



# (12) 发明专利

(10) 授权公告号 CN 109993693 B

(45) 授权公告日 2023.04.25

(21) 申请号 201711473491.1  
(22) 申请日 2017.12.29  
(65) 同一申请的已公布的文献号  
申请公布号 CN 109993693 A

(43) 申请公布日 2019.07.09

(73) 专利权人 澜至电子科技(成都)有限公司  
地址 610200 四川省成都市双流区东升街  
道成都芯谷产业园区集中区内

(72) 发明人 贺梦驰 孙滨璇 吴可 张家祯

(74) 专利代理机构 北京市君合律师事务所  
11517

专利代理师 毛健 杜小锋

(51) Int. Cl.  
G06T 3/40 (2006.01)

(56) 对比文件  
CN 102638679 A, 2012.08.15  
CN 104700361 A, 2015.06.10

CN 104700360 A, 2015.06.10  
CN 101216935 A, 2008.07.09  
CN 106204453 A, 2016.12.07  
CN 104794692 A, 2015.07.22  
JP 2009182733 A, 2009.08.13  
JP 2013218523 A, 2013.10.24  
MJ Chen.et al. "A fast edge-oriented algorithm for image Interpolation".《Image and Vision Computing》.2005, 第23卷(第9期),  
V Patel.et al. "A review on different image interpolation techniques for image enhancement".《Journal of emerging technology and advanced》.2013,  
张亚等. "一种有效的数字图像传感器彩色插值算法".《计算机测量与控制》.2008, (第04期),  
刘晓松等. "基于统计特征的彩色图像快速插值方法".《电子学报》.2004, (第01期),

审查员 宋亚静

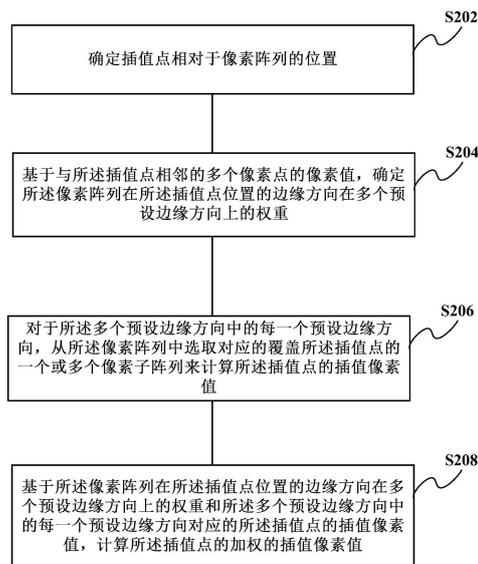
权利要求书3页 说明书9页 附图7页

## (54) 发明名称

用于对图像进行插值的方法和装置

## (57) 摘要

本申请公开了一种用于对图像进行插值的方法,所述图像包括由多个像素点组成的像素阵列,所述方法包括:确定插值点相对于所述像素阵列的位置;基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向在多个预设边缘方向上的权重;对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的覆盖所述插值点的一个或多个像素子阵列来计算所述插值点的插值像素值;以及基于所述像素阵列在所述插值点位置的边缘方向在多个预设边缘方向上的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值,计算所述插值点的加权的插值像素值。



CN 109993693 B

1. 一种用于对图像进行插值的方法,所述图像包括由多个像素点组成的像素阵列,其特征在于,包括:

确定插值点相对于所述像素阵列的位置;

基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重;

对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的一个或多个像素子阵列来计算所述插值点的插值像素值,其中所述一个或多个像素子阵列覆盖所述插值点;以及

基于所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值,计算所述插值点的加权的插值像素值。

2. 根据权利要求1所述的方法,其特征在于,基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重包括:

将与所述插值点相邻的多个像素点划分为多个像素单元;

计算所述多个像素单元中每个像素单元的边缘方向在所述多个预设边缘方向上的权重分量;

基于所述多个像素单元中每个像素单元的边缘方向在所述多个预设边缘方向上的权重分量,确定所述像素阵列在所述插值点位置的边缘方向沿所述多个预设边缘方向的权重。

3. 根据权利要求2所述的方法,其特征在于,所述像素单元的边缘方向在所述多个预设边缘方向上的权重分量与所述像素单元相对所述插值点位置的距离、以及所述像素单元的边缘方向的剧烈程度相关。

4. 根据权利要求1所述的方法,其特征在于,所述多个预设边缘方向包括 $-45^{\circ}$ 方向、 $0^{\circ}$ 方向、 $45^{\circ}$ 方向和 $90^{\circ}$ 方向。

5. 根据权利要求1所述的方法,其特征在于,对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的一个或多个像素子阵列的边界构成平行四边形,并且所述平行四边形的第一组相对边与所述预设边缘方向平行。

6. 根据权利要求5所述的方法,其特征在于,对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的一个或多个像素子阵列来计算所述插值点的插值像素值,包括:

在每个所述像素子阵列中,沿与所述平行四边形的第一组相对边不同的第二组相对边的方向,计算每一行或列的像素在与所述插值点位置对应的位置的插值像素值,其中与所述插值点位置对应的位置是指沿所述预设边缘方向穿过所述插值点的直线与所述行或列的交点的位置;

基于所述每一行或列的像素在与所述插值点位置对应的位置的插值像素值,计算所述插值点的插值像素值。

7. 根据权利要求1所述的方法,其特征在于,基于所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值,计算所述插值点的加权的插值像素值,包括:

基于所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重计算与每一个像素子阵列对应的插值像素值的权重；

基于每一个像素子阵列的所述插值像素值与所述插值像素值的权重计算所述插值点的加权的插值像素值。

8. 一种用于对图像进行插值的装置,所述图像包括由多个像素点组成的像素阵列,其特征在于,包括:

处理器;

存储器,所述存储器中存储有能够在所述处理器上运行的程序指令,并且当所述程序指令在所述处理器上运行时,使得所述处理器:

确定插值点相对于所述像素阵列的位置;

基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重;

对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的一个或多个像素子阵列来计算所述插值点的插值像素值,其中所述一个或多个像素子阵列覆盖所述插值点;以及

基于所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值,计算所述插值点的加权的插值像素值。

9. 根据权利要求8所述的装置,其特征在于,基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重包括:

将与所述插值点相邻的多个像素点划分为多个像素单元;

计算所述多个像素单元中每个像素单元的边缘方向在所述多个预设边缘方向上的权重分量;

基于所述多个像素单元中每个像素单元的边缘方向在所述多个预设边缘方向上的权重分量,确定所述像素阵列在所述插值点位置的边缘方向沿所述多个预设边缘方向的权重。

10. 根据权利要求9所述的装置,其特征在于,所述像素单元的边缘方向在所述多个预设边缘方向上的权重分量与所述像素单元相对所述插值点位置的距离,以及所述像素单元的边缘方向的剧烈程度相关。

11. 根据权利要求8所述的装置,其特征在于,所述多个预设边缘方向包括 $-45^\circ$ 方向、 $0^\circ$ 方向、 $45^\circ$ 方向和 $90^\circ$ 方向。

12. 根据权利要求8所述的装置,其特征在于,对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的一个或多个像素子阵列的边界构成平行四边形,并且所述平行四边形的第一组相对边与所述预设边缘方向平行。

13. 根据权利要求12所述的装置,其特征在于,对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的一个或多个像素子阵列来计算所述插值点的插值像素值,包括:

