

支持多模态交互的桌面增强显示系统

王子珩^{1,2}, 姜忠鼎^{1,2}

(1. 复旦大学 软件学院, 上海 200438; 2. 复旦大学 上海市数据科学重点实验室, 上海 200438)

摘要: 传统的显示设备在受限的物理空间内难以向用户呈现大量画面和复杂内容, 而AR头戴式显示设备通过将三维的可视化内容悬浮显示在用户眼前, 在不占用额外物理空间的条件下可增强真实世界的画面显示, 呈现形式更为丰富的内容。设计AR虚拟空间与真实电脑画面虚实融合的桌面增强显示系统。通过基于二维码识别的空间定位技术将真实电脑画面映射至虚拟空间内, 实现交互空间的统一, 同时构建窗口布局计算模型使得系统可以根据用户自定义参数自动生成窗口并设置其布局。在此基础上, 利用蓝牙通信、网络传输、操作系统底层映射和结合视线检测的语音识别等技术支持手势、键鼠和语音的多模态交互方式, 设计鼠标移动策略以扩展鼠标在三维空间下的多种操作模式。实验结果表明, 与隔空手势交互、鼠标交互等传统交互方式相比, 该系统在处理常见电脑任务时平均耗时节省10%~30%, 具有较高的交互效率, 且在跨窗口连续移动和瞬间跳转时能够正确显示鼠标位置。

关键词: 头戴式显示设备; 增强显示; 虚实融合; 多模态交互; 人机交互

开放科学(资源服务)标志码(OSID):



中文引用格式: 王子珩, 姜忠鼎. 支持多模态交互的桌面增强显示系统[J]. 计算机工程, 2022, 48(7): 177-188.

英文引用格式: WANG Z H, JIANG Z D. Enhanced desktop display system supporting multi-modal interaction[J]. Computer Engineering, 2022, 48(7): 177-188.

Enhanced Desktop Display System Supporting Multi-Modal Interaction

WANG Ziheng^{1,2}, JIANG Zhongding^{1,2}

(1. Software School, Fudan University, Shanghai 200438, China;

2. Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 200438, China)

[Abstract] Traditional display devices are limited in their ability to present a large amount of complex content to users within a given physical space. In contrast, Augmented Reality (AR) Head Mounted Display (HMD) devices can enhance users' perception of the real world by overlaying a three-dimensional visualization on their field of view to display richer content in several forms without occupying additional physical space. This study presents the design of an enhanced desktop display system that integrates virtual content in AR space with real screen. Through spatial positioning technology based on QR code recognition, real computer screens are mapped to a unified virtual interactive space. By establishing a window layout calculation model, the system can automatically generate windows and set their layouts according to user-defined parameters. The system supports multi-modal gesture interaction modes including keyboard and mouse input as well as voice control through technologies such as Bluetooth communication, network transmission, an underlying mapping of the operating system, and voice recognition based on gesture detection. The mouse movement strategy of the system is designed to expand the various operation modes of standard mouse devices into a three-dimensional space. The results of experiments show that users are able to complete common tasks with the proposed system in about 10%~30% less time on average compared with traditional input methods such as air gesture-based and mouse interaction. Moreover, the proposed approach exhibited a high interaction efficiency, and is able to display the cursor position correctly when moving across windows continuously and jumping instantaneously.

[Key words] Head Mounted Display (HMD) device; enhanced display; integration of virtuality and reality; multi-modal interaction; human-computer interaction

DOI: 10.19678/j.issn.1000-3428.0061698

基金项目: 教育部第二批“新工程”研究与实践项目(E-XTYR20200621)。

作者简介: 王子珩(1996—), 男, 硕士, 主研方向为计算机图形学、人机交互、增强现实; 姜忠鼎(通信作者), 副教授、博士。

收稿日期: 2021-05-20 修回日期: 2021-07-14 E-mail: wzh196423@gmail.com

0 概述

近年来,头戴式显示(Head Mounted Display, HMD)设备在重量、分辨率、续航能力和视场等方面不断改进,以便于供个人用户使用。由于HMD设备可以将三维可视内容悬浮呈现在用户眼前,因此与传统桌面显示器相比,HMD设备在不占用额外物理空间的条件下,能够显示数量更多、形式类型更丰富的内容^[1],有望被广泛应用于日常办公、学习场景中。

学术界已有较多关于使用HMD设备进行信息呈现的研究。文献[2]提出 Personal Cockpit 概念,探索了在AR空间下多窗口布局的参数对用户交互效率的影响,通过多项用户对比实验获得AR空间下多窗口布局需要遵循的规律,但是 Personal Cockpit 未研究相对于隔空手势更加高效和准确的交互方式。文献[3]提出 Spatial Analytic Interfaces,使日常的可视化分析任务在任何合适的时间和地点都能够就地执行,但是该研究同样没有进一步研究增强显示HMD的交互问题。文献[4-5]提出以VR设备作为远程办公平台,文献[6-7]提出基于VR设备的虚拟办公应用,可以在外出甚至交通工具上进行办公,并且将远程用户以虚拟化的形式呈现在本地用户的VR空间中,但是该应用仅实现基本的画面呈现和简单交互。Facebook 提出 Infinite Office 概念^[8],在VR空间中将多个可视化窗口环绕在用户眼前,支持隔空手势和键盘输入,满足使用者日常化办公的需求。但是 Facebook Infinite Office 只实现了基于隔空手势的跨窗口交互,并且VR设备呈现的均为虚拟画面,因此无法利用空间定位技术实现与真实空间的空间对齐,并且在该概念中,仅支持使用手势在多窗口间进行切换和交互,以及利用键盘进行输入,未考虑到鼠标交互这种更为高效且准确的交互方式。

关于使用HMD设备与真实空间虚实融合进行交互和显示的研究已取得一定的进展。早在1999年,文献[9]提出基于虚实融合建立连续工作空间的概念。Mirage^[10]是一个将AR头显内画面作为电脑扩展显示屏,实现AR空间与真实电脑画面的虚实融合,但是该应用没有实现交互空间的统一,用户只能使用电脑自带的键盘鼠标与AR空间内的扩展屏交互,而在AR空间内用户无法与电脑屏幕进行交互。文献[11]实现AR头显与大屏的虚实融合,并通过空间定位技术建立虚拟画面和真实大屏之间的联系,但是该系统并不涉及多窗口显示,因此无需考虑窗口间切换的交互问题,并且使用的也是手势交互,不支持键盘和鼠标。文献[12]将物理桌面屏与AR眼镜中的多维数据图表相结合,增强了复杂数据的呈现形式,但是仅支持数据图表增强现实,具有一般的泛用性。文献[13]提出使用AR头显增强现实中的纸质文件系统 HoloDoc,但是该系统的应用场景较局限,无法应用在使用电脑的数字化办公场景中。

文献[14]设计使用智能手机在AR头显中进行多窗口管理的系统。文献[15-16]分别提出 BISHARE 和 VESAD,实现了AR头显与手机屏幕的虚实融合,但是在该系统中仅支持基于多种手势的交互,交互效率、准确性都有所欠缺。文献[17]通过空间定位将真实世界中的触摸屏映射到VR空间内,从而打破了真实世界与VR空间的交互界限。在VR空间内,用户通过在触摸屏上移动手指来操作虚拟画面上的光标。该系统的不足之处在于,与基于键鼠的交互相比,基于触摸屏的交互准确度较低、交互动作也更少。由于该系统基于VR头显实现,因此无法对真实空间中的画面进行增强显示。文献[18]提出 Gluey,利用AR眼镜内的画面作为“剪切板”,在注册系统中的多设备之间传递图像并重定向交互设备的输入,尽管该系统同时涉及了真实和虚拟画面,且交互设备输入的重定向大幅提高了交互效率,但AR眼镜中的虚拟画面仅起到临时画布的作用,并没有与真实设备达到虚实融合的效果。最新的研究进展来自文献[19],他们利用 Windows Duplication API 采集本机电脑画面,在 Unity Editor 中运行并通过 Holographic Remoting Player 将采集的画面实时传输并显示在AR眼镜中,同时使用电脑相连的键盘和鼠标进行正常交互。该成果仅将AR眼镜用作虚拟外接显示器,所有的交互只能通过与电脑本身相连的设备才能完成,在AR眼镜中只能观看。该系统必须运行 Unity Editor 以及AR眼镜中的 Remoting Player,并非运行于AR眼镜的本机应用,并且只能显示一台电脑的桌面内容,其可扩展性和实用性都较低。

