



(12)发明专利申请

(10)申请公布号 CN 110365979 A

(43)申请公布日 2019.10.22

(21)申请号 201910738846.8

G06K 9/46(2006.01)

(22)申请日 2014.07.16

H04N 19/30(2014.01)

(30)优先权数据

H04N 19/40(2014.01)

T02013A000629 2013.07.24 IT

(62)分案原申请数据

201480041502.7 2014.07.16

(71)申请人 新运全球有限公司

地址 英属维尔京群岛托尔托拉

(72)发明人 S·帕斯彻拉克斯

(74)专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 周磊

(51)Int.Cl.

H04N 19/14(2014.01)

H04N 19/90(2014.01)

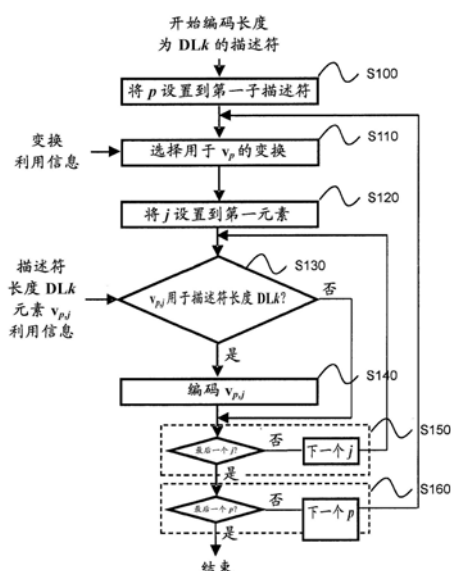
权利要求书2页 说明书12页 附图22页

(54)发明名称

基于梯度直方图编码图像描述符的图像处理装置和方法

(57)摘要

本公开涉及基于梯度直方图编码图像描述符的图像处理装置和方法。描述了用于基于梯度直方图(hp)将图像描述符(H)编码为包括变换后的子描述符(vp)的变换后的描述符(V)的方法,其中每个梯度直方图(hp)包括多个直方图区间(hi)并且每个子描述符(vp)包括一组值(vj),其特征在于所述子描述符(vp)是根据元素利用次序列表生成的,所述元素利用次序列表指定所述一组子描述符(vp)的子描述符索引(p)和所述一组值(vj)的元素索引(j)。



1. 一种编码图像描述符的方法,用于基于梯度直方图将图像描述符编码为包括变换后的子描述符的变换后的描述符,

其中每个梯度直方图包括多个直方图区间并且每个子描述符包括一组值,

其中对于每个子描述符,通过变换相应梯度直方图的每个直方图区间并接着对得到的变换后的值执行标量量化来获得子描述符的所述一组值,以及

其中所述子描述符是根据元素利用次序列表生成的,其中所述元素利用次序列表指定一组子描述符的子描述符索引和所述一组值的元素索引。

2. 如权利要求1所述的方法,其中所述子描述符根据它们离图像描述符中心的距离被分成子描述符组,并且其中在每个子描述符组内,子描述符组的对应值在所述元素利用次序列表中都被分配相同的编码优先级。

3. 如权利要求2所述的方法,其中所述子描述符被分成包含具有到图像描述符中心的最小距离的全部子描述符的第一组和包含全部其余子描述符的第二组。

4. 如权利要求2所述的方法,其中所述子描述符被分成包含具有离描述符中心的最大距离的子描述符的第一组、包含具有到描述符中心的第二大距离的第一子描述符集合的第二组、包含具有到描述符中心的第二大距离的第二子描述符集合的第三组以及包含具有到描述符中心的最小距离的子描述符的第四组。

5. 如权利要求4所述的方法,其中所述第一组、第二组、第三组和第四组包含相同数量的子描述符。

6. 如权利要求1所述的方法,其中所述子描述符根据它们离图像描述符中心的距离并且根据它们的相互距离被分成子描述符组,并且其中在每个子描述符组内,子描述符组的对应值在所述元素利用次序列表中都被分配相同的编码优先级。

7. 如权利要求6所述的方法,其中所述子描述符被分成包含具有离描述符中心的最大距离的子描述符的第一组、包含具有到描述符中心的第二大距离的第一子描述符集合的第二组、包含具有到描述符中心的第二大距离的第二子描述符集合的第三组以及包含具有到描述符中心的最小距离的子描述符的第四组。

8. 如权利要求6所述的方法,其中子描述符组不包含到彼此的距离低于预定阈值的子描述符。

9. 如权利要求6所述的方法,其中所述相互距离包括最大化子描述符之间的总距离。

10. 如权利要求1所述的方法,其中所述子描述符根据它们离图像描述符中心的距离被分成子描述符组并且根据它们的相互距离被排序,并且其中在每个子描述符组内,子描述符组的对应值在所述元素利用次序列表中都被分配相同的编码优先级。

11. 如权利要求10所述的方法,其中子描述符组不包含到彼此的距离低于预定阈值的连续子描述符。

12. 如权利要求10所述的方法,其中所述相互距离包括最大化连续子描述符之间的距离。

13. 如权利要求1所述的方法,其中所述子描述符根据它们离图像描述符中心的距离被分成子描述符组并且每个组的子描述符以根据它们对应的编码特性定义的顺序被编码,并且其中在每个子描述符组内,子描述符组的对应值在所述元素利用次序列表中都被分配相同的编码优先级。

14. 如权利要求13所述的方法,其中所述编码特性是对应的变换。

15. 如权利要求14所述的方法,其中对于每个子描述符组,该组中第一子描述符的变换是第一类型的变换,该组中第二子描述符的变换是所述第一类型的变换,该组中第三子描述符的变换是第二类型的变换并且该组中第四子描述符的变换是所述第二类型的变换。

16. 如权利要求13所述的方法,其中所述编码特性是量化的类型和等级,或者是量化的类型和等级的组合。

17. 如权利要求2所述的方法,其中特定类的值被赋予较高的优先级。

18. 如权利要求2所述的方法,其中最接近描述符中心的子描述符组被赋予较高的优先级。

19. 如权利要求2至18中任一项所述的方法,其中所述元素利用次序列表和图像描述符一起被存储或发送。

20. 如权利要求1至19中任一项所述的编码图像描述符的方法,其中所述图像描述符通过截断所述描述符的一个或多个最后的元素被转换成具有减小的长度的描述符。

21. 如权利要求1所述的编码图像描述符的方法,其中所述元素利用次序列表采取16元素或24元素或32元素或128元素的有序列表的形式。

22. 一种图像处理装置,包括用于执行如权利要求1至21中任何一项所述的方法的部件。

基于梯度直方图编码图像描述符的图像处理装置和方法

[0001] 本申请是申请日为2014年7月16日、申请号为201480041502.7、题为“用于基于梯度直方图编码图像描述符的图像处理装置和方法”的发明专利申请的分案申请。

技术领域

[0002] 本发明涉及用于高效地编码、转码、解码和处理在图像感兴趣关键点周围的局部区域中计算出的图像描述符的方法，并且涉及包括用于编码、转码、解码和处理这种描述符的部件的图像处理设备。

背景技术

[0003] 这种图像描述符已经在许多计算机视觉应用中发现了广泛的适用性，这些应用包括对象识别、基于内容的图像检索和图像配准，仅举几例。

[0004] 对这种描述符进行编码的现有方法呈现出某些缺点。

[0005] 例如，现有的编码方法产生这样的描述符，它们需要解析整个描述符来执行转码，借此给定描述符长度的描述符被转换为不同描述符长度的描述符，或者它们需要解析整个描述符来执行不同长度的描述符的解码和比较。

[0006] 作为另一个例子，现有的编码方法就编码复杂性而言是低效的，因为它们忽略了产生变长图像描述符所需的操作中的共性和冗余。

[0007] 由本申请人自己提交的尚未公布的意大利专利申请No. T02012A000602描述了局部图像描述符的编码，借此基于梯度直方图的变换从采用所述梯度直方图的图像描述符计算健壮、可区分、可扩展且紧凑的图像描述符，其中所述变换捕获包含在梯度直方图中的显著且健壮的信息，这些信息是梯度直方图的区间(bin)值之间的关系和分布的形状的形式的。

[0008] 在所述尚未发表的意大利专利申请中，公开了所述描述符的编码方法，其就产生容易扩展的位流而言比现有技术方法更高效。

[0009] 这种描述符在以上提到的尚未发表的意大利专利申请No. T02012A000602中公开，该申请公开了基于梯度直方图的变换从采用所述梯度直方图的图像描述符计算健壮、可区分、可扩展且紧凑的图像描述符，其中所述变换捕获包含在梯度直方图中的显著且健壮的信息，这些信息是梯度直方图的区间值之间的关系和分布的形状的形式的。

[0010] 根据尚未发表的意大利专利申请No. T02012A000602的从采用梯度直方图的图像描述符(尤其是SIFT图像描述符)计算健壮、可区分、可扩展且紧凑的图像描述符的重要方面在下文进行描述。

[0011] 简要地说，利用SIFT方法，局部图像描述符是如下形成的：首先，执行跨多个图像尺度和位置的搜索，以识别和定位对尺度和朝向不变的稳定的图像关键点；接着，对每个关键点，基于局部图像梯度确定一个或多个主导朝向，从而允许后续的局部描述符计算相对于每个关键点的分配的朝向、尺度和位置来执行，因此实现这些变换的不变性。接着，关键点周围的局部图像描述符如下形成：首先，在关键点周围的区域中的图像样本点处计算梯

度量值和朝向信息;接着,这些样本被累积到总结 $n \times n$ 个子区域上的内容的朝向直方图中。

[0012] 仅仅作作为说明,SIFT关键点描述符的例子在图1a和1b中示出,其中图1a示出了将局部区域R细分成 4×4 子区域SR,并且图1b示出了对于每个朝向直方图将 360° 朝向范围细分成八个区间,其中每个箭头的长度对应于该直方图条目的量值。因此,如图1中所示的局部图像描述符具有 $4 \times 4 \times 8 = 128$ 个元素。SIFT技术的更多细节可以在International Journal of Computer Vision, 60, 2 (2004), pp. 91-110由David G. Lowe所写的“Distinctive image features from scale-invariant keypoints”中找到。

[0013] 根据尚未发表的意大利专利申请No. T02012A000602,健壮、可区分、可扩展且紧凑的图像描述符可以从SIFT描述符如下计算。

[0014] 在以下描述中,H是包括16个梯度直方图h的整个SIFT描述符,每个梯度直方图h具有八个区间h,而V是根据本发明的包括16个子描述符v的整个局部描述符,每个子描述符v具有八个元素v。

[0015] 令H表示包括16个梯度直方图 h_0-h_{15} 的SIFT局部图像描述符,如图2a中所示,每个直方图包括八个区间值 h_0-h_7 ,如图2b中所示。更健壮、可区分、可扩展且紧凑的图像描述符可以通过变换H的 h_0-h_{15} 中的每一个并接着对结果产生的变换后的值执行标量量化来计算。更具体而言,根据图3的变换利用信息, h_0-h_{15} 中的每一个根据变换A或变换B被变换,如以下所示,即变换A应用到 $h_0, h_2, h_5, h_7, h_8, h_{10}, h_{13}, h_{15}$ 并且变换B应用到 $h_1, h_3, h_4, h_6, h_9, h_{11}, h_{12}, h_{14}$,给出变换后的描述符V具有分别对应于 h_0-h_{15} 的子描述符 v_0-v_{15} ,并且每个子描述符包括元素 v_0-v_7 ,从而给出总共128个元素。

[0016] 变换A (1)

[0017] $v_0 = h_2 - h_6$

[0018] $v_1 = h_3 - h_7$

[0019] $v_2 = h_0 - h_1$

[0020] $v_3 = h_2 - h_3$

[0021] $v_4 = h_4 - h_5$

[0022] $v_5 = h_6 - h_7$

[0023] $v_6 = (h_0 + h_4) - (h_2 + h_6)$

[0024] $v_7 = (h_0 + h_2 + h_4 + h_6) - (h_1 + h_3 + h_5 + h_7)$

[0025] 变换B (1)

[0026] $v_0 = h_0 - h_4$

[0027] $v_1 = h_1 - h_5$

[0028] $v_2 = h_7 - h_0$

[0029] $v_3 = h_1 - h_2$

[0030] $v_4 = h_3 - h_4$

[0031] $v_5 = h_5 - h_6$

[0032] $v_6 = (h_1 + h_5) - (h_3 + h_7)$

[0033] $v_7 = (h_0 + h_1 + h_2 + h_3) - (h_4 + h_5 + h_6 + h_7)$

[0034] 接着,每个元素经受粗标量量化,例如三元(3级)量化,其中量化阈值被选择为在用于每个元素的量化区间之间实现特定的发生概率分布。这种标量量化产生量化后的描述

