RIEGL-VZ ROS 2 Driver API

1. Coordinate Systems

RIEGL uses hierarchically structured coordinate systems:

SOCS (Scanner's Own Coordinate System): Angle data and range data are the base for calculation of the data in the Scanner's Own Coordinate System (SOCS).

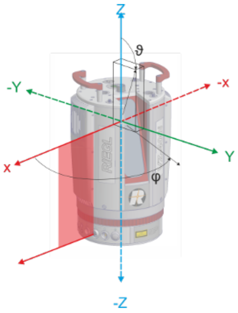

Figure 1: SOCS

PRCS (Project Coordinate System): A number of scan positions and the data acquired therein make up a scan project.The center of the project's coordinate system (PRCS) usually coincides horizontally with the center of the first scan position. The axes of PRCS are strictly pointing to east (x-axis, red), north (y-axis, green) and up (z-axis, blue), respectively.

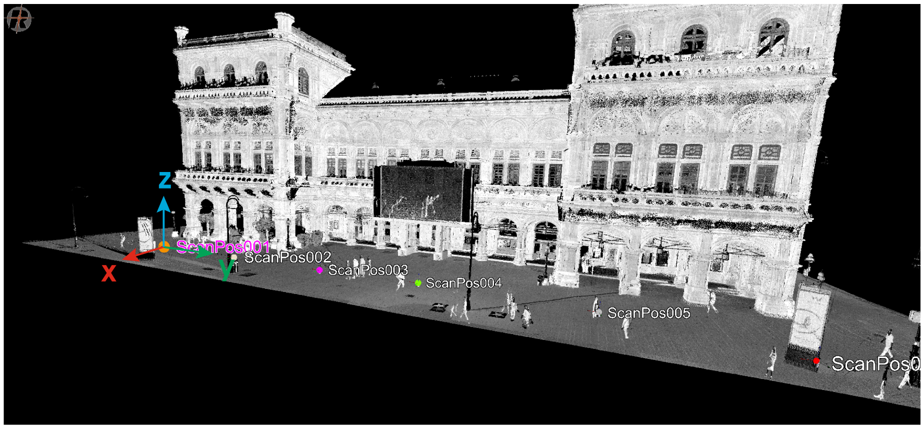

Figure 2: PRCS

VOCS (Voxel Coordinate System): This is an intermediate coordinate system.

Origin and orientation are identical to PRCS at the first scan position. Each scan will be registered in the VOCS coordinate system.

- Every scan position is described by SOPV (Scan Orientation and Position in VOCS).
- The position of the VOCS is described by VOP (VOCS Orientation and Position in PRCS).
- With every scan the VOP, especially the orientation, will be readjusted.
- The SOP (Scan Position and Orientation in PRCS) has to be recalculated after each newly registered scan from SOPV and updated VOP.

```
SOCS ---
 |      |
sopv    |
 |      |
VOCS   sop
 |      |
vop     |
 |      |
PRCS ---
```

2. RIEGL Interfaces

2.1 Messages

riegl_vz_interfaces/Status:

tbd…

```
uint8 warnings
uint8 errors
uint8 scan_progress
uint8 memory_usage
```

2.2 Services

riegl_vz_interfaces/GetPointCloud:

```
# Two-integer timestamp that is expressed as seconds and nanoseconds.
builtin_interfaces/Time stamp
---
bool success   # indicate successful run of service
string message # informational, e.g. for error messages
PointCloud2 pointcloud
```

A zero timestamp {seconds=0,nanoseconds=0} implicitly refers to the last scan position.
See PointCoud2 definition: sensor_msgs/PointCloud2

riegl_vz_interfaces/GetPose:

```
---
bool success   # indicate successful run of service
string message # informational, e.g. for error messages
PoseStamped poses[]
```

See PoseStamped definition: sensor_msgs/PoseStamped
The ‘frame_id’ in the header is either ‘SOCS’ or ‘VOCS’.

riegl_vz_interfaces/SetPose:

```
PoseStamped poses[]
---
bool success   # indicate successful run of service
string message # informational, e.g. for error messages
```

3. Nodes

3.1 riegl_vz

3.1.1 Parameters

~**hostname** (string, default: “”) :

The scanners hostname or IP address.

~**working_dir** (string, default: “~/.ros_riegl_vz”) :

The root working directory for runtime execution.

~**ssh_user** (string, default: “user”) :

The linux user name for SSH login on the scanner.

~**ssh_password** (string, default: “user”) :

The linux user password for SSH login on the scanner.

~**project_name** (string, default: “”) :

The scan project name used by service ‘set_project’. An existing project will be loaded, otherwise a new project will be created. If string is empty, a default project name will be composed from current local time (date and time).

~**scan_pattern** (double[], default: {30.0,130.0,0.04,0.0,360.0,0.04})

Specifies the field of view (FOV) for scanning and the scan increments.
[0]: Line Start Angle

[1]: Line Stop Angle
[2]: Line Angle Increment
[3]: Frame Start Angle
[4]: Frame Stop Angle
[5]: Frame Angle Increment

~**meas_program** (integer, default: 0) :

This is the laser scanner measurement program, which specifies the laser scanner frequency.

~**scan_publish** (bool, default: “True”) :

Enable publishing of point cloud data on topic ‘pointcloud’ after scan acquisition has finished.

~**msm** (integer[], default: 1) :

The scan data MSM (monitor step multiplier), used for point cloud data reduction, default disabled.

3.1.2 Published Topics

pointcloud (sensor_msgs/PointCloud2) :

Point cloud with scan data from the laser scanner in SOCS.

status (riegl_vz_interfaces/Status) :

Riegl VZ scanner status, provided once per second.

3.1.3 Services

set_project (std_srvs/Trigger) :

Create a new or load an existing project on the scanner with name from parameter ‘~project_name’.

Response:
success = True -> message: Project Name
success = False -> message: Error Message

scan (std_srvs/Trigger) :

Acquire laser scan data. When the scan has finished data is published on ‘pointcloud’ topic if parameter ‘~pointcloud_publish’ is enabled. Use ‘is_busy’ service to check if data acquisition has finished.

Response:
success = True -> message: Measurement Identifier
success = False -> message: Error Message

register_scan (std_srvs/SetBool) :

Start laser scan registration within actual project. Use ‘is_busy’ service to check if scan registration has finished.

Request:
data: enable coarse positioning (not supported atm)
Response:
success = True -> message: success
success = False -> message: Error Message

is_busy (std_srvs/SetBool) :

Check if scan data acquisition or registration is busy.

Request:
data: set blocking execution
Response:
success = True -> message: RIEGL VZ is busy
success = False -> message: RIEGL VZ is ready

get_pointcloud (riegl_vz_interfaces/GetPointCloud) :

Get point cloud of a previous scan data acquisition.

get_pose (riegl_vz_interfaces/GetPose) :

Request position {VOP, SOPV} of the previously acquired scan.

get_all_poses (riegl_vz_interfaces/GetPose) :

Request positions {VOP, SOPV[]} for all previously acquired scans in actual project.

stop (std_srvs/Trigger) :

Stop laser scan acquisition or registration.

Response:
success = True -> message: “RIEGL VZ has been stopped”

shutdown (std_srvs/Trigger) :

Stop data acquisition and power down the laser scanner.

Response:
success = True -> message: “RIEGL VZ is shutting down”

3.1.4 Services (Extension)

tbd…

set_pose (riegl_vz_interfaces/SetPose) :

Set position of the scanner origin in a reference coordinate system. This is useful for scan registration.

get_voxel (riegl_vz_interfaces/GetPointcloud) :

Get voxel data of a previous scan data acquisition.