

实验二 动作捕捉系统与协同灭火

一、 实验目的

1. 掌握光学动作捕捉系统及 VICON 软件的使用。
2. 了解 ROS 系统、话题的基本概念。
3. 掌握无人机与 EP 机器人结合动作捕捉系统的定位方法。

二、 实验设备

1. 无人机与机器人硬件：Tello Talent 无人机，Robomaster EP 机器人，笔记本电脑。
2. 软件：Ubuntu20.04+ros-noetic+VICON+python3+Robomaster SDK。

三、 实验原理

3.1 Vicon 光学动作捕捉系统

VICON 光学动作捕捉系统（Optical Motion Capture System）是一种用于捕捉、记录和分析人体和物体运动的技术系统。该系统利用光学原理和相机阵列来跟踪特定的标记点或标记器件，以获取物体在三维空间中的位置和姿态信息。光学动作捕捉系统通常包括以下组件：

摄像头：系统使用多个高速和高分辨率的摄像头组成的阵列。这些摄像头通常被放置在一个房间或空间中，以获得全方位的视野。摄像头的数量和布局根据具体需求而定，可以根据场地大小和被跟踪对象的复杂性进行配置。

标记点或标记器件：被跟踪的对象或人体身上贴有特殊的标记点或标记器件。这些标记点可以是反光球、LED 灯、惯性传感器等。摄像头通过检测和跟踪这些标记点或标记器件的位置来确定物体或人体的姿态和运动。

软件系统：光学动作捕捉系统通常附带专用的软件套件，用于处理和分析从摄像头捕获到的数据。这些软件能够将标记点或标记器件的位置数据转换为可视化的三维模型，从而实现对运动的重建和分析。用户可以使用这些软件进行姿态估计、动作追踪、运动模拟等操作。

光学动作捕捉系统的工作原理是通过多个摄像头同时拍摄被跟踪对象或人

体上的标记点或标记器件。每个摄像头捕获到的图像被传输到计算机中，通过对这些图像进行分析和处理，系统能够确定标记点或标记器件在三维空间中的位置，并将其转换为具体的运动数据。

3.2 ROS 话题

Vicon 软件以 ROS 话题的方式发布刚体的位置信息，ROS 话题是机器人操作系统（ROS）中用于实现节点之间异步消息传递的机制。通过话题，节点可以发布和订阅与特定主题相关联的消息，实现在分布式系统中共享数据和信息的目的。话题采用发布者-订阅者模式，允许多个节点同时发布和订阅同一个话题，实现灵活的通信和系统设计。

ROS 话题消息常用命令：

```
rostopic -h （查看帮助）  
rostopic list （查看已经发布或订阅的话题）  
rosviz rqt_graph rqt_graph （查看各话题间的发布订阅关系）  
rostopic echo + 话题名称 (查看发布内容)  
rostopic type + 话题名称 (查看消息类型)  
rostopic pub [topic] [msg_type] [args] （在命令行发布话题内容，可以使用 tab 查看联想内容）  
rostopic show + 消息类型 （查看消息类型的细节，可以使用 tab 查看系统提供的各种消息类型）
```

四、 实验内容

4.1 VICON 动捕系统定位

动捕系统在使用前需要进行标定，标定的目的是建立捕捉系统的坐标系和物体的运动关系，以确保准确的姿态和位置数据。打开桌面上的 VICON TRACKER 软件，进入 CALIBRATE 菜单栏，点击 CALIBRATE CAMERAS 下的 START，此时所有相机的 LED 会闪烁。打开标定杆上方的开关，开启后标定杆上的标记点亮起，手持标定杆，走进 VICON 动捕视场中，朝相机挥舞标定杆，相机下方的显示屏上有圆形进度条，当采集到足够多的数据后相机会变绿，当所有相机 LED 都变绿

