Data Paper

The International Journal of Robotics Research
2017, Vol. 36(10) 1045–1052

DOI: 10.1177/0278364917720510
journals.sagepub.com/home/ijr

Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields

Nived Chebrolu[1], Philipp Lottes[1], Alexander Schaefer[2], Wera Winterhalter[2], Wolfram Burgard[2] and Cyrill Stachniss[1]

Abstract

There is an increasing interest in agricultural robotics and precision farming. In such domains, relevant datasets are often hard to obtain, as dedicated fields need to be maintained and the timing of the data collection is critical. In this paper, we present a large-scale agricultural robot dataset for plant classification as well as localization and mapping that covers the relevant growth stages of plants for robotic intervention and weed control. We used a readily available agricultural field robot to record the dataset on a sugar beet farm near Bonn in Germany over a period of three months in the spring of 2016. On average, we recorded data three times per week, starting at the emergence of the plants and stopping at the state when the field was no longer accessible to the machinery without damaging the crops. The robot carried a four-channel multi-spectral camera and an RGB-D sensor to capture detailed information about the plantation. Multiple lidar and global positioning system sensors as well as wheel encoders provided measurements relevant to localization, navigation, and mapping. All sensors had been calibrated before the data acquisition campaign. In addition to the data recorded by the robot, we provide lidar data of the field recorded using a terrestrial laser scanner. We believe this dataset will help researchers to develop autonomous systems operating in agricultural field environments. The dataset can be downloaded from http://www.ipb.uni-bonn.de/data/sugarbeets2016/.

Keywords

Agricultural robotics, precision farming, plant classification, sugar beet, localization and mapping

1. Introduction

Recently, there has been a growing interest in robots for precision agriculture, as they have the potential to significantly reduce the need for manual weed removal, or to lower the amount of herbicides and pesticides applied to a field. Unlike traditional weed eradication approaches, which treat the whole field uniformly, robots are able to selectively apply herbicides and pesticides to individual plants, thus using resources more efficiently. In order to increase the yield further, sustainable farming uses innovative techniques based on frequent monitoring of key indicators of crop health. Here, robots can serve as autonomous platforms for continuously collecting large amounts of data. In this context, this dataset aims at providing real-world data to researchers who develop autonomous robot systems for tasks like plant classification, navigation, and mapping in agricultural fields.

We collected the dataset on a sugar beet farm over an entire crop season using the agricultural robot depicted in Figure 1. The key idea was to observe a typical operational cycle of the robot: it starts in the garage, reaches the field, drives along the crop rows in the field, and finally returns back to the garage. The collected data amounts to approximately 5 TB. It includes visual plant data captured by an RGB-D sensor and a four-channel camera which, in addition to RGB information, also measures light emissions in the near-infrared (NIR) spectrum. The data related to navigation comprises wheel odometry, lidar scans, and two types of global positioning system (GPS) measurement: precise point positioning (PPP) and real time kinematic (RTK) information. In addition to the sensor data, we provide the intrinsic and extrinsic calibration parameters for all sensors, as well as development tools for accessing and manipulating the data, scripted in Python. Furthermore,

[1]University of Bonn, Bonn, Germany
[2]University of Freiburg, Freiburg im Breisgau, Germany

Corresponding authors:
Nived Chebrolu, University of Bonn, Photogrammetrie, IGG, Nussallee 15, 53115, Bonn, Germany.
Email: nived.chebrolu@igg.uni-bonn.de
Philipp Lottes, University of Bonn, Photogrammetrie, IGG, Nussallee 15, 53115 Bonn, Germany.
Email: philipp.lottes@igg.uni-bonn.de

Fig. 1. Field robot BoniRob operating on the field. Left: data acquisition three days after plant emergence. Right: data acquisition five weeks after emergence.

we provide an initial set of ground truth data for plant classification, that is, labeled images captured by the four-channel multi-spectral camera. The label classes comprise sugar beet plants and several weed species. In addition to that, early in the season we used a terrestrial laser scanner to obtain a precise three-dimensional (3D) point cloud of the field. This point cloud is also part of the dataset.