在每个所述像素子阵列中,沿与所述平行四边形的第一组相对边不同的第二组相对边的方向,计算每一行或列的像素在与所述插值点位置对应的位置的插值像素值,其中与所

述插值点位置对应的位置是指沿所述预设边缘方向穿过所述插值点的直线与所述行或列的交点的位置；

基于所述每一行或列的像素在与所述插值点位置对应的位置的插值像素值，计算所述插值点的插值像素值。

14. 根据权利要求8所述的装置，其特征在于，基于所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值，计算所述插值点的加权的插值像素值，包括：

基于所述像素阵列在所述插值点位置的边缘方向沿多个预设边缘方向的权重计算与每一个像素子阵列对应的插值像素值的权重；

基于每一个像素子阵列的所述插值像素值与所述插值像素值的权重计算所述插值点的加权的插值像素值。

## 用于对图像进行插值的方法和装置

### 技术领域

[0001] 本申请涉及图像处理技术领域,更具体地,涉及一种用于对图像进行插值的方法和装置。

### 背景技术

[0002] 在手机、平板电脑、计算机或数字电视显示中,经常需要使用图像缩放算法对图像进行处理。例如,在对图像进行放大的过程中,需要使用插值算法,用于计算图像放大后缺失的像素。

[0003] 现有技术中广泛使用的插值算法包括最近邻元法、双线性内插法、三次内插法等,但使用这些方法获得的放大图像的锯齿现象明显。

### 发明内容

[0004] 本申请的一个目的在于提供一种用于对图像进行插值的方法和装置,能够在放大图像时减弱锯齿现象和减少人工痕迹。

[0005] 在本申请的一个方面,提供了一种用于对图像进行插值的方法,所述图像包括由多个像素点组成的像素阵列,所述方法包括:确定插值点相对于所述像素阵列的位置;基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向在多个预设边缘方向上的权重;对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的覆盖所述插值点的一个或多个像素子阵列来计算所述插值点的插值像素值;以及基于所述像素阵列在所述插值点位置的边缘方向在多个预设边缘方向上的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值,计算所述插值点的加权的插值像素值。

[0006] 在本申请的另一方面,还提供了一种用于对图像进行插值的装置,所述图像包括由多个像素点组成的像素阵列,所述装置包括:处理器;存储器,所述存储器中存储有能够在所述处理器上运行的程序指令,并且当所述程序指令在所述处理器上运行时,使得所述处理器:确定插值点相对于所述像素阵列的位置;基于与所述插值点相邻的多个像素点的像素值,确定所述像素阵列在所述插值点位置的边缘方向在多个预设边缘方向上的权重;对于所述多个预设边缘方向中的每一个预设边缘方向,从所述像素阵列中选取对应的覆盖所述插值点的一个或多个像素子阵列来计算所述插值点的插值像素值;以及基于所述像素阵列在所述插值点位置的边缘方向在多个预设边缘方向上的权重和所述多个预设边缘方向中的每一个预设边缘方向对应的所述插值点的插值像素值,计算所述插值点的加权的插值像素值。

[0007] 以上为本申请的概述,可能有简化、概括和省略细节的情况,因此本领域的技术人员应该认识到,该部分仅是示例说明性的,而不旨在以任何方式限定本申请范围。本概述部分既非旨在确定所要求保护主题的关键特征或必要特征,也非旨在用作为确定所要求保护主题的范围的辅助手段。

## 附图说明

[0008] 通过下面说明书和所附的权利要求书并与附图结合,将会更加充分地清楚理解本申请内容的上述和其他特征。可以理解,这些附图仅描绘了本申请内容的若干实施方式,因此不应认为是对本申请内容范围的限定。通过采用附图,本申请内容将会得到更加明确和详细地说明。

[0009] 图1示出了一图像的像素点组成的像素阵列100;

[0010] 图2示出了根据本申请一个实施例的用于对图像进行插值的方法的流程图;

[0011] 图3示出了根据本申请一个实施例的像素阵列200;

[0012] 图4示出了根据本申请一个实施例的确定像素阵列在插值点位置的边缘方向在多个预设边缘方向上的权重的流程图;

[0013] 图5示出了根据本申请一个实施例的与插值点相邻的多个像素点的示意图;

[0014] 图6示出了根据本申请一个实施例的从像素阵列中选取的像素子阵列形状的示意图;

[0015] 图7示出了根据本申请一个实施例的从像素子阵列来计算插值点的插值像素值的示意图;

[0016] 图8示出了根据本申请一个实施例的用于对图像进行插值的装置的示意图。

## 具体实施方式

[0017] 在下面的详细描述中,参考了构成其一部分的附图。在附图中,类似的符号通常表示类似的组成部分,除非上下文另有说明。详细描述、附图和权利要求书中描述的说明性实施方式并非旨在限定。在不偏离本申请的主题的精神或范围的情况下,可以采用其他实施方式,并且可以做出其他变化。可以理解,可以对本申请中一般性描述的、在附图中图解说明的本申请内容的各个方面进行多种不同构成的配置、替换、组合,设计,而所有这些都明确地构成本申请内容的一部分。

[0018] 首先,结合图1对多相位插值算法进行介绍,图1示出了一图像的像素点组成的像素阵列100,x和y轴方向的直线的每个交叉表示该图像的一个现有像素点。(i,j)表示坐标为i,j的现有像素点,图1中以圆点表示;(x,y)表示需要进行插值的像素点位置,图1中以圆圈表示,其到像素点(i,j)的水平距离为v,垂直距离为u。在图1所示的例子中,使用4×4个现有像素作为输入,计算插值点的像素值。具体地,以(i+u,j+v)表示当前插值点位置的坐标,则插值结果f(i+u,j+v)可以使用下面的公式(1)计算:

$$f(i+u,j+v) = [A] \times [B] \times [C] = [S(u+1) S(u+0) S(u-1) S(u-2)] * \begin{bmatrix} f(i-1,j-1) & f(i-1,j+0) & f(i-1,j+1) & f(i-1,j+2) \\ f(i+0,j-1) & f(i+0,j+0) & f(i+0,j+1) & f(i+0,j+2) \\ f(i+1,j-1) & f(i+1,j+0) & f(i+1,j+1) & f(i+1,j+2) \\ f(i+2,j-1) & f(i+2,j+0) & f(i+2,j+1) & f(i+2,j+2) \end{bmatrix} * \begin{bmatrix} S(v+1) \\ S(v+0) \\ S(v-1) \\ S(v-2) \end{bmatrix} \quad \text{公式(1)}$$

[0020] 其中,矩阵[B]为输入的4×4个现有像素点的像素值,[A]和[C]矩阵是每个现有像素点在水 平和垂直方向上的权重。三个矩阵相乘相当于对[B]中的每个像素值乘以两个方向上的权重 后再相加。例如,对于像素点(i,j),其在两个方向上到插值点位置的距离分别为u和v,从 而其像素值与权重系数S(u+0)和S(v+0)相乘;对应的,像素点(i-1,j-1)的像素

值对应的权重为 $S(u+1)$ 和 $S(v+1)$ ，以此类推。函数 $S(x)$ 为多相位插值算法的核函数，用于根据每个现有像素点到插值点位置的距离计算得到对应的权重，下面的公式(2)给出一个核函数 $S(x)$ 的示例。不同的应用场景中可以选择不同的核函数，但基本思想是现有像素点距离插值点越近则权重越大。

$$[0021] \quad S(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 & (0 \leq |x| < 1) \\ 4 - 8|x| + 5|x|^2 - |x|^3 & (1 \leq |x| < 2) \\ 0 & (|x| \geq 2) \end{cases} \quad \text{公式 (2)}$$

