计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(3):125−132 [doi: 10.15888/j.cnki.csa.008974]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# 基于实时视频感知的虚拟体育交互系统①

陆启迪[1,2], 陈志祥[2,3], 魏　鑫[2,3], 高梓玉[1,2], 丁浩然[2], 赵海峰[2], 张　燕[1,2,3]

[1](南京师范大学 计算机与电子信息学院/人工智能学院, 南京 210023)
[2](金陵科技学院 软件工程学院, 南京 211169)
[3](南京邮电大学 计算机学院、软件学院、网络空间安全学院, 南京 210023)
通信作者: 赵海峰, E-mail: zhf@jit.edu.cn

**摘　要**: 针对疫情常态化背景下, 传统体育项目受场地、器材等限制, 市场上相关产品价格昂贵、可扩展性不足等问题, 提出了一种基于实时视频感知的虚拟体育交互系统. 该系统设计视频数据采集模块和人体关节点提取模块, 结合 OpenPose 获取人体的关节点坐标, 实时捕捉人体手势以及肢体动作. 动作语义理解模块包括运动动作理解和绘图动作理解. 前者根据运动中肢体关节点的相对位置关系, 识别运动动作语义. 后者将手腕部关节点绘图动作轨迹生成为草图图像, 使用 AlexNet 进行识别分类, 解析为对应的绘制动作语义. 该模型在边缘端设备的分类准确率为 98.83%. 采用基于 Unity 设计的草图游戏应用作为可视化交互界面, 实现在虚拟场景中的运动交互. 该系统使用实时视频感知交互方式实现居家运动健身, 无需其他的外部设备, 具有更强的参与度和趣味性.

**关键词**: 草图识别; 动作识别; 动作语义; 虚拟体育; 人机交互; 边缘计算

引用格式: 陆启迪,陈志祥,魏鑫,高梓玉,丁浩然,赵海峰,张燕.基于实时视频感知的虚拟体育交互系统.计算机系统应用,2023,32(3):125–132. http://www.c-s-a.org.cn/1003-3254/8974.html

## Virtual Sports Interaction System Based on Real-time Video Perception

LU Qi-Di[1,2], CHEN Zhi-Xiang[2,3], WEI Xin[2,3], GAO Zi-Yu[1,2], DING Hao-Ran[2], ZHAO Hai-Feng[2], ZHANG Yan[1,2,3]

[1](School of Computer and Electronic Information/School of Artificial Intelligence, Nanjing Normal University, Nanjing 210023, China)
[2](School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China)
[3](School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

**Abstract**: A virtual sports interaction system based on real-time video perception is proposed in response to the problems that traditional sports are limited by venues and equipment in the context of ongoing COVID-19 response, and the related products in the market are expensive and not scalable. The system is designed with a video data acquisition module and a human joint point extraction module, which can acquire human joint point coordinates in combination with OpenPose and capture human gestures and body movements in real time. The action semantic understanding module includes motion action understanding and drawing action understanding. The former recognizes the motion action semantics depending on the relative position relationship of the limb joints in motion. The latter generates the drawing action trajectories of wrist joints as sketch images, uses AlexNet to recognize and classify them, and resolves them into the corresponding drawing action semantics. The classification accuracy of the model is 98.83% in edge-side devices. A Unity-based sketch game application is used as the visual interaction interface to realize motion interaction in a virtual scene. The system adopts the interaction mode of real-time video perception to achieve home exercise and fitness without other external devices, which is more participatory and interesting.

**Key words**: sketch recognition; action recognition; action semantics; virtual sports; human-computer interaction; edge computing

① 基金项目: 江苏省高校自然科学研究重大项目 (21KJA520001); 江苏省国际科技合作项目 (BZ2020069)
收稿时间: 2022-07-22; 修改时间: 2022-08-26; 采用时间: 2022-09-16; csa 在线出版时间: 2022-11-18
CNKI 网络首发时间: 2022-11-23

## 1 引言

近期, 元宇宙作为互联网新地标矗立在科技变革最前沿[1], 虚拟体育也随之受到了越来越多的关注. 虚拟体育[2] 是将现代计算机科学技术与传统体育项目相结合的产物. 传统体育项目常常会受到场地、器材等条件的限制, 而虚拟体育应用可以让人们摆脱上述条件的束缚, 帮助人们随时随地进行安全科学的运动健身. 国际奥委会主席巴赫表示[3], “奥林匹克虚拟系列赛, 旨在增进与虚拟体育领域爱好者和观众的关系, 并进一步鼓励人们、特别是年轻人参与体育运动, 弘扬奥林匹克价值观”. 国家体育总局印发《“十四五”体育发展规划》, 鼓励体育信息化建设. 全民健身事业正处于高质量发展阶段, 采用现代科技信息技术推动全民健身走向更高水平是“十四五”时期体育发展的战略选择[4].

