



(12)发明专利申请

(10)申请公布号 CN 109727187 A

(43)申请公布日 2019.05.07

(21)申请号 201910005972.2

(22)申请日 2019.01.03

(71)申请人 北京地平线机器人技术研发有限公司

地址 100080 北京市海淀区中关村大街1号
3层318

(72)发明人 李军 李建军 张鑫语 李晓森
黄畅

(74)专利代理机构 北京市正见永申律师事务所
11497

代理人 黄小临 冯玉清

(51)Int.Cl.

G06T 1/60(2006.01)

G06K 9/00(2006.01)

G06K 9/32(2006.01)

权利要求书2页 说明书11页 附图3页

(54)发明名称

用于调整多个感兴趣区域数据的存储位置的方法和装置

(57)摘要

公开了一种用于调整多个感兴趣区域数据的存储位置的方法和装置。该方法可以包括：根据要调整的多个感兴趣区域数据的得分，确定与存储要调整的多个感兴趣区域数据的存储区域的地址范围相对应的多个局部地址范围；针对作为当前要调整的感兴趣区域数据的第一数据，确定要调整的多个局部地址范围中的第一局部地址范围内的第一地址；将第一地址处的第二数据从存储区域存储到缓冲存储器中；以及将第一数据存储在存储区域中的第一地址处。通过根据本公开的实施例的方法和装置，能够高效率地实现多个感兴趣区域数据的存储位置的就地调整。



1. 一种用于调整多个感兴趣区域数据的存储位置的方法,包括:

根据所述多个感兴趣区域数据的得分,确定与存储所述多个感兴趣区域数据的存储区域的地址范围相对应的多个局部地址范围;

针对作为当前要调整的感兴趣区域数据的第一数据,确定所述多个局部地址范围中的第一局部地址范围内的第一地址;

将所述第一地址处的第二数据从所述存储区域存储到缓冲存储器中;以及

将所述第一数据存储到所述存储区域中的第一地址处。

2. 根据权利要求1所述的方法,其中,所述多个局部地址范围中的每个局部地址范围的大小取决于所述多个感兴趣区域数据中具有与该局部地址范围相对应的得分的感兴趣区域数据的数量。

3. 根据权利要求1所述的方法,其中,所述第一数据具有与所述第一局部地址范围相对应的得分,并且所述第一地址的第二数据不是已调整的感兴趣区域数据。

4. 根据权利要求3所述的方法,其中,确定所述多个局部地址范围中的第一局部地址范围内的第一地址包括:

根据与所述第一局部地址范围相对应的计数器的计数值确定所述第一地址。

5. 根据权利要求4所述的方法,还包括:

更新与所述第一局部地址范围相对应的计数器的计数值,使得更新后的计数值对应于所述第一局部地址范围内的未被访问过的地址。

6. 根据权利要求1所述的方法,其中,将所述第一地址处的第二数据从所述存储区域存储到缓冲存储器中包括:

将缓存在所述缓冲存储器的第一缓冲区中的第一数据存储到所述缓冲存储器的第二缓冲区中;以及

将所述第二数据从所述存储区域缓存到所述缓冲存储器的第一缓冲区中。

7. 根据权利要求6所述的方法,其中,将所述缓冲存储器中的第一数据存储到所述存储区域中的第一地址处包括:

将所述第二缓冲区中的第一数据存储到所述存储区域中的第一地址处。

8. 根据权利要求1至7中的任一项所述的方法,还包括:

在所述第二数据是未调整的感兴趣区域数据的情况下,确定所述缓冲存储器中的第二数据作为下一个要调整的感兴趣区域数据。

9. 根据权利要求1至7中的任一项所述的方法,还包括:

在所述第二数据不是未调整的感兴趣区域数据并且所述存储区域中的已调整的感兴趣区域数据的数量小于所述多个感兴趣区域数据的数量的情况下,确定所述存储区域中的未被访问过的第二地址处的数据作为下一个要调整的感兴趣区域数据。

10. 根据权利要求9所述的方法,其中,确定所述存储区域中的未被访问过的第二地址处的数据作为下一个要调整的感兴趣区域数据包括:

确定所述多个局部地址范围中的第二局部地址范围,所述第二局部地址范围内包括至少一个未被访问过的地址;以及

根据与所述第二局部地址范围相对应的计数器中的计数值确定所述第二地址。

11. 一种计算机可读的非临时性存储介质,在其上存储有程序指令,所述程序指令在

被执行时至少执行根据权利要求1至10中的任一项所述的方法。

12. 一种用于调整多个感兴趣区域数据的存储位置的装置, 包括:

缓冲存储器, 被配置为存储当前要调整的感兴趣区域数据; 以及

处理器, 被配置为在启动时至少执行根据权利要求1至10中的任一项所述的方法。

13. 根据权利要求12所述的方法, 其中, 所述缓冲存储器包括两个缓冲区, 每个缓冲区的容量取决于单个感兴趣区域数据的大小。

14. 根据权利要求12或13所述的装置, 还包括:

多个计数器, 每个计数器分别与一个局部地址范围相对应, 所述多个计数器的数量取决于每个感兴趣区域数据的得分的精度。

用于调整多个感兴趣区域数据的存储位置的方法和装置

技术领域

[0001] 本公开总体上涉及人工智能的技术领域,并且具体地涉及一种用于调整多个感兴趣区域数据的存储位置的方法和装置。

背景技术

[0002] 可以采用诸如RCNN (Regions with CNN features)、空间金字塔池化网络 (Spatial Pyramid Pooling Network, SPP-Net)、快速RCNN (Fast RCNN)、更快速RCNN (Faster RCNN) 等方法检测输入图像或视频中的一个或多个关注对象,而这些对象检测方法需要先生成(或推荐)多个感兴趣区域 (Region of Interest, ROI),然后基于所生成的ROI数据执行后续处理。

[0003] 在基于ROI的对象检测方法中,需要根据ROI的得分重新调整所生成的各个ROI数据的存储位置。

[0004] 通常的用于控制调整ROI数据的位置的过程的时空复杂度均很高,不适合在硬件资源有限的人工智能芯片上高效率地实现。

发明内容

[0005] 根据本公开的一个方面,提供了一种用于调整多个感兴趣区域数据的存储位置的方法和装置。该方法可以包括:根据要调整的多个感兴趣区域数据的得分,确定与存储要调整的多个感兴趣区域数据的存储区域的地址范围相对应的多个局部地址范围;针对作为当前要调整的感兴趣区域数据的第一数据,确定要调整的多个局部地址范围中的第一局部地址范围内的第一地址;将第一地址处的第二数据从存储区域存储到缓冲存储器中;以及将第一数据存储到存储区域中的第一地址处。

[0006] 根据本公开的另一个方面,还提供了一种用于调整多个感兴趣区域数据的存储位置的装置。该装置可以包括缓冲存储器和处理器,其中,缓冲存储器可以被配置为存储当前要调整的感兴趣区域数据,并且处理器可以被配置为在启动时至少执行上述方法。

