

基于综合模型ROSTrans视觉图像转发实验

1. 实验目的

本实验旨在演示如何通过 RosTrans 中间件将 RflySim3D 渲染的视觉传感器数据（图像、点云等）转发为标准 ROS 主题，实现基于 ROS 的视觉数据接收与可视化；同时提供一个单机随机飞行控制脚本，使无人机在小范围空间内无限随机飞行，配合 UE4 跟随视角展示视觉传感器的动态采集效果。

主要目标：

- 了解 RosTrans 的 VisionAssistant 插件如何将 RflySim3D 视觉数据桥接为 ROS 主题
- 掌握 `Config.json` 视觉传感器配置文件的结构与参数含义
- 学会在 Rviz/Rviz2 中订阅和可视化 RGB 图像、LiDAR 点云、红外图像等传感器数据
- 理解 RflySim 视觉仿真中传感器 ID (SeqID) 与 ROS 主题命名的映射关系
- 掌握通过 ROS 主题控制无人机随机飞行，并使用 UE4 跟随视角实时观察飞行与视觉数据

2. 实验要求

- 软件要求：Windows 10 及以上版本；RflySim 工具链^[1]；WSL2 Ubuntu 环境（用于运行 ROS/ROS2 节点和 Rviz）。
- 硬件要求：笔记本/台式电脑 1 台^[2]（需支持 GPU 渲染）。

3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\4.RosTransExps\4.VisionSitransNoPx4](#)

4. 实验内容或步骤

4.1 启动仿真环境

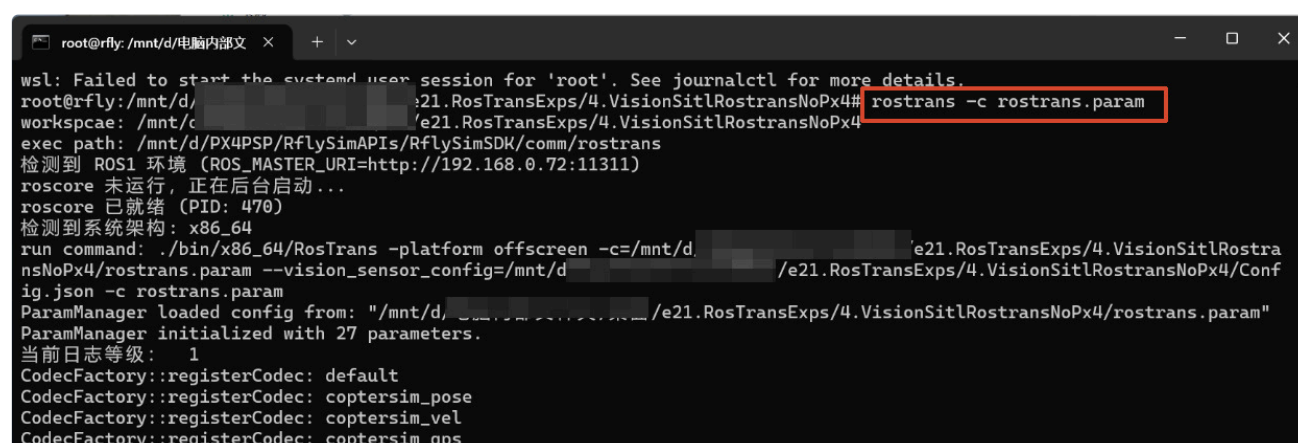
1、双击运行 `CopterSimN0px4.bat` 文件，等待仿真环境初始化完成。脚本将会启动 1 个 CopterSim、1 个 RflySim3D 软件，查看 CopterSim 软件，将显示使用 DLL 模型文件为 MulticopterN0px4，等待所有 CopterSim 窗口的仿真模式，均显示 `Simulink&DLL SIL` 模式，即表示初始化完成。



4.2 启动 RosTrans 中间件

2、在 WSL 中启动 RosTrans 并加载参数。双击 'WinWSL.bat'，在弹出的对话框中输入：

```
1 | rostrans -c rostrans.param
```



该指令将加载本目录下的 `rostrans.param` 配置文件。RosTrans 将自动：

- 连接 CopterSim (DLL Model 模式)，接收遥测数据并发布到 ROS 主题
- 启用 VisionAssistant 插件，读取 `Config.json` 配置，请求 RflySim3D 的视觉传感器数据
- 将接收到的图像和点云数据转发为标准 ROS 主题 (Image / PointCloud2)