符 \tilde{V} , 具有子描述符 $\tilde{v}_0 - \tilde{v}_{15}$, 每个子描述符包括元素 $\tilde{v}_0 - \tilde{v}_7$, 再次具有总共128个元素。这种紧凑的描述符捕获包含在原始梯度直方图中的最可区分且健壮的信息, 其中信息是它们的区间值之间的关系和分布的形状的形式的。

[0035] 描述符 V 以及其量化版本 \tilde{V} 的关键优点是它是高度可扩展的, 并且, 如果应用的存储需求或传输通道的特性要求的话, 其维度可以通过简单地消除其一个或多个元素来容易地减小。为了简化, 在以下描述中, 将就量化之前的具有子描述符 $v_0 - v_{15}$ 的描述符 V 的编码来描述本发明的重要方面, 其中每个子描述符包括元素 $v_0 - v_7$, 并且, 除非另外声明, 否则应当理解, 量化后的描述符 \tilde{V} 的编码以类似的方式进行。

[0036] 图4a-4e示出了已经被找出以对五个目标描述符长度产生卓越的可区分能力和健壮性的示例性元素集合, 从描述符长度0 (DL0)、仅利用20个描述符元素的最短描述符长度到描述符长度4 (DL4)——利用所有128个元素的最长描述符长度。更具体而言, 图4a示出了用于包括20个元素的描述符长度DL0的示例性元素集合, 图4b示出了用于包括40个元素的描述符长度DL1的示例性元素集合, 图4c示出了包括64个元素的描述符长度DL2的示例性元素集合, 图4d示出了用于包括80个元素的描述符长度DL3的示例性元素集合, 并且图4e示出了用于包括全部128个元素的描述符长度DL4的示例性元素集合。因此, 对于每个描述符长度, 每个子描述符的每个元素将根据图4a-4e的元素利用集合来编码或者不根据图4a-4e的元素利用集合来编码。

[0037] 这种可扩展性属性的关键在于对每个描述符长度被利用的元素集合必须与对所有更高描述符长度被利用的元素集合相同或者是其子集, 如图4a-4e中所示。通过对具有更高描述符长度的描述符的过多元素的简单消除, 使得其被减小至与具有较低描述符长度的描述符的相同元素集合, 这允许具有不同长度的描述符的转码和比较。

[0038] 这种描述符的简明的编码方法包括以“按子描述符”次序计算和编码元素, 即, 在一般情况下是作为 $v_{0,0}, v_{0,1}, \dots, v_{0,7}, v_{1,0}, v_{1,1}, \dots, v_{1,7}, \dots, v_{15,0}, v_{15,1}, \dots, v_{15,7}$, 其中 $v_{i,j}$ 表示子描述符 v_i 的元素 v_j 。这意味着通过使用适当的变换来编码变换后的直方图 v_0 的元素 v_0, v_1, \dots, v_7 , 接着编码变换后的直方图 v_1 的元素 v_0, v_1, \dots, v_7 , 等等, 例如图3中所示, 并且还期望的描述符长度使用适当的元素利用集合, 例如图4中所示, 以决定哪些元素应当被编码。

[0039] 这种编码, 例如对描述符长度DL0产生描述符 $v_{0,0}, v_{1,0}, v_{2,0}, v_{3,0}, v_{4,0}, v_{5,0}, v_{5,6}, v_{6,0}, v_{6,6}, v_{7,0}, v_{8,0}, v_{9,0}, v_{9,6}, v_{10,0}, v_{10,6}, v_{11,0}, v_{12,0}, v_{13,0}, v_{14,0}, v_{15,0}$ 并且对描述符长度DL1产生描述符 $v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}, v_{2,0}, v_{2,1}, v_{3,0}, v_{3,1}, v_{4,0}, v_{4,1}, v_{5,0}, v_{5,1}, v_{5,2}, v_{5,6}, v_{6,0}, v_{6,1}, v_{6,2}, v_{6,6}, v_{7,0}, v_{7,1}, v_{8,0}, v_{8,1}, v_{9,0}, v_{9,1}, v_{9,2}, v_{9,6}, v_{10,0}, v_{10,1}, v_{10,2}, v_{10,6}, v_{11,0}, v_{11,1}, v_{12,0}, v_{12,1}, v_{13,0}, v_{13,1}, v_{14,0}, v_{14,1}, v_{15,0}, v_{15,1}$ 。

[0040] 图5作为一系列步骤示出了这种简明的编码器的操作。在下面的描述中以及在编码器操作的后续描述中, 除非另有规定, 否则这种步骤序列对应于概念性的步骤并且不对应于软件实现、组件和指令的具体硬件, 而是代表编码器的整体操作。更具体而言, 图5示出了编码器对描述符长度DLk的操作, 例如对应于图4中所示的一个描述符长度。在图5的步骤S100中, 描述符的编码在第一子描述符(即 v_0)处开始。在步骤S110中, 为被处理的子描述符选择适当的变换, 例如根据图3的变换利用。应当指出的是, 根据这里描述的两种不同变换

从描述符H计算描述符V只是例子。从描述符H计算描述符V还可以根据单个变换进行,例如仅变换A或仅变换B,从而使步骤S110变得不必要,或者可以根据两个以上的变换。在步骤S120中,被处理的子描述符的编码在第一个子描述符元素(即 v_0)处开始。接着,在步骤S130中,特定子描述符的特定元素(即 $v_{0,0}$)的使用与否对照用于描述符长度 DL_k 的元素利用信息进行检查,例如使用图4的其中一个利用集合。如果元素未被使用,则处理移至步骤S150。如果元素对描述符长度 DL_k 在使用中,则其编码在步骤S140中发生。在这里以及在编码器操作的后续描述中,除非另有规定,否则“编码”一词是指使元素 $v_{0,0}$ 成为局部图像描述符的一个或多个动作或者其组合,所述动作包括(作为例子但不是限制)根据前面看到的(1)或(2)的适当变换函数的计算、在不了解哪些元素最终将在描述符中被使用时全部元素都被预先计算的情况下用于包括到局部图像描述符中的元素的选择、元素值的量化、元素在易失性或非易失性存储器中的存储以及元素沿传输信道的传输。在步骤S140之后,或者如果在步骤S130中决定元素不对描述符长度 DL_k 使用,则处理移至步骤S150。在步骤S150中,如果当前元素不是子描述符的最后一个元素,则处理移至子描述符的下一个元素,否则处理移至步骤S160。在步骤S160中,如果当前子描述符不是局部图像描述符的最后一个子描述符,则处理移至局部图像描述符的下一个子描述符,否则处理结束。因此,很清楚,步骤S100、S120、S150和S160涉及处理被执行的次序,而步骤S110、S130和S140涉及局部图像描述符的实际编码。

[0041] 这种描述符的另一种简明的编码方法包括以“按元素”次序计算和编码元素,即在一般情况下作为 $v_{0,0}, v_{1,0}, \dots, v_{15,0}, v_{0,1}, v_{1,1}, \dots, v_{15,1}, \dots, v_{0,7}, v_{1,7}, \dots, v_{15,7}$,即再次通过使用适当的变换来编码用于子描述符 v_0, v_1, \dots, v_{15} 的元素 v_0 ,接着编码用于子描述符 v_0, v_1, \dots, v_{15} 的元素 v_1 ,等等,例如如图3中所示,并且还对期望的描述符长度使用适当的元素利用集合,例如如图4中所示,以决定哪些元素应当被编码。这种编码器可以以与图5的编码器类似的方式操作,具有适当的步骤重排序。一般而言,以上提到的两种方法中任一种都不提供优于另一种的优点。以相关的计算机视觉应用为目的,为了转码、解码和处理,解码器也必须知道编码过程和元素排序以及利用集合,以便能够处理和比较(可能具有不同长度的)描述符。因此,元素利用集合必须或者永久地被固定或者与描述符一起被存储/发送。在该上下文中,该简明的编码过程是不利的。

[0042] 更具体而言,这种编码忽略了编码次序中不同元素之间的相对重要性。因此,就转码而言,借此给定描述符长度的描述符被转换成不同描述符长度的描述符,或者就解码和通过比较两个描述符之间的对应元素来比较不同长度的描述符而言,这种编码必需对描述符进行解析以获得期望的结果。

[0043] 此外,这种编码忽略了不同元素之间的相对重要性中的冗余模式,并且关于决定特定元素是否应当被编码是不必要地复杂的。

发明内容

[0044] 因此,本发明的目标是公开用于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置,该方法相对于现有技术的方法更高效。

[0045] 本发明的另一个目标是公开用于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置,该方法更为灵活。

[0046] 本发明的另一个目标是公开用于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置,该方法允许实现优化的编码器实现。

[0047] 本发明的另一个目标是公开用于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置,该方法允许获得任何长度的图像描述符。

[0048] 本发明的这些和其它目标是通过用于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置来实现的,如作为本说明书的组成部分的权利要求所要求保护的。

[0049] 简单地说,公开了用于高效地编码图像描述符的方法,诸如以上描述的通过根据元素利用次序编码它们的方法,从而产生可以通过简单的描述符截断而不是解析被转换成较低描述符长度的可扩展描述符。

[0050] 编码是根据子描述符组执行的,子描述符组是根据所述子描述符组的对应元素之间的相对重要性中的冗余模式形成的。

[0051] 更具体而言,所述分组是通过对它们的对应元素在按描述符的所有元素根据其就实现高识别性能而言的相对重要性的排序中具有相似重要性的子描述符进行分组来执行的,更具体而言是通过根据它们离描述符中心的距离来分组子描述符并附加地通过根据它们之间的距离来分组子描述符和/或附加地通过根据对应的编码特性排序一组的子描述符和/或附加地通过根据它们之间的距离排序一组的子描述符。

[0052] 根据本发明的编码方法就效率、计算复杂性和/或生成可扩展位流所需的信息量而言有利地比现有技术方法更高效。

[0053] 本发明的更多特征在要作为本说明书的组成部分的权利要求中阐述。

附图说明

[0054] 具体参考附图,从以下对用于基于梯度直方图编码图像描述符的方法以及相关图像处理装置的详细描述,以上目标将变得更加明晰,其中:

[0055] -图1a和图1b示出了现有技术的关键点描述符的例子;

[0056] -图2a和图2b分别示出了图1的关键点描述符的梯度直方图和与所述梯度直方图之一相关的区间值;

[0057] -图3示出了被应用到图2的梯度直方图的示例性变换;

[0058] -图4a-图4e示出了用于五个相应的目标描述符长度的示例性元素集合;

[0059] -图5代表示出使用图4的元素集合的编码器操作的流程图;

[0060] -图6示出了由根据本发明的方法使用的元素利用次序;

[0061] -图7示出了使用图6的元素利用次序的编码器的操作;

[0062] -图8示出了局部图像描述符的区域和子区域的中心;

[0063] -图9示出了按照根据本发明的方法的第一实施例或第四实施例的局部图像描述符的子描述符的第一分组;

[0064] -图12和图14分别示出了按照根据本发明的方法的第一实施例的局部图像描述符的子描述符的第二和第三示例性分组;

[0065] -图10、图13和图15分别示出了与图9、图12、图14和图17的分组相关的第一、第二和第三组-元素利用次序;

[0066] -图11示出了按照根据本发明的方法的第一、第二或第四实施例的编码器的操作;

- [0067] -图16示出了按照根据本发明的方法的第一实施例的第四示例性分组；
- [0068] -图17示出了按照根据本发明的方法的第二和第三实施例的局部图像描述符的子描述符的第五示例性分组；
- [0069] -图18示出了按照根据本发明的方法的第三实施例的使用图17的元素利用次序的编码器的操作；
- [0070] -图19示出了按照根据本发明的方法的第四实施例的编码器的操作；
- [0071] -图20示出了按照根据本发明的方法的第五实施例的局部图像描述符的子描述符的示例性分组；
- [0072] -图21a-图21e示出了根据图20的分组的示例性元素集合；
- [0073] -图22a-图22e分别示出了当被转换成组-元素利用集合时图21a-图21e的元素集合；
- [0074] -图23示出了用于描述符编码的编码器的操作，用于提供图22a-图22e的转换后的元素集合；
- [0075] -图24示出了适于执行根据本发明的方法的图像处理设备。

具体实施方式

[0076] 根据本发明，不是根据用于不同描述符长度的元素利用集合来进行编码，更高效的编码器可以根据元素利用次序操作，从而产生其元素根据元素利用次序排序并且其可以通过简单的描述符截断被转换成更低描述符长度的描述符。这种元素利用次序可以采取128元素有序列表的形式，其可以在112个字节中进行编码，每个列表条目指定子描述符索引和元素索引，如图6中所示。因此，例如，图6示出了元素优先级列表，借此元素 $v_{5,0}$ 被赋予最高优先级，元素 $v_{9,0}$ 被赋予第二高优先级，等等。因此这种元素利用次序编码器可以通过编码列表中顶部的1个元素来产生长度为1的描述符。

