
Cartographer ROS Documentation

Release 1.0.0

The Cartographer Authors

Apr 12, 2017

Contents

1	Configuration	1
2	ROS API	3
2.1	Command-line Flags	3
2.2	Subscribed Topics	3
2.3	Published Topics	4
2.4	Services	4
2.5	Required tf Transforms	4
2.6	Provided tf Transforms	4
3	Public Data	5
3.1	2D Cartographer Backpack – Deutsches Museum	5
3.2	3D Cartographer Backpack – Deutsches Museum	7
3.3	PR2 – Willow Garage	7
4	Frequently asked questions	11
4.1	The laser data in the 3D bags is much higher than the maximum reported 20 Hz rotation speed that the VLP-16 can do. Why?	11
5	System Requirements	13
6	Building & Installation	15
7	Running the demos	17

Configuration

Note that Cartographer’s ROS integration uses [tf2](#), thus all frame IDs are expected to contain only a frame name (lower-case with underscores) and no prefix or slashes. See [REP 105](#) for commonly used coordinate frames.

Note that topic names are given as *base* names (see [ROS Names](#)) in Cartographer’s ROS integration. This means it is up to the user of the Cartographer node to remap, or put them into a namespace.

The following are Cartographer’s ROS integration top-level options, all of which must be specified in the Lua configuration file:

map_frame The ROS frame ID to use for publishing submaps, the parent frame of poses, usually “map”.

tracking_frame The ROS frame ID of the frame that is tracked by the SLAM algorithm. If an IMU is used, it should be at its position, although it might be rotated. A common choice is “imu_link”.

published_frame The ROS frame ID to use as the child frame for publishing poses. For example “odom” if an “odom” frame is supplied by a different part of the system. In this case the pose of “odom” in the *map_frame* will be published. Otherwise, setting it to “base_link” is likely appropriate.

odom_frame Only used if *provide_odom_frame* is true. The frame between *published_frame* and *map_frame* to be used for publishing the (non-loop-closed) local SLAM result. Usually “odom”.

provide_odom_frame If enabled, the local, non-loop-closed, continuous pose will be published as the *odom_frame* in the *map_frame*.

use_odometry If enabled, subscribes to [nav_msgs/Odometry](#) on the topic “odom”. Odometry must be provided in this case, and the information will be included in SLAM.

use_laser_scan If enabled, the node subscribes to [sensor_msgs/LaserScan](#) on the “scan” topic. If 2D SLAM is used, either this or *use_multi_echo_laser_scan* must be enabled.

use_multi_echo_laser_scan If enabled, the node subscribes to [sensor_msgs/MultiEchoLaserScan](#) on the “echoes” topic. If 2D SLAM is used, either this or *use_laser_scan* must be enabled.

num_point_clouds Number of 3D lasers to subscribe to. Must be a positive value if and only if using 3D SLAM. Subscribes to [sensor_msgs/PointCloud2](#) on the “points2” topic for one laser, or topics “points2_1”, “points2_2”, etc for multiple lasers.

lookup_transform_timeout_sec Timeout in seconds to use for looking up transforms using [tf2](#).

submap_publish_period_sec Interval in seconds at which to publish the submap poses, e.g. 0.3 seconds.

pose_publish_period_sec Interval in seconds at which to publish poses, e.g. 5e-3 for a frequency of 200 Hz.

The following ROS API is provided by `cartographer_node`.

Command-line Flags

- configuration_directory** First directory in which configuration files are searched, second is always the Cartographer installation to allow including files from there.
- configuration_basename** Basename (i.e. not containing any directory prefix) of the configuration file (e.g. `backpack_3d.lua`).

