



(12) 发明专利

(10) 授权公告号 CN 111381919 B

(45) 授权公告日 2024.06.25

(21) 申请号 201911396260.4

(22) 申请日 2019.12.30

(65) 同一申请的已公布的文献号
申请公布号 CN 111381919 A

(43) 申请公布日 2020.07.07

(30) 优先权数据
18306885.7 2018.12.29 EP

(73) 专利权人 达索系统公司
地址 法国韦利济—维拉库布莱

(72) 发明人 E·梅尔 F·M·桑切斯贝穆德斯

(74) 专利代理机构 永新专利商标代理有限公司
72002

专利代理人 戴开良

(51) Int. Cl.
G06F 9/451 (2018.01)

(56) 对比文件

Gopal Sharma et al..CSGNet:Neural shape parser for constructive solid geometry.http://arxiv.org/pdf/1712.08290.pdf.2018,第1卷(第1期),1-16.

Mrchenzhangwu.InverseCSG: Automatic Conversion of 3D Models to CSG Trees.https://blog.csdn.net/Mrchenzhangwu/article/details/103097253.2019,第1卷(第1期),1.

Gopal Sharma et al..CSGNet:Neural shape parser for constructive solid geometry.http://arxiv.org/pdf/1712.08290.pdf.2018,第1卷(第1期),1-16.

审查员 李敏

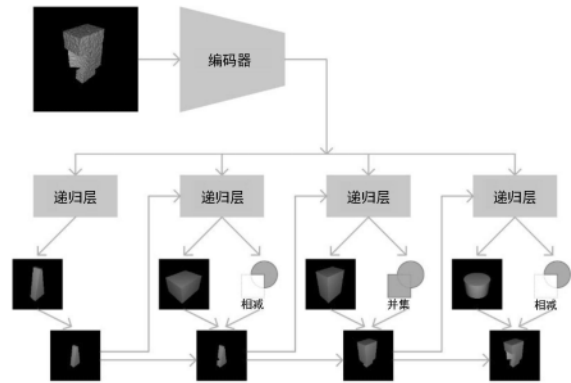
权利要求书2页 说明书28页 附图5页

(54) 发明名称

形成用于推断可编辑特征树的数据集

(57) 摘要

本发明尤其涉及一种计算机实现的方法,该方法用于形成被配置用于学习神经网络的数据集。神经网络被配置用于从3D形状的离散几何表示推断出表示3D形状的可编辑特征树。可编辑特征树包括应用于叶几何形状的几何运算的树布置。该方法包括提供相应的数据段,以及将数据段的一部分分别作为相应的训练样本插入到数据集中。插入到数据集中的一个或多个第一数据段中的每个第一数据段的相应的3D形状与未插入到数据集中的相应一个或多个第二数据段的相应的3D形状相同。该方法形成针对数字化的改进的解决方案。



1. 一种用于神经网络的计算机实现的方法,所述方法用于形成被配置用于学习所述神经网络的数据集,所述神经网络被配置用于从3D形状的离散几何表示推断出表示所述3D形状的可编辑特征树,所述可编辑特征树包括应用于叶几何形状的几何运算的树布置,所述方法包括:

提供相应的数据段,每个数据段包括:

表示3D形状的可编辑特征树,所述可编辑特征树包括应用于叶几何形状的几何运算的树布置,以及

所述3D形状的离散几何表示,所述离散几何表示与将几何运算的所述树布置应用于所述叶几何形状的结果相对应;

将所述数据段的一部分分别作为相应的训练样本插入到所述数据集中,插入到所述数据集中的—个或多个第一数据段中的每个第一数据段的所述相应的3D形状与未插入到所述数据集中的相应的—个或多个第二数据段的所述相应的3D形状相同,

其中,针对每个第一数据段,所述第一数据段的所述可编辑特征树与每个相应的第二数据段的所述可编辑特征树在结构上是兼容的,并且在以下情况下,第一可编辑特征树和第二可编辑特征树在结构上是兼容的:

所述第一可编辑特征树的几何运算的所述树布置与所述第二可编辑特征树的几何运算的所述树布置至少部分地匹配,以及

每对对应的叶几何形状具有相同的相应的基元形状类型,并且对于每个其它参数,每对对应的叶几何形状具有相同的相应的参数值。

2. 根据权利要求1所述的方法,其中,关于基元形状类型的离散集合以及对于每种基元形状类型,相应的连续参数中的每个连续参数的—个或多个参数域的相应的离散集合,每个参数域具有所述相应的连续参数的相应的参数值,每个叶几何形状是由相应的基元形状类型利用—个或多个参数域的所述相应的离散集合中的每个参数域的相应的参数值形成的。

3. 根据权利要求2所述的方法,其中,对于—种或多种基元形状类型中的每一种基元形状类型,所述—个或多个相应的连续参数包括—个或多个尺寸参数和/或—个或多个定位参数,和/或所述基元形状类型的离散集合包括长方体类型、球体类型、—种或多种圆柱体类型、和/或—种或多种棱柱体类型。

4. 根据权利要求2或3所述的方法,其中,所述提供所述相应的数据段包括:

提供初始数据,所述初始数据包括所述基元形状类型的离散集合和—个或多个参数域的每个相应的离散集合;以及

基于所述初始数据来合成所述相应的数据段。

5. 根据权利要求4所述的方法,其中,对于—种或多种基元形状类型中的每种基元形状类型,所述—个或多个相应的连续参数包括—个或多个尺寸参数,并且所述合成包括:

基于所述初始数据来生成至少—个初始数据段,包括针对每个尺寸参数来选择相应的参数值;以及然后

基于所述初始数据段来确定至少—个将来的数据段,包括迭代地从所述初始数据段开始,修改相应的尺寸参数中的每个尺寸参数的—个或多个参数值和/或删除—个或多个叶几何形状。

6. 根据权利要求1至3中任一项所述的方法,其中,所述插入包括:
提供成对的相应的数据段,每对包括第三数据段和第四数据段;以及
针对每对进行以下操作:
测试所述第三数据段的所述3D形状与所述第四数据段的所述3D形状之间的同一性;以
及
在所述测试产生肯定结果时,将所述第三数据段和所述第四数据段中的一个数据段识别为第二数据段。
7. 根据权利要求6所述的方法,其中,所述测试包括:
计算第一体积值,所述第一体积值表示所述第三数据段的所述离散几何表示的体积;
计算第二体积值,所述第二体积值表示所述第四数据段的所述离散几何表示的体积;
以及
将所述第一体积值与所述第二体积值进行比较。
8. 根据权利要求1至3中任一项所述的方法,其中:
关于基元形状类型的离散集合并且对于每个基元形状类型,相应的连续参数中的每个连续参数的一个或多个参数域的相应的离散集合,每个参数域具有所述相应的连续参数的相应的参数值,每个叶几何形状是由相应的基元形状类型利用一个或多个参数域的所述相应的离散集合中的每个参数域的相应的参数值形成的,
对于一种或多种基元形状类型中的每种基元形状类型,所述一个或多个相应的连续参数由一个或多个尺寸参数和一个或多个其它参数组成,所述一个或多个其它参数包括一个或多个定位参数。
9. 根据权利要求1至3中任一项所述的方法,其中,对于一个或多个第一数据段中的每个第一数据段,所述插入包括:根据所述第一数据段的所述可编辑特征树的规范性来选择所述第一数据段。
10. 一种用于神经网络的计算机实现的方法,所述方法包括:执行根据权利要求1-9中的任一项所述的方法以形成数据集;以及
基于所述数据集来学习所述神经网络。
11. 一种其上存储有指令的非暂时性计算机可读存储介质,所述指令在由计算机运行时,使得所述计算机能够执行根据权利要求1至9中任一项所述的方法和/或根据权利要求10所述的方法。
12. 一种用于神经网络的装置,所述装置包括:
处理器;
耦合到所述处理器的存储器,所述存储器上存储有指令,所述指令在由所述处理器执行时,使得所述处理器执行根据权利要求1-9中任一项所述的方法和/或根据权利要求10所述的方法。
13. 一种计算机程序产品,其包括计算机程序,所述计算机程序在由计算机执行时,使得所述计算机执行根据权利要求1至9中任一项所述的方法和/或根据权利要求10所述的方法。

形成用于推断可编辑特征树的数据集

技术领域

[0001] 本公开内容涉及计算机程序和系统的领域,并且更具体地,本公开内容涉及与学习神经网络有关的解决方案,该神经网络被配置用于从3D形状的离散几何表示推断出表示3D形状的可编辑特征树。

背景技术

[0002] 市场上提供了许多用于对象的设计、工程和制造的系统 and 程序。CAD是计算机辅助设计的缩写,例如,其涉及用于设计对象的软件解决方案。CAE是计算机辅助工程的缩写,例如,其涉及用于模拟未来产品的物理行为的软件解决方案。CAM是计算机辅助制造的缩写,例如,其涉及用于定义制造过程和操作的软件解决方案。在这种计算机辅助设计系统中,图形用户界面在技术效率方面起着重要作用。这些技术可以嵌入产品生命周期管理(PLM)系统中。PLM是一种业务战略,其帮助企业在扩展企业的整个概念中共享产品数据,应用通用流程以及利用企业知识来开发从概念到其寿命结束的产品。Dassault Systèmes (商标为CATIA、ENOVIA和DELMIA)提供的PLM解决方案提供了工程中心(其组织产品工程知识)、制造中心(其管理制造工程知识)以及企业中心(其使企业能够整合和连接成工程中心和制造中心两者)。该系统共同提供了一个开放的对象模型,该模型将产品、过程、资源链接在一起,以实现动态的、基于知识的产品创建和决策支持,其推动优化的产品定义、制造准备、生产和服务。

[0003] 在这种背景下,研究人员一直在寻找从3D形状的离散表示中推断出表示3D形状的合理的可编辑特征树的解决方案。特别地,寻找与原始输入网格相对应的合理的CSG树已经引起了注意。这个问题属于更一般的主题,被称为“数字化”。

[0004] 以下参考文献特别涉及数字化:

[0005] - [1] Bénéière等人:Topology Reconstruction for B-Rep Modeling from 3D Mesh in Reverse Engineering Applications, 2012年。

[0006] - [2] Flavien Boussuge等人:Extraction of generative processes from B-Rep shapes and application to idealization transformations, 2014年。

[0007] - [3] S. Gauthier等人:Digitized 3d mesh segmentation based on curvature analysis, 2017年。

[0008] - [4] D. Cohen-Steiner等人:Restricted Delaunay Triangulation and Normal Cycle, 2003年。

[0009] - [5] B. Levy等人:Lp Centroidal Voronoi Tessellation and its application, 2010年。

[0010] - [6] F. Boussuge: Idealization of CAD assemblies for FE structural analyses, chapter 4: Extraction of generative processes from B-Rep shapes to structure components up to assemblies, PhD Thesis, Université de Grenoble, 2014年。

[0011] - [7] J. J. Shah, D. Anderson, Y. Se Kim, S. Joshi: A Discourse on Geometric Feature Recognition From CAD Models, Journal of Computing and Information Science in Engineering, 2001年。

[0012] - [8] Y. Zhang, X. Luo, Y. Zhao, An Approach to the Automatic Recognition of Boolean Decomposition Loops for Swept Volume Decomposition, 25th International Meshing Roundtable, 2016年。

[0013] - [9] Jonathan Richard Shewchuk, Robust Adaptive Floating-Point Geometric Predicates, 1996年。

[0014] - [10] G. Sharma, R. Goyal, D. Liu, E. Kalogerakis, S. Maji. CSGNet: Neural Shape Parser for Constructive Solid Geometry. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018年。

[0015] 论文[1]涉及从网格中提取足够的信息以使用平均曲率构建精确的边界表示(B-Rep)模型。通过分析相对于整个网格曲率分布而言网格的三角形上的曲率,建立子网格,这些子网格是作为B-Rep表示的面的潜在候选。当子网格不接触它们的邻居时,所提出的方法将它们扩展,直到可以恢复相邻面之间的某些限制。所提出的方法通过基于到已建立的子网格的距离的概念来聚合三角形,从而扩展子网格。对于复杂且曲折的几何形状,不能保证所提出的过程能够恢复有效的限制。因此,在复杂的网格几何图形的情况下,该方法可能无法构建B-Rep。此外,这种方法仅专注于重建精确的B-Rep,而不是可编辑特征树。在从原始网格构建可编辑特征树的背景下,B-Rep表示不一定是中间阶段,并且其可能会损害该方法。在复杂几何图形的情况下,通过中间的B-Rep表示将使整个过程变得脆弱。如果B-Rep构造失败,则根本不会检索任何特征。

[0016] 论文[2]涉及从B-Rep表示开始的“过程树”重构。为了构建流程树,所提出的方法分析了B-Rep的面和面之间的相交处以构建草图,稍后可以将该草图拉伸。生成的过程树仅限于实体加法,即不生成任何减法运算。由于精确的B-Rep是可用的,因此算法迭代地进行,即,通过布尔运算从当前B-Rep中识别并且移除特征实体,直到处理完整个实体为止。因此,所提出的方法存在传播错误的风险。

[0017] 论文[8]提出了一种确定扫描实体之间连续的布尔运算所产生的连接的相交曲线/边的非平面环路。为了闭合环路,该论文指出可能必须创建额外的虚拟相交曲线/边。然而,该论文没有详细介绍如何创建它们。此外,该论文没有超出环路检测的范围,并且特别地,其没有显示如何根据环路检测来确定布尔运算顺序。

[0018] 论文[10]提出了一种完全不同的方法,该方法基于神经网络模型从3D形状推断特征树表示。该学习是基于特征树表示的候选叶实体的离散集合的。另外,该方法要求将输入形状离散为具有有限分辨率的常规体素网格。该论文没有提到如何获得标记的数据集进行监督学习。所提出的方法缺乏准确性。

[0019] 在这种背景拿下,仍然需要用于数字化的改进的解决方案。

发明内容

[0020] 因此,提供了一种计算机实现的(数据集形成)方法,该方法用于形成被配置用于学习神经网络的数据集。神经网络被配置用于从3D形状的离散几何表示推断出表示所述3D

形状的可编辑特征树。所述可编辑特征树包括应用于叶几何形状的几何运算的树布置。该方法包括提供相应的数据段,以及将所述数据段的一部分分别作为相应的训练样本插入到所述数据集中。每个数据段包括表示3D形状的可编辑特征树。可编辑特征树包括应用于叶几何形状的几何运算的树布置。每个数据段还包括3D形状的离散几何表示。所述离散几何表示与将几何运算的所述树布置应用于所述叶几何形状的结果相对应。插入到所述数据集中的一个或多个第一数据段中的每个第一数据段的所述相应的3D形状与未插入到所述数据集中的相应的一个或多个第二数据段的所述相应的3D形状相同。

[0021] 该方法可以包括以下各项中的一项或多项:

[0022] -关于基元形状类型的离散集合以及对于每种基元形状类型,相应的连续参数中的每个连续参数的一个或多个参数域的相应的离散集合,每个参数域具有所述相应的连续参数的相应的参数值,每个叶几何形状是由相应的基元形状类型利用一个或多个参数域的所述相应的离散集合中的每个参数域的相应的参数值形成的;

