



(12)发明专利申请

(10)申请公布号 CN 111080671 A

(43)申请公布日 2020.04.28

(21)申请号 201911378607.2

(22)申请日 2019.12.27

(71)申请人 深圳大学

地址 518060 广东省深圳市南山区南海大道3688号

(72)发明人 胡瑞珍 黄惠 闫子豪

(74)专利代理机构 深圳市君胜知识产权代理事务所(普通合伙) 44268

代理人 王永文

(51)Int.Cl.

G06T 7/20(2017.01)

G06N 3/08(2006.01)

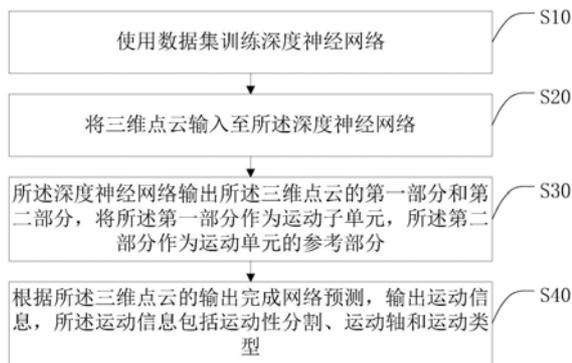
权利要求书3页 说明书13页 附图6页

(54)发明名称

一种基于深度神经网络的运动预测方法和智能终端

(57)摘要

本发明公开了一种基于深度神经网络的运动预测方法和智能终端,所述方法包括:使用数据集训练深度神经网络;将三维点云输入至所述深度神经网络;所述深度神经网络输出所述三维点云的第一部分和第二部分,将所述第一部分作为运动子单元,所述第二部分作为运动单元的参考部分;根据所述三维点云的输出完成网络预测,输出运动信息,所述运动信息包括运动性分割、运动轴和运动类型。本发明实现了在非结构化并且可能是部分扫描的各种铰链式物体在静止状态下同时运动和部件的预测结果,能够十分准确地预测物体部件的运动。



1. 一种基于深度神经网络的运动预测方法,其特征在于,所述基于深度神经网络的运动预测方法包括:

使用数据集训练深度神经网络;

将三维点云输入至所述深度神经网络;

所述深度神经网络输出所述三维点云的第一部分和第二部分,将所述第一部分作为运动子单元,所述第二部分作为运动单元的参考部分;

根据所述三维点云的输出完成网络预测,输出运动信息,所述运动信息包括运动性分割、运动轴和运动类型。

2. 根据权利要求1所述的基于深度神经网络的运动预测方法,其特征在于,在训练所述深度神经网络时,所使用的损失函数为:

$$L(\mathcal{D}_t, \mathcal{S}, \mathcal{M}) = \sum_{t=1}^n (L_{\text{rec}}(\mathcal{D}_t) + L_{\text{disp}}(\mathcal{D}_t)) + L_{\text{seg}}(\mathcal{S}) + L_{\text{mob}}(\mathcal{M});$$

其中, \mathcal{D}_t 表示位移图, \mathcal{S} 表示分割, \mathcal{M} 表示拟合运动参数, L_{rec} 是重建误差, L_{disp} 是位移误差, L_{seg} 是分割误差, L_{mob} 是运动参数的回归误差;

重建误差表示形状的扭曲程度,位移误差表示运动部分的精确度,分割误差和回归误差则刻画了运动信息的正确程度,包括对运动与不动的划分、运动轴的位置、方向和运动类型。

3. 根据权利要求2所述的基于深度神经网络的运动预测方法,其特征在于, L_{rec} 刻画了预测的运动后点云与真实的运动后点云之间的几何误差;

将点云 \mathcal{P}_0 分成参照部分和运动部分,在经历过运动后 $\mathcal{P}_t = \mathcal{P}_{t-1} + \mathcal{D}_t$ 后,参照部分保持静止,运动部分为刚性运动,其中, \mathcal{P}_{t-1} 和 \mathcal{P}_t 表示两个邻接的点云帧,因此 L_{rec} 分为两部分:

$$L_{\text{rec}}(\mathcal{D}_t) = L_{\text{ref}}(\mathcal{P}_t) + L_{\text{mov}}(\mathcal{P}_t);$$

$L_{\text{ref}}(\mathcal{P}_t)$ 是参照部分的误差, $L_{\text{mov}}(\mathcal{P}_t)$ 是运动部分的误差;

$L_{\text{ref}}(\mathcal{P}_t)$ 是每个点误差距离的平方和:

$$L_{\text{ref}}(\mathcal{P}_t) = \sum_{p \in \mathcal{P}_{\text{ref}}^t} \|p - p^{\text{gt}}\|^2;$$

其中, p^{gt} 是点 p 真实的位置;

$L_{\text{mov}}(\mathcal{P}_t)$ 的构成为:

$$L_{\text{mov}}(\mathcal{P}_t) = L_{\text{shape}}(\mathcal{P}_{\text{mov}}^t, \mathcal{P}_{\text{mov}}^{t, \text{gt}}) + L_{\text{density}}(\mathcal{P}_{\text{mov}}^t, \mathcal{P}_{\text{mov}}^{t, \text{gt}});$$

其中, L_{shape} 是用来惩罚不与目标形状匹配的点, L_{density} 是预测点云与目标点云局部点密度, $\mathcal{P}_{\text{mov}}^t$ 指所述深度神经网络生成的第 t 帧的点云中的运动部分, $\mathcal{P}_{\text{mov}}^{t, \text{gt}}$ 指正确的第 t 帧的点云中的运动部分, gt 是 ground truth 的缩写,表示正确的意思。

4. 根据权利要求3所述的基于深度神经网络的运动预测方法,其特征在于,通过误差损失函数预测运动信息与目标运动信息之间的差别;所述运动类型包括旋转运动和平移运动。

5. 根据权利要求4所述的基于深度神经网络的运动预测方法,其特征在于,对于旋转运动,损失函数如下:

$$L_{\text{disp}}^R(\mathcal{D}_t) = L_{\text{dir}}^R(\mathcal{D}_t) + L_{\text{ang}}^R(\mathcal{D}_t) + L_{\text{len}}^R(\mathcal{D}_t);$$

$L_{\text{dir}}^R(\mathcal{D}_t)$ 刻画了预测的位移是否垂直于真实的运动轴,具体的计算公式为:

$$L_{\text{dir}}^R(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} |\text{dot}(\mathcal{D}_t(p)/\|\mathcal{D}_t(p)\|_2, d_{\text{gt}})|$$

其中, dot表示点乘, $\mathcal{D}_t(p)$ 表示第t帧点云p的位移图, d_{gt} 是正确的运动轴的方向; $L_{\text{ang}}^R(\mathcal{D}_t)$ 则是各个点旋转角的偏差,所有点旋转角度一致,具体计算公式为:

$$L_{\text{ang}}^R(\mathcal{D}_t) = \sigma^2 \{\text{dot}(\text{proj}(p), \text{proj}(\mathcal{D}_t(p) + p))\}_{\forall p \in \mathcal{P}_{\text{mov}}^{t-1}};$$

其中, σ 为常数, $\text{proj}(p)$ 表示p点与将点p投影到正确运动轴上的投影点的距离, $\text{proj}(p) = (p - \pi(p)) / \|p - \pi(p)\|_2$;

$L_{\text{len}}^R(\mathcal{D}_t)$ 要求每个点旋转前和旋转后距离真实转轴的距离相同,约束其运动的圆周性,具体计算公式为:

$$L_{\text{len}}^R(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} \left| \|p - \pi(p)\|_2 - \|(\mathcal{D}_t(p) + p) - \pi(\mathcal{D}_t(p) + p)\|_2 \right|$$

6. 根据权利要求5所述的基于深度神经网络的运动预测方法,其特征在于,对于平移运动,损失函数如下:

$$L_{\text{disp}}^T(\mathcal{D}_t) = L_{\text{dir}}^T(\mathcal{D}_t) + L_{\text{len}}^T(\mathcal{D}_t);$$

$L_{\text{dir}}^T(\mathcal{D}_t)$ 刻画了预测的位移是否平行于真实的运动轴,具体计算公式为:

$$L_{\text{dir}}^T(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} 1 - |\text{dot}(\mathcal{D}_t(p)/\|\mathcal{D}_t(p)\|_2, d_{\text{gt}})|$$

$L_{\text{len}}^T(\mathcal{D}_t)$ 则要求每个点移动的距离相同,方差为0,具体计算公式为:

$$L_{\text{len}}^T(\mathcal{D}_t) = \sigma^2 \left(\|\|\mathcal{D}_t(p)\|_2\|_{\forall p \in \mathcal{P}_{\text{mov}}^{t-1}} \right)$$

7. 根据权利要求6所述的基于深度神经网络的运动预测方法,其特征在于,运动信息损失函数为:

$$L_{\text{mob}}(\mathcal{M}) = \|d - d^{\text{gt}}\|_2 + \|x - x^{\text{gt}}\|_2 + H(t, t^{\text{gt}});$$

其中, d 、 x 和 t 分别为运动轴方向、运动轴位置和运动类型, d^{gt} 是正确的运动轴方向, x^{gt} 是正确的运动轴位置, t^{gt} 是正确的运动类型, H 是交叉熵。

8. 根据权利要求1所述的基于深度神经网络的运动预测方法,其特征在于,所述三维点云的点数为1024个。

9. 一种智能终端,其特征在于,所述智能终端包括:存储器、处理器及存储在所述存储器上并可在所述处理器上运行的基于深度神经网络的运动预测程序,所述基于深度神经网络的运动预测程序被所述处理器执行时实现如权利要求1-8任一项所述的基于深度神经网络

络的运动预测方法的步骤。

10. 一种存储介质,其特征在于,所述存储介质存储有基于深度神经网络的运动预测程序,所述基于深度神经网络的运动预测程序被处理器执行时实现如权利要求1-8任一项所述基于深度神经网络的运动预测方法的步骤。