The main contribution of this paper is a comprehensive dataset of a sugar beet field that covers the time span relevant to crop management and weed control: from the emergence of the plants to a pre-harvest state at which the field is no longer accessible to the machines. The primary objective is to push developments and evaluations of different applications for autonomous robots operating in agricultural field environments. These applications can range from crop and weed classification to localization, mapping, and navigation on fields with plants in different growth states (Ball et al., 2016; Hall et al., 2015; Lottes et al., 2016b; Tellaeche et al., 2008; Underwood et al., 2015)

2. The agricultural robot platform: BoniRob

The BoniRob platform is a multi-purpose robot by Bosch DeepField Robotics. BoniRob is developed for applications in precision agriculture, that is, for mechanical weed control and selective herbicide spraying, as well as for plant and soil monitoring. It provides mounts for installing different tools for these specific tasks. BoniRob is equipped with four wheels which can be steered independently of each other, which allows for flexible movements and navigation on rough terrain.

2.1. Sensor setup

Figure 2 illustrates the locations of all sensors mounted on the BoniRob. They deliver (i) visual, (ii) depth, (iii) 3D laser, (iv) GPS, and (v) odometry data. In the following subsections, we give a brief overview of these sensors and describe their functions in relation to the perception system of the agricultural robot.

2.1.1. JAI AD-130GE camera. This camera is a prism-based 2-charge-coupled device (CCD) multi-spectral vision sensor, which provides image data of three bands inside the visual spectrum (RGB) and observes one band of the NIR spectrum. As plant leaves exhibit high reflectivity in the NIR spectrum due to their chlorophyll content (Rouse et al., 1974), the NIR channel is useful for separating vegetation from soil and other background data. Figure 3 depicts some example RGB and NIR images. The Bayer mosaic color CCD and the monochrome CCD of the JAI camera, both of size $\frac{1}{3}$", provide an image resolution of 1296 px $\times$ 966 px, respectively. One key feature of this camera system is its prism-based design: as the optical paths of the RGB and of the NIR channels are identical, the RGB and NIR data can be treated as one four-channel image. We mounted the camera to the bottom of the robot chassis at a height of around 85 cm above the soil, looking straight downwards. Using a Fujinon TF8-DA-8 lens with 8 mm focal length, this setup yields a ground resolution of approximately 3 px/mm and a field of view of 24 cm $\times$ 31 cm on the ground. The main purpose of the JAI AD-130GE camera is to capture detailed visual information of the plants for the crop and weed perception system of the robot as proposed in our previous work (Lottes et al., 2016a,b) and for detailed visual monitoring of the plant growth by extraction of key indicators for phenotyping applications. In order to be independent of natural light sources, we built an opaque shroud mounted on the bottom of the robot chassis and used controlled high-performance artificial light sources; see Figure 2.

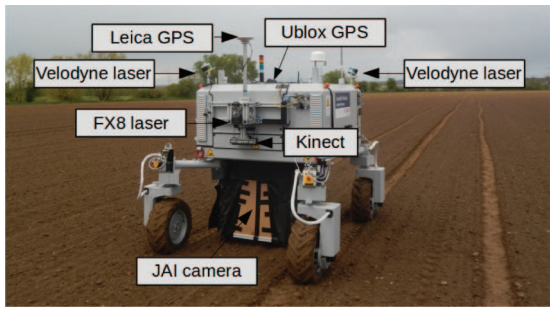

Fig. 2. Agricultural field robot BoniRob with all sensors. The JAI camera is mounted inside the shroud under the robot chassis and looks straight downwards.

2.1.2. Kinect One (Kinect v2). The Kinect is a time-of-flight camera by Microsoft, which provides the RGB and depth information of the scene. We mounted the Kinect sensor to the front of the robot, outside the shroud, and tilted it towards the ground. The main reason for positioning it outside the shroud was to avoid the interference with the JAI camera particularly in the NIR spectrum. In the dataset, we provide the rectified RGB, NIR, and depth images. As their pixels correspond to each other, they can be used for creating 3D point clouds. Figure 4 illustrates some examples of Kinect sensor data.