[0023] -对于一种或多种基元形状类型中的每一种基元形状类型,所述一个或多个相应的连续参数包括一个或多个尺寸参数和/或一个或多个定位参数;

[0024] -所述基元形状类型的离散集合包括长方体类型、球体类型、一种或多种圆柱体类型和/或一种或多种棱柱体类型;

[0025] -提供相应的数据段包括:提供初始数据,所述初始数据包括所述基元形状类型的离散集合和一个或多个参数域的每个相应的离散集合;以及基于所述初始数据来合成所述相应的数据段;

[0026] -对于一种或多种基元形状类型中的每种基元形状类型,所述一个或多个相应的连续参数包括一个或多个尺寸参数,并且所述合成包括:基于所述初始数据来生成至少一个初始数据段,包括针对每个尺寸参数来选择相应的参数值;以及然后基于所述初始数据段来确定至少一个将来的数据段,包括迭代地从所述初始数据段开始,修改相应的尺寸参数中的每个尺寸参数的一个或多个参数值和/或删除一个或多个叶几何形状;

[0027] -插入包括:提供成对的相应的数据段,每对包括第三数据段和第四数据段,并且针对每对进行以下操作:测试所述第三数据段的所述3D形状与所述第四数据段的所述3D形状之间的同一性;以及在所述测试产生肯定结果时,将所述第三数据段和所述第四数据段中的一个数据段识别为第二数据段。

[0028] -测试包括:计算第一体积值,所述第一体积值表示所述第三数据段的所述离散几何表示的体积;计算第二体积值,所述第二体积值表示所述第四数据段的所述离散几何表示的体积;以及将所述第一体积值与所述第二体积值进行比较;

[0029] -针对每个第一数据段,所述第一数据段的所述可编辑特征树与每个相应的第二数据段的所述可编辑特征树在结构上是兼容的;

[0030] -关于基元形状类型的离散集合并且对于每个基元形状类型,相应的连续参数中的每个连续参数的一个或多个参数域的相应的离散集合,每个参数域具有所述相应的连续参数的相应的参数值,每个叶几何形状是由相应的基元形状类型利用一个或多个参数域的所述相应的离散集合中的每个参数域的相应的参数值形成的;对于一种或多种基元形状类型中的每种基元形状类型,所述一个或多个相应的连续参数由一个或多个尺寸参数和一个或多个其它参数组成,所述一个或多个其它参数包括一个或多个定位参数;并且在以下情

况下,第一可编辑特征树和第二可编辑特征树在结构上是兼容的:所述第一可编辑特征树的几何运算的所述树布置与所述第二可编辑特征树的几何运算的所述树布置至少部分地匹配,并且每对对应的叶几何形状具有相同的相应的基元形状类型,并且对于每个其它参数,每对对应的叶几何形状具有相同的相应的参数值;和/或

[0031] -对于一个或多个第一数据段中的每个第一数据段,所述插入包括:根据所述第一数据段的所述可编辑特征树的规范性来选择所述第一数据段。

[0032] 还提供了表示根据数据集形成方法可形成的数据集的数据结构。

[0033] 还提供了一种使用数据集的计算机实现的(使用)方法。该方法包括基于数据集来学习神经网络。

[0034] 还提供了一种计算机程序,该计算机程序包括用于执行数据集形成方法和/或使用方法的指令。

[0035] 还提供了一种设备,该设备包括具有记录在其上的数据结构和/或程序的数据存储介质。该设备可以形成或用作非暂时性计算机可读介质,例如在SaaS(软件作为服务)或其它服务器、或基于云的平台等上。该设备可以替代地包括耦合到数据存储介质的处理器。因此,该设备可以全部或部分地形成计算机系统(例如,该设备是整个系统的子系统)。该系统还可以包括耦合到处理器的图形用户界面。

附图说明

[0036] 现在将通过非限制性示例的方式并且参考附图来描述实施例,其中:

[0037] -图1示出了系统的图形用户界面的示例;

[0038] -图2示出了系统的示例;以及

[0039] -图3-11说明了这些方法。

具体实施方式

[0040] 据此提出了与神经网络的学习有关的计算机实现的方法。在本文中,神经网络是一种功能,其被配置为:将相应的3D形状中的每个3D形状的离散几何表示中的任何一个作为输入;处理该输入;以及作为处理的结果,输出被配置用于推断表示与输入相对应的3D形状的可编辑特征树的相应数据。因此,这样的神经网络可以用于推断与3D形状的离散几何表示相对应的可编辑特征树。

[0041] 因此,提出了一种用于执行这种推断的计算机实现的方法,该方法可以被称为“推断方法”。

[0042] 该推断方法可以包括提供3D形状的离散几何表示。该推断方法还可以包括将神经网络应用于3D形状的离散几何表示。该推断方法还可以包括基于应用的结果(即,当输入离散几何表示时由神经网络输出的数据)来推断表示3D形状的至少一个可编辑特征树。输出数据可以(直接)包括可编辑特征树,和/或被处理以(间接)确定可编辑特征树,例如自动地和/或根据预定算法。

[0043] 在间接确定的情况下,输出数据可以例如包括用于选择定义至少一个可编辑特征树的值的一个或多个概率分布,并且预定算法可以例如包括基于一个或多个概率分布来确定至少一个可编辑特征树。该确定可以例如惩罚选择具有低概率的值。该确定可以例如等

于基于一个或多个概率分布来搜索预定数量 k (例如,其中 k 高于或等于1和/或低于10)个最可能的可编辑特征树。可以根据波束搜索算法来执行搜索。这提供了计算效率。

[0044] 在提高准确性的示例中,该推断方法还可以包括例如自动地细化每个(初始地)推断出的可编辑特征树。细化可以包括编辑推断出的可编辑特征树(例如,其参数),以便使得其所得的3D几何形状(例如,可通过“播放”可编辑特征树获得)更类似于输入的离散几何表示。可以通过使损失最小化来增加相似度,该损失惩罚由推断出的可编辑特征树提供的3D几何形状与输入的离散几何表示之间的差异。最小化可以包括探索最初推断出的可编辑特征树的候选版本,例如其叶几何形状的版本。稍后提供这种版本的示例。该细化选项通过利用特定于本公开内容的背景的事实来提高准确性。即,该细化利用了以下事实:可编辑特征树可以用于计算与离散几何表示直接可比的3D几何形状,例如通过播放可编辑特征树(即,按照由可编辑特征树的树排列定义的顺序将可编辑特征树的几何运算应用于可编辑特征树的叶几何形状)。换句话说,可以直接并且立即地评估神经网络的输出的准确性。

[0045] 从机器学习的领域中知道,神经网络对输入的处理包括对输入应用运算,该运算由包括权重值的数据定义。因此,神经网络的学习包括基于被配置用于这种学习的数据集来确定权重的值。为此,数据集包括各自形成相应的训练样本的数据段。训练样本表示在学习之后要使用神经网络的情况的多样性。本文引用的任何数据集都可以包括高于1000、10000、100000或1000000的多个训练样本。

[0046] 因此,特别提出了一种用于形成这种数据集的计算机实现的方法,该方法可以被称为“数据集形成方法”。还提出了一种用于执行这种学习的计算机实现的方法,该方法可以被称为“学习方法”。

[0047] 机器学习过程可以包括数据集形成方法和/或学习方法,以及可选地包括推断方法。

[0048] 机器学习过程可以特别地包括数据集形成方法和学习方法两者。在数据集形成方法之后的示例中,学习方法可以例如至少部分地基于由数据集形成方法形成的数据集来执行。这种机器学习过程特别有效并且提供提高的准确性。

[0049] 替代地,机器学习过程可以包括数据集形成方法,并且基于数据集来执行用于学习神经网络的任何其它计算机实现的方法(与所提出的学习方法相比)。又替代地,机器学习过程可以包括对由任何其它计算机实现的方法(与所提出的数据集形成方法相比)(诸如用于形成数据集或检索数据集的另一种方法)提供的数据集执行学习方法。

[0050] 在介绍数据集形成方法和学习方法之前,现在讨论其中涉及的数据结构。将会认识到的是,本文提供的数据结构定义和示例可以应用于通过数据集形成方法形成的数据集的至少一部分(例如全部)和/或在学习方法中提供的数据集的至少一部分(例如全部)、和/或推断方法的至少一个输入和/或至少一个输出。

[0051] 3D形状的离散几何表示在本文中是一种数据结构,其包括数据段的离散集合。每段数据表示位于3D空间中的相应的几何实体。每个几何实体表示3D形状的相应位置(换句话说,由3D形状表示的实体的材料组成的相应部分)。几何实体的聚合(即并集或并列)在一起表示3D形状。在示例中,本文中的任何离散几何表示可以包括高于100(例如,高于1000)的多个这样的数据段。

[0052] 本文中的任何离散几何表示可以例如是3D点云,每个几何实体是点。替代地,本文

中的任何离散几何表示可以是3D网格,每个几何实体是网格图块或面。本文中的任何3D网格可以是规则的或不规则的(即,是否由相同类型的面组成)。本文中的任何3D网格可以是多边形网格,例如三角形网格。替代地,本文中的任何3D网格可以是B-Rep。本文中的任何3D网格可以是3D点云获得的,例如通过对3D点云进行三角剖分(例如,利用DeLaunay三角剖分)。

[0053] 本文中的任何3D点云可以是根据真实对象的物理测量结果来确定的,例如在3D重建过程中(除了离散几何表示是综合生成的之外,例如在数据集形成方法中)。3D重建过程可以包括:提供真实对象;提供每个被配置用于获取相应的物理信号的一个或多个物理传感器;以及通过在真实对象上操作一个或多个物理传感器(即,利用每个传感器来扫描真实对象)来获取一个或多个相应的物理信号。然后,根据任何已知技术,3D重建可以基于测量结果自动确定3D点云和/或3D网格。一个或多个传感器可以包括多个(例如,RGB和/或图像或视频)相机,并且确定可以包括运动结构分析。一个或多个传感器可以替代地或另外包括一个或多个深度传感器(例如在RGB深度相机上),并且确定可以包括根据深度数据的3D重建。一个或多个深度传感器可以例如包括激光器(例如激光雷达)或超声发射器-接收器。

[0054] 替代地,本文中的任何3D点云或3D网格可以是表示实体(例如B-Rep模型)的皮肤(即外表面)的3D建模对象中获得的,例如,通过在3D建模对象上进行射线投射或镶嵌3D建模对象。可以根据任何3D建模对象渲染过程来执行镶嵌。可以在任何CAD系统上对这种渲染过程进行编码,以便显示3D建模对象的图形表示。3D建模对象可以由用户利用CAD系统进行设计或已经是设计好的(除了离散几何表示是综合生成的之外,例如在数据集形成方法期间)。

[0055] 建模对象是由例如在数据库中存储的数据定义的任何对象。通过扩展,表达“建模对象”表示数据本身。根据用于设计建模对象的系统的类型,可以通过不同种类的数据来定义建模对象。该系统确实可以是CAD系统、CAE系统、CAM系统、PDM系统和/或PLM系统的任何组合。在那些不同的系统中,建模对象由相应的数据定义。因此,可以提到CAD对象、PLM对象、PDM对象、CAE对象、CAM对象、CAD数据、PLM数据、PDM数据、CAM数据、CAE数据。然而,这些系统并不是彼此排斥的,因为可以通过与这些系统的任何组合相对应的数据来定义建模对象。因此,系统很可能既是CAD系统又是PLM系统。

[0056] 所谓CAD系统,另外意指至少适于基于建模对象的图形表示来设计建模对象的任何系统,诸如CATIA。在这种情况下,定义建模对象的数据包括允许表示建模对象的数据。CAD系统可以例如使用边缘或线(在某些情况下具有面或表面)来提供CAD建模对象的表示。线、边缘或表面可以以各种方式来表示,例如非均匀有理B样条(NURBS)。具体而言,CAD文件包含可以从中生成几何形状的规范,进而允许生成表示。建模对象的规范可以存储在单个CAD文件或多个CAD文件中。表示CAD系统中的建模对象的文件的典型大小在每零件1兆字节的范围内。并且建模对象通常可以是数千个零件的组合。

[0057] 在CAD的背景下,建模对象通常可以是3D建模对象,例如,其表示产品(诸如零件或零件组合)或者可能是产品组合。所谓“3D建模对象”,意指由允许其3D表示的数据建模的任何对象。3D表示允许从各个角度查看零件。例如,当表示3D时,可以处理3D建模对象并且围绕其任何轴或围绕显示表示的屏幕中的任何轴来旋转它们。特别地,这不包括未被3D建模的2D图标。3D表示的显示有助于设计(即,提高设计者统计完成其任务的速度)。由于产品的

设计是制造过程的一部分,因此这可以加快工业制造过程。

[0058] CAD系统可以是基于历史的。在这种情况下,可以通过包括几何特征历史的数据进一步定义建模对象。建模对象的确可以由自然人(即设计者/用户)使用标准建模特征(例如,拉伸、旋转、切割和/或倒圆)和/或标准表面特征(例如,扫描、混合、放样、填充、变形和/或平滑)来设计。支持这样的建模功能的许多CAD系统是基于历史的系统。这意味着设计特征的创建历史通常是通过非循环数据流保存的,该非循环数据流通过输入和输出链接将所述几何特征链接在一起。自80年代初以来,基于历史的建模范例已广为人知。可以通过两个持久数据表示来描述建模对象:历史和B-rep(即边界表示)。B-rep是在历史中定义的计算的结果。当表示建模对象时,显示在计算机屏幕上的零件的形状为B-rep(例如镶嵌)。零件的历史是设计意图。基本上,历史收集关于建模对象已执行的操作的信息。B-rep可以与历史一起保存,以使显示复杂零件更加容易。历史可以与B-rep一起保存,以便允许根据设计意图对零件进行设计更改。

[0059] 推断方法允许在馈送3D形状的离散几何表示而没有任何这种历史时(例如,由于未保存历史,或者离散几何表示或B-Rep是从远程系统接收的,或者是在没有历史的数据库中检索到的),通过可编辑特征树检索几何特征的这种历史。

[0060] 本文中的任何3D形状可以表示在现实世界中要制造的产品几何形状,诸如(例如机械的)零件或零件组合(或等效地零件组合,因为从这些方法的角度来看,零件组合可以看作是零件本身,或者这些方法可以独立地应用于组合的每个零件),或更一般地,可以是任何刚体组合(例如,移动机构)。该产品可以是各种且无限的工业领域中的产品之一,包括:航空航天、建筑、建造、消费品、高科技设备、工业设备、运输、海洋和/或海上石油/天然气生产或运输。因此,3D形状可以表示工业产品,其可以是任何机械零件,诸如陆地运载工具(包括例如汽车和轻型卡车设备、赛车、摩托车、卡车和汽车设备、卡车和公共汽车、火车)的零件、航空运载工具(包括例如机身设备、航空航天设备、推进设备、国防产品、航空设备、太空设备)的零件、海军运载工具(包括例如海军设备、商船、近海设备、游艇和工作船、船舶设备)的零件、通用机械零件(包括例如工业制造机械、重型移动机械或设备、已安装的设备、工业设备产品、金属制品、轮胎制造产品)、机电或电子零件(包括例如消费类电子产品、安全和/或控制和/或仪器产品、计算和通信设备、半导体、医疗设备和设备)、消费品(包括例如家具、家庭和花园产品、休闲用品、时尚产品、硬商品零售商的产品、软商品零售商的产品)、包装(包括例如食品和饮料和烟草、美容和个人护理、家用产品包装)。