本文基于 HoloLens 2 AR 智能眼镜,利用混合现实工具包^[20] (Mixed Reality Toolkit, MRTK) 设计远程桌面增强显示系统,以支持多模态交互。采用基于二维码识别的空间定位技术将真实电脑画面投射至AR虚拟空间内,通过坐标空间转换和交互接口底层映射的技术,实现隔空手势、键盘鼠标与真实电脑桌面以及AR虚拟空间内画面的交互。

1 虚实融合的多窗口显示

1.1 远程PC控制的实现方法

实现AR虚拟空间与真实电脑画面虚实融合的主要步骤是首先将真实电脑屏幕准确映射到AR虚拟空间中,并在AR空间中与其物理位置空间对齐;其次AR头显通过网络向电脑发送有序、正确的交互指令。

真实电脑屏幕在AR空间内空间对齐的实现是利用 HoloLens 2 眼镜上4个黑白摄像头对黑白图案(二维码)进行识别^[21],当识别完成后以该二维码平面为基准面,建立一个局部坐标系。本文考虑到 HoloLens 2 关于二维码识别的官方文档^[21]中要求二维码边长在5~10 cm之间且对距离、环境有所限制,在正常光照的房内,测试了不同尺寸、同样内容的二维码能够识别的最远距离,结果如表1所示。

表 1 不同尺寸二维码的最远识别距离

Table 1 The farthest recognition distance of QR code with different sizes		cm
二维码边长	最远识别距离	
5	无法稳定识别	
6	60	
8	75	
10	90	

从表 1 可以看出,边长 6 cm 的二维码能够在 60 cm 以内的范围内稳定识别。由于考虑到定标二维码对屏幕画面的遮挡,同时在坐姿状态下人眼距离屏幕的距离约为 50~60 cm,因此本文选择边长为 6 cm 的二维码,做成卡片放置在电脑屏幕右下角进行定标,如图 1 所示。

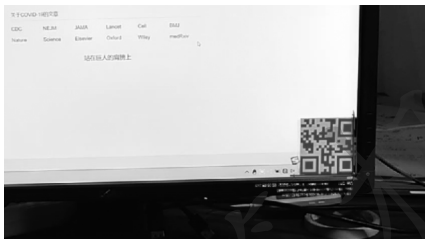


图 1 二维码定标
Fig.1 QR code calibration

HoloLens 2 识别该二维码后建立的局部坐标系如图 2 所示(彩色效果见《计算机工程》官网 HTML 版)。



图 2 坐标系建立示意图
Fig.2 Schematic diagram of coordinate system establishment

在此基础上,本文通过测量屏幕的长、宽等物理信息,将真实电脑屏幕在 AR 空间中实现空间对齐,如图 3 所示。从图 3 可以看出,空心圆点为手部射线与真实屏幕在 AR 空间内映射结果的交点,实心圆点为该交点映射到真实电脑屏幕上的显示点,两者几乎重合,表明实现了空间对齐。由于二维码紧贴在屏幕表面,因此无论是屏幕发生移动、旋转还是翻转,二维码都能在 AR 空间中准确确定物理屏幕的位置。

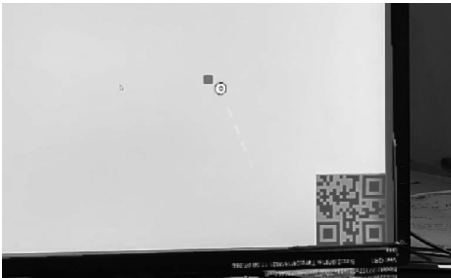


图 3 基于二维码实现的空间对齐
Fig.3 Space alignment based on QR code

当真实电脑画面在 AR 空间中实现空间对齐后,本文将物理画面映射到虚拟空间中,使得后续在虚拟空间中能够对物理画面(即其映射结果)和虚拟画面统一对待。

本文系统利用 AR 头显和连接在 AR 头显上的交互设备向远程电脑发送模拟鼠标和键盘输入数据的方式,实现交互空间的统一。本文系统使用 Windows10 操作系统,其系统自带的动态链接库 user32.dll 可以提供 mouse_event() 和 SendInput() 接口,将鼠标、键盘的输入信息映射成操作系统底层的鼠标和键盘。因此,远程电脑只需要获取 AR 空间内交互光标的点击状态、交互光标在电脑屏幕上的归一化坐标和键盘的按键数据,将 AR 头显及其连接交互设备的输入映射到远程电脑上,从而实现交互空间的统一。

HoloLens 2 眼镜支持与蓝牙鼠标和蓝牙键盘的无线连接,通过 Stream Socket 与蓝牙设备通信,准确获取鼠标的点击状态、绝对位置和键盘的按键数据。对于光标在电脑屏幕上归一化坐标的获取。HoloLens 2 眼镜将光标的归一化坐标、点击状态和键盘的按键数据通过网络发送至远程电脑,远程电脑通过 Windows 操作系统底层接口映射成键鼠事件,从而实现 AR 眼镜对远程电脑的控制。

1.2 多窗口布局

由于本文系统涉及真实电脑画面在虚拟空间中的映射结果以及多个虚拟画面窗口,因此需要考虑并设计包括多个真实和虚拟窗口的布局,方便用户使用。文献[22]提出 Ethereal planes 设计框架。基于该设计框架的理念,ENS B 等人提出 Personal Cockpit^[1]并通过多项用户对比实验,根据不同参数下用户操作的准确性和疲劳感,确定局部最优的参数组合。然而,对于 Personal Cockpit,所有的窗口均为虚拟窗口,因此无需考虑任何现实的物理因素。在本文系统中,将真实电脑画面通过空间对齐映射到虚拟空间的“真实窗口”,这是本文系统多窗口布局设计必须额外考虑的因素。此外,Personal Cockpit 的测试环境并非真正的 AR 眼镜,因此对于参数的选择考虑较少,同时在对比实验中所有的参数组合都是预先设置,无法根据不同用户的个性化体验进行修改,因此缺乏实际使用价值。

本文系统则支持根据用户配置文件中预先设置的参数,引入到窗口布局位置计算模型中,自动计算位置并对多窗口在AR空间中进行布局,同时在运行时,还支持用户通过面板修改配置,更新计算模型参数,重新计算并进行窗口布局。该窗口布局计算模型包括屏幕坐标系、用户坐标系(又称相机坐标系)和世界坐标系之间的坐标变换,如图4所示。

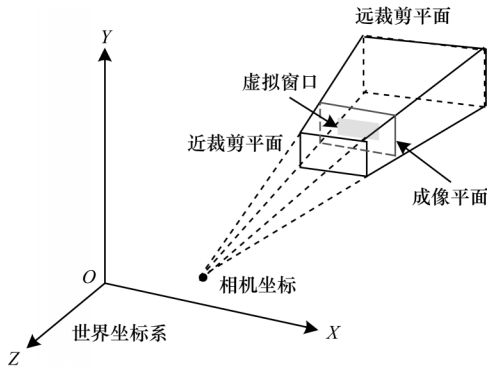


图4 窗口布局计算模型

Fig.4 Window layout calculation model

本文构建的窗口布局计算模型考虑在相机可视范围内只有一个虚拟窗口的情况。在三维的AR空间中,三维的物体经过多次变换,其世界坐标最终被映射成屏幕空间下的三维坐标(Z坐标仅用于保留深度信息),三维物体以二维图像的形式显示在用户眼前。用户只需配置虚拟窗口在屏幕空间下的尺寸比例和偏移量。屏幕空间坐标系如图5所示。

