

对象处理方法、装置、设备、计算机可读存储介质及计算机程序产品

相关申请的交叉引用

5 本申请基于申请号为 202210684708.8、申请日为 2022 年 6 月 17 日的中国专利申请提出，并要求以上中国专利申请的优先权，以上中国专利申请的全部内容在此引入本申请作为参考。

技术领域

本申请涉及计算机技术领域，尤其涉及一种对象处理方法、装置、设备、计算机可读存储介质及计算机程序产品。

10 背景技术

多边形模型是计算机图形中最流行、最重要，且得到最广泛支持的模型表示方法。而二维装箱的基本过程是：依次将每个多边形按照某种规则放入容器；每次放入之后更新容器的可放置区域，然后再放置下一个多边形，直到所有多边形已放入或者容器已满。图 1 给出了装箱过程的一个中间状态，其中左侧是尚未放置的多边形，右侧是已被放置好的多边形以及容器的剩余空间。装箱问题不仅在学术界得到了广泛研究，其成果在工业界也得到了大量应用。已有相关技术提供的装箱算法在得到装箱结果之后，往往会存在较大的空隙，即便进行进一步的优化仅仅是对部分多边形进行扩展，或者对所有多边形进行同类型的扩展，装箱填充率的提升效果有限，无法超越人工装箱优化效果。

发明内容

20 本申请实施例提供一种对象处理方法、装置、计算机可读存储介质及计算机程序产品，能够提高目标容器中资源对象的填充率。

本申请实施例的技术方案是这样实现的：

本申请实施例提供一种对象处理方法，应用于贴图渲染场景，所述方法应用于计算机设备，包括：

25 获取待处理的多个资源对象在目标容器中的初始填充结果；

基于所述初始填充结果，将各个资源对象移动至各自对应的可用空间的中心位置；

获取所述各个资源对象的对称属性信息和包含关系信息；

基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种；

30 基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象。

本申请实施例提供一种对象处理装置，包括：

第一获取模块，配置为获取待处理的多个资源对象在目标容器中的初始填充结果；

35 第一移动模块，配置为基于所述初始填充结果，将各个资源对象移动至各自对应的可用空间的中心位置；

第二获取模块，配置为获取所述各个资源对象的对称属性信息和包含关系信息；

第一确定模块，配置为基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种；

5 第一扩展模块，配置为基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象。

本申请实施例提供一种装置，包括：

存储器，用于存储可执行指令；

10 处理器，用于执行所述存储器中存储的可执行指令时，实现本申请实施例提供的对象处理方法。

本申请实施例提供一种计算机可读存储介质，存储有可执行指令，用于引起处理器执行时，实现本申请实施例提供的对象处理方法。

本申请实施例提供一种计算机程序产品，包括计算机程序或指令，所述计算机程序或指令被处理器执行时实现本申请实施例提供的对象处理方法。

15 本申请实施例具有以下有益效果：

在本申请实施例提供的对象处理方法中，在获取待处理的多个资源对象的初始填充结果之后，首先基于该初始填充结果进行全局均匀化处理，以将各个资源对象移动至各自对应的可用空间的中心位置；然后再获取所述各个资源对象的对称属性信息和包含关系信息，并基于资源对象的对称属性信息和包含关系信息确定各个资源对象的扩展类型，最后基于各个资源对象的扩展类型进行扩展处理，得到处理后的各个资源对象，由于是对目标容器中的每个资源对象都进行了扩展，并且在扩展之前将资源对象移动至中心位置，从而能够在有限的空间内实现最大比例的扩展，而且资源对象的扩展类型是基于资源对象的对称属性信息和包含信息自适应确定出的，其中扩展类型包括各向同性扩展和各向异性扩展，相较于相关技术中仅能够对资源对象进行各向同性扩展，在本申请20 实施例中，对于只能进行各向同性扩展的资源对象进行各向同性扩展，但是对于能够进行各向异性扩展的资源对象通过各向异性扩展能够实现对资源对象最大程度上的扩展，如此能够模拟出人工装箱调整的效果，进而最大限度的提高填充率。

附图说明

图 1 为装箱问题的示意图；

30 图 2 是本申请实施例提供的对象处理系统 100 的架构示意图；

图 3 是本申请实施例提供的第二终端 400 的结构示意图；

图 4A 是本申请实施例提供的对象处理方法的一种实现流程示意图；

图 4B 是本申请实施例提供的对象处理方法的另一种实现流程示意图

35 图 5 是本申请实施例提供的将各个资源对象移动至对应可用空间中心的实现流程示意图；

图 6 是本申请实施例提供的将所述第一目标对象移动至第一目标可用空间的实现流程示意图；

图 7 是人工修改装箱结果的示意图；

图 8 是本申请实施例提供的多边形分类扩展的示意图；

40 图 9 是本申请实施例提供的对象处理方法的再一种实现过程示意图；

图 10 是本申请实施例提供的 UV 岛不处于局部扩展空隙中心的示意图；

图 11 是本申请实施例提供的各向异性扩展的示意图；

图 12 为本申请实施例提供的各向同性扩展的示意图;

图 13 为本申请实施例提供的各向异性扩展的实现过程示意图;

图 14 为本申请实施例提供的各向同性扩展的实现过程示意图;

图 15 为本申请实施例提供的小中大规模 UV 岛扩展前后的装箱效果对比图。

5 具体实施方式

为了使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请作进一步地详细描述,所描述的实施例不应视为对本申请的限制,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例,都属于本申请保护的范围。

在以下的描述中,涉及到“一些实施例”,其描述了所有可能实施例的子集,但是可以理解,“一些实施例”可以是所有可能实施例的相同子集或不同子集,并且可以在不冲突的情况下相互结合。

在以下的描述中,所涉及的术语“第一/第二/第三”仅仅是是区别类似的对象,不代表针对对象的特定排序,可以理解地,“第一/第二/第三”在允许的情况下可以互换特定的顺序或先后次序,以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。

除非另有定义,本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的,不是旨在限制本申请。

对本申请实施例进行进一步详细说明之前,对本申请实施例中涉及的名词和术语进行说明,本申请实施例中涉及的名词和术语适用于如下的解释。

1) 装箱问题(Packing Problem),一般意义上的装箱问题是指将一组规则或者不规则的二维或者三维物体,按照某种规则排列在一个给定的规则或者不规则容器中,从而达到特定目的。本申请实施例所涉及的装箱问题聚焦于将二维不规则多边形装箱在一个二维容器中,如图 1 所示。在美术制作术语中,也将待装箱的二维不规则多边形称为 UV 岛,在本申请实施例中,将二维不规则多边形称为资源对象。

2) 非拟合多边形(NFP, No Fit Polygon),是指一个多边形 B 沿着另外一个多边形 A 的内壁进行滑动时,其边界上的某个参考点的轨迹所形成的多边形。NFP 定义了对多边形 B 在多边形 A 所定义的区域进行装箱时,该参考点被放置的可行区域。

3) 闵可夫斯基和(Minkowski Sum),给定向量集合 A 和 B,定义它们的闵可夫斯基和为 $A+B=\{\mathbf{a}+\mathbf{b}|\mathbf{a}\in A,\mathbf{b}\in B\}$ 。

4) 遗传算法(GM, Genetic Algorithm),是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式,利用计算机仿真运算,将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果。

5) 空间搜索树(AABB Tree):一种基于 K-d 树原理实现的空间搜索树,其每个节点代表一个三维几何图元的空间包围盒。该数据结构可以快速报告查询图元是否与树中的图元相交,以及具体的相交类型、相交位置等。

6) 出血值(Padding):装箱问题中的美术专业用语,指装箱过程中对 UV 岛间进行间距保护使之留有一定空隙。一般 UV 岛装箱中常用的做法是根据装箱空间分辨率,每 128 个像素取一个像素作为出血值,例如 256*256 分辨率通常采用 2 个像素值作为出血值。

为了更好地理解本申请实施例提供的用于装箱优化的对象处理方法，首先对相关技术中的装箱算法进行说明。

5 在相关技术中，常用的装箱算法有基于 NFP 和最低重心的方法，在实现时，依次为每个待放置的多边形确定多个候选角度以及依据候选角度确定多个候选位置，按照重心最低的候选位置放置相应多边形。按照这种装箱方法放置后，容器中会存在较大空隙，因此填充率较低。

10 在相关技术中，提出一种针对已有装箱结果进行位置优化以提升填充率的方案。通过迭代地将一些 UV 岛进行各向同性扩展并且放入更大的空隙中，来逐步提高整个装箱的填充率。这种方法通过近似均匀地扩大已装箱的 UV 岛并优化它们的位置，可以使得装箱效果得到大幅提升。但是它依然无法最大限度地提高填充率，其原因有二：1) UV 岛只有在进行位置更新的时候才可以进行等比例扩大，这样就限制了对没有进行位置更新的 UV 岛的扩展；2) 对于部分允许进行各向异性扩展的 UV 岛，该算法并不支持，从而影响了填充率的进一步提升。

15 基于此，本申请实施例提供一种对象处理方法、装置、设备、计算机可读存储介质及计算机程序产品，可以在保持资源对象各向同性/各向异性特征的前提下，最大限度地提高装箱利用率，从而满足特定需求。下面说明本申请实施例提供的计算机设备的示例性应用，本申请实施例提供的设备可以实施为笔记本电脑，平板电脑，台式计算机，机顶盒，移动设备（例如，移动电话，便携式音乐播放器，个人数字助理，专用消息设备，20 便携式游戏设备）等各种类型的用户终端，也可以实施为服务器。下面，将说明设备实施为终端时示例性应用。

25 参见图 2，图 2 是本申请实施例提供的对象处理系统 100 的架构示意图，如图 2 所示，该系统包括第一终端 200、网络 300 和第二终端 400，其中，第一终端 200 和第二终端 400 通过网络建立有通信连接，网络 300 可以是广域网或者局域网，又或者是二者的组合。

30 用户可以利用第一终端 200 完成虚拟对象的三维模型设计，该虚拟对象可以是游戏场景中的虚拟对象，还可以是虚拟现实场景或者增强现实场景中的虚拟对象，例如可以是虚拟人物，虚拟建筑物，以及虚拟植物等，而显示在虚拟场景中的虚拟对象是通过纹理映射技术（Texture Mapping）对该虚拟对象的三维模型进行纹理贴图后得到的，从而可提升显示在游戏画面中的虚拟对象的真实性和生动性。在本申请实施例中，第一终端 200 在完成对虚拟对象的三维模型设计后，将三维模型数据发送至第二终端 400，第二终端 400 可以对虚拟对象的三维模型进行面片剖分，得到二维的资源对象（在一些实施例中，可以称为 UV 岛或者多边形），然后在对二维的资源对象进行装箱处理。在实现时可以对二维的资源对象进行装箱处理，从而得到初始填充结果，该初始填充结果可以35 是利用预设的装箱算法对多个资源对象进行装箱处理得到的，还可以是在利用预设的装箱算法对多个资源对象进行装箱处理后，再对得到的装箱结果进行进一步的全局优化得到的，在本申请实施例中，在获取到初始填充结果后，基于所述初始填充结果，将各个资源对象移动至各自对应的可用空间的中心位置；然后再通过各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种，之后是基于所述各个资源对象的扩展类型，对所述40 各个资源对象进行扩展处理，得到处理后的各个资源对象，由于第一目标对象占用的空间小于该最大空隙，并且将第一目标对象移动之后进行了放大处理，从而使得容器的总空余空间变小，提升填充率。

45 在一些实施例中，第一终端 200 和第二终端 400 可以是同一个终端，也即通过同一个终端完成对象设计和纹理渲染功能。在一些实施例中，第二终端 400 还可以是从服务

器中获取待渲染的三维虚拟对象，然后第二终端 400 对三维虚拟对象进行面片剖分，得到多个二维的资源对象，并将多个资源对象进行装箱处理以及全局优化处理，从而提高填充率。

这里的服务器可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器。第一终端 200 和第二终端 400 可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表、车载智能终端等，但并不局限于此。终端以及服务器可以通过有线或无线通信方式进行直接或间接地连接，本申请实施例中不做限制。

参见图 3，图 3 是本申请实施例提供的第二终端 400 的结构示意图，图 3 所示的第二终端 400 包括：至少一个处理器 410、存储器 450、至少一个网络接口 420 和用户接口 430。终端 400 中的各个组件通过总线系统 440 耦合在一起。可理解，总线系统 440 用于实现这些组件之间的连接通信。总线系统 440 除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图 3 中将各种总线都标为总线系统 440。