一种基于深度神经网络的运动预测方法和智能终端

技术领域

[0001] 本发明涉及深度学习技术领域,尤其涉及一种基于深度神经网络的运动预测方法、智能终端及存储介质。

背景技术

[0002] 近年来,计算机图形学以及相关领域例如计算机视觉和机器人领域已经聚焦于推断三维物体和它们部件的可能运动,因为这个问题对于理解物体直观功能性(affordances)以及功能性等问题密切相关。该问题比较难解决的是当仅仅给定一个三维物体的几个静止状态,机器是否以及如何能够学习来预测部件运动或者部件可运动性。

[0003] 已有方法提出根据物体运动来获得并且重建,表示并且理解物体运动,甚至根据静止物体预测部件运动,这些工作背后的动机是更全面地理解物体的运动有助于图形学应用,例如,动画,物体位姿修正和重建,以及机器人应用,如3D场景中人机交互的建模。

[0004] 在机器人领域,大量工作聚焦于功能可见性预测的问题,它们的目标是识别出物体中能进行特定交互的区域,例如,抓或者推。最近应用到深度神经网络来标记功能可见性标签的图像,或者物理模拟来得到与功能可见性密切相关的人类效用。功能可见性分析的更通用的方法是基于人体位姿假设的想法,预测拟合给定场景上下文的最佳人体位姿以辅助理解场景。基于人与物体的交互,人体位姿假设也能用于预测物体的功能类别。与功能可见性和人体位姿分析密切相关的是活动识别,其中一个例子是在输入场景中检测活动区域,这些区域支持具体类别的人类活动,例如吃饭或者看电视。尽管功能可见性检测识别能进行特定运动类型的区域,例如转动或者滑动;预测的运动仅仅用标签来描述,并且局限于和人的交互。因此,他们不能表示一个物体的一般运动。功能可见性分析的更通用的方法的焦点是以高层次来理解与特定物体交互的动作或者是在给定场景下的动作,然而这些方法不能检测或者建模与这些动作相关的具体运动或者部件运动。

[0005] 在计算机视觉中,已提出基于当前物体的描述,推测未来物体的状态的方法,这些方法隐式预测图像中的物体正进行的运动以及未来的运动。通用的解决方法是利用在视频数据上训练生成对抗网络(GANs)来生成输入图像的后续帧。另一方面,将视频分解成内容和运动组件,之后根据选定的内容和运动,将分解得到的内容和运动组件创造视频的后续帧。

[0006] 计算机图形学的工作也有针对三维物体进行运动推断的问题。通过从部件的几何部署预测机械部件和整个组件的可能运动来说明机械组件的运动。例如从概念草图创建图表动画。对于更一般的形状,引入了互动地形(interaction landscapes),它是物体以某种方式被使用的动作表示,例如,杯子被人用来喝水。然后,这种表示可以用于将运动分类为不同类型的交互并且还用于预测物体在其运动的几秒内支持的交互。例如使用一个称为运动树的结构,得到了场景中的物体的相对运动。结构树是根据在不同的几何配置中找到物体的不同实例推断得到的。当给定一个部件分割好的三维物体,基于从每个物体的少量静止运动状态的物体数据集学习得到的模型来预测物体部件可能进行的运动以及运动参数。

这个模型有效地将物体的几何关联到它的可能运动。从两个未分割的功能性相似的实例或者运动一样但在不同的运动状态的物体来预测物体的部件进行的可能运动。虽然能够推断场景中物体的运动,但是它受限于要得到场景中出现的多个物体实例这个假设。数据驱动方法缺点是需要物体被很好地分割。有的不足是设计的网络的输入需要一对运动状态一样而旋转角度大小不同的物体作为输入。当需要在三维场景中直接获得功能性预测,例如,在机器人导航中,希望要么预分割的物体,要么旋转的物体对,这都是不现实的。

[0007] 因此,现有技术还有待于改进和发展。

发明内容

[0008] 本发明针对现有技术的上述缺陷,本发明提供一种基于深度神经网络的运动预测方法、智能终端及存储介质。

[0009] 本发明解决技术问题所采用的技术方案如下:

[0010] 一种基于深度神经网络的运动预测方法,其中,所述基于深度神经网络的运动预测方法包括:

[0011] 使用数据集训练深度神经网络;

[0012] 将三维点云输入至所述深度神经网络;

[0013] 所述深度神经网络输出所述三维点云的第一部分和第二部分,将所述第一部分作为运动子单元,所述第二部分作为运动单元的参考部分;

[0014] 根据所述三维点云的输出完成网络预测,输出运动信息,所述运动信息包括运动性分割、运动轴和运动类型。

[0015] 所述的基于深度神经网络的运动预测方法,其中,在训练所述深度神经网络时,所使用的损失函数为:

$$[0016] \quad L(\mathcal{D}_t, \mathcal{S}, \mathcal{M}) = \sum_{t=1}^n (L_{\text{rec}}(\mathcal{D}_t) + L_{\text{disp}}(\mathcal{D}_t)) + L_{\text{seg}}(\mathcal{S}) + L_{\text{mob}}(\mathcal{M});$$

[0017] 其中, \mathcal{D}_t 表示位移图, \mathcal{S} 表示分割, \mathcal{M} 表示拟合运动参数, L_{rec} 是重建误差, L_{disp} 是位移误差, L_{seg} 是分割误差, L_{mob} 是运动参数的回归误差;

[0018] 重建误差表示形状的扭曲程度,位移误差表示运动部分的精确度,分割误差和回归误差则刻画了运动信息的正确程度,包括对运动与不动的划分、运动轴的位置、方向和运动类型。

[0019] 所述的基于深度神经网络的运动预测方法,其中, L_{rec} 刻画了预测的运动后点云与真实的运动后点云之间的几何误差;

[0020] 将点云 \mathcal{P}_0 分成参照部分和运动部分,在经历过运动后 $\mathcal{P}_t = \mathcal{P}_{t-1} + \mathcal{D}_t$ 后,参照部分保持静止,运动部分为刚性运动,其中, \mathcal{P}_{t-1} 和 \mathcal{P}_t 表示两个邻接的点云帧,因此 L_{rec} 分为两部分:

$$[0021] \quad L_{\text{rec}}(\mathcal{D}_t) = L_{\text{ref}}(\mathcal{P}_t) + L_{\text{mov}}(\mathcal{P}_t);$$

[0022] $L_{\text{ref}}(\mathcal{P}_t)$ 是参照部分的误差, $L_{\text{mov}}(\mathcal{P}_t)$ 是运动部分的误差;

[0023] $L_{\text{ref}}(\mathcal{P}_t)$ 是每个点误差距离的平方和:

$$[0024] \quad L_{\text{ref}}(\mathcal{P}_t) = \sum_{p \in \mathcal{P}_{\text{ref}}^t} \|p - p^{\text{gt}}\| ;$$

[0025] 其中, p^{gt} 是点 p 真实的位置;

[0026] $L_{\text{mov}}(\mathcal{P}_t)$ 的构成为:

$$[0027] \quad L_{\text{mov}}(\mathcal{P}_t) = L_{\text{shape}}(\mathcal{P}_{\text{mov}}^t, \mathcal{P}_{\text{mov}}^{t, \text{gt}}) + L_{\text{density}}(\mathcal{P}_{\text{mov}}^t, \mathcal{P}_{\text{mov}}^{t, \text{gt}});$$

[0028] 其中, L_{shape} 是用来惩罚不与目标形状匹配的点的, L_{density} 是预测点云与目标点云局部点密度, $\mathcal{P}_{\text{mov}}^t$ 指所述深度神经网络生成的第 t 帧的点云中的运动部分, $\mathcal{P}_{\text{mov}}^{t, \text{gt}}$ 指正确的第 t 帧的点云中的运动部分, gt 是 ground truth 的缩写, 表示正确的意思。

[0029] 所述的基于深度神经网络的运动预测方法, 其中, 通过误差损失函数预测运动信息与目标运动信息之间的差别; 所述运动类型包括旋转运动和平移运动。

[0030] 所述的基于深度神经网络的运动预测方法, 其中, 对于旋转运动, 损失函数如下:

$$[0031] \quad L_{\text{disp}}^R(\mathcal{D}_t) = L_{\text{dir}}^R(\mathcal{D}_t) + L_{\text{ang}}^R(\mathcal{D}_t) + L_{\text{len}}^R(\mathcal{D}_t);$$

[0032] 其中, dot 表示点乘, $\mathcal{D}_t(p)$ 表示第 t 帧点云 p 的位移图, d_{gt} 是正确的运动轴的方向; $L_{\text{dir}}^R(\mathcal{D}_t)$ 刻画了预测的位移是否垂直于真实的运动轴, 具体的计算公式为:

$$[0033] \quad L_{\text{dir}}^R(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} |\text{dot}(\mathcal{D}_t(p) / \|\mathcal{D}_t(p)\|_2, d_{\text{gt}})| ;$$

[0034] $L_{\text{ang}}^R(\mathcal{D}_t)$ 则是各个点旋转角的偏差, 所有点旋转角度一致, 具体计算公式为:

$$[0035] \quad L_{\text{ang}}^R(\mathcal{D}_t) = \sigma^2 \{ \text{dot}(\text{proj}(p), \text{proj}(\mathcal{D}_t(p) + p)) \}_{\forall p \in \mathcal{P}_{\text{mov}}^{t-1}} ;$$

[0036] 其中, σ 为常数, $\text{proj}(p)$ 表示 p 点与将点 p 投影到正确运动轴上的投影点的距离, $\text{proj}(p) = (p - \pi(p)) / \|p - \pi(p)\|_2$;

[0037] $L_{\text{len}}^R(\mathcal{D}_t)$ 要求每个点旋转前和旋转后距离真实转轴的距离相同, 约束其运动的圆周性, 具体计算公式为:

$$[0038] \quad L_{\text{len}}^R(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} | \|p - \pi(p)\|_2 - \|(\mathcal{D}_t(p) + p) - \pi(\mathcal{D}_t(p) + p)\|_2 |$$

