Abstract: This is the first TrimBot2020 Periodic Report. The project is in good shape, with a prototype vehicle capable of autonomous navigation in the project test gardens, using a real-time 10 camera stereo system. 3D data is matched to sketch map, by which users specified the desired destination. Two prototype end effectors have been developed and initial experiments with them attached to the proposed robot arm have been trialled. For later integration, prototype scene-flow, structure and motion 3D processes are working, as is a prototype semantic segmentation algorithm. A functioning web site is performing well, with 10,000 page views so far, showing public deliverables, and example photos and videos.
The overall goal of the TrimBot2020 project is to develop and evaluate the technology needed by an autonomous hedge trimmer. It addresses 2 core issues: 1) advancing outdoor robotic technology: low cost robot vehicles for interacting with the natural environment and reliable 3D sensing for robot localisation and scene understanding, and 2) demonstration of a hedge and rose cutting prototype, for potential economic exploitation.

The overall objectives are to develop and demonstrate: 1) novel robotic end effectors for garden trimming, 2) a low cost mobile platform for deploying the end effectors, and 3) reliable outdoor 3D sensing for robot navigation, object and scene structure recognition, and trimming control.

This requires innovations in: reliable outdoor 3D sensing, low-cost ground-based mobile robotics with a manipulator arm, development of new end effectors for hedge, bush, and rose trimming, algorithms for scene modelling, scene structure recognition, robot navigation and servoing, deformable map registration, cut planning, servo and control.

1 Expanded Project Highlights
The overall goal of the TrimBot2020 project is to develop and evaluate the technology needed by an autonomous hedge trimmer. This application will require innovations in: reliable outdoor 3D sensing, low-cost ground-based mobile robotics with a manipulator arm, development of new end effectors for hedge, bush, and rose trimming, algorithms for scene modelling, scene structure recognition, robot navigation and servoing, deformable map registration, cut planning, servo and control. The robot will be evaluated in two outdoor model gardens.

The project has made great progress in the first review period. The highlights are:

- **Prototype Robot Vehicles**

  Eventually three versions of the mobile platforms will be built. Platforms 1 and 2 have now been built (without a mounted arm), and the third (final demonstrator) platform designed. Figure ?? (left) shows platform 2, which will be demonstrated at the first review. It has five stereo camera pairs in a pentagonal setup used for visual SLAM, a network router for online operation and additional batteries for onboard computers and sensors. A CAD model of the final demonstrator with arm and trimming tool is shown in Figure ?? (right).

  Platform 1 was used for the ground truth data collection. Examples of the vehicle and ground truth data are shown in the video of milestone 2: [https://www.youtube.com/watch?v=BtL2fuPVSs](https://www.youtube.com/watch?v=BtL2fuPVSs).

- **Autonomous Navigation**

  Navigation software is integrated into platform 2 to support the online autonomous operation, where a user indicates a destination using the SketchMap User Interface (UI), and the vehicle navigates (open-loop) to the indicated destination. Figure ?? shows the vehicle in front of a bush and the remote laptop with the UI which was used to define the goal positions for the vehicle. The vehicle is also able to perform an approaching manoeuvre to reach a desired trimming position in front of a target object.
- **Novel End Effectors**

  The project designed and built two novel servo motor controlled end effectors, one designed for bush clipping, the other for rose stem cutting. For the bush trimming, a configuration of circular counter-rotating saw blades was chosen to ensure a high success rate to cut small branches in one go. For the single branch cutting, a commercially available electrical pruner was modified in such a way that it can be mounted on a manipulator. This device can cut through 14 mm of softwood. Fig. ?? shows the two prototype end effectors.

![Prototype End Effectors](image)

Figure 3: Constructed version 1 of the boxwood trimming tool (left image) and components of the prototype version 1 of the rose cutter (right image).

The project chose the six degrees of freedom lightweight Kinova Jaco2 arm to be mounted on the mobile platform, and to carry the end effectors. A test rig was fabricated that allows to securely place the manipulator with tools in a stationary position in either a laboratory setting or in the open field.

Initial algorithms for fitting spherical shapes to plant data have been created. Operational tests on real plants have successfully been carried out, including planning and cutting of spherical bushes. See Fig. ?? for examples of the first rose and bush cutting experiments. A video of the first experiments can be seen here (GIT password protected):
Integration and Robot Control

The design and concept phase for hardware and software is completed. All data structures and component interfaces have been designed, and are implemented as Robot Operating System (ROS) messages, as are the control interfaces. The first successful integration of the vision pipeline and the motion planning pipeline was demonstrated. The core vehicle components have been integrated: vehicle control, user sketch map, navigation and obstacle avoidance, stereo-based 3D image data capture, and SLAM garden modelling. Future work will integrate the depth-net 3D scene data, scene semantic segmentation, trimming planning and servo-based trimming.

3D Sensing

A multi-camera system was built using 10 cameras (Fig. ??(left)), arranged into 5 stereo pairs covering the full 360° horizontal field-of-view (FOV), and is intended for mounting around the robot arm. Each stereo pair consists of a colour and a grayscale camera. The colour camera is needed for semantic processing while the grayscale camera is more light sensitive. An FPGA implementation of ETHZ’s stereo algorithm produces a depth image registered to the colour image. All cameras are synchronised and produce a video stream consisting of the intensity and depth images at 12 frames/second. Fig. ??(right) shows an example panorama captured by the cameras.
Figure 5: (left) The multi-camera system, consisting of 5 stereo pairs arranged in a pentagon shape, used by the TrimBot2020 robot. (right) Panoramic view composed of the five individual stereo camera views, one per column, from the multi-camera system. A video animation of the panoramic view can be found at https://polybox.ethz.ch/index.php/s/DtfR8akFjYYDSpa.

Figure 6: A 3D map generated by our SLAM system from data captured in the test garden in Wageningen. The black points correspond to the recovered 3D structure while the coloured dots correspond to the poses estimated for each camera in the multi-camera system.

- **3D Data Processing**

A SLAM (Simultaneous Localisation and Mapping) system was developed to estimate the motion of the robot and the 3D structure of the scene from the data captured by the multi-camera rig. The SLAM system uses all 10 cameras in the rig, modelled mathematically as a generalised camera, for accurate and stable estimation. The main novelty of the approach is an online self-calibration procedure that computes the extrinsic parameters of the multi-camera system. This avoids the tedious manual process of acquiring enough images of the marker boards for offline extrinsic calibration. Fig. ?? shows a 3D map generated by our system.

Innovative approaches were developed for estimating optical flow from videos, disparities from stereo images, and depth from a single moving camera. These innovations are based on the recent advances in convolutional networks, which were engineered to formulate the above-mentioned tasks for the very first time as learning problems. The advantages of this approach are: 1) interactive frame rates while providing state-of-the-art accuracy, and 2) the approaches can learn about and exploit garden-specific structure relationships. The DeMoN network for two-frame structure from motion uses not only the motion parallax
between the image pair for depth estimation, but it also uses subtle cues from shading and texture and the typical shape of objects to fill in information where measurements are missing. The disparity estimation network (DispNet) can be employed wherever there is a stereo camera with horizontal baseline. The optical flow estimation network (FlowNet) can be employed to estimate motion in image sequences, for example, to track the branches of a bush over time. The depth estimation network (DeMoN) is a first building block for dense depth estimation from a moving monocular camera and can be employed also for visual odometry from dense image analysis in cases where classical sparse descriptors have problems. Figure ?? shows a depth image computed from two consecutive input images.

A video illustrating DeMoN is here: [https://www.youtube.com/watch?v=Rat9-nyVd2s](https://www.youtube.com/watch?v=Rat9-nyVd2s)

A video illustrating FlowNet 2.0 is here: [https://www.youtube.com/watch?v=JSzUdVBmQP4](https://www.youtube.com/watch?v=JSzUdVBmQP4)

- **Scene Semantic Decomposition**
  Reliable garden navigation and trimming requires identifying different scene structures. Depth/3D gives the location of the structures, but colour image data helps with the identification of the class of the structure (e.g. whether grass (driveable) or gravel (not driveable)). The project approaches the scene analysis problem in a data-driven fashion. The final goal is to semantically understand a scene captured by the robot’s camera system using deep learning methods. A novel deep-net architecture has been proposed to solve image decomposition problem in a data-driven fashion. Figure ?? illustrates this decomposition (using synthetic image data).

- **Evaluation + Demonstrators highlights**
  The progress of the project will be demonstrated in three stages: navigation, trimming and the two combined. In 2017 the effort concentrates on the first demonstrator (although activity towards the second has started).

  The project team will demonstrate the robot autonomously navigating to user selected locations in the Renningen test garden (weather permitting - or a video will be shown).
To test the tools and later the integrated robot, two test gardens were constructed at the Wageningen University & Research Centre and at the Robert Bosch Renningen campus. Figure 8 shows a sketch map of the Wageningen garden design and a photo of the actual garden. The Bosch test garden is mainly used for navigation tasks, however it also contains some bushes, roses and a hedge. Figure 9 show some images from the test garden in Renningen.

Figure 8: Example of image decomposition into intrinsic characteristics

![Input RGB](image1)
![Albedo GT](image2)
![Our Albedo Prediction](image3)
![Shading GT](image4)
![Our Shading Prediction](image5)

Figure 9: Sketch map of test garden in Wageningen and photo of the realised garden.

- **Ground-truth data collection**

The project will evaluate the positional accuracy of the robot navigation and trimming, as well as the accuracy of its visual understanding. The positional accuracy will be measured in the test garden.

For the scene understanding (and for providing the training data needed for developing the components), ground-truth data was collected. Sixty varying video sequences were captured in the two test gardens built for the project purposes. 3D semantic point clouds were obtained for both gardens and for 742 images a 2D semantic map was created. Semantic annotation of the captured data was carried out on 6 sequences so far and custom GUI applications were developed for this purpose.

