Proceedings of the IEEE IROS workshop on

Agricultural Robotics

learning from Industry 4.0 and moving into the future

September 28, 2017

Vancouver, Canada

Tsampikos Kounalakis
Frits van Evert
David Michael Ball
Gert Kootstra
Lazaros Nalpantidis

<https://doi.org/10.18174/434834>

Table of content

Objectives	3
Organisers.....	4
Peer-reviewed papers	5
Timo Korthals, Mikkel Kragh, Peter Christiansen and Ulrich Rückert Towards Inverse Sensor Mapping in Agriculture	6
Florian Kraemer, Alexander Schaefer, Andreas Eitel, Johan Vertens and Wolfram Burgard From Plants to Landmarks: Time-invariant Plant Localization in Agricultural Fields that uses Deep Pose Regression.....	12
Ruud Barth, Jorijs Ijsselmuiden, Jochen Hemming and Eldert J. van Henten Optimising Realism of Synthetic Agricultural Images using Cycle Generative Adversarial Networks	18
Thibault Tourrette, Roland Lenain, Raphael Rouveure and Thomas Solatges Tracking footprints for agricultural applications: a low cost lidar approach	23
Chanyoung Ju, Sungjun Park, Sangsoo Park and Hyoung Il Son A Haptic Teleoperation of Agricultural Multi-UAV	29
Lars Grimstad, Kristine Skattum, Eirik Solberg, Gabriel Da Silva Martins Loureiro and Pål Johan From Thorvald II Configuration for Wheat Phenotyping	35

Objectives

Agriculture provides humans with food and raw materials although with a large impact on the environment. Precision farming is a promising pathway to increase the sustainability of farming by increasing farm profitability, reducing manual labour, and reducing environmental impact. In particular it is expected that robotized precision farming solutions through high precision targeted interventions will increase yield and create a more viable ecological solution. Whilst robotics is moving out of the lab into industrial settings, many recent advances in robotics have not yet found their way into the agricultural domain. Agriculture demands high levels of intelligence, mobility, autonomy, manipulation and robustness. While subsets of those characteristics have been targeted by the industrial-, field- and space- robotics communities no all-encompassing solutions have yet been reported. The successes of intelligent robots in semi-structured and semi-controlled industrial environments give rise to optimism about robots working in uncontrolled agricultural environments.

Learning from Industry 4.0, agricultural robots will have to possess attributes such as *modularity*, *cyber-physical monitoring*, *decentralized decision making* and *Internet of Things (IoT)* implementation which are the key features of the ongoing industrial revolution:

- *Modular* structured agricultural systems using tools according to the examined agricultural application.
- *Cyber-physical systems monitoring* providing supervision and control over various physical processes.
- *Decentralized decision making* by using external computational resources while simplifying system requirements and design.
- *Implementation the Internet of Things* using information distributed between sensors and other robotic systems in the same area of operation, and applying big data analytics for taking decisions.

Robots for this envisioned future farming require advances in a broad range of topics that surpass current state-of-the-art capabilities. This workshop aspires to shed light in the latest advancements towards this goal and open the discussion about the required future steps and research directions.

Topics of interest

The workshop covered, but was not restricted to, the following topics.

- Robotized precision farming.
- Farming sustainability, profitability and viable ecological solution.
- Repurposing of industrial-, field- and space-robots for agriculture.
- Analogies between Industry 4.0 and agricultural applications.
- Modular structured agricultural systems.
- Cyber-physical system agricultural production monitoring.
- Decentralized decision and computation for agricultural applications.
- Internet of Things on agriculture sensing devices.
- Novel perception for agricultural robots including passive and active methods.
- Long-term autonomy and navigation in unstructured environments.
- Adaptive sampling and informative data collection.
- Deployment strategies and user interfaces for end-users.

Support by IEEE RAS AgRA TC

This workshop was endorsed by the IEEE RAS Technical Committee on Agricultural Robotics and Automation (AgRA TC)

Website

<https://agrifoodroboticsworkshop.com/iros2017/>

Organisers

The workshop was organised by:

Tsampikos Kounalakis

Department of Mechanical Production and Management Engineering
Aalborg University Copenhagen
Denmark
Tel.: +45-93562204
tkoun@make.aau.dk

Frits van Evert

Agrosystems Research
Wageningen Plant Research
Netherlands
frits.vanevert@wur.nl

David Michael Ball

Robert Bosch Startup Deepfield Robotics
Germany
DavidMichael.Ball@de.bosch.com

Gert Kootstra

Farm Technology
Wageningen University
gert.kootstra@wur.nl

Lazaros Nalpantidis

Department of Mechanical Production and Management Engineering
Aalborg University Copenhagen
Denmark
lanalpa@make.aau.dk

Advisory Committee

- Marcel Bergerman, Carnegie Mellon University, USA.
- Eldert van Henten, Wageningen University, Netherlands.

Peer-reviewed papers

Submitted papers were subject to thorough peer-reviewing process, with peers selected from the members of the Program Committee. The PC selected twelve papers for oral and poster presentations. The remainder of this document provides the full texts of these contributions.

For citing the papers in your own work, please use the following standard:

[*authors*] (2017) [*title*]. In Kounalakis, T., van Evert, F., Ball, D.M., Kootstra, G., and Nalpantidis, L. (Eds.), *Proceedings of the IROS Workshop on Agricultural Robotics: learning from Industry 4.0 and moving into the future*. September 28, 2017, Vancouver, Can

Towards Inverse Sensor Mapping in Agriculture

Timo Korthals¹, Mikkel Kragh², Peter Christiansen², and Ulrich Rückert¹

Abstract—In recent years, the drive of the Industry 4.0 initiative has enriched industrial and scientific approaches to build self-driving cars or smart factories. Agricultural applications benefit from both advances, as they are in reality mobile driving factories which process the environment. Therefore, accurate perception of the surrounding is a crucial task as it involves the goods to be processed, in contrast to standard indoor production lines. Environmental processing requires accurate and robust quantification in order to correctly adjust processing parameters and detect hazardous risks during the processing. While today approaches still implement functional elements based on a single particular set of sensors, it may become apparent that a unified representation of the environment compiled from all available information sources would be more versatile, sufficient, and cost effective. The key to this approach is the means of developing a common information language from the data provided. In this paper, we introduce and discuss techniques to build so called inverse sensor models that create a common information language among different, but typically agricultural, information providers. These can be current live sensor data, farm management systems, or long term information generated from previous processing, static drone images, or satellites. In the context of Industry 4.0, this enables the interoperability of different agricultural systems and allows information transparency.

I. INTRODUCTION

Agricultural vehicles are complex, mobile processors of biological products that operate in unstructured and constantly changing environment. While the operation of these vehicles was initially relatively, today their setup and use requires trained specialists due to the requirement of increasing efficiency and lowering overall costs. However, without automation and the augmenting of parameter optimization in the process chain, throughputs, and farming yields would be much smaller than usual. For instance, automated steering systems employed in harvesting use LiDAR systems to scan the area between the crop and stubble in order to automatically guide the harvester along the edge; and seed drills save GPS data and the machine parameters of sowing which are used later to minimize the utilization of fertilizer spreaders.

Focusing the automation and in particular its implementation, all applications follow the same paradigm of having a distinctive set of sensors, a processing unit, and an actuator interface to steer the vehicle or manipulate process parameters. While this approach allows simple, distributed

and modular modification, with increases in automated functionality its installation and maintenance becomes unfeasible due to the sheer number of sensors and processing units required. Furthermore, the potential for sensor fusion is completely squandered. An alternative approach is pursued by the authors, that of building a common inner semantical representation of the environment based on occupancy grid maps, from which all further automation is derived [1], [2]. These grid maps are arranged in multiple overlapping layers, where each one is occupied by localized classifications.

While the authors have already provided a proof-of-concept of semantical grid mapping approaches in agriculture [3], requisite information and instructions for building sensor models based on sensors and other data sources is still lacking. In contrast to robotic and automotive approaches, where grid mapping based applications are well known, agricultural environments and applications especially vary greatly and therefore have to be treated accordingly. With respect to Fig. 1 and [4], this contribution focuses on the *Inverse Sensor Modeling* component.

The paper is organized as follows: Section II presents a brief introduction to occupancy grid maps, their extension to the semantical representation. Section III presents the gathered experience and approaches to building sensor models derived from previous agricultural research projects. Finally, Section IV presents further ideas and points to next steps in agricultural applications in Industry 4.0.

II. RELATED WORK

Occupancy grid maps are used in static obstacle detection for robotic systems, which are a well-known and a commonly studied scientific field [5], [6], [7]. They are a component of almost all navigation and collision avoidance systems designed to maneuver through cluttered environments. Another important application is the creation of obstacle maps for traversing an unknown area and the recognition of known obstacles, so supporting the localization. Recently, occupancy grid maps have been applied to combine LiDAR and RADAR in automotive applications, with the goal of creating a harmonious, consistent and complete representation of the vehicle's environment as a basis for advanced driver assistance systems [8], [9], [10].

A. Occupancy Grid Mapping

Two-dimensional occupancy grid maps (OGM) were originally introduced by Elfes [11]. In this representation, the environment is subdivided into a regular array or a grid of quadratic cells. The resolution of the environment representation directly depends on the size of the cells. In addition to

¹Bielefeld University, Cluster of Excellence Cognitive Interaction Technologies, Cognitronics & Sensor Systems, Inspiration 1, 33619 Bielefeld, Germany, <http://www.ks.cit-ec.uni-bielefeld.de/>, {tkorthals, rueckert}@cit-ec.uni-bielefeld.de

²Aarhus University, Department of Engineering, Finlandsgade 22, DK-8200 Aarhus N, Denmark <http://eng.au.dk/>, {mkha, pech}@eng.au.dk

Fig. 1: Semantic occupancy grid mapping framework

this compartmentalization of space, a probabilistic measure of occupancy is associated with each cell. This measure takes any real number in the interval $[0, 1]$ and describes one of the two possible cell states: unoccupied or occupied. An occupancy probability of 0 represents a space that is definitely unoccupied, and a probability of 1 represents a space that is definitely occupied. A value of 0.5 refers to an unknown state of occupancy.

An occupancy grid is an efficient approach to representing uncertainty, combining multiple sensor measurements at the decision level, and to incorporating different sensor models [10]. To learn an occupancy grid M given sensor information z , different update rules exist [5]. For the authors' approach, a Bayesian update rule is applied to every cell $m \in M$ at position (w, h) as follows: Given the position x_t of a vehicle at time t , let $x_{1:t} = x_1, \dots, x_t$ be the positions of the vehicle's individual steps until t , and $z_{1:t} = z_1, \dots, z_t$ the environmental perceptions. For each cell m of the occupancy probability grid the probability that this cell is occupied by an obstacle. Thus, occupancy probability grids seek to estimate

$$P(m|z_{1:t}, x_{1:t}) = \text{Odds}^{-1} \left(\prod_{t=1}^T \frac{P(m|z_t, x_t)}{\text{Odds}(P(m|z_t, x_t))} \right) \quad (1)$$

This equation already describes the online capable, recursive update rule that populates the current measurement z_t to the grid, where $P(m|z_{1:t}, x_{1:t})$ is the so called inverse sensor model (ISM). The ISM is used to update the OGM in a Bayesian framework, which deduces the occupancy probability of a cell, given the sensor information.

B. Extension to Agriculture Applications

The adaptation of OGM techniques to agricultural applications appears to be merely a matter of time but is not that obvious and intuitive to apply on the second sight. Robotic and automotive applications have in common that they both want to detect non-traversable areas or objects occupying their path. Such unambiguous information is used to quantify the whole environment sufficiently for all derivable tasks, such as path planning or obstacle avoidance, to be completed. When assumptions like a flat operational plane or minimum

obstacle heights are made, sensors frustums oriented parallel to the ground are sufficient for all tasks

In agricultural applications, obstacle recognition is not essential as they act on and process their environment. Therefore, quantification of the environment involves features such as processed areas, processability, crop quality, density, and maturity level in addition to traversability. In order to map these features, single occupancy grid maps are no longer sufficient and therefore, semantic occupancy grid maps that allow different classification results to be mapped are used. Furthermore, sensor frustums are no longer oriented parallel to the ground, but rather oriented at an angle to gather necessary crop information (cf. Fig. 2).

The extension to semantic occupancy grid maps (SOGM) or inference grids is straightforward and is defined by an OGM M with W cells in width, H cells in height, and N semantic layers (c.f. Fig. 1):

$$M : \{1, \dots, W\} \times \{1, \dots, H\} \rightarrow m = \{0, \dots, 1\}^N \quad (2)$$

Compared to a single layer OGM which allows the classification into three classes $\{\text{occupied}, \text{unoccupied}, \text{unknown}\}$, the SOGM supports a maximum of $|\{\text{occupied}, \text{unoccupied}, \text{unknown}\}|^N = 3^N$ different classes allowing much higher differentiability in environment and object recognition. The corresponding ISMs are fused by means of the occupancy grid algorithm to their n th associated semantical occupancy grid.

The location of information in the maps is required to be completed by *mapping under known poses* approaches [6]. As proposed by REP-105¹ and realized by the authors in [4], information is mapped locally via Kalman filtered odometry and inertial navigation measurement. The maps themselves are globally referenced which on the one side allows smooth local mapping in the short term without the discrete jumps caused by global positioning systems using a Global Navigation Satellite System (GNSS), but also allows global consistent storing and loading of information.

While the actual features are very diverse of agriculture applications, this publication does not primarily focus on classification, but rather on geographical interpretation and sensor building.

¹<http://www.ros.org/reps/rep-0105.html>

Fig. 2: Ground oriented LiDAR for crop rectification

III. EXPLICIT ISM GENERATION FOR SPECIFIC SENSORS

A. Local Sensor Based ISM

1) *LiDAR based Mapping*: LiDAR sensors measure the distance to an object and depending on their capabilities, also the reflectance. The distance can directly be used to deduce free (s.t. the area between the measured distance and the sensor) and occupied space (s.t. the location of measured distance) in a planar environment. This is commonly utilized for robotic and automotive tasks, where a well-known inverse sensor modelling technique directly derives the corresponding ISM. In agriculture, however, it is common for LiDAR sensors to face downwards as shown in Fig. 2, in order to detect the soil or crop that needs to be processed. This results in the circumstance that the measurement can only be taken at the corresponding target point, and no implications can be done along the measurement.

Naively mapping the related classification in the point of measurement in the vehicles coordinate frame would result in scattered maps from which further applications are hardly derivable (c.f. Fig. 3). Therefore, the actual Gaussian measurement uncertainty σ_S needs to be introduced as in the common planar model, but with its appropriate error propagation. Assuming σ_ϕ , σ_ξ , σ_γ being gaussian noise in the angular positioning caused by vehicle's steering, and σ_x , σ_y , σ_z to be the positioning caused by vibrations of the vehicle it is possible to calculate the resulting full covariance matrix $\sum_{X_S}$ at the point of interest as follows: First, the transformation of the scalar distance measurement S in the LiDAR frame to the euclidean point X_S in the vehicle frame is

$$X_S = \begin{pmatrix} c_\phi c_\gamma \\ c_\xi s_\gamma + c_\gamma s_\phi s_\xi \\ s_\xi s_\gamma - c_\gamma s_\phi c_\xi \end{pmatrix} S + T(x, y, z) \quad (3)$$

where T is the translation between the sensor and the vehicle frame. For error propagation, the functions need to be linearized by calculating the Jacobian:

$$J^T = \begin{pmatrix} c_\phi c_\gamma & c_\xi s_\gamma + c_\gamma s_\phi s_\xi & s_\xi s_\gamma - c_\gamma s_\phi c_\xi \\ -S s_\phi c_\gamma & S c_\gamma c_\phi s_\xi & -S c_\gamma c_\phi c_\xi \\ -S c_\phi s_\gamma & S c_\xi c_\gamma - S s_\gamma s_\phi s_\xi & S s_\xi c_\gamma + S s_\gamma s_\phi c_\xi \\ 0 & -S s_\xi s_\gamma + S c_\gamma s_\phi c_\xi & S c_\xi s_\gamma + S c_\gamma s_\phi s_\xi \end{pmatrix}^T \quad (4)$$

Fig. 3: Harvesting scenario (left), resulting SOGM from crop classification ISM with (middle) and without (right) error propagation

$$\sum_{X_S} = J \text{diag}(\sigma_s^2, \sigma_\phi^2, \sigma_\gamma^2, \sigma_\xi^2) J^T + \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2) \quad (5)$$

The Jacobian is a function of its arguments $J(S, \phi, \gamma, \xi)$, which means that it is required to be evaluated for every new sensor measurement. Equation 5 describes the full covariance matrix which can be applied to calculate the uncertainty distribution for every measurement.