[0077] 图7示出了使用图6的元素利用次序的这种编码器的操作。利用图7的编码器，描述符的编码以按元素利用次序具有顶部优先级（优先级1）的元素（即子描述符 v_5 的元素 v_0 ）开始。在步骤S200中，根据这个元素所属的子描述符来选择适当的变换，例如根据图3的变换利用。应当指出的是，根据这里描述的两种不同变换从描述符H计算描述符V只是例子。在不同的实施例中，从描述符H计算描述符V还可以根据单个变换进行，例如仅变换A或仅变换B，从而使步骤S200变得不必要，或者从描述符H计算描述符V还可以根据两个以上的变换。接着，元素（即 $v_{5,0}$ ）的编码在步骤S210中发生。接着，在步骤S220中，如果描述符的期望数目1个元素还未被编码，则处理移至按元素利用次序具有下一个最高优先级的元素，否则处理结束。因此，步骤S220涉及有多少元素要进行编码的控制，而步骤S200和S210涉及局部图像描述符的实际编码。

[0078] 因此，图7的编码器通过使用如图6中的元素利用次序，产生其元素根据元素利用次序排序并且其可以通过简单的描述符截断（即，消除描述符的最后的元素）被转换成较低描述符长度的描述符，并且比使用如图4中的元素利用次序的图5的编码器更灵活。

[0079] 描述符中的元素数目1可以与描述符一起被存储/发送，有可能在图像级。为了转码、解码和处理，解码器也必须知道元素排序以便能够为了相关的计算机视觉应用进行处理。因此，元素利用集合必须或者永久地被固定或者与描述符一起被存储/发送。

[0080] 但是,在实践当中,图7的编码器的效率可以提高。这是因为图6中所示的元素利用次序包含显著数量的冗余,并且就某种程度来说其生成和使用是不实际的。

[0081] 其原因是V的128个元素的完整集合不对应于其元素可以根据单个优先级列表排序的单个描述符,而是对应于16个不同的8元素子描述符,每个子描述符是根据特定的变换从不同的梯度直方图提取的并且因此所有子描述符之间的对应元素捕获具有相同角度分离的区间之间的关系。

[0082] 因此,已经发现,就利用有限的元素集合实现高识别性能而言,描述符需要在均匀的元素分布(即,从尽可能多的子描述符选择(一个或多个)元素)与离描述符中心的距离(即,赋予到描述符中心更近的子描述符更高的优先级)之间寻求平衡。同时,还已经发现,当子描述符到描述符中心的距离相同时,来自不同子描述符的对应元素的重要性近似相同,而来自不同子描述符的对应元素的重要性随着子描述符到描述符中心的距离减小而增加。

[0083] 在该上下文中,子描述符v到描述符中心的距离指的是对应于引起子描述符v的梯度直方图h的子区域的中心与包括这些子区域的区域的中心之间的距离,如图8中所示。虽然有可能返回去参考图像区域和子区域的维度来计算所述距离,但这是没必要的,因为只对比较所述距离感兴趣。因此,所述距离可以通过假设每个子区域的每一边具有例如单位长度来计算。此外,在本描述中,所述距离是欧几里得距离,但是其它合适的距离度量也可以被使用。

[0084] 因此,据观察,为了图像描述符的高效编码,其子描述符可以被分组使得用于每一组的子描述符的对应描述符元素按利用次序被分配共同的重要性并且被联合编码。

[0085] 第一实施例

[0086] 在本发明的第一实施例中,子描述符是根据其离描述符中心的距离来分组的。

[0087] 例如,一个这样的分组在图9中示出,其中有三个组,即,包含具有到描述符中心的最大距离的子描述符的 $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$,包含具有到描述符中心的第二大距离的子描述符的 $g_1 = \{v_1, v_2, v_4, v_7, v_8, v_{11}, v_{13}, v_{14}\}$,及包含具有到描述符中心的最小距离的子描述符的 $g_2 = \{v_5, v_6, v_9, v_{10}\}$ 。在每个组内,子描述符按升序子描述符索引次序来排序,但这不是约束性的并且可以使用其它次序,诸如从组中最顶部最左边的子描述符开始按顺时针方向。在每个组内,组的子描述符的对应元素全都被分配相同的编码优先级。

[0088] 基于这种分组,可以生成组-元素利用次序,这可以采用24元素有序列表的形式,其可以在15个字节中编码,每个列表条目指定子描述符组和元素索引,如图10中所示。应当指出的是,图10的组-元素利用次序仅仅是例子,并且不同的组-元素利用次序可以通过改变列表中条目的优先级而生成。因此,就长度和编码大小而言,图10的组-元素利用次序比图6的元素利用次序经济得多。因此,例如,图10示出了组-元素优先级列表,借此组 g_2 的元素 v_0 被赋予最高优先级,从而指示编码器前四个要编码的元素是 $v_{5,0}, v_{6,0}, v_{9,0}$ 和 $v_{10,0}$,组 g_1 的元素 v_0 被赋予第二高优先级,从而指示编码器接下来要编码的八个元素是 $v_{1,0}, v_{2,0}, v_{4,0}, v_{7,0}, v_{8,0}, v_{11,0}, v_{13,0}$ 和 $v_{14,0}$,等等。

[0089] 图11示出了使用图10的组-元素利用次序并且被配置为按所述组-元素利用次序编码顶部m个元素组的这种编码器的操作。利用图11的编码器,描述符的编码以按组-元素利用次序具有顶部优先级(优先级1)的元素组(即,包括子描述符 v_5, v_6, v_9 和 v_{10} 的组 g_2 的元

素 v_0)开始。在步骤S300中,描述符的编码在该组的第一个子描述符(即, v_5)处开始。在步骤S310中,为该子描述符选择适当的变换,例如根据图3的变换利用。应当指出的是,根据这里描述的两种不同变换从描述符H计算描述符V只是例子。在不同的实施例中,从描述符H计算描述符V还可以根据单个变换进行,例如仅变换A或仅变换B,从而使步骤S310变得不必要,或者从描述符H计算描述符V还可以根据两个以上的变换。接着,元素(即, $v_{5,0}$)的编码在步骤S320中发生。在步骤S330中,如果当前的子描述符不是组中最后一个子描述符,则处理移至下一个子描述符,否则处理移至步骤S340。接着,在步骤S340中,如果期望数目的 m 组元素还未被编码,则处理移至按组-元素利用次序具有下一个最高优先级的元素组,否则处理结束。因此,步骤S300、S330和S340涉及处理被执行的次序并且涉及有多少元素组要被编码的控制,而步骤S310和S320涉及局部图像描述符的实际编码。

[0090] 描述符中组的数目 m 或者对应的元素数目 l 可以与描述符一起被存储/发送,有可能在图像级。

[0091] 图10的组-元素利用次序比图6的元素利用次序更经济的事实产生更高效和经济的编码器。此外,如前面所看到的,为了转码、解码和处理,解码器也必须知道编码过程和元素利用次序,以便能够为了相关的计算机视觉应用而处理和比较描述符,这意味着元素利用次序必须或者永久地被固定或者与描述符一起被发送。不同的应用可能需要改变元素利用次序,有可能在图像或子图像级,例如通过向最接近描述符中心的子描述符赋予更高的优先级,或者通过对特定类的元素(例如, v_7 而不是 v_2)赋予更高的优先级,以便利用有限的元素集合实现高识别性能。在这种情况下,元素利用次序必须与描述符一起被存储或发送。考虑低位速率描述符通常是几百个字节的大小,图10的组-元素利用次序代表比图6的元素利用次序低得多的开销。此外,子描述符分组可以是固定的并且对编码器和解码器都已知,或者可以与描述符一起被发送。例如,对于到目前为止所考虑的分组,组的数目以及每个组的组成的大小可以在小于10个字节中编码。

[0092] 作为根据本发明第一实施例的另一个例子,不同的分组在图12中示出,其中有两个组,即,包含所有外围子描述符的 $g_0 = \{v_0, v_1, v_2, v_3, v_4, v_7, v_8, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$ 和包含具有到描述符中心的最小距离的子描述符(即,所有中心子描述符)的 $g_1 = \{v_5, v_6, v_9, v_{10}\}$ 。因此,对于这个例子,组 g_0 包含具有到描述符中心的变化距离的子描述符,但总是比组 g_0 的子描述符离中心更远。在每个组内,该组的子描述符的对应元素全都被分配相同的编码优先级。基于这种分组,可以生成组-元素利用次序,这可以采取16元素有序列表的形式,其可以在8个字节中编码,每个列表条目指定子描述符组和元素索引,如图13中所示。于是,图11的编码器可以再次被用来根据图13的组-元素利用次序编码描述符。应当指出的是,图13的组-元素利用次序仅仅是例子,并且不同的组-元素利用次序可以通过改变列表中条目的优先级而生成。

[0093] 作为根据本发明第一实施例的另一个例子,不同的分组在图14中示出,其中有四个组,即,包含具有到描述符中心的最大距离的子描述符的 $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$,包含具有到描述符中心的第二大距离的一组四个子描述符的 $g_1 = \{v_1, v_2, v_4, v_7\}$,包含再次具有到描述符中心的第二大距离的一组四个子描述符的 $g_2 = \{v_8, v_{11}, v_{13}, v_{14}\}$,以及包含具有到描述符中心的最小距离的子描述符的 $g_3 = \{v_5, v_6, v_9, v_{10}\}$ 。因此,对于这个例子,组 g_1 和 g_2 的子描述符离描述符的中心有相同的距离。这种分组是通过将原始组 g_1 细分成新组 g_1 和 g_2 从图9的

分组获得的。其优点在于它产生具有相同数目的子描述符的组，这在优化的编码器实现中是期望的。在每个组内，该组的子描述符的对应元素全都被分配相同的编码优先级。基于这种分组，可以生成组-元素利用次序，这可以采取32元素有序列表的形式，其可以在20个字节中编码，每个列表条目指定子描述符组和元素索引，如图15中所示。于是，图11的编码器可以再次被用来根据图15的组-元素利用次序编码描述符。应当指出的是，图15的组-元素利用次序仅仅是例子，并且不同的组-元素利用次序可以通过改变列表中条目的优先级而生成。

[0094] 很清楚，图14分成四个组的分组不是唯一的。例如，另选分组在图16中示出，包含组 $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$ 、 $g_1 = \{v_1, v_2, v_{13}, v_{14}\}$ 、 $g_2 = \{v_4, v_7, v_8, v_{11}\}$ 和 $g_3 = \{v_5, v_6, v_9, v_{10}\}$ 。因此，在图16中，组 g_0 和 g_3 与图14中的完全相同，但 g_1 和 g_2 的子描述符与图14相比被交换了，使得那两个组中的每一个包含来自子描述符网格的左上、右上、左下和右下部分的一个子描述符。

[0095] 图7的编码器与图11的编码器之间的一个差异是，不像产生任意长度的描述符的前者，后者允许对其长度具有由子描述符组决定的粒度的描述符的编码。在实践当中，后者可以被配置为产生任意长度的描述符，如随后将被证明的。

[0096] 第二实施例

[0097] 在本发明的第二实施例中，子描述符首先根据它们离中心的距离（第一条件）并且其次根据它们的相互距离（第二条件）来分组。

[0098] 在该上下文中，子描述符之间的距离可以再次采取欧几里德距离或另一种合适距离度量的形式，诸如曼哈顿距离。

[0099] 第二个条件可以是，例如，组不应当包含到彼此的距离低于预定阈值的子描述符。因此，预定阈值可以被设置为防止例如分组邻近的子描述符。

[0100] 第二条件的目的是确保不同组的子描述符在任何可能的时候都取自子描述符网格相对远的位置，由此增加具有非常低数目的特征的描述符的信息内容。应当指出的是，第二条件可能不是总被满足，例如，对于包含所有中心子描述符 v_5, v_6, v_9 和 v_{10} 的组，不可能满足该条件。

[0101] 例如，一种这样的分组在图17中示出，其中由四个组，即，包含具有到描述符中心的最大距离的子描述符的 $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$ ，包含具有到描述符中心的第二大距离的一组四个子描述符的 $g_1 = \{v_1, v_7, v_8, v_{14}\}$ ，包含再次具有到描述符中心的第二大距离的一组四个不同子描述符的 $g_2 = \{v_2, v_4, v_{11}, v_{13}\}$ ，以及包含具有到描述符中心的最小距离的子描述符的 $g_3 = \{v_5, v_6, v_9, v_{10}\}$ 。因此，对于这个例子，组 g_0, g_1 和 g_2 的子描述符满足它们不包含邻近子描述符的条件。在每个组内，该组的子描述符的对应元素都被分配相同的编码优先级。基于这种分组，可以生成诸如图15中所示的组-元素利用次序，并且诸如图11中所示的编码器则可以再次被用来根据图15的组-元素利用次序编码描述符。