当前, 在新冠疫情背景下, 虚拟体育的发展迎来了新的契机. 虚拟体育的实现方式多种多样. 例如, Switch 健身环采用重力感应方式, 实现了动作指令传输和运动交互. Kinect[5] 采用 RGB-D 摄像头获取深度数据, 大大促进了动作识别以及虚拟体育领域的发展. An 等人[6] 从 Kinect 获取的骨骼数据中提取静态特征和动态特征, 并进行特征融合, 采用支持向量机 (support vector machine, SVM)[7] 分类, 从而实现动作识别. NVIDIA Jetson[8] 是一种嵌入式计算人工智能平台. Bokovoy 等人[9] 采用 NVIDIA Jetson 进行基于视觉的深度重建, 采用全卷积神经网络 (full convolutional neural network, FCNN), 引入增强功能, 提高模型推理效率. 近年来深度学习的发展使得如 C3D[10]、TSN[11]、I3D[12]、TSM[13] 和 SlowFast[14] 等基于视频的动作识别算法性能得到很大提升. 然而, 这些算法模型复杂, 计算量较大, 不易在边缘端设备上部署.

通过以上方式实现虚拟体育运动的成本较高, 需要专门的设备以及 PC 来实现. 同时, 不同的运动需要购买不同的外部设备, 支持的交互方式单一, 不具有通用性. 针对以上问题, 结合当前人工智能与物联网技术的发展, 本文提出了采用边缘设备进行实时视频感知的交互方式, 不需要其他的外部设备, 即可实现虚拟与现实交互. 本文的优点在于整体系统运行在边缘端, 支持多种客户端的连接, 能够让用户在足不出户的情况下进行体育锻炼, 成本也更加实惠.

本文提出了一个基于实时视频感知的虚拟体育交互系统, 该系统中设计了视频数据采集模块、人体关节点提取模块和动作语义理解模块. 视频数据采集模块通过边缘端设备实时采集用户的运动视频数据. 人体关节点提取模块结合 OpenPose[15] 获取人体骨骼关节点坐标, 将处理后的坐标数据输入动作语义理解模块. 该模块负责解析人体动作语义, 并将结果发送到客户端可视化应用, 实现虚拟场景中的运动交互. 动作语义理解模块包括运动动作理解和绘制动作理解. 前者根据运动中肢体关节点的位置相对关系, 识别运动动作语义. 后者获取人体手腕部关节点坐标, 并将绘图动作轨迹生成为草图图像, 训练草图分类[16] 深度学习模型, 将其识别为动作语义. 从而在前端可视化应用中实现绘图娱乐操作, 使得运动交互过程更具趣味性.

本文的主要贡献如下: 一是设计了一个虚拟体育交互系统, 采用边缘设备实现所有功能模块. 二是设计了一套自然人机交互方式, 提升用户的运动体验. 三是本应用系统进行大量的系统测试和用户研究, 验证了系统的可用性.

## 2 系统设计

### 2.1 系统架构

本系统设计了 3 个功能模块, 如图 1 所示, 分别为视频数据采集模块、人体关节点提取模块和动作语义理解模块. 视频数据采集模块负责采集人体运动视频数据. 人体关节点提取模块获取关节点坐标数据, 并进行归一化处理. 动作语义理解模块负责解析动作语义, 最终将解析结果传输至客户端应用, 在虚拟场景中实现运动交互. 下面将分别介绍系统各个模块的功能和流程.

### 2.2 视频数据采集模块

本文采用地平线 X3 SDB 开发板和 X3-CAM-01-S-H 摄像头作为采集单元. 地平线 X3 SDB 开发板搭载了全新一代 AIoT 边缘 AI 芯片旭日 3, 具有高性能、低功耗以及降低研发成本、加速应用落地等优点. 首先完成软硬件系统环境配置. 将 X3-CAM-01-S-H 摄像头与地平线 X3 SDB 开发板连接, 运行本系统脚本获取实时视频数据. 视频数据为 30 帧/s 的 RGB 彩色图像帧, 每帧图像大小为 1920×1080.

