Contrastive 3D Shape Completion and Reconstruction for Agricultural Robots using RGB-D Frames

Federico Magistri Elias Marks Sumanth Nagulavancha Ignacio Vizzo Thomas Läbe
Jens Behley Michael Halstead Chris McCool Cyrill Stachniss

***Abstract*—Monitoring plants and fruits is important in modern agriculture, with applications ranging from high-throughput phenotyping to autonomous harvesting. Obtaining highly accurate 3D measurements under real agricultural conditions is a challenging task. In this paper, we address the problem of estimating the 3D shape of fruits when only a partial view is available. We propose a pipeline that exploits high-resolution 3D data in the learning phase but only requires a single RGB-D frame to predict the 3D shape of a *complete* fruit during operation. To achieve this, we first learn a latent space of potential fruit appearances that we can decode into an SDF volume. With the pretrained, frozen decoder, we subsequently learn an encoder that can produce meaningful latent vectors from a single RGB-D frame. The experiments presented in this paper suggest that our approach can predict the 3D shape of whole fruits online, needing only 4 ms for inference. We evaluate our approach in controlled environments and illustrate its deployment in greenhouses without modifications.**

***Index Terms*—Robotics and Automation in Agriculture and Forestry, Deep Learning for Visual Perception, RGB-D Perception**

I. Introduction

A challenge that agricultural production faces today is meeting the rising demand for food, feed, fiber, and fuel for an ever-growing world population. This situation is aggravated by several factors including: climate change, lack of workers, and decreasing biodiversity [14]. A promising solution to tackle this challenge is by means of autonomous robotic systems. The use of robotic systems can benefit the agricultural production sector across the whole plant growth period, from sowing to harvesting, with the goal of increasing yield while reducing human labor and agrochemical inputs [47].

In recent years, different studies showcased the range of applications in the context of agricultural robotics. They differ between arable fields and horticulture. In the first case, the use of robotic systems, both ground and aerial, prove its benefit in the context of weed management from mapping [29], [41] to intervention [33], but also for pest control [16] and phenotyping [48]. In horticulture, fruit detection [42] and counting [20] are first applications of robots. Such techniques can be used as a basis for fruit picking [46], phenotyping [40], harvesting [3], [2], and for ripeness estimation [20]. A common, yet not solved, problem for both arable field and horticulture is the estimation of the 3D shape of crops or fruits in real worlds conditions. Such conditions are particularly challenging due to the complexity of the environment. As an example, a leaf in the lower part of the canopy is often occluded by other leaves. Making it difficult to estimate the 3D shape of the complete plant. A similar scenario is often present in horticulture, where a fruit can be hidden behind leaves or other fruits. Nevertheless, the estimation of 3D models could have benefits for several applications: detailed yield estimation by providing crop information such as volume or autonomous harvesting providing precise fruit size, position, and orientation.

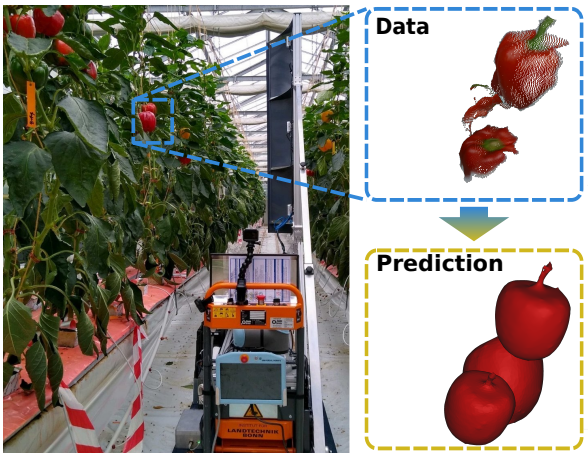

Fig. 1: Being able to reconstruct 3D shapes of fruit in greenhouses is important for applications ranging from yield estimation to harvesting. However, it is an extremely challenging task due to the complexity of the environment. On the left, we show our robot monitoring a sweet pepper greenhouse near Bonn, Germany. Given an RGB-D frame (top right) our approach is able to complete and reconstruct the 3D shape of fruits (bottom right).

This paper tackles the problem of estimating the 3D shape of fruits using commonly used RGB-D cameras while exploiting costly laser scanning systems only to learn a prior for the shape of fruits. Obtaining complete point clouds of fruits in real world scenarios is quite challenging and labor-intensive. A robot working in such conditions often only views a small

Manuscript received: Feb 23, 2022; Revised: Jun 7, 2022; Accepted: Jul 17, 2022. This paper was recommended for publication by Editor Hyungpil Moon upon evaluation of the Associate Editor and Reviewers' comments.

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob and by the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support programme under funding no 28DK108B20 (RegisTer). All authors are with the University of Bonn, Germany. Cyrill Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Digital Object Identifier (DOI): see top of this page.

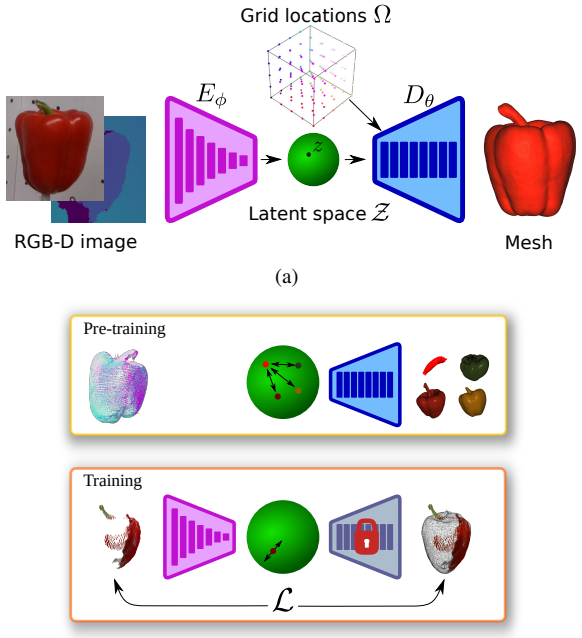

Fig. 2: Overview of our approach. (a) Generation of a triangular mesh using our architecture exploiting a pre-trained encoder that produces a latent vector (indicated as a point z in latent space $\mathcal{Z}$) and locations Ω on a regular grid to determine via a decoder D (blue) an SDF value that can be used for generating a mesh using marching cubes. (b) Training: we pre-train the shape decoder and optimize the latent shape space such that different sweet peppers are separated. Next, we freeze the shape decoder and train the encoder.

portion of a fruit. As can be seen in Fig. 1, the point clouds obtained by a robot are noisy and incomplete making the estimation of 3D shapes difficult. Our goal is to recover the complete 3D shape of fruits using only partial views from an RGB-D camera at inference time and exploiting a highly accurate 3D laser scanner to learn a prior over the target fruit species.