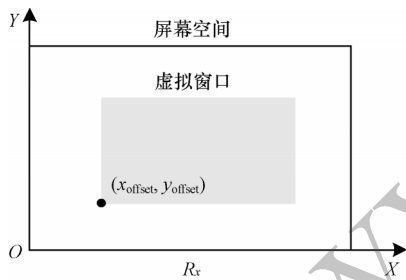


图5 屏幕空间坐标系

Fig.5 Screen space coordinate system

窗口中心在屏幕空间下的三维坐标 $(x_{\text{screenPos}}, y_{\text{screenPos}}, z_{\text{screenPos}})$ 如式(1)所示:

$$\left(\frac{R_x \times \alpha}{2} + x_{\text{offset}}, \frac{R_x \times \alpha}{2 \times r_{\text{aspect}}} + y_{\text{offset}}, d \right) \quad (1)$$

其中: d 为相机的焦平面距离; R_x 为相机横向分辨率; r_{aspect} 为宽高比; α 为用户配置的窗口比例; $(x_{\text{offset}}, y_{\text{offset}})$ 为配置窗口的左下角偏移量。

窗口布局计算模型将坐标系从屏幕空间转换到相机坐标系中。相机空间坐标系如图6所示。

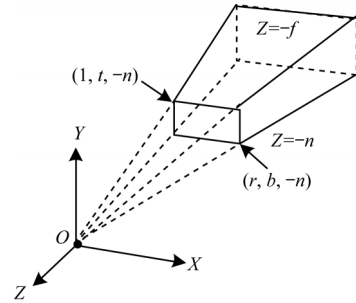


图6 相机空间坐标系

Fig.6 Camera space coordinate system

透视相机根据其前后裁剪平面的距离 n, f , 近裁剪平面的上下左右距离 t, b, l, r , 以及其前裁剪平面的左上和右下的顶点坐标 $(l, t, -n)$ 和 $(r, b, -n)$, 确定一个四棱台。在该四棱台的六个面组成的包围盒中,通过透视投影矩阵将三维坐标(由于三维坐标不能通过矩阵变换实现平移,在计算机图形学中一般增加一个 w 分量构建四维齐次坐标,利用 4×4 矩阵实现缩放、旋转、平移变换)转换成屏幕空间下的三维坐标。该相机空间坐标系透视投影的变换矩阵如式(2)所示:

$$M_{\text{persp}} = \begin{bmatrix} \frac{2 \times n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2 \times n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2 \times f \times n}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2)$$

根据图6的四棱台在相机空间中的对称性,可得:

$$r = -l \quad (3)$$

$$t = -b \quad (4)$$

对于 XOZ 平面的视场角 h_{fov} , 以及裁剪平面的宽高比 r_{aspect} , 根据三角函数,可得:

$$\tan \frac{h_{\text{fov}}}{2} = \frac{r}{n} \quad (5)$$

$$r_{\text{aspect}} = \frac{r}{t} \quad (6)$$

将式(3)~式(6)代入到式(2)中,可得:

$$M_{\text{persp}} = \begin{bmatrix} \cot \frac{h_{\text{fov}}}{2} & 0 & 0 & 0 \\ 0 & r_{\text{aspect}} \times \cot \frac{h_{\text{fov}}}{2} & 0 & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2 \times f \times n}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (7)$$

本文将坐标从屏幕空间转换到相机空间的矩阵,即为上述投影矩阵的逆矩阵,该矩阵如式(8)所示:

$$\mathbf{M}_{\text{screen} \rightarrow \text{cam}} = \mathbf{M}_{\text{persp}}^{-1} = \begin{bmatrix} \tan \frac{h_{\text{fov}}}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{r_{\text{aspect}}} \times \tan \frac{h_{\text{fov}}}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{f-n}{2 \times f \times n} & \frac{f+n}{2 \times f \times n} \end{bmatrix} \quad (8)$$

至此,本文得到虚拟窗口在相机空间中的三维坐标。窗口布局计算模型将坐标从相机坐标系转换到世界坐标系。相机在世界坐标系下的坐标为 $(x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}})$, 相机坐标系的基向量(分别对应世界坐标系下的基向量 $\vec{x}, \vec{y}, \vec{z}$) 在世界坐标系下的表示为 $\vec{u}, \vec{v}, \vec{w}$, 如式(9)所示:

$$\begin{bmatrix} \vec{u} \\ \vec{v} \\ \vec{w} \end{bmatrix} = \mathbf{M}_1 \times \begin{bmatrix} \vec{x} \\ \vec{y} \\ \vec{z} \end{bmatrix} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix} \times \begin{bmatrix} \vec{x} \\ \vec{y} \\ \vec{z} \end{bmatrix} \quad (9)$$

其中: \mathbf{M}_1 为将基向量从世界坐标转换到相机坐标的变换矩阵。将基向量从相机坐标转换到世界坐标的变换矩阵 \mathbf{M}_2 , 如式(10)所示:

$$\mathbf{M}_2 = \mathbf{M}_1^{-1} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix}^{-1} = \begin{bmatrix} x_u & x_v & x_w \\ y_u & y_v & y_w \\ z_u & z_v & z_w \end{bmatrix} \quad (10)$$

考虑到平移,窗口布局计算模型需要将三维坐标转换为四维的齐次坐标。因此,从相机坐标系转换到世界坐标系需要先将基向量从相机坐标转换到世界坐标,再将相机坐标原点 $(x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}})$ 通过平移与世界坐标的原点对齐。相机坐标到世界坐标的变换矩阵如式(11)所示:

$$\mathbf{M}_{\text{cam} \rightarrow \text{world}} = \begin{bmatrix} 1 & 0 & 0 & x_{\text{cam}} \\ 0 & 1 & 0 & y_{\text{cam}} \\ 0 & 0 & 1 & z_{\text{cam}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{M}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} x_u & x_v & x_w & x_{\text{cam}} \\ y_u & y_v & y_w & y_{\text{cam}} \\ z_u & z_v & z_w & z_{\text{cam}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

因此,窗口布局计算模型将窗口坐标从屏幕坐标系转换到世界坐标系,如式(12)所示:

$$\begin{bmatrix} x_{\text{worldPos}} \\ y_{\text{worldPos}} \\ z_{\text{worldPos}} \\ w_{\text{worldPos}} \end{bmatrix} = \mathbf{M}_{\text{cam} \rightarrow \text{world}} \times \mathbf{M}_{\text{screen} \rightarrow \text{cam}} \times \begin{bmatrix} x_{\text{screenPos}} \\ y_{\text{screenPos}} \\ z_{\text{screenPos}} \\ 1 \end{bmatrix} \quad (12)$$

$$\mathbf{M}_{\text{rotateR}} = \begin{bmatrix} r_{ix}^2(1 - \cos \theta_v) + \cos \theta_v & r_{ix}r_{iy}(1 - \cos \theta_v) - r_{iz}\sin \theta_v & r_{ix}r_{iz}(1 - \cos \theta_v) + r_{iy}\sin \theta_v & 0 \\ r_{ix}r_{iy}(1 - \cos \theta_v) + r_{iz}\sin \theta_v & r_{iy}^2(1 - \cos \theta_v) + \cos \theta_v & r_{iy}r_{iz}(1 - \cos \theta_v) - r_{ix}\sin \theta_v & 0 \\ r_{ix}r_{iz}(1 - \cos \theta_v) - r_{iy}\sin \theta_v & r_{iy}r_{iz}(1 - \cos \theta_v) + r_{ix}\sin \theta_v & r_{iz}^2(1 - \cos \theta_v) + \cos \theta_v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