### 2.3 人体关节点提取模块

人体关节点提取模块, 结合 OpenPose[15] 将视频帧数据解析为关节点数据流, 实时获取人体骨骼关节点坐标. 基于开发板内置的数据可视化套件, 可将关节点解析结果在浏览器界面中显示. 该模块对手腕部关节点和其他关节点数据分别进行归一化、标准化等数据预处理操作, 再将处理后的数据传输至动作语义理解模块.

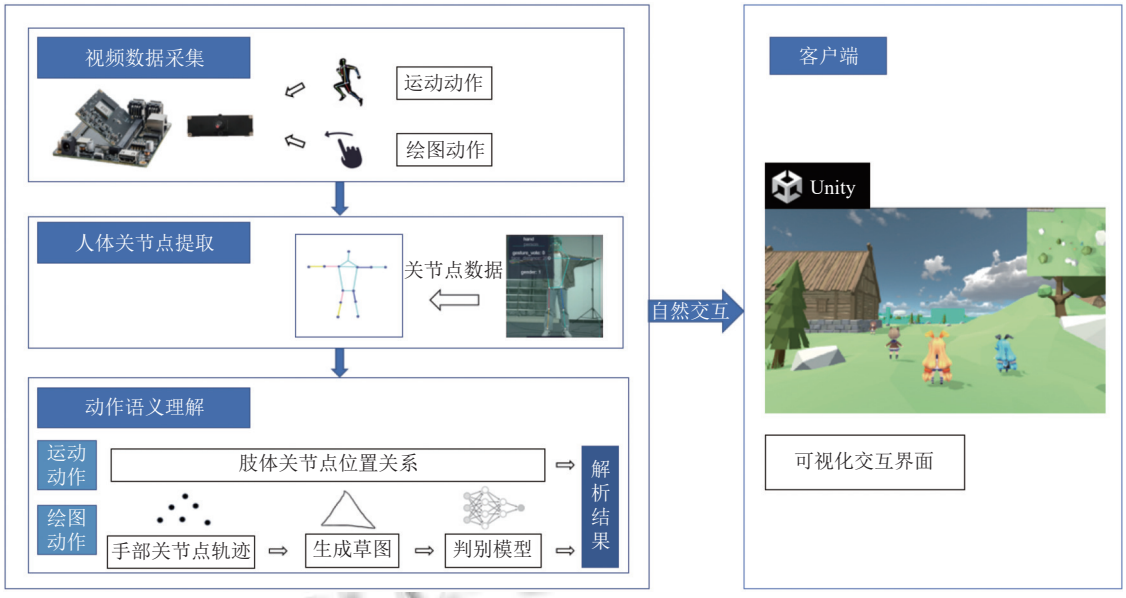



图 1 系统架构图

OpenPose[15] 是基于监督学习和卷积神经网络[17]的人体姿态估计算法. 它采用自下而上的方法, 提出了一种新的特征表示方法——部分亲和域 (part affinity fields, PAFs), 编码肢体部分的位置和方向信息. 每种类型的肢体都有相应的亲和域, 连接着与其相关的肢体部分. 同时, 在 CPM (convolutional pose machines)[18]的基础上, 结合中继监督优化连续阶段的预测结果, 采用部分置信图, 标记每个关节点, 通过匈牙利算法[19] 获取最优匹配, 最终获得实时高质量的检测结果.

结合我们的应用需求以及未来拓展的可能, 我们选择保留了人体 17 个骨骼关节点的检测 (头部眼鼻耳 5 个关节点、左手腕、右手腕、左手肘、右手肘、左肩、右肩、左髋、右髋、左膝、右膝、左脚踝、右脚踝). 在板端开启 Web Service 等服务, 将视频数据采集模块获取的 RGB 视频流作为输入, 结合 OpenPose[15] 检测人体关节点, 以图像左上角为坐标原点, 从板端读取关节点二维坐标数据, 保存跟踪的 17 个关节点数据, 采用 WebSocket 通信协议将数据流传输至服务端, 服务端通过 Python 实现 WebSocket 通信接收数据, 并做数据预处理, 效果如图 2 所示.

### 2.4 动作语义理解模块

动作语义理解模块主要由两个子模块构成: 一是绘图动作理解. 通过跟踪用户手腕部关节点, 连接并绘制手势轨迹, 生成草图图像, 再将草图输入分类模型进行识别, 最后解析为相应的动作语义. 二是运动动作理解. 通过各个肢体关节点的坐标位置关系, 设计奔跑、跳跃、后退、左转、右转、开始绘图等动作语义.