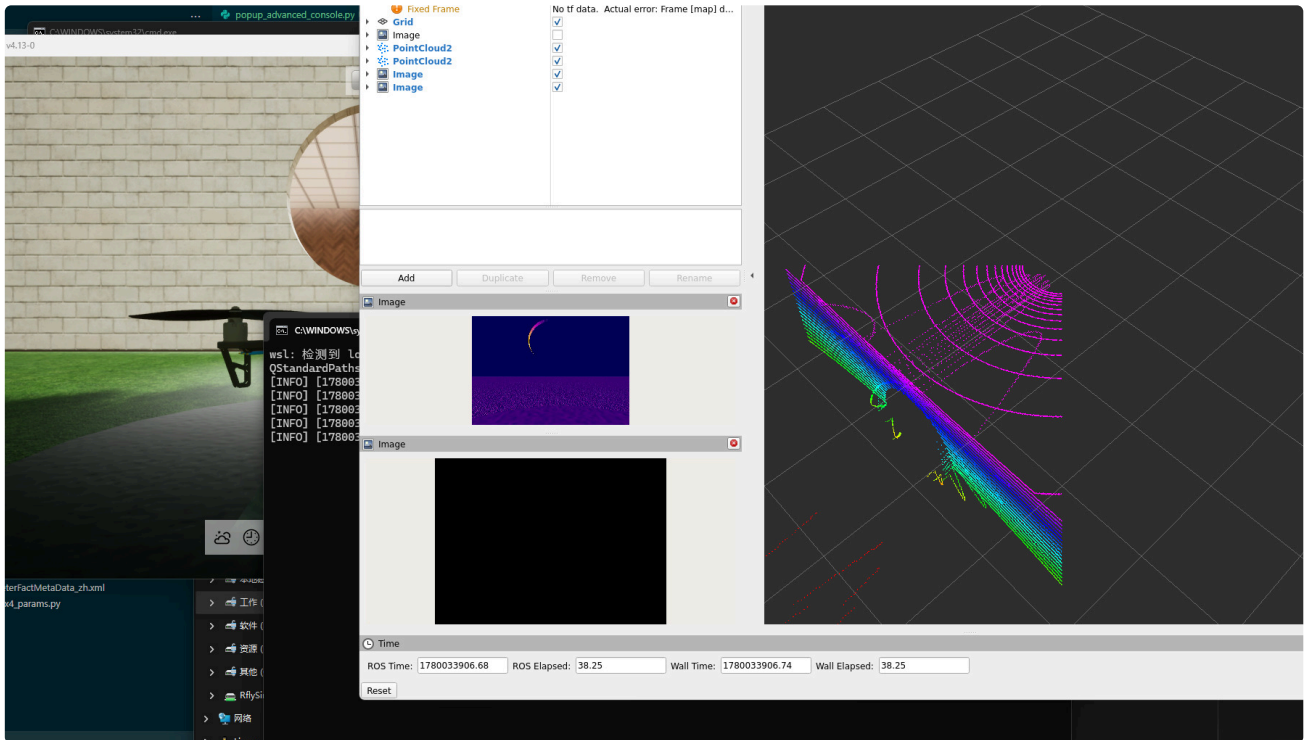
`rostrans.param` 关键配置说明：

```
1 | enable_dllModel=true           # 启用 DLL 综合模型 (无 PX4)
2 | enable_vision=true            # 启用视觉传感器转发
3 | vision_sensor_config=Config.json # 视觉传感器配置文件
4 | vision_sensor_req_ip=127.0.0.1 # RflySim3D 视觉请求 IP
5 | vision_imu_freq=200           # IMU 数据发送频率 (Hz)
```