通过将式(1)、式(8)和式(11)代入到式(12)中,根据任何一块虚拟窗口的屏幕空间坐标,计算其在AR眼镜中世界空间的坐标。

针对多窗口布局问题,本文只需要在相机空间下计算窗口各自的空间坐标,最后统一根据式(11)将坐标转换到世界空间中。根据用户配置窗口间水平方向和垂直方向的夹角 θ_h, θ_v , 在相机空间下窗口间旋转夹角的俯视图和侧视图分别如图7所示。

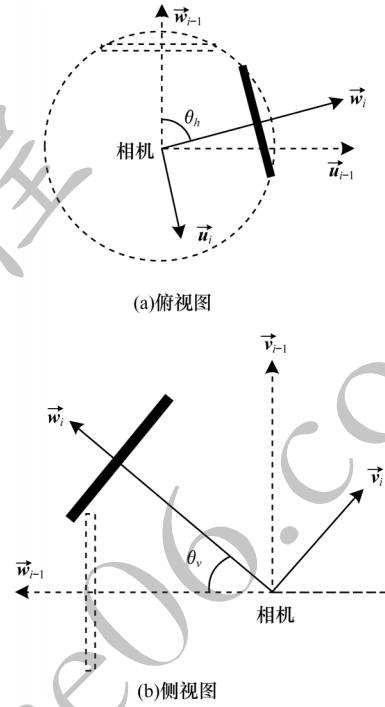


图7 在相机空间下窗口间旋转夹角示意图

Fig.7 Schematic diagram of rotation angle between windows in camera space

从图7可以看出,对于窗口 i ,在相机空间下先将窗口 $i-1$ 围绕相机原点处的 \vec{Y} 轴旋转 θ_h , 旋转矩阵如式(13)所示:

$$\mathbf{M}_{\text{rotateY}} = \begin{bmatrix} \cos \theta_h & 0 & \sin \theta_h & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_h & 0 & \cos \theta_h & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

旋转后窗口 i 的前向量 \vec{f}_i 和相机的 \vec{Y} 轴方向确定窗口的右方向向量 \vec{r}_i , 如式(14)所示:

$$\vec{r}_i = \vec{Y} \times \vec{f}_i \quad (14)$$

窗口 i 再围绕相机原点的 \vec{r}_i 方向为轴旋转 θ_v , 即可得到窗口 i 的相机空间坐标。将每个窗口的相机空间坐标代入到式(11)中,即可计算出每个窗口的世界坐标。绕 \vec{r}_i 方向的旋转矩阵如式(15)所示:

多窗口布局计算模型根据用户自定义的窗口比例、窗口在屏幕中的偏移量、多窗口之间的水平和垂直夹角等数据,自动计算并设置每个虚拟窗口的布局。此外,除参数化的窗口位置计算模型之

外,考虑到用户的个性化需求,本文系统还支持用户任意移动、缩放虚拟窗口。用户自定义调整虚拟窗口示意图如图8所示,其中虚拟窗口用黑色虚线框标示。



图8 用户自定义调整的虚拟窗口示意图

Fig.8 Schematic diagram of the virtual window with user-defined adjustment

2 多模态交互的实现方法

与 Facebook Infinite Office^[8]相似,本文系统使用的 HoloLens 2 眼镜也支持基于隔空手势的交互方式,但该交互方式相较于传统电脑的键盘和鼠标交互方式,交互效率和准确性都较差。考虑到电脑本身的交互方式是基于键鼠,本文在隔空手势交互的基础上实现基于键盘鼠标的交互方式。同时为了使交互更加自然,两种交互方式需要通过自然切换方式达到多模态交互的效果。此外,考虑到在使用场景中存在用户双手都被当前任务占用但仍需进行简单操作的情况,本文系统还支持基于视线检测的语音输入方式。

2.1 隔空手势交互

隔空手势交互需要将手部射线映射成远程电脑的鼠标,其中射线位置信息映射成鼠标位置,不同手势则对应不同的鼠标点击状态。

在 AR 空间中手部射线和画面窗口都是三维的,因此隔空手势交互方式下交互光标在画面上归一化坐标的计算,只需要在三维空间中通过射线碰撞检测得到手部射线和画面窗口的碰撞交点,然后在画面所在平面内根据交点坐标、画面长宽计算得到位置信息。

隔空手势交互计算示意图如图9所示。在三维空间中手部射线与画面窗口交于点 P ,画面窗口确定平面 XOY 。在该平面内,交点 P 的横纵坐标为 (a,b) ,而画面窗口的长宽为 W 和 H ,因此该交点的归一化坐标为 $(\frac{a}{W}, \frac{b}{H})$ 。该坐标映射到远程电脑上即为鼠标在电脑窗口内的归一化坐标,根据本文 1.1 节,将该坐标发送给远程桌面,通过操作系统底层接口映射成鼠标位置。

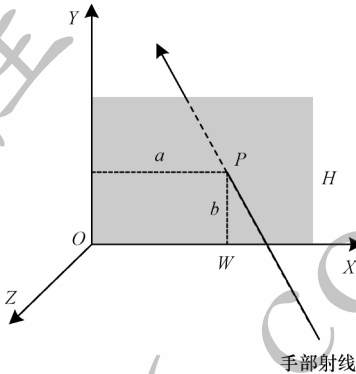


图9 隔空手势交互计算示意图

Fig.9 Schematic diagram of calculation of air gesture-based interaction

手势与鼠标点击状态的映射是根据常见手势在 AR 空间下的语义,合理映射为鼠标的点击状态,映射结果如表2所示。

表2 手势与鼠标点击状态的映射关系

Table 2 Mapping relationship between gestures and mouse click states

手势	语义	映射结果
食指与拇指捏合后张开	选择	左键 Click
手掌展开并移动	浏览	鼠标移动
食指拇指捏合并移动	拖动	按住左键移动

上述原理实现了手部射线与远程桌面鼠标之间(包括位置、点击状态)的映射,从而在 AR 眼镜空间内使用隔空手势交互远程操作电脑桌面。

2.2 鼠标交互

本文解决鼠标坐标从二维空间到三维空间的转换,以及鼠标跨窗口交互的问题。

2.2.1 鼠标坐标转换

Facebook Infinite Office 和本文系统实现了基于隔空手势的交互。由于手部射线和窗口画面均处于三维空间中,因此只需要利用射线的碰撞检测进行求交计算。但是鼠标本身的坐标是二维的,因此通过算法将二维坐标映射到三维空间中进行后续的求交计算,得到鼠标在画面上的归一化坐标。

无论是真实电脑画面在AR空间内的映射结果,还是完全虚拟画面,如果不考虑画面呈现在AR三维空间中的厚度,每个单独的画面都可以基于其面向使用者显示画面的表面,确定唯一的平面。三维画面确定的平面示意图如图10所示。

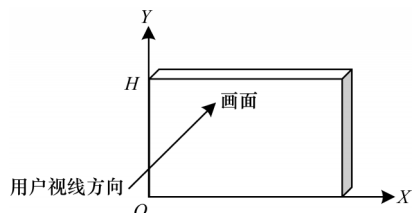


图10 三维画面确定的平面示意图

Fig.10 Plane schematic diagram of the three-dimensional image determination

三维画面确定的平面是一个三维空间中的平面,因此将得到平面上光标 P 在该平面内的归一化坐标 (a, b) ,计算 P 的三维坐标,如式(16)所示:

$$\mathbf{P}_{\text{pos}} = a \times \overrightarrow{OW} + b \times \overrightarrow{OH} \quad (16)$$

