

前言

本文主要内容为V1Pro相机定位算法模块使用示例说明，本示例基于ROS2平台，经测试在Ubuntu22.04(humble)Linux系统上编译运行，通过调用相机SDK实现建图（采图）、定位等功能。在查阅本文档前请先阅读相机SDK使用说明《LxCameraApi（C、C++）开发者指南.pdf》，同时，若运行于Linux平台请按照提示运行 `install.sh` 脚本，完成环境配置。

定位算法模块

异常代码

```
1 // status
2 // 算法返回值：异常码
3 Unknown      = -1,  //!< 重定位中
4 Tracking     = 0,  //!< 正常定位中
5 RelocFailed  = 1,  //!< 重定位失败
6 Slipping     = 2,  //!< 打滑
7 Relocating   = 6,  //!< 重定位中
8 NoTopImage   = 7,  //!< 相机图像输入异常
9 NoOdom       = 8,  //!< 里程计超时异常
10 Mapping     = 20, //!< 正常建图中，建图需要odom,图像以及激光数据（激光数据or激光位姿至少满足一个，如无可发虚拟数据）
11 MappingFailed = 21, //!< 建图异常
12 NoLaser     = 22, //!< 激光超时异常，建图时使用
13 NoLaserPose = 23, //!< 激光位姿数据超时异常，建图时使用
14
15 // 附加返回值：重定位异常码
16 -1: Unknown      //!< 重定位中（若重定位位姿(x,y,yaw)全为0，会判断为误发，返回代码-1）
17 0: Normal        //!< 重定位成功
18 1: No image      //!< 重定位失败，图像输入异常
19 2: No detection  //!< 重定位失败，特征检测异常
20 3: No expectation //!< 重定位失败，期望地图异常
21 4: No both       //!< 重定位失败，特征检测、期望地图都异常
22 5: No match      //!< 重定位失败，检测与地图无匹配
23 6: Off>30cm     //!< 重定位失败，检测与地图存在匹配，距离超过阈值
24 7: invalid request //!< 重定位失败，重定位请求位姿无效，如NaN
```

定位置信度

在正常定位的状态（异常码为0时），可通过置信度衡量定位结果的质量，定位置信度分布于0~4，其中：

- 0：不好
- 1：较差
- 2-3：一般
- 4：较好

在正常定位的状态时，建议取置信度大于等于1时的定位结果。

建图

- 输入数据
 - 机器人里程计（绝对）的数据，类型为nav_msgs/Odometry
 - 机器人激光雷达的数据（辅助建图所需，部分场景可不用激光，可以不输入，请联系技术支持确认），类型为sensor_msgs/LaserScan
 - 机器人激光位姿的数据（若需视觉地图和激光地图坐标系统一则需传入，否则可不输入），类型为geometry_msgs/PoseStamped
- 输出数据
 - 导出相机中录制的原始数据，离线建图，然后导入视觉地图至相机

定位

- 输入数据
 - 机器人里程计（绝对）数据，类型为nav_msgs::msg::Odometry
- 输出数据
 - 发送重定位请求，机器人重定位成功后，输出机器人在视觉地图中的位姿，类型为geometry_msgs::msg::PoseStamped

编译

ROS节点创建步骤：

1. 进入示例程序代码文件目录 `Lanxin-MRDVS/Sample/ROS/lx_camera_node_ws`
2. `colcon build`
3. `source install/setup.bash`

运行

运行ROS示例前，若相机中已经部署ROS主从，需先关闭ROS主从相关设置，否则会因ROS产生相关通讯异常。若相机中有ROS2(humble)环境，需上电后等待约20s左右，再运行ROS示例程序。

通讯

可以使用虚拟数据调通SDK通讯，确认与相机之间通讯正常。

1. `source install/setup.bash`
2. `ros2 launch lx_camera_ros sensor_sim.launch.py`

如下图所示，该节点发出虚拟的激光数据 `/sim/scan`、里程计数据 `/sim/odom` 和激光位姿数据 `/sim/scan_pose`，相机通过SDK去接收这些数据，联调通讯。

```
1 from launch import LaunchDescription
2 from launch_ros.actions import Node
3
4 def generate_launch_description():
5     return LaunchDescription([
6         Node(
7             package="lx_camera_ros",
8             executable="sensor_sim_node",
9             output="screen",
10            emulate_tty=True,
11            parameters=[
12                #<!-- -135° -->
13                {"min_ang": -2.35619449},
```

```

14     #<!-- 135° -->
15     {"max_ang": 2.35619449},
16     {"angle_increment": 0.00582},
17     {"time_increment": 0.00006167129},
18     {"range_min": 0.05},
19     {"range_max": 101.0},
20     {"init_range": 100.0},
21
22     #<!-- 虚拟激光数据topic -->
23     {"laser_frameId": "laser_link"},
24     {"laser_topic_name": "/sim/scan"},
25     #<!-- 虚拟里程计数据topic -->
26     {"odom_frame_id": "base"},
27     {"odom_topic_name": "/sim/odom"},
28     #<!-- 虚拟激光位姿topic -->
29     {"laserpose_frame_id": "base"},
30     {"laserpose_topic_name": "/sim/scan_pose"}}])
31