[0061] 如从CAD领域已知的,可编辑特征树在本文中是可编辑数据结构,其包括表示应用于叶几何形状的几何运算的树布置的数据。树布置的每个叶节点表示相应的叶几何形状,并且树布置的每个非叶节点表示要应用于其子节点的相应的几何运算,也被称为“特征”。因此,将几何运算的树布置应用于叶几何形状等同于:从叶几何形状开始,并且按照树布置依次应用每个非叶节点的几何运算。由可编辑特征树表示的3D形状对应于可编辑特征树的根节点的结果。

[0062] 叶几何形状可以是任何类型。叶几何形状可以例如各自包括相应的3D形状(例如,由其组成),诸如相应的3D实体。几何运算可以是任何类型。在示例中,几何运算全部是相同类型的,例如全部是加法/并集运算。在这样的示例中,推断方法可以被配置为推断限于仅具有所述单一类型的几何运算的那些可编辑特征树。在替代和更好的示例中,几何运算可

以包括不同类型的运算(例如不同的布尔运算类型(即二进制和连续运算符,例如,其将新的3D形状与两个3D形状进行关联)),包括加法/并集运算类型、减法运算类型和/或相交运算类型。在这样的示例中,推断方法可以被配置为利用可用的所有类型的几何运算来推断可编辑特征树。特别地,本文中的任何可编辑特征树可以是从实体建模领域已知的构造实体几何(CSG)树。在特别准确的示例中,推断方法被配置为仅推断CSG树。树布置可以是任何类型。特别地,本文中的任何可编辑特征树可以是单根树、二叉树和/或满二叉树(即,每个非叶节点具有至少一个叶节点作为子节点类型的二叉树)。在特别高效的示例中,推断方法被配置为仅推断呈现这种布置类型的树。如将认识到的是,推断方法可以被配置为经由被提供给学习方法的对应的神经网络架构来推断受限于某些条件的可编辑特征树。

[0063] 用户可选择和编辑可编辑特征树的每个叶几何形状。在其中几何运算可以包括不同类型的运算的可编辑特征树的情况下,示例中的几何运算中的一个或多个几何运算可由用户选择和编辑。在替代示例中,几何运算不是可编辑的。在所有示例中,可编辑特征树的分支可以被删除和/或可以添加分支。分支本身就是应用于叶几何形状的几何运算的树布置。

[0064] 叶几何形状或几何操作的编辑可以由用户以在CAD领域中已知的任何方式来执行,诸如包括选择其具有当前参数值的至少一个参数以及修改所述参数值。与对离散几何表示(诸如原始网格)的编辑相比,这种编辑呈现人体工程学特性。特别地,推断出的可编辑特征树可以包括比输入网格的离散几何实体的数量低超过5倍(例如超过10倍)的数量的叶几何形状,和/或包括低于100(例如低于50)的数量的叶几何形状。每个叶的几何形状可以由低于20(例如低于10)的数量的参数定义。因此,与离散几何表示相比,可编辑特征树以更紧凑和更有组织的方式表示3D形状。此外,如从CAD领域所知,作用于可编辑特征树的单个参数可能呈现全局影响,而在离散几何表示作用于单个几何实体的情况下,则仅呈现局部影响。例如,本文中的任何可编辑特征树可以被配置用于通过修改小于5(例如小于3)的数量的参数值来按比例缩放3D形状的(例如弯曲的,例如圆柱形的)部分或整个3D形状,而不是移动/增加大量的网格三角形。另外,与离散几何表示相比,可编辑特征树在语义上对用户更有意义。

[0065] 在示例中,推断方法可以包括显示可编辑特征树的图形表示,包括表示每个对应的叶几何形状或几何运算的图形项目的树布置。该编辑可以包括例如通过图形交互来选择图形项。本文中的图形交互意指操作触觉设备以激活与显示器上的图形项目的位置相对应的位置处的像素。例如,可以将鼠标光标放置在显示的图形项目上,或者可以经由触摸屏直接接触显示的图形项目,或者经由触模板间接接触显示的图形项目。该选择可以触发对话框等的打开,以及由用户例如经由键盘输入一个或多个参数值。

[0066] 在示例中,推断方法可以包括显示离散几何表示的图形表示,并且同时,由用户启动神经网络的应用和可编辑特征树的推断,因此,与显示离散几何表示的图形表示同时显示可编辑特征树的图形表示。然后,用户可以隐藏离散几何表示的图形表示,以编辑可编辑特征树。替代地,推断方法可以继续同时显示两个图形表示,并且在用户执行编辑时,该方法可以相应地更新(例如,改变)离散几何图形的图形表示。这向执行编辑的用户提供视觉反馈。

[0067] 在示例中,推断方法被配置用于将根据对真实对象的物理测量结果确定的3D网格

或3D点云作为输入,并且该推断方法允许确定表示真实对象的3D形状的对应的可编辑特征树。然后,用户可以执行对可编辑特征树的编辑,例如以创建类似于真实对象但与其不同的设计。在替代和额外的示例中,推断方法被配置用于将不具有对应的可编辑特征树的、从远程计算机系统接收或从非易失性存储器取得的3D建模对象的3D网格作为输入,并且该推断方法允许确定表示3D形状的对应的可编辑特征树。然后,用户可以执行对可编辑特征树的编辑,例如以创建类似于3D网格但与其不同的设计,或完成由3D网格形成的设计。

[0068] 除了以更符合人体工程学的方式执行对3D形状的编辑的可能性之外或者替代该可能性,获得可编辑特征树还可以用于其它目的,例如3D形状合并或机械仿真。

[0069] 3D形状合并方法可以包括:在计算机系统处提供表示第一3D形状的第一可编辑特征树;提供第二3D形状的离散几何表示;对离散几何表示执行推断方法以获得表示第二3D形状的第二可编辑特征树;以及将第一可编辑特征树和第二可编辑特征树合并。可以根据任何合并技术来执行合并和/或表示第一3D形状与第二3D形状之间的机械组装或机械配合。合并可以例如包括将第二树连接到第一树。3D形状合并方法还可以包括:将几何模型应用于合并的树,以获得由此表示的3D形状的图形表示;和/或向用户显示所述图形表示。

[0070] 可以基于可编辑特征树来执行机械仿真。实际上,由于可编辑特征树以紧凑的形式(相对于离散几何表示)和参数化方式表示3D形状,因此可以在可编辑特征树上更准确地执行机械计算。特别地,更容易并且更直接根据CSG来计算有限元网格(如论文“Finite-element mesh generation from constructive-solid-geometry models”Edwin Boender, Willem F. Bronsvort, Frits H. Post, 见“计算机辅助设计”,第26卷,第5期,1994年5月,第379-392页;或论文“Finite element mesh generation methods: a review and classification”, K Ho-Le, 第38页,“计算机辅助设计”,第20卷(1988年1月1日/2月1日)所教导的)。此外,CSG还保证了表面的防水性。因此,机械仿真可以包括根据可编辑特征树(例如,CSG)和/或其一个或多个修改来计算一个或多个有限元网格,并且然后,基于每个有限元网格来执行有限元分析。此外,由于CSG是参数化的,因此可以根据仿真结果来容易地优化CSG。因此,一个或多个修改可以用于基于每个先前的有限元分析结果来优化可编辑特征树(例如,CSG)。

[0071] 现在一般地介绍数据集形成方法和学习方法。

[0072] 数据集形成方法包括提供相应的数据段。每个相应的数据段包括表示3D形状的可编辑特征树。可编辑特征树包括应用于叶几何形状的几何运算的树布置。每个相应的数据段还包括3D形状的离散几何表示。离散几何表示对应于将几何运算的树布置应用于叶几何形状的结果。数据集形成方法还包括将数据段的一部分作为训练样本插入到数据集中。插入到数据集中的一个或多个第一数据段中的每个数据段的相应的3D形状与未插入到数据集中的相应的一个或多个第二数据段的相应的3D形状相同。

[0073] 所谓“提供相应的数据段”,意指至少在数据集形成方法的某个时刻,该数据段被记录在计算机系统的存储器上并且可供处理器访问。

[0074] 所谓“数据集”,意指包括训练样本并且被配置用于机器学习的任何数据结构,该数据结构被记录在计算机系统的存储器上并且可供处理器访问。数据集形成方法可以包括将数据集记录在非易失性存储器上。

[0075] 所谓将数据段作为训练样本“插入”或“不插入”数据集,意指至少在数据集形成方

法的某个时刻(例如结尾),在如下的状态中将数据集记录在计算机系统的(例如,非易失性)存储器上:其中,数据集包括所有“插入的”数据段(即,其因此在表示数据集的数据结构内被记录在存储器上),而数据集排除(即不包括)所有“不插入的”数据段(即,其因此至少在表示数据集的数据结构内没有被记录在存储器上)。因此,至少在所述时刻并且相对于记录在存储位置上的同一数据集,所有第一数据段同时被记录,而没有记录第二数据段。然后,在进一步处理数据集之后的示例中,学习可以是基于数据集的。该学习可以涉及(例如,基本上仅涉及)第一数据段(例如,基本上所有第一数据段),而基本上不涉及第二数据段。因此,学习到的神经网络将与第一数据段一致,而不与第二数据段一致。

[0076] 数据集形成方法形成针对数字化的改进的解决方案。形成的数据集包括训练样本,每个训练样本包括相应的3D形状的离散几何表示和表示3D形状的相应的可编辑特征树。因此,该数据集被配置用于学习神经网络,在利用所提出的学习方法的示例中,该神经网络可用于将离散几何表示转换为可编辑特征树表示。另外,关于预期学习标记数据集。实际上,神经网络的潜在输入(离散几何表示)在数据集中分别由地面实况输出(相应的可编辑特征树)标记。因此,可以监督学习并且因此特别有效。此外,数据集中未填充所有提供的数据段。相反,在训练样本中,存在与3D形状相对应的一个或多个(第一)数据段,该3D形状与所提供的一个或多个(第二)数据段的相应的3D形状相同,但不是训练样本的一部分。第二数据段的这种滤除改进了学习。实际上,在最好的情况下,第二数据段形成相对于第一数据段的冗余信息。在最坏的情况下,第二数据段形成相对于第一数据段不一致的信息,从而扰乱了学习。滤除可以被称为“冗余”减少/管理。

[0077] 该学习方法包括提供数据集,该数据集包括相应的3D形状的离散几何表示。该学习方法还包括提供叶几何形状的候选集合。该学习方法还包括基于数据集和候选集合来学习神经网络。候选集合包括叶几何形状的至少一个连续子集。

[0078] 所谓“提供叶几何形状的候选集合”,意指能够使该学习方法(特别是学习步骤)将所述候选集合的每个元素视为推断的候选(至少在某个时刻)。从机器学习的领域中知道,学习步骤可以包括自由地设置神经网络的权重值,以使得推断方法倾向于执行与馈送到学习方法的数据集一致的推断。所谓“提供叶几何形状的候选集合”,意指以某种方式预先确定神经网络架构,并且以某种方式执行学习步骤,以使得候选集合的每个元素在推断期间都可用于潜在地推断。在示例中,候选集合的每个元素是每个推断的叶几何形状的候选(即,树布置中的任何位置)。

[0079] 所谓“连续的”,意指可以通过子集的另一候选叶几何形状的特定连续变形(例如,利用固定数量的参数进行参数化)来获得子集的每个候选叶几何形状,以使得连续变形的每个中间状态本身就是该子集的候选叶几何形状。在示例中,叶几何形状的候选集合包括若干连续的子集(例如,每基元形状类型一个子集)。

[0080] 该学习方法形成了针对数字化的改进的解决方案。值得注意的是,学习到的神经网络可用于将离散几何表示转换为可编辑特征树。如已经讨论的,这在数字化的背景下是特别有用的。此外,利用学习到的神经网络来执行这种数字化允许实现机器学习领域所提供的益处,诸如快速或实时执行,不需要繁琐的先前手动编程/编码以及结果的准确性(例如,此处的准确性是指在播放输出的可编辑特征树的结果与输入的离散几何表示之间评估的3D相似度的值)。因此,该学习方法在3D设计的背景下改进了人体工程学。此外,与论文

[10]中提出的解决方案相反,叶几何形状的候选集合包括叶几何形状的至少一个连续的子集。这提高了学习的准确性。实际上,由于3D形状的至少一个连续体全部都可以访问以成为推断的可编辑特征树的叶,因此可以在这种连续体中以良好的方式执行推断。

[0081] 学习可以包括基于数据集的至少一部分的监督训练。在这样的情况下,数据集的所述至少一部分包括针对每个离散几何表示的与离散几何表示一起形成(标记的)训练样本的相应的可编辑特征树。所述数据集的至少一部分可以根据任何方法来提供,例如根据数据集形成方法来形成。

[0082] 替代地或另外,学习可以包括基于数据集的至少另一部分的无监督训练。在这样的情况下,数据集的所述至少另一部分的离散几何表示各自形成相应的(未标记的)训练样本。可以根据任何方法来提供数据集的所述至少另一部分,例如作为离散几何表示的商业数据库来检索。

[0083] 在特别高效的示例中,学习方法可以包括基于通过数据集形成方法形成的数据集的监督训练。在这样的示例的稳健示例中,学习还可以包括基于另一数据集的无监督训练,例如在监督训练之后。如从机器学习领域中所知,每次训练可以包括迭代地处理相应的数据集,例如逐个小批处理,以及沿着迭代处理修改神经网络的权重值。这可以根据随机梯度下降来执行。对于每次训练,可以以任何方式初始化权重值。在示例中,可以利用在监督训练的结尾处获得的权重值来初始化无监督训练的权重值。可以以任意方式将权重值初始化以用于监督训练,例如随机地或将每个权重值初始化为零值。因此,数据集形成方法和监督训练形成了初始化权重值的方法,以用于进一步的训练,即可以基于另一数据集的无监督训练。另一数据集可以例如是未标记的数据集,其比通过数据集形成方法形成的标记的数据集更能代表现实。因此,与在监督训练之后停止相比,无监督训练可以产生更准确的结果。然而,如从机器学习领域所知,无监督训练可能非常长,并且如果任意初始化,则会提供不正确的结果。在所提出的方法中,监督学习提供了相对良好的初始化。

[0084] 该方法是计算机实现的。

[0085] 这意味着方法的步骤(或基本上所有步骤)由至少一个计算机或任何类似系统执行。因此,方法的步骤由计算机可能完全自动地或半自动地执行。在示例中,可以通过用户计算机交互来执行该方法的至少一些步骤的触发。所需的用户-计算机交互级别可能取决于预见的自动级别,并且与实现用户意愿的需求保持平衡。在示例中,该级别可以是用户定义的和/或预定义的。