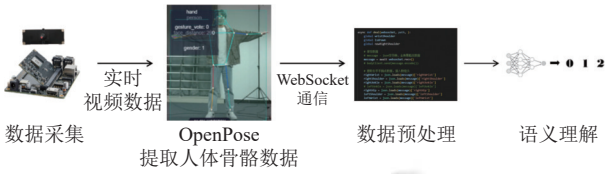



图 2 提取人体关节点

#### 2.4.1 基于草图分类的手势绘图动作理解

不同于传统直接通过视频帧进行动作识别的方法, 本文采用草图作为中间桥梁, 将用户手腕部关节点的手势轨迹, 生成为绘图动作的草图, 从而将动作识别问题转换为草图识别问题. 与基于视频帧方法相比, 本文的模型参数更少, 更加轻量, 易于量化部署在边缘端.

为训练草图识别模型, 本文自建了草图图形数据集, 形状类别为圆形、方形、三角形, 如表 1 所示. 为了使数据集中的草图更接近真实的手绘, 我们收集了用户绘制的真实草图作为样例, 邀请了 30 名用户为我们的草图数据集做出了贡献. 每个用户提供 3 种图形的 60–70 张手绘草图, 总计 5 942 张草图图像, 训练集和测试集的划分比例约为 7:3.

表 1 自构建草图图形数据集

| 形状 | 圆形 | 方形 | 三角形 |
|---|---|---|---|
| 数量 | 1 995 | 1 969 | 1 978 |

草图识别模型选用 3 种经典卷积神经网络架构进行对比实验, 分别为 ResNet[20]、VGGNet[21]、AlexNet[22]. 对比内容包括使用 PyTorch 框架训练的浮点模型精度, 以及经过模型转换后的混合异构模型精度, 如表 2 所示, 是 3 种经典模型的实验结果, 可以看出, 由于 AlexNet 相对于 ResNet 和 VGGNet 更加轻量, 对于只有 3 类的草图识别问题, 反而取得最好的识别效果.

表 2 采用不同骨干网络的模型准确率对比 (%)

| 网络 | 浮点模型 | 混合异构模型 |
|---|---|---|
| ResNet[20] | 96.9 | 85.6 |
| VGGNet[21] | 98.4 | 91.6 |
| AlexNet[22] | 99.4 | 98.83 |

最终模型采用 AlexNet 作为骨干网络, 通过 Kaiming 分布初始化, 并采用交叉熵作为损失函数对模型进行训练. 模型训练所使用的服务器系统环境为 Ubuntu 16.04, 硬件配置为 Tesla V100 GPU 32 GB×2. 草图分类模型训练流程如图 3 所示.

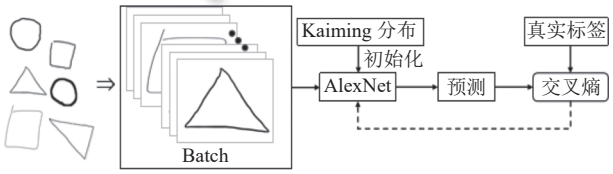



图 3 草图分类模型训练流程

### 2.4.2 模型转换

若想在开发板上调用深度学习框架训练的模型, 则需要完成模型转换的操作. 首先将用 PyTorch 框架训练的草图分类识别模型转换为 ONNX 模型格式, 再将 ONNX 模型转换为可以在开发板上使用的混合异构模型, 具体分为 4 个步骤: 模型验证、数据校准、模型转换以及性能评估.

模型验证: 将由 PyTorch 训练的浮点模型转换而成的 ONNX 模型文件传入开发板, 运行模型验证脚本, 检验模型是否符合工具链要求.

数据校准: 通过调整数据预处理方式, 对输入数据进行数据校准操作, 系统中的草图分类模型采用了归一化和标准化等操作进行数据预处理.

模型转换: 为了使得转换的模型在开发板上高效运行, 需要完成模型的量化及优化, 编写配置文件并运行模型转换脚本, 将浮点模型转换为适用于开发板的混合异构模型.

性能评估: 在应用部署前, 需要验证模型性能是否达到应用要求, 通过性能调优测试, 我们的草图分类模型精度已符合应用需求.

### 2.4.3 基于关节点位置关系的运动动作理解

本系统的运动动作包括奔跑、跳跃、后退、右转、左转、开始绘图等 6 种动作, 提出基于关节点位置关系的方法理解运动动作语义.