[0007] 根据本公开的另一个方面,还提供了一种计算机可读的非临时性存储介质,在其上存储有程序指令,该程序指令在被执行时至少执行上述方法。

[0008] 通过根据本公开的实施例的方法和装置,能够高效率地实现多个感兴趣区域数据的位置的就地调整。

附图说明

[0009] 通过结合附图对本公开实施例进行更详细的描述,本公开的上述以及其他目的、特征和优势将变得更加明显。附图用来提供对本公开实施例的进一步理解,并且构成说明书的一部分,与本公开实施例一起用于解释本公开,并不构成对本公开的限制。在附图中,相同的参考标号通常代表相同部件或步骤。

[0010] 图1示出根据本公开的实施例的存储着N个ROI数据的存储区域SR的示例。

- [0011] 图2示出根据本公开的实施例的用于调整ROI数据的存储位置的方法的示例。
- [0012] 图3示出在对于如图1所示的N个ROI数据执行图2的示例方法的过程的示例。
- [0013] 图4示出执行图2的示例方法中的步骤S130和步骤S140的示例。
- [0014] 图5示出在对于如图1所示的N个ROI数据执行图2的示例方法的过程的示例。
- [0015] 图6示出根据本公开的实施例的用于调整ROI数据的存储位置的装置的示例。

具体实施方式

[0016] 下面,将参考附图详细地描述根据本公开的示例实施例。显然,所描述的实施例仅仅是本公开的一部分实施例,而不是本公开的全部实施例,应理解,本公开不受这里描述的示例实施例的限制。

[0017] 概述

[0018] 在诸如RCNN、SPP-Net、Fast RCNN和Faster RCNN这样的检测效果优良的多阶段对象检测方法中,通常需要对所产生大量(例如,数千个、数万个、甚至数十万个)的ROI进行排序,即,调整ROI数据在存储器中的存储位置。

[0019] 在通常的用于控制调整ROI数据的存储位置的方案中,通常需要所配置的存储器能够提供大于或等于ROI数据的总数据量的两倍的存储空间,其中,存储器可以包括前述的缓冲存储器和/或用于存储最初的ROI数据和最终的排序结果的存储器。

[0020] 对于人工智能芯片而言,由于硬件成本和芯片空间等不同方面的考虑和/或限制,通常只配置容量较低的片上存储器,例如,总容量可以为几百K字节到几M字节的静态存储器或高速缓冲存储器等。因此,人工智能芯片在不与片外存储器进行数据交换的情况下无法独立地对数量可达数十万或数万的ROI数据进行排序。

[0021] 即使只对数千个ROI数据进行排序,在人工智能芯片中,由于需要在片上存储器中分配例如容量至少与存储所有ROI数据的存储区域相同的存储区域,因此也将占用更多的片上存储器资源,甚至需要设置更多的片上存储器,这将增加人工智能芯片的硬件成本。

[0022] 另外,可以将人工智能芯片装配到诸如手机、导航仪这样的终端设备中,以便检测例如经由终端设备上的摄像头等拍摄的图像或视频中的一个或多个关注对象。在这样的情况下,可以控制人工智能芯片将所产生的ROI数据存储到设置在终端设备之内且在人工智能芯片之外的片外存储器中,甚至可以控制由终端设备中的处理器执行对存储于片外存储器中的ROI数据的排序。

[0023] 然而,诸如手机、导航仪这样的终端设备对处理效率、功耗和硬件成本等通常也具有较高要求。通常的ROI排序方案的较高的空间复杂度(例如,需要容量与存储器中存储所有ROI数据的存储区域相同的缓存区域)将导致终端设备需要更高的硬件成本,例如需要配置更多容量的存储器。

[0024] 因此,总是期望能够在确保执行的效率以及结果的准确性和精度的同时,尽可能降低人工智能芯片或装配了人工智能芯片的终端设备的硬件成本。

[0025] 根据本公开的实施例的用于调整ROI数据的存储位置的方法和装置,旨在至少部分地解决或缓解上述技术问题中的一个或多个,使得人工智能芯片或装配了人工智能芯片的终端设备能够高效率地实现ROI数据的存储位置的就地调整,而不必配置容量可能需要大于或等于存储器中用于存储所有ROI数据的存储区域的缓存区域。

[0026] 示例性方法

[0027] 图1示出已经生成并连续地存储在存储器的存储区域SR中的N个ROI数据ROI₁、ROI₂、……、ROI_N,其中,N是大于1的整数,例如数十万、数万或数千,并且ROI₁、ROI₂、……、ROI_N分别存储于存储区域SR中的地址ADDR₁、ADDR₂、……、ADDR_N处。换句话说,N个ROI数据ROI₁、ROI₂、……、ROI_N连续地存储于对应于连续的地址空间ADDR₁至ADDR_N(在本文中,也称为“完整地址范围”)的存储区域SR中。

[0028] 这里,由于每个ROI数据通常可以包括多个数据项(例如,ROI的坐标、宽度、高度、得分等),所以存储区域SR中的地址ADDR₁、ADDR₂、……、ADDR_N中的每个地址对应的是存储区域SR中用于存储每个ROI数据的单位存储区域的首地址,而不是存储区域SR的每个物理上的存储单元的首地址。在本文中,为了简单,假设每个ROI数据只占据存储区域SR的一个存储单元,或者在存储区域SR是根据二维存储器确定的二维存储区域的情况下只占据存储区域SR的一个存储行或缓存行或缓冲区。

[0029] 需要根据各个ROI数据的得分调整各个ROI数据在存储区域SR中的存储位置,使得按照一种次序,对于在调整之后分别存储于ADDR_i和ADDR_{i+1}的ROI_i和ROI_{i+1}(i是大于或等于1且小于N的任意整数),ROI_i的得分小于或等于ROI_{i+1}的得分,或者按照另一种次序,对于在调整之后分别存储于ADDR_i和ADDR_{i+1}的ROI_i和ROI_{i+1},ROI_i的得分大于或等于ROI_{i+1}的得分。

[0030] 图2示出根据本公开的实施例的用于调整ROI₁、ROI₂、……、ROI_N在SR中的存储位置的方法的示例,该示例方法可以包括步骤S110、S120、S130和S140。

[0031] 在步骤S110中,可以根据ROI₁、ROI₂、……、ROI_N中的每一个ROI的得分,确定与完整地址范围ADDR₁至ADDR_N相对应的多个局部地址范围LS₁至LS_K,其中,K为大于1的整数,例如256,并且LS₁至LS_K中的每个局部地址范围的大小(或者在每个局部地址范围中所包括的连续的地址的数量,或者每个局部地址范围对应的存储容量)可以取决于ROI₁、ROI₂、……、ROI_N中具有与该局部地址范围相对应的得分的ROI数据的数量。

