



(12)发明专利申请

(10)申请公布号 CN 106231214 A

(43)申请公布日 2016.12.14

(21)申请号 201610655090.7

(22)申请日 2016.08.11

(71)申请人 复旦大学

地址 200433 上海市杨浦区邯郸路220号

(72)发明人 解玉凤 周乐成 徐徐然 马建国

(74)专利代理机构 上海正旦专利代理有限公司

31200

代理人 陆飞 陆尤

(51)Int.Cl.

H04N 5/374(2011.01)

H04N 19/176(2014.01)

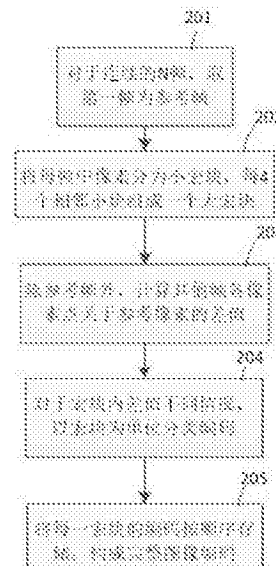
权利要求书2页 说明书5页 附图3页

(54)发明名称

基于可调宏块的高速CMOS传感器图像近似无损压缩方法

(57)摘要

本发明属于CMOS传感器技术领域,具体为一种基于可调宏块的高速CMOS传感器图像近似无损压缩方法。本发明方法包括:取高速CMOS传感器产生的N帧连续图像作为一组处理对象,并将第一帧作为参考帧;对每一帧图像采用自适应分块,以宏块为单位,在与参考帧作差后进行数据编码;对与参考帧近似相同的宏块,采用很短的标识码代替常规的无损编码;对其他与参考帧不相同的宏块,在差分编码后采用普通的无损压缩编码进行无损压缩。本发明方法能够保证数据压缩后有用信息不丢失;减小算法复杂度,提高算法执行速度;最大化压缩比,降低存储成本。



1. 一种基于可调宏块的CMOS高速传感器图像的近似无损压缩方法,其特征在于具体步骤为:

(1)取高速CMOS传感器产生的N帧连续图像作为一组处理对象,并将第一帧作为参考帧;

(2)对每一帧图像采用自适应分块,以宏块为单位,在与参考帧作差后进行数据编码;

(3)对与参考帧近似相同的宏块,采用很短的标识码代替常规的无损编码;

(4)对其他不相同的宏块,在差分编码后采用普通的无损压缩编码进行无损压缩。

2. 根据权利要求1所述的方法,其特征在于,所述的自适应分块包括:

对每一帧图像,划分为大小为 $m*m$ 的小宏块,每4个相邻的小宏块组成一个大小为 $(2m)*(2m)$ 的大宏块,以宏块为单位进行像素值的编解码;

除参考帧外,其他帧内的每一宏块,都有参考帧内相同位置的宏块与之对应,称为该宏块的参考宏块;

除参考帧外,其他帧内的每一个像素点,都有参考帧内相同位置的像素与之对应,称为该像素点的参考像素;

根据大小宏块内各像素点与参考像素的差值情况,判断以大宏块为单位编码,或以小宏块为单位编码。

3. 根据权利要求2所述的方法,其特征在于,所述的以宏块为单位进行的编码包括:

采用差分编码方式进行帧间压缩,即求取每一像素点与其参考像素的差值,用二进制表示;

计算差分编码后小宏块内的像素差值之和,若差值之和小于所设阈值,则认为该小宏块与其参考宏块近似相同,利用近似无损压缩的思想,用很短的标记位表示该宏块的编码,即在解码时该小宏块可直接用其参考宏块替代;

若相邻4个小宏块都与参考宏块近似相同,其组成的大宏块也与参考宏块相同,用一不同的标记位对于整个大宏块进行编码,即在解码时该大宏块直接用其参考宏块替代,进一步增大压缩比率;

对于与参考宏块不相同的小宏块,则采用无损压缩编码对其内部像素差值进行压缩,其中无损压缩编码的码字不包含大小宏块的标记位;

