Towards 4D Crop Analysis in Precision Agriculture: Estimating Plant Height and Crown Radius over Time via Expectation-Maximization

Luca Carlone, Jing Dong, Stefano Fenu, Glen C. Rains, Frank Dellaert

***Abstract*— In previous work [1] we showed how to apply modern sensor fusion and computer vision techniques to obtain a dense 3D reconstruction of a crop for precision agriculture. In this work we consider the case in which we have multiple 3D reconstructions, obtained from sensor data collected over several weeks. From this collection of dense reconstructions, we want to estimate how the size and height of each plant evolve over time. The problem is challenging since the 3D reconstructions may contain very partial views of each plant. Moreover, the presence of multiple plants (and background) requires solving the *data association* problem, which makes our goal even more challenging. We propose a general probabilistic model to estimate *shape* and *appearance* of objects (the plants) using *factor graphs*. Then, we tailor the formulation to precision agriculture, and we show that the choice of a suitable parametrization and the use of *expectation-maximization* enables fast inference on plant growth. Our approach provides high-level description of the status of each plant, and this can inform the farm manager and enhance situational awareness. Current results are extremely encouraging, and open several avenues for future research.**

I. Introduction

The goal of *precision agriculture* (PA) is to provide farm managers with relevant information about the status of the crop. The resulting awareness supports timely decision making, allowing early treatment of biotic (e.g., pests, insects, or pathogens) and abiotic (e.g., inadequate moisture or nutrition) stresses, thereby preventing yield loss [2].

Standard techniques in agriculture are based on manual measurements or soil sampling from a human operator. This approach is expensive, labor intensive, and provides a very sparse overview on the status of the crop. Early attempts to automate the monitoring process were based on satellite imaging [3]. These techniques have significant limitations in terms of resolution as well as coverage in space and time. Advances in mobile robotics are offering an effective alternative: the farm manager can adopt one or more robots (*Unmanned Ground Vehicles* (UGV) or *Unmanned Aerial Vehicles* (UAV)) to perform continuous monitoring of the crops. These vehicles are relatively inexpensive and provide high resolution measurements from on-board sensors.

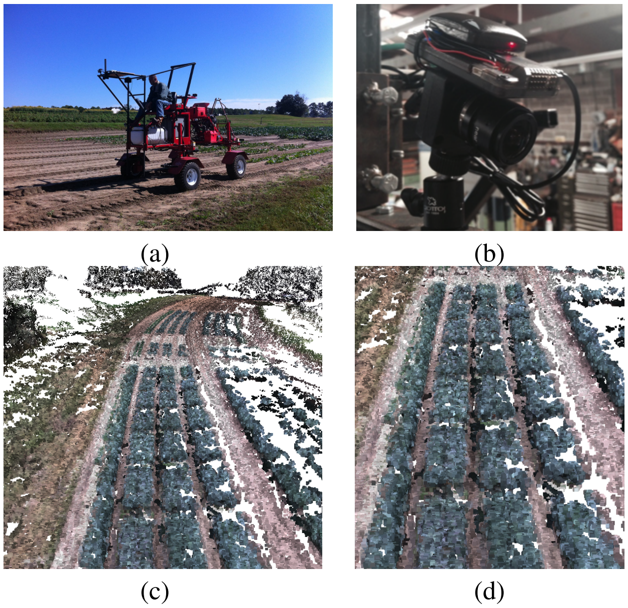


Fig. 1. (a) Rover used for data collection; (b) Sensor equipment: high resolution camera, Inertial Measurements Unit (IMU), and GPS. (c) Example of 3D reconstruction produced by our system. (d) Zoomed view.

As discussed in Section II, a large body of related work deals with the use of robotic or computer vision techniques to support autonomous guidance of robots operating on the field. In this work, instead, we are interested in perception and monitoring. Our objective is to provide farm managers with interpreted information about the crops that can be directly used for decision making. The contribution of the present paper leverages our previous results [1].

In [1], we discussed the use of sensor fusion and computer vision techniques for PA. Our crop monitoring system relies on a tractor rover to collect data in the field (Fig. 1a). The tractor is equipped with a high resolution camera, an Inertial Measurements Unit (IMU), and a GPS (Fig. 1b). The tractor collects the data weekly and we use *SLAM* (Simultaneous Localization and Mapping) techniques [4] and *multi-view stereo* algorithms [5], [6] to obtain a dense 3D reconstruction (a colored point cloud) from images and sensor data (Fig. 1c-d). Each 3D reconstruction provides a snapshot of the field at a given moment, and the collection of 3D reconstructions over time provides a 4D model of the crop.

The motivation for the present work is that, while 3D point clouds provide a detailed view of the field, it is hard for a non-expert user to interpret those and obtain information of interest from them. Even more difficult is the task of detecting patterns from multiple 3D reconstructions over time. For this reason, the goal of this paper is to use and *interpret* the set of 3D reconstructions produced by the system [1]. Ideally, the farmer is interested in tracking the size (*crown radius*), the height, and the color, for each plant, to detect subnormal growth rates or discoloration. This problem is challenging: each point cloud contains millions of

L. Carlone, J. Dong, S. Fenu, and F. Dellaert are with the College of Computing, Georgia Institute of Technology, USA. {luca.carlone,jdong,sfenu3}@gatech.edu, frank@cc.gatech.edu

G. C. Rains is with the Department of Entomology, University of Georgia, USA. grains@uga.edu

This work was partially funded by the NSF Award 11115678 "RI: Small: Ultra-Sparsifiers for Fast and Scalable Mapping and 3D Reconstruction on Mobile Robots", and by the USDA NIFA Award GEOW-2014-09160 "Multipurpose robotic sampling to optimize crop production".

points and we need to solve for the *data association* (which point corresponds to which plant) and infer the shape and appearance of a plant from a set of points.

We propose to formulate 4D plant description in terms of joint inference over a factor graph. We simultaneously perform inference over the *labels* which associate each point to a plant, and plant shape and appearance. This formulation is general and applicable to the 4D description of generic objects. Then we tailor this formulation to precision agriculture and show that the choice of a specific parametrization for the shape of a plant, together with the use of *expectation-maximization* [7], enables fast inference.

As a by-product of our pipeline (Fig. 2), we can easily provide other information, such as canopy size and height, which is also a major goal in related work, see, e.g., [8].

We test our approach on both simulated and real data collected in the field. We are able to correctly estimate the evolution of size and height of the plants. The approach currently needs some prior information (the number of plants in each canopy), and in the conclusion of this work we discuss our current effort towards relaxing these assumptions.