As part of the ground truth data recording sessions both test gardens were measured using a Leica ScanStation P15. The resulting point clouds can be used as reference point clouds for ground truth data processing. Figure 9 shows a reference point cloud from the entire test garden in Wageningen.
Figure 10: Test garden in Renningen (a) all roses in Renningen, (b) a row of overgrown boxwoods.

Figure 11: Reference point cloud of the test garden in Wageningen.

- **Dissemination and Exploitation**

  Four major planning documents have been developed and released to help shape the project’s Dissemination and Exploitation strategy:


  A dissemination plan was developed and dissemination activities have been organised in
five phases: Preparatory phase, Scientific dissemination, Communication to general public, Teaching/education, Project exploitation. In the preparatory phase, now concluded, all the tools for dissemination have been configured and are being used.

The project website is available at the URL: [http://www.trimbot2020.org](http://www.trimbot2020.org) and raised awareness of the TrimBot2020 project to about 2000 users (with more than 3400 started visitor sessions). Figure ?? shows the growth of webhits on the project website.

The project has also set up social media profiles on the four most popular community interaction platforms:

- Twitter: [https://twitter.com/trimbot2020](https://twitter.com/trimbot2020)
- Facebook: [https://www.facebook.com/pg/trimbot2020](https://www.facebook.com/pg/trimbot2020)
- ResearchGate: [www.researchgate.net/project/EU-TrimBot2020-Gardening-robotics](http://www.researchgate.net/project/EU-TrimBot2020-Gardening-robotics)
- A YouTube channel with 7 videos: [https://www.youtube.com/channel/UCbPCq-c_Gsamuyjgl81rWGA](https://www.youtube.com/channel/UCbPCq-c_Gsamuyjgl81rWGA)
2 Expanded Original Contributions
So far, the project has made the following original scientific and engineering contributions:

1. **A novel low cost robot for garden trimming** - A Bosch Indego robot base was adapted as a first prototype and then used as inspiration for a new robot base design (in development). The first prototype enables vision-guided autonomous navigation in a garden context, including sensing, wifi, enhanced power, communication, and computing capability.

2. **Hedge trimming end-effector** - A hedge trimming tool with circular counter-rotating saw blades was built and passed initial tests. The blades are driven by means of a flat electrical servo motor and gearbox, which allows for variable speeds and current monitoring for cutting feedback.

3. **Rose stem trimming end-effector** - A commercially available Bosch Ciso electrical pruner was modified to be mounted on the manipulator. The original DC motor was replaced by a servo controlled DC motor for monitoring cutting. It has passed initial tests.

4. **Garden vehicle navigation module** - A state machine adapted for outdoor garden navigation, along with suitable planning and execution modules have been developed. Planning is sketch-map based, and takes account of obstacles/proto-objects and driveable regions. Execution takes account of unexpected obstacles and includes trajectory replanning. The developments are extensions of state-of-art ROS navigation (move_base).

5. **Pentagonal 10 camera ring** - The project has developed a 5 stereo pair / 10 camera system for TrimBot2020 use, which provides 360 degree coverage around the vehicle. Each camera pair has one colour and one monochrome camera, and all cameras are synchronised for simultaneous acquisition, using previously developed FPGA technology.

6. **Real-time multi-camera stereo** - The FPGA was adapted from ETHZ’s previously developed stereo algorithm to provide real-time depth maps from for the five stereo pairs. In addition, the FPGA allows for real-time multi-baseline stereo, possibly for use on the arm as well as the vehicle.

7. **360 degree SLAM** - We implemented a feature-based SLAM approach for the multi-camera system, modelling the camera system as a generalised camera (multiple centres of projection). A novel self-calibration module (based on SfM) computes the extrinsic calibration between the cameras from scratch during online operation, based on data acquired as the camera system moves, and exploiting the rigid motion constraint on the 10 cameras.

8. **Binocular and Trinocular Stereo** - The binocular dominant plane sweep stereo algorithm was enhanced to preserve quality in textureless areas through a content-aware adaptive window aggregation scheme. A novel cost function and a novel hypothesis generator were added to achieve state-of-the-art results on the well-known KITTI and Middlebury benchmarks. An extension to trinocular stereo by using a third image after camera motion combines binocular and motion stereo. The main novelty of our algorithm is an integrated baseline recovery step.
9. **Dataset and benchmarking of (Multi-View) Stereo and Local Features** - A new benchmark dataset was created and used to compare hand crafted vs deep learned features for the task of feature-based SfM. The hand-crafted features (SIFT based) were faster and performed better, whereas the learned features are greatly affected by scene characteristics. A novel benchmark dataset containing both high-resolution images and video sequences depicting a diverse set of real-world scenes was created for evaluation of (multi-view) stereo algorithms. This dataset will be used to train and evaluate some of the TrimBot2020 algorithms.

10. **Dataset for training outdoor semantic segmentation algorithms** - A large-scale synthetic dataset was created, which is dedicated for garden semantics, with different ground truth information: pixel-wise semantic annotation, intrinsics, depth, scene flow, surface normals, etc.

11. **Deep network for intrinsic image decomposition** - A novel deep-net architecture was constructed to solve image decomposition problem in a data-driven fashion. This decomposes an image into intrinsic properties, such as albedo and shading.

12. **Optical flow deep network** - A deep network was trained end-to-end for optical flow estimation with state-of-the-art accuracy and interactive frame rates.

13. **Two-frame structure from motion deep network** - A deep network was trained end-to-end for two-frame structure from motion for interactive frame rates and improved accuracy compared to traditional two-frame structure from motion techniques.

14. **ROS node for noisy data sphere fitting** - A ROS node was created for fitting a sphere to a 3D point cloud, here applied to the data from a spherical bush observed with a single stereo pair.

15. **A ROS architecture for control of a garden trimming robot** - A robot control system based on ROS messages between all components was designed. The state-machine based control system takes account of route planning around known obstacles and no-go areas (non-traversable terrain), navigation to locations specified by users using a garden sketch map user interface, and unexpected obstacle detection and replanning. Future developments will extend the state machine for trimming actions.

16. **Successful demonstration of garden navigation using a low-cost garden trimming robot** - The robot can navigate to user-specified locations avoiding obstacles and non-traversable surfaces.

17. **Outdoor robot navigation dataset** - We have created a set of ROS bags recording raw data for 6 runs captured in the Wageningen test garden. Each bag contains 10 camera intensity, colour and depth video, 3D-LIDAR (Velodyne VLP-16), high precision IMU (e.g. STIM 300) and external laser tracker. This unique dataset may be of interest to researchers investigating vision based localisation navigation, in that one can compare visually estimated position to that measured by the IMU and laser tracker.

18. **Tool for 2D-3D scene ground-truth labelling** - This improved the efficiency of ground-truth labelling of stereo image data, by projection of a semantically labelled 3D point
cloud into the individual camera images. It also allows tracking of motion over consecutive image frames to propagate labels. Some manual corrections are always subsequently needed. The labels are classes of objects seen outdoors: grass, paths, trees, bushes, etc.

19. **Semantically annotated garden dataset (2D+3D)** - A semantically annotated garden dataset (2D+3D) was released, among the first of its kind featuring natural outdoor scenes (Fig. ??). A subset is being used currently in the ICCV 2017 workshop “3D Reconstruction meets Semantics” challenge. The full dataset will be released publicly soon.
3 Workpackage Progress
3.1 WP1: Mobile platform construction

Mobile Platforms

In WP1 three mobile platforms will be built. Platforms 1 and 2 have now been built and for the final demonstrator the conception has been completed. All platforms are based on modified Indego lawn mowers. Platform 1 was equipped with 6 monochrome and two colour cameras from ETHZ. The cameras were placed on a tower in the centre of the vehicle and were arranged in an octagonal setup. For the second platform the camera ring was replaced by an pentagonal camera setup with five stereo camera pairs. Figure ?? shows both vehicles. On the left is the platform 1 vehicle with the octagonal camera setup and on the right is the platform 2 vehicle with the pentagonal camera setup.

![Platform 1](a) ![Platform 2](b)

Figure 13: (a) mobile platform 1 used for ground truth data recording. (b) mobile platform 2 supports online operation and navigation.

In addition to the cameras several sensors for ground truth data recording are integrated into the vehicle. A Velodyne VLP16 LIDAR sensor is mounted above the cameras to record a reference point cloud from the same point of view. An IMU sensor and an external position tracking system are used to estimate the 6-DOF ground truth position of the vehicle. The external position tracking system requires a prism reflector on the vehicle. For the online operation capability of platform 2 an additionally onboard laptop for SLAM and a Wifi router for the network connection is added the the vehicle. The network configuration with the Wifi router allows to control the vehicle from the remote laptop. The vehicle is also extended by an odometer. A more detailed description of platforms 1 and 2 is given in deliverables D1.1: Platform 1: Sensor Data Collection including Ground Truth, D1.2: Platform 2: Supports online operation for demonstrator 1 and D6.1: First Integrated Platform.

For the last demonstrator 3, the concept and design of the mobile platform is also completed. This platform will have the drive unit from the Indego lawn mower mounted on an aluminium frame. The use of an aluminium frame instead of the original lawn mower body provides more...
flexibility for the integration of the arm, sensors and supports needed to stabilise the vehicle during trimming. A first prototype of this vehicle including the lawn mover drive unit, the aluminium frame, the batteries and the supporters is shown in Figure ??.

A CAD model of the final design of the vehicle including also the arm and the trimming tool is shown in Figure ??.

In the next steps the electronics have to be integrated into the prototype and the control of the supporters has to be improved.