若相邻4个小宏块都与参考宏块不相同,则对大宏块整体采用无损压缩编码对其内部像素差值进行压缩。

4. 根据权利要求3所述的方法,其特征在于,所述阈值应使背景区域的识别度最大,利用先验信息确定;或者通过下述方法确定:使用传感器拍摄已知为静态背景的图像,并按权利要求3中所述方法统计各宏块中像素差值之和,选取一尽可能小的值,使该值大于各宏块内像素差值之和,作为设定阈值。

5. 根据权利要求3所述的方法,其特征在于,对图像参考帧直接进行无损压缩编码,为便于解码,参考帧以宏块为单位进行编码,宏块内部从左至右,从上至下编码,帧内按宏块顺序,可按照4个小宏块构成一个大宏块,大宏块则遵循从左至右,从上至下的编码顺序。

6. 一种基于可调宏块的CMOS高速传感器图像的近似无损解压缩方法,其特征在于,步骤为:

(1)首先对参考帧进行解码,还原出各参考像素值;

(2)对其后每一帧,按宏块顺序解码,当读取到标记位表示大宏块相同时,直接将该位置处编码还原成参考帧对应的大宏块,由参考帧对应的4个小宏块组成;

(3)当读取到标记位表示小宏块相同时,直接将该位置处编码还原成参考帧对应的小宏块;

否则按正常步骤对宏块内像素进行哈夫曼解码,还原出宏块内每一像素点与参考像素的差值,并根据参考像素值还原出实际像素值;

(4)对每一帧内的编码进行上述处理,得到的解压结果以小宏块为单位顺序排列,在最终解码输出前需要重新调整各像素点位置,再将每一帧按顺序输出,得到连续的图像流。

## 基于可调宏块的高速CMOS传感器图像近似无损压缩方法

### 技术领域

[0001] 本发明属于CMOS传感器技术领域,具体涉及一种对高帧频传感器图像的近似无损压缩方法。

### 背景技术

[0002] 近几年来,随着CMOS传感器的高速发展与广泛应用,各相关领域大量应用高帧频CMOS相机采集图像数据。目前的高速CMOS传感器拍摄速度最高可以达到1000fps,传统的数据处理以及存储方法已无法满足其应用的需求。按照每帧50万像素,每像素点12bit数据计算,在传感器高速拍摄时,每秒平均将会有750MB的信息输出,若不进行处理,对如此庞大数据流的处理和存储会非常困难。CMOS传感器特殊的应用领域又要求记录所有关键数据,在普通的有损压缩会导致信息丢失的情况下,迫切地需要一种可靠、快速的无损或近似无损压缩方法,解决传感器输出数据吞吐量太大无法实时处理或存储的问题。

[0003] 目前已有针对CMOS传感器图像提出的压缩算法,中国专利(申请号201310530804.8)提出了一种CMOS图像传感器像素值的无损压缩方法。在一帧内选取参考像素和邻近像素,对当前像素求差值以减少存储位数。对于单像素12bit量化的CMOS传感器图像,存储像素差值仅需要6bit量化,实现了图像压缩。但该算法采用模拟减法器作差,差值需要额外的AD进行量化,增加了压缩成本;对差值采用低精度AD压缩,存在丢失信息的风险。另外由于算法仅实现帧内压缩,对于高帧频图像没有充分利用帧间冗余度高的特性,压缩率较低。

[0004] 在《基于CMOS图像传感器LUPA300的高速摄像头关键技术的研究》(合肥工业大学硕士学位论文,2012)一文中,提出了一种对高速CMOS传感器图像的压缩方法。在N帧内将相邻图像帧作差,再对差值进行动态哈夫曼压缩编码的方法。在解码时仅需要解码完整的第一帧,其后的所有帧按照记录的差值恢复。但采用相邻帧作差再压缩的方法,在图像噪声存在的情况下,难以对非零像素差值进行高比例压缩,依然会存储大量冗余信息,LUPA300传感器的帧频为250fps,对于更高帧频的传感器,帧间数据的冗余更为明显,二进制差值数据将带有大量连续的0值。另外在传感器图像中,常见大部分为静止背景,仅有部分区域运动的情况,对于这类图像若对背景区域和动态区域同样处理,在图像存在噪声干扰或光照变化的情况下,无法充分利用背景区域的帧间冗余,难以达到最佳压缩效果。