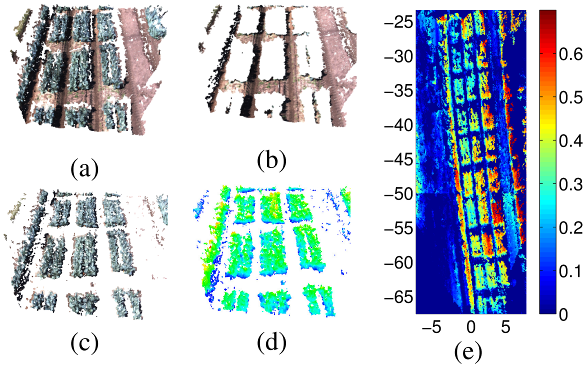


Fig. 2. As a by-product of our pipeline we get a segmentation of the field in canopies, from which we can compute canopy height. (a) original point cloud. Using ground plane extraction we distinguish the ground (b) from the plants (c), and compute the height of points in each canopy (d-e).

II. Related Work

A first set of related work deals with **robotics and computer vision in precision agriculture**. A large body of work focuses on the use of vision to guide a UAV while weeding, spraying, or fertilizing [9], [10], [11], [12], [13], [14], [15]. Along this line, we also find contributions on actuation [16] (including vision-aided fruit picking [17], [18], [19]), control [20] and coverage [21], [22], [23]. Less attention has been devoted to estimation aspects (yield evaluation, field monitoring, data interpretation). Related work on produce monitoring and maturity evaluation are usually designed for specific fruits, e.g., mangoes [24], apples [25], grape [26], or to detect weeds [27]. Polder and Hofstee [28] propose a plant phenotyping system based on light-field camera. Anthony *et al.* [8] propose an approach to estimate crop height using a 2D laser scanner mounted on a UAV.

The second area of related work is **4D reconstruction**. Schindler *et al.* [29], [30] consider the problem of understanding urban development from a set of historical images. Pollard and Mundy [31], [32] propose a voxel-based approach to detect changes over time. Kosecka [33] detects changes in urban environments from Google Street view images. Sakurada *et al.* [34] detect changes by comparing depth estimates from multiple cameras.

The third branch of related work regards **object detection** in point clouds. This body of work usually assumes the availability of object templates and search for those templates in a point clouds using local 3D shape features [35], [36], [37], [38], voting schemes [39], 3D shape contexts [35], 3D Zernike descriptors [36], and spin image signatures [37], [38]. Recent work focuses on the use of large collections of weakly labeled data from the web as templates [40], [41], [42], [43]. de Figueiredo *et al.* [44] propose a grid-based Bayesian filter for multiple object detection. Golovinskiy *et al.* [45] propose to perform object recognition by a sequence of steps, including clustering, segmentation, and classification. Koppula *et al.* [46] investigate object detection via approximate inference on a graphical model, to detect objects in cluttered indoor scenarios. Patterson *et al.* [47] use both global and local descriptors.

Our work differs from related work in many aspects. Our goal is similar to [8], but our system relies on a high resolution camera, rather than a 2D laser. This choice has multiple advantages: monocular cameras are cheap and allow to obtain a dense reconstruction of the field, which provides a large amount of information. Contrarily to the literature on 4D mapping, we reason in terms of objects (the plants), rather than voxels: this is similar in spirit to SLAM++ [48], however, we do not have templates for the objects, and our objects are time-varying (the plants grow over time).

III. A General Formulation to Track Shape and Appearance of Objects over Time

Our objective is to track the size and the height of each plant in the field over time, given a set of 3D reconstructions. Towards this goal, we first propose a general formulation that is agnostic to the specific application in PA.

A. Problem Statement

A general statement of the problem discussed in this section is as follows: *given a set of 3D point clouds, representing the same scene at different time instants, our goal is to track objects of interest, and provide a* description *(e.g., position, shape, appearance) of each object over time.*

The input to our problem is a set of 3D point clouds $\mathcal{P} \doteq \{\mathcal{P}_t : t = 1, \ldots, T\}$. The index t is a time stamp: each point cloud $\mathcal{P}_t$ describes a 3D reconstruction of the environment at time t. The environment is assumed to be reasonably static during a single data collection, but reconstructions at different times (i.e., for different t) may be different, because of the presence of time-varying elements in the environment. Each point cloud contains N_t colored points: $\mathcal{P}_t \doteq \{\boldsymbol{p}_j^t \doteq (\boldsymbol{z}_j^t, \boldsymbol{c}_j^t) : j = 1, \ldots, N_t\}$, where $\boldsymbol{z}_j^t$ is the 3D position of the point and $\boldsymbol{c}_j^t$ is its (rgb) color.

Each point cloud $\mathcal{P}_t$ contains partial and noisy measurements of a set of M_t *objects*: $\mathcal{O}_t = \{\mathcal{O}_i^t : i = 1, \ldots, M_t\}$. An object i is *described* by a pair: $\mathcal{O}_i^t \doteq (\mathcal{S}_i^t, \mathcal{A}_i^t)$, where $\mathcal{S}_i^t$ is the *shape* of the object and $\mathcal{A}_i^t$ is its *appearance*. The label t in $(\mathcal{S}_i^t, \mathcal{A}_i^t)$ remarks that both quantities are time-varying.

The output of our approach is an estimate of shape $\mathcal{S}_i^t$ and appearance $\mathcal{A}_i^t$ for all objects i and for all times t.

B. Probabilistic Formulation using Factor Graphs

In this section we model the problem as a *factor graph*. Factor graphs are a general representation of estimation problems in terms of graphical models [49]. A factor graph includes a set of *hidden variables* (that we want to estimate) and a set of *factors*, that encode probabilistic relations among the hidden variables. The factors essentially model available measurements (and prior knowledge) on the hidden variables.

In our problem, the set of hidden variables is $\mathcal{O} \doteq \{\mathcal{O}_i^t : t = 1,\ldots,T;\ i = 1,\ldots,M_t\}$ (the description of the objects over time), while the available measurements are the set of point clouds $\mathcal{P} \doteq \{\boldsymbol{p}_j^t : t = 1,\ldots,T;\ j = 1,\ldots,N_t\}$.

Standard inference over factor graphs is performed via *maximum a-posteriori* (MAP) estimation: the best estimate $\mathcal{O}^\star$ for the objects $\mathcal{O}$ is the one maximizing the posterior probability of $\mathcal{O}$ given the measurements $\mathcal{P}$:

$$\mathcal{O}^\star = \arg\max_{\mathcal{O}} \ \mathbb{P}\left(\mathcal{O} \mid \mathcal{P}\right) \tag{1}$$

We assume to have a nominal model that tells us how $\mathcal{O}_i^t$ varies from time $t-1$ to time t:

$$\mathbb{P}\left(\mathcal{O}_i^t \mid \mathcal{O}_i^{t-1}\right) \tag{2}$$

This is usually referred to as a *process model* in robotics [50]. In PA, the reader may think about (2) as a model for the *nominal* growth of each plant (more details in Section IV).