While the use of RGB-D sensors is increasing in the agricultural robotics context thanks to their flexibility and affordability, the use of the depth channel has been limited to supporting fruit detection [25], [34], [26]. We believe that this sensor can also be exploited to recover complete 3D geometries of fruits. Thus allowing robots to perform complex tasks such as growth monitoring on a per-fruit basis in challenging environments and allowing more accurate intervention. However, given the complexity of the task, such a detailed growth analysis is still bound to controlled environment [31], [10], [11] while, in field conditions, robots allow to monitor growth at a field level [7], [9].

The main contribution of this paper is a novel method to infer in real-time the 3D shape of fruits using a single RGB-D frame. We exploit the prior knowledge about the appearance of fruits, by encoding such prior information in the weights of a neural network trained using high resolution point cloud data. At inference time, our approach only requires a single RGB-D frame to estimate the 3D shape of a complete fruit in around 4 ms, thus making our approach suitable for robotics applications. In sum, our approach can (i) estimate in real-time the 3D shape of fruits using single RGB-D frames while leveraging costly laser scanning systems to learn a prior about the target fruit, (ii) be deployed in real greenhouses while trained in controlled environment, and (iii) can be adapted to species for which prior knowledge is not available.

II. Related Work

In agricultural environments such as greenhouses, orchards, and arable fields, observing crops and fruits entirely is challenging. This challenge is mainly due to the complexity of the environment. For example, a sweet pepper can be occluded by different fruits or can be hidden behind groups of leaves. However, having a complete observation of the target crop or fruit is important for many applications, ranging from harvesting to phenotyping. In recent years, a diverse number of studies focused on sensor placement to obtain more informative views of a target fruit. Lehnert et al. [27] exploit a camera array to compute the next best view in order to maximize fruit coverage. In a follow-up work, Zaenker et al. [49] combine local and global viewpoint planning allowing larger fruit coverage. Gibbs et al. [15] propose an active sensing algorithm to obtain high quality 3D surface reconstruction of plants. Such studies assume that a sensor can move more or less freely around the target object. This may be the case when paired with a robotic manipulator but will not be possible in all settings. Instead, our approach, by inferring 3D shape from a single frame, does not require any specific robot configuration.

Both, Blok et al. [5] and Kierdorf et al. [21] estimate the occluded parts of fruit or crops in 2D images. In the first case, the authors propose to directly learn a semantic mask including non-visible parts. In the second case, the authors use a GAN-based approach to generate images without occlusions. In contrast to such works, our approach estimates a 3D shape instead of a 2D representation of fruits without occlusion. More importantly, while such works require paired data to have a ground truth image without occlusion and an input image to be used as input, our approach does not require an input-ground truth pair.

In the context of image-based phenotyping, Kirk et al. [23] estimates mass and volume of strawberries using RGB images only, whereas Halstead et al. [20] proposes an object detector algorithm as a basis to estimate quantity and ripeness of fruits in real greenhouses. Halstead et al. [18] enhances this technique by providing a crop agnostic monitoring approach, which includes area estimation of the crop. However, they require a different model for different crop types. Generally, these approaches target a single task, instead, by estimating a complete 3D model our approach can be a basis to evaluate different traits without tailored methods.

In our previous work [30], [32], we exploit prior knowledge about the appearance of a plant by deforming a template to align with partial observations using gradient descent. Our new approach is different in two ways. First, we learn prior knowledge about our target fruit directly from real data instead of relying on a pre-determined template. Second, instead of solving an optimization problem, our approach only requires a single forward pass of a neural network to obtain a complete 3D model making it faster and thus suitable for online

applications on real robotic platforms.

Similar to DeepSDF [38], we solve the task of completing 3D shapes from partial observations. However, we are predicting signed distance fields (SDF) with a single forward pass instead of solving the problem by searching over a latent space. Thus, making our approach suitable for online robotics applications. Stutz et al. [45] propose a variational autoencoder pretrained on ShapeNet [8] to infer complete 3D models on real data. Instead of being a variational autoencoder, our network take as input a RGB-D frame and output a SDF volume. Additionally we use a contrastive loss [17] to enforce different views of the same object to generate the same complete shape. During training, we additionally exploit camera poses to define a self-supervision signal by comparing a local SDF with the predicted one.

3D scene completion, outside the agricultural domain, has recently gained attention. Dai et al. [13] proposed a sparse generative network for completing indoor scenes using RGB-D. Similar to our approach, this network is able to complete the scene beyond the sensor measurements. Rodriguez et al. [39] exploits a latent space representation to transfer grasping skills. The main difference to our approach is that we are not densifying sensor data but complete and reconstruct a target object.

III. Our Approach to Fruit Completion and Reconstruction

In this paper, we study the problem of estimating complete 3D shapes of fruit from a single RGB-D frame. For learning shape priors, we exploit a high-resolution, but slow and costly laser scanning system to learn a prior about the 3D shape of fruits. During operation, our proposed architecture takes as input a single RGB-D frame cropped to a single fruit, which can be obtained with any object detection approach [1] and outputs a complete 3D model of the fruit. To obtain such results, we first pre-train a decoder-only fully connected neural network (FCN) that learns to predict SDF values from a complete point cloud. Secondly, we train an encoder that learns to map an RGB-D frame to a complete 3D model using the pre-trained decoder. At inference time, we only need a single forward pass of a single RGB-D frame to obtain a complete 3D model, see Fig. 2.