[0022] 本申请的发明人经过研究发现，与最近邻元法、双线性内插法相比，上述的多相位插值算法使用插值点位置周围更多的像素点作为输入，能够减小放大图像中的锯齿现象，但在放大倍数较大时依旧能看到明显锯齿。虽然可以通过选用不同的核函数使边缘模糊来改善锯齿，但很难同时兼顾锯齿和图像的清晰度。另外，一些基于边缘检测的算法通过检测输入图像的边缘方向以选用不同的插值算法来减小锯齿，例如，通过使用canny算子来检测图像中的边缘方向，根据检测结果使用双线性插值或其他方式得到输出图像。然而canny算子计算复杂并且在图像复杂区域难以得到准确的结果，另外，基于边缘方向的双线性插值每次只能使图像放大一倍，在放大倍数大时需要多次放大。

[0023] 基于此，本申请提供了一种通过检测图像中边缘方向来融合不同像素子阵列的插值结果的插值方法。该方法可以计算图像在插值点位置的边缘方向与多个预设边缘方向的趋近程度，而不是试图计算出一个准确的边缘方向，避免了边缘检测误差所造成的输出图像失真；另外，该方法在每个插值位置使用多个不同的像素子阵列计算多个插值像素值，并对多个插值结果进行融合，而不是对每个插值位置寻求使用最合适的子阵列得到插值结果，或对同一个像素子阵列使用不同的插值算法，从而可以避免在图像边缘变化剧烈区域产生人工痕迹。

[0024] 下面结合图2对本申请实施例的用于对图像进行插值的方法进行详细说明。图2示出了根据本申请一实施例用于对图像进行插值的方法的流程图，该方法可以在对图像进行缩放的过程中计算插值。

[0025] 图3示出了根据本申请一实施例的需要进行放大的图像200的示意图。如图3所示，该图像200为二维位图图像，包括以圆点示出的多个像素点组成的像素阵列，每个像素点都具有特定的位置。该图像200可以为灰度图像，或采用RGB、CMYK或其他适宜的编码格式进行编码的彩色图像，每个像素点都具有特定的灰度值或颜色通道值。

[0026] 在一些实施例中，可以使用图2所示的用于对图像进行插值的方法，对图3所示的图像200进行插值。以下描述中，在没有特殊说明的情况下，像素值表示灰度值或各颜色通道的值，不再进行区分。本领域技术人员应当理解，在对彩色图像进行插值时，需要分别针对各颜色通道进行插值。

[0027] 如图2所示，在步骤S202中，确定插值点相对于像素阵列的位置。

[0028] 在对二维图像进行放大的过程中，通常需要在水平方向和垂直方向上对图像进行插值。例如，如果要将水平和垂直方向为 $360 \times 640$ 像素阵列的图像放大到 $720 \times 1280$ ，这就需要 对图像分别在水平方向插入额外的360行像素和在垂直方向上插入额外的640列像素，即在原来没有像素点的位置插入新的像素点，形成新的图像。因此，在确定插值点相对于像素阵列的位置时，需要结合图像的放大倍数和插值处理在各行和各列的处理顺序。以

图3 为例,图像200中的点X被确定为当前插值点的位置,以圆圈表示。

[0029] 接着,在步骤S204中,基于与插值点相邻的多个像素点的像素值,确定像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的权重。

[0030] 在步骤S204中,基于与当前插值点相邻的多个像素点的像素值,确定像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的权重,而并不是给出一个确定的边缘方向 角度,可以有效减少或避免边缘检测误差所造成的图像失真。

[0031] 在一些实施例中,如图4所示,步骤S204包括子步骤S204a-S204c。

[0032] 具体地,在步骤S204a中,将与插值点相邻的多个像素点划分为多个像素单元。

[0033] 在一些实施例中,与插值点相邻的多个像素点是指像素阵列中与插值点距离最近的多 个现有像素点,这些像素点通常构成一个正方形阵列。例如,如图5所示,选择插值点X周围的 $4 \times 4$ 个像素点用于边缘检测。需要说明的是,在其他实施例中,基于具体的应用场景,并结合计算量和放大倍数,也可以使用更多或更少的像素点。

[0034] 接着,这些与插值点相邻的多个像素点再被划分为多个像素单元,如图5所示,像素点abcd被划分为一个像素单元,像素点bd与其右侧的两个像素点又被划分为另一个像素单元,像素点cd与其下方的两个像素点又被划分为另一个像素单元,以此类推,图5中的 $4 \times 4$ 个像素点可以被划分为 $3 \times 3$ 个像素单元。本实施例中,每个像素单元中包含4个像素点,但在其他实施例中,基于计算量和选取的与插值点相邻的像素点数量,进行划分后,每个像素单元也可以包含其他数量的像素点。

[0035] 接着,在步骤S204b中,计算多个像素单元中每个像素单元的边缘方向在多个预设边缘方向上的权重分量。

[0036] 通常预设边缘方向选取的数量越多,最终输出的图像边缘会越平滑,但由于每种预设边缘方向都需要对应的插值计算方法,所以计算量也会加大。在一实施例中,如图5所示,选取 $-45^\circ$ 方向、 $0^\circ$ 方向、 $45^\circ$ 方向和 $90^\circ$ 方向作为4个预设边缘方向,可以满足大部分应用场景。当然,在其他实施例中,为了使输出图像边缘更平滑,可以使用6个、8个或更多个的预设边缘方向。

[0037] 为了计算多个像素单元中每个像素单元的边缘方向在多个预设边缘方向上的权重分量,首先计算每个像素单元的边缘方向。继续以图5为例,若每个像素点的像素深度为8位,即以8位二进制数表示灰度值或各颜色深度值,对于像素单元abcd,其边缘方向角度A的正切值可以通过下面的公式(3)获得:

[0038]  $\tan(A) = -128 \times (b+d-a-c) / (c+d-a-b)$  公式(3);

[0039] 其中,a、b、c、d表示对应像素点的像素值,则 $(b+d-a-c)$ 表示像素单元abcd的水平方向梯度、 $(c+d-a-b)$ 表示像素单元abcd的垂直方向梯度;系数-128基于像素点的像素深度确定,在不同颜色深度的图像中,可以取不同值。

[0040] 对上述正切值通过反正切函数求值即可得到像素单元abcd的边缘方向的角度A。在一具体实施例中,可以采用下面的公式(4)计算,其为反正切函数的近似函数。需要说明的是,公式(4)只给出了0到1023的输入范围,但由于函数 $\arctan(x)$ 关于原点对称,对于-1023到0的输入直接将其对应正值的输出取负值作为输出即可。因此,公式(4)适用于输入范围为-1023到1023,输出范围为-256到256,对应角度为 $-90^\circ$ 到 $90^\circ$ 。

$$[0041] \quad \arctan(x) = \begin{cases} 138 \times x \gg 7 & (0 < x \leq 100) \\ 60 \times x + 7808 \gg 7 & (100 < x \leq 250) \\ 25 \times x + 16538 \gg 7 & (250 < x \leq 400) \\ 10 \times x + 22528 \gg 7 & (400 < x \leq 600) \\ 40 \times x + 16128 \gg 7 & (600 < x \leq 1023) \end{cases} \quad \text{公式 (4)};$$

