



(10) 授权公告号 CN 118377384 B

(45) 授权公告日 2024. 09. 10

(21) 申请号 202410799538.7

G06N 3/0442 (2023.01)

(22) 申请日 2024.06.20

G06N 3/047 (2023.01)

(65) 同一申请的已公布的文献号

G06N 3/092 (2023.01)

申请公布号 CN 118377384 A

G06N 7/01 (2023.01)

(43) 申请公布日 2024.07.23

G06T 17/00 (2006.01)

G06V 40/20 (2022.01)

(73) 专利权人 深圳市星野信息技术有限公司

(56) 对比文件

地址 518057 广东省深圳市南山区粤海街

CN 112509151 A, 2021.03.16

道高新区社区白石路3609号深圳湾科

CN 114035678 A, 2022.02.11

技生态园二区9栋417

审查员 胡平

(72) 发明人 方俊钦 毛佳乐

(74) 专利代理机构 天津创信方达专利代理事务

所(普通合伙) 12247

专利代理师 李京京

(51) Int. Cl.

G06F 3/01 (2006.01)

权利要求书4页 说明书13页 附图2页

(54) 发明名称

一种虚拟场景交互方法和系统

(57) 摘要

本发明属于虚拟交互技术领域,本发明公开了一种虚拟场景交互方法和系统;包括:获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列;将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来n个虚拟场景状态;基于未来n个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备,大幅提升了用户的交互体验。

S1、获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列

S2、将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列

S3、根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来n个虚拟场景状态;基于未来n个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备

1.一种虚拟场景交互方法,其特征在于,包括:S1、获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列;

S2、将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;

S3、根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来n个虚拟场景状态;基于未来n个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备;

所述用户当前交互指令数据包括用户输入设备操作数据和生物特征数据;

用户输入设备操作数据包括按键数据、手势数据和设备位移数据;生物特征数据包括眼球运动数据和肢体运动数据;

相关环境数据包括用户所处虚拟位置数据、视角方向数据和用户所处环境状态数据;用户所处环境状态数据包括场景物体位置数据和光照数据;所述按键数据包括按键事件的时间戳、按键编码和扫描码;

所述手势数据的获取方式包括:

采用深度相机以固定的帧率持续采集用户手部区域的点云数据,构成点云数据序列;

从点云数据序列中提取手部精确点云数据,将手部区域点云数据进行降采样,得到完整手部点云数据;

将完整手部点云数据进行特征提取,得到手掌法向量、指尖位置向量、指根位置向量和指间角度向量;将手掌法向量、指尖位置向量、指根位置向量和指间角度向量拼接得到手势特征向量;

将获得的手势特征向量按照时间的先后排列成一个手势特征向量序列,将手势特征向量序列输入至预先训练完成的手势识别模型中,输出对应的手势类型的概率分布,取概率分布中概率最大的手势类型的编码作为手势数据;

所述将用户当前交互指令数据进行预处理的方式包括:

将用户当前交互指令数据中不同模态的数据进行归一化处理,得到归一化指令数据;

将归一化指令数据进行时间序列上的对齐,得到对应的多模态序列;将多模态序列在时间维度上拼接,形成一个具有g个特征通道的特征时间序列数据;

将特征时间序列数据构建为一个三维张量,其中两个维度表示时间和特征通道,第三个维度用于表示对应的数据;

所述基于深度强化学习的时间预测模型的构建方式包括:

时间预测模型包括序列建模网络和强化学习网络;将门控循环单元网络作为序列建模网络的基础框架,门控循环单元网络包括网络输入层、GRU隐藏层和状态表示输出层;

收集N1组的交互序列数据,每个交互序列数据包含时间步特征表示和对应的标签;

将交互序列数据按照批次逐步输入至门控循环单元网络得到对应批次的最终状态向量,将对应批次的最终状态向量输入到预设的分类器中得到预测的标签概率的分布;计算标签概率和标签之间的交叉熵损失,对门控循环单元网络和分类器的可训练参数使用优化器进行梯度更新,并最小化交叉熵损失,得到序列建模网络;

定义强化学习网络为一个半马尔可夫决策过程,为半马尔可夫决策过程定义一个元组(S1,N2,P1,R1);其中,S1是低层管理器的决策的状态空间,N2是高层管理器可选择的子任

务空间, $P1$ 是低层管理器的决策的状态转移函数, $R1$ 是低层管理器的决策的奖赏函数;

将半马尔可夫决策过程划分为高层管理器的策略和低层管理器的策略; 定义半马尔可夫决策过程的目标是学习一个最优的高层管理器的策略和 Q 个最优的低层管理器的策略, 将交互序列数据通过门控循环单元网络获取的对应批次的最终状态向量, 作为半马尔可夫决策过程的状态, 训练半马尔可夫决策过程, 并使用策略梯度算法分别优化高层管理器的策略和低层管理器的策略; 使得在半马尔可夫决策过程中计算的期望累积奖赏最大化; 得到构建完成的强化学习网络;

所述概率图模型的构建方式包括:

定义概率图模型的随机变量, 随机变量为相关环境数据中的数据; 根据虚拟场景中各个物体之间的相互影响关系, 构建无向图结构; 无向图结构中的节点表示随机变量, 无向图结构中的边表示随机变量之间的相互依赖关系; 为每个节点定义一个节点潜在函数, 为每条边定义一个边潜在函数;

收集 $M1$ 组的历史数据, 历史数据为虚拟场景状态的时间序列数据; 初始化节点潜在函数和边潜在函数的参数, 利用虚拟场景状态的时间序列数据对概率图模型进行训练, 得到节点潜在函数和边潜在函数的参数; 训练的方法为最大似然估计方法; 即完成了概率图模型的构建;

所述未来 n 个虚拟场景状态的获取方式包括:

定义概率图模型的转移概率, 基于概率图模型的转移概率, 对未来 n 个时间步长进行推理, 得到每个时间步长下虚拟场景各个物体的概率分布, 在进行推理过程中, 将用户交互序列按时间步长依次输入概率图模型;

对于每个时间步长, 根据概率分布采样或选取场景物体的状态, 构建出该时间步长下的虚拟场景状态, 继而得到未来 n 个虚拟场景状态。

2. 根据权利要求1所述的虚拟场景交互方法, 其特征在于, 所述进行降采样的方式包括:

将点云数据序列有序地插入到预先构建完成的八叉树中, 然后以固定的空间分辨率对八叉树进行遍历, 得到完整手部点云数据;

所述八叉树的构建方式包括:

将点云数据序列所在的三维空间划分为规则的立方体单元, 根据点云数据序列中点的坐标, 将每个点插入到对应的立方体单元中, 对于包含 m 个点的立方体单元, 将其继续递归划分为8个子单元, 直到每个立方体单元内最多只有一个点, 即初步完成八叉树的构建;

所述对八叉树进行遍历的方式包括:

设置一个距离阈值 d , 从八叉树的根节点开始, 按照深度优先或广度优先的遍历八叉树的每个节点, 对于每个非空的节点, 计算节点中所有点到节点中心的距离, 选取距离节点中心最近的点作为该节点的代表点;

初始化一个空的列表 L , 遍历八叉树的每个非空的节点 X , 计算 X 的中心点 c 和代表点 p 之间的距离 $d(c, p)$, 若 $d(c, p)$ 大于或等于 d , 则将 p 加入列表 L , 最终列表 L 中的点即为完整手部点云数据;

其中, 距离 $d(c, p)$ 的计算公式为:

$d(c, p) = \sqrt{(c_x - p_x)^2 + (c_y - p_y)^2 + (c_z - p_z)^2}$; 其中, (c_x, c_y, c_z) 为中心点c的三维坐标, (p_x, p_y, p_z) 为点p的三维坐标。

3. 根据权利要求2所述的虚拟场景交互方法, 其特征在于, 所述进行特征提取的方式包括:

将完整手部点云数据进行主成分分析, 得到主方向向量 $\vec{v}_1$, 手掌法向量 $\vec{n}$ 为 $\vec{v}_1$ 与手部所在的三维坐标系的z轴负方向的矢量积, 即 $\vec{n} = \vec{v}_1 \times (-\vec{k})$, 其中, $\vec{k}$ 为z轴的单位向量;

