# TrimBot2020 Deliverable D7.4

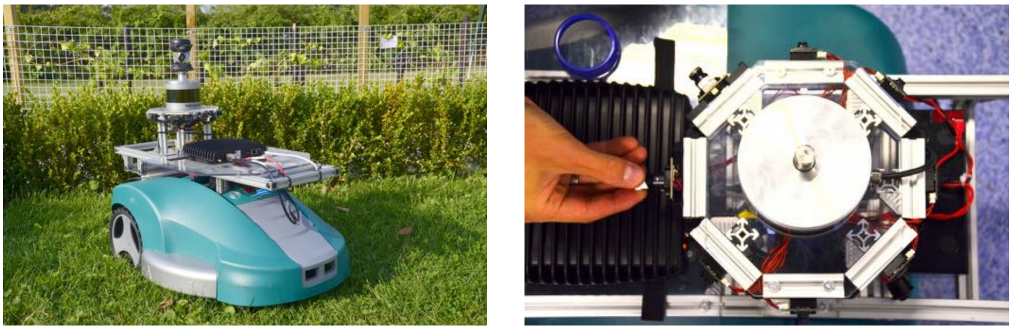
# Ground-truth data definitions and acquisition

Principal Author: UEDIN
Contributors: UEDIN
Dissemination: PU

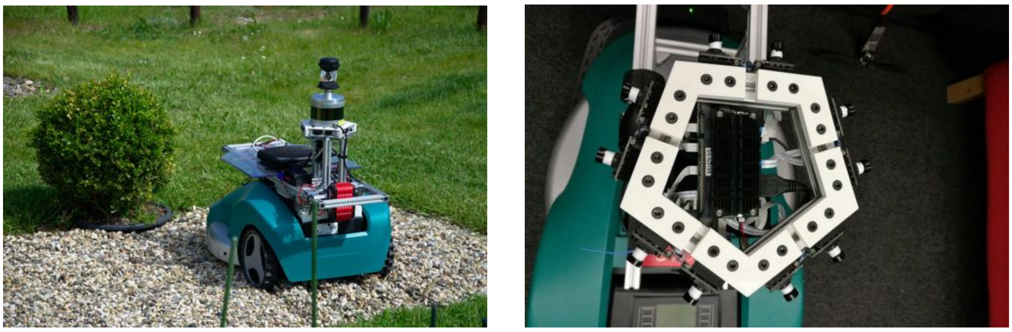
**Abstract:** This document specifies a set of standardized data for experimental evaluation needed by component development and evaluation described in D7.3. Content, data formats and acquisition details are given in general and specifically for each dataset captured in project test gardens.

Deliverable due: Month 12+



(a) Initial octagon camera setup (8 x 1).



(b) Improved pentagon camera setup (5 x 2).

Figure 1: First integrated platform for GT data collection.

# 1 Introduction

The captured datasets consist of sets of videos from the stereo sensors during different tasks: forward motion in the garden, approaching an obstacle, crossing different surface types, approaching a hedge, approaching a bush, trimming a bush, etc. The data were taken under different lighting conditions. The data were acquired by manual movement of the sensors, platform and arm along a known trajectory. This data was then marked up to identify important regions and locations, with attached 3D shape information.

# 2 Data acquisition

The prototype used for the capture was Platform 1 described in *D6.1 - First integrated platform* (Fig. 1). It is based on a modified *Bosch Indego* lawn mower with additional sensors. A camera rig from ETHZ and a *Velodyne VLP16* lidar scanner were mounted on the platform to capture data from the environment. A reflecting prism for the *Topcon Total Station* is mounted on top of the vehicle to track the position of the vehicle. A high-accuracy IMU is mounted to track the vehicle orientation. An embedded PC was also integrated into the vehicle to control it and record the sensor data.

The data for ground truth (GT) was captured in our two test gardens during several data collection meetings listed in Tab. 1. The first of them (WU1) is described in detail in *D1.1 - Sensor Data Collection including Ground Truth* and the next ones followed a similar program, except for BOSCH1 when a new camera setup tested and laser scanning and tracking was

| *Date* | *Location* | *Short name* | *Camera setup* | *Scan+track* |
|---|---|---|---|---|
| September 2016 | Wageningen, NL | WU1 | 8 x 1 mono+color | Yes |
| December 2016 | Renningen, DE | BOSCH1 | 5 x 2 mono | No |
| March 2017 | Renningen, DE | BOSCH2 | 5 x 2 mono+color | Yes |
| May 2017 | Wageningen, NL | WU2 | 5 x 2 mono+color | Yes |

Table 1: List of data collection events in chronological order.

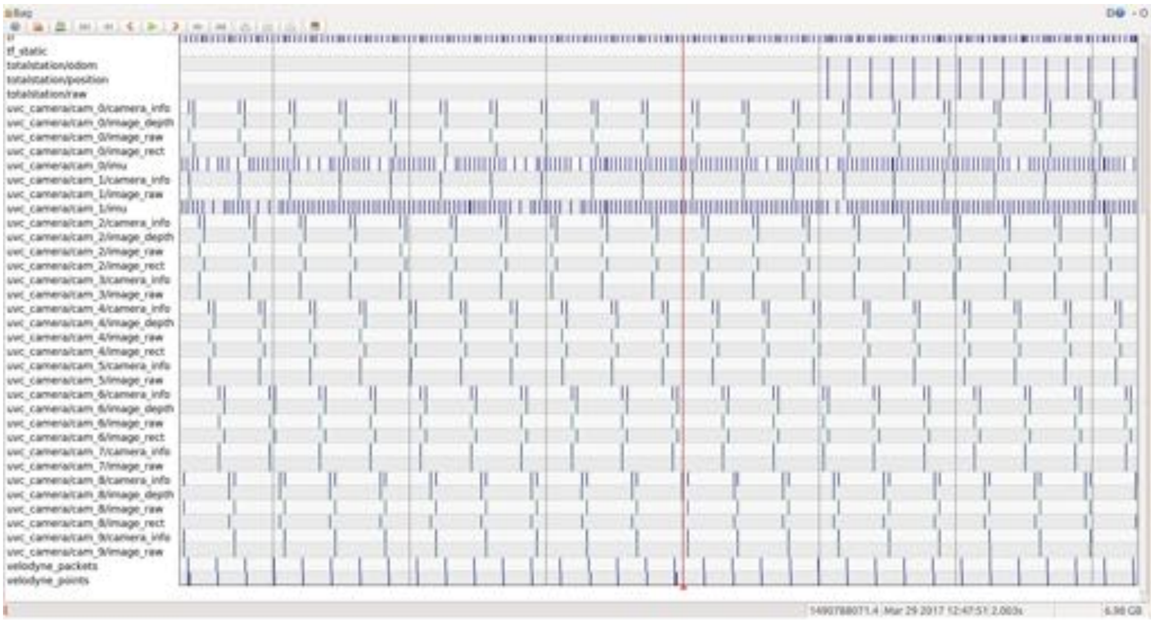


Figure 2: Visualization of a sample recorded bagfile index with `rqt_bag` tool in ROS. Left: topic list (vertical axis). Right: messages as short vertical line segments in time (horizontal axis).

omitted.