[0032] 然后,可以从ROI₁、ROI₂、……、ROI_N中的任何一个数据(例如,ROI₁)开始,对属于ROI₁、ROI₂、……、ROI_N中并且当前缓存在缓冲存储器BUF中的当前要调整的数据ROI_{cur}(在本文中,也称为“第一数据”)执行步骤S120至S140。

[0033] 在步骤S120中,可以确定LS₁至LS_K中的LS_k(在本文中,也称为“第一局部地址范围”,其中,k为大于或等于1且小于或等于K的整数)内的ADDR_{des}(在本文中,也称为“第一地址”),其中,ROI_{cur}具有与LS_k相对应的得分,并且ADDR_{des}处的数据ROI_{next}(在本文中也称为“第二数据”)不是已调整的ROI数据。

[0034] 在步骤S130中,可以将ADDR_{des}处的数据ROI_{next}从存储区域SR存储到缓冲存储器BUF中。

[0035] 在步骤S140中,可以将ROI_{cur}作为已调整的ROI数据存储到存储区域SR中的ADDR_{des}处。

[0036] 需要说明的是,在缓存在缓冲存储器BUF中的ROI_{next}不是已调整的ROI数据或者存储区域SR中还包括未调整的ROI数据的情况下,可以例如根据缓冲存储器BUF中的ROI_{next}确定新的要调整的ROI数据(即,新的ROI_{cur})继续执行上述的步骤S120至S140。如此反复,直至存储区域SR中的所有ROI数据均成为已调整的ROI数据为止。

[0037] 在本文中,存储区域SR中的“已调整的ROI数据”可以意味着该ROI数据是ROI₁、

ROI₂、……、ROI_N中的一个ROI数据,并且其在存储区域SR中的存储位置是调整后的存储位置,因此在后续的处理(如果有)中可以跳过该ROI数据;存储区域SR中的“未调整的ROI数据”可以意味着该ROI数据是ROI₁、ROI₂、……、ROI_N中先前从未被访问和/或处理的一个原始的ROI数据;“不是已调整的ROI数据”可以意味着该数据可以是未调整的ROI数据,也可以是具有诸如空值(NULL)或“-1”等特殊值的特殊ROI数据或特殊数据(例如,可以用于进行标记,或者可以起到简化处理等作用)。

[0038] 通过根据本公开的实施例的方法,可以直接在原始分配的存储区域SR上对ROI数据进行“就地”调整。除了用于存储原始的各个ROI数据本身的存储区域SR之外,用于在调整过程中提供辅助的缓存区域的缓冲存储器BUF可以包括两个缓存行或缓冲区(在另外的实施例中,缓冲存储器BUF也可以是在包括存储区域SR的存储器中的不同于存储区域SR的两个存储区),用于缓冲在调整过程中涉及的中间数据,即前述的ROI_{cur}和ROI_{nxt}。即使相对于通常只配置几百K字节到几M字节的片上存储器的人工智能芯片而言,缓冲存储器BUF的硬件成本也是可以忽略不计的。

[0039] 而且,在根据本公开的实施例的方法中,不需要保存例如用于指示各个ROI数据的前后关系的链接、指针等额外数据。这意味着,存储区域SR和缓冲存储器BUF中的每一个的容量均只取决于ROI数据本身的大小。例如,如果一个ROI数据本身(可以包括ROI坐标、宽度、高度、置信度等与ROI本身有关的信息,但不包括用于指示与其他ROI数据的前后关系的链接、指针等信息)的大小为s个字节,则在不考虑诸如数据对齐等必须的额外开销的情况下,根据本公开的实施例的方法所需的存储空间(包括存储区域SR和缓冲存储器BUF)的总容量只有(N+2)*s个字节。

[0040] 另外,根据本公开的实施例的方法能够在一次性地遍历完所有的ROI数据的同时完成对所有ROI数据的存储位置的调整,具有简单且高效的控制逻辑。

[0041] 因此,通过根据本公开的实施例的示例方法,能够以极小的硬件成本,高效率地实现ROI数据的就地调整,具有极低的空间复杂度和时间复杂度。例如,在将根据本公开的实施例的示例方法实施于人工智能芯片,以对初始存储于片上存储器中的ROI数据的存储位置进行调整的情况下,人工智能芯片可以不必对片上存储器进行扩容或者增设另外的例如容量可能需要大于或等于片上存储器中用于存储所有ROI数据的存储区域的缓冲存储器,并且人工智能芯片能够通过简单的控制逻辑高效率地执行ROI数据的存储位置的调整。

[0042] 应当理解,在根据步骤S120至S140确定的不同循环中,第一数据ROI_{cur}、第一局部地址范围LS_k、第一地址ADDR_{des}、第二数据ROI_{nxt}等以及在下文可能出行的诸如“第二地址”、“第二局部地址范围”等中的每一个可以具有或对应于不同的数据值/数据项/数据内容。

[0043] 下面结合示例描述根据本公开的实施例的方法的更多细节。

[0044] 在不同的实施例中,可以通过区域推荐网络、选择性搜索等适当的方式针对输入的图像或视频中的一个或多个关注对象(例如,图像或视频中的人物、动物、车辆、标志物、建筑物、车道线等)生成ROI数据,其中,所生成的每个ROI数据具有相应的得分,这样的得分通常可以是该ROI的置信度,并且在一些实施例中可以是在计算ROI的置信度的过程中通过省略非线性处理而得到的置信度替代值。

[0045] 在基于ROI的对象检测方法(包括诸如RCNN、SPP-Net、Fast RCNN 和Faster RCNN 这样的检测效果优良的多阶段对象检测方法以及诸如Tiny-SSD等单阶段的对象检测方法)

中,需要根据各个ROI数据的得分对各个ROI数据进行排序,或者说,根据各个ROI数据的得分调整各个ROI数据在存储器中的存储位置,使得在调整之后,各个ROI数据在存储器中是有序地连续存储的。

[0046] 由于硬件方面的限制,例如用于寄存或缓存ROI数据的得分的寄存器或缓冲存储器通常具有有限的位数,所以实际生成的ROI数据的得分以及ROI数据的其他数据项(例如,ROI的坐标、ROI的宽度、ROI的高度等)均只能使用具有预定精度的数值表示。例如,可以采用能够支持32比特精度的浮点数的硬件配置,从而能够表示ROI数据的 $2^{32}=4294967296$ 种得分。

[0047] 如前文所述,在基于GPU或TPU的人工智能系统中,对于资源和功耗可以没有任何限制,因此可以毫无压力地采用32比特精度的浮点数表示方式。然而,对于人工智能芯片以及诸如手机等需要对硬件成本进行严格控制的终端设备而言,32比特精度的浮点数表示方式将导致较高的硬件成本。

[0048] 通过分析基于ROI的对象检测方法的算法原理并进行大量的实验发现,至少对于ROI数据的得分,可以采用比通常的32比特精度更低的精度。例如,通过大量的工作证实,可以将ROI的得分量化到6比特以上的预定精度,例如6比特精度或8比特精度,而不影响最终检测结果的精度。