![Figure 14: Design for final demonstrator 3.](image)

Garden Reference Point Clouds

The Leica ScanStation P15 was used to record a high quality 3D point cloud from the entire test gardens at Wageningen and Renningen. The ScanStation combines distance measurements from a laser scanner and with the corresponding RGB values from a colour camera. The maximum measurement distance for the laser scanner is 40 m with an accuracy of 3 mm at 40 m. However, the range and the RGB measurements are performed sequentially thus the measurements are not synchronised in time. For moving objects this can lead to wrong RGB values: for example if a leaf of a tree moves between the distance and the colour measurement, then it gets the blue colour from the sky because the camera sees the blue sky at the point where the leaf was before. This effect can be seen on the trees of the point cloud from the Wageningen test garden.

The point cloud from the test garden in Wageningen is shown in Figure ??.

This point cloud consists of 280 million points and has a size of about 12 GB as a text file. Due to the measurement range of the Leica scan station, objects outside the test garden are also included in this point cloud. To reduce the amount of data a second point cloud which is cropped to an area close to the test garden is also provided. However, this cropped point cloud has still 235 million points and a size of 10 GB as text file. To further reduce the amount of data, the point clouds were downsampled and compressed to a binary file format. The 3D voxel grid filter from the Point Cloud Library (PCL) was used for the downsampling step. This filter uses a 3D filtering object and approximates all the included points by their centroid. Thus, the size of the downsampled point cloud depends on the size of the filter object. For the Wageningen
test garden a filter objects of 0.01 m is used. Downsampling the point cloud with a 0.01 m filter object leads to a point cloud with 16 million points and a file size of 254 Mb for the binary PCL format. A detailed list of all point cloud files from the Wageningen test garden is shown in Appendix of D1.1.

![Figure 15: Detailed 3D reference point cloud with included colour data from the test garden in Wageningen. The point cloud was recorded with a Leica ScanStation P15 during the first ground truth data collection session.](image)

**Navigation Software**

The navigation software package includes navigation planning, navigation execution and a vehicle driver. The inputs to the navigation software come from the Semantic SLAM package and the Garden User Interface (GardenUI). The Semantic SLAM package provides for the position of the vehicle w.r.t. the map. The inputs from the GardenUI are the registered map as an occupancy grid and the goals to where the vehicle has to navigate. A goal can either be a position in the garden or an object that has to be trimmed e.g. a bush, a hedge or a rose. If the goal for the navigation is a trimming object, the navigation planning has to calculate a position in front of this object and an approaching manoeuvre to reach the best trimming position for this object. If the goal is a position and not a trimming object only a collision free path to this position is planned by the navigation planning. However, for each user defined goal the navigate planning checks the position by taking into account the traversable areas in the map and if the position is not reachable an alternative position is calculated by the navigation planning component. This is needed to avoid goal positions which are inside an obstacle or too close to an obstacle and therefore not reachable by the vehicle.

In the first navigation version the ROS move_base component is used for the navigation execution. This component navigates the vehicle to a given destination. The destination is provided by the navigation planning. Based on the goal position calculated by the navigation planning, move_base calculates a collision free path and navigates the vehicle along this path. The vehicle driver is used to send the velocity commands from move_base to the vehicle itself. This node also provides odometry and IMU data from the vehicle which can be used by
other components like the Semantic SLAM package.
3.2 WP2: Clipper & Control

The objectives of WP2 ‘Manipulator Construction and Control’ are: Design and construct a manipulator and tools; Design and implement a motion planning program for the manipulator; Design and implement a navigation and trimming servoing mechanism. In the period reported here, work was conducted in two tasks: T2.1 Arm/Clipper Mechatronics and T2.2 Arm/Clipper Motion Planning. The task on servoed navigation and trimming will start in the 2nd period of the project. In the following the progress of the WP is given grouped by the individual tasks of the work package.

T2.1 Arm/Clipper Mechatronics (M1-M18)

In agreement with the project plan, WP2 defined the requirements for the robotic manipulator that will be placed on the mobile TrimBot2020 vehicle. After that, an extensive market research on available manipulators was conducted. Most industrial type robotic arms were not suitable due to their high weight. For candidate manipulators reachability studies were carried out, including software simulations. The arm selected and acquired for the project is the Kinova Jaco2 (Kinova Robotics, Canada). This robotic arm is developed primarily for persons with disabilities. Its six degrees of freedom allow for the dexterous movement that is required to perform the trimming task. The links are made from carbon fibre and due to that the weight is only 4.4kg. The maximum manipulator payload is 2.2kg, the positioning repeatability is 3mm and the positioning accuracy is 8 mm. Due to the planned use of active perception and visual servo control this rather low accuracy was not considered to be a major drawback. A ROS driver to control the arm was readily available.

A mobile test rig with a robot manipulator and laptop mount and accessories storage bins was designed and realised (Figure ??). This test rig allows to securely position the setup to a stationary position in either a laboratory setting or in the open field.

For the design of the first version of the trimming and cutting tools a structured design method was applied. With this method it was determined which functions and components were needed, and that the design of the hardware was in agreement with the overall design of sensors, software and other hardware components. For the two tasks needed, namely boxwood/hedge trimming and single branch cutting (roses), two different type of tools were designed and developed.

For the boxwood trimming a configuration of circular counter-rotating saw blades was chosen to ensure a high success rate to cut small branches in one go. Figure ?? show drawings of this design. The knives are driven by means of a flat electrical servo motor (Maxon Motor, Switzerland) with gearbox. The motor controller allows variable setting of the rotational speed by software commands. Furthermore the servo control allows to monitor and control the motor current which might be important to set the right rotational knife speed in relation to the speed of the manipulator. The constructed end-effector is displayed in the left photo of Figure ??.

For the single branch cutting a commercially available Bosch Ciso electrical pruner was chosen to be modified in such a way that it can be mounted on the manipulator. The main features of this device are that it can cut through 14 mm of softwood and it has a sensor built in to sense when the knife is completely open or closed. To monitor the drive system the original DC motor was replaced by a servo controlled DC motor. 3D printed plastic enclosures were
Figure 16: Mobile test rig for manipulator. CAD drawing (left picture) and realised (right picture). For the design of first version of the trimming and cutting tools a structured design method was applied.

produced to surround the components of the tool. The total weight of the tool is 0.5 kg. Figure ?? (right) shows a photo of the parts of the single branch cutting tool.

Both tools are designed to be easily attached to the last joint of the robotic arm. Further details on the mechatronic design of the robot and the tools are described in D2.1.

In order to test the tools and later the integrated robot a test garden was realised at the location of Wageningen University & Research Centre. This garden has a collection of boxwood plants (spherical and cube shaped), a boxwood hedge, an ivy hedge and a number of individual rose plants. Moreover, different types of terrain (flat, and elevated) and different types of floor (lawn, pebble stones, wood chips, pavement) are present. Figure ?? shows a sketch map of the test garden in Wageningen and a recent photo of the garden.

With both type of tools, initial operational tests have successfully been carried out in the laboratory and also in the test garden (Figure ??). For these first tests, the movement of the robotic arm was manually controlled by means of a remote control.

D2.1 was completed during the first period and details the arm and platform integration concept. This concept is separated in three parts: mechanical integration (including arm and sensor mounting), electrical integration (incl. emergency stop concept) and software integration (using ROS). Figure ?? shows a drawing and photo of the platform integrated with the robotic arm.

T2.2 Arm/Clipper Motion Planning (M10-M36)

The design of the motion planning software builds upon a review of the literature about coverage-aware motion planning and vision-based replanning that is part of D2.1. Differently from classical coverage path planning algorithms, the algorithm being developed will focus not be just on where the robot moves, but also on how it moves. The motion planning objective of
the bush trimming operation is to calculate a time sequence of arm configurations such that the error between the final bush surface and the target bush surface is below a certain threshold. The plan will need to consist of cutting and non-cutting motions. Figure ?? shows the bush trimming framework overview. The reference trajectory will be generated using MATLAB, the reference trajectory will be then forwarded to V-REP through Simulink/ROS. Simulations can be carried out using V-REP and finally the motion trajectory can be sent to the controller and executed by the robotic arm.

Initial algorithms for fitting spherical shapes to plant data have been created. Operational tests on real plants have successfully been carried out, including planning and cutting of spherical bushes. The planning setup module processes the input bush target shape to generate the search space for the actual planning task. The coverage module produces a list of intermediate joint configurations allowing to achieve coverage of the cutting area with the tool. Figure ?? shows an example of a computed tool coverage plan. It is worth to remark that in that example (stationary setup and a target bush diameter of 40 cm) only 1/4 of the bush surface can be
actually swept by the trimming tool. In the future, the vehicle will need to reposition a number
of times around the plant in order to perform full trimming of the bush.

Furthermore, as agreed on within the consortium, a ROS based FlexBE\footnote{http://wiki.ros.org/flexbe} state machine was
designed and implemented for the trimming actions of the arm. This state machine module can be
easily integrated as sub-state machine in the overall FlexBE state machine of the robot in the
future.

Deliverable 2.2 (Manipulator and tools version 1, running open-loop motion planning) was
submitted. In this deliverable the successful integration of the vision pipeline for cameras on
the arm (partner ALUF) and the motion planning pipeline (partner WU) was shown.

This stationary robot demonstration serves as a stepping stone towards TrimBot2020’s mo-
bile trimming robot prototype.

Being a demonstrator type of deliverable, D2.2 consists of the following parts: a video clip
showing the robot in action; a spherical boxwood bush; a test rig that can be moved manually;
a robot arm; a stereo camera; a custom end-effector; computing hardware; a vision algorithm
with implementation; a motion planning algorithm with implementation and a motion controller
with implementation. Figure 19 shows two screenshots from the D2.2 video clip. The video of
the first experiments can be seen here (GIT password protected): https://gitlab.inf.ed.ac.uk/TrimBot2020/General/tree/master/deliverables/D2.2/video/trimbot_d2-2.mp4

A trimming result evaluation system (size, symmetry, smoothness) is currently under devel-
opment at Wageningen University & Research Centre.