[0005] 在《高帧频数字图像序列实时压缩算法》(大连理工大学学报,2009,第49卷第3期)一文中,提出了一种自适应近无损压缩算法,对连续N帧图像的第一帧进行基于帧内感兴趣区域的压缩,其他帧在帧间差分阈值处理后进行游程编码。该算法采用3级形态Harr小波变换,并检测高频子带中的感兴趣区域,但分解3层小波变换后,需对每一子带分别进行编码,计算量大,不利于高速传感器图像的实时处理;根据传感器应用条件不同,感兴趣区域的检测可能出现不准确的问题,在其他区域采用有损压缩可能导致关键数据丢失。

[0006] 国外文献也提出了相关的传感器数据压缩方法,*A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and*

*Time-Domain CDS*一文中,提出了一种使用动态框架的方法,仅识别图像中动态的部分,对于其他非动态部分仅存储初始帧。但该算法要求改变传感器本身的结构,需要增加相应的动态监测结构,增加了硬件成本,且该传感器帧频无法达到很高。

[0007] *An Adaptive DPCM Method for Efficient Data Compression in Aerospace Sensor Systems*一文中,提出了一种自适应DPCM编码的方法,对于不同的差值范围采用不同位数进行编码,使差分编码的冗余度最小。但该方法增加了状态机,对单一像素可能需要多次判断,增加了编码的复杂度,并且该算法对差值采用不同位数存储,大大增加了解码的复杂度;另外该算法对于高帧频信号的帧间压缩率不够高,由于算法依然对整帧采用无损压缩的方法,对于图像中的背景区域,由于噪声、光照等因素的存在,DPCM编码在这些区域仍会产生大量非零差值,每一像素至少用4bit进行编码存储,无法获得最佳压缩比例。

[0008] 因此,有必要提出一种针对高帧频的图像流 充分利用冗余性提高压缩比、满足高帧频拍照通常不希望丢失数据、处理速度要求高的无损压缩算法。

## 发明内容

[0009] 针对现有技术中存在的不足,本发明提供一种对CMOS高速传感器图像进行近似无损压缩的方法,能够对传感器图像进行即时压缩,满足高帧频CMOS传感器的图像存储需求,同时不丢失任何有用信息,以便于后期数据处理。

[0010] 本发明主要为实现三个目标:保证数据压缩后有用信息不丢失;减小算法复杂度,提高算法执行速度;最大化压缩比,降低存储成本。由于高帧频CMOS传感器帧间冗余信息较多,对于一些帧间基本无变化的部分,往往是图像中的静态背景,可以取较大压缩比率。而相对于一些复杂的关注区域判别方法,本发明所提出的算法直接以自适应宏块为单位对数据分类处理,大大减小了算法复杂度,优势在于平衡了算法执行速度和压缩比。

[0011] 本发明提出的针对CMOS高速传感器图像的近似无损压缩方法,具体步骤为:

(1)取高速CMOS传感器产生的N帧连续图像作为一组处理对象,并将第一帧作为参考帧;

(2)对每一帧图像采用自适应分块,以宏块为单位,在与参考帧作差后进行数据编码;

(3)对与参考帧近似相同的宏块,采用很短的标识码代替常规的无损编码;

(4)对其他与参考帧不相同的宏块,在差分编码后采用普通的无损压缩编码进行无损压缩。

[0012] 本发明中,步骤(2)中,所述的自适应分块包括:

对每一帧图像,划分为大小为 $m*m$ 的小宏块,每4个相邻的小宏块组成一个大小为 $(2m)*(2m)$ 的大宏块,以宏块为单位进行像素值的编解码;

除参考帧外,其他帧内的每一宏块(无论大小),都有参考帧内相同位置的宏块与之对应,称为该宏块的参考宏块;

除参考帧外,其他帧内的每一个像素点,都有参考帧内相同位置的像素与之对应,称为该像素点的参考像素;