[0049] 于是,在本公开的实施例,可以将ROI数据的得分量化到预定精度,例如可以低于通常的32比特精度,例如6比特精度、8比特精度、10比特精度、16比特精度等。

[0050] 例如,在量化到6比特精度的情况下,各个ROI数据将具有 $2^6=64$ 种得分;在量化到8比特精度的情况下,各个ROI数据将具有 $2^8=256$ 种得分;在量化到10比特精度的情况下,各个ROI数据将具有 $2^{10}=1024$ 种得分;诸如此类。采用比通常的32比特精度更低的预定精度表示ROI数据的得分,能够显著地降低硬件成本和提高处理效率。

[0051] 如前文所述,生成并存储在存储区域SR中的ROI₁、ROI₂、……、ROI_N的数量N通常可以是数十万、数万或数千。在将各个ROI的得分量化到诸如6比特精度或8比特精度预定精度的情况下,可能出现的ROI得分的最大种数(例如,在6比特精度的情况下为64种,在8比特精度的情况下为256种)也将远小于N,使得ROI₁、ROI₂、……、ROI_N中的一个或多个ROI数据可以具有相同的得分。

[0052] 在一个实施例,对于在预定精度下的任何一种得分,可以在生成ROI₁、ROI₂、……、ROI_N中计数具有这种得分的ROI数据的数量。例如,如图3所示,可以针对在预定精度下的每种ROI得分设置相应的计数器CNT₁至CNT_M,其中,例如在采用8比特精度的情况下,M可以等于256,并且例如计数器CNT₁用于计数具有第一种ROI得分的ROI数据的数量,计数器CNT₂用于计数具有第二种ROI得分的ROI数据的数量,以此类推。然而,应当理解,这并不意味着第一种ROI得分的得分值一定是1并且第一种ROI得分的得分值一定是2。在本文中,为了描述上的方便,使用计数器CNT₁至CNT_M中的每一个的下标(即,1至M中的每一个)代替表示与该计数器相对应的ROI得分的得分值,或者说,在一个实施例,可以将每个ROI数据的得分映射成在与预定精度相对应的包括1至M的自然数范围内的某个数值。

[0053] 为了后续调整/排序的方便,同时也为了描述上的方便,可以规定,对于任何两个计数器CNT_p和CNT_n($1 \leq p < n \leq M$),与计数器CNT_p相对应的ROI得分小于计数器CNT_n相对应的ROI得分。然而,本公开不局限于此,可以根据需要(例如,排序的要求)对计数器与ROI得分

之间的对应关系做出其他规定,例如也可以规定与计数器 CNT_p 相对应的ROI得分大于计数器 CNT_n 相对应的ROI得分。

[0054] 显然,对于计数器 CNT_1 至 CNT_M 中的任何一个计数器 CNT_s (s 是大于或等于1且小于或等于 M 的任一整数),其在生成所有的ROI数据之后的最终计数值或者在执行根据本公开的实施例的方法步骤S110之前的初始值可以是范围 $[0, N]$ 内的值,其中,0可以意味着在所生成并存储在SR中的所有ROI数据中的任何一个ROI数据的得分在预定精度下的值均不为 s ,而 N 可以意味着所生成并存储在SR中的所有ROI数据的得分在预定精度下的值均为 s 。实际上,如上所述,在预定精度为6比特以上的情况下,计数器 CNT_s 在步骤S110之前的初始值通常不会为 N ,甚至通常不会是接近于的某个值。

[0055] 然后,在步骤S110中,可以根据计数器 CNT_1 至 CNT_M 中的每个计数器 CNT_s 的初始值确定与存储区域SR的完整地址范围 $ADDR_1$ 至 $ADDR_N$ 相对应的多个局部地址范围 LS_1 至 LS_K 。

[0056] 例如, K 可以与 M 相同,并且如果计数器 CNT_1 的初始值为 $C1$ 并且计数器 CNT_2 的初始值为 $C2$,即,在存储于SR中的 ROI_1 、 ROI_2 、……、 ROI_N 中有 $C1$ 个ROI数据的得分在预定精度下的值为1,并且有 $C2$ 个ROI数据的得分在预定精度下的值为2,则与计数器 CNT_1 对应的局部地址范围 LS_1 可以包括存储区域SR中的连续的 $C1$ 个地址 $ADDR_1$ 至 $ADDR_{C1}$,并且与计数器 CNT_2 对应的局部地址范围 LS_2 可以包括存储区域SR中的连续的 $C2$ 个地址 $ADDR_{C1+1}$ 至 $ADDR_{C1+1+C2}$,其中,如果 $C1=0$,则与计数器 CNT_2 对应的局部地址范围 LS_2 可以包括存储区域SR中的连续的 $C2$ 个地址 $ADDR_1$ 至 $ADDR_{1+C2}$,并且此时可以禁用计数器 CNT_1 ,使得局部地址范围 LS_1 实际上变得无效并因此而可以不必考虑。在另外的示例中,实际确定的局部地址范围的数量 K 可以小于或等于实际设置的计数器的数量 M 。例如,在一个或多个计数器被禁用的情况下, K 可以小于 M 。

[0057] 类似地,可以确定局部地址范围 LS_1 至 LS_K 中的其他局部地址范围,并且类似地,初始值为0的计数器可以被禁用。

[0058] 为了描述上的方便,在图3的示例中,计数器 CNT_1 至 CNT_M 中的每一个的初始值均不为0。相应地,局部地址范围 LS_1 至 LS_K 中的每一个可以包括完整地址范围 $ADDR_1$ 至 $ADDR_N$ 内一个或多个的连续的地址,并且完整地址范围 $ADDR_1$ 至 $ADDR_N$ 中的任一个地址均只属于局部地址范围 LS_1 至 LS_K 中的一个局部地址范围。

[0059] 在图3的示例中,当前要调整的数据 ROI_{cur} 是 ROI_1 、 ROI_2 、……、 ROI_N 中的 ROI_i (i 是大于或等于1且小于或等于 N 的整数),并且可以在执行步骤S120之前已经缓存在缓冲存储器BUF的缓存行或缓冲区 $L1$ (在本文中也称为“第一缓冲区”)中。

[0060] 例如,如果 ROI_{cur} (例如,图3的示例中的 ROI_i) 是 ROI_1 、 ROI_2 、……、 ROI_N 中的第一个被调整的ROI数据,则可以在执行步骤S120之前,从存储区域SR中存储 ROI_{cur} 的原始地址 $ADDR_{org}$ (例如,图3的示例中的 $ADDR_i$) 处读取 ROI_{cur} ,并将其缓存到缓冲存储器BUF的缓存行 $L1$ 中。

[0061] 在确定 ROI_{cur} 之后,例如,在将 ROI_{cur} 缓存到缓冲存储器BUF中之后,可以将 $ADDR_{org}$ 处的数据标记为不再是未处理的ROI数据。