处理器 410 可以是一种集成电路芯片，具有信号的处理能力，例如通用处理器、数字信号处理器（DSP，Digital Signal Processor），或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等，其中，通用处理器可以是微处理器或者任何常规的处理器等。

用户接口 430 包括使得能够呈现媒体内容的一个或多个输出装置 431，包括一个或多个扬声器和/或一个或多个视觉显示屏。用户接口 430 还包括一个或多个输入装置 432，包括有助于用户输入的用户接口部件，比如键盘、鼠标、麦克风、触屏显示屏、摄像头、其他输入按钮和控件。

存储器 450 可以是可移除的，不可移除的或其组合。示例性的硬件设备包括固态存储器，硬盘驱动器，光盘驱动器等。存储器 450 可选地包括在物理位置上远离处理器 410 的一个或多个存储设备。

存储器 450 包括易失性存储器或非易失性存储器，也可包括易失性和非易失性存储器两者。非易失性存储器可以是只读存储器（ROM，Read Only Memory），易失性存储器可以是随机存取存储器（RAM，Random Access Memory）。本申请实施例描述的存储器 450 旨在包括任意适合类型的存储器。

在一些实施例中，存储器 450 能够存储数据以支持各种操作，这些数据的示例包括程序、模块和数据结构或者其子集或超集，下面示例性说明。

操作系统 451，包括用于处理各种基本系统服务和执行硬件相关任务的系统程序，例如框架层、核心库层、驱动层等，用于实现各种基础业务以及处理基于硬件的任务；

网络通信模块 452，用于经由一个或多个（有线或无线）网络接口 420 到达其他计算设备，示例性的网络接口 420 包括：蓝牙、无线相容性认证（WiFi）、和通用串行总线（USB，Universal Serial Bus）等；

呈现模块 453，用于经由一个或多个与用户接口 430 相关联的输出装置 431（例如，显示屏、扬声器等）使得能够呈现信息（例如，用于操作外围设备和显示内容和信息的用户接口）；

输入处理模块 454，用于对一个或多个来自一个或多个输入装置 432 之一的一个或多个用户输入或互动进行检测以及翻译所检测的输入或互动。

在一些实施例中，本申请实施例提供的装置可以采用软件方式实现，图 3 示出了存

储在存储器 450 中的对象处理装置 455, 其可以是程序和插件等形式的软件, 包括以下软件模块: 第一获取模块 4551、第一移动模块 4552、第二获取模块 4553、第一确定模块 4554 和第一扩展模块 4555, 这些模块是逻辑上的, 因此根据所实现的功能可以进行任意的组合或进一步拆分。将在下文中说明各个模块的功能。

5 在另一些实施例中, 本申请实施例提供的装置可以采用硬件方式实现, 作为示例, 本申请实施例提供的装置可以是采用硬件译码处理器形式的处理器, 其被编程以执行本申请实施例提供的对象处理方法, 例如, 硬件译码处理器形式的处理器可以采用一个或多个应用专用集成电路 (ASIC, Application Specific Integrated Circuit)、DSP、可编程逻辑器件 (PLD, Programmable Logic Device)、复杂可编程逻辑器件 (CPLD, Complex Programmable Logic Device)、现场可编程门阵列 (FPGA, Field-Programmable Gate Array) 或其他电子元件。

将结合本申请实施例提供的终端的示例性应用和实施, 说明本申请实施例提供的对象处理方法。

15 本申请实施例提供一种对象处理方法, 应用于计算机设备, 该计算机设备可以是终端也可以是服务器, 在本申请实施例中, 以计算机设备为终端为例进行说明。图 4A 是本申请实施例提供的对象处理方法的一种实现流程示意图, 以下将结合图 4A 对本申请实施例提供的对象处理方法进行说明。

步骤 S101, 获取待处理的多个资源对象在目标容器中的初始填充结果。

20 这里, 多个资源对象可以是指两个或两个以上资源对象, 在本申请实施例中, 资源对象可以是对虚拟对象的三维模型进行剖分得到的二维多边形 (下述简称为多边形), 该多边形可以是带孔洞的多边形, 也可以是不带孔洞的多边形。在一些实施例中, 该资源对象可以是游戏制作行业中的 UV 岛, 其中, UV 岛的本质即为上述的多边形。另外, 该资源对象还可以是对实际对象进行展开及分块处理后得到的。目标容器用于承载资源对象, 该目标容器也可以为形状为多边形的二维物体。

25 作为示例, 初始填充结果可以是利用预设的装箱算法, 将待处理的多个资源对象依次添加到目标容器得到的。例如可以利用基于 NFP+遗传算法的装箱算法、基于 NFP 和最低重心原则的自动装箱算法等等。另外, 初始填充结果还可以是对利用预设的装箱算法确定出的装箱结果进行全局位置优化之后得到的。初始填充结果可以包括各个资源对象的中心点所在的位置, 还可以包括各个资源对象的外轮廓顶点的位置信息、目标容器的空余位置等。

步骤 S102, 基于所述初始填充结果, 将各个资源对象的对象中心移动至各自对应的可用空间的中心位置。

30 在一些实施例中, 可以根据初始填充结果依次确定将各个资源对象的对象中心移动至所述第 i 个资源对象所在可用空间的中心位置的移动信息, 并在基于所述移动信息, 确定所述第 i 个资源对象的移动距离大于预设的距离阈值时, 将所述第 i 个资源对象的对象中心移动至所述可用空间的中心位置。

步骤 S103, 获取所述各个资源对象的对称属性信息和包含关系信息。

35 资源对象的对称属性信息可以包括资源对象的左右对称属性信息和上下对称属性信息。左右对称属性信息能够表征资源对象是否为左右对称的, 上下对称属性信息能够表征资源对象是否为上下对称的。在判断资源对象是否为左右对称时, 将资源对象以经过资源对象的对象中心点且垂直于 x 轴的竖直中心线为分割线进行划分, 得到左右两个部分, 并确定左右两部分的重合率, 如果两者重合率大于预设的重合阈值 (例如 90%), 则确定该资源对象是左右对称的。判断资源对象是否为上下对称的实现过程与上述过程
45 是类似的。

每个资源对象的包含关系信息可以包括第一对象集合和第二对象集合，其中，第一对象集合中包括的是该资源对象所包含的资源对象的对象标识，第二对象集合中包括的是包含该资源对象的其他资源对象的对象标识。

判断第 i 个资源对象是否包含第 p 个资源对象在实现时，可以首先获取第 i 个资源对象的各个孔洞信息。在本申请实施例中，资源对象可以是二维多边形，当一个二维多边形内部包含有其他多边形时，被包含的多边形可以称为是该二维多边形的孔洞。资源对象的各个孔洞信息可以包括孔洞上的各个顶点名称、顶点位置以及顶点之间的连接关系。然后再从第 p 个资源对象的外边界上随机确定一个参考点，判断所述参考点是否位于第 i 个资源对象中的孔洞内部。作为示例，可以利用 CGAL 提供的库函数检查该参考点是否在第 i 个资源对象中的孔洞内部。在一些实施例中，依次判断该参考点是否位于第 i 个资源对象中的各个孔洞内部，如果该参考点不位于第 i 个资源对象中所有孔洞内部，那么确定该参考点不位于第 i 个资源对象中的孔洞内部，确定第 i 个资源对象不包含第 p 个资源对象；如果该参考点位于第 i 个资源对象中某个孔洞内部，则确定该参考点位于第 i 个资源对象中孔洞内部，确定所述第 i 个资源对象包含第 p 个资源对象。

步骤 S104，基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型。

其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种，各向同性扩展是指资源对象在各个方向的扩展比例是相同的，各向异性扩展是指资源对象在不同方向的扩展比例可以是相同的，也可以是不同的。

在一些实施例中，可以首先判断资源对象是否具备内部对称特征，当该资源对象不具备内部对称特征时，可以将该资源对象的扩展类型确定为各向同性扩展，如果该资源对象还包含有其他子资源对象，那么其他子资源对象的扩展类型也为各向同性扩展；当资源对象具备内部对称特征时，确定该资源对象是否被父资源对象包含，如果未被父资源对象包含，那么确定该资源对象的扩展类型为各向异性扩展；如果被其他父资源对象包含，如果其父资源对象的扩展类型为各向同性扩展，那么该资源对象也为各向同性扩展；如果其父资源对象的扩展类型为各向异性扩展，那么还需要确定其兄弟资源对象的是否具备内部对称特征，如果其兄弟资源对象均具备内部对称特征，确定该资源对象和其兄弟资源对象的扩展类型为各向异性扩展；如果存在至少一个不具备内部对称特征的兄弟资源对象，则确定该资源对象和其兄弟资源对象的扩展类型为各向同性扩展。

步骤 S105，基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象。

在一些实施例中，如果资源对象的扩展类型为各向同性扩展，首先确定出资源对象的第一扩展比例，确定该资源对象不包含其他子资源对象时，按照第一扩展比例对该资源对象进行扩展，得到扩展后的资源对象；如果该资源对象包含其他子资源对象时，那么其他子资源对象的扩展类型也为各向同性扩展，此时会对该资源对象和该资源对象包含的子资源对象同时按照第一扩展比例进行扩展，如此不仅可以避免该资源对象和其包含的子资源对象发生交叠，而且可以提高扩展效率。如果资源对象的扩展类型为各向异性扩展，那么确定资源对象在各个扩展方向的第二扩展比例，如果该资源对象不包含其他子资源对象，那么将该资源对象在各个扩展方向上按照对应的第二扩展比例进行扩展即可，如果该资源对象包含其他子资源对象，且其他子资源对象的扩展类型也为各向异性扩展时，那么将该资源对象和其子资源对象在各个扩展方向上按照对应的第二扩展比例同时进行扩展，从而提高扩展比例；如果该资源对象包含的子资源对象的扩展类型不为各向异性扩展时，那么则不能同时扩展该资源对象和子资源对象，从而避免该资源对象和子资源对象产生重叠。

在本申请实施例提供的对象处理方法中,在获取待处理的多个资源对象的初始填充结果之后,首先基于该初始填充结果进行全局均匀化处理,以将各个资源对象移动至各自对应的可用空间的中心位置;然后再获取所述各个资源对象的对称属性信息和包含关系信息,并基于资源对象的对称属性信息和包含关系信息确定各个资源对象的扩展类型,最后基于各个资源对象的扩展类型进行扩展处理,得到处理后的各个资源对象,由于是对目标容器中的每个资源对象都进行了扩展,并且在扩展之前将资源对象移动至中心位置,从而能够在有限的空间内实现最大比例的扩展,而且资源对象的扩展类型是基于资源对象的对称属性信息和包含信息自适应确定出的,其中扩展类型包括各向同性扩展和各向异性扩展,相较于相关技术中仅能够对资源对象进行各向同性扩展,在本申请实施例中,对于只能进行各向同性扩展的资源对象进行各向同性扩展,但是对于能够进行各向异性扩展的资源对象通过各向异性扩展能够实现对资源对象最大程度上的扩展,如此能够模拟出人工装箱调整的效果,进而最大限度的提高填充率。

在一些实施例中,在步骤 S105 之后,为了再进一步确定步骤 S105 得到的扩展结果是否已经达到预期的填充率,是否还具有更优的扩展结果,如图 4B 所示,还可以执行下述的步骤 S106 至 S112:

步骤 S106,在对第 1 个至第 N 个资源对象进行扩展处理后,确定所述目标容器的第一填充率。

步骤 S105 中是按照资源对象的标识,从第 1 个资源对象到第 N 个资源对象进行顺序扩展的,在扩展之后,可以基于目标容器的总容量和扩展后的各个资源对象已占用的空间,确定出目标容器的第一填充率,也即将各个资源对象已占用的空间求和,得到已占用总空间,再将已占用总空间除以目标容器的总容量即得到第一填充率。

步骤 S107,确定第一填充率是否小于预设的填充率阈值。

其中,当确定所述第一填充率小于预设的填充率阈值时,进入步骤 S108;当确定第一填充率大于或者等于该填充率阈值时,说明已经达到预期的扩展效果,此时进入步骤 S112。

