 万个像素。对于 64 像素的区域,一个帧将具有 32,400 个区域,并且一组 32 位参引项将需要约 130 千字节的存储。另一种实现方式是识别一个用于存储器区段池的基存储地址,如 64 百万字节,其可以存储 24 或 32 位色(例如,1 字节用于每个 RGB 以及一条阿尔法通道用于每个像素),并构建指示符作为与该基存储地址的偏移量。尽管逐字节定址 64 百万字节需要一个 26 位的地址空间,但每个区域具有一个 256 字节的容量(用于 4 字节像素)。因此,更低的 8 位将维持不变,并且 18 位可以用于每个指示符。另一种展现表面的方法的示例是识别先前表面并为每个被建立成包含不同于该先前表面的对应区域的像素数据区域提供该区域的外显识别以及一个存储用于该区域的像素数据的存储器区段。这些示例为说明性的并且不具有限制性。

[0035] 在一种方法中,表面 105 展现一系列具有相同格式和分辨率的表面中的一个表面,从而使得展现映射 120 的数据可以为一系列预定义顺序下的参引项,如存储器参引项,并且不需要结合指向存储其像素信息的存储器区段的参引项来外显地识别每个区域。因此,像素汇 107 可以通过读取存储器区段(这些存储器区段存储用于该表面的每个区域的像素数据)来存取用于表面的像素数据;这些指向这些存储器区段的参引项的顺序隐含地定义像素数据区域的相对空间安排(如用于显示)。反过来,可以为每个区域内的像素数据的格式建立一个惯例。

[0036] 结合将像素数据写入至所选定的存储器区段,计数 13 的集合被维持用于这些存储器区段,其如以下所解释的,跟踪有多少个表面将每个存储器区段用作一个像素数据源。

[0037] 更新空闲区段列表 14, 以清除用于存储像素数据 128 的存储器区段。此示例示出了单独的空闲区段列表 14, 该空闲区段列表识别准备接收像素数据的空闲区段。然而, 其他实现方式可以直接使用这些计数, 并识别哪些存储器区段具有值为 0 的计数, 以及选定有待被用于该基础上的存储器区段。

[0038] 因此, 图 4 描绘了映射 120 识别存储定义表面 105 的像素数据的一个存储器区段集合, 并且共同地, 映射 120 和该存储器区段集合展现表面 105 的被渲染像素数据。每个映射例如可以通过指针或存储地址来识别映射 120。此类地址和指针可以被提供至像素汇 7, 该像素汇由此使用该被识别的映射确定从其上可以检索用于表面的像素数据的任何多个不同的存储器区段。因此, 用于三倍缓冲情况的翻转链将包括指向三个映射的指针或存储地址序列。此方法容纳了通过从该序列上清除一个对应的指针或存储地址并对与存储用于该表面的像素数据的存储器区段相关联的计数进行减值来丢弃该序列中的表面之一。因此, 如下所述, 如果表面 105 被输出用于显示或用于进一步的处理, 则存取映射 120 以识别存储用于表面 105 的像素数据并且然后从那些存储器区段上读取该像素数据。然而, 在描述此类操作前, 以下直接描述了展现进一步被渲染表面的数据的存储。

[0039] 图 5 描绘了表面 106 作为一组像素数据区域, 包括存在于表面 105 内的区域(即, 这些区域内的像素数据与表面 105 的相应区域内的相同)。然而, 表面 106 的一些区域不同于表面 105, 并且具体地, 区域 116、117 和 118 不同。

[0040] 图 6 描绘了映射 120 被复制到映射 121, 该映射 120 映射了用于表面 105 的像素数据区域所存储的位置。从空闲区段列表 14 上获得用于不同于表面 105 的每个像素数据区域的空闲存储器区段, 并且像素数据被存储在那些存储器区段内。可以通过更新映射 121 来生成映射 122, 以使用新获得的现在存储用于表面 106 的区域的像素数据的存储器区段来替换用于存储用于表面 105 的像素数据的存储器区段。如所描绘的, 现在区域 116 至 118 具有映射 122 中所列的相应的存储器区段参引项。

[0041] 此外, 如关于图 7A 所解释的, 维持了用于存储器区段的计数 13。图 7A 描绘了存储器区段池 12, 这些存储器区段共同存储用于被渲染表面的像素数据。现在, 某些存储器区段存储用于表面 105 和用于表面 106 两者的像素数据, 具体地, 区域 110 和 111 内的像素数据相同并且存储此类数据的存储器区段(当写表面 105 时被填充)用于定义表面 106。对计数 13 进行增值以反映用于那些存储器区段的数据现在用于定义两个表面。在一些实现方式中可以在专用的存储部分内维持计数; 例如, 为了区域大小的最大支持分辨率和最大粒度, 被设计成能够三倍缓冲的处理器可以提供足够大的内存存储以容纳 3 深翻转链(2 位计数器)。例如, 在 1080p 显示器以及 64 像素区域中, 约 64k 位可以被分配用于此类存储器。在一些情况下, 可以将区域大小固定在该架构中。然而, 最大分辨率可以是可改变的。而且, 当用专用翻转链支持多个显示器或者在根据本披露的存储器内展现其他工作渲染表面时, 将需要相应地调整此类存储器的容量。在一些示例中, 存储器控制器可以分配预先确定的存储容量, 用于存储多达预先确定数量的全分辨率帧。也可以在此类存储器中使用高速缓存技术。

[0042] 图 7B 描绘了一个示例, 其中, 每个存储器区段可以由多个可能位于不连续的存储器区域内的部分组成; 具体地, 存储用于区域 116 的像素数据的区段被描绘成由部分 116A 至 116C 组成。

[0043] 图 8 根据本披露描绘了一种用于存储用于被渲染表面的像素数据的示例方法。图 8 描绘了接收 205 源数据以便在渲染的表面时产生像素数据,以及产生 207 此类像素数据。在一些实现方式中,可以在以上如被描述用于表面 105 和 106 的那些区域中产生该像素数据。如果该像素数据没被产生或另外被分成多个区域,则该像素数据被分成 209 多个区域。在 211 处,对是否用来自之前被渲染表面的数据展现该被渲染表面作出决定。通常,当正在被渲染的表面的像素数据区域与之前所渲染的表面相同时,则来自之前所渲染的表面的区域可以用于定义正在被渲染的表面,并且如以上所介绍的,复制(213)用于先前的表面的映射(例如,映射 120),并且对于正在被渲染的表面的每个区域而言,对该像素数据是否与该先前的表面的相应区域相同或不同做出确定(215)。如果用于该区域的像素数据没有变化(217),则用用于该存储器区段(将用于该未改变的区域的像素数据存储在该存储器区段内)的指示符更新(231)用于被渲染表面的映射,以及在 232 处,更新计数 13。