```

3. ros2 launch lx_camera_ros mapping.launch.py

如下图所示，该节点为接收虚拟传感器数据，然后通过调用相机SDK传入数据。

```

1  from launch import LaunchDescription
2  from launch_ros.actions import Node
3
4  def generate_launch_description():
5      return LaunchDescription([
6          Node(
7              package="lx_camera_ros",
8              executable="lx_localization_node",
9              output="screen",
10             emulate_tty=True,
11             parameters=[
12                 #<!-- 相机IP，默认为空时按设备索引获取 -->
13                 {"ip": "192.168.100.82"},
14                 #<!-- 图像显示使能 -->
15                 {"is_show": False},
16                 #<!-- 接收或发布ROS话题名 -->
17                 {"LxCamera_UploadScan": "/sim/scan"},
18                 {"LxCamera_UploadOdom": "/sim/odom"},
19                 {"LxCamera_UploadLaserPose": "/sim/scan_pose"},
20                 {"LxCamera_Error": "LxCamera_Error"},
21                 {"LxCamera_Command": "LxCamera_Command"},
22                 {"LxCamera_Mapping": "LxCamera_Mapping"},
23                 {"LxCamera_Location": "LxCamera_Location"},
24                 {"LxCamera_SetParam": "LxCamera_SetParam"},
25                 {"LxCamera_SwitchMap": "LxCamera_SwitchMap"},
26                 {"LxCamera_DownloadMap": "LxCamera_DownloadMap"},
27                 {"LxCamera_UploadMap": "LxCamera_UploadMap"},
28                 {"LxCamera_LocationResult": "LxCamera_LocationResult"},
29                 {"LxCamera_UploadReloc": "LxCamera_UploadReloc"},
30                 #<!-- 自动曝光期望值，范围[0-100]，默认:50 -->
31                 {"auto_exposure_value": 50},
32                 #<!-- 建图使能，true or false -->
33                 {"mapping_mode": True},
34                 #<!-- 定位使能，true or false -->

```

```

35         {"localization_mode": False},
36         #<!-- 相机至小车外参, 单位:米, 度数, 格式[x, y, z, yaw, pitch, roll]
-->
37         {"camera_extrinsic_param": [0.34, 0.00, 1.3, -90, 0, 0]},
38         #<!-- 激光雷达至小车外参, 单位:米, 度数, 格式[x, y, yaw] -->
39         {"laser_extrinsic_param": [0.34, 0.11, 0.0]} ])
40     ])

```

4. 待算法返回异常码为20（Mapping）时，通讯即正常。

建图

1. 准备好建图所需的传感器数据

- 机器人里程计（绝对）的topic，类型为nav_msgs::msg::Odometry
- 机器人激光雷达的topic（部分场景可不用激光，请联系技术支持确认，此时可输入虚拟激光数据），类型为sensor_msgs::msg::LaserScan
- 机器人激光位姿的topic（若需视觉地图和激光地图坐标系统一则需传入），类型为geometry_msgs::msg::PoseStamped