A. Shape Decoder Pre-Training

With the pre-training of the decoder, we want to learn a prior of the complete 3D shape of a typical target fruit. We represent this prior with the weights of a decoder-only FCN via a latent space representation [38]. At this stage, the training data for the decoder network D_θ is a point cloud, $\mathcal{P}$ of a complete fruit obtained with a high accuracy laser scanning system paired with a latent vector, $\boldsymbol{z}$, and the output is its SDF representation $\text{SDF}(\boldsymbol{x})$, where $\text{SDF}(\boldsymbol{p}) = 0$ for $\boldsymbol{p} \in \mathcal{P}$. Formally, we aim at learning a function that maps a point $\boldsymbol{x} \in \mathbb{R}^3$ to its SDF value $s \in \mathbb{R}$:

$$D_\theta(\boldsymbol{z}, \boldsymbol{x}) = \text{SDF}(\boldsymbol{x}). \tag{1}$$

To learn the weights of the decoder θ, we regress the value of the SDF from complete 3D point clouds of fruits as follows.

We define a set of point clouds $\mathcal{P}$ with elements of the form $\mathcal{P}_k = \{(\boldsymbol{p}_k^0, \boldsymbol{n}_k^0), \ldots, (\boldsymbol{p}_k^N, \boldsymbol{n}_k^N)\}$, where k identifies a fruit instance and each element is given by the 3D location $\boldsymbol{p} \in \mathbb{R}^3$ of the point and its normal vector $\boldsymbol{n} \in \mathbb{R}^3, ||\boldsymbol{n}|| = 1$, estimated using principal component analysis. For each point $\boldsymbol{p}$, we generate the target SDF value by translating $\boldsymbol{p}$ along its normal by a small random value s. Thus, our training set is given by $\mathcal{X} = \{(\boldsymbol{x}, s) \mid \boldsymbol{x} = \boldsymbol{p} + s \cdot \boldsymbol{n}, s \sim \mathcal{U}(-\sigma, \sigma)\}$, where σ can be interpreted as the truncation value in standard TSDF pipelines. In this way, s represents ground truth SDF value $s = \text{SDF}(\boldsymbol{x})$ at location $\boldsymbol{x}$. We define a loss function to regress the SDF value for each point $\boldsymbol{x}$:

$$\mathcal{L}_{\text{sdf}}(\boldsymbol{x}, \boldsymbol{z}, s) = \Big|\text{clamp}(D_\theta(\boldsymbol{z}, \boldsymbol{x}), \tau) - \text{clamp}(s, \tau)\Big|, \tag{2}$$

where $\text{clamp}(x, \tau) = \max(-\tau, \min(x, \tau))$ is a clamping function restricting the values to be between $-\tau$ and $+\tau$. Additionally, we use a L1 regularization term over the latent vectors to force them to be unit-norm vectors:

$$\mathcal{L}_{\text{reg}}(\boldsymbol{z}) = \Big|1 - ||\boldsymbol{z}||\Big|. \tag{3}$$

The input of the decoder is then a point cloud of a fruit with its respective latent vector, which is randomly initialized and optimized during training together with the weights of the network. Following Park et al. [38], the FCN architecture consists of 8 fully connected layers. With the latent vector that is passed to the first and fourth layer. For an in-depth discussion, we refer to the DeepSDF publication [38]. In this way, by inputting only point clouds of complete fruits, we will bias the decoder to generate SDFs of complete shapes.

At test time, we can get a mesh of the object by calling the network with a regular 3D grid of points and thus get a standard SDF volume. Running the marching cubes algorithm [28] will then give us the predicted mesh.

B. Learning a Latent Representation from RGB-D Frames

Using a decoder-only architecture for shape completion as done in DeepSDF entails learning a latent vector during inference to predict complete 3D models. This leads to a rather high inference time, which we would like to avoid. Furthermore, learning from high-resolution fruit models beforehand is totally acceptable for our application and allows us to incorporate background knowledge about the appearance of fruits into our model. Therefore, we propose an encoder-decoder architecture Fig. 2, where the encoder E_ϕ is responsible for generating latent vectors that result in plausible 3D shapes after decoding. Our pipeline takes as input a single RGB-D frame $\text{I} \in \mathbb{R}^{H \times W \times 4}$ and returns an SDF volume of the fruit in the frame. During this training phase, we only optimize the weights ϕ of the encoder E_ϕ and keep the weights θ of the decoder D_θ frozen. Reducing the problem to a mapping function between an image and a latent vector, $\mathbb{R}^{H \times W \times 4} \mapsto \mathbb{R}^M$. Thus, the encoder is responsible to generate latent vectors $\boldsymbol{z} \in \mathbb{R}^M$ given an input image $\text{I} \in \mathbb{R}^{H \times W \times 4}$, namely $\boldsymbol{z} = E_\phi(\text{I})$, where ϕ represents the weights of the encoder. Substituting the encoder in Eq. (1):

$$D_\theta(E_\phi(\text{I}), \boldsymbol{x}) = \text{SDF}(\boldsymbol{x}). \tag{4}$$

At this stage, for each fruit k we have a set of image-pose pairs $\mathcal{I}_k = \{(\mathrm{I}_k^1, \mathsf{T}_k^1), \ldots, (\mathrm{I}_k^i, \mathsf{T}_k^i), \ldots, (\mathrm{I}_k^N, \mathsf{T}_k^N)\}$, where I_k^i is an RGB-D frame and $\mathsf{T}_k^i \in \mathbb{R}^{4\times4}$ is its pose in homogeneous coordinates. For each frame, we build a local SDF, denoted as SDF_k^i, using the poses T_k^i, which provides a local view and supervision for learning a consistent latent vector $\boldsymbol{z}_k^i = E_\phi(\mathrm{I}_k^i)$ for each frame.

An immediate challenge for our proposed solution is to align the latent vectors generated from the encoder to meaningful latent vectors for the decoder without direct one-to-one correspondences. Thus during training, we force the latent vectors $\boldsymbol{z}_k^i$ generated from the encoder E_ϕ to be unit-norm vectors using Eq. (3).