[0042] 其中,算符“>>7”表示计算结果右移7位。需要说明的是,公式(4)中的常数与计算结果移动的位数与像素点的像素深度有关,在不同实施例中可能不同。

[0043] 在使用公式(3)和(4)获得图5中9个像素单元的边缘方向的角度后,接着计算每个像素单元的边缘方向在多个预设边缘方向(例如,-45°方向、0°方向、45°方向和90°方向)上的权重分量。在一些实施例中,使用公式(5)计算各个像素单元的边缘方向在预设边缘方向上的权重分量:

[0044]  $\text{edg}(i) = \text{fw}(|\arctan(x) - k|) \times \text{edg\_deg} \times \text{vert\_coeff} \times \text{hori\_coeff} \gg 23$  公式(5);其中,fw为近似高斯函数,arctan(x)对应当前像素单元的边缘方向,k对应当前使用的预设边缘方向;edg\_deg为该像素单元的边缘剧烈程度,以图5中的像素单元abcd为例,边缘剧烈程度为|b+d-a-c|与|c+d-a-b|中的较大值;算符“>>23”表示计算结果右移23位;vert\_coeff和hori\_coeff为根据当前插值点位置确定的竖直方向和水平方向的权重系数。

[0045] 在一些实施例中,vert\_coeff和hori\_coeff使用上文的多相位插值算法中的公式(2)来获得。对应于图1中的4×4的像素阵列,此处以3×3个像素单元在一预设边缘方向上的权重分量作为输入,计算竖直方向和水平方向的权重系数vert\_coeff和hori\_coeff。类似地,vert\_coeff和hori\_coeff也可以采用其他算法进行计算,但基本思想是像素单元距离插值点位置越近则权重越大。

[0046] 在一些实施例中,近似高斯函数使用公式(6)表示:

$$[0047] \quad \text{fw}(x) = \begin{cases} 256 & (x = 0) \\ -19 \times x + 2014 \gg 3 & (0 < x \leq 50) \\ -8 \times x + 1440 \gg 3 & (50 < x \leq 150) \\ -2 \times x + 618 & (150 < x \leq 256) \\ 1 & (256 < x) \end{cases} \quad \text{公式 (6)}$$

[0048] 其中,算符“>>3”表示计算结果右移3位。

[0049] 针对不同的预设边缘方向,公式(5)中k的取值不同。根据公式(4)的近似反正切函数,其输出范围为-256到256,对应角度为-90°到90°;则在该范围内-128对应-45°,因此,公式(5)中k取-128表示计算和预设边缘方向-45°的差异。类似地,预设边缘方向45°对应的k取值为128,预设边缘方向0°对应的k取值为0,预设边缘方向90°对应的k取值为|arctan(x) - 256|与|arctan(x) + 256|中的较小值。本领域技术人员应当理解,公式(5)和(6)中k的取值、常数值和计算结果的移动位数也均与像素点的像素深度有关,在不同实施例中可能不同。

[0050] 接着,在步骤S204c中,基于多个像素单元中每个像素单元的边缘方向在多个预设边缘方向上的权重分量,确定像素阵列在插值点位置的边缘方向在多个预设边缘方向上的权重。

[0051] 具体地,由于在前述步骤S204b中已经计算了所有9个像素单元在各个预设边缘方

向(即,  $-45^\circ$ 方向、 $0^\circ$ 方向、 $45^\circ$ 方向和 $90^\circ$ 方向)上的权重分量,本步骤S204c中,将9个像素单元在某个特定预设边缘方向上的权重分量相加,即可得到像素阵列在插值点位置的边缘方向在该预设边缘方向上的权重。换句话说,完成对各个像素单元对应的权重分量的计算后需要融合所有像素单元的结果;由公式(5)可以看出,融合时的权重由两部分组成,离当前插值点位置越近的像素单元的权重越大(即,  $vert\_coeff$ 和 $hori\_coeff$ 越大),边缘剧烈程度越大的像素单元权重越大(即,  $edg\_deg$ 越大);其中,每个像素单元的距离直接由该像素单元的中心到当前插值点位置得到,而边缘剧烈程度可以用该像素单元两个方向上梯度的较大值表示。

[0052] 以图5为例,步骤S204c可计算获得像素阵列200在插值点X的边缘方向在4个预设边缘方向 $-45^\circ$ 、 $45^\circ$ 、 $0^\circ$ 和 $90^\circ$ 的4个权重值 $edg0$ ,  $edg1$ ,  $edg2$ 和 $edg3$ 。

[0053] 回到图2,继续执行步骤S206,对于多个预设边缘方向中的每一个预设边缘方向,从像素阵列中选取对应的覆盖插值点的一个或多个像素子阵列来计算插值点的插值像素值。

[0054] 在一些实施例中,对于多个预设边缘方向中的每一个预设边缘方向,从像素阵列中选取对应的覆盖插值点的一个或多个像素子阵列的边界构成平行四边形,并且平行四边形的第一组相对边与预设边缘方向平行。与现有技术中仅使用正方形或长方形的区域的像素子阵列进行插值的方法相比,本申请实施例对像素子阵列的选取考虑了预设边缘方向的角度,包括多种不同形状的像素子阵列,有利于减少插值误差。

[0055] 继续以预设边缘方向为 $-45^\circ$ 、 $0^\circ$ 、 $45^\circ$ 和 $90^\circ$ 为例,对从像素阵列中选取对应的覆盖插值点的一个或多个像素子阵列进行说明。例如,从像素阵列中选取的像素子阵列与图6所示的5个平行四边形对应。倾斜的预设边缘方向角度包括 $-45^\circ$ 和 $45^\circ$ ,对于它们,偏向水平和偏向竖直的边缘方向需要使用两种不同的像素子阵列。例如,对于预设边缘方向角度为 $45^\circ$ ,选择与图6中(a)和(c)的平行四边形对应的像素子阵列;(a)和(c)的一组相对边与预设边缘方向角度平行,即也为 $45^\circ$ ;(a)和(c)的另外一组相对边分别平行于水平方向和竖直方向。类似地,对于预设边缘方向角度为 $-45^\circ$ ,选择与图6中(b)和(d)的平行四边形对应的像素子阵列。对于预设边缘方向角度为 $0^\circ$ 和 $90^\circ$ ,选择与图6中(e)的正方形对应的像素子阵列。

[0056] 由于预设边缘方向的角度越接近于垂直方向时,对应像素子阵列需要覆盖垂直方向上的更多行,导致需要更多的行内存,因此在一些实施例中预设边缘方向的角度仅使用 $-45^\circ$ 、 $0^\circ$ 、 $45^\circ$ 和 $90^\circ$ 四种,对应选取与图6所示的5个平行四边形像素子阵列。在其他实施例中,根据具体应用场景,选择其他数量的边缘方向时,则需要选择其他适宜形状的像素子阵列。但需要说明的是,由于插值计算中的输入必须在插值方向上均匀分布并且为原始图像像素阵列上的像素点,因此非矩形像素子阵列所对应的平行四边形区域的斜边需要在每一行/列上经过输入像素阵列上的像素点,否则插值计算过程中输入的像素点本身就是一个待插值的点。因此,非矩形像素子阵列斜边的倾斜角度只能为 $\arctan(n)$ 和 $\arctan(1/n)$ ,其中 $n$ 为任意整数。