[0039] 所述的基于深度神经网络的运动预测方法, 其中, 对于平移运动, 损失函数如下:

$$[0040] \quad L_{\text{disp}}^T(\mathcal{D}_t) = L_{\text{dir}}^T(\mathcal{D}_t) + L_{\text{len}}^T(\mathcal{D}_t);$$

[0041] $L_{\text{dir}}^T(\mathcal{D}_t)$ 刻画了预测的位移是否平行于真实的运动轴, 具体计算公式为:

$$[0042] \quad L_{\text{dir}}^T(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} 1 - |\text{dot}(\mathcal{D}_t(p) / \|\mathcal{D}_t(p)\|_2, d_{\text{gt}})| ;$$

[0043] $L_{\text{len}}^T(\mathcal{D}_t)$ 则要求每个点移动的距离相同, 方差为 0, 具体计算公式为:

$$[0044] \quad L_{\text{len}}^T(\mathcal{D}_t) = \sigma^2 \left(\|\mathcal{D}_t(p)\|_2 \right)_{\forall p \in \mathcal{P}_{\text{mov}}^{t-1}} ;$$

[0045] 所述的基于深度神经网络的运动预测方法, 其中, 运动信息损失函数为:

$$[0046] \quad L_{\text{mob}}(\mathcal{M}) = \|d - d^{\text{gt}}\|_2 + \|x - x^{\text{gt}}\|_2 + H(t, t^{\text{gt}});$$

[0047] 其中, d 、 x 和 t 分别为运动轴方向、运动轴位置和运动类型, d^{gt} 是正确的运动轴方向, x^{gt} 是正确的运动轴位置, t^{gt} 是正确的运动类型, H 是交叉熵。

[0048] 所述的基于深度神经网络的运动预测方法, 其中, 所述三维点云的点数为1024个。

[0049] 一种智能终端, 其中, 所述智能终端包括如上所述的基于深度神经网络的运动预测系统, 还包括: 存储器、处理器及存储在所述存储器上并可在所述处理器上运行的基于深度神经网络的运动预测程序, 所述基于深度神经网络的运动预测程序被所述处理器执行时实现如上所述的基于深度神经网络的运动预测方法的步骤。

[0050] 一种存储介质, 其中, 所述存储介质存储有基于深度神经网络的运动预测程序, 所述基于深度神经网络的运动预测程序被处理器执行时实现如上所述基于深度神经网络的运动预测方法的步骤。

[0051] 本发明使用数据集训练深度神经网络; 将三维点云输入至所述深度神经网络; 所述深度神经网络输出所述三维点云的第一部分和第二部分, 将所述第一部分作为运动子单元, 所述第二部分作为运动单元的参考部分; 根据所述三维点云的输出完成网络预测, 输出运动信息, 所述运动信息包括运动性分割、运动轴和运动类型。本发明实现了在非结构化并且可能是部分扫描的各种铰链式物体在静止状态下同时运动和部件的预测结果, 能够十分准确地预测物体部件的运动。

附图说明

[0052] 图1是本发明基于深度神经网络的运动预测方法的较佳实施例的流程图;

[0053] 图2是本发明基于深度神经网络的运动预测方法的较佳实施例中深度神经网络从一个训练集学习深度预测模型, 该训练集涵盖了不同物体的各种运动的示意图;

[0054] 图3是本发明基于深度神经网络的运动预测方法的较佳实施例中长短期记忆网络的结构示意图;

[0055] 图4是本发明基于深度神经网络的运动预测方法的较佳实施例中运动类型为旋转运动的示意图;

[0056] 图5是本发明基于深度神经网络的运动预测方法的较佳实施例中运动类型为平移运动的示意图;

[0057] 图6是本发明基于深度神经网络的运动预测方法的较佳实施例中在完整和部分扫描的多种形状的不同运动进行运动和部件预测结果集的示意图;

[0058] 图7是本发明基于深度神经网络的运动预测方法的较佳实施例中预测课桌的并行运动的示意图;

[0059] 图8是本发明基于深度神经网络的运动预测方法的较佳实施例中基准预测网络“BaseNet”架构的示意图;

[0060] 图9是本发明基于深度神经网络的运动预测方法的较佳实施例中MAPP-NET与BaseNet之间的可视化对比的示意图;

[0061] 图10是本发明基于深度神经网络的运动预测方法的较佳实施例中与没有重建损失项 L_{rec} 或者位移损失项 L_{disp} 得到的预测的可视化对比的示意图;

[0062] 图11是本发明基于深度神经网络的运动预测方法的较佳实施例中运动参数和分割不是通过网络预测得到的结果的可视化对比的示意图；

[0063] 图12为本发明智能终端的较佳实施例的运行环境示意图。

具体实施方式

[0064] 为使本发明的目的、技术方案及优点更加清楚、明确，以下参照附图并举实施例对本发明进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。基于本发明中的实施例，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他实施例，都属于本发明保护的范围。

[0065] 本发明较佳实施例所述的基于深度神经网络的运动预测方法，如图1所示，一种基于深度神经网络的运动预测方法，其中，所述基于深度神经网络的运动预测方法包括以下步骤：

[0066] 步骤S10、使用数据集训练深度神经网络；

[0067] 步骤S20、将三维点云输入至所述深度神经网络；

[0068] 步骤S30、所述深度神经网络输出所述三维点云的第一部分和第二部分，将所述第一部分作为运动子单元，所述第二部分作为运动单元的参考部分；

[0069] 步骤S40、根据所述三维点云的输出完成网络预测，输出运动信息，所述运动信息包括运动性分割、运动轴和运动类型。

[0070] 本发明引入了一个基于学习的方法，它同时预测单一未分割的点云，可能是一个三维物体的部分扫描形状的旋转部件以及它们的运动。本发明的深度神经网络，将输入的三维物体视为一个运动单元，并且输出点云的两个部分，将其中一个部分作为运动子单元而另一个部分作为运动单元的参考部分，迭代地应用本发明提出的网络得到的部分能够预测更精细的部件运动，从而得到层次性运动同时基于运动的物体分割的预测，如图2所示。MAPP-NET(深度神经网络)从一个训练集学习深度预测模型，该训练集涵盖了不同物体的各种运动。尽管从单一的配置中进行运动性预测与分割的问题本质上是不适定的，本发明的基于学习的方法能够汇聚丰富的线索，例如从训练数据中得到部件几何以及它们的上下文场景，从而推测没有见过的三维物体。

[0071] 点云的运动性预测的核心要点能被视为是预测点对以及随着时间变化的位移场，它允许网络处理非结构化的低层次的输入并且利用运动的瞬时性特点，具体来说，本发明的MAPP-NET通过循环神经网络来实现，它的输入是点云，然后预测在后续帧中每个点的位移情况，而接下来的每一帧的输入点云是参考点。网络的架构由编码器-解码器对组成同时交叉着长短时记忆网络(LSTM, Long Short-Term Memory)，它同时预测输入点云的位移场；本发明还在网络中加入了额外的层来推断基于运动性的分割以及预测得到的位移场的运动参数。因此，给定一个点云，MAPP-NET既推断点的几何变换的运动类型和运动参数(如旋转轴)，又根据预测的运动状态预测他们的可旋转的部分的分割。

[0072] 本发明的目的是分割出给定三维物体中可运动的部分，判断物体运动类型，并生成物体下几帧的运动序列。其中物体是用单个，未经分割的点云来表示的。本发明采用深度神经网络在数据集上预训练来达到上述目标。因此本发明的主要技术问题为如何设计网络结构和损失函数以完成上述任务。

[0073] 本发明输入是点数为1024的三维点云,假设该点云只有一个运动单元,即点云的点要么是参照不动的,要么同属一个运动。输出为一个点云序列,序列中的每个点云都有1024个点并与输入点云中的点一一对应。同时网络还预测输出了运动性分割S,运动轴(d, x)、和运动类型t。运动轴信息包含轴的方向d和轴上一点x的位置,把它们统称为运动信息 $M = (t, d, x)$ 。

[0074] 网络的核心是用一个循环神经网络去预测点云中点的位移,该位移即为运动的表示。采用循环神经网络是因为这种网络在处理序列数据时有不错的效果。更具体的,本发明采用的是长短期记忆网络,并运用了PointNet++中的网络结构集合抽象层SA与特征传递层FP。图3具体说明了网络的结构,输入的点云 P_0 在经过一个集合抽象层之后便进入循环神经网络,它包含了若干子网络,子网络由一个特征传递层和全连接层构成,每个子网络输出某一帧的运动预测,即位移D。之后与输入点云相加便得到了运动后若干帧的点云P。有了这些点云和位移,在经过一些层就能分析出分割、运动信息。将若干帧位移信息一起传入一个全连接层后便能得到该点云的分割。运动信息也可以通过类似的方法分别得到,不过传入的信息是运动后的若干帧点云信息而不是位移,并且由于要整体考虑,需要在全连接层前加入一个集合抽象层。之所以用点云而不用位移是因为在实验中发现前者能有更高的精度,具体结构可见图3

[0075] 网络训练和损失函数

[0076] 为了训练上述多重输出网络,本发明设计了如下的损失函数:

$$[0077] \quad L(\mathcal{D}_t, \mathcal{S}, \mathcal{M}) = \sum_{t=1}^n (L_{\text{rec}}(\mathcal{D}_t) + L_{\text{disp}}(\mathcal{D}_t)) + L_{\text{seg}}(\mathcal{S}) + L_{\text{mob}}(\mathcal{M});$$

[0078] 其中, \mathcal{D}_t 表示位移图,S表示分割,M表示拟合运动参数, L_{rec} 是重建误差, L_{disp} 是位移误差, L_{seg} 是分割误差, L_{mob} 是运动参数的回归误差;

[0079] 重建误差表示形状的扭曲程度,位移误差表示运动部分的精确度,分割误差和回归误差则刻画了运动信息的正确程度,包括对运动与不动的划分、运动轴的位置、方向和运动类型。