[0086] 这也意味着神经网络、数据集、一个或多个(例如所有)输入和/或一个或多个(例如所有)输出可以分别由相应的数据结构表示。一个或多个(例如所有)数据结构可以分别被非暂时性地记录在相应的数据存储介质上。

[0087] 方法的计算机实现的典型示例是使用适合于此目的的系统来执行该方法。该系统可以包括耦合到存储器和图形用户界面(GUI)的处理器,该存储器具有记录在其上的计算机程序,该计算机程序包括用于执行该方法的指令。该存储器还可以存储数据库。该存储器是适合于这种存储的任何硬件,可能包括若干物理上不同的部分(例如,一个用于程序,而可能一个用于数据库)。该系统可以是CAD系统。

[0088] 图1示出了本文中的任何系统的GUI的示例,其中,该系统是CAD系统。

[0089] GUI 2100可以是典型的类似于CAD的界面,其具有标准菜单栏2110、2120以及底部

和侧面工具栏2140、2150。这种菜单和工具栏包含一组用户可选图标,每个图标与如本领域中已知的一种或多种操作或功能相关联。这些图标中的一些图标与软件工具相关联,适合于编辑和/或在GUI 2100中显示的3D建模对象2000上工作。软件工具可以被分组为工作台。每个工作台均包含软件工具的子集。特别地,工作台之一是适合于编辑建模产品2000的几何特征的编辑工作台。在操作中,设计者可以例如预先选择对象2000的一部分,并且然后通过选择适当的图标来启动操作(例如,更改尺寸、颜色等)或编辑几何约束。例如,典型的CAD操作是对显示在屏幕上的3D建模对象的打孔或折叠进行建模。GUI可以例如显示与所显示的产品2000有关的数据2500。在该图的示例中,被显示为“特征树”的数据2500和其3D表示2000涉及包括制动钳和盘的制动组件。GUI还可以示出各种类型的图形工具2130、2070、2080,例如用于促进对象的3D取向,用于触发对被编辑产品的操作的仿真或渲染所显示产品2000的各种属性。光标2060可以由触觉设备控制以允许用户与图形工具进行交互。

[0090] 3D建模对象2000可以是3D形状的离散几何表示的图形表示,并且推断方法可以允许确定可编辑特征树2500。因此,推断方法允许从没有与之相关联的特征树的3D形状的离散几何表示2000开始,并且然后,检索可编辑特征树2500。这允许使用CAD系统的所有能力。

[0091] 图2示出了本文中的任何系统的示例,其中,该系统是客户端计算机系统,例如用户的工作站。

[0092] 该示例的客户端计算机包括连接到内部通信总线1000的中央处理单元(CPU)1010、也连接到总线的随机存取存储器(RAM)1070。客户端计算机还被提供有图形处理单元(GPU)1110,其与连接到总线的视频随机存取存储器1100相关联。视频RAM 1100在本领域中也称为帧缓冲器。大容量存储设备控制器1020管理对大容量存储设备(诸如硬盘驱动器1030)的访问。适合于有形地体现计算机程序指令和数据的大容量存储设备包括所有形式的非易失性存储器,通过举例的方式,包括:半导体存储设备(诸如EPROM、EEPROM和闪存设备);磁盘(诸如内部硬盘和可移动磁盘);磁光盘;以及CD-ROM盘1040。上述任何项可以通过专门设计的ASIC(专用集成电路)进行补充或合并。网络适配器1050管理对网络1060的接入。客户端计算机还可以包括触觉设备1090,诸如光标控制设备、键盘等。在客户端计算机中使用光标控制设备以允许用户将光标选择性地定位在显示器1080上的任何所需位置。另外,光标控制设备允许用户选择各种命令,并且输入控制信号。光标控制设备包括多个信号生成设备,以用于将控制信号输入到系统。通常,光标控制设备可以是鼠标,该鼠标的按钮用于生成信号。替代地或另外,客户端计算机系统可以包括敏感垫和/或敏感屏幕。

[0093] 本文中的任何计算机程序可以包括可由计算机执行的指令,该指令包括用于使得上述系统执行该方法的装置。该程序可以记录在任何数据存储介质上,包括系统的存储器。该程序可以例如在数字电子电路中或者在计算机硬件、固件、软件中或者它们的组合中实现。该程序可以被实现为装置,例如有形地体现在机器可读存储设备中以由可编程处理器执行的产品。方法步骤可以通过执行指令程序的可编程处理器来执行,以通过对输入数据进行操作并且生成输出来执行该方法的功能。因此,处理器可以是可编程的并且被耦合以从数据存储系统、至少一个输入设备和至少一个输出设备接收数据和指令,以及向数据存储系统、至少一个输入设备和至少一个输出设备发送数据和指令。如果需要,可以用高级过程或面向对象的编程语言或汇编或机器语言来实现应用程序。在任何情况下,该语言都可以是编译语言或解释语言。该程序可以是完整的安装程序或更新程序。在任何情况下,该程

序在系统上的应用都产生用于执行该方法的指令。

[0094] 本文中的任何方法都可以是用于设计3D建模对象的过程的一部分。“设计3D建模对象”指定任何动作或一系列动作,其是拟定3D建模对象的过程的至少一部分。因此,该方法可以包括从头开始创建3D建模对象。替代地,该方法可以包括提供先前创建的3D建模对象,并且然后修改3D建模对象。

[0095] 本文中的任何方法都可以包括在制造过程中,该制造过程可以包括在执行该方法之后,产生与建模对象相对应的物理产品。在任何情况下,通过该方法设计的建模对象都可以表示制造对象。因此,建模对象可以是建模实体(即表示实体的建模对象)。制造对象可以是产品,诸如零件或零件组合。因为该方法改进了建模对象的设计,所以该方法还改进了产品的制造,并且因此提高了制造过程的生产率。

[0096] 现在讨论叶几何形状的示例。

[0097] 在数据集形成方法中提供一种或多种(例如所有)叶几何形状(作为相应的数据段的可编辑特征树的一部分),在学习方法中提供一种或多种(例如所有)候选叶几何形状,和/或由推断方法提供的一个或多个(例如所有)叶几何形状(作为推断出的可编辑特征树的一部分)可以分别与这些示例一致。

[0098] 给定基元形状类型的离散集合,并且对于每种基元形状类型,一个或多个参数域的相应的离散集合各自具有相应的连续参数,其中每个参数域具有相应的连续参数的相应的参数值,可以由相应的基元形状类型利用一个或多个参数域的相应的离散集合中的每个参数域的相应的参数值来形成叶几何形状。

[0099] 因此,数据集形成方法和/或学习方法可以是基于基元形状类型的这种离散集合的,并且对于每种基元形状类型,是基于一个或多个参数域的这种相应的离散集合的。对于数据集形成方法和学习方法,这些集合可以是相同或不同的。在集合是相同的情况下,机器学习过程特别准确,因为训练样本的叶几何形状与用于推导的叶几何形状候选一致。

[0100] 现在,学习方法中的叶几何形状的候选集合可以包括基元形状类型的离散集合之间的集合乘积,并且对于每种基元形状类型,包括一个或多个参数域的相应的离散集合。集合乘积中的每个元素表示相应的候选叶几何形状。

[0101] 在数据集形成方法的情况下,训练样本的每个叶几何形状可以属于这样的集合乘积。在示例中,数据集形成方法可以包括:提供初始数据,该初始数据包括基元形状类型的离散集合(的数据表示),并且对于每种基元形状类型,包括一个或多个参数域的相应的离散集合(的数据表示)。在这样的示例中,可以基于这样的初始数据来提供(例如,合成)叶几何形状。

[0102] 所谓“基元形状类型”,意指一类几何形状(例如,实体),其中每个几何形状呈现出共同的通用形式(例如,用户可识别的形式)。基元形状类型与参数相关联。每个参数取被称为“参数域”(其可以是间隔,即实数的连续范围)的集合中的值,并且为每个参数域提供特定的参数值定义了基元形状类型的实例。基元形状类型可以由一个或多个参数的相应集合以及预定算法来定义,该预定算法被配置用于在被馈送相应集合的参数值时生成基元形状类型的这种几何形状实例。所述实例形成可以干预一个或多个可编辑特征树的叶几何形状。

[0103] 基元形状类型可以分别与一个或多个连续参数相关联。所谓参数的“连续”,意指

输出叶几何形状的函数相对于参数至少分段地连续。也可以说参数域是至少分段地连续的。这可以允许对各种形状的紧凑表示。

[0104] 在示例中,对于一个或多个基元形状类型中的每个基元形状类型,一个或多个相应的连续参数包括一个或多个尺寸参数和/或一个或多个定位参数。尺寸参数是定义给定基元形状类型的形状的特征尺寸的大小的参数。尺寸参数对于每个基元形状类型可能是特定的。定位参数是允许在3D空间中定位形状的参数。对于所有基元形状类型,定位参数可以是相同的,例如相对于(0,X,Y,Z)参照定义的(x,y,z)坐标。

[0105] 在关于学习的计算时间和/或收敛与真实情况的多样性的表示性方面的特别高效的示例中,基元形状类型的离散集合可以包括长方体类型、球体类型、一个或多个圆柱体类型、和/或一种或多种棱柱体类型。在示例中,基元形状类型的离散集合可以包括以下类型的任何一种类型或任何组合(例如,全部):

[0106] -长方体,例如通过三个定位参数和三个尺寸参数进行参数化;

[0107] -球体,例如通过三个定位参数和一个尺寸参数(直径)进行参数化

[0108] -沿X,Y或Z轴(因此对应于三种基元形状类型)定向的右圆柱体,例如通过三个定位参数、一个表示高度的尺寸参数和一个表示直径的尺寸参数进行参数化;

[0109] -沿(X,Y)、(X,Z)、(Y,Z)、(Y,X)、(Z,X)、(Z,Y)轴(因此对应于六种基元形状类型)定向的具有等腰梯形底面的直棱柱体,例如通过三个定位参数、一个表示大梯形底面的尺寸参数、一个表示小梯形底面的长度的尺寸参数、一个表示梯形的高度的尺寸参数、一个表示棱柱体的深度的尺寸参数进行参数化。

[0110] 基元形状类型的离散集合可以包括任何其它基元形状类型,诸如具有三个以上的旋转参数的未定向圆柱体、多边形底面圆柱体、非直线圆柱体、多边形底面棱柱体、非定向棱柱体和/或非规则多面体。

[0111] 本文提供的任何叶几何形状(例如,在存储器上)都可以被记录(例如,在提供的数据段、训练样本和/或推断的输出内)作为数据结构,该数据结构包括表示基元形状类型的数据(例如,文本信息),并且对于每个参数,包括表示参数值的相应数据(例如,值,诸如浮点数)。

[0112] 现在讨论数据集形成方法的示例。

[0113] 在第一示例中,插入可以包括:对于一个或多个(例如所有)第一数据段(即,插入到数据集中)中的每个第一数据段,根据第一数据段的可编辑特征树的规范性来选择第一数据段。例如,对于一个或多个(例如所有)第一数据段中的每个第一数据段,第一数据段的可编辑特征树比一个或多个(例如,基本上例如所有)相应的第二数据段(即,未插入到数据集中)中的每个第二数据段的可编辑特征树更规范。所谓表示3D形状的第一可编辑特征树相对于表示相同3D形状的第二可编辑特征树而言“更规范”,意指第一可编辑特征树的叶几何形状(例如实体)的大小(例如体积)的总和小于(或等于)第二可编辑特征树的叶几何形状(例如实体)的大小(例如体积)的总和。所谓“基本上”,意指插入可以旨在使每个相对于它们相应的第二数据段的第一数据段的规范性最大化,但是决策可能很少是非最优的,以使得有时没有达到真正最小值。为训练样本的可编辑特征树提供高规范性通过减少(例如避免)在授权非规范性的情况下可能产生的歧义,从而提高学习的准确性。

[0114] 在可与第一示例结合的第二示例中,提供相应的数据段可以包括合成数据段。

[0115] 所谓“合成”,意指包括生成可编辑特征树和离散几何表示的自动过程。在示例中,该方法可以首先合成可编辑特征树,并且然后,通过将几何运算的树布置应用于叶几何形状来确定离散几何表示(包括)。数据集形成方法可以例如包括提供用于播放可编辑特征树的预定方案,并且确定离散几何表示可以通过将该方案应用于可编辑特征树来自动执行。该方法还可以可选地进一步将噪声添加到一个或多个离散几何表示中(例如,Perlin噪声)。这允许获得更现实的离散几何表示,并且因此提高了学习的稳健性。

[0116] 合成可能是次优的(就其元素的冗余和/或规范性而言)。因此,至少一些合成数据段可以被滤除并且因此不被插入到数据集中,可选地在基于它们确定其它数据之后(例如,至少一个这样的其它数据段最终被插入到数据集中)。换句话说,该数据集未填充所有合成数据,而是填充了其选定的严格部分。

[0117] 在这样的第二示例的第一选项中(所述第一选项也适用于除了第二示例之外的其它示例中),滤除可以包括提供(第三和第四)数据段对,并且测试至少一(例如,每)对(第三和第四)数据段的3D形状之间的同一性。测试可以是预定的并且定义一对3D形状之间的相同字符:如果一对3D形状的代表性数据段之间的测试为肯定,则认为这对3D形状相同。因此,在测试产生肯定结果(并且因此确定第三和第四数据段是冗余的)时,滤除可以将第三和第四数据段中的数据段识别为第二数据段(不插入到数据集中),从而滤除所述第二数据。所识别的数据段可以可选地是最不规范的数据段(即,具有最不规范的可编辑特征树的数据段)。

[0118] 在可与第一选项组合的第二示例的第二选项中,数据段可以包括一个或多个(例如,不同和/或不相交的)数据段组,相应组的所有数据段表示相同的3D形状。换句话说,每组的所有可编辑特征树都导致表示相同3D形状的离散几何表示。在这样的情况下,数据集形成方法可以保留针对至少一个(例如,每个)组在数据集中仅插入一个或多个(例如,一个)数据段作为训练样本(可选地根据规范性(例如具有最规范的可编辑特征树的数据段),并且丢弃/删除(即不包括在数据集中)组中的任何其它数据段)。

[0119] 在第二选项的特定示例中,可以像第一选项中一样对组进行处理,即,使用“数据段对”方式进行处理,包括测试步骤。在这样的示例中,可以以任何方式提供组和对。例如,该处理可能不具有该组的先验知识。在这样的情况下,测试允许确保第三和第四数据段在同一组中。