步骤 S108,依次对第 N 个资源对象至第 1 个资源对象按照各自对应的扩展类型进行扩展处理,得到第二次处理后的第 N 个资源对象至第 1 个资源对象。

在该步骤中,将初始填充结果中的各个资源对象按照倒序,从第 N 个资源对象至第 1 个资源对象依次基于各自对象的扩展类型进行扩展处理,得到第二次处理后的第 N 个资源对象至第 1 个资源对象。

需要说明的是,在倒序对资源对象进行扩展处理时,对于单个资源对象来说,扩展过程与顺序对资源对象进行扩展处理的实现方式是类似的,只是处理顺序不同。

步骤 S109,确定所述目标容器的第二填充率。

步骤 S109 的实现过程与步骤 S106 确定第一填充率的实现过程是类似的,步骤 S109 在实现时,可参考步骤 S106 的实现过程。

步骤 S110,判断第二填充率是否大于第一填充率。

其中,当所述第二填充率大于所述第一填充率时,说明经过步骤 S108 的再次扩展处理,提高了目标容器的填充率,此时进入步骤 S111;当第二填充率小于或者等于第一填充率时,说明经过步骤 S108 的再次扩展处理,并没有提高目标容器的填充率,此时进入步骤 S112。

步骤 S111,将所述第二次处理后的第 N 个资源对象至第 1 个资源对象确定为目标扩展结果。

步骤 S112,将处理后的第 1 个资源对象至第 N 个资源对象确定为目标扩展结果。

在本申请实施例中,在顺序对目标容器中的各个资源对象进行扩展之后,通过上述

步骤 S106 至步骤 S112, 在确定目标容器的第一填充率未达到预设的填充率阈值时, 按照倒序对目标容器中的各个资源对象进行扩展, 并得到第二填充率, 最终将第一填充率和第二填充率中的较大值对应的扩展结果确定为目标扩展结果, 从而能够进一步提升目标容器的填充率。

5

在一些实施例中, 上述步骤 S102“基于所述初始填充结果, 将各个资源对象的对象中心移动至各自对应的可用空间的中心位置”可以通过如图 5 所示的步骤 S21 至步骤 S212 实现, 以下结合图 5 对各个步骤进行说明。

步骤 S21, 获取预设的移动迭代次数阈值和单个资源对象的移动次数阈值。

10

其中, 移动迭代次数阈值是大于或者等于单个资源对象的移动次数阈值的, 移动迭代次数阈值和移动次数阈值可以根据资源对象总数确定的, 还可以是用户根据自身需求自行设定的。

步骤 S22, 获取当前移动迭代次数。

在本申请实施例中, 可以设定当前移动迭代次数变量, 初始值为 0, 每完成第 1 个资源对象至第 N 个资源对象的一轮移动之后, 当前移动迭代次数加 1。

15

步骤 S23, 判断当前移动迭代次数是否小于移动迭代次数阈值。

其中, 当所述当前移动迭代次数小于所述移动迭代次数阈值时, 进入步骤 S24; 当前移动迭代次数大于或者等于该移动迭代次数阈值时, 进入步骤 S103。

步骤 S24, 获取第 i 个资源对象的当前移动次数。

20

$i=1, 2, \dots, N$, N 为资源对象总数, N 为大于 2 的整数。

步骤 S25, 判断第 i 个资源对象的当前移动次数是否小于移动次数阈值。

其中, 当所述第 i 个资源对象的当前移动次数小于所述移动次数阈值时, 进入步骤 S26, 当第 i 个资源对象的当前移动次数大于或者等于移动次数阈值时, 进入步骤 S29。

25

步骤 S26, 确定将所述第 i 个资源对象的对象中心移动至所述第 i 个资源对象所在可用空间的中心位置的移动信息。

在本申请实施例中, 第 i 个资源对象的对象中心可以用第 i 个资源对象对应的包围盒中心来表征, 其中, 第 i 个资源对象的包围盒可以是第 i 个资源对象的坐标轴对齐包围盒, 也可以是第 i 个资源对象的紧致包围盒, 资源对象的包围盒一般为矩形, 包围盒中心为矩形的两个对角线的交点。

30

在一些实施例中, 可以将第 i 个资源对象沿 x 轴分别向左、向右移动, 确定向左移动到与其他资源对象接触时的向左移动量, 确定向右移动到与其他资源对象接触时的向右移动量, 按照 $(\text{向右移动量}-\text{向左移动量})/2$ 计算出第 i 个资源对象在 x 轴方向的移动信息。移动信息可以包括移动量和移动方向, 其中, $(\text{向右移动量}-\text{向左移动量})/2$ 的绝对值表征移动量, $(\text{向右移动量}-\text{向左移动量})/2$ 的正负表征移动方向, 正表示向右移动, 负表示向左移动。同样地, 将第 i 个资源对象沿 y 轴分别向上、向下移动, 确定向上移动到与其他资源对象接触时的向上移动量, 确定向下移动到与其他资源对象接触时的向下移动量, 按照 $(\text{向上移动量}-\text{向下移动量})/2$ 计算出第 i 个资源对象在 y 轴方向的移动信息。

35

举例来说, 第 i 个资源对象向左移动量为 3, 向右移动量为 5, 向上移动量为 4, 向下移动量为 8, 那么在 x 轴方向的移动信息为 $(5-3)/2=1$, 也即移动量为 1, 移动方向为向右, 在 y 轴方向的移动信息为 $(4-8)/2=-2$, 也即移动量为 2, 移动方向为向下。

40

步骤 S27, 判断所述第 i 个资源对象的移动距离是否大于预设的距离阈值。

在一些实施例中, 首先基于步骤 S26 确定出来的移动信息确定出第 i 个资源对象的移动距离, 由于步骤 S26 确定出的移动信息包括 x 轴方向的移动信息 d_1 和 y 轴方向的

移动信息 d_2 ，那么第 i 个资源对应的移动距离即为 $\sqrt{d_1^2 + d_2^2}$ ，然后判断第 i 个资源对象的移动距离是否大于预设的距离阈值，当确定所述第 i 个资源对象的移动距离大于预设的距离阈值时，进入步骤 S28；当确定第 i 个资源对象的移动距离小于或者等于该距离阈值时，进入步骤 S29。

5 步骤 S28，按照所述移动信息将所述第 i 个资源对象的对象中心移动至所述可用空间的中心位置，并更新所述第 i 个资源对象的当前移动次数。

该步骤在实现，可以确定第 i 个资源对象是否包含其他子资源对象，如果第 i 个资源对象包含有子资源对象，那么在移动第 i 个资源对象时，是将第 i 个资源对象和其包含的子资源对象同时进行移动，此时在移动第 i 个资源对象时，第 i 个资源对象与其包含的子资源对象之间的相对关系不变，因此，在该步骤中仅更新第 i 个资源对象的当前移动次数，而不用更新第 i 个资源对象的子资源对象的当前移动次数。

步骤 S29，将 $i=i+1$ 。

步骤 S210，判断 i 是否等于 $N+1$ 。

其中，当 i 等于 $N+1$ ，说明已经完成第 1 个资源对象至第 N 个资源对象的本轮移动迭代过程，此时进入步骤 S211。当 i 不等于 $N+1$ ，说明还没有完成本轮移动迭代过程，此时进入步骤 S24。

步骤 S211，在完成第 1 个资源对象至第 N 个资源对象的当前移动迭代过程后，更新所述当前移动迭代次数，得到更新后的当前移动迭代次数。

更新当前移动迭代次数，也即将当前移动迭代次数加 1。

20 步骤 S212，判断更新后的当前移动迭代次数是否小于移动迭代次数阈值。

其中，确定所述更新后的当前移动迭代次数小于所述移动迭代次数阈值时，进入步骤 S24，继续移动所述第 1 个资源对象至第 N 个资源对象；确定所述更新后的当前移动迭代次数大于或者等于所述移动迭代次数阈值时，说明已经达到移动结束条件，进入步骤 S103。

25 由于在将各个资源对象移动至各自对应的可用空间的中心位置时，一个资源对象的移动会对其邻近的其他资源对象在对应可用空间的相对位置产生影响。比如说，在移动一个资源对象之前，该资源对象的相邻资源对象位于该相邻资源对象对应可用空间的中心位置，而移动该资源对象之后，该资源对象的相邻资源对象对应的可用空间的大小、可用空间的外边缘会发生变化，从而使得相邻资源对象不再位于该可用空间的中心位置。因此进行一轮移动迭代，往往不能实现大多数资源对象位于各自对应的可用空间的中心位置，因此，可以对资源对象进行多轮移动迭代，在本申请实施例中，对单个资源对象的移动次数阈值和整体的移动迭代次数阈值进行设定，通过设定单个资源对象的移动次数阈值，能够避免由于某个或某些资源对象发生循环移动而导致的超时；通过设定移动迭代次数阈值，能够避免在资源对象岛特别多的情况下，导致效果不明显的多次迭代并最终超时。在一些实施例中，可以根据资源对象的总数设定移动迭代次数阈值和单个资源对象的移动次数阈值，从而能够在保证执行效率的同时，达到最佳移动效果，以实现执行效率和执行效果之间的自适应平衡。

40 在一些实施例中，上述步骤 S104“基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型”，可以通过图 6 所示的步骤 S41 至步骤 S48 实现，以下结合图 6 对各个步骤进行说明。

步骤 S41，基于第 i 个资源对象的对称属性信息判断第 i 个资源对象是否具备内部对称特征。

其中， $i=1, 2, \dots, N$ ， N 为资源对象总数， N 为大于 2 的整数。

由于第*i*个资源对象的对象属性信息包括左右对称属性信息和上下对称属性信息，在一些实施例中，当基于左右对称属性信息确定第*i*个资源对象是左右对称，或者当基于上下对称属性信息确定第*i*个资源对象是上下对称的，确定第*i*个资源对象具备内部对称特征时，进入步骤 S42；当基于左右对称属性信息确定第*i*个资源对象不是左右对称，并且基于上下对称属性信息确定第*i*个资源对象不是上下对称的，确定第*i*个资源对象不具备内部对称特征时，进入步骤 S45。

也就是说，第*i*个资源对象满足左右对称和上下对称至少之一时，确定第*i*个资源对象具备内部对称特征；第*i*个资源对象既不满足左右对称又不满足上下对称，确定第*i*个资源对象不具备内部对称特征。

步骤 S42，基于第*i*个资源对象的包含关系信息判断第*i*个资源对象是否被父资源对象包含。

由于第*i*个资源对象的包含关系信息中包括第一对象集合和第二对象集合，其中，第一对象集合中包括的是该资源对象所包含的子资源对象的对象标识，第二对象集合中包括的是包含该资源对象的父资源对象的对象标识，在一些实施例中，可以判断第二对象集合是否为空，当第二对象集合不为空，说明第*i*个资源对象被父资源对象包含，此时进入步骤 S43。当第二对象集合为空，说明第*i*个资源对象未被父资源对象包含，此时进入步骤 S48。

步骤 S43，获取所述父资源对象的扩展类型。

步骤 S44，判断该父资源对象的扩展类型是否为各向同性扩展。

其中，当所述父资源对象的扩展类型为各向同性扩展时，进入步骤 S45；当该父资源对象的扩展类型为各向异性扩展时，进入步骤 S46。

步骤 S45，确定第*i*个资源对象的扩展类型为各向同性扩展。

步骤 S46，获取所述父资源对象包含的其他子资源对象。

步骤 S47，判断其他子资源对象是否均具备内部对称特征。

其中，当基于所述其他子资源对象的对称属性信息，确定所述其他子资源对象均具备内部对称特征时，进入步骤 S48；当基于所述其他子资源对象的对称属性信息，确定至少存在一个其他子资源对象不具备内部对称特征时，进入步骤 S45。

步骤 S48，确定第*i*个资源对象的扩展类型为各向异性扩展。

在上述步骤 S41 至步骤 S48 所在实施例中，在确定第*i*个资源对象的扩展类型时，如果第*i*个资源对象不具备内部对称特征，则直接将第*i*个资源对象的扩展类型确定为各向同性扩展，从而保证在进行扩展处理后不发生形变，而如果第*i*个资源对象具备内部对称特征时，需要进一步确定第*i*个资源对象是否被父资源对象包含，如果没有被包含就可以将第*i*个资源对象的扩展类型确定为各向异性扩展，进而能够实现各个方向的最大扩展，而如果第*i*个资源对象被父资源对象包含时，如果父资源对象的扩展类型为各向同性扩展，那么为了避免扩展后造成资源对象之间的重叠，将第*i*个资源对象的扩展类型也确定为各向同性扩展，而如果父资源对象的扩展类型为各向异性扩展，那么如果该父资源对象的所有子资源对象均具备内部对称特征时，就可以将第*i*个资源对象以及其他子资源对象的扩展类型确定为各向异性扩展，如此在扩展过程中不仅能避免不同资源对象之间的重叠，还能保证在各个扩展方向都进行了最大比例的扩展，从而提高目标容器的填充率。