Apart from seasonal changes of environment (ie. growth and color of grass and bushes) the major difference between individual data collections was in the camera setup, which was improved from octagonal rig (Fig. 1a) to pentagonal (Fig. 1b) and the monochromatic sensors were also upgraded to color as shown in Tab. 1. The final setup provides complete coverage of the scene around the vehicle with depth and color information.

# 3 Data formats

The representation of data in the project is derived from the ROS protocols based on messages of different types. The data captured by sensors of Platform 1 are accordingly published as standard ROS messages[1] and recorded into ROS bagfiles[2]. A bagfile is essentially an archive of messages indexed by the source name (topic) and time of acquisition as shown in Fig. 2.

[1] http://wiki.ros.org/msg
[2] http://wiki.ros.org/Bags

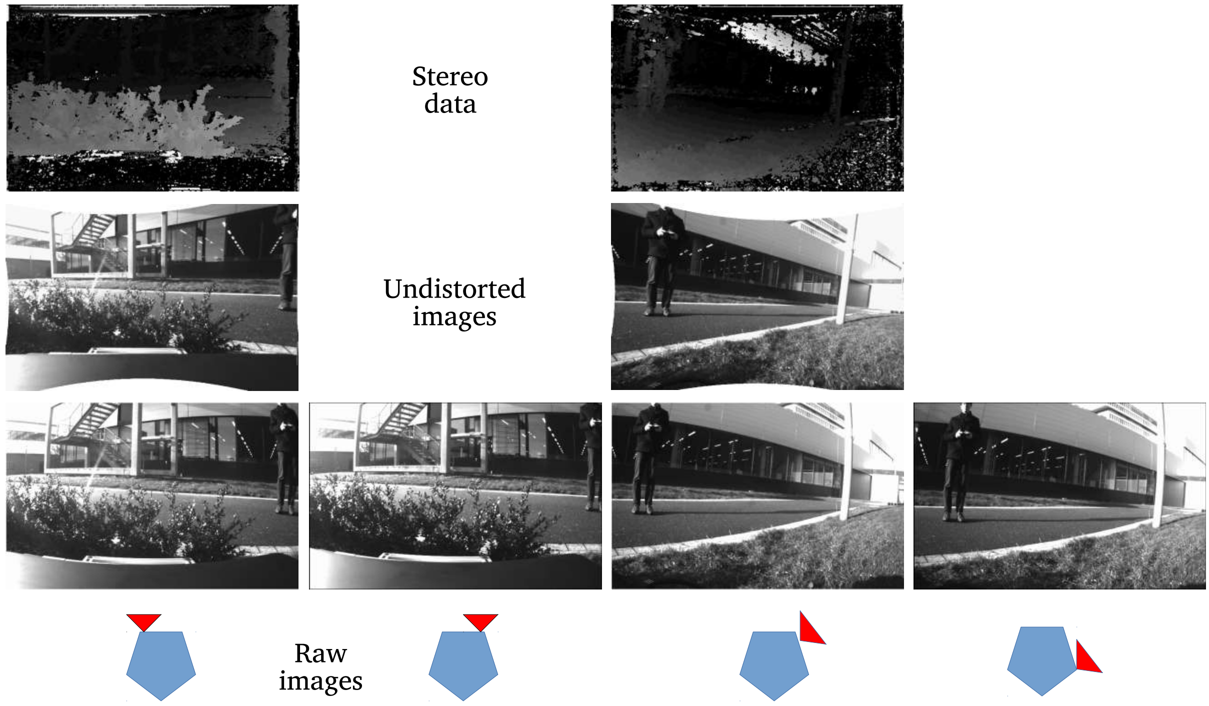



Figure 3: Sample images from two camera pairs of the pentagon (5x2) camera setup.

## 3.1 Camera data

The primary recorded data are image and depth streams from cameras (Fig. 3) with the following specification:

- Base topic: `/uvc_camera/cam_X`
  - `/image_raw` ... original captured image
  - `/image_rect` ... undistorted and rectified image
  - `/image_depth` ... depth from stereo matching (5x2 setup only)
- Data type: `sensor_msgs/Image`
- Resolution: 752x480 (WVGA)
- Encoding: Bayer RGGB or mono, 8 bits/channel
- Frame rate: 5 fps
- Synchronized: Yes, with time stamp in message header[3]

[3]Messages are recorded to bagfile sequentially with delay.

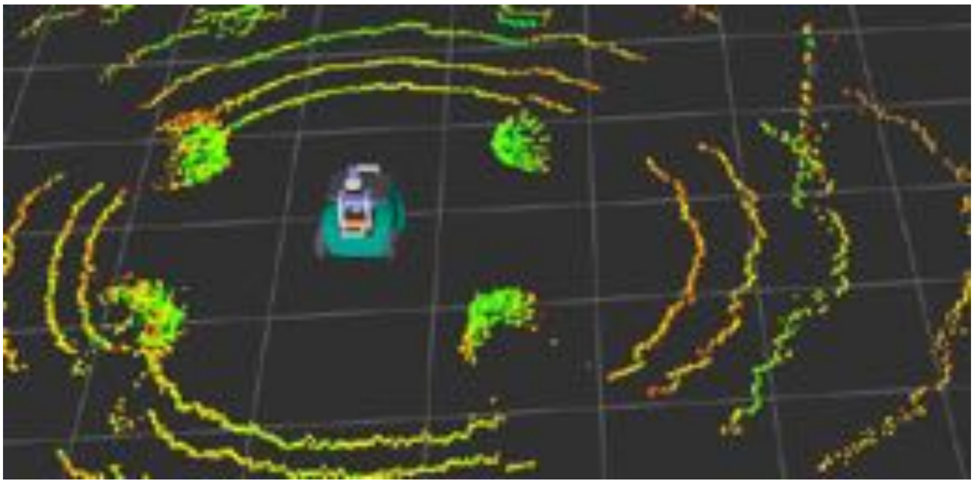


Figure 4: Sample range data from mounted laser scanner.

## 3.2 Position data

The position of the vehicle from the laser tracker and orientation calculated from IMU data is combined in the following pose message:

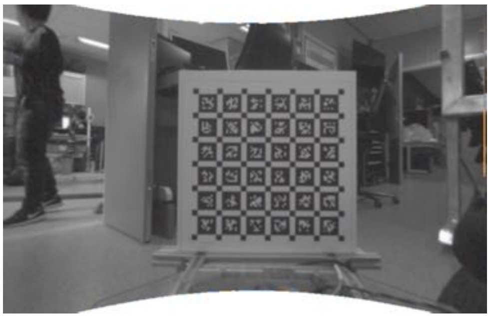
- Topic: `/ground_truth/odom`
- Data type: `nav_msgs/Odometry`
- Frame rate: 12 fps
- Accuracy of tracking: 1 cm pose
- Synchronized: No
- Coordinate system: center of mounted tracking prism relative to static point cloud from laser scanner

## 3.3 Range data

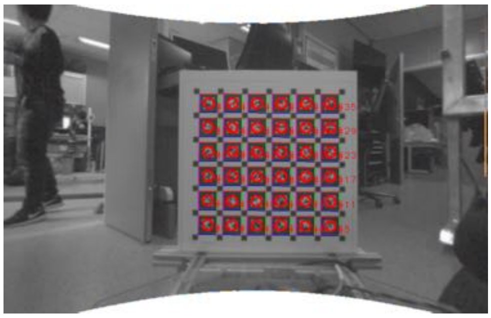
The range data from the mounted laser scanner (Velodyne) is provided as a point cloud (Fig. 4):

- Topic: `/velodyne_points`
- Data type: `sensor_msgs/PointCloud2`
- Frame rate: 10 fps
- Synchronized: No
- Coordinate system: center of the scanner relative to vehicle



(a) Pattern fixed to the vehicle frame.