Subscribed Topics

The following range data topics are mutually exclusive. At least one source of range data is required.

scan (`sensor_msgs/LaserScan`) Supported in 2D and 3D (e.g. using an axially rotating planar laser scanner). If `use_laser_scan` is enabled in the *Configuration*, this topic will be used as input for SLAM.

echoes (`sensor_msgs/MultiEchoLaserScan`) Supported in 2D and 3D (e.g. using an axially rotating planar laser scanner). If `use_multi_echo_laser_scan` is enabled in the *Configuration*, this topic will be used as input for SLAM. Only the first echo is used.

points2 (`sensor_msgs/PointCloud2`) Only supported in 3D. If `num_point_clouds` is set to 1 in the *Configuration*, this topic will be used as input for SLAM. If `num_point_clouds` is greater than 1, multiple numbered points2 topics (i.e. `points2_1`, `points2_2`, `points2_3`, ... up to and including `num_point_clouds`) will be used as inputs for SLAM.

The following additional sensor data topics may also be provided.

imu (`sensor_msgs/Imu`) Supported in 2D (optional) and 3D (required). This topic will be used as input for SLAM.

odom (`nav_msgs/Odometry`) Supported in 2D (optional) and 3D (optional). If `use_odometry` is enabled in the *Configuration*, this topic will be used as input for SLAM.

Published Topics

map (`nav_msgs/OccupancyGrid`) Only supported in 2D. If subscribed to, a background thread will continuously compute and publish the map. The time between updates will increase with the size of the map. For faster updates, use the submaps APIs.

scan_matched_points2 (`sensor_msgs/PointCloud2`) Point cloud as it was used for the purpose of scan-to-submap matching. This cloud may be both filtered and projected depending on the *Configuration*.

submap_list (`cartographer_ros_msgs/SubmapList`) List of all submaps, including the pose and latest version number of each submap, across all trajectories.

Services

submap_query (`cartographer_ros_msgs/SubmapQuery`) Fetches the requested submap.

finish_trajectory (`cartographer_ros_msgs/FinishTrajectory`) Finishes the current trajectory by flushing all queued sensor data, running a final optimization, and writing artifacts (e.g. the map) to disk. The *stem* argument is used as a prefix for the various files which are written. Files will usually end up in `~/ros` or `ROS_HOME` if it is set.

Required tf Transforms

Transforms from all incoming sensor data frames to the *configured tracking_frame* and *published_frame* must be available. Typically, these are published periodically by a *robot_state_publisher* or a *static_transform_publisher*.

Provided tf Transforms

The transformation between the *configured map_frame* and *published_frame* is always provided.

If *provide_odom_frame* is enabled in the *Configuration*, a continuous (i.e. unaffected by loop closure) transform between the *configured odom_frame* and *published_frame* will be provided.

2D Cartographer Backpack – Deutsches Museum

This data was collected using a 2D LIDAR backpack at the [Deutsches Museum](#). Each bag contains data from an IMU, data from a horizontal LIDAR intended for 2D SLAM, and data from an additional vertical (i.e. push broom) LIDAR.

License

Copyright 2016 The Cartographer Authors

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Data

ROS Bag	Known Issues
b0-2014-07-11-10-58-16.bag	
b0-2014-07-11-11-00-49.bag	
b0-2014-07-21-12-42-53.bag	
b0-2014-07-21-12-49-19.bag	1 gap in vertical laser data
b0-2014-07-21-12-55-35.bag	
b0-2014-07-21-13-11-35.bag	
b0-2014-08-14-13-23-01.bag	
b0-2014-08-14-13-36-48.bag	