在本申请实施例中，在对多个资源对象进行装箱处理，所得到的初始填充结果中，会存在一个资源对象内部还嵌套包含有其他资源对象的情况，也可能存在几个资源对象为对称关系的情况，此时，上述步骤 S105“基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象”通过以下步骤 S51 至步骤

S519 实现, 以下对各个步骤进行说明。

步骤 S51, 判断目标容器中是否存在第 i 个资源对象的对称对象。

其中, $i=1, 2, \dots, N$, N 为资源对象总数, N 为大于 2 的整数; 当确定所述目标容器中不存在第 i 个资源对象的对称对象时, 进入步骤 S52; 当确定目标容器中存在第 i 个资源对象的对称对象时, 进入步骤 S517。

步骤 S52, 判断第 i 个资源对象的扩展类型是否为各向同性扩展。

当确定第 i 个资源对象的扩展类型为各向同性扩展时, 进入步骤 S53; 当确定第 i 个资源对象的扩展类型为各向异性扩展时, 进入步骤 S58。

步骤 S53, 将所述第 i 个资源对象的对象中心移动至对应的可用空间的中心位置。

该步骤的实现过程与步骤 S102 的实现过程是类似的, 在实现时可参考步骤 S102 的实现过程。

需要说明的是, 虽然在步骤 S102 中已经将目标容器中的多个资源对象进行了全局均匀化处理, 但是在对资源对象进行扩展处理后, 会对未进行扩展处理的资源对象的位置产生影响, 因此会使得资源对象不再处于对应可见空间的中心位置, 因为在对第 i 个资源对象进行扩展之前, 再次将第 i 个资源对象的对象中心移动至对应的可用空间的中心位置, 以保证确定出的最大扩展比例。

步骤 S54, 确定所述第 i 个资源对象的第一扩展比例。

在一些实施例中, 可以采用双向射线法确定第 i 个资源对象的第一扩展比例: 首先从第 i 个资源对象的对象中心, 沿着第 i 个资源对象的外边界顶点朝外发射射线, 得到第 i 个资源对象外部的其他资源对象的交点, 并据此计算对应的第一候选扩展比例; 然后从第 i 个资源对象外部的其他资源对象的外边界的顶点, 朝第 i 个资源对象的对象中心发射射线, 得到和第 i 个资源对象的外边界的最近交点, 并据此计算对应的第二候选扩展比例。将第一候选扩展比例和第二候选扩展比例中的最小者作为第一扩展比例, 从而避免进行扩展时与其他资源对象重叠。

在一些实施例中, 上述步骤 S54“确定所述第 i 个资源对象的第一扩展比例”可以通过以下步骤实现:

步骤 S541, 获取位于所述第 i 个资源对象外部的其他资源对象的外边界信息、所述第 i 个资源对象的外边界信息和所述第 i 个资源对象的对象中心。

获取第 i 个资源对象的外边界信息在实现时, 可以首先获取资源对象上的每条边, 以及各条边的两个相邻面, 然后判断各条边的相邻面是否均属于资源对象内部, 如果一条边的相邻面均属于资源对象内部, 则确定该条边为内部边, 否则确定该条边为外边界, 将一个资源对象的各个外边界连接起来, 就构成了该资源对象的外边界。

第 i 个资源对象的对象中心可以用第 i 个资源对象的包围盒的中心来表示。第 i 个资源对象外部的其他资源对象的外边界信息可以包括第 i 个资源对象外部的其他资源对象外边界上的顶点的顶点标识、顶点索引等信息, 第 i 个资源对象的外边界信息包括第 i 个资源对象外边界上的顶点标识、顶点索引等信息。

步骤 S542, 从所述第 i 个资源对象的对象中心向所述第 i 个资源对象的外边界上的各个第一顶点发射第一射线, 确定各条第一射线与所述其他资源对象的外边界的各个第一最近交点。

在一些实施例中, 可以基于第 i 个资源对象外部的其他资源对象的外边界信息构建第一空间搜索树, 该第一空间搜索树上的图元为其他资源对象的外边界上的连接边, 然后通过第一空间搜索树确定各条第一射线与其他资源对象的外边界的各个第一最近交点。

步骤 S543, 基于所述对象中心的中心坐标、所述各个第一顶点的第一顶点坐标和

各个第一最近交点的第一交点坐标，确定各个第一顶点对应的第一候选扩展比例。

在一些实施例中，该步骤可以通过以下步骤实现：

步骤 S5431，获取间隔距离。

该间隔距离对应其他实施例中的出血值，该间隔距离可以是预先设置的，也可以是
5 根据装箱空间的分辨率自动确定出来的，例如，可以假设每 128 个像素取一个像素作为
间隔距离，那么 256*256 分辨率通常采用 2 个像素值作为间隔距离，如此根据装箱空间
的分辨率自动计算并预留间隔距离，能够有效避免人工装箱手动保留间隔距离可能引起
的错误。

步骤 S5432，基于该间隔距离和所述各个第一最近交点的交点坐标确定各个第一最
10 近交点对应的第一目标点的第一目标坐标。

其中，第一目标点为第一射线上的点，第一目标点与第一最近交点之间的距离为该
间隔距离，且第一目标点在所述第一最近交点和对象中心之间。

步骤 S5433，基于所述对象中心的中心坐标和所述各个第一顶点的第二顶点坐标确
定所述各个第一顶点对应的第二距离。

15 在一些实施例中，在已知对象中心的中心坐标和各个第一顶点的第二顶点坐标时，
可以基于距离公式，确定出各个第一顶点对应的第二距离。该第二距离也即第一顶点与
对象中心之间的距离。

步骤 S5434，基于所述对象中心的中心坐标和各个第一目标点的第一目标坐标确定
所述各个第一顶点对应的第二距离。

20 与步骤 S5433 类似，在已知对象中心的中心坐标和各个第一目标点的第一目标坐标
的前提下，基于距离公式即可确定出各个第一顶点对应的第二距离。第二距离也即第一
目标点与对象中心之间的距离。

步骤 S5435，基于所述各个第一顶点对应的第二距离和第二距离，确定各个第一顶
点对应的第二候选扩展比例。

25 在一些实施例中，将各个第一顶点对应的第二距离除以第一距离即得到了各个第一
顶点对应的第二候选扩展比例。

步骤 S544，从所述其他资源对象的外边界上的各个第二顶点向所述第 i 个资源对象
的对象中心发射第二射线，确定各条第二射线与所述第 i 个资源对象的外边界的各个第
二最近交点。

30 在一些实施例中，可以基于第 i 个资源对象的外边界信息构建第二空间搜索树，该
第二空间搜索树上的图元为第 i 个资源对象的外边界上的连接边，然后通过第二空间搜
索树确定各条第二射线与其他资源对象的外边界的各个第二最近交点。

步骤 S545，基于所述对象中心的中心坐标、所述各个第二顶点的第二顶点坐标和
各个第二最近交点的第二交点坐标，确定各个第二顶点对应的第二候选扩展比例。

35 这里，首先基于该间隔距离和所述各个第二最近交点的交点坐标确定各个第二最近
交点对应的第二目标点的第二目标坐标。其中，第二目标点为第二射线上的点，第二目
标点与第二顶点之间的距离为该间隔距离，且第二目标点在所述第二顶点和对象中心之
间。再基于所述对象中心的中心坐标和所述各个第二最近交点的交点坐标确定所述各个
第二顶点对应的第五距离，并基于所述对象中心的中心坐标和各个第二目标点的第二目
40 标坐标确定所述各个第二顶点对应的第六距离；各个第二顶点对应的第六距离除以对
应的第五距离，得到各个第二顶点对应的第二候选扩展比例。

步骤 S546，将多个第一候选扩展比例和多个第二候选扩展比例中的最小值确定为
所述第 i 个资源对象的第一扩展比例。

在上述步骤 S541 至步骤 S546 所在的实施例中，在确定第 i 个资源对象进行各向同

性扩展时采用的第一扩展比例时,采用双向射线法,首先从第*i*个资源对象的对象中心,沿着第*i*个资源对象的外边界顶点朝外发射第一射线,得到第*i*个资源对象外部的与其他资源对象的第一最近交点,并据此计算对应的第一候选扩展比例;然后从第*i*个资源对象外部的其他资源对象的外边界的顶点,朝第*i*个资源对象的对象中心发射第二射线,得到和第*i*个资源对象的外边界的第二最近交点,并据此计算对应的第二候选扩展比例,并且在计算第一候选扩展比例和第二候选扩展比例的时候,预留出了间隔距离,也即预留出了部分空隙,从而能够避免在进行资源对象扩展时与其他资源对象发生交叠。

下面接续步骤 S54 继续进行说明。

步骤 S55, 确定第*i*个资源对象是否包含其他子资源对象。

其中,当基于所述第*i*个资源对象的包含关系信息,确定所述第*i*个资源对象中包含有其他子资源对象时,进入步骤 S56;当确定第*i*个资源对象不包含其他子资源对象时,进入步骤 S57。

步骤 S56, 将所述第*i*个资源对象和所述第*i*个资源对象中包含的其他子资源对象按照所述第一扩展比例进行扩展处理,得到处理后的第*i*个资源对象和处理后的其他子资源对象。

在一些实施例中,是以第*i*个资源对象的对象中心为扩展中心,将第*i*个资源对象和第*i*个资源对象中包含的其他子资源对象按照第一扩展比例进行扩展处理,得到处理后的第*i*个资源对象和处理后的其他子资源对象。

步骤 S57, 将所述第*i*个资源对象按照所述第一扩展比例进行扩展处理,得到处理后的第*i*个资源对象。

当确定第*i*个资源对象不包含其他子资源对象时,就仅需要对第*i*个资源对象进行扩展处理即可,在一些实施例中,以第*i*个资源对象的对象中心为扩展中心按照第一扩展比例对第*i*个资源对象进行各向同性扩展处理,得到处理后的第*i*个资源对象。

步骤 S58, 获取所述第*i*个资源对象的主轴信息,并基于所述第*i*个资源对象的主轴信息确定旋转角度和第一旋转方向。

第*i*个资源对象的主轴是指经过第*i*个资源对象的对象中心的多条基线中宽度最大的一条基线。第*i*个资源对象的主轴信息为第*i*个资源对象的一种属性信息,第*i*个资源对象的主轴信息包括主轴与*x*轴之间的夹角,在将第*i*个资源对象填充至目标容器或者在移动第*i*个资源对象时,第*i*个资源对象的主轴信息会发生变化。假设第*i*个资源对象的主轴信息为45度,那么根据该主轴信息,可以确定旋转角度为45度,第一旋转方向为顺时针。

步骤 S59, 将*N*个资源对象按照所述第一旋转方向旋转所述旋转角度,得到旋转后的*N*个资源对象。

在该步骤中,是将目标容器中的*N*个资源对象均进行了旋转,从而保证第*i*个资源对象与其他资源对象之间的相对位置保持不变。其中,旋转后的第*i*个资源对象的主轴为水平方向,也即旋转后的第*i*个资源对象的主轴与*x*轴平行。