\section*{T2.3 Servoed Navigation and Trimming (18-48)}

This task will start in the second period of the project. For bush/hedge trimming the vision-
based replanning system will be built on top of the open-loop trimming planning module. It
will need to be researched how much visual servoing actions are possible for the bush trimming
action as the trimming tool will operate at very close range and the bush will move due to the
cutting action. Strategies like performing a movement and then looking back to adapt future
trajectories will be further investigated. For the individual branch cutting actions (roses) visual
Figure 20: First boxwood trimming experiments (left image) and first rose branch cutting experiments (right image).

servoing will be applied to adapt orientation and position of the tool according to the sensed error while approaching the cutting point.
Figure 21: Mechanical integration concept of platform and arm (left) and photo of first stability tests with real hardware in May 2017 (right).

Figure 23: Bush trimming framework overview.

Figure 22: Bush trimming framework overview
Figure 23: Example of a computed tool coverage plan (left). Example of computed tool coverage plan, visualized on top of the target bush boundary (right)

Figure 24: Two screenshots of the video of D2.2 showing an open loop bush trimming action.
3.3 WP3: 3D Acquisition and Fusion

The following details the activities undertaken as part of our work on WP3, ordered by topic. In addition, we describe the goals achieved during our work, as well as ongoing research and planned activities.

3.3.1 Multi-Camera System and Calibration

In order to allow the robot to freely and safely move through the garden, it is important that the full 360° horizontal field-of-view (FOV) around the robot is observable. Thus, we decided to use a multi-camera system for the TrimBot2020 project. This system is mounted on the base of the robot and is used for SLAM, re-localization, 3D reconstruction, and to capture the images used by WP4.

Design of the camera system. We designed the layout of the camera system under a set of constraints: In order to be able to obtain accurate depth estimates at all times, there needs to be sufficient visual overlap between adjacent cameras in the multi-camera rig to facilitate stereo computations. This overlap is also important for self-calibration during SLAM as insufficient overlap results in failure of the calibration process. The full 360° horizontal FOV can be created using either few cameras with a large FOV or several cameras with smaller FOV. Cameras with large FOV are subject to heavy distortion effects resulting in a non-uniform resolution. Additionally, the neural networks used for semantic scene parsing in WP4 are typically trained on images with little radial distortion, limiting the maximum FOV that can be used. We analyzed multiple configurations both theoretically and experimentally. For the latter, we used a state-of-the-art incremental Structure-from-Motion pipeline. In these experiments, we determined which camera configurations led to stable 3D reconstructions.

Fig. ?? (left) shows the rig configuration finally agreed upon all partners using the cameras. It consists of 10 cameras, arranged into 5 stereo pairs in a pentagon shape. There is an overlap in the field of view of neighbouring cameras as the FOV is 82° for the individual cameras while the pentagon shape has only 72° corners. The bandwidth consumed by the 10 cameras, providing images at a resolution of 752 × 480 pixels at 12 frames per second, is 330.46 Mbit/s.

In order to synchronize the image capture between all cameras we use a FPGA. In addition, the FPGA also computes a depth map for one camera in each stereo pair by running a stereo algorithm onboard. The depth maps are also provided at 12 frames per second. The FPGA is based on previous work at ETHZ. However, modifications were necessary: We added Bayer and infra-red filters onto some of the cameras to record colour images, which are required for the semantic scene analysis in WP4, and adapted the stereo algorithm to still use the raw pixel intensities (as it operates on grayscale images). The Bayer filter allows to generate a colour image but also blurs the image as the colour channels are not captured at full resolution. Sensors with a Bayer filter are slightly less light sensitive compared to an image sensor without a Bayer filter. With a combination of colour and greyscale cameras in a stereo pair the advantages of both versions can be exploited. In addition, we extended the image undistortion model implemented in the FPGA to handle the sensors used in the TrimBot2020 project.

Sensor calibration. There are two approaches to calibrate the multi-camera system: The first is an offline procedure based on an Aprilgrid target (cf. Fig. ?? (middle)). The robot, while
Figure 25: (left) The configuration of the multi-camera rig used in the TrimBot2020 project. (middle) The Aprilgrid target used to calibrate the intrinsic and extrinsic parameters of the multi-camera system. (right) The calibration target, mounted at a known position, used to compute the pose of the camera system with respect to the base frame of the robot.

standing still, records images of the target that is carried around it manually. The images are then fed into the Kalibr toolbox to compute the intrinsic parameters of each camera and the extrinsic parameters between the cameras. The pose of camera rig relative to the base of the robot is then obtained from an image of a pattern fixed to a known measured location on the base frame (cf. Fig. ??(right)).

As can be seen in Fig. ?? (middle), the target used for calibration is rather large, making the calibration process cumbersome. Unfortunately, a large target is necessary to ensure that a large enough part of the pattern is seen by adjacent cameras from different stereo pairs. In order to automate this process, we developed an online calibration approach that is part of the SLAM system. More details about the online calibration procedure, which estimates the extrinsic parameters of the multi-camera rig, are provided in Sec. ??.

Details on the calibration procedures can also be found in deliverable D3.1.

**Status and planned activities.** We are currently contemplating to include estimating the intrinsic parameters of the cameras into the online calibration procedure. Besides this point, we consider this part of WP3 completed.

### 3.3.2 3D Data Fusion

Fusing data from different sensors, e.g., binocular and motion stereo, to obtain an accurate 3D model of the garden is a central aspect of WP3. The main activity undertaken so far in this part of WP3 is the definition and implementation of the data structures used for 3D data fusion.

The fusion is performed by the 3D data processing package that is being developed by UEDIN and ALUF. It fuses data from the SLAM system, binocular and motion stereo, and scene flow into a single volumetric representation while taking the uncertainties of each component into account. Based on the volumetric data, a 3D surface mesh can then be extracted.

The SLAM system provides a sparse point cloud of the garden, where each point is associated with the local image features it was triangulated from. The binocular and motion stereo systems and the scene flow component provide depth maps in the form of dense point clouds together with functionality to estimate the covariances of their measurements. The data processing package then generates a voxel volume describing the scene. Each voxel contains position, occupancy, and colour information together with covariance estimates, obtained during the fusion process, for each quantity.
Figure 26: Depth maps computed by (left) the FPGA, (middle) binocular stereo, (right) trinocular stereo.

Figure 27: A schematic overview of the SLAM pipeline developed in WP3.

As the individual components providing data are developed by different partners and since these individual components need to interact with each other, it was decided that each partner implements their component as a ROS (Robot Operating System) package. These packages communicate with each other by sending messages that contain the data they provide. Thus, the data structures used for 3D data fusion are implemented as ROS messages. An advantage of this architecture is that it easily allows us to link the geometric information generated in WP3 with the semantic information extracted in WP4, e.g., for semantic SLAM, simply by including one message in another.

A more detailed description of the functionality of the individual components and the messages that they exchange can be found in deliverable D3.1.

Status and planned activities. The design of the data structures used for 3D data fusion has been completed and the data structures are implemented by the partners. An extensive review of existing work about estimating the uncertainty for different sensors and using these estimates for data fusion has been completed. Based on this review, a data fusion algorithm has been designed that is currently being implemented.

As part of WP3, algorithms for binocular and binocular + motion stereo have been developed by UEDIN. As shown in Fig. ??, these algorithms are able to provide more accurate depth maps compared to the FPGA stereo algorithm. The latter runs on a platform with more restricted computational resources while the former run on a standard PC. The computational resource of the robot’s PC need to be shared between all partners. Thus, UEDIN and ETHZ are planning to explore using the coarser depth maps estimated in real-time by the FPGA as initialization for the more costly, but also more precise, stereo algorithms from UEDIN.

With the current progress and the ongoing activities, this part of WP3 is well on track.
3.3.3 (Semantic) SLAM & Re-Localization

The SLAM system employed in the TrimBot2020 project is used to simultaneously estimate a map of the garden and determine the position of the robot with respect to the map. Since most existing SLAM approaches either use only a monocular camera or a binocular stereo camera, and are thus not able to take full advantage of our multi-camera system, we developed a multi-camera SLAM approach. Rather than handling each image captured by the multi-camera rig individually, our SLAM method models the rig as a generalized camera, i.e., a camera with multiple centers of projection. This enforces that all cameras of the rig move consistently.

An overview of the developed SLAM system is provided in Fig. 28. Our approach follows a classical feature-based approach, where local image features are used to establish correspondences between images captured at different time steps. Image features from previous time steps correspond to 3D points in the SLAM map, enabling us to estimate the pose of the generalized camera with respect to the SLAM map. Finally, the map is extended by triangulating new 3D points and performing a local refinement via bundle adjustment. Periodically, loop closure is performed to reduce drift and a larger part of the map is refined. A map of the test garden in Wageningen, generated by our SLAM system, is shown in Fig. 28. More details on the SLAM system and its evaluation will be provided in deliverable D3.2 (due in month 30).

The main novelty of our SLAM system is that it integrates an online calibration procedure. This process is able to calibrate the extrinsic parameters of the multi-camera system using images of the garden captured by the robot. The calibration procedure is based on a novel variant of global Structure-from-Motion. Global Structure-from-Motion approaches estimate the poses of all cameras simultaneously rather than incrementally. We adapt the typical global Structure-from-Motion pipeline, consisting of rotation and translation averaging followed by bundle adjustment, to incorporate the constraint that the cameras in the rig should move consistently over time. The self-calibration procedure iteratively captures data. Once it determines that it has obtained enough constraints, it attempts self-calibration. After some time, this calibration is then later on refined using the data captured in between. Our experiments show that our approach is able to estimate the extrinsic calibration of the rig from a random initialization. A schematic representation of the self-calibration pipeline is given in Fig. 28. More details on the self-calibration procedure can be found in deliverable D3.1.