[0057] 在对应每一个预设边缘方向,从像素阵列中选取对应的覆盖插值点的像素子阵列之后,就可以基于所选择的像素子阵列来计算插值点的插值像素值。具体地,在每个像素子阵列中,沿与平行四边形的上述的第一组相对边不同的第二组相对边的方向,计算每一

行或列的像素在与插值点位置对应的位置的插值像素值；基于每一行或列的像素在与插值点位置对应的位置的插值像素值，计算插值点的插值像素值。其中，与插值点位置对应的位置是指沿预设边缘方向穿过插值点的直线与该行或列的交点的位置。

[0058] 参考图7(a)，其示出了对应于图6(a)所示的平行四边形的像素子阵列，其中，沿预设边缘方向穿过插值点的直线(在7(a)中以虚线示出)与该像素子阵列中4个行相交，首先计算这4行像素与该直线交点位置的水平插值结果，再使用这4个水平插值结果计算在该直线的插值点位置的插值像素值。对应地，参考图7(b)，对应于图6(c)所示的平行四边形的像素子阵列，沿预设边缘方向穿过插值点的直线(在7(b)中以虚线示出)与该像素子阵列中4个列相交，首先计算这4列像素与该直线交点位置的竖直插值结果，再使用这4个竖直插值结果计算在该直线的插值点位置的插值像素值。其他三种像素子阵列的计算方法与此类似，在此不再赘述。计算结束后可以获得对应于图6所示的5个像素子阵列的插值像素值。

[0059] 继续参考图2，执行步骤S208，基于该像素阵列在插值点位置的边缘方向在多个预设边缘方向上的权重和多个预设边缘方向中的每一个预设边缘方向对应的插值点的插值像素值，计算插值点的加权的插值像素值。

[0060] 在步骤S204和步骤S206中分别计算了像素阵列在插值点位置的边缘方向在多个预设边缘方向上的权重以及对应每一个预设边缘方向的该插值点的插值像素值，则对上述结果进行加权融合后，即可获得该插值点的加权的插值像素值。

[0061] 具体地，在一些实施例中，首先基于该像素阵列在插值点位置的边缘方向在多个预设边缘方向上的权重计算与每一个像素子阵列对应的插值像素值的权重，再基于每一个像素子阵列的插值像素值及其对应的权重计算该插值点的加权的插值像素值。

[0062] 继续以预设边缘方向 $-45^\circ$ 、 $0^\circ$ 、 $45^\circ$ 和 $90^\circ$ 为例，图6中的每一个像素子阵列对应的插值像素值的权重可以通过公式(7)计算：

$$[0063] \quad \begin{cases} \text{wedg} = f\left(256 \times \frac{\text{edg0}}{\text{edg0}+\text{edg1}}, \text{thr}\right) \times f\left(256 \times \frac{\text{edg2}}{\text{edg2}+\text{edg3}}, \text{thr}\right) \\ w5 = 256 - \text{wedg} \\ w1 = (\text{edg0} \times \text{edg2})/(\text{edg1} \times \text{edg3}) \\ w2 = (\text{edg1} \times \text{edg2})/(\text{edg0} \times \text{edg3}) \\ w3 = (\text{edg0} \times \text{edg3})/(\text{edg1} \times \text{edg2}) \\ w4 = (\text{edg1} \times \text{edg3})/(\text{edg0} \times \text{edg2}) \end{cases} \quad \text{公式(7)}$$

[0064] 其中，edg0, edg1, edg2和edg3为像素阵列在插值点的边缘方向在4个预设边缘方向 $-45^\circ$ 、 $45^\circ$ 、 $0^\circ$ 和 $90^\circ$ 的4个权重值；w1, w2, w3, w4和w5为图5中的每一个像素子阵列对应的插值像素值的权重值；并且，

$$[0065] \quad f(x) = \begin{cases} 32768 - (256 - \text{thr}) \times x \gg 7 & (0 < x \leq 128) \\ (256 - \text{thr}) \times x + 256 \times \text{thr} - 32768 \gg 7 & (128 < x \leq 256) \end{cases} \quad \text{公式(8)}$$

[0066] 其中，公式(8)的作用是当两个输入edg0和edg1(或edg2和edg3)越接近时输出越小，相同时输出thr，而两个输入差异越大时输出越大。其中，thr越大，f(x)的输出也越大，基于边缘或者说基于非正方形的像素子阵列计算获得的插值像素值的权重也会越大；从而可以认为thr控制最后加权的插值像素值结果更倾向于考虑边缘方向还是更倾向于传统的多相位插值计算方式。在一些实施例中，thr取值为128时，能得到较好的效果。

[0067] 可以看出,wedg为图6中(a)、(b)、(c)和(d)对应的插值像素值权重的和,其计算思想是,如果在步骤S204中获得的像素阵列在当前插值点位置的边缘方向对 $\pm 45^\circ$ 的接近程度都很大,或对 $0/90^\circ$ 的接近程度都很大,说明对该边缘方向的判断不准确,更倾向于使用通过图6(e)计算获得的插值像素值;而当对该边缘方向判断准确的时候,更多的使用通过图6的(a)、(b)、(c)和(d)计算获得的插值像素值。因此,计算出 wedg后将其分配给与图6的(a)、(b)、(c)和(d)对应的 $w_1, w_2, w_3$ 和 $w_4$ ,而 剩余权重分配给与图6的(e)对应的 $w_5$ 。 $w_1$ 对应的像素子阵列适合的边缘方向在 $-45^\circ - 0^\circ$ 之间,因此其权重用 $-45^\circ$ 和 $0^\circ$ 的权重(edg0和edg2)除以 $45^\circ$ 和 $90^\circ$ 的权重(edg1和 edg3)得到。 $w_2, w_3$ 和 $w_4$ 的计算方法与此类似,在此不再赘述。

[0068] 接着,基于每一个像素子阵列的插值像素值与插值像素值的权重通过公式(9)该插 值点的加权的插值像素值:

[0069] 
$$OUT = (wedg) \times (w_1 \times out_1 + w_2 \times out_2 + w_3 \times out_3 + w_4 \times out_4) / (w_1 + w_2 + w_3 + w_4) + w_5 \times out_5 >> 8$$
 公式(9)

[0070] 其中,OUT表示插值点的加权的插值像素值,算符“>>8”输出结果右移8位。

[0071] 虽然上文以预设边缘方向 $-45^\circ, 0^\circ, 45^\circ$ 和 $90^\circ$ 为例对步骤S208进行说明,但应当理解 的是,其计算思想也适用于其他的预设边缘方向。换句话说,如果像素阵列在插值点位 置的边缘方向对一像素子阵列对应的预设边缘方向的接近程度越大,则该像素子阵列 对应 的插值像素值的权重也越大;但由于对图像复杂区域及噪声大的区域的边缘方向检测 会有 一定的误差,因此希望只在边缘方向检测结果有信心的区域增大对使用非矩形 像素子阵 列获得的插值像素值的权重,而对没有信心的区域则更多的使用矩形或正方形 像素子阵列 获得的插值像素值。由于边缘方向检测结果是真实的边缘方向对各个预设边 缘方向的接近 程度,即检测结果为相对于各个预设边缘方向的权重,可以认为真实边缘方 向是在权重最 大的两个预设边缘方向之间;因此每个像素子阵列的插值像素值对应的权 重可以用其对应 的预设边缘方向的权重乘以其相邻的权重大的那个预设边缘方向的权 重,再除以其他预 设边缘方向的权重来获得;最后将所有权重归一化作为各个插值像素值 结果的系数。需要 说明的是,虽然上文的权重计算策略可应用于不同数量的预设边缘方 向,但为了得到更好 的结果,当选取不同数量的预设边缘方向时,可以对权重计算策略优 化设计。