Another challenge for our pipeline is to generate the same latent vector $\boldsymbol{z}_k$ for different views I_k^i of the same fruit k. To enforce this relation, we define a contrastive loss that aims at minimizing the difference between latent vectors generated from different views $\boldsymbol{z}_m^i$ of the same fruit m and to repel the latent vectors $\boldsymbol{z}_n^j$ generated from different fruit j. Thus, we define a loss function inspired by contrastive representation learning pipelines [17] composed by a hinged repelling term for latents of different fruit, $m \neq n$ and a non-hinged attraction term for latents of the same fruit:

$$\mathcal{L}_\mathrm{c} = \begin{cases} ||\boldsymbol{z}_m^i - \boldsymbol{z}_n^j|| & , \text{if } m = n \\ \left[\delta - ||\boldsymbol{z}_m^i - \boldsymbol{z}_n^j||\right]^+ & , \text{otherwise,} \end{cases} \tag{5}$$

where $\boldsymbol{z}_m^i$ and $\boldsymbol{z}_n^j$ are latent vectors generated from different RGB-D frames and $[x]^+ = \max(0, x)$, which hinges the loss. This is modulated by the parameter δ, which allows the latents to move around improving training stability [6].

Additionally, we optimize our network with an L1 loss between the predicted and the local SDF values, namely SDF_k^i, generated from a single RGB-D frame I_k^i. Here, we compare the predicted SDF values with the estimated SDF values on spatial locations Ω on a regular grid, here we use the bounding box of the fruit $\mathcal{B}$ with dimensions $w_\mathcal{B}$, $h_\mathcal{B}$, and $l_\mathcal{B}$ for width, height, and length, respectively. Then, the grid locations Ω are defined as follows:

$$\Omega = \left\{ r \begin{pmatrix} i \\ j \\ k \end{pmatrix} - \frac{1}{2} \begin{pmatrix} e \\ e \\ e \end{pmatrix} \middle| \begin{array}{l} i \in \{0, 1, \ldots, D-1\} \\ j \in \{0, 1, \ldots, D-1\} \\ k \in \{0, 1, \ldots, D-1\} \end{array} \right\}, \tag{6}$$

where $e = \max(w_\mathcal{B}, h_\mathcal{B}, l_\mathcal{B})$ is the extent of the grid and $r = eD^{-1}$ corresponds to the resolution with D grid positions.

In line with prior work on scan completion [13], [35], we log-transform the predicted and target values before applying the L1 loss. We mask out regions with high local SDF values to account for non-observed voxels. The SDF loss $\mathcal{L}_v$ is, then, defined as:

$$\mathcal{L}_\mathrm{v}(s, \hat{s}) = |\log(s) - \log(\hat{s})|, \tag{7}$$

where $s = SDF(\boldsymbol{x})_k^i$ is the observed SDF value from the local view and $\hat{s} = D_\theta(E_\phi(\mathrm{I}_i^k), \boldsymbol{x})$ is the predicted SDF value at grid positions $\boldsymbol{x} \in \Omega$. The loss function given in Eq. (7) guarantees that the predicted volume closely represents the input fruit. Without this loss, there is no one-to-one correspondence between input and output. During training we minimize the weighted sum of the defined losses:

$$\mathcal{L} = w_\mathrm{c}\mathcal{L}_\mathrm{c} + w_\mathrm{reg}\mathcal{L}_\mathrm{reg} + w_\mathrm{v}\mathcal{L}_\mathrm{v}, \tag{8}$$

where w_c, w_reg, w_v are scalars balancing the different terms.

Our encoder architecture consists of 7 blocks consisting of one convolutional layer and one pooling layer with leaky ReLU activations and one final fully connected layer to match the desired latent dimension. At each block, we halved the size of the first 2 dimensions of the feature map while we double the size of the last dimension.

Note that we need camera poses and multiple frames of the same fruit while training, but our pipeline only needs a single RGB-D frame without a pose at inference time.

C. Adaptation to Different Species

As a next step, we want to avoid collecting a large amount of training data for each species. Instead, we want to leverage high resolution point cloud of one fruit (e.g. sweet peppers) and adapt the decoder weights to predict the shapes of another species (e.g. strawberries) from RGB-D images. To solve this question in our pipeline we only need to update the weights of the decoder during the second stage of the training, using the same loss function as in Eq. (8).

D. Instance-Based Semantic Segmentation

To deploy our system in real conditions, a necessary first step is to detect each fruit in a given image and identify the pixels belonging to a fruit instance. To solve such a task, we exploit the super- and sub-class network proposed for objected detection by Halstead et al. [20] and extended for instance segmentation [19]. This approach takes as input an RGB frame and generates, for each fruit in the image, a binary segmentation mask and estimates fruit ripeness. Note that any instance segmentation approach can be used.

IV. Data Collection

We collect a fruit dataset to learn a shape prior but also to evaluate our proposed solution. Such a dataset of fruits consists of high accuracy point clouds and RGB-D frames. To obtain the point clouds, we use a sub-millimeter accurate Perceptron V5 laser scanner and a Romer Infinite measuring arm with a scanning accuracy of 0.012mm. With the same hardware setup as described by Schunck et al. [43] for more details on the used hardware. To record the RGB-D images, we use a RealSense D435. We show a few samples of our dataset in Fig. 3. In total, we scan 82 strawberries and 84 sweet peppers, resulting in 4000 and 5000 images.