[0044] 如果该区域内的像素数据已经变化,则在 225 处,存储器区段被识别(如来自空闲区段列表 14)并用于写入(227)用于该区域的被渲染像素数据。在 229 处,更新用于正在被渲染的表面的映射以使该区域与存储用于该区域的像素数据的存储器区段相关。在 230 处,更新计数 13。

[0045] 在各情况下,可以向前推进(233)至有待考虑的后续区域。通过循环回 215 所描绘的,可以根据上述流程处理正在被渲染的表面的每个区域,以便考虑该后续区域。在多倍缓冲(例如,三倍缓冲)的背景下,为了丢弃这些表面之一,从一个序列和用于存储该被丢弃的表面内的像素数据区域的存储器区段的计数上清除指向用于该表面的映射的存储地址或指针。

[0046] 然而,如果在定义有待被渲染的表面时,不使用用于先前表面的像素数据,则在 219 处,获得用于存储像素区域的存储器区段,在 221 处,用于多个区域的像素数据被写入至相应的存储器区段,以及在 223 处,创建这些区域和这些存储器区段之间的映射。

[0047] 因此,图 8 描绘了对于正在被渲染的给定表面而言(或者,更通常地,对于有待通过所存储的像素数据展现的表面而言),可以使用存储用于现有表面的区域的像素数据的存储器区段来展现该表面,其中存储了不同于该现有表面的像素数据以及带有说明可以在哪儿找到每个像素数据区域的映射。在这些示例中,用于一个现有表面的像素数据区域用于定义正在被渲染的表面;然而,在其他实现方式中,可以使用来自多个现有表面的像素数据。

[0048] 可以执行各种方法来确定与现有表面的相应区域相比给定的区域是否包括不同的像素数据。在一个示例中,可以创建并且存储相应区域内的像素数据的循环冗余校验(CRC)或散列。可以产生用于后续表面的区域的像素数据和 CRC。可以比较这些 CRC,以及如果其不匹配,则确定这些区域是不同的。在另一个示例中,可以进行逐像素比较,如在硬件 XOR 单元中。仍然可以执行其他试探法。例如,可以有先验地确定是否保证表面的给定区域不是不同的方法。例如,在合成两个表面的背景下(例如,当使用用户界面叠加在背景上时,如图像),该用户界面可以具有已知的边界,从而使得这些已知边界外的像素区域将不会改变。例如,应用程序编程界面调用可以具有关于受该调用影响的表面的边界的信息,如 `glScissor()` 调用,这将允许排除受这种界面调用影响的像素数据区域。

[0049] 在使用分片(tiling)方法的渲染架构的具体背景下,其中,一起处理被发现存在

于一片屏幕空间内的几何体(而不是按正确的提交顺序处理该几何体,不管几何体的每个元件被映射到屏幕空间哪里)。在一个示例中,当像素源正在产生用于一系列表面的像素数据时,当那些表面被组织成多个分片时,在该像素源写入该像素数据前,该像素源在打算存储用于每个分片的像素数据的存储位置上获得一个写锁。在产生该像素数据的过程中,该像素源或该系统内的另一个元件将像素数据写入用于任何具有不同于之前被渲染表面内的相应分片的像素数据的对应存储位置。因此,通过该像素源获得该写锁是对像素汇的一种指示,指示已经更新并且需要按顺序读取该具体分片内的像素数据,以汇编该表面(例如,用于显示)。然而,对于其他该像素源没有为其获得写锁的存储位置,像素汇可以使用该信息推断该像素数据与之前被渲染表面相同并且将为来自该之前被渲染表面的相应分片的那些分片获得像素数据。

[0050] 在一种实现方式中,可以为整个表面预先分配一个存储位置集合,但只用于具有已改变像素数据的分片的像素数据填充。作为这种分配的副产物,可以建立分片和存储位置之间的对应关系并被一个或多个像素源和一个或多个像素汇参引。在另一种实现方式中,不预先分配用于一个帧的所有分片的存储位置;可以根据需要分配存储位置。为了解释可变分片写入要求和分配的可变等待时间,可以维持一个预先分配的存储位置子集(例如,可以维持4个、8个或16个位置)。在该分配过程中,维持边带数据,该边带数据识别哪个分片的像素数据被存储在每个位置上。在后面的实现方式中,为用于所识别的分片的像素数据分配存储器的事实是对一个或多个像素汇的一种指示,指示已经更新用于该分片的像素数据。

[0051] 以上描述示出了像素源如何可以用保存读取和写入存储宽带方式生成描述被渲染表面的数据以及如何可以总体提高操作(如合成操作)效率。例如,可以合成一个输出帧的整个区域,或者仅已知具有某些修改可能性的区域。作为示例,在渲染表面或表面的一部分的过程中,可以维持影响每个像素的合成数据(例如,深度分类和混合操作)。在合成过程中,如深度分类过程中(不具有半透明性),可以确定用于每个像素的一个图像数据源。例如,在开窗口环境中,确定像素是否将来源于给定的窗口或背景。对于每个区域而言,可以维持一个标记,该标记保留用于该区域内的像素的像素数据源的运行记录。如果发现任何像素不同于相应的像素,则可以标记包含该像素的区域,从而可以存储用于该区域的像素数据。在一些实现方式中,将会影响一个给定的像素或一个像素集合的那些图像部件在不同的时间可以是可供使用的。例如,在 8×8 的块中,指向用于每个像素的可见表面的参引项可以是可供使用的。在背景上合成窗口时,可以用对应的多对三角形图元(被映射到展现存储器内的窗口或背景的纹理上)展现该窗口和该背景。这些数据可以之前已经由处理器75或者由视频解码器70生成或者使得可以从一些其他来源获得这些数据。用于该窗口和该背景各自的图元在一定范围内可以具有标识符,以及硬件掩码可以确定一条像素线上的所有像素标识符在给定范围内是否具有一个标识符(例如,对每个像素ID的逐位掩码的输出进行与操作)。

[0052] 在纹理化过程中还可以完成源识别,作为确定纹理源的副产物。在许多情况下,纹理化是一个逐像素过程,从而使得当对每个像素进行纹理化时,可以更新该源标记。对于这种源测试,实现方式可以具有提前退出;例如,如果关于参引图像已经确定任一像素或像素组需要更新,则不需要检查该分片中的剩余像素,因为将输出用于该整个分片的像素数据,