(b) Tags detected in the pattern.

Figure 5: Estimation of camera to vehicle transformation.

## 3.4 Calibration data

Details of the calibration of cameras are described in *D3.1 - Data representation design and implementation, sensor calibration*. The resulting camera parameters of the camera rig, both extrinsic and intrinsic, are present in a YAML file following the conventions of the Kalibr toolbox[4].

The pose of the camera rig relative to the base of the robot was obtained from an image of a pattern fixed to a known measured location on the base frame. The pattern contains 36 unique tags [2] and is visible at the front camera of the rig (Fig. 5a). This setting allows to calculate the 3D position of each tag wrt. the coordinate system of the robot base. The 2D locations of tags detected in the image (Fig. 5b) are matched with their known 3D coordinates and these correspondences are used to estimate the relative camera pose with the PnP algorithm. The resulting transformation calculated wrt. the tracking prism was added to the YAML calibration file.

## 3.5 Garden geometry

The garden geometry captured with a stationary laser scanner (Leica) is provided as a point cloud file (Fig. 6) in multiple resolutions, as listed in *D1.1 - Appendix A: Leica Point Clouds*.

- Fields: XYZ RGB[5]
- Coordinate system: world, same as tracker
- Native file format: PTS - Leica ASCII file
  - Number of individual scans: 22
  - Accuracy: 3 mm (at 40 m)
  - Original size: 235M points
  - Filtered resolution: 10 mm

[4] https://github.com/ethz-asl/kalibr/wiki

[5] Color information was captured with delay and does not correspond exactly due to dynamic movements in the garden.

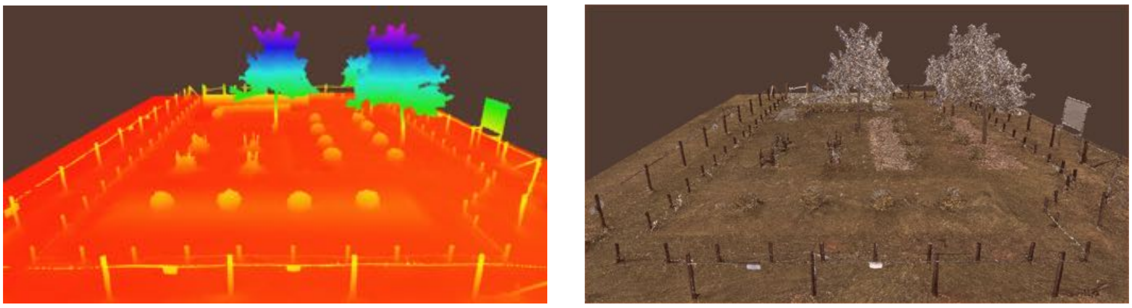


Figure 6: Point cloud of the test garden. Left: height-colored. Right: captured RGB color.

- Exported to standard formats:
  - PCD[6] - Point Cloud Library format (ASCII / binary)
  - PLY[7] - Polygon File Format (ASCII / binary)

[6] http://pointclouds.org/documentation/tutorials/pcd_file_format.php
[7] http://paulbourke.net/dataformats/ply/

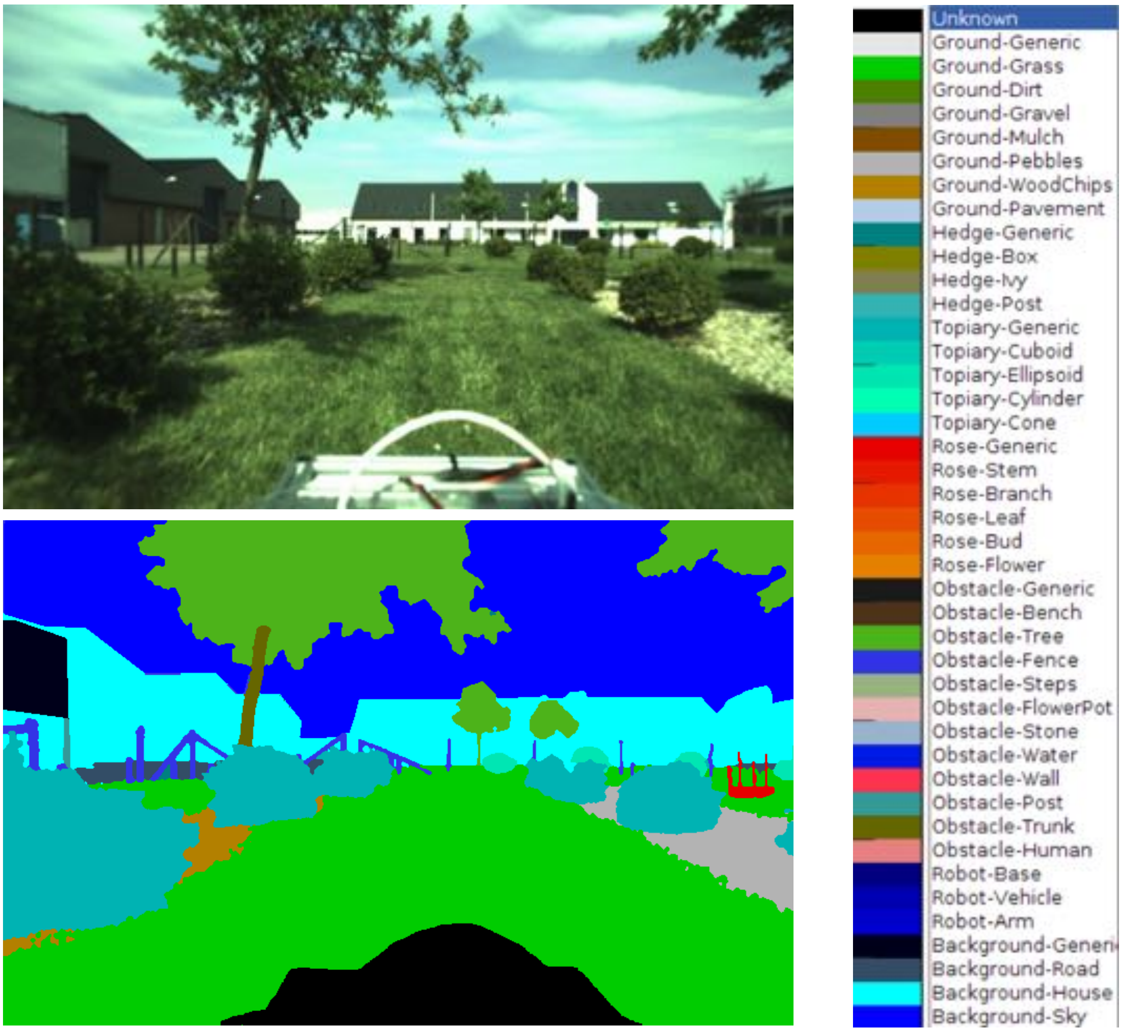



Figure 7: Sample annotated image. The captured image (top left) and its semantic map (bottom left) with color-coded semantic classes (right).