该方法采用滑动窗口缓存最近的视频帧数据, 用于判断当前的运动动作语义. 当滑动窗口内所有相邻的视频帧都符合某一运动动作语义所要求的参数时, 则判定为该运动动作, 否则设置当前动作为静止.

在系统实现中, 保存最近 6 帧的全身关节点数据为 $P$, 包括了关节点坐标: 头部$(x_h,y_h)$, 左手腕$(x_{lw},y_{lw})$, 右手腕$(x_{rw},y_{rw})$, 左肩$(x_{ls},y_{ls})$, 右肩$(x_{rs},y_{rw})$, 左脚踝$(x_{la},y_{la})$, 右脚踝$(x_{ra},y_{ra})$, 右髋关节$(x_{rh},y_{rh})$. 设某一帧的时刻为$t$.

初始化: 适配人体与摄像头不同距离下的画面, 设身高$H=y_h-y_{ra}$, 阈值$\theta$表示运动幅度或容差范围, 其取值表示所占身高的比例.

奔跑: 用户在摄像头前奔跑, 如图 4 所示, 即$P$满足$|y_{ra}^{(t-1)}-y_{ra}^{(t)}|<\theta H$, 其中, $\theta=0.1$.


图 4 奔跑动作示意图

跳跃: 双脚跳离地面一定高度, 如图 5 所示. 为避免与奔跑动作判定混淆, 这里取髋部关节点判断, 即$P$满足$|y_{rh}^{(t-1)}-y_{rh}^{(t)}|<\theta H$ , 其中, $\theta=0.15$.


图 5 跳跃动作示意图

后退: 双手向身前平举, 手腕与肩膀坐标基本重合, 如图 6 所示, 即$P$满足:

$$\begin{cases} |x_{lw}-x_{ls}|<\theta H \\ |x_{lw}-y_{ls}|<\theta H \\ |x_{rw}-x_{rs}|<\theta H \\ |x_{rw}-y_{rs}|<\theta H \end{cases}$$，其中，$\theta=0.1$

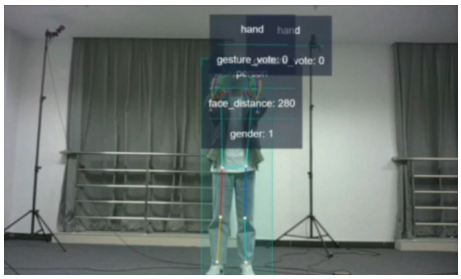



图 6　后退动作示意图

右转: 右手向身体右侧平举, 如图 7 所示, 即$P$满足$|y_{rw}-y_{rs}|<\theta H$，其中，$\theta=0.1$.

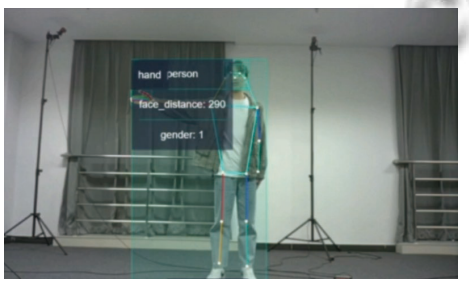



图 7　右转动作示意图

左转: 左手向身体左侧平举, 如图 8 所示, 即$P$满足$|y_{lw}-y_{ls}|<\theta H$, 其中, $\theta=0.1$.

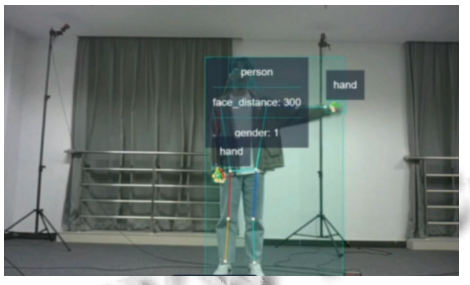



图 8　左转动作示意图

开始绘图: 双手向两侧平举, 即可进入绘图模式, 如图 9 所示, 即$P$满足$\begin{cases} |y_{lw}-y_{ls}|<\theta H \\ |y_{rw}-y_{rs}|<\theta H \end{cases}$, 其中, $\theta=0.1$.

进入绘图模式, 调用基于草图分类的绘图动作理解模型, 解析 3 种绘制草图 (圆形, 方形, 三角形) 的动作语义.

## 3　用户交互界面设计

本系统采用基于 Unity 设计的一款草图游戏来实现前端可视化应用交互, 支持 PC、平板等多种客户端设备, 用户可自行选择进行体验.