Continued on next page

Table 3.1 – continued from previous page

ROS Bag	Known Issues
b0-2014-10-07-12-13-36.bag	
b0-2014-10-07-12-34-42.bag	
b0-2014-10-07-12-43-25.bag	
b0-2014-10-07-12-50-07.bag	
b1-2014-09-25-10-11-12.bag	
b1-2014-10-02-14-08-42.bag	
b1-2014-10-02-14-33-25.bag	
b1-2014-10-07-12-12-04.bag	
b1-2014-10-07-12-34-51.bag	
b2-2014-11-24-14-20-50.bag	
b2-2014-11-24-14-33-46.bag	
b2-2014-12-03-10-14-13.bag	
b2-2014-12-03-10-33-51.bag	
b2-2014-12-03-10-40-04.bag	
b2-2014-12-12-13-51-02.bag	
b2-2014-12-12-14-18-43.bag	
b2-2014-12-12-14-41-29.bag	
b2-2014-12-12-14-48-22.bag	
b2-2014-12-17-14-33-12.bag	
b2-2014-12-17-14-53-26.bag	
b2-2014-12-17-14-58-13.bag	
b2-2015-02-16-12-26-11.bag	
b2-2015-02-16-12-43-57.bag	
b2-2015-04-14-14-16-36.bag	
b2-2015-04-14-14-39-59.bag	
b2-2015-04-28-13-01-40.bag	
b2-2015-04-28-13-17-23.bag	
b2-2015-05-12-12-29-05.bag	2 gaps in laser data
b2-2015-05-12-12-46-34.bag	14 gaps in laser data
b2-2015-05-26-13-15-25.bag	
b2-2015-06-09-14-31-16.bag	
b2-2015-06-25-14-25-51.bag	
b2-2015-07-07-11-27-05.bag	
b2-2015-07-21-13-03-21.bag	
b2-2015-08-04-13-39-24.bag	
b2-2015-08-18-11-42-31.bag	
b2-2015-08-18-11-55-04.bag	
b2-2015-08-18-12-06-34.bag	
b2-2015-09-01-11-55-40.bag	
b2-2015-09-01-12-16-13.bag	
b2-2015-09-15-14-19-11.bag	
b2-2015-11-24-14-12-27.bag	
b2-2016-01-19-14-10-47.bag	
b2-2016-02-02-14-01-56.bag	1 gap in laser data
b2-2016-03-01-14-09-37.bag	
b2-2016-03-15-14-23-01.bag	
b2-2016-04-05-14-44-52.bag	
b2-2016-04-27-12-31-41.bag	

3D Cartographer Backpack – Deutsches Museum

This data was collected using a 3D LIDAR backpack at the [Deutsches Museum](#). Each bag contains data from an IMU and from two Velodyne VLP-16 LIDARs, one mounted horizontally (i.e. spin axis up) and one vertically (i.e. push broom).

License

Copyright 2016 The Cartographer Authors

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Data

ROS Bag	Duration	Size	Known Issues
b3-2015-12-10-12-41-07.bag	1466 s	7.3 GB	1 large gap in data
b3-2015-12-10-13-10-17.bag	718 s	5.5 GB	1 gap in data
b3-2015-12-10-13-31-28.bag	720 s	5.2 GB	2 large gaps in data
b3-2015-12-10-13-55-20.bag	429 s	3.3 GB	
b3-2015-12-14-15-13-53.bag	916 s	7.1 GB	
b3-2016-01-19-13-26-24.bag	1098 s	8.1 GB	
b3-2016-01-19-13-50-11.bag	318 s	2.5 GB	
b3-2016-02-02-13-32-01.bag	47 s	366 MB	
b3-2016-02-02-13-33-30.bag	1176 s	9.0 GB	
b3-2016-02-09-13-17-39.bag	529 s	4.0 GB	
b3-2016-02-09-13-31-50.bag	801 s	6.1 GB	
b3-2016-02-10-08-08-26.bag	3371 s	25 GB	
b3-2016-03-01-13-39-41.bag	382 s	2.9 GB	
b3-2016-03-01-15-42-37.bag	3483 s	17 GB	6 large gaps in data
b3-2016-03-01-16-42-00.bag	313 s	2.4 GB	
b3-2016-03-01-16-48-39.bag	375 s	2.8 GB	
b3-2016-03-02-10-09-32.bag	1150 s	6.6 GB	3 large gaps in data
b3-2016-04-05-13-54-42.bag	829 s	6.1 GB	
b3-2016-04-05-14-14-00.bag	1221 s	9.1 GB	
b3-2016-04-05-15-51-36.bag	30 s	231 MB	
b3-2016-04-05-15-52-20.bag	377 s	2.7 GB	
b3-2016-04-05-16-00-55.bag	940 s	6.9 GB	
b3-2016-04-27-12-56-11.bag	2905 s	21 GB	
b3-2016-05-10-12-56-33.bag	1767 s	13 GB	
b3-2016-06-07-12-42-49.bag	596 s	3.9 GB	3 gaps in horizontal laser data

PR2 – Willow Garage

This is the Willow Garage data set, described in:

- “An Object-Based Semantic World Model for Long-Term Change Detection and Semantic Querying.”, by Julian Mason and Bhaskara Marthi, IROS 2012.