根据大小宏块内各像素点与参考像素的差值情况,判断以大宏块为单位编码,或以小宏块为单位编码。

[0013] 本发明中,所述的以宏块为单位进行的编码包括:

采用差分编码方式进行帧间压缩,即求取每一像素点与其参考像素的差值,可以用二进制表示;

计算差分编码后小宏块内的像素差值之和,若差值之和小于所设阈值,则认为该小宏块与其参考宏块近似相同,利用近似无损压缩的思想,仅用很短的标记位表示该宏块的编码,即在解码时该小宏块可直接用其参考宏块替代;

所述阈值应使背景区域识别度最大,可利用先验信息确定;一种确定阈值的方法为,使用传感器拍摄已知为静态背景的图像,并按前述方法统计各宏块中像素差值之和,选取一尽可能小的值,使该值大于各宏块内像素差值之和,作为设定阈值;

若相邻4个小宏块都与参考宏块近似相同,其组成的大宏块也与参考宏块相同,可用一不同的标记位对于整个大宏块进行编码,即在解码时该大宏块可直接用其参考宏块替代,进一步增大压缩比率;

对于与参考宏块不相同的小宏块,则采用无损压缩编码对其内部像素差值进行压缩,其中无损压缩编码的码字不能包含大小宏块的标记位;

若相邻4个小宏块都与参考宏块不相同,则对大宏块整体采用无损压缩编码对其内部像素差值进行压缩。

[0014] 最优地,仅需使用2位标记位,分别表示小宏块近似相同,小宏块不同,大宏块近似相同,大宏块不同,可同步进行解码。

[0015] 本发明中,对图像参考帧直接进行无损压缩编码,为便于解码,参考帧以宏块为单位进行编码,宏块内部从左至右,从上至下编码,帧内按宏块顺序编码,可按照4个小宏块构成一个大宏块,大宏块则遵循从左至右,从上至下的编码顺序;

对除参考帧外,每一帧内所有宏块进行前述的差分编码,编码的顺序应与参考帧相同,直至该组处理对象内各帧全部编码完成;

对于其他各组处理对象,也采用相同的压缩算法进行编码。

[0016] 本发明还提出对应上述的压缩方法的近似无损解压缩方法,具体步骤为:

(1)首先对参考帧进行解码,还原出各参考像素值;

(2)对其后每一帧,按宏块顺序解码,当读取到标记位表示大宏块相同时,直接将该位置处编码还原成参考帧对应的大宏块,由参考帧对应的4个小宏块组成;

(3)当读取到标记位表示小宏块相同时,直接将该位置处编码还原成参考帧对应的小宏块;

否则按正常步骤对宏块内像素进行哈夫曼解码,还原出宏块内每一像素点与参考像素的差值,并根据参考像素值还原出实际像素值;

(4)对每一帧内的编码进行上述处理,得到的解压结果以小宏块为单位顺序排列,在最终解码输出前需要重新调整各像素点位置,再将每一帧按顺序输出,得到连续的图像流。

[0017] 本发明方法实现了预期的目标:保证数据压缩后有用信息不丢失;减小算法复杂度,提高算法执行速度;最大化压缩比,降低存储成本。

## 附图说明

[0018] 图1 为本发明提出的高速CMOS传感器图像近似无损压缩算法的应用结构示意图。

[0019] 图2为本发明提出的近似无损压缩方法的总体流程图。

[0020] 图3为本发明提出的可适应宏块的编码结构示意图。

[0021] 图4 为本发明提出的可适应宏块的编码分支流程图。

[0022] 图5为本发明提出的近似无损解压缩算法的流程图。

## 具体实施方式

[0023] 下面结合实施例对本发明的具体实施方式做进一步的详细描述。

[0024] 图1为本发明提出的近似无损压缩方法在实际应用中的结构示意图,由高速CMOS传感器产生的图像数据101,直接作为本发明提出的近似无损压缩算法102的输入,经过压缩后进入存储介质103中,可大大节省存储空间,降低存储设备成本;在需要对图像数据进行处理或预览时,可将压缩数据通过相应的解压缩算法104还原成原始图像数据。