[0072] 参考图8,图8示出了根据本申请一实施例用于对图像进行插值的装置800的示意 图。在一些实施例中,该装置可以用于对图3所示的包括由多个像素点组成的像素阵列的 图像 进行插值。

[0073] 如图8所示,该装置800包括:处理器802和存储器804;存储器802中存储有能够 在 处理器804上运行的程序指令,并且当该程序指令在处理器804上运行时,使得处理器 804 能够执行以下步骤:确定插值点相对于该像素阵列的位置;基于与该插值点相邻的多 个像 素点的像素值,确定该像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的 权重;对于多个预设边缘方向中的每一个预设边缘方向,从该像素阵列中选取对应的覆盖 该 插值点的一个或多个像素子阵列来计算该插值点的插值像素值;以及基于该像素阵列 在 该插值点位置的边缘方向在多个预设边缘方向上的权重和该多个预设边缘方向中的每 一个 预设边缘方向对应的插值点的插值像素值,计算该插值点的加权的插值像素值。

[0074] 在一些实施例中,基于与该插值点相邻的多个像素点的像素值,确定该像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的权重包括:将与该插值点相邻的多个像素点划分为多个像素单元;计算该多个像素单元中每个像素单元的边缘方向在多个预设边缘方向上的权重分量;基于多个像素单元中每个像素单元的边缘方向在所述多个预设边缘方向上的权重分量,确定该像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的权重。

[0075] 在一些实施例中,该像素单元的边缘方向在多个预设边缘方向上的权重分量与该像素单元相对所述插值点位置的距离以及该像素单元的边缘方向的剧烈程度相关。

[0076] 在一些实施例中,多个预设边缘方向包括 $-45^\circ$ 方向、 $0^\circ$ 方向、 $45^\circ$ 方向和 $90^\circ$ 方向。

[0077] 在一些实施例中,对于多个预设边缘方向中的每一个预设边缘方向,从该像素阵列中选取对应的覆盖该插值点的一个或多个像素子阵列的边界构成平行四边形,并且该平行四边形的第一组相对边与该预设边缘方向平行。

[0078] 在一些实施例中,对于多个预设边缘方向中的每一个预设边缘方向,从该像素阵列中选取对应的覆盖该插值点的一个或多个像素子阵列来计算该插值点的插值像素值,包括:在每个所述像素子阵列中,沿与该平行四边形的第一组相对边不同的第二组相对边的方向,计算每一行或列的像素在与该插值点位置对应的位置的插值像素值;基于每一行或列的像素在与该插值点位置对应的位置的插值像素值,计算与该插值点的插值像素值。

[0079] 在一些实施例中,基于该像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的权重和多个预设边缘方向中的每一个预设边缘方向对应的该插值点的插值像素值,计算该插值点的加权的插值像素值,包括:基于该像素阵列在该插值点位置的边缘方向在多个预设边缘方向上的权重计算与每一个像素子阵列对应的插值像素值的权重;基于每一个像素子阵列的插值像素值与该插值像素值的权重计算该插值点的加权的插值像素值。

[0080] 应当注意,尽管在上文详细描述中提及了用于对图像进行插值的方法的若干步骤,和用于对图像进行插值的装置的若干模块,但是这种划分仅仅是示例性的而非强制性的。实际上,根据本申请的实施例,上文描述的两个或更多模块的特征和功能可以在一个模块中具体化。反之,上文描述的一个模块的特征和功能可以进一步划分为由多个模块来具体化。

[0081] 那些本技术领域的一般技术人员可以通过研究说明书、公开的内容及附图和所附的权利要求书,理解和实施对披露的实施方式的其他改变。在权利要求中,措词“包括”不排除其他的元素和步骤,并且措辞“一”、“一个”不排除复数。在本申请的实际应用中,一个零件可能执行权利要求中所引用的多个技术特征的功能。权利要求中的任何附图标记不应理解为对范围的限制。