[0080] 重建损失函数, L_{rec} 刻画了预测的运动后点云与真实的运动后点云之间的几何误差;

[0081] 将点云 P_0 分成参照部分和运动部分,在经历过运动后 $\mathcal{P}_t = \mathcal{P}_{t-1} + \mathcal{D}_t$ 后,参照部分保持静止,运动部分为刚性运动,其中, \mathcal{P}_{t-1} 和 \mathcal{P}_t 表示两个邻接的点云帧,因此 L_{rec} 分为两部分:

$$[0082] \quad L_{\text{rec}}(\mathcal{D}_t) = L_{\text{ref}}(\mathcal{P}_t) + L_{\text{mov}}(\mathcal{P}_t);$$

[0083] $L_{\text{ref}}(\mathcal{P}_t)$ 是参照部分的误差, $L_{\text{mov}}(\mathcal{P}_t)$ 是运动部分的误差;

[0084] $L_{\text{ref}}(\mathcal{P}_t)$ 是每个点误差距离的平方和:

$$[0085] \quad L_{\text{ref}}(\mathcal{P}_t) = \sum_{p \in \mathcal{P}_{\text{ref}}^t} \|p - p^{\text{gt}}\|^2;$$

[0086] 其中, p^{gt} 是点p真实的位置;

[0087] $L_{\text{mov}}(\mathcal{P}_t)$ 的构成为:

$$[0088] \quad L_{\text{mov}}(\mathcal{P}_t) = L_{\text{shape}}(\mathcal{P}_{\text{mov}}^t, \mathcal{P}_{\text{mov}}^{t, \text{gt}}) + L_{\text{density}}(\mathcal{P}_{\text{mov}}^t, \mathcal{P}_{\text{mov}}^{t, \text{gt}});$$

[0089] 其中, L_{shape} 是用来惩罚不与目标形状匹配的点的, L_{density} 是预测点云与目标点云局部点密度, $\mathcal{P}_{\text{mov}}^t$ 指所述深度神经网络生成的第 t 帧的点云中的运动部分, $\mathcal{P}_{\text{mov}}^{t, \text{gt}}$ 指正确的第 t 帧的点云中的运动部分, gt 是 ground truth 的缩写, 表示正确的意思。

[0090] 通过误差损失函数预测运动信息与目标运动信息之间的差别; 所述运动类型包括旋转运动和平移运动。

[0091] 对于位移损失函数(误差损失函数), 此位移损失函数可以衡量预测运动信息与目标运动信息之间的差别, 如前所述, 这也是针对点云中运动部分而言的。由于有不同的运动类型, 因而也对应有不同形式。本发明只考虑了旋转与平移两种类型的运动。

[0092] 对于旋转运动, 见图4, 损失函数如下:

$$[0093] \quad L_{\text{disp}}^R(\mathcal{D}_t) = L_{\text{dir}}^R(\mathcal{D}_t) + L_{\text{ang}}^R(\mathcal{D}_t) + L_{\text{len}}^R(\mathcal{D}_t);$$

[0094] $L_{\text{dir}}^R(\mathcal{D}_t)$ 刻画了预测的位移是否垂直于真实的运动轴, 具体的计算公式为:

$$[0095] \quad L_{\text{dir}}^R(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} |\text{dot}(\mathcal{D}_t(p) / \|\mathcal{D}_t(p)\|_2, d_{\text{gt}})|$$

[0096] 其中, dot 表示点乘, $\mathcal{D}_t(p)$ 表示第 t 帧点云 p 的位移图, d_{gt} 是正确的运动轴的方向; $L_{\text{ang}}^R(\mathcal{D}_t)$ 则是各个点旋转角的偏差, 所有点旋转角度一致, 具体计算公式为:

$$[0097] \quad L_{\text{ang}}^R(\mathcal{D}_t) = \sigma^2 \{ \text{dot}(\text{proj}(p), \text{proj}(\mathcal{D}_t(p) + p)) \}_{\forall p \in \mathcal{P}_{\text{mov}}^{t-1}};$$

[0098] 其中, σ 为常数, $\text{proj}(p)$ 表示 p 点与将点 p 投影到正确运动轴上的投影点的距离, $\text{proj}(p) = (p - \pi(p)) / \|p - \pi(p)\|_2$;

[0099] $L_{\text{len}}^R(\mathcal{D}_t)$ 要求每个点旋转前和旋转后距离真实转轴的距离相同, 约束其运动的圆周性, 具体计算公式为:

$$[0100] \quad L_{\text{len}}^R(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} \| \|p - \pi(p)\|_2 - \|(\mathcal{D}_t(p) + p) - \pi(\mathcal{D}_t(p) + p)\|_2 |$$

[0101] 对于平移运动, 见图5, 损失函数如下:

$$[0102] \quad L_{\text{disp}}^T(\mathcal{D}_t) = L_{\text{dir}}^T(\mathcal{D}_t) + L_{\text{len}}^T(\mathcal{D}_t);$$

[0103] $L_{\text{dir}}^T(\mathcal{D}_t)$ 刻画了预测的位移是否平行于真实的运动轴, 具体计算公式为:

$$[0104] \quad L_{\text{dir}}^T(\mathcal{D}_t) = \sum_{p \in \mathcal{P}_{\text{mov}}^{t-1}} 1 - |\text{dot}(\mathcal{D}_t(p) / \|\mathcal{D}_t(p)\|_2, d_{\text{gt}})|$$

[0105] $L_{\text{len}}^T(\mathcal{D}_t)$ 则要求每个点移动的距离相同, 方差为0, 具体计算公式为:

$$[0106] \quad L_{\text{len}}^T(\mathcal{D}_t) = \sigma^2 (\| \|\mathcal{D}_t(p)\|_2 \|_{\forall p \in \mathcal{P}_{\text{mov}}^{t-1}})$$

[0107] 分割损失函数 $L_{\text{seg}}(S)$ 是预测分割与真实分割的多项逻辑斯特回归交叉熵 (softmax cross entropy)。

[0108] 运动信息损失函数为：

$$[0109] \quad L_{\text{mob}}(\mathcal{M}) = \|d - d^{\text{gt}}\|_2 + \|x - x^{\text{gt}}\|_2 + H(t, t^{\text{gt}});$$

[0110] 其中, d 、 x 和 t 分别为运动轴方向、运动轴位置和运动类型, d^{gt} 是正确的运动轴方向, x^{gt} 是正确的运动轴位置, t^{gt} 是正确的运动类型, H 是交叉熵。

[0111] 本发明通过引入一种新的循环神经网络结构和若干新颖的损失函数来完成对物体未来运动的预测, 这包括未来几个时刻的点云状态, 运动部分的分割, 运动类型和运动参数。

[0112] 进一步地, 本发明展示了使用MAPP-NET得到的运动性预测, 并且评估该方法的不同组件。本发明使用如下公式 (1) 定义的损失函数和Adam随机优化子训练网络。在本发明的实验中, 使用了运动单元数据集。本发明采样了单元的可见表面来创建点云, 称为“完整扫描”。本发明按照90/10的划分比例, 将数据集划分为训练/测试单元。本发明还从测试集得到部分扫描的集合, 用于额外的评估。

$$[0113] \quad L(D_t, S, M) = \sum_{t=1}^T (L_{\text{reg}}(D_t) + L_{\text{disp}}(D_t)) + L_{\text{seg}}(S) + L_{\text{mob}}(M); \quad (1)$$

[0114] 图6展示了对于完整和部分扫描, 在测试单元上进行运动预测的例子。对于每个例子, 展示了每个输入 (input) 点云的前5帧的预测帧 (frame), 绘制了预测的变换轴, 参考部件和运动部件。可以观察到MAPP-NET对于不同的运动类型的不同物体, 如何预测正确的部件运动并且生成对应的运动序列。例如, 本发明的方法准确地预测不同轴方向和位置的形状的旋转运动, 包括水平和垂直的轴方向, 例如第一行 (左) 展示的翻盖式手机以及第二行 (左) 展示了转动的闪存驱动设备 (U盘)。本发明的方法还准确地预测轴位置, 如第四行 (左) 展示的行李箱和第二行 (右) 展示的堆垛机的例子。

[0115] 还可以看到, 对于平移运动, 例如第五行 (右) 抽屉的运动, MAPP-NET能够通过平移预测其打开的正确方向, 尽管数据只显示抽屉的前表面, 而没有内部结构; 因为包裹该物体的参考部件太大了。发现第三行 (左) 的抽屉的把手一个相似的结果, 但是预测了不同的运动类型。更进一步, 我们能够从第五行 (左) 和最后一行 (右) 展示的例子中发现, 针对那些已经接近结束帧的输入点云, 本发明方法在找到运动的停止状态后, 已学会停止生成新的帧, 这表明该方法能够推测运动的范围。

[0116] 此外, MAPP-NET还能预测相同物体的多个部件的运动。给定多于一个运动部件的物体, 本发明方法能够要么迭代地预测多个运动, 如图2所示; 要么同时预测不同组件的运动, 特别是不同运动类型的组件。这是可行的, 因为本发明训练单个网络来预测所有的不同运动类型, 例如平移和旋转。如图7的同时运动的例子, 本发明展示了预测的分割的全部5帧连续的帧。生成的帧 (红色) 的运动部件当它们与输入帧更接近时, 用更浅的颜色表示。

[0117] 针对测试集通过MAPP-NET预测的运动性, 本发明进行定量评估, 通过度量运动参数和分割的误差, 因为本发明有基准可以使用。具体来说, 对于每个测试单元, 本发明使用两种度量来计算预测的变换轴 $M = (d, x)$ 与基准轴 $M^{\text{gt}} = (d^{\text{gt}}, x^{\text{gt}})$ 相比较的误差。第一个度量方式阐述了预测的轴方向的误差:

$$[0118] \quad E_{\text{angle}} = \arccos(|\text{dot}(d/\|d\|_2, d^{\text{gt}}/\|d^{\text{gt}}\|_2)|);$$