[0025] 图2为本发明提出的高速CMOS传感器图像近似无损压缩算法的总体流程图,本算法利用高帧频传感器图像较大的帧间冗余性,分宏块判断图像中某一区域相对参考帧的像素变化情况,若像素几乎不变,认为该区域为静态背景,不影响重要信息的处理,可用参考帧的对应区域替代,故采用近似无损压缩,不保留该宏块部分的像素信息,以获取较大的压缩比率;而对于像素变化较大的区域,采用差分后无损压缩编码的方式,避免主要数据的丢失。

[0026] 对于从CMOS传感器得到的图像数据,首先取出连续的N帧作为一组处理对象,取其中第一帧作为参考帧(201);

对每一帧图像进行分块,先将图像分为小宏块,再由相邻的4个小宏块构成大宏块(202),使用大宏块可以进一步增大压缩比率,但小宏块可以使静态和动态区域的划分更精确,不同的宏块大小构成自适应宏块的无损压缩算法;

需要指出的是,本实施例中采用由4个小宏块构成一个大宏块的结构,本领域普通技术人员无需创造性劳动也可以将该结构扩展至由其他数量的小宏块构成大宏块的结构;

将除参考帧外的每一帧均与参考帧相减,得到各像素点关于参考像素的差值(203);

根据宏块内像素差值的不同情况,分类对宏块内数据进行编码(204),编码的分类将在后一实施例阐述;

最后将每一宏块的编码结果按顺序输出,构成完整的图像编码结果(205)。

[0027] 图3为本算法所述可适应宏块的编码结构示意图,分(a)、(b)、(c)三个子图,分别表示3种不同情况下的宏块编码结构;

对于一组处理对象(301),取其中一非参考帧(302),将其划分为小宏块,相邻小宏块构成大宏块,本实施例中302内带有标号的为大宏块,由4个小宏块构成,取其中的一个大宏块,其内部小宏块的顺序如303中编号所示;

根据小宏块内像素差值的不同情况,判定宏块与参考宏块是否近似相同,并对宏块内部数据进行编码并组合,本实施例中,以两位二进制位作为标示位:第一位表示以大宏块或小宏块为单位进行编码,“0”为小宏块,“1为大宏块”;第二位表示宏块是否近似相同,“0”表示相同,“1”表示不同;

以图(a)为例,303中编号为1、4的两个小宏块与参考宏块不相同,而2、3两个小宏块与参考宏块近似相同,故第2、3个宏块直接以标示位“00”作为整块的编码,而1、4两个小宏块增加标示位“01”,其后对宏块内差值进行无损压缩编码,紧跟于标示位后,将各小宏块编码

按宏块顺序输出,即该大宏块303的编码结果如304所示;

以图(b)为例,303中4个小宏块均与参考宏块不相同,此时可知整个大宏块与参考宏块不相同,直接对大宏块内数据整体采用无损压缩编码,并在编码前增加标示位“11”,即该大宏块的编码结果如304所示;

以图(c)为例,303中4个小宏块均与参考宏块近似相同,此时可知整个大宏块与参考宏块近似相同,直接以标示位“10”作为整个大宏块303的编码,如304所示;

以自适应宏块进行编码可获得很大的压缩比率,以1000fps,每帧50万像素,每像素点12bit量化数据的高速CMOS传感器的输出图像为例,每秒产生750MB的数据;采用每20帧为一组处理对象,小宏块4\*4大小,大宏块8\*8大小,若图像中有75%左右的部分为静态背景,假设平均每帧中有4000个大宏块,8000个小宏块可直接以标示位编码,使用基于自适应宏块的近似无损压缩后每秒产生的数据量可按下式计算:

$$\{500000 \times 12 + 19 \times [(8000 + 4000) \times 2 + 116000 \times 12]\} \times 50 / 8 = 206 \text{ MB},$$