2.1.3. Velodyne VLP16 Puck. This 3D lidar sensor provides distance and reflectance measurements obtained by a

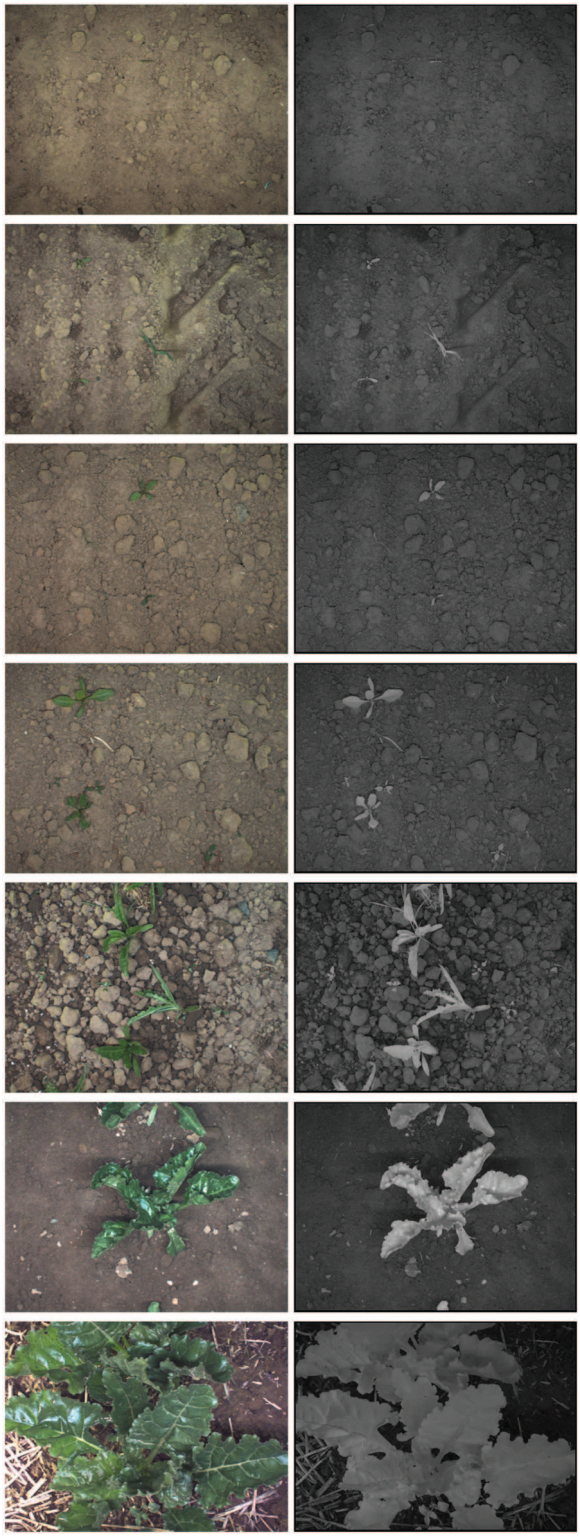

Fig. 3. Sugar beets and weeds captured with the JAI AD-130GE multi-spectral camera. The left column shows RGB images; the right one, the corresponding NIR images. The NIR channel shows a higher reflectivity for the vegetative parts. The image data in this dataset contains sugar beet data from its emergence (first row) up to the growth stage at which machines are no longer used for weed control, because their operation would damage the crops (last row).

rotating column of 16 laser diodes. Thus, the sensor has 16 scan planes, each of which provides a 360° horizontal field of view and a 30° vertical field of view with a horizontal resolution of 0.4° and a vertical resolution of approximately 2°. The sensor provides measurements up to a range of 100 m at a frequency of 20 Hz for a full 360° scan. Figure 5 illustrates a section of a single scan. The BoniRob is equipped with two of these sensors, one in the front right top corner of the chassis and the other in the rear left top corner. They are slightly tilted towards the ground to better detect objects close to the robot. The main purpose of the Velodyne sensors is to provide data for creating a 3D map of the environment, for localization, and for navigation tasks like obstacle detection.

2.1.4. Nippon Signal FX8. The FX8 is a 3D laser range sensor by Nippon Signal that provides distance measurements up to a maximum range of 15 m. It has a horizontal field of view of 60° and a vertical field of view of 50°. The data is provided at a rate of 4 Hz with a resolution of 97 px × 61 px. Typical images look like the one in Figure 6. The sensor is mounted on the front of the robot and tilted slightly towards the ground. It can be utilized for obstacle avoidance and to detect plant rows when navigating the field.

2.1.5. Leica RTK GPS. In order to track the robot's position, we employ a RTK GPS system by Leica, which provides accurate position estimates. The RTK GPS receiver tracks the signal of the satellites and additionally obtains observations from a nearby base station with a known location. With this information, the receiver computes corrections of the standard GPS signal and improves the position estimation to an accuracy of only a few centimeters. For details on this approach see Grewal et al. (2013). We recorded the position of the GPS antenna mounted on the robot with respect to the World Geodetic System 1984 (WGS84) at a frequency of 10 Hz. As an example, Figure 7 depicts all recorded paths during the data acquisition campaign.