[0119] 简单地表示了预测与基准轴之间偏差的角度, 范围在 $[0, \pi/2]$ 。第二个度量方式计算了轴位置的误差:

[0120] $E_{\text{dist}} = \min(\|x - \pi(x)\|_2, 1)$;

[0121] $\pi(x)$ 将点 x 投影到由 $M^{\text{gt}} = (d^{\text{gt}}, x^{\text{gt}})$ 决定的基准运动轴。因为所有的形状被正则化到一个单位体中, 本发明截断最大距离到1。注意平移没有定义好的轴的位置。因此对于平移, 仅仅计算轴方向。当分类错误时, 变换类型误差 E_{seg} 设为1; 反之为0。分割误差 E_{seg} 仅仅度量被指定为错误标签的点的百分比。

[0122] 然后, 计算两个数据集的每个误差的均值: 完整和部分扫描。本发明方法的误差可以在表1中看到: 可以观察到所有的误差都相对很低, 表明预测的运动的准确性很高; 此外, 本发明方法针对于完整和部分扫描都达到相当的结果, 表明本发明方法的鲁棒性。

	Data type	Method	E_{angle}	E_{dist}	E_{seg}	E_{type}
[0123]	Complete scans	BaseNet	0.324	0.301	0.036	0.045
		Ours	0.209	0.153	0.038	0.047
	Partial scans	BaseNet	0.354	0.215	0.059	0.058
		Ours	0.255	0.147	0.050	0.059

[0124] 表1: 本发明方法和BaseNet的运动预测误差

[0125] 与BaseNet的对比, 为了展示利用MAPP-NET的优势, 能够在预测所有的运动相关的参数之前生成位移图, 本发明与基准进行了比较, 将它称为“BaseNet”。BaseNet将点云 P_0 作为输入, 使用标准的网络架构直接估计分割 S 和运动参数 M 。网络由编码器/解码器对和全连接层组成, 如图8所示。BaseNet的损失函数为:

[0126] $L(S, M) = L_{\text{seg}}(S) + L_{\text{motion}}(M)$;

[0127] 使用了公式(1)定义的两个损失函数项。

[0128] 表1展示了MAPP-NET和BaseNet在完整和部分扫描之间的比较, 可以发现BaseNet的分割误差 E_{seg} 和运动类型误差 E_{type} 与本发明方法相当, 但是它的轴方向误差 E_{angle} 和轴位置误差 E_{dist} 比本发明的至少高了5%。结果差异的主要原因可能是分割和分类任务相比运动预测更简单。网络架构像PointNet++已经表明了在那两个任务上能取得好的结果, 然而对于运动预测, 单独的输入帧可能会导致推测的歧义。

[0129] 在本发明的深度学习框架中, 本发明使用循环神经网络来生成多个描述运动的帧的序列, 它更多地限制了推断。结果, 运动参数地预测更加准确。

[0130] 图9展示了本发明方法与BaseNet在一些例子上的可视化对比。因为BaseNet没有生成运动帧, 展示它在输入点云上分割和预测的轴, 然而对于本发明方法, 一起展示了预测的分割和轴的5帧连续的帧。生成的帧的运动部件当它们更接近输入帧时用更浅的颜色表示。对于完整和部分扫描的平移和旋转, BaseNet更容易预测错误的运动类型, 从而造成复杂形状的预测错误, 例如, 对于在课桌下的键盘抽屉, 错误预测了滑动运动的方向。

[0131] 为了进一步验证本发明的损失函数, 在完整的扫描上进行三个消融研究实验。

[0132] L_{rec} 和 L_{dtsp} 的重要性。为了显示 L_{rec} 和 L_{dtsp} 的重要性, 这两项是预测的位移图 D_t 或者点云 P_t 与基准比较的损失函数项, 将本发明方法的结果与没有添加这两项中的任意一项得到的结果进行比较。表2的第二和第三行展示了这个实验得到的误差值, 对比第六行使用本发明的完整的损失函数。对比本发明损失函数完整的版本, 去除 L_{res} 和 L_{disp} 两者之一增大了误差, 并且更重要的是, 如图10所示, 中间预测的序列对比那些使用完整的损失函数得到的

结果,它的质量更差。

Method	E_{angle}	E_{dist}	E_{seg}	E_{type}
w/o L_{rec}	0.329	0.190	0.048	0.095
w/o L_{disp}	0.218	0.166	0.055	0.071
w/o L_{mob}	0.513	0.565	0.108	0.068
w/o L_{mob} and L_{seg}	0.623	0.499	0.213	0.065
Ours	0.209	0.153	0.038	0.047

[0134] 表2:对比完整的MAPP-NET和去除某个损失函数项的方法的消融实验,注意损失函数的所有项的重要性得到最低的误差(末行)

[0135] 没有重建损失项 L_{rec} ,尽管多亏了位移损失项 L_{disp} ,运动部件的运动看起来合理,点(特别是那些在参考部件的点)更容易移动到不可预料的位置。

[0136] 另一方面,当去掉位移损失项 L_{disp} ,运动部件的点的运动变得不一致,这导致了运动部件的扭曲。相比之下,本发明的完整的方法能够对于运动部件预测一个准确且平滑的运动并且还能保持参考部件不发生改变。

[0137] L_{mob} 和 L_{seg} 的重要性。在第二个消融实验,验证运动损失项 L_{mob} 和和分割损失项 L_{seg} 的用法,通过将本发明的完整网络与根据预测的位移图推测运动参数M和分割S而不是通过网络的额外层来预测它们的方法作比较。具体来说,本发明的网络从位移图 D_t 生成一个点云运动序列 P_t ,它能直接用于拟合运动参数M;然而对于分割S,本发明能过滤一些点,这依赖于它们是否比位移图 P_t 的合适阈值 θ 移动得更多,从而将点分为运动和静止(参考)点。

[0138] 在实验中,使用阈值 $\theta=0.01$ 来决定分割。为了拟合每对邻接帧的运动轴,计算最佳刚性变换矩阵,该矩阵具有将一帧变换到下一帧的最小的均值方差,并且提取:平移的轴方向、旋转的轴方向和位置。对于评估,计算了平移的轴方向误差 E_{angle} ,旋转的轴方向误差 E_{angle} 和轴位置误差 E_{dict} 。最终,计算所有的测试序列的所有邻接帧的均值误差。表2第四和第五行展示了这个实验的误差值。

[0139] 这种运动拟合方法对于噪声很敏感,导致很大误差;然而利用本发明的完整网络得到的预测更稳定并且提供更好的结果。运动参数拟合结果与本发明的结果的对比如图11所示,可以看到,没有运动损失项 L_{mob} 和分割损失项 L_{seg} ,一些离群点会造成轴拟合的大误差。然而对于没有运动损失项 L_{mob} 的例子,尽管分割看起来是正确的,不同点的位移的噪声也能造成轴拟合的大误差。例如,第二行展示的轮子,除了物体的更低部分的点有运动外,大多数点没有运动,导致了拟合出来的轴的位置偏离了轮子的中心。

[0140] 基于 P_t 定义 L_{rec} 和 L_{mob} 的重要性。此外,因为本发明的网络提供了位移图 D_t 和点云 P_t 作为中间输出,除了位移损失 L_{disp} ,所有的损失项都能基于 D_t 和 P_t 两者任一定义。因此,进行第三个消融实验来表现基于 P_t 的重建项 L_{rec} 和运动损失项 L_{mob} 的定义。正如本发明方法做的那样,这个定义比在 D_t 上定义更好。表3展示了这个实验证明的这个点。这个结果的主要原因是位移图 D_t 是在两个邻接的点云帧 T_{t-1} 和 P_t 之间定义的。因此,定义在 D_t 上的误差会影响 P_t 的生成,也会影响 D_{t+1} 。如果仅仅在每个 D_t 上独立地度量重建损失项 L_{rec} ,之后在学习过程中不能准确地将累计的误差加入考虑。相反, P_t 是通过应用所有的先前的位移图到输

入点云 P_0 而得到的。因此,通过在每个 P_t 上定义重建损失项 L_{rec} ,损失项在生成的序列的误差中提供了更全局的限制。运动损失项 L_{mob} 的定义也是采用相似的参数应用。

Method	E_{angle}	E_{dist}	E_{seg}	E_{type}
L_{mob} on \mathcal{D}_t	0.441	0.214	0.170	0.073
[0141] L_{rec} on \mathcal{D}_t	0.350	0.203	0.145	0.074
both on \mathcal{D}_t	0.569	0.331	0.308	0.095
Ours	0.209	0.153	0.038	0.047

[0142] 表3:使用 \mathcal{D}_t 而不是 P_t 定义重建损失项 L_{rec} 和运动损失项 L_{mah} 的比较。最后一行对应于本发明将两个损失项都定义在 P_t 上的方法,得到最低的误差

[0143] 正如实验所强调的,本发明的方法呈现了预测具有单个运动部件的物体的运动的高准确性。因此,对于在更多情形下预测物体的运动,本发明的方法算是一个不错的基础模块。例如,图2和图7显示了本发明的方法对于检测一个物体中多个运动部件的潜能,包括以并行方式发生的运动或者按照层次性顺序的运动。然而,对于这项更复杂任务,还需要进一步的实验来定量地评估本发明方法,可能需要构造具有多个可运动部件的物体的数据集和它们已知的运动参数以及分割。另外,本发明当前的数据集假定了形状是有意义的朝向并且数据集相对小,由276个运动单元组成。另一个更直接的改进方法可应用到更复杂场景的方向是强化本发明的数据集,通过应用随机变换到运动单元,以便本发明的网络能够以位姿不变的方式运作,或者利用部分扫描来训练网络从而改善它的抗干扰能力。