Figure 28: A schematic overview over the novel self-calibration procedure integrated into SLAM.
Status and planned activities. The current version of the SLAM system has been integrated with the path planning system developed by BOSCH. The SLAM system achieves an average frame rate of about 5 frames per second (which is already sufficient given the travel speed of the robot). Still, we are working on improving its run-time performance. We are planning to evaluate our SLAM approach on more data recorded in the test gardens. The SLAM system was developed as part of a Master Thesis. We plan to submit a paper with a focus on the self-calibration part to the International Conference on Robotics and Automation (ICRA) 2018.

There is an ongoing Master thesis at ETHZ on integrating semantics into the SLAM process. The semantic SLAM system proposed in the thesis uses semantic image segmentations to assign each 3D map point a semantic label. This semantic information is then used during the optimization of the camera poses and the 3D map to reduce drift. We are planning to use the resulting SLAM system in the TrimBot2020 project as part of Task 3.5 (“Semantic SLAM”).

There is ongoing work at ETHZ on incorporating semantic information into re-localization against pre-built semantic 3D maps. The core idea of the approach is to combine 3D information from depth maps with semantic labels extracted from images to obtain feature descriptors that are robust to strong changes in viewing conditions. Current results demonstrate that our novel descriptors can be used to re-localize under strong viewpoint changes. We are planning to extend this approach to enable re-localization against maps that contain both appearance and geometric changes, e.g., due to cutting operations or seasonal changes. To the best of our knowledge, this is an unsolved problem. In order to obtain data for this scenario, a student at ETHZ is currently collecting data with the TrimBot2020 camera system on a weekly basis. We plan to incorporate the resulting approach into the project as part of Tasks 3.5 and 3.7 (“Arm and Vehicle Localisation”).

There have been exploratory talks between ALUF and ETHZ to incorporate the DeMoN approach from ALUF, which jointly estimates depth and camera motion via a neural network, into a novel dense SLAM system. Both parties plan to explore this topic further.

With the current progress and the ongoing activities, this part of WP3 is well on track.
3.4 WP4: 3D Data Analysis and Scene Understanding

Workpackage 4 (WP4) approaches the scene analysis problem in a data-driven fashion. The final goal is to semantically understand a scene captured by the robot’s camera system using deep learning.

3.5 Creating synthetic dataset for garden-specialized training

Because of the absence of large-scale datasets annotated with garden-specialized semantics, which requires a good amount of man-hour effort but play a key role in deep learning algorithms, we decided to get started with a synthetic design. From computer graphics softwares, we construct CAD garden models with various arrangement of objects and lighting settings. The models can be rendered into photo-realistic images (as shown in Figure ?? (left)) as well as other data type such as depth, reflectance, shading (Figure ?? (middle)), and pixel-wise object labelling (Figure ??(right)). Targeting large scale datasets, the CAD models, which were first designed manually, are studied and adapted so that they can be constructed and thus generated autonomously.

![Figure 29: Sample scenes from synthetic garden dataset](image)

3.6 Developing a deep network for intrinsic image decomposition

Intrinsic image decomposition is an ill-posed and under-constrained problem, as given a pixel value of the observed image, there are multiple unknowns and multiple solutions. Thus, most of the early work relies on deriving priors about scene characteristics and imposing constraints on reflectance and shading intrinsics by formulating an optimization routine for pixel-wise decomposition. As intrinsic image datasets emerge, more recent research has moved towards learning approaches. Nevertheless, current datasets are restricted in a sense that they either have object-centric images, and thus do not comprise scene-level intrinsic image decomposition or they do not contain enough data. To that end, the scene-level large-scale synthetic garden dataset we created is also beneficial for our data-driven intrinsic image decomposition, as intrinsic ground truth images can be extracted instantly with semantic labels. By using the dataset, we propose a novel architecture, shown in Figure ??, to solve the intrinsic image decomposition problem in a purely data-drive manner. Given the fact that large derivatives in an image correspond to reflectance changes, whereas small ones correspond to illumination changes, the model exploits this gradient information in addressing intrinsic image decomposition. Some of the initial decomposition results are shown in Figure ??.

Furthermore, initial evaluation results...
are shown in Table ???. Note that the error metric is Local Mean Squared Error (LMSE), the lower the better.

Table 1: Evaluation results on the test set

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline method</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>0.209</td>
<td>0.080</td>
</tr>
</tbody>
</table>

3.7 Developing a deep network for semantic segmentation

The current dataset, with around 30k images, enables us to train and test a baseline semantic segmentation system. We choose the SegNet semantic segmentation architecture Figure ?? as it is originally designed for autonomous driving purposes. The network can learn the general scene structure through the encoder layers (first half) before going into each detailed region in the decoder layers (second half).

In Figure ??(left), we show the baseline results in 3 measurements: global accuracy, local or per-class accuracy, and mean intersection-over-union (IOU). The presence of a large gap in accuracy between sunset images (with dark and dim appearance) and the other scenes (with bright appearance) shows the impact of lighting conditions on the system performance. The impact is also shown in Figure ??(right), which compares the accuracy between systems using RGB image and one using reflectance images, where all the lighting effects are removed. The good and a failure case of the system is shown in Figure ??.
Figure 31: Baseline intrinsic image decomposition results

Figure 32: SegNet architecture
Figure 33: Baseline results using synthetic images

Figure 34: A good (left) and a failure (right) segmentation
3.8 WP5: 3D for Motion Planning and Visual Servoing

We worked towards a scene flow (optical flow and depth) solution that runs at interactive frame rates and can handle well the special case of plants with lots of self-occlusion, thin objects, and special reflection properties. Many design decisions have been made together with the partners, such as the choice of the camera and the coupling of depth sensing, visual servoing and control.

Improving the deep architectures for disparity and optical flow estimation

FlowNet and DispNet are deep network architectures that can learn optical flow and disparity estimation end-to-end from a sufficiently large set of training data (synthetic, rendered data). These approaches were developed by the ALUF team between submission of the project proposal and the start of the project. An illustration of the network architecture of DispNet is in Fig. ??.

![Figure 35: DispNet deep net architecture](image)

Directly after the project started, we worked towards improving this framework to match the goals of the project. In particular, we investigated the relationship between shortcomings of the networks and properties of the training data. We developed specialized datasets that improved the network performance on small displacements, where classical variational approaches excelled. We also found learning schedules using multiple datasets with different properties sequentially to improve the performance of FlowNet drastically. Moreover, we developed a stacking strategy that runs networks on top of the output of a previous network to learn refinement of the previous solution. This finally led to the FlowNet 2.0 architecture, shown in Fig. ??.

FlowNet 2.0 reached state-of-the-art optical flow accuracy while being 2-3 orders of magnitude faster than previous state-of-the-art methods. This makes it an excellent technology to be used in a robotics setup, where both quality and interactive frame rates are necessary.

Developing a deep network for depth and camera motion estimation from two views

Based on the success of FlowNet, which formulates optical flow estimation completely as a learning problem, with large advantages for the accuracy/runtime ratio, we tried to formulate also structure from motion as an end-to-end learning problem. The input is two images from a monocular video and the desired output is the depth map and the egomotion of the camera. This building block is a decisive component in all SLAM and large-scale structure from motion approaches. Formulating it as a learning problem is more difficult than formulation of optical flow estimation, because the depth map estimation and the egomotion estimation are mutually dependent, but while the depth map requires an encoder-decoder network architecture to regress
a high-resolution output, the egomotion parameters are low-dimensional and must be inferred with a fully-connected network. Consequently, it took us a rather long time to figure out an architecture that learns this task. The architecture shown in Fig. ?? is an obvious choice building upon previous experience.

However, we found that it does not use the motion parallax between the two input images. In fact, making one of the two images black, the results were as good (or bad) as with two images as input. This shows, that the network fell back to learning only depth from a single image, an approach published earlier by Eigen et al. Since, depth from a single image is ambiguous and highly dependent on the test data fitting to the training data, it should be inferior to two-frame structure from motion, where the motion parallax provides for most points in the image real measurements to decide on the depth. We tried multiple ways to improve the learning formulation and network architecture, building on the FlowNet, from which we already know that it estimates correspondences between two images. In the end, the research led to a working solution based on stacking networks (as in FlowNet 2.0) and an additional recurrent (iterative)

Figure 36: FlowNet deep net architecture

Figure 37: Depth and camera motion deep net architecture
component to refine the outputs without the need for learning new weights for each iteration, as seen in Fig. ??

![Stacked and recurrent net architecture](image)

**Figure 38: Stacked and recurrent net architecture**

Of major importance is the stacking of a FlowNet and a successive encoder-decoder network for depth and egomotion estimation, as illustrated in Fig. ??.

![Integrated FlowNet and encoder-decoder network architecture](image)

**Figure 39: Integrated FlowNet and encoder-decoder network architecture**

Only by first forcing the network to compute optical flow, thereby using the two input images, and then deriving the depth map from these estimates and the input images, we succeeded making good use of the motion parallax. The difference to depth from a single image is clearly visible in scenes that were sufficiently different from scenes in the training data. An example is shown in Fig. ??.

![Example comparing depth from one and two images](image)

**Figure 40: Example comparing depth from one and two images.**
Creating training data specialized for the camera used in TrimBot2020

All of the above networks were trained on data that fitted a certain camera model: the camera model of the Sintel benchmark in case of FlowNet and DispNet and the camera model of a Kinect in case of DeMoN. In the TrimBot2020 project, a special camera will be used. Thus, we investigated on how the correct modeling of the camera in the training data affects the results. The result below, where DispNet was trained with the Sintel camera model and the Bumblebee Stereo camera model, respectively, show that correct modeling of the camera during training, particularly lens distortion and blur, improves results considerably, especially in the areas of the image most affected by the distortion of the camera.