To use the data from different sensors (laser scanner and RGB-D images in our case) in the same pipeline we must register the data to each other. We start by registering the RGB-D images to each other with a standard TSDF fusion pipeline [12], [37]. The results of this step are the camera poses for each RGB-D frame in a local coordinate system and a mesh of the object. To finally register the point cloud of the laser scanner to the points of the mesh from the RGB-D sensor, we apply an iterative closest point (ICP) [4] algorithm, see Fig. 3 for an example. This algorithm needs an approximate, initial

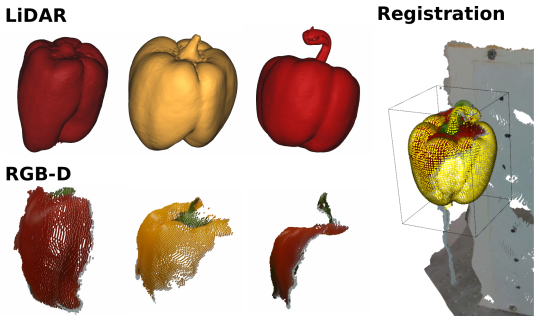

Fig. 3: Exemplary overview of our dataset, reference 3D models obtained with a high precision laser scanner (top left). RGB-D frames obtained with a depth camera (bottom left). Registration between ground a truth model and the aggregated RGB-D frames.

transformation between the two point clouds. To compute it, we scan the fruits inside three perpendicular planes and exploit this structure in the registration.

Given the result of the ICP with this approximation, we compute the pose of every RGB-D frame in the coordinate system of the laser model. Note that in this study such transformations are used during training to define the SDF loss in Eq. (7) but it is not needed at inference time.

Additionally, for each frame, we compute a semantic mask employing the approach outlined in Section III-D by fine-tuning a model trained from the BUP20 dataset [18]. The model was trained for twenty epochs at a learning rate of $1e^{-3}$ using the stochastic gradient descent optimizer with a momentum of 0.9 and weight decay of $5e^{-4}$.

To sum up, for each fruit in our dataset we have the ground truth 3D model from the high-resolution LiDAR and the measuring arm, a diverse number of RGB-D frames paired with binary masks and registration parameters, both the frame-to-frame and frame-to-model.

V. Experimental Evaluation

We validate our approach for shape completion using a mixture of data collected in a controlled environment and in a real glasshouse and compare our results to both learning and non-learning-based existing methods. From now on, we refer to our approach with CoRe, an abbreviation for Completion and Reconstruction. Specifically, we show experiments whose results support our three claims: (i) estimate in real-time the 3D shape of fruits using single RGB-D frames while leveraging costly laser scanning systems to learn prior about the target fruit, (ii) be deployed in real greenhouse while trained in controlled environment, and (iii) can be adapted to species for which a prior knowledge is not available.

A. Metrics

To measure the accuracy of our approach we use different metrics: f-score, precision, recall, and Chamfer distance. Defined over point clouds, the Chamfer distance D_{C} is the average symmetric squared distance $\bar{d}^2$ of each point to its nearest neighbor in the other point cloud:

$$D_{\mathrm{C}}(\mathcal{G},\mathcal{R}) \quad = \frac{\bar{d}^2(\mathcal{G},\mathcal{R})}{2} + \frac{\bar{d}^2(\mathcal{R},\mathcal{G})}{2}, \tag{9}$$

with $\bar{d}^2(\mathcal{P}_i,\mathcal{P}_j) = \frac{1}{|\mathcal{P}_i|}\sum_{\boldsymbol{x}_i\in\mathcal{P}_i} \min_{\boldsymbol{x}_j\in\mathcal{P}_j} \|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2$, where $\mathcal{G}$ and $\mathcal{R}$ are respectively the ground truth point cloud and the point cloud obtained by sampling the reconstructed mesh. We use the f-score metric as given by Knapitsch et al. [24]. To compute the f-score, we first define precision p, and recall r, given a threshold ρ:

$$\begin{aligned} p(\rho) &= \frac{100}{|\mathcal{R}|}\sum_{\boldsymbol{r}\in\mathcal{R}} \left[\!\!\left[\min_{\boldsymbol{g}\in\mathcal{G}} ||\boldsymbol{r}-\boldsymbol{g}|| < \rho \right]\!\!\right], \\ r(\rho) &= \frac{100}{|\mathcal{G}|}\sum_{\boldsymbol{g}\in\mathcal{G}} \left[\!\!\left[\min_{\boldsymbol{r}\in\mathcal{R}} ||\boldsymbol{g}-\boldsymbol{r}|| < \rho \right]\!\!\right], \end{aligned} \tag{10}$$

where $\mathcal{G}$ and $\mathcal{R}$ are defined as in Eq. (9), $\boldsymbol{g}$ and $\boldsymbol{r}$ are points from $\mathcal{G}$ and $\mathcal{R}$ and the operator $[\![\cdot]\!]$ is the Iverson bracket, i.e., if the condition within the brackets is satisfied it evaluates to 1, otherwise to 0. Intuitively, such metrics compute the percentage of points in one set whose distance to the closest point in the other set is smaller than a fixed threshold. As usual, the f-score is the harmonic mean of precision and recall. In all the experiments, we set the threshold ρ to 5 mm. Additionally, we report the average inference time needed to obtain the complete 3D shape. In our experiments, we used an NVIDIA Quadro RTX 5000 GPU.

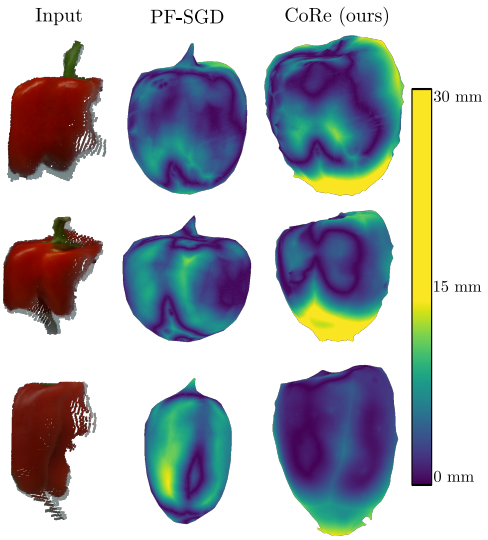

Fig. 4: While PF-SGD [32] obtains better results in terms of reconstruction accuracy, our approach produces competitive 3D models in a fraction of time. We show in yellow every point with an error greater than 15mm to highlight where our approach fails.

B. Fruit Reconstruction in Controlled Environments

We design the first experiment to show how our approach can estimate the 3D shape of a target fruit only using a single RGB-D frame. We first train the DeepSDF decoder network using the hyperparameters suggested by Park et al. [38]. For completeness, we report the results of this network when the input is represented by highly accurate point clouds of complete fruits. We refer to this as upper bound in Tab. II.