[0062] 在一个实施例中,可以在将 ROI_{cur} 从存储区域SR缓存到缓冲存储器BUF的缓存行 $L1$ 之后,将存储区域SR中的地址 $ADDR_{org}$ 处的数据清空或者设置为诸如空数据(NULL)或“-1”等特殊数据,也可以修改 $ADDR_i$ 处的ROI数据,使得在 $ADDR_{org}$ 处的修改后的ROI数据中的某个数据项(例如得分)具有例如“-1”或空数据这样的特殊值。

[0063] 然后,在步骤S120中,可以根据缓存到缓冲存储器BUF中的 ROI_{cur} 的得分,确定局部地址范围 LS_1 至 LS_M 中与该得分相对应的局部地址范围 LS_k 。例如,可以根据缓冲存储器BUF中的 ROI_{cur} 的得分,确定上述的计数器 CNT_1 至 CNT_M 中与该得分相对应的计数器 CNT_k ,从而确定局部地址范围 LS_k 。

[0064] 显然,此时的局部地址范围 LS_k 中的至少一个地址处的数据不是已调整的ROI数据(可以是未调整的ROI数据,也可以是空数据或其他特殊数据)。于是,可以确定局部地址范围 LS_k 中的一个地址 $ADDR_{des}$ (例如,图3的示例中的 $ADDR_j$),在地址 $ADDR_{des}$ 的数据 ROI_{nxt} (例如,图3的示例中的 ROI_j)不是已调整的ROI数据。

[0065] 在一个实施例中,可以逐个地检测局部地址范围 LS_k 中的每个地址处的数据状态,从而确定上述地址 $ADDR_{des}$ 。

[0066] 在另一个实施例中,可以根据与局部地址范围 LS_k 相对应的计数器 CNT_k 中的计数值或数据确定上述地址 $ADDR_{des}$ 。例如,可以根据计数器 CNT_k 中的计数值确定相对于局部地址范围 LS_k 的基准地址(例如, LS_k 的首地址)的偏移量,并且可以根据该偏移量确定局部地址范围 LS_k 中的地址 $ADDR_{des}$ 。然后,可以在适当的时机,例如在确定地址 $ADDR_{des}$ 之后、在读取地址 $ADDR_{des}$ 处的 ROI_{nxt} 之后、在将 ROI_{cur} 写入到地址 $ADDR_{des}$ 处之后,更新 CNT_k 中的数据或计数值,使得更新后的数据对应于局部地址范围 LS_k 内的未被访问过的地址。

[0067] 例如,可以确定在计数器 CNT_k 的每个计数值与局部地址范围 LS_k 中的每个地址之间的对应关系,并且利用计数器 CNT_k 的当前计数值从局部地址范围 LS_k 中确定上述地址 $ADDR_{des}$ 。例如,可以控制计数器 CNT_k 在适当的时机从初始值(即,具有与 k 相对应的得分的ROI数据的数量)开始递减地计数,并将计数器 CNT_k 的每个计数值作为例如从局部地址范围 LS_k 的首地址开始向局部地址范围 LS_k 的尾地址方向的偏移量。当初始值不为0的计数器 CNT_k 的计数值成为0时,意味着所有的具有与 k 相对应的得分的ROI数据的存储位置均已经调整完毕。然后,可以根据计数器 CNT_k 的当前计数值 $CNT_k.V$ 和局部地址范围 LS_k 的首地址,确定地址 $ADDR_{des}$ 。由此,可以确保局部地址范围 LS_k 中的地址 $ADDR_{des}$ 处的ROI数据不是已调整的ROI数据,并且在地址 $ADDR_{des}$ 之后直至局部地址范围 LS_k 的尾地址中的每个地址处的ROI数据是已调整的ROI数据。

[0068] 在一个实施例中,可以设置分别与局部地址范围 LS_1 至 LS_M 或计数器 CNT_1 至 CNT_M 相对应的地址寄存器 AR_1 至 AR_M (图3中未示出),其中,地址寄存器 AR_k 用于保存局部地址范围 LS_k 的例如首地址。也就是说,与局部地址范围 LS_k 相对应的寄存器和/或计数器例如可以包括地址寄存器 AR_k 。如前文所述,由于 M 的值较小,所以这样的硬件开销即使对于人工智能芯片来说也是完全可接受的。

[0069] 在另外的实施例中,例如可以在与存储器或缓冲存储器中存储局部地址范围 LS_1 至 LS_M 中的每个局部地址范围的首地址或尾地址,而不必单独地设置地址寄存器。

[0070] 应当理解,本公开不局限于上述示例。例如,由于相对地址表示形式(例如,计数值或地址偏移量)与绝对地址表示形式可以容易相互转换,所以作为示例在本文中描述的计数器 CNT_1 至 CNT_M 的使用方式可以修改为基于相对地址表示形式(例如,计数值或地址偏移量)的其他任何适当的处理方式,并且可以修改为与基于绝对地址表示形式的处理方式。

[0071] 在确定地址 $ADDR_{des}$ 之后,可以在步骤S130中将原始存储于 $ADDR_{des}$ 处的 ROI_{nxt} 作为 ROI_{nxt} 缓存到缓冲存储器BUF中,并且在步骤S140中将缓冲存储器BUF中的 ROI_{cur} 以覆盖

ADDR_{des}处的原始数据的方式存储到ADDR_{des}处。

[0072] 在一个实施例中,可以在步骤S130中,将缓冲存储器BUF中的ROI_{cur}从缓冲存储器BUF的缓存行L1移动到缓冲存储器BUF的另一个缓存行或缓冲区L2(在本文中也称为“第二缓冲区”)中,然后将存储于ADDR_{des}处的ROI_{next}缓存到缓冲存储器BUF的缓存行L1中。然后,可以在步骤S140中,将缓存行L2中的ROI_{cur}以覆盖ADDR_{des}处的原始数据(即,ROI_{next})的方式存储到ADDR_{des}处。

[0073] 在另一个实施例中,缓冲存储器BUF的缓存行L1中的ROI_{cur}可以在先前的步骤S120中移动到缓冲存储器BUF的缓冲区L2中。然后,在步骤S130中,ADDR_{des}处的ROI_{next}可以直接缓存到缓冲存储器BUF的缓存行L1中。然后,在步骤S140中,可以将缓存行L2中的ROI_{cur}以覆盖ADDR_{des}处的原始数据的方式存储到ADDR_{des}处。在该实施例中,可以控制在一个读写周期内实现步骤S130和S140。

[0074] 例如,如图4所示,可以在一个时钟的上升沿执行步骤S130以控制将ADDR_{des}处的ROI_{next}缓存到缓冲存储器BUF的缓存行L1中,并且在该时钟的下降沿执行步骤S140以控制将缓存行L2中的ROI_{cur}以覆盖ADDR_{des}处的原始数据的方式存储到ADDR_{des}处。

