2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Efficient and Accurate Transformer-Based 3D Shape Completion and Reconstruction of Fruits for Agricultural Robots

Federico Magistri Rodrigo Marcuzzi Elias Marks Matteo Sodano Jens Behley Cyrill Stachniss

(a) Data Acquisition (b) RGB-D Mapping (c) Completed Shapes

Fig. 1: Robots used for collecting data in agricultural environments (a) will inherently suffer from occlusions due to the cluttered nature of such environments (b). Our approach completes 3D shapes of fruits if only a partial observation is available (c).

***Abstract*— Robots that operate in agricultural environments need a robust perception system that can deal with occlusions, which are naturally present in agricultural scenarios. In this paper, we address the problem of estimating 3D shapes of fruits when only partial observations are available. Generally speaking, such a shape completion can be realized by exploiting prior knowledge about the geometry of the fruit. This is typically done by template matching using traditional optimization algorithms, which are slow but accurate, or by encoding such knowledge into the weights of a neural network, leading to faster but often less accurate estimates. Our approach combines the best of both worlds. It exploits the benefit of having a template representing our object of interest with the advantages of using a neural network to learn how to deform a template. Our experimental evaluation demonstrates that our approach yields accurate estimation at a competitively low inference time in challenging greenhouse environments.**

I. Introduction

Nowadays, the agricultural production system has to cope with labor shortages and increased demand for food, feed, and fiber for an ever-growing population [12]. Robotic systems have the potential to tackle several issues by taking over tasks commonly executed by humans or by performing time-demanding or specialized tasks. Recently robotic solutions have been proposed for spot spraying to reduce chemical inputs [1], [35], for high-throughput phenotyping to support breeders in developing more resistant plant varieties [3], [38], and to increase yield or for autonomous harvesting [2], [4]. However, the cluttered nature of the agricultural environment poses challenges to the aforementioned tasks. For example, a phenotypic trait can be misinterpreted because a plant is only partially visible, or an autonomous grasp may fail because the robot wrongly estimates a partially occluded fruit shape.

In this paper, we investigate the problem of estimating the complete 3D shape of fruits when only a partial view can be gathered by the robotic system. This is a typical scenario in different agricultural environments, from greenhouses and orchards to arable fields, see Fig. 1 for an example. By exploiting prior knowledge about fruit shape, it is possible to solve a non-rigid registration problem [7], [32] to align a simple 3D mesh, with the partial observation. A different direction consists of encoding the prior knowledge into the weights of a neural network [28], [33], [36] that takes as input partial observation and outputs complete shapes. In the first case, it is possible to obtain high-fidelity shape estimation at the cost of high execution time. In the second case, the estimated shapes are less accurate but the inference step is typically one order of magnitude faster. Such approaches [22], [28] typically rely on a discretization of the scene which is suboptimal to represent fine-grained details.

The main contribution of this paper is a novel approach for completing 3D shapes combining template matching with deep learning. First, we use a 3D sparse convolutional backbone to extract point-wise features. We then aggregate such features into vertex features and feed