# 4 Semantic annotation

A subset of the captured data is manually annotated to indicate the semantic classes of the objects appearing both in the images (2D) and the geometry (3D). We have chosen a list of labels (Tab. 2) relevant to the garden environment. Labels have assigned unique colors used for visualization as shown in Fig. 7.

If details of the object are not identifiable a generic label (eg. *Rose-Generic*) is used. Specific labels (eg. *Rose-Stem*) are used when objects get close enough (ie. cover more than 10 pixel area). *Obstacles* are present inside of garden area. Anything outside garden perimeter (ie. behind fence, wall) is generally considered *Background*.

## 4.1 Annotation workflow

The time consuming task of human annotation of image sequences can be facilitated by projection of annotated 3D geometry (semantic point cloud) into images given the camera poses. For

```yaml
Unknown: 0          # invalid or ignored segments
Ground:    # terrain surface
  Generic: 1        # general ground label
  Grass: 2          # tussocks (green)
  Dirt: 3           # fine bare soil (brown/grey)
  Gravel: 4         # small stone chippings (grey)
  Mulch: 5          # shredded bark (dark brown)
  Pebbles: 6        # larger round stones (grey)
  WoodChips: 7      # chipped pieces of wood (light brown)
  Pavement: 8       # solid or tiled concrete (grey)
Hedge:     # line of dense bushes with rectangular profile
  Generic: 10         # general label
  Box: 11             # dense buxus plant hedge
  Ivy: 12             # cimbing hedera plant on a support
  Post: 13            # support
Topiary:    # individual bushes of primitive shapes
  Generic: 20         # unspecified shape
  Cuboid: 21          # cube and similar
  Ellipsoid: 22       # sphere and similar
  Cylinder: 23        # round or elliptic
  Cone: 24            # upper and lower diameter
Rose:       # rose bushes
  Generic: 30         # general rose label
  Stem: 31            # main part of the plant
  Branch: 32          # fork the stem
  Leaf: 33            # attached to branches
  Bud: 34             # before blossom
  Flower: 35          # blossoming
Obstacle:   # structures blocking space
  Generic: 100 # general obstacle label
  Bench: 101      # chair, perch
  Tree: 102       # tree crown
  Fence: 103      # wooden, wired
  Steps: 104      # edge
  FlowerPot: 105 # ceramic
  Stone: 106      # large
  Water: 107      # pond, stream
  Wall: 108       # separating
  Post: 109       # pole, lamp
  Trunk: 110      # below crown
  Human: 111      # person
Robot:    # itself or other
  Base: 200       # charging
  Vehicle: 201    # chassis
  Arm: 202        # mounted
Background: # outside garden
  Generic: 220    # unspec.
  Road: 221       # driveway
  House: 222      # buildings
  Sky: 223        # above
```

Table 2: List of semantic object classes distinguished in the project (YAML format).

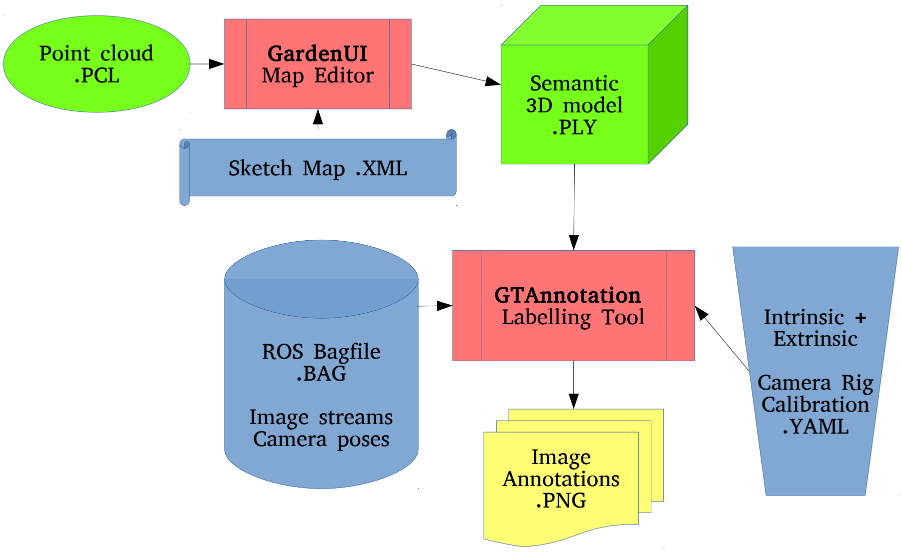



Figure 8: Semantic annotation workflow.

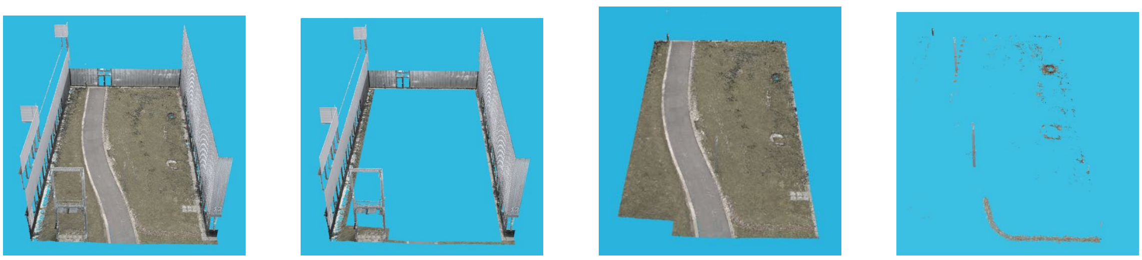
this purpose we have designed a workflow (Fig. 8) consisting of the following steps:

#### 4.1.1 Point cloud segmentation

The point cloud is split in three parts as in Fig. 9:

1. **Background** outside of garden perimeter is manually cropped out with CloudCompare[8] software.
2. **Ground surface** (terrain) is separated with segmentation method [3].
3. **Individual objects** are identified as connected components of the remaining point cloud.

The process is semi-automatic, i.e. the results after each step are inspected and manually fixed as needed.



(a) input (b) background (c) ground (d) objects

Figure 9: Segmentation of a point cloud. The input (left) is split into three parts (right).

[8] http://www.cloudcompare.org/

### 4.1.2 Initialization of 3D sketch map geometry

The sketch map editor in `GardenUI` (described in Sec. 4.2) is employed to generate a sketch map from the segmented point cloud (background excluded):

- **Objects** are initialized as bounding boxes around cloud segments.
- **Ground mesh** is initialized as Delaunay triangulation (DT) of uniformly sampled ground surface point cloud segment.