We train our encoder using a batch size of 16 RGB-D frames (each frame is zero-padded to reach a dimension of $256\times256\times4$) for 250 epochs using the Adam [22] optimizer starting from a learning rate of $1e^{-5}$ with an exponential decay modulated by $\gamma = 0.97$ and using the following values to weight the different terms of the loss function Eq. (8):

TABLE I: Strawberry - reconstruction results. Bold numbers indicate best performance in learning- and non-learning-based approaches.

Approach	$D_{\mathbf{C}}$ [mm] ↓ avg (std)	**f-score** [%] ↑ avg (std)	**precision** [%] ↑ avg (std)	**recall** [%] ↑ avg (std)	**inference time** [s] ↓ avg	**partial?**	**learning?**
CPD [36]	5.13 (0.91)	57.93 (9.29)	**94.09(7.94)**	42.34 (8.59)	**0.57**	✓	✗
PF-SGD [32]	**2.71(0.58)**	**86.08(8.73)**	88.82 (7.01)	**83.90(11.26)**	8.1	✓	✗
DeepSDF [38]	3.61 (0.13)	74.01 (18.21)	83.76 (11.64)	68.32 (21.10)	36.84	✓	✓
CoRe (ours)	**2.67(0.93)**	**86.01(13.53)**	**87.97(11.34)**	**84.85(16.32)**	**0.004**	✓	✓
upper bound	1.21 (0.26)	98.28 (1.94)	97.50 (3.25)	99.12 (1.42)	37.23	✗	✓

TABLE II: Sweet pepper - reconstruction results. Bold numbers indicate best performance in learning- and non-learning-based approaches.

Approach	$D_{\mathbf{C}}$ [mm] ↓ avg (std)	**f-score** [%] ↑ avg (std)	**precision** [%] ↑ avg (std)	**recall** [%] ↑ avg (std)	**inference time** [s] ↓ avg	**partial?**	**learning?**
CPD [36]	12.36 (1.29)	39.84 (9.34)	**76.68(16.08)**	27.07 (6.67)	**15.62**	✓	✗
PF-SGD [32]	**3.97(0.97)**	**68.95(11.93)**	71.20 (11.10)	**66.94(12.73)**	17.48	✓	✗
DeepSDF [38]	29.78 (28.66)	37.12 (17.63)	32.96 (19.44)	46.06 (14.65)	44.13	✓	✓
CoRe (ours)	**7.83(1.76)**	**52.85(9.68)**	**47.38(9.61)**	**60.00(9.74)**	**0.004**	✓	✓
upper bound	2.84 (4.10)	94.94 (4.97)	95.08 (6.39)	94.34 (3.96)	44.46	✗	✓

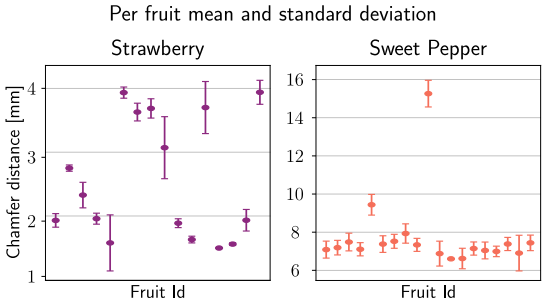

Fig. 5: We evaluate the consistency of our prediction as the poses change by computing mean and standard deviation for the chamfer distance by grouping frames representing the same fruit. We use around 500 frames for the strawberry and around 900 for the sweet pepper. We notice few outlier cases for both datasets, otherwise the difference of our prediction are in the millimeter order.

$w_c = 0.1$, $w_{reg} = 1$ and $w_v = 50$. In our experiments we define train, test and validation set based on the fruit ids. The sets stay consistent for the different training stages. For both datasets we use 70 % of the fruits as training, 20 % as testing and 10 % as validation. We compare our solution with both learning-based and non-learning methods and summarize the quantitative analysis in Tab. I and Tab. II. From the results, it is clear that our approach is more suitable for online robotics applications given the low inference time. We need only 4 ms to make an inference while DeepSDF needs more than 44 s on average and the best non-learning-based methods need more than 15 s. In terms of reconstruction accuracy, we can see that our approach can infer more accurate shapes than DeepSDF considering both D_{C} and f-score. This is due to the fact, that to learn complete shapes, DeepSDF has to be trained on the data collected with the laser scanning system described in Sec. IV and has no adaptation capability to infer on data collected with a RGB-D sensor. In this sense, our training strategy is beneficial to exploit both sensors. Considering the non-learning-based approaches, on one side the approach proposed by Marks et al. [32], PF-SGD, can estimate the 3D shape of the target fruit more closely but needing a much higher inference time. On the other side, the coherent point drift algorithms, CPD, [36] tends to produced collapsed meshes to closely align with the input point cloud, this can be seen by looking at the huge gap between precision and recall. In figure Fig. 4, we show a few qualitative examples of estimated 3D models.

C. Fruit Reconstruction in Greenhouses

In the second experiment, we illustrate that our approach can be directly deployed in real world conditions. To this end, we use the sweet pepper dataset BUP20 [44] collected in a greenhouse near Bonn, Germany. From the raw RGB-D images, we first predict the instance segmentation masks using the network proposed by Halstead et al. [18]. Afterward, for each segmented fruit we predict a 3D shape using our architecture. We show an overview of a full pipeline in real world conditions in Fig. 6. We do not have 3D reference models for this dataset, thus we can only show that our pipeline can be deployed in such conditions, providing visually plausible results.

D. Ablation Study

We, additionally, performed a diverse number of ablation studies to highlight the effects of our loss function design. As a first ablation, we train our encoder model using the setting described in Sec. V-B using all combinations of loss terms, see Tab. III. From these experiments, it is clear that the local SDF loss $\mathcal{L}_{\mathrm{v}}$, given in Eq. (7), is the term that is mostly influential for the results, it is only with the addition of this term that we reach performances around 8 mm for D_{C} and around 50 % for the f-score. This has to be expected since this term is the only one, which establishes a direct correspondence between input and prediction, while the other terms only consider the latent vectors. This explains the worst results when imposing loss terms only on the latent vectors.