Two assumptions have been made in this model to make the error model tractable: first, that the uncertainty in angular movements resides in the coordinate frame of the laser scanner and second, that the uncertainty in translation is uncorrelated from the angular ones. The assumptions do not fully hold, due to the fact that rolling, pitching and yawing do not occur in the laser scanner frame, but in some other arbitrary frame, depending on the current ground conditions and vehicle's steering. To simplify the model even more, the uncertainty in z can be omitted, because in the later sensor modeling component, only the projection into the xy -plane is important. Further, rolling is omitted as it is negligible in comparison to the other influences [12]:

$$X'_S = \begin{pmatrix} c_\phi c_\gamma \\ s_\gamma \\ -c_\gamma s_\phi \end{pmatrix} S + T(x, y, z) \\ J' = \begin{pmatrix} c_\phi c_\gamma & -S s_\phi c_\gamma & -S c_\phi s_\gamma \\ s_\gamma & 0 & S c_\gamma \\ 0 & -S s_\xi s_\gamma + S c_\gamma s_\phi c_\xi & S c_\xi s_\gamma + S c_\gamma s_\phi s_\xi \end{pmatrix} \quad (6) \\ \sum_{X'_S} = J' \text{diag}(\sigma_s^2, \sigma_\phi^2, \sigma_\gamma^2) J'^T + \text{diag}(\sigma_x^2, \sigma_y^2)$$

The influences of error propagation are depicted in Fig. 3 where a two class classifier for crop derives the ISMs which are mapped to the global coordinate system. The resulting map without error propagation is very sparse which makes further functionality derivation without heuristical post processing unfeasible. Introducing error propagation and respecting the model uncertainties, on the other hand, results in a much more sufficient and consistent map where further classification can easily be applied.

Further improvements in classification can be achieved by first mapping the raw LiDAR data to a globally referenced representation from which further ISMs with much higher quality can be derived. More advanced LiDAR systems scanning in multiple planes bypass the raw mapping and directly enable rich classifiers like Support Vector Machines to process the data as proposed by [3].

Fig. 4: Inverse Perspective Mapping of RGB image

Fig. 5: (Left) Grass and human predictions in a mowing application classified by a fully convolutional network for semantic segmentation [15] and the corresponding ISMs generated by IPM (right)

2) *Inverse Perspective Mapping*: Inverse Perspective Mapping (IPM) is a geometrical transformation that projects an image to a ground plane surface as shown in Fig. 4. For a flat surface, the perspective effect is removed by transforming the viewpoint from a camera view to a birds eye view. This technique has been used in automotive applications where assumptions about camera pose and a flat world with respect to the street are sufficient [13], [12]. However, even slight deviations in camera inclination and height result in large errors, more advanced, adaptive techniques have been developed which calculate the camera pose online by using the borders of the road or lane markers [14].

However, an unstructured agricultural environments does permit such dynamic techniques and thus, they are either treated as a static scenario, where the camera pose relative to ground surface does not change, or the transformation between the extrinsic and flat plane is calculated dynamically with support of an inertial measurement unit (IMU). The whole IPM for mapping image coordinates $\mathbf{x}_P|_{px} = (u, v, 1)^T$ to surface $\mathbf{x}_{FP}|_m = (x, y, z \equiv 0, 1)^T$ is defined by three parameter transformations: the intrinsic ${}^P\mathbf{T}_C$ from the camera perspective to the camera frame, the extrinsic ${}^C\mathbf{T}_V$ from the camera frame to the vehicle frame, and ${}^V\mathbf{T}_{FP}$ which transforms from the vehicle frame to the flat plane (FP) frame. This leads to

$$\mathbf{x}_P|_{px} = {}^P\mathbf{T}_C \cdot {}^C\mathbf{T}_V \cdot {}^V\mathbf{T}_{FP} \cdot \mathbf{x}_{FP}|_m \quad (7)$$

To build the actual ISM, the image first needs to be classified and then transformed to the flat plane by means of Equation 7 (c.f. Fig. 5).

Values of an ISM are the probability of a grid cell being occupied by a giving classification. As indicated in Fig. 5, the area that is not visible by the camera is set to 0.5 to represent the fact no information is provided for areas that are not visible to the camera. Visible areas with no detections are set below 0.5 to indicate that the area is not expected to

Fig. 6: Input image (left), classification based on semantic segmentation (middle) and corresponding ISM with detection cut-off after class occurrence along the focal axis

Fig. 7: Bounding box detection to ISM

be occupied by the given class. Values above 0.5 indicate that the area is expected to be occupied by the given class.

For detecting flat class elements such as road-lane markings or grass, the IPM algorithm is able to provide good approximations of the actual inverse perspective mapping. Elevated elements violate the IPM ground plane assumption and will stretch elements unnaturally and incorrectly across large areas as indicated in Fig. 4.

To avoid the stretching artifacts of tall objects, different approaches are proposed. A naive approach for pixel based classifiers states that all objects classified as being other than ground are standing perpendicular on the ground. Therefore, one can perform a ray trace along the focal axis and mark all cells behind a detected object as unknown (c.f. Fig. 6) [16], [3].

Another approach generates three dimensional object location hypotheses by first estimating the distance to the corresponding detection. This can be achieved by either using the abovementioned naive approach or using a depth sensor like a stereo camera or LiDAR which is registered to the camera.

Second, when using classifiers like YOLO [17] which offers classified bounding boxes, the four bounding box corners are mapped to real world coordinates using the estimated distance to a detection and the intrinsic camera parameters. The bounding box position and extent are derived in 3D and is represented as depicted in Fig. 7 by cylinder specified by a center, height, and width.

Detections are mapped to values above 0.5 with a Gaussian distribution to indicate the existence of an obstacle with corresponding localization uncertainties. The localization uncertainty for the camera depends on the radial coordinate (distance to the object) and angular coordinate (angle to object), where accuracy degrades with increasing distance and angle. The procedure for converting a 2D bounding box to an ISM using distance estimates is presented in Fig. 7. Using the estimated distance of a detected object and the intrinsic camera parameters, the four bounding box corners are mapped to world coordinates.

Lastly, the concept of contradicting IPM is introduced for crop processing in harvesting scenarios. In comparison

Fig. 8: Simplified error assumption in flat plane assumption according to height

Fig. 9: RGB input image and scanline based classification for crop plane (left), inverse perspective mapping of classification for crop and ground plane (middle) and corresponding fused contradicting ISM (right)

with the abovementioned IPM scenarios, this discrimination is necessary as the camera rectifies no common ground in the lower areas of the image as depicted in Fig. 9 which refutes former assumptions. Neglecting this fact would result in drastically wrong localization of detections, as visualized in Fig. 8, which indicates that the localization error σ_d in depth d depends on the error σ_h of height h as follows:

$$\sigma_d = \frac{d}{h} \sigma_h. \quad (8)$$

If this simple error propagation is applied to a hypothetical example of small crop with for example a height of 0.5 meters and a camera installation height of 1.5 meters where a feature 10 meters away should be mapped, the resultant error is one of 3 meters. Therefore, two flat plane assumptions are calculated, one for the ground and one for the crop height resulting in two different ISMs. These can then be combined by Dempsters rule of combination leading to contradictions [18], which is visualized in Fig. 9. From the emerging contradictions in Fig. 9 (right), it can be seen that vehicle traces appear which are actually the contradicting occlusion in both IPMs.

3) *Ambiguous Sensor Mapping*: Ambiguous sensor readings originate from sensors with very bad angular or distance resolution by definition of the authors. As depicted in Fig. 10 LiDAR systems can achieve very accurate positioning and are therefore the preferred sensors for mapping. However, they are by far the most cost- and power intensive systems. Other sensing techniques are more cost and power efficient but are commonly neglected due to their high noise or inaccuracy. Nevertheless, the authors have demonstrated that even with poorly embedded sensors, sufficient environment detection can be achieved [19] by designing an inverse particle filter which samples from the sensors uncertainty distribution. At present, this technique has only been applied in laboratory conditions and therefore, real agricultural applications remain pending.

Fig. 10: Standard error contour of qualitative sensor cones ($\bullet$: Sensor position, x: Obstacle, -)

Fig. 11: Top view of crop field with an applied inverse sensor model for the cutter bar: gray shaded area being of high probability that the cutter bar has been applied on that region

B. Application Models

Application models are straight forward to implement and only depends on the localizing accuracy. Building such a model is only dependent on the geometrical shape of the agricultural implement. That means on the other hand, that ISM is a static and primitive shape in the local frame of the vehicle which leaves a probabilistic footprint where the implement has been applied to the crop as depicted in Fig. 11. When incorporating inaccurate localization, the shape needs to be transformed accordingly.

C. Map Services

Geodata acquired by satellites, drones, or planes with high recording frequencies as well as its partially free availability, make this information increasingly attractive for agriculture. In this context worth mentioning are the Sentinel program², the hyperspectral system EnMap³, the RapidEye constellation⁴ as well as the start-up companies Skybox Imaging⁵ and Planet Labs⁶. In addition, the release of the long-standing Landsat archive now offers many opportunities for agricultural applications, such as the generation of profit potential maps. There is a trend towards direct access to such data and towards appropriate image excerpts using web servers or APIs. As part of spatial data infrastructures, data (e.g. land and terrain data) are published interoperably and often free of charge via web services. In particular, Annex III of the INSPIRE Directive⁷ requires EU member states to provide data. However, for a precision farming service or a precision farming application further different data sources have to be

²http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview4/

³<http://www.enmap.org/>

⁴<http://blackbridge.com/rapideye/>

⁵<http://www.skyboximaging.com/>

⁶<https://www.planet.com/>

⁷<http://inspire.ec.europa.eu/>

Fig. 12: Classification decomposition of hand labeled orthographic photograph [3]

linked (for example, weather data play a crucial role in most agricultural processes), or complex procedures and algorithms are required to derive the desired information from the data. Subsequent downstream services will continue to play an increasingly important role in agriculture. The European Union, for example, specifically supports the development of such services based on Copernicus data by SMEs. At the endpoint of the downstream services, information products (such as humidity maps, biomass maps and yield forecast maps) are often available, which can be integrated into other applications or devices. The combination and the inclusion of all the information sources and their derivation for the identification of machine parameters is one essential part which can be handled by ISMs. As an example, a static and classified drone image can be easily transferred to a semantic ISM by decomposing all classes and loading the appropriate area during operation (c.f. Fig. 12).

IV. CONCLUSION AND OUTLOOK

The authors have presented an information representation as semantic grids which can be maintained among different modalities and sources. It utilizes the idea of the ISOBUS standard, which was designed with machinery interoperability in mind, and allows every sensory source to publish or access its information in a general grid format. The main aspect of this contribution focused on different techniques, originating from literature, practical experiments, and experience, of actually building these representations.

As the acquisition and localization of data are sufficiently solved, further research will concentrate on planning and control of such diverse data. Furthermore, learning approaches have not been confronted in this application which directly maps a sensor reading to the appropriate locality and probability. These techniques were introduced by Thrun [6] and have been applied by the authors. However, following the engineering path of building inverse sensor models is far more robust and intuitive. At present, only a few approaches are known to the authors and therefore, more applications extending from direct control architectures up to holistic farm management systems are of great interest. Approaching rich control architectures in agricultural environments allows an interesting area of overlap between robotics and Industry

4.0 to emerge, s.t. simultaneously planning and processing. Mathematical frameworks exist, where in agriculture the particular issue will be driven by the information representation and how it is incorporated into environmental processing.

ACKNOWLEDGMENT

This research was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG) and by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems Ost-WestfalenLippe" (it's OWL) and managed by the Project Management Agency Karlsruhe (PTKA).

REFERENCES

- [1] T. Korthals, A. Skiba, and T. Krause, "Evidenzkarten-basierte Sensorfusion zur Umfelderkennung und Interpretation in der Ernte," in *Informatik in der Land-, Forst und Ernährungswirtschaft*, 2016, pp. 15–18.
- [2] —, "Einsatz Event-Basierter Systemarchitektur für Erntemaschinen zur Elektronischen Umfelderkennung," in *74. Tagung LAND.technik*. VDI e.V., 2016.
- [3] M. Kragh, P. Christiansen, T. Korthals, T. Jungeblut, H. Karstoft, and R. N. Jørgensen, "Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture," in *International Conference on Agricultural Engineering*, Aarhus, 2016.
- [4] T. Korthals, J. Exner, T. Schöpping, and M. Hesse, "Semantical Occupancy Grid Mapping Framework," in *European Conference on Mobile Robotics*. IEEE, 2017.
- [5] D. Hähnel, "Mapping with Mobile Robots," Ph.D. dissertation, University of Freiburg, 2004.
- [6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Mass.: MIT Press, 2005.
- [7] C. Stachniss, *Robotic Mapping and Exploration*, 2009.
- [8] R. Garcia, O. Aycard, and T.-d. Vu, "High Level Sensor Data Fusion for Automotive Applications using Occupancy Grids," no. December, pp. 17–20, 2008.
- [9] M. E. Bouzouraa and U. Hofmann, "Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 294–300.
- [10] H. Winner, *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Wiesbaden: Vieweg+Teubner Verlag, 2015.
- [11] A. Elfes, "Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception," in *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990.
- [12] M. Konrad, D. Nuss, and K. Dietmayer, "Localization in digital maps for road course estimation using grid maps," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 87–92, 2012.
- [13] M. Bertozzi and a. Broggi, "Real-time lane and obstacle detection on the GOLD system," *Proceedings of Conference on Intelligent Vehicles*, 1996.
- [14] N. Simond and P. Rives, "Homography from a vanishing point in urban scenes," *International Conference on Intelligent Robots and Systems*, pp. 1005–1010, 2003.
- [15] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.
- [16] S. Kohlbrecher, "Grid-based occupancy mapping and automatic gaze control for soccer playing humanoid robots," ... *Humanoid Soccer Robots* ..., no. October, 2011.
- [17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Cvpr 2016*, pp. 779–788, 2016.
- [18] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.
- [19] T. Korthals, M. Barther, and S. Herbrechtsmeier, "Occupancy Grid Mapping with Highly Uncertain Range Sensors based on Inverse Particle Filters," 2016.

From Plants to Landmarks: Time-invariant Plant Localization that uses Deep Pose Regression in Agricultural Fields

Florian Kraemer Alexander Schaefer Andreas Eitel Johan Vertens Wolfram Burgard

Abstract—Agricultural robots are expected to increase yields in a sustainable way and automate precision tasks, such as weeding and plant monitoring. At the same time, they move in a continuously changing, semi-structured field environment, in which features can hardly be found and reproduced at a later time. Challenges for Lidar and visual detection systems stem from the fact that plants can be very small, overlapping and have a steadily changing appearance. Therefore, a popular way to localize vehicles with high accuracy is based on expensive global navigation satellite systems and not on natural landmarks. The contribution of this work is a novel image-based plant localization technique that uses the time-invariant stem emerging point as a reference. Our approach is based on a fully convolutional neural network that learns landmark localization from RGB and NIR image input in an end-to-end manner. The network performs pose regression to generate a plant location likelihood map. Our approach allows us to cope with visual variances of plants both for different species and different growth stages. We achieve high localization accuracies as shown in detailed evaluations of a sugar beet cultivation phase. In experiments with our BoniRob we demonstrate that detections can be robustly reproduced with centimeter accuracy.

I. INTRODUCTION

The growth of the world population is predicted to not saturate until 2050 and its increasing demands must be met. The agricultural output is expected to rise by 60 % while the available farmland will experience competition from the growing interest in bio-fuels. At the same time, modern agricultural systems are based on untargeted use of pesticides and inorganic nutrients. Their use is already in question nowadays and some demand permacultures and less interference with biological systems that contribute naturally to the agricultural output [3].

The concept of Precision Farming might lower environmental impact while promising affordable, organic production with high yields. Agricultural robots form the backbone of this system. They navigate the fields autonomously and intervene only in a targeted way. For example, weeds are removed precisely by mechanical tools, and fertilizers can be placed selectively for individually monitored plants.

However, a mobile robot must be able to localize itself in order to work autonomously. This is particularly true for agricultural high-precision tasks. Current agricultural robot implementations for fields rely on high-precision global navigation satellite systems (GNSS) to localize precisely. Examples are the system of Bakker et al. [1], the BoniRob

All authors are with the Department of Computer Science at the University of Freiburg, Germany. This work has partially been supported by the European Commission under the grant number H2020-ICT-644227-FLOURISH.

Fig. 1: Deducing the Stem Emerging Points (SEPs) from RGB and NIR input images: We train a fully convolutional neural network (FCNN) to predict a pose likelihood map, from which we extract the SEP positions.

setup without our extension [12], and AgBot/SwarmBot [2], which can also localize temporarily by visual row tracking. But this kind of GNSS requires the setup of a base station. Additionally, poor signals due to forest borders and at longer distances from the base station can lower the accuracy. An alternative to GNSS is that the robot maps its surrounding and localizes itself based on reproducible features. The semi-structured and changing field environment, depicted in Fig. 3, poses a difficult situation when not relying on artificial landmarks. Here, it is highly challenging to reproduce standard image and point cloud features after some days. Therefore, this work proposes time-invariant plant localization from multi-spectral images to lower the dependency on GNSS systems. The estimated locations can be used in robotic mapping and localization approaches as landmarks and allow the robot to pursue its precision tasks.

The key idea for the time-invariant plant localization is that the point of plant emergence never changes. To detect the Stem Emerging Points (SEPs) in images we use a fully convolutional neural network to generate a pose likelihood map, as depicted in Fig. 1. From the pose likelihood map, we extract SEPs for all plant species occurring in sugar beet fields and for different plant growth stages.

