北京航空航天大學
BEIHANG UNIVERSITY

科研课堂课程报告

基于大模型的 VR 场景意图识别

学院 人工智能学院

专业 人工智能

学号 23373436

姓名 唐晓阳

2025 年 7 月 2 日

1 Introduction

1.1 流程

我们通过眼动、手势、语音三方面信息作为输入，通过大模型的输出与场景做出意图识别交互。

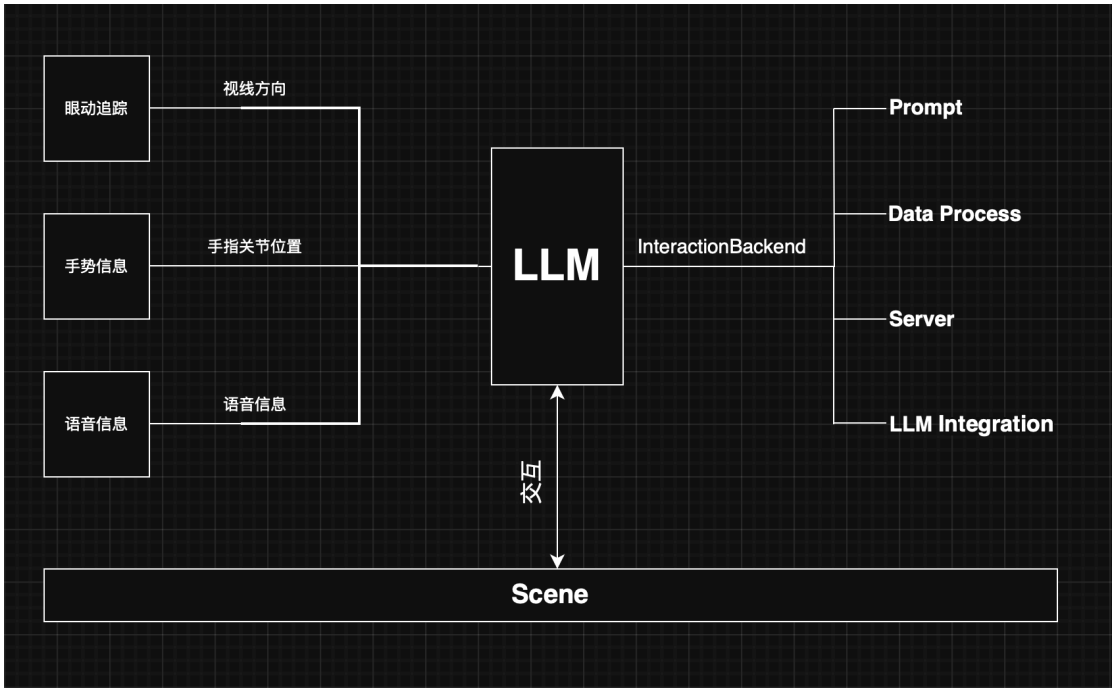

图 1 流程

1.2 工作内容

- 语音识别与信息提取
- 数据处理和集成
- Prompt
- 场景实验

1.3 开发基础

图 2 硬件设备：PICO 4 Pro

图 3 开发平台:Unity

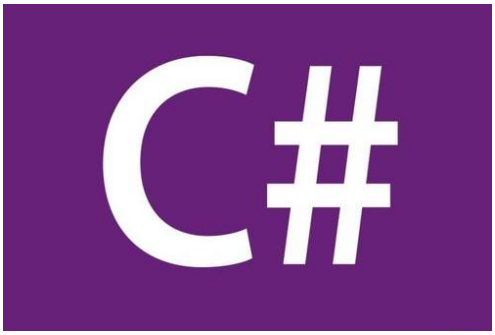

图 4 信息识别：Csharp

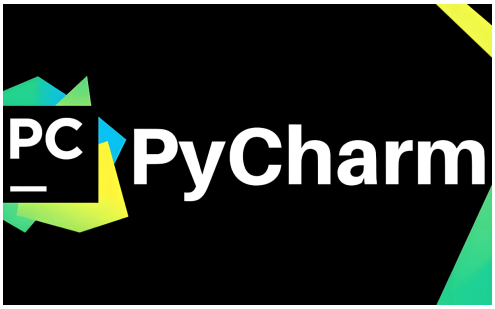

图 5 交互后端：Python

2 语音识别与信息提取

该部分主要实现了 4 个功能:Unity 不限时长录音、关键词识别与标注、实时接收结果、VR 语音识别流程

2.1 不限时长录音

此部分对应代码中的 micUnlimitedDuration.cs。实现了 Unity 中的无限时长麦克风录音功能，通过动态管理音频缓冲区突破 Unity 原生录音时长限制，支持录音控制、音频回放和 WAV 文件保存。