游戏设计界面如图 10 所示, 游戏场景为全开放式, 呈现为一座岛屿. 玩家角色位于界面中心位置, 在玩家角色顶端放置 Camera, 获取高空俯瞰视角, 以此设计小地图模块, 并在用户界面右上角展示, 如图 11 所示.


图 9　开始绘图动作示意图

图 10　Unity 草图游戏设计界面

图 11　草图游戏用户界面

玩家角色运动动作设计有: 奔跑、跳跃、后退、左转、右转、开始绘图等 6 种. 绘图动作包括绘制圆形、绘制方形和绘制三角形等 3 种. 绘制圆形将从玩家角色身前上空掉落一颗白球, 用于消除障碍; 方形则掉落一个木箱, 可以帮助玩家跨越河流; 三角形则掉落一棵松树, 用于后期小岛建设. 绘图草图生成三维模型如图 12 所示.

图 12　绘图草图生成三维模型

用户与游戏角色进行身份绑定, 在摄像头前做出指定动作, 通过系统各模块连接运行, 即可生成对应的动作指令. 用户可以在游戏中自由奔跑、跳跃, 通过手势绘制图形来生成对应模型, 从而获得更具趣味的运动体验.

## 4 系统测试

为了验证系统的可用性, 提供良好的用户体验, 我们进行了系统功能性测试和模型性能测试.

### 4.1 功能性测试

系统测试采用 PC 作为可视化交互设备, 系统环境为 Windows 10, 安装应用所需依赖环境, 以及 Unity 应用软件, 系统启动流程如下.

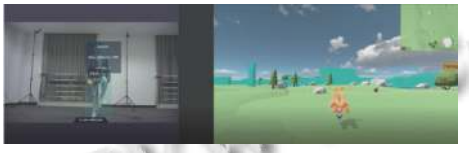
(1) 在边缘设备执行调用 server 文件, 启动通信服务.

(2) 通过运行 shell 脚本文件, 调用开发板 Web Service 等服务, 运行人体关节点提取模块和动作语义理解模块.

(3) 运行数据通信及数据处理的程序文件, 通过浏览器获取摄像头的视频流.

(4) Unity 启动草图游戏.

用户通过其肢体动作控制游戏中的角色, 通过跑、跳和双手向前平举控制角色的奔跑、跳跃和后退等交互, 如图 13 所示. 通过左手向左侧平举、右手向右侧平举和双手向两侧平举控制角色的左转、右转和开始绘图等交互, 如图 14 所示. 通过用户在空中画圆、画方和画三角, 在游戏中生成白球、木箱和松树等三维模型, 如图 15 所示. 整个系统运行流畅, 达到了实时性的要求.



(a) 奔跑

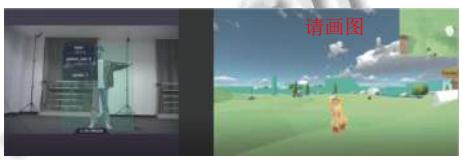
(b) 跳跃

(c) 后退

图 13 奔跑、跳跃、后退等交互效果

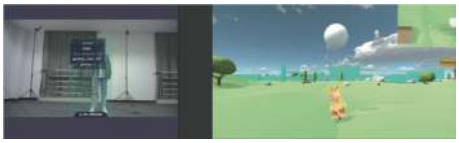
(a) 左转

(b) 右转




(c) 开始绘图

图 14 左转、右转、开始绘图等交互效果



(a) 绘制圆形

(b) 绘制方形

(c) 绘制三角形

图 15 绘制草图生成三维模型交互效果

### 4.2 模型性能测试

模型性能测试分为两组实验: 第 1 组实验测试了本文提出的运动动作理解和绘图动作理解算法. 第 2 组实验比较了本文与当前 5 种基于视频的深度学习算法的性能.

#### 4.2.1 动作语义理解

本系统的动作语义理解模型实时运行在边缘端设备上, 采用人与系统实时交互的方式, 测试模型的准确率. 对于每一类动作, 包括运动动作和绘图动作, 测试人员在系统前执行 200 次动作, 统计系统对每类动作的正确响应次数, 以此作为本系统动作语义理解模型的准确率. 测试结果如表 3 所示. 可以看出, 本系统提

出的动作语义理解模型具有较高的准确率, 满足用户的实时交互需要.