压缩量超过3.5倍,在未计算无损压缩算法产生的效果时,已有较高的压缩率。

[0028] 图4为本算法所述可适应宏块编码的分支流程图,该实施例作为前一实施例的补充,说明了判断某一宏块采用哪种编码类型的过程,需要首先计算帧内各像素与参考像素的差值;

统计小宏块中所有像素差值之和,(401),若该值小于所设定的阈值,则认为该小宏块和参考宏块近似相同(403),否则认为该小宏块和参考宏块不同(402);

所述阈值的一种确定方法为,使用传感器拍摄已知为静态背景的图像,并按前述方法统计各宏块中像素差值之和,选取一尽可能小的值,使该值大于各宏块内像素差值之和,作为设定阈值,该方法相当于利用先验信息,得到能使背景区域识别度最大的阈值;

在判断完能够成大宏块的相邻4个小宏块的情况后,判断是否需要以大宏块为单位进行编码,若4个小宏块均与参考宏块不同,则以大宏块为单位对其内像素进行无损压缩编码(407);若4个小宏块均与参考宏块近似相同,则以标示位直接表示整个大宏块的编码(408);否则以小宏块为单位进行编码,认为相同的宏块以标示位直接表示(409),不同的宏块采用无损压缩编码(406)。

[0029] 图5为本算法所述近似无损解压算法的流程图,旨在说明如何对压缩后的编码进行解压,首先解压出参考帧,还原成像素形式(501),对其后各帧,首先将读取某一宏块的标示位,可判断出该宏块是否与参考宏块是否相同,若是,则直接译码为参考宏块像素值;否则对该宏块之后的压缩编码进行译码,直至读取到下一宏块的标示位;