即使有一个像素是不同的。

[0053] 图 9 和图 10 用于解释用于存取根据本披露展现的像素数据的表面的示例过程。在 305 处, 决定存取或输出表面(例如, 图 10 中重新产生的表面 105), 例如, 以便执行合成操作或输出、产生可显示帧。在 307 处, 该过程包括选定和检索用于该表面的映射(例如, 映射 120)。在 309 处, 使用这种映射识别存储用于该表面的像素数据的存储器区段。例如, 参引映射 120, 识别用于区域 110 至 114 的存储器区段, 以及在 311 处, 从池 12 中检索用于这些区域的像素数据。在 313 处, 对用于在 311 处读取的存储器区段的(计数器 13 的)每个计数器进行减值在 315 处, 识别已经达到 0 的任何计数器(例如, 用于存储用于区域 112 至 114 的像素数据的存储器区段的计数器)并将其释放返回空闲区段列表 14。

[0054] 如果正在读取该像素数据用于显示, 则在 319 处, 进一步的示例过程部分可以包括临时存储该像素数据, 以及在 320 处, 可以用适合用于显示的格式输出该像素数据(例如, 如图 10 中所示)。显示控制器可以执行图 9 的过程; 例如, 像素汇 7 可以包括一个显示控制器。在其他示例中, 图 9 的过程可以由一个处理器执行, 如图形处理器, 以便获得用于表面的像素数据以及使用该像素数据进行进一步的图形操作, 并且仍然在其他实现方式中, 图形处理器和显示控制器两者都可以执行根据图 9 的过程。

[0055] 图 11 描绘了一种示例系统的多个元件, 可以在该系统中实践所披露的多个方面。在图 11 中, 用处理器 75 表示像素源 6 (来自图 1), 以及用显示控制器(85) 表示像素汇 7, 尽管如以上所披露的, 但处理器 75 也还可以起到像素汇的作用, 而且除了处理器 75 外可以存在像素数据源。此外, 视频解码器 70 以及存储器 10 可以通过总线 20 耦联到处理器 75 上。显示器 25 可以耦联到显示控制器 85 上。

[0056] 用于处理 75 的 API50 和驱动器 52 可以和软件如操作系统 45 和应用程序 40 一起被提供至界面处理器 75。应用程序 40 可以检索输入 35, 以及产生用于渲染表面的数据、设置和命令。其他渲染输入可以包括加速结构 66、纹理 67、几何体 68 以及着色器代码 69, 可以通过驱动器 52 或通过应用程序 40 或者两者存取这些输入。操作系统 45 还可以接收来自用户界面 30 的输入, 例如, 该用户界面包括视觉 UI 元件 31、键盘 32 以及鼠标 33 中的一个或多个。例如, 应用程序 40 可以使用 OpenGL®、OpenCL®、OpenRL®、DirectX® 中的一个或多个或者另一个合适的 API 与处理器 75 连接。驱动器 52 可以接收来自应用程序 40 的 API 调用, 以及将这种调用转换成适用于处理器 75 的计算设置。

[0057] 应用程序 40 可以在 OS45 上运行, 该 OS 可以提供允许在可供使用的屏幕空间内改变窗口的大小和移动窗口的 API 调用并且可以提供多种效果, 如半透明性。同样地, 开窗口过程 47 可以跟踪所有窗口和任何可用于显示的其他事项(如桌面上的图标、屏幕保护程序、背景壁纸等)的位置的尺寸。

[0058] 应用程序 40 可以接收从 UI 元件生成的输入, 通过示例包括 UI 元件 25 (例如, 能够引起触摸输入的屏幕上菜单)、键盘 26 以及鼠标 27。此类输入可以被 OS45 处理并被引导至应用程序 40。应用程序 40 还可以使用 API 与 OS45 通信。OS45 可以发挥各种不同的通过这种 API 可供应用程序 40 使用的功能。

[0059] 处理器 75 可以使用延迟着色方法来渲染表面, 其中, 延迟纹理和其他面向像素的操作, 直到确定可见表面之后。可以在基于分片的渲染器内渲染表面(例如, 表面 105)的区域, 在一个示例中, 分片和区域之间存在 1:1 的对应关系。

[0060] 显示控制器 85 读取使用映射识别的存储器区段,这些区段各自对应于一个可用于读取的表面。显示控制器 85 可以具有或控制可供使用的临时缓冲区,以便从所读取的存储器区段上汇编像素数据。在一些装置中,显示控制器 85 还可以能够执行如缩放、旋转、剪切功能以及多监视器操作。因此,本披露不暗示显示控制器 85 读取的像素数据必须被显示为无修改。一些实现方式可以提供一个显示控制器,该显示控制器根据本说明书运行,但是其可以控制常驻有显示器的永久性存储(例如,用于显示器的后背存储器,如 LCD)。例如,针对一个具体的帧,这种显示控制器可以引起更新这种后背存储器的多个区域,同时针对该帧,该存储器的其他区域不变。

[0061] 图 12 描绘了在某种背景下的处理器 75 的示例构造的多个方面。处理器 75 可以包括一个可编程计算集群 156、一组协处理器 157 以及一个调度器 155。这些单元中的每一个或其某一部分可以与高速缓存层级 150 通信,反过来,该高速缓存层级与存储器总线 160 通信。各种数据管理器可以设置有待在处理器 75 中调度和执行的计算;示例包括顶点数据管理器 151、像素数据管理器 152 以及一个通用数据管理器 153。主机接口 158 可以通过控制和总线 161 与处理器 75 通信。存储器总线 160 以及控制和寄存器总线 161 可以为图 1 的总线 20 的一部分。系统存储器接口 162 将存储器总线 160 连接到系统存储器层级 164 上。

[0062] 图 12 中描绘的元件的每一个可以具有组成元件,为了清晰性目的,没有描绘这些组成元件。例如,可编程计算集群 156 可以具有多个计算集群,每个带有一个本地调度器,并且调度器 155 可以全局地在那些集群之间分配计算。协处理器 170 可以包括纹理协处理器或用于其他种类的可以有益地在固定功能硬件内被执行的算法(或部分这些算法)的协处理器,与在可编程集群 165 中的执行相反。例如,存储器 10 可以由多个不同的物理存储器组成,包括在图中没有外显地识别的工作存储器。例如,上述计数可以存储在处理器 75 的本地工作存储器内,该处理器可以具有维持这些计数的固定的或部分固定的功能硬件。

[0063] 为了描述的清晰性,没有描述或抽象地描绘了常规元件。可以针对不同的实现方式有区别地执行所述功能性,从而使得不同的实现方式可以具有执行各种和不同组别功能性的元件。通过示例,可以在一个与被分配至计算线程以及用于其他目的存储器共享的存储器内实现存储像素数据的存储器。在一些实现方式中,可以有用于存储像素数据的专用视频存储器,而可以在一个或多个单独的存储器内提供用于应用程序和线程的工作存储器。这些存储器可以使用存储器层级的共同元件,如共享主存储器。所描绘的各种数据结构和数据类型还可以存在同一物理存储器内。这种数据源可以包括网络连接以及网络装置内的缓冲区。以上所披露的示例最直接地与以下情况相关:图形处理器,如 3D 图形处理器(其可以通过图形处理单元或被编程执行图形处理的通用处理器中的一个或多个实现)为一个像素源,以及显示控制器为一个像素汇。