后说明标定完成，在 VICON TRACKER 中停止标定，软件会自行计算误差，Image error 误差小于 0.3mm 即可。

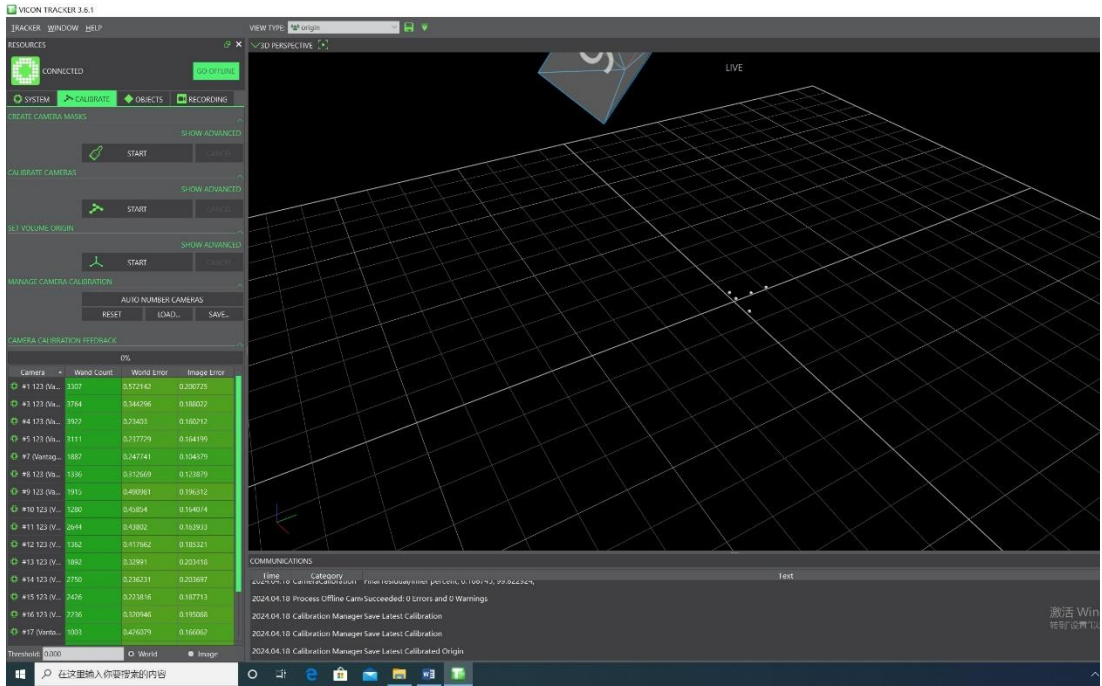


图 1 VICON 标定界面

完成相机标定后，设置坐标系，将标定杆放在地面上合适的位置，在 VICON TRACKER 软件中 SET VOLUME ORIGIN 栏下 点击 START，即可完成坐标系的设置，其中坐标原点为标定杆的交叉点，标定杆指向的方向分别为三维坐标系的 x、y 轴，垂直方向为 z 轴。

动捕标记小球的反光涂层材料可以帮助动捕系统跟踪和定位物体，在物体上安装三个及以上的标记小球即可被动捕系统准确追踪，动捕标记小球尽量不要对称安装，图 2 是一种安装方法。



图 2 动捕标记小球安装

为了确定无人机与消防机器人之间的位置关系，我们在 Tello Talent 无人机和 EP 移动机器人上都安装三个动捕标记小球。将安装了动捕标记小球的无人机和机器人放置在 VICON 动捕系统的视场中，在 VICON TRACKER 软件的视图中可以看到相应的标记球，如图 2 所示。