将完整手部点云数据构造成集合P, 基于集合P构造完整的凸壳CH(P);

对完整的CH(P) 进行层次聚类, 得到k+1个层次, 记作L0, L1, ..., Lk-1, Lk; 其中, Lk-1即为指尖点集;

对Lk-1中的每个点vj, 遍历CH(P) 上的所有顶点, 计算vj到所有顶点的欧几里得距离并取最小值dj, 计算完所有vj的dj后, 取dj的最小值对应的点, 即为指尖位置, 将指尖位置进行向量化, 即得到指尖位置向量 $\vec{v}_t$; 其中, t为手指的索引;

计算每个指尖位置向量 $\vec{v}_t$ 到手掌平面的距离dt, 沿手掌平面的法线 $\vec{N}$ 方向从 $\vec{v}_t$ 往回偏移α得到对应指根位置向量 $\vec{q}_t = \vec{v}_t - \alpha \cdot d_t \cdot \vec{N}$; 其中, α为经验系数;

对于任意两相邻的指尖位置向量 $\vec{v}_t$ 和 $\vec{v}_{t+1}$, 计算 $\vec{v}_t$ 的矢量 $\vec{u}_t = \vec{v}_t - \vec{q}_t$ 和 $\vec{v}_{t+1}$ 的矢量 $\vec{u}_{t+1} = \vec{v}_{t+1} - \vec{q}_{t+1}$, 其中, $\vec{q}_{t+1}$ 为t+1手指对应的指根位置向量;

则基于矢量 $\vec{u}_t$ 和 $\vec{u}_{t+1}$ 计算得到指间角度 $\theta_t = \arccos\left(\frac{\vec{u}_t \cdot \vec{u}_{t+1}}{\|\vec{u}_t\| \cdot \|\vec{u}_{t+1}\|}\right)$, 将指间角度向量化, 即得到指间角度向量。

4. 根据权利要求3所述的虚拟场景交互方法, 其特征在于, 所述凸壳CH(P) 的构造方式包括:

初始时, 凸壳CH(P) 为任意4个不共面的点构成的四面体, 对于集合P中的点b, 若b在CH(P) 内, 则忽略b, 若b在CH(P) 上, 则将b添加到CH(P) 的顶点集合, 若b在CH(P) 外, 则用b对CH(P) 进行遍历划分, 生成新的CH(P), 最终得到完整的CH(P);

所述进行层次聚类的方式包括:

初始化L0等于CH(P), 对于每一个层次Lr, r属于区间[0, k], 且k为大于1的正整数; 计算Lr中所有点对(vp, vq) 的欧几里得距离d(vp, vq), 提取最大的欧几里得距离对应的点对(pi, pj), 将(pi, pj) 合并为一个新的簇, 移除pi和pj, 将剩余的点和新的簇组成新的点集Lr+1; 重复直至Lr+1为空集或只包含一个簇, 即完成了对完整的CH(P) 的层次聚类。

5. 根据权利要求4所述的虚拟场景交互方法, 其特征在于, 所述手势识别模型的训练方式包括:

收集M组手势动作的手势特征向量序列及其对应的手势类型的编码, 构成训练数据集; 定义时间卷积网络作为手势识别模型的骨干网络, 时间卷积网络由h个残差时间卷积块组

成,每个残差时间卷积块包括时间卷积层和dropout层,并在时间卷积层和dropout层中加入残差连接;

时间卷积层的计算公式为:

$$y_s = \sum_{i=0}^{k-1} f(x_{s-i}) \Theta w_i + B; \text{其中, } y_s \text{ 为时间卷积层在时间步 } s \text{ 时的输出特征, } k \text{ 为卷积核的大小, } i \text{ 为卷积核在时间维度上的偏移量, } x_{s-i} \text{ 为第 } s-i \text{ 个时间步的输入特征, } \Theta \text{ 表示卷积操作, } w_i \text{ 为卷积核系数, } B \text{ 为偏置项, } f(\cdot) \text{ 为ReLU函数, } f(x_{s-i}) \text{ 是将输入特征 } x_{s-i} \text{ 通过ReLU函数进行变换后的结果;}$$

核的大小, i 为卷积核在时间维度上的偏移量, x_{s-i} 为第 $s-i$ 个时间步的输入特征, Θ 表示卷积操作, w_i 为卷积核系数, B 为偏置项, $f(\cdot)$ 为ReLU函数, $f(x_{s-i})$ 是将输入特征 x_{s-i} 通过ReLU函数进行变换后的结果;

最后一个残差时间卷积块的输出是上下文特征序列,将骨干网络添加一个分类头,将上下文特征序列输入分类头,对上下文特征序列中每个时间步的特征使用全连接层进行分类,得到手势类型的概率分布;

进行分类的公式为: $P(y_s | c_s) = \text{Softmax}(W_{q1} \cdot c_s + B_{q1})$; 其中, W_{q1} 为全连接层的权重, B_{q1} 为全连接层的偏置, c_s 为上下文特征序列在时间步 s 处的特征;

$$\text{定义手势识别模型的损失函数 } Loss = -\frac{1}{S} \sum_{s=1}^S \log P(\hat{y}_s | c_s); \text{其中, } S \text{ 为上下文特征序列的总长度, } \hat{y}_s \text{ 为第 } s \text{ 个时间步的真实的手势类型的编码;}$$

征序列的总长度, $\hat{y}_s$ 为第 s 个时间步的真实的手势类型的编码;

利用训练数据集训练手势识别模型,过程中最小化损失函数,并不断优化手势识别模型的参数,得到训练完成的手势识别模型。

6.一种虚拟场景交互系统,用于实现权利要求1至5中任一项所述的虚拟场景交互方法,其特征在于,包括:数据采集处理模块,用于获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列;

预测模块,用于将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;

虚拟环境调控模块,用于根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来 n 个虚拟场景状态;基于未来 n 个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备,各个模块之间通过有线和/或无线的方式进行连接。

一种虚拟场景交互方法和系统

技术领域

[0001] 本发明涉及虚拟交互技术领域,更具体地说,本发明涉及一种虚拟场景交互方法和系统。

背景技术

[0002] 申请公开号为CN115525149A的专利公开了一种虚拟场景与真实场景的数据交互方法、装置及系统,包括真实场景数据获取模块、虚拟场景构建模块和语音融合模块,真实场景数据获取模块将真实用户的手势操作进行识别,得到手势交互数据;虚拟场景构建模块基于手势交互数据和控制交互数据构建虚拟建筑场景;语音融合模块将真实用户的语音信息融合至虚拟建筑场景,得到最终虚拟建筑场景,通过将手势交互数据和语音信息融合,构建虚拟建筑场景,解决了现有的虚拟场景与真实场景的数据交互装置交互的数据不充分的问题。

[0003] 但是依然无法充分利用用户手势等丰富的多模态交互信息,这对于虚拟场景交互来说尤为重要,因为手势是用户表达意图的重要方式,由于无法全面捕捉用户的交互意图,导致无法准确预测用户的行为,影响了交互的智能性,其次,现有的虚拟场景交互方法通常只根据用户当前的交互指令来渲染虚拟场景,无法提前预测用户未来的交互行为序列;这就意味着,场景无法及时做出相应的变化,交互体验难免会显得不够流畅和自然,此外,现有方法缺乏对虚拟场景上下文环境的建模能力,无法充分利用场景中物体位置、光照条件等重要信息来预测场景的变化,这就导致交互的准确性受到限制,无法真正实现用户与虚拟场景的自然交互。

[0004] 鉴于此,本发明提出一种虚拟场景交互方法和系统以解决上述问题。

发明内容

[0005] 为了克服现有技术的上述缺陷,为实现上述目的,本发明提供如下技术方案:一种虚拟场景交互方法,包括:S1、获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列;

[0006] S2、将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;

[0007] S3、根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来n个虚拟场景状态;基于未来n个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备。

[0008] 进一步地,所述用户当前交互指令数据包括用户输入设备操作数据和生物特征数据;