2.1.6. Ublox GPS. In addition to the rather expensive RTK GPS solution by Leica, we used the low-cost Ublox EVK7-P GPS receiver to track the robot's position. The sensor's underlying principle of position estimation is PPP (Grewal et al., 2013). The advantage of this approach is its low price and the need for only one receiver. We tracked the position with this sensor at 4 Hz with respect to the WGS84.

2.2. Computer setup

The BoniRob has an onboard PC with a dual core i7 processor and 6 GB DDR3 memory; its operating system is Ubuntu 14.04. Apart from the Kinect, all sensor drivers are run on this PC, using the popular robot operating system (ROS) as middleware. The two Velodyne scanners, the JAI camera, and the FX8 scanner are connected to the onboard

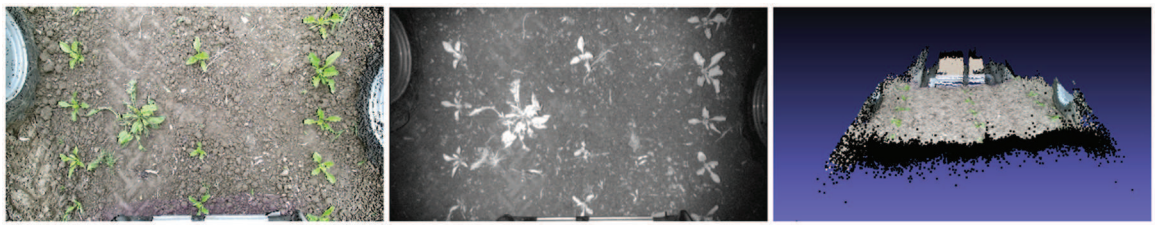

Fig. 4. Examples of the data recorded by the Kinect sensor. From left to right: rectified RGB image, infrared image, and processed point cloud by exploiting additional depth information.

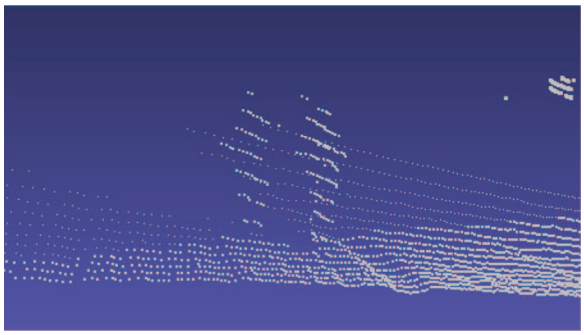

Fig. 5. Section of a scan resulting from a single revolution of the 16 laser diodes of the Velodyne VLP-16 sensor. The 3D point cloud shows two people walking close to the robot.

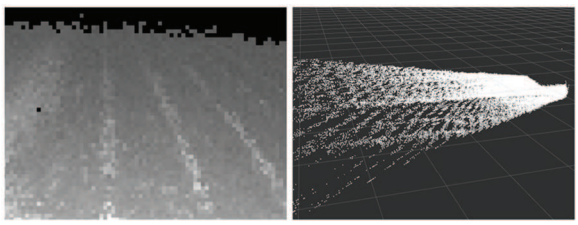

Fig. 6. Left: range image obtained using the FX8 laser scanner. Right: side view the of corresponding point cloud provided by the FX8.

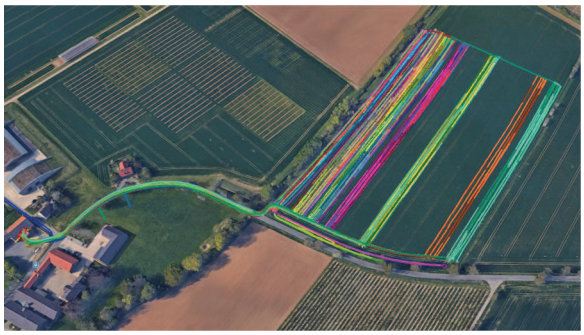

Fig. 7. Determined paths by the GPS sensor of the entire data acquisition campaign at the Campus Klein-Altendorf. Different colors refer to recordings on different days. Best viewed in color.

computer via an Ethernet hub. Due to the high data bandwidth required by the Kinect, we connected that sensor to a separate computer which was software-synchronized via network with the main PC before recording.