1. 启动录音

2. 动态数据保存（突破时长限制的关键）

- 双区交替保存策略：

当录音位置达缓冲区前 60% → 保存前半段数据（0 50%)

当录音位置达缓冲区后 10% 50% → 保存后半段数据（50% 100%）

- 循环操作避免数据覆盖：

”不限时长录音”

```
while (!isMicRecordFinished) {
    if (isSaveFirstHalf) {
        // 等待位置进入60%-100%区间
        micClip.GetData(micDataTemp, 0); // 取前半段
    } else {
        // 等待位置进入10%-50%区间
        micClip.GetData(micDataTemp, length / 2); // 取后半段
    }
    micDataList.AddRange(micDataTemp); // 动态累积数据
}
```

3. 停止录音

4. 保存 WAV 文件

2.2 关键词识别和标注

此部分对应代码 ReadTextMannager.cs

Key Step：SclectKeyWord 算法

采用增量式字符匹配策略：

1. 将文本内容转为字符数组逐字符处理

2. 动态构建当前检测词 buffWord

3. 实时检查是否匹配关键词前缀

4. 对完整匹配词添加高亮标签

2.3 实时接收识别结果

此部分对应代码 TextConnect.cs

2.4 语音识别流程

此部分对应代码 SpeechtoText.cs

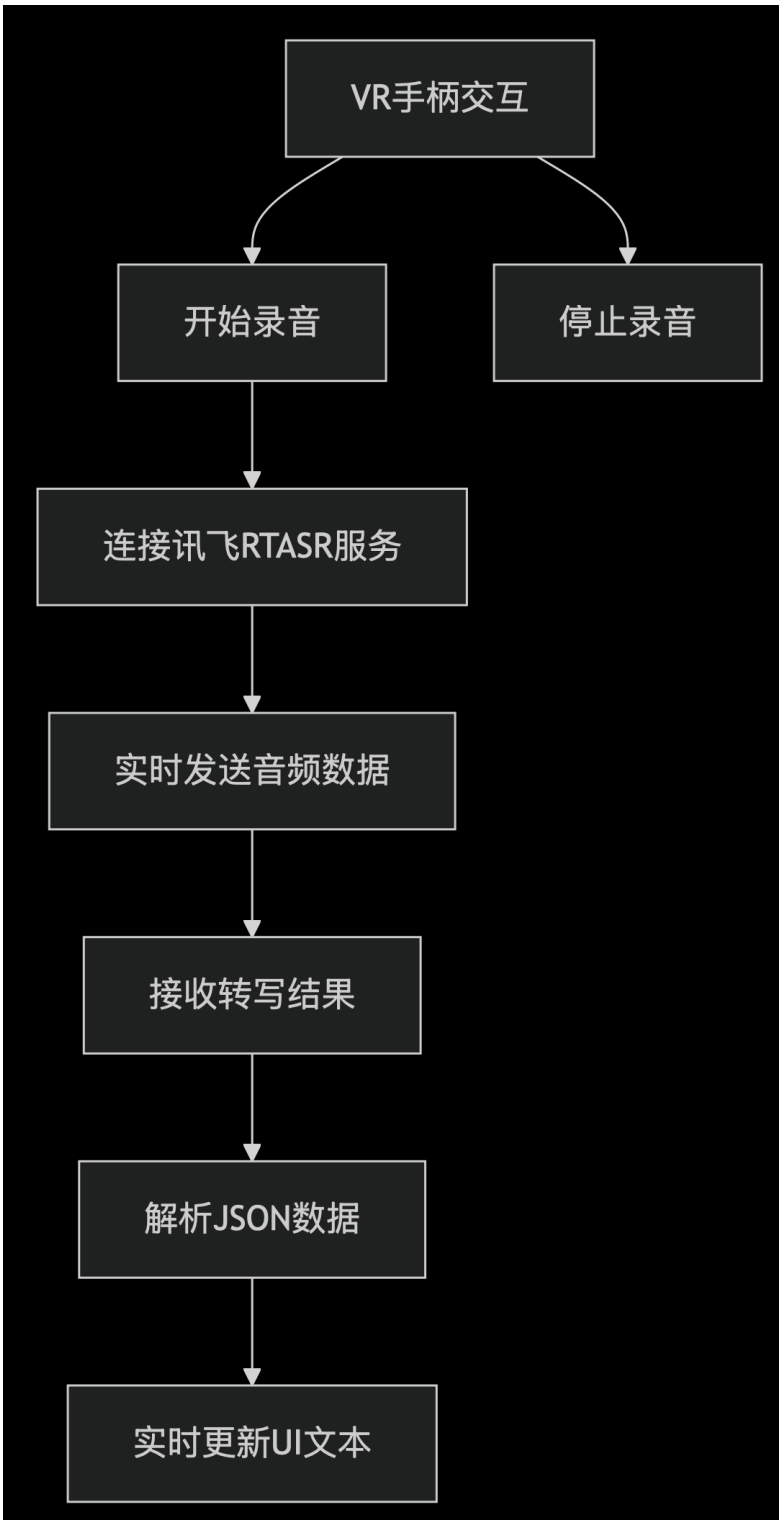

图 6 语音识别

3 数据处理

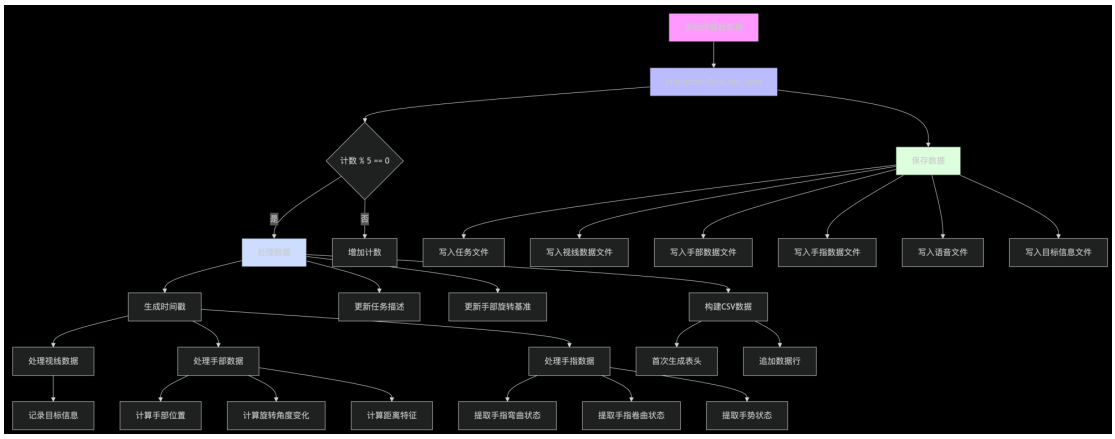

图 7 数据处理

3.1 手部数据处理与收集

手部数据处理

-提取左右手 3D 坐标 (x, y, z)

-计算手部旋转角度变化

-计算关键距离特征：手-头距离、左右手间距

手指数据处理

-提取每根手指的状态特征：弯曲度、卷曲度

-手势状态标志：捏合、指向

3.2 数据处理核心

”数据处理”

```
def add_data(self, data):
    # data = json.loads(line.strip().replace("'", "\""))
    if self.cnt == 0:
        self.prev_left_hand_rot = data['hand_info']['left_hand_rot']
        self.prev_right_hand_rot = data['hand_info']['right_hand_rot']

    if self.cnt % 5 == 0:
        gaze_res = dict()
        hand_res = dict()
        finger_res = dict()
```

```
        gaze_res['timestamp'] = self.cnt // 5
        hand_res['timestamp'] = self.cnt // 5
        finger_res['timestamp'] = self.cnt // 5
        gaze_res.update(data['gaze_info'])
        hand_res.update(processing_hand_info(data['hand_info'], self.
            prev_left_hand_rot, self.prev_right_hand_rot))
        finger_res.update(processing_finger_info(data['finger_info']))

        self.task = data['task']

        self.prev_left_hand_rot = data['hand_info']['left_hand_rot']
        self.prev_right_hand_rot = data['hand_info']['right_hand_rot']

        if self.is_empty:
            self.gaze_data = ','.join(gaze_res.keys())
            self.hand_data = ','.join(hand_res.keys())
            self.finger_data = ','.join(finger_res.keys())
            self.is_empty = False

        self.gaze_data += '\n' + ','.join(str(value) for value in gaze_res.
            values())
        self.hand_data += '\n' + ','.join(str(value) for value in hand_res.
            values())
        self.finger_data += '\n' + ','.join(str(value) for value in
            finger_res.values())
        self.target_info[data['gaze_info']['target_name']] = data['
            target_info']

    self.cnt += 1
```