[0009] 用户输入设备操作数据包括按键数据、手势数据和设备位移数据;生物特征数据包括眼球运动数据和肢体运动数据;

[0010] 相关环境数据包括用户所处虚拟位置数据、视角方向数据和用户所处环境状态数

据;用户所处环境状态数据包括场景物体位置数据和光照数据;所述按键数据包括按键事件的时间戳、按键编码和扫描码;

[0011] 所述手势数据的获取方式包括:

[0012] 采用深度相机以固定的帧率持续采集用户手部区域的点云数据,构成点云数据序列;

[0013] 从点云数据序列中提取手部精确点云数据,将手部区域点云数据进行降采样,得到完整手部点云数据;

[0014] 将完整手部点云数据进行特征提取,得到手掌法向量、指尖位置向量、指根位置向量和指间角度向量;将手掌法向量、指尖位置向量、指根位置向量和指间角度向量拼接得到手势特征向量;

[0015] 将获得的手势特征向量按照时间的先后排列成一个手势特征向量序列,将手势特征向量序列输入至预先训练完成的手势识别模型中,输出对应的手势类型的概率分布,取概率分布中概率最大的手势类型的编码作为手势数据。

[0016] 进一步地,所述进行降采样的方式包括:

[0017] 将点云数据序列有序地插入到预先构建完成的八叉树中,然后以固定的空间分辨率对八叉树进行遍历,得到完整手部点云数据;

[0018] 所述八叉树的构建方式包括:

[0019] 将点云数据序列所在的三维空间划分为规则的立方体单元,根据点云数据序列中点的坐标,将每个点插入到对应的立方体单元中,对于包含 m 个点的立方体单元,将其继续递归划分为8个子单元,直到每个立方体单元内最多只有一个点,即初步完成八叉树的构建;

[0020] 所述对八叉树进行遍历的方式包括:

[0021] 设置一个距离阈值 d ,从八叉树的根节点开始,按照深度优先或广度优先的遍历八叉树的每个节点,对于每个非空的节点,计算节点中所有点到节点中心的距离,选取距离节点中心最近的点作为该节点的代表点;

[0022] 初始化一个空的列表 L ,遍历八叉树的每个非空的节点 X ,计算 X 的中心点 c 和代表点 p 之间的距离 $d(c, p)$,若 $d(c, p)$ 大于或等于 d ,则将 p 加入列表 L ,最终列表 L 中的点即为完整手部点云数据;

[0023] 其中,距离 $d(c, p)$ 的计算公式为:

[0024]
$$d(c, p) = \sqrt{(c_x - p_x)^2 + (c_y - p_y)^2 + (c_z - p_z)^2}$$
;其中, (c_x, c_y, c_z) 为中心点 c 的三维坐标, (p_x, p_y, p_z) 为代表点 p 的三维坐标。

[0025] 进一步地,所述进行特征提取的方式包括:

[0026] 将完整手部点云数据进行主成分分析,得到主方向向量 $\vec{v}_1$,手掌法向量 $\vec{n}$ 为 $\vec{v}_1$ 与手部所在的三维坐标系的 z 轴负方向的矢量积,即 $\vec{n} = \vec{v}_1 \times (-\vec{k})$,其中, $\vec{k}$ 为 z 轴的单位向量;

[0027] 将完整手部点云数据构造集合 P ,基于集合 P 构造完整的凸壳 $CH(P)$;

[0028] 对完整的CH(P)进行层次聚类,得到k+1个层次,记作 $L_0, L_1, \dots, L_{k-1}, L_k$;其中, L_{k-1} 即为指尖点集;

[0029] 对 L_{k-1} 中的每个点 v_j ,遍历CH(P)上的所有顶点,计算 v_j 到所有顶点的欧几里得距离并取最小值 d_j ,计算完所有 v_j 的 d_j 后,取 d_j 的最小值对应的点,即为指尖位置,将指尖位置进行向量化,即得到指尖位置向量 $\vec{v}_t$;其中,t为手指的索引;

[0030] 计算每个指尖位置向量 $\vec{v}_t$ 到手掌平面的距离 d_t ,沿手掌平面的法线 $\vec{N}$ 方向从 $\vec{v}_t$ 往回偏移 α 得到对应指根位置向量 $\vec{q}_t = \vec{v}_t - \alpha \cdot d_t \cdot \vec{N}$;其中, α 为经验系数;

[0031] 对于任意两相邻的指尖位置向量 $\vec{v}_t$ 和 $\vec{v}_{t+1}$,计算 $\vec{v}_t$ 的矢量 $\vec{u}_t = \vec{v}_t - \vec{q}_t$ 和 $\vec{v}_{t+1}$ 的矢量 $\vec{u}_{t+1} = \vec{v}_{t+1} - \vec{q}_{t+1}$,其中, $\vec{q}_{t+1}$ 为t+1手指对应的指根位置向量;

[0032] 则基于矢量 $\vec{u}_t$ 和 $\vec{u}_{t+1}$ 计算得到指间角度 $\theta_t = \arccos\left(\frac{\vec{u}_t \cdot \vec{u}_{t+1}}{\|\vec{u}_t\| \cdot \|\vec{u}_{t+1}\|}\right)$,将指间角

度向量化,即得到指间角度向量。

[0033] 进一步地,所述凸壳CH(P)的构造方式包括:

[0034] 初始时,凸壳CH(P)为任意4个不共面的点构成的四面体,对于集合P中的点b,若b在CH(P)内,则忽略b,若b在CH(P)上,则将b添加到CH(P)的顶点集合,若b在CH(P)外,则用b对CH(P)进行遍历划分,生成新的CH(P),最终得到完整的CH(P);

[0035] 所述进行层次聚类的方式包括:

[0036] 初始化 L_0 等于CH(P),对于每一个层次 L_r ,r属于区间 $[0, k]$,且k为大于1的正整数;计算 L_r 中所有点对 (v_p, v_q) 的欧几里得距离 $d(v_p, v_q)$,提取最大的欧几里得距离对应的点对 (p_i, p_j) ,将 (p_i, p_j) 合并为一个新的簇,移除 p_i 和 p_j ,将剩余的点和新的簇组成新的点集 L_{r+1} ;重复直至 L_{r+1} 为空集或只包含一个簇,即完成了对完整的CH(P)的层次聚类。

[0037] 进一步地,所述手势识别模型的训练方式包括:

[0038] 收集M组手势动作的手势特征向量序列及其对应的手势类型的编码,构成训练数据集;定义时间卷积网络作为手势识别模型的骨干网络,时间卷积网络由h个残差时间卷积块组成,每个残差时间卷积块包括时间卷积层和dropout层,并在时间卷积层和dropout层中加入残差连接;

[0039] 时间卷积层的计算公式为:

[0040] $y_s = \sum_{i=0}^{k-1} f(x_{s-i}) \Theta w_i + B$;其中, y_s 为时间卷积层在时间步s时的输出特征,k为

卷积核的大小,i为卷积核在时间维度上的偏移量, x_{s-i} 为第s-i个时间步的输入特征, Θ 表示卷积操作, w_i 为卷积核系数,B为偏置项,f()为ReLU函数, $f(x_{s-i})$ 是将输入特征 x_{s-i} 通过ReLU函数进行变换后的结果;

[0041] 最后一个残差时间卷积块的输出是上下文特征序列,将骨干网络添加一个分类头,将上下文特征序列输入分类头,对上下文特征序列中每个时间步的特征使用全连接层

进行分类,得到手势类型的概率分布;

[0042] 进行分类的公式为: $P(y_s | c_s) = \text{Softmax}(W_{q1} \cdot c_s + B_{q1})$;其中, W_{q1} 为全连接层的权重, B_{q1} 为全连接层的偏置, c_s 为上下文特征序列在时间步 s 处的特征;

[0043] 定义手势识别模型的损失函数 $Loss = -\frac{1}{S} \sum_{s=1}^S \log P(\hat{y}_s | c_s)$;其中, S 为上下文特征序列的总长度, $\hat{y}_s$ 为第 s 个时间步的真实的手势类型的编码;