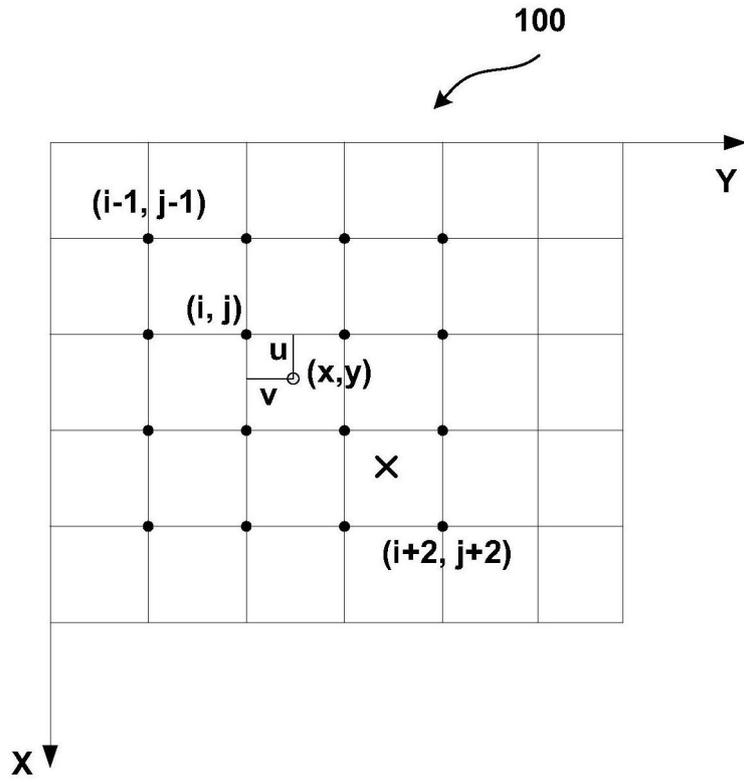


图1

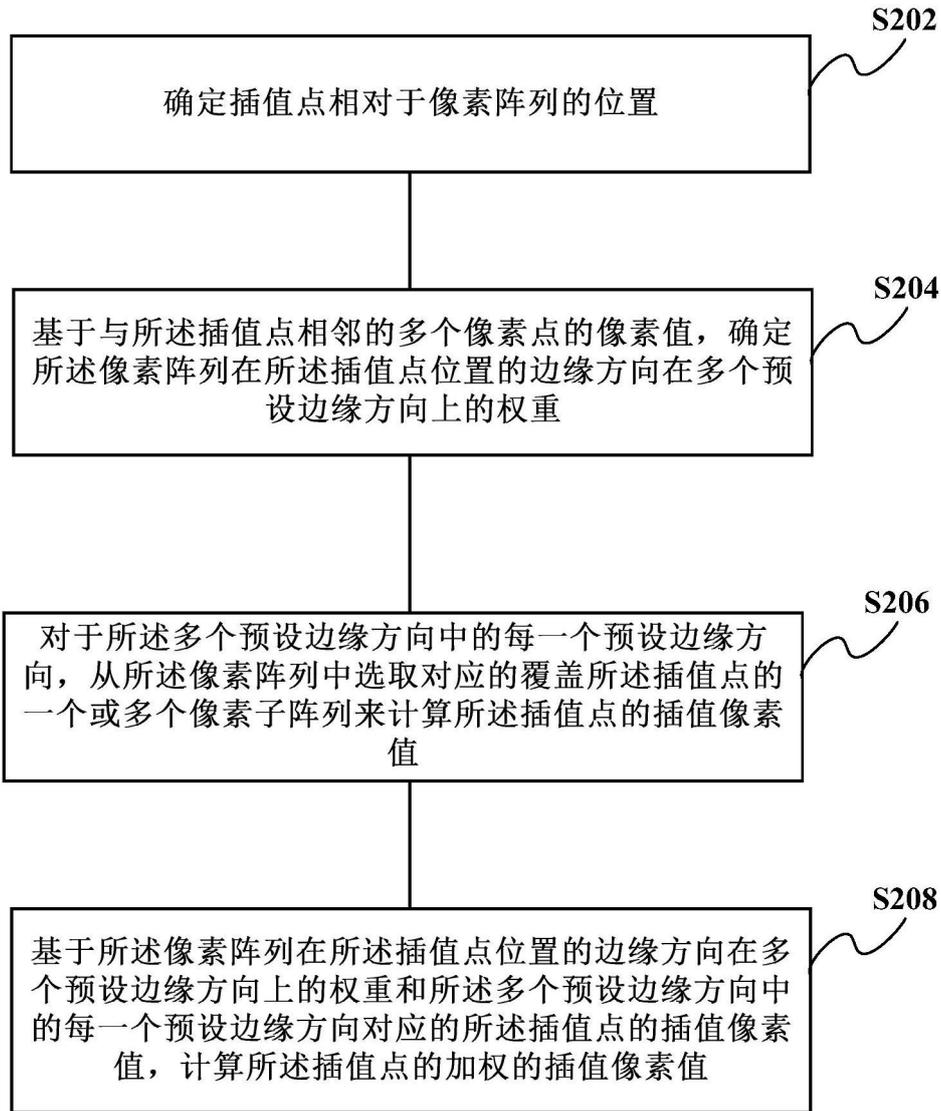


图2

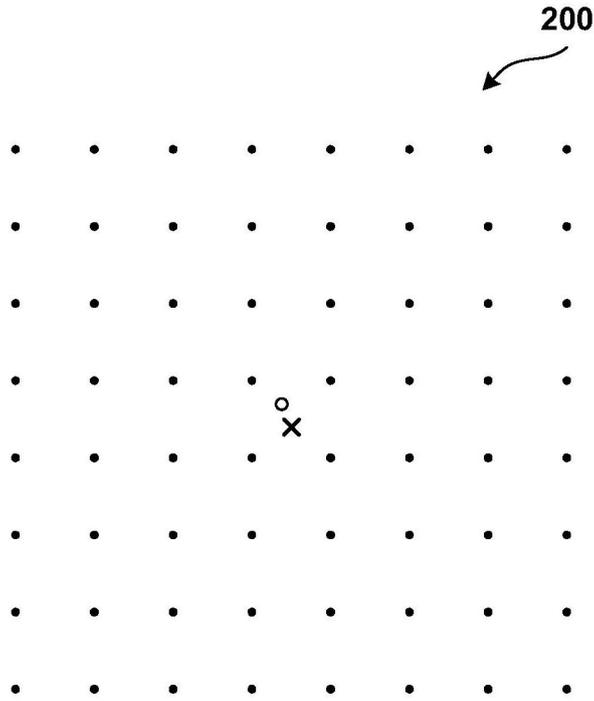


图3

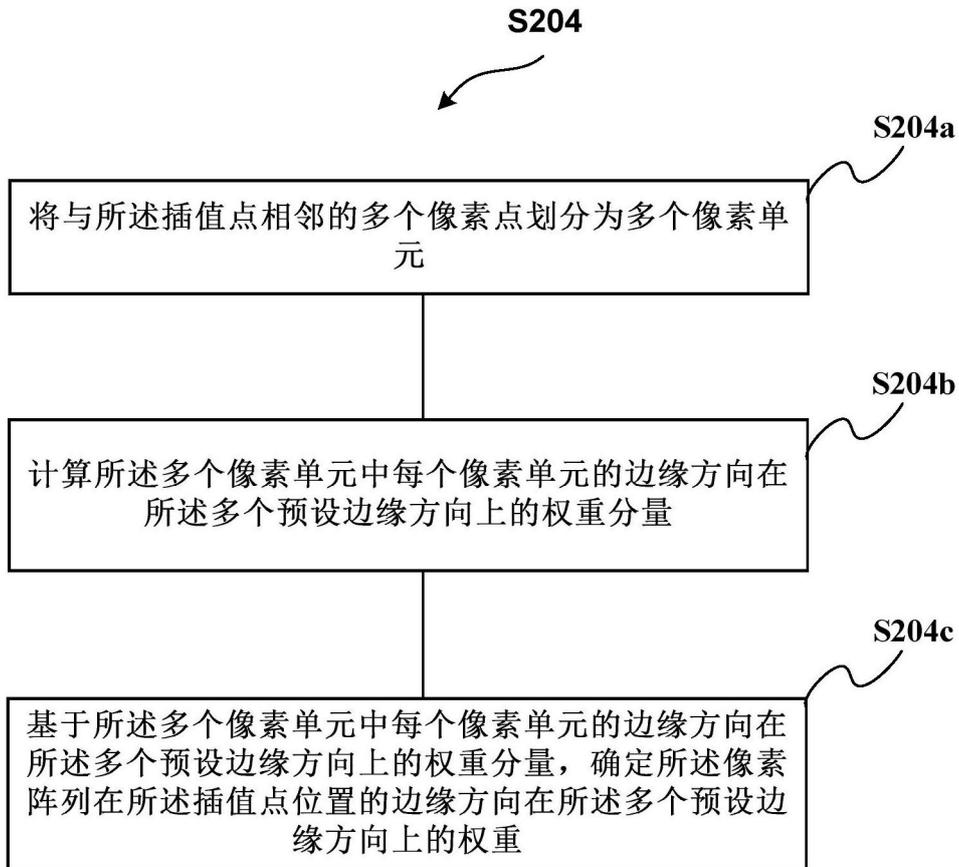


图4

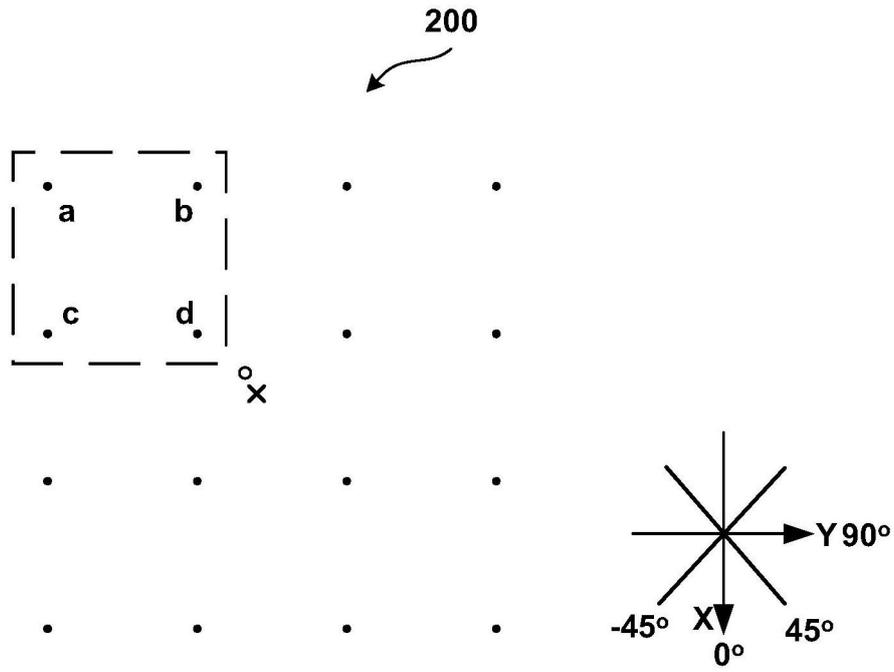