[0064] 在各实施例中,相对于图形处理单元,像素源可以包括一个视频解码器(例如,可以输出压缩视频流以及输出像素数据的表面用于显示的处理元件)和一个无线显示接收器。无线显示发送器可以是像素汇。在一些实现方式中,像素源和像素汇可以在同一系统内,或者在芯片上的同一系统或同一装置内,而在其他实现方式中,像素源和像素汇可以被一个或多个网络分开或临时地被分开,或者为两种情况的组合。例如,通过一次写入像素数据并在下一次读取该像素数据,图形处理器和视频解码器可以是像素源和像素汇两者。

[0065] 如从本披露中将明显的,可以用硬件、软件、固件或它们的组合的形式实现所披露的一些部件和功能性。如果以固件和 / 或软件被实现,则这些功能可以作为机器可读介质上的一个或多个指令或代码被存储。在一个示例中,该介质为非瞬态的。示例包括一个带有数据结构编码的计算机可读介质和一个用计算机程序编码的计算机可读介质。其他种类的介质包括传输介质。非瞬态介质可以是任何可以被机器存取的有形介质。通过举例,但不限于,这种介质可以包括 RAM、ROM、EEPROM、CD-ROM 或其他光盘存储、磁盘存储或其他磁性存储装置,或任何其他可以被用来存储所希望的程序代码的介质,该程序代码为指令或数据结构的形式并且可以被机器存取。

[0066] 为了示例硬件和软件的这种互换性,不同的示例部件、框图、模块、电路以及步骤已经在本文中就它们的功能方面而言进行了总体上进行了描述。这种功能作为硬件还是软件被实现取决于具体的应用以及总体系统强加的设计约束条件。熟练的业内人士可指向每一特定应用以不同方式实施所描述的功能,但这种实施决策不应被解释为导致偏离本披露的范围。

[0067] 这些方面和特征的描述被提供成使本领域技术人员能够制作和使用这些系统、装置以及执行所披露的方法。各种修改对本领域技术人员而言将非常明显,并且此文件中所述的原理可以在不脱离本披露的精神或范围的情况下应用到其他方面上。因此,本说明书不旨在限制权利要求书。相反,权利要求书将被给予与在此披露的原理和新颖特征一致的内容。

[0068] 关于示例过程,附图包括相对排序的过程部件,仅作为对理解本说明书的辅助。这种相对排序不是对过程部件的排序或安排的任何具体限制的隐含披露。通常,在不脱离本披露的范围的情况下,过程限制可以是顺序上互换的,并且权利要求书中的装置加功能并不旨在涵盖那些被描述为执行所列举的不仅包括结构等效物还包括等效结构的的功能的结构。

[0069] 进一步地,已经选定了部件和过程的某些细分,但根据本披露的这些实现方式并不被要求维持这些细分,以及可以有区别地分配或功能或对功能进行分组。例如,不同于、除了或少于此处识别的那些以外的部件内有区别地分配或执行这种功能性。相反,披露了所描述的特征和步骤,作为所附权利要求书的范围内的系统和方法的部件的示例。