Ideally, we would also like to have a mathematical model that relates the points in $\mathcal{P}$ to the unknowns $\mathcal{O}$ (*measurement model*). However, it is hard to write all point measurements as an analytic function of all objects. Instead, we can describe the point measurements corresponding to the observation of a specific object. We define a measurement model:

$$\mathbb{P}\left(\boldsymbol{p}_j^t \mid \mathcal{O}_{\mathcal{L}_j^t}^t, \mathcal{L}_j^t\right) \tag{3}$$

where $\mathcal{L}_j^t$ is the point *label*, that assigns an object id i to each point observation $\boldsymbol{p}_j^t$. The measurement model (3) describes the probability of measuring a point $\boldsymbol{p}_j^t$ from a specific object (specified by $\mathcal{L}_j^t$), having given shape and appearance (specified by $\mathcal{O}_{\mathcal{L}_j^t}^t$).

The fact that the point labels $\mathcal{L} \doteq \{\mathcal{L}_j^t : t = 1,\ldots,T;\ j = 1,\ldots,N_t\}$ are unknown requires to include those as part of the inference process. Assuming independent noise in the process model (2) and in the measurement model (3), we can write the joint posterior $\mathbb{P}\left(\mathcal{O},\mathcal{L} \mid \mathcal{P}\right)$ as [50]:

$$\mathbb{P}\left(\mathcal{O},\mathcal{L} \mid \mathcal{P}\right) = \tag{4}$$

$$\mathbb{P}\left(\mathcal{O}^1\right) \prod_{t=1}^{T} \left\{ \prod_{i=1}^{M_t} \left[\mathbb{P}\left(\mathcal{O}_i^t \mid \mathcal{O}_i^{t-1}\right) \prod_{j=1}^{N_t} \mathbb{P}\left(\boldsymbol{p}_j^t \mid \mathcal{O}_{\mathcal{L}_j^t}^t, \mathcal{L}_j^t\right) \right] \right\}$$

where $\mathbb{P}\left(\mathcal{O}^1\right)$ denotes probability priors (if available).

In order to get the MAP estimate (1), we have to marginalize the latent variables $\mathcal{L}$ from (4):

$$\mathcal{O}^\star = \arg\max_{\mathcal{O}} \sum_{\mathcal{L}} \mathbb{P}\left(\mathcal{O},\mathcal{L} \mid \mathcal{P}\right) \tag{5}$$

The marginalization in (5) essentially means that we are looking for the estimate $\mathcal{O}$ that best explains the measurements $\mathcal{P}$, considering all possible data association outcomes.

The formulation described so far is very general, but it is intractable in general, due to the marginalization (5) [51]. In order to write explicit expressions for the process and measurement models, we need to choose a parametrization for shape and appearance of the objects. For this reason, in Section IV-A we commit to a parametrization (which is tailored to our PA application). Then, in Section V we apply expectation-maximization to make the computation of the MAP estimate (5) tractable.

IV. Tailoring the Formulation to Precision Agriculture

Here we choose a parametrization for the shape and appearance of the plants which enables to write simple expressions for the probabilities in (4).

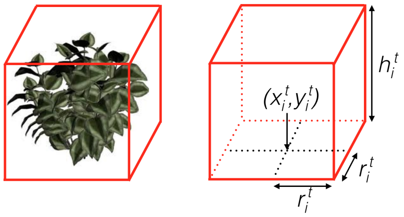


Fig. 3. We parametrize the shape of each plant i (at time t) using the geometry of the corresponding bounding box. The bottom/top of the box is a square with side $2r_i^t$, and center (x_i^t, y_i^t). The height is h_i^t.

A. A Simple Descriptor for Plants

We are interested in the size of the crown and the height of each plant in the field. Therefore the simplest shape descriptor is a *box*. We characterize each plant by the size/height of the corresponding bounding box (Fig. 3). The box is defined by 4 parameters $x_i^t, y_i^t, r_i^t, h_i^t$. The base of the bounding box is a square of side $2r_i^t$. The parameters (x_i^t, y_i^t) describe the Cartesian position of the center of this square; h_i^t is the height of the box.

We parametrize the appearance of an object as the average (rgb) color $\boldsymbol{c}_i^t \in \mathbb{R}^3$ of the points in its bounding box.

We stack these parameters in a vector:

$$\boldsymbol{o}_i^t = [x_i^t \ \ y_i^t \ \ r_i^t \ \ h_i^t \ \ \boldsymbol{c}_i^t] \tag{6}$$

We call $\boldsymbol{o}_i^t$ a *descriptor* of plant i at time t.

B. Process model

The process model (2) describes a nominal evolution of plant descriptor over time. For each plant, the pair (x_i^t, y_i^t) is an estimate of the position of the root of the plant, and this is stationary over time:

$$x_i^t = x_i^{t-1} + \nu_x, \qquad y_i^t = y_i^{t-1} + \nu_y \tag{7}$$

where the random noise terms ν_x and ν_y can accomodate small changes in (x_i^t, y_i^t), e.g., due to external actions.[1]

We assume to have a nominal model that describes the evolution of the height h_i^t and the crown radius r_i^t of a plant

[1] If one omits the noise terms, eq. (7) becomes a deterministic relation, that can be still used within our framework as it only reduces the number of hidden variables.

over time. Related literature offers models for the evolution of those quantities, see, e.g., [52]. We write this model as:

$$r_i^t = r_i^{t-1} + n_r + \nu_r, \qquad h_i^t = h_i^{t-1} + n_h + \nu_h \tag{8}$$

where n_r and n_h are nominal growth factors and ν_r and ν_h are noise terms, that account for random mismatches w.r.t. the nominal growth model. Since the terms n_r and n_h are possibly time-varying, the model (8) is fairly general.

Similarly, the change in appearance is modeled as:

$$\boldsymbol{c}_i^t = \boldsymbol{c}_i^{t-1} + \boldsymbol{n}_c + \boldsymbol{\nu}_c \tag{9}$$

where $\boldsymbol{n}_c$ and $\boldsymbol{\nu}_c$ describe nominal color change and random noise, respectively.

Equations (7)-(9) describe -for each object i- the nominal evolution over time of all the parameters we want to estimate. Putting these relations in compact matrix form, and assuming Gaussian noise, we obtain the following process model:

$$\mathbb{P}\left(\mathcal{O}_i^t \mid \mathcal{O}_i^{t-1}\right) = \alpha \exp\left\{\frac{1}{2}\left\|\boldsymbol{o}_i^t - \boldsymbol{A}_i^t \boldsymbol{o}_i^{t-1} - \boldsymbol{b}_i^t\right\|^2_{\boldsymbol{\Phi}_i^t}\right\} \tag{10}$$

where the matrix $\boldsymbol{A}_i^t$ and the vector $\boldsymbol{b}_i^t$ write eqs. (7)-(9) in compact form, α is a normalization constant (irrelevant for the subsequent development), and $\boldsymbol{\Phi}_i^t$ is the covariance of the noise vector $[\nu_x \; \nu_y \; \nu_r \; \nu_h \; \boldsymbol{\nu}_c]$.