已知当前画面的分辨率 $R_x \times R_y$,当AR程序启动时,本文获取当前鼠标的二维坐标 (x_0, y_0) ,对应当前画面确定平面的原点(即图10中的 O 点)的归一化坐标 $(0, 0)$,记为 (a_0, b_0) 。之后的每一帧 i (其中 $i = 1, 2, \dots, n$),对于获取到的鼠标二维坐标 (x_i, y_i) ,其对应的平面上坐标 (a_i, b_i) 如式(17)所示:

$$(a_i, b_i) = (a_{i-1}, b_{i-1}) + \left(\frac{x_i - x_{i-1}}{R_x}, \frac{y_i - y_{i-1}}{R_y} \right) \quad (17)$$

在每帧计算后通过式(18)将 a_i 和 b_i 都归一化到 $[0, 1]$ 区间内:

$$(a_i, b_i) = \left(\min \left(\max (0, a_i), 1 \right), \min \left(\max (0, b_i), 1 \right) \right) \quad (18)$$

(a_i, b_i) 是鼠标在窗口上的归一化坐标,如果该窗口是真实电脑屏幕在AR空间中的映射窗口,则将该归一化坐标发送给远程电脑,通过操作系统底层接口映射成鼠标位置。

将 (a_i, b_i) 代入到式(16)中,即可获取每一帧鼠标的二维坐标 (x_i, y_i) 正确映射到三维空间中的三维坐标 $\mathbf{P}_{\text{pos}_i}$,从而在AR空间中显示三维光标作为提示。

2.2.2 鼠标跨窗口交互

鼠标跨窗口交互分为跨窗口连续移动和跨窗口的瞬间跳转。

1) 跨窗口连续移动

在三维空间内的隔空手势可以自然地在多窗口之间连续移动,对于鼠标交互,也应达到在多块桌面显示器窗口之间连续移动的效果,交互更自然,降低使用者的学习成本。

跨窗口连续移动的实现原理和传统鼠标跨多显示器移动类似。鼠标向左移出当前窗口示意图如图11所示。在移出瞬间,鼠标在当前窗口画面所在平面的归一化坐标为 $(0, b_i)$ 。为保证鼠标在窗口间连续移动时,其位置相对于整个窗口的比例固定,在移入左侧窗口的瞬间,鼠标在左侧窗口所在平面的归一化坐标为 $(1, b_i)$ 。在该平面内代入式(16)得到当前鼠标的三维坐标。之后的每一帧通过式(16)~式(18)计算出鼠标的三维坐标。

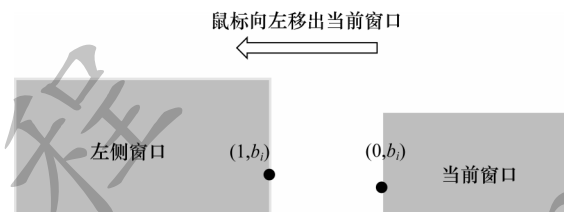


图11 鼠标向左移出当前窗口的示意图

Fig.11 Schematic diagram of moving the mouse to the left out of current window

鼠标向右移出当前窗口的示意图如图12所示。在移出瞬间,鼠标在当前窗口画面所在平面的归一化坐标为 $(1, b_i)$,在移入右侧窗口的瞬间,其在右侧窗口所在平面的归一化坐标为 $(0, b_i)$,在该平面内代入式(16)得到当前鼠标的三维坐标。

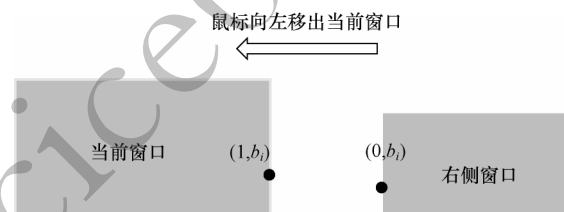


图12 鼠标向右移出当前窗口的示意图

Fig.12 Schematic diagram of moving the mouse to the right out of current window

如果仅根据归一化坐标的横坐标临界值和鼠标运动方向判断鼠标是否需要跨窗口连续移动。因此,本文系统增加了对鼠标移动速度的判断。如果鼠标移动速度小于阈值,那么用户很可能只是在窗口边缘进行交互。只有当鼠标归一化横坐标达到临界值且移动速度大于阈值时,才会认为用户有跨窗口连续移动鼠标的意图。

2) 跨窗口的瞬间跳转

如果仅实现鼠标跨窗口的连续移动,那么在窗口数量较多或用户需要一次性跨多个窗口交互时效率较低,因此还需实现鼠标跨窗口的瞬间跳转。文献[23]设计并实现的 Multi-monitor Mouse,利用鼠标特殊按键在多个物理窗口之间瞬间跳转,在鼠标跳转后将其保持在与之前页面等比例的位置上,能够有效提高效率。他们还设计了当鼠标跳转到新页

面时,将鼠标自动移动到上一次离开该页面的历史位置,由于操作系统限制,因此无法提供额外的信息提示。

本文受 Multi-monitor Mouse 切换策略的启发,用户将视线移动到目标页面后,通过键盘的组合按键 CTRL+TAB 触发鼠标的瞬间跳转,在鼠标跳转后,将其保持在与之前页面等比例的位置。瞬间跳转后鼠标位置显示的示意图如图 13 所示,鼠标在瞬间跳转至新窗口后,在跳转窗口内保持与原窗口相同的归一化坐标 (a,b) 。

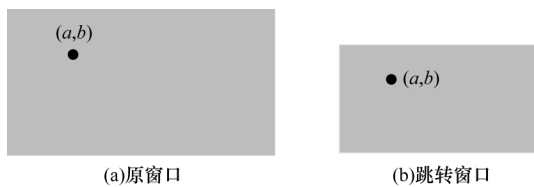


图 13 鼠标瞬间跳转后位置显示示意图

Fig.13 Schematic diagram of position display after mouse jumps instantaneously

瞬间切换改变了用户交互的当前上下文,并且用户需在新窗口内回忆历史交互记录。Multi-monitor Mouse 受底层操作系统的限制,无法为用户提供鼠标历史位置的提示,但是 AR 空间中可以在当前窗口上叠加一个鼠标最近一次离开页面时的历史位置,并将其作为提示,以便于用户在瞬间跳转后迅速切换至当前窗口的历史上下文,提高切换效率。

对于历史位置的计算,如果是通过跨窗口连续移动的方式离开页面,则历史位置为鼠标离开页面时在页面边缘临界处的坐标,其归一化坐标为 $(0,b_i)$ 或 $(1,b_i)$ 。如果用户通过跨窗口的瞬间跳转方式切换页面,则历史位置为用户按 CTRL+TAB 键时鼠标在上一个页面中的归一化坐标 (a,b) 。

本文系统将历史位置记录完成后,鼠标在瞬间跳转到新页面时,将会在新页面上叠加显示上一次离开该页面时的历史坐标,其原理如图 14 所示。

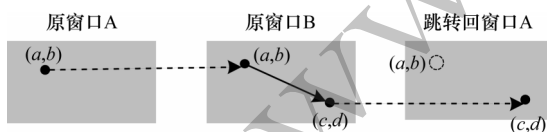


图 14 鼠标跨窗口瞬间跳转后历史位置显示原理

Fig.14 Display principle of historical position after mouse jumps across the window instantaneously

