



(12) 发明专利

(10) 授权公告号 CN 101542524 B

(45) 授权公告日 2012.09.05

(21) 申请号 200680056329.3

CN 1799064 A, 2006.07.05, 全文.

(22) 申请日 2006.11.08

US 6788309 B1, 2004.09.07, 说明书第3栏
第27行至第6栏第43行、附图2-5.

(85) PCT申请进入国家阶段日

US 6788309 B1, 2004.09.07, 说明书第3栏
第27行至第6栏第43行、附图2-5.

2009.05.08

(86) PCT申请的申请数据

审查员 蒋一明

PCT/CN2006/002982 2006.11.08

(87) PCT申请的公布数据

W02008/055381 EN 2008.05.15

(73) 专利权人 英特尔公司

地址 美国加利福尼亚

(72) 发明人 W·李 E·李

(74) 专利代理机构 永新专利商标代理有限公司

72002

代理人 赵腾飞 王英

(51) Int. Cl.

G06T 1/20 (2006.01)

G06T 1/60 (2006.01)

H04N 1/393 (2006.01)

(56) 对比文件

US 2004/0174368 A1, 2004.09.09, 全文.

JP 特开 2006-165825 A, 2006.06.22, 说明
书第4页第46行至第7页第21行、附图1-7.

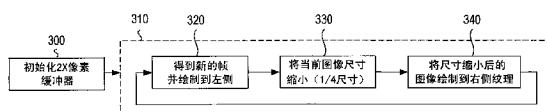
权利要求书 2 页 说明书 5 页 附图 3 页

(54) 发明名称

图形处理器的流水线化归约操作

(57) 摘要

通常，在一个方案中，本公开内容描述了一种
用于初始化纹理缓冲器以及通过利用该纹理缓冲
器对归约操作进行流水线操作的方法。



1. 一种能够对多个图像同时执行归约操作的方法,所述方法包括:

初始化纹理缓冲器,所述纹理缓冲器大于要分析的帧并且包括第一侧和第二侧,其中,所述第一侧用于存储一新的完整帧,所述第二侧用于存储所述完整帧的多个不同的尺寸缩小版本,其中,所述多个不同的尺寸缩小版本依次减小;

将一新的完整帧绘制在所述第一侧中;

对所述纹理缓冲器进行尺寸缩小以便同时对所述完整帧和所述多个不同的尺寸缩小版本执行尺寸缩小;

偏移尺寸缩小后的所述完整帧和所述多个不同的尺寸缩小版本,以便将尺寸缩小后的完整帧复制至所述第二侧中的第一个位置处,并将经过尺寸缩小后的所述第二侧中的每一个所述尺寸缩小版本复制至所述第二侧中的下一个位置处;

重复所述绘制步骤、所述尺寸缩小步骤和所述偏移步骤,其中,使所述重复流水线化,以使得所述纹理缓冲器包含多个不同帧,其中每个帧具有一不同尺寸缩小级别。

2. 如权利要求1所述的方法,其中,

所述初始化包括:初始化所述纹理缓冲器,所述纹理缓冲器的所述第二侧中的最后一个位置为一个像素,以及

所述重复使得所述第二侧中的所述最后一个位置中的内容是单个帧的归约操作的期望平均值。

3. 如权利要求1所述的方法,其中,

对所述纹理缓冲器的所述尺寸缩小将所述完整帧尺寸缩小为1/4尺寸,以及将所述多个不同的尺寸缩小版本尺寸缩小为1/4尺寸。

4. 如权利要求1所述的方法,其中,所述初始化包括:初始化所述纹理缓冲器,以使得所述纹理缓冲器的尺寸是要分析的帧的尺寸的两倍。

5. 如权利要求1所述的方法,其中,所述重复包括:在处于稳定状态中时同时对多个帧进行尺寸缩小。

6. 如权利要求2所述的方法,其中,在处于稳定状态中时,每次重复操作都会产生对于单个帧的归约操作的期望平均值。

7. 一种能够对多个图像同时执行归约操作的图形处理单元,所述图形处理单元包括:

用于初始化纹理缓冲器的第一模块,所述纹理缓冲器大于要分析的帧并且包括第一侧和第二侧,其中,所述第一侧用于存储一新的完整帧,所述第二侧用于存储所述完整帧的多个不同的尺寸缩小版本,其中,所述多个不同的尺寸缩小版本依次减小;以及