[0102] 很清楚，基于组内子描述符的距离的另选条件也可以被采用，诸如最大化子描述符之间的总距离，等等。

[0103] 第三实施例

[0104] 在本发明的第三实施例中，子描述符是根据它们离中心的距离分组的并且每个组的子描述符是以根据其对应的编码特性（诸如其对应的变换）所定义的顺序被编码的。

[0105] 例如,结合图3的变换部署模式考虑本发明第二实施例的图17的分组,可以看到,在每个组内,两个子描述符根据变换A被变换并且两个子描述符根据变换B被变换。因此,有可能设置公共的编码顺序条件,其对于每个组,变换利用顺序必须是“A A B B”,即对于每个子描述符组,使得组中第一子描述符的变换是变换A、组中第二子描述符的变换也是变换A、组中第三子描述符的变换是变换B并且组中第四子描述符的变换是变换B的编码顺序。因此,存在包含具有到描述符中心的最大距离的子描述符的 $g_0 = \{v_0, v_{15}, v_3, v_{12}\}$,包含具有到描述符中心的第二大距离的一组四个子描述符的 $g_1 = \{v_7, v_8, v_1, v_{14}\}$,包含再次具有到描述符中心的第二大距离的一组四个不同子描述符的 $g_2 = \{v_2, v_{13}, v_4, v_{11}\}$,以及包含具有到描述符中心的最小距离的子描述符的 $g_3 = \{v_5, v_{10}, v_6, v_9\}$ 。在每个组内,该组的子描述符的对应元素都被分配相同的编码优先级。此外,在每个组内,用于该组中四个子描述符的变换利用信息总是“A A B B”,这意味着高效的编码器实现不需要识别每个子编码器的适用变换。

[0106] 图18示出了使用图15的组-元素利用次序并且被配置为按所述组-元素利用次序编码顶部m组元素的这种编码器的操作。对于图18的编码器,描述符的编码以按组-元素利用次序具有顶部优先级(优先级1)的元素组(即,包括子描述符 v_5, v_{10}, v_6 和 v_9 的组 g_3 的元素 v_0)开始。在步骤S400中,描述符的编码在该组的第一个子描述符(即, v_5)处开始。给定所有组都具有公共且固定的变换利用次序,处理移至步骤S410,在那里,元素(即, $v_{5,0}$)的编码发生。在步骤S420中,如果当前的子描述符不是组中的最后一个子描述符,则处理移至下一个子描述符,否则处理移至步骤S430。接着,在步骤S430中,如果期望数目的m组元素还未被编码,则处理移至按组-元素利用次序具有下一个最高优先级的元素组,否则处理结束。因此,步骤S400、S420和S430涉及处理被执行的次序并且涉及有多少元素组要被编码的控制,而只有步骤S410涉及局部图像描述符的实际编码。

[0107] 在以上描述中,公共编码顺序条件对在每个组内被利用的变换定义,但是这个条件还可以对其它编码特性定义,诸如量化的类型和等级,或者量化的类型和等级的组合。

[0108] 很清楚,如果需要,则不同的编码顺序条件可以应用到子描述符的不同组。例如,在本发明的第一实施例中,根据图9的分组产生三个组,即,包含具有到描述符中心的最大距离的四个子描述符的 g_0 、包含具有到描述符中心的第二大距离的八个子描述符的 g_1 以及包含具有到描述符中心的最小距离的四个子描述符的 g_2 。在这种情况下,由于不同的组大小,因此变换利用顺序必须是“A A B B”的编码顺序条件可以应用到组 g_0 和 g_2 ,并且变换利用顺序必须是“A A A A B B B B”的编码顺序条件可以应用到组 g_1 。

[0109] 第四实施例

[0110] 在本发明的第四实施例中,子描述符是根据其离中心的距离分组的并且每个组的子描述符根据它们之间的距离来排序。

[0111] 排序条件可以是,例如,组内连续子描述符之间的距离不应当低于预定阈值。因此,预定阈值可以被设置为防止例如组内的连续的邻近子描述符。

[0112] 不同的排序条件可以是,例如,组内连续子描述符之间的距离被最大化。

[0113] 当采用大的子描述符组时,为了减小组-元素利用次序的大小以及部分组编码,这种排序特别有用,如以下解释的。应当指出的是,这种类型的排序可能不是总是可行,例如,对于包含所有中心子描述符 v_5, v_6, v_9 和 v_{10} 的组,不具有连续的邻近描述符是不可能的。

[0114] 例如,考虑图9中所示的分组,可以看到,每个组内的子描述符可以被排序,以最大

化连续子描述符之间的距离,从组内具有最低索引的子描述符开始,从而产生 $g_0 = \{v_0, v_{15}, v_3, v_{12}\}$, $g_1 = \{v_1, v_{14}, v_2, v_{13}, v_7, v_8, v_{11}, v_4\}$ 和 $g_2 = \{v_5, v_{10}, v_6, v_9\}$ 。在每个组内,该组的子描述符的对应元素都被分配相同的编码优先级。

[0115] 基于这种分组以及每个组内子描述符的排序,可以生成诸如图10中所示的组-元素利用次序,并且诸如图11中所示的编码器则可以再次被用来根据图10的组-元素利用次序编码描述符。另选地,诸如图19中所示的编码器可以被采用。

[0116] 更具体而言,图7的编码器与图11和图18的编码器之间的一个区别在于,不像产生任意长度的描述符的前者,后者允许对其长度具有由子描述符组决定的粒度的描述符进行编码。但是,在实践当中,在已经达到期望的描述符长度时通过最后一个元素组的部分编码,图11和图18的每个编码器都可以容易地被配置为编码任意长度的描述符。在该上下文中,图19示出了图11的编码器的这种修改。本质上,图19的编码器是通过简单地交换图11的编码器的步骤S330和S340来获得的,从而一旦期望数目的元素已经被编码就允许图19的编码器终止组的编码。很清楚,类似的修改也适用于图18的编码器。

[0117] 在该上下文中,组内的子描述符根据它们之间距离的排序是非常有利的,因为它导致连续的元素取自子描述符网格的相对远的位置,在组的部分编码的情况下,这增加了具有低数目的特征的描述符的信息内容。

[0118] 第五实施例

[0119] 本发明前面的实施例证明了根据组-元素利用次序的图像描述符的高效编码,其中组-元素利用次序是得自将子描述符基于它们离描述符中心的距离和/或它们到彼此的距离分组成子描述符组。

[0120] 在本发明的另选实施例中,子描述符可以被分组,使得组内的每个子描述符具有与组内所有其它子描述符完全相同的元素利用集合。

[0121] 例如,考虑图20的分组,其中有四个组,即, $g_0 = \{v_5, v_6, v_9, v_{10}\}$ 、 $g_1 = \{v_1, v_7, v_8, v_{14}\}$ 、 $g_2 = \{v_2, v_4, v_{11}, v_{13}\}$ 和 $g_3 = \{v_0, v_3, v_{12}, v_{15}\}$ 。这是与图17中相同的分组,但是具有重新分配的组索引(即, g_0 和 g_3 已经交换),使得具有较低索引的组包含更靠近描述符中心的子描述符。组合这种分组与图4a-4e的元素利用集合导致图21a-21e的组-元素利用集合。

[0122] 由于用于每个描述符长度的被利用元素集合必须与用于所有更高描述符长度的被利用元素集合相同或者是其子集,因此图21a-21e的组-元素利用集合可以被转换成如图22a-22e中所示的递增的组-元素利用集合,其中,对于每个描述符长度(例如图22c的DL2),与紧挨着更低描述符长度(例如图22b的DL1)相比,对应的元素利用集合只示出构成所述描述符长度的描述符的附加元素。

[0123] 基于图22a-22e的递增的组-元素利用集合,编码器可以生成可以通过简单的描述符截断被转换成较低描述符长度的描述符。

[0124] 图23示出了用于长度为DLk的描述符的编码的这种编码器的操作。更具体而言,在步骤S600中,处理通过编码最低描述符长度(即,DL0)的描述符开始。在步骤S610中,DL0描述符的编码以第一子描述符组(即, g_0)的编码开始,并且在步骤S620中,第一子描述符组的编码以第一个元素(即, v_0)的编码开始。在步骤S630中,如果根据图22的递增组-元素利用集合,元素 v_0 对处于描述符长度DL0的 g_0 而言不在使用中,则处理移至步骤S680,否则处理移至步骤S640。在步骤S640中,选择组 g_0 的第一个子描述符,即 v_5 ,并且在步骤S650中,选择适

当的变换函数,例如根据图3。应当指出的是,根据这里描述的两种不同变换从描述符H计算描述符V只是例子。在不同的实施例中,从描述符H计算描述符V还可以根据单个变换进行,例如仅变换A或仅变换B,从而使步骤S650变得不必要,或者从描述符H计算描述符V还可以根据两个以上的变换。接着,在步骤S660中,元素 $v_{5,0}$ 的编码发生。在步骤S670中,如果当前的子描述符不是组中的最后一个子描述符,则处理移至该组中的下一个子描述符,否则处理移至步骤S680。在步骤S680中,如果当前元素不是最后一个元素,即 v_7 ,则处理移至下一个元素,否则处理移至步骤S690。在步骤S690中,如果当前的子描述符组不是最后一个子描述符组,则处理移至下一个子描述符组,否则处理移至步骤S695。在步骤S695中,如果当前的子描述符长度不是目标子描述符长度,则处理移至下一个子描述符长度,从而编码由递增组-元素利用集合指定的附加元素。否则,处理结束。

[0125] 虽然对于从SIFT图像描述符计算健壮、可区分、可扩展和紧凑的图像描述符详细地给出了本发明的实施例和各方面,但是本发明适用于基于梯度直方图的其它图像描述符,如在尚未发表的意大利专利申请No. T02012A000602中概述的。

[0126] 仅仅作为例子,图24示出了用于执行根据本发明的方法的概念性图像处理装置。更具体而言,处理装置1100从输入装置1000接收输入,这可以包括诸如图像或视频数据的视觉数据、基于梯度直方图的预先计算的描述符、根据本发明方法的预先计算的紧凑描述符、编程指令或用户输入,其中输入装置1000可以采取用户输入装置、媒体阅读器或所发送信号的接收器的形式。处理装置1100包括以下主要处理块:控制其它处理块的操作的中央处理单元1110、易失性存储器1120、非易失性存储器1130、可选的被配置为基于梯度直方图生成描述符的描述符提取器块1140、被配置为执行根据本发明的方法的紧凑描述符编码器块1150、以及可选的被配置为例如为了建立或验证视觉对应性而处理所述紧凑描述符的紧凑描述符处理器块1160。处理装置1100连接到输出装置1900,输出装置1900可以采取视觉显示单元、媒体书写器或信号发送器的形式,它提供可以包括以下各项的输出:诸如图像或视频数据的带注释的视觉数据、诸如已建立或经过验证的视觉对应性的处理信息或者根据本发明的方法的紧凑描述符。应当理解的是,图24中所示的处理块和体系架构仅仅是概念性的并且可以不确切地对应于实现根据本发明的方法的每个装置。

[0127] 在不背离发明性构思的新颖精神的情况下,本文作为例子描述的用于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置可以经受许多可能的变化;还应当清楚,在本发明的实际实现中,所示出的细节可以具有不同的形式(shape)或者用其它技术上等同的元件代替。

[0128] 因此,很容易理解,本发明不限于基于梯度直方图编码图像描述符的方法以及相关的图像处理装置,而是可以在不背离发明性构思的情况下经受许多修改、改进或等同部分和元件的替代,如权利要求中清楚地指示的。