Additionally, we train our encoder-decoder architecture without the pretrain, frozen decoder using the same loss function described in Eq. (8). As expected, without exploiting the complete point clouds in the decoder pre-train, our architecture does not manage to precisely estimate the complete 3D shape from the RGB-D frames. In the last ablation study, we compute

Fig. 6: Qualitative results in greenhouse. After an instance segmentation network (left), our pipeline outputs a 3D model of the segmented fruits (right). The point clouds (with the predictions of our network) are shown from a slighly different viewpoint, with respect to the images, to better visualize the 3D models. Note that our network is only trained on data collected in a controlled environment.

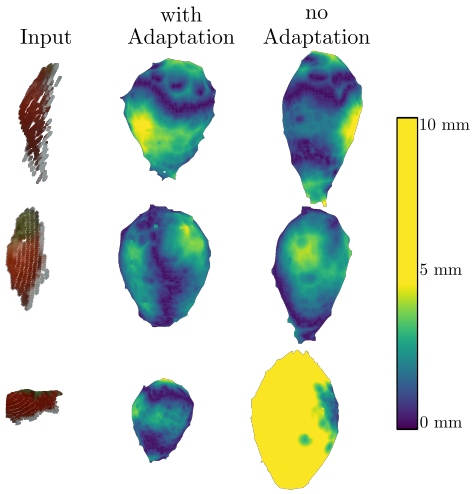

Fig. 7: Qualitative examples showing the benefits of adapting the network to a different species. We show in yellow points with an error greater than 5mm to highlight predicted parts with high errors.

per-fruit mean and standard deviation to better evaluate the consistency of the reconstructions as the poses change. We show such results in Fig. 5 where we see small per-fruit standard deviation with the exeption of few outliers.

E. Transferring to Different Species

In this last experiment, we show that our approach can be adapted to different species without the need to re-train the decoder D_θ. Here, we use the same hyper-parameters defined in Sec. V-B, the only change being that we also update the weight of the decoder instead of keeping it frozen. We train our network with and without adapting the decoder which was pre-trained on another species. We report the quantitative evaluation in Tab. IV for both datasets, the adaptation of the decoder improves both the chamfer distance (from 8.26 mm to 4.75 mm for the Strawberry and from 5.94 mm to 5.79 mm for the Sweet Pepper) and the f-score (from 32.93 % to 52.48 % for the Strawberry and from 53.82 % to 56.18 % for the Sweet Pepper) meaning that we are able to keep a strong prior about general fruit shape while adapting it to a new species. Interestingly, adapting the Strawberry decoder leads to better reconstruction than learning the decoder directly on sweet peppers. We believe this is the result of a larger diversity in the strawberry dataset. We additionally show qualitative results in Fig. 7, where it is evident that, without the adaptation, the network outputs unreliable shapes.

TABLE III: Sweet pepper - ablation

Loss	$D_{\mathbf{C}}$ [mm] ↓ avg (std)	**f-score** [%] ↑ avg (std)
$\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{c}}$	13.06 (3.16)	27.76 (10.07)
$\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{v}}$	8.12 (1.75)	51.18 (9.87)
$\mathcal{L}_{\text{c}} + \mathcal{L}_{\text{v}}$	7.86 (1.63)	51.39 (9.23)
$\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{c}} + \mathcal{L}_{\text{v}}$	**7.83(1.76)**	**52.85(9.68)**
no prior, $\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{c}} + \mathcal{L}_{\text{v}}$	16.41 (3.47)	24.60 (6.75)

TABLE IV: Adaptation to different species

Strawberry		Sweet Pepper		
$D_{\mathbf{C}}$ [mm] ↓ avg (std)	**f-score** [%] ↑ avg (std)	$D_{\mathbf{C}}$ [mm] ↓ avg (std)	**f-score** [%] ↑ avg (std)	**adaptation**
8.26 (1.74)	32.93 (12.20)	5.94 (1.03)	53.82 (6.74)	✗
4.75(1.34)	**52.48(18.86)**	**5.79(1.06)**	**56.18(8.01)**	✓

VI. Conclusion

In this paper, we presented a novel approach to estimate the 3D shape of fruits from single RGB-D images fast enough for online operation at sensor frame rate. Our method exploits a prior knowledge learned from high quality point clouds during training. This allows us to successfully estimate in real time a 3D mesh of target fruits. We implemented and evaluated our approach on different datasets obtained in controlled environment and real world conditions. We provided comparisons to other existing techniques both learning-based and non-learning-based and supported all claims made in this paper. The experiments suggest that we can achieve competitive results in terms of reconstruction accuracy while running orders of magnitude faster than the baselines, thus making our approach suitable for robotics applications.

References

[1] Y. Amit, P. Felzenszwalb, and R. Girshick. Object detection. *Computer Vision: A Reference Guide*, 2020.

[2] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Hellström, J. Hemming, P. Kurtser, O. Ringdahl, T. Tielen, et al. Development of a sweet pepper harvesting robot. *Journal of Field Robotics (JFR)*, 37(6):1027–1039, 2020.

[3] R. Barth, J. Hemming, and E.J. van Henten. Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, 146:71–84, 2016.

[4] P. Besl and N. McKay. A Method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[5] P.M. Blok, E.J. van Henten, F.K. van Evert, and G. Kootstra. Image-based size estimation of broccoli heads under varying degrees of occlusion. *Biosystems Engineering*, 208:213–233, 2021.

[6] B.D. Brabandere, D. Neven, and L.V. Gool. Semantic instance segmentation with a discriminative loss function. In *Deep Learning for Robotic Vision workshop, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] L. Carlone, J. Dong, S. Fenu, G. Rains, and F. Dellaert. Towards 4d crop analysis in precision agriculture: Estimating plant height and crown radius over time via expectation-maximization. In *ICRA Workshop on Robotics in Agriculture*, 2015.

[8] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint:1512.03012*, 2015.