[0044] 利用训练数据集训练手势识别模型,过程中最小化损失函数,并不断优化手势识别模型的参数,得到训练完成的手势识别模型。

[0045] 进一步地,所述将用户当前交互指令数据进行预处理的方式包括:

[0046] 将用户当前交互指令数据中不同模态的数据进行归一化处理,得到归一化指令数据;

[0047] 将归一化指令数据进行时间序列上的对齐,得到对应的多模态序列;将多模态序列在时间维度上拼接,形成一个具有 g 个特征通道的特征时间序列数据;

[0048] 将特征时间序列数据构建为一个三维张量,其中两个维度表示时间和特征通道,第三个维度用于表示对应的数据。

[0049] 进一步地,所述基于深度强化学习的时间预测模型的构建方式包括:

[0050] 时间预测模型包括序列建模网络和强化学习网络;将门控循环单元网络作为序列建模网络的基础框架,门控循环单元网络包括网络输入层、GRU隐藏层和状态表示输出层;

[0051] 收集 $N1$ 组的交互序列数据,每个交互序列数据包含时间步特征表示和对应的标签;

[0052] 将交互序列数据按照批次逐步输入至门控循环单元网络得到对应批次的最终状态向量,将对应批次的最终状态向量输入到预设的分类器中得到预测的标签概率的分布;计算标签概率和标签之间的交叉熵损失,对门控循环单元网络和分类器的可训练参数使用优化器进行梯度更新,并最小化交叉熵损失,得到序列建模网络;

[0053] 定义强化学习网络为一个半马尔可夫决策过程,为半马尔可夫决策过程定义一个元组 $(S1, N2, P1, R1)$;

[0054] 将半马尔可夫决策过程划分为高层管理器的策略和低层管理器的策略;定义半马尔可夫决策过程的目标是学习一个最优的高层管理器的策略和 Q 个最优的低层管理器的策略,将交互序列数据通过门控循环单元网络获取的对应批次的最终状态向量,作为半马尔可夫决策过程的状态,训练半马尔可夫决策过程,并使用策略梯度算法分别优化高层管理器的策略和低层管理器的策略;使得在半马尔可夫决策过程中计算的期望累积奖赏最大化;得到构建完成的强化学习网络。

[0055] 进一步地,所述概率图模型的构建方式包括:

[0056] 定义概率图模型的随机变量,随机变量为相关环境数据中的数据;根据虚拟场景中各个物体之间的相互影响关系,构建无向图结构;无向图结构中的节点表示随机变量,无向图结构中的边表示随机变量之间的相互依赖关系;为每个节点定义一个节点潜在函数,为每条边定义一个边潜在函数;

[0057] 收集 $M1$ 组的历史数据,历史数据为虚拟场景状态的时间序列数据;初始化节点潜

在函数和边潜在函数的参数,利用虚拟场景状态的时间序列数据对概率图模型进行训练,得到节点潜在函数和边潜在函数的参数;训练的方法为最大似然估计方法;即完成了概率图模型的构建;

[0058] 所述未来n个虚拟场景状态的获取方式包括:

[0059] 定义概率图模型的转移概率,基于概率图模型的转移概率,对未来n个时间步长进行推理,得到每个时间步长下虚拟场景各个物体的概率分布,在进行推理过程中,将用户交互序列按时间步长依次输入概率图模型;

[0060] 对于每个时间步长,根据概率分布采样或选取场景物体的状态,构建出该时间步长下的虚拟场景状态,继而得到未来n个虚拟场景状态。

[0061] 一种虚拟场景交互系统,用于实现所述的一种虚拟场景交互方法,包括:数据采集处理模块,用于获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列;

[0062] 预测模块,用于将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;

[0063] 虚拟环境调控模块,用于根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来n个虚拟场景状态;基于未来n个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备,各个模块之间通过有线和/或无线的方式进行连接。

[0064] 本发明一种虚拟场景交互方法和系统的技术效果和优点:

[0065] 本发明能够融合多种用户交互数据,如按键、手势、眼球运动等,特别的对于虚拟交互来说手势数据的准确性较为重要,全面捕捉用户的交互意图,从而更准确地预测用户的行为,提升了交互的智能性,其次,采用基于深度强化学习的时间预测模型,能够提前预测用户未来的交互行为序列,并驱动概率图模型生成未来虚拟场景的状态变化,大大增强了交互的流畅性和自然性,此外,还利用概率图模型建模虚拟场景的上下文环境,如物体位置、光照条件等,更好地预测场景的变化,从而进一步提高了交互的准确性,总的来说,通过多模态融合、时间预测建模和概率图建模,大幅提升了用户的交互体验。

附图说明

[0066] 图1为本发明的一种虚拟场景交互方法示意图;

[0067] 图2为本发明的一种虚拟场景交互系统示意图。

具体实施方式

[0068] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。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。

[0069] 实施例1

[0070] 请参阅图1所示,本实施例一种虚拟场景交互方法,包括:

[0071] S1、获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行

预处理,得到融合交互特征序列;

[0072] S2、将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;

[0073] S3、根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来 n 个虚拟场景状态;基于未来 n 个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备。

[0074] 进一步的,用户当前交互指令数据包括用户输入设备操作数据和生物特征数据;

[0075] 用户输入设备操作数据包括按键数据、手势数据和设备位移数据;生物特征数据包括眼球运动数据和肢体运动数据;

[0076] 相关环境数据包括用户所处虚拟位置数据、视角方向数据和用户所处环境状态数据;用户所处环境状态数据包括场景物体位置数据和光照数据;

[0077] 按键数据包括按键事件的时间戳、按键编码和扫描码,需要说明的是,时间戳由输入设备的操作系统内核或驱动程序在收到硬件中断时获取,例如,当用户按下按键时,键盘硬件会触发一个中断信号,操作系统内核或键盘驱动程序收到该中断,内核或驱动程序调用系统时间API,获取当前的系统时间,精确到微秒或纳秒级别,获取到的时间值就是按键事件发生的时间戳。

[0078] 按键编码由输入设备的键盘驱动程序根据扫描码查询按键映射表获得;扫描码来自输入设备的键盘硬件电路,具体的,键盘硬件电路由多个按键矩阵组成,每个按键对应矩阵中的一个行列交点,当按下一个按键时,对应的行列交点会形成电路通路,键盘硬件电路检测到这个通路,并根据行列位置生成一个硬件扫描码,即扫描码。

[0079] 时间戳、按键编码和扫描码结合起来就完整描述了一个按键事件,例如,按下“A”键时,记录数据[12:30:45.782,65,30]。

[0080] 手势数据的获取方式包括:

[0081] 采用深度相机以固定的帧率(如30fps)持续采集用户手部区域的点云数据,构成点云数据序列;深度相机采集的手部区域的点云数据不是仅仅包含手部还存在一些背景区域;

[0082] 从点云数据序列中提取手部精确点云数据,将手部区域点云数据进行降采样,减少数据量,提高后续处理效率,得到完整手部点云数据。

[0083] 进行降采样的方式包括:

[0084] 将点云数据序列有序地插入到预先构建完成的八叉树中,然后以固定的空间分辨率对八叉树进行遍历,得到完整手部点云数据。

[0085] 八叉树的构建方式包括:

[0086] 将点云数据序列所在的三维空间划分为规则的立方体单元,根据点云数据序列中点的坐标,将每个点插入到对应的立方体单元中,对于包含 m 个点的立方体单元,将其继续递归划分为8个子单元,直到每个立方体单元内最多只有一个点,即初步完成八叉树的构建。

[0087] 对八叉树进行遍历的方式包括:

[0088] 设置一个距离阈值 d ,表示相邻两个采样的点之间的最小距离, d 的值越大,采样点越稀疏, d 的值越小,采样点越密集;

[0089] 从八叉树的根节点开始,按照深度优先或广度优先的遍历八叉树的每个节点,对于每个非空的节点(包含点云点),计算节点中所有点到节点中心的距离,选取距离节点中心最近的点作为该节点的代表点。

