# 基于VR头戴式显示设备的面向多用户的Locomotion系统

本文档正文将包含四个部分：第一部分为系统实现的功能简介，介绍了目前该系统所完成的一系列功能，并配以对应的参考文献；第二部分为系统待解决和遗留的问题；第三部分为对于阅读过的相关论文的总结和梳理，主要包含对于相关论文提出了哪些存在的问题，应用了哪些改善措施，论文的解决方法是否有效和数据证明以及还存在哪些遗留问题和改进措施。第四部分为参考文献

## 第一部分 系统功能简介

目前，多用户Locomotion系统复现了多篇论文的移动方式，包括：1.用户可以自定义传送终点高度，并配以预览窗口增强用户对终点周围环境的认识；2.群体传送方式，提供4种队形并提供优化传送后站位方案；3.用户可以自定义传送终点旋转角度，并配以预览用户化身，帮助用户调整视线角度；4.提供World-In-Miniature，极大提高传送的效率，用户可以通过选择微型世界当中的坐标来进行传送，并配以预览窗口，且可以调整用户的视线方向。

* 1. 自定义传送终点高度

1.1.1之前存在的问题

之前的传送系统大多数只能允许用户把传送的地点设定在地面上（设置了Teleportation Area）或者某些具体的点位上（设置了Teleportation Anchor），如果用户想要移动到半空中俯瞰整个全景，需要借助VR当中的梯子或者从屋中的楼梯爬到屋顶进行观赏，这非常不方便，无法让用户随心所欲地传送到半空中任何一个点位。

1.1.2 解决方案

在设定传送终点时用户可以自定义传送的高度，当用户的右手柄发出的射线和­­Teleportation Area有接触时，用户右手柄发出的射线会变白（图1），表示此处可以进行传送。当用户按下Trigger键会在目的地处创造一个光柱（图2）。此时用户可以抬起右手柄，随着右手柄角度和初始角度相差得越来越大，光柱会持续上升，一旦光柱的高度超出了用户的视野范围内，会提供一个预览窗口来帮助用户理解传送目的地的视野（图3）。此时用户可以通过预览窗口观看第三人称下的目的地视角，直到用户对视角感到满意，此时按下Grip键可以传送到预览窗口所显示的位置上去。



图1.当用户右手柄发出的射线和可传送的区域接触时，射线变成白色

图2.按下Trigger键会在目的地处创造透明光柱

图3.当光柱高度超过用户视野范围内会提供预览窗口

1.1.3 实现细节

首先通过XR Interaction Toolkit当中的Teleportation Area组件设置可以传送的区域，然后当用户的手柄射线接触到这片区域时就可以变成白色，提示用户可以进行传送。

当按下Trigger键会创造出光柱，然后读取右手柄此刻所处于的角度作为初始角度，随着用户右手的抬起和防下将改变光柱的高度。为了防止细微的抖动导致光柱高度变化，设置了Dead Zone范围，在这个范围内的角度变化不会影响光柱的高度。经过测试，这个值被设置为了10°。

当改变的角度处于（0°，VZ）或者（180°-VZ，180°）这个范围内，由于用户抬起的高度或降低的幅度较大，将把角度差和光柱上升的速度进行线性关联，在这个范围内每增加1°，那么光柱上升的速度将会提高0.1m/s（最快3m/s）。

当改变的角度处于（Press+DZ，180°-VZ）和（VZ，Press-DZ）的范围内时，将把当前角度和初始角度的差作为光柱高度变化的线性关联，在这个范围内每增加1°，那么光柱的高度则会增加0.1m。（图4）

一旦光柱超过了用户的视野范围，则会将预览窗口设置为可见，并且设置它的Transform为光柱正前方，然后生成在光柱上方生成用户的化身和摄像机，通过把光柱上方的摄像机Z坐标向后移动，把视角改为第三视角。此时设置预览窗口材料的纹理为第三视角摄像机的Output Texture，这样就可以在预览窗口中显示出光柱上方所展示的第三视角画面，也方便用户通过窗口观察将要传送的目的地的位置。

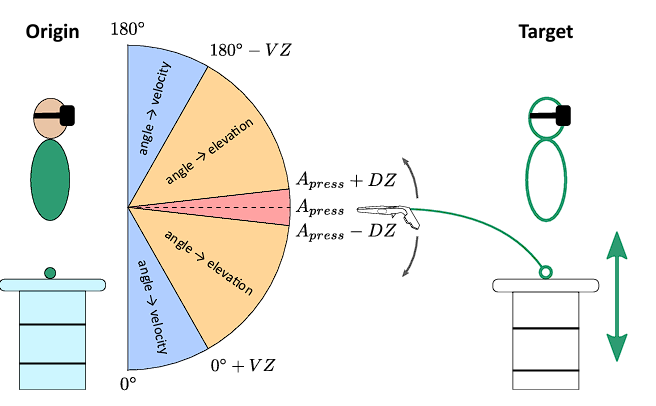


图4.用户手柄抬起的角度不同的范围：DZ（Dead Zone），VZ（Velocity Zone）

* 1. 群体传送方式

1.2.1之前存在的问题

之前的传送方案中如果需要旅游的小组一起移动到一个位置，需要每个成员一个一个来执行传统的单人传送功能，等一个人到了位置，下一个成员再执行自己的单人传送，这很容易造成小组成员丢失、移动效率较低、其他成员对环境不熟悉导致迷路、让游客过于费心思在移动上而不是欣赏风景等等问题。所以针对这种场景，需要有群体传送的解决方案。