Sugar beets, among other root vegetables like carrots, onions, or radish, offer a good experimental environment for agricultural robots, since they are planted in well-traversable rows. Sugar beets grow near the ground and must be monitored for a period of six to eight weeks until the field enters the pre-harvesting phase [4]. The field may contain severe plant overlaps and different weed species, as shown in example field images in Fig. 2. To account for difficult real world scenarios, we extract plant locations not only for sugar beet plants, but also for weeds present in these fields. This allows including all plants in mapping and localization tasks, as well as targeting weed removal actions more precisely. Our evaluation covers the full evolution of a sugar beet field

a) Young, well separated sugar beet b) Intruding grass weed
c) Deformation producing side view d) Overlaps at later growth stages

Fig. 2: Four examples of typically observed plants: a) and b) are taken 6 weeks, c) 7 weeks and d) 8 weeks after germination.

observed from a ground robot. It shows the SEP localization accuracy for different growth stages and weed densities.

We make the following contributions: 1) A SEP localization approach based on pose regression and fully convolutional neural networks, 2) a comprehensive evaluation on hand-annotated image data with several detection measures focusing on long-term robustness and comparing the novel approach with two baseline approaches, 3) a comparison of landmark maps generated by our BoniRob, showing the reliability of SEP landmarks over time, and 4) a dataset for plant localization and evaluation tools, available at: <http://plantcentroids.cs.uni-freiburg.de>.

II. RELATED WORK

Plant localization is useful for agricultural robots in a number of ways. In orchards, the semi-structured environment has been used for mapping and localization purposes: Cheein et al. [5] find olive tree stems by camera and laser scanner detection, which then work as landmarks in a SLAM system.

Several approaches exist for individual field plant detection and SEP localization, many of which are highly use-case adapted and based on heuristics. Midtby et al. [11] follow sugar beet leaf contours to find the SEPs. This works only for very early growth stages without overlapping leaves. Jin et al. [9] focus on corn plants, but use RGB-D data. Again, they make strong model assumptions and follow the plants' image skeletons towards minima in the height domain. Weiss and Biber [15] segment plants based on point cloud clusters and estimate the plant positions at the median points of the clusters. Since this might fail for overlapping plants, Gai et al. [6] extend this approach and follow leaf

Fig. 3: Flourish/BoniRob on a sugar beet field 4 weeks and 8 weeks after germination [4]. The experimental setup is mounted under the robot, shields sunlight and records high-resolution RGB and NIR plant images.

ridges detected in RGB-images to the center. Their approach only works for specific crop types and their ground-plant segmentation fails for early growth stages. Most similar to our work, Haug et al. [7] perform machine learning-based SEP localization in an organic carrot field. They classify key points sampled equally-spaced from plant areas in NDVI images. As a classifier, they employ a Random Forest with hand-crafted statistical and geometrical features for different patch sizes. Their evaluation over 25 images takes one point in time into account and they find weaknesses of the classifier for plant overlaps and locally misleading plant shapes. Our work overcomes these by supplying a broader field of view to the classifier and by avoiding hand-crafted features that might only work for limited scenarios and plant species.

The field of image landmark localization is important to our pose regression approach. Tompson et al. [13] estimate likelihood maps of hand joint locations based on depth images. However, without any upsampling operation in their network architecture, they generate a coarse output of less than a fifth of the input size. Zhang et al. [16] overcome this drawback by using a fully convolutional neural network (FCNN). This allows them to generate likelihood maps with input image resolution.

III. APPROACH

We localize the SEPs in multi-spectral images by solving a pose regression problem with a neural network. FCNNs can learn in an end-to-end manner to output a pose likelihood map with high-resolution output. FCNNs generally feature a contracting path similar to commonly used image classification networks. In addition, FCNNs upsample the coarsely-resolved bottleneck feature maps and localize features in an expanding path [10]. Formally, the problem is to assign every pixel a likelihood $p : \mathbf{x} \mapsto \mathbb{R}$ depending on the distance to the next SEP location. We generate the ground truth values as $r : \mathbf{x} \mapsto [0, 1]$ to obtain a normalized network output that allows postprocessing it as likelihood maps.

In order to compute the pixel-wise ground truth regression values r , we annotate the plant locations for every input image by hand, following a set of rules: For rosette-type plants, like sugar beet, the SEPs can be determined efficiently by following the leaf stems. For herbaceous plants, like

Fig. 4: An example of the likelihood map ground truth generation taken from the sixth week after sugar beet germination. The RGB image (left) shows two sugar beet and several grass weed plants. The likelihood values (right) have been computed with $\sigma = 50$ px for visualization purposes.

grass weeds, the region of emergence can be arbitrarily shaped. In these cases, the emergence region is marked with a polygon contour. A post-processing step then estimates the SEP coordinates as the region’s center of mass. Cases in which the human expert is uncertain are filtered out.

From the ground truth SEP coordinates in every image set, we generate a likelihood map that is used for training the network. First, a distance transform produces a map of distances to the closest ground truth SEP position $\mathbf{x}_{\text{SEP}}$:

$$d(\mathbf{x}) = \min_{\mathbf{x}_{\text{SEP}} \in \text{GT}} \|\mathbf{x} - \mathbf{x}_{\text{SEP}}\|_2 \quad (1)$$

Next, we transform the resulting distance map to a Gaussian ground truth likelihood map using

$$r(\mathbf{x}) = \exp \frac{-d(\mathbf{x})^2}{2\sigma^2}, \quad (2)$$

where σ is a parameter that determines the decay of likelihood with increasing distance. Fig. 4 shows a corresponding likelihood map. We use the validation image set to tune σ for the best precision-recall relation. In our use-case, we achieve good results for σ values from 15 px to 19 px.

We use the L2-Norm, or Euclidean distance, of the network’s regression score p and the ground truth r to compute the loss during network training. The weight optimization by stochastic gradient descent can accordingly be formulated as minimizing the energy function for all pixel locations Ω :

$$E = \sum_{\mathbf{x} \in \Omega} \|r(\mathbf{x}) - p(\mathbf{x})\|_2^2 \quad (3)$$

Solving the pose regression problem does not depend on a specific fully convolutional neural network architecture. Instead, architectures commonly used for semantic segmentation problems can be adapted to our problem. We employ the architecture of MultiNet (the expert network from [14]), which has been successfully deployed with the restrictions of mobile robotics. The low number of parameters, when compared to e.g. FCN8 [10], fits the memory capabilities of current embedded devices and allows processing our image stream in real-time at about 4 Hz. Additionally, the number of parameters hinders overfitting, when working with small data sets. MultiNet extends the ResNet50 architecture in

Fig. 5: Trained network likelihood map output (right), RGB image depicted for input visualization (left): Big grass weeds produce spread-out likelihood peaks, which are more spiked for rosette-type plants. Even hidden, small grass weeds are detected (see bottom image, upper left corner).

several domains: The first feature maps are kept at a higher resolution since they need to be reused in the expanding path later on. Additionally, it extends the residual blocks to two parallel convolutions, one of which has a wider kernel, to learn multi-scale features. We use a Euclidean loss layer and implement the neural network in the Caffe framework [8]. The image input in the experiments is downsampled by a factor of 2 to capture plants entirely. We stack the RGB and NIR images channel-wise and crop them to fit the network input size of $4 \times 512 \times 512$ (channels×width×height).

Examples for a trained network’s output can be seen in Fig. 5 where the likelihood map contains flat basins for regions distant to SEPs. Furthermore, high likelihood regions correspond to the Gaussian likelihoods given in the ground truth, but they are usually more elongated. For herbaceous plants, the peaks can be arbitrarily shaped. Generally, the peaks reach different likelihood values. Their height can be seen as an indicator for the network’s confidence or pseudo-probability for a detection.

The final step of our SEP localization approach is to extract the SEP coordinates from the likelihood map. First, we segment the likelihood map into basins and peak regions using Otsu’s method. SEP locations are then determined as the center of mass for every likelihood peak region. The confidence score for a detection is deduced as a region’s average likelihood. Scoring by the average value of a peak region reflects that detections are less confident when the likelihood peak is spread out. This way of confidence scoring performs better in our tests than relying on a peak region’s maximum likelihood value.

IV. EXPERIMENTS

First, we compare our pose regression approach with two baselines using a hand-annotated field image test data set. Then, we evaluate the reliability of the our approach, when plant locations are used as landmarks in a field mapping task.

As a first baseline, we re-implement the approach of Haug et al. [7]. We found that it copes better with the big variances in our dataset when the feature set and classifier are adapted. Instead of hand-crafted features, we use HOG-features of NDVI images. We sample the key points with a stride of 10 px and derive features by using patch sizes of 40 px, 80 px, 160 px and 320 px. The classification of every key point's features is performed by an SVM with a radial basis function kernel.

As a second baseline, we extend the first baseline to also show the performance of a state-of-the-art classifier in this scenario. Here, instead of coarsely sampled key points, we use the pixel-wise classifications of a segmentation FCNN. The FCNN learns to classify for every pixel whether it belongs to the Background, Vegetation or Stem Emerging Region class. The Vegetation class is introduced to enable the learning of richer features through finer class partitioning. Also, the intermediate step of estimating regions of stem emergence rather than one stem emerging *point* per plant has been introduced to keep class imbalances low. The SEPs are then determined in a post-processing step as the center of mass for a given stem emerging connected component in the segmentation mask. We estimate a confidence score for every region/SEP as the mean of pixel-wise network score differences between the stem emerging and the other classes. In contrast to our regression approach, here, we annotate pixel-wise to generate ground truth segmentation masks.

A. Test Image Set Comparison

In this experiment, we evaluate the two baselines and our approach, using hand-annotated ground truth images. All data is sampled from the Flourish Data Acquisition Campaign 2016 [4] (see Fig. 3). Among other sensors, it features high-precision GNSS, odometry and a multi-spectral camera. The camera records high-resolution image streams of the traversed sugar beet rows with an RGB spectrum of 400 nm to 680 nm and an NIR spectrum of 730 nm to 950 nm. We use the aligned image channels to create a dataset. Accordingly, we label several sequences of about 300 images each, taken from different rows and on different days. We denote them X-Wy where X is a unique identifier and y the number of weeks after sugar beet germination. We use entire sequences to split the dataset into training and test partitions, since the images sequences are locally correlated.

To evaluate the output of an approach, we compute the distance $d = \min_{i \in \text{GT}} \|\mathbf{x}_i - \mathbf{x}\|_2$ between the approaches' SEP detections and the ground truth SEP locations. Following standards in detection evaluation, we sort them by their confidence score in descending order. Subsequently, we check whether the distance error lies below a threshold and only accept the highest scored detection per ground truth label. The threshold is varied from 6 px to 18 px,

Fig. 6: Precision-recall relationship of SEP localizations derived for three test image sets taken 4, 5 and 8 weeks after sugar beet germination. The B-W5 data set additionally has a very high weed density. Acceptance ranges are 6 px and 18 px. The first baseline is abbreviated as SVM, the second as Segm. and our approach as Regr.

which corresponds to a ground resolution of about 3.6 mm to 8.4 mm at the image nadir (neglecting ground unevenness). The binary result of being accepted or not allows us to use the quantitative measures of precision and recall. In order to depict the distance errors better, we compute their mean for all accepted detections, the mean accepted distance (MAD), and associate it with the Average Precision (AP). In this case, we accept a detection if it is closer than the minimal plant distance in the dataset, 20 px, which we see as a natural boundary for acceptance.

All approaches of this comparison need to be trained. The SVM baseline is trained on the H-W5 sequence only, since more samples make the fitting process problematic. The segmentation (second baseline) and regression networks are trained on the G-W5, H-W5, I-W6, J-W8 and K-W8 sequences, which add up to 1398 images. They cover four weeks of sugar beet field evolution and have a varying weed content. We augment the image set by systematically sampling 8 rotated and 2 mirrored versions as well as 4 randomly cropped versions of the input images. Then, we train both networks for 50 000 iterations and a mini-batch size of 4 from scratch.

Dataset	Approach	AP	MAD [px]
A-W4	SVM	0.515	10.12
	Segm.	0.798	5.82
	Regr.	0.796	5.90
B-W5	SVM	0.054	10.58
	Segm.	0.285	8.43
	Regr.	0.432	7.96
C-W8	SVM	—	—
	Segm.	0.447	10.98
	Regr.	0.513	10.05

TABLE I: Average Precision (AP) and Mean Accepted Distances (MAD) for the test image set comparison. 1 pixel corresponds to about 0.6 mm ground resolution. The first baseline is abbreviated as SVM, the second as Segm. and our approach as Regr.

Early Growth Stage: The A-W4 sequence of 100 images was recorded when most sugar beet plants were in a growth stage of two to four leaves and one plant’s leaves rarely overlap. The individuals are usually well separated by soil. Also, only few weeds are visible such that the ground truth contains 166 instances of sugar beet and only 10 instances of weed plants. Fig. 6 shows large differences in precision and recall between the SVM baseline and the neural network based approaches. However, the network approaches can not clearly be separated. Tab. I shows both neural network approaches performing very similar with about 80 % AP and only 5.9 px MAD.

Weed-Affected: The B-W5 sequence is an example for a more weed affected situation. The ground truth features 330 instances of sugar beet and 1427 instances of weed plants. This high weed density can occur due to non-effective pest control or on the borders of a field. Accordingly, the images contain a higher visual variance, as different plant species occur and leaves often overlap. Here, the precision recall relationships in Fig. 6 separate the approaches better. Our approach can handle the high number of weed plants better than the baselines for both distance thresholds. As described in Tab. I, the pose regression approach reaches as low as 7.96 px in MAD, which is a relative improvement of 5.9 % over the segmentation network approach. More importantly, it performs a relative 51.6 % better in AP. The SVM baseline underperforms in this scenario.

Late Growth Stage: The C-W8 sequence was recorded at the end of the crop management phase. Pest control has been successful for our test field so that this sequence features 369 instances of sugar beet and only 5 instances of weed plants (see Fig. 2d for an example image). The SVM baseline would need a separate training specifically for this growth stage. Also, the neural network baseline performed generally better in the other test sequences, for which reason we exclude the SVM for the late growth stage. The results in Fig. 6 show that the regression approach achieves higher precision rates. Tab. I shows the mean accepted distance errors where regression ranks again before segmentation. It performs a relative 14.7 % better in AP and has a 9.3 % relative improvement in mean accepted distance. The lower accuracy compared to the A-W4 sequence, with comparably low amounts of weed plants, could be explained by the older sugar beet plants containing less texture in their center.

Fig. 7: 4 day comparison between geo-referenced images taken in the scope of the E-W5 (left) and F-W6 sequences. Detections from regression approach (circles). One weed has died and can not be matched. Only little growth is observable, the soil structure is still the same.

B. Landmark Reproducibility

This experiment evaluates the pose regression SEP localization approach in the robotic use-case by comparing two mapping runs of the agricultural robot. Effectively, this shows how system uncertainties influence the geo-referencing and reproducibility of SEP landmarks. We evaluate the long-term reliability by varying the time between two mapping runs, taking the data again from [4].

The pipeline to produce a metric SEP landmark map from the image detections is as follows: A high-precision GNSS is employed for all mapping runs to assign a reference pose to every image. Empirical evaluation of the GNSS showed that 50 % of position errors ranged below 3.1 mm and 95 % below 7.9 mm. Additionally, we improve the estimated positions by averaging the GPS fixes during constant robot motion and fuse odometry measurements in an Unscented Kalman Filter. We use the Kalman filter’s pose estimations as the origin for a pinhole camera projection (neglecting ground unevenness). As the image recording produces overlapping image regions, we merge landmarks in the metric map if they are closer than the GNSS system’s 95 % error range.

The evaluation is then done by matching the landmarks for minimal Euclidean distance in two overlaid metric landmark maps. From the resulting error distance distribution we choose the 3σ -range (about 70 mm in both experiments) as an acceptance range for a match. We show figures for precision and recall rates given this acceptance range and evaluate the continuous distance errors.

Evolution over 4 Days: The E-W5 has been recorded 4 days before the F-W6 sequence in the same traversing direction. Still, Fig. 7 gives an impression of how different perceptions of the field environment can be even for a short time-span: The ground might still show the same soil structure, but plants can hardly be associated by their appearance. The leaf angle distribution can change and the earlier sequence generally shows more shimmering leaf surfaces due to more moisture on that day. The error distributions in x and y directions have a very low bias ($\mu_x = 1.2$ mm, $\sigma_x = 16.7$ mm, $\mu_y = -2.9$ mm, $\sigma_y = 13.3$ mm), which shows that the underlying pose-estimation is working properly.

As a result, 302 plants are matched successfully and only 12 outliers are found. Since the second mapping run finds 345 SEPs, the corresponding recall rate is 96.2 % and precision equals 87.5 %. The resulting errors with a mean of $\mu = 18.41$ mm are depicted in Fig. 9. Note that this is an

Fig. 8: Geo-referenced images with a 28 day difference and severe changes in plant and soil appearance. Detections from regression approach (circles).