[0090] 在八叉树中,深度优先为从根节点开始,选择一个子节点,继续向下遍历该子节点的子节点,直到遇到叶子节点或空节点,然后回溯到上一层节点,继续遍历其他子节点。

[0091] 在八叉树中,广度优先是从根节点开始,先访问根节点,然后依次访问所有第一层节点,再访问所有第二层节点,以此类推。

[0092] 节点中心指的是该立方体空间区域的几何中心,在遍历八叉树进行降采样时,对于每个非空节点(包含点云点),需要计算该节点中所有点云点到节点中心的距离,然后选取距离最近的那个点云点作为该节点的代表点;这样做的目的是为了在降采样时,保留对该区域最有代表性的点云点。

[0093] 初始化一个空的列表L,用于存储采样点,遍历八叉树的每个非空的节点X,计算X的中心点c和代表点p之间的距离 $d(c, p)$,若 $d(c, p)$ 大于或等于d,则将p加入列表L,最终列表L中的点即为完整手部点云数据;

[0094] 其中,距离 $d(c, p)$ 的计算公式为:

[0095] $d(c, p) = \sqrt{(c_x - p_x)^2 + (c_y - p_y)^2 + (c_z - p_z)^2}$;其中, (c_x, c_y, c_z) 为中心点c的三维坐标, (p_x, p_y, p_z) 为代表点p的三维坐标。

[0096] 将完整手部点云数据进行特征提取,得到手掌法向量、指尖位置向量、指根位置向量和指间角度向量;将手掌法向量、指尖位置向量、指根位置向量和指间角度向量拼接得到手势特征向量。

[0097] 进行特征提取的方式包括:

[0098] 将完整手部点云数据进行主成分分析(PCA),得到主方向向量 $\vec{v}_1$,手掌法向量 $\vec{n}$ 为 $\vec{v}_1$ 与手部所在的三维坐标系的z轴负方向的矢量积,即 $\vec{n} = \vec{v}_1 \times (-\vec{k})$,其中, $\vec{k}$ 为z轴的单位向量(0,0,1)。

[0099] 将完整手部点云数据构造成集合P,基于集合P构造完整的凸壳CH(P);

[0100] 凸壳CH(P)的构造方式包括:

[0101] 初始时,凸壳CH(P)为任意4个不共面的点构成的四面体,对于集合P中的点b,若b在CH(P)内,则忽略b,若b在CH(P)上,则将b添加到CH(P)的顶点集合,若b在CH(P)外,则用b对CH(P)进行遍历划分,生成新的CH(P),最终得到完整的CH(P)。

[0102] 对完整的CH(P)进行层次聚类,得到k+1个层次,记作 $L_0, L_1, \dots, L_{k-1}, L_k$;其中, L_{k-1} 即为指尖点集;

[0103] 对 L_{k-1} 中的每个点 v_j ,遍历CH(P)上的所有顶点,计算 v_j 到所有顶点的欧几里得距离并取最小值 d_j ,计算完所有 v_j 的 d_j 后,取 d_j 的最小值对应的点,即为指尖位置,将指尖位置进行向量化,即得到指尖位置向量 $\vec{v}_t$;其中,t为手指的索引。

[0104] 进行层次聚类的方式包括:

[0105] 初始化 L_0 等于CH(P),对于每一个层次 L_r ,r属于区间 $[0, k]$,且k为大于1的正整数;

计算Lr中所有点对(vp,vq)的欧几里得距离d(vp,vq),提取最大的欧几里得距离对应的点对(p i,pj),将(p i,pj)合并为一个新的簇,移除pi和pj,将剩余的点和新的簇组成新的点集Lr+1;重复直至Lr+1为空集或只包含一个簇,即完成了对完整的CH(P)的层次聚类。

[0106] 计算每个指尖位置向量 $\vec{v}_t$ 到手掌平面的距离 d_t ,沿手掌平面的法线 $\vec{N}$ 方向从 $\vec{v}_t$ 往回偏移 α 得到对应指根位置向量 $\vec{q}_t = \vec{v}_t - \alpha \cdot d_t \cdot \vec{N}$;其中, α 为经验系数,通常是0到1之间的一个小数,它反映了手指长度占整个手部长度的比例。

[0107] 对于任意两相邻的指尖位置向量 $\vec{v}_t$ 和 $\vec{v}_{t+1}$,计算 $\vec{v}_t$ 的矢量 $\vec{u}_t = \vec{v}_t - \vec{q}_t$ 和 $\vec{v}_{t+1}$ 的矢量 $\vec{u}_{t+1} = \vec{v}_{t+1} - \vec{q}_{t+1}$,其中, $\vec{q}_{t+1}$ 为t+1手指对应的指根位置向量;

[0108] 则基于矢量 $\vec{u}_t$ 和 $\vec{u}_{t+1}$ 计算得到指间角度 $\theta_t = \arccos\left(\frac{\vec{u}_t \cdot \vec{u}_{t+1}}{\|\vec{u}_t\| \cdot \|\vec{u}_{t+1}\|}\right)$,将指间角

度向量化,即得到指间角度向量。

[0109] 将获得的手势特征向量按照时间的先后排列成一个手势特征向量序列,将手势特征向量序列输入至预先训练完成的手势识别模型中,输出对应的手势类型的概率分布,取概率分布中概率最大的手势类型的编码作为手势数据,如编码6代表“挥手”手势。

[0110] 手势识别模型的训练方式包括:

[0111] 收集M组手势动作的手势特征向量序列及其对应的手势类型的编码,构成训练数据集;定义时间卷积网络作为手势识别模型的骨干网络,时间卷积网络由h个残差时间卷积块组成,每个残差时间卷积块包括时间卷积层和dropout层,并在时间卷积层和dropout层中加入残差连接,加入dropout层防止过拟合,加入残差连接,有助于梯度传播。

[0112] 时间卷积层的计算公式为:

[0113]
$$y_s = \sum_{i=0}^{k-1} f(x_{s-i}) \Theta w_i + B$$
 其中, y_s 为时间卷积层在时间步s时的输出

特征,k为卷积核的大小,i为卷积核在时间维度上的偏移量,范围是0到k-1, x_{s-i} 为第s-i个时间步的输入特征, Θ 表示卷积操作, w_i 为卷积核系数,B为偏置项,f()为ReLU函数,f(x_{s-i})是将输入特征 x_{s-i} 通过ReLU函数进行变换后的结果;

[0114] 最后一个残差时间卷积块的输出是上下文特征序列,将骨干网络添加一个分类头,将上下文特征序列输入分类头,对上下文特征序列中每个时间步的特征使用全连接层进行分类,得到手势类型的概率分布。

[0115] 进行分类的公式为: $P(y_s | c_s) = \text{Soft max}(W_{q1} \cdot c_s + B_{q1})$;其中, W_{q1} 为全连接层的权重, B_{q1} 为全连接层的偏置, c_s 为上下文特征序列在时间步s处的特征;

[0116] 定义手势识别模型的损失函数 $Loss = -\frac{1}{S} \sum_{s=1}^S \log P(\hat{y}_s | c_s)$;其中,S为上下

文特征序列的总长度,即时间步的总数, $\hat{y}_s$ 为第s个时间步的真实的手势类型的编码;

[0117] 利用训练数据集训练手势识别模型,过程中最小化损失函数,并不断优化手势识

别模型的参数,得到训练完成的手势识别模型。

[0118] 通过设备内置的加速度传感器或陀螺仪传感器检测设备的运动状态,运动状态包括位移和旋转,通过融合数据,获取设备位移数据。

[0119] 眼球运动数据通过设备的前置摄像头结合眼球追踪算法来获取,眼球追踪算法包括基于图像处理的算法或基于3D模型的算法,图像处理的算法或基于3D模型的算法可以实时检测用户的眼部运动轨迹。

[0120] 肢体运动数据通过设备的深度摄像头或RGB摄像头,结合骨骼跟踪算法来获取;骨骼跟踪算法基于机器学习模型,从图像或视频中检测并跟踪人体关键点,从而获取肢体的运动轨迹数据。