步骤 S510, 将所述旋转后的第*i*个资源对象的对象中心移动至对应的可用空间的中心位置。

该步骤的实现过程与步骤 S102 的实现过程是类似的,在实现时可参考步骤 S102 的实现过程。

步骤 S511, 确定所述旋转后的第*i*个资源对象在预设的各个扩展方向上的第二扩展比例。

在一些实施例中,可以从旋转后的第*i*个资源对象的外边界上确定各个扩展方向对应的多个候选点;从各个候选点延对应的各个扩展方向发射第三射线,确定各条第三射

线与所述其他资源对象的外边界的最近交点，基于最近交点确定旋转后的第 i 个资源对象在各个扩展方向上的第二扩展比例。

在一些实施例中，上述步骤 S511“确定所述旋转后的第 i 个资源对象在预设的各个扩展方向上的第二扩展比例”可以通过以下步骤实现：

5 步骤 S5111，获取位于所述旋转后的第 i 个资源对象外部的其他资源对象的外边界信息和所述旋转后的第 i 个资源对象的包围盒。

步骤 S5112，从所述旋转后的第 i 个资源对象的外边界上确定第 j 个扩展方向对应的多个候选点。

10 由于资源对象的外边界的顶点是按照逆时针方向排序的，在一些实施例中，如果第 j 个扩展方向为向上方向，那么将第 i 个资源对象的外边界上起点 x 坐标大于终点 x 坐标的连接边确定为候选连接边，将候选连接边的顶点确定为候选点，然后将第 i 个资源对象外部的其他资源对象上的外边界上的顶点沿第 j 个扩展方向（也即向上）的反方向（也即向下）做投影，将投影到候选连接边上的点也确定为第 j 个扩展方向上的候选点。类似的，如果第 j 个扩展方向为向下，那么将第 i 个资源对象的外边界上起点 x 坐标小于终点 x 坐标的连接边确定为候选连接边，将候选连接边的顶点确定为候选点，然后将第 i 个资源对象外部的其他资源对象上的外边界上的顶点沿第 j 个扩展方向（也即向下）的反方向（也即向上）做投影，将投影到候选连接边上的点也确定为第 j 个扩展方向上的候选点。

20 步骤 S5113，从各个候选点沿所述第 j 个扩展方向发射第三射线，确定各条第三射线与所述其他资源对象的外边界的第三最近交点。

j 为小于或者等于 M 的正整数， M 为扩展方向总数， M 为大于或者等于 1 的整数。在一些实施例中，可以基于第 i 个资源对象的外边界信息构建第二空间搜索树，该第二空间搜索树上的图元为第 i 个资源对象的外边界上的连接边，然后通过第二空间搜索树确定各条第三射线与其他资源对象的外边界的各个第三最近交点。

25 步骤 S5114，确定所述各个候选点到所述包围盒的目标边的各个第三距离，并确定所述各个第三最近交点到所述目标边的各个第四距离。

所述目标边为所述包围盒上与所述第 j 个扩展方向的相反方向对应的边。比如第 j 个扩展方向为向上，那么目标边为包围盒上的下侧水平边，第 j 个扩展方向为向左，那么目标边为包围盒上的右侧竖直边。

30 步骤 S5115，基于所述各个第三距离和对应的所述各个第四距离，确定各个候选点对应的第三候选扩展比例。

在一些实施例中，将各个候选点的第四距离除以对应的第三距离，即得到了各个候选点对应的第三候选扩展比例。

35 步骤 S5116，将多个第三候选扩展比例中的最小值确定为所述第 j 个扩展方向对应的第二扩展比例。

在上述步骤 S5111 至步骤 S5116 所在的实施例中，确定对第 i 个资源对象进行各向异性扩展时在各个扩展方向上的第二扩展比例时，可以首先从第 i 个资源对象的外边界上确定各个扩展方向对应的多个候选点，然后从各个候选点向各个扩展方向发射第三射线，确定各条第三射线与所述其他资源对象的外边界的第三最近交点，并根据第 i 个资源对象的包围盒上各个扩展方向的反方向对应的目标边、第三最近交点以及候选点确定出各个扩展方向对应的第二扩展比例，如此能够实现对第 i 个资源对象在各个扩展方向上进行最大程度的扩展。

下面接续步骤 S511 继续进行说明。

步骤 S512，确定所述旋转后的第 i 个资源对象中是否包含有其他子资源对象。

其中，当基于所述第 i 个资源对象的包含关系信息，确定所述旋转后的第 i 个资源对象中包含有其他子资源对象时，进入步骤 S513；当确定旋转后的第 i 个资源对象未包含其他资源对象时，进入步骤 S514。

5 步骤 S513，将所述旋转后的第 i 个资源对象和所述旋转后的第 i 个资源对象中包含的其他子资源对象在所述各个扩展方向上按照各自对应的第二扩展比例进行扩展，得到扩展后的第 i 个资源对象和扩展后的其他子资源对象。

由于在步骤 S58 和步骤 S59 中已经将第 i 个资源对象的主轴旋转至水平方向，那么第 i 个资源对象的包围盒的相邻边一个是水平方向，一个是竖直方向，扩展方向可以是上下左右四个方向至少之一，在一些实施例中，以第 i 个资源对象的包围盒上各个扩展方向对应的反方向上的边为基准，将第 i 个资源对象和第 i 个资源对象中包含的其他子资源对象在各个扩展方向上按照各自对应的第二扩展比例进行扩展，得到扩展后的第 i 个资源对象和扩展后的其他子资源对象。

10 以扩展方向为向上为例进行说明，是以第 i 个资源对象包围盒下侧的水平边为基准（也即下侧的水平边保持不动），将第 i 个资源对象和第 i 个资源对象中包含的其他子资源对象向上扩展对应的第二扩展比例。

15 步骤 S514，将旋转后的第 i 个资源对象在所述各个扩展方向上按照各自对应的第二扩展比例进行扩展，得到扩展后的第 i 个资源对象。

步骤 S515，将所述扩展后的第 i 个资源对象和所述扩展后的其他子资源对象按照第二旋转方向旋转所述旋转角度，得到处理后的第 i 个资源对象和处理后的其他子资源对象。

20 所述第二旋转方向为所述第一旋转方向的反方向。在该步骤中，将该目标容器中的 N 个资源对象按照与第一旋转方向的反方向旋转该旋转角度，从而将目标容器中的 N 个资源对象恢复至旋转前的姿态。

25 步骤 S516，基于所述第 i 个资源对象的扩展类型，分别确定所述第 i 个资源对象的第一参考扩展比例和所述对称对象的第二参考扩展比例。

当第 i 个资源对象具有对称对象时，那么第 i 个资源对象的扩展类型和其对称对象的扩展类型是相同的。在一些实施例中，如果第 i 个资源对象的扩展类型为各向同性扩展，那么可以利用参考下述步骤 S541 至步骤 S546 类似的过程，分别确定出第 i 个资源对象的第一参考扩展比例和所述对称对象的第二参考扩展比例；如果第 i 个资源对象的扩展类型为各向异性扩展，那么可以参考上述步骤 S58 至步骤 S511 确定出第 i 个资源对象在各个扩展方向的第一参考扩展比例和所述对称对象在各个方向的第二参考扩展比例。

步骤 S517，将所述第一参考扩展比例和所述第二参考扩展比例中的最小值，确定为所述第 i 个资源对象和所述对称对象的目标扩展比例。

35 在一些实施例中，如果第 i 个资源对象的扩展类型为各向同性扩展，那么第 i 资源对象和对称对象各自对应一个参考扩展比例，那么将第一参考扩展比例和至少一个第二参考扩展比例中的最小值确定为第 i 个资源对象和所述对称对象的目标扩展比例；如果第 i 个资源对象的扩展类型为各向异性扩展，那么第 i 个资源对象和对称对象在各个扩展方向上均对应一个参考扩展比例，假设扩展方向有四个，分别为上、下、左、右，那么将第 i 个资源对象和对称对象在每个扩展方向上对应的参考扩展比例中最小值确定为各个扩展方向的目标扩展比例。

40 步骤 S518，按照所述目标扩展比例对所述第 i 个资源对象和所述对称对象进行扩展处理，得到处理后的第 i 个资源对象和处理后的对称对象。

在一些实施例中，如果对第 i 个资源对象和对称对象进行扩展后，与其他资源对象

发生交叠，那么将第*i*个资源对象和对称对象恢复至扩展前。

通过上述步骤 S51 至步骤 S518，能够在对第*i*个资源对象进行扩展时，基于第*i*个资源对象是否具有对称对象以及第*i*个资源的扩展类型进行不同的扩展方式，其中第*i*个资源对象不具有对称对象时，那么就基于第*i*个资源对象的扩展类型进行扩展，
5 然而当第*i*个资源对象中包含其他子资源对象时，还可以将第*i*个资源对象和第*i*个资源对象的子资源对象同时进行扩展，如此不仅能够避免第*i*个资源对象扩展时与其内部包含的子资源对象之间发生交叠，并且还能同时扩展多个资源对象，从而提高对象处理效率；而当第*i*个资源对象具有对称对象时，需要分别确定出第*i*个资源对象的第一参考扩展比例和对称对象的第二参考扩展比例，并取最小值作为目标扩展比例，从而保证
10 第*i*个资源对象和对称对象在扩展后仍能够保持大小一致。

下面，将说明本申请实施例在一个实际的应用场景中的示例性应用。

对于诸如 2UV 展开这一特定类型的装箱问题，往往允许对 UV 岛进行一定幅度的扩展，从而改进装箱效果。例如对于图 7 所示的 2UV 展开，原始装箱结果如左侧 701
15 所示，那么大部分美术工作人员会对这部分 UV 岛进行扩展，达到右侧 702 所示的效果，以便于在渲染时提高内存利用率，从而最大化纹理分辨率。

本申请实施例提供的对象处理方法，可以应用于自动 2UV 展开工具中，可以对具有烘焙需求的多种游戏场景模型实现 2UV 展开的自动化，包括角色模型、车辆模型、建筑模型等等。在一些实施例中，终端响应于针对自动 2UV 展开工具中的“模型导入”
20 的操作指令，呈现模型选择界面，并基于通过模型选择界面接收到的模型选择操作，确定目标三维模型数据，终端响应于接收到的 UV 展开操作指令，通过自动 2UV 工具对目标三维模型进行面片剖分，从而得到多个二维的 UV 岛（资源对象），再对多个 UV 岛进行装箱处理，得到初始装箱结果，并基于各个 UV 岛的对称属性信息和包含关系信息确定各个 UV 岛的扩展类型，其中，扩展类型包括各向同性扩展和各向异性扩展中的一种，基于各个 UV 岛的扩展类型，对各个 UV 岛进行扩展处理，得到处理后的各个
25 UV 岛。对原始装箱结果中的每个 UV 岛都进行了扩展，并且在扩展之前将 UV 岛移动至中心位置，从而能够在有限的空间内实现最大比例的扩展，如此能够最大限度的减少空余空间，从而提高目标容器的填充率。之后利用最终的装箱结果对三维模型进行贴图处理，在实现时，响应于接收到的 UV 岛选择指令，确定本次进行贴图的目标 UV 岛，
30 终端响应于针对目标 UV 岛的贴图操作，呈现目标 UV 岛的多个候选贴图图像，然后终端接收贴图选择操作，并基于该贴图选择操作确定目标贴图图像，基于该目标贴图图像对目标 UV 岛进行贴图处理。由于在装箱过程中对 UV 岛进行了扩展处理，因此在对三维模型中的各个 UV 岛进行贴图时，能够提高纹理映射的分辨率。

在本申请实施例中，通过分析美术人工的做法，提炼蕴含在人工操作中的思想来设计对应的算法。以图 8 左侧的初始装箱结果 801 为例，对于美术人工而言，会移动各个
35 UV 岛，使得不同 UV 岛之间的间距大致相同；然后开始对不同的 UV 岛进行分类扩展。例如对于右上侧矩形框 8011 所示的具有的内部对称特征的 UV 岛，为了最大限度地利用其周围空间，会较多地进行垂直方向的拉伸，同时稍微进行水平方向的拉伸，得到如图 8 中 802 右上侧矩形框 8021 所示的扩展结果。虽然这种各向异性的拉伸会导致发生一定的形变，但是由于该 UV 岛相对比较规整，所以并不会引起纹理的扭曲。而对于左
40 下侧矩形框 8012 所示的没有内部对称特征的 UV 岛，为了避免引起纹理扭曲，往往对它进行各向同性的等比例扩展。得到如图 8 中 802 左下侧矩形框 8022 所示的扩展结果。

基于以上分析，本申请实施例提供的对象处理方法在实现时，可以分为图 9 所示的两个步骤：

45 步骤 S901，全局均匀化处理。

在一些实施例中，对所有 UV 岛进行全局均匀化处理，以使得不同 UV 岛之间的间距大致相同。

步骤 S902，分类扩展。

其中，图 9 所示的 UV 岛仅仅包含其外部边界和内部孔洞，而不再包含多边形区域的内部边。这是因为区域内部边并不会对装箱过程和结果有任何影响。所以为了提高计算效率，首先提取各个 UV 岛的外轮廓（包含外部边界和内部孔洞），并且在后续装箱优化中仅仅考虑各个 UV 岛的外轮廓。