More details about these data can be found in:

- “Unsupervised Discovery of Object Classes with a Mobile Robot”, by Julian Mason, Bhaskara Marthi, and Ronald Parr. ICRA 2014.
- “Object Discovery with a Mobile Robot” by Julian Mason. PhD Thesis, 2013.

License

Copyright (c) 2011, Willow Garage All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Data

ROS Bag	Known Issues
2011-08-03-16-16-43.bag	Missing base laser data
2011-08-03-20-03-22.bag	
2011-08-04-12-16-23.bag	
2011-08-04-14-27-40.bag	
2011-08-04-23-46-28.bag	
2011-08-05-09-27-53.bag	
2011-08-05-12-58-41.bag	
2011-08-05-23-19-43.bag	
2011-08-08-09-48-17.bag	
2011-08-08-14-26-55.bag	
2011-08-08-23-29-37.bag	
2011-08-09-08-49-52.bag	
2011-08-09-14-32-35.bag	
2011-08-09-22-31-30.bag	
2011-08-10-09-36-26.bag	

Continued on next page

Table 3.2 – continued from previous page

ROS Bag	Known Issues
2011-08-10-14-48-32.bag	
2011-08-11-01-31-15.bag	
2011-08-11-08-36-01.bag	
2011-08-11-14-27-41.bag	
2011-08-11-22-03-37.bag	
2011-08-12-09-06-48.bag	
2011-08-12-16-39-48.bag	
2011-08-12-22-46-34.bag	
2011-08-15-17-22-26.bag	
2011-08-15-21-26-26.bag	
2011-08-16-09-20-08.bag	
2011-08-16-18-40-52.bag	
2011-08-16-20-59-00.bag	
2011-08-17-15-51-51.bag	
2011-08-17-21-17-05.bag	
2011-08-18-20-33-16.bag	
2011-08-18-20-52-30.bag	
2011-08-19-10-12-20.bag	
2011-08-19-14-17-55.bag	
2011-08-19-21-35-17.bag	
2011-08-22-10-02-27.bag	
2011-08-22-14-53-33.bag	
2011-08-23-01-11-53.bag	
2011-08-23-09-21-17.bag	
2011-08-24-09-52-14.bag	
2011-08-24-15-01-39.bag	
2011-08-24-19-47-10.bag	
2011-08-25-09-31-05.bag	
2011-08-25-20-14-56.bag	
2011-08-25-20-38-39.bag	
2011-08-26-09-58-19.bag	
2011-08-29-15-48-07.bag	
2011-08-29-21-14-07.bag	
2011-08-30-08-55-28.bag	
2011-08-30-20-49-42.bag	
2011-08-30-21-17-56.bag	
2011-08-31-20-29-19.bag	
2011-08-31-20-44-19.bag	
2011-09-01-08-21-33.bag	
2011-09-02-09-20-25.bag	
2011-09-06-09-04-41.bag	
2011-09-06-13-20-36.bag	
2011-09-08-13-14-39.bag	
2011-09-09-13-22-57.bag	
2011-09-11-07-34-22.bag	
2011-09-11-09-43-46.bag	
2011-09-12-14-18-56.bag	
2011-09-12-14-47-01.bag	
2011-09-13-10-23-31.bag	

Continued on next page

Table 3.2 – continued from previous page

ROS Bag	Known Issues
2011-09-13-13-44-21.bag	
2011-09-14-10-19-20.bag	
2011-09-15-08-32-46.bag	

Frequently asked questions

The laser data in the 3D bags is much higher than the maximum reported 20 Hz rotation speed that the VLP-16 can do. Why?