表 3 动作语义理解准确率测试

| 动作语义 | 正确分类 | 错误分类 | 总测试数 | 识别准确率 (%) |
|---|---|---|---|---|
| 奔跑 | 197 | 3 | 200 | 98.5 |
| 跳跃 | 197 | 3 | 200 | 98.5 |
| 后退 | 198 | 2 | 200 | 99.0 |
| 左转 | 198 | 2 | 200 | 99.0 |
| 右转 | 198 | 2 | 200 | 99.0 |
| 开始绘图 | 198 | 2 | 200 | 99.0 |
| 圆形 | 198 | 2 | 200 | 99.0 |
| 方形 | 197 | 3 | 200 | 98.5 |
| 三角形 | 198 | 2 | 200 | 99.0 |
| 总计 | 1 779 | 21 | 1 800 | 98.83 |

#### 4.2.2 与基于视频的深度学习方法比较

我们选用了近年来深度学习中经典的 5 种动作识别模型, C3D[10]、TSN[11]、I3D[12]、TSM[13] 和 SlowFast[14], 进行了对比实验. 由于上述模型无法在边缘端设备上直接运行, 不能使用实时交互的方式进行准确率测试. 因此, 采用录制视频的方式作为替代.

我们录制了用户的交互动作视频, 制作成视频数据集, 在服务器上进行测试. 数据集包含系统交互所需的 9 类动作, 每类动作 150 个视频, 一共 1 350 个视频. 对于每一类动作, 将其中的 105 个视频作为训练, 其余 45 个视频作为测试. 使用 MMAction2[23] 框架调用预训练模型, 并在上述数据集上进行微调, 所有参数保持与预训练模型相同, 验证识别模型的性能.

测试结果如表 4 所示. 可以看出, 基于视频的深度学习模型在 Top1 准确率上与本文模型有较大差距. 这是由于本文模型是在人体关节点检测的基础上运行, 运动动作理解模型计算关节点的相对位置关系, 绘图动作理解直接将动作轨迹转换为草图, 问题复杂度降低, 算法受外界干扰较小. 而深度学习模型直接从视频帧提取特征, 具有较高的复杂度. 在 Top5 准确率上, 深度学习模型有较好的识别性能.

表 4 识别模型性能对比 (%)

| 模型 | Top1-acc | Top5-acc |
|---|---|---|
| C3D[10] | 75.56 | 90.12 |
| TSN[11] | 73.33 | 97.04 |
| I3D[12] | 79.51 | 100 |
| TSM[13] | 86.17 | 100 |
| SlowFast (SlowOnly)[14] | 78.77 | 99.75 |
| 本文 | **98.83** | — |

## 5 用户研究

我们的应用系统已经在连接边缘设备的 PC 机上完全实现. 我们邀请了 60 位用户来体验我们的应用, 参与者构成多样, 有涉及相关领域研究的学者, 也有从未涉及此领域的体验者. 在进行应用体验之前, 我们为每位参与者准备了如何使用本应用系统的教程. 每位参与者可以自由做出所设定的交互动作进行运动体验.

我们邀请上述用户在体验后填写一份简单的调查问卷, 为我们的用户研究提供依据. 调查问卷主要内容包括: 用户个人信息, 用户运动健身背景调查, 用户体验满意度以及对应用的建议等.

调查问卷的结果表明, 60 位用户中 50 人表示是有趣的体验, 8 人表示体验一般, 2 人表示体验感不足; 收集用户对于应用的建议大致归纳如下: (1) 游戏人物动作设计比较简单, 动作指令不够丰富; (2) 绘图生成的图形模型较少, 缺乏开放性; (3) 游戏界面设计比较粗糙, 期待优化游戏画面. 调查问卷部分数据可视化如图 16 和图 17 所示. 图 17 中从左到右, 蓝色表示关于草图游戏界面设计的建议, 绿色表示关于交互动作指令的建议, 黄色表示关于草图图形类别的建议, 红色表示关于系统整体灵敏度的建议.