4.3 启动 Rviz 可视化

3、双击运行 `RunWslRviz.bat`，脚本将自动启动 VcXsrv (X11 转发) 并在 WSL 中打开 Rviz/Rviz2，加载预配置的可视化布局。

```
C:\Windows\system32\cmd.exe
[INFO] [1780024906.978910211]: rviz version 1.14.25
[INFO] [1780024906.979043347]: compiled against Qt version 5.15.3
[INFO] [1780024906.979120114]: compiled against OGRE version 1.9.0 (Ghadamon)
[INFO] [1780024907.041914041]: Forcing OpenGL version 0.
1[INFO] [1780024909.736170487]: Stereo is NOT SUPPORTED
[INFO] [1780024909.736289116]: OpenGL device: D3D12 (Intel(R) Iris(R) Xe Graphics)
[INFO] [1780024909.736326793]: OpenGL version: 4.1 (GLSL 4.1) limited to GLSL 1.4 on Mesa system.
```



也可在 WSL 中手动启动（ROS 版本可在 4.2 步骤弹出的对话框中查看）。再次双击 '[WinWSL.bat](#)' 脚本，在弹出的对话框中运行如下代码：

```
1 # ROS1
2 rviz -d vision_ros1.rviz
3
4 # ROS2
5 rviz2 -d vision_ros2.rviz
```

Rviz 启动后将自动订阅各传感器主题并显示数据。

4.4 启动随机飞行控制脚本

4、再次双击 '[WinWSL.bat](#)' 脚本，在弹出的对话框中运行飞行控制脚本，使无人机在小范围空间内无限随机飞行：

```
1 # 使用默认参数（起飞高度 5m，飞行范围 ±5m，高度 3~8m）
2 python3 vision_copter_flight.py
```

可在 RflySim3D 中观察到无人机解锁起飞并开始随机飞行，同时 Rviz 中的视觉数据随飞行实时刷新。

也可再次双击 '[WinWSL.bat](#)' 脚本，在弹出的对话框中运行如下代码自定义参数：

```

1 | # 自定义参数
2 | python3 vision_copter_flight.py --alt 5 --range 6 --speed 1.5 --alt-min 2 --alt-max 10

```

脚本启动后将自动：

- 配置 UE4 为**跟随视角**（后方 4m, 上方 2m, 俯角 15°），始终跟随无人机飞行
- 显示飞行轨迹和无人机编号标签
- 解锁起飞至目标高度后，在设定范围内不断生成随机航点并飞往
- 每到达一个航点后随机悬停 0.5~2.0 秒，然后飞向下一个航点
- 按 **Ctrl+C** 后优雅退出：自动回到起始位置 → 逐步降落 → 上锁

命令行参数说明：

1	参数	默认值	说明
2	-----	-----	-----
3	--alt	5.0	起飞高度 (m)
4	--range	5.0	飞行范围半径 XY (m)，即东西/南北方向 ±range
5	--alt-min	3.0	随机飞行最低高度 (m)
6	--alt-max	8.0	随机飞行最高高度 (m)
7	--speed	1.0	巡航飞行速度 (m/s)
8	--wp-r	0.5	航点到达判定半径 (m)
9	--win	0	UE4 窗口 ID

4.5 实验观察

观察 RflySim3D 窗口和 Rviz 中的实时效果：

RflySim3D 窗口（UE4 跟随视角）：

- 相机始终跟随无人机后方，实时观察无人机的随机飞行轨迹
- 飞行轨迹以线条形式持续绘制在场景中

Rviz 窗口（视觉传感器数据）：

- **RGB 图像**：来自传感器 1 的前视摄像头画面，随飞行方向实时变化
- **3D LiDAR 点云**：来自传感器 2 的车载 LiDAR 扫描数据（彩色渲染）
- **Livox LiDAR 点云**：来自传感器 3 的 Livox 型 LiDAR 数据
- **红外灰度图像**：来自传感器 4 的红外灰度图
- **红外伪彩图像**：来自传感器 5 的红外伪彩色图

5. 关键知识点

5.1 视觉传感器配置（Config.json）

Config.json 定义了 5 个视觉传感器，通过 SeqID 标识：

SeqID	TypeID	传感器类型	分辨率	频率	ROS 主题
1	1	RGB 摄像头	640×480	30Hz	/rflysim/sensor1/
2	20	车载 LiDAR	900×32	10Hz	/rflysim/sensor2/

SeqID	TypeID	传感器类型	分辨率	频率	ROS 主题
3	22	Livox LiDAR	250×40	10Hz	/rflsim/sensor3/
4	40	红外灰度相机	640×480	30Hz	/rflsim/sensor4/
5	41	红外伪彩相机	640×480	30Hz	/rflsim/sensor5/