用于将一新的完整帧绘制在所述第一侧中的第二模块;

用于对所述纹理缓冲器进行尺寸缩小以便同时对所述完整帧和所述多个不同的尺寸缩小版本执行尺寸缩小的第三模块;

用于偏移尺寸缩小后的所述完整帧和所述多个不同的尺寸缩小版本,以便将尺寸缩小后的完整帧复制至所述第二侧中的第一个位置处,并将经过尺寸缩小后的所述第二侧中的每一个所述尺寸缩小版本复制至所述第二侧中的下一个位置处的第四模块;

其中,对第二模块至第四模块的操作进行重复和流水线化,以使得所述纹理缓冲器包含多个不同帧,其中每个帧具有一不同尺寸缩小级别。

8. 如权利要求7所述的图形处理单元,其中,

所述第一模块初始化所述纹理缓冲器，所述纹理缓冲器的所述第二侧中的最后一个位置为一个像素，以及

第二至第四模块的重复操作使得所述第二侧中的所述最后一个位置中的内容是单个帧的归约操作的期望平均值。

9. 如权利要求 7 所述的图形处理单元，其中，

所述第三模块包括将所述完整帧尺寸缩小为 1/4 尺寸，以及将所述多个不同的尺寸缩小版本尺寸缩小为 1/4 尺寸。

10. 如权利要求 7 所述的图形处理单元，其中，所述第一模块初始化所述纹理缓冲器，以使得所述纹理缓冲器的尺寸是要分析的帧的尺寸的两倍。

11. 如权利要求 7 所述的图形处理单元，其中，当在稳定状态中操作时，同时对多个帧进行尺寸缩小。

12. 如权利要求 8 所述的图形处理单元，其中，当在稳定状态中操作时，对于每次重复操作都会产生对单个帧的归约操作的期望平均值。

图形处理器的流水线化归约操作

背景技术

[0001] 图形处理单元 (GPU) 是可以用在例如个人计算机、工作站或游戏主机中的专用图形渲染设备。GPU 在操作和显示计算机图形方面非常有效率。GPU 包含多个处理单元，这些处理单元同时执行独立的操作（例如，在像素级上的颜色空间转换）。它们的高并行化结构可以使它们在复杂算法范围内比普通的中央处理单元 (CPU) 效率高。GPU 可以采用使图形基元操作运行的比使用主 CPU 直接在屏幕上绘制快得多的方式，来实现多个图形基元。

[0002] GPU 上的通用编程正在成为一种用于加速计算的高效且普遍的方式，并且结合 CPU 充当重要的计算单元。在实践中，可以对庞大数量的现有通用处理内核（例如纹理处理、矩阵和矢量计算）进行优化以便运行在 GPU 上。然而，GPU 具有某些硬件约束和结构限制。例如，GPU 不具有全局变量的概念，并且不能使用几个全局变量来在运行中保存临时数据。因此，GPU 不能有效地处理某些通常使用的归约 (reduction) 操作（例如，对一串数据元素进行平均和求和计算）。

发明内容

[0003] 根据一个方面，本发明公开了一种能够对多个图像同时执行归约操作的方法，所述方法包括：初始化纹理缓冲器，所述纹理缓冲器大于要分析的帧并且包括第一侧和第二侧，其中，所述第一侧用于存储一新的完整帧，所述第二侧用于存储所述完整帧的多个不同的尺寸缩小版本，其中，所述多个不同的尺寸缩小版本依次减小；将一新的完整帧绘制在所述第一侧中；

[0004] 对所述纹理缓冲器进行尺寸缩小以便同时对所述完整帧和所述多个不同的尺寸缩小版本执行尺寸缩小；偏移尺寸缩小后的所述完整帧和所述多个不同的尺寸缩小版本，以便将尺寸缩小后的完整帧复制至所述第二侧中的第一个位置处，并将经过尺寸缩小后的所述第二侧中的每一个所述尺寸缩小版本复制至所述第二侧中的下一个位置处；重复所述绘制步骤、所述尺寸缩小步骤和所述偏移步骤，其中，使所述重复流水线化，以使得所述纹理缓冲器包含多个不同帧，其中每个帧具有一不同尺寸缩小级别。