3. Data acquisition campaign

In the spring of 2016, we started to conduct a two-month data acquisition campaign at Campus Klein-Altendorf, a farm near Bonn in Germany. Specifically, we collected data on a sugar beet field during a crop season, covering the various growth stages of the plants; see Figure 3. On average, we acquired data on two to three days a week, leading to 30 days of recordings in total. In a typical day's recording, the robot covered between four and eight crop rows, each measuring 400 m in length. We controlled the robot manually during the data collection process, keeping its average speed at 300 mm/s. We recorded about 5 TB of uncompressed data during the whole data acquisition campaign: high-resolution images of the plants, depth information from the Kinect, 3D point clouds of the environment from the Velodyne and FX8 laser scanners, GPS positions of the antennas, and wheel odometry.

The data collection process was phased over time to cover the different growth stages of the sugar beet crop starting at germination. Our intention was to capture the key variations of the field during the time relevant to weed control and crop management. The robot visited several regions of the field multiple times during the data collection period. The dataset also captured different weather and soil conditions ranging from sunny and dry to overcast and wet. However, no collection was made during heavy rain, as the robot's tires would have sunk into the wet soil. In addition to the on-field recordings, we provide the data captured by the sensors while the robot drove from the garage to the field and back.

3.1. Sensor calibration

The dataset contains a complete set of calibration parameters. This involves intrinsic, that is, sensor-specific, calibration parameters for an appropriate interpretation of the sensor data, and a set of static extrinsic calibration parameters, which encode the relative poses of the sensors with respect to the robot's coordinate frame *base_link*. This information is essential for fusing the measurements obtained by the different sensors. The *base_link* frame is defined as the center of the bottom plane of the robot chassis, as illustrated by Figure 8.

3.1.1. Extrinsics. In order to allow for fusion of measurements of different sensors, we provide the 3D transformations from the robot frame *base_link* to the coordinate

Table 1. Extrinsic parameters for the transformation from the robot's coordinate frame *base_link* to the frame of each sensor. The translation is given by x, y, and z; the rotation is given by the quaternion q.

Sensor	x [m]	y [m]	z [m]	qx	qy	qz	qw
JAI AD-130GE	0.081	0.138	−0.073	−0.698	0.716	0.008	0.008
Microsoft Kinect v2	0.876	0.040	0.152	0.712	−0.702	0.019	−0.022
Velodyne VLP-16 (front)	0.690	−0.612	0.977	0.168	−0.183	0.968	0.015
Velodyne VLP-16 (rear)	−0.705	0.528	0.972	−0.070	−0.253	−0.638	0.724
Nippon Signal FX8	1.073	−0.086	0.473	0.845	−0.005	0.535	0.010
Leica GPS	−0.861	−0.081	−1.146	–	–	–	–
Ublox GPS	−0.685	−0.053	−0.764	–	–	–	–

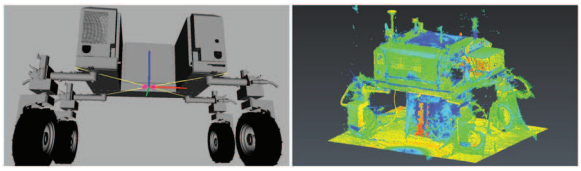

Fig. 8. Left: illustration of the robot's coordinate frame, called *base_link*: the x-axis is colored red, the y-axis is green, and the z-axis is blue. Right: reconstructed 3D model of the field robot. The sensor coordinate systems were determined by measuring the poses of the sensor casings in this model and then looking up the sensor coordinate system with respect to the casing in the data sheets.

system of each sensor in Table 1. The positions of the sensors on the robot are depicted in Figure 2. We determined the pose of each sensor in the following manner: first, we built a 3D model of the robot using the FARO X130 terrestrial laser scanner and extracted the poses of the sensor casings from it. Second, we derived the reference pose of the sensor (for example the projection center of the camera) from the mechanical drawings provided by the manufacturer. For both Velodyne scanners and the FX8, we additionally performed a high-precision alignment procedure based on sensor data: we positioned the robot in a structured environment with multiple walls and then used the overlap of the fields of view of the front Velodyne and of the FX8 or of the rear Velodyne, respectively, to accurately align the scans based on scan matching. The pose corrections computed by the scan matcher resulted in the final calibration poses. Following this procedure, the inter-sensor rotations obtained have an uncertainty in the order of 1° and the translations between sensors have an uncertainty of around 1 cm.