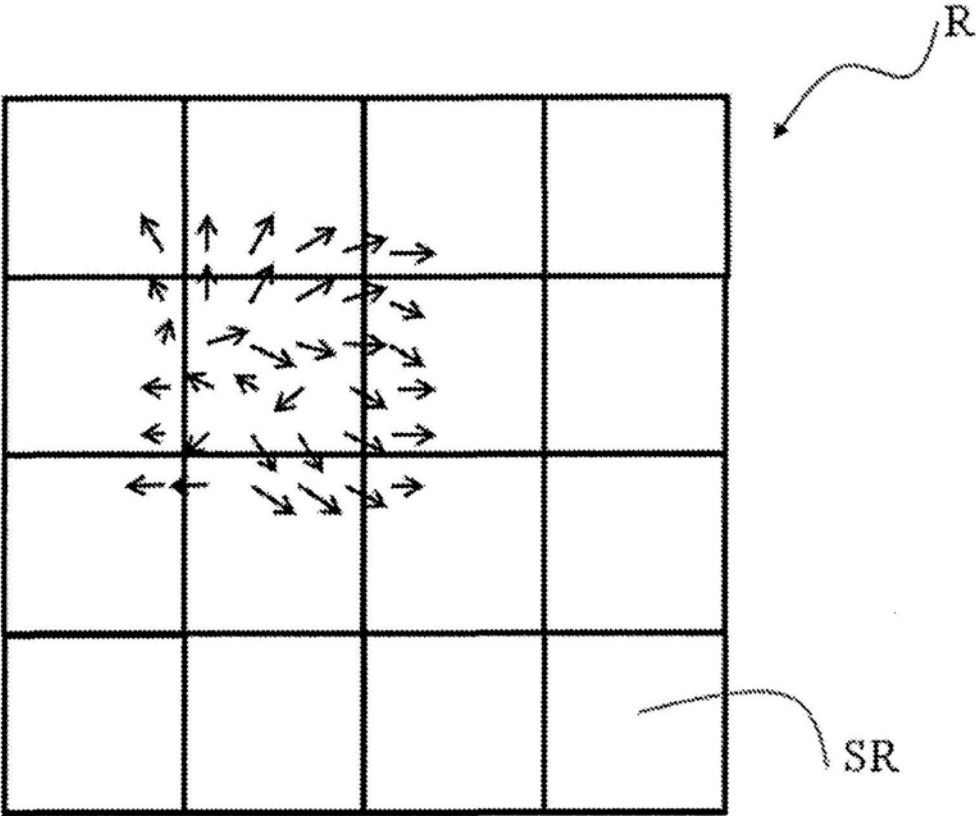


图1a

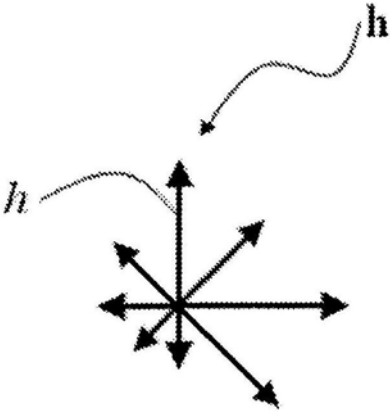


图1b

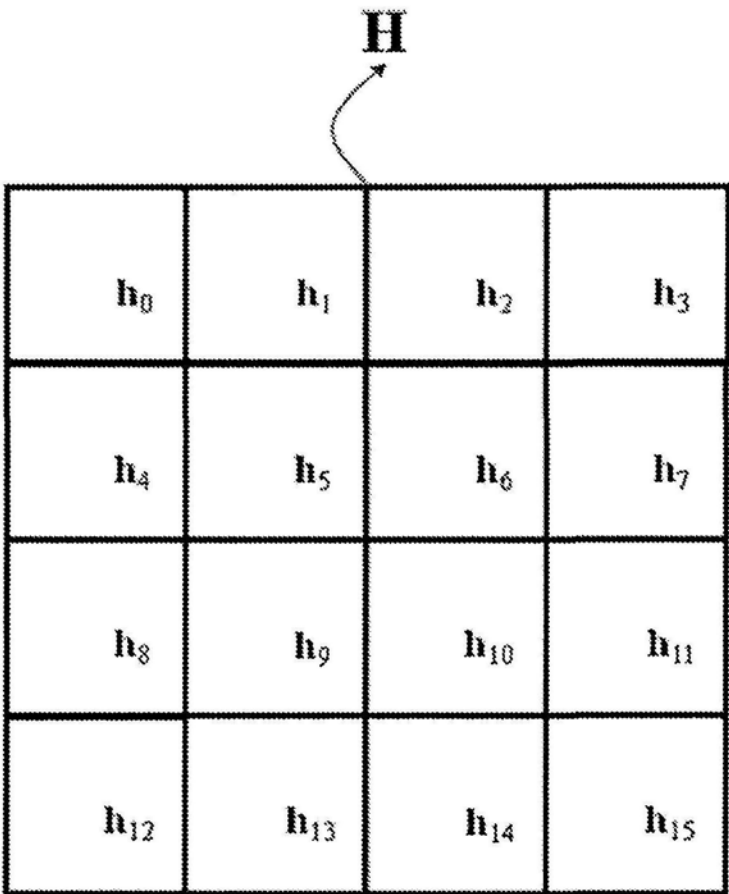


图2a

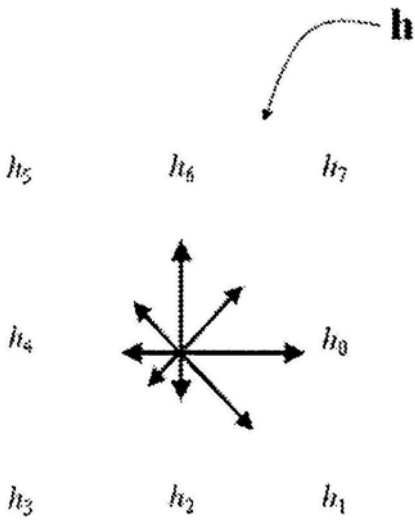


图2b

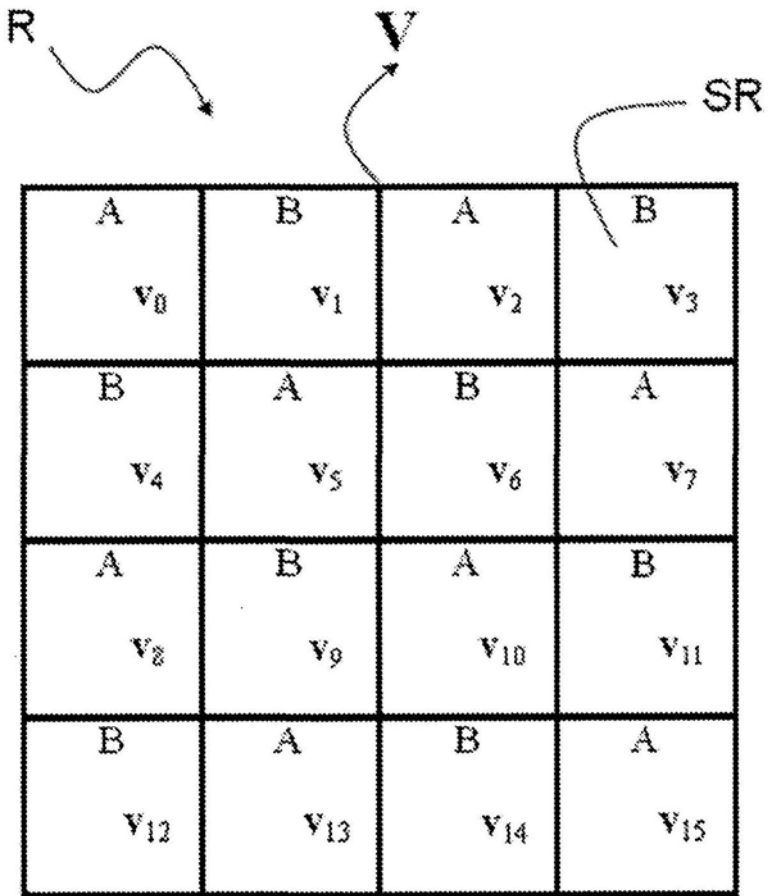


图3

v ₀					v ₁					v ₂					v ₃				
v ₀					v ₀					v ₀					v ₀				
v ₄					v ₅					v ₆					v ₇				
v ₀					v ₀				v ₆	v ₀				v ₆	v ₀				
v ₈					v ₉					v ₁₀					v ₁₁				
v ₀					v ₀				v ₆	v ₀				v ₆	v ₀				
v ₁₂					v ₁₃					v ₁₄					v ₁₅				
v ₀					v ₀					v ₀					v ₀				

图4a

v_0					v_1					v_2					v_3				
v_0	v_1				v_0	v_1				v_0	v_1				v_0	v_1			
v_4					v_5					v_6					v_7				
v_0	v_1				v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1			
v_8					v_9					v_{10}					v_{11}				
v_0	v_1				v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1			
v_{12}					v_{13}					v_{14}					v_{15}				
v_0	v_1				v_0	v_1				v_0	v_1				v_0	v_1			

图4b

v_0					v_1					v_2					v_3				
v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6
v_4					v_5					v_6					v_7				
v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6
v_8					v_9					v_{10}					v_{11}				
v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6
v_{12}					v_{13}					v_{14}					v_{15}				
v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6	v_0	v_1	v_2		v_6

图4c

v ₀						v ₁						v ₂						v ₃													
v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇
v ₄						v ₅						v ₆						v ₇													
v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇
v ₈						v ₉						v ₁₀						v ₁₁													
v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇
v ₁₂						v ₁₃						v ₁₄						v ₁₅													
v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇

图4d

v_0								v_1								v_2								v_3							
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_4								v_5								v_6								v_7							
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_8								v_9								v_{10}								v_{11}							
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_{12}								v_{13}								v_{14}								v_{15}							
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7