Figure 41: Example showing the effect of using different camera models.

Meanwhile, the same modeling has been done for the camera used in TrimBot2020, i.e., the networks can be trained on data that is best suited for the camera in use.

Creating specialized synthetic data for training large networks on garden scenes

The advantage of the above learning based approaches is that they can be adapted and specialized easily to specific application scenarios, such as scenes from a car or scenes in a garden. However, this adaptation requires training data that reflects the properties of the application area. In case of the TrimBot2020 project, the scenes will largely consist of plants, which come with special challenges for depth estimation. Within this project period we already started to plan and set up tools to create and render plant and garden scenes for large scale training with a deep network. While it is easy to manually set up scenes and rendering them, creating a large scale training dataset with sufficient diversity this way is very tedious. Therefore, we plan to set up a procedural method that generates realistic garden models randomly and in arbitrary amounts automatically. This work is not yet finished and will be continued in the next project period.

Implementing a ROS node for fitting a spherical bush from a stereo view

For allowing the WU team to make progress with the development of the controller for the trimming arm, we developed a ROS node that runs the DispNet to estimate the depth of the scene and then fits a sphere into a spherical bush. This sphere is used in Workpackage 2 to build a first open loop controller for trimming spherical bushes. The work will be extended to using stereo pairs from multiple viewpoints in order to fit also more complex shapes.
3.9 WP6: Platform integration

Software Architecture

One task in WP6 is the definition of the software architecture. A basic architecture was therefore defined and documented at the beginning of the project. Due to the continuous integration process this software architecture will be adapted during the project. The defined software architecture is based on the computational components listed in the System Requirement Document D7.1. For the integration of all these components into one system the ROS (Robot Operating System) framework is used. ROS provides a communication interface for interprocess communication, robot-specific features and several tools. In ROS each process that performs a computation is called node. These nodes are combined together into a graph using streaming topics for the communication between nodes. A topic is a named bus over which a message is exchanged. These topics and messages are the interfaces between the computational components defined in D7.1. Additionally to the ROS standard message several custom messages are defined and used in this project. A list of all custom messages and their detailed definition is maintained in the project wiki (General/wikis/msg). For the software realisation with ROS each of the components has to be implemented as at least one ROS node or a wrapper class to ROS.

For a better software organisation all components which belong to the same task are grouped together to packages (http://wiki.ros.org/Packages) or meta packages (http://wiki.ros.org/Metapackages). Most of these packages are also related to one project partner. Thus the interfaces between these packages are also the interfaces between the project partners. The TrimBot2020 software is organised in seven ROS packages:

- Camera (ETHZ)
- Semantic SLAM (ETHZ)
- 3D Data Processing (UEDIN/ALUF)
- Garden Object Detection (UVA)
- Sketch Map (UEDIN)
- Navigation (BOSCH)
- Trimming (WU/WR)

A description of these software packages and the interfaces (topics and message types) between these packages are defined in the software architecture document [https://gitlab.inf.ed.ac.uk/TrimBot2020/General/tree/master/architecture/software](https://gitlab.inf.ed.ac.uk/TrimBot2020/General/tree/master/architecture/software) in order to clarify the interfaces and the data flow between the individual packages, an overview plan was created and also added to software architecture document. Due to the continuous integration strategy this document is a living document and will be update during the project. Each integration step is planned based on this document. An example of how a software architecture is derived for an integration step from this document is the software architecture used for the first navigation integration. The overview plan defining the interfaces and the data flow for this integration step is shown in Figure ???. This integration is needed for the first demo
which is about navigation. The software includes therefore only four packages Camera, SLAM, Sketch Map and Navigation.

To enable online operation for the demonstrators a network with a Wifi router is integrated into the vehicle. The configuration with the Wifi route on the vehicle allows to have the same network configuration anywhere. Thus, a platform can either be used in the test garden in Wageningen or in the test garden in Renningen without changing the network configuration. An embedded PC and an onboard laptop can be connected via Ethernet to the router and a remote Laptop can be connected via Wifi. The Network Time Protocol (NTP) is used to synchronise all computers in this network to achieve synchronised data.

Garden Task Management

The garden task management includes two parts: 1) the Garden User Interface called GardenUI and 2) the master state machine which implements the behaviour of the robot. The GardenUI allows an user to specify a 2.5D semantic sketch map of a garden (Figure ??). The sketch map has two main parts, the terrain represented by a mesh surface and a set of objects represented with primitive shapes (cube, sphere, etc.). Every segment of the terrain and every object has attached a semantic label (grass, hedge, topiary etc.). See D7.4 - Ground-truth data definitions and acquisition for complete list of semantic labels and their colour coding.
Figure 42: Overview of the software architecture for demonstrator 1.
Figure 43: Garden User Interface. Magenta line: sequence of objects selected for trimming. Blue markers: control points of the surface mesh.

The sketch map must be initially registered to the SLAM map in order to issue correct coordinates for navigation. Currently the registration transform is rigid because the sketch map is drawn with the help of metric measurements from the garden plan. A later version will allow less accurate initial sketch maps that will be subsequently refined by deformable registration with the sketch map.

The interface with a loaded and registered map can be used to issue commands to the robot. The robot, whose pose is also displayed in the map, can be sent to a specific location or a currently selected trimming object. The current action can be cancelled with a dedicated button. GardenUI also updates the occupancy grid from SLAM with semantic information, i.e. indicates terrain which is not driveable. Only surface tiles labelled as grass or pavement are considered safe to drive over.

The master state machine controls the behaviour of the robot. It is implemented with FlexBE\(^2\) which is based on SMACH\(^3\) and allows an easy modelling and implementation of state machines for ROS. FlexBE has some pre-implemented states, e.g. ROS message subscriber, logging states or calculation states and it also allows to interleave state machines. In TrimBot2020, this concept is used to keep the master state machine simple and therefore encapsulates the state machines for navigation and trimming. The first version of the master state machine is now completed and will be used for the first demo. Figure ?? shows this state machine.

The grey boxes are interleaved state machines and the yellow boxes are states. The arrows between these boxes are the transitions with their conditions. The master state machine consists of an Init state machine where the vehicle and all sensors are initialised. After a successful initialisation (pink transition) the master state machine circulates between two ROS message subscriber states (green transitions). Between the two subscriber states two additional Pause

---

\(^{2}\) [http://wiki.ros.org/flexbe]

\(^{3}\) [http://wiki.ros.org/smach]
Figure 44: Master State Machine: The pink transition is activated after a successful initialisation. The green transitions build a circle in which the state machine waits for new user goals. The orange transitions show the procedure if a message for a trimming object is received and the purple transition show the procedure if a message for a simple goal is received.

states are included. The first subscriber state is the Check for Trimming Object which checks if a gardenObject message is received and if such a message is received the orange transition to the Navigate to Object is activated and the navigation state machine will be entered. After a successful navigation the state machine will go to the Trimming state. In this version of the state machine the Trimming and Trimming Successful states are empty and only passed to go back to the wait for user goal circle (green transitions). The second subscriber state is the Check for Simple Goal which checks if a simpleGoal message is received. If this message is received the purple transition to the Navigate to Simple Goal is activated and this state machine will be entered. After a successful navigation the state machine will return to the wait for user goal circle (green transitions). A more detailed description of the state machines used for navigation is given in D1.2 Platform 2: Supports online operation for demonstrator 1. For the final version of the master state machine the Trimming and Trimming Successful states will be replaced by more complexed state machines and a recovery behaviour will also be integrated.

Conception and Integration Meetings

Several meetings were held to develop both the hardware and the software concept. For the definition of the software architecture, Skype meetings were held mainly between BOSCH and UEDIN. However, with other partners also Skype meetings and a brainstorm session during the consortium meeting in Freiburg were held to define the interfaces. Additional to the Skype meetings technical meetings have been carried out or are planned for:
• Arm selection and arm/platform integration concept  
  (16.02.2016, Renningen, (BOSCH, WU))

• Camera concept and camera/platform integration concept  
  (30.03.2016 Renningen, (BOSCH, ETHZ))

• Integration of camera ring and ground truth data recording  
  (19-23.9.2016 Wageningen (UEDIN, WU, ETHZ, BOSCH))

• Integration of five stereo camera pairs and data recording  
  (12-13.12.2016 Rennigen (BOSCH UEDIN, ETHZ))

• Integration of final pentagonal camera setup and data recording  
  (28-29.3.2017 Renningen,(BOSCH, ETHZ, UEDIN))

• Integration of Garden UI and Navigation  
  (7-8.6.2017 Renningen, (BOSCH, UEDIN))

• First integration of SLAM, Navigation and UI  
  (3-6.7.2017 Renningen, (BOSCH, ETHZ, UEDIN))

• Second integration of SLAM, Navigation and UI  
  (15-16.8.2017 Zürich, (BOSCH, ETHZ, UEDIN))

• Third integration of SLAM, Navigation and UI  
  (4-8.9.2017 Renningen, (BOSCH, ETHZ))
3.10 WP7: Evaluation + Demonstrator

System specification and evaluation

Initially we concretized the plans laid out in the proposal and identified functional components of the prototype system as shown in Fig. ?? . The individual components can be grouped into logical parts, starting with the environment observed by sensors that feed data into computational modules. The stream of visual information is processed to provide sufficient understanding of the scene to task planners and controllers. This is performed in two main modes: garden navigation or bush trimming, depending on the current task of the master controller, responsible for high-level decisions. Finally commands are issued to the robotic platform, which either starts driving in the garden or moves the arm with activated clipper to perform trimming.