3.1.2. Intrinsics. The JAI camera provides two types of images, an RGB image and an NIR image. For both images, we included the camera calibration parameters based on the pinhole model in the dataset. We estimated these parameters using the OpenCV camera calibration library (Bradski, 2000) by registering images of checkerboard patterns. As far as the Kinect calibration is concerned, the dataset comes with camera parameters for the color and the NIR image, for the relative orientation between those two, and a depth correction parameter. In order to obtain these parameters, we used the procedure described at https://github.com/code-iai/iai_kinect2. The Kinect data provided is already registered and modified according to the depth correction. Therefore, no further correction is required by the user. For the Velodyne data, we specify the distance correction and the offset parameter values for each of the 16 laser diodes. As with the Kinect, we have already applied these corrections to the point clouds in the dataset.

4. Data description

In this section, we describe the structure of the dataset, all types and the data formats used, and how to access its individual parts.

4.1. Dataset structure

The whole dataset is divided into multiple folders, each of which contains the data recorded on a certain day of the acquisition campaign. Furthermore, we divided each day's recording into smaller chunks of data. We originally recorded the dataset using rosbag, the standard tool for data recording provided by ROS. The resulting so-called bag files (**.bag*), which contain all recorded data, were split whenever they reached the file size limit of 4 GB. We converted them to standard raw formats for portability. The chunks of raw data correspond to the split bag files. In order to accommodate both users familiar and users unfamiliar with ROS, the dataset contains both the original ROS bag files and the converted raw data files. The chunks can be downloaded as individual zip archives. All calibration parameters are provided in a separate zip file. They are valid for all the recordings provided in the dataset.

The chunks are further subdivided: each sensor modality, like image data, laser data, odometry measurements, and GPS positions, has its own folder. The *camera* folder consists of data from the JAI camera and the Kinect, the *laser* folder holds Velodyne and FX8 data, the *gps* folder contains the GPS positions read from the Ublox and Leica GPS receivers, and the *odometry* folder contains wheel odometry estimates. An overview of the folder hierarchy of a chunk is illustrated in Figure 9.

Some of the chunks do not contain all sensor information. During the first days of the campaign, we experienced

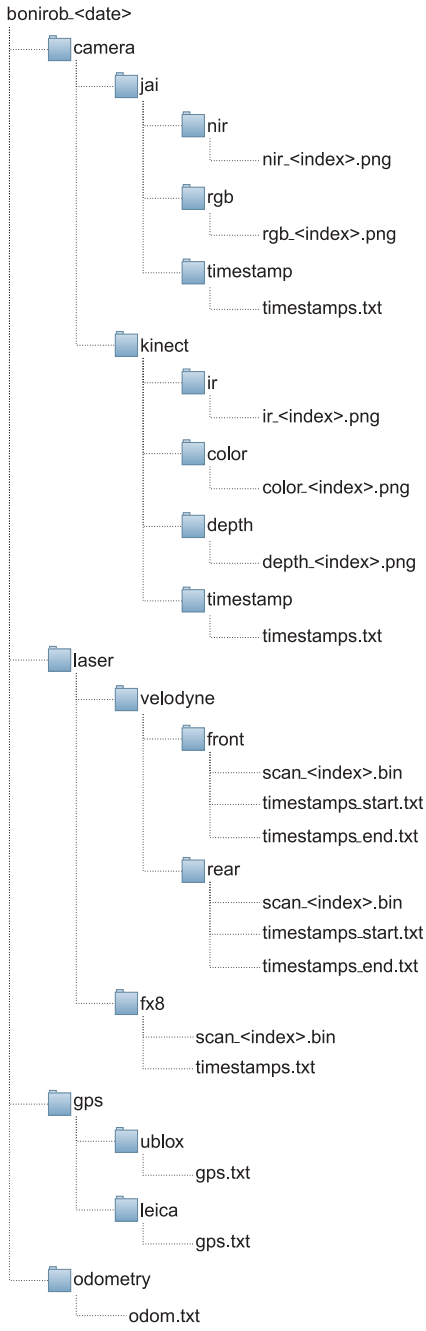

Fig. 9. Folder structure for each chunk of data. The term <*date*> refers to the date and time of the acquisition of a certain chunk, while the term <*index*> identifies each piece of data within a chunk.

some issues with the GPS hardware. We also experienced crashes of the drivers for the Kinect and the rear Velodyne sensor. A list of all missing sensor measurements per chunk is provided in the file *missing_measurements.txt*.

4.2. Data types and formats

In this section, we briefly describe the file formats of the raw data for each sensor.