[0121] 相关环境数据通过虚拟场景的三维模型数据获取,具体的,在构建虚拟场景时,对场景中的人物和各个物体进行三维建模,并记录它们的精确位置坐标、用户所处虚拟位置数据和视角方向数据,通过设置虚拟场景的光照参数获取光照数据。

[0122] 需要说明的是,将上述数据均进行了量化,使得适应于后续的数据处理。

[0123] 进一步的,将用户当前交互指令数据进行预处理的方式包括:

[0124] 将用户当前交互指令数据中不同模态的数据进行归一化处理,得到归一化指令数据,使其数值范围处于相同的量级;归一化方法包括Min-Max归一化和Z-Score归一化。

[0125] 将归一化指令数据进行时间序列上的对齐,使得不同模态在同一时间点都有对应的数据,得到对应的多模态序列,使用插值、上采样或下采样实现对齐。

[0126] 将多模态序列在时间维度上拼接,形成一个具有g个特征通道的特征时间序列数据;例如,按键数据、手势数据、眼球运动数据并列形成多个特征通道,这种拼接融合方式保留了每个模态的原始时间动态,易于后续建模;

[0127] 将特征时间序列数据构建为一个三维张量,其中两个维度表示时间和特征通道,第三个维度用于表示对应的数据;便于输入到能够处理多通道数据的模型中。

[0128] 将已量化的多模态的数据进行有效融合,为后续的特征提取和建模过程做好数据准备。

[0129] 进一步的,基于深度强化学习的时间预测模型的构建方式包括:

[0130] 定义时间预测模型包括序列建模网络和强化学习网络;将门控循环单元网络作为序列建模网络的基础框架,门控循环单元网络包括网络输入层、GRU隐藏层和状态表示输出层。

[0131] 收集N1组的交互序列数据,每个交互序列数据包含时间步特征表示和对应的标签(用户的最终行为);

[0132] 将交互序列数据按照批次逐步输入至门控循环单元网络得到对应批次的最终状态向量,将对应批次的最终状态向量输入到预设的分类器中(全连接层+softmax函数)得到预测的标签概率的分布;计算标签概率和标签之间的交叉熵损失,对门控循环单元网络和分类器的可训练参数使用优化器(如Adam)进行梯度更新,并最小化交叉熵损失,得到训练完成的门控循环单元网络。

[0133] 网络输入层用于接收融合交互特征序列作为输入,融合交互特征序列是一个长度为Y的向量序列,向量序列中的每个向量对应一个时间步;

[0134] GRU隐藏层由F个GRU单元组成,每个GRU单元对应处理向量序列中的一个时间步;

对于第 a 个时间步,利用当前时间步的输入向量和上一时间步的隐状态向量,计算出一个更新门控制器向量,更新门控制器向量的每个元素值在0到1之间,表示对应的隐状态元素应当保留来自上一状态的信息的程度。

[0135] 同样利用当前时间步的输入向量和上一时间步的隐状态向量,计算出一个重置门控制器向量;重置门控制器向量的每个元素值在0到1之间,表示对应的隐状态元素应当舍弃来自上一状态的信息的程度;

[0136] 根据当前输入向量、重置门控制器向量和上一时间步的隐状态向量,计算出一个候选新隐状态向量;候选新隐状态向量综合了来自当前输入和上一状态的信息。

[0137] 利用更新门控制器向量、上一时间步的隐状态向量和候选新隐状态向量,计算出当前时间步的最终隐状态向量;最终隐状态向量是对上一隐状态和候选新隐状态的融合,由更新门控制器决定两者的融合比例。

[0138] 状态表示输出层经过 Y 个时间步的递归计算,得到最后一个时间步的隐状态向量,将该向量作为整个融合交互特征序列的编码表示,即得到最终状态向量。

[0139] 需要解释的是,隐状态向量是GRU单元在每个时间步保存的一个向量,用于编码来自之前时间步的输入序列的信息,具体来说,在第 a 个时间步,GRU单元会根据当前时间步的输入向量 $x(a)$ 和上一时间步的隐状态向量 $h(a-1)$ 来计算更新门控制器向量 $z(a)$,更新门控制器向量每个元素的值在0到1之间,用于控制该时间步的隐状态向量 $h(a)$ 应当保留来自上一隐状态 $h(a-1)$ 的信息的程度,同样重置门控制器向量、候选新隐状态向量和最终状态向量亦是如此获取。

[0140] 更新门控制器向量 $z(a)$ 是上一隐状态 $h(a-1)$ 和当前输入 $x(a)$ 的函数,通过学习得到的权重矩阵 $W(z)$ 和偏置 $b(z)$ 对它们进行affine变换,再通过sigmoid激活函数得到0到1之间的值;

[0141] $z(a)$ 的每个元素值接近0,表示该元素在新的隐状态 $h(a)$ 中应当较多保留来自上一隐状态 $h(a-1)$ 的信息;反之,值接近1,表示该元素应当较多融入来自当前输入 $x(a)$ 的新信息。

[0142] 所以,隐状态向量 $h(a-1)$ 不仅携带着之前时间步的序列信息,还通过更新门控制器向量 $z(a)$ 控制了新的隐状态 $h(a)$ 应当如何更新,以此来编码整个输入序列的动态信息。

[0143] 强化学习网络的构建方式包括:

[0144] 定义强化学习网络为一个半马尔可夫决策过程,为半马尔可夫决策过程定义一个元组 $(S1, N2, P1, R1)$;其中, $S1$ 是低层管理器的决策的状态空间, $N2$ 是高层管理器可选择的子任务空间, $P1$ 是低层管理器的决策的状态转移函数, $P(s' | s1, n1, \pi_n)$ 表示在状态 $s1$ 下,执行子任务 $n1$ 对应的策略 π_n 后,转移到状态 s' 的概率, $R1$ 是低层管理器的决策的奖赏函数, $R(s1, n1, \pi_n)$ 表示在状态 $s1$ 下,执行子任务 $n1$ 对应的策略 π_n 所获得的奖赏。

[0145] 将半马尔可夫决策过程划分为高层管理器的策略和低层管理器的策略;高层管理器的策略 $\pi_m(n1 | s1)$ 表示在状态 $s1$ 下选择子任务 $n1$ 的概率,低层管理器的策略 $\pi_n(a1 | s1)$ 表示在状态 $s1$ 下,执行子任务 $n1$ 时选择动作 $a1$ 的概率。

[0146] 定义半马尔可夫决策过程的目标是学习一个最优的高层管理器的策略和 Q 个最优的低层管理器的策略,将交互序列数据通过门控循环单元网络获取的对应批次的最终状态向量,作为半马尔可夫决策过程的状态,训练半马尔可夫决策过程,并使用策略梯度算法分

别优化高层管理器的策略和低层管理器的策略;使得在半马尔可夫决策过程中获得的期望累积奖赏最大化;得到构建完成的强化学习网络。

[0147] 期望累积奖赏的计算公式为:

[0148] $J(\pi_m, \{\pi_n\}) = E_{\tau \sim \pi_m, \{\pi_n\}} [\sum_{cd=1}^{CD} \gamma^{cd} \cdot R(s_{cd}, n_{cd}, \pi_{n_{cd}})]$; 其中, $J(\pi_m, \{\pi_n\})$ 表示在给定高层管理器的策略 π_m 和低层管理器的策略构成的集合 $\{\pi_n\}$ 时, 期望获得的累积奖赏, 是需最大化的目标函数; $E_{\tau \sim \pi_m, \{\pi_n\}} [\]$ 表示对轨迹 τ 取期望值, 其中轨迹 τ 是根据高层管理器的策略 π_m 和低层管理器的策略构成的集合 $\{\pi_n\}$ 生成, CD 为最终状态向量的时间步的数量, cd 为最终状态向量的时间步索引, γ^{cd} 为时间步 cd 时的折现因子, $R(s_{cd}, n_{cd}, \pi_{n_{cd}})$ 表示在状态 s_{cd} 下, 执行子任务 n_{cd} 对应的低层管理器的策略 $\pi_{n_{cd}}$ 所获得的即时奖赏, s_{cd} 为时间步 cd 时的状态, n_{cd} 表示时间步 cd 执行的子任务, 由高层管理器的策略选择, $\pi_{n_{cd}}$ 表示执行子任务 n_{cd} 时采用的低层管理器的策略。