C. Measurement model

The measurement model (3) describes the likelihood of a measurement $\boldsymbol{p}_j^t$, given that the observation comes from a given object $\mathcal{O}_i^t$. Since we parametrized the object using the descriptor $\boldsymbol{o}_i^t$, the measurement model describes the relation between a point measurement and the descriptor. We recall that each measurement $\boldsymbol{p}_j^t$ is a colored point: $\boldsymbol{p}_j^t \doteq (\boldsymbol{z}_j^t, \boldsymbol{c}_j^t)$. We consider the following model for the measured appearance:

$$\boldsymbol{c}_j^t = \boldsymbol{c}_i^t + \boldsymbol{\epsilon}_c \tag{11}$$

which means that the measured color of the point $\boldsymbol{p}_j^t$ corresponds to the appearance of the objects (its average color) plus random noise $\boldsymbol{\epsilon}_c$. Note that we can account for large intra-object color variability by operating on the covariance of the noise term $\boldsymbol{\epsilon}_c$.

The position $\boldsymbol{z}_j^t$ of the 3D point is more difficult to model as it requires disambiguating points on the top of the bounding box from points on the sides. For instance, a point on the top of the box will be described by:

$$\boldsymbol{a}_h^{\mathsf{T}} \boldsymbol{z}_j^t = h_i^t + \text{noise} \tag{12}$$

where $\boldsymbol{a}_h$ is a vector that selects the height of the point from $\boldsymbol{z}_j^t$. Eq. (12) says that the height of the point is a noisy measurement of the height of the box.

On the other hand, a point on the lateral sides of the box is described by a different measurement model:

$$\boldsymbol{a}_x^{\mathsf{T}} \boldsymbol{z}_j^t = x_i^t \pm r_i^t + \text{noise} \tag{13}$$

where $\boldsymbol{a}_x$ select the x component from the point $\boldsymbol{z}_j^t$, and we use $+r_i^t$ if the point is on the right side, and $-r_i^t$ if it is on the left. Roughly speaking, a point on the right (resp. left) of the box has a positive (resp. negative) shift of r_i^t wrt to the center of the box, see Fig. 3.

The general expression that relates the measured point position $\boldsymbol{z}_j^t$ to the descriptor $\boldsymbol{o}_i^t$ is the following:

$$\begin{bmatrix} b_x & b_y & b_h \end{bmatrix} \boldsymbol{z}_j^t = \begin{bmatrix} b_x & b_y & b_r & b_h & \mathbf{0} \end{bmatrix} \boldsymbol{o}_i^t + \epsilon_z \tag{14}$$

where the variables $b_x, b_y, b_h \in \{0, 1\}$ and $b_r \in \{-1, 0, +1\}$. For suitable values of the integer variables $\boldsymbol{b}_j^t = [b_x \; b_y \; b_r \; b_h]$, the model (14) describes points observations of every side of the box (excluding the bottom which is not of interest for our application).

In order to write an analytic expression for the measurement model (14), we had to introduce extra (integer) variables $\boldsymbol{b}_j^t$, hence the likelihood (3) becomes:

$$\mathbb{P}\left(\boldsymbol{p}_j^t \mid \mathcal{O}^t_{\mathcal{L}_j^t}, \mathcal{L}_j^t, \boldsymbol{b}_j^t\right) = \beta \exp\left\{\frac{1}{2}\left\|\boldsymbol{F}_j^t \boldsymbol{p}_j^t - \boldsymbol{G}_j^t \boldsymbol{o}_i^t\right\|^2_{\boldsymbol{\Omega}_j^t}\right\} \tag{15}$$

where β is a normalization constant, $\boldsymbol{\Omega}_j^t$ is the measurement covariance, and the matrices $\boldsymbol{F}_j^t$ and $\boldsymbol{G}_j^t$ are functions of the integers $\boldsymbol{b}_j^t$ and write eq. (14) in compact form.

D. Priors

In precision agriculture it is not uncommon to record the GPS position of the seeds at the moment of planting. In our experiments we do not use priors, but this information, when available, can be included in our formulation (4):

$$\mathbb{P}\left(\mathcal{O}^1\right) = \prod_{i=1}^{M_1} \mathbb{P}\left(\mathcal{O}_i^1\right) = \prod_{i=1}^{M_1} \gamma_i \exp\left\{\frac{1}{2}\left\|\boldsymbol{P}_i \boldsymbol{o}_i^1 - \boldsymbol{s}_i\right\|^2_{\boldsymbol{\Psi}_i^t}\right\} \tag{16}$$

where the matrix $\boldsymbol{P}_i$ selects the (x, y) position of object i from the descriptor $\boldsymbol{o}_i^1$, the vector $\boldsymbol{s}_i$ contains the GPS measurement of the seed, and γ_i, $\boldsymbol{\Psi}_i^t$ are a normalization constant and a given covariance matrix, respectively.

V. Computing the MAP Estimate using EM

Since the marginalization (5) is intractable in general, in this section we propose an approach based on *expectation-maximization* [7] to compute the MAP estimator.

In Section IV we wrote explicit expressions for the process model (10), measurement model (15), and possible priors (16), for our specific PA application. The probabilistic models are particularly simple, due to our choice of the descriptors (6), however, it was necessary to introduce extra variables $\boldsymbol{b}_j^t$ for each point. We call these extra variables $\mathcal{B} \doteq \{\boldsymbol{b}_j^t : t = 1, \ldots, T, \; j = 1, \ldots, N_t\}$. With the introduction of these variables the MAP estimator (5) becomes

$$\mathcal{O}^\star = \arg\max_{\mathcal{O}} \sum_{\mathcal{L},\mathcal{B}} \mathbb{P}\left(\mathcal{O}, \mathcal{L}, \mathcal{B} \mid \mathcal{P}\right) =$$

$$\mathbb{P}\left(\mathcal{O}^1\right) \prod_{t=1}^{T} \left\{ \prod_{i=1}^{M_t} \left[\mathbb{P}\left(\mathcal{O}_i^t \mid \mathcal{O}_i^{t-1}\right) \prod_{j=1}^{N_t} \mathbb{P}\left(\boldsymbol{p}_j^t \mid \mathcal{O}^t_{\mathcal{L}_j^t}, \mathcal{L}_j^t, \boldsymbol{b}_j^t\right) \right] \right\} \tag{17}$$

where we have to marginalize out the latent variables $\mathcal{L}$ (which associate a point to an object) and $\mathcal{B}$ (which associate the point to a specific face of the object).