Fig. 9: Comparison of two metric landmark maps for a 4 day and 28 day time span. The curves show the euclidean distance errors if every SEP landmark from an initial map is matched with the closest one in a later map.

upper bound, as it also contains the GNSS uncertainty.

Evolution over 28 Days: The D-W4 sequence shows images from an early growth stage of two main leaves. The C-W8 sequence has been recorded 28 days later in opposite driving direction, shortly before the pre-harvesting phase of the field starts. Accordingly, the plants have grown a lot, overlapping plants are common and the leaves extend a lot more as can be seen in the comparison in Fig. 8.

The evaluation shows 34 outliers for 291 SEPs landmarks in the earlier map. This corresponds to a recall rate of 88.3 % and precision rate of 92.8 %. The distance errors of matched landmarks are shown in Fig. 9 ($\mu = 20.8$ mm).

This map comparison shows slightly higher mapping errors than the experiment over 4 days and at earlier growth stages did. This corresponds to the image comparison experiment where later growth stages produced larger distance errors. We conclude that the landmark matching over this long period shows good reliability and accuracy.

V. CONCLUSIONS

We present an approach for time-invariant plant localization on RGB and NIR image data. Based on an FCNN and pose regression, it outperforms state-of-the-art approaches. We show that the resulting localizations, when compared with an image ground truth, achieve high precision and recall rates, while mean distance errors can be expected to be as low as 5 px to 11 px. The image data set, ground truth annotations, and evaluation tools are publicly available. For the agricultural robot scenario, we show that the SEP detections can be used as anonymous landmarks: We were able

to reproduce hundreds of landmarks with precision/recall rates of about 90 % and mean errors as low as 20 mm. For future work, we expect better regression network results if the likelihood decay parameter is chosen depending on the plant size. This might increase accuracies for plants that offer less texture in the plant center, like sugar beets in late growth stages. Our approach will also benefit from advances made in the field of FCNNs. We will implement a robot localization approach based on our work, which enables an agricultural robot to navigate autonomously without the disadvantages of a high-precision GNSS.

REFERENCES

- [1] Tijmen Bakker, Kees van Asselt, Jan Bontsema, Joachim Müller, and Gerrit van Straten. Autonomous navigation using a robot platform in a sugar beet field. *Biosystems Engineering*, 109(4):357 – 368, 2011.
- [2] David Ball, Patrick Ross, Andrew English, Peter Milani, Daniel Richards, Andrew Bate, Ben Upcroft, Gordon Wyeth, and Peter Corke. Farm workers of the future: Vision-based robotics for broad-acre agriculture. *IEEE Robotics & Automation Magazine*, 2017.
- [3] Riccardo Bommarco, David Kleijn, and Simon G Potts. Ecological intensification: harnessing ecosystem services for food security. *Trends in ecology & evolution*, 28(4):230–238, 2013.
- [4] Nived Chebrolu, Philipp Lottes, Alexander Schaefer, Wera Winterhalter, Wolfram Burgard, and Cyrill Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research (IJRR)*, 2017.
- [5] F. Auat Cheein, G. Steiner, G. Perez Paina, and R. Carelli. Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. *Computers and Electronics in Agriculture*, 78(2):195 – 207, 2011.
- [6] Jingyao Gai, Lie Tang, and Brian Steward. Plant localization and discrimination using 2D+3D computer vision for robotic intra-row weed control. In *2016 ASABE Annual International Meeting*, page 1. American Society of Agricultural and Biological Engineers, 2016.
- [7] S. Haug, P. Biber, A. Michaels, and J. Ostermann. Plant stem detection and position estimation using machine vision. In *Workshop Proceedings of IAS-13, ser. 13th Intl. Conf. on Intelligent Autonomous Systems*, pages 483–490, 2014.
- [8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- [9] Jian Jin and Lie Tang. Corn plant sensing using real-time stereo vision. *Journal of Field Robotics*, 26(6-7):591–608, 2009.
- [10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [11] Henrik S Midtby, Thomas M Giselsso, and Rasmus N Jørgensen. Estimating the plant stem emerging points (PSEPs) of sugar beets at early growth stages. *Biosystems engineering*, 111(1):83–90, 2012.
- [12] Arno Ruckelshausen, Peter Biber, Michael Dorna, Holger Gremmes, Ralph Klose, Andreas Linz, Florian Rahe, Rainer Resch, Marius Thiel, Dieter Trautz, et al. BoniRob—an autonomous field robot platform for individual plant phenotyping. *Precision agriculture*, 9(841):1, 2009.
- [13] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.
- [14] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, Singapore, 2017.
- [15] Ulrich Weiss and Peter Biber. Plant detection and mapping for agricultural robots using a 3D Lidar sensor. *Robotics and autonomous systems*, 59(5):265–273, 2011.
- [16] Ning Zhang, Evan Shelhamer, Yang Gao, and Trevor Darrell. Fine-grained pose prediction, normalization, and recognition. *arXiv preprint arXiv:1511.07063*, 2015.

Optimising Realism of Synthetic Agricultural Images using Cycle Generative Adversarial Networks.

Ruud Barth¹, Joris IJsselmuiden², Jochen Hemming¹ and Eldert J. van Henten²

Abstract—A bottleneck of state-of-the-art machine learning methods, e.g. deep learning, for plant part image segmentation in agricultural robotics is the requirement of large manually annotated datasets. As a solution, large synthetic datasets including ground truth can be rendered that realistically reflect the empirical situation. However, a dissimilarity gap can remain between synthetic and empirical data by incomplete manual modelling. This paper contributes to closing this gap by optimising the realism of synthetic agricultural images using unsupervised cycle generative adversarial networks, enabling unpaired image-to-image translation from the synthetic to empirical domain and vice versa. For this purpose, the *Capsicum annuum* (sweet- or bell pepper) dataset was used, containing 10,500 synthetic and 50 empirical annotated images. Additionally, 225 unlabelled empirical images were used. We hypothesised that the similarity of the synthetic images with the empirical images increases qualitatively and quantitatively when translated to the empirical domain and investigated the effect of the translation on the factors color, local texture and morphology. Results showed an increased mean class color distribution correlation with the empirical dataset from 0.62 prior and 0.90 post translation of the synthetic dataset. Qualitatively, synthetic images translate very well in local features such as color, illumination scattering and texture. However, global features like plant morphology appeared not to be translatable.

I. INTRODUCTION

A key success factor of agricultural robotics performance is a robust underlying perception methodology that can distinguish and localise object parts [1], [2], [3]. In order to train state-of-the-art machine learning methods that can achieve this feat, large empirical annotated datasets are required. Synthetic data can help bootstrapping such methods in order to reduce the required amount of empirical data [4]. However, a gap in realism remains between the modelled synthetic data and the empirical images, plausibly restraining synthetic bootstrapping performance.

In this paper we report on optimising the realism of synthetic images modelled from empirical data [5]. The long term goal of this research is to improve plant part segmentation performance by synthetically bootstrapped deep convolutional neural networks (CNN) [4]. For the intermediate goal presented here, we hypothesise the dissimilarity between corresponding synthetic and empirical images can

be qualitatively and quantitatively reduced using unpaired image-to-image translation by cycle-consistent adversarial networks (cGAN) [6].

Convolutional neural networks currently show state-of-the-art performance on image segmentation tasks [7], [8], [9]. However, CNNs require large annotated datasets on a per-pixel level in order to successfully train the deep network. Moreover, in agriculture the high amount of image variety due to a wide range of species, illumination conditions and morphological seasonal growth differences, leads to an increased dataset size dependency. Satisfying this requirement can quickly become a bottleneck for learning.

One solution is to bootstrap CNNs with synthetically generated images including automatically computed ground truth [10], [11]. Consequently the bootstrapped network can be fine-tuned with and applied to empirical images, resulting in increased performance over methods without synthetic bootstrapping [4].

Previously we have shown methods to create such a dataset by realistically rendering 3D modelled plants [5]. Despite intensive manual optimisation for geometry, color and textures, we have shown that a discrepancy remains between synthetic and empirical data. Although this dataset can be used for successful synthetic bootstrapping and learning empirical images, there remains a gap between achieved performance and theoretical optimal performance [4].

Recently, the advent of generative adversarial networks (GAN) introduced another method of image data generation [12]. In GANs two models are trained simultaneously and adversarially: a generative model G and a discriminative model D . The generative model's goal is to capture the feature distribution of a dataset by learning to generate images thereof from latent variables (e.g. random noise vectors). The discriminative model in turn evaluates to what extent the generated image is a true member of the dataset. In other words, model G is optimised to trick model D while model D is optimising to not get fooled by model G . As both models can be implemented as CNNs, the error of both models can be back-propagated to minimise the loss of both models simultaneously. The result after training is a model G that can generate new random images highly similar to the learned dataset.

In later approaches, GANs were also conditioned by additional input images; both the generator and discriminator observe an input image [13]. The discriminator's goal is to compare such pairs on coherency of their co-occurrence whereas the generator aims to create an image-to-image translation from the conditional image to an image adhering

¹Ruud Barth and Jochen Hemming are with Wageningen University & Research, Greenhouse Horticulture, P.O. Box 644, 6700 AP, Wageningen, The Netherlands. ruud.barth@wur.nl and jochen.hemming@wur.nl

²Joris IJsselmuiden and Eldert J. van Henten are with Wageningen University & Research, Farm Technology Group, Droevendaalsesteeg 1, 6708 PB, Wageningen, The Netherlands. joris.ijsselmuiden@wur.nl and eldert.vanhenten@wur.nl

to the same coherency of the other pairs in the dataset. The result is a generator G that can translate images from one domain X (e.g. summer photographs) to images in another Y (e.g. winter photographs), formally notated as $G : X \rightarrow Y$.

A requirement for conditional GANs is the availability of geometrically paired images, but for many tasks these will not be available. For example in agriculture, obtaining a geometrically paired synthetic image of an empirical scene would defeat the purpose of circumventing manual annotation time. Instead, only unpaired synthetic images can be generated without additional manual efforts.

A recent approach aimed to dissolve this requirement by investigating unpaired image-to-image translation. In cycle-consistent adversarial networks (Cycle-GAN) [6], a mapping $G : X \rightarrow Y$ is learned whilst also an inverse mapping $F : Y \rightarrow X$. Both domains X and Y have corresponding discriminators D_X and D_Y . Hence, D_X ensures G to translate X similar to Y whilst D_Y safeguards a indistinguishable conversion of Y to X .

However since the domains are unpaired, the translation at this point does not guarantee that an individual image $x \in X$ is mapped to an geometrically similar image in domain Y (or $y \in Y$ to X). This because there are boundless mappings from x that result in the same target distribution of Y . Therefore the mapping needs to be constrained in a way the original geometry is maintained.

To achieve that, a cycle consistency loss was added to further regularise the learning. Given a sample $x \in X$ and $y \in Y$, a loss was added to the optimisation such that $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Hence the learning was therefore constrained by the intuition that if an image is translated from one domain to the other and back again, an equal image should be retrieved. This forces the generators G and F to achieve unpaired geometrically consistent image-to-image translation from one domain to the other and vice versa.

The key contribution of our research presented here is that we show that Cycle-GAN can translate agricultural images in the synthetic domain to images in the empirical domain, to improve the realism of the synthetic data and close the dissimilarity gap further. Hence, this will increase the amount of realistic training data for machine learning computer vision methods. This can be seen as an important step towards improved sensing for agricultural robotics by minimising the dependency on manual annotated datasets.

The scope of this paper was limited to results of the translation and a similarity comparison, whereas future research will investigate the impact of translated synthetic images on learning.

II. MATERIALS

A. Image dataset

The unpaired image dataset of *Capsicum annuum* (sweet- or bell pepper) was used [5] that consists of 50 empirical images of a crop in a commercial high-tech greenhouse and 10,500 corresponding synthetic images, modelled to approximate the empirical set visually and geometrically. In both sets 8 classes were annotated on a per-pixel level,

Fig. 1: Uncropped examples of empirical and synthetic color images (2nd and 3rd image respectively) and their corresponding ground truth labels. Class labels: ● background, ● leaves, ● peppers, ● peduncles, ● stems, ● shoots and leaf stems, ○ wires and ○ cuts where pepper where harvested.

either manually for the empirical dataset or automatically for the synthetic dataset. In Figure 1 examples of images in the dataset are shown. The dataset was publicly released at: <http://dx.doi.org/10.4121/uuid:884958f5-b868-46e1-b3d8-a0b5d91b02c0>

Both synthetic and empirical images were cropped to 424x424 pixels to exclude the robot end-effector’s suction cup in the image, because initial image-to-image translation experiments showed that this hardware was replicated undesirably in other parts of the image. This is in line with comments of the methodology of the Cycle-GAN authors; color and texture translation often succeeds though large geometric changes are translated with less success.

From the *Capsicum annum* dataset, the synthetic images 1-1,000 were used for training and the remainder for testing. For the empirical images, an unreleased and unlabelled dataset consisting of 225 images was used, of which a random subset was previously labelled and included in the *Capsicum annum* dataset. 175 Images of this set were used for training and 50 for testing.

B. Software

The Berkeley AI Research (BAIR) laboratory implementation of unpaired image-to-image translation using cycle-consistent adversarial networks was used [6].

III. METHODS

A. Image-to-Image Translation

Hyper-parameters of the Cycle-GAN were manually optimised. The full resolution of the cropped images was used.

The number of generative and discriminative filters were set to 50 and the learning rate was set to 0.0002 with an ADAM [14] momentum term of 0.5. The basic discriminator model was used, whereas for the generative model the RESNET [15] 6 blocks model. Weights for the cycle loss were set to 10 for each direction.

B. Quantitative Translation Comparision

Although the success of the translation will already be quantitatively captured by the adversarial loss, this measure is biased and mathematically obfuscated. It is interesting to look more specifically at key image features like color.

For this purpose, we compared for each class the synthetic color distribution prior and post translation with those of the empirical distribution. We hypothesise the color difference post translation will be reduced. To determine this quantitatively, the correlation between color distributions was obtained.

The color spectrum of each class was obtained by first transforming the color images to HSI colorspace. The hue channel in the transformed image represented for each pixel which color was present, irregardless of illumination and saturation intensity. The histogram of this channel was then taken to count the relative color occurrence per class.

IV. RESULTS

In Figure 3 the results of the image-to-image translations are shown. The second column is of most interest to our research, as it shows synthetic images which are translated to the empirical domain. However, as reference also the

Fig. 2: Color distributions (discretized to 256 values in the hue channel) per class of the synthetic, empirical and synthetic translated to empirical images. Integral per distribution amounts to 1.

	backgr.	leafs	peppers	peduncles	stems	shoots	wires	cuts	avg.
correlation(synthetic, empirical)	0.25	0.78	0.42	0.93	0.76	0.83	0.45	0.48	0.62
correlation(synthetic→empirical, empirical)	0.86	0.94	0.93	0.93	0.92	0.98	0.81	0.79	0.90

TABLE I: Color distribution correlation per class between the synthetic and synthetic translated to empirical with the empirical image dataset.

Fig. 3: Image-to-image translation examples using Cycle-GAN. Source images prior translation are shown in the outer columns; synthetic images (left) and empirical images (right). The second column shows translated synthetic images to empirical ones and the third column shows empirical images translated to synthetic ones.

translation from empirical to the synthetic domain is shown in the third column.

The color distributions for each class for the synthetic, empirical and synthetic→empirical translation are shown in Figure 2. The corresponding correlations between the empirical images and the synthetic or synthetic→empirical images are shown in Table I.

V. DISCUSSION

Qualitative evaluation of the results subjectively showed a remarkable feat of translated synthetic images to empirical looking images and vice versa. Notably the scattering of illumination and color of each plant part were converted realistically. It appears the model learns to distinguish plant parts without any supervised information, as often the (partially) ripe and unripe fruit are translated to the other domain with altered maturity levels. A difference in camera focus seemed translated properly, indicating that local features (e.g. edge blur and texture) can be mapped.

Artifacts do arise however, especially the translation to overexposed area's like sunshine or fruit reflections. The explanation might be that the model cannot generate this information correctly because information beyond the maximum range of the image was previously collapsed into a single the maximum value (e.g. 255) of the image. Furthermore, a faint checker-like texture seems to have been added to the translated local textures.

Larger morphological features (e.g. plant part shape and their geometry) were not translated, indicating a limitation of the Cycle-GAN approach. This suggests that the source synthetic data should be geometrically highly similar to the empirical situation, for a realistic translation to succeed.

If the translated images are later to be used for supervised learning, the morphological structure should be retained however. This because the underlying ground truth cannot be translated correspondingly, as no supervision is used in Cycle-GAN.

The method is not suited when one image set contains additional parts absent in the other set, e.g. the inclusion of a suction cup in our earlier experiments.

In Figure 2 the translation effect on color distribution can be seen for each plant part and background. Quantitatively, the mean correlation of the color distributions increased of the synthetic data with the empirical data prior (0.62) and post translation (0.90), confirming our hypothesis that color difference post translation with the empirical data is reduced.