[0149] 需要说明的是, 在半马尔可夫决策过程中, 存在高层管理器 (高级策略) 和低层管理器 (低级策略) 两个层次, 高层管理器决定选择执行哪个子任务, 而低层管理器则决定如何执行该子任务 (执行哪些具体动作); 轨迹的生成方式包括:

[0150] 初始化初始的状态 s_0 , 该状态由门控循环单元网络获取的最终状态向量表示; 根据当前状态和高层管理器的策略选择一个子任务 nt 。具体做法是从策略的概率分布中采样一个子任务; 根据当前状态和选择的子任务 nt , 利用对应的低层管理器的策略执行该子任务, 即根据低层管理器的策略的概率分布连续采样动作序列, 直到子任务完成或达到最大步数限制; 执行子任务过程中, 记录所经历的状态序列、执行的动作序列, 以及获得的即时奖励; 子任务执行完毕后, 根据状态转移函数, 从当前状态转移到下一状态; 重复直到达到终止条件 (如最大轨迹长度); 通过以上过程, 就生成了一条完整的轨迹 τ , 包含了状态序列、子任务序列、动作序列和奖励序列。

[0151] 将用户交互过程分解为多个子任务, 由高层管理器选择当前要执行哪个子任务, 而低层管理器则执行对应子任务的具体动作序列;

[0152] 分层的方式, 可以更好地捕捉交互过程中的层次结构, 提高样本效率, 高层管理器关注更高层次的子任务选择, 而低层管理器则专注于具体的动作执行, 两者相互协作。

[0153] 具体来说, 时间预测模型的输入是融合交互特征序列, 先通过序列建模网络获得当前状态的表示, 然后基于强化学习网络输出下一时间步的预测动作 (即预测的用户交互), 执行该动作会产生新的状态表示, 以此类推, 直至预测出整个未来时间段的交互序列。

[0154] 在训练过程中, 通过最大化与真实交互序列的拟合度, 时间预测模型不断调整参数, 提高预测的准确性。

[0155] 进一步的, 概率图模型的构建方式包括:

[0156] 定义概率图模型的随机变量, 随机变量为相关环境数据中的数据; 例如物体位置、光照条件、用户位置和视角等, 每个随机变量取有限个离散状态, 或者连续状态; 根据虚拟场景中各个物体之间的相互影响关系, 构建无向图结构; 无向图结构中的节点表示随机变量, 无向图结构中的边表示随机变量之间的相互依赖关系, 通常近邻的节点之间存在边连

接,远距离节点之间可能没有直接连接;

[0157] 为每个节点定义一个节点潜在函数,用于描述该节点取不同状态的概率分布,节点潜在函数是参数化的函数,例如高斯函数或逻辑斯蒂函数;为每条边定义一个边潜在函数,用于描述相邻节点之间的相互作用;边潜在函数依赖于相邻节点的状态,用编码节点之间的兼容性或不兼容性拟合;

[0158] 收集M1组的历史数据,历史数据包括虚拟场景状态的时间序列数据和用户交互行为的时间序列数据,

[0159] 初始化节点潜在函数和边潜在函数的参数,利用虚拟场景状态的时间序列数据对概率图模型进行训练,得到节点潜在函数和边潜在函数的参数;训练的方法为最大似然估计方法;即完成了概率图模型的构建。

[0160] 未来n个虚拟场景状态的获取方式包括:

[0161] 将用户交互序列作为概率图模型的控制输入,用于驱动场景的变化,并定义概率图模型的转移概率,基于概率图模型的转移概率,对未来n个时间步长进行推理,得到每个时间步长下虚拟场景各个物体的概率分布,在进行推理过程中,将用户交互序列按时间步长依次输入,模拟用户在虚拟场景中的交互行为;

[0162] 对于每个时间步长,根据概率分布采样或选取场景物体的状态,构建出该时间步长下的虚拟场景状态,继而得到未来n个虚拟场景状态;场景状态包括物体位置、光照条件、用户位置和视角等多个方面的信息;

[0163] 需要说明的是,转移概率描述了在给定当前状态和用户交互行为的条件下,场景转移到下一状态的概率分布,具体来说,当前时间步为 e ,用户的交互行为表示为 $u(e)$,将 $u(e)$ 作为概率图模型的输入,它将影响各个节点的转移概率,在时间步 e ,概率图模型的联合概率分布表示为节点潜在函数和边潜在函数的乘积,边潜在函数与用户交互 $u(e)$ 相关,接着,定义从时间步 e 转移到时间步 $e+1$ 的转移概率,等于每个节点根据其邻居节点状态和用户交互 $u(t)$ 转移到下一状态的条件概率的乘积,单个节点的转移概率通过学习获得,包含了用户交互对该节点转移的影响。

[0164] 定义了概率图模型的转移概率后,在推理过程中,利用该转移概率模型,结合用户的交互序列作为输入,对未来n个时间步长进行滚动推理,在每个时间步长,得到虚拟场景中各个物体状态的概率分布,再通过采样等方式从概率分布中选取物体的具体状态,就可以构建出该时间步长下的完整虚拟场景状态。

[0165] 虚拟场景画面的渲染方式包括:

[0166] 从未来n个虚拟场景状态中解析出每个时刻场景的相关环境数据,根据解析出的相关环境数据,加载所需的三维模型资源,利用加载的三维模型资源构建三维虚拟场景;具体的渲染引擎、编解码器和传输协议而有所调整,同时,考虑实时性、低延迟、带宽利用率等因素,以确保虚拟场景的流畅呈现,高效、逼真地渲染为对应的虚拟场景画面,为用户提供比较好的交互体验。

[0167] 本实施例,能够融合多种用户交互数据,如按键、手势、眼球运动等,特别的对于虚拟交互来说手势数据的准确性较为重要,全面捕捉用户的交互意图,从而更准确地预测用户的行为,提升了交互的智能性,其次,采用基于深度强化学习的时间预测模型,能够提前预测用户未来的交互行为序列,并驱动概率图模型生成未来虚拟场景的状态变化,大大增

强了交互的流畅性和自然性,此外,还利用概率图模型建模虚拟场景的上下文环境,如物体位置、光照条件等,更好地预测场景的变化,从而进一步提高了交互的准确性,总的来说,通过多模态融合、时间预测建模和概率图建模,大幅提升了用户的交互体验。

[0168] 实施例2

[0169] 请参阅图2所示,本实施例未详细叙述部分见实施例1描述内容,提供一种虚拟场景交互系统,包括:数据采集处理模块,用于获取用户当前交互指令数据和相关环境数据,将用户当前交互指令数据进行预处理,得到融合交互特征序列;

[0170] 预测模块,用于将融合交互特征序列输入至预先构建完成的基于深度强化学习的时间预测模型,得到一个未来时间段内的用户交互序列;

[0171] 虚拟环境调控模块,用于根据相关环境数据和用户交互序列,利用预先构建完成的概率图模型,得到未来n个虚拟场景状态;基于未来n个虚拟场景状态渲染出对应的虚拟场景画面,将虚拟场景画面传输至用户端设备,各个模块之间通过有线和/或无线的方式进行连接,实现模块间的数据传输。

[0172] 实施例3

[0173] 本实施例公开提供了一种电子设备,包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,处理器执行计算机程序时实现上述所提供的一种虚拟场景交互方法的运行方式。

[0174] 由于本实施例所介绍的电子设备为实施本申请实施例中一种虚拟场景交互方法所采用的电子设备,故而基于本申请实施例中所介绍的一种虚拟场景交互方法,本领域所属技术人员能够了解本实施例的电子设备的实施方式以及其各种变化形式,所以在此对于该电子设备如何实现本申请实施例中的方法不再详细介绍。只要本领域所属技术人员实施本申请实施例中一种虚拟场景交互方法所采用的电子设备,都属于本申请所欲保护的范围。

[0175] 上述公式均是去量纲取其数值计算,公式是由采集大量数据进行软件模拟得到最近真实情况的一个公式,公式中的预设参数以及阈值选取由本领域的技术人员根据实际情况进行设置。