Both parts are manually adjusted, e.g. to prevent overlaps of the object bounding boxes.

### 4.1.3 Assignment of semantic labels to sketch map

A semantic label is manually assigned to every object and every ground mesh face in the `GardenUI` (Sec. 4.2). Where required the vertices of the ground surface mesh are moved to match boundaries between different surface types.

### 4.1.4 Transfer of labels to point cloud

Every point in the original point cloud gets an index of a semantic label and the corresponding label color. First, object bounding boxes are used to label points inside of them. The remaining ground points get the label of a surface mesh face onto which they vertically project.

### 4.1.5 Projection of point cloud to image frames

The previously produced semantic point cloud is loaded in `GTAnnotation` tool (described in Sec. 4.3) together with recorded images and camera calibration. Since the camera poses are approximately known, the points can be projected onto the images. Holes between the projected 2D points are covered with Delaunay triangulation, then all triangles with the same label of vertices and similar depth are filled inside with that label.

### 4.1.6 Manual adjustments of labels in editor

The editing capabilities of the `GTAnnotation` tool (Sec. 4.2) are used to refine the projected semantic map to match the corresponding image.

### 4.1.7 Transfer of annotation to the next frame

The labels can be transfered from the current frame to the next one using the correspondences from optical flow [1]. The obtained labels are approximately correct, but usually need further adjustments.

## 4.2 Sketch map editor

In order to obtain a semantic point cloud, the essential step mentioned above is to mark regions in the 3D space corresponding to different objects. The Garden User Interface (`GardenUI`) allows to draw a sketch map of the garden, where 2.5D geometry of terrain and garden objects have shape and semantic labels assigned (Fig. 10).

In the orthogonal 2D view (Fig. 10a) user has the following editing options:

- insert or remove vertices of the ground mesh (control points),
- move a selected vertex (location $X$, $Y$),
- adjust the elevation ($Z$) of a selected vertex or face,
- insert objects of primitive shapes (spheres, cubes, cylinders, cones),
- change dimensions (diameters $DX$, $DY$, $DZ$) and orientation (rotation angles $RX$, $RY$, $RZ$),
- assign the semantic label from the list (Tab. 2) to a selected face of the ground mesh or object.

The 3D view mode (Fig. 10b) allows arbitrary rotation of the map but the points or objects cannot be moved or inserted.

It works with point clouds to support workflow given in Sec. 4.1:

- import a segmented point cloud and initialize objects from its components,
- export a semantic point cloud with labels corresponding to the current sketch map.

Additional capabilities for registration and navigation are described in *D1.2 - Platform 2*.

## 4.3 Annotation tool

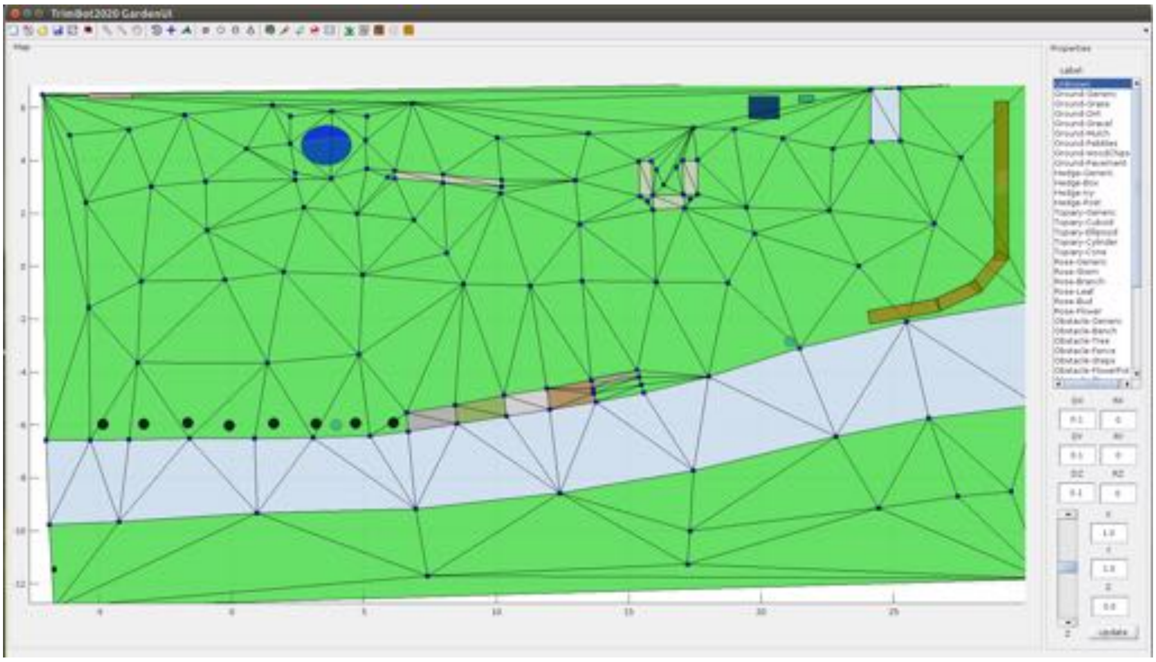
The GT annotation tool (`GTAnnotation`) allows to load a bagfile with multiple image streams together with camera calibration and semantic 3D model from a point cloud, which can be projected into the images. The workspace (ie. calibration + 3D model + bagfile) can be saved in a configuration file (YAML) and loaded later.

The drawing interface (Fig. 11) has the following functionality:

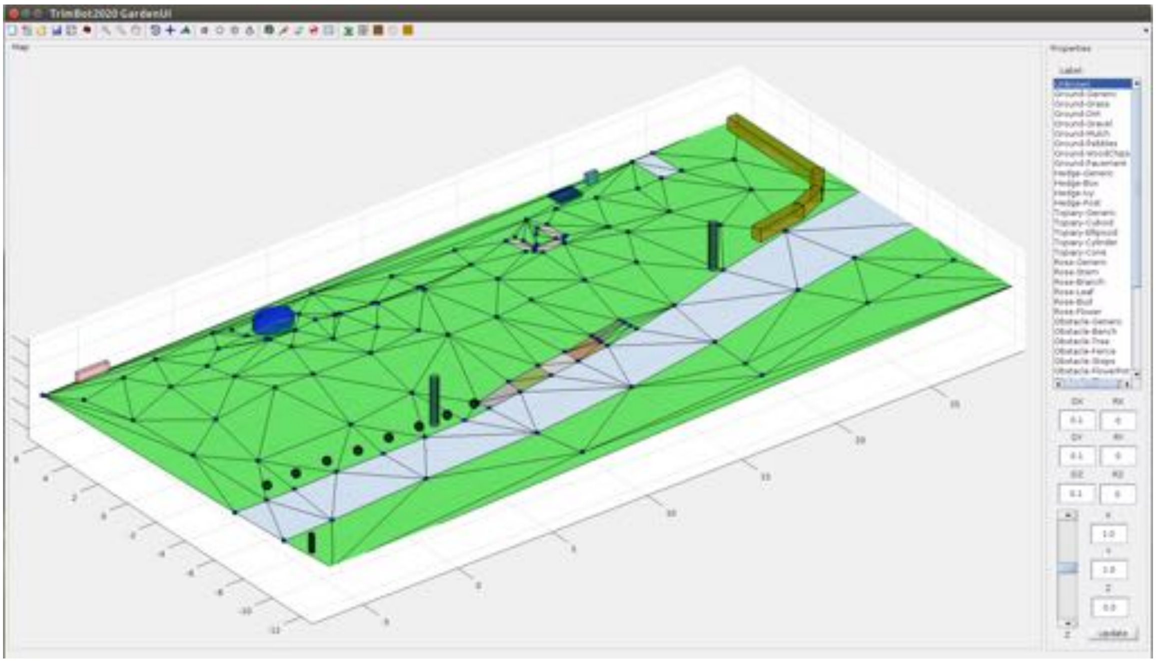
- transparently overlay semantic labels on original image (adjustable opacity),
- navigate in multiple camera topics and image frames,
- draw user selected semantic labels with a brush of adjustable size,
- draw region boundary lines,
- fill region (semantic / image),
- automatically refine annotation boundaries to align with contours of the original image,
- translate and rotate the current semantic map in the image frame.