VI. CONCLUSION

This work contributed to the field of agricultural robotics by providing a method for optimising realism in synthetic training data to improve state-of-the-art machine learning methods that semantically segment plant parts.

Our hypothesis that dissimilarity between synthetic and empirical images can be reduced by using adversarial generative networks (e.g. Cycle-GAN) has been confirmed qualitatively and quantitatively by increasing the color distribution

correlation with empirical images prior and post translation of synthetic images.

Future research will investigate the impact on learning with empirically translated synthetic images. Due to the improved realism, it might become feasible to circumvent the need of any manual annotation of empirical data by solely bootstrapping on translated synthetic data, without the requiring empirical fine-tuning.

ACKNOWLEDGMENT

This research was partially funded by the European Commission in the Horizon2020 Programme (SWEEPER GA no. 644313).

REFERENCES

- [1] C. Bac, J. Hemming, and E. van Henten, "Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper," *Computers and Electronics in Agriculture*, vol. 96, pp. 148 – 162, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168169913001099>
- [2] A. Gongal, S. Amaty, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," *Computers and Electronics in Agriculture*, vol. 116, pp. 8 – 19, 2015.
- [3] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014. [Online]. Available: <http://dx.doi.org/10.1002/rob.21525>
- [4] R. Barth, J. IJsselmuiden, J. Hemming, and E. van Henten, "Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation," 2017, submitted to the Journal of Computers and Electronics in Agriculture.
- [5] R. Barth, J. IJsselmuiden, J. Hemming, and E. J. van Henten, "Data synthesis methods for semantic segmentation in agriculture: a capicum annum dataset," 2017, submitted to the Journal of Computers and Electronics in Agriculture.
- [6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.
- [7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," *ArXiv e-prints*, June 2017.
- [8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015. [Online]. Available: <http://arxiv.org/abs/1412.7062>
- [10] F. Dittrich, H. Woern, V. Sharma, and S. Yayilgan, "Pixelwise object class segmentation based on synthetic data using an optimized training strategy," in *Networks Soft Computing (ICNSC), 2014 First International Conference on*, Aug 2014, pp. 388–394.
- [11] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez, "The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes," 2016.
- [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [13] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: <http://arxiv.org/abs/1611.07004>
- [14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: <http://arxiv.org/abs/1512.03385>

Tracking footprints for agricultural applications: a low cost lidar approach

Thibault Tourrette¹, Roland Lenain¹, Raphael Rouveure¹ and Thomas Solatges²

Abstract—The accuracy of a mobile robot is an important topic, which is subjected to an important attention. It is particularly crucial when considering off-road applications such as agricultural robot, since it rises questions for both localization and control. A possible way to avoid the use of expensive sensors (such as RTK-GPS) or building specific landmarks, lies in the exploitation of tire footprints, let in the soil by previous actions achieved manually or autonomously. These footprints are particularly detectable by several kind of sensors. This paper presents a method for tracking footprints with a low cost lidar sensor. The proposed way is to detect an expected footprint template and then aggregate successive detections to construct a 3D map. This map finally permit to compute a control law for path tracking regarding mobile robot position compared with footprint location.

I. INTRODUCTION

Agricultural robotics is a promising field of application, as it permits reducing arduousness, and risks for humans, while ensuring a high level of production [3]. Another interest of agricultural robotics is to reduce chemical products usage. Currently, mechanical solutions are often too expensive because they need more time than chemical solutions, in that way an autonomous robot could appear as a good idea. As a result, research is paying more and more attention to this area, leading to important advances in terms of perception and control [1]. In particular, the problem of path tracking is deeply studied. It is particularly important considering farm applications, since robots have to move following the vegetation rows. Many solutions are based RTK-GPS, since it supplies a highly accurate absolute localization [2]. Nevertheless, the use of an RTK-GPS is still quite expensive and is not always reliable, since GNSS signals are disturbed close to building [6], and cannot be used inside greenhouses. Moreover, when working on a field, the task is related to the vegetation and not necessarily to an absolute position.

As a result, alternative solutions may be used, based on exteroceptive sensors such as vision [7] [8]. Nevertheless, such approaches rely on lightning conditions, which cannot be controlled in the different applications of agriculture, especially in open fields. Lidar then raises as an interesting solution, as it is less sensitive to light conditions, while remaining relatively cheap. Several algorithms have been developed, especially based on simultaneous localization

and mapping [4] [11] or occupation grid [9]. This implies the use of a map, requiring the detection of specific landmarks [5], which cannot be necessarily visible, because of vegetation changes.

In this paper a lidar sensor is used in order to detect and follow the footprints let by the wheels during previous field operations. These footprints indeed remain unchanged and detectable regardless of the vegetation growth, and are supposed to be an accurate reference for the robot position. The idea is to take part of the inclination of a lidar settled in front of the robot and inclined with respect to the soil (as depicted on Fig. 1). Such an inclination permits to anticipate row curvature, and to build a smoothed local trajectory.

Fig. 1. Illustration of footprint tracking using lidar

The algorithm proposed in that paper is based on several steps. First, the detection of a footprint is performed in the local framework of a laser scan, using a correction of global shape of the soil. This is achieved thanks to a correlation method, with respect to an expected shape with given properties. This permits to define a point belonging to a global trajectory. This latter is then built by the aggregation of such points, obtained by the detection achieved in the successive lidar scans. This accumulation of points is based on the robot odometry, the drift of which is not significant, as only a local trajectory is required. Thanks to these points, a local trajectory is computed on-line in order to obtain all the required variables to feed a simple path tracking control law. Experimental results show the efficiency of the approach through motion in actual field, using footprints on the soil, let by a farm tractor.

This paper is organized as follows. First, the model used for the mobile robot and the structure detection principle are

¹ Irstea, Technologies and Information Support System Research Unit, 9 avenue Blaise Pascal, CS 20085, 63178 Aubière, France firstname.lastname@irstea.fr

² SITIA, 7 rue de l'halbrane, 44340 Bouguenais, France t.solatges@sitia.fr

presented. Then, the algorithm developed to build on-line a trajectory from successive footprints detections is detailed. Experimental results are finally proposed and discussed, allowing to highlight the performances of the proposed approach.

II. MODELING

A. Mobile robot modeling and control

In this paper, the control law used is designed for a car-like mobile robot, which can be classically viewed as a bicycle model. Fig. 2 depicts this point of view and introduces the notations used in this paper. These variables are defined as follows:

- ◇ v is the robot linear speed,
- ◇ $\dot{\theta}$ is the robot angular speed,
- ◇ δ is the robot steering angle,
- ◇ y is the lateral deviation which is basically considered as the path tracking error, as long as the desired offset is null,
- ◇ $\tilde{\theta}$ is the angular deviation. It is the difference between the robot heading in the global space and the trajectory tangent orientation at the closest point to the control point of the robot belonging to the desired trajectory. The point to be controlled on the robot is the middle of the rear axle,
- ◇ L is the robot wheelbase,
- ◇ $c(s)$ is the trajectory curvature at curvilinear abscissa s .

Fig. 2. Simple bicycle model applies to path tracking

This paper does not focus on the path tracking control algorithm, which is ensued from previous works and classical approaches (see [10]). As a result, Eq. (1) expresses the control law applied in this paper, which has been proven to be efficient, as long as assumptions are respected, basically that sliding and velocity are limited. This permits to achieved successive tasks using the same paradigm with several kind of sensors for control (row following, half turn, trajectory tracking to go back to initial position)

$$\delta = \arctan\left(L \cdot \left(\frac{c(s) \cdot \cos(\tilde{\theta})}{1 - y \cdot c(s)} + \frac{\cos^3(\tilde{\theta})}{(1 - y \cdot c(s))^2} \cdot (-K_D \cdot (1 - y \cdot c(s)) \cdot \tan(\tilde{\theta}) - K_P \cdot y + c(s) \cdot (1 - y \cdot c(s)) \cdot \tan^2(\tilde{\theta}))\right)\right) \quad (1)$$

This equation is defined if the condition $1 - y \cdot c(s) \neq 0$ is true. This means that the center of the curvature should not be superimposed on the center of the robot P_R . The goal of that control law is to bring lateral deviation y and angular deviation $\tilde{\theta}$ to zero. In expression (1), K_P and K_D are tuning gain allowing to defined the robot behavior. We assume that the velocity of the robot will be low enough to ensure that actuators will not be saturated.

$$\dot{\theta} = \frac{v \cdot \tan(\delta)}{L} \quad (2)$$

The robot used for experimental results is a skid-steering robot i.e. its control variable is an angular speed instead of a steering angle. As the efficiency of control law (1) is proofed, we use it with an equivalent class defined by Eq. (2) which is also described in [10].

B. Terrain and footprint models in lidar framework

This paper aims at detecting the footprint let by tires in a previous field operation, typically by a farm tractor, thanks to a single-beam lidar, mounting in front of the robot and oriented in soil direction. A desired shape for this footprint is then expected, and the objective of the first step consists in finding a matching shape. For that purpose, a correlation algorithm is used in order to match the current lidar scan and a pre-defined model. This template, as represented on Fig. 3, is representative of one tire footprint, allowing the detection with sufficient accuracy, while limiting the false positive detection result.

Fig. 3. Example of laser scan with two tire footprints and the associate template use in correlation

We assume that the model and the laser scan have the same number of points. The correlation expression used is defined by :

$$r(k) = \frac{\sum_{j=1}^N (z_{j+k} - \bar{z}) \cdot (m_j - \bar{m})}{\sum_{i=1}^N \sqrt{(z_i - \bar{z}) \cdot (m_i - \bar{m})}} \quad (3)$$

Where :

- ◊ N is the total number of points of both lidar data and template,
- ◊ $r(k)$ is the correlation score for the offset k between laser data and template; with k moving from $-\frac{2N-1}{2}$ to $\frac{2N-1}{2}$,
- ◊ z values are the lidar scan data,
- ◊ m values are the model data,
- ◊ $\bar{z}$ and $\bar{m}$ are the mean value of laser and model data respectively.

The correlation score r must reach a maximum when laser data delayed by k are closest to the template. It is assumed that for k varying from $-\frac{2N-1}{2}$ to $\frac{2N-1}{2}$, it exists a value k_{max} , for which r reaches a maximum r_{max} . Nevertheless, the challenge is to be sure that this maximal value r_{max} makes sense, and can be considered as a confident detection of the expected template. This problem is addressed in the next part of this paper.

III. STRUCTURE RECONSTRUCTION

The goal of this application is to follow a trajectory defined by a structure in the soil with an autonomous mobile robot, thanks to a lidar detection. The approach then consists in detecting the expected structure in each laser scan, and aggregating the successive scans in a global map. The accumulation of points in the 3D space during robot motion will then defined the trajectory to follow.

A. LIDAR scan processing

The first step is to detect the templates defined previously in each lidar scan. Fig. 3 defines the expected footprint in the framework of laser scan and suggests the use of correlation approach, as expressed by Eq. (3). However, in an off-road context, the terrain may quickly appear to be uneven, unlike the expected template. It leads to estimation error or false detection. As a result, a preliminary step consists in estimating the general terrain profile in order to compensate for soil inclination. This is achieved via a least mean square approximation. Data are approximated to a third degree polynomial expression as in Fig. 4. The polynomial degree is chosen in a way that it removes large variations of ground, but does not minimize other variations, which are representative of the template structure. The typical result of this terrain compensation is depicted in Fig. 4.

Once the general shape of the soil is compensated, the correlation equation (3) is used. As pointed out previously, the aim is to recognize the expected template in the lidar scan. In Fig. 3, the model is here defined as a negative hole in the ground with a depth and a width of ten centimeters. This is achieved by applying the correlation Eq. (3), with a varying offset k . The best position for model in data is represented by the maximum of correlation score $r(k)$, such as depicted on Fig. 5. As the Eq. (3) is normalized, a

Fig. 4. Removing ground variations

threshold can be chosen in order to validate that the maximal value $r_{max} = r(k_{max})$ is representative of a good detection. This threshold has been experimentally set up to 0.2, as it limits the false positive detection rate while maintaining useful results. As it will be presented in the experimental results part, this threshold is particularly important when using a template composed of multiple footprints in one lidar scan. In that case, not only the maximum value is kept as a good position but all the local maxima are considered as a possibility for structure position.

Fig. 5. Result of correlation computation on the data depicted on Fig. 3

The maximum correlation score value gives the shift k_{max} between template position (in the example, two maximum are found as there is two tire footprints in the data) and structure in real data. Therefore, as the three dimensional position of each laser point is known in a global map with a coordinate transformation between lidar and robot, the position of the footprint can be extracted from a reference map.

B. Global map reconstruction

With the previous part of this paper, the structure position can be extracted at each lidar scan and gives a lateral tracking deviation. In order to be stable, the tracking control law (1) requires an angular deviation which could be compute only with several points. The next step then consists in aggregating data with the mobile robot motion in order to define a trajectory to follow in a global map. By means of this

data collection, aberrant information can be removed. The main problem about aggregation lies in the accuracy of the robot motion estimation in the global map between each laser scan. In order to avoid the use of expensive sensors, the robot odometry is used for this step. The drift in the use of this kind of data will not affect the accuracy of path tracking, since only a local part will be used to evaluate lateral and angular deviations. In fact, only data between lidar impact and robot tires are kept; this distance of a few meters is not significantly impacted by odometry drift.

Since the footprints are expected to be realized by a previous vehicle in row, one can assume that desired trajectory can be approximated by a local line. It is then possible to aggregate the hole detected at each lidar scan iteration in a continuous line with a Least Mean Square algorithm. The Fig. 6 shows this approximation at three different moments. In the real time application, the linear estimation is made at each iteration of control.

Fig. 6. Linear approximation (red lines) of accumulated structure positioning data (in blue) for the first, second and third part of data

One can take part of the aggregation step to check the validity of the template detection in each laser scan. It has indeed been assumed that the trajectory to be estimated should be continuous. As a result, if the optimal k_{max} is discontinuous, one can deduce that this previous estimation is not relevant.

The last step consists in computing a steering control for the mobile robot. As shown in Eq. (1), it is necessary to extract a lateral and angular deviation between mobile robot and the structure to follow. If the linear estimation of trajectory gives an expression $y = a.x + b$, the gaps can be computed from:

$$y = \frac{a.X_{robot} - Y_{robot} + b}{\sqrt{a^2 + 1}}, \quad (4)$$

$$\tilde{\theta} = \theta_{robot} - \tan^{-1}(a). \quad (5)$$

With this algorithm, the computed control is always updated regarding both motion of the robot and new structure detection.

However, a limitation appears with this method: a minimum number of distant points is necessary to approximate

a line and so to compute a relevant control. That means the start of the robot is made in a "blind" process if another positioning system is not reachable. Data acquisition on about two meters length seems to be the minimum value to compute a reliable control. That is why the lidar position and orientation on the robot are important, because it permits seeing further than the robot front, and it also avoids laser masking.

IV. EXPERIMENTAL RESULTS

For the field experiments, we use a skid-steering mobile robot presented on Fig. 1, whose features are presented in Table I. This robot has two driven wheels at the rear and two caster wheels at the front. In this application, there are two control inputs: the linear speed and the angular speed.

TABLE I
ROBOT FEATURES

Weight (with cultivator)	2100 kg
Wheel track	1.40 m to 1.80 m
Maximum angular speed	2 rad.s ⁻¹
Maximum speed	1.5 m.s ⁻¹

For the detection process, the chosen LIDAR is an outdoor LIDAR Sick LMS151. All features are presented in Table II. In this application, only 120° in the center of view are kept. In fact, other data are not useful because the ground is not present on it so they are not reliable for the detection algorithm and it would just take more computation time for a useless result.

TABLE II
LIDAR FEATURES

Designation	LMS151
Maximum aperture angle	270°
Rotation frequency	50 Hz
Angular resolution	0.5°
Operating Range	0.5 to 20 m

All algorithms are implemented on ROS (Robot Operating System); several nodes which compute all actions presented in this paper are used:

- ◇ A mobile robot control node which is publishing the odometry data and subscribing to the control message (steering angle and constant linear speed).
- ◇ A detection node which is subscribing to the LIDAR data and is publishing the position of detected structure.
- ◇ A control node which extracts the control from the lateral and angular deviation between robot and structure positions.

In the next part of this paper, we will show some experiments made in a real world context. First experiment aims at validating the localization based on lidar. The second part

shows the effective control of the robot inside a greenhouse for weeds removing.

A. Detection accuracy: GPS validation

To demonstrate the accuracy of the algorithm, we use a RTK GPS which has a centimetric accuracy. The experiments presented in this part are conducted in an open field to ensure that a fix GPS measurement is always available. The tracked tire marks are made with the robot itself with GPS position recording. Then, the footprint detection algorithm is used to follow it, always with GPS recording. As GPS and footprints detection does not work in the same coordinates (global or local), we choose the lateral deviation as precision metric which is not coordinate system dependent.

In experiments displayed here, the robot begins with a lateral deviation of about 0.5 m . The initialization distance for laser detection is fixed at 2 m .

The comparison of robot positions between the footprints creation and the use of the laser following algorithm is presented Fig. 7. Note that the reference GPS trajectory (in blue) was filtered to smooth it while the other data are not.