4.2.1. Camera data. All camera images have been stored in losslessly compressed PNG files. The files are named according to the following convention:

`camera/<sensor>/<type>/<type>_<index>.png`

where *sensor* is either *jai* or *kinect*, type is *rgb* or *nir* for the JAI camera and *color*, *ir*, or *depth* for the Kinect, and *index* is the image index. The Kinect image depth is 16 bit. The *timestamps* folder provides the timestamps of the individual images. We have taken care to synchronize the timestamps of all images for a given camera. The intrinsic and extrinsic calibration parameters are provided separately in the *calibration* folder. Note that for the JAI camera, the RGB and the NIR images are captured through a prism. Therefore, their relative orientation is identity. For the Kinect, the point cloud can be generated from the given raw data using the `generate_kinect_pointcloud` function in the development tools.

4.2.2. Laser data. The laser data has been logged using two Velodyne laser scanners (front and rear) and a Nippon Signal FX8 scanner. The resulting point clouds are in a binary format containing the fields [*x, y, z, intensity, ring*]. The first three fields yield the position of the detected point in meters, and *intensity* is a value in [0, 255]; higher values denote higher reflectance. For the Velodyne VLP-16, each ring number corresponds to a certain laser diode. Each of the 16 laser diodes measures a profile on a certain scan plane. This yields a 3D point cloud even when the robot is not moving around. The *ring* value is set to -1 for all FX8 scans, as this information is not applicable. Furthermore, the intensity values of the point cloud correspond to the infrared reflectance. The binary files are stored as

`laser/<sensor>/scan_<index>.bin`

where *sensor* is *velodyne/front*, *velodyne/rear*, or *fx8*, and *index* is the scan index in a chunk. Again, the *timestamps* folder holds the timestamp of each scan in seconds. Note that for the Velodyne laser scanners, both the start and end times of each scan are provided. This allows for interpolation of the timestamps for the individual laser diode firings (see Velodyne manual for details). The intrinsic calibration information is already applied to all laser scans. We provide functions to access the resulting point cloud data in the software tools that come with the dataset.

4.2.3. GPS data. GPS data was logged using two devices, a Leica RTK system and a low-cost Ublox EVK7-PPP. This data is saved as text files in

`gps/<sensor>/gps.txt`

where *sensor* is either *leica* or *ublox*. Each line in the GPS log file corresponds to a position. This position refers to the WGS84 system and is formatted [*timestamp, latitude, longitude, altitude*], where *latitude* and *longitude* are specified in degrees, while the *altitude* measurements are given

Fig. 10. Left: special extra-high tripod equipped with a FARO X130 terrestrial laser scanner. Right: part of the registered point cloud of the sugar beet field. The black lines show false measurements.

Fig. 11. Left: RGB image captured by the JAI camera. Right: corresponding ground truth image encoding sugar beet (red) and several weed species (other colors).

in meters. The Leica RTK measurements were logged at 10 Hz, and the Ublox measurements, at 4 Hz. We noticed that the RTK GPS receiver occasionally lost its signal, particularly when the robot was moving along the border of the field close to trees. Please note that the Leica RTK GPS data are unfiltered raw measurements, whereas the Ublox driver provides filtered data only, without explicitly specifying the underlying algorithm.

4.2.4. Odometry data. The wheel odometry data was saved to the text file

`odometry/odom.txt`

Each line in this file corresponds to an odometry measurement. The measurements are formatted [*timestamp,* $\dot{x}$*,* $\dot{y}$*,* $\dot{z}$*,* ω*,* x*,* y*,* ϕ]. The dotted variables and ω refer to the translational velocity in meters per second and the rotational speed around the z-axis in radians per second, respectively, whereas x, y, and ϕ denote the position in meters and the heading in radians of the robot. The position information is obtained by integrating the velocities from the beginning of the data acquisition session on that day. Thus, the robot position with respect to the position at the beginning of a chunk can simply be obtained by subtracting the position of the first measurement of the chunk from each new measurement. Note that wheel slippage varies throughout the dataset depending on the position of the robot on the field and on the dampness of the soil.

4.2.5. Terrestrial Laser Scanner data. In addition to the data captured by the robot, we collected 3D laser scans of the sugar beet field with a FARO X130 terrestrial laser scanner mounted on a stationary tripod. We scanned the field on 10 May 2016, when the plants were small. Localizing a robot in such an environment without relying on GPS would be a challenging task. See Figure 10 for an illustration of the Terrestrial Laser Scanner (TLS) data. We recorded several scans from different view points to cover almost the whole sugar beet field. In order to obtain a complete 3D scan of the field, we registered the individual scans using checkerboard targets on the field and an iterative closest point procedure. Finally, leveraging the TLS's GPS, compass, and inclinometer, we computed the pose of the registered point cloud with respect to the WGS84. The scans were stored in a text file, which contains the x, y, and z coordinates of each point in meters along with the *intensity* values.