关键配置参数说明

```
1 {
2   "SeqID": 1,           // 传感器序号, 对应 ROS 主题中的 sensorN
3   "TypeID": 1,          // 传感器类型 (1=RGB, 20=LiDAR, 22=Livox, 40=IR灰度, 41=IR伪彩)
4   "TargetCopter": 1,    // 挂载目标飞机 ID
5   "TargetMountType": 0, // 挂载方式 (0=机体固连)
6   "DataWidth": 640,     // 数据宽度 (像素/扫描点数)
7   "DataHeight": 480,    // 数据高度 (像素/扫描线数)
8   "DataCheckFreq": 30,  // 数据请求频率 (Hz)
9   "CameraFOV": 90.0,    // 视场角 (度)
10  "SensorPosXYZ": [0.3, 0, 0], // 传感器安装偏移 (前,右,下), 单位米
11  "SensorAngEular": [0, 0, 0], // 传感器安装角度 (roll, pitch, yaw), 单位度
12  "SendProtocol": [1, ...], // 发送协议配置 (1=启用)
13  "otherParams": [...],    // 传感器特定参数 (LiDAR: 最大距离/角度范围等)
14 }
```

LiDAR 特有参数 (otherParams)

以传感器 2 (车载 LiDAR, TypeID=20) 为例:

索引	值	说明
0	200.0	最大探测距离 (m)
1	0.05	角度分辨率 (度)
2	-45.0	水平起始角 (度)
3	45.0	水平结束角 (度)
4	-20.0	垂直起始角 (度)
5	20.0	垂直结束角 (度)

5.2 ROS 主题命名规则

视觉传感器主题遵循以下命名模式:

```
1 | /rflsim/sensor{SeqID}/{数据类型}
```

数据类型	ROS 消息类型	传感器 TypeID
img_rgb	sensor_msgs/Image	1 (RGB)
vehicle_lidar	sensor_msgs/PointCloud2	20 (车载 LiDAR)
livox_lidar	sensor_msgs/PointCloud2	22 (Livox LiDAR)
img_Infrared_Gray	sensor_msgs/Image	40 (红外灰度)

数据类型	ROS 消息类型	传感器 TypeID
img_Infrared	sensor_msgs/Image	41 (红外伪彩)

5.3 IMU 与视觉传感器请求数据

除视觉数据外，RosTrans 还转发以下辅助数据：

主题	消息类型	说明
/rflsim/pod{SeqID}/vision_sensor	rflsim_msgs/VisionSensorReq	传感器请求/状态信息
/rflsim/camera{SeqID}/camera_data	rflsim_msgs/CameraData	相机参数与图像数据

5.4 随机飞行控制脚本（[vision_copter_flight.py](#)）

飞行控制原理

`vision_copter_flight.py` 使用 DllModelAssistant 插件（SimMode=3, 无 PX4 固件）控制单架无人机，通过 MAVROS 兼容的 ROS 主题进行通信。飞行流程分为 5 个阶段：

阶段	说明
Phase 0	等待遥测数据就绪 + 配置 UE4 跟随视角
Phase 1	发送 Pre-arm setpoint 流 → 切换 OFFBOARD 模式 → 解锁 → 起飞
Phase 2	爬升至目标高度，到达后悬停稳定 2 秒
Phase 3	无限循环 : 随机生成航点 → 飞向航点 → 悬停 → 下一个航点
Phase 4	<code>Ctrl+C</code> 触发: 回到起始位置 → 逐步降落 → AUTO.LAND → 上锁

ROS 控制主题

脚本使用以下 ROS 主题与服务：

方向	主题/服务	消息类型	用途
TX	/mavros/setpoint_raw/local	PositionTarget	发布位置/速度 setpoint
TX	/mavros/setpoint_position	PoseStamped	发布位置 setpoint (备用)
RX	/mavros/local_position/pose	PoseStamped	接收当前位置
RX	/mavros/state	State	接收飞行模式和解锁状态
SRV	/mavros/set_mode	SetMode	切换飞行模式
SRV	/mavros/cmd/arming	CommandBool	解锁/上锁
SRV	/mavros/cmd/takeoff	CommandTOL	起飞