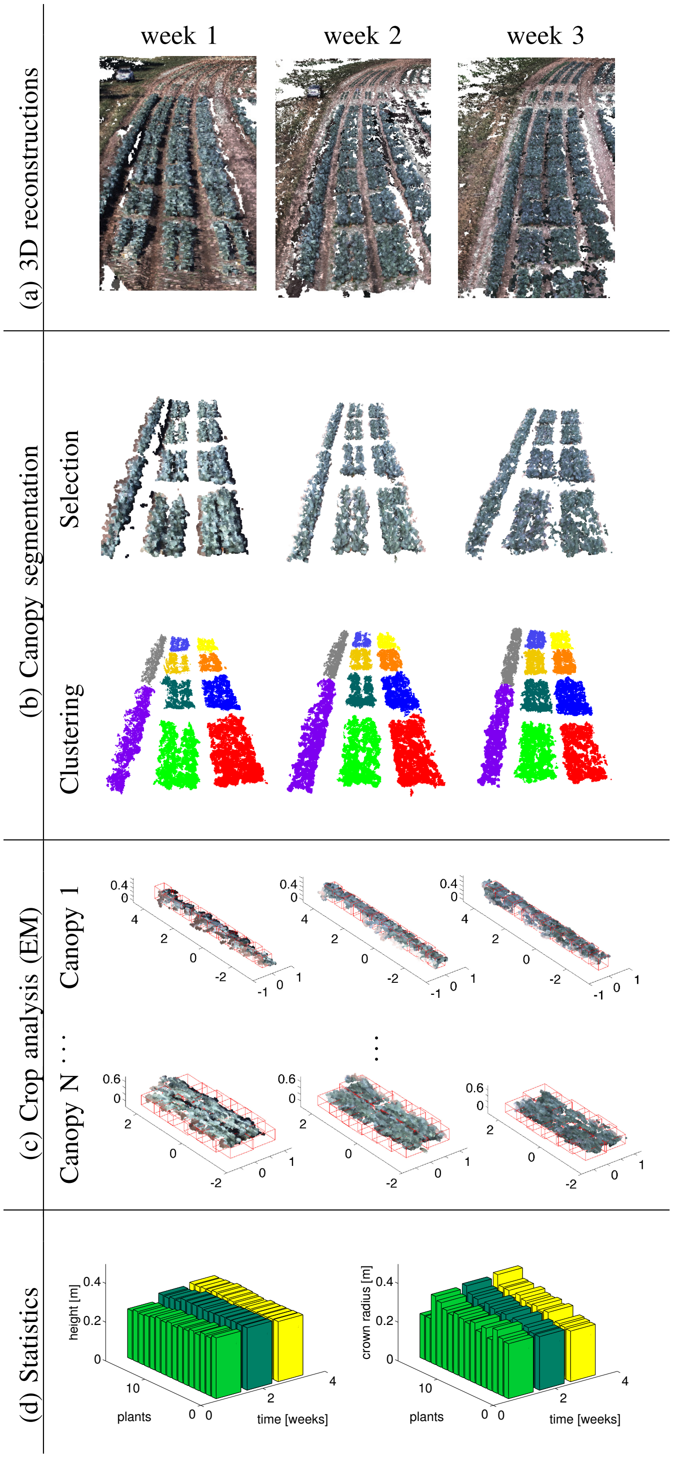


Fig. 4. Overview of our crop analysis pipeline.

Usually, rather than maximizing the posterior probability one minimizes the negative log-posterior:

$$\mathcal{O}^{\star} = \arg\min_{\mathcal{O}} \; -\log\left\{\sum_{\mathcal{L},\mathcal{B}} \mathbb{P}\left(\mathcal{O},\mathcal{L},\mathcal{B} \mid \mathcal{P}\right)\right\} \tag{18}$$

Problem (18) is intractable in general, as the sum would produce a combinatorial number of terms, each one describing a Gaussian posterior, for a particular choice of $\mathcal{L}_j^t, \boldsymbol{b}_j^t$.

We use *Expectation-Maximization* (EM) to make the problem tractable. Rather than solving directly (18), EM solves:

$$\min_{\mathcal{O}} \; \mathbb{E}_{\mathcal{L},\mathcal{B}|\bar{\mathcal{O}}}\left\{-\log\mathbb{P}\left(\mathcal{O},\mathcal{L},\mathcal{B} \mid \mathcal{P}\right)\right\} \tag{19}$$

where $\mathbb{E}_{\mathcal{L},\mathcal{B}|\bar{\mathcal{O}}}$ returns the expected values of $\mathcal{L},\mathcal{B}$ given the current guess $\bar{\mathcal{O}}$. The optimization problem (19) is solved multiple times. At each iteration the problem returns an estimate for $\mathcal{O}$, that is used as guess $\bar{\mathcal{O}}$ for the following iteration. It is known [7] that (19) iteratively converges to a minimum of the original cost (18), therefore we can use EM to compute our map estimate $\mathcal{O}^{\star}$.

The practical advantage in the use of EM is that $-\log\mathbb{P}\left(\mathcal{O},\mathcal{L},\mathcal{B} \mid \mathcal{P}\right)$ can be developed as follows:

$$\begin{aligned} -\log\mathbb{P}\left(\mathcal{O},\mathcal{L},\mathcal{B} \mid \mathcal{P}\right) = -\log\mathbb{P}\left(\mathcal{O}^1\right) + \\ \sum_{t=1}^{T}\left\{\sum_{i=1}^{M_t}\left[-\log\mathbb{P}\left(\mathcal{O}_i^t \mid \mathcal{O}_i^{t-1}\right) + \right.\right. \\ \left.\left.\sum_{j=1}^{N_t} -\log\mathbb{P}\left(\boldsymbol{p}_j^t \mid \mathcal{O}^t_{\mathcal{L}_j^t}, \mathcal{L}_j^t, \boldsymbol{b}_j^t\right)\right]\right\} \end{aligned} \tag{20}$$

Substituting eqs. (10), (15) and (16) in (20) we get:

$$\begin{aligned} -\log\mathbb{P}\left(\mathcal{O},\mathcal{L},\mathcal{B} \mid \mathcal{P}\right) = \sum_{i=1}^{M_1}\left\|\boldsymbol{P}_i\boldsymbol{o}_i^1 - \boldsymbol{s}_i\right\|^2_{\boldsymbol{\Psi}_i^t} + \\ \sum_{t=1}^{T}\left\{\sum_{i=1}^{M_t}\left[\left\|\boldsymbol{o}_i^t - \boldsymbol{A}_i^t\boldsymbol{o}_i^{t-1} - \boldsymbol{b}_i^t\right\|^2_{\boldsymbol{\Phi}_i^t} + \right.\right. \\ \left.\left.\sum_{j=1}^{N_t}\left\|\boldsymbol{F}_j^t\boldsymbol{p}_j^t - \boldsymbol{G}_j^t\boldsymbol{o}_i^t\right\|^2_{\boldsymbol{\Omega}_j^t}\right]\right\} \end{aligned} \tag{21}$$

which is a quadratic cost in the unknown descriptors $\boldsymbol{o}_i^t$.