鼠标最初在窗口 A 内的归一化坐标为 (a,b) ,瞬间跳转至窗口 B 时保持鼠标在 A 中的归一化坐标 (a,b) ,此时记录鼠标在窗口 A 中的历史位置 (a,b) 。鼠标在窗口 B 内移动至 (c,d) 处后,又瞬间跳转至窗口 A,并保持坐标为 (c,d) ,同时系统获取到鼠标在窗

口 A 内的历史位置 (a,b) ,叠加显示在窗口 A 上,提示用户在上一次交互结束时的位置信息。

在窗口 A 中,为了增强提示作用,历史位置的提示标志会持续闪烁,在以下两种条件下该闪烁提示才会消失:(1)用户将鼠标移至历史位置附近,表示用户接受该提示;(2)用户在页面内的任意位置触发鼠标点击事件,表示不再需要该历史位置的提示。

鼠标在跨窗口瞬间跳转后,保持与上一个页面相同归一化坐标的位置,使得鼠标的瞬间跳转十分自然,并且符合用户的历史经验,这一点在 Multi-monitor Mouse^[23]中已经得到证实。用户通过历史位置的提示显示可以快速回忆之前交互的上下文,即使用户不需要历史信息提示,该提示信息也不会干扰用户的正常交互,相较于 Multi-monitor Mouse^[23],本文系统提高了交互的效率。对于鼠标跨窗口瞬间跳转的实现效果见本文 3.3 节。

2.3 隔空手势与鼠标交互间的切换

本文在 2.1 节和 2.2 节中实现了在 AR 空间内使用隔空手势和鼠标这两种交互方式,在此基础上,还需要支持用户在不同交互方式之间的自然切换,降低用户的学习成本,提高交互的准确性和效率。

本文考虑到在日常办公和学习中,用户大量使用键盘和鼠标与电脑进行交互,因此用户对这类交互方式更熟悉。在 AR 空间内大多数用户会选择使用键盘和鼠标进行交互。一旦用户伸手将手部射线朝向画面窗口,就将交互方式切换为隔空手势交互,不再响应鼠标的任何输入。为了避免用户抬手误触,又考虑到大多数用户是右利手,因此本文在检测用户手部射线时只检测右手的手部射线,将左手射线均视作误触。当右手的手部射线与任一画面发生碰撞并产生交点时,本文系统才将交互方式切换为隔空手势交互,并在碰撞结束后自动切换为鼠标交互。

本文实现了隔空手势与鼠标交互的切换,还需要考虑在两者切换时交互光标的显示问题。本文采取形状不同的光标对两者进行区分。当鼠标交互切换至隔空手势交互时,本文记录鼠标在先前画面内的历史位置,随后隐藏鼠标的交互光标,将手部射线光标设置在右手射线与发生射线碰撞画面的交点处。在隔空手势交互结束后,切换至鼠标交互时,隐藏手部射线光标,本文将鼠标位置设置为手部射线与画面窗口最后一次发生碰撞的交点处,并在该位置显示鼠标交互光标。

本文实现了隔空手势与鼠标交互间的自然切换,并最大程度地避免了用户手势的误操作。同时,在两种交互方式之间进行切换时,本文设置了切换

前后不同光标的显示位置,避免用户在切换交互方式时丢失光标。

2.4 结合视线检测的语音输入

隔空手势结合键盘鼠标的交互方式基本能满足一般工作场景的需求。但是考虑到学习、工作中双手被任务占用时还需要进行其他操作的情况,因此本文系统实现了结合视线检测的语音输入交互方式。在该场景下用户的双手被任务占用,对于操作目标的选定无法再依赖于需要手的交互方式,因此本文选择通过获取用户视线的聚焦对象,为用户完成语音控制前的目标选定。系统通过 MRTK 中的 Gaze Provider 获取用户视线的方向,在该方向上进行实时的射线碰撞检测,检测到支持语音交互的物体时,用户可随之进行语音输入。

与键盘鼠标输入以及隔空手势相比,由于语音输入的效率和准确度较低,因此语音输入主要支持复杂程度较低的操作。语音输入支持的指令如表 3 所示。

表 3 语音输入支持的指令描述
Table 3 Description of instructions supported by voice input

语音指令	操作	使用场景
Pick up	选取窗口	窗口移动
Put down	释放窗口	
Follow me	窗口始终视线跟随	
Zoom in	窗口放大 10%	窗口缩放
Zoom out	窗口缩小 10%	
Near	窗口靠近 0.1 m	窗口距离调整
Far	窗口离远 0.1 m	
Close	关闭窗口	关闭当前窗口
Face to me	窗口正对用户	调整窗口朝向
Reset layout	重置窗口布局	重置布局

表 3 中的指令在对应场景下可以让用户无需中断手头当前任务,就可以进行调整窗口位置、大小、距离等操作,在多窗口场景下能够有效提高用户的工作效率,减少交互上下文切换产生的不必要开销。

3 实验结果与分析

3.1 实验环境

系统基于 Unity 游戏引擎开发,版本为 2019.4.11f1 (64 bit),在 Microsoft HoloLens 2 增强现实眼镜上运行。真实电脑采用 Intel i7-7700 HQ 处理器,2.80 GHz,16 GB 内存,显卡为 NVIDIA GTX1060 3 GB,Windows 10 64 bit 操作系统,运行一个使用 C# 开发的 Socket 服务器。眼镜和电脑连接在同一个局域网中。另有 6 台相同配置的电脑负责推流给 AR 眼镜,从而呈现虚拟画面。

3.2 跨窗口连续移动时鼠标位置

在使用鼠标交互的状态下,本文首先在窗口 A 将鼠标移动至归一化坐标为 (1.0,0.5) 的临界处,鼠标向右快速水平移动至窗口 B,观察此时鼠标光标的位置。跨窗口连续移动时的鼠标位置如图 15 所示。

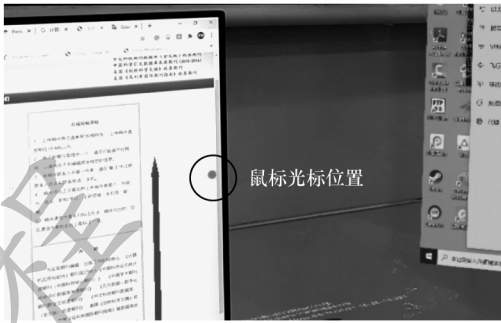


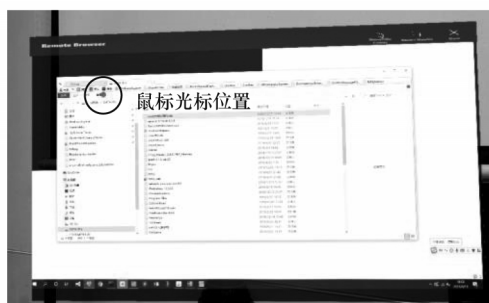
图 15 跨窗口连续移动时的鼠标位置

Fig.15 Mouse position when moving across windows continuously

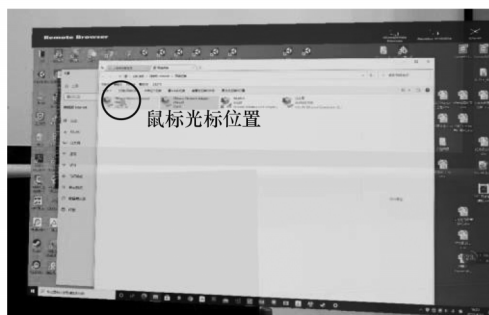
从图 15(a)可以看出,鼠标离开上一个窗口时临界位置的归一化坐标 (1.0,0.5)。从图 15(b)可以看出,鼠标连续移动后,其进入新窗口时的临界值归一化坐标为 (0.0,0.5)。当鼠标进入新窗口和离开上一个窗口临界位置的横坐标均为临界值时,纵坐标相同。该实验说明当鼠标跨窗口连续移动时,在不同窗口间的归一化坐标计算和显示是连续的,验证了跨窗口连续移动时鼠标位置计算和显示的正确性。