Fig. 7. Result of footprint tracking evaluate with RTK GPS

During the initialization process, the robot is just moved forward without any angular speed. Consequently, it first goes away from reference trajectory and comes back to it when the command process takes over.

As mentioned earlier, the lateral deviation is a good criteria for algorithm validation in comparison with GPS. In fact, it will also be the only available metric without GPS (like in greenhouse) to conclude on algorithm efficiency. In Fig. 8, lateral deviation obtained both by GPS and lidar detection during the same experiment as before are presented.

The result shows that both plot have the same shape even if we can note some gap of the order of 5 cm . Actually, this value can be defined as the precision of our algorithm taking into account the relative error of RTK GPS. Then we can consider that the lateral deviation given by lidar detection algorithm is relevant to qualify the footprints following.

In Fig. 8, we can also observe that the robot takes around 10 m to get back on footprints which is coherent with the command law control gain used in this experiment : K_d is

Fig. 8. Lateral deviation compare between lidar detection algorithm and RTK GPS

equal to 0.7 and $K_p = \frac{K_d^2}{4}$. This value could be decreased but that could bring some overshoots or oscillations which are not wanted during the final application in greenhouse.

Fig. 9. Reconstruction result with two parallel slots

The last thing on detection validation is to know if the algorithm gives us noisy points or not. It would define the robustness of our application. Fig. 9 shows the points found by the algorithm during the experiment. As presented before, local positioning use only the last 2.5 m . In this plot, we can notice that there are no holes in detection process and no false positive points. The distance between two reconstructed lines is also the same as the wheel track of the robot.

At last, we can see that points are close to a line which confirms that the linear approximation used for local positioning is relevant.

B. Greenhouse experiments: final application validation

As presented in the introduction to this paper, the goal of this work is to mechanically remove weeds with an autonomous robot in greenhouses. The previous part gave proof of the algorithm precision in an open field; in this part we will present some results obtained in a 40 m long greenhouse.

The experiments were carried out for different values of the robot speed, along with whether the cultivator is used

or not. This is an important part as it allows us to test our algorithm in real working conditions.

Fig. 10. Lateral deviation at different speed and with or without cultivator

We choose two different speeds: 0.5 m.s^{-1} and 1.0 m.s^{-1} . These values represent the actual speed limits used to work with a manual farm tractor. The cultivator used is also an actual tool for pulling weeds. It is important to do some experiment with it because it could change the robot reaction and so our algorithm answer.

In Fig. 10, lateral deviations of four representative experiments are plotted. We cannot observe differences between each tests: in all cases the robot stays with a lateral deviation close to zero after the settling distance of 10 m . In these tests, the robot traveled around 30 m autonomously. This distance allows us to confirm the algorithm stability and that whether the cultivator is used or not is not significant for the result.

We have also done some repeatability tests at 0.5 m.s^{-1} . After eight experiences on different parts of the greenhouse, which represent a cumulated traveled distance of around 250 m , we found an average lateral deviation of 2 cm with a standard deviation of 7 cm . Since we do not observe precision variations, these values confirm that our algorithm is efficient when used in working conditions.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we present an approach to detect footprints and follow it with a mobile robot in real time operation with a 2D laser telemeter sensor. The proposed method is based on a local mapping which does not need a global positioning system or a previous mapping operation.

The method and experiments presented here show that the idea of using a single low cost lidar can be a relevant solution to detect and follow footprints or little ground structures. Accumulating data with mobile robot moves allows reducing positioning errors due to vehicle odometry use. As mentioned in the paper, the main disadvantage of this method is its beginning. In fact, this algorithm requires a first movement of the robot to give a relevant position in local map. This problem could be resolved with a global autonomous approach, for example other positioning systems can be used to bring the robot at greenhouse and to make a U-turn. The

robot could be positioned with another solution during the laser initialization distance so there is no more blind process.

The perspectives for this work can address other problems like prediction, detection and estimation of sliding, adaptive control or obstacles avoidance. The method of detection presented in this paper could be adapted to extract more information from lidar measurements. In the same way, more efficient control law could be used or developed to improve mobile robot navigation efficiency in agricultural robotics. This application can also be adapted to different templates; that could permit to follow other shapes, such as structured vegetation.

VI. ACKNOWLEDGEMENTS

This work is related to the PUMAGRI project which is funded as part of the FUI 20 by Economy, Industry and Digital French Minister and the French region of Pays de Loire and Auvergne.

REFERENCES

- [1] Marcel Bergerman, John Billingsley, John Reid, and Eldert van Henten. Robotics in Agriculture and Forestry. In *Springer Handbook of Robotics*, pages 1463–1492. Springer International Publishing, 2016.
- [2] D.M. Bevil, J. Ryu, and J.C. Gerdes. Integrating INS Sensors With GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering Stiffness. *IEEE*, 7(4):483–493, December 2006.
- [3] Simon Blackmore. Towards robotic agriculture. In *SPIE Commercial+ Scientific Sensing and Imaging*, pages 986603–986603. International Society for Optics and Photonics, 2016.
- [4] Guillaume Bresson, Thomas Fraud, Romuald Aufre, Paul Checchin, and Roland Chapuis. Real-time monocular slam with low memory requirements. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1827–1839, 2015.
- [5] ZJ Chong, Baoxing Qin, Tirthankar Bandyopadhyay, Marcelo H Ang, Emilio Frazzoli, and Daniela Rus. Synthetic 2d lidar for precise vehicle localization in 3d urban environment. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1554–1559. IEEE, 2013.
- [6] Tomislav Kos, Ivan Markezic, and Josip Pokrajcic. Effects of multipath reception on gps positioning performance. In *Elmar, 2010 Proceedings*, pages 399–402. IEEE, 2010.
- [7] Thomas Lemaire, Cyrille Berger, Il-Kyun Jung, and Simon Lacroix. Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364, 2007.
- [8] Peyman Moghadam, Wijerupage Sardha Wijesoma, and Dong Jun Feng. Improving path planning and mapping based on stereo vision and lidar. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 384–389. IEEE, 2008.
- [9] Julien Moras, Véronique Cherfaoui, and Philippe Bonnifait. Credibilist occupancy grids for vehicle perception in dynamic environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 84–89. IEEE, 2011.
- [10] Claude Samson, Pascal Morin, and Roland Lenain. Modeling and control of wheeled mobile robots. In *Springer Handbook of Robotics*, pages 1235–1266. Springer, 2016.
- [11] Guillaume Trehard, Zayed Alsayed, Evangeline Pollard, Benazouz Bradai, and Fawzi Nashashibi. Credibilist simultaneous localization and mapping with a lidar. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2699–2706. IEEE, 2014.

A Haptic Teleoperation of Agricultural Multi-UAV

Chanyoung Ju, Sungjun Park, Sangsoo Park, and Hyoung Il Son, *Member, IEEE*,

Abstract—In this study, we propose a distributed swarm control algorithm for an agricultural multiple unmanned aerial vehicle (UAV) system that enables a single operator to remotely control a multi-UAV system. The system has two control layers that consist of a teleoperation layer through which the operator inputs teleoperation commands via a haptic device and a UAV control layer through which the motion of UAVs is controlled by a distributed swarm control algorithm. In the teleoperation layer, the operator controls the desired velocity of the UAV by manipulating the haptic device and simultaneously receives the haptic feedback. In the UAV control layer, the distributed swarm control consists of the following three control inputs: 1) velocity control, 2) formation control, and 3) collision avoidance control. The three controls are input to each UAV for the distributed system. The proposed algorithm is implemented in the dynamic simulator, and experimental results using four UAVs are presented to evaluate and verify the algorithm.

Keywords—agricultural UAV, multi-UAV system, distributed swarm control, haptic teleoperation, UAV simulator.

I. INTRODUCTION

When drones or unmanned aerial vehicles (UAVs) are used in agriculture, it is possible to create a map that reconstructs the location of farmland and crops in three dimensions. Subsequently, the yield of crops is estimated through sensors mounted on the UAV [1]. Furthermore, the health index and vegetation index of crops are calculated through visual, infrared, and thermal information measured by the UAV and the method using agricultural UAV more accurately detects pests and diseases that cannot be observed well on the ground and facilitate eco-friendly and high-efficiency agriculture by applying variable amounts of water, fertilizer, and pesticide based on the state of the crop [2]. Thus, agricultural UAVs display excellent potential in addition to low maintenance costs that are used in a variety of farming applications and are growing rapidly as a major technology for agriculture [3].

However, an individual UAV involves limitations in terms of weight, size, and energy consumption for itself as well as for the sensors that it carries [4]. A method to alleviate these problems involves performing farming by using multiple vehicles that cooperate to achieve the mission goal (i.e., crop scouting, seeding, pesticide spraying, and transport). Various improvements occur when a multi-UAV system that can improve robustness is used and include a reduction in the completion time and an increase in the amount of work relative to time when compared to those when a single-UAV system

is applied [5]. Therefore, the deployment of multiple UAVs further increases agricultural working efficiency by utilizing the advantages of a multi-UAV system. The application involving multi-UAV systems is used for detecting, localizing, tracking, surveillance, and intercepting [6]. However, in a few cases, the multi-UAV system was applied to agriculture, and several studies continue to progress (e.g., multi-UAV based cooperative remote sensing for real-time water management and irrigation control) [7]. In this study, we propose a swarm control algorithm that is applied to a multi-UAV system for agriculture.

If a multi-UAV system is introduced for agriculture, a more advanced and smarter system is constructed when compared with the current agricultural environment. However, several difficulties persist in the actual application of the agricultural multi-UAV system. Thus, to adapt to the real agricultural environment, we designed a multi-UAV system by considering the following two methods 1) a UAV control method; and 2) a swarm control method. The first method (UAV control method) involves three typical methods to control the UAV, namely a method of autonomous driving by using Simultaneous Localization and Mapping (SLAM), a method of driving on a certain path as specified by the operator, and a method of teleoperation by the operator in real time. Currently, most methods of controlling UAV in agriculture use the second method by using ground station software such as a Mission Planner or QGroundControl. However, this method specifies the path each time, and thus the system becomes more complicated and difficult when the use of the UAVs increases, and it usually uses a centralized controller while attempting to control multi-UAV. Therefore, it is not suitable to use agricultural multiple UAVs. The best method involves performing agricultural tasks with fully automated UAVs. Unfortunately, it is not currently possible to guarantee the safety of using multiple UAVs that automatically avoid obstacles and fly the path by using SLAM or Machine vision (MV) in unstructured, unspecified, and uncertain agricultural environments. In such cases, fully autonomous control of the UAVs is typically impossible, and instead, a teleoperation of their behaviors is desired if not absolutely necessary to impose human intelligence on the task of coping with the forementioned uncertainties [8], [9]. Additionally, when an operator needs more precise or additional work with an agricultural multi-UAV, the use of a system in which a teleoperation command by the operator is entered in a supplementary manner to control the multi-UAV is more advantageous when applied to agricultural work than the use of an incomplete automated UAV system.

In case of teleoperation, efficiency decreases and costs increase when several operators are mobilized to control multiple UAVs. Thus, an efficient system in which multiple

Chanyoung Ju, Sungjun Park, Sangsoo Park, Hyoung Il Son are with the Department of Rural and Biosystems Engineering, Chonnam National University, 77 Yongbong-ro, Buk-gu, Gwangju 61186, Republic of Korea. Corresponding author : Hyoung Il Son, E-mail: hison@jnu.ac.kr, URL: hralab.jnu.ac.kr, Phone: +82-62-530-2152, Fax: +82-62-530-2159.

Manuscript received July 31, 2017; revised August xx, 2017.

UAVs are remotely controlled by a single operator is required [10]. In [7], several operators were mobilized to control an agricultural multi-UAV used for area coverage and remote sensing. Furthermore, the method that is most commonly used in agriculture currently is the 1:1 control method. Nevertheless, the proposed method in the current study corresponds to a 1:N swarm control method in which it is not necessary to mobilize several operators. In the study, we considered a swarm control method involving the agricultural environment. The following three typical methods exist for swarm control: centralized method, decentralized method, and distributed method. The centralized method is a way to handle and control the whole process by a single reader that contains complete information. This method is relatively easy in terms of implementation of the system and management of data. However, the centralized method is not suitable for outdoor environments, such as agriculture, owing to the lack of a direct connection between UAVs. Thus, swarm control is not performed when an error occurs in the leader as well as when an error occurs in only a single UAV [11]. The distributed method is a way to maintain the swarm control system even if a certain leader includes errors because the distributed controller is mounted in each UAV, and this method is the most stable way to cope with unpredictable errors or accidents. Among the three methods, we swarm control agricultural multiple UAVs by using a more stable and flexible decentralized method, and this system copes with the uncertainty in the agricultural environment.

In summary, the method that is currently used is a 1:1 control method for agricultural multi-UAV via several operators and a system that uses swarm control via a single operator does not exist. Therefore, in the present study, we proposed a distributed swarm control for an agricultural multi-UAV system that enables a single operator to remotely control the multi-UAV. The distributed swarm control consists of the following three control inputs: 1) a velocity control of UAV by using an operator's teleoperation command, 2) a formation control to form the desired formation by using an artificial potential field, and 3) a collision avoidance control to avoid obstacles. In a teleoperation, an operator uses a haptic device to control a multi-UAV, and appropriate haptic feedback is given to an operator. Formation control forms a desired formation by using a potential function that is defined as the relative distance between UAVs. Collision avoidance control automatically avoids obstacles via a repulsive potential field defined as the distance between UAVs and Obstacles. The three controls are input to each UAV for the distributed system. In this study, we implemented a multi-UAV system with a distributed swarm control algorithm into a dynamic simulator and subsequently performed experiments to validate and evaluate the algorithm.

The structure of this paper is as follows. Section II introduces the control architecture based on a distributed swarm control algorithm. Section III shows an overall hardware-in-the-loop simulation that includes an experiment setup, an experimental task, and data analysis. Finally, Section IV and Section V show experimental results by using four UAVs and discuss the conclusions of this study, respectively.

II. DISTRIBUTED SWARM TELEOPRATION

A. UAV Dynamics

We consider N quadrotor-type UAVs with 3-DOF Cartesian positions that are denoted by $p_i \in \mathbb{R}^3$, $i = 1, 2, \dots, N$. Flight control of UAVs is derived from the following under-actuated Lagrangian dynamics equation in $SE(3)$ [12]

$$m_i \ddot{p}_i = -\lambda_i R_i e_3 + m_i g e_3 + \delta_i \quad (1)$$

$$J_i \dot{w}_i + S(w_i) J_i w_i = \gamma_i + \zeta_i, \quad \dot{R}_i = R_i S(w_i) \quad (2)$$

where $m_i > 0$ denotes mass, $p_i := [p_1; p_2; \dots; p_N] \in \mathbb{R}^{3N}$ denotes the Cartesian center-of-mass position represented in the north-east-down (NED) inertial frame $\{O\} := \{N^O, E^O, D^O\}$, $\lambda_i \in \mathbb{R}$ denotes thrust control input, $R_i \in SO(3)$ denotes the rotational matrix describing the body-frame $B := \{N^B, E^B, D^B\}$ of UAV w.r.t. to the inertial frame $\{O\}$, g is the gravitation constant, $e_3 = [0, 0, 1]^T$ denotes the basis vector representing the down direction and representing that thrust and gravity act in the D direction, $J_i \in \mathbb{R}^{3 \times 3}$ denotes the UAV's inertia matrix w.r.t. the body frame $\{B\}$, $w_i \in \mathbb{R}^3$ denotes the angular velocity of the UAV relative to the inertial frame $\{O\}$ represented in the body frame $\{B\}$, $\gamma_i \in \mathbb{R}^3$ denotes the attitude torque control input, $\delta_i, \zeta_i \in \mathbb{R}^3$ denote the aerodynamic perturbations, and $S(w_i) : \mathbb{R}^3 \rightarrow so(3)$ denotes the skew-symmetric operator defined s.t. for $\alpha, \beta \in \mathbb{R}^3$, $S(\alpha)\beta = \alpha \times \beta$. For typical UAV flying, $\delta_i, \zeta_i \approx 0$

B. Distributed Swarm Control

Our goal involves using the distributed method as opposed to the centralized method to simultaneously control the 3-DOF Cartesian position of N UAVs. The distributed swarm control algorithm approaches the following three fundamental requirements: 1) UAV control; 2) formation control; and 3) obstacle avoidance control [8], [13].

First, to describe the manner in which the UAVs are connected via communication to form the N -nodes, we define the dynamic undirected *connectivity graph* $\mathcal{G} \triangleq \{\mathcal{V}, \mathcal{E}\}$ by the vertex set $\mathcal{V} \triangleq \{1, 2, \dots, n\}$, representing the UAVs and the edge set $\mathcal{E} \triangleq \{e_{ij} : i = 1, 2, \dots, n, j \in \mathcal{N}_i\}$ representing the connectivity among the UAVs, where the dynamic neighbor set $\mathcal{N}_i$ of UAV i is defined as follows:

$$\mathcal{N}_i \triangleq \{j \in \mathcal{V} : \text{UAV } i \text{ receives information from UAV } j, i \neq j\} \quad (3)$$

Subsequently, we implement the following distributed swarm control on each UAV, for the i th UAV,

$$\dot{p}_i(t) := u_i^u + u_i^c + u_i^o \quad (4)$$

where the meaning of the three control inputs $u_i^u \in \mathbb{R}^3$, $u_i^c \in \mathbb{R}^3$ and $u_i^o \in \mathbb{R}^3$ represents the velocity terms of the UAV.