2. **按照要求修改launch文件夹 mapping.launch 下文件（切记请核对）**，修改实际传入数据的topic名，如激光数据 /scan、里程计数据 /odom 和激光位姿数据 /scan_pose。设置已存在地图名（若初次使用或不知道已存在地图名，可设置为默认 "example_map1"）、相机外参、激光雷达外参等

```

1  from launch import LaunchDescription
2  from launch_ros.actions import Node
3
4  def generate_launch_description():
5      return LaunchDescription([
6          Node(
7              package="lx_camera_ros",
8              executable="lx_localization_node",
9              output="screen",
10             emulate_tty=True,
11             parameters=[
12                 #<!-- 相机IP, 默认为空时按设备索引获取 -->
13                 {"ip": "192.168.100.82"},
14                 #<!-- 图像显示使能 -->
15                 {"is_show": False},
16                 #<!-- 接收或发布ROS话题名 -->
17                 {"LxCamera_UploadScan": "/scan"},
18                 {"LxCamera_UploadOdom": "/odom"},
19                 {"LxCamera_UploadLaserPose": "/scan_pose"},
20                 {"LxCamera_Error": "LxCamera_Error"},
21                 {"LxCamera_Command": "LxCamera_Command"},
22                 {"LxCamera_Mapping": "LxCamera_Mapping"},
23                 {"LxCamera_Location": "LxCamera_Location"},
24                 {"LxCamera_SetParam": "LxCamera_SetParam"},
25                 {"LxCamera_SwitchMap": "LxCamera_SwitchMap"},
26                 {"LxCamera_DownloadMap": "LxCamera_DownloadMap"},
27                 {"LxCamera_UploadMap": "LxCamera_UploadMap"},
28                 {"LxCamera_LocationResult": "LxCamera_LocationResult"},
29                 {"LxCamera_UploadReloc": "LxCamera_UploadReloc"},
30                 #<!-- 自动曝光期望值, 范围[0-100], 默认:50 -->
31                 {"auto_exposure_value": 50},
32                 #<!-- 建图使能, true or false -->
33                 {"mapping_mode": True},

```

```

34     #<!-- 定位使能, true or false -->
35     {"localization_mode": False},
36     #<!-- 相机至小车外参, 单位:米, 度数, 格式[x, y, z, yaw, pitch, roll]
-->
37     {"camera_extrinsic_param": [0.34, 0.00, 1.3, -90, 0, 0]},
38     #<!-- 激光雷达至小车外参, 单位:米, 度数, 格式[x, y, yaw] -->
39     {"laser_extrinsic_param": [0.34, 0.11, 0.0]} ])
40 })

```

3. source install/setup.bash

4. ros2 launch lx_camera_ros mapping.launch

5. 待算法返回异常码为20（Mapping）时，可以推动机器人录制数据

6. 导出录制数据。通过SDK接口实现该功能，**需先关闭建图使能**，然后导出。或者通过示例的ros2 topic pub的形式导出数据，示例如下：

先关闭建图使能：

```

1 ros2 topic pub -1 /lx_localization_node/LxCamera_Mapping
std_msgs/msg/String "{data: '0'}"

```

然后，请根据实际情况修改路径，示例：

```

1 ros2 topic pub -1 /lx_localization_node/LxCamera_DownloadMap
std_msgs/String "{data: 'your_dir/Lanxin-
MRDVS/Sample/ROS2/lx_camera_node_ws/src/lx_camera_ros/map/download_map'}"

```

同时通过 /lx_localization_node/LxCamera_Message 话题查看下发情况：

```

1 ros2 topic echo /lx_localization_node/LxCamera_Message | grep DownloadMap

```

若返回结果如下，且在则导出相应路径中可以看到导出的地图，例如

在 /home/fr1511b/v1pro/Lanxin-MRDVS/Sample/ros-v1pro/map 路径下生成

download_map.zip 文件，则导出数据成功。**然后请联系MRDVS相关人员技术支持，并提供建图数据。**

```

1 data: "{\"cmd\":\"LxCamera_DownloadMap\",\"result\":0}

```

7. 建图完成后，**需先关闭建图使能**，然后将地图导入至相机。通过SDK接口实现该功能，或者通过示例的ros2 topic pub的形式导入数据，示例如下：

```

1 ros2 topic pub -1 /lx_localization_node/LxCamera_UploadMap
std_msgs/String "{data: '/home/fr1511b/v1pro/Lanxin-MRDVS/Sample/ros-
v1pro/map/xz9_231216.zip'}"

```

同时通过 /lx_localization_node/LxCamera_Message 话题查看下发情况：

```

1 ros2 topic echo /lx_localization_node/LxCamera_Message | grep UploadMap