[0120] 例如,合成可以包括生成至少一个初始数据段,并且然后,基于该初始数据段来确定至少一个将来的数据段。每个初始数据段和其将来的数据段形成相应的组。确定将来的数据段的步骤可以包括:从初始数据段开始,迭代地执行可编辑特征树编辑的步骤,(可编辑特征树编辑的步骤的)每个迭代的输出形成将来的数据段。所谓“从初始数据段开始迭代地”,意指将可编辑特征树编辑的步骤重复一次或多次,在每个时间 t 处使用先前时间的输出作为输入,并且初始(即开始/首次)输入(在初始时间 $t=0$ 处)是初始数据段的可编辑特征树。在每次迭代中,先前时间可以是之前的时间(即,对应于先前迭代,即 $t-1$)或任何其它先前时间(诸如初始时间 $t=0$)。该迭代可以对应于该组的盲探,例如旨在找到最规范的数据段,以确定作为要插入到数据集中的第一数据段。与提供原始(无组织)合成数据段集合相比,迭代地合成和提供数据段可以提高计算效率。实际上,与随机地(并且因此严酷地)对已经合成但未组织的数据进行处理相比,与合成进行迭代处理允许组织数据段并且自然地

逐组工作,从而至少减少了不必要的测试。

[0121] 特别地,测试可以与合成交织执行。例如,可以(例如,基本上仅)在可编辑特征树编辑的每个步骤之后执行测试。在这样的示例中,第三数据段可以是在当前时间确定的将来的数据段,和/或第四数据段可以是另一可用的数据段,诸如初始数据段或在先前时间确定的将来的数据段,例如输入到可编辑特征树编辑的当前步骤的数据段。测试允许知道迭代是否停留在组中,换句话说,可编辑特征树编辑的步骤是否修改了所表示的3D形状。如果测试产生肯定结果,则可以如上所述地执行滤除,并且迭代可以继续(例如,以探索组)。换句话说,滤除与合成交织,并且由此遵循合成的顺序。替代地,如果测试产生否定结果,则意味着第三数据段不属于该组。在这样的情况下,可以出于任何目的删除/丢弃或者替代地保留第三数据段(将其插入到数据集中,或者从该数据集开始另一次单独的迭代以探索单独的组)。在任何情况下,从当前迭代中丢弃第三数据段,这可以例如通过返回到第四数据段并且执行不同的可编辑特征树编辑来继续进行。

[0122] 已经讨论了根据第二选项对一个组的处理(例如,包括生成与该组相对应的初始数据段)。在第二选项的示例中,可以针对至少一个(例如,每个)其它组执行相同的处理(例如,包括生成与每个其它组相对应的初始数据段)。所有组可以并行或顺序地处理。

[0123] 在可与第一和/或第二示例组合的第三示例中,对于每个第一数据段(即,插入到数据集中),第一数据段的可编辑特征树在结构上可以与每个相应的第二数据段(即对应于第一数据段并且未插入到数据集中)的可编辑特征树兼容(例如,并且比其更规范)。当第一可编辑特征树和第二可编辑特征树潜在地产生相同的3D形状时,它们被视为“在结构上兼容”。“结构兼容性”可以对应于任何预定的树结构兼容性标准。例如,当第一可编辑特征树和第二可编辑特征树至少部分地彼此匹配时,它们可以被视为“在结构上兼容”。预定树结构兼容性标准可以测量两个可编辑特征树的元素相同的程度。实际上,具有大量相同元素的两个可编辑特征潜在地产生相同的3D形状。两个相同的可编辑特征必须在结构上兼容。然而,在结构上兼容的两个可编辑特征树可以是相同或不同的。第三示例提高了效率。

[0124] 例如,在第二示例的第一选项中,对于包括测试步骤的一个或多个(例如所有)“数据段对”方式的过程,第三和第四数据段在结构上可以兼容(即具有在结构上兼容的可编辑特征树)。这允许仅在特别有可能是肯定结果的情况下执行测试,从而提高了效率。

[0125] 替代地或另外,仍然在第二示例的第一选项中,测试可以包括体积比较方案。体积比较方案可以包括计算第一体积值和第二体积值(使用基于离散几何表示的任何体积计算方案),并且将第一体积值与第二体积值进行比较。第一体积值表示第三数据段的离散几何表示的体积。第二体积值表示第四数据段的离散几何表示的体积。这提供了快速的同一性测试。当测试中涉及的一对3D形状具有在结构上兼容的可编辑特征树时,可以以相对较低的稳健性成本实现所述速度。

[0126] 替代地或另外,在第二示例的第二选项中,至少一个(例如,每个)组的数据段可以是两两在结构上兼容的。例如,可编辑特征树编辑的步骤可以确保(即,被限制为使得)每个输出数据段在结构上与输入数据段兼容。可编辑特征树之间的结构兼容性可以形成等价关系。这提高了效率,特别是当第二示例的所述第二选项包括对第二示例的第一选项的测试时。

[0127] 在第三示例的选项中,当满足以下两个(例如,必要和/或充分)条件时,可以将第

一可编辑特征树和第二可编辑特征树视为“在结构上兼容”：

[0128] 条件1、第一可编辑特征树具有几何操作的相应的(第一)树布置,其至少部分地匹配第二可编辑特征树的几何操作的(第二)树布置。换句话说,第一树布置的至少一部分与第二树布置的至少一部分匹配(即,相等/相同)。在示例中,第一树布置可以与第二树布置匹配(即,相等/相同),或者第一(或相应地第二)树布置可以被包括在第二(或相应地第一)树布置中(即,与其一部分相匹配)。另外,每对对应的几何运算是相同的。条件1还可以可选地规定,第一树布置和第二树布置彼此完全匹配,除了它们中的一者或两者相对于另一者具有一个或多个额外的单叶分支(即,仅具有一个叶节点的分支)。换句话说,第一树布置和第二树布置可以彼此不同,仅因为第一树布置可以相对于第二树布置具有额外的叶几何形状和/或因为第二树布置可以相对于第一树布置具有额外的叶几何形状。

[0129] 条件2、每对对应的叶几何形状具有相同的相应的基元形状类型,并且对于每个非尺寸参数(除了尺寸参数之外的每个参数),每对对应的叶几何形状具有相同的相应的参数值。换句话说,每对对应的叶几何形状除了它们的尺寸参数值外都是相同的。

[0130] 所谓针对一对元素的“对应”,意指第一可编辑特征树的第一元素和第二可编辑特征树的第二元素,它们位于它们相应的可编辑特征树的匹配位置(根据条件1提供的树布置匹配)。

[0131] 满足这两个条件的可编辑特征树潜在地产生相同的3D形状,因为即使尺寸参数值可能不同,但是几何运算、基元形状类型和其它参数可能使得产生的3D形状相同。并且在示例中,由数据集形成方法提供的相应的数据段可以有效地包括一个或多个相应的数据段集合,其中每个集合的相应的可编辑特征树在结构上全部是兼容的。在示例中,一个或多个集合可以各自包括与表示不同的(即,不相同的)3D形状的可编辑特征树相对应的至少两个数据段。但是替代地或另外,该集合可以包括与具有至少一个不同的尺寸参数(对于至少一种叶几何形状的基元形状类型并且仍然表示相同的3D形状)的可编辑特征树相对应的至少两个数据段。数据集形成方法可以插入两个数据段中的单个数据段。所述单个数据段的确定可以以任何方式执行。

[0132] 现在将进一步讨论第二示例。

[0133] 在示例中,数据集形成方法可以包括提供初始数据,该初始数据包括基元形状类型的离散集合,以及对于每种基元形状类型,包括一个或多个参数域的相应的离散集合。这意味着对于每种基元形状类型,初始数据包括与对象类(指代面向对象的编程)相对应的相应的数据结构,该对象类具有与基元形状类型的参数相对应的字段。在这样的情况下,可以基于初始数据来执行合成相应的数据段。例如,提供至少一个相应的数据段可以包括基于初始数据来合成可编辑特征树。合成可以包括生成操作的树布置,并且对于树布置的每个叶节点,基于初始数据来生成叶几何形状。特别地,叶几何形状的生成可以包括选择基元形状类型,以及为基元形状类型的每个参数选择参数值。这对应于创建给定类的实例。

[0134] 在方法不限于涉及单一类型的几何运算的树的情况下,初始数据还可以包括几何运算集合。在这样的情况下,生成操作的树布置可以包括生成树布置,并且对于每个非叶节点,选择几何操作。该几何运算集合可以是离散的,例如包括一个或多个布尔运算,包括加法/并集、减法和/或交集。这尤其适用于生成CSG树。

[0135] 生成树布置可以包括生成节点和连接节点的树结构。如较早讨论的,树布置可以

限于单根树、二叉树和/或满二叉树。特别地,可以在既是单根也是全二进制的类型的树内生成一个或多个可编辑特征树,并且其中每个非叶节点具有至少一个叶节点作为子节点。在这样的情况下,生成树布置可以包括选择树布置的深度,即,非叶节点的数量。该信息足以完全定义所述类型的树。

[0136] 合成可编辑特征树时执行的任何选择可以经由(例如,随机)采样来执行。这允许生成相对较好地表示真实情况的多样性的数据。因此,这导致最终的准确学习。

[0137] 任何(例如随机)采样可以包括集合内的均匀(例如随机)采样,其中要选择的变量取值(例如,随机均匀采样是根据均匀概率分布的随机采样)。

[0138] 例如:

[0139] -可以根据所有大于N1且小于N2的整数集合上的均匀分布对深度进行采样,其中N1是大于0和/或小于5(例如等于1)的“最小深度”

[0140] 整数,并且N2是大于N1和/或小于100、50或20(例如等于7)的

[0141] “最大深度”整数。

[0142] -可以根据基元形状类型的离散集合上的均匀分布对基元形状类型进行采样。

[0143] -可以根据布尔运算的离散集合(例如,由加法/并集、减法和交集组成)上的均匀分布每个布尔运算进行采样。

[0144] -可以根据超矩形(间隔的笛卡尔积,每个间隔形成特定于基元形状类型的相应参数的参数域)中的均匀分布对每种所选的基元形状类型的参数值进行采样。为简化起见,数据集形成方法可以在超立方体 $[a, b]^l$ 中对上述每个基元形状类型的所有参数进行均匀地采样,其中 l 是基元形状类型的参数数量,并且 a 和 b 是实数, a 可以大于或等于0和/或小于或等于1(例如等于0.2)。 b 可以大于或等于1和/或小于或等于10(例如等于2)。

[0145] 现在进一步讨论第二示例中的第二选项的具体示例。

[0146] 当对于一种或多种基元形状类型中的每种基元形状类型,一个或多个相应的连续参数包括一个或多个尺寸参数时,在示例中,合成可以包括以下两种操作之间的混合:分别根据初始数据(如上所述)生成一个或多个初始数据段;以及基于每个这样生成的初始数据段,基于对所述生成的数据段的修改来推导出一个或多个相应的其它(将来的)数据段。在示例中,合成可以包括基于初始数据对至少一个(例如,若干)初始数据段的初始生成。对于(每个叶几何形状的)每个尺寸参数,初始生成包括初始参数值的较早描述的选择(例如,经由采样,可选地经由随机采样)。在这样的示例中,合成然后可以包括(例如,对于一个或多个(例如每个)生成的初始数据段):从初始数据段开始,迭代地确定一个或多个将来的数据段;修改相应的尺寸参数的一个或多个参数值和/或删除一个或多个叶几何形状。

[0147] 特别地,从初始数据段开始,迭代过程可以包括:减小(例如,最大程度地)一个或多个(例如,尽可能多的)尺寸参数值和/或删除一个或多个(例如,尽可能多的)叶几何形状,同时保持表示的3D形状相同。换句话说,迭代过程的目的在于用表示相同3D形状的更规范的数据段替换初始数据段,例如用通过减小尺寸参数值和/或删除叶几何形状可获得的并且仍然表示与初始数据段相同的3D形状的最规范的数据段来替换初始数据段。

[0148] 在示例中,迭代处理可以包括从初始数据段开始的约简算法,该约简算法包括:

[0149] a、初始化:

[0150] o第三数据段(即变量),以及

- [0151] o第四数据段(即变量)；
- [0152] b、将第四数据段的值设置为初始数据段；
- [0153] c、标记为“未访问”；
- [0154] o第四数据段的可编辑特征树的所有叶几何形状,以及
- [0155] o第四数据段的可编辑特征树的所有叶几何形状的所有尺寸参数；
- [0156] d、选择未访问叶并且将所选择的叶标记为已访问(否则,如果没有未访问叶是可用的,则跳到步骤k)；
- [0157] e、选择所选择的叶的未访问尺寸参数并且将所选择的尺寸参数标记为已访问(否则,如果没有未访问尺寸参数是可用的,则返回到步骤d)；
- [0158] f、初始化大于0且小于所选择的尺寸参数的参数值的参数约简值(例如,所选择的尺寸参数的参数值的一半,其对应于二等分算法)；
- [0159] g、将第三数据段的值设置为从所选择的尺寸参数的参数值中减去参数约简值的结果；
- [0160] h、测试第三数据段的3D形状与第四数据段的3D形状之间的同一性(例如,经由对第三数据段和第四数据段的离散几何表示之间的简单体积比较,因为它们的可编辑树在结构上是兼容的)；
- [0161] i、如果测试产生肯定结果,则：
 - [0162] o用第三数据段的值替换第四数据段的值(第四数据段被识别为第二数据段,并且因此可以删除其值)；
 - [0163] o返回到步骤f；
- [0164] j、否则(即测试产生否定结果)：
 - [0165] o减小参数约简值(例如减半,其对应于二等分算法)；
 - [0166] o如果参数约简值高于预定门限(例如0.1或0.05),则返回到步骤g,否则返回到步骤e；
- [0167] k、将第四数据段的可编辑特征树的所有叶几何形状标记为“未访问”；
- [0168] l、选择未访问叶并且将所选择的叶标记为已访问(否则,如果没有未访问叶是可用的,则结束该算法)；
- [0169] m、将第三数据段的值设置为从第四数据段中删除所选择的叶的结果；
- [0170] n、测试第三数据段的3D形状与第四数据段的3D形状之间的同一性(例如,经由对第三数据段和第四数据段的离散几何表示之间的简单体积比较,因为它们的可编辑树在结构上是兼容的)；
- [0171] o、如果测试产生肯定结果,则：
 - [0172] o用第三数据段的值来替换第四数据段的值(第四数据段被识别为第二数据段,并且因此可以删除其值)；
 - [0173] o返回到步骤I。
- [0174] 该约简算法包括:在叶上的第一“for”循环,以用于减少尽可能多的几何形状叶；以及在嵌入在第一“for”循环中的尺寸参数上的第二“for”循环,以用于最大程度地减小每个尺寸参数；以及然后在叶上进行的第三“for”循环,以用于删除尽可能多的几何形状叶。
- [0175] 每个基元形状类型的减小可以逐尺寸参数来执行,例如根据X、Y或Z轴。可以任意

设置要考虑的尺寸顺序。一个尺寸的减小可能影响至少一个参数。例如,如果合成尝试缩小的是立方体,则如果该立方体的三个尺寸(长度、宽度、深度)在X、Y、Z上对齐,则每个尺寸只有一个影响参数。相反,在更复杂的情况下,诸如对沿轴(X,Y)定向的右等腰梯形棱柱体的减小,若干参数可能会并行影响。