Therefore each EM iteration involves two steps. First, the computation of the expectation of the quadratic cost (21): this is easy when the expectation is taken over discrete variables as in (21), and the result of the expectation is a quadratic function, see e.g., [53]. Second, we have to minimize the quadratic cost resulting from the expectation step, according to (19): this only requires to solve a sparse linear system.

VI. Practical Implementation and Experiments in Precision Agriculture

In this section we provide a practical description of our current crop analysis pipeline. We refer the reader to the video attachment [54] for an example of execution.

The pipeline is summarized in Fig. 4. The input is a set of 3D point clouds (Fig. 4a). From each point cloud, we first segment the canopies (Fig. 4b) as a preprocessing. Then, for each canopy, we apply the EM algorithm of Section V, which estimates the bounding box of each plant (Fig. 4c).

We report details of the canopy segmentation and on the EM implementation in Sections VI-A and VI-B, respectively. Finally, we describe current results in Section VI-C

A. Preprocessing: Canopy Segmentation

In this section we propose a preprocessing of the point clouds that allows a large computational saving and enables to cope with hilly fields (Fig. 5), in which the height of points in the point clouds can be influenced by a slope of the field.

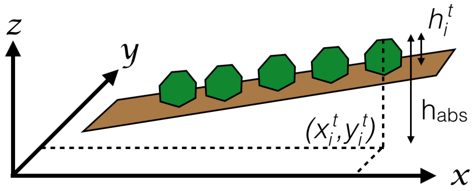


Fig. 5. The points in the original point clouds are expressed in a global reference frame. Therefore, if the field has an upward slope, plants aligned along a row, thus having the same height h_i^t, would produce point observations at different heights h_{abs}. Since the quantity we are interested in is h_i^t, rather than h_{abs}, we fit a local ground plane to the point cloud and we normalize the heights of the points in the cloud w.r.t. this plane.

For each point cloud, this preprocessing includes 3 steps:

1) *Plant selection via ground plane extraction*: we extract the ground plane from the point cloud using a standard RANSAC algorithm. Then, we select the subset of the points in the cloud that are above the ground: these are the points belonging to the plants (Fig. 4b). This largely reduces the number of points as we get rid of uninteresting points, corresponding to the ground. In order to have a better estimate of the ground plane, we divide the field in small patches and we perform the ground plane fitting on each patch.
2) *Height normalization*: using the ground plane computed at the previous step, we normalize the height of the points with respect to the local ground plane, such that the plant heights are expressed w.r.t. ground (Fig. 5).
3) *Canopy clustering*: from the plant segmentation of Fig. 4b, we apply standard Euclidean clustering algorithms [55], that allow to cluster the plants in each canopy, and helps to filter our small imperfections.

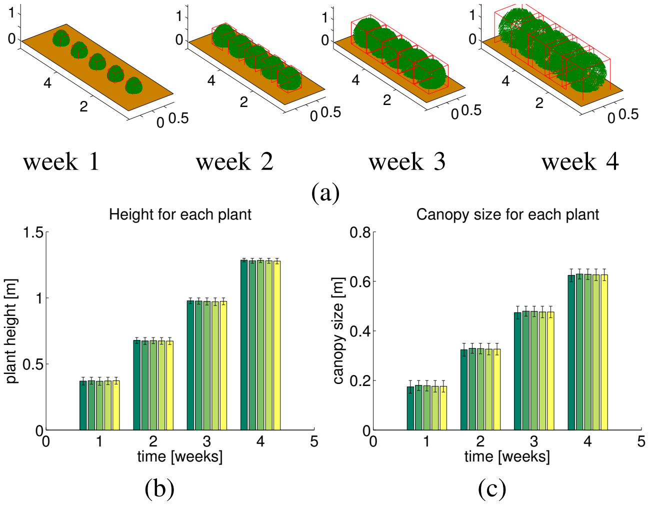
The advantage of canopy clustering is that we can perform EM for each canopy independently. This implies a computational advantage: intuitively, restricting the attention on a single cluster, EM has to search in a smaller space of point-to-plant labels, hence making inference faster.

The output of the preprocessing is a set of point clouds $\mathcal{C}_{\text{canopy}} = \{\mathcal{C}^1_{\text{canopy}}, \ldots, \mathcal{C}^T_{\text{canopy}}\}$, for each canopy. As we remarked in Fig. 2, a useful by-product of this preprocessing is that, after canopy segmentation, we can easily estimate canopy height and size. In the following we go further, and we discuss our EM implementation that estimates crown radius and height *for each plant* in the canopies.

B. Settings for EM

The settings we use for the EM algorithm are as follows.

Initialization. We initialize EM by setting plants position uniformly spaced within the canopy. The algorithm will be then in charge of refining this initial guess. In order to initialize uniformly spaced plants, we need 2 pieces of information: the number of plants in the canopy, and the number of rows along which the plants are arranged. These are the only manual inputs required by our pipeline, and in Section VII we comment on our current effort to automatically estimate those quantities. All plants are initialized to have the same size, which can be inferred from the knowledge of canopy size and number of rows in the canopy. As initialization for plant height, we simply consider the average height of points within a radius (20cm) around the plant position guess.



(a)

(b)

(c)

Fig. 6. Crop analysis on simulated data. We simulate 5 plants growing over 4 weeks and we apply our EM algorithm to the corresponding point clouds. The results are the statistics in figure (b) and (c). Each bar also reports the error intervals in black. Errors are less than 4 cm in both height and canopy size estimates.

EM settings. To implement EM we need to define the following parameters: growth factors n_r, n_h, and the covariance matrices $\mathbf{\Phi}_i^t$, $\mathbf{\Omega}_j^t$, and $\mathbf{\Psi}_i^t$, in eqs. (10), (15), (16). In the following experiments we set $n_r = 0$ and $n_h = 0$ and set all covariances to be identity matrices. Moreover, we do not use any priors on plant descriptors (Section IV-D).