[0075] 由此,当前要调整的ROI_{cur}(例如,图3的示例中的ROI_i)可以作为已调整的ROI数据存储到SR的ADDR_{des}处,并且与LS_k相对应的计数器CNT_k的计数值可以在此过程中的适当时机更新,例如减1,并在计数值仍不为0的情况下成为与ADDR_{des}的前一个地址ADDR_{des-1}相对应的值。如果CNT_k的计数值成为0,则意味着所有的具有与k相对应的得分的ROI数据的存储位置均已经调整完毕,并且在一个实施例中可以禁用计数器CNT_k。

[0076] 如图3所示,在一种情况下,每次循环中要处理的ROI_{cur}或者在每次循环中确定的要在下一次循环中处理的ROI_{next}可以直接是ROI₁至ROI_N中的一个未调整的ROI数据。

[0077] 于是,可以将ROI_{next}(即,ROI_j)作为下一次循环的ROI_{cur}。例如,可以在通过步骤S140将ROI_{cur}=ROI_i存储到存储区域SR的ADDR_{des}=ADDR_j处之后并且进入下一次循环的步骤S120之前,将变量ROI_{cur}从ROI_i更新为ROI_j。然后,针对新的ROI_{cur}=ROI_j再次执行步骤S120至S140。

[0078] 例如,可以通过对循环次数进行计数,并且可以在循环次数达到N次时结束处理。

[0079] 在另外的一些情况(例如,当前的ROI_{cur}是ROI₁至ROI_N中的最后一个需要调整的ROI数据,在步骤S120中确定的ADDR_{des}与读取ROI_{cur}的ADDR_{org}相同,在步骤S120中确定的ADDR_{des}在先前的循环中已经被处理过,等等)下,在步骤S130中缓存到缓冲存储器BUF中的ROI_{next}可能不是未调整的ROI数据,而是例如在先前的循环中已处理过的ROI数据或者被施加标记(例如,数据本身被设置为诸如空数据、“-1”等特殊数据,或者数据中的诸如得分等数据项具有诸如“-1”等特殊值)的ROI数据,或者是在当前循环中正在处理的ROI_{cur}(此时,ADDR_{des}与ADDR_{org}相同),等等。

[0080] 为此,可以在步骤S140之后并且在执行下一次循环的步骤S120之前,例如,在BUF中当前缓存的ROI_{next}不是未调整的ROI数据并且存储区域SR中的已调整的ROI数据的数量小于N的情况下,确定存储区域SR中的另一个未被访问过的地址ADDR_{new}(在本文中,也称为“第二地址”),并使用地址ADDR_{new}处的ROI数据更新BUF中的ROI_{next}并作为下一个要调整的ROI_{cur}。

[0081] 为了确定新的地址ADDR_{new},在一个实施例中,可以确定LS₁至LS_K中的LS_k, (在本文

中,也称为“第二局部地址范围”,其中, k' 为大于或等于1且小于或等于K的整数, k' 与k可以相同或不同),其中, LS_k 内可以包括至少一个未被访问过的地址,例如,与 LS_k 相对应的计数器 $CNT_{k'}$ 的计数值当前不为0。然后,可以根据计数器 $CNT_{k'}$ 的计数值确定新的地址 $ADDR_{new}$ 。

[0082] 在一个实施例中,如图5所示,可以在步骤S140之后进入到步骤S150,以检测当前缓存在BUF中的 ROI_{next} 是否为未调整的ROI。在一个实施例中,可以检测当前缓存在BUF的缓存行L1中的 ROI_{next} 是否为诸如空数据或“-1”等特殊数据。在另一个实施例中,可以检测当前缓存在BUF的缓存行L1中的 ROI_{next} 的某些数据项或字段(例如得分)是否具有诸如“-1”等特殊值。

[0083] 在步骤S150返回“是”(例如, ROI_{next} 不是特殊数据或者不包括特殊数据项)的情况下,可以将 ROI_{next} 作为新的 ROI_{cur} ,并且针对新的 ROI_{cur} 执行步骤S120。

[0084] 在步骤S150返回“否”的情况下,可以继续到步骤S160,以重新确定 ROI_{next} 。例如,可以根据 CNT_1 至 CNT_M 中任何一个的计数值仍不为0的计数器 $CNT_{k'}$,从而确定 $LS_{k'}$,然后根据该计数器 $CNT_{k'}$ 的当前计数值确定属于 $LS_{k'}$ 的地址 $ADDR_{new}$ 。然后,可以将地址 $ADDR_{new}$ 处的ROI数据作为新的 ROI_{next} 替换在步骤S130中缓存到BUF中的 ROI_{next} ,并继续执行步骤S120(即,图5的示例中的“成功确定新的 ROI_{next} ”的情况)。

[0085] 如果在步骤S160中发现 CNT_1 至 CNT_M 中每一个的计数值均为0,则可以意味着存储区域SR中的已调整的ROI数据的数量已经达到N,即,所有的ROI均已调整完毕。另外,例如,也可以通过循环次数进行计数,并且可以在循环次数达到N次时结束处理。

[0086] 在上面的示例中,在步骤S150中,通过检测当前缓存在BUF的缓存行L1中的 ROI_{next} 是否为诸如空数据或“-1”等特殊数据或者当前缓存在BUF的缓存行L1中的 ROI_{next} 的某些数据项或字段(例如得分)是否具有诸如“-1”等特殊值,判断当前的 ROI_{next} 是否为未调整的ROI。在另外的实施例中,例如可以设置一个地址寄存器,该寄存器中的初始数据可以是第一个处理的ROI数据在存储区域SR中的原始存储地址。然后,可以在步骤S150中检测在步骤S120中确定是 $ADDR_{des}$ 是否与该地址寄存器中存储的地址相同。如果相同,则步骤S150可以返回“否”,否则返回“是”。然后,在步骤S160中,如果成功地确定新的 ROI_{next} ,可以使用地址 $ADDR_{new}$ 更新该地址寄存器中的数据,并使方法继续到S120。

[0087] 如上所述,计数器 CNT_1 至 CNT_M 的数量可以远小于 ROI_1 至 ROI_N 的数量。而且,在一些实施例中,可以建立计数器的列表,控制在某个计数器的计数值成为0时将该计数器从列表中删除或禁用,然后可以在步骤S160中根据位于列表的表头或表尾处的计数器尝试确定新的 ROI_{next} 。因此,步骤S150和S160的执行过程可以非常迅速,并且甚至不涉及另外的循环处理。

[0088] 根据本公开的实施例的方法能够在遍历完 ROI_1 至 ROI_N 中的每个ROI数据的同时完成对每个ROI数据的存储位置的就地调整。

[0089] 因此,通过根据本公开的实施例的示例方法,能够以极小的硬件成本,高效率地实现ROI数据的就地调整,具有极低的空间复杂度和时间复杂度。

[0090] 例如,在将根据本公开的实施例的示例方法实施于人工智能芯片,以对初始存储于片上存储器中的ROI数据的存储位置进行调整的情况下,人工智能芯片可以不必对片上存储器进行扩容或者增设另外的例如容量可能需要大于或等于片上存储器中用于存储所有ROI数据的存储区域的缓冲存储器,并且人工智能芯片能够通过简单的控制逻辑高效率