[0176] 以上所述仅是本发明的优选实施方式,本发明的保护范围并不仅限于上述实施例,凡属于本发明思路下的技术方案均属于本发明的保护范围。应当指出,对于本技术领域的普通技术用户来说,在不脱离本发明原理前提下的若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

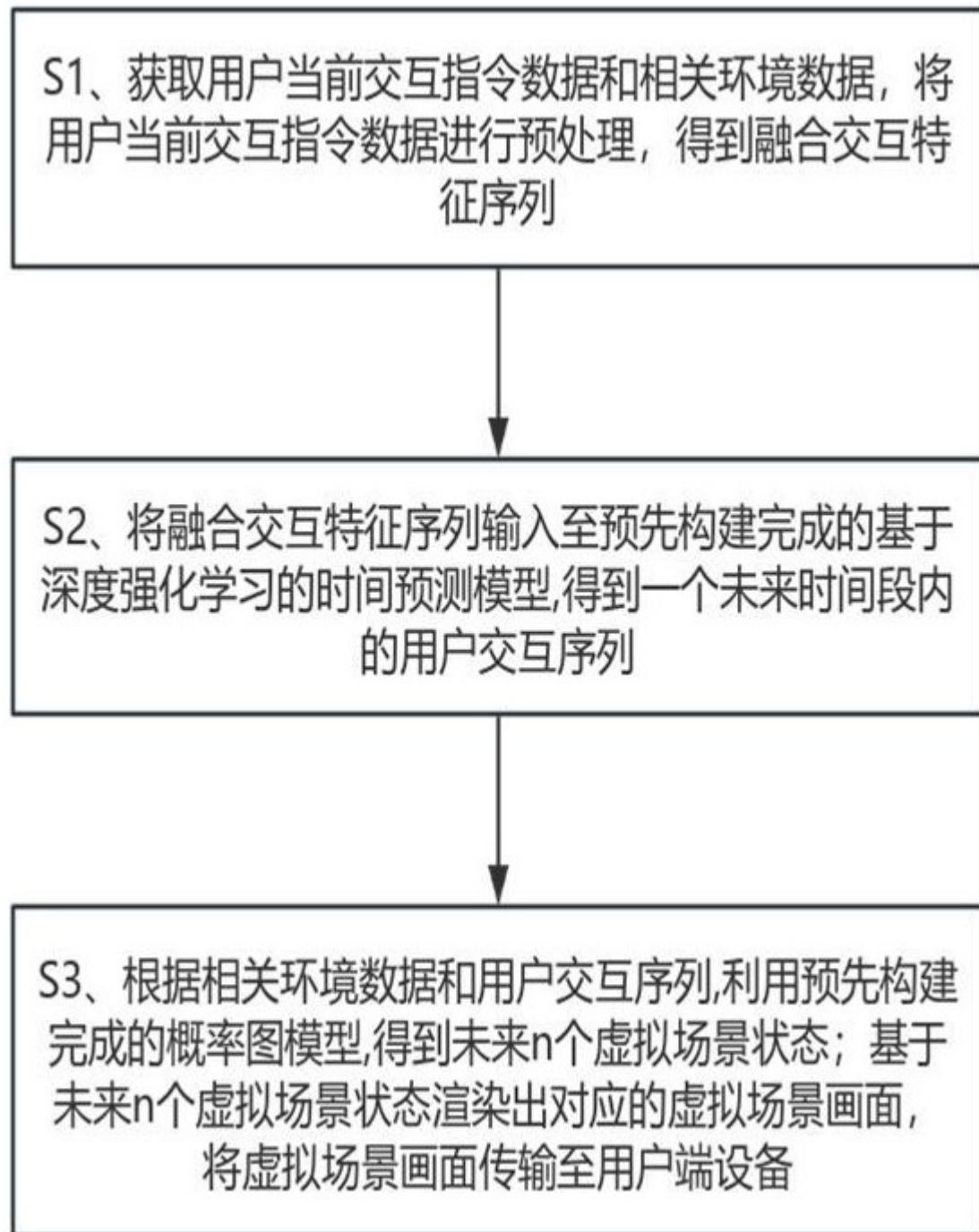


图 1

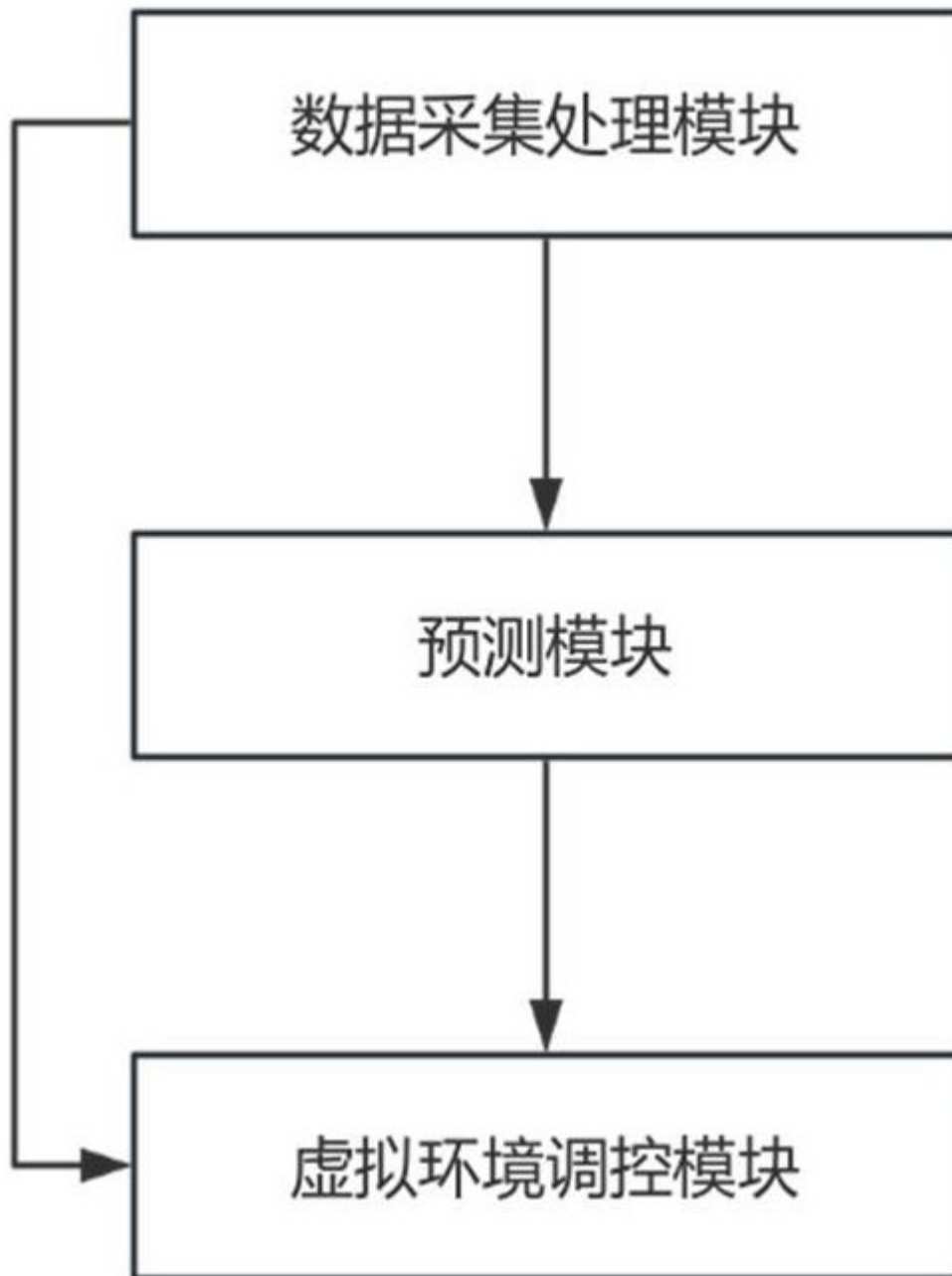


图 2