[9] N. Chebrolu, T. Läbe, and C. Stachniss. Robust long-term registration of uav images of crop fields for precision agriculture. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3097–3104, 2018.

[10] N. Chebrolu, T. Läbe, and C. Stachniss. Spatio-temporal non-rigid registration of 3d point clouds of plants. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[11] N. Chebrolu, F. Magistri, T. Läbe, and C. Stachniss. Registration of Spatio-Temporal Point Clouds of Plants for Phenotyping. *PLOS ONE*, 16(2), 2021.

[12] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.

[13] A. Dai, C. Diller, and M. Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] T. Duckett, S. Pearson, S. Blackmore, B. Grieve, W. Chen, G. Cielniak, J. Cleaversmith, J. Dai, S. Davis, C. Fox, et al. Agricultural robotics: the future of robotic agriculture. *arXiv preprint:1806.06762*, 2018.

[15] J.A. Gibbs, M. Pound, A. French, D. Wells, E. Murchie, and T. Pridmore. Active vision and surface reconstruction for 3d plant shoot modelling. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 2019.

[16] F. Görlich, E. Marks, A.K. Mahlein, K. König, P. Lottes, and C. Stachniss. UAV-Based Classification of Cercospora Leaf Spot Using RGB Images. *Drones*, 5(2), 2021.

[17] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[18] M. Halstead, A. Ahmadi, C. Smitt, O. Schmittmann, and C. McCool. Crop agnostic monitoring driven by deep learning. *Frontiers in Plant Science*, 12, 2021.

[19] M. Halstead, S. Denman, C. Fookes, and C. McCool. Fruit detection in the wild: The impact of varying conditions and cultivar. In *Proc. of Digital Image Comp.: Techniques and Applications (DICTA)*, 2020.

[20] M. Halstead, C. McCool, S. Denman, T. Perez, and C. Fookes. Fruit quantity and ripeness estimation using a robotic vision system. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):2995–3002, 2018.

[21] J. Kierdorf, I. Weber, A. Kicherer, L. Zabawa, L. Drees, and R. Roscher. Behind the leaves–estimation of occluded grapevine berries with conditional generative adversarial networks. *arXiv preprint*, 2105.10325, 2021.

[22] D. Kingma and J.Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 1412.6980, 2014.

[23] R. Kirk, M. Mangan, and G. Cielniak. Non-destructive soft fruit and mass estimation for phenotyping in agriculture. In *International Conference on Computer Vision Systems (ICVS)*, 2021.

[24] A. Knapitsch, J. Park, Q.Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. on Graphics (TOG)*, 36(4), 2017.

[25] K. Kusumam, T. Krajník, S. Pearson, G. Cielniak, and T. Duckett. Can you pick a broccoli? 3d-vision based detection and localisation of broccoli heads in the field. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[26] J. Le Louëdec and G. Cielniak. 3d shape sensing and deep learning-based segmentation of strawberries. *Computers and Electronics in Agriculture*, 190:106374, 2021.

[27] C. Lehnert, D. Tsai, A. Eriksson, and C. McCool. 3d move to see: Multi-perspective visual servoing for improving object views with semantic segmentation. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[28] W. Lorensen and H. Cline. Marching Cubes: a High Resolution 3D Surface Construction Algorithm. In *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1987.

[29] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss. UAV-Based Crop and Weed Classification for Smart Farming. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[30] F. Magistri, N. Chebrolu, J. Behley, and C. Stachniss. Towards In-Field Phenotyping Exploiting Differentiable Rendering with Self-Consistency Loss. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[31] F. Magistri, N. Chebrolu, and C. Stachniss. Segmentation-Based 4D Registration of Plants Point Clouds for Phenotyping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[32] E. Marks, F. Magistri, and C. Stachniss. Precise 3d reconstruction of plants from uav imagery combining bundle adjustment and template matching. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[33] C. McCool, J. Beattie, J. Firn, C. Lehnert, J. Kulk, O. Bawden, R. Russell, and T. Perez. Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics. *IEEE Robotics and Automation Letters (RA-L)*, 3(2):1184–1190, 2018.

[34] H.A. Montes, J. Le Louedec, G. Cielniak, and T. Duckett. Real-time detection of broccoli crops in 3d point clouds for autonomous robotic harvesting. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[35] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[36] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 32(12):2262–2275, 2010.

[37] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[38] J.J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] D. Rodriguez, C. Cogswell, S. Koo, and S. Behnke. Transferring grasping skills to novel instances by latent space non-rigid registration. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[40] J.C. Rose, A. Kicherer, M. Wieland, L. Klingbeil, R. Töpfer, and H. Kuhlmann. Towards automated large-scale 3d phenotyping of vineyards under field conditions. *Sensors*, 16(12):2136, 2016.

[41] I. Sa, C. Lehnert, A. English, C. McCool, F. Dayoub, B. Upcroft, and T. Perez. Peduncle Detection of Sweet Pepper for Autonomous Crop Harvesting - Combined Colour and 3D Information. *IEEE Robotics and Automation Letters (RA-L)*, 2(2):765–772, 2017.

[42] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool. Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8):1222, 2016.

[43] D. Schunck, F. Magistri, R. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann, and L. Klingbeil. Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis . *PLOS ONE*, 16(8):1–18, 2021.

[44] C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool. Pathobot: A robot for glasshouse crop phenotyping and intervention. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[45] D. Stutz and A. Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] N. Wagner, R. Kirk, M. Hanheide, G. Cielniak, et al. Efficient and robust orientation estimation of strawberries for fruit picking applications. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[47] A. Walter, R. Finger, R. Huber, and N. Buchmann. Opinion: Smart farming is key to developing sustainable agriculture. *Proc. of the National Academy of Sciences*, 114(24):6148–6150, 2017.

[48] J. Weyler, , F. Magistri, P. Seitz, J. Behley, and C. Stachniss. In-Field Phenotyping Based on Crop Leaf and Plant Instance Segmentation. In *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2022.

[49] T. Zaenker, C. Lehnert, C. McCool, and M. Bennewitz. Combining local and global viewpoint planning for fruit coverage. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2021.