对一帧宏块完成上述解码后,帧内像素按宏块顺序排列,根据需要使用,可以对像素输出顺序进行调整,还原为正常顺序(504)。



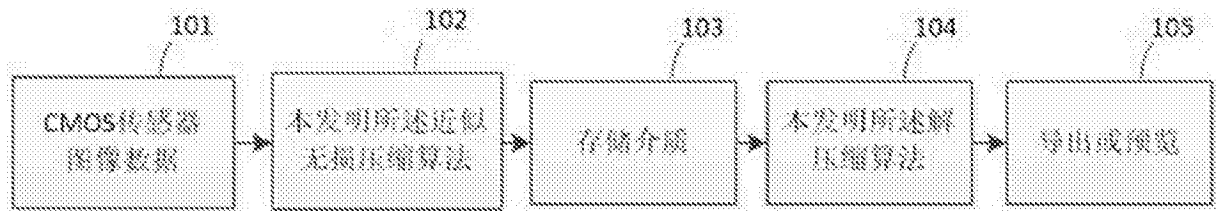


图1

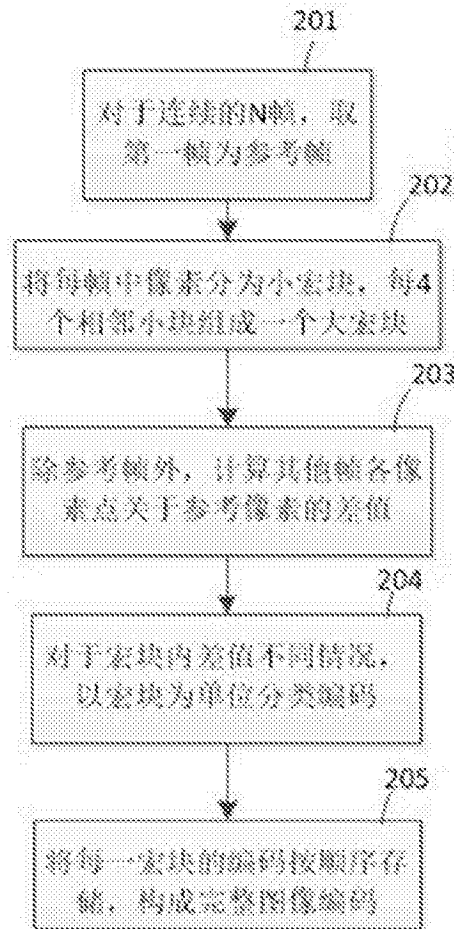