图4e

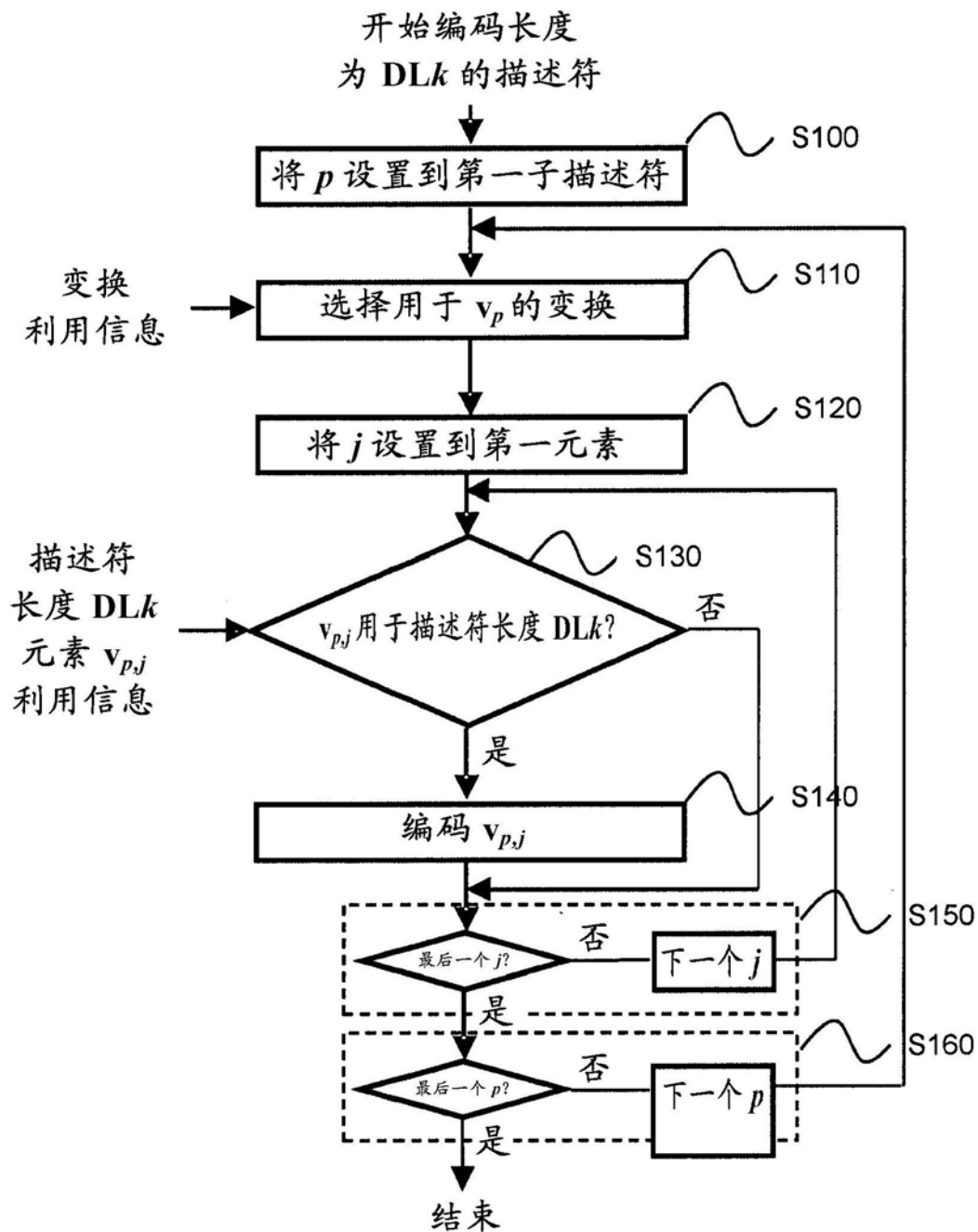


图5

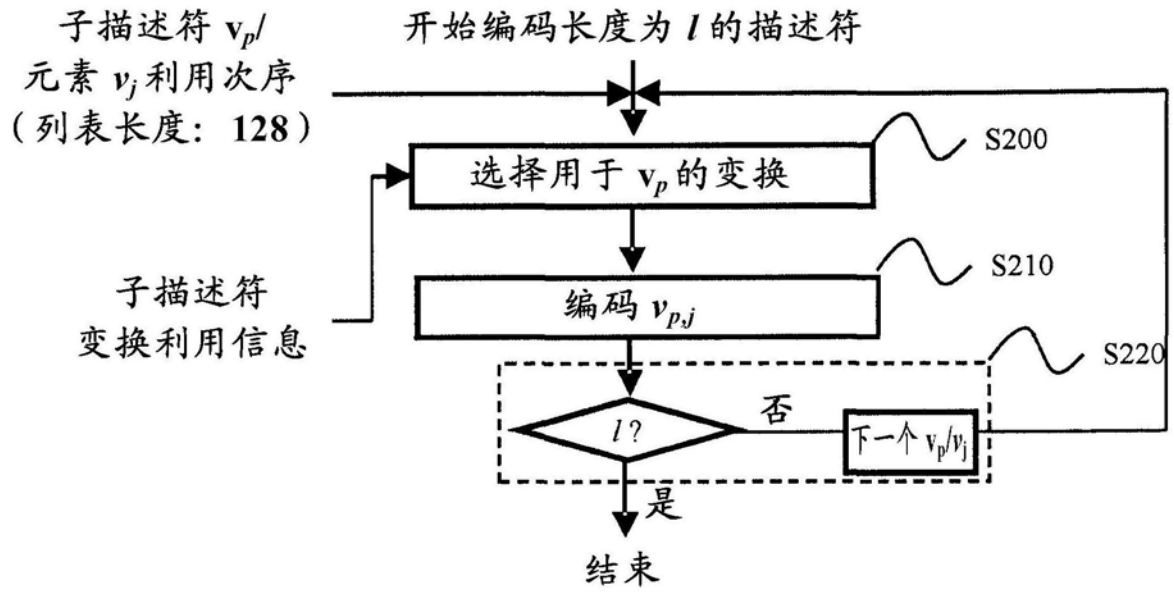


图7

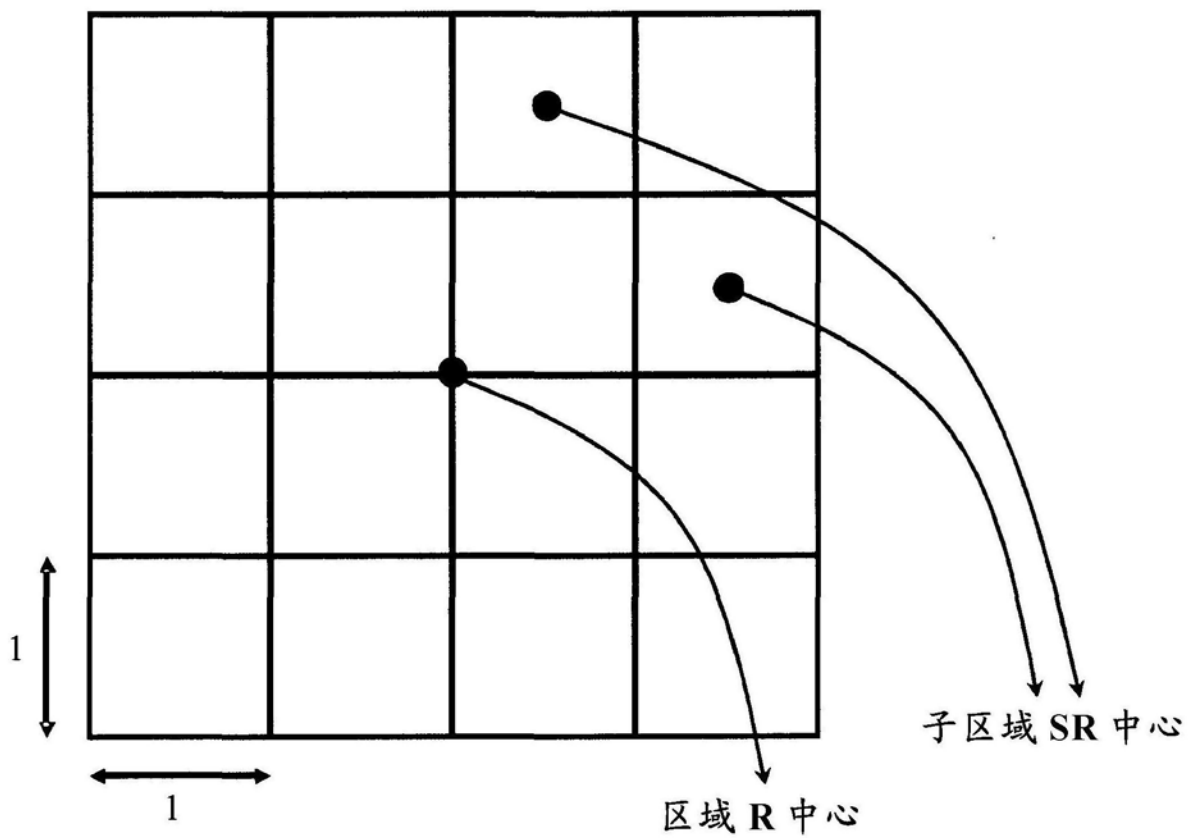


图8

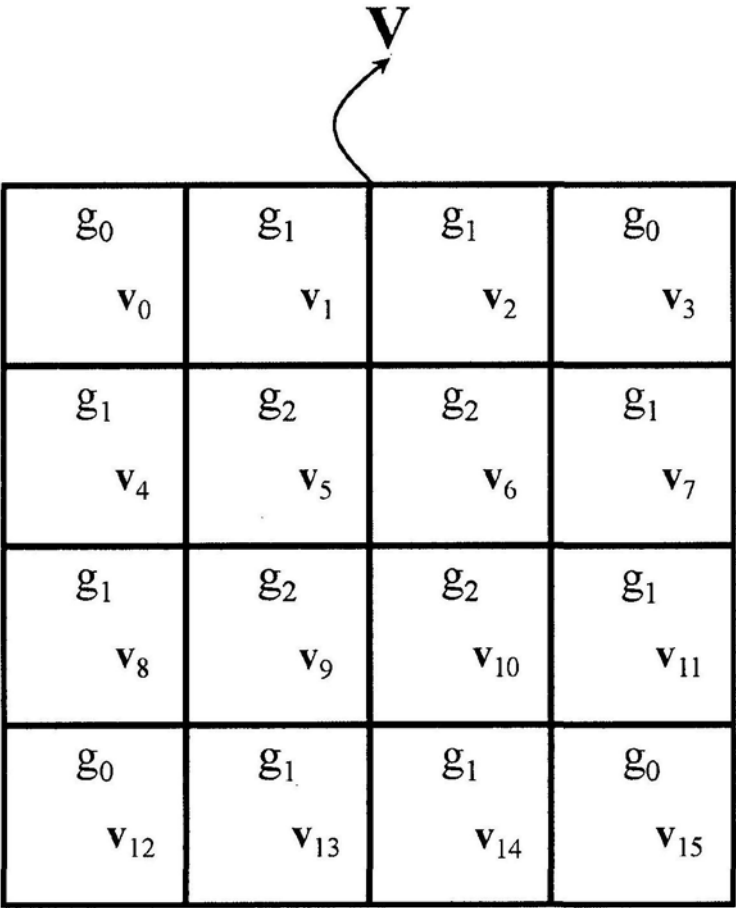


图9

优先级	组 g_x	元素 v_j
1	2	0
2	1	0
3	0	0
4	2	6
5	2	1
6	1	1
7	0	1
8	2	2
9	1	2
10	0	2
11	1	6
12	0	6
13	2	7
14	1	7
15	0	7
16	2	3
17	1	3
18	0	3
19	2	4
20	1	4
21	0	4
22	2	5
23	1	5
24	0	5

图10

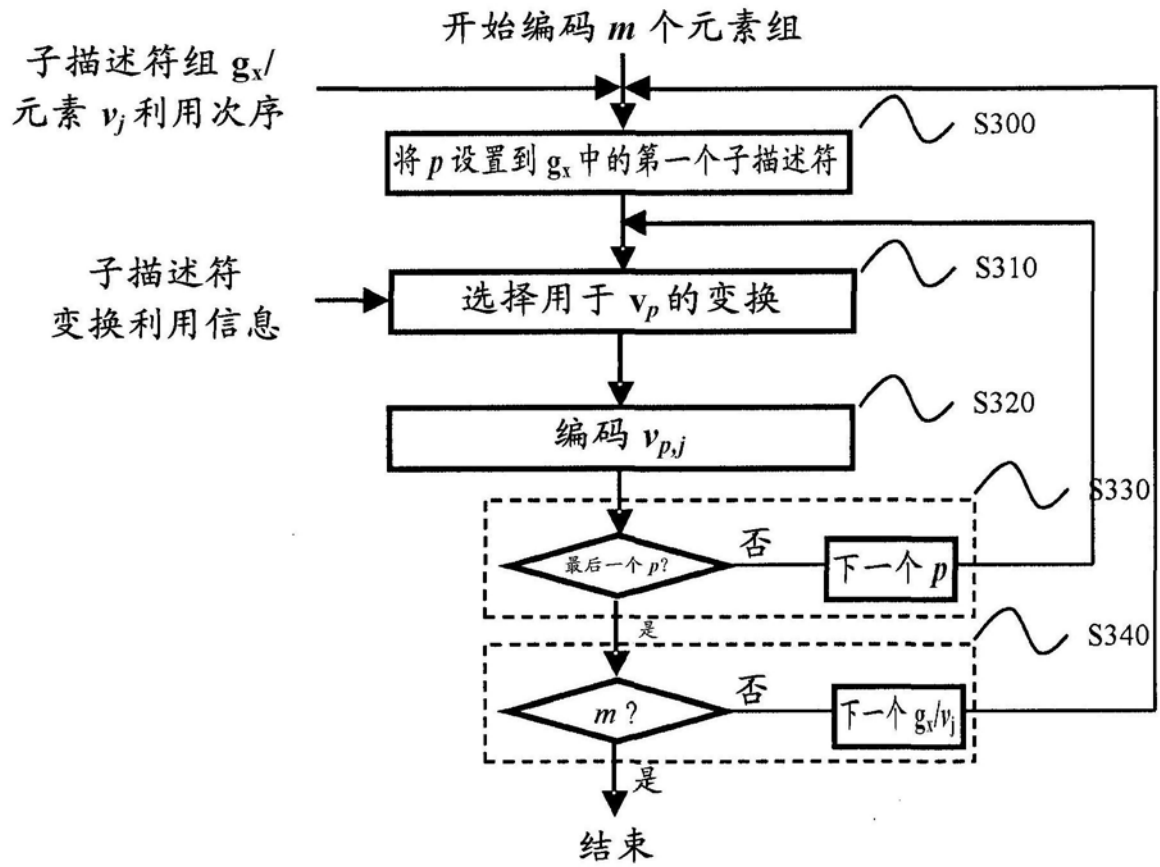


图11

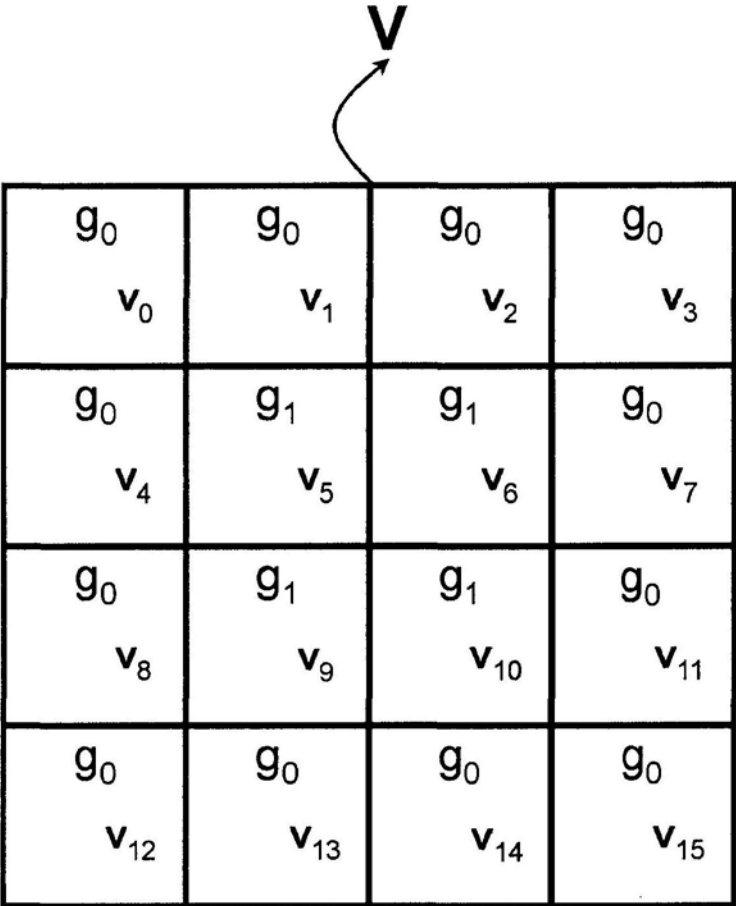


图12

优先级	组 g_i	元素 v_j
1	1	0
2	0	0
3	1	6
4	1	1
5	0	1
6	1	2
7	0	2
8	0	6
9	1	7
10	0	7
11	1	3
12	0	3
13	1	4
14	0	4
15	1	5
16	0	5