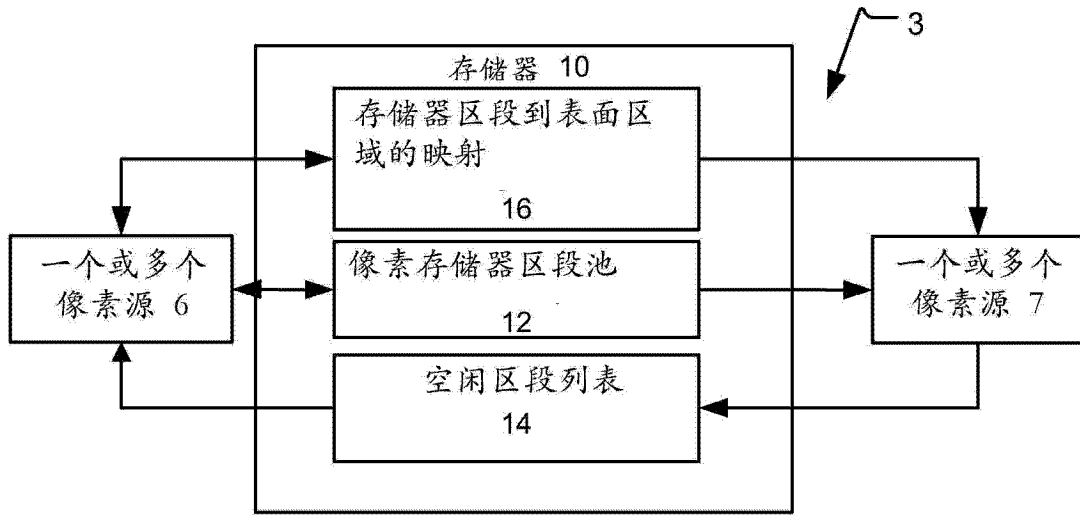


图 1

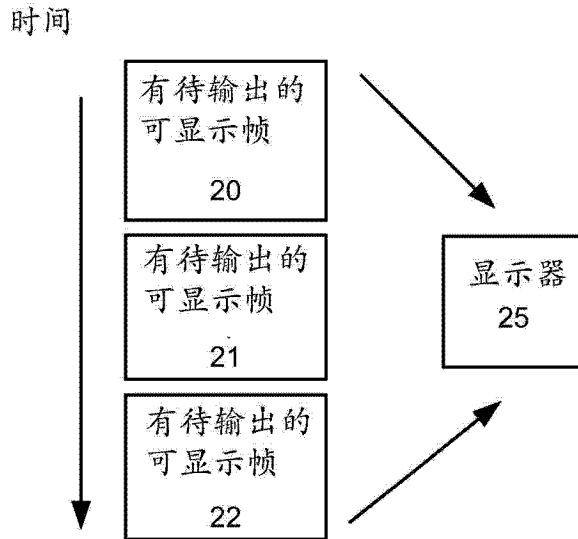


图 2

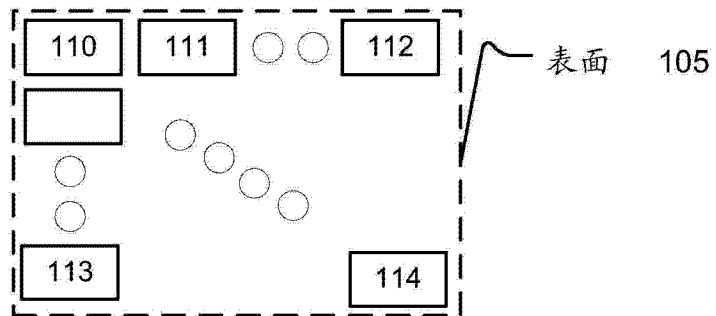


图 3