The camera pose associated with the current frame (translation and rotation) can be also manually adjusted to better fit the projection of a semantic point cloud to the image (Sec. 4.1.5).



(a) 2D view



(b) 3D view

Figure 10: User interface of the sketch map editor showing mesh of terrain and objects fitted to GT point cloud. Blue markers are control points of the terrain mesh.

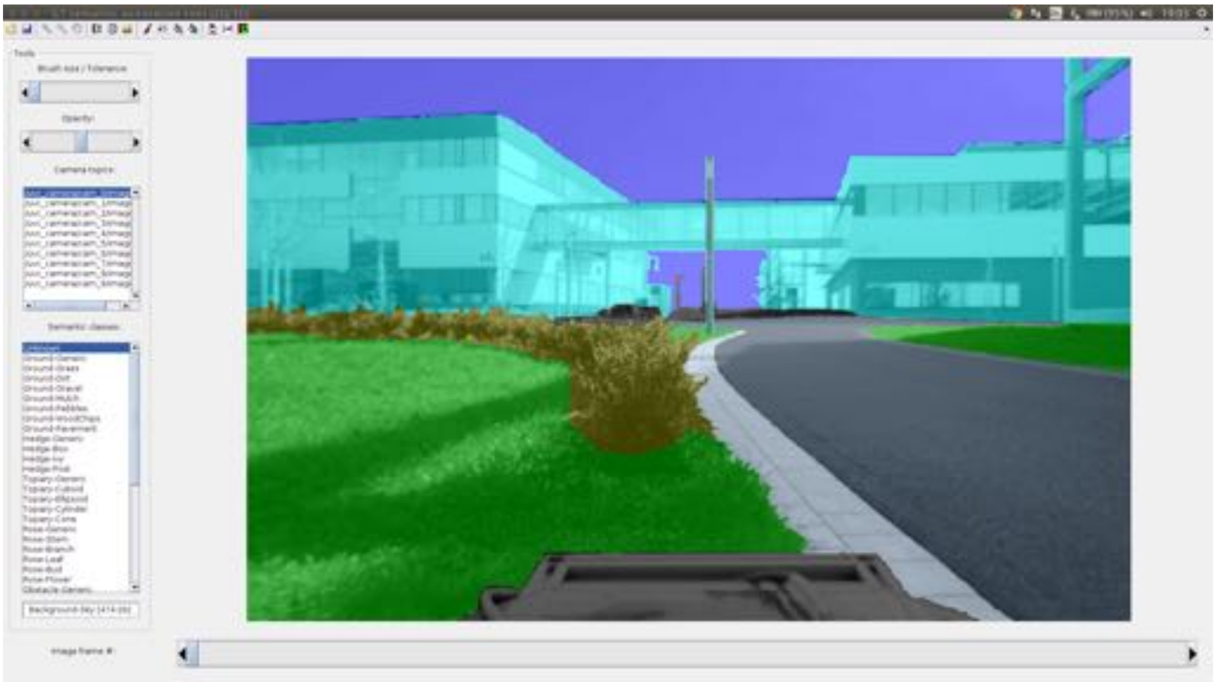


Figure 11: User interface of the developed semantic annotation tool.

## 4.4 Data formats

### 4.4.1 Label specification

Corresponds to Tab. 2.

- Format: YAML
- labels.yml
- colors.yml

### 4.4.2 Semantic point cloud

Geometry as in Sec. 3.5 extended with semantic information.

- Format: PLY
- Fields: XYZ RGB L
  - color according to semantic label of a point
  - additional field L contains the semantic label index
  - values as defined in Sec. 4.4.1

### 4.4.3 Semantic bitmaps

Pixelwise semantic map as shown in Fig. 7.

- Format: Indexed PNG

- Range: values from label set in Sec. 4.4.1
- Colormap: embedded, follows Sec. 4.4.1.

# 5 Datasets

The collected data are organized into datasets according to the data collection events as given in Tab. 3. Appendix A enumerates different recorded and annotated scenarios, which are summarized by types in Tab. 4.

| *Short name* | *Repository (GitLab)* | *Scenarios* | *Annotated* | *Note* |
|---|---|---|---|---|
| **WU1** | GardenDatasetWageningen2016 | 18 | 0 | old setup |
| **WU2** | GardenDatasetWageningen2017 | 18 | 4 | challenge |
| **BOSCH1** | GardenDatasetBosch2016 | 10 | 0 | no tracking |
| **BOSCH2** | GardenDatasetBosch2017 | 14 | 2 | |

Table 3: List of collected datasets in alphabetical order.

| *Scenario type* | *Subject* | **WU1** | **WU2** | **BOSCH1** | **BOSCH2** | *Total* | *Annotated* |
|---|---|---|---|---|---|---|---|
| Approach | boxwood | 4 | | | 3 | 7 | 1 |
| | hedge | 1 | 4 | | 1 | 6 | 1 |
| | obstacle | 4 | | | | 4 | |
| | surface | 1 | | 1 | 2 | 4 | |
| Drive around | boxwood | 6 | 5 | 1 | 3 | 15 | 2 |
| | hedge | 8 | 4 | 3 | 1 | 16 | 1 |
| | roses | 4 | 1 | 1 | | 6 | |
| | garden | 3 | 4 | 2 | 2 | 11 | 1 |
| Hand-held | boxwood | 7 | | | | 7 | |
| | hedge | 3 | | | | 3 | |

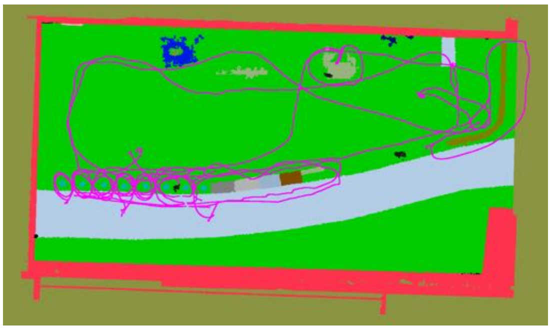
Table 4: Number of collected and annotated sequences of different types in datasets.