```

若返回结果如下，则导入地图成功。

```

1 data: "{\"cmd\":\"LxCamera_UploadMap\",\"result\":0}

```

定位

1. 准备好建图所需的传感器数据

- o 机器人里程计的topic，类型为nav_msgs::msg::Odometry

2. 按照要求修改launch文件夹下 localization.launch (切记请核对)，设置地图名 (已有地图名，如刚才导入的地图)、相机内参等

```
1  from launch import LaunchDescription
2  from launch_ros.actions import Node
3
4  def generate_launch_description():
5      return LaunchDescription([
6          Node(
7              package="lx_camera_ros",
8              executable="lx_localization_node",
9              output="screen",
10             emulate_tty=True,
11             parameters=[
12                 #<!-- 相机IP, 默认为空时按设备索引获取 -->
13                 {"ip": "192.168.100.82"},
14                 #<!-- 图像显示使能 -->
15                 {"is_show": False},
16                 #<!-- 接收或发布ROS话题名 -->
17                 {"LxCamera_UploadScan": "/sim/scan"},
18                 {"LxCamera_UploadOdom": "/sim/odom"},
19                 {"LxCamera_UploadLaserPose": "/sim/scan_pose"},
20                 {"LxCamera_Error": "LxCamera_Error"},
21                 {"LxCamera_Command": "LxCamera_Command"},
22                 {"LxCamera_Mapping": "LxCamera_Mapping"},
23                 {"LxCamera_Location": "LxCamera_Location"},
24                 {"LxCamera_SetParam": "LxCamera_SetParam"},
25                 {"LxCamera_SwitchMap": "LxCamera_SwitchMap"},
26                 {"LxCamera_DownloadMap": "LxCamera_DownloadMap"},
27                 {"LxCamera_UploadMap": "LxCamera_UploadMap"},
28                 {"LxCamera_LocationResult": "LxCamera_LocationResult"},
29                 {"LxCamera_UploadReloc": "LxCamera_UploadReloc"},
30                 #<!-- 自动曝光期望值, 范围[0-100], 默认:50 -->
31                 {"auto_exposure_value": 50},
32                 #<!-- 建图使能, true or false -->
33                 {"mapping_mode": False},
34                 #<!-- 定位使能, true or false -->
35                 {"localization_mode": True},
36                 #<!-- 重要: 定位时地图名需有效, 已上传相机并存在; 建图时,如果相机中无地图可
默认输入"example_map1" -->
37                 {"map_name": "xz9_231216"},
38                 #<!-- 相机至小车外参, 单位:米, 度数, 格式[x, y, z, yaw, pitch, roll]
-->
39                 {"camera_extrinsic_param": [0.34, 0.00, 1.3, -90, 0, 0]},
40                 #<!-- 激光雷达至小车外参, 单位:米, 度数, 格式[x, y, yaw] -->
41                 {"laser_extrinsic_param": [0.34, 0.11, 0.0]} ])
42
43
```

3. source install/setup.bash

4. 发送重定位请求。将机器人移动至指定地点，发送该点位在视觉地图中的坐标即可（一般地，录图起始点坐标为(x=0, y=0, yaw=0)）（注意：直接发重定位请求位姿（x=0, y=0, yaw=0）会被当成误发，可以规避坐标零点，如重定位请求位姿（x=0.01, y=0.00, yaw=0）），此部分可以通过SDK接口实现，或在示例的ros2 topic pub的形式发布重定位请求：

```
1  ros2 topic pub -1 /lx_localization_node/LxCamera_UploadReloc
   geometry_msgs/msg/PoseWithCovarianceStamped "{
2    header: {
3      stamp: { sec: 0, nanosec: 0 },
4      frame_id: 'map'
5    },
6    pose: {
7      pose: {
8        position: { x: 0.01, y: 0.00, z: 0.0 },
9        orientation: { x: 0.0, y: 0.0, z: 0.0, w: 1.0 }
10     },
11     covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
12                  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
13                  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
14   }
15 }"
```

同时通过 /lx_localization_node/LxCamera_Message 话题查看下发情况：

```
1  ros2 topic echo /lx_localization_node/LxCamera_Message | grep UploadReloc
```

若返回结果如下，则导入地图成功。

```
1  data: "{\"cmd\":\"LxCamera_UploadReloc\",\"result\":0}"
```

5. 待算法返回异常码为0（Tracking）时，并且可以订阅到 /lx_localization_node/LxCamera_LocationResult 即定位OK。此时，相机返回的为机器人在视觉地图中的位姿。

常见问题

运行该示例出现问题时，可通过相机SDK日志快速排查，常见问题如下：

1. 若通过ROS示例会出现卡顿、数据通讯异常，考虑ROS主从问题，去掉相关设置即可
2. 若关闭ROS示例程序后重新运行失败，考虑相机流占用问题，请耐心等待10-20s，再重新打开
3. 若出现SDK库加载失败、网络频繁丢包等问题，请确认配置脚本 `install.sh` 是否运行生效
4. 重定位成功时程序崩溃，考虑ROS相机内部配置问题，请联系相关人员解决

参考

- LxCameraApi（C、C++）开发者指南.pdf
- LxCameraViewer使用说明书.pdf
- V1Pro用户使用手册.pdf