以下对上述步骤 S901 和步骤 S902 的实现过程进行说明。

一、全局均匀化

对于一个给定的装箱结果，所要扩展的 UV 岛未必处于其所在空隙的中心位置，如图 10 的两个圆圈 1001 和 1002 中的 UV 岛所示。这种情况将导致其扩展受到影响。为了最大程度地消除这种影响，需要首先对所有 UV 岛进行全局“摇匀”，也即全局均匀化处理。进行全局均匀化处理的原理是：想象各个 UV 岛之间的空隙处存在着“虚拟弹簧”，其弹力将促使各个 UV 岛之间因相互排斥而运动，最终达到受力平衡。基于这一思想，全局均匀化处理在实现时可以通过以下步骤实现：

步骤 S9011，根据 UV 岛数量，定义最大迭代次数 P 和单 UV 岛最大移动次数 Q。

其中，P 为正整数，Q 为小于或者等于 P 的正整数。

步骤 S9012，对于每个 UV 岛，如果确定其累计移动次数小于 N，则计算它所在的局部空隙的中心点，并移动该 UV 岛，使其包围盒中心点同局部空隙中心点重合，并更新其累计移动次数。

步骤 S9013，如果在步骤 S9012 中没有 UV 岛被移动，则算法结束。

如果在一轮移动迭代中，所有的 UV 岛的包围盒中心点均位于各自对应的局部空隙中心，那么在本轮迭代中没有 UV 岛被移动，此时全局均匀化处理结束。

步骤 S9014，更新当前迭代次数，如果确定已经达到了最大迭代次数 M，则算法结束，否则转步骤 S9012。

在本申请实施例中，之所以定义单个 UV 岛的最大移动次数 Q，是因为在实践中发现在某些情况下会发生循环移动导致超时；之所以定义最大迭代次数 P，也是发现对于 UV 岛特别多的情况，如果不定义最大迭代次数，则容易导致效果不明显的多次迭代并最终超时。在实践中，为了达到效率和效果之间的自适应平衡，可以设置当 UV 岛数量小于 10 个时， $P=Q=8$ ；当 UV 岛数量小于 20 时， $P=Q=5$ ；而当 UV 岛数量大于 20 时， $P=Q=3$ 。

二、分类扩展

在诸如 2UV 展开这类应用中，往往对不同的 UV 岛存在不同的扩展需求。例如对于图 11 左侧中圆圈 1101 和 1102 内的 UV 岛，由于它们存在明显的内部对称特征，所以对其 x 方向和 y 方向分别施加不同的扩展比例并不会引起纹理扭曲。在这种情况下，在本申请实施例中，原则是尽量充分扩展 UV 岛，以最大限度地提高填充率。将这种扩展方式称为各向异性扩展，得到的扩展结果如图 11 右侧椭圆 1103 和 1104 所示。

而对于不存在内部对称特征的 UV 岛，如图 12 中左侧的圆圈 1201 内所示，一旦进行各向异性的扩展，则容易引起纹理扭曲。在这种情况下，原则是首先保证纹理不扭曲，然后再对 UV 岛进行尽量充分的扩展。此时，必须对 UV 岛的各个方向施加相同的扩展比例。在本申请实施例中，称这种扩展方式为各向同性扩展，其扩展结果如图 12 中的右侧中圆圈 1202 所示。

以上多边形分类仅仅是针对 2UV 展开这一特定需求而设计的。对于其它需求，可以设计不同的 UV 岛分类，并对其实施不同类型的扩展。

在对各向同性扩展和各向异性扩展的实现过程进行说明之前,首先对各向同性扩展和各向异性扩展的一种数据结构:基于轴对齐包围盒的空间搜索树(AABB Tree, Axis-Aligned Bounding Box Tree)进行说明。AABB树是一种基于K-树原理的空间搜索树,其每个节点代表一个几何图元的空间包围盒。该数据结构可以快速报告查询图元是否与树中的图元相交,以及具体相交类型、相交位置等。如果查询图元是点,则还可以查询最近距离。

在本申请实施例中,采用(带孔)多边形的边来构造空间搜索树。基于这种数据结构,分类扩展可以通过以下步骤实现:

- 步骤 S9021, 记录每个 UV 岛的孔洞所包含的 UV 岛集合;
- 步骤 S9022, 按照正序, 对每个 UV 岛分别执行步骤 S9023, 得到顺序扩展结果;
- 步骤 S9023, 如果当前待扩展 UV 岛具有内部对称特征, 则执行各向异性扩展, 否则执行各向同性扩展;
- 步骤 S9024, 如果顺序扩展结果的填充率已经满足要求, 则算法结束, 否则转步骤 S9025;
- 步骤 S9025, 按照逆序, 对每个 UV 岛再次分别执行步骤 3), 得到逆序扩展结果;
- 步骤 S9026, 比较顺序扩展结果和逆序扩展结果, 选择填充率最高者, 作为最终的扩展结果。

在本申请实施例中,之所以在步骤 S9021 中首先记录每个 UV 的孔洞中所包含的 UV 岛集合,是因为如果当前 UV 岛的孔洞中包含其它 UV 岛,那么在扩展过程中其内部 UV 岛也需要做同步扩展。这样不仅可以避免和内部 UV 岛发生交叠,而且可以提高扩展效率。以下对各向异性扩展和各向同性扩展的实现过程进行说明。

各向异性扩展:各向异性扩展允许 UV 岛在不同方向上具有不同的扩展比例,因此可以在不同方向上分别进行独立扩展。在本申请实施例中,各向异性扩展的实现过程可以包括以下步骤:

- 步骤 S231, 旋转所有 UV 岛,使得当前 UV 岛的主轴和 x 轴平行,并记录旋转角度。
- 在一些实施例中,可以通过当前 UV 岛的主轴信息确定将主轴旋转至与 x 轴平行时的旋转角度和旋转方向,然后旋转所有 UV 岛,从而使得当前 UV 岛的主轴和 x 轴平行。
- 步骤 S232, 提取当前 UV 岛外部的所有 UV 岛的边界线段,并构建对应的空间搜索树 T。
- 步骤 S233, 将当前 UV 岛移动到其局部间隙的中心。
- 在一些实施例中,将当前 UV 岛的包围盒的中心移动到当前 UV 岛对应的局部间隙的中心。
- 步骤 S234, 沿着上、下、左、右四个方向通过下述两个步骤分别做最大扩展:
 - A) 对于当前 UV 岛外边界上的每个候选点,沿着当前方向向外发射射线,利用空间搜索树 T 计算最近交点,从而确定当前顶点(线段)朝外的最大扩展比例;
 - B) 在所有的最大扩展比例中,选择最小者作为该方向上的最终扩展比例,以其反方向为基准,扩展该 UV 岛及其孔洞中所包含的所有 UV 岛。
- 步骤 S235, 根据步骤 S231 所记录的旋转角度,将所有 UV 岛旋转回原始位置。

图 13 为本申请实施例提供的各向异性扩展的实现过程示意图,其中,图 13 中 1301 为原始装箱结果中的一部分,其中待扩展 UV 岛为 1301 中心部位的矩形,由于该待扩展 UV 岛具备内部对称特征,因此进行各向异性扩展,在实现时向上、下、左、右扩展,其中,通过图 13 中的 1302 可以看出,在向上扩展时,可以从矩形上边界的边上选择 3 个点作为候选点,向上发射射线,以确定向上扩展的扩展比例;向下扩展时,可以如图 13 中的 1303 所示,从矩形的下边界上选择 2 个点作为候选点,向下发射射线,以确定

向下扩展的扩展比例；向左进行扩展时，可以如图 13 中的 1304 所示，从矩形的左边界上选择 9 个候选点，向左发射射线，以确定向左扩展时的扩展比例；向右进行扩展时，可以如图 13 中的 1305 所示，从矩形的右边界上选择 5 个候选点，向右发射射线，以确定向右扩展时的扩展比例，在上下左右四个方向进行扩展后，可以得到图 13 中右下角 1306 所示的最终扩展结果。可以看到在这种情况下，原始 UV 岛在不同方向上扩展了不同的比例，并且均得到了最充分的扩展。

各向同性扩展：为了避免纹理扭曲，对于不具备内部对称特征的 UV 岛需要采用各向同性扩展。在本申请实施例中，各向同性扩展的实现过程可以通过以下步骤实现：

步骤 S241，提取当前 UV 岛外部的所有 UV 岛边界并构建对应的空间搜索树 T_1 。

步骤 S242，将当前 UV 岛移动至其局部间隙中心。

步骤 S243，提取当前 UV 岛的边界，并构建其对应的空间搜索树 T_2 。

步骤 S244，采用双向射线法确定最大扩展比例：

A) 从当前 UV 岛中心，沿着其边界顶点朝外发射射线，得到和 T_1 的最近交点，并据此计算对应的扩展比例；

B) 从其余 UV 岛边界顶点，朝当前 UV 岛的中心发射射线，得到和 T_2 的最近交点，并据此计算对应的扩展比例；

步骤 S245，计算步骤 S244 中得到的所有扩展比例中的最小者作为最终扩展比例，然后对当前 UV 岛及其内部岛进行各向同性的等比例扩展。

图 14 为本申请实施例提供的各向同性扩展的实现过程示意图，其中，图 14 中的 1401 是原始待扩展的 UV 岛；1402 是部分朝外发射的射线；1403 是部分朝内发射的射线；1404 是最终的扩展结果。由于 1402 和 1403 中所示的每条射线都对应一个扩展比例，所以只要选择所有射线所对应的扩展比例中的最小者，并据此对当前 UV 岛进行扩展，就可以确保不会和周围 UV 岛发生交叠。而此时的扩展比例也是该 UV 岛在不和其它 UV 岛发生交叠的前提下所能扩展的最大比例。

在实施自动 2UV 展开项目的美术资源中，往往存在一组或者多组对称 UV 岛。有时用户希望在装箱过程中保持这些对称 UV 岛的大小。基于此，在进行分类扩展时，可以通过以下步骤实现：

步骤 S9021'，记录每个 UV 岛的孔洞所包含的 UV 岛集合。

步骤 S9022'，提取对称 UV 岛，将对称 UV 岛集合记为 A，其余非对称 UV 岛集合记为 B。

步骤 S9023'，对于集合 A 中的每组对称 UV 岛，执行以下步骤：

a) 对于该组对称 UV 岛中的每个 UV 岛，将其移动到局部空隙中心，并根据其分类计算最大扩展比例；

b) 获得 a) 中所有扩展比例中的最小者，作为该组对称 UV 岛的最终扩展比例，并对该组所有对称 UV 岛进行同比例扩展；如果扩展导致发生了交叠，则恢复原样。

步骤 S9024'，对于集合 B 中的每个非对称 UV 岛，仍然按照步骤 S9023 提供的方法进行分类扩展。

在上述步骤 S9021' 至步骤 S9024' 中，优先考虑针对对称 UV 岛的分类扩展。这里的关键之处在于步骤 S9023'。在该步骤中，首先判断当前组的对称 UV 岛是否是具备内部对称特征的，如果是，则采用各向异性方式计算最大扩展比例；否则采用各向同性方式来计算最大扩展比例。在组内所有 UV 岛计算出最大扩展比例后，取其中的最小者作为最终扩展比例，以便最大限度地保证不发生交叠。然而同组对称 UV 岛也有可能相邻，使得按照最小比例扩展依然发生交叠。所以在实质扩展之后需要进行检查，一旦发现发生了交叠则恢复原样，并跳过对该组对称 UV 岛的扩展。如此，能够保证对称 UV 岛同

比例的扩展，因而能够满足美术人员的特定需求。

利用本申请实施例提供的模型处理方法，分别针对小规模 UV 岛、中等规模 UV 岛以及大规模 UV 岛进行了实验，得到了如图 15 所示的扩展结果。其中，在图 15 中，1501、1503 和 1505 分别为小规模 UV 岛、中等规模 UV 岛以及大规模 UV 岛扩展之前的装箱效果，1502、1504 和 1506 分别为小规模 UV 岛、中等规模 UV 岛以及大规模 UV 岛扩展之后的装箱效果，通过对比可以看出，经过多边形分类扩展之后，不仅装箱填充率得到了最大限度的提高，而且 UV 岛各向同性/各向异性的特性也得到了完好的保存。这两点非常符合 2UV 展开中美术工作人员的需求。