[0005] 根据另一方面，本发明公开了一种能够对多个图像同时执行归约操作的图形处理单元，所述图形处理单元包括：用于初始化纹理缓冲器的第一模块，所述纹理缓冲器大于要分析的帧并且包括第一侧和第二侧，其中，所述第一侧用于存储一新的完整帧，所述第二侧用于存储所述完整帧的多个不同的尺寸缩小版本，其中，所述多个不同的尺寸缩小版本依次减小；以及用于将一新的完整帧绘制在所述第一侧中的第二模块；用于对所述纹理缓冲器进行尺寸缩小以便同时对所述完整帧和所述多个不同的尺寸缩小版本执行尺寸缩小的第三模块；用于偏移尺寸缩小后的所述完整帧和所述多个不同的尺寸缩小版本，以便将尺寸缩小后的完整帧复制至所述第二侧中的第一个位置处，并将经过尺寸缩小后的所述第二侧中的每一个所述尺寸缩小版本复制至所述第二侧中的下一个位置处的第四模块；其中，对第二模块至第四模块的操作进行重复和流水线化，以使得所述纹理缓冲器包含多个不同帧，其中每个帧具有一不同尺寸缩小级别。

[0006] 根据另一方面，本发明公开了一种系统，所述系统包括：中央处理单元 (CPU)；图形处理单元 (GPU)，包括：用于初始化纹理缓冲器的第一模块，所述纹理缓冲器大于要分析的帧并且包括第一侧和第二侧，其中，所述第一侧用于存储一新的完整帧，所述第二侧用于存储所述完整帧的多个不同的尺寸缩小版本，其中，所述多个不同的尺寸缩小版本依次减小；以及用于将一新的完整帧绘制在所述第一侧中的第二模块；用于对所述纹理缓冲器进行尺寸缩小以便同时对所述完整帧和所述多个不同的尺寸缩小版本执行尺寸缩小的第三模块；用于偏移尺寸缩小后的所述完整帧和所述多个不同的尺寸缩小版本，以便将尺寸缩小后的完整帧复制至所述第二侧中的第一个位置处，并将经过尺寸缩小后的所述第二侧中的每一个所述尺寸缩小版本复制至所述第二侧中的下一个位置处的第四模块；其中，对第二模块至第四模块的操作进行重复和流水线化，以使得所述纹理缓冲器包含多个不同帧，其中每个帧具有一不同尺寸缩小级别。

附图说明

- [0007] 根据以下详细描述，各个实施例的特征和优点将会变得显而易见，其中：
- [0008] 图 1 示出根据一个实施例的实例视频挖掘应用；
- [0009] 图 2A 示出根据一个实施例的用于归约操作的实例滤波器循环方法的流程图；
- [0010] 图 2B 示出根据一个实施例的滤波器循环方法的实例应用；
- [0011] 图 3A 示出根据一个实施例的用于归约操作的实例流水线纹理方法的流程图；以及
- [0012] 图 3B 示出根据一个实施例的流水线纹理循环方法的实例应用。

具体实施方式

[0013] 图 1 示出了一个实例视频挖掘应用 100。该视频挖掘应用 100 包括特征提取 110 和拍摄图片 (shot) 边界检测 120。特征提取 110 包括几个归约操作。这些归约操作可以包括以下的至少一些子集：(1) 确定一帧内所有像素的平均灰度值；(2) 提取一个指定区域内的像素的黑值和白值；(3) 计算一帧的每个颜色通道的 RGB 直方图；以及 (4) 计算两个连续帧之间的平均灰度差值。

[0014] 可以通过以分层形式将一个帧（例如 720x576 个像素）尺寸缩小为更小的块并且提取这些块的特征，来执行特征提取。最终将该帧归约为 1 个像素，并且该像素是该帧的平均值。

[0015] 一种块形成方法可以使用大滤波器将帧的尺寸缩小为大块。例如，可以使用 36x32 滤波器着色器 (shader) 将 720x576 个像素的帧的尺寸缩小为 20x18 个像素的图像，随后可以使用 20x18 的滤波器着色器将该 20x18 个像素的图像的尺寸缩小为单个像素。该方法仅需要几个步骤（例如，两个尺寸缩小事件），但是这些步骤需要大量的内存存取时间。在具有很多独立处理单元（流水线）的 GPU 上使用这种方法会造成一个流水线过分占用对存储器的存取，并造成其他流水线停顿。