4.2.6. Ground truth data for plant classification. For a small portion of the JAI images, we provide labeled ground truth data. Figure 11 depicts an RGB image captured by the JAI camera and its corresponding ground truth annotation. The ground truth data does not only encode vegetative (colored) and non-vegetative (black) parts, but also distinguishes different classes of the former: sugar beets (red) and several weed species. In sum, we manually labeled around 300 images as accurately as possible, identifying sugar beets and nine different types of weed. In the future, further labeled data will be made available on the website.

5. Software tools

Along with the raw data, we provide a basic set of Python tools for accessing and working with the dataset. After loading the dataset into memory, its hierarchical structure is mapped to a nested object in Python, which can easily be accessed using the dot operator. The tools use the same naming convention as the one employed for storing the data in various folders on the disk. For example, after loading the camera data by calling `dataset.load_camera()` images from all cameras are stored in `dataset.camera`. If we are interested in the JAI camera data, we access it using `dataset.camera.jai`. Going further down the hierarchy, RGB and NIR images from the JAI camera are represented by `dataset.camera.jai.rgb` and `dataset.camera.jai.nir`, respectively.

In addition to these basic methods to access the data, we provide further utility functions. For example, the `generate_kinect_pointcloud` method computes the point cloud from Kinect raw data, and `save_kinect_pointcloud_as_ply` saves these point clouds as standard PLY files. The latter can be processed by tools such as Meshlab, MATLAB, and so on. Along with the tools, we provide an example script that explains how to use the various methods. The development tools can be downloaded from the dataset website as well.

6. Summary

We present a large-scale agricultural robot dataset for development of plant classification systems as well as robot localization and mapping applications on agricultural fields. To the best of our knowledge, no comparable, publicly available dataset exists. The data was collected during one crop season, capturing the various changes in the field as the crops grew. In sum, we collected 5 TB of data from vision, laser, GPS, and odometry sensors. We also provide a basic set of software tools to access the data easily. Furthermore, we annotated a subset of images for classification. The main intention of this work is to provide researchers with a challenging real-world dataset that helps develop autonomous capabilities for field robots.

Acknowledgements

The authors would like to thank the team at the Campus Klein-Altendorf for their contributions concerning this data acquisition campaign and for granting access to the fields.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has partly been supported by the European Union (grant number H2020-ICT-644227-FLOURISH).

References

Ball D, Upcroft B, Wyeth G, et al. (2016) Vision-based obstacle detection and navigation for an agricultural robot. *Journal of Field Robotics* 33(8): 1107–1130.

Bradski G (2000) *Dr. Dobb's Journal of Software Tools*. Available at: http://code.opencv.org/projects/opencv/wiki/CiteOpenCV

Grewal MS, Andrews AP and Bartone CG (2013) *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. New York, NY: Wiley-Interscience.

Hall D, McCool C, Dayoub F, et al. (2015) Evaluation of features for leaf classification in challenging conditions. In: *2015 IEEE winter conference on applications of computer vision (WACV),* IEEE, 5 January 2015, pp. 797–804.

Lottes P, Hoeferlin M, Sander S, et al. (2016a) An effective classification system for separating sugar beets and weeds for precision farming applications. In: *Proceedings of the IEEE international conference on robotics & automation (ICRA)*, IEEE, May 2016, pp. 5157–5163.

Lottes P, Hörferlin M, Sander S, et al. (2016b) Effective vision-based classification for separating sugar beets and weeds for precision farming. *Journal of Field Robotics*.

Rouse JW Jr, Haas RH, Schell JA, et al. (1974) Monitoring vegetation systems in the great plains with ERTS. *NASA Special Publication* 351: 309–317.

Tellaeche A, Burgos-Artizzu X, Pajares G, et al. (2008) A vision-based method for weeds identification through the Bayesian decision theory. *Pattern Recognition* 41(2): 521–530.

Underwood JP, Jagbrant G, Nieto JI, et al. (2015) Lidar-based tree recognition and platform localization in orchards. *Journal of Field Robotics* 32(8): 1056–1074.