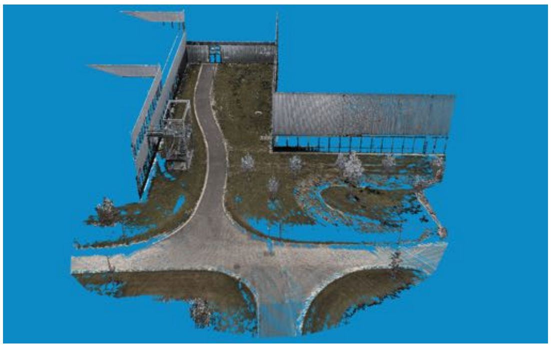
## 5.1 Renningen garden (BOSCH)

This garden is surrounded by buildings on three sides and the last side is open. It features additional obstacles and hilly terrain.(Fig. 12). The list of completed annotations is given in Tab. 5.

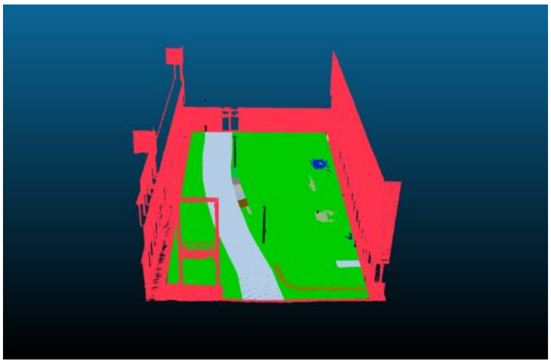
(a) General view



(b) Map view with recorded vehicle trajectories (magenta: BOSCH2)



(c) Color point cloud



(d) Semantic label colored point cloud

Figure 12: Bosch garden, Renningen, DE.

| *Dataset* | *Scenario* | *Cameras* | *Frames* | *Range* | *Labeling effort* |
|---|---|---|---|---|---|
| **BOSCH2** | around_hedge | #0 | 117 | 175:1:292 | 5 man-days |
| **BOSCH2** | slalom_boxwoods | #0 | 63 | 180:10:800 | 2 man-days |

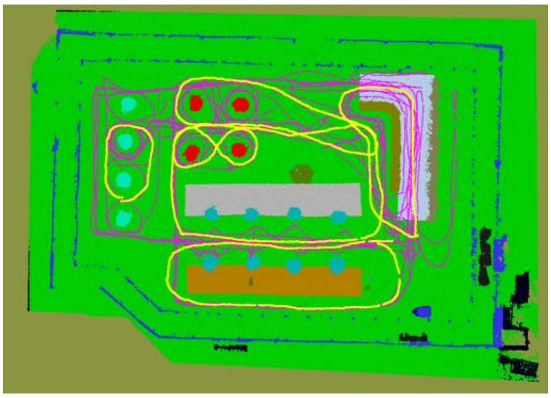
Table 5: List of annotated scenarios for BOSCH garden.

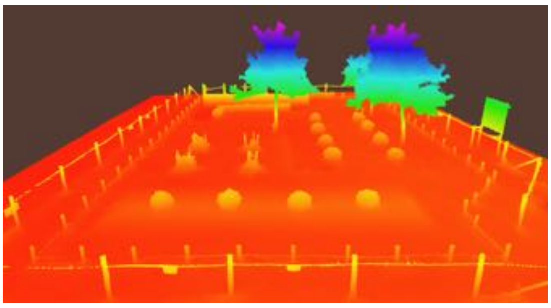
## 5.2 Wageningen garden (WU)

This garden is enclosed with a double fence and features a variety of bushes and mostly flat terrain (Fig. 13). The list of completed annotations is given in Tab. 6.

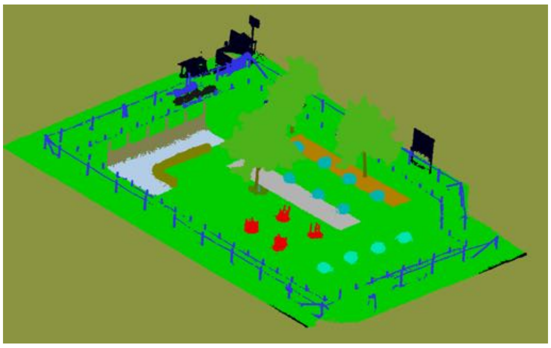
(a) General view



(b) Map view with recorded vehicle trajectories (magenta: WU1, yellow:WU2)



(c) Depth-colored point cloud



(d) Semantic label colored point cloud

Figure 13: WU garden, Wageningen, NL.

| *Dataset* | *Scenario* | *Cameras* | *Frames* | *Range* | *Labeling effort* |
|---|---|---|---|---|---|
| **WU2** | train_around_hedge | #0, #2 | 92 | 100:10:550 | 11 man-days |
| **WU2** | train_boxwood_row | #0, #2 | 114 | 90:10:650 | 11 man-days |
| | | #0 | 42 | 140:1:185 | 4 man-days |
| **WU2** | train_boxwood_slope | #0, #2 | 46 | 120:10:340 | 4 man-days |
| **WU2** | test_around_garden | #0, #2 | 268 | 140:10:1480 | 20 man-days |

Table 6: List of annotated scenarios for WU garden.

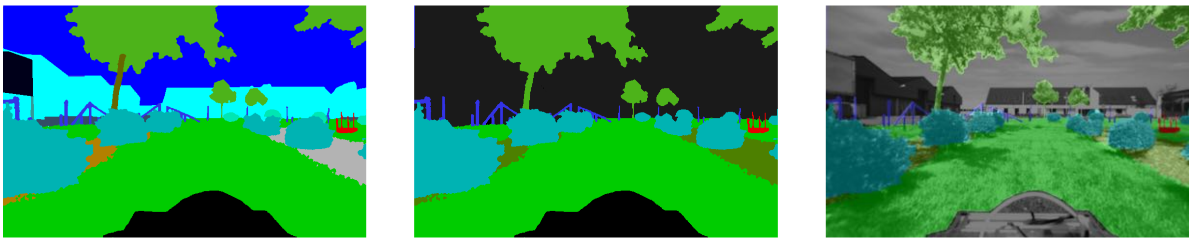
### 5.2.1 Workshop challenge

The first part of the dataset was publicly released in July 2017 to provide training and testing data for a challenge connected with a workshop[9] organized by the project members.

[9] *3D Reconstruction meets Semantics*, in conjunction with ICCV 2017 conference. http://trimbot2020.webhosting.rug.nl/events/3drms/

```
Unknown: 0
Grass: 1
Ground: 2
Pavement: 3
Hedge: 4
Topiary: 5
Rose: 6
Obstacle: 7
Tree: 8
Background: 9
```

Table 7: Reduced list of semantic object classes for workshop challenge (YAML format).



(a) Full set annotation. (b) Reduced set annotation. (c) Reduced over captured image.

Figure 14: Reduction of label set.

The full label set (Tab. 2) was largely reduced for the purpose of the challenge as shown in Tab. 7. The WU2 annotations (all from Tab. 6) were transformed to export bitmaps with reduced label set (Fig. 14). The resulting dataset is available from a public repository[10].