其次，之前对于将其他成员加入小队的方法上缺少多人场景，只有提出了通过两个用户之间的虚拟手相互重叠，然后一起按下Grip键达0.5秒之后给两人建立小队的情况。但这种情况并不适合小组成员达到6-10人的情况。（一个一个握手效率太低）

1.2.2解决方案

针对之前存在的问题，该传送方案引出了旅游小组的两种角色——导游和游客。小组的传送位置由导游来决定，其他的小组成员只需要把重点放在欣赏风景上。

首先，当导游想要把游客加入到自己的小队时，导游可以通过右手柄按下Grip键，然后围绕待被加入的游客的地下画出一个圆圈，在圆圈内的游客就会被导游加入小组（图8），这样就大大提高了入队的效率。

当导游的右手柄和地面有接触时，同样手柄发出的射线会变成白色，表示此处可以进行传送。然后当用户按下Trigger键后，会在导游选择的目的地处生成整个小组的预览化身（图6）。

此外为了方便用户在游览风景中适应不同的场景（比如一起讨论、在狭窄的缝隙中移动等等），为小组设置了4种不同的队形（图5），导游可以通过右手柄的摇杆来控制选择哪种队形。如果游客想要一起讨论一个他们感兴趣的物体，可以让导游选择Circle或者HorseShoe来进行站位。如果想要通过某些狭窄的通道，导游可以选择Grid或者Queue来进行站位。

同时，为了支持传送当中需要规避障碍，为导游提供了预览队形的碰撞检测的功能。当要前往的位置与当前的站位有冲突时，用户化身的脚下的小球会显示红色，并且整个小组的成员之间也会有线连接，当线都为绿色时代表此处无碰撞，可以传送。如果发生了碰撞，可以选择改变预览的队形，比如改成Grid甚至Queue。

为了满足旅游当中的社交属性，旅游团队中可能包含两人一组或者三人一组的情况，所以系统需要保证传送后的队形依旧让他们能够保持接近。之前的4种基本队列只考虑了小组的整体，而具体小组内的站位则需要考虑组内的人际关系。对此，该系统完成了3种基本的站位计算方式和1种站位改善方式。首先，导游可以选择根据欧式距离损失最小的站位。其次，导游可以选择角距离损失最小的站位。再次，导游可以选择两者的综合最小值，两种损失函数的权重可以由用户自己提前设置，该系统分配的权重为各0.5。最后，因为小组的站位此前都是均匀分布的，但由于传送前可能关系亲密的用户们站得很近，而传送后突然增加距离的话会让用户感到陌生，可能需要重新转头寻找自己的朋友。所以，该系统增加了站位改善方式，通过传送前用户的两两站位距离和传送后的位置，优化小组成员的站位，让传送前距离相近的用户继续站在一起，增强用户的体验。

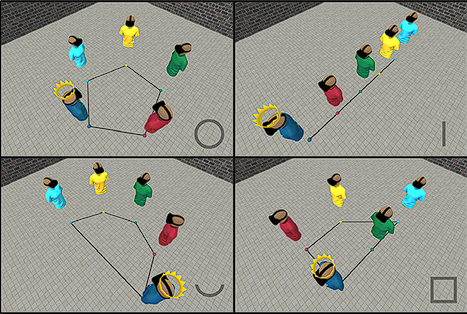


图5.4种不同的小组队形，分别是Circle、Queue、HorseShoe和Grid（从左上角到右下角）



图6.当导游按下Trigger键，会在目的地处展示整个小组的预览化身

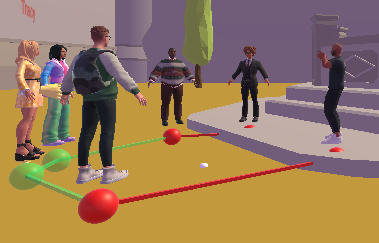


图7.当小组的预览化身与场景有碰撞时，会把用户脚下的小球和连接线变成红色，表示无法传送到此位置



图8.当导游按下Grip键绕着小组成员画圈时，被围在其中的成员会入队

1.2.3实现细节

首先通过XR Interaction Toolkit当中的Teleportation Area组件设置可以传送的区域，然后当导游的手柄射线接触到这片区域时就可以变成白色，提示用户可以进行传送。