All partners worked together to detail the required capabilities of each physical and computational component, while responsibilities for their development were also decided, as captured in D7.1 - System Requirements Document.

The above introduced requirements were further quantified in a subsequent communication leading to D7.3 - Component and System Evaluation Plan, where a range of component properties was specified. Most of them can be generally characterized as accuracy or error tolerances. The principal purpose of this effort was to make sure that combined tolerances of individual components will not exceed the maximum allowed error of the system as a whole. Simultaneously error measures and simple scenarios to obtain them were proposed, they will be later implemented for evaluation and testing during integration.

The dependencies between individual components shown in Fig. ?? established a basis for data interfaces in the software architecture designed in WP6.

Ground-truth data acquisition

We have captured 4 datasets consisting of sets of videos from the stereo sensors during different tasks. The prototype used for the capture was Platform 1 described in WP6 (Fig. ??).

The data for ground truth (GT) was captured in our two test gardens during several data collection meetings listed in Tab. ?? . The GT recording started in the second half of 2016, then significant changes to the camera setup have led to the second round of recording in 2017. As detailed in D1.1 - Sensor Data Collection including Ground Truth from WP1 the events usually followed a similar program:

- Day 1: garden scanning, platform setup, testing and calibration.
- Day 2: manual driving with the platform in the garden, data capture.
- Day 3: evaluation of collected data, additional capture.

The data captured from the moving vehicle included synchronized image streams from the camera rig, range data from the mounted laser scanner, robot position from the laser tracker and orientation from the IMU.

Semantic annotation of ground-truth data

A subset of the captured data has been manually annotated to indicate the semantic classes of the objects appearing both in the images (2D) and the geometry (3D). The time consuming task of
Figure 45: Scheme of TrimBot2020 system components and dependencies.

Figure 46: Platform 1 during GT data collection with octagon camera setup (8 x 1, left) and pentagon (5 x 2, right). Video: [https://youtu.be/aV7ainghMCA](https://youtu.be/aV7ainghMCA).

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Scenarios</th>
<th>Annotated</th>
<th>Camera setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2016</td>
<td>Wageningen, NL</td>
<td>18</td>
<td>0</td>
<td>8 x 1 mono+colour</td>
</tr>
<tr>
<td>December 2016</td>
<td>Renningen, DE</td>
<td>10</td>
<td>4</td>
<td>5 x 2 mono only</td>
</tr>
<tr>
<td>March 2017</td>
<td>Renningen, DE</td>
<td>14</td>
<td>0</td>
<td>5 x 2 mono+colour</td>
</tr>
<tr>
<td>May 2017</td>
<td>Wageningen, NL</td>
<td>18</td>
<td>2</td>
<td>5 x 2 mono+colour</td>
</tr>
</tbody>
</table>

Table 2: List of ground-truth data collection events in chronological order.
Figure 47: Overview of collected GT data in the two project gardens. Video: https://youtu.be/BtL2fuPVsCs
human annotation of image sequences was facilitated by projection of annotated 3D geometry (semantic point cloud) into images given the camera poses. To this end we have designed a workflow (Fig. ??) for which two applications with GUI were developed: a sketch map editor for production of a 3D semantic model, and a labeling tool for the 2D part (Fig. ??).

An example of the resulting semantic map is shown in Fig. ??). In total 742 image frames were annotated in this way, which required over two person-months of work. 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 organized by the project members.

**Demonstrators**

The early *D7.2 - Three Demonstrator Plan* was produced to show how the project will demonstrate that it has achieved its main goal: the development of a garden trimming robot. The plan lists required components to accomplish this task and it is continuously updated to match the changes in software and hardware architecture arising during the development, as well as the progress of integration in WP6.

Demonstrator 1 tests the capability of vehicle localisation and navigation in the garden. For this purpose Platform 2 has been developed in WP1. It is equipped with the camera system from WP3 and other essential components, which allow safe navigation to a specified location. It has been clarified that the task can rely on the information from the sketch map and SLAM, hence the visual servoing planned in D7.2 does not have to be included in order to reach the goal and the integration of related components (3D fusion and surfacing, scene understanding) was postponed.

The measures from D7.3 will be used to evaluate the real performance of Demonstrator 1 in D7.5 (Month 24).

---

Figure 48: Sample annotated image. The captured image (top left) and its semantic map (bottom left) with colour-coded semantic classes (right).
Figure 49: User interface of the developed sketch map editor (left) and the semantic annotation tool (right).

Figure 50: Semantic annotation workflow.
3.11 WP8: Dissemination and Exploitation

Workpackage 8 is devoted to dissemination activities. Dissemination activities are of great importance to the success of TrimBot2020 and all research efforts and results will be widely communicated to the scientific and relevant industrial communities and the general public.

Four deliverables have been released within the first 18 months of the project:

1. D8.1 - Dissemination plan (responsibility of RUG)
2. D8.2 - Data management plan (responsibility of UEDIN)
3. D8.3 - Website and social media presence (responsibility of RUG)
4. D8.4 - Report on relevant stakeholders (responsibility of RUG)

Workpackage Activities

In deliverable D8.1, a plan of dissemination activities was designed that includes five phases:

1. Preparatory phase, now over, in which the dissemination tools have been designed and integrated with each other
2. Scientific dissemination, to help spreading the impact of scientific innovations created in the project (at the initial steps)
3. Communication to general public, to raise awareness of science and robotics innovation among people (planned to be taken care of after the first demos)
4. Teaching/education, for the training of young researchers and development of teaching materials (in the process for all the partners)
5. Project exploitation, for BOSCH to identify market possibilities for the technology developed in the context of the project

In the following, activities related with the first three phases taken in the first 18 months of the project are reported.

Website Release

The TrimBot2020 website is the focal communication point between the project and the stakeholders, including general audience. The website is hosted on the servers of the University of Groningen at the URL [http://www.trimbot2020.org/](http://www.trimbot2020.org/) and is provided with an area for public dissemination of contents and a member area that is available only for consortium members. The private area hosts internal material and deliverables, allows for management of scientific publication list, etc. Figure ?? shows the homepage.

The activities that are constantly measured on the website, showed a consistent increase of the project’s web visibility. In Figure ??, a graph of the user visits on the TrimBot2020 website is reported, which indicates a substantial growth. The statistics created with Google Analytics indicate:

- Total users: 1900
Setup of Social Media Channels

Profiles on social media have been set up primarily to increase the visibility of the project and raise public awareness of the project and its added value for the general audience. The social media presence has been set up on the channels illustrated in the following, for which we report the actual statistics and dissemination achievements. You can see the front pages of our channels at:

- Twitter (Fig. ??): [https://twitter.com/trimbot2020](https://twitter.com/trimbot2020)
- YouTube (Fig. ??): [https://www.youtube.com/channel/UCbPCq-c_Gsamuyjgl81rWGA](https://www.youtube.com/channel/UCbPCq-c_Gsamuyjgl81rWGA)
- ResearchGate (Fig. ??): [https://www.researchgate.net/project/EU-TrimBot2020-Gardening-robotics](https://www.researchgate.net/project/EU-TrimBot2020-Gardening-robotics)
- Facebook (Fig. ??): [https://www.facebook.com/trimbot2020/](https://www.facebook.com/trimbot2020/)

Data Management Plan

A data management plan (Deliverable D8.2) was released on month 6. The data plan covers the types, formats and quantities of the data, the metadata, and relevant standards. It ensures reliable data analysis and RIA interpretation and will enable broad use of the data produced in the context of the project by the scientific community.
Figure 52: Twitter Front Page

Figure 53: YouTube Channel page

Figure 54: ResearchGate Channel Front Page
Analysis of Stakeholders

The content created by the TrimBot2020 consortium aims at reaching a wide audience distributed all over the world, with the objectives of stimulating the market of gardening robotics, increasing appreciation for science and engineering and promoting the scientific results and breakthroughs. Stakeholder engagement is an important process for the success of the project, as it allows to gain knowledge about needs and interests of the entities who interact with the project consortium.

The stakeholders (Deliverable 8.4) of the TrimBot2020 project are organized in target groups, so to help direct the communication and dissemination strategy. The organization in target groups also helps the management of the needs and interests of various stakeholders in an efficient way.

<table>
<thead>
<tr>
<th>Target Group</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>End users</td>
<td>garden owners, garden centres, parks, golf courses, farmers, etc.</td>
</tr>
<tr>
<td>Scientific community</td>
<td>Scientific community working in the areas of consumer robotics, gardening robots, outdoor robotics, 3D sensing, 3D scene understanding and Computer Vision and data fusion.</td>
</tr>
<tr>
<td>Industry</td>
<td>Big enterprises (Bosch, etc.), SMEs</td>
</tr>
<tr>
<td>Public (general)</td>
<td>People, media</td>
</tr>
</tbody>
</table>

According to this categorization, user groups were identified and lists of possible stakeholders were provided in the deliverable D8.4:

- Governmental bodies
- Industry
- Civic park gardeners, golf courses, large private estates, farmers and garden centres
- Media
- Academia/Research
- General public
4 Administrative Details

4.1 Status of Project Objectives

1. Autonomous garden robotic bush and rose trimming: This is the overall objective. We have some initial trimming experiments, and an ability to navigate autonomously to a user-specified garden location. The robot arm will be mounted on the vehicle in the next period.

2. Development of a low-cost physical robot able to do trimming: A prototype based on Bosch’s Indego lawn mower platform is being used so far, but will be revised when the arm is added.

3. Development of a computer controlled trimmer and cutter suitable for mounting as a robot end-effector: Prototype trimmer and cutter have been developed, and have had initial tests over a planned trimming of a spherical bush.

4. Development of algorithms to fuse multiple shape-from-X capabilities to create a reliable 3D scene shape description: This is in progress. We have prototype stereo, optical flow and shape from motion 3D data extraction processes, along with a prototype SLAM data fusion algorithm.