地执行ROI数据的存储位置的调整。

[0091] 应当理解,根据本公开的实施例的示例方法可以应用于任何期望对已经生成并连续地存储的多个ROI数据进行就地排序的场景,而不局限于人工智能芯片中的应用。

[0092] 示例性装置

[0093] 图6示出根据本公开的实施例的用于调整多个感兴趣区域数据的存储位置的装置的示例DEV,其可以包括处理器或控制器CON和缓冲存储器BUF。

[0094] 处理器或控制器CON例如可以是基于现场可编程门阵列、ARM处理器等开发的处理器电路或控制逻辑电路,并且可以被配置为在启动(例如,在通电)时根据预定指令执行根据本公开的实施例的方法(例如,图3或图5所示的示例方法)中的步骤。

[0095] 缓冲存储器BUF可以是诸如高速缓冲存储器、寄存器、静态存储器等任何适当类型的存储部件。缓冲存储器BUF可以根据处理器CON的指令或者在处理器CON的控制下存储或缓存处理器CON的处理或逻辑控制过程中涉及的中间数据,包括例如前文所述的ROI_{cur}和/或ROI_{next}。

[0096] 在一个实施例中,缓冲存储器BUF可以包括两个缓冲区或缓存行(例如,图4的示例中的L1和L2),并且每个缓冲区的容量可以取决于单个ROI数据的大小。

[0097] 另外,如图6所示,示例装置DEV还可以包括多个计数器CNT₁至CNT_M,以便分别对具有与1至M相对应的得分的ROI数据的数量进行计数,从而协助处理器CON确定与存储器中存储着ROI₁至ROI_N的存储区域SR的地址范围相对应的多个局部地址范围。

[0098] 在一个实施例中,计数器CNT₁至CNT_M的数量M可以取决于每个ROI数据的得分的精度。例如,在采用8比特精度的情况下,M可以等于256,并且例如计数器CNT₁用于计数具有第一种ROI得分的ROI数据的数量,计数器CNT₂用于计数具有第二种ROI得分的ROI数据的数量,以此类推。

[0099] 应当理解,根据本公开的实施例的装置不局限于图6中的示例。例如,在另外的实施例中,根据本公开的实施例的装置还可以包括诸如用于指令时序控制的定时控制器、用于指令中断控制的中断控制器、用于实现部件之间的互联的交叉开关和/或多路选择器、协处理器等其他部件/模块/电路/单元/元件。

[0100] 另外,在图6中,连线两端的装置/部件/元件/模块/电路可以例如通过数据总线 and/或指令总线等直接地连接或耦接在一起,也可以经由诸如交叉开关(Crossbar)等其他中介装置/部件/元件/模块/电路间接地连接或耦接在一起,其中,连线的箭头可以表示被关注的指令和/或数据的流向,但并不意味着指令和/或数据只能按照箭头所示的方向流动。

[0101] 在一个实施例中,根据本公开的实施例的装置例如可以实施为人工智能芯片的片上装置或片上装置的一部分。在另外的实施例中,根据本公开的实施例的装置可以应用于任何期望对已经生成并连续地存储的多个ROI数据进行就地排序的场景或装置中。

[0102] 示例性计算机程序产品和计算机可读存储介质

[0103] 除了上述方法和设备以外,本公开的实施例还可以是计算机程序产品,其包括计算机程序指令,该计算机程序指令在被处理器运行时使得处理器执行本说明书上述“示例性方法”部分中描述的根据本公开各种实施例的方法中的步骤。

[0104] 计算机程序产品可以以一种或多种程序设计语言的任意组合来编写用于执行本

公开实施例操作的程序代码,程序设计语言可以包括面向对象的程序设计语言,诸如Java、C++等,还包括常规的过程式程序设计语言,诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。

[0105] 此外,本公开的实施例还可以是计算机可读存储介质,例如计算机可读的非临时性存储介质,其上存储有程序指令,程序指令在被处理器运行时使得处理器执行本说明书上述“示例性方法”部分中描述的根据本公开各种实施例的方法中的步骤。

[0106] 计算机可读存储介质可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以包括但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪速存储器)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0107] 以上结合具体实施例描述了本公开的基本原理,但是,需要指出的是,在本公开中提及的优点、优势、效果等仅是示例而非限制,不能认为这些优点、优势、效果等是本公开的各个实施例必须具备的。另外,上述公开的具体细节仅是为了示例的作用和便于理解的作用,而非限制,上述细节并不限制本公开为必须采用上述具体的细节来实现。

[0108] 本公开中涉及的器件、装置、设备、系统的方框图仅作为例示性的例子并且不意图要求或暗示必须按照方框图示出的方式进行连接、布置、配置。如本领域技术人员将认识到的,可以按任意方式连接、布置、配置这些器件、装置、设备、系统。诸如“包括”、“包含”、“具有”等等的词语是开放性词汇,指“包括但不限于”,且可与其互换使用。这里所使用的词汇“或”和“和”指词汇“和/或”,且可与其互换使用,除非上下文明确指示不是如此。这里所使用的词汇“诸如”指词组“诸如但不限于”,且可与其互换使用。

[0109] 还需要指出的是,在本公开的装置、设备和方法中,各部件或各步骤是可以分解和/或重新组合的。这些分解和/或重新组合应视为本公开的等效方案。

[0110] 在本文中,诸如“第一”、“第二”等不带有量词的修饰词旨在用于区分不同的元件/部件/电路/模块/装置/步骤,而不用于强调次序、位置关系、重要程度、优先级别等。与此不同,诸如“第一个”、“第二个”等带有量词的修饰词可以用于强调不同的元件/部件/电路/模块/装置/步骤的次序、位置关系、重要程度、优先级别等。

[0111] 提供所公开的方面的以上描述以使本领域的任何技术人员能够做出或者使用本公开。对这些方面的各种修改对于本领域技术人员而言是非常显而易见的,并且在此定义的一般原理可以应用于其他方面而不脱离本公开的范围。因此,本公开不意图被限制到在此示出的方面,而是按照与在此公开的原理和新颖的特征一致的最宽范围。

[0112] 为了例示和描述的目的已经给出了以上描述。此外,此描述不意图将本公开的实施例限制到在此公开的形式。尽管以上已经讨论了多个示例方面和实施例,但是本领域技术人员将认识到其某些变型、修改、改变、添加和子组合。