Fig. 1. Concept of a multi-UAV System: haptic teleoperation of multiple UAVs

1) *UAV control* u_i^u : The UAV control method mainly uses the following three methods: (i) the method of fully autonomous driving, (ii) the method of driving on a certain path specified by the operator, and (iii) the method of teleoperation by the operator in real time (Fig.1). However, as mentioned in Section I, we considered only the method of (iii) given the agricultural environment.

Therefore, we consider a 3-DOF haptic device for master as modeled by the following nonlinear Lagrangian dynamics equation [14]

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = \tau + f_h \quad (5)$$

where $q \in \mathbb{R}^3$ denotes the configuration of the haptic device (e.g., the position of end effector), $M(q) \in \mathbb{R}^{3 \times 3}$ denotes the positive-definite/symmetric inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ denotes the Coriolis matrix, and $\tau \in \mathbb{R}^3$, $f_h \in \mathbb{R}^3$ denote the control input and human forces, respectively.

The velocity term, $u_i^u \in \mathbb{R}^3$, represents the teleoperation command for the desired velocity input of the UAV that is directly controlled by the operator by using the configuration of the haptic device q

$$u_i^u = \Lambda q \quad \forall i \quad (6)$$

where $\Lambda \in \mathbb{R}^+$ denotes a constant scale factor used to match different scales between q and the UAV desired velocity u_i^u , and $q \in \mathbb{R}^3$ denotes the position of end effector. In (6), multiple UAVs with an unbounded workspace can fly without the limitations of workspace by controlling the desired velocity by using the configuration of the haptic device with a bounded workspace.

Simultaneously, to allow the operator to telesense a few UAVs and their surrounding obstacles, we consider the haptic feedback $y(t) \in \mathbb{R}^3$ s.t., as defined by

$$y_f(t) := \frac{1}{\Lambda N} \sum_{i=1}^N u_i^o, \quad y_v(t) := q - \frac{1}{\Lambda N} \sum_{i=1}^N \dot{x}_i. \quad (7)$$

$$y(t) = y_f + y_v \quad (8)$$

where $\dot{x}_i$ denotes i th UAV's velocity, and u_i^o denotes i th UAV's obstacle avoidance control, $y_f(t)$ denotes force-cue feedback, and $y_v(t)$ denotes velocity-cue feedback. The force-cue feedback plays the role of a repulsive force from the environment. It is related to the difference between the position of UAVs

and the location of the obstacles. The velocity-cue feedback represents the difference between the commanded velocity as specified by q and the average velocity of UAVs. This difference is caused due to various reasons and is described in detail in [13]. Finally, $y(t)$ denotes the sum of these two signals y_f and y_v and is sent to the master via the communication channel.

The ultimate objective involves implementing a multi-UAV system that enables a single operator to control N UAVs' Cartesian positions (1) $p := [p_1; p_2; \dots; p_N] \in \mathbb{R}^{3N}$ based on the distributed swarm control algorithm via the single 3-DOF haptic device (5) as shown in Fig.1. The operator teleoperates more intuitively by receiving visual feedback as well as haptic feedback based on the information of the N UAV's state and its surrounding environment.

2) *Formation control* u_i^c : The second velocity term, $u_i^c \in \mathbb{R}^3$ denotes a control input to avoid a collision among UAVs, preserves connectivity, and achieves a certain desired formation as specified by the desired distances $d_{ij}^c \in \mathbb{R}^+$ $\forall i = 1, \dots, N$, and $\forall j \in \mathcal{N}_i$, as defined by

$$u_i^c := - \sum_{j \in \mathcal{N}_i} \frac{\partial \varphi_{ij}^c (\|p_i - p_j\|^2)^T}{\partial p_i} \quad (9)$$

where φ_{ij}^c denotes a certain artificial potential function to create an attractive action if $\|p_i - p_j\| > d_{ij}^c$, a repulsive action if $\|p_i - p_j\| < d_{ij}^c$, and a null action if $\|p_i - p_j\| = d_{ij}^c$. The potential function φ_{ij}^c is described in further detail, and φ_{ij}^c consists of V_{ij} and W_{ij} [15]. V_{ij} denotes a repulsive potential function to avoid collision among UAVs and requires the following properties.

- 1) V_{ij} denotes a function of the square norm of the distance between UAVs i, j , not based on vector

$$V_{ij} = V_{ij}(\|p_i - p_j\|^2) = V_{ij}(\beta_{ij}) \quad (10)$$

- 2) V_{ij} attains its maximum value whenever $\beta_{ij} \rightarrow 0$. In other words, we require that $V_{ij} \rightarrow \infty$ whenever $\beta_{ij} \rightarrow 0$.
- 3) It is continuously differentiable from everywhere.
- 4) $\partial V_{ij} / \partial p_i = 0$ and $V_{ij} = 0$ whenever $\beta_{ij} > (d_{ij}^c)^2$.
- 5) $\partial V_{ij} / \partial \beta_{ij} < 0$ whenever $0 < \beta_{ij} < (d_{ij}^c)^2$ and $\partial V_{ij} / \partial \beta_{ij} = 0$ whenever $\beta_{ij} \geq (d_{ij}^c)^2$.

Additionally, W_{ij} denotes an attractive potential function between UAVs i and $j \in \mathcal{N}_i$, which is required to exhibit the following properties for aggregation.

- 1) W_{ij} denotes a function of squared norm of the distance between UAVs i, j , that is, not based on a vector

$$W_{ij} = W_{ij}(\|p_i - p_j\|^2) = W_{ij}(\beta_{ij}) \quad (11)$$

- 2) W_{ij} attains its maximum value whenever $\beta_{ij} \rightarrow \infty$. Thus, we require that $W_{ij} \rightarrow \infty$ whenever $\beta_{ij} \rightarrow \infty$.
- 3) It is continuously differentiable from everywhere.
- 4) $\partial W_{ij} / \partial p_i = 0$ and $W_{ij} = 0$ whenever $\beta_{ij} < (d_{ij}^c)^2$.
- 5) $\partial W_{ij} / \partial \beta_{ij} > 0$ whenever $(d_{ij}^c)^2 < \beta_{ij}$ and $\partial W_{ij} / \partial \beta_{ij} = 0$ whenever $\beta_{ij} \leq (d_{ij}^c)^2$.

Finally, the distributed formation control for each UAV i is given as the sum of the negative gradients of the two

Fig. 2. Two control layers while using teleoperation

Fig. 3. Experimental Setup

potentials (10), (11) in the p_i direction: $u_i^c := -\sum_{j \in \mathcal{N}_i} \frac{\partial V_{ij}}{\partial p_i} - \sum_{j \in \mathcal{N}_i} \frac{\partial W_{ij}}{\partial p_i}$. Therefore, given the presence of two vertical asymptotes in u_i^c about the bounded potential field, the prevention of intervehicle collisions and the preservation of intervehicle connectivity is guaranteed.

3) *Obstacle avoidance control* u_i^o : The final velocity term, $u_i^o \in \mathbb{R}^3$, is expressed by the following equation as a control input based on a potential field that allows multiple UAVs to avoid obstacles through a certain distance threshold: $\mathcal{D}_o \in \mathbb{R}^+$

$$u_i^o := - \sum_{r \in \mathcal{O}_i} \frac{\partial \varphi_{ir}^o (\|p_i - p_r^o\|)^T}{\partial p_i} \quad (12)$$

where $\mathcal{O}_i$ denotes the set of obstacles of the i th UAV with an obstacle point p_r^o that corresponds to the position of the r th obstacle in the environment, and φ_{ir}^o denotes a certain artificial potential function that produces a repulsive action if $\|p_i - p_r^o\| < \mathcal{D}_o$, and a null action if $\|p_i - p_r^o\| \geq \mathcal{D}_o$. When the distance between the UAVs and the obstacles becomes closer to $\mathcal{D}_o$, then the repulsive potential function increases to infinity.

In short, each UAV is controlled by distributed swarm control(4), and this corresponds to the sum of the three control inputs. As a result, there are two control layers that consist of a teleoperation layer, which the operator inputs control commands via the haptic device, and a UAV control layer in which a slave's motion is controlled by the distributed swarm control algorithm.

III. EVALUATION OF DISTRIBUTED SWARM CONTROL

A. Experimental Setup

The apparatus mainly consists of a desktop, a monitor that displays the simulation, and a haptic device (Fig.3). In the

Fig. 4. Experimental Task

desktop, we used the Ubuntu14.04 LTS version of the Linux environment to simulate the virtual environment by using the Robot Operating System (ROS) and Gazebo (3D robot simulator). A virtual environment was constructed to simulate UAVs dynamics and their control laws. The dynamical control of multiple UAVs based on the distributed swarm control algorithm was simulated in a virtual environment by using Open Dynamic Engine(ODE). This environment was presented based on the Gazebo for 3D graphical rendering and on the ROS for haptic rendering, control, and communication. In the simulation, the visual scene is rendered from a camera perspective at a specific distance from the starting points of the UAVs. The graphical and haptical simulations are run at 60 Hz and 1000 Hz, respectively.

Additionally, an operator can control the virtual multi-UAV by manipulating the single haptic device. The haptic device is used as the master robot and corresponds to the commercial available Novint Falcon by Novint Technologies, Inc. with actuated linear 3-DOF and un-actuated rotational 3-DOF. It generates 3-DOF force feedback up to 8.9 N at the nominal position and its workspace is cube-shaped with an edge (4" × 4" × 4"). The device is connected to a desktop through USB with 1000 Hz servo-rate. Therefore, the desktop performs serial communication with the haptic device and communicates with the multiple UAVs through ROS messages (e.g., topics, services).

B. Experimental Task

The experimental task involves the flight of a multiple UAVs based on the distributed swarm control algorithm along a certain path with respect to the operator's teleoperation command (Fig.4). In the experiment, we used four UAVs to implement our goal of multiple UAV systems. During the experiment, we confirmed that multiple UAVs form the desired formation by the formation control. At this time, UAVs form a triangular pyramid with a constant relative distance. We also placed several obstacles in the path of the Task and confirmed that the UAVs avoid collisions by using the obstacle avoidance control. As shown in the Fig.4, the path is defined until UAVs arrive at the goal area from the starting point, and the experiment ends when the multi-UAVs reach the goal area.

Fig. 5. Trajectory of UAVs projected on the XY planes during the task.

Fig. 6. Trajectory of UAVs projected on the XZ planes during the task.

C. Data Analysis

To confirm that the UAV forms well in the desired formation, avoids obstacles, and behaves in accordance with teleoperation commands, we recorded the position of the UAV(p_i), velocity of the UAV($\dot{p}_i$), position of the obstacle(p_j^o), and simulation time(ms, t) at 1000 Hz. Most of the data was sent and received via *rostopic*, and the *rosviz* was used to save the desired *rostopic*. Data analysis confirmed the following points: the multi-UAV forms a triangular pyramid shape with a constant relative distance between each other; and the UAV is automatically avoided through repulsive action based on a potential field when the distance to the obstacle is less than the distance threshold, and it should act based on the operator's desired velocity input. The results of the experiment are detailed in the following Section IV.

IV. EXPERIMENTAL RESULTS

In the experiment, a scenario was assumed wherein the operator controls the agricultural multi-UAV from the starting point to the ending point. Obstacles, such as trees and utility poles, are placed in the field (virtual world), and the agricultural task proceeds such that it passes above or beside the obstacles. The multi-UAV is assumed as a UAV for agricultural observation, spraying, or sowing. The results of the experiment obtained by using the distributed swarm control are shown in Fig.5-9.

In Fig.5-6, we present the trajectories of multiple UAVs during the experiment. This figure shows the trajectories of the UAVs controlled by the operator's teleoperation command and indicate as to whether the UAVs form the desired formation. The multi-UAV is controlled by the distributed swarm control algorithm, and the formation of the UAVs is

Fig. 7. Velocity of UAVs $\dot{p}_i$

Fig. 8. Height of UAVs with Obstacles

well formed as shown in Fig.5-6. Therefore, when the multi-UAV is applied to the agriculture by using the algorithm, the operator can perform agricultural work while maintaining the desired formation of UAVs. Additionally, the operator controls via teleoperation, and thus, it is possible to work more precisely on the desired area and also to cope with an unexpected accident. Furthermore, the formation of multi-UAV is controlled by modifying the artificial potential function to form the desired formation, and it is possible to perform a flexible agricultural task by forming a one-column-array formation or a quadrilateral formation as necessary.

Fig.7 shows the velocity of each UAV. In this case, the velocity control is input to each UAV as the sum of the teleoperation control of UAVs, the formation control, and the collision avoidance control. In this study, the largest input corresponds to the velocity control of teleoperation command, and the supplementary inputs correspond to the formation control and the collision avoidance control. The distributed swarm control is represented by the sum of three control inputs, and thus the formation may change based on the control input that is mainly applied. It may not be controlled as desired, and thus it is necessary to consider the most important control input. As shown in the figure, the three control inputs control each UAV via a distributed controller, and agricultural work is performed uniformly and constantly because the velocity of each UAV does not significantly differ.

Fig. 9. Relative distance between UAVs $\|p_i - p_j\|$

Fig.8 shows the height of the UAVs and the position of the obstacles. The experimental results are mainly controlled by the collision avoidance control. In the simulation, the multi-UAV approaches upwards with respect to the first and second obstacles and approaches the side of the middle for the other two obstacles. As shown in the figure, when the distance between the UAVs and the obstacles is within a certain distance threshold, the repulsive potential field increases while approaching the obstacle, and multiple UAVs avoid the obstacles. In this manner, it is possible to avoid the increasingly frequent crashing problem caused by the collision with the utility pole when the algorithm is applied to an agricultural multi-UAV system and to safely avoid specific obstacles when the obstacles are detected by a sensor during agricultural tasks.

Fig.9 presents the relative distances between UAVs, and the experimental results are mainly controlled by the formation control. Theoretically, it is necessary to maintain a relative distance between UAVs at a certain desired distance although it is almost impossible to implement the same. This is because position measurement errors occur at times, and the UAVs may become unstable due to disturbance, or the position of the UAVs constantly changes. Thus, it is extremely difficult to match the accurate value. Therefore, in the simulator, the relative distance between UAVs is maintained at a distance between 1 m and 1.2 m. Hence, specification of the desired range between UAVs in this manner prevents oscillation by the potential field more than that when the relative distance between UAVs is set to be a desired distance. The repulsive action was derived when the distance between the UAVs became closer to 1.0 m, and the attractive action was derived when the distance between the UAVs exceeds 1.2 m such that the formation is maintained. As shown in the figure, the distance between the UAVs does not exceed 1.2 m and is not below 1.0 m. It is assumed that a real multi-UAV system exists for agriculture. Therefore, even if multiple UAVs are used in a system when compared with a conventional system that uses a single UAV, it can be controlled stably because it maintains the desired formation while avoiding collision with others. The multiple UAVs work together, and this simultaneously increases the amount of work and the area of work, reduces the working time, and significantly increases the agricultural efficiency.

V. CONCLUSION

In this study, we proposed a distributed swarm control algorithm for agricultural multiple UAVs by using a teleoperation architecture that consisted of the following two layers: 1) a UAV control layer that controls the multiple UAVs to form a desired formation as specified by the desired distances to avoid obstacles specified by the distance threshold and to drive a specific path by UAV control method; and 2) a teleoperation layer that controls multiple UAVs such that they move at a desired velocity given the operator's teleoperation command, and the operator simultaneously receives haptic feedback with respect to sensing UAVs and their surrounding obstacles. To apply the same to actual agricultural multi-UAV systems, we

used a teleoperation method for UAV control and a distributed method for swarm control, and the methods are optimized for the agricultural environment.

ACKNOWLEDGMENT

This research was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant NRF-2015R1C1A1A02036875, and in part by a grant (115062-2) funded by the Ministry of Agriculture, Food, and Rural Affairs (MAFRA, Korea).