[0144] 未来工作的另一个方向是利用本发明的方法预测的运动性来合成输入形状的运动。作为这个更大的运动合成问题的一部分,一个有趣的子问题是学习如何补全物体的几何形状,当运动发生时物体的几何可能会丢失,例如,从橱柜中拉出的抽屉应显示其内部,如果对形状进行扫描或未对其内部进行建模,则会丢失内部。一个可能的方法是从预测的运动和存在的部件几何学习如何合成丢失的几何。这种方法至少需要以预分割的物体的形式建立一个训练集,并对其所有内部细节进行建模。

[0145] 本发明介绍了由重建损失函数和位移损失函数组成的损失函数,它保证了在保持物体形状的同时也准确地预测运动。重建损失衡量了在运动过程中,维持物体形状的程度,而位移损失衡量了位移场刻画运动的程度。表明:对比其他可选择的方法,这个损失函数能带来最准确的预测。循环神经网络(RNN)架构的使用允许本发明不仅预测运动的后续帧,也使本发明能够决定当运动停止时,除了推断运动参数还能推断预测的运动的范围,如:门能开多大角度。

[0146] 本发明展示了MAPP-NET能够十分准确地预测物体部件的运动,这些物体是具有不同运动类型(包括旋转和平移变换)的多种物体,既可以是3D物体的完整点云也可以是部分扫描的结果。此外,还验证本发明方法的合理性并且与基准方法进行了比较。最终,本发明展示了初步的结果,即本发明提出的网络有能按层次性方式来分割具有多个运动部分组成的物体的潜能,同时预测多个部件的运动。

[0147] 技术效果:

[0148] (1) 本发明将功能可见性分析问题归为对于输入的几何进行分割并且标记每个分割的运动类型和参数;因此,本发明提出的深度神经网络从预分割和已知运动的三维形状学习,之后执行分割和预测。

[0149] (2) 本发明的深度神经网络MAPP-NET从一个三维点云形状预测部件的运动,然而并不需要该形状的分割;本发明通过训练一个深度学习模型来同时分割输入形状和预测它的部件的运动从而实现的。

[0150] (3) 本发明的网络是在基准分割和运动参数标识好了的运动单元数据集上训练的;一旦训练完毕,它能够用来预测单个未分割的表示物体的一个静止状态的点云的运动。

[0151] (4) 本发明介绍了由重建损失函数和位移损失函数组成的损失函数,它保证了在保持物体形状的同时也准确地预测运动;重建损失衡量了在运动过程中,维持物体形状的程度,而位移损失衡量了位移场刻画运动的程度;表明:对比其他可选择的方法,这个损失函数能带来最准确的预测。

[0152] (5) 循环神经网络(RNN)架构的使用允许本发明不仅预测运动的后续帧,也使本发明能够决定当运动停止时,除了推断运动参数还能推断预测的运动的范围,如:门能开多大。

[0153] (6) 本发明展示了MAPP-NET能够十分准确地预测物体部件的运动,这些物体是具有不同运动类型(包括旋转和平移变换)的多种物体,既可以是3D物体的完整点云也可以是部分扫描的结果。

[0154] (7) 本发明展示了初步的结果,本发明提出的网络有能按层次性方式来分割具有多个运动部分组成的物体的潜能,同时预测多个部件的运动。

[0155] 进一步地,如图12所示,基于上述基于深度神经网络的运动预测方法,本发明还相应提供了一种智能终端,所述智能终端包括处理器10、存储器20及显示器30。图7仅示出了智能终端的部分组件,但是应理解的是,并不要求实施所有示出的组件,可以替代的实施更多或者更少的组件。

[0156] 所述存储器20在一些实施例中可以是所述智能终端的内部存储单元,例如智能终端的硬盘或内存。所述存储器20在另一些实施例中也可以是所述智能终端的外部存储设备,例如所述智能终端上配备的插接式硬盘,智能存储卡(Smart Media Card, SMC),安全数字(Secure Digital, SD)卡,闪存卡(Flash Card)等。进一步地,所述存储器20还可以既包括所述智能终端的内部存储单元也包括外部存储设备。所述存储器20用于存储安装于所述智能终端的应用软件及各类数据,例如所述安装智能终端的程序代码等。所述存储器20还可以用于暂时地存储已经输出或者将要输出的数据。在一实施例中,存储器20上存储有基于深度神经网络的运动预测程序40,该基于深度神经网络的运动预测程序40可被处理器10所执行,从而实现本申请中基于深度神经网络的运动预测方法。

[0157] 所述处理器10在一些实施例中可以是一中央处理器(Central Processing Unit, CPU),微处理器或其他数据处理芯片,用于运行所述存储器20中存储的程序代码或处理数据,例如执行所述基于深度神经网络的运动预测方法等。

[0158] 所述显示器30在一些实施例中可以是LED显示器、液晶显示器、触控式液晶显示器以及OLED(Organic Light-Emitting Diode, 有机发光二极管)触摸器等。所述显示器30用于显示在所述智能终端的信息以及用于显示可视化的用户界面。所述智能终端的部件10-

30通过系统总线相互通信。

[0159] 在一实施例中,当处理器10执行所述存储器20中基于深度神经网络的运动预测程序40时实现以下步骤:

[0160] 使用数据集训练深度神经网络;

[0161] 将三维点云输入至所述深度神经网络;

[0162] 所述深度神经网络输出所述三维点云的第一部分和第二部分,将所述第一部分作为运动子单元,所述第二部分作为运动单元的参考部分;

[0163] 根据所述三维点云的输出完成网络预测,输出运动信息,所述运动信息包括运动性分割、运动轴和运动类型。

[0164] 本发明还提供一种存储介质,其中,所述存储介质存储有基于深度神经网络的运动预测程序,所述基于深度神经网络的运动预测程序被处理器执行时实现所述基于深度神经网络的运动预测方法的步骤;具体如上所述。

[0165] 综上所述,本发明提供了一种基于深度神经网络的运动预测方法和智能终端,所述方法包括:使用数据集训练深度神经网络;将三维点云输入至所述深度神经网络;所述深度神经网络输出所述三维点云的第一部分和第二部分,将所述第一部分作为运动子单元,所述第二部分作为运动单元的参考部分;根据所述三维点云的输出完成网络预测,输出运动信息,所述运动信息包括运动性分割、运动轴和运动类型。本发明实现了在非结构化并且可能是部分扫描的各种铰链式物体在静止状态下同时运动和部件的预测结果,能够十分准确地预测物体部件的运动。

[0166] 当然,本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关硬件(如处理器,控制器等)来完成,所述的程序可存储于一计算机可读的存储介质中,所述程序在执行时可包括如上述各方法实施例的流程。其中所述的存储介质可为存储器、磁碟、光盘等。

[0167] 应当理解的是,本发明的应用不限于上述的举例,对本领域普通技术人员来说,可以根据上述说明加以改进或变换,所有这些改进和变换都应属于本发明所附权利要求的保护范围。