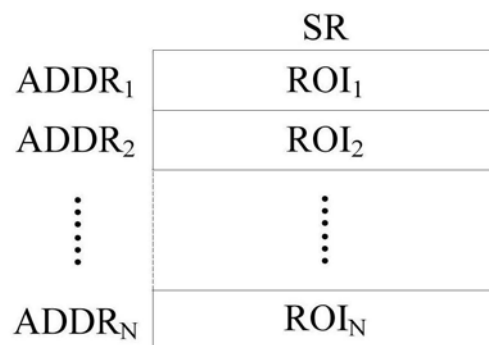


图1



图2

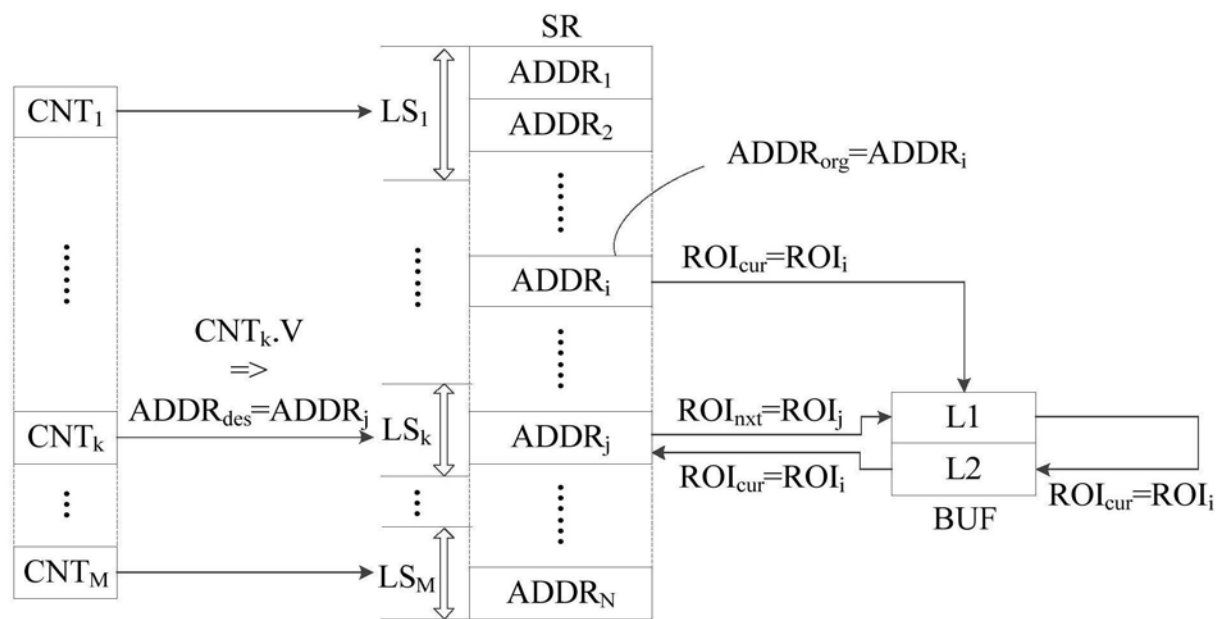


图3



图4

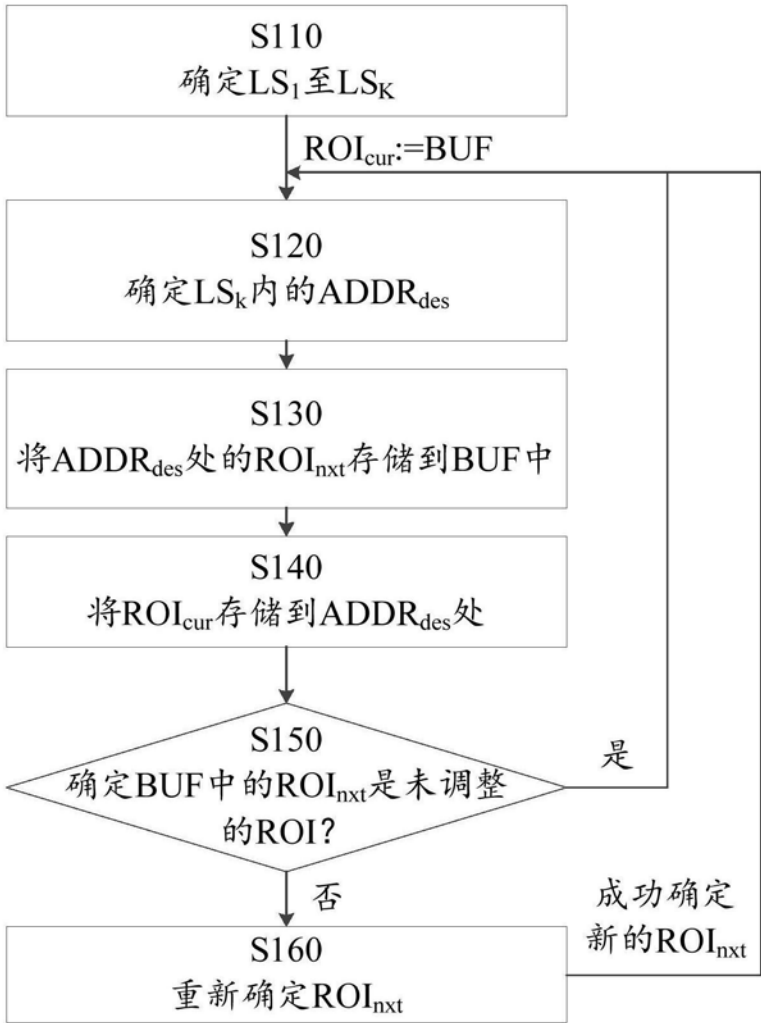


图5

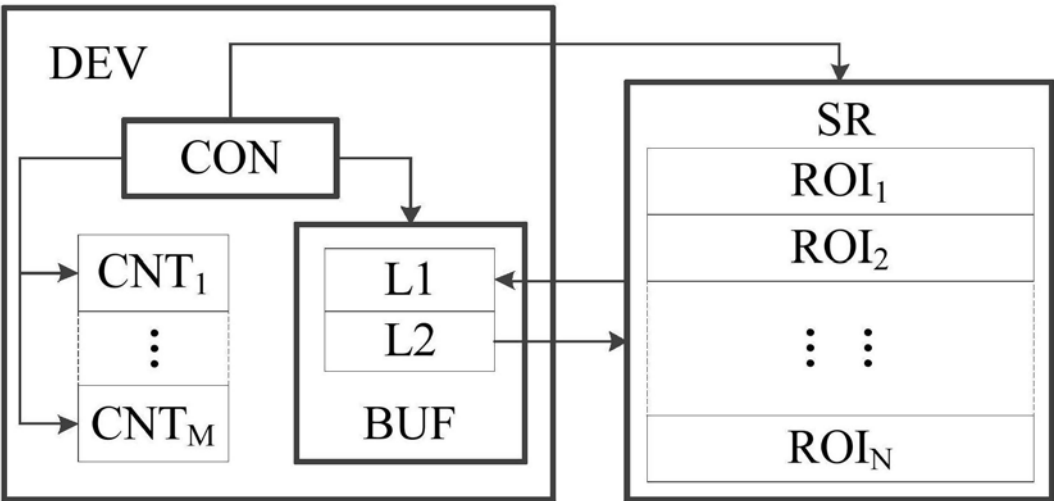


图6