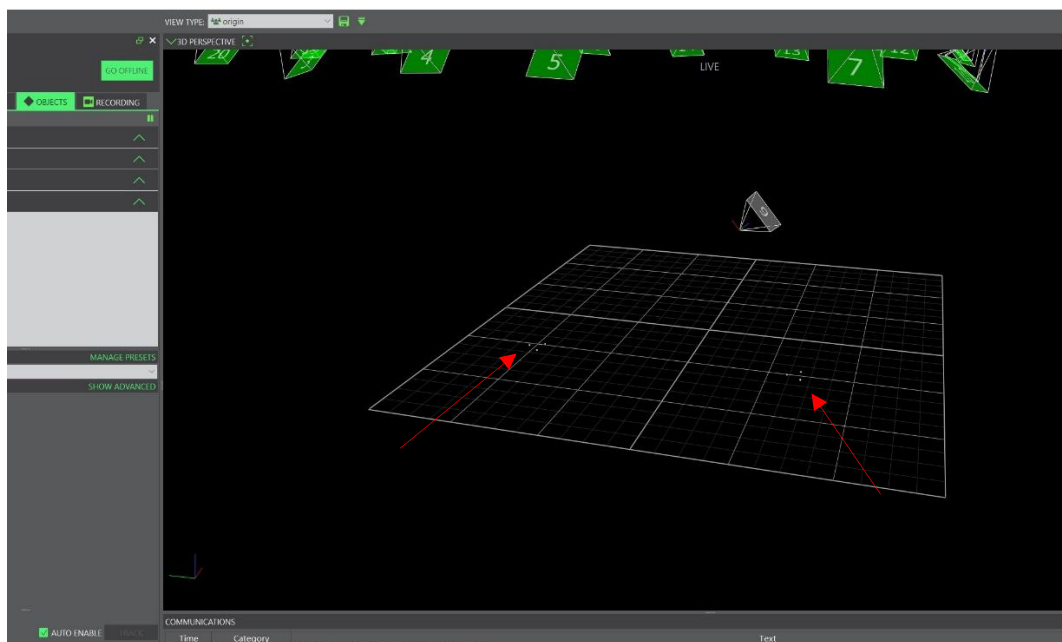


图 3 标记点在 VICON 视图下

为了获取无人机与 EP 机器人的位置信息，首先要在 VICON 中创建目标，选中无人机或 EP 机器人对应的三个标记小球（按住 ctrl 以选中多个标记点），选中之后标记点和相机之间有蓝色线条，在左侧的 OBJECTS 栏下的 Create Object 中输入名称后，点击右侧 Create 创建目标，如图 3 所示。

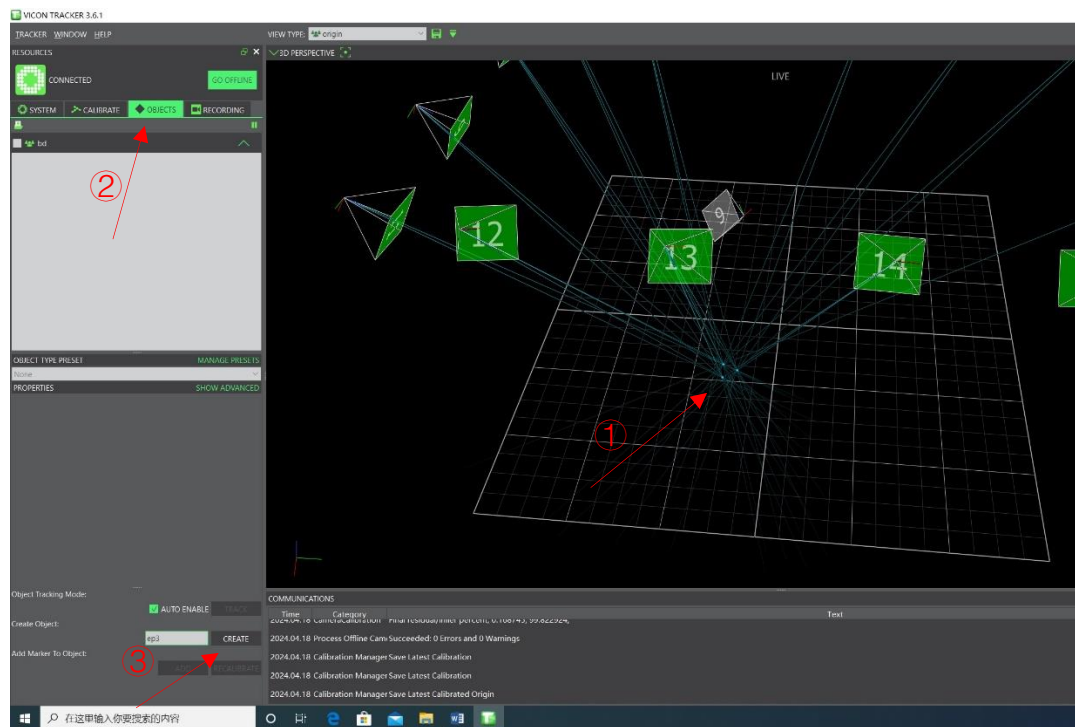


图 4 VICON 创建目标

Vicon 软件通过 WiFi 发布刚体的位置信息，我们下载其开源的 ROS 功能包后，可以通过连接其 Wifi 来接收刚体的位置信息。

由于 ROS 中的功能包需要位于 ROS 工作空间之下，因此先创建 ROS 工作空间，接着克隆功能包的源代码。打开一个新的终端：

```
mkdir -p ~/vicon_ws/src    # 创建工作空间 vicon_ws 的文件夹
cd vicon_ws/src
gitclone https://github.com/ethz-asl/vicon\_bridge    #克隆功能包的源代码
```