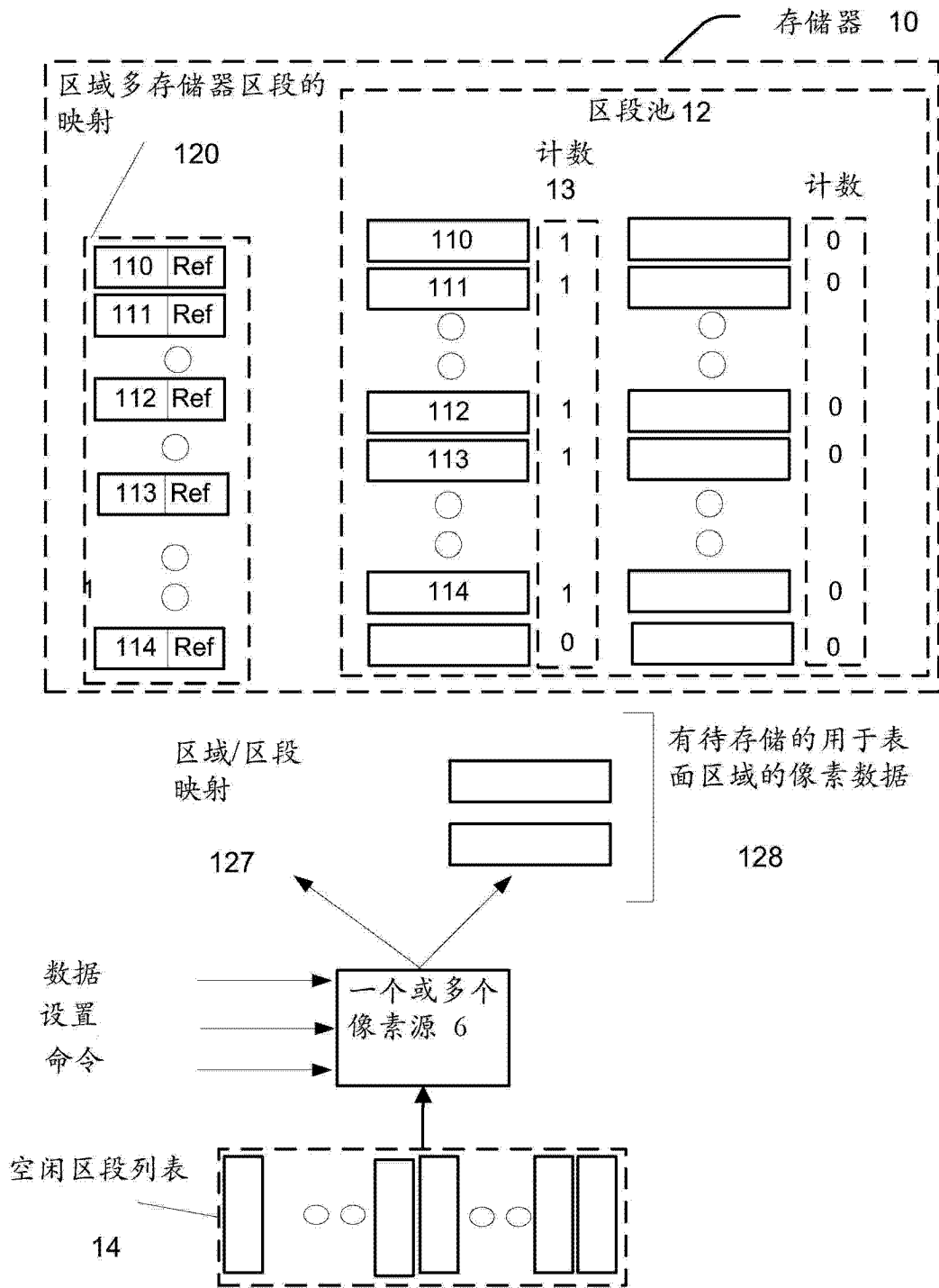


图 4

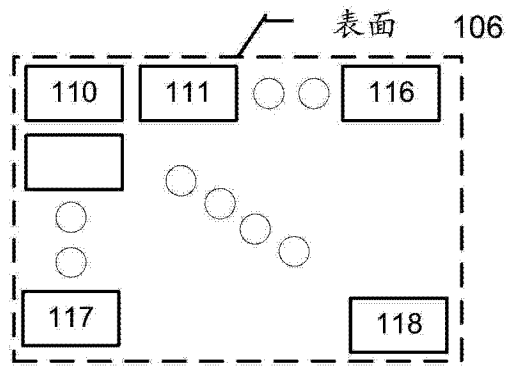


图 5

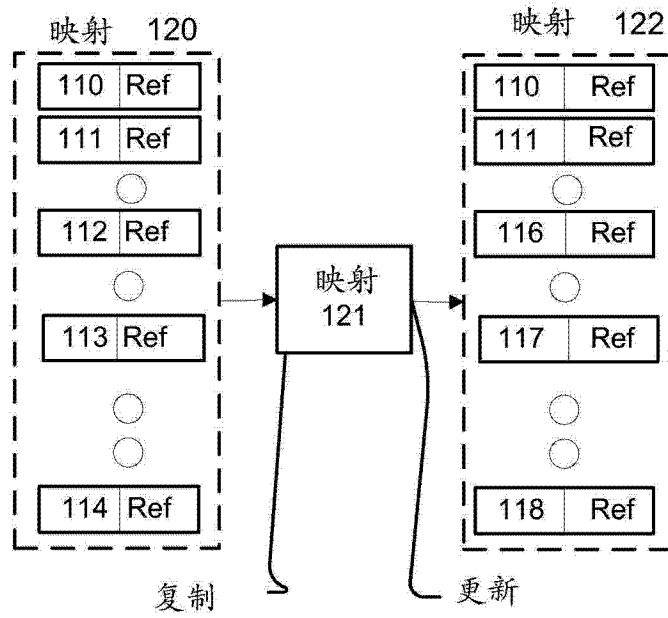


图 6

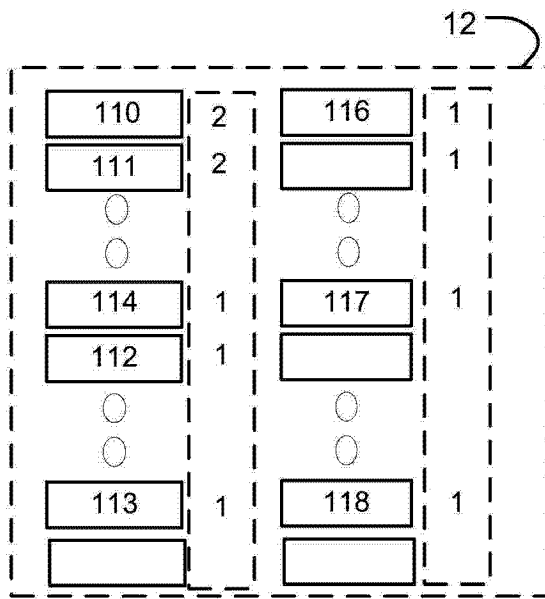