## 5.3 Data storage and availability

The full datasets are currently available from internal project repository to consortium members. More specifically the files are hosted at UEDIN's Gitlab server[11] except for bagfiles and large point clouds which can be downloaded from a dedicated MARV Robotics[12] site with total 377 GB in 102 files of 2h09m duration.

Selected parts of the dataset will be released publicly as presented in *D8.2 - Data Management Plan and Framework*.

## 5.4 Semantic annotation effort

The choice of scenarios for the semantic annotation was mainly driven by the goal to cover both gardens in terms of diverse trajectories and scenario types. The WU2 challenge dataset we

[10] https://gitlab.inf.ed.ac.uk/3DRMS/Challenge2017
[11] https://gitlab.inf.ed.ac.uk/TrimBot2020
[12] https://github.com/ternaris/marv-robotics

| *Site* | *Frames* | *Man-days* | *Avg. frames/day* | *Avg. min/frame* |
|---|---|---|---|---|
| BOSCH garden | 180 | 7 | 25.7 | 20 |
| WU garden | 562 | 50 | 11.3 | 42 |
| *Total* | 742 | 57 | 13.0 | 39 |

Table 8: Summary of semantic annotation effort.

focused on almost meets such a goal for the Wageningen garden. The current extent of Bosch garden annotation is still rather limited, also due to privacy restrictions at the Bosch campus.

The effort put into semantic GT annotation so far (July 2017) is summarized in the statistics in Tab. 8. The given time totals correspond to the primary work of several annotators (mostly students working part-time). Additionally estimated 20% more time was spent in the second round with inspection and corrections.

The WU garden is considerably more complex than the BOSCH garden, which results in higher annotation time. Similarly labeling of consecutive frames can leverage the label transfer (Sec. 4.3) to speed up the annotation process, compared to the case when every 10th frame is labeled and the view change is too large for label transfer.

# 6 Future plans

## 6.1 Additional data collection

Compared to Platform 1 used so far for the data collection the current design of Platform 3 (with arm mounted) will result in the change of height of the camera rig and different self-occlusions from the modified vehicle frame and supports. Additional data collection of vehicle data with Platform 3 will be performed as a part of integration meetings, particularly for the purpose of evaluation of navigation and reconstruction as proposed in *D7.3 - Component and System Evaluation Plan*.

The Platform 3 will also include a camera rig mounted on the arm. When the arm camera placement and rig setup is fixed we will capture several scenarios of the arm moving around different garden objects; the extent of this recording is yet to be determined.

## 6.2 Additional semantic annotation

We will continue with the semantic annotation on the BOSCH2 dataset to cover the area of the garden, in particular for different surfaces and obstacles. This is required in order to provide sufficient diversity and quantity for scene understanding training.

Detailed annotation for rose clipping has not been carried out so far. For this purpose we will enhance the annotation tool to transfer labels and clipping sites from 2D to 3D.

# A Appendix

This section enumerates all scenarios captured for the project GT datasets as of July 2017. Multiple recorded sequences (bagfiles) for given scenarios are listed with their time stamp and duration. Semantically annotated sequences are also indicated.

## A.1 Scenarios recorded for WU1 dataset

See *D1.1 - Sensor Data Collection including Ground Truth, Appendix B and C* for the detailed list of recorded data in Wageningen, September 2016. The following sections describe data recorded after D1.1 was released.

## A.2 Scenarios recorded for WU2 dataset

| *Scenario* | *Date and start* | *Duration* | *Annotated* |
|---|---|---|---|
| around boxwood row | 2017-05-16 17:07 | 2m 07s | |
| | 2017-05-17 10:28 | 2m 23s | |
| | 2017-05-17 10:32 | 2m 07s | |
| | 2017-05-17 16:15 | 1m 47s | |
| | 2017-05-17 16:49 | 2m 07s | Y |
| slalom boxwood on slope | 2017-05-16 15:56 | 1m 09s | |
| | 2017-05-17 10:37 | 2m 13s | Y |
| | 2017-05-17 10:39 | 1m 38s | |
| | 2017-05-17 16:46 | 1m 33s | |
| around hedge | 2017-05-16 17:05 | 1m 02s | |
| | 2017-05-17 10:19 | 1m 43s | |
| | 2017-05-17 10:21 | 1m 23s | |
| | 2017-05-17 14:57 | 1m 12s | Y |
| around garden | 2017-05-16 17:07 | 3m 59s | |
| | 2017-05-17 10:53 | 4m 41s | Y |
| | 2017-05-17 16:52 | 5m 23s | |
| | 2017-05-17 16:59 | 4m 16s | |
| around rose | 2017-05-17 10:04 | 1m 24s | |

Table 9: List of vehicle scenarios in WUR2.

## A.3 Scenarios recorded for BOSCH1 dataset

| *Scenario* | *Date and start* | *Duration* |
|---|---|---|
| around boxwood | 2016-12-13 10:05 | 1m 10s |
| around obstacles | 2016-12-13 13:13 | 2m 10s |
| | 2016-12-13 10:09 | 1m 30s |
| loop over hill | 2016-12-13 13:07 | 3m 48s |
| | 2016-12-13 10:22 | 2m 45s |
| around hedge | 2016-12-13 09:59 | 18s |
| | 2016-12-13 10:02 | 1m 02s |
| | 2016-12-13 13:04 | 1m 20s |
| around roses | 2016-12-13 10:31 | 1m 13s |
| approach pebbles | 2016-12-13 10:07 | 57s |

Table 10: List of vehicle scenarios in BOSCH1.

## A.4 Scenarios recorded for BOSCH2 dataset

| *Scenario* | *Date and start* | *Duration* | *Annotated* |
|---|---|---|---|
| slalom boxwood | 2017-03-29 14:29 | 2m 08s | |
| | 2017-03-29 14:32 | 2m 40s | Y |
| | 2017-03-29 12:47 | 2m 47s | |
| around different surfaces | 2017-03-29 12:51 | 3m 09s | |
| | 2017-03-28 15:18 | 3m 27s | |
| around obstacles | 2017-03-29 12:39 | 1m 52s | |
| around hedge | 2017-03-29 14:24 | 2m 51s | Y |
| approach hedge | 2017-03-28 14:49 | 3m 35s | |
| around garden | 2017-03-28 14:59 | 7m 18s | |
| | 2017-03-29 12:28 | 6m 44s | |
| around boxwood | 2017-03-29 13:07 | 1m 01s | |
| | 2017-03-29 13:01 | 1m 34s | |
| | 2017-03-28 13:42 | 2m 34s | |
| along boxwood row | 2017-03-29 13:04 | 2m 23s | |

Table 11: List of vehicle scenarios in BOSCH2.

# References

[1] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Computer Vision and Pattern Recognition*, 2015.

[2] Andrew Richardson, Johannes Strom, and Edwin Olson. AprilCal: Assisted and repeatable camera calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

[3] Wuming Zhang, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xie, Xiaoyan Wang, and Guangjian Yan. An easy-to-use airborne lidar data filtering method based on cloth simulation. *Remote Sensing*, 8(6), 2016.