使用 ROS 提供的 catkin_make 命令编译工作空间

```
cd ~/vicon_ws
catkin_make
```

修改功能包内 `vicon_bridge/launch/vicon.launch` 文件中的“`datastream_hostport`”的值为“192.168.10.1:801”，将工作空间的路径添加到系统环境变量中：

连接名为“VICON”的 wifi，运行 `vicon.launch` 即可看到对应发布的位姿话题。

```
source ~/vicon_ws/devel/setup.bash  
roslaunch vicon_bridge vicon.launch
```

打开一个新终端，列出当前所有的 ROS 话题：

```
rostopic list
```

这时应该能看到名为 `/vicon/XX/XX` 的话题，即为我们在 VICON 软件中创建的刚体，使用 `rostopic` 命令查看该话题携带的消息数据：

```
rostopic echo /vicon/XX/XX
```

终端中会显示选定刚体的位置和姿态信息。

4.2 空地协同灭火

无人机通过直连模式连接 PC1，控制无人机在街区进行巡逻，发现火灾后悬停，消防机器人与另一台 PC2 通过组网模式连接至 VICON 动捕系统的 Wifi，这样 PC2 既可通过动捕接收到无人机与 EP 机器人的位置信息，又可以控制 EP 机器人的移动。由于组网模式下多个 EP 机器人连接至同一个 Wifi，为了区分开来，在初始化时应添加其唯一的 SN 号：

```
ep_robot.initialize(conn_type="sta", sn="XXXXXXXXXXXX")
```

其中 EP 机器人的 SN 号在中控台上方贴标处。

由于 VICON 动捕系统通过 ROS 话题发布目标的位置信息，在 ROS 中，要接收特定话题上发布的消息，需要编写订阅者来订阅相关话题，为了处理位置信息，首先要确定其数据类型，在终端中使用 `rostopic type` 查看 VICON 发布的信息类型：

rostopic type /vicon/XX/XX

发现该话题类型为 geometry_msgs/TransformStamped，TransformStamped 消息类型包含以下字段：

header: 消息头部，包含时间戳（stamp）和坐标系（frame_id）信息。

child_frame_id: 子坐标系的名称，表示相对于父坐标系的变换关系。

transform: 描述从父坐标系到子坐标系的变换信息，包括平移（translation）和旋转（rotation）。

其中 transform 部分中的 translation 数据是我们感兴趣的部分，它代表了目标相对于 VICON 坐标系的平移变换，也就是目标在 VICON 坐标系下的三维坐标信息，提取出该部分信息，就能够获取目标的三维坐标信息。rospy 是用于 Python 编程语言的 ROS 客户端库，我们通过 rospy 库来追踪 VICON 发布的无人机与 EP 机器人的话题，并提取出相关的位置信息。下面这段代码演示了如何获取无人机的位置信息

```
1. import rospy
2. from time import sleep
3. from rospy_message_converter import message_converter
4. from geometry_msgs.msg import TransformStamped
5.
6. def callback_tt(data):
7.     global x_tt
8.     global y_tt
9.     received_data = message_converter.convert_ros_message_to_dictionary(data)
10.    x_tt = received_data['transform']['translation']['x']
11.    y_tt = received_data['transform']['translation']['y']
12.
13. if __name__ == '__main__':
14.     rospy.init_node('listener', anonymous=True)#
15.     rospy.Subscriber('/vicon/tt/tt', TransformStamped, callback_tt)#
16.     while(1):
17.         sleep(1)
18.         print("x_tt:",x_tt, "y_tt:",y_tt)
```

其中，第 14 行初始化一个 ROS 节点，第 15 行创建一个订阅者对象，用

于订阅/vicon/tt/tt 话题，并将接收到的消息传递给名为 callback_tt 的回调函数进行处理。6-11 行是回调函数，用来处理接收到的消息，当接收到订阅的话题上发布的消息时，回调函数会被调用。7-8 行定义两个全局变量，分别表示目标的 x、y 坐标，第 9 行将消息数据转换成 python 下的字典类型，方便进行数据处理，10-11 行提取出 translation 下的 x、y 数据。

请自行在以上示例代码中添加代码，同时获取无人机和 Ep 机器人的位置信息，根据两者的位置相对关系控制移动机器人移动至无人机下方附近并发射水弹灭火。