3.3 跨窗口瞬间跳转时鼠标位置与历史位置

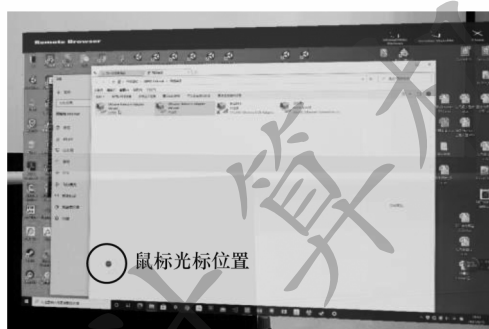
在使用鼠标交互的状态下,本文首先在窗口 A 将鼠标移动至归一化坐标为 (0.15,0.84) 处,通过组合键将鼠标瞬间跳转至窗口 B,观察此时鼠标光标的位置,然后在窗口 B 内将鼠标光标移动至归一化坐标为 (0.17,0.18) 处,通过组合键将鼠标瞬间跳转回窗口 A,观察此时鼠标的鼠标位置,以及显示的历史位置。跨窗口瞬间跳转时的鼠标位置及历史位置如图 16 所示。



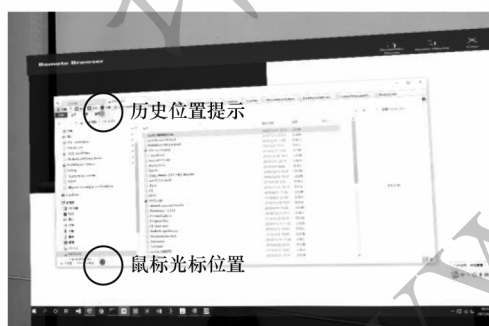
(a)鼠标在窗口A内的初始位置



(b)瞬间跳转后鼠标在窗口B内的位置



(c)鼠标在窗口B内移动后的位置



(d)瞬间跳转回窗口A后的位置显示

图 16 跨窗口瞬间跳转时鼠标位置及历史位置

Fig.16 Mouse position and historical position when jumping across windows instantaneously

从图 16(a)和图 16(b)可以看出,鼠标瞬间跳转后,其在新窗口内保持与之前窗口相同比例的位置。从图 16(c)和图 16(d)可以看出,鼠标再次瞬间跳转回先前页面时,页面上显示正确的历史位置提示。该实验表明鼠标瞬间跳转时,其跳转位置和历史位置的计算和显示均正确。

3.4 隔空手势、鼠标连续移动和瞬间跳转的效率对比

本文在多窗口场景下,测试隔空手势、鼠标连续移动和鼠标瞬间跳转的交互效率。实验流程是在

AR眼镜中使用 2×3 的窗口布局,每个窗口占相机画面的比例为0.3,宽高比为16/9。窗口间横向和纵向圆心角的夹角均为 27° ,实验窗口布局示意图如图 17所示。每个窗口内的画面显示如图 18所示,有编号为1~4的四个按钮,排列顺序随机。实验参与者需要根据指令依次点击窗口中的按钮。“指令”为一个包含3个元素的元组,3个元素分别表示按下的按钮所在窗口的行号、列号和按钮的编号。例如,1、3、4表示参与者需要点击第1行、第3列窗口中的4号按钮。每位参与者需要进行2次实验,每次实验需要完成的指令个数分别为10和15个,每次实验中,参与者用3种交互方式各完成一遍指令并计算耗时。实验共有3名参与者。

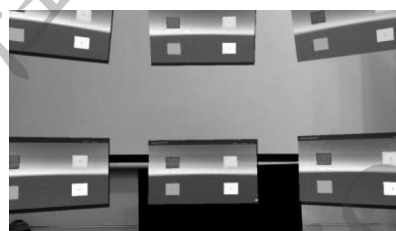


图 17 实验窗口布局示意图

Fig.17 Schematic diagram of experimental window layout

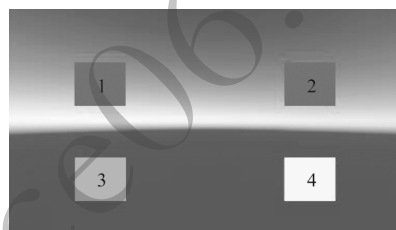


图 18 窗口内按钮分布示意图

Fig.18 Schematic diagram of button layout inside the window

不同交互方式的平均耗时对比如图 19所示,隔空手势交互和鼠标连续移动交互的效率均低于鼠标瞬间跳转方式。其原因为瞬间跳转在窗口数量较多时节省了光标在不同位置之间移动的时间。在交互指令数量较少时,隔空手势交互效率高于鼠标连续移动,但两者之间的差距在指令数量增多时有所减小,这是因为隔空手势需要用户始终举着一只手,在指令较多、交互时间较长时容易让人产生疲劳感,而鼠标交互方式则没有此问题。

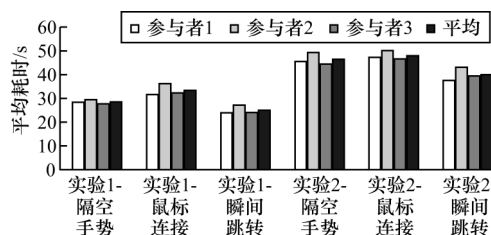


图 19 不同交互方式的平均耗时对比

Fig.19 Average time consumption comparison among different interaction methods

实验结果表明,与隔空手势交互方式相比,本文系统实现的鼠标交互方式在没有降低交互效率的同时减轻了用户使用时的疲劳感,在长时间交互中优势将逐渐凸显。而本文提出的鼠标跨窗口瞬间跳转的交互方式在避免疲劳的基础上还显著提高了交互效率,验证了本文系统实现多模态交互方式的正确性、有效性和必要性。

3.5 与相关工作效率对比

本文设计3组任务对本文系统和对照系统(文献[19]提出的系统)进行工作效率对比。对照系统与本文系统的实验环境相同,运行Unity版本、AR眼镜、计算机CPU、内存、显卡、操作系统均与本文系统相同。对照系统使用与计算机相连的有线键鼠,本文系统使用与AR眼镜通过蓝牙相连的无线键鼠。

本文系统和对照系统在初始状态下均使用2个虚拟的桌面画面窗口和1个真实电脑窗口,其中真实窗口在中间,虚拟窗口在两侧并正对使用者。本文实验的窗口布局如图20所示。为保证实验结果的准确性,本文实验分别在对照系统和本文系统中搭建相似窗口布局的实验环境。

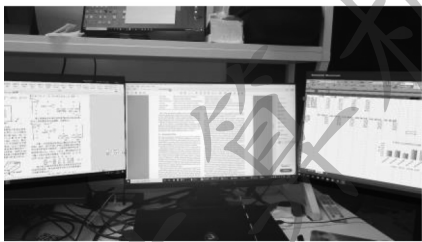


图20 本文实验的窗口布局示意图

Fig.20 Schematic diagram of window layout of the experiment in this paper

本文系统的窗口布局可由使用者自行修改,而对照系统窗口布局是固定的。其中对照系统的3个窗口均为同一台计算机的3个分屏窗口,本文系统的1个真实电脑窗口运行在一台计算机上,2个虚拟桌面窗口呈现的是2台不同计算机上的远程桌面画面。实验共分为3个任务,3名参与者每人均需在2个系统中分别完成3个任务。

任务1是数字对比,3个窗口各显示一个Excel文件,其中左右两个虚拟窗口各有50个数,按照顺序一一对应,其中有30%的数字不同,其余相同。实验者需要将50个数录入到中间窗口的空表中,相同数字直接录入,不同数字录入较小的那一个。