[0176] 现在讨论学习方法的示例。

[0177] 在示例中,神经网络可以包括递归神经网络(RNN)单元(即,相同RNN的单元)。这样的示例允许学习准确的神经网络,并且由此执行准确的推断,该神经网络呈现(经由RNN)与要推断的可编辑特征树的顺序特征相对应的顺序架构。

[0178] 特别地,每个RNN单元在相应的时间步处输出用于推断相应的基元形状类型的相应的第一数据和用于推断一个或多个参数域的相应的离散集合的每个参数域的相应的参数值的相应的第二数据。因此,输出的数据允许推断相应的叶几何形状。可以以任何方式(例如预定的(例如,所推断的可编辑特征树可以限于加性树,以使得每个几何运算都是加法))来推断几何运算。替代地,每个RNN单元还可以在相应的时间步处输出用于推断相应的几何运算的相应数据(例如,在预定的几何运算集合中,例如包括加法/并集、减法和/或交集或由其组成)。时间步的数量可以是预定的。替代地,每个RNN单元还可以输出用于推断可编辑特征树的结束或继续决策(即,结束标记符)的相应数据。

[0179] 现在讨论RNN输出的特别有效的示例。

[0180] 相应的第一数据可以包括概率的相应的离散分布,每个概率归因于基元形状类型的离散集合中的相应的离散集合。换句话说,每个RNN单元输出在基元形状类型上的概率分布,以使得可以针对与相应的时间步相对应的叶几何形状来推断出可能的基元形状类型。

[0181] 替代地或另外,相应的第二数据包括(对于每种基元形状类型)针对一个或多个参数域的相应的离散集合中的每个参数域的相应的参数值。换句话说,RNN单元直接输出用于推断出的基元形状类型的参数值。这简化了学习并且提高了收敛性和准确性,特别是因为参数是连续的并且很难学习输出密度分布的神经网络。

[0182] 另外,用于推断相应的几何运算的可选的相应数据可以包括概率的相应(例如离散)分布,每个概率归因于预定的(例如离散)几何运算集合中的相应的几何运算。此外,用于推断结束标记符的相应数据可以包括结束的概率和/或继续的概率(例如,伯努利分布),这与经典RNN一样。

[0183] 现在讨论产生特别准确的结果的神经网络的架构。

[0184] 在选项中,神经网络可以包括初始编码器,该初始编码器将离散几何表示或其表示作为输入。在示例中,编码器可以是点云编码器(例如,PointNet编码器),并且直接将点云(形成离散几何表示或表示离散几何表示和/或从离散几何表示提取)作为输入。在其它示例中,编码器可以是2D卷积神经网络(CNN),并且将2D图像数据(表示离散几何表示和/或从离散几何表示中提取)作为输入。2D图像数据可以是离散几何表示的单通道渲染(即,单视图图像)或多通道渲染(即,多视图图像)。在高效示例中,多通道渲染可以是基于在离散几何表示周围分布的四个视点(例如,规则地)的四视图渲染,例如基本上在离散几何表示周围形成正方形。

[0185] 在替代或额外选项中,每个RNN单元可以将当前预测序列(反馈回路)的结果作为(额外)输入。这提高了准确性。反馈回路是可能的,因为由于所涉及的连续参数,因此反馈

回路的输入是可区分的。在这样的选项中,神经网络可以包括应用于每个RNN单元的结果以计算当前预测序列的预定几何建模器(即,静态的,即是不学习的,换句话说,不包括任何可修改的权重)。建模器可以是提供可编辑特征树与其表示(例如多通道渲染)之间的映射的任何数据结构。几何建模器可以包括用于播放可编辑特征树,以及然后提取所得3D形状的方案。

[0186] 在这样的选项的示例中,每个RNN单元可以包括相应的编码器。相应的编码器可以应用于RNN的输入,例如以计算表示输入的全局特征。取决于几何建模器的输出,在示例中,如果几何建模器输出点云,则每个相应的编码器可以是点云编码器(例如,PointNet编码器)。在其它示例中,如果几何建模器输出2D图像数据,则每个相应的编码器可以是2D CNN。2D图像数据可以是单通道渲染(即,单视图图像)或多通道渲染(即,多视图图像),诸如在先前介绍的示例中。

[0187] 在这样的示例中,每个相应的编码器在每个时间步处可以相同。替代地或另外,一个或多个(例如,全部)相应的编码器可以可选地与初始编码器相同。这提高了过程的稳健性。

[0188] 对于与每个离散几何表示(每个离散几何表示在此与相应的地面真值可编辑特征树相关联)的每个叶几何形状相对应的时间步,导致准确结果的监督训练的损失的一个示例惩罚以下量中的一者或两者:

[0189] -归因于相应的地面真值叶几何形状的相应的基元形状类型的相应的第一数据的概率的低下,和/或

[0190] -相应的地面真值叶几何形状的一个或多个相应的参数值与相应的第二数据的一个或多个相应的参数值之间的差异。

[0191] 换句话说,监督训练可以通过使这种损失最小化而作用于神经网络的权重,从而倾向于使每个RNN单元输出的相应概率针对相应的地面真值基元形状类型接近1(即不低)并且针对其它基元形状类型接近0(即低),和/或使每个单元输出的相应的连续参数值接近其地面真值。这样的损失准确地处理问题的离散/连续混合。

[0192] 在该示例的选项中,对于与每个离散几何表示的每个叶几何形状相对应的时间步,损失还可能类似地惩罚额外量中的一者或两者:

[0193] -归因于对应的地面真值几何运算的相应概率(由RNN单元输出)的低下,和/或

[0194] -基于用于推断结束标记符的相应数据,到达相应的地面真值可编辑特征树的树布置(例如深度或长度)的概率(由RNN单元输出)的概率是低的。

[0195] 现在讨论导致精确结果的无监督训练的示例,其中相应的第一数据包括相应的离散概率分布,每个概率均归因于基元形状类型的离散集合中的相应的离散集合。

[0196] 众所周知,无监督训练可以使损失最小化。最小化包括探索相应的候选离散概率分布。换句话说,最小化迭代地修改神经网络权重以探索输出的相应的离散概率分布并且使它们与未标记的数据集一致(其中每个离散几何表示均不与任何地面真值可编辑特征树相关联)。从某种意义上说,对相应的候选离散概率分布的探索是间接的,并且是对权重的直接探索的结果。在选项中,最小化还包括探索在几何运算上的相应的候选离散概率分布和/或针对结束标记符的相应的候选离散概率分布。在示例中,最小化还包括探索一个或多个参数域的相应的离散集合的每个参数域的相应的候选参数值。

[0197] 现在,对于数据集的每个离散几何表示以及对于每个候选,损失惩罚对应的离散几何表示的差异。相应的离散几何表示是根据相应的可编辑特征树推导出的,可编辑特征树本身是可基于所探索的相应的候选离散概率分布来推断的。换句话说,给定神经网络的权重值(在无监督训练的任何迭代中),可以向神经网络输入未标记数据集的离散几何表示,以执行可编辑特征树的推断。每个推断出的可编辑特征树本身可以是基于推导出离散几何表示(通过播放树并且确定结果的离散的几何表示)的。可以将这种推导出的离散几何表示与初始对应的数据集的离散几何表示进行比较。

[0198] 在示例中,无监督训练可以包括用于至少一个(例如,每个)离散几何表示的采样步骤。训练可以包括提供小批量,并且采样可以包括对于每个小批量,通过遵循神经网络输出的数据(包括概率分布和RNN输出的连续参数值),针对每个小批量的离散几何表示对可编辑特征树进行采样。然后,采样可以计算损失,其将输入的离散几何表示(或其表示,例如渲染)与对应于采样的可编辑特征树的离散几何表示进行比较(即,对它们之间的差异进行惩罚)。由于采样是随机且离散的,因此该损失相对于采样是不可区分的。因此,由于不存在真实梯度,因此无法执行相对于变量的这种真实梯度的经典反向传播。

[0199] 所提出的无监督训练的示例建议使用强化方法来提供以下各项:损失相对于表示相应的第一数据的相应的候选离散分布而言的变量的伪梯度;以及损失相对于表示在几何运算上的相应的候选离散分布而言的变量的伪梯度;和/或损失相对于表示结束标记符分布而言的变量的伪梯度。然后,最小化可以包括这种损失的伪梯度的反向传播(例如,随机下降)以学习神经网络。由于伪梯度是由增强算法提供的,因此即使反向传播不是真实梯度之一,该学习也是准确的。最小化还可以包括损失相对于一个或多个参数域的相应的离散集合中的每个参数域而言的(真实)梯度的反向传播(因为损耗相对于这些变量而言是可微分的)。因此,强化方法解决了引入离散概率分布而引起的可微性问题,并且仅解决了这一问题。

[0200] 现在讨论包括方法的示例的示例机器学习过程。

[0201] 递归神经网络

[0202] 示例机器学习过程是基于递归神经网络(RNN)的。递归神经网络是具有内部存储器状态的深度神经网络。因此,与前馈神经网络不同,递归神经网络可以表现出时间行为。它们隐式地考虑过去的条目以预测下一个结果。它们可以写为

$y_t = f_w^1(x_t, h_t)$, $h_{t+1} = f_w^2(x_t, h_t)$,其中 y 是输出, x 是输入,并且 h 是内部隐藏状态(也被称为存储器)。 w 是RNN的权重。存储器 h 不是堆积所有过去输入的硬存储器,它是一个固定大小的向量,通过 f_w^2 来学习,以使得RNN学习仅保留来自过去步骤的有用信息。

[0203] 示例机器学习过程的目的

[0204] 从原始网格开始,示例机器学习过程允许在没有任何监督的情况下构建CSG树,该CSG树由利用布尔运算按顺序合并的基元序列组成。为了将原始网格转换为CAD格式,该示例机器学习过程可能特别有用。这种CSG数字化可以允许将原始表面转换为可编辑的3D模型,然后可以将其用于变形模型,将其连接到机械组件或对其进行仿真。这些原始网格可以来自真实3D机械零件的激光扫描,或者可以来自3D市场上获得的B-rep模型。

[0205] 示例机器学习过程的原理

[0206] 示例机器学习过程允许学习递归神经网络,该递归神经网络根据原始网格来预测CSG树。首先,随机生成CSG树的合成数据集。数据集中的每个CSG树都被简化为其最小表达,以便移除任何冗余。该数据集用于以监督方式训练递归神经网络,以根据原始网格来预测CSG树。然后,通过在原始网格(其CSG树是未知的)的真实数据集上进行强化学习,以无监督方式来细化网络。最后,该网络用于根据新的原始网格来推断出合理的CSG树。使用优化方案来细化树的连续参数,以便使CSG树最适合输入网格。

[0207] 此外,示例机器学习过程中的任何CSG树都是单根和满二叉树(在这种类型中,每个非叶节点具有至少一个叶节点作为子节点)。

[0208] 示例机器学习过程的效果

[0209] 与基于曲率分析的其它确定性方法相比,示例机器学习过程对输入网格中的噪声和不完整数据的稳健性更高。出于若干原因,示例机器学习过程提供了相比于论文[10]的改进。首先,论文[10]仅适用于离散化基元,而示例机器学习过程则处理连续基元(参数实体)。其次,示例机器学习过程提供了稳健的最终优化,以不断细化CSG树。第三,示例机器学习过程的RNN包括反馈回路,该反馈回路简化了推断过程。第四,示例机器学习过程的预训练步骤更加合理,因为其被应用于CSG数据集,其中根据新算法,所有CSG树均已减少以避免冗余。

[0210] 对示例机器学习过程的描述

[0211] 1、生成合成CSG数据集

[0212] 在第一步中,示例机器学习过程将生成CSG树的合成数据集。树是利用每个基元之间的布尔运算进行交织的基元序列。每个布尔运算是二进制运算符,其被应用于接下来的基元和先前运算的结果。三种常见的布尔运算是相加/相减/相交。因此,将序列中的每个基元与序列的当前结果(相加/相交/相减),如图3中所示。

[0213] 基元属于一组基本类型(被标记为 $\{1, \dots, L\}$,诸如长方体、圆柱体、棱柱体等)。每个基元类型都具有其自己的一组连续参数(例如,长方体的大小和位置)。

[0214] 为了对CSG树进行采样,示例机器学习过程对其深度、基元类型和布尔运算进行随机采样。对于每个基元,示例机器学习过程还对其连续参数进行采样。因此,示例机器学习过程获得随机CSG树。通过将CSG的布尔运算顺序地应用于其基元序列,可以将该CSG转换为原始网格。

[0215] 示例机器学习过程最终获得具有对应的CSG树的原始网格的数据集。预训练步骤的主要问题之一是CSG树的无限序列可以对应于完全相同的原始几何形状。例如,有无穷尽的方法使两个立方体相交,从而得到相同的产生的立方体。图4示出了示例。

[0216] 训练将相同的原始输入网格映射到不同的CSG树的网络要困难得多。为了减轻这个问题,示例机器学习过程涉及一种算法,该算法将每个采样的CSG树减少到其最小表达,以便在CSG树与其产生的网格之间强制实现双射。

[0217] 首先,示例机器学习过程使用二等分算法,以便减小每个基元沿其所有轴的尺寸,同时将产生的3D形状约束为保持不变。为了获得快速测试以检查结果是否保持不变,示例机器学习过程对产生的3D形状的面积进行比较。示例机器学习过程不断缩小每个基元,而产生的CSG的面积保持不变。

[0218] 其次,如果不改变结果,则示例机器学习过程在树中移除任何基元以及其接下来

的操作。

[0219] 该归简算法由图5-6示出。

[0220] 最后,示例机器学习过程还可以将噪声(诸如Perlin噪声,这种噪声取决于应用)添加到产生的网格中。

[0221] 2、监督学习递归神经网络

[0222] 示例机器学习过程定义了如下的递归神经网络(RNN)。RNN将3D网格的离散表示I(例如多通道图像I,其中每个通道表示输入网格、3D体素网格、3D点云等的渲染视图或深度视图)作为输入进行数字化。

[0223] RNN包括应用于其输入的编码器,以计算表示输入形状的全局特征。这由图7示出。

[0224] 如果输入是图像,则编码器可以例如是2D CNN;如果输入是点云,则编码器可以例如是PointNet编码器,诸如以下论文所述:

[0225] -Charles R Qi,Hao Su,Kaichun Mo,and Leonidas J Guibas.Pointnet:Deep learning on point sets for 3d classification and segmentation.In Computer Vision and Pattern Recognition,2017年。

[0226] CNN在CSG的每个步骤处都可以相同。在CSG的步骤处使用的所述CNN本身可以与初始CNN相同或不同。在相同的情况下,该过程更加稳健。

[0227] 每个RNN单元也可以可选地将当前预测序列的结果(反馈回路)作为输入。

[0228] 当前预测序列的结果是利用几何建模器计算的。建模器m可以被看作是CSG序列与其产生的表示(例如,多通道渲染)之间的映射。

[0229] 每个RNN单元在每个时间步t处输出基元类型的离散分布、布尔运算上的分布o以及结束标记符(其指示CSG是否结束)上的伯努利分布q;q是关于当前步t不是当前CSG序列的最后一步的概率,并且因此,1-q是关于当前步是序列中的最后一步的概率。 p_y 是基元类型y的概率,并且 o_y 是运算y的概率。