3.3 Prompt

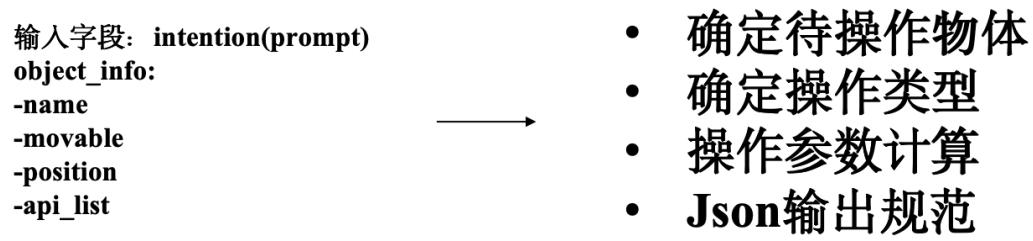

图 8 提示词主要内容

”提示词”

```
operation_prompt_end = """
字段解释:
intention: 表示用户的交互意图，即你需要完成的操作任务;
objects_info: 所有相关的物体的列表，每一个元素为一个相关物体，其中: name为物体名称，movable为物体能否位移，position为物体位置坐标，api_list为物体可调用的api列表。

坐标系补充说明: xz平面为水平面（x轴为左右方向，z轴为前后方向），y轴为高度轴。

你需要按照以下步骤进行分析:
一、确定唯一的待操作物体
一般来说，如果意图中只有一个物体，那就仅操控该物体；如果意图中有两个物体，请根据常识，选择操控其中一个物体。即无论怎样，都仅操控一个物体实现交互意图，将该物体确定为待操作物体（controlled_object）。

二、确定操作类型
操作类型分为两种: 1、对待操作物体进行位移; 2、调用待操作物体的api
如果交互意图中涉及到需要移动、旋转待操作物体的操作，且待操作物体的movable为True，且待操作物体的api_list中的api信息和交互意图不相关，则操作类型type=1（一般来说，如果交互意图中出现两个物体，大概率是此类操作类型）;
如果提供的待操作物体的api_list不为空，且其中某个api的名称和交互意图强相关，即若调用api即可完成交互意图，则操作类型type=2（你不能将交互意图强制和api_list中的某个api联系起来，如果确实找不到合适的api，则将操作类型type设为0）;
否则，如果你认为交互意图无法通过上述两种类型的操作完成，则设type=0

三、计算具体操作信息
```

```
若type=0:
则无需操作。
若type=1:
请计算待操作物体最终需要被移动到哪个位置（destination_position）、到达目标位置后是否需要让被操作物体做一些小范围运动（motion）、到达目标位置后是否需要倾斜被操作物体（incline）。
做小范围运动的含义为：若只是简单的移动、放置等操作，则到达目标位置就结束，motion=0；若类似于拿笔写字，刀切食物、擦拭物品等操作，根据常识当被操作物体到达目标位置后还需要做一些小范围移动（铅笔的左右移动、刀的上下移动等），这种情况下motion=1。
倾斜物体的含义为：若交互意图类似倒水、倒牛奶、倒咖啡、浇水等需要倾斜被操作物体，则incline=1；否则，incline=0。
此外，如果objects_info有两个物体，一般来说，则将待操作物体移到另一个物体处。
若type=2：请分析具体要使用api_list中的哪个api，api必须来源于提供的api_list，需要和提供的某个api名称一致，你不能凭空捏造。

四、按json格式输出
若type=0，输出的json格式为：
{"type": 0}
若type=1，输出的json格式为：
{"type": 1, "controlled_object": , "destination_position": , "motion": , "incline": }
若type=2，输出的json格式为：
{"type": 2, "controlled_object": , "api": }

controlled_object和api请使用中文，确保与提供的objects_info中对应的内容一致

无论如何，请严格按上述json格式进行输出，只需要输出json结果，结果前后也不要包含json相关说明，确保第一个字符为{，最后一个字符为}，保证整个输出内容能够被json模块解析。分析过程由你内部完成，不需要提供给我。
"""
```

4 实验

此部分见录制视频