RIEGL-VZi 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).

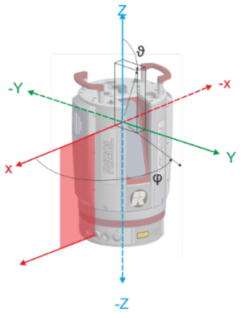

Figure 1: SOCS

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.

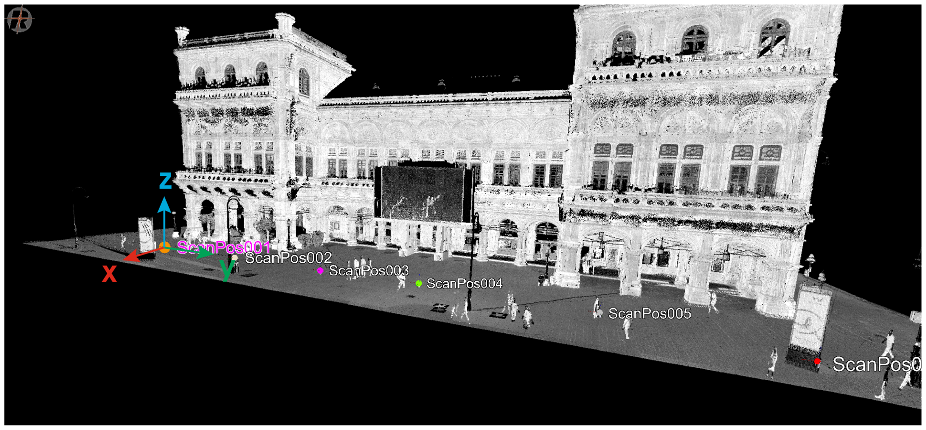

Figure 2: PRCS

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 Orienqtation 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__vzi__interfaces/Status:

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

tbd…

2.2 Services

riegl__vzi__interfaces/GetPointCloud:

```
uint32 index
---
PointCloud2 pointcloud
```

See PointCoud2 definition: sensor_msgs/PointCloud2

riegl__vzi__interfaces/GetPose:

```
int32 index_first
int32 index_last
---
PoseStamped poses[]
```

A negative value of an index points to the last scan position. 0 ist the first scan position.

See PoseStamped definition: sensor_msgs/PoseStamped[]

3. Nodes

3.1 riegl_vz

3.1.1 Parameters

~**hostname** (string, default: “”) :

The scanners hostname or IP address.

~**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.

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

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

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

The scan data MSM (monitor step multiplier), used for point cloud data reduction, default disabled ([0]: lines, [1]: shots).

3.1.2 Published Topics

pointcloud (sensor_msgs/PointCloud2) :

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

status (riegl_vzi_interfaces/Status) :

Riegl VZ scanner status, provided once per second.

3.1.3 Services

create_project (std_srvs/Trigger) :

Create a new or load an existing project on the scanner with name composed from current local time (date and time).

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

get_pointcloud (riegl_vzi_interfaces/GetPointCloud) :

Get point cloud data of a previously acquired scan.

register_scan (std_srvs/Trigger) :

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

Response:
success = True -> message: SUCCESS
success = False -> message: Error Message

is_busy (std_srvs/SetBool) :

Check if scanner data acquisition or registration is busy.

Request:
data: set blocking execution
Response:
success = True -> message: BUSY
success = False -> message: READY

get_pose (riegl_vzi_interfaces/GetPose) :

Request positions for a number of previously acquired scans in actual project. {VOP, SOPV[], SOP[]}

shutdown (std_srvs/Trigger) :

Power down the laser scanner.

Response:
success = True -> message: SUCCESS
success = False -> message: Error Message