The VLP-16 in the example bags is configured to rotate at 20 Hz. However, the frequency of UDP packets the VLP-16 sends is much higher and independent of the rotation frequency. The example bags contain a `sensor_msgs/PointCloud2` per UDP packet, not one per revolution.

In the corresponding [Cartographer configuration file](#) you see `TRAJECTORY_BUILDER_3D.scans_per_accumulation = 160` which means we accumulate 160 per-UDP-packet point clouds into one larger point cloud, which incorporates motion estimation by combining constant velocity and IMU measurements, for matching. Since there are two VLP-16s, 160 UDP packets is enough for roughly 2 revolutions, one per VLP-16.

[Cartographer](#) is a system that provides real-time simultaneous localization and mapping (SLAM) in 2D and 3D across multiple platforms and sensor configurations. This project provides Cartographer's ROS integration.

System Requirements

See Cartographer's [system requirements](#).

The following [ROS distributions](#) are currently supported:

- Indigo
- Kinetic

Building & Installation

We recommend using `wstool` and `rosdep`. For faster builds, we also recommend using `Ninja`.

```
# Install wstool and rosdep.
sudo apt-get update
sudo apt-get install -y python-wstool python-rosdep ninja-build

# Create a new workspace in 'catkin_ws'.
mkdir catkin_ws
cd catkin_ws
wstool init src

# Merge the cartographer_ros.rosinstall file and fetch code for dependencies.
wstool merge -t src https://raw.githubusercontent.com/googlecartographer/
↪cartographer_ros/master/cartographer_ros.rosinstall
wstool update -t src

# Install deb dependencies.
rosdep init
rosdep update
rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y

# Build and install.
catkin_make_isolated --install --use-ninja
source install_isolated/setup.bash
```


CHAPTER 7

Running the demos

Now that Cartographer and Cartographer's ROS integration are installed, download the example bags (e.g. 2D and 3D backpack collections of the [Deutsches Museum](#)) to a known location, in this case ~/Downloads, and use roslaunch to bring up the demo:

```
# Download the 2D backpack example bag.
wget -P ~/Downloads https://storage.googleapis.com/cartographer-public-data/
↳bags/backpack_2d/cartographer_paper_deutsches_museum.bag

# Launch the 2D backpack demo.
roslaunch cartographer_ros demo_backpack_2d.launch bag_filename:=${HOME}/
↳Downloads/cartographer_paper_deutsches_museum.bag

# Download the 3D backpack example bag.
wget -P ~/Downloads https://storage.googleapis.com/cartographer-public-data/
↳bags/backpack_3d/b3-2016-04-05-14-14-00.bag

# Launch the 3D backpack demo.
roslaunch cartographer_ros demo_backpack_3d.launch bag_filename:=${HOME}/
↳Downloads/b3-2016-04-05-14-14-00.bag

# Download the Revo LDS example bag.
wget -P ~/Downloads https://storage.googleapis.com/cartographer-public-data/
↳bags/revo_lds/cartographer_paper_revo_lds.bag

# Launch the Revo LDS demo.
roslaunch cartographer_ros demo_revo_lds.launch bag_filename:=${HOME}/
↳Downloads/cartographer_paper_revo_lds.bag

# Download the PR2 example bag.
wget -P ~/Downloads https://storage.googleapis.com/cartographer-public-data/
↳bags/pr2/2011-09-15-08-32-46.bag

# Launch the PR2 demo.
roslaunch cartographer_ros demo_pr2.launch bag_filename:=${HOME}/Downloads/
↳2011-09-15-08-32-46.bag
```

```
# Download the Taurob Tracker example bag.
wget -P ~/Downloads https://storage.googleapis.com/cartographer-public-data/
↳bags/taurob_tracker/taurob_tracker_simulation.bag

# Launch the Taurob Tracker demo.
roslaunch cartographer_ros demo_taubob_tracker.launch bag_filename:=${HOME}/
↳Downloads/taurob_tracker_simulation.bag
```

The launch files will bring up `roscore` and `rviz` automatically.