[0016] 滤波器循环方法将一个滤波器着色器使用几次，从而连续地将一个帧的尺寸缩小到特定量，并且随后一旦该图像被归约到特定尺寸，就使用第二个滤波器着色器将其归约到 1 个像素。该方法对每个帧使用多个像素缓冲器来保存各种尺寸缩小的版本。例如，可

以使用 2×2 滤波器将图像进行 $1/4$ 缩小 5 次，因此将图像归约为 22×18 个像素。然后可以使用 22×18 滤波器着色器将该 22×18 个像素的图像的尺寸缩小为 1 个像素。这个实例会需要 5 个像素缓冲器来存储各种尺寸缩小的版本（每个进行了 $1/4$ 尺寸缩小后的图像）。

[0017] 图 2A 示出用于归约操作的一个实例滤波器循环方法的流程图。初始化像素缓冲器（例如，5 个）(200)。将该帧提供给尺寸缩小循环 (210)。该循环 210 通过使用 2×2 滤波器将当前图像的尺寸缩小为 $1/4$ 尺寸 (220)。将缩小尺寸后的图像绘制到下一个像素缓冲器 (230)。在当前图像为 22×18 个像素（例如，在经过 5 次 $1/4$ 尺寸缩小之后），该循环 210 结束。然后使用 22×18 滤波器着色器将该 22×18 个像素的图像归约为 1 个像素。这一个像素的结果就是该帧的期望平均值。

[0018] 该滤波器循环方法对于每个帧都需要 6 次尺寸缩小操作，并且需要 5 个额外的像素缓冲器来捕获尺寸缩小后的图像。在该滤波器循环方法中的关键路径是使用 22×18 滤波器着色器的尺寸缩小。

[0019] 图 2B 示出了该滤波器循环方法（例如 210）的应用。当接收到该帧时，将其存储在缓冲器 250 中。将该帧的尺寸缩小到 $1/4$ ，然后存储在下一个缓冲器 260 中。然后对尺寸缩小后的图像再进行尺寸缩小，并依次存储在缓冲器 265-280 中。存储在缓冲器 280 的图像是 22×18 个像素的图像，其尺寸被 22×18 滤波器着色器缩小到 1 个像素。

[0020] 流水线纹理方法将各个帧的尺寸缩小操作重叠在一起，从而增加了并行性。可以将多个滤波步骤合并为单个滤波步骤，从而可以将多个（例如 8 个）具有不同尺寸的连续的帧一起进行尺寸缩小。最初，对比一个帧大（例如 2 倍）的纹理缓冲器进行初始化。该纹理缓冲器可以包含两侧，第一侧用于存储新的帧，第二侧用于存储尺寸缩小后的帧。将一个帧读取到该纹理缓冲器中。然后对该纹理缓冲器进行尺寸缩小，并且偏移尺寸缩小后的图像。该过程持续进行，从而使滤波操作可以一次对多个帧进行尺寸缩小。一旦该操作处于稳定状态（该纹理缓冲器是满的），则执行单个尺寸缩小就足以获得最终结果。

[0021] 图 3A 示出了用于归约操作的实例流水线纹理方法的流程图。初始化 $2X$ 的像素缓冲器 (300)。然后初始化流水线滤波器操作 (310)。该流水线滤波器操作 310 包括读取一个新的帧，并将其绘制到纹理的左侧 (320)。然后使用 2×2 滤波器将纹理中的图像尺寸缩小到 $1/4$ (330)，并将所有尺寸缩小后的图像绘制到纹理的右侧 (340)。事实上，纹理中的每个图像都被尺寸缩小，然后向右侧偏移。原始帧（例如 720×576 个像素）被尺寸缩小到 $1/4$ （例如 360×288 个像素）并绘制到纹理缓冲器右侧上的第一个位置处。纹理的右侧上的图像被尺寸缩小，然后再次绘制到右侧上的下一个位置处。

[0022] 持续地重复该尺寸缩小处理。每次尺寸缩小操作都对一个新的帧以及在纹理缓冲器右侧中的尺寸缩小后的图像一起进行尺寸缩小。需要 7 次 $1/4$ 尺寸缩小操作来将一个帧归约为单个像素。纹理缓冲器的右侧能够保存 7 个尺寸缩小后的图像。当纹理缓冲器是满的时，其一共保存了各种尺寸缩小版本的 8 个帧，并对这些图像中的各图像同时进行尺寸缩小。在第 7 次尺寸缩小操作之后的值后一个像素，并且表示该帧的期望平均值。