图 7A

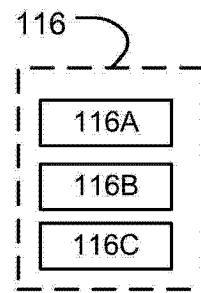


图 7B

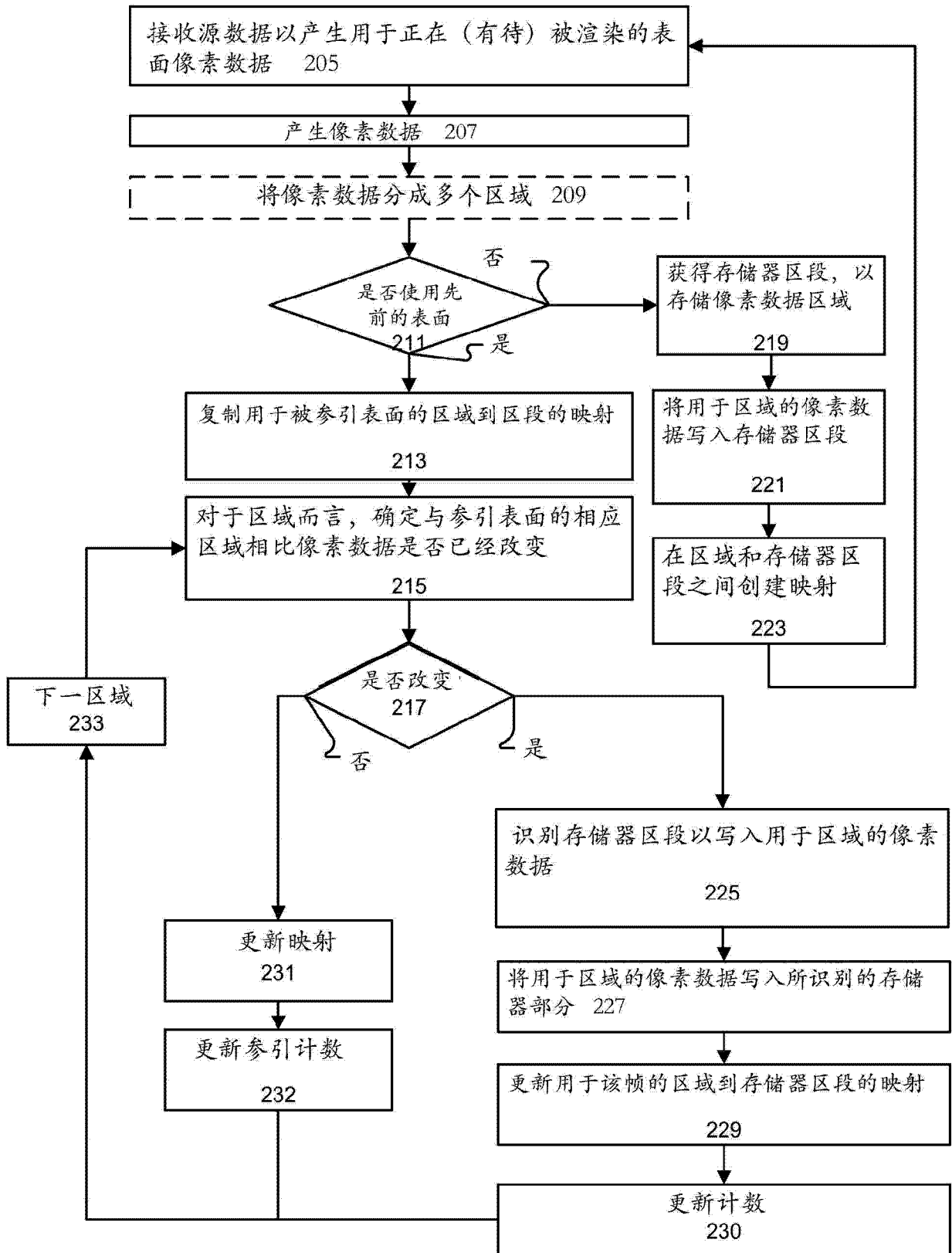


图 8

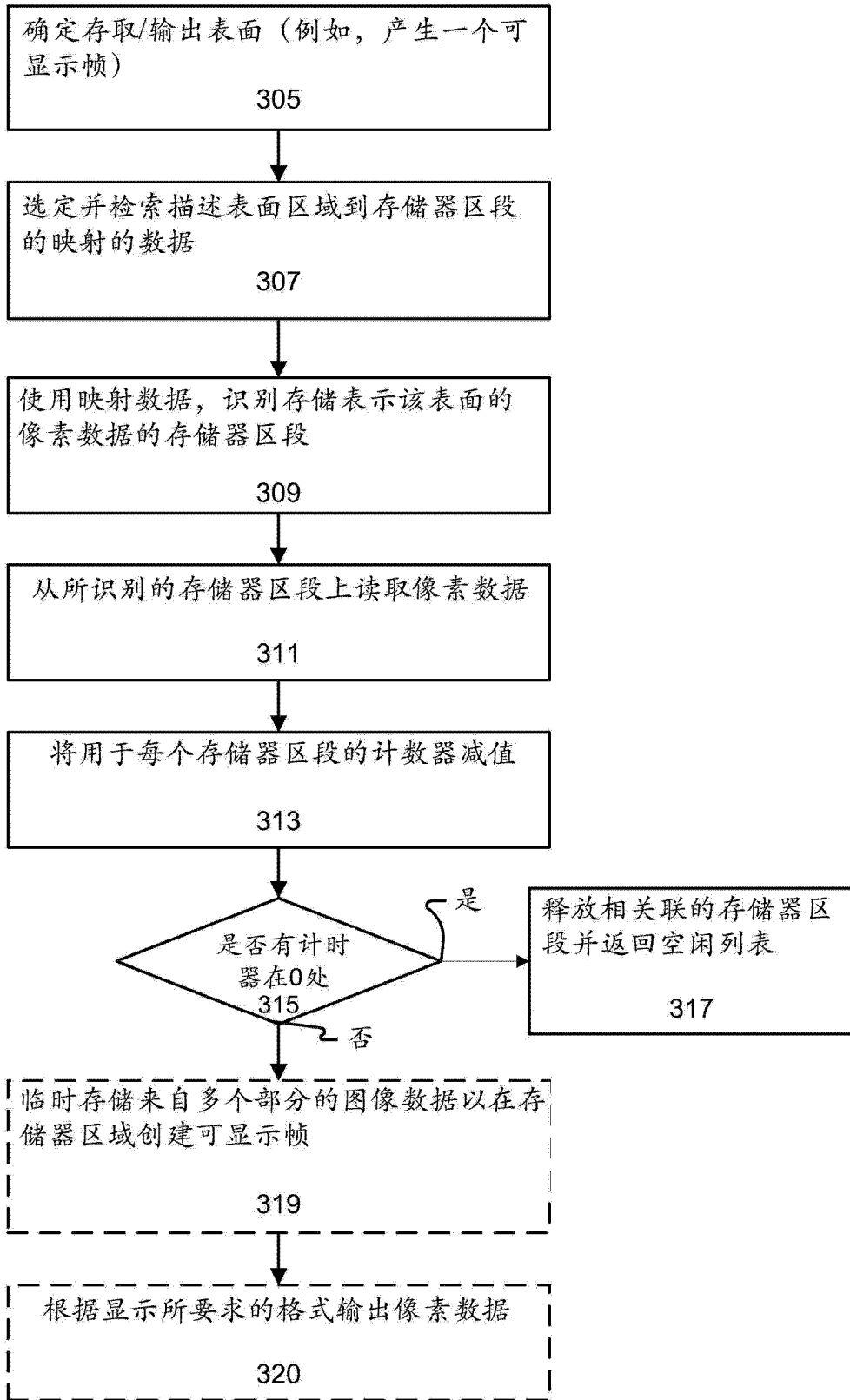