为了进一步验证本申请实施例提供的模型处理方法对自动 2UV 展开项目的填充率提升，在某游戏公司的 14 个项目的美术资源上进行了批量测试，得到了如表 1 所示的统计结果。将美术资源的名称分别用 A，B，...，N 来代替。

表 1、多边形分类扩展算法有益效果的统计对比

指标	美术资源						
	A	B	C	D	E	F	G
资源总数	1440	851	138	388	46	449	197
人工填充率	0.62	0.64	0.60	0.78	0.69	0.62	0.72
扩展前填充率	0.60	0.60	0.60	0.63	0.63	0.69	0.63
扩展后填充率	0.73	0.76	0.68	0.77	0.74	0.79	0.77
填充率提升比	0.13	0.16	0.08	0.14	0.11	0.10	0.14
扩展前超越人工占比	48.8%	42.3%	51.9%	19.5%	26.3%	66.2%	26.8%
扩展后超越人工占比	79.6%	81.0%	88.3%	50.5%	73.7%	86.3%	62.1%
超越人工占比提升比	20.8%	38.7%	36.4%	31.0%	47.4%	20.1%	35.3%
	H	I	J	K	L	M	N
资源总数	50	77	450	789	219	48	17
人工填充率	0.67	0.74	0.61	0.81	0.73	0.64	0.67
扩展前填充率	0.63	0.61	0.60	0.66	0.67	0.63	0.59
扩展后填充率	0.75	0.78	0.73	0.89	0.83	0.72	0.66
填充率提升比	0.12	0.17	0.13	0.23	0.16	0.09	0.07
扩展前超越人工占比	46.7%	17.9%	48.3%	19.4%	38.8%	52.6%	16.7%
扩展后超越人工占比	80.0%	67.6%	87.5%	69.6%	85.2%	76.3%	33.3%

超越人工占比提升比	33.3%	49.7%	39.2%	50.2%	46.4%	23.7%	16.6%
-----------	-------	-------	-------	-------	-------	-------	-------

通过表 1 可以看出，在加了多边形分类扩展之后，填充率有了 10%-15% 的提高。尤其值得注意的是，无多边形分类扩展之前的平均填充率往往不及人工装箱结果；但是加了多边形扩展之后，自动装箱结果的平均填充率除了项目 D 之外全部超越人工填充率。表 1 同样揭示出，增加了多边形分类扩展之后，装箱填充率超越人工的占比可以有平均 30% 左右的提升。

下面继续说明本申请实施例提供的对象处理装置 455 的实施为软件模块的示例性结构，在一些实施例中，如图 3 所示，存储在存储器 450 的对象处理装置 455 中的软件模块可以包括：

10 第一获取模块 4551，配置为获取待处理的多个资源对象在目标容器中的初始填充结果；

第一移动模块 4552，配置为基于所述初始填充结果，将各个资源对象移动至各自对应的可用空间的中心位置；

15 第二获取模块 4553，配置为获取所述各个资源对象的对称属性信息和包含关系信息；

第一确定模块 4554，配置为基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种；

20 第一扩展模块 4555，配置为基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象。

在一些实施例中，该第一移动模块，还配置为：

获取预设的移动迭代次数阈值和单个资源对象的移动次数阈值；

获取当前移动迭代次数，当所述当前移动迭代次数小于所述移动迭代次数阈值时，

25 获取第 i 个资源对象的当前移动次数， $i=1,2,\dots,N$ ，N 为资源对象总数，N 为大于 2 的整数；

当所述第 i 个资源对象的当前移动次数小于所述移动次数阈值时，确定将所述第 i 个资源对象的对象中心移动至所述第 i 个资源对象所在可用空间的中心位置的移动信息；

30 基于所述移动信息，确定所述第 i 个资源对象的移动距离大于预设的距离阈值时，将所述第 i 个资源对象的对象中心移动至所述可用空间的中心位置，并更新所述第 i 个资源对象的当前移动次数。

在一些实施例中，该装置还包括：

第一更新模块，配置为在完成第 1 个资源对象至第 N 个资源对象的当前移动迭代过程后，更新所述当前移动迭代次数，得到更新后的当前移动迭代次数；

35 第二移动模块，配置为确定所述更新后的当前移动迭代次数小于所述移动迭代次数阈值时，继续移动所述第 1 个资源对象至第 N 个资源对象。

在一些实施例中，该第一确定模块，还配置为：

当基于第 i 个资源对象的对称属性信息表征所述第 i 个资源对象具备内部对称特征，且基于所述第 i 个资源对象的包含关系信息确定所述第 i 个资源对象未被父资源对象包含时，确定所述第 i 个资源对象的扩展类型为各向异性扩展，其中， $i=1,2,\dots,N$ ，N 为资源对象总数，N 为大于 2 的整数。

40 在一些实施例中，该第一确定模块，还配置为：

当基于第 i 个资源对象的对称属性信息表征所述第 i 个资源对象不具备内部对称特

征时, 确定所述第 i 个资源对象的扩展类型为各向同性扩展。

在一些实施例中, 该第一确定模块, 还配置为:

当所述第 i 个资源对象被父资源对象包含时, 获取所述父资源对象的扩展类型;

当所述父资源对象的扩展类型为各向同性扩展时, 确定所述第 i 个资源对象的扩展
5 类型为各向同性扩展。

在一些实施例中, 该第一确定模块, 还配置为:

当所述父资源对象的扩展类型为各向异性扩展时, 获取所述父资源对象包含的其他
子资源对象;

当基于所述其他子资源对象的对称属性信息, 确定所述其他子资源对象均具备内部
10 对称特征时, 确定所述第 i 个资源对象的扩展类型为各向异性扩展;

当基于所述其他子资源对象的对称属性信息, 确定至少存在一个其他子资源对象不
具备内部对称特征时, 确定所述第 i 个资源对象的扩展类型为各向同性扩展。

在一些实施例中, 该第一扩展模块, 还配置为:

当确定所述目标容器中不存在第 i 个资源对象的对称对象, 且第 i 个资源对象的扩
15 展类型为各向同性扩展时, 确定所述第 i 个资源对象的第一扩展比例, $i=1, 2, \dots, N$, N
为资源对象总数, N 为大于 2 的整数;

当基于所述第 i 个资源对象的包含关系信息, 确定所述第 i 个资源对象中包含有其
他子资源对象时, 将所述第 i 个资源对象和所述第 i 个资源对象中包含的其他子资源对
象按照所述第一扩展比例, 对所述第 i 个资源对象和所述第 i 个资源对象中包含的其他
20 子资源对象进行扩展处理, 得到处理后的第 i 个资源对象和处理后的其他子资源对象。

在一些实施例中, 该第一扩展模块, 还配置为:

获取位于所述第 i 个资源对象外部的其他资源对象的外边界信息、所述第 i 个资源
对象的外边界信息和所述第 i 个资源对象的对象中心;

从所述第 i 个资源对象的对象中心向所述第 i 个资源对象的外边界上的各个第一顶
25 点发射第一射线, 确定各条第一射线与所述其他资源对象的外边界的各个第一最近交
点;

基于所述对象中心的中心坐标、所述各个第一顶点的第二顶点坐标和各个第一最近
交点的第一交点坐标, 确定各个第一顶点对应的第二候选扩展比例;

从所述其他资源对象的外边界上的各个第二顶点向所述第 i 个资源对象的对象中心
30 发射第二射线, 确定各条第二射线与所述第 i 个资源对象的外边界的各个第二最近交点;

基于所述对象中心的中心坐标、所述各个第二顶点的第二顶点坐标和各个第二最近
交点的第二交点坐标, 确定各个第二顶点对应的第二候选扩展比例;

将多个第一候选扩展比例和多个第二候选扩展比例中的最小值确定为所述第 i 个资
源对象的第一扩展比例。

35 在一些实施例中, 该第一扩展模块, 还配置为:

获取预设的间隔距离, 基于所述间隔距离和所述各个第一最近交点的交点坐标确定
各个第一最近交点对应的第一目标点的第一目标坐标;

基于所述对象中心的中心坐标和所述各个第一顶点的第二顶点坐标确定所述各个
第一顶点对应的第二距离;

40 基于所述对象中心的中心坐标和各个第一目标点的第一目标坐标确定所述各个第
一顶点对应的第二距离;

基于所述各个第一顶点对应的第二距离和第二距离, 确定各个第一顶点对应的第一
候选扩展比例。

在一些实施例中, 该第一扩展模块, 还配置为:

当第 i 个资源对象的扩展类型为各向异性扩展时, 获取所述第 i 个资源对象的主轴信息, 并基于所述第 i 个资源对象的主轴信息确定旋转角度和第一旋转方向, $i=1, 2, \dots, N$, N 为资源对象总数, N 为大于 2 的整数;

5 将 N 个资源对象按照所述第一旋转方向旋转所述旋转角度, 得到旋转后的 N 个资源对象, 其中, 旋转后的第 i 个资源对象的主轴为水平方向;

将所述旋转后的第 i 个资源对象的对象中心移动至对应的可用空间的中心位置;

确定所述旋转后的第 i 个资源对象在预设的各个扩展方向上的第二扩展比例;

10 当基于所述旋转后的第 i 个资源对象的包含关系信息, 确定所述旋转后的第 i 个资源对象中包含有其他子资源对象时, 将所述旋转后的第 i 个资源对象和所述旋转后的第 i 个资源对象中包含的其他子资源对象在所述各个扩展方向上按照各自对应的第二扩展比例进行扩展, 得到扩展后的第 i 个资源对象和扩展后的其他子资源对象;

将所述扩展后的第 i 个资源对象和所述扩展后的其他子资源对象按照第二旋转方向旋转所述旋转角度, 得到处理后的第 i 个资源对象和处理后的其他子资源对象, 所述第二旋转方向为所述第一旋转方向的反方向。

15 在一些实施例中, 该第一扩展模块, 还配置为:

获取位于所述旋转后的第 i 个资源对象外部的其他资源对象的外边界信息和所述旋转后的第 i 个资源对象的包围盒;

从所述旋转后的第 i 个资源对象的外边界上确定第 j 个扩展方向对应的多个候选点;

20 从各个候选点沿所述第 j 个扩展方向发射第三射线, 确定各条第三射线与所述其他资源对象的外边界的第三最近交点;

确定所述各个候选点到所述包围盒的目标边的各个第三距离, 并确定所述各个第三最近交点到所述目标边的各个第四距离, 所述目标边为所述包围盒上与所述第 j 个扩展方向的相反方向对应的边;

25 基于所述各个第三距离和对应的所述各个第四距离, 确定各个候选点对应的第三候选扩展比例;

将多个第三候选扩展比例中的最小值确定为所述第 j 个扩展方向的第二扩展比例。

在一些实施例中, 该装置还包括:

30 第二确定模块, 配置为当确定所述目标容器中存在第 i 个资源对象的对称对象时, 基于所述第 i 个资源对象的扩展类型, 分别确定所述第 i 个资源对象的第一参考扩展比例和所述对称对象的第二参考扩展比例;

第三确定模块, 配置为将所述第一参考扩展比例和所述第二参考扩展比例中的最小值, 确定为所述第 i 个资源对象和所述对称对象的目标扩展比例;

第二扩展模块, 配置为按照所述目标扩展比例对所述第 i 个资源对象和所述对称对象进行扩展处理, 得到处理后的第 i 个资源对象和处理后的对称对象。

35 在一些实施例中, 该装置还包括:

第四确定模块, 配置为在对第 1 个至第 N 个资源对象进行扩展处理后, 确定所述目标容器的第一填充率;

40 第三扩展模块, 配置为当确定所述第一填充率小于预设的填充率阈值时, 依次对第 N 个资源对象至第 1 个资源对象按照各自对应的扩展类型进行扩展处理, 得到第二次处理后的第 N 个资源对象至第 1 个资源对象;

第五确定模块, 配置为确定所述目标容器的第二填充率;

第六确定模块, 配置为当所述第二填充率大于所述第一填充率时, 将所述第二次处理后的第 N 个资源对象至第 1 个资源对象确定为目标扩展结果。

需要说明的是, 本申请实施例针对对象处理装置的描述, 与上述方法实施例的描述

是类似的,具有同方法实施例相似的有益效果。对于本装置实施例中未披露的技术细节,请参照本申请方法实施例的描述而理解。

5 本申请实施例提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行本申请实施例上述的对象处理方法。

10 本申请实施例提供一种存储有可执行指令的计算机可读存储介质,其中存储有可执行指令,当可执行指令被处理器执行时,将引起处理器执行本申请实施例提供的对象处理方法,例如,如图4A、图4B、图5和图6示出的对象处理方法。