[0023] 图 3B 示出了流水线纹理循环方法（例如 310）的应用。纹理缓冲器 350 的尺寸是一个帧尺寸的两倍（例如， 1440×576 个像素），并且包括左侧 355 和右侧 360。左侧 355 存储新的帧（帧 N）365。右侧 360 存储 7 个先前帧的尺寸缩小版本。先前帧（帧 N-1）370 是 $1/4$ 尺寸的并且被存储在右侧 360 上的第一个槽位 (slot) 中。帧 N-2375 到帧 N-7397 存

储在右侧的连续槽位中，并且每一个槽位都是前一个槽位的 1/4 尺寸。应该注意，帧 N-6 和 N-7 尺寸非常小，以至于不能清楚地观看它们，因此为了简单起见，将它们一起进行标记。

[0024] 当执行尺寸缩小操作时，其将 7 个图像（7 个不同帧的不同阶段）一起进行归约，并且将图像重新绘制在右侧，然后在左侧绘制一个新的帧。因此，当流水线纹理方法处于稳定状态中时，每次简约操作都能够得到一个帧的结果。

[0025] 在 GPU 上使用流水线纹理方法能够同时处理多个帧的多个计算。在没有复杂的编程和 SIMD 优化的情况下，这种处理是不能在 CPU 上执行的。

[0026] 以上所述的纹理缓冲器的实施例讨论了在长度上是一个帧的两倍的缓冲器（例如，720x576 对比 1440x576），但是并不局限于此。相反，缓冲器可以在高度上延长（例如 720x1152），而不会脱离本发明的范围。此外，实施例示出了将一个新的帧绘制到左侧，而将尺寸缩小后的帧绘制到右侧，但是并不局限于此。相反，可以将新的帧绘制到右侧，而将尺寸缩小后的帧绘制到左侧，可以将新的帧绘制到上部，而将尺寸缩小后的帧绘制到下部，或者可以将新的帧绘制到下部，而将尺寸缩小后的帧绘制到上部，而不会脱离本发明的范围。简单的事实是可以对缓冲器中的所有内容执行一次尺寸缩小操作，然后将尺寸缩小后的图像重新绘制在缓冲器中的下一个位置。

[0027] 虽然本公开内容已经参考具体实施例进行了陈述，但是显而易见的是，本公开内容并不局限于此，因为在不脱离本发明的范围的情况下可以对具体实施例继续各种改变和修改。对“一个实施例”或“实施例”的指代意思是在此所述的特定的特征、结构或特点包含在至少一个实施例中。因此，在通篇说明书中的各个位置出现的短语“在一个实施例中”或“在实施例中”并非必然指代同一实施例。