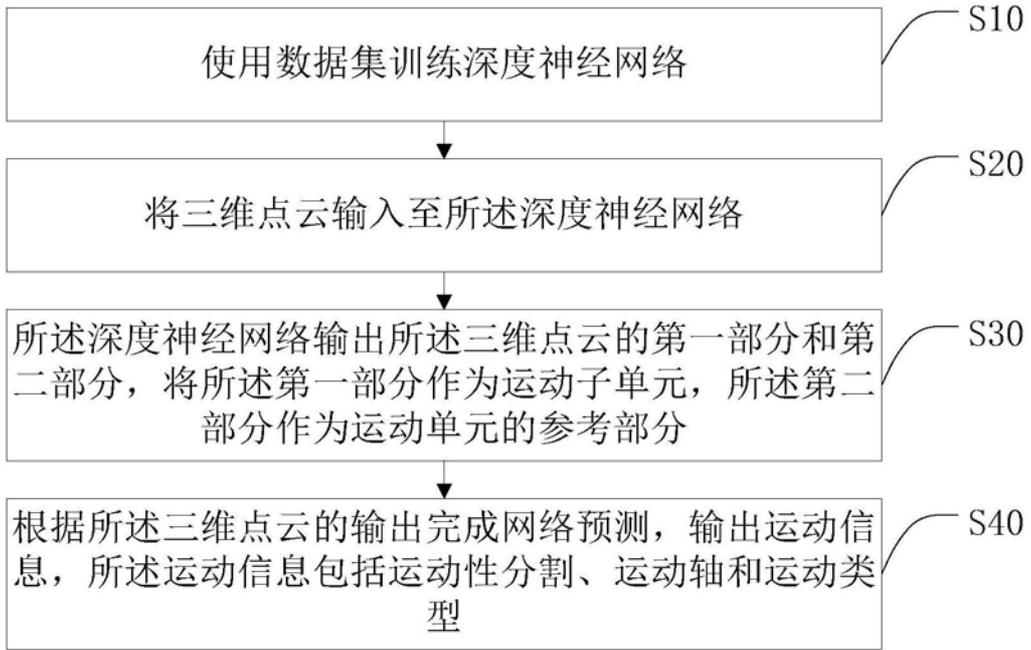


图1

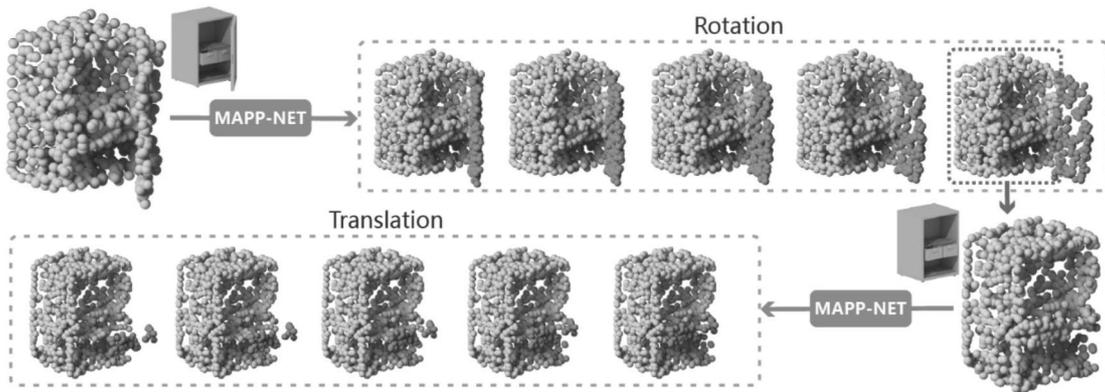


图2

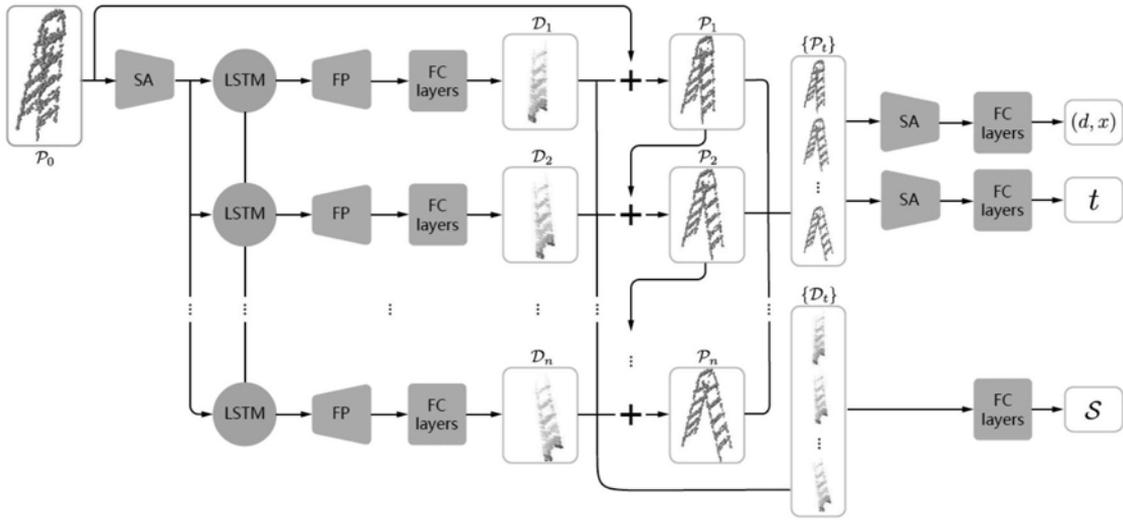


图3

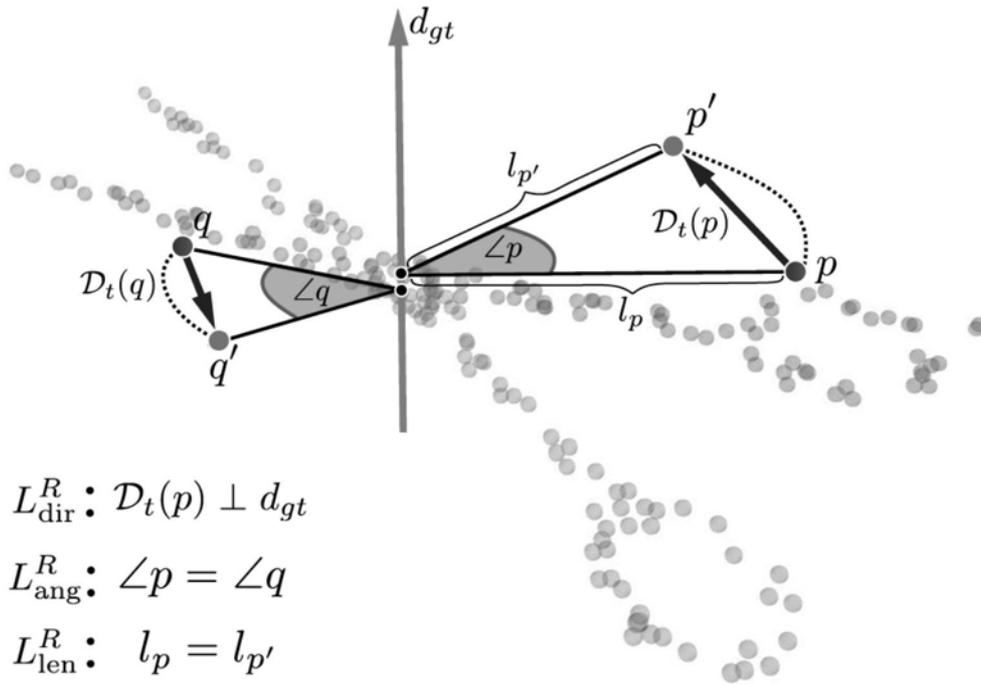


图4

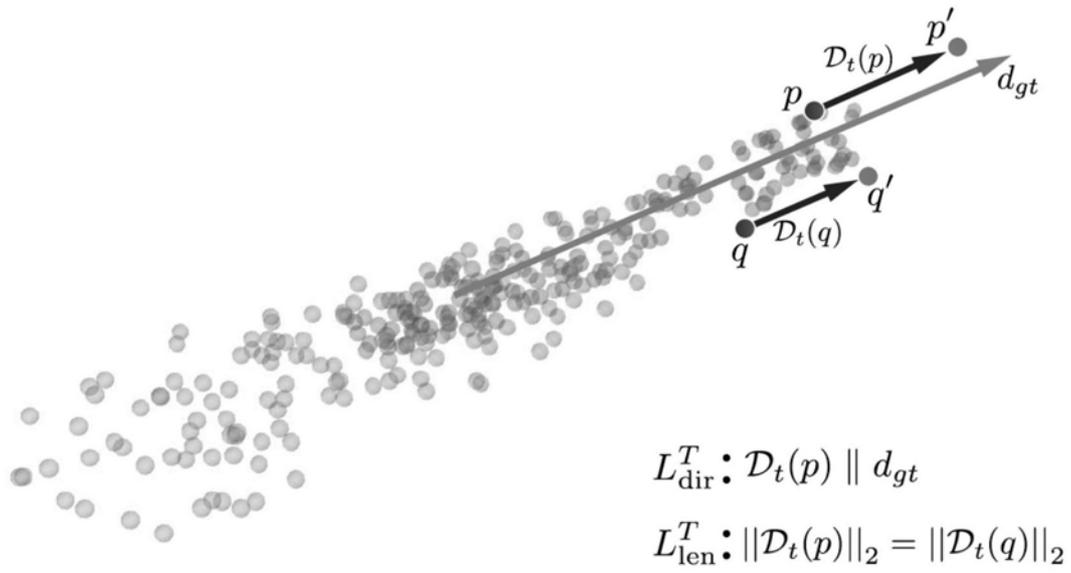


图5

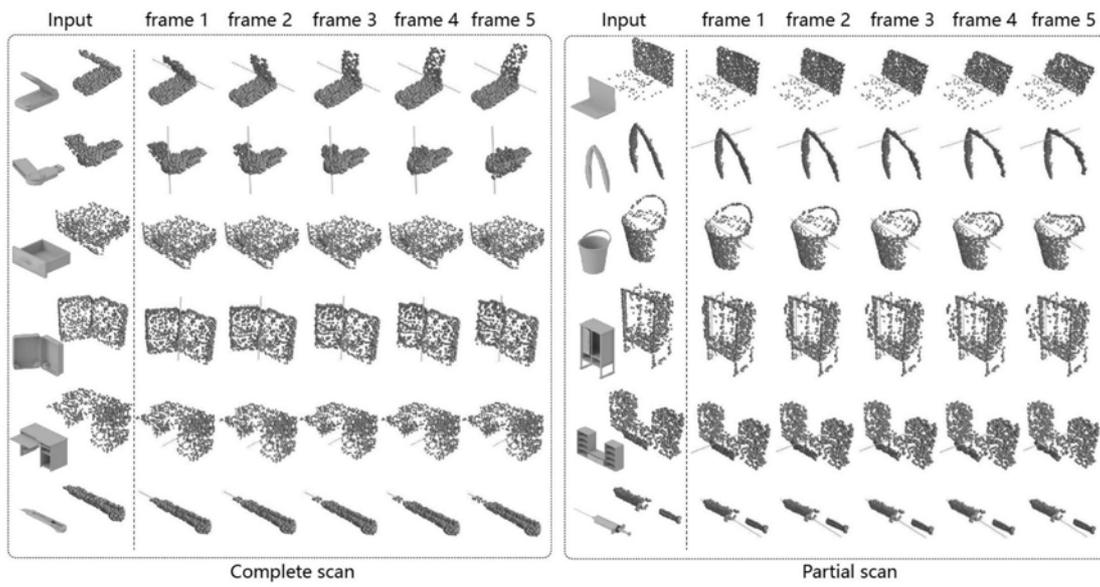


图6