[0230] 其还在每个时间步t处输出每种基元类型l的连续参数 g_l 。

[0231] 图8示出了通过示例机器学习过程学习的神经网络的示例架构。

[0232] 使用合成数据集以监督方式对该RNN进行了预训练。注意,也可以使用真实的CSG数据集。

[0233] 令N为数据集的大小,以及 \widehat{T}_n 为第n CSG序列的深度。每个CSG序列被写为基元类型序列 $(\widehat{a}_1^n, \dots, \widehat{a}_{\widehat{T}_n}^n)$ 和运算序列 $\widehat{b}_2^n, \dots, \widehat{b}_{\widehat{T}_n}^n$ 。对于每个基元 \widehat{a}_t^n ,令 \widehat{x}_t^n 为其参数。第n CSG的输入表示被表示为 I^n 。

[0234] 我们还通过 \widehat{I}_t^n 来表示在时间步t处的已知CSG树n的中间结果的表示,即 $\widehat{I}_t^n = m([\widehat{a}_1^n, \dots, \widehat{a}_t^n], [\widehat{x}_1^n, \dots, \widehat{x}_t^n], [\widehat{b}_2^n, \dots, \widehat{b}_t^n])$ (并且 $\widehat{I}_0^n = 0$)。

[0235] 字母上的帽子指示CSG序列的地面真值标签。

[0236] 我们的RNN的一般等式:
$$\begin{pmatrix} p \\ o \\ q \\ g \end{pmatrix}_t = f_w^1(I, I_{t-1}, h_t) \quad , \quad h_t = f_w^2(I, I_{t-1}, h_{t-1}), h_0$$

=0,其中 w 是神经网络的权重。我们回想反馈回路(RNN的输入 I_{t-1})是可选的。

[0237] 示例机器学习过程为了训练网络的权重而使最小化的损失(使用标准的深度学习技术,诸如小批量处理上的ADAM求解器)可能是如下:

$$[0238] \quad L_1(w) = L_{\text{primitives}}(w) + L_{\text{operations}}(w) + L_{\text{length}}(w) + \lambda L_{\text{parameters}}(w)$$

$$[0239] \quad L_{\text{primitives}}(w) = - \sum_{n=1}^N \frac{1}{\widehat{T}_n} \sum_{t=1}^{\widehat{T}_n} \log [p_{a_t^n}(I^n, \widehat{I}_{t-1}^n, h_t^n)]$$

$$[0240] \quad L_{\text{operations}}(w) = - \sum_{n=1}^N \frac{1}{\widehat{T}_n - 1} \sum_{t=2}^{\widehat{T}_n} \log [o_{b_t^n}(I^n, \widehat{I}_{t-1}^n, h_t^n)]$$

$$[0241] \quad L_{\text{length}}(w) = - \sum_{n=1}^N \left[\frac{1}{\widehat{T}_n} \log [1 - q(I^n, \widehat{I}_{\widehat{T}_n-1}^n, h_{\widehat{T}_n}^n)] \right.$$

$$\left. + \frac{1}{\widehat{T}_n} \sum_{t=1}^{\widehat{T}_n-1} \log [q(I^n, \widehat{I}_{t-1}^n, h_t^n)] \right]$$

$$[0242] \quad L_{\text{parameters}}(w) = \sum_{n=1}^N \frac{1}{\widehat{T}_n} \sum_{t=1}^{\widehat{T}_n} \left\| \widehat{x}_t^n - g_{a_t^n}(I^n, \widehat{I}_{t-1}^n, h_t^n) \right\|_2^2$$

[0243] λ 是离散变量的分类损失 $L_{\text{primitives}}(w) + L_{\text{operations}}(w) + L_{\text{length}}(w)$ 与连续回归损失 $L_{\text{parameters}}(w)$ 之间的折衷参数。例如,可以将其初始化为

$$\lambda = \frac{L_{\text{primitives}}(w_0) + L_{\text{operations}}(w_0) + L_{\text{length}}(w_0)}{L_{\text{parameters}}(w_0)}, \text{其中 } w_0 \text{ 是神经网络的随机初始化权重。}$$

[0244] 除了 \log (交叉熵项)之外的任何低惩罚函数都可以用于项 $L_{\text{primitives}}(w)$, $L_{\text{operations}}(w)$ 和 $L_{\text{length}}(w)$ 中的任何项,和/或除了L2距离之外的任何距离 d 和/或除了平方函数之外的任何递增和/或正函数 incr 可以用于 $L_{\text{parameters}}(w)$ 。

[0245] 在使用反馈回路的情况下,当在推断时使用RNN从输入的3D表示中预测CSG时,示例机器学习过程可能无法访问地面真值中间表示 \widehat{I}_t 。在推断期间, \widehat{I}_t 可能被RNN替换为其推断出的版本:

$$[0246] \quad I_t = m([a_1, \dots, a_t], [x_1, \dots, x_t], [b_2, \dots, b_t]),$$

$$[0247] \quad \text{其中 } a_i \text{ 和 } b_i \text{ 是从 } p_t \text{ 和 } o_t \text{ 采样得到的, 并且 } x_i = g_t^{a_t}(I, I_{t-1}, h_t).$$

[0248] 在示例机器学习过程使用反馈回路的情况下,一旦通过使损失 $L_1(w)$ 最小化学习了RNN,示例机器学习过程就可以使接下来的损失 $\widetilde{L}_1(w)$ 最小化,以便训练RNN将其自己的预测中间表示 I_t 用于其反馈回路。

[0249] $\widetilde{L}_1(w)$ 是与 $L_1(w)$ 完全相同的损失,除了示例机器学习过程将所有地面真值中间

表示

$$[0250] \quad \widehat{I}_t^n = m([\widehat{a}_1^n, \dots, \widehat{a}_t^n], [\widehat{x}_1^n, \dots, \widehat{x}_t^n], [\widehat{b}_2^n, \dots, \widehat{b}_t^n])$$

[0251] 替换为:

$$[0252] \quad \widetilde{I}_t^n = m([\widehat{a}_1^n, \dots, \widehat{a}_t^n], [\widetilde{x}_1^n, \dots, \widetilde{x}_t^n], [\widehat{b}_2^n, \dots, \widehat{b}_t^n]),$$

[0253] 其中, $\widetilde{x}_t^n = g_{\widehat{a}_t^n}(I^n, \widetilde{I}_{t-1}^n, h_t^n)$ 是RNN推断出的参数。

[0254] 这意味着示例机器学习过程将使用RNN预测的推断的连续参数来计算中间表示(但保留CSG树的地面真值结构)。

[0255] 现在, \widetilde{I}_t^n 不再是地面真值固定表示, 而是取决于神经网络的权重(经由 g_1, \dots, g_t , 其取决于权重)。因此, 为了使 $\widetilde{L}_1(w)$ 最小化, 可以使示例机器学习过程能够相对于 x_1, \dots, x_t 来区分 \widetilde{I}_t^n , 即, 区分几何建模器 m 。可以利用有限的差来计算映射 m 的微分。此外, 还可以应用平滑内核来平滑结果。

[0256] 3、递归神经网络的无监督学习

[0257] 已经在不表示真实数据集的合成数据集上对RNN进行了预训练。因此, 示例机器学习过程可以在真实数据集上对其进行微调。然而, 真实数据集通常不与其CSG树相关联。即, 示例机器学习过程提出了一种强化方法在没有监督的情况下微调网络。此外, 这种方法允许网络预测更一致的树, 而预训练步骤则必须与合成数据集匹配, 尽管采用了约简算法, 但是合成数据集可以呈现3D形状非常相似的不同CSG树。

[0258] 还可以使用与此处介绍的微调步骤相同的技术, 以无监督方式直接训练RNN。在这种情况下, 示例机器学习过程可以直接从随机初始化的RNN开始。但是, 收敛要比从预训练的网络开始慢得多。

[0259] 在该无监督学习步骤中, 数据集仅包含 N 个原始网格, 而没有任何标签、CSG信息等。

[0260] 损失可能只是将预测的CSG的渲染与输入网格的渲染进行比较。为了计算预测的CSG, 示例机器学习过程可以根据RNN的分布(以输入为条件)对其进行采样。可以根据稍后在第4节介绍的推断方法的示例但基于当前的权重值来执行采样。

[0261] 示例机器学习过程可以从 $I_0^n = \mathbf{0}$ 开始, 并且从 $p(I^n, I_0^n, h_1^n)$ 中对 a_1^n 进行采样, 从 $o(I^n, I_0^n, h_1^n)$ 中对 b_1^n 进行采样, 并且设置 $I_1^n = m([a_1^n], [x_1^n], [b_2^n])$, 其中 $x_1^n = g_{a_1^n}(I^n, I_0^n, h_1^n)$ 。

[0262] 示例机器学习过程还可以从 $q(I^n, I_0^n, h_1^n)$ 中对结束/继续标记符进行采样。如果对结束标记符进行采样, 则在此处停止。否则, 示例机器学习过程将迭代该采样过程, 直到对结束标记符进行采样, 或者时间步达到最大允许时间步 T , 即最大允许时间步(即, CSG序列的最大长度): 示例机器学习过程采样:

[0263] 从 $p(I^n, I_{t-1}^n, h_t^n)$ 中对 a_t^n 进行采样,

[0264] 从 $o(I^n, I_{t-1}^n, h_t^n)$ 中对 b_t^n 进行采样,

[0265] 设置 $I_t^n = m([a_1^n, \dots, a_t^n], [x_1^n, \dots, x_t^n], [b_2^n, \dots, b_t^n])$, 其中 $x_t^n = g_{a_t^n}(I^n, I_{t-1}^n, h_t^n)$ 。

[0266] 我们通过 T_n 来表示第 n 个输入原始网络的采样 CSG 的深度。

[0267] 有了这些记号, 损失可以很容易地写成

$$[0268] \quad L_2(w) = \sum_{n=1}^N d(I^n, I_{T_n}^n)$$

[0269] 其中 d 是 I^n 和 $I_{T_n}^n$ 之间的相似度测量。例如, 如果示例机器学习过程使用多通道图像或 3D 体素网格作为 3D 输入, 则示例机器学习过程可以使用

$$[0270] \quad L_2 = \sum_{n=1}^N \|I^n - I_{T_n}^n\|_2^2。$$

[0271] 如果示例机器学习过程使用点云, 则示例机器学习过程可以使用倒角距离。

[0272] 该损失相对于连续参数 x_1^n, \dots, x_t^n 是可区分的。因此, 使该损失最小化仅允许学习 (或如果已经对 RNN 进行了预训练, 则进行微调) 与那些连续参数有关的神经网络的权重, 但是无法学习离散分布 p, o, q 。实际上, 离散分布 (相对于网络的权重是可区分的) 用于推断离散变量 (基元类型、布尔运算、CSG 长度), 并且损失相对于离散变量是无法区分的。

[0273] 为了也微调这些分布, 示例机器学习过程可以反向传播损失相对于这些分布而言的梯度。示例机器学习过程可以使用强化算法来获得离散分布 p, o, q 的这种梯度。强化算法限于这些离散分布, 并且因此, 无监督学习可以获得其它变量的真实梯度。这限制了近似, 从而影响精度, 并且这允许优化分布, 否则将不会更新分布, 并且因此不会学习。

[0274] 强化算法可以是以下论文中描述的 REINFORCE 算法:

[0275] -R.J.Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 229—256, 1992 年。

[0276] 这提供了以下梯度公式:

$$[0277] \quad \frac{\partial L_2}{\partial p^y} = \frac{L_2 - \beta}{p^y} 1_{t \leq T_i} 1_{y=a_t^i}$$

$$[0278] \quad \frac{\partial L_2}{\partial o^y} = \frac{L_2 - \beta}{o^y} 1_{t \leq T_i} 1_{y=b_t^i}$$

$$[0279] \quad \frac{\partial L_2}{\partial q} = -\frac{L_2 - \beta - \rho}{q} 1_{t=T_i}$$

[0280] 其中 ρ 是鼓励 CSG 树针对最小尺寸具有最大表现力的额外稀疏奖励, 并且 β 是用于减少估计梯度的方差的基准。

[0281] 通常 ρ 可以近似等于在用于预训练网络的监督数据集上计算出的 L_2 损失。

[0282] 令 w_0 是无监督学习之前的权重 (w_0 来自随机初始化或来自监督预训练步骤)。然后, 可以将 β 选为常量, 例如, 如果网络已经被预训练, 则为 $\frac{1}{2} L_2(w_0)$, 或者如果我们是从小权重开始 w_0 的, 则为 $\frac{1}{10} L_2(w_0)$ 。或者, 示例机器学习过程也可以将 β 设置为训练期间的 L_2 损

失的移动平均值。

[0283] 要注意的是,在无监督学习场景中,出于与先前的部分(具有反馈回路的监督学习)相同的原因,还可以使示例机器学习过程能够区分映射 m 。区别在于,在无监督场景中,即使示例机器学习过程不使用反馈回路,示例机器学习过程也可以区分建模器,因为损失明确地取决于最终的CSG表示 $I_{T_n}^n$ 。

[0284] 4、推断和最终细化

[0285] 一旦对网络进行了训练,示例机器学习过程就可以使用其来根据原始输入网格推断出最可能的合理的CSG树。为此,示例机器学习过程可以应用具有宽度 k 的波束搜索算法,因为贪婪的搜索对于深层的CSG树是禁止的。

[0286] 在第一步 $t=1$ 处,示例机器学习过程可以对最可能的 k 个CSG序列进行采样。然后,在每个接下来的步处,对于每个当前序列,示例机器学习过程可以计算 k 个最可能的运算和基元。在每个步处,这都给我们 k^2 个CSG序列,其中示例性机器学习过程可能只保持 k 个最可能的CSG序列。它们是基于新的潜在序列的概率来选择的,新的潜在序列的概率是当前序列的概率乘以基元的概率乘以运算的概率再乘以结束标记符的概率。

[0287] 最后,示例机器学习过程获得了 k 个合理的CSG序列。对于每个序列,示例机器学习过程可以细化该序列所有基元的连续参数,例如通过使用可微分的优化(例如BFGS)使损失 L_2 最小化。最后,示例机器学习过程可以只保留使该损失最小化的CSG树,或者示例机器学习过程可以向用户提出所有 k 个CSG树。

[0288] 图9-11示出了基于与示例机器学习过程一致地学习的神经网络对示例机器学习过程的推断方法的应用的测试。图9示出了表示机械工具的原始输入网格。图10示出了在细化之前(在将几何建模器应用于推断出的可编辑特征树之后)应用推断方法的结果。图11示出了细化后的结果。