任务2是根据指令点击指定按钮。在3个窗口内分别显示图18中的4个按钮,编号为1~4,顺序为乱序的。实验者需要根据指令点击窗口的对应按钮,指令长度为20。

任务3是文件归档。左侧窗口的桌面有4个名为A、B、C、D的文件夹,以及20个文件,文件名分别为1~20的数字,文件格式为pdf、txt或doc。右侧窗口有一个需求文档,包含20个文件的归档要求,归

档要求形如“将所有编号能被3整除的pdf文件放入文件夹B”。实验者需要根据归档要求,将所有文件拖入指定文件夹进行归档。

对照系统和本文系统对各任务的平均耗时对比如图21所示。对于任务1和任务2,相比对照系统,本文系统的平均耗时分别降低了10%和33%。对于任务3,本文系统与对照系统的平均耗时相近。实验结果表明,对于任务2,实验者使用鼠标和手势同时进行交互的方式,大幅提升跨窗口交互的效率。而对于任务1和任务3,实验者均根据需要调整了虚拟窗口的位置,方便对比和查看。

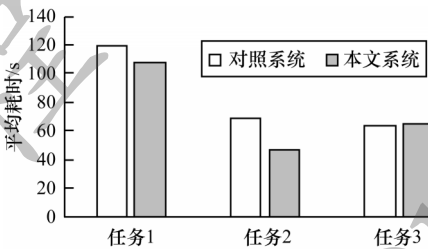


图21 本文系统和对照系统的平均耗时对比

Fig.21 Average time consumption comparison between the proposed system and contrast system

实验结果表明,与对照系统相比,本文系统通过多模态的交互方式以及多窗口布局策略,在常见的办公任务中显著提升了交互效率,验证了本文系统设计、功能和算法的先进性。此外,相比对照系统,本文系统支持同时显示并操作多台计算机,而对照系统仅支持同时显示一台计算机的多个桌面窗口。

4 结束语

本文基于AR眼镜设计远程桌面增强显示系统,以支持多模态交互。通过构建窗口布局计算模型,根据用户自定义参数自动调整多窗口的布局,并增加基于键盘鼠标的跨窗口交互方式以及结合视线检测的语音输入方式,在丰富交互方式的同时提高交互效率和准确性,实现交互空间的统一。实验结果表明,与传统隔空手势交互、鼠标交互等方式相比,本文系统具有较高的交互效率。下一步将优化远程画面传输的网络模块和编解码模块,并设计合理的调度策略,以解决当显示大量远程桌面虚拟窗口时存在AR眼镜CPU性能瓶颈的问题。

参考文献

[1] BI X J, BALAKRISHNAN R. Comparing usage of a large high-resolution display to single or dual desktop displays for daily work[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, USA: ACM Press, 2009: 1005-1014.

[2] ENS B M, FINNEGAN R, IRANI P P. The Personal Cockpit: a spatial interface for effective task switching on

- head-worn displays [C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, USA; ACM Press, 2014; 3171-3180.
- [3] ENS B, IRANI P. Spatial analytic interfaces; spatial user interfaces for in situ visual analytics[J]. IEEE Computer Graphics and Applications, 2017, 37(2): 66-79.
- [4] BELLGARDT M, PICK S, ZIELASKO D, et al. Utilizing immersive virtual reality in everydaywork[C]//Proceedings of the 3rd Workshop on Everyday Virtual Reality. Washington D. C. , USA; IEEE Press, 2017; 1-4.
- [5] FEREYDOONI N, WALKER B N. Virtual reality as aremote workspace platform; opportunities and challenges [C]//Proceedings of Microsoft New Future of Work Virtual Symposium. New York, USA; [s. n.], 2020; 1-8.
- [6] OFEK E, GRUBERT J, PAHUD M, et al. Towards a practical virtual office for mobile knowledge workers [EB/OL]. [2021-04-15]. <https://arxiv.org/abs/2009.02947>.
- [7] GRUBERT J, OFEK E, PAHUD M, et al. The office of the future; virtual, portable, and global[J]. IEEE Computer Graphics and Applications, 2018, 38(6): 125-133.
- [8] Facebook Company. Facebook Infinite Office [EB/OL]. [2021-04-15]. <https://www.oculus.com/blog/facebook-connect-oculus-quest-2-aaa-gaming-partnerships-and-more/>.
- [9] REKIMOTO J, SAITOH M. Augmented surfaces; a spatially continuous work space for hybrid computing environments[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, USA; ACM Press, 1999; 378-385.
- [10] Microsoft Company. Mirage; Virtual Monitors[EB/OL]. [2021-04-15]. <https://www.mirage-app.com/>.
- [11] REIPSCHLAGER P, FLEMISCH T, DACHSELT R. Personal augmented reality for information visualization on large interactive displays [J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 27(2): 1182-1192.
- [12] BUTSCHER S, HUBENSCHMID S, MÜLLER J, et al. Clusters, trends, and outliers; how immersive technologies can facilitate the collaborative analysis of multidimensional data [C]//Proceedings of CHI Conference on Human Factors in Computing Systems. New York, USA; ACM Press, 2018; 1-12.
- [13] LI Z, ANNETT M, HINCKLEY K, et al. HoloDoc; enabling mixed reality workspaces that harness physical and digital content[C]//Proceedings of CHI Conference on Human Factors in Computing Systems. New York, USA; ACM Press, 2019; 1-14.
- [14] REN J, WENG Y T, ZHOU C C, et al. Understanding window management interactions in AR headset+ smartphone interface [C]//Proceedings of Extended Abstracts of CHI Conference on Human Factors in Computing Systems. New York, USA; ACM Press, 2020; 1-8.
- [15] ZHU F Y, GROSSMAN T. BISHARE; exploring bidirectional interactions between smartphones and head-mounted augmented reality [C]//Proceedings of CHI Conference on Human Factors in Computing Systems. New York, USA; ACM Press, 2020; 1-14.
- [16] NORMAND E, MCGUFFIN M J. Enlarging a smartphone with AR to create a handheld VESAD (virtually extended screen-aligned display) [C]//Proceedings of IEEE International Symposium on Mixed and Augmented Reality. Washington D. C. , USA; IEEE Press, 2018; 123-133.
- [17] BIENER V, SCHNEIDER D, GESSLEIN T, et al. Breaking the screen; interaction across touchscreen boundaries in virtual reality for mobile knowledge workers [J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(12): 3490-3502.
- [18] SERRANO M, ENS B, YANG X D, et al. Gluey; developing a head-worn display interface to unify the interaction experience in distributed display environments[C]// Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services. New York, USA; ACM Press, 2015; 161-171.
- [19] PAVANATTO L, NORTH C, BOWMAN D A, et al. Do we still need physical monitors? an evaluation of the usability of AR virtual monitors for productivity work[C]// Proceedings of IEEE Virtual Reality and 3D User Interfaces. Washington D. C. , USA; IEEE Press, 2021; 759-767.
- [20] Microsoft Company. MRTK [EB/OL]. [2021-04-15]. <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtk-unity-2021-04>.
- [21] Microsoft Company. QR code tracking [EB/OL]. [2021-04-15]. <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/platform-capabilities-and-apis/qr-code-tracking>.
- [22] ENS B, HINCAPIÉ -RAMOS J D, IRANI P. Ethereal planes; a design framework for 2D information space in 3D mixed reality environments[C]//Proceedings of the 2nd ACM Symposium on Spatial User Interaction. New York, USA; ACM Press, 2014; 2-12.
- [23] BENKO H, FEINER S. Multi-monitor mouse [C]// Proceedings of CHI' 05 Extended Abstracts on Human Factors in Computing Systems. New York, USA; ACM Press, 2005; 1208-1211.