方向	主题/服务	消息类型	用途
SRV	<code>/mavros/cmd/land</code>	<code>CommandTOL</code>	降落
SRV	<code>/mavros/param/set</code>	<code>ParamSet</code>	设置参数
TX	<code>/rflsim/win0/ue4_cmd</code>	<code>UE4Cmd</code>	UE4 视口控制命令

UE4 跟随视角

脚本自动配置 UE4 跟随视角，使用以下 UE4 命令：

命令	说明
<code>RflyChangeViewKeyCmd N 1</code>	切换到地球静止跟随模式（第 1 视角）
<code>RflyCameraPosAng -4 0 -2 0 -15 0</code>	相机位于飞机后方 4m、上方 2m，俯角 15°
<code>RflyCameraFovDegrees 90</code>	设置视场角为 90°
<code>RflyChangeViewKeyCmd S</code>	显示无人机编号标签
<code>RflyChangeViewKeyCmd T 8</code>	显示飞行轨迹

随机航点生成

航点在以起飞点为原点的 ENU 坐标系立方体空间内均匀随机生成：

1	<code>X</code> ∈ [-range, +range] （东西方向，默认 ±5m）
2	<code>Y</code> ∈ [-range, +range] （南北方向，默认 ±5m）
3	<code>Z</code> ∈ [alt_min, alt_max] （高度，默认 3~8m）

每到达一个航点后，无人机会随机悬停 0.5~2.0 秒后再飞向下一个航点，实现自然的随机飞行效果。

6. 参考资料

1. [RflySim 官方文档](#)
2. [RflySim 视觉仿真指南](#)
3. [ROS sensor_msgs/Image 消息定义](#)
4. [ROS sensor_msgs/PointCloud2 消息定义](#)

7. 常见问题

Q1：Rviz 中没有显示任何图像或点云数据

A1：请检查以下几点：

- 确认 CopterSim 和 RflySim3D 均已启动
- 确认 RosTrans 已启动并加载了正确的 `rostrans.param`（`enable_vision=true`）
- 确认 WSL2 与 Windows 网络互通（可运行 `WinWSL.bat` 配置网络桥接）

- 在 Rviz 中检查主题名称是否正确（如 `/rflsim/sensor1/img_rgb`）

I Q2: LiDAR 点云显示异常（全黑或无数据点）

A2: 请确认:

- `Config.json` 中 LiDAR 传感器的 `DataCheckFreq` 不宜设置过高（建议 10Hz）
- `otherParams` 中的探测距离和角度范围是否合理
- Rviz 中 PointCloud2 的 `Style` 建议设为 `Flat Squares`，`Size` 设为 3 像素

I Q3: RunWslRviz.bat 无法启动 Rviz 图形界面

A3: 请确认:

- VcXsrv (X11 转发服务器) 已安装在 `%PSP_PATH%\VcXsrv` 目录下
- WSL 发行版名称为 `RflySim-20.04`（如不同需修改 bat 脚本中的 `-d` 参数）
- 在 WSL 中已正确配置 `DISPLAY` 环境变量

I Q4: vision_copter_flight.py 启动后无人机不动或报错

A4: 请检查以下几点:

- 确认 CopterSim 已启动且处于运行状态 (SimMode=3, DLL Model 模式)
- 确认 RosTrans 已启动并成功连接 CopterSim (DllModelAssistant 插件已加载)
- 若提示 `rflsim_msgs` 导入失败, UE4 视口控制功能将不可用, 但飞行控制不受影响
- 确认 ROS 环境变量已正确设置 (`ROS_VERSION`、`ROS_MASTER_URI` 等)
- 可通过 `rostopic list` (ROS1) 或 `ros2 topic list` (ROS2) 检查 `/mavros/state` 等主题是否存在

I Q5: UE4 跟随视角没有生效

A5: 请确认:

- `rflsim_msgs` 包已正确安装, 脚本启动时未出现 `rflsim_msgs 未安装` 的提示
- RflySim3D 窗口已打开且正常显示场景
- 若手动修改了视角, 可重新运行脚本或在 RflySim3D 中按 `N` 键切换回跟随模式

-
1. <https://rflsim.com/> ↩
 2. 推荐配置请见: <https://rflsim.com/doc/zh/HowToInstall.pdf> ↩