在一些实施例中,计算机可读存储介质可以是FRAM、ROM、PROM、EPROM、EEPROM、闪存、磁表面存储器、光盘、或CD-ROM等存储器;也可以是包括上述存储器之一或任意组合的各种设备。

15 在一些实施例中,可执行指令可以采用程序、软件、软件模块、脚本或代码的形式,按任意形式的编程语言(包括编译或解释语言,或者声明性或过程性语言)来编写,并且其可按任意形式部署,包括被部署为独立的程序或者被部署为模块、组件、子例程或者适合在计算环境中使用的其它单元。

20 作为示例,可执行指令可以但不一定对应于文件系统中的文件,可以被存储在保存其它程序或数据的文件的一部分,例如,存储在超文本标记语言(HTML, Hyper Text Markup Language)文档中的一个或多个脚本中,存储在专用于所讨论的程序的单个文件中,或者,存储在多个协同文件(例如,存储一个或多个模块、子程序或代码部分的文件)中。

25 作为示例,可执行指令可被部署为在一个计算设备上执行,或者在位于一个地点的多个计算设备上执行,又或者,在分布在多个地点且通过通信网络互连的多个计算设备上执行。

以上所述,仅为本申请的实施例而已,并非用于限定本申请的保护范围。凡在本申请的精神和范围之内所作的任何修改、等同替换和改进等,均包含在本申请的保护范围之内。

权利要求书

- 1、一种对象处理方法，应用于贴图渲染场景，所述方法应用于计算机设备，包括：
获取待处理的多个资源对象在目标容器中的初始填充结果；
基于所述初始填充结果，将各个资源对象移动至各自对应的可用空间的中心位置；
5 获取所述各个资源对象的对称属性信息和包含关系信息；
基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种；
基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象。
- 10 2、根据权利要求1中所述的方法，其中，所述基于所述初始填充结果，将各个资源对象移动至各自对应的可用空间的中心位置，包括：
获取预设的移动迭代次数阈值和单个资源对象的移动次数阈值；
获取当前移动迭代次数，当所述当前移动迭代次数小于所述移动迭代次数阈值时，
15 获取第*i*个资源对象的当前移动次数， $i=1,2,\dots,N$ ，*N*为资源对象总数，*N*为大于2的整数；
当所述第*i*个资源对象的当前移动次数小于所述移动次数阈值时，确定将所述第*i*个资源对象的对象中心移动至所述第*i*个资源对象所在可用空间的中心位置的移动信息；
当基于所述移动信息，确定所述第*i*个资源对象的移动距离大于预设的距离阈值时，
20 将所述第*i*个资源对象的对象中心移动至所述可用空间的中心位置，并更新所述第*i*个资源对象的当前移动次数。
- 3、根据权利要求2中所述的方法，其中，所述方法还包括：
在完成第1个资源对象至第*N*个资源对象的当前移动迭代过程后，更新所述当前移动迭代次数，得到更新后的当前移动迭代次数；
25 确定所述更新后的当前移动迭代次数小于所述移动迭代次数阈值时，继续移动所述第1个资源对象至第*N*个资源对象。
- 4、根据权利要求1中所述的方法，其中，所述基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，包括：
当基于第*i*个资源对象的对称属性信息表征所述第*i*个资源对象具备内部对称特征，且基于所述第*i*个资源对象的包含关系信息确定所述第*i*个资源对象未被父资源对象包含时，
30 确定所述第*i*个资源对象的扩展类型为各向异性扩展，其中， $i=1,2,\dots,N$ ，*N*为资源对象总数，*N*为大于2的整数。
- 5、根据权利要求4中所述的方法，其中，所述基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，包括：
当基于第*i*个资源对象的对称属性信息表征所述第*i*个资源对象不具备内部对称特征时，
35 确定所述第*i*个资源对象的扩展类型为各向同性扩展。
- 6、根据权利要求4中所述的方法，其中，所述基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，包括：
当所述第*i*个资源对象被父资源对象包含时，获取所述父资源对象的扩展类型；
当所述父资源对象的扩展类型为各向同性扩展时，确定所述第*i*个资源对象的扩展类型
40 为各向同性扩展。
- 7、根据权利要求6中所述的方法，其中，所述基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，包括：
当所述父资源对象的扩展类型为各向异性扩展时，获取所述父资源对象包含的其他

子资源对象：

当基于所述其他子资源对象的对称属性信息，确定所述其他子资源对象均具备内部对称特征时，确定所述第 i 个资源对象的扩展类型为各向异性扩展；

5 当基于所述其他子资源对象的对称属性信息，确定至少存在一个其他子资源对象不具备内部对称特征时，确定所述第 i 个资源对象的扩展类型为各向同性扩展。

8、根据权利要求 1 中所述的方法，其中，所述基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象，包括：

10 当确定所述目标容器中不存在第 i 个资源对象的对称对象，且第 i 个资源对象的扩展类型为各向同性扩展时，确定所述第 i 个资源对象的第一扩展比例， $i=1, 2, \dots, N$ ， N 为资源对象总数， N 为大于 2 的整数；

当基于所述第 i 个资源对象的包含关系信息，确定所述第 i 个资源对象中包含有其他子资源对象时，将所述第 i 个资源对象和所述第 i 个资源对象中包含的其他子资源对象按照所述第一扩展比例，对所述第 i 个资源对象和所述第 i 个资源对象中包含的其他子资源对象进行扩展处理，得到处理后的第 i 个资源对象和处理后的其他子资源对象。

15 9、根据权利要求 8 中所述的方法，其中，所述确定所述第 i 个资源对象的第一扩展比例，包括：

获取位于所述第 i 个资源对象外部的其他资源对象的外边界信息、所述第 i 个资源对象的外边界信息和所述第 i 个资源对象的对象中心；

20 从所述第 i 个资源对象的对象中心向所述第 i 个资源对象的外边界上的各个第一顶点发射第一射线，确定各条第一射线与所述其他资源对象的外边界的各个第一最近交点；

基于所述对象中心的中心坐标、所述各个第一顶点的第一顶点坐标和各个第一最近交点的第一交点坐标，确定各个第一顶点对应的第一候选扩展比例；

从所述其他资源对象的外边界上的各个第二顶点向所述第 i 个资源对象的对象中心发射第二射线，确定各条第二射线与所述第 i 个资源对象的外边界的各个第二最近交点；

25 基于所述对象中心的中心坐标、所述各个第二顶点的第二顶点坐标和各个第二最近交点的第二交点坐标，确定各个第二顶点对应的第二候选扩展比例；

将多个第一候选扩展比例和多个第二候选扩展比例中的最小值确定为所述第 i 个资源对象的第一扩展比例。

30 10、根据权利要求 9 所述的方法，其中，所述基于所述对象中心的中心坐标、所述各个第一顶点的第一顶点坐标和各个第一最近交点的第一交点坐标，确定各个第一顶点对应的第一候选扩展比例，包括：

获取预设的间隔距离，基于所述间隔距离和所述各个第一最近交点的交点坐标确定各个第一最近交点对应的第一目标点的第一目标坐标；

35 基于所述对象中心的中心坐标和所述各个第一顶点的第一顶点坐标确定所述各个第一顶点对应的第一距离；

基于所述对象中心的中心坐标和各个第一目标点的第一目标坐标确定所述各个第一顶点对应的第二距离；

基于所述各个第一顶点对应的第一距离和第二距离，确定各个第一顶点对应的第一候选扩展比例。

40 11、根据权利要求 1 中所述的方法，其中，所述基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象，包括：

当第 i 个资源对象的扩展类型为各向异性扩展时，获取所述第 i 个资源对象的主轴信息，并基于所述第 i 个资源对象的主轴信息确定旋转角度和第一旋转方向， $i=1, 2, \dots, N$ ， N 为大于 2 的整数，表示资源对象总数；

将N个资源对象按照所述第一旋转方向旋转所述旋转角度,得到旋转后的N个资源对象,其中,旋转后的第i个资源对象的主轴为水平方向;

将所述旋转后的第i个资源对象的对象中心移动至对应的可用空间的中心位置;

确定所述旋转后的第i个资源对象在预设的各个扩展方向上的第二扩展比例;

5 当基于所述第i个资源对象的包含关系信息,确定所述旋转后的第i个资源对象中包含有其他子资源对象时,将所述旋转后的第i个资源对象和所述旋转后的第i个资源对象中包含的其他子资源对象在所述各个扩展方向上按照各自对应的第二扩展比例进行扩展,得到扩展后的第i个资源对象和扩展后的其他子资源对象;

10 将所述扩展后的第i个资源对象和所述扩展后的其他子资源对象按照第二旋转方向旋转所述旋转角度,得到处理后的第i个资源对象和处理后的其他子资源对象,所述第二旋转方向为所述第一旋转方向的反方向。

12、根据权利要求11中所述的方法,其中,所述确定所述旋转后的第i个资源对象在预设的各个扩展方向上的第二扩展比例,包括:

15 获取位于所述旋转后的第i个资源对象外部的其他资源对象的外边界信息和所述旋转后的第i个资源对象的包围盒;

从所述旋转后的第i个资源对象的外边界上确定第j个扩展方向对应的多个候选点,j为小于或者等于M的正整数,M为扩展方向总数,M为大于或者等于1的整数;

从各个候选点沿所述第j个扩展方向发射第三射线,确定各条第三射线与所述其他资源对象的外边界的第三最近交点;

20 确定所述各个候选点到所述包围盒的目标边的各个第三距离,并确定所述各个第三最近交点到所述目标边的各个第四距离,所述目标边为所述包围盒上与所述第j个扩展方向的相反方向对应的边;

基于所述各个第三距离和对应的所述各个第四距离,确定各个候选点对应的第三候选扩展比例;

25 将多个第三候选扩展比例中的最小值确定为所述第j个扩展方向的第二扩展比例。

13、根据权利要求1中所述的方法,其中,所述方法还包括:

当确定所述目标容器中存在第i个资源对象的对称对象时,基于所述第i个资源对象的扩展类型,分别确定所述第i个资源对象的第一参考扩展比例和所述对称对象的第二参考扩展比例;

30 将所述第一参考扩展比例和所述第二参考扩展比例中的最小值,确定为所述第i个资源对象和所述对称对象的目标扩展比例;

按照所述目标扩展比例对所述第i个资源对象和所述对称对象进行扩展处理,得到处理后的第i个资源对象和处理后的对称对象。

14、根据权利要求1至13任一项所述的方法,其中,所述方法还包括:

35 在对第1个至第N个资源对象进行扩展处理后,确定所述目标容器的第一填充率;

当确定所述第一填充率小于预设的填充率阈值时,依次对第N个资源对象至第1个资源对象按照各自对应的扩展类型进行扩展处理,得到第二次处理后的第N个资源对象至第1个资源对象;

确定所述目标容器的第二填充率;

40 当所述第二填充率大于所述第一填充率时,将所述第二次处理后的第N个资源对象至第1个资源对象确定为目标扩展结果。

15、一种对象处理装置,所述装置包括:

第一获取模块,配置为获取待处理的多个资源对象在目标容器中的初始填充结果;

第一移动模块,配置为基于所述初始填充结果,将各个资源对象移动至各自对应的

可用空间的中心位置；

第二获取模块，配置为获取所述各个资源对象的对称属性信息和包含关系信息；

第一确定模块，配置为基于所述各个资源对象的对称属性信息和包含关系信息确定所述各个资源对象的扩展类型，其中，所述扩展类型包括各向同性扩展和各向异性扩展中的一种；

5

第一扩展模块，配置为基于所述各个资源对象的扩展类型，对所述各个资源对象进行扩展处理，得到处理后的各个资源对象。

16、一种计算机设备，所述计算机设备包括：

存储器，用于存储可执行指令；

10

处理器，用于执行所述存储器中存储的可执行指令时，实现权利要求 1 至 14 任一项所述的对象处理方法。

17、一种计算机可读存储介质，存储有可执行指令，所述可执行指令被处理器执行时实现权利要求 1 至 14 任一项所述的对象处理方法。

18、一种计算机程序产品，包括计算机程序或指令，所述计算机程序或指令被处理器执行时实现权利要求 1 至 14 任一项所述的对象处理方法。

15