图13

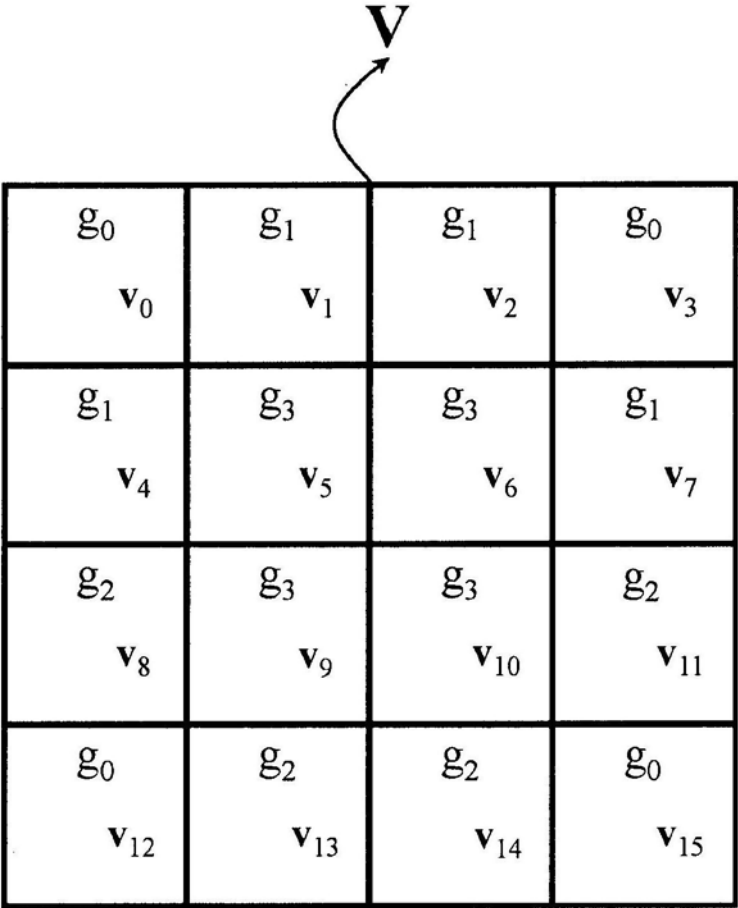


图14

优先级	组 g_i	元素 v_j
1	3	0
2	1	0
3	2	0
4	0	0
5	3	6
6	3	1
7	1	1
8	2	1
9	0	1
10	3	2
11	1	2
12	2	2
13	0	2
14	1	6
15	2	6
16	0	6
17	3	7
18	1	7
19	2	7
20	0	7
21	3	3
22	1	3
23	2	3
24	0	3
25	3	4
26	1	4
27	2	4
28	0	4
29	3	5
30	1	5
31	2	5
32	0	5