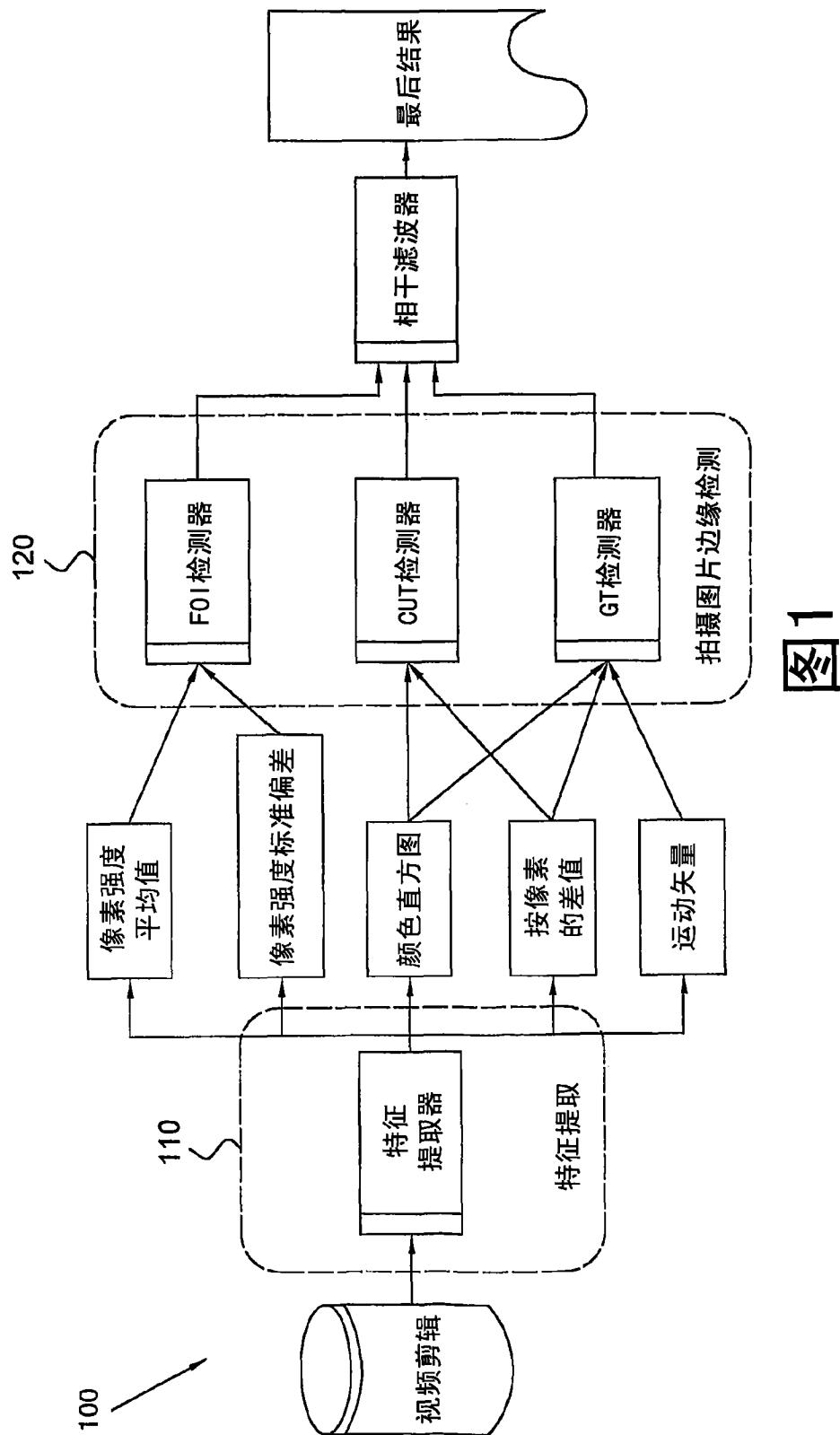
[0028] 实施例可以用硬件、软件、固件、微代码或者其任意组合来实现。当用软件、固件或微代码实现时，实施例的要素是用于执行必要任务的程序代码或代码段。代码可以是执行操作的实际代码，或者是对操作进行模拟或仿真的代码。代码段可以表示过程、函数、子程序、程序、例程、子例程、模块、软件包、类、或者指令、数据结构或程序声明的任意组合。代码段可以通过传递和 / 或接收信息、数据、自变量、参数或存储器内容而耦合到另一代码段或硬件电路。信息、自变量、参数、数据等可以借助于任何手段，包括存储器共享、消息传递、令牌传递、网络传输等等，进行传递、转发或发送。程序或代码段可以存储在处理器可读介质中，或者由体现在载波中的计算机数据信号或由载波调制的信号而通过传输介质进行发送。“处理器可读或可存取介质”或“机器可读或可存取介质”可以包括任何能够存储、发送或传递信息的介质。处理器 / 机器可读 / 可存取介质的实例包括电子电路、半导体存储器件、只读存储器 (ROM)、闪存存储器、可擦除 ROM (EROM)、软盘、致密盘 (CD-ROM)、光盘、硬盘、光纤介质、无线电频率 (RF) 链路等等。计算机数据信号可以包括能够在诸如电子网络信道、光纤、空中、电磁、RF 链路等等之类的传输介质上传播的任何信号。代码段可以经由诸如互联网、内联网等等之类的计算机网络进行下载。机器可存取介质可以体现为制造品。机器可存取介质可以包括当由机器存取时使该机器执行以下所述的操作的数据。术语“数据”在此指代能够为了机器可读的目的而被编码的任何类型的信息。因此，其可以包括程序、代码、数据、文件等等。

[0029] 实施例的一部分或者全部可以用软件来实现。软件可以具有彼此耦合的几个模块。一个软件模块耦合到另一模块，以接收变量、参数、自变量、指针等等，和 / 或产生或传

