



(12) 发明专利申请



(10) 申请公布号 CN 118963538 A

(43) 申请公布日 2024. 11. 15

(21) 申请号 202410839877.3

(22) 申请日 2024.06.26

(71) 申请人 北京工业大学

地址 100020 北京市朝阳区平乐园100号

(72) 发明人 赵雪锋 厉望秉 杨千太 刘思雨

(74) 专利代理机构 北京思海天达知识产权代理有限公司 11203

专利代理师 王兆波

(51) Int. Cl.

G06F 3/01 (2006.01)

G06F 3/0346 (2013.01)

权利要求书3页 说明书6页 附图9页

(54) 发明名称

一种混合现实环境下虚拟物体位姿控制方法

(57) 摘要

本发明公开了一种混合现实环境下虚拟物体位姿控制方法,具体包括以下五个步骤:混合现实手势数据获取与处理;虚拟物体空间位姿灵敏度调节;虚拟物体空间位姿自由度控制;虚拟物体空间位姿坐标系映射;封装、测试与发布。本发明提出并引入敏感度参数,灵活调整虚拟物体对于手势移动的响应程度,实现空间姿态的精细控制;同时提出虚拟物体轴向自由度独立控制方法,实现空间位置的准确控制;此外,提出虚拟物体坐标系参照转换方法,实现空间位姿的按需控制。本发明能够满足不同场景下虚拟物体操作的应用需求,减轻用户学习负担,提升人机交互体验,为混合现实、虚拟现实和人机交互领域的虚拟物体位姿控制提供更加高效实用的方法。

混合现实手势数据获取与处理

虚拟物体空间位姿灵敏度调节

虚拟物体空间位姿自由度控制

虚拟物体空间位姿坐标系映射

封装、测试与发布

1.一种混合现实环境下虚拟物体位姿控制方法,其特征在于,该方法包括如下步骤:

步骤1,混合现实手势数据获取与处理,在混合现实开发平台Unity引擎中利用Mixed Reality Toolkit即MRTK混合现实开发工具包配置混合现实开发框架,在Visual Studio中通过C#编程语言实现混合现实手势处理器注册、手势识别逻辑定义、手势更新事件监听,以实现手势交互数据接口,基于MRTK获取基于Vector3的平移手势和基于Quaternion的旋转手势交互数据,并将手势数据转换为虚拟物体空间位姿数据;

步骤2,创新提出并引入灵敏度参数,包括平移灵敏度 S_t 和旋转灵敏度 S_r ,首先进行平移灵敏度调节,按照平移灵敏度参数 S_t 对基于Vector3的空间位姿平移数据增量 $\vec{l} = (i_x, i_y)$ 进行缩放,其中 i_x 和 i_y 分别是沿着X轴和Y轴的输入量,然后将处理后的空间位姿平移增量数据与当前空间位姿平移位置数据 $\vec{v}_{initial}$ 相加,输出空间位姿平移目标位置中间数据 \vec{v}_1 ,具体的表达式如下:

$$\vec{v}_1 = \vec{v}_{initial} + S_t \cdot \vec{l}$$

进行旋转灵敏度调节,按照旋转灵敏度参数 S_r 对基于Quaternion的空间位姿旋转数据增量 R_Δ 进行缩放,需要将空间位姿旋转四元数转换为欧拉角数据 $\vec{\theta}_\Delta = (\theta_x, \theta_y, \theta_z)$,在欧拉角中乘以旋转灵敏度参数实现旋转增量的缩放 $\vec{\theta}_{scaled} = S_r \cdot \vec{\theta}_\Delta$,然后将处理后的欧拉角 $\vec{\theta}_{scaled}$ 重新转换为旋转四元数 R_{scaled} ,再基于四元数乘法将处理后的空间位姿旋转增量四元数数据与当前空间位姿旋转位置四元数数据 $R_{initial}$ 相乘,输出空间位姿旋转目标位置中间数据 R_1 ,具体的表达式如下:

$$R_1 = R_{initial} \otimes R_{scaled};$$

步骤3,虚拟物体空间位姿自由度控制,创新提出轴向自由度独立控制方法,首先进行平移自由度控制,将空间位姿平移目标位置中间数据 \vec{v}_1 向xyz三轴映射,形成 V_{1x} 、 V_{1y} 和 V_{1z} 三个自由度单独数据, $\vec{v}_1 = (V_{1x}, V_{1y}, V_{1z})$,同时将当前空间位姿平移位置数据 \vec{v}_{c1} 向xyz三轴映射,形成 V_{c1x} 、 V_{c1y} 和 V_{c1z} 三个自由度单独数据, $\vec{v}_{c1} = (V_{c1x}, V_{c1y}, V_{c1z})$,根据轴向平移自由度控制要求,在 V_{1x} 、 V_{1y} 、 V_{1z} 、 V_{c1x} 、 V_{c1y} 和 V_{c1z} 中选取相应的自由度数据,选择过程可以通过引入函数 F 来描述, F 作用于每一个轴向的平移自由度,决定采用 \vec{v}_1 或 \vec{v}_{c1} 的对应分量,定义控制向量 $\vec{F} = (F_x, F_y, F_z)$,其中每个 $F_i (i=x, y, z)$ 为1时选择 \vec{v}_{1i} ,为0时选择 \vec{v}_{c1i} ,形成新的自由度数据,并输出空间位姿平移目标位置最终数据 $\vec{v}_2 = (V_{2x}, V_{2y}, V_{2z})$,其中

$V_{2i} = F_i \cdot V_{1i} + (1 - F_i) \cdot V_{c1i}$, i 代表x、y或z,具体的表达式如下:

$$\vec{v}_2 = (F_x \cdot V_{1x} + (1 - F_x) \cdot V_{c1x}, F_y \cdot V_{1y} + (1 - F_y) \cdot V_{c1y}, F_z \cdot V_{1z} + (1 - F_z) \cdot V_{c1z})$$

其次进行旋转自由度控制,本发明创新提出一种四元数和欧拉角相结合的空间位姿旋转自由度独立控制方法,将空间位姿旋转目标位置中间数据 R_1 的四元数数据转换为空间位姿旋转目标位置中间数据的欧拉角数据 O_1 , $O_1 = \text{Euler}(R_1)$,然后将欧拉角数据 O_1 向xyz三轴映射, $O_1 = (O_{1x}, O_{1y}, O_{1z})$,形成 O_{1x} 、 O_{1y} 和 O_{1z} 三个自由度单独数据,再将当前空间位姿旋转位置四元数数据 R_{c1} 转换为欧拉角数据 O_{c1} , $O_{c1} = \text{Euler}(R_{c1})$,并向xyz三轴映射, $O_{c1} = (O_{c1x}, O_{c1y}, O_{c1z})$,

形成 O_{C1x} 、 O_{C1y} 和 O_{C1z} 三个自由度单独数据,根据轴向旋转自由度控制要求,在 O_{1x} 、 O_{1y} 、 O_{1z} 、 O_{C1x} 、 O_{C1y} 和 O_{C1z} 中选取相应的自由度数据,选择过程类似于平移自由度控制,引入一个控制向量 $\vec{F}_R = (F_{Rx}, F_{Ry}, F_{Rz})$,其中每个 $F_{Ri} (i = x, y, z)$ 决定了是选择 O_{1i} 的分量还是 O_{C1i} 的分量,形成新的空间位姿旋转目标位置最终数据的欧拉角数据 $O_2 = (O_{2x}, O_{2y}, O_{2z})$,其中 $O_{2i} = F_{Ri} \cdot O_{1i} + (1 - F_{Ri}) \cdot O_{C1i}$, i 代表x、y或z,最后输出空间位姿旋转目标位置最终数据的四元数数据 R_2 ,具体的表达式如下:

$$R_2 = \text{Quaternion } (O_2);$$

步骤4,虚拟物体空间位姿坐标系映射,本发明提出虚拟物体坐标系参照转换方法,根据坐标系参照要求,将 \vec{V}_2 和 R_2 同时向虚拟物体全局坐标或局部坐标映射,实现虚拟物体在全局坐标系下或在相对于父物体的局部坐标系,也称为构件坐标系下的空间位姿更新;

步骤5,封装、发布与测试,在Unity引擎中构建用于空间位姿控制的混合现实人机交互界面,包括平移和旋转灵敏度滑动调节按钮、自由度独立控制按钮、坐标系参照切换按钮,利用Visual Studio将项目编译为ARM64格式并连接HoloLens 2进行发布,在真实空间中佩戴HoloLens 2设备,根据人机交互界面选取不同的灵敏度、自由度、坐标轴参照进行虚拟物体空间位姿灵敏度调节测试、自由度控制测试和坐标系映射测试。

2.如权利要求1所述的混合现实环境下虚拟物体位姿控制方法,其特征在于,在步骤1中,可在Unity引擎中依次可执行如下步骤:

步骤1.1:基于MRTK的混合现实开发框架配置;

步骤1.2:手势交互数据接口实现;

步骤1.3:基于MRTK的手势交互数据获取;

步骤1.4:手势数据处理。

3.如权利要求1所述的混合现实环境下虚拟物体位姿控制方法,其特征在于,在步骤2中,依次可执行如下步骤:

步骤2.1:虚拟物体空间位姿平移灵敏度调节;

步骤2.2:虚拟物体空间位姿旋转灵敏度调节。

4.如权利要求1所述的混合现实环境下虚拟物体位姿控制方法,其特征在于,在步骤3中,依次可执行如下步骤:

步骤3.1:虚拟物体空间位姿平移自由度控制;

步骤3.2:虚拟物体空间位姿旋转自由度控制。

5.如权利要求1所述的混合现实环境下虚拟物体位姿控制方法,其特征在于,在步骤4中,依次可执行如下步骤:

步骤4.1:在Visual Studio中将 \vec{V}_2 和 R_2 同时向全局坐标或局部坐标映射;

步骤4.2:在Unity中实现虚拟物体空间位姿更新。

6.如权利要求1所述的混合现实环境下虚拟物体位姿控制方法,其特征在于,在步骤5中,依次可执行如下步骤:

步骤5.1:构建用于空间位姿控制的混合现实人机交互界面;

步骤5.2:用Visual Studio打开所保存的项目,将项目编译为ARM64格式,并发布成安装包;连接HoloLens 2,部署并调试HoloLens 2的使用环境,将HoloLens 2设置为使用

Windows设备门户,通过Wi-Fi连接HoloLens 2设备,进行项目发布;

步骤5.3:佩戴HoloLens 2设备,根据人机交互界面选取不同的灵敏度、自由度、坐标轴参照进行虚拟物体空间位姿灵敏度调节测试、自由度控制测试和坐标系映射测试。

一种混合现实环境下虚拟物体位姿控制方法

技术领域

[0001] 本发明涉及混合现实、虚拟现实和人机交互领域,特别涉及一种混合现实环境下虚拟物体位姿控制方法。

背景技术

[0002] 混合现实(MixedReality,MR)技术作为物理与数字交互领域中的新兴技术,可以结合真实和虚拟世界创造新的环境和可视化,形成一个人、计算机、环境相结合的混合现实空间,使人可以结合计算机数字信息加强对真实环境的理解;同时用户并可在该环境中将虚拟世界与现实世界进行模型交互与信息获取,实现虚拟与现实的实时互动,增强用户体验的真实感。随着元宇宙、数字孪生等概念的火爆发展,当前混合现实技术已经在教育与培训、医疗与保健、制造业和维修、建筑与土木、军事与安全、旅游与文化、娱乐与游戏等行业和领域得到了广泛且深入的应用和发展。

[0003] 混合现实环境下虚拟物体的空间位姿是影响虚拟物体在真实世界中呈现效果的关键,同时也是人机交互体验的重要影响因素。但是当前混合现实技术的虚实交互和人机交互主要基于手势展开,给虚拟物体的空间位姿控制造成了一定的挑战:1)当前混合现实环境下基于手势的虚拟物体交互逻辑,手势的微小运动会造成虚拟物体三维位姿的大幅度变化,导致在真实的物理空间中虚拟物体的空间位姿难以准确控制,造成较差的人机交互体验,同时增加用户的学习成本;2)不同应用场景对虚拟物体位姿控制要求不同,当前混合现实环境下平动和转动六自由度的虚拟物体位姿控制方法,难以满足需要对虚拟物体进行更加精细或者具有特定约束的操作场景应用需求,例如细长或大尺度工程构件的装配场景、空间位姿微调场景、教育培训等,限制了应用效果;3)从空间旋转姿态原理角度解析,通常有四元数和欧拉角两个方法,但是对于虚拟物体空间旋转姿态的特定轴向自由度的独立控制,四元数处理通常非常复杂,没有一个简单直接的方法来禁用四元数的某个轴向旋转,欧拉角虽然可以直观映射特定的旋转轴,但作为有顺序的非连续变换,存在“万向锁”的问题,当前针对虚拟物体空间旋转姿态的轴向自由度独立控制在技术上面临挑战。

[0004] 因此,实现混合现实环境下虚拟物体位姿控制具有重要的现实意义。将其应用在混合现实、虚拟现实和人机交互领域实现混合现实环境下灵敏度可动态调节、自由度可灵活设置、坐标系可按需切换的虚拟物体位姿控制,可以补充完善混合现实基于手势的虚拟物体位姿控制人机交互逻辑,满足不同场景下虚拟物体操作的应用需求,减少误操作、减轻用户学习负担,提升人机交互体验。

发明内容

[0005] 为根据场景需求动态调节基于手势交互的混合现实环境下虚拟物体位姿控制的灵敏度、自由度和坐标系参照,本发明提供一种混合现实环境下虚拟物体位姿控制方法,该方法包括以下步骤:

[0006] 步骤1,混合现实手势数据获取与处理;

- [0007] 步骤2,虚拟物体空间位姿灵敏度调节;
- [0008] 步骤3,虚拟物体空间位姿自由度控制;
- [0009] 步骤4,虚拟物体空间位姿坐标系映射;
- [0010] 步骤5,封装、测试与发布。
- [0011] 进一步,在步骤1中,可在Unity引擎中依次可执行如下步骤:
- [0012] 步骤1.1:基于MRTK的混合现实开发框架配置;
- [0013] 步骤1.2:手势交互数据接口实现;
- [0014] 步骤1.3:基于MRTK的手势交互数据获取;
- [0015] 步骤1.4:手势数据处理。
- [0016] 进一步,在步骤2中,依次可执行如下步骤:
- [0017] 步骤2.1:虚拟物体空间位姿平移灵敏度调节;
- [0018] 步骤2.2:虚拟物体空间位姿旋转灵敏度调节。
- [0019] 进一步,在步骤3中,依次可执行如下步骤:
- [0020] 步骤3.1:虚拟物体空间位姿平移自由度控制;
- [0021] 步骤3.2:虚拟物体空间位姿旋转自由度控制。
- [0022] 进一步,在步骤4中,依次可执行如下步骤:
- [0023] 步骤4.1:在Visual Studio中将 \vec{V}_2 和 R_2 同时向全局坐标或局部坐标映射;
- [0024] 步骤4.2:在Unity中实现虚拟物体空间位姿更新。
- [0025] 步骤4.3:定义预制体、材质球、虚拟载体网格匹配关系。
- [0026] 进一步,在步骤5中,依次可执行如下步骤:
- [0027] 步骤5.1:构建用于空间位姿控制的混合现实人机交互界面;
- [0028] 步骤5.2:用Visual Studio打开所保存的项目,将项目编译为ARM64格式,并发布成安装包;连接HoloLens 2,部署并调试HoloLens 2的使用环境,将HoloLens 2设置为使用Windows设备门户,通过Wi-Fi连接HoloLens 2设备,进行项目发布;
- [0029] 步骤5.3:佩戴HoloLens 2设备,根据人机交互界面选取不同的灵敏度、自由度、坐标轴参照进行虚拟物体空间位姿灵敏度调节测试、自由度控制测试和坐标系映射测试。
- [0030] 本发明的优点及积极效果为:
- [0031] (1) 灵活调整全局坐标系或局部坐标系下虚拟物体的空间位姿,实现空间位姿的按需控制;
- [0032] (2) 灵活动态控制虚拟物体平移和旋转的自由度,实现空间位姿的准确控制;
- [0033] (3) 灵活动态调节虚拟物体平移和旋转的灵敏度,实现空间位姿的精细控制;
- [0034] (4) 以脚本方式进行封装,快速灵活低成本部署。
- [0035] 基于此,本发明提出了一种混合现实环境下灵敏度可动态调节、自由度可灵活设置、坐标系可按需切换的虚拟物体位姿控制方法,补充完善了混合现实基于手势的虚拟物体位姿控制人机交互逻辑;满足不同场景下虚拟物体操作的应用需求,极大提升了混合现实技术在虚拟装配、教育培训、遥操作等需要对虚拟物体进行更加精细或者具有特定约束的操作场景的应用适用性;减少误操作、减轻用户学习负担,提升混合现实虚实交互和人机交互体验;为混合现实、虚拟现实和人机交互领域的虚拟物体位姿控制提供更加高效实用的方法,促进混合现实、虚拟现实和人机交互领域的创新及可持续发展。

附图说明

[0036] 通过结合以下附图所作的详细描述,本发明的上述和/或其他方面和优点将变得更清楚和更容易理解,此附图只是示意性的,并不限制本发明,其中:

[0037] 图1是混合现实环境下虚拟物体位姿控制方法流程图。

[0038] 图2是在Unity引擎中配置混合现实开发环境示例图。

[0039] 图3是用于空间位姿控制的混合现实人机交互界面示例图。

[0040] 图4是佩戴HoloLens 2设备操作及测试示例图。

[0041] 图5是虚拟物体空间位姿灵敏度调节示例图。

[0042] 图6是虚拟物体空间位姿自由度控制示例图。

[0043] 图7-1是虚拟物体的全局坐标系示意图;

[0044] 图7-2是虚拟物体在全局坐标系下进行平移示意图1;

[0045] 图7-3是虚拟物体在全局坐标系下进行平移示意图2;

[0046] 图7-4展示了虚拟物体的局部坐标系示意图;

[0047] 图7-5展示了虚拟物体在局部坐标系下进行平移示意图1;

[0048] 图7-6展示了虚拟物体在局部坐标系下进行平移示意图2;

具体实施方式

[0049] 为使本发明的上述目的、特征和优点能够更加明显易懂,下面结合附图和具体实施方式对本发明做进一步详细的说明。

[0050] 如图1所示,本发明提供一种混合现实环境下虚拟物体位姿控制方法,该方法包括以下步骤:

[0051] 步骤1,混合现实手势数据获取与处理;

[0052] 步骤2,虚拟物体空间位姿灵敏度调节;

[0053] 步骤3,虚拟物体空间位姿自由度控制;

[0054] 步骤4,虚拟物体空间位姿坐标系映射;

[0055] 步骤5,封装、测试与发布。

[0056] 进一步,在步骤1中,可在Unity引擎中依次执行如下步骤:

[0057] 步骤1.1:在混合现实开发平台Unity引擎中利用MixedRealityFeatureTool工具导入Mixed Reality Toolkit (MRTK) 混合现实开发工具包,主要包括“Mixed Reality Toolkit Examples”“Mixed Reality Toolkit Extensions”“Mixed Reality Toolkit Utilities”“Mixed Reality Toolkit Tools”资源包,完成混合现实开发环境配置,如图2所示;

[0058] 步骤1.2:在Visual Studio中通过C#编程语言实现混合现实手势处理器注册、手势识别逻辑定义、手势更新事件监听,以实现手势交互数据接口,主要包括下面两个MRTK方法“IMixedRealityGestureHandler<Vector3>”“IMixedRealityGestureHandler<Quaternion>”;

[0059] 步骤1.3:在Visual Studio中基于C#语言编写手势数据获取路径,基于MRTK获取基于Vector3的平移手势和基于Quaternion的旋转手势交互数据;

[0060] 步骤1.3:将从MRTK获取到的混合现实手势数据赋予虚拟物体空间位姿位置

“targetPosition”及旋转“targetRotation”数据。

[0061] 进一步,在步骤2中,依次可执行如下步骤:

[0062] 步骤2.1:创新提出并引入灵敏度参数,包括平移灵敏度 S_t 和旋转灵敏度 S_r ,首先进行平移灵敏度调节,按照平移灵敏度参数 S_t 对基于Vector3的空间位姿平移数据增量 $\vec{I} = (i_x, i_y)$ 进行缩放,其中 i_x 和 i_y 分别是沿着x轴和Y轴的输入量,然后将处理后的空间位姿平移增量数据与当前空间位姿平移位置数据 \vec{V}_{initial} 相加,输出空间位姿平移目标位置中间数据 \vec{V}_1 ,具体的表达式如下:

$$[0063] \quad \vec{V}_1 = \vec{V}_{\text{initial}} + S_t \cdot \vec{I}$$

[0064] 步骤2.2:进行旋转灵敏度调节,按照旋转灵敏度参数 S_r 对基于Quaternion的空间位姿旋转数据增量 R_{Δ} 进行缩放,需要将空间位姿旋转四元数转换为欧拉角数据 $\vec{\theta}_{\Delta} = (\theta_x, \theta_y, \theta_z)$,在欧拉角中乘以旋转灵敏度参数实现旋转增量的缩放 $\vec{\theta}_{\text{scaled}} = S_r \cdot \vec{\theta}_{\Delta}$,然后将处理后的欧拉角 $\vec{\theta}_{\text{scaled}}$ 重新转换为旋转四元数 R_{scaled} ,再基于四元数乘法将处理后的空间位姿旋转增量四元数数据与当前空间位姿旋转位置四元数数据 R_{initial} 相乘,输出空间位姿旋转目标位置中间数据 R_1 ,具体的表达式如下:

$$[0065] \quad R_1 = R_{\text{initial}} \otimes R_{\text{scaled}}$$

[0066] 进一步,在步骤3中,依次可执行如下步骤:

[0067] 步骤3.1:虚拟物体空间位姿自由度控制,创新提出轴向自由度独立控制方法,首先进行平移自由度控制,将空间位姿平移目标位置中间数据 \vec{V}_1 向xyz三轴映射,形成 V_{1x} 、 V_{1y} 和 V_{1z} 三个自由度单独数据, $\vec{V}_1 = (V_{1x}, V_{1y}, V_{1z})$,同时将当前空间位姿平移位置数据 \vec{V}_{C1} 向xyz三轴映射,形成 V_{C1x} 、 V_{C1y} 和 V_{C1z} 三个自由度单独数据, $\vec{V}_{C1} = (V_{C1x}, V_{C1y}, V_{C1z})$,根据轴向平移自由度控制要求,在 V_{1x} 、 V_{1y} 、 V_{1z} 、 V_{C1x} 、 V_{C1y} 和 V_{C1z} 中选取相应的自由度数据,选择过程可以通过引入函数F来描述,F作用于每一个轴向的平移自由度,决定采用 \vec{V}_1 或 \vec{V}_{C1} 的对应分量,定义控制向量 $\vec{F} = (F_x, F_y, F_z)$,其中每个 F_i ($i = x, y, z$) 为1时选择 \vec{V}_{1i} ,为0时选择 \vec{V}_{C1i} ,形成新的自由度数据,并输出空间位姿平移目标位置最终数据 $\vec{V}_2 = (V_{2x}, V_{2y}, V_{2z})$,其中 $V_{2i} = F_i \cdot V_{1i} + (1 - F_i) \cdot V_{C1i}$, i 代表x、y或z,具体的表达式如下:

$$[0068] \quad \vec{V}_2 = (F_x \cdot V_{1x} + (1 - F_x) \cdot V_{C1x}, F_y \cdot V_{1y} + (1 - F_y) \cdot V_{C1y}, F_z \cdot V_{1z} + (1 - F_z) \cdot V_{C1z})$$

[0069] 步骤3.2:进行旋转自由度控制,本发明创新提出一种四元数和欧拉角相结合的空间位姿旋转自由度独立控制方法,将空间位姿旋转目标位置中间数据 R_1 的四元数数据转换为空间位姿旋转目标位置中间数据的欧拉角数据 O_1 , $O_1 = \text{Euler}(R_1)$,然后将欧拉角数据 O_1 向xyz三轴映射, $O_1 = (O_{1x}, O_{1y}, O_{1z})$,形成 O_{1x} 、 O_{1y} 和 O_{1z} 三个自由度单独数据,再将当前空间位姿旋转位置四元数数据 R_{C1} 转换为欧拉角数据 O_{C1} , $O_{C1} = \text{Euler}(R_{C1})$,并向xyz三轴映射, $O_{C1} = (O_{C1x}, O_{C1y}, O_{C1z})$,形成 O_{C1x} 、 O_{C1y} 和 O_{C1z} 三个自由度单独数据,根据轴向旋转自由度控制要求,在 O_{1x} 、 O_{1y} 、 O_{1z} 、 O_{C1x} 、 O_{C1y} 和 O_{C1z} 中选取相应的自由度数据,选择过程类似于平移自由度控制,引入一个控制向量 $\vec{F}_R = (F_{Rx}, F_{Ry}, F_{Rz})$,其中每个 F_{Ri} ($i = x, y, z$) 决定了是选择 O_1 的分量还是 O_{C1} 的分量,形成新的空间位姿旋转目标位置最终数据的欧拉角数据 $O_2 = (O_{2x}, O_{2y}, O_{2z})$,其中

$O_{2i} = F_{Ri} \cdot O_{1i} + (1 - F_{Ri}) \cdot O_{C1i}$, i 代表x、y或z,最后输出空间位姿旋转目标位置最终数据的四元数数据 R_2 ,具体的表达式如下:

[0070] $R_2 = \text{Quaternion}(O_2)$ 。

[0071] 进一步,在步骤4中,依次可执行如下步骤:

[0072] 步骤4.1:虚拟物体空间位姿坐标系映射,本发明提出虚拟物体坐标系参照转换方法,根据坐标系参照要求,将 \vec{V}_2 和 R_2 同时向虚拟物体全局坐标或局部坐标映射,全局坐标系平移为 $\text{transform.position} = \vec{V}_2$,旋转为 $\text{transform.rotation} = R_2$,局部坐标系平移为 $\text{transform.localPosition} = \vec{V}_2$,旋转为 $\text{transform.localRotation} = R_2$;

[0073] 步骤4.2:进行虚拟物体在全局坐标系下或在相对于父物体的局部坐标系下的空间位姿更新,全局坐标系下的空间位姿平移表示为 $\text{targetPosition} = \text{transform.position}$,旋转表示为 $\text{targetRotation} = \text{transform.rotation}$;局部坐标系下的空间位姿平移表示为 $\text{targetPosition} = \text{transform.localPosition}$,旋转表示为 $\text{targetRotation} = \text{transform.localrotation}$ 。

[0074] 进一步,在步骤5中,可在Unity引擎中依次可执行如下步骤:

[0075] 步骤5.1:在Unity引擎中构建用于空间位姿控制的混合现实人机交互界面,调用Button组件构建交互按钮,包括平移和旋转灵敏度滑动调节按钮、自由度独立控制按钮、坐标系参照切换按钮,如图3所示;

[0076] 步骤5.2:用Visual Studio打开所保存的项目,将项目编译为与HoloLens 2合适的格式(ARM64格式),并发布成安装包,连接HoloLens 2,部署并调试HoloLens 2的使用环境,将HoloLens设置为使用Windows设备门户,通过Wi-Fi连接HoloLens 2设备,进行项目发布;

[0077] 步骤5.3:在真实空间中佩戴HoloLens 2设备,根据人机交互界面选取不同的灵敏度、自由度、坐标轴参照进行虚拟物体空间位姿灵敏度调节测试、自由度控制测试和坐标系映射测试,如图4-图7所示,如图5所示,以平移灵敏度控制为例,仅释放x轴平移自由度,实线代表手势的移动距离,5-1表示灵敏度为最高设置,可以看出虚拟物体位置变化对于手势移动几乎进行实时响应,白实线为直线,5-2表示当灵敏度设置为低灵敏度时,可以看出虚拟物体位置变化对于手势移动响应变慢,白实线开始弯曲,表明需要较大的手势移动距离,才能将虚拟物体进行较小的平移位置变化,证明灵敏度调节已经生效,图6锁定了x轴的平移,可以看到当手势沿x轴向右拖拽虚拟物体时,虚拟物体空间位姿在x轴不会发生移动,说明x轴平移自由度已经实现独立控制,旋转自由度独立控制同理,图7-1展示了虚拟物体的全局坐标系,可以观察到y轴正方向朝正北方向,此时只释放y轴平移自由度,在全局坐标系下手势向正北方向移动,虚拟物体只会在y轴正方向即正北方向上进行平移,如图7-2和图7-3所示,图7-4展示了虚拟物体的局部坐标系,可以观察到y轴正方向朝西北45度方向,此时同样只释放y轴平移自由度,在局部坐标系下手势向正北方向移动,但是虚拟物体只会在y轴正方向即西北45度方向上进行平移,如图7-5和图7-6所示,这也就证明了坐标轴参照已经切换成功。

[0078] 本发明提供的混合现实环境下虚拟物体位姿控制方法,创新提出并引入敏感度参数,灵活调整虚拟物体对于手势移动的响应程度,实现空间姿态的精细控制;同时提出虚拟

物体轴向自由度独立控制方法,实现空间位置的准确控制;此外,提出虚拟物体坐标系参照转换方法,实现空间位姿的按需控制;本发明解决了当前混合现实应用中难以根据场景需求动态调节基于手势交互的混合现实环境下虚拟物体位姿控制的自由度和灵敏度的共性问题;本发明补充完善了混合现实基于手势的虚拟物体人机交互逻辑,满足不同场景下虚拟物体操作的应用需求,减轻用户学习负担,提升人机交互体验,为混合现实、虚拟现实和人机交互领域的虚拟物体位姿控制提供更加高效实用的方法

[0079] 本发明不局限于上述实施方式,任何人在本发明的启示下都可得出其他各种形式的产品,但不论在其形状或结构上作任何变化,凡是具有与本申请相同或相近似的技术方案,均落在本发明的保护范围之内。

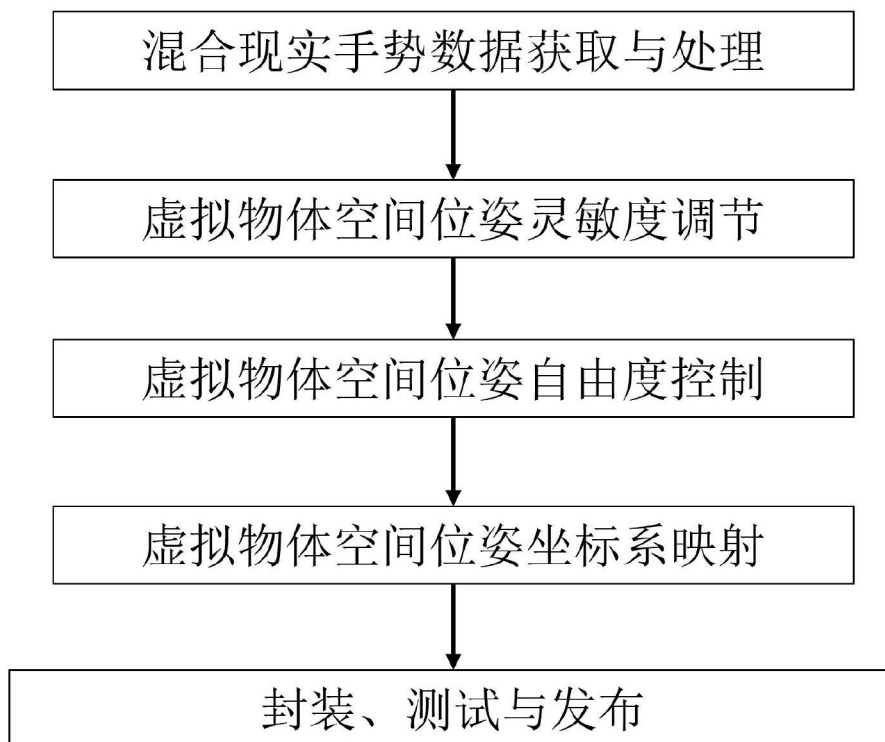


图1

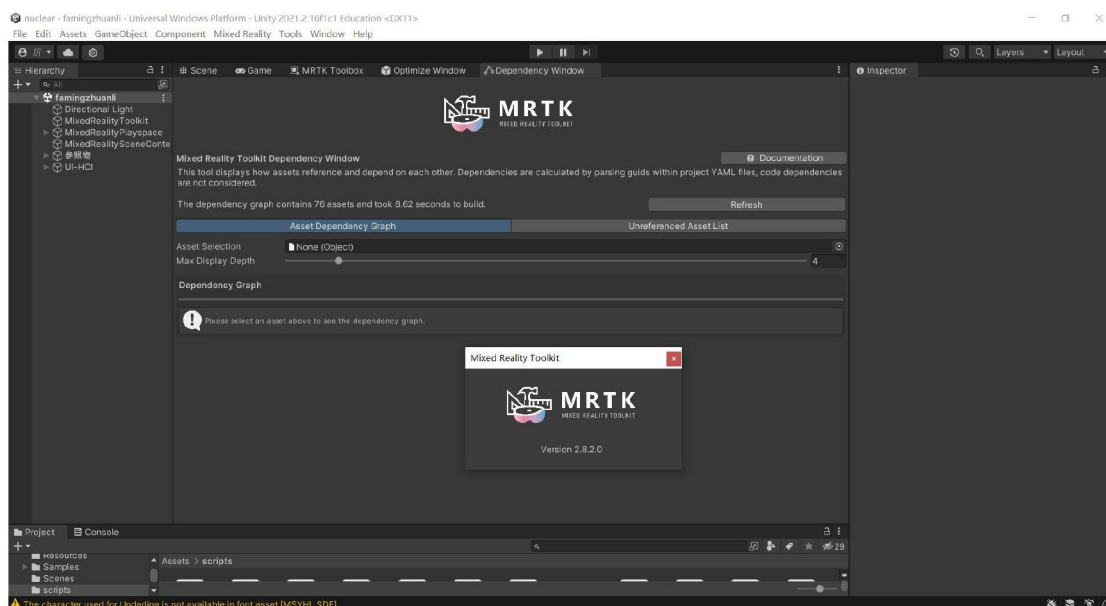


图2

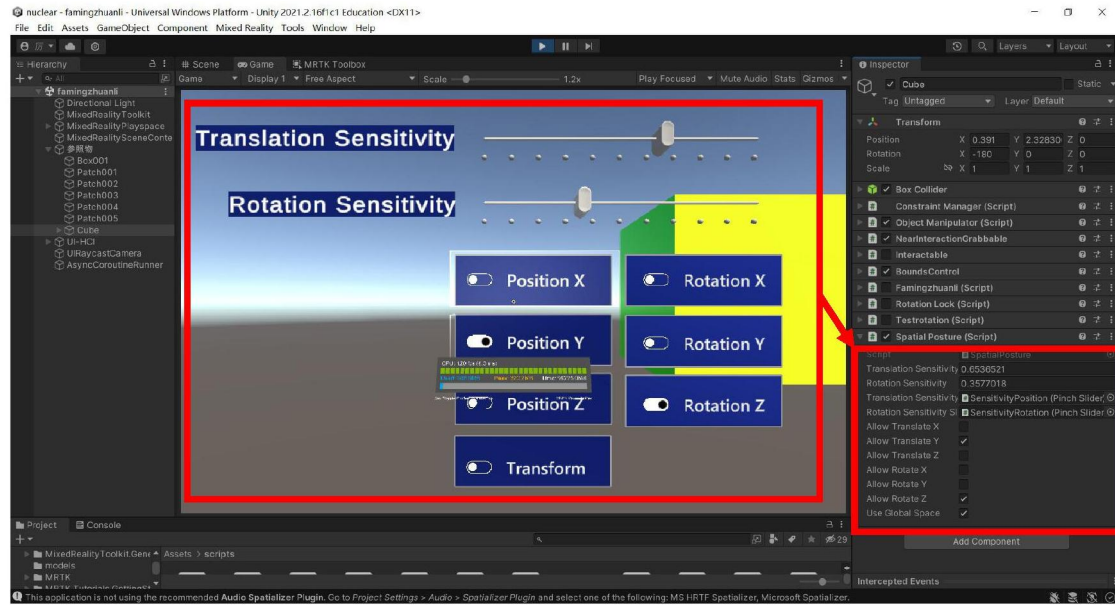


图3



图4

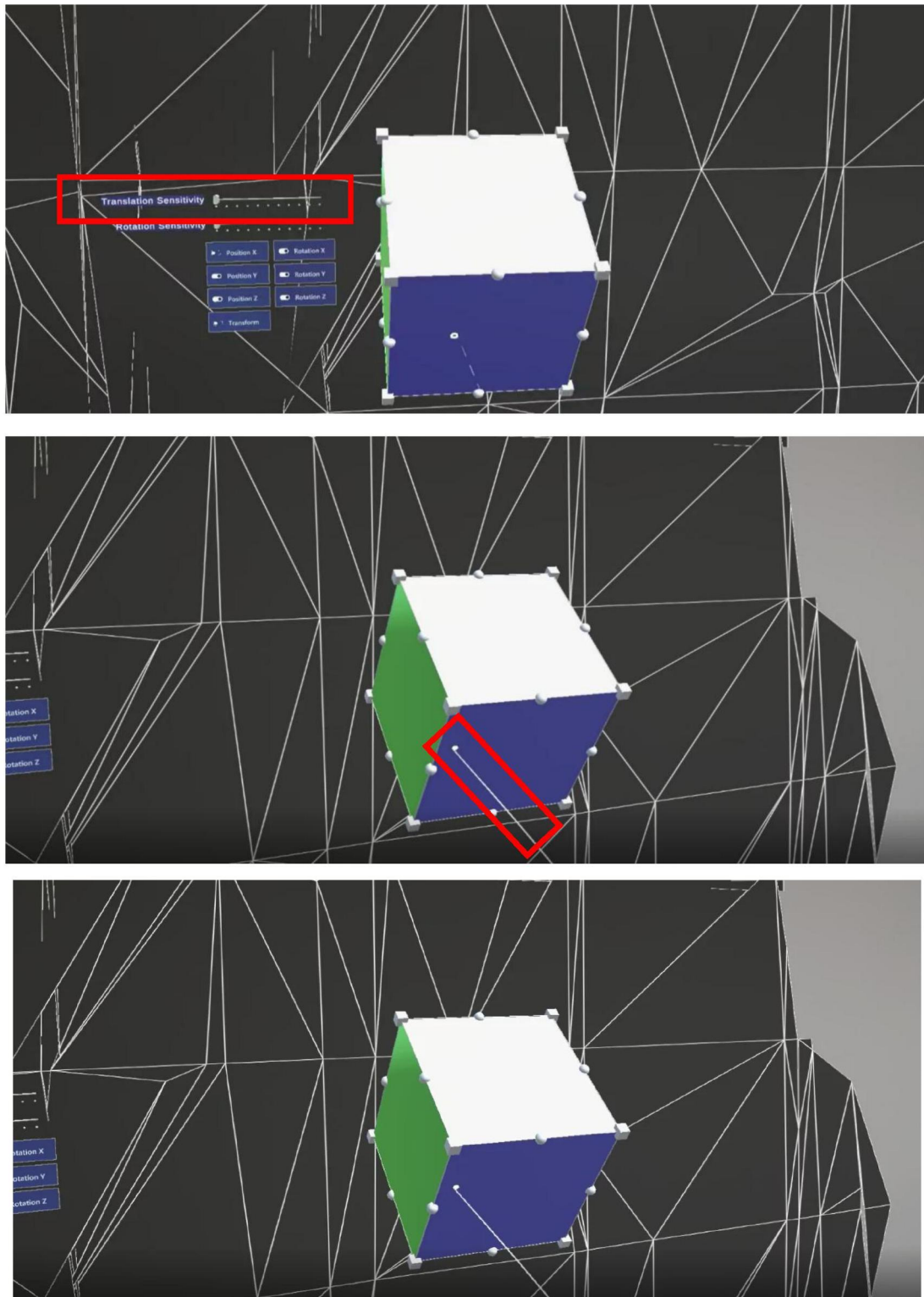


图5-1

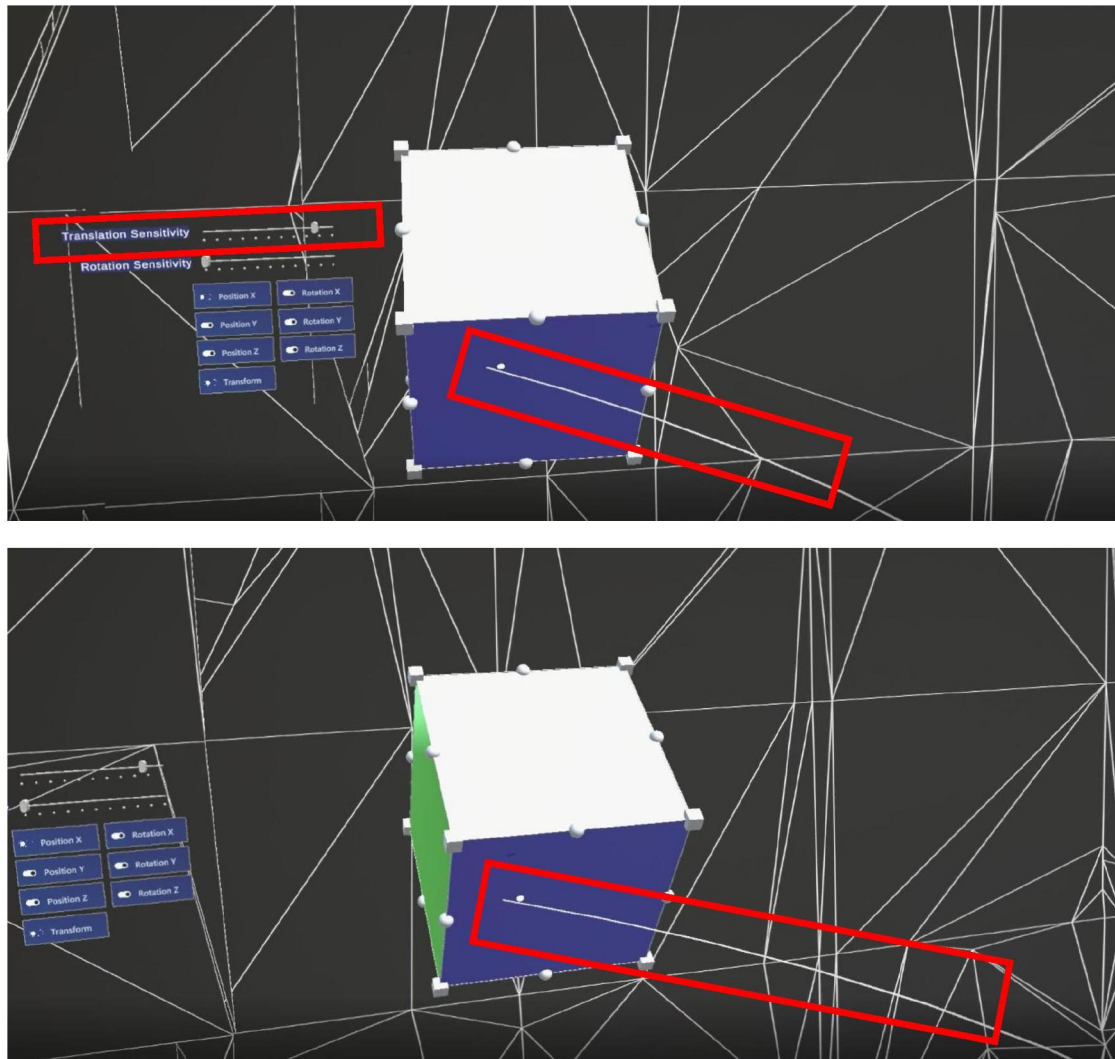


图5-2

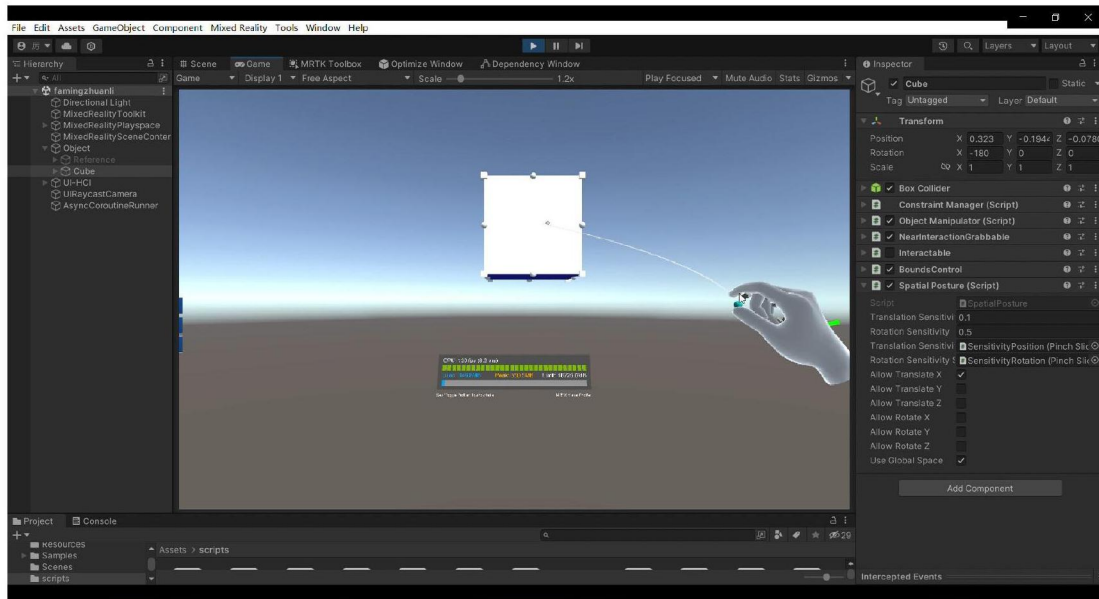


图6

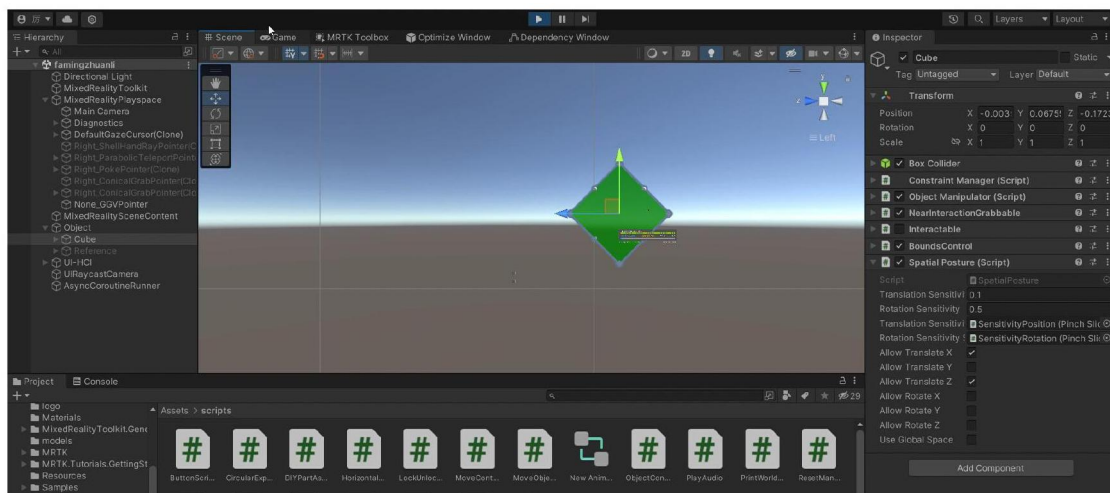


图7-1

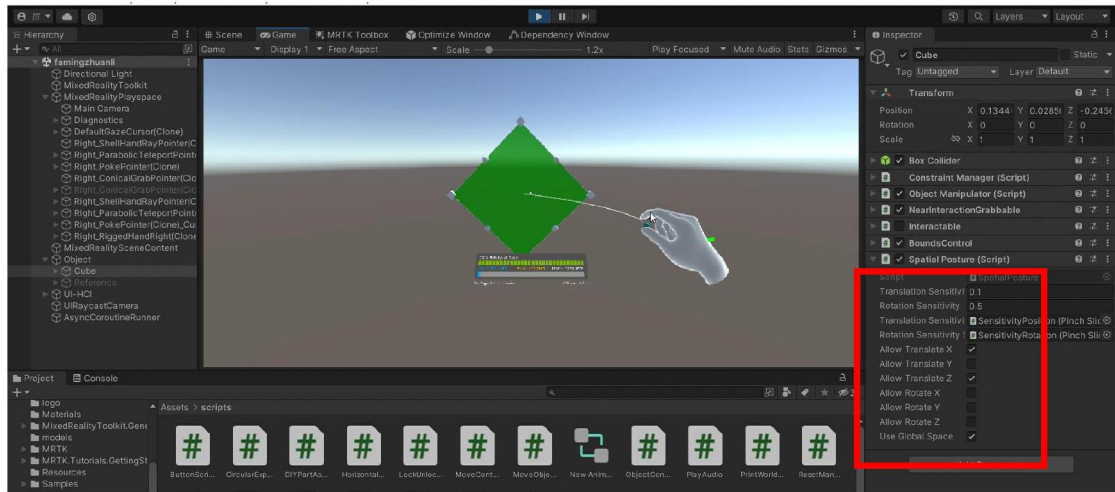


图7-2

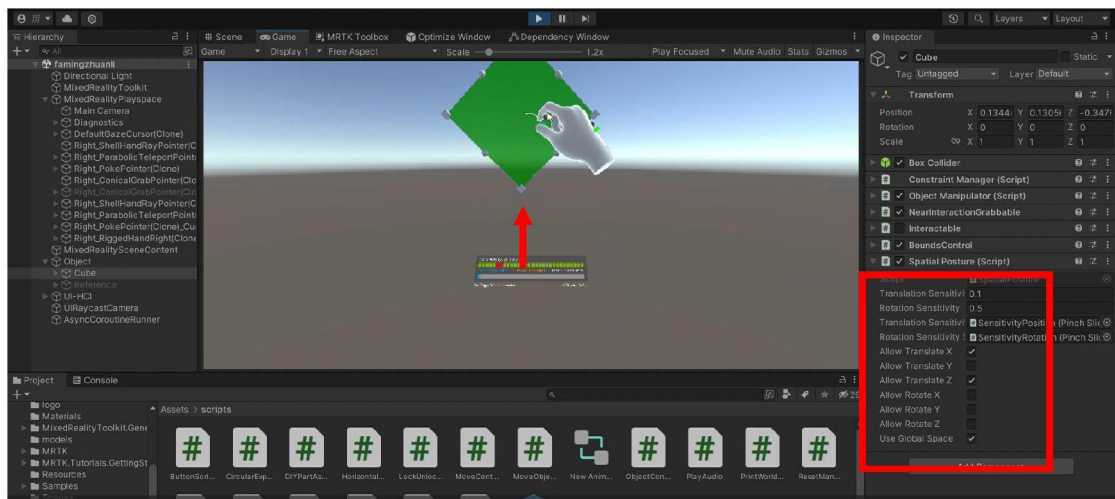


图7-3

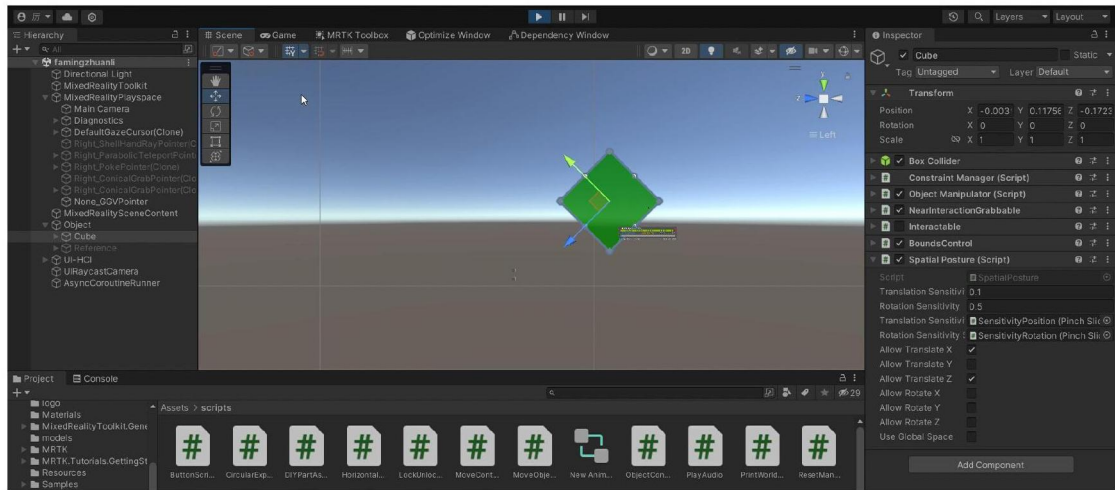


图7-4

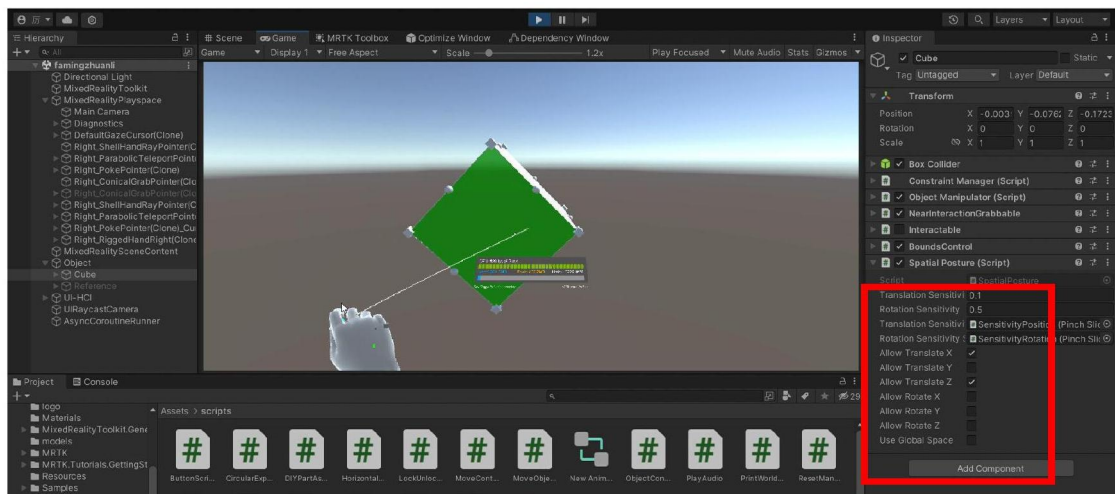


图7-5

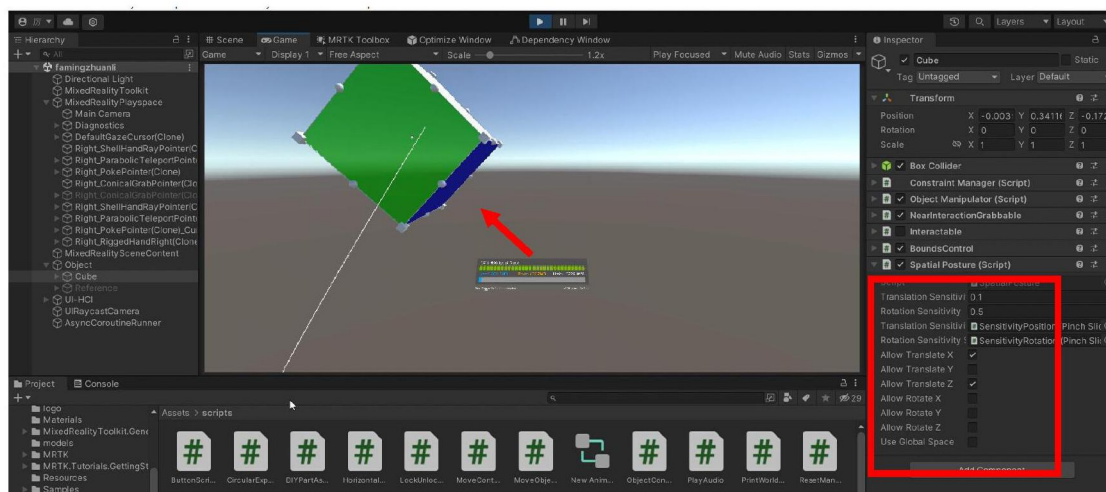


图7-6