图 9

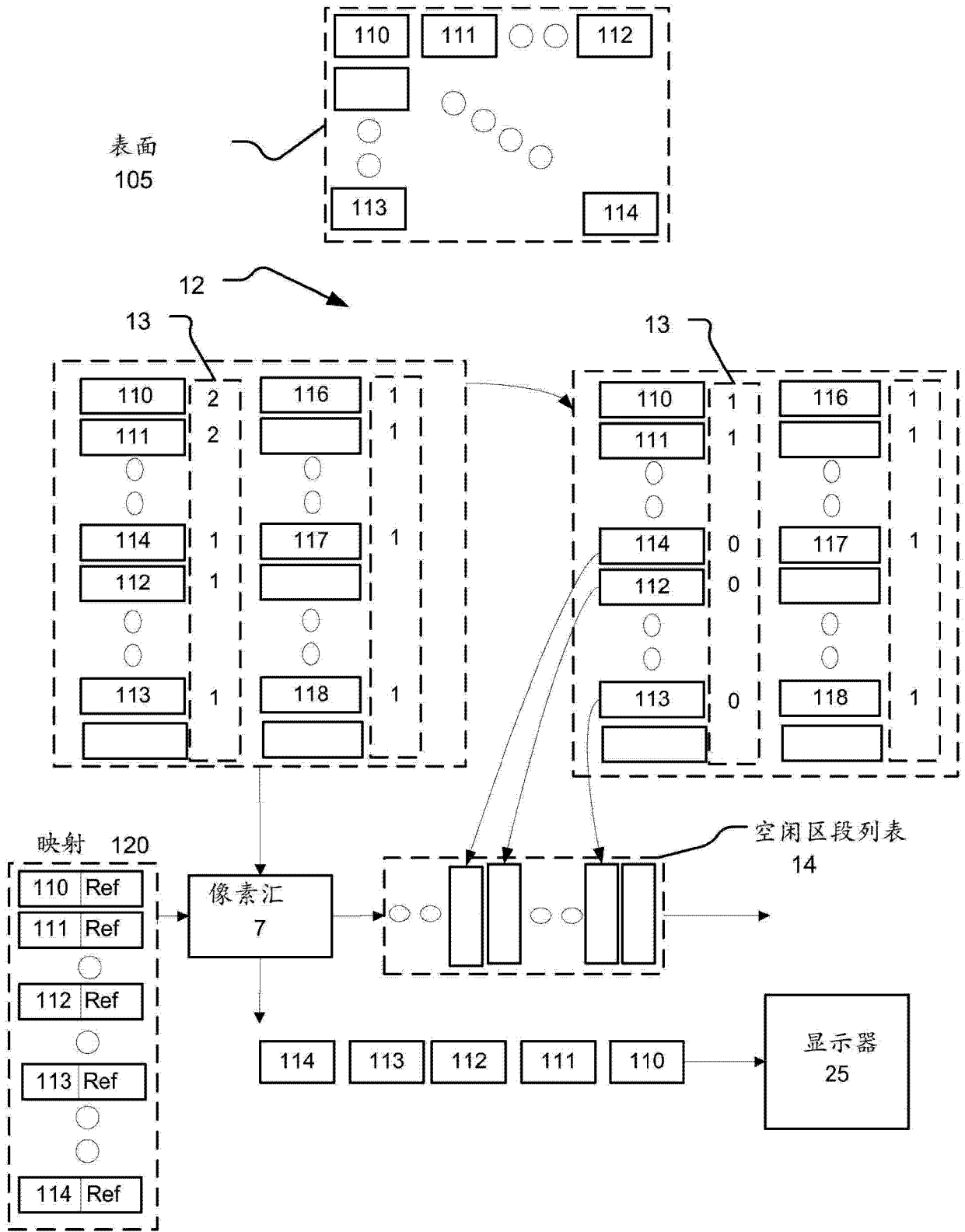


图 10

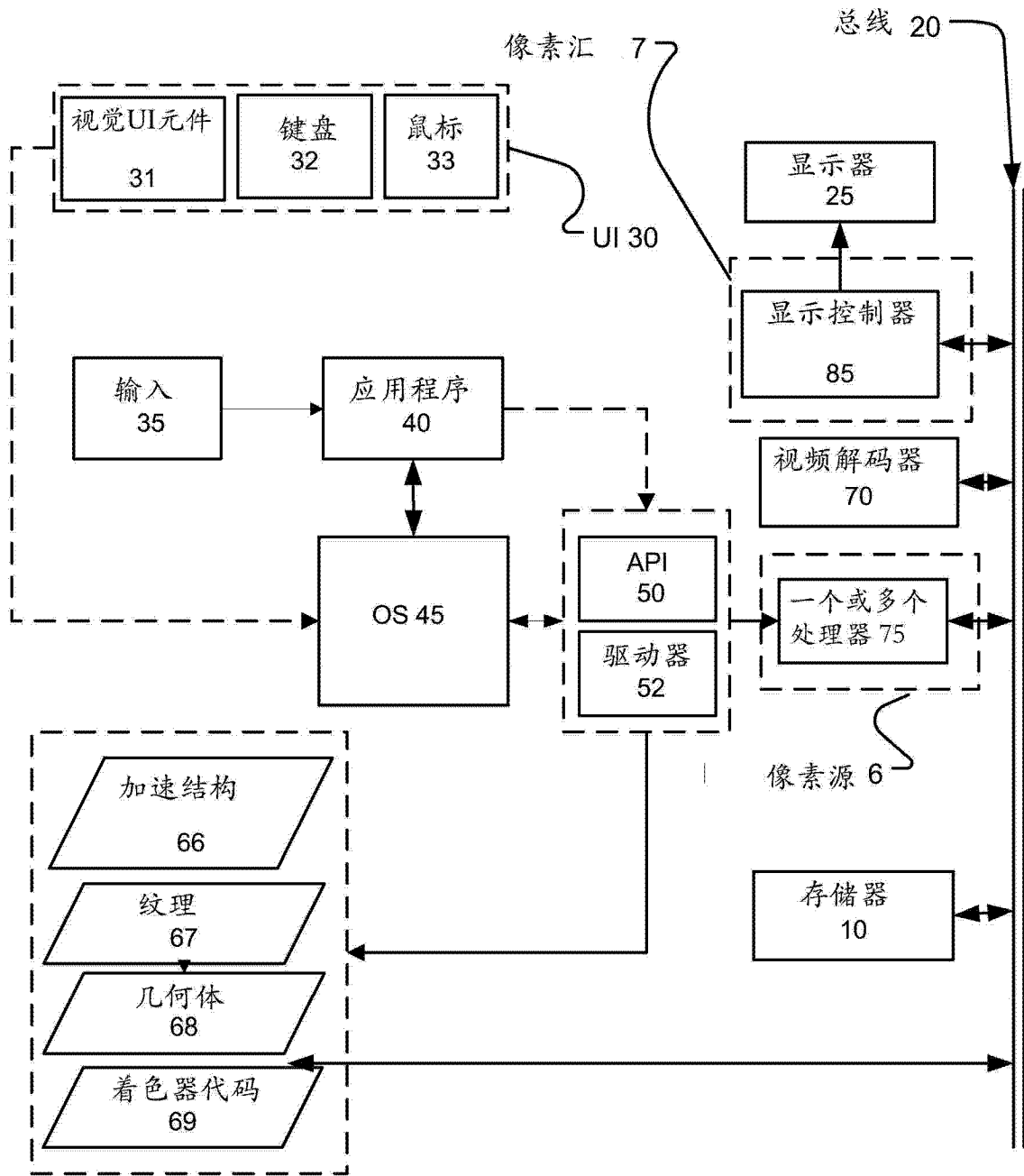


图 11

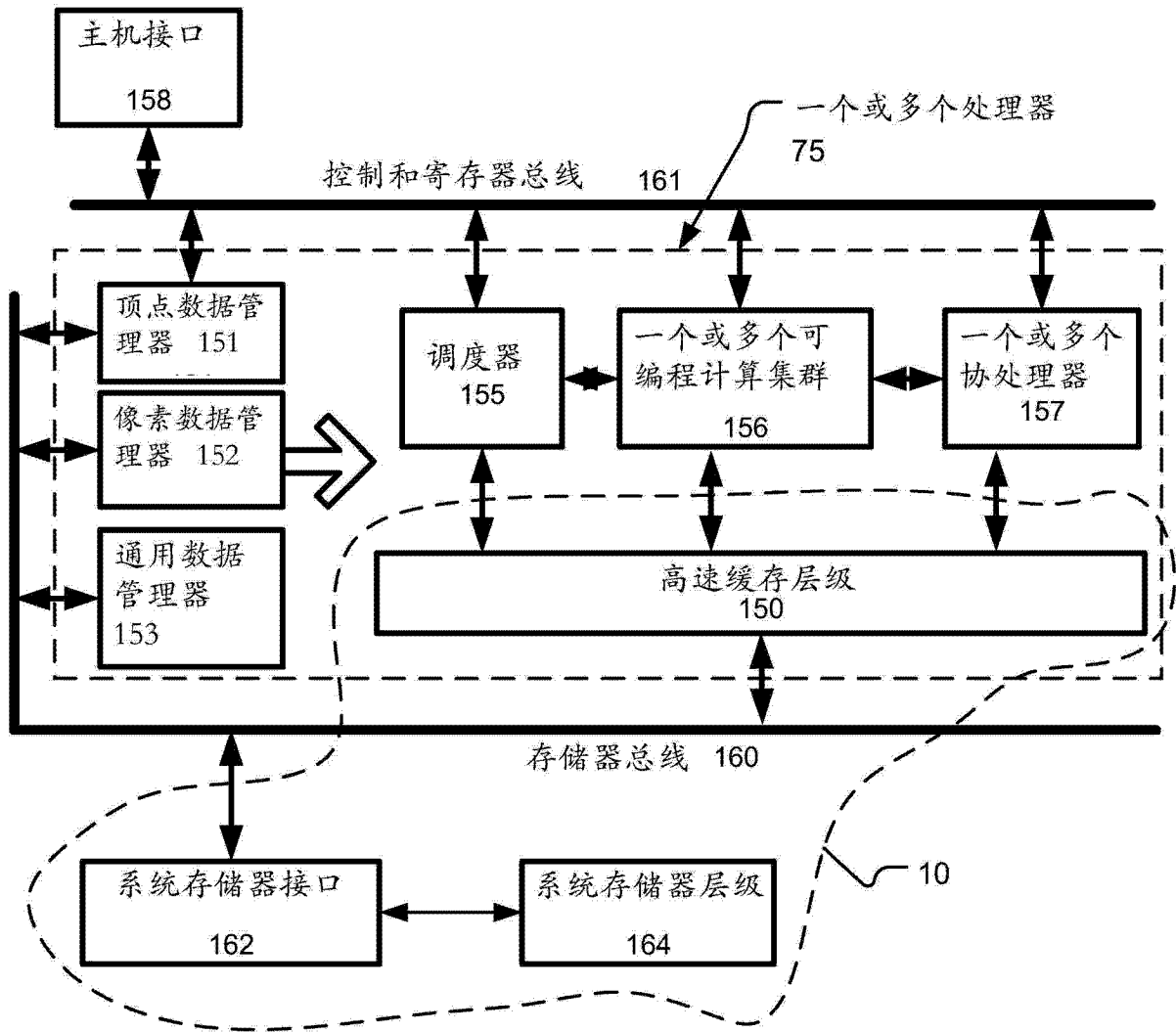


图 12