图2

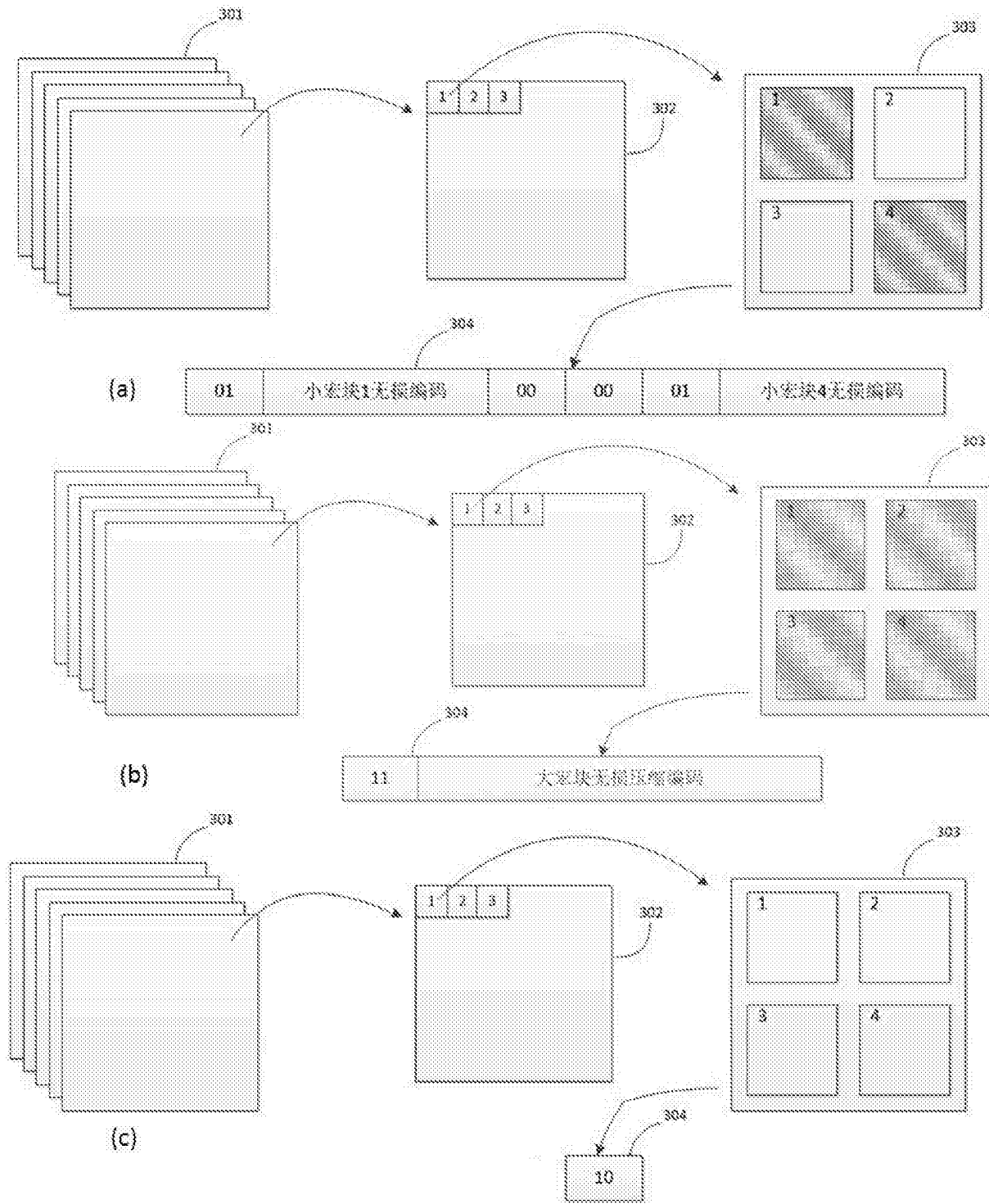


图3

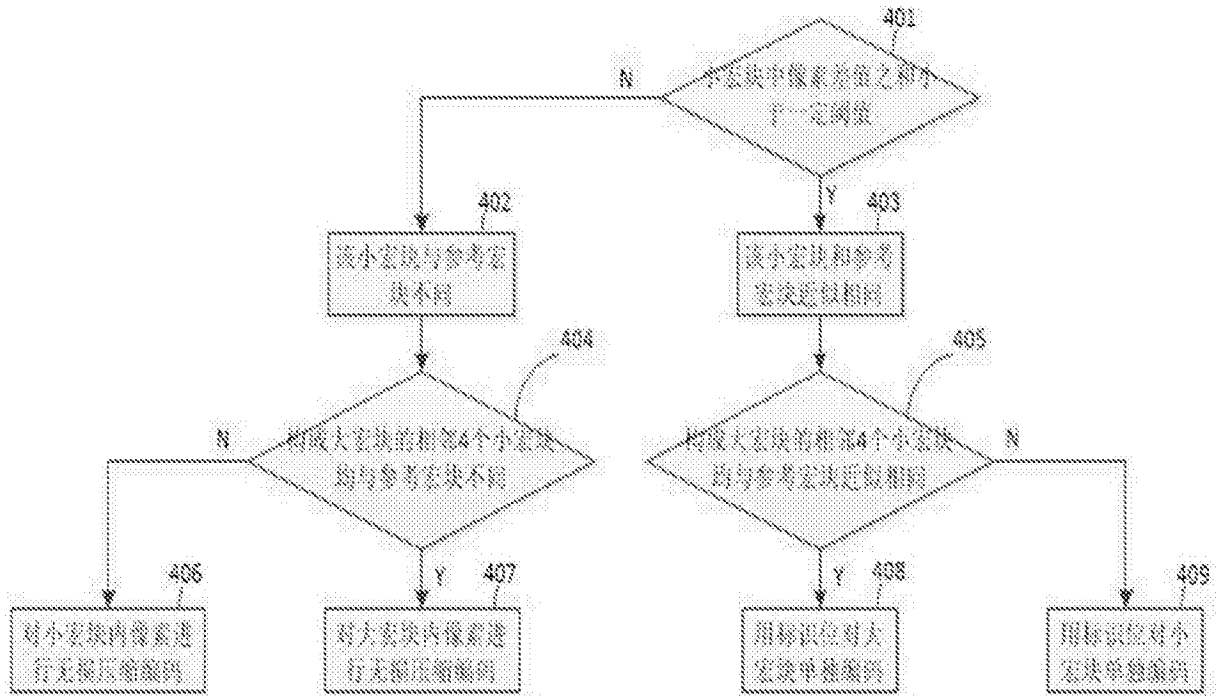


图4

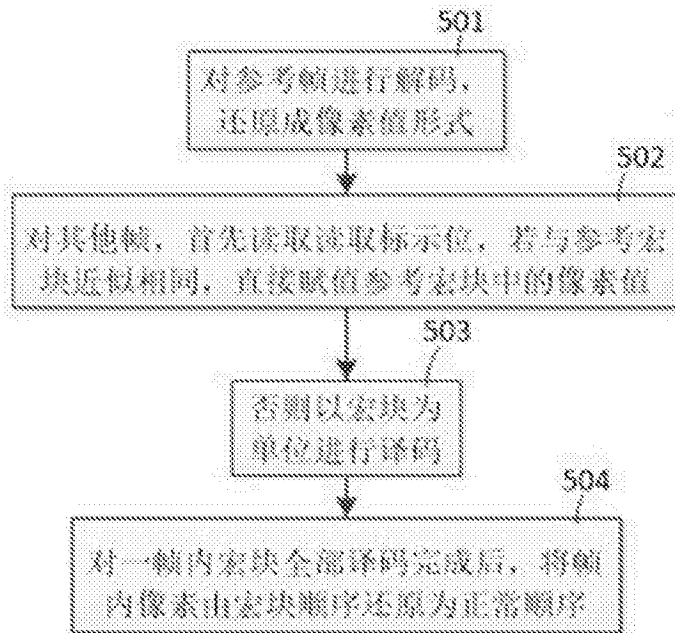


图5