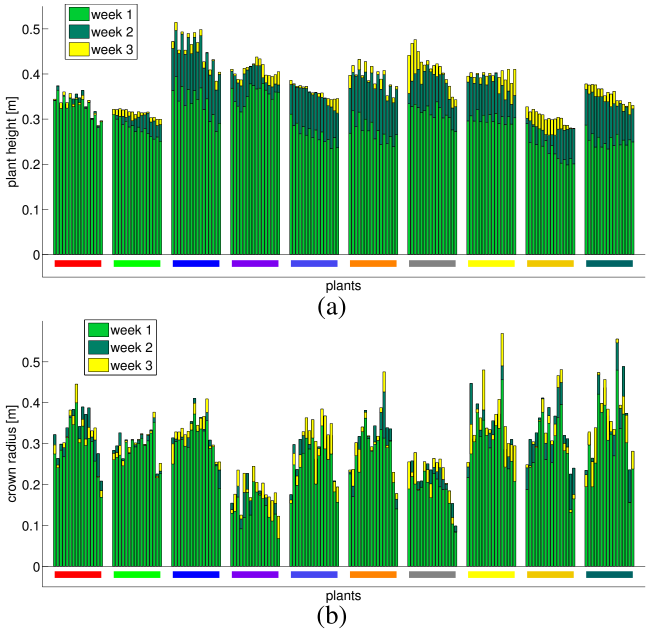
C. Results

We tested our algorithm in both simulated and real data. Results from simulations show that the algorithm is very accurate in estimating plants height and size. Real tests show that the approach can perform crop analysis using field data, and provides reasonable results.

Simulated data. We created 4 point clouds, describing 5 plants growing over 4 weeks, see Fig. 6a. Then we applied the EM algorithm with the settings specified in Section VI-B. The corresponding results are shown in Fig. 6. Fig. 6a shows the red boxes estimated by the EM algorithm, superimposed on the plants. The statistics in Fig. 6b-c show the estimated height and crown radius over time, together with the error intervals in black. Errors are computed as differences between the estimate and the corresponding ground truth (available in simulation). Estimation errors are less than 4cm.

Real data. For the real tests we used the point clouds produced by the system [1]: we performed 3 data collections on a test field in Tifton, Georgia, over 3 consecutive weeks. From sensor data we used the pipeline described in [1] to reconstruct 3 dense point clouds, on which we applied the crop analysis pipeline presented in this paper. Note that while we use the term "week 1" to denote the first data collection, it does not mean that the plants are 1 week old.

All the images in Fig. 4 show intermediate results of our pipeline on the real data. Fig. 4c contains a visualization of the outcome of the EM pipeline for 2 of the 10 canopies in the analysed portion of the field. EM estimates a bounding box for each plant. From the estimated size/height of the



(a)

(b)

Fig. 7. Results from field tests. On the x-axis we plot each plant (arranged by clusters, with the same color code of Fig. 4b). (a) For each plant we show the estimated height at week 1 (light green), week 2 (dark green), and week 3 (yellow). (b) Estimated crown radius for each plant.

bounding box, we can directly plot high-level statistics as the ones in Fig. 4c, which show the evolution of crown radius and height for each plant.

Statistics for all the canopies are shown in Fig. 7; for each plant we plot estimated height and crown radius at week 1,2 and 3. Plants are clustered with the same color code of Fig. 4 (e.g., the blue bar underlines the statistics related to the plants in the blue canopy of Fig. 4).

We currently do not have ground truth data to compare against and we can only observe that the estimates for the height are reasonable for the crops under analysis (cabbage and broccoli). Moreover, the height increases over the second week, while the growth is slower (in average) between week 2 and 3. The statistics show larger fluctuations of the crown radius estimates. This is an intrinsic limitation of the problem: often the point clouds are not dense enough to correctly discern the boundary of each plant, resulting in noisy radius estimates.

VII. Discussion and Future Work

In this paper we propose an approach to estimate crown radius and height for each plant in a field, from a set of dense 3D reconstructions. Our initial results are encouraging: we can correctly segment canopies and obtain reasonable estimates for crown radius and height of each plant.

The current pipeline requires two manual inputs: for each canopy a human operator has to specify the number of plants and the arrangement (1 or 2 rows). Moreover, we initialize EM assuming that the plants are uniformly spaced in the canopies. This assumption is reasonable for the cultures under analysis (broccoli and cabbage, which are manually transplanted at regular distances), but may not be realistic for other cultures (e.g., corn) for which the density of plants can largely vary across the field.

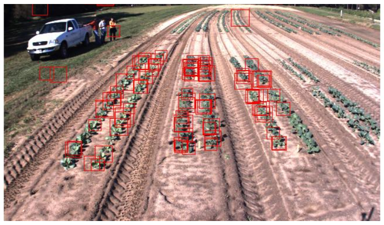


Fig. 8. Preliminary results on image-based plant detection for plants at early stage of growth.

We are currently working towards relaxing both assumptions, towards the goal of having a fully automated crop analysis pipeline that works for general crops.

First, we are working on image-based plant detection algorithms to count the number of plants in each canopy and have a better guess on plant position. While we do not expect to have good results for mature crops using image-based detectors, the detector is expected to perform well at early stages of growth, when the plants are well separated (see our preliminary results in Fig. 8).

Second, we are currently collecting data using a UAV (besides the tractor system of [1]). The UAV allows to cover more points of view with respect to the tractor, hence enabling denser reconstructions. Moreover, the UAV is the only viable alternative to monitor fast growing crops (as corns), that after few weeks are already too tall to be monitored by our tractor system.

We are also exploring other applications of 4D imagining in agriculture. Yield estimates can be made earlier in the season by identifying fruiting structures and flowering patterns. This may be useful in managing contracts with buyers, storage facility availability, labor and machinery logistics. Identifying the size and shape of the plant canopy and the amount of foliage would be beneficial in directing chemical applications at the right locations and in the right volumes to improve pest control efficiency. Variable application of plant growth regulators in crops such as cotton based on 4D imaging could improve fruit production and yield by identifying the locations of excessive plant growth. Weed identification could determine which weeds are in a field, where they are and how fast they are growing, increasing the efficiency of post-emergence herbicides.

References

[1] J. Dong, L. Carlone, G. C. Rains, T. Coolong, and F. Dellaert, "4D mapping of fields using autonomous ground and aerial vehicles," in *2014 ASABE and CSBE/SCGAB Annual International Meeting*, 2014.

[2] E. C. Oerke, "Crop losses to pests," *The Journal of Agricultural Science*, vol. 144, no. 1, pp. 31–43, 2006.

[3] S. Moulin, Bondeau, and R. Delecolle, "Combining agricultural crop models and satellite observations: from field to regional scales," in *International Journal of Remote Sensing*, 1998, pp. 1021–1036.

[4] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, pp. 217–236, Feb 2012.

[5] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 8, pp. 1362–1376, 2010.

[6] T. Treibitz and Y. Schechner, "Instant 3descatter," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 1861–1868.

[7] T. Minka, "Expectation-Maximization as lower bound maximization," November 1998.

[8] D. Anthony, S. Elbaum, A. Lorenz, and C. Detweiler, "On crop height estimation with uavs," in *IROS*, 2014.

[9] J. E. Reid and S. W. Searcy, "An algorithm for computer vision sensing of a row crop guidance directrix," *Transactions of the SAE*, vol. 100, no. 1, pp. 93–105, 1991.