REFERENCES

- [1] Das, Jnaneshwar, et al. "Devices, systems, and methods for automated monitoring enabling precision agriculture." *Automation Science and Engineering (CASE)*, 2015 IEEE International Conference on. IEEE, 2015.
- [2] Katsigiannis, Panagiotis, et al. "An autonomous multi-sensor UAV system for reduced-input precision agriculture applications." *Control and Automation (MED)*, 2016 24th Mediterranean Conference on. IEEE, 2016.
- [3] Xiang, Haitao, and Lei Tian. "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV)." *Biosystems engineering* 108.2 (2011): 174-190.
- [4] Doering, Dionisio, et al. "Design and optimization of a heterogeneous platform for multiple uav use in precision agriculture applications." *IFAC Proceedings Volumes* 47.3 (2014): 12272-12277.
- [5] Jones, Chris, et al. "Principled approaches to the design of multi-robot systems." *Proc. of the Workshop on Networked Robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*. 2004.
- [6] Han, Jinlu, et al. "Low-cost multi-UAV technologies for contour mapping of nuclear radiation field." *Journal of Intelligent & Robotic Systems* (2013): 1-10.
- [7] Chao, Haiyang, et al. "Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control." *IFAC Proceedings Volumes* 41.2 (2008): 11744-11749.
- [8] Lee, Dongjun, et al. "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles." *IEEE/ASME Transactions on Mechatronics* 18.4 (2013): 1334-1345.
- [9] Lee, Dongjun, et al. "Haptic teleoperation of multiple unmanned aerial vehicles over the internet." *Robotics and automation (icra)*, 2011 IEEE international conference on. IEEE, 2011.
- [10] Franchi, Antonio, et al. "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs." *IEEE Robotics & Automation Magazine* 19.3 (2012): 57-68.
- [11] Borrelli, Francesco, Tams Keviczky, and Gary J. Balas. "Collision-free UAV formation flight using decentralized optimization and invariant sets." *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 1. IEEE, 2004.
- [12] Yang, Hyunsoo, et al. "Multi-rotor drone tutorial: systems, mechanics, control and state estimation." *Intelligent Service Robotics* (2017): 1-15.
- [13] Son, Hyoung Il, et al. "Human-centered design and evaluation of haptic cueing for teleoperation of multiple mobile robots." *IEEE Transactions on Cybernetics* 43.2 (2013): 597-609.
- [14] Rodriguez-Seda, Erick J., et al. "Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance." *IEEE Transactions on Control Systems Technology* 18.4 (2010): 984-992.
- [15] Dimarogonas, Dimos V., and Kostas J. Kyriakopoulos. "Connectedness preserving distributed swarm aggregation for multiple kinematic robots." *IEEE Transactions on Robotics* 24.5 (2008): 1213-1223.

Thorvald II Configuration for Wheat Phenotyping

Lars Grimstad¹, Kristine Skattum¹, Eirik Solberg¹, Gabriel Da Silva Martins Loureiro² and Pål Johan From¹

Abstract—This paper presents a mobile agricultural robot specifically designed for phenotyping tasks. The robot is a modified version of the Thorvald II agricultural robot. It uses the same modules as the standard robot, but is re-designed to have a higher ground clearance, narrower wheels, and more protective covers to protect the plants than the standard version. The robot can move autonomously in the field and collects data for plant phenotyping. This type of system will be a game changer when it comes to efficient and high-throughput phenotyping and will be an important factor in increasing the world's food production over the next decades.

I. INTRODUCTION

Plant phenotyping refers to a quantitative description of the plants anatomical, ontogenetical, physiological and biochemical properties [1], which is used to breed more resilient plants with higher yields. Phenotyping of plants is thus one of the main tools available to plant breeders to increase productivity, and therefore important in increasing the world's food production as well as the quality and reliability of food production throughout the world.

Phenotyping currently depends on manual labor to perform in-field measurements [2]. The work is tedious and traits like plant coverage are commonly still determined by visual inspection. High-throughput phenotyping will increase genetic gains in plant breeding, but this requires systems for collecting vast amounts of accurate and relevant data from the fields. These fields are normally huge areas with several hundreds or thousands of plots with varieties of plants. Robotic solution in the form of drones or ground robots thus arise as a natural tool for collecting this information in a cost-efficient and reliable manner. Collecting data with specified intervals in time and space can be performed efficiently with robots. In this paper we thus propose to use ground robots that can be programmed to autonomously collect this data from the fields, save the data, and finally analyze the data and present it in a format useful to plant breeders.

Some robots that have been designed for phenotyping tasks can be found in literature. On the larger end of the scale, huge gantry cranes are used to carry advanced sensory systems over relatively large areas. These systems are extremely costly and can only cover the area defined by the workspace of the robot. An example of such systems is the 30-tonnes LemnaTec Field Scanalyzer at the University

of Arizona's Maricopa Agricultural Center used to monitor 1.5 acres. On the other end of the scale, we have smaller mobile robots like the TERRA-MEPP robot developed by the University of Illinois, which is a small light-weight robot that can move between rows of sorghum plants. Other concepts closer to the one presented in this paper are the BoniRob developed by Bosch [3] and TED developed by Nao Technologies.

In this paper we will present the phenotyping robot including mechanical design, control and navigation. The robot is assembled from modules from the Thorvald II modular robotic system and has been specially configured for phenotyping tasks. This is done in close collaboration with researchers on plant breeding to satisfy all requirements, such as ground clearance, wheel width, weight, accuracy, and so on.

The paper is structured as follows: First we discuss robotic platforms and describe the robot especially assembled for phenotyping. We then briefly describe the camera setup before we go through the current system for setting up and running automatic data gathering with the robot. We finish the paper with some concluding remarks.

II. ROBOTIC PLATFORMS

In our previous work [4] we have investigated the positive and negative sides of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) for wheat phenotyping. In short, UAVs are good for covering large areas quickly as the images are well suited for calculating key spectral indices like the NDVI (normalized differential vegetation index) and EVI (enhanced vegetation index). However, the images captured by the UAV lack the level of detail that we get from images captured by the ground vehicle. In fact we used the images from the UGV as ground truth when trying to determine key events in the plant's life, like heading date, from the UAV images. The UGV can also carry a far greater payload than the UAV, which is important for certain sensors, especially hyper spectral cameras which tends to be rather large. We will therefore continue to use both UAVs and UGVs in future works.

In the aforementioned previous endeavors we also concluded that our then robot, Thorvald I, was unsuited for the task of wheat phenotyping, and that a more specialized robot was needed.

A. Thorvald I

Thorvald I was developed to be used as a multipurpose tool carrier [5]. The all electric robot has four-wheel drive

¹Faculty of Science and Technology, Norwegian University of Life Sciences, Drøbakveien 31, 1432 Ås, Norway
larg@nmbu.no, kristine.skattum@nmbu.no,
eirik.solberg@nmbu.no paf@nmbu.no

²Escola Politécnica, Universidade Federal do Rio de Janeiro - Av. Athos da Silveira Ramos, 149 - Cidade Universitária, Rio de Janeiro - RJ, 21941-909, Brasil gloureiro@poli.ufrj.br

and four-wheel steering, and is designed for payloads of up to 150 kg.

The reason for the robot being deemed unsuited for wheat phenotyping is the robot's fixed height of 59 cm. It could only operate as long as the plants were shorter than this, or else the robot may have damaged the plants. As wheat plants grow far higher than 59 cm, the robot's limited height was in some cases a considerable problem, and one of the reasons for developing Thorvald II. Fig. 1 shows Thorvald I in a field of wheat. Here the plants have clearly outgrown the robot.

Fig. 1: Thorvald I in a field of wheat. Here the plants are so high that they may be damaged by the robot.

B. Thorvald II

Thorvald II is Thorvald I's successor, but instead of being one single robot, Thorvald II is a family of an infinite number of different robots [6]. The robotic concept is based on modules and the idea is that a handful standard modules can be assembled in different configurations to create robots with widely different properties for different agricultural environments. The modules connect through standardized mechanical and electrical interfaces. Assembling or re-configuring a robot is done with basic tools, and can in many cases be achieved in less than one hour. An example of such a module is the drive module. This module contains a motor with transmission connected to a wheel and is used to propel the robot. Other examples of a modules are the battery enclosure, which contains batteries, computers and other electronics, and the steering module, which is used to turn a drive modules in the horizontal plane. Examples of different robot configurations are shown in Fig. 2.

Although several different robots have been built using the Thorvald II modules, the standard configuration is the four-wheel drive, four-wheel steering version with passive suspension modules in all corners shown in Fig. 3. The standard configuration is good for hard work in rough terrain. It can carry heavy payloads and has great pulling power.

A robot for phenotyping does not need the power of the standard Thorvald II robot. By reducing the number of motors and gears, the total cost of the robot will drop significantly. The ground clearance of the standard Thorvald

Fig. 2: Examples of a few different robots made from Thorvald II modules. The standard configuration is on the far left.

II robot is also too small for driving over fully grown wheat plants. For these reason a different configuration was chosen.

Fig. 3: The standard configuration of the Thorvald II agricultural robot

III. THORVALD II CONFIGURED FOR PHENOTYPING

As we wish to keep the costs of the robot to a minimum, a two-wheel differential drive robot design is chosen. Two drive modules are connected directly to the front of the robots frame, with caster wheels at the rear. A tall custom arch connects the sides of the platform together. The resulting robot is tall enough for driving over fully grown wheat plants without harming them, and it also has six less motors and gears than the standard Thorvald II configuration, which means that it is less costly to make. The presented robot is shown in Fig. 4.

A. Task Specific Modifications

Wheat plants tend to lean over when they grow tall, effectively covering the wheel tracks in between plots. It is therefore important that the robot is able to separate the

Fig. 4: A Thorvald II robot configured for wheat phenotyping

plants in neighboring plots by gently pushing them to either side when it is drives through the field. This is important so that the plants do not get run over or tangled in the robot's wheels or frame. To deal with this problem the robot's front wheels and lower frame are fitted with custom covers. The wheel covers stick out in front of the robot and tapers at an angle of 60° from the horizontal plane. This helps separate plants from neighboring plots so that the robot can pass through without causing damage.

As the spacing between the plots in this particular research field is small, about 15 cm, the 16 cm wide standard Thorvald II wheels are replaced by 10 cm narrow wheels. Fig. 5 shows the the robot in the field.

B. Configuration Review

The main disadvantage with the selected configuration compared to the standard Thorvald II configuration is the considerable loss off traction that results from having two less drive modules and no modules for suspension. If one of the drive wheels loses traction, the robot may turn unexpectedly. The robot's control system will try to compensate for this, but if the problem remains, the robot is effectively stuck. However, as the arch that makes up the upper part of the robot's frame is not completely stiff, it will allow some flex. This helps in keeping the drive wheels in contact with the ground.

Another disadvantage of the proposed robot design is the

Fig. 5: The robot in the wheat field. The narrow spacing between the plots require precise navigation

loss of maneuverability which results from omitting steering modules and opting for differential drive. On the standard Thorvald II robot, steering modules allow 360° independent rotation of each drive module about the vertical axis. This makes the standard configuration far more maneuverable than the presented phenotyping robot. When the presented robot turns, the rear will sling out to the side. If the robot decides to adjust it's heading too vividly in between plots, it can potentially harm plants.

A fortunate consequence of using the Thorvald II system is that the Thorvald platform easily can be fitted with more drive modules, suspension modules or modules for independent steering of each drive module. If the somewhat sparse original design turns out not to be suited for the task, the robot can easily be upgraded with the needed modules. Fig. 6 shows a tall Thorvald II configuration with four-wheel drive and four-wheel steering that was assembled before the authors decided on the presented design. This particular robot was created by reconfiguring the initial Thorvald II prototype, a robot that has seen several configuration since the one shown here.

IV. CAMERA SETUP

Two pairs of cameras are mounted to the top of the robot's frame. One camera pair is facing directly down, while the other pair is mounted to the side at a 55° angle. Each camera pair consists of one RGB camera and one monochromatic IR-camera. The cameras connect to the robot's Ethernet network and are triggered from the robots main computer. All four cameras fire simultaneously when triggered and the captured images are stored locally on

Fig. 6: A tall Thorvald II configuration with four-wheel drive and four-wheel steering. This robot is not fitted with row dividing covers.

the robot. A rectangular piece of polytetrafluoroethylene (PTFE) is mounted in the field of view of both camera pairs as a universal white reference present in all images. In our aforementioned previous work, using only cameras positioned vertically towards the ground, we noticed that the images suffered from a few very exposed points at the top of the plants and shadowed parts underneath. With our current configuration the cameras facing down are reliable as long as the crop is short while the angled cameras improve the measurement's reliability as the crop grows tall, as shown in Fig. 7.

The cameras used are from Teledyne Dalsa's Genie Nano Series GigE Vision. The models are two Genie Nano-M2420 and two Genie Nano-C2420. The specifications are:

- Resolution of 2448×2048 ;
- Maximum frame rate of 35 fps ;
- Pixel size of $3.45 \mu\text{m}$;
- Output format GigE Vision;
- Size of $21 \times 29 \times 44 \text{ mm}$;
- Mass of 46 g ;

V. NAVIGATION AND AUTONOMOUS DATA GATHERING

Whenever the presented robot is turning, each of the two drive wheels will be tangent to the circle on which each wheel is driving. As the drive wheels are in the front of the robot, this means that the rear will sling out to the side. If the robot decides to turn too aggressively whilst driving in between the plots, it may run over and damage plants. Damaging plants in the research plots is of course not acceptable. This means that the robot needs to have a precise navigation system.

Fig. 7: The pictures from the downward facing camera pair (top) suffer from high exposure concentrated at the apex of the plants, and strong shadows in between plants. The angled cameras improve the reliability of the NDVI measurements where this is the case.

A. Sensors For Navigation

For this setup the the robot runs ROS (Robot Operatig System) and is fitted with a few different sensors. The robot has an Xsens MTi-30 IMU and a Septentrio APS-NR2 RTK-GNSS receiver, and the robot's motors are fitted with encoders. The GNSS system receives correction signals through CPOS, a service provided by the Norwegian Mapping Authority. At the location of testing we obtained stable positions and had accuracy on a centimeter level. The robot also calculates its absolute heading from GNSS position readings over time whenever driving in a straight line. The heading estimate is maintained by the IMU when turning. Translational velocity calculated from the encoders and rotational velocity (about the vertical axis) from the IMU are fused with GNSS positions and absolute heading estimate calculated from the GNSS readings to produce a accurate pose estimate.

In the future, the robot will be equipped with a LIDAR for obstacle detection, similar to what is used on other Thorvald II robots.

B. Running the Robot

The seeds in the research plots were seeded by a manually driven research seeder. As a result the rows are not straight and the exact position of the rows are not known.

The robot rely on GNSS for localization, and navigates by means of waypoints. The waypoints are recorded manually by driving the robot across the field. In more detail, the robot is programmed to run at constant speed, with an operator manually adjusting the steering. For safety, the operator

also has the possibility of pausing the robot, or switching it to fully teleoperated mode. The robot records its pose for every 10 cm of travel. Whenever the robot comes to a position selected for data collection, the operator presses a button to record the robot's pose. The recorded poses are ordered in sets corresponding to the different rows in the field, and are used as waypoints by the robots navigation system. The full path can be created by combining sets of waypoints made from different recordings, and different speeds can be selected for each sub-path. After the path has been established, each time the robot is scheduled to take pictures, the operator will bring the robot to the correct field and start the program. The robot will then automatically follow the predetermined path and record data at the correct locations.

C. System Verification

A path was recorded in one of the wheat fields at the NMBU research farm, Vollebekk. Points for image capturing were selected by the operator during the recording. The robot was able to replicate the path and record pictures at the selected locations. All system worked as intended, which means that the robot soon will be transferred to its permanent location at the NMBU research farm, where it will be put to good use in phenotyping projects.

VI. CONCLUDING REMARKS

In this paper we have illustrated how robots made from modules help us in quickly adapting to new environments. Here, only a few custom parts were needed to create a robot completely different from the standard design. The fact that the robot can be reconfigured decreases the level of risk involved when designing a task specific robot such as the one presented here. If the robot does not behave as intended, the design can easily be tweaked or even changed beyond recognition with only basic tools. It is our firm belief that modular robots significantly speed up the process of determining the best design for a given application.

A. Future Work

In the future the robot will be equipped with more sensors for data collection. Effort will be put into making the robot user friendly, as most users of the robot will be plant researchers, not roboticists. The robot will also be fitted with other equipment. A sprayer for automated application of pesticides on research plots has already been developed and is ready for field trials.

REFERENCES

- [1] A. Walter, F. Liebisch, and A. Hund, "Plant phenotyping: from bean weighing to image analysis," *Plant Methods*, vol. 11, p. 14, Mar 2015.
- [2] J. Underwood, A. Wendel, B. Schofield, L. McMurray, and R. Kimber, "Efficient in-field plant phenomics for row-crops with an autonomous ground vehicle," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1061–1083, 2017.
- [3] A. Ruckelshausen, P. Biber, M. Dorna, H. Gremmes, R. Klose, A. Linz, F. Rahe, R. Resch, M. Thiel, D. Trautz, and U. Weiss, "Boniroban autonomous field robot platform for individual plant phenotyping," *Precision agriculture*, vol. 9, no. 841, 2009.

- [4] I. Burud, G. Lange, M. Lillemo, E. Bleken, L. Grimstad, and P. J. From, "Exploring robots and uavs as phenotyping tools in plant breeding," in *IFAC 2017 World Congress*, 2017.
- [5] L. Grimstad, C. D. Pham, H. N. T. Phan, and P. J. From, "On the design of a low-cost, light-weight, and highly versatile agricultural robot," in *2015 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO 2015)*, 2015.
- [6] L. Grimstad and P. J. From, "Thorvald ii - a modular and re-configurable agricultural robot," in *IFAC 2017 World Congress*, 2017.