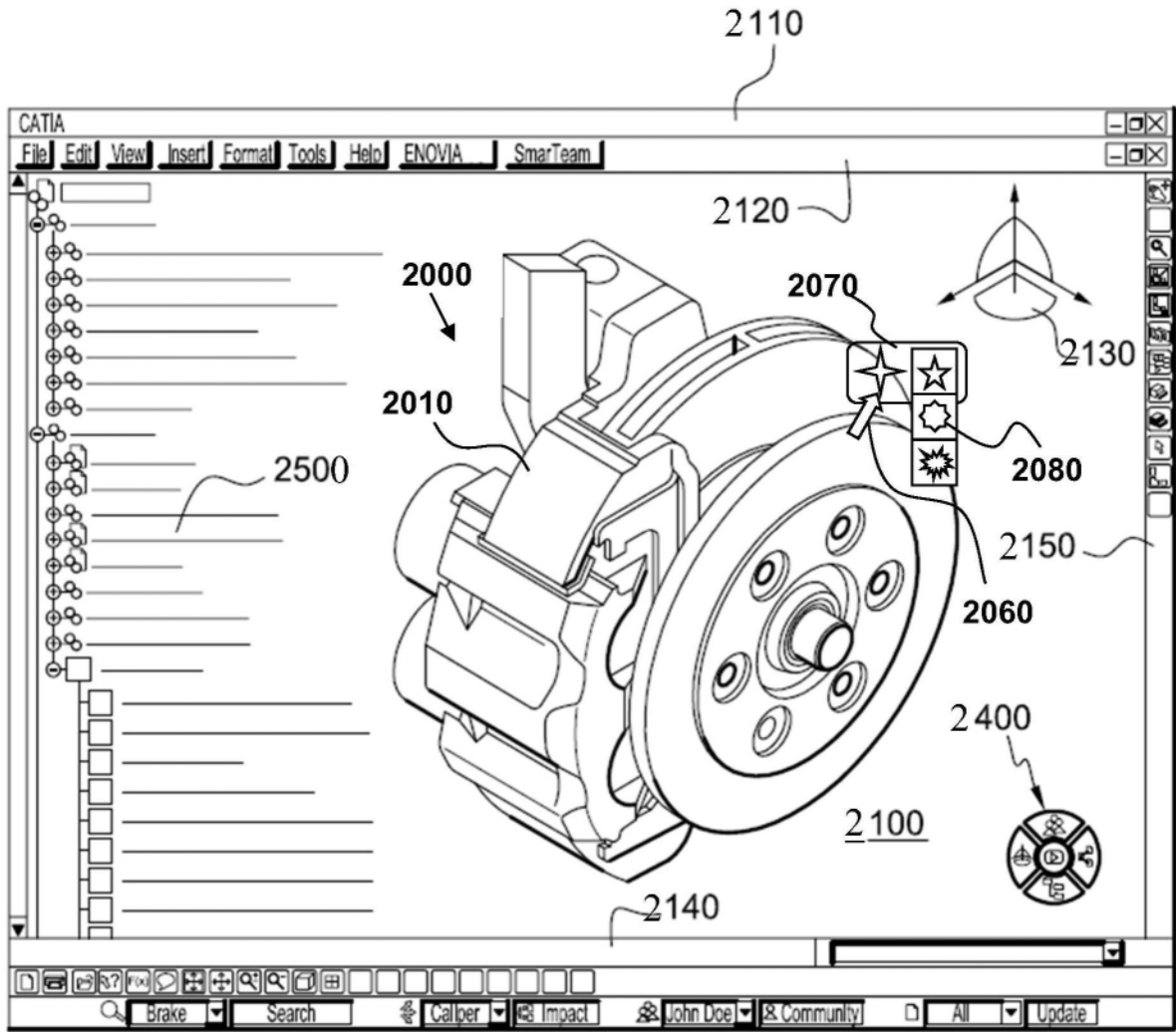


图1

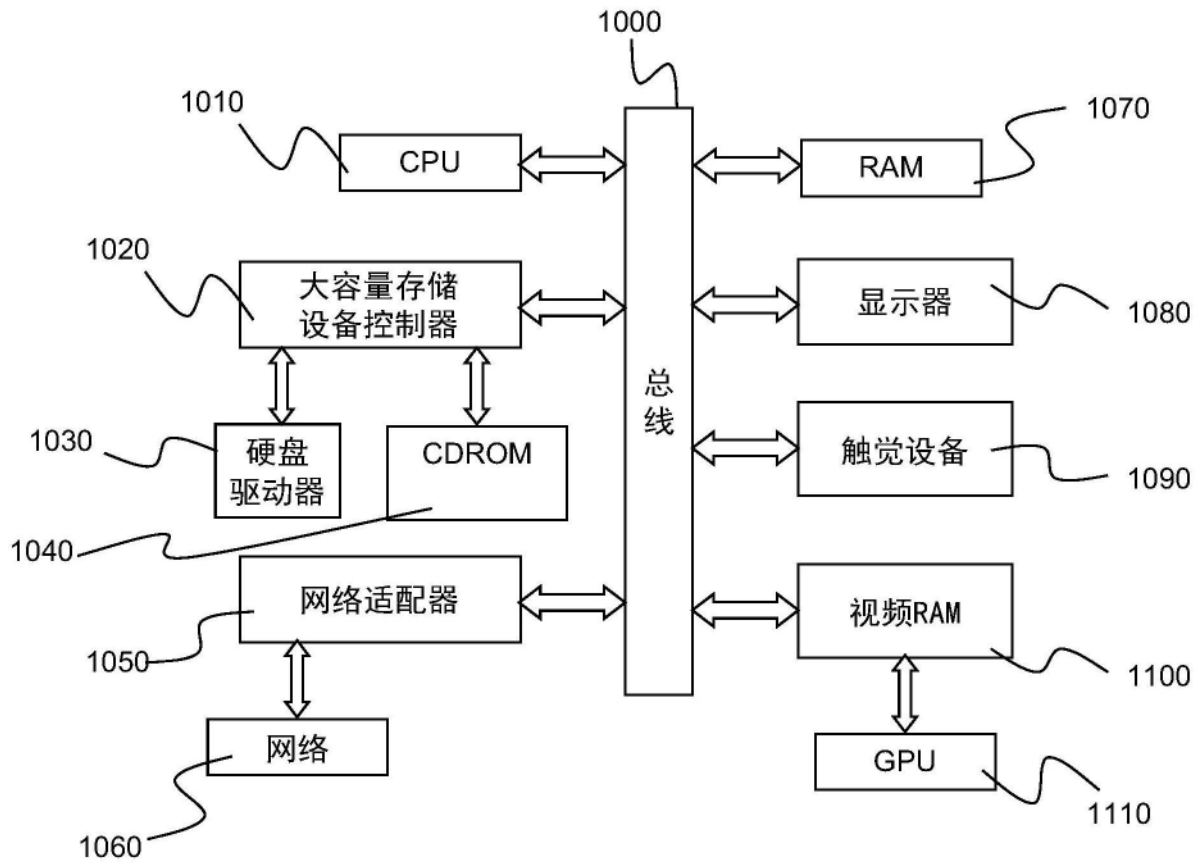


图2

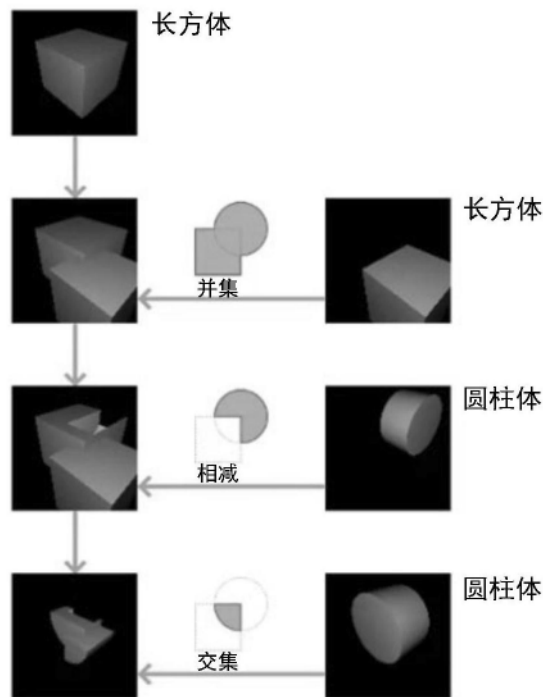


图3

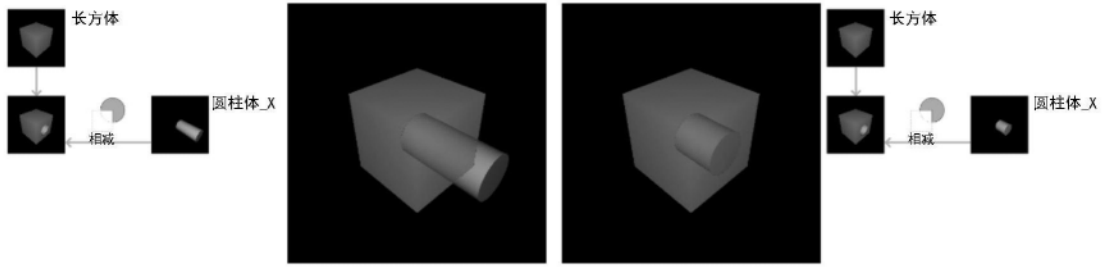


图4

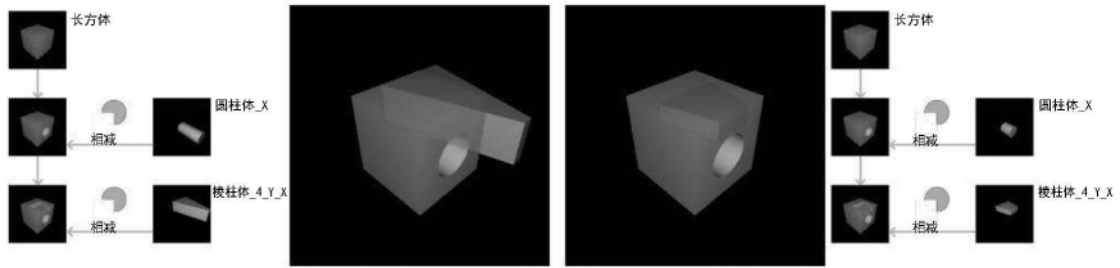


图5

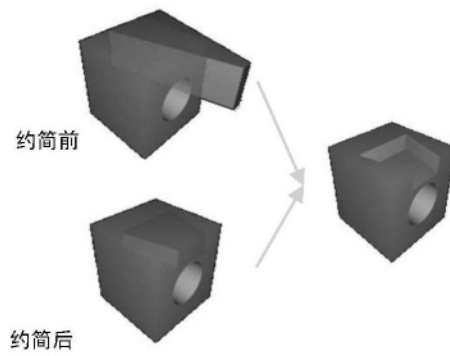


图6

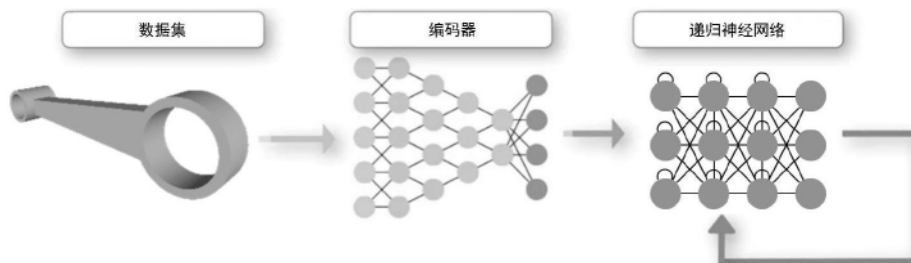


图7

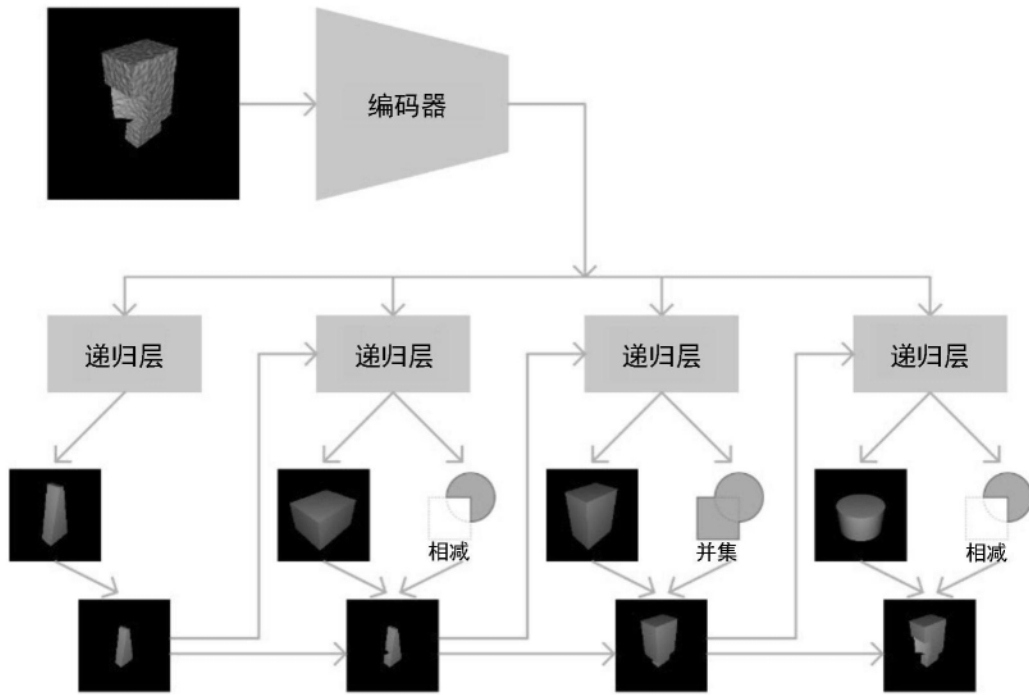


图8



图9

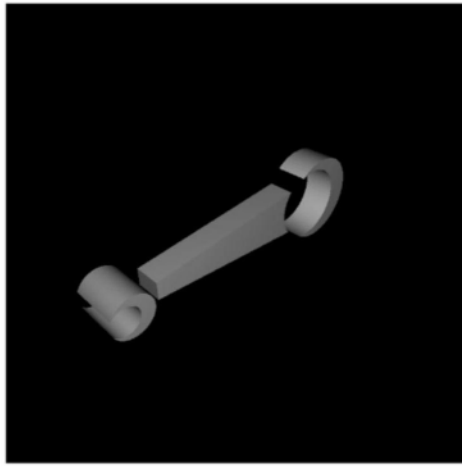


图10

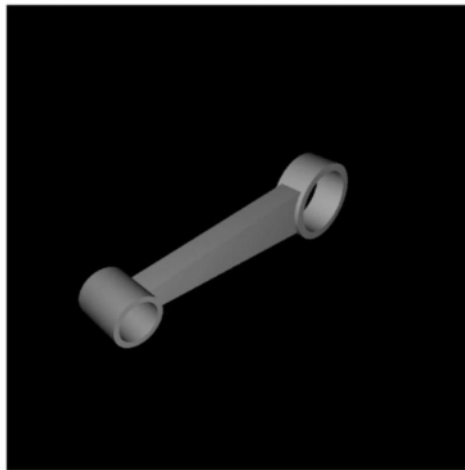


图11