图5

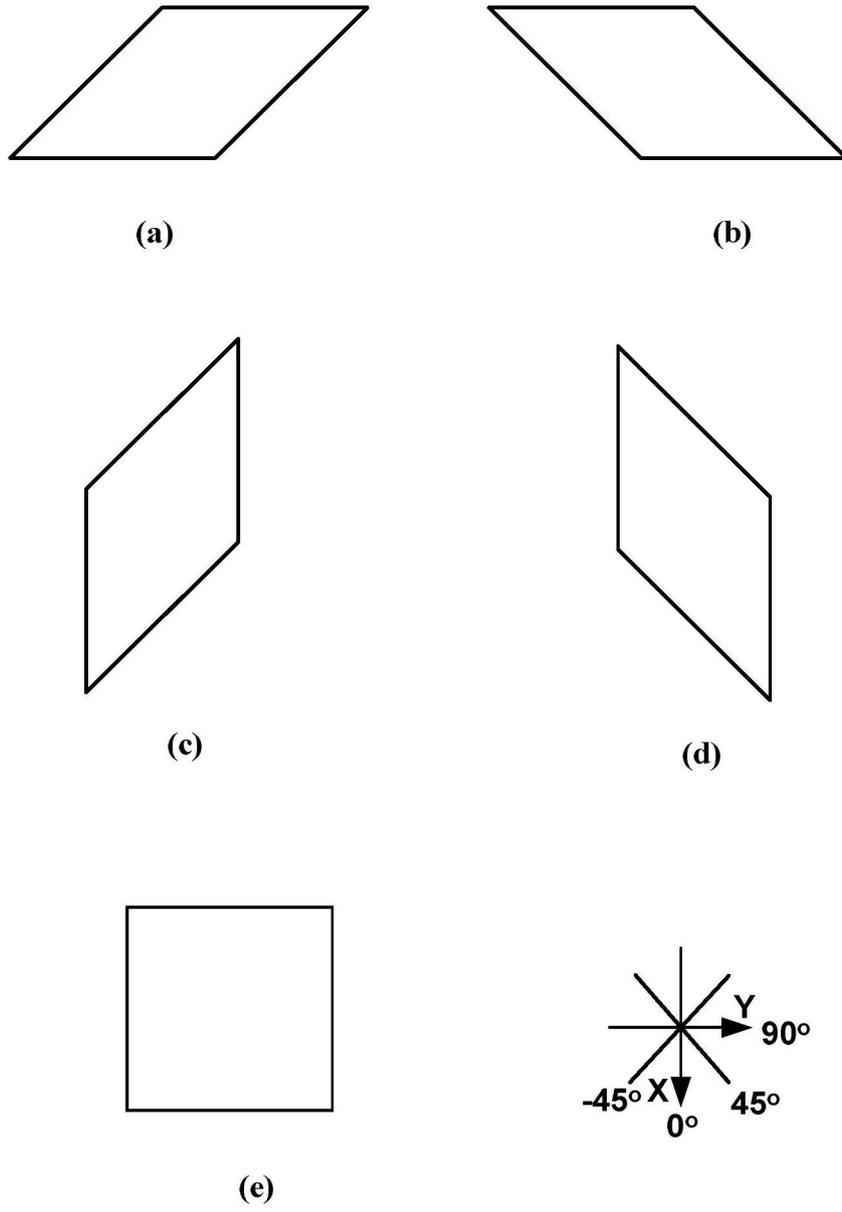
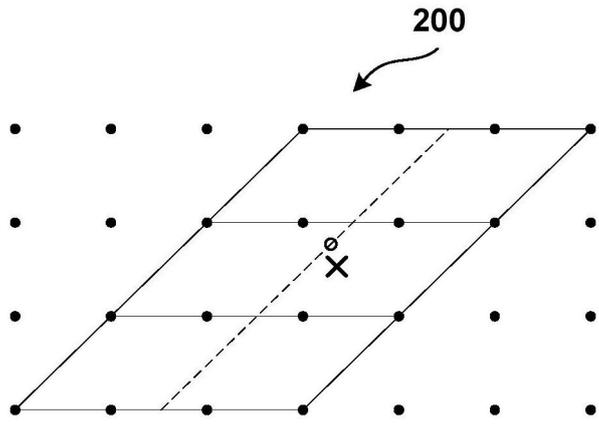
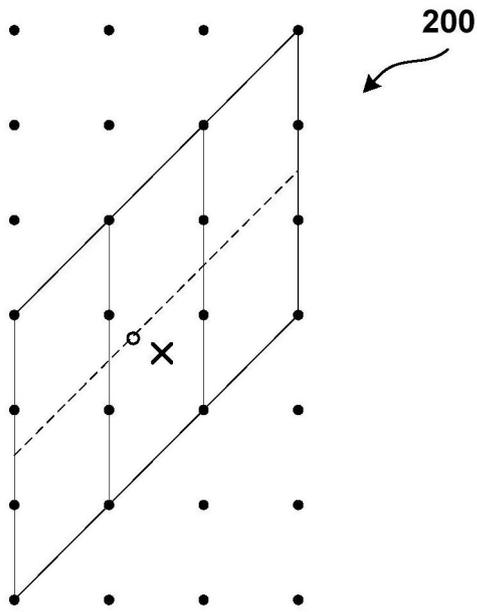


图6



(a)



(b)

图7

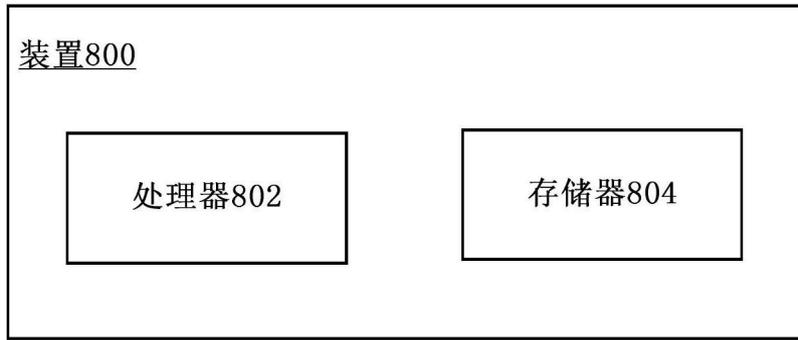


图8