5. Development of algorithms to acquire an accurate outdoor 3D scene map and description: There is a prototype SLAM data fusion algorithm, and prototype scene pixel semantic labelling algorithm.

6. Development of algorithms for navigation in a restricted and cluttered outdoor environment with changeable illumination conditions: We have developed prototype navigate to locations and navigate to sketch map objects capability.

7. Development of algorithms for relating visual SLAM-based 3D scene maps to user-supplied sketch maps of the garden and ideal topiary shape: We have developed the user sketch map construction interface and acquired SLAM-based 3D garden data. We are still developing the deformable registration algorithms.

8. Development of algorithms for overgrown bush analysis: Preliminary 3D bush shape fitting work is in progress.

9. Development of algorithms for 3D visual servoed bush trimming: This has not started.

10. Evaluation of the complete integrated robot: We have collected and ground-truthed garden data for evaluation of the robot performance so far.

11. Training of junior research staff in robotics and 3D vision skills: We estimate 24 person-years of pre- and post-doctoral researcher project time has contributed to their training.

Milestones status
The project website (M1) can be seen at [http://www.trimbot2020.org](http://www.trimbot2020.org). Videos showing the working platform driving (manual) in the Wageningen test garden can be seen at [https://www.youtube.com/watch?v=PCQDBM4uki8](https://www.youtube.com/watch?v=PCQDBM4uki8) and [https://www.youtube.com/watch?v=uxf7evLiHNw](https://www.youtube.com/watch?v=uxf7evLiHNw). There will be a live vehicle navigation demonstration at the project review meeting (weather permitting) or a video otherwise.

### 4.2 Workplan Status

The Workplan Status is: proceeding according to plan. The red vertical line shows the current time (month 21). All due Deliverables and Milestones have been met.
4.3 Risks Status

Three potential project risks identified in the proposal have been encountered and overcome:

1. **Unstable vehicle**: a redesigned platform with controllable stabilisers has been designed.

2. **Inaccurate vehicle servo to plant locations**: a redesigned pentagonal sensor ring was designed, replacing the body-mounted cameras.

3. **Accurate real-world ground-truth too expensive**: image and video simulators were designed for producing training data for the deep convolutional network algorithms.

There have not been any unforeseen risks so far.

4.4 Summary of Staffing

These are the people who have contributed substantially to the TrimBot2020 project (although not all are paid by the project).
<table>
<thead>
<tr>
<th>Name</th>
<th>Partner</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert Fisher</td>
<td>UEDIN</td>
<td>Coordinator &amp; PI</td>
</tr>
<tr>
<td>Radim Tylecek</td>
<td>UEDIN</td>
<td>Researcher</td>
</tr>
<tr>
<td>Can Pu</td>
<td>UEDIN</td>
<td>Researcher</td>
</tr>
<tr>
<td>James Garforth</td>
<td>UEDIN</td>
<td>Researcher</td>
</tr>
<tr>
<td>Joanne Pennie</td>
<td>UEDIN</td>
<td>Admin (contracts and finance)</td>
</tr>
<tr>
<td>Suzanne Perry</td>
<td>UEDIN</td>
<td>Admin</td>
</tr>
<tr>
<td>Eldert van Henten</td>
<td>WU</td>
<td>PI</td>
</tr>
<tr>
<td>Joris IJsselmuiden</td>
<td>WU</td>
<td>Researcher</td>
</tr>
<tr>
<td>Dejan Kaljaca</td>
<td>WU</td>
<td>Researcher</td>
</tr>
<tr>
<td>Thomas Brox</td>
<td>ALUF</td>
<td>PI</td>
</tr>
<tr>
<td>Nichlas Mayer</td>
<td>ALUF</td>
<td>Researcher</td>
</tr>
<tr>
<td>Benjamin Ummenhofer</td>
<td>ALUF</td>
<td>Researcher</td>
</tr>
<tr>
<td>Eddy Ilg</td>
<td>ALUF</td>
<td>Researcher</td>
</tr>
<tr>
<td>Huizhong Zhou</td>
<td>ALUF</td>
<td>Researcher</td>
</tr>
<tr>
<td>Theo Gevers</td>
<td>UVA</td>
<td>PI</td>
</tr>
<tr>
<td>Fares Alnajar</td>
<td>UVA</td>
<td>Researcher</td>
</tr>
<tr>
<td>Hoang-An Le</td>
<td>UVA</td>
<td>Researcher</td>
</tr>
<tr>
<td>Anil Baslamisli</td>
<td>UVA</td>
<td>Researcher</td>
</tr>
<tr>
<td>Nicolai Petkov</td>
<td>RUG</td>
<td>PI</td>
</tr>
<tr>
<td>Michael Wilkinson</td>
<td>RUG</td>
<td>Researcher</td>
</tr>
<tr>
<td>Nicola Strisciuglio</td>
<td>RUG</td>
<td>Researcher</td>
</tr>
<tr>
<td>Marc Pollefeys</td>
<td>ETHZ</td>
<td>PI</td>
</tr>
<tr>
<td>Torsten Sattler</td>
<td>ETHZ</td>
<td>Researcher</td>
</tr>
<tr>
<td>Dominik Honegger</td>
<td>ETHZ</td>
<td>Researcher</td>
</tr>
<tr>
<td>Johannes Schönberger</td>
<td>ETHZ</td>
<td>Researcher</td>
</tr>
<tr>
<td>Ian Cherabier</td>
<td>ETHZ</td>
<td>Researcher</td>
</tr>
<tr>
<td>Marcel Geppert</td>
<td>ETHZ</td>
<td>Researcher</td>
</tr>
<tr>
<td>Sara Meier</td>
<td>ETHZ</td>
<td>Researcher</td>
</tr>
<tr>
<td>Jochen Hemming</td>
<td>WR</td>
<td>PI</td>
</tr>
<tr>
<td>Toon Tielen</td>
<td>WR</td>
<td>Researcher</td>
</tr>
<tr>
<td>Bart van Tuijl</td>
<td>WR</td>
<td>Researcher</td>
</tr>
<tr>
<td>Pieter de Visser</td>
<td>WR</td>
<td>Researcher</td>
</tr>
<tr>
<td>Angelo Mencarelli</td>
<td>WR</td>
<td>Researcher</td>
</tr>
<tr>
<td>Peter Biber</td>
<td>Bosch</td>
<td>PI</td>
</tr>
<tr>
<td>Sebastian Haug</td>
<td>Bosch</td>
<td>Researcher</td>
</tr>
<tr>
<td>Michael Blaich</td>
<td>Bosch</td>
<td>Researcher</td>
</tr>
<tr>
<td>Simon Feeser</td>
<td>Bosch</td>
<td>Researcher</td>
</tr>
<tr>
<td>Stefan Laible</td>
<td>Bosch</td>
<td>Researcher</td>
</tr>
<tr>
<td>Maximilian Wenger</td>
<td>Bosch</td>
<td>Researcher</td>
</tr>
</tbody>
</table>

### 4.5 Summary of Dissemination Activities

The project has developed a comprehensive website ([http://www.trimbot2020.org](http://www.trimbot2020.org)) listing project members, hosting public deliverables (8) and datasets (1), with links to project publications (1 journal, 7 conference) and student dissertations (1), links to public posters and
presentations (2), link to the ResearchGate and Twitter pages, project highlight photos and videos, and links to events (1) that the consortium is hosting. More details are found in Section ??.


4.6 Summary of Project Meetings

We have had 5 two-day full-consortium meetings where normally all PIs and researchers attend: 8 Jan 2016 (Kickoff, ETHZ), 10 May 2016 (Groningen), 31 August 2016 (Edinburgh), 16 Jan 2017 (Freiburg), and 18 May 2017 (Amsterdam). Attendance is typically 20-25 people. Normally, the agenda consists of an administrative review, a review of each workpackage, research reports by all young researchers, and a set of project topic brainstorming sessions.

Three (of 4) Science and Exploitation Advisory Board members (Christensen, Eklundh, Hlavac) attended the 18 May 2017 consortium meeting.

There were also a number of technical working meetings:

WR/WU + Bosch (Feb 16, 2016) Arm and Platform integration discussion @ Renningen
BOSCH + ETHZ (March 30, 2016) Camera concept and camera/platform integration concept @ Renningen
WU/WR + UVA (Aug 22, 2016) Vision integration discussion @ Wageningen
WR/BOSCH (Aug 24, 2016) Test garden / Sensors on platform discussion @ Renningen
BOSCH + UEDIN + ETHZ (Dec 12-13, 2016) Integration of five stereo camera pairs and data recording @ Renningen
BOSCH + UEDIN + ETHZ (Mar 28-29, 2017) @ Renningen
BOSCH + UEDIN (Jun 7-8, 2017) Integration of Garden UI and Navigation @ Renningen
ALUF + ETHZ (Jun 12, 2017) Brainstorming for deep learned SLAM @ Freiburg
WU/ALUF/WR (20-23 June 2017) Integration of vision pipeline @ Wageningen
BOSCH + UEDIN + ETHZ (Jul 3-6, 2017) First integration of SLAM, Navigation and UI @ Renningen
BOSCH + UEDIN + ETHZ (15-16 August 2017) Integration for demo 1 @ Bosch
BOSCH + ETHZ (Sep 4-8, 2017) Third integration of SLAM, Navigation and UI @ Renningen

We also had 4 events where ground truth data from the gardens was collected:
WR/WU with other partners (Sep 10-23, 2016) Garden data collection @ Wageningen
BOSCH with other partners (Dec 2016) Garden data collection @ Renningen
BOSCH with other partners (Mar 2017) Garden data collection @ Renningen
WR/WU with other partners (May 16-17, 2017) Garden data collection @ Wageningen