递结果、更新的变量、指针等等。软件模块还可以是与运行在平台上的操作系统交互的软件驱动器或接口。软件模块还可以是用于配置、设置、初始化数据以及将数据发送到硬件设备或从硬件设备接收数据的硬件驱动器。

[0030] 实施例可以描述为一个处理，处理通常被绘制为流程图、流程图表、结构图或框图。虽然流程图可以将操作描述为顺序处理，但是其中的很多操作可以并行或同时执行。另外，可以重新排列操作的次序。当一个处理的操作完成时，该处理可以结束。处理可以对应于方法、函数、过程、子例程、子程序等等。当处理对应于函数时，该处理的结束可以对应于该函数返回到调用函数或主函数。

[0031] 各个实施例旨在受到所附权利要求的精神和范围内的最宽泛的保护。



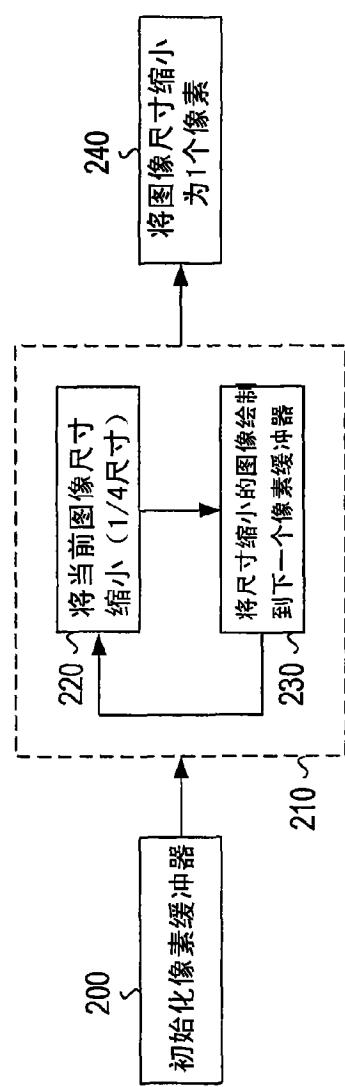


图 2A

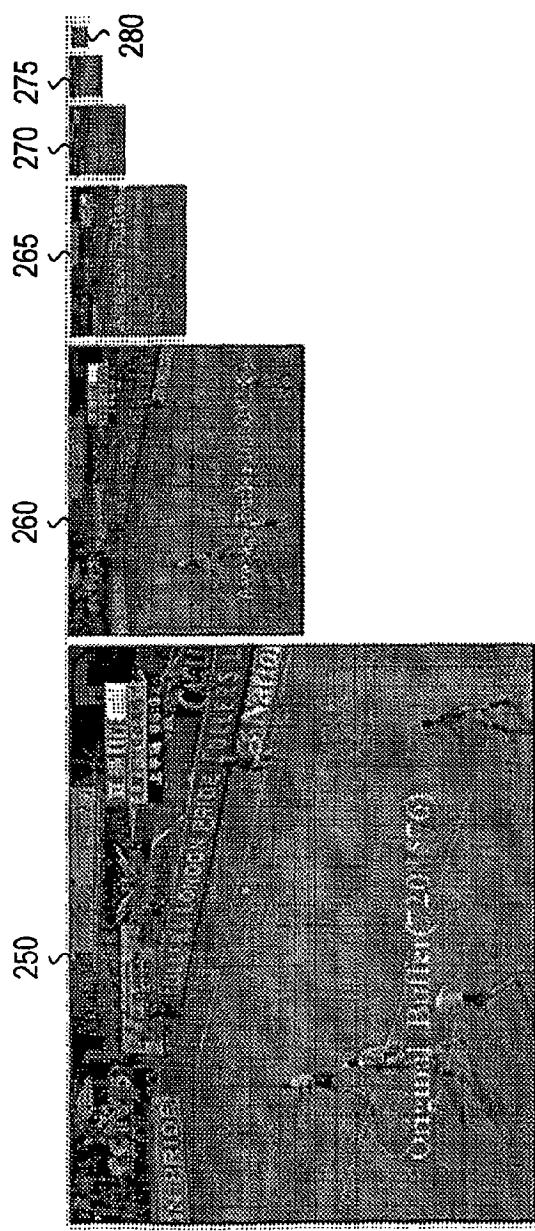


图 2B