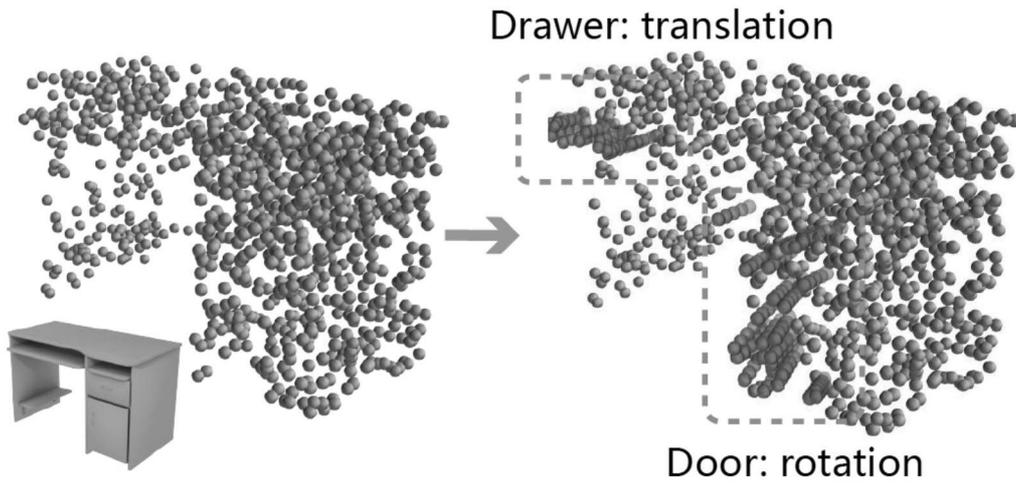


图7

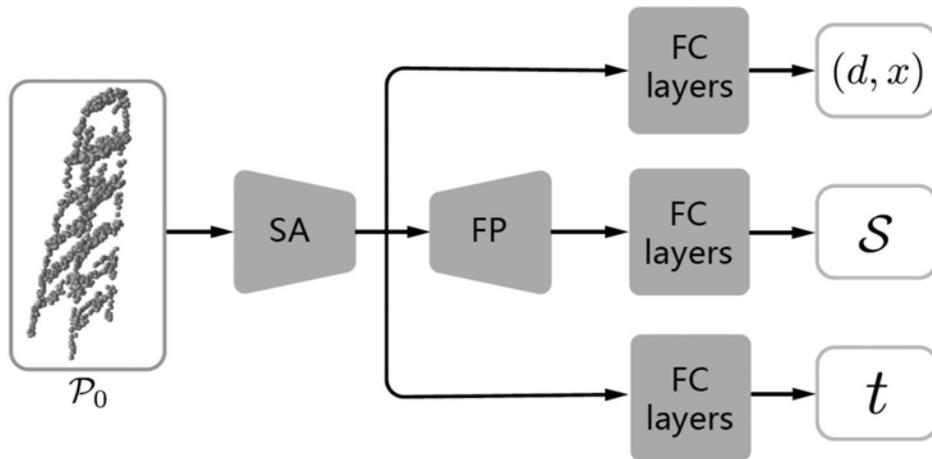


图8

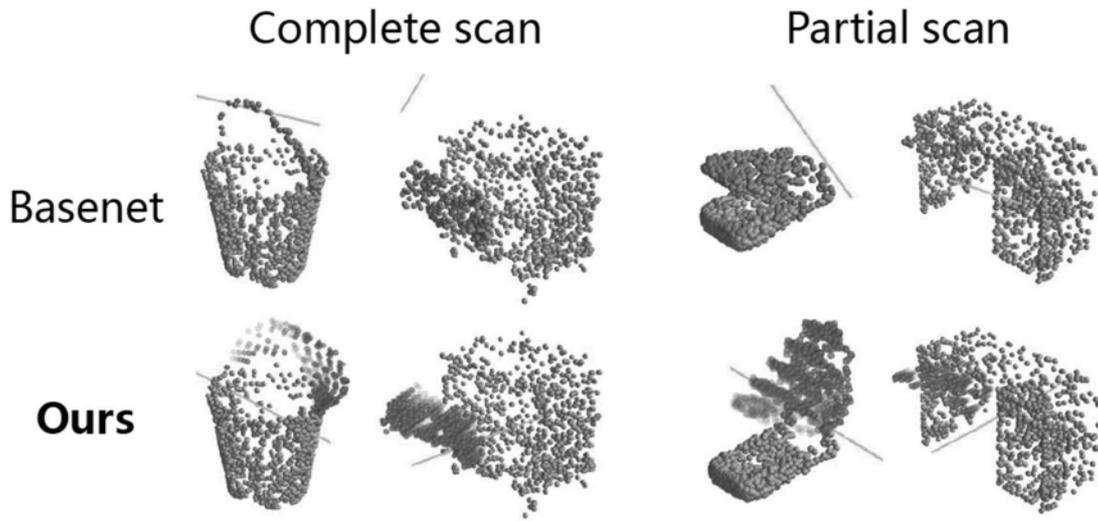


图9

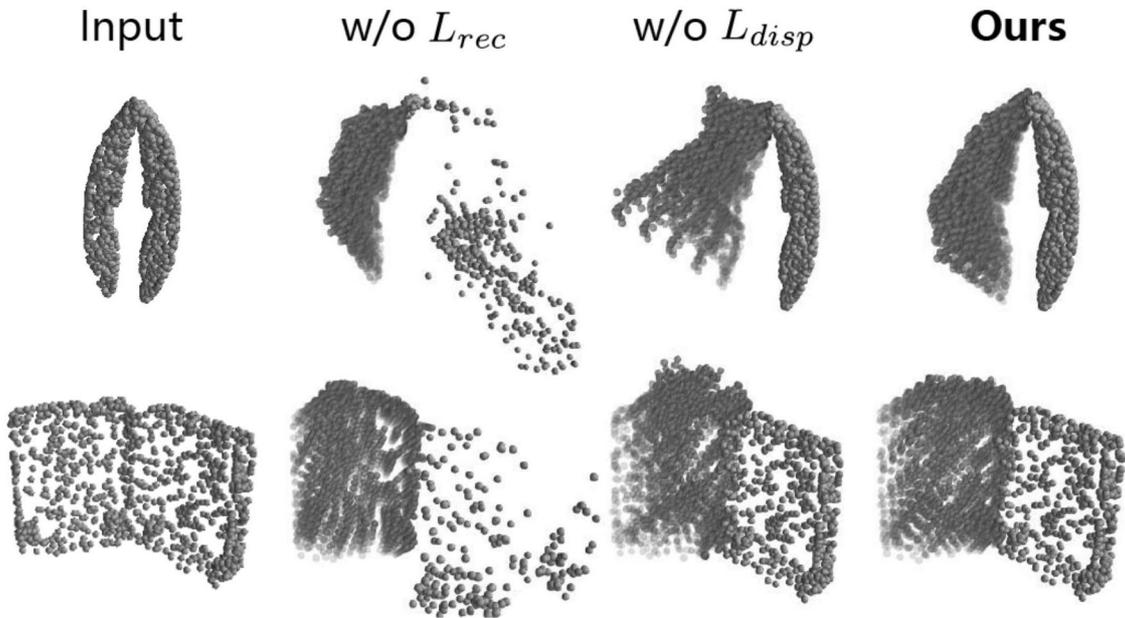


图10

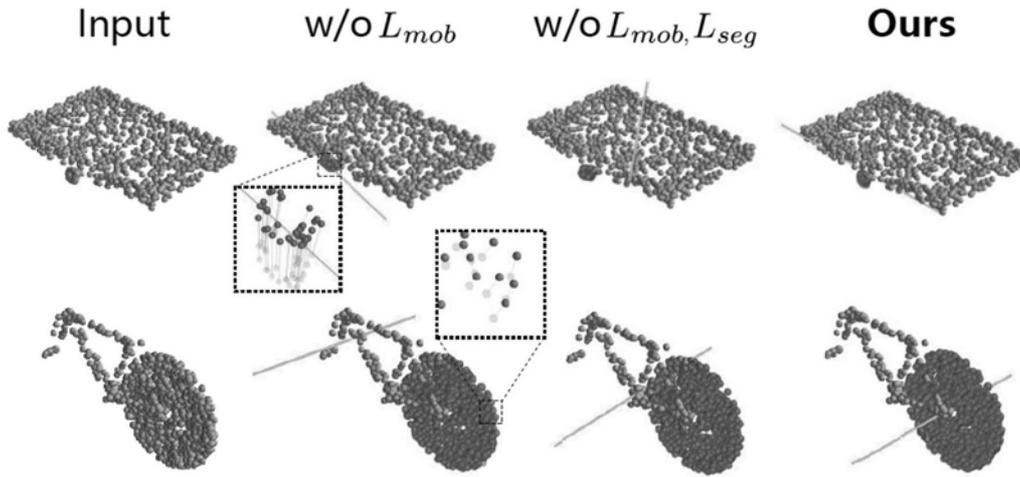


图11

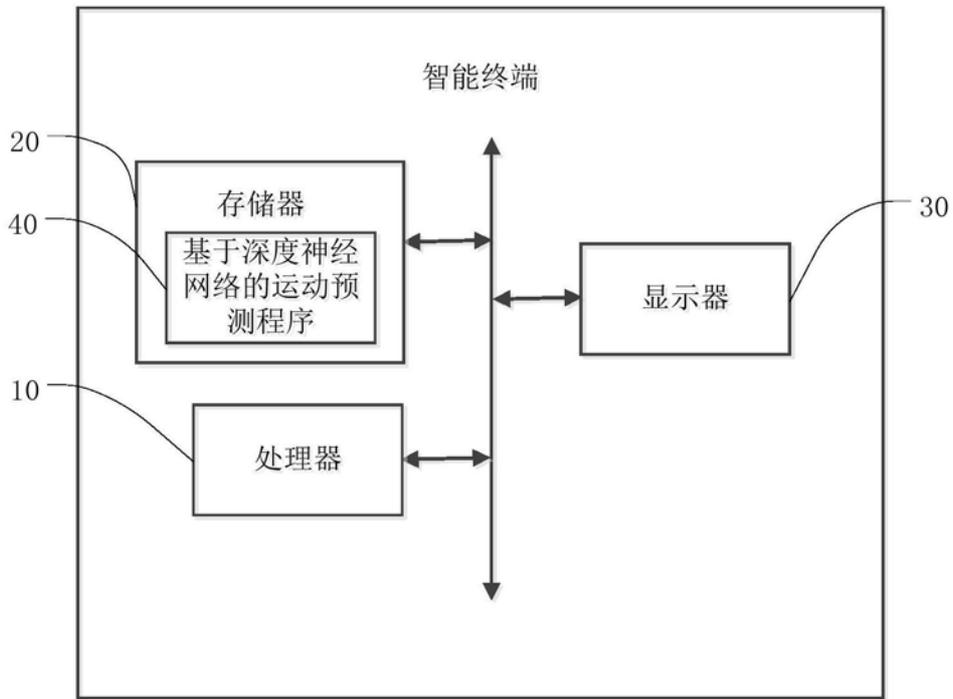


图12