图15

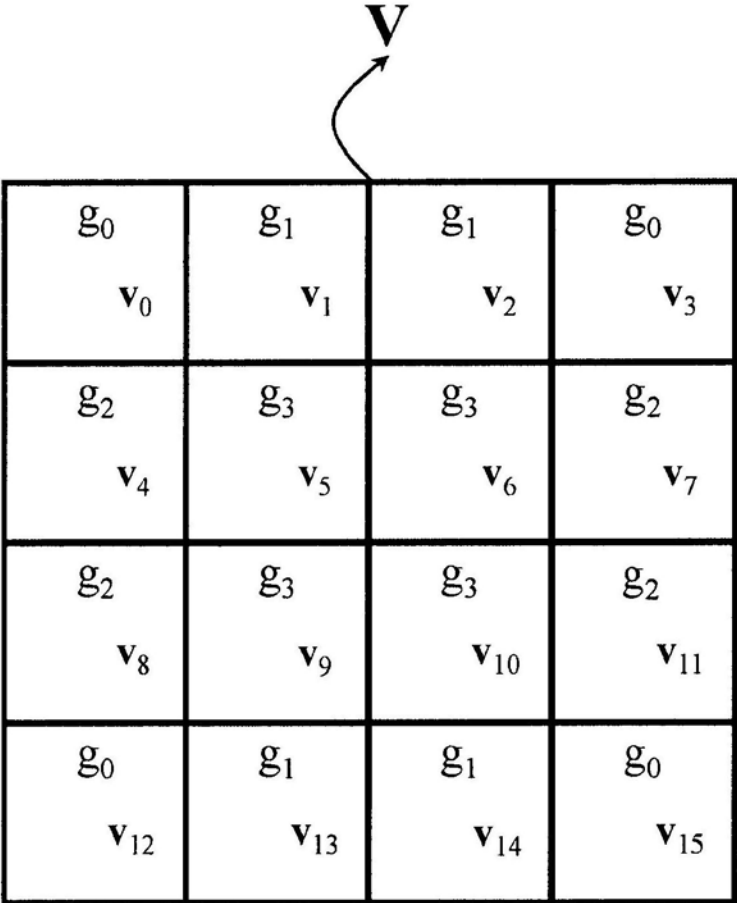


图16

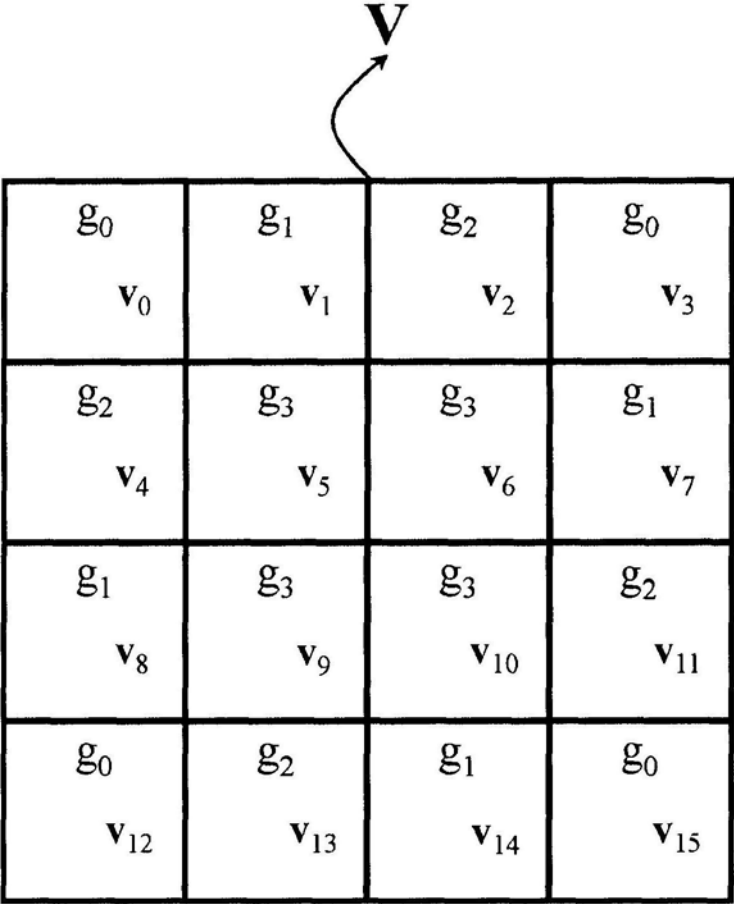


图17

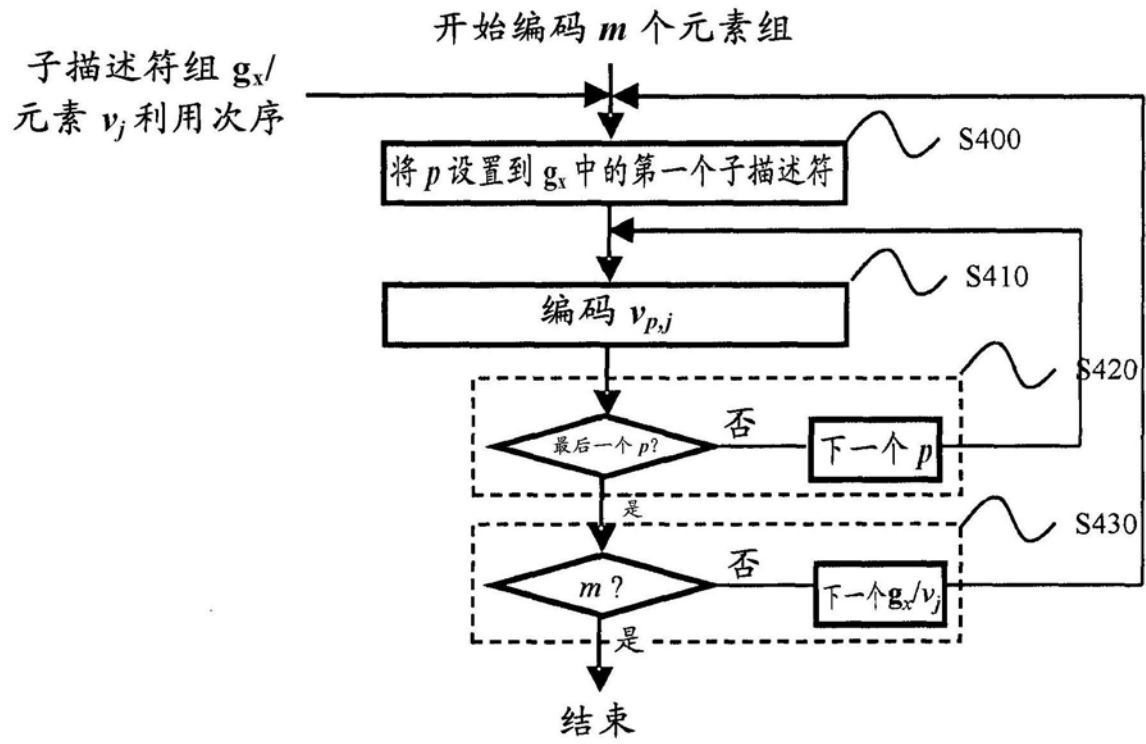


图18

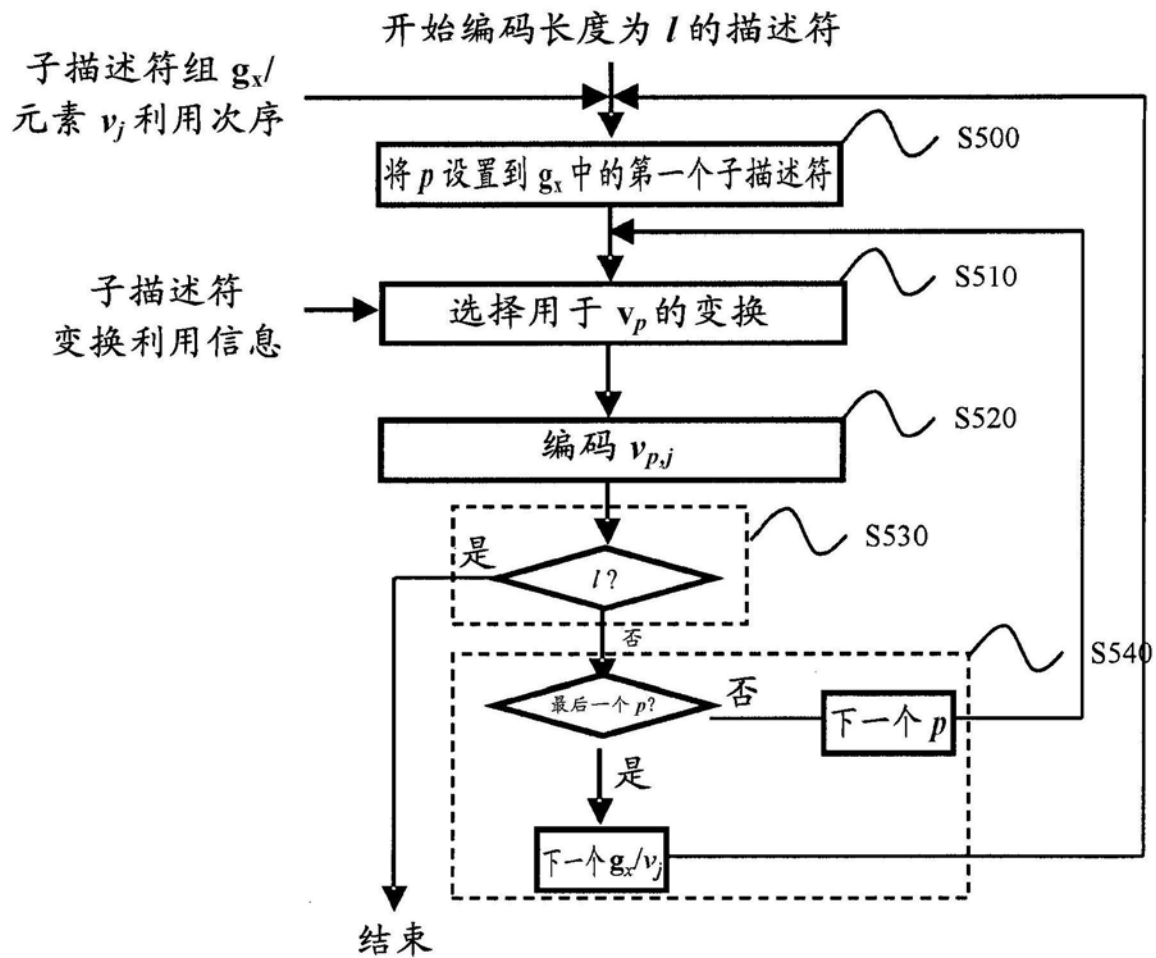


图19

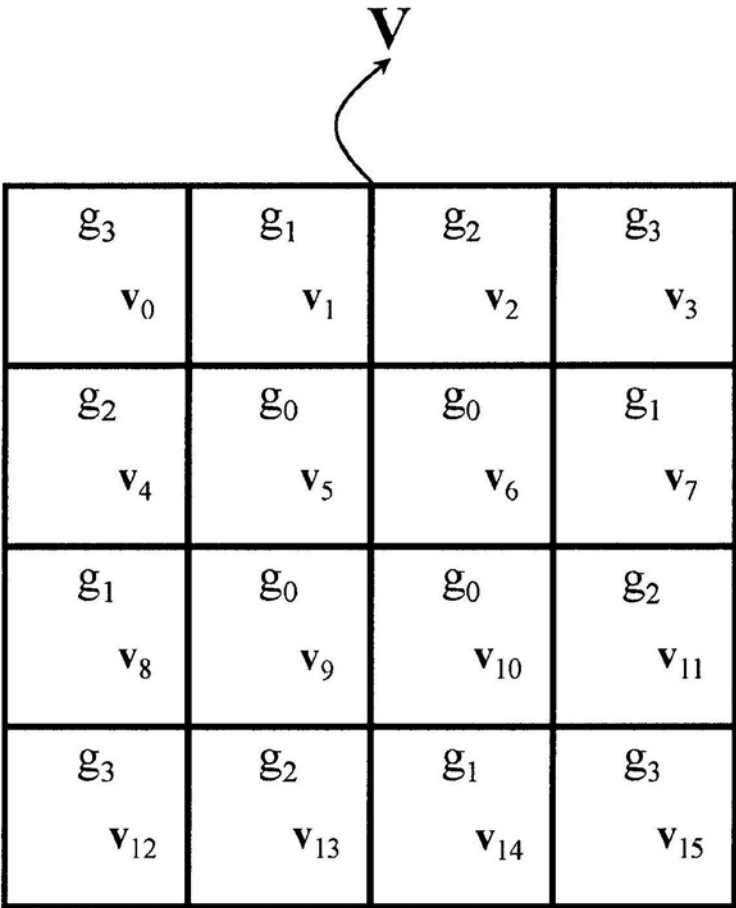


图20

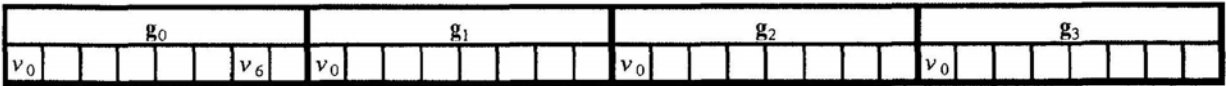


图21a

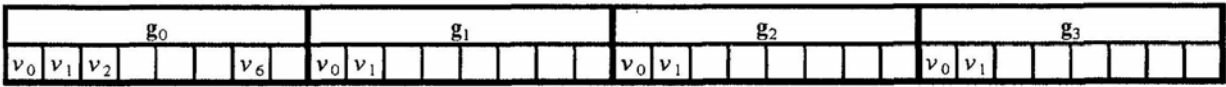


图21b

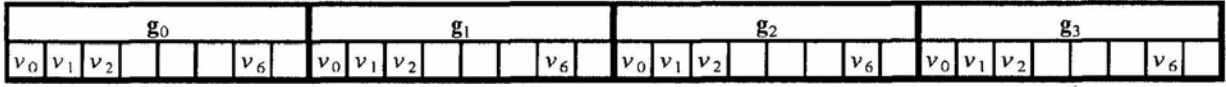


图21c

g ₀							g ₁							g ₂							g ₃						
v ₀	v ₁	v ₂					v ₆	v ₇	v ₀	v ₁	v ₂				v ₆	v ₇	v ₀	v ₁	v ₂							v ₆	v ₇

图21d

g ₀								g ₁								g ₂								g ₃							
v ₀	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	v ₀	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	v ₀	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	v ₀	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇

图21e

g ₀							g ₁							g ₂							g ₃						
v ₀						v ₆	v ₀							v ₀							v ₀						

图22a

g ₀							g ₁							g ₂							g ₃						
	v ₁	v ₂					v ₁							v ₁							v ₁						

图22b

g ₀							g ₁							g ₂							g ₃						
								v ₂				v ₆			v ₂			v ₆				v ₂				v ₆	

图22c

g ₀							g ₁							g ₂							g ₃						
						v ₇						v ₇						v ₇								v ₇	

图22d

g ₀							g ₁							g ₂							g ₃						
			v ₃	v ₄	v ₅				v ₃	v ₄	v ₅				v ₃	v ₄	v ₅							v ₃	v ₄	v ₅	

图22e

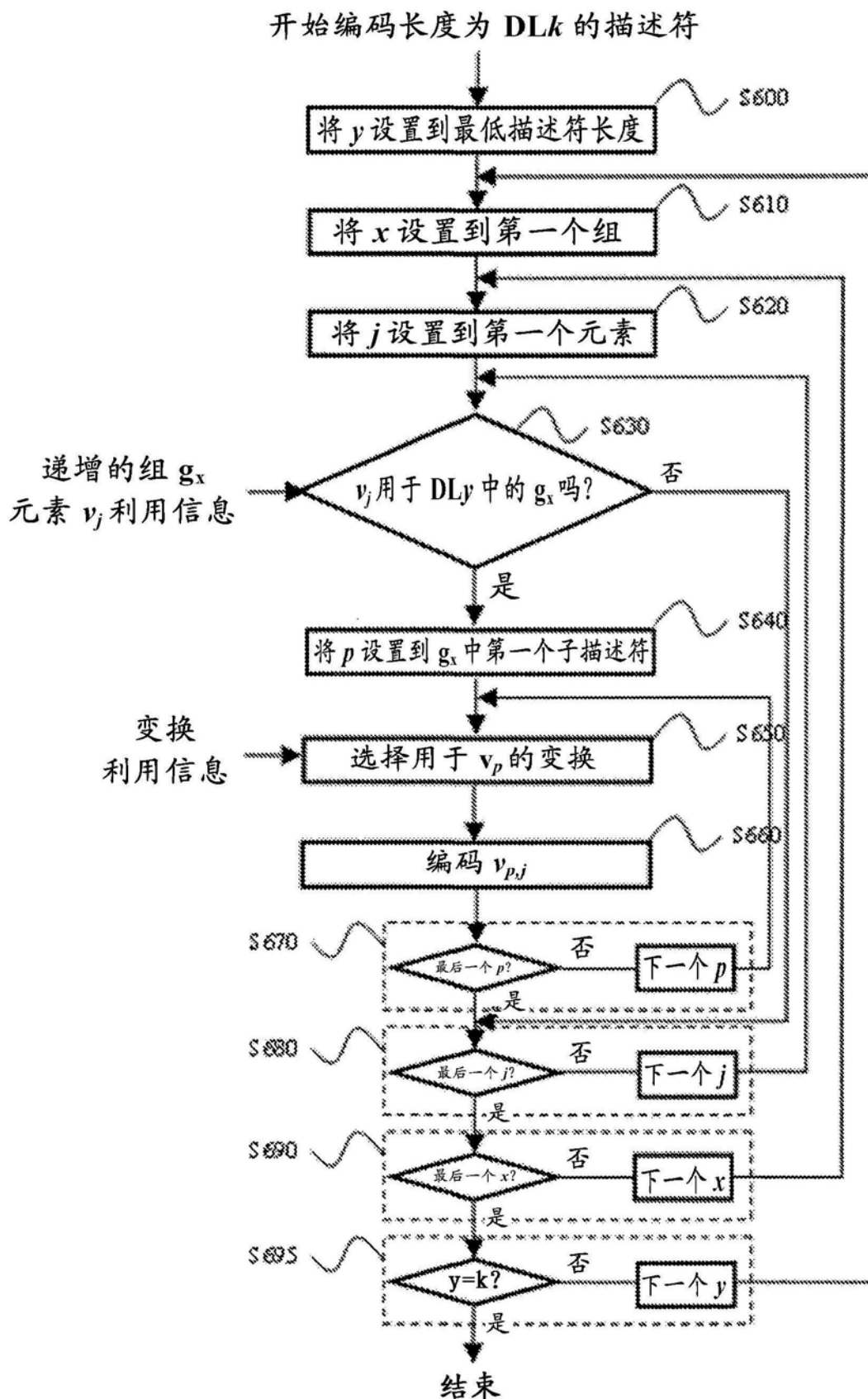


图23

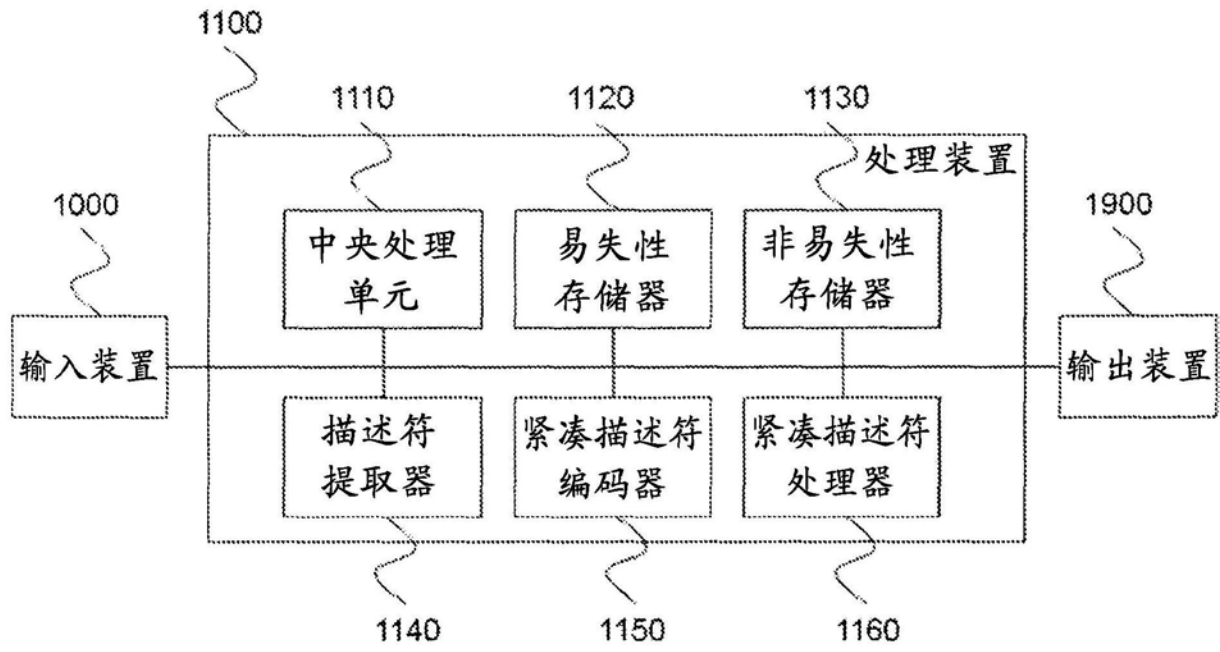


图24