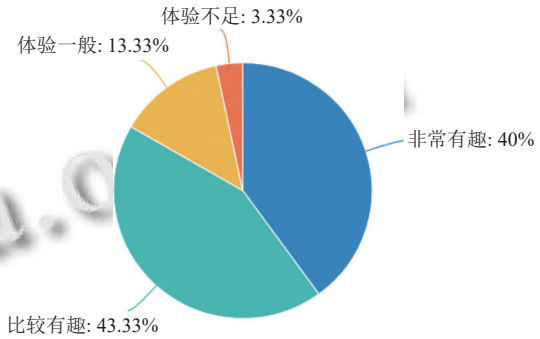



图 16 用户体验满意度调查

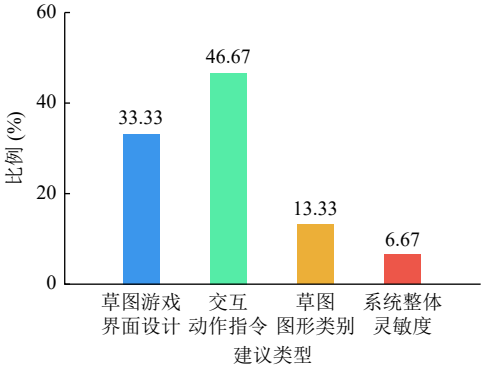



图 17 用户建议占比

## 6 结语

本文提出了一个基于实时视频感知的虚拟体育交互系统. 该应用系统结合 OpenPose 的人体姿态估计算法实时采集人体骨骼关节点数据, 提出基于关节点位置关系的运动动作理解实现运动交互; 同时, 设计一种基于草图分类的绘图动作理解模型实现娱乐绘图功能. 用户界面采用 Unity 设计的一款草图游戏来呈现. 系统测试和用户研究表明, 我们的系统开放性强, 趣味性高, 居家即可实现运动健身.

同时, 本应用系统也存有不足, 如运动、绘图等指令设计较少, 后续会通过更多的设计来丰富指令库. 此外, 目前用户界面设计较为初级, 后期会逐步更新迭代, 优化界面, 以提升用户的交互体验.

## 参考文献


1 喻国明, 耿晓梦. 元宇宙: 媒介化社会的未来生态图景. 新疆师范大学学报 (哲学社会科学版), 2022, 43(3): 110–118.

2 阚新玉. 虚拟体育的概念来源及其内涵分析. 贵州体育科技, 2006, (1): 11–14.

3 王剑冰. 史上首次!奥林匹克虚拟系列赛即将开赛. http://linyi.dzwww.com/tyyl/202104/t20210423_8394342.htm. [2022-09-05].

4 戴健, 史小强, 程华. “十四五”时期我国全民健身发展的环境变化与战略转型. 体育学研究, 2022, 36(5): 1–8.

5 石曼银. Kinect 技术与工作原理的研究. 哈尔滨师范大学自然科学学报, 2013, 29(3): 83–86. [doi: 10.3969/j.issn.1000-5617.2013.03.025]

6 An JH, Cheng XR, Wang Q, *et al*. Human action recognition based on Kinect. Journal of Physics: Conference Series, 2020, 1693: 012190. [doi: 10.1088/1742-6596/1693/1/012190]

7 Platt JC. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report, Redmond: Microsoft, 1998.

8 齐健. NVIDIA Jetson TX2 平台: 加速发展小型化人工智能终端. 智能制造, 2017, (5): 20–21. [doi: 10.3969/j.issn.1671-8186.2017.05.005]

9 Bokovoy A, Muravyev K, Yakovlev K. Real-time vision-based depth reconstruction with NVidia Jetson. Proceedings of 2019 European Conference on Mobile Robots. Prague: IEEE, 2019. 1–6.

10 Tran D, Bourdev L, Fergus R, *et al*. Learning spatiotemporal features with 3D convolutional networks. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 4489–4497.

11 Wang LM, Xiong YJ, Wang Z, *et al*. Temporal segment networks: Towards good practices for deep action recognition. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 20–36.

12 Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the Kinetics dataset. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 4724–4733.

13 Lin J, Gan C, Han S. TSM: Temporal shift module for efficient video understanding. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 7082–7092.

14 Feichtenhofer C, Fan HQ, Malik J, *et al*. SlowFast networks for video recognition. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 6201–6210.

15 Cao Z, Simon T, Wei SE, *et al*. Realtime multi-person 2D pose estimation using part affinity fields. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 1302–1310.

16 Eitz M, Hays J, Alexa M. How do humans sketch objects? ACM Transactions on Graphics, 2012, 31(4): 44.

17 LeCun Y, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278–2324. [doi: 10.1109/5.726791]

18 Wei SE, Ramakrishna V, Kanade T, *et al*. Convolutional pose machines. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 4724–4732.

19 Kuhn HW. The Hungarian method for the assignment problem. Naval Research Logistics, 2005, 52(1–2): 7–21.

20 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

21 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proceedings of the 3rd International Conference on Learning Representations. San Diego: ICLR, 2015.

22 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2012. 1097–1105.

23 MMAction2 Contributors. OpenMMLab ’s next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2. [2022-09-05].


(校对责编: 孙君艳)