[10] J. N. Wilson, "Guidance of agricultural vehicles a historical perspective," *Computers and Electronics in Agriculture*, vol. 25, no. 1-2, pp. 3–9, 2000.

[11] G. Jiang, C. Zhao, and Y. SI, "A machine vision based crop rows detection for agricultural robots," in *International Conference on Wavelet Analysis and Pattern Recognition*, 2010.

[12] S. Ericson and B. Åstrand, "Row-detection on an agricultural field using omnidirectional camera," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[13] ——, "Avision-guided mobile robot for precision agriculture," in *European Conference on Precision Agriculture*, 2009, pp. 623–630.

[14] X. Jinlin and X. Liming, "Autonomous agricultural robot and its row guidance," in *International Conference on Measuring Technology and Mechatronics Automation*, 2010.

[15] L. C. Lulio, M. L. Tronco, and A. Porto, "JSEG-based image segmentation in computer vision for agricultural mobile robot navigation," in *International Symposium on Computational Intelligence in Robotics and Automation*, 2009, pp. 240–245.

[16] G. T. Tangerino, E. P. Godoy, R. A. Tabile, R. Y. Inamasu, and A. Porto, "Hydraulic networked control of four wheel steering agricultural robot," in *International Conference on Control and Automation*, 2011, pp. 142–147.

[17] M. Nezhad, J. Massh, and H. Komleh, "Tomato picking machine vision using with the open CV's library," in *Iranian Machine Vision and Image Processing Conference*, 2011, pp. 1–5.

[18] F. Qingchun, Z. Wengang, Q. Quan, J. Kai, and G. Rui, "Study on strawberry robotic harvesting system," in *International Conference on Computer Science and Automation Engineering*, 2012, pp. 320–324.

[19] Y. Takahashi, J. Ogawa, and K. Saeki, "Automatic tomato picking robot system with human interface using image processing," in *Annual Conference of the IEEE Industrial Electronics Society*, 2001, pp. 433–438.

[20] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osorio, and P. A. Vargas, "The use of unmanned aerial vehicles and wireless sensor network in agricultural applications," in *IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 5045–5048.

[21] J. Valente, J. D. Cerro, A. Barrientos, and D. Sanz, "Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach," *Computers and Electronics in Agriculture*, vol. 99, pp. 153–59, 2013.

[22] J. Valente, D. Sanz, A. Barrientos, J. D. Cerro, A. Ribeiro, and C. Rossi, "An air-ground wireless sensor network for crop monitoring," *Sensors*, vol. 11, no. 6, pp. 6088–6108, 2011.

[23] A. Barrientos, J. Colorado, J. D. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *J. of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.

[24] T. Chalidabhongse, P. Yimyam, and P. Sirisomboon, "2D/3D vision-based mango's feature extraction and sorting," in *International Conference on Control, Automation, Robotics and Vision*, 2006, pp. 1–6.

[25] P. Yimyam and A. F. Clark, "Agricultural produce grading by computer vision using genetic programming," in *International Conference on Robotics and Biomimetics*, 2012, pp. 458–463.

[26] S. Nuske, S. Achar, T. Bates, S. Narasimhan, and S. Singh, "Yield estimation in vineyards by visual grape detection," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011, pp. 2352–2358.

[27] M. H. Siddiqi, I. Ahmad, and S. B. Sulaiman, "Weed recognition based on erosion and dilation segmentation algorithm," in *International Conference on Education Technology and Computer*, 2009, pp. 224–228.

[28] G.Polder and J. W. Hofstee, "Phenotyping large tomato plants in the greenhouse using a 3D light-field camera," in *2014 ASABE and CSBE/SCGAB Annual International Meeting*, 2014.

[29] G. Schindler, F. Dellaert, and S. Kang, "Inferring temporal order of images from 3D structure," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[30] G. Schindler and F. Dellaert, "Probabilistic temporal inference on reconstructed 3d scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[31] T. Pollard and J. L. Mundy, "Change detection in a 3-d world," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[32] A. Ulusoy and J. L. Mundy, "Image-based 4-d reconstruction using 3-d change detection," in *European Conf. on Computer Vision (ECCV)*, 2014.

[33] J. Kosecka, Y. Ma, S. Soatto, and R. Vidal, "Detecting changes in images of street scenes," in *Asian Conf. on Computer Vision (ACCV)*, 2012.

[34] K. Sakurada, O. Takayuki, and K. Deguchi, "Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[35] M. Körtgen, G. J. Park, M. Novotni, and R. Klein, Eds., *3d shape matching with 3d shape contexts*, 2003.

[36] M. Novotni and R. Klein, "3d zernike descriptors for content based shape retrieval," in *ACM symposium on Solid modeling and applications*, 2003, pp. 216–225.

[37] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 5, pp. 433–449, 1999.

[38] J. Assfalg, M. Bertini, A. D. Bimbo, and P. Pala, "Content-based retrieval of 3D objects using spin image signatures," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 589–599, 2007.

[39] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 998–1005.

[40] K. Lai and D. Fox, "Object recognition in 3D point clouds using web data and domain adaptation," *Intl. J. of Robotics Research*, vol. 29, no. 8, pp. 1019–1037, 2010.

[41] T. Malisiewicz and A. Efros, "Recognition by association via learning per-examplar distances," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[42] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: a database and web-based tool for image annotation," *Intl. J. of Computer Vision*, vol. 77, 2008.

[43] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: a large dataset for non-parametric object and scene recognition," in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 11, 2008, pp. 1958–1970.

[44] R. de Figueiredo, P. Moreno, A. Bernardino, and J.Santos-Victor, "Multi-object detection and pose estimation in 3D point clouds: A fast grid-based Bayesian filter," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 4250–4255.

[45] A. Golovinskiy, V. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," in *Intl. Conf. on Computer Vision (ICCV)*. IEEE, 2009, pp. 2154–2161.

[46] H. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[47] A. Patterson, P. Mordohai, and K. Daniilidis, "Object detection from large-scale 3D datasets using bottom-up and top-down descriptors," in *European Conf. on Computer Vision (ECCV)*, 2008.

[48] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[49] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, February 2001.

[50] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.

[51] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "Structure from motion without correspondence," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2000.

[52] P. J. Peper, E. G. McPherson, and S. M. Mori, "Equations for predicting diameter, height, crown width, and leaf area of san joaquin valley street trees," *J. Arboricult.*, vol. 27, pp. 306–307, 2001.

[53] F. Dellaert, "The expectation maximization algorithm," College of Computing, Georgia Institute of Technology, Tech. Rep. GIT-GVU-02-20, February 2002.

[54] L. Carlone, J. Dong, S. Fenu, G. C. Rains, and F. Dellaert, "Towards 4D crop analysis in precision agriculture: Estimating plant height and crown size over time using factor graphs. supplementary material." [Online]. Available: www.lucacarlone.com/index.php/resources/research/agSensing

[55] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.