其次，当导游按下Grip键绕着小组成员脚下的地面画圈时，被围绕在其中的成员会成功入队。通过RayInteractor的raycastHit获取与地面的交点来获取一系列的坐标点，然后通过LineRenderer的SetPosition函数将每个点位设置进去，最后就可以展示出导游所画的范围了。通过LineRenderer的点的集合里得到X和Z的坐标最大值和最小值，如果小组成员的X和Z坐标在这个范围内，那么就判断这些成员处于导游所画的范围内，然后把他们加入到小组里面。

接下来是展示小组的预览化身，当导游把想加入小组的成员加入组内，他就可以通过右手的射线在地面上选择传送点，然后按下Trigger键激活小组预览化身，默认小组是Circle站位。如果导游想要调整可以通过右手柄的摇杆来控制变化，将摇杆向右是Queue，摇杆向下是Grid，摇杆向左是HorseShoe，摇杆向上是Circle。队形是通过当前射线的落点，以及小组内人数，半径等计算出来的。以Circle站位为例，当射线和地面接触时，获取到它的坐标点，然后根据用户设置的半径和人数算出圆上应有多少个点位，并且通过圆心坐标和半径来计算出各个点位的坐标。然后将小组内的用户化身展示在这些坐标上。

接着是碰撞检测，当生成预览的用户化身时，会在用户脚下也生成一个小球，然后为其添加碰撞检测脚本，如果用户化身和周围的物体发生碰撞，就把小球变成红色。默认的绿色代表该用户可以传送到目标位置，红色代表用户和目标位置有碰撞，需要重新调整队形。同时各个小球之间也有连线，如果连线的一个小球变红了，那么整条线都会变红，让导游可以更加明显地看出碰撞。另一方面，添加的连线也可以让导游更好地了解目前小组所占的空间有多大。

最后是优化小组成员之间的站位，之前只解决了整个小组的队列问题，我们需要进一步优化小组内成员分别该站在哪个候选点上。对此，引出了3种计算方式。首先是基于最优欧式距离的计算，这样可以保证传送前临近的小组成员传送后依旧处于相邻的位置。主要采取的是穷举的方法，将之前队列计算出的每个位置轮流安排不同的组员，然后通过欧式距离当做损失函数，, ,U代表小组成员集合，D代表不同情况的损失函数，先讨论欧氏距离作为损失函数的情况。因为玩家都处于地面上，所以坐标仅考虑了x坐标和z坐标，通过传送前的欧氏距离减去传送后的欧氏距离误差最小来确定哪种站位方式是最适合的。

其次是角距离的计算，参考的是其他成员的位置相对于用户的位置和用户水平视线的角度差，这样是为了保证传送前用户视野范围内的成员，在传送后还会出现在他们的视野范围之内。同样采取穷举的方法，但是需要更换损失函数。如果要计算用户Ui相对于Uj的角距离，举个例子，此时用户Ui的Forward Vector V=(v1,v2)，然后通过Uj的位置向量减去Ui的位置向量，得到从Ui的位置指向Uj的位置的一个向量W。

随后通过atan2函数来计算向量W和向量V之间的角距离作为损失函数，公式如下：

然后选择角距离损失最小的站位安排。

再次是综合欧式距离和角距离的计算，为这两个损失函数赋予不同的权重，这个值可以由用户自行分配。代表基于欧式距离的损失，代表基于角距离的损失。的取值范围为(0,1),综合损失公式如下：

最后是站位的优化，由于4种队列算出的候选站位都是均匀分布的，需要运用算法来优化他们传送后的站位，让之前聚在一起的用户还聚在一起，而不是均匀分布。为此，选用了插值的方法。具体步骤如下：首先，先选用欧式距离的损失函数找出K个最优的分配站位方案，接着用插值优化这K个方案的站位。以Queue队形举例，固定最上面和最下面两个站位节点不变，然后算出目标队列中两两相邻的节点他们传送前之间的距离，把这些距离加在一起算出总和，然后根据总和和两者之间的距离进行插值计算，分配新的位置，这样就可以让传送前距离相近的两人传送后依旧接近。最后，根据这K个不同的方案的站位再用角距离损失函数计算一遍，选出最适合的优化的传送后站位方案。

* 1. 自定义传送终点角度

1.3.1之前存在的问题

在观赏VR世界中某个用户感兴趣的物体时，经常需要用户不断地调整角度和位置来从不同的视角观测它们，但现有的移动方式并不适合这种场景，总是需要用户在传送后转动身体来调整视角，尤其很多用户是坐着使用VR头显的，这就让转动身体这一行为更加困难，为他们的操作增加了负担。

1.3.2解决方案

1.3.3实现细节

* 1. World In Miniature传送

1.2.1之前存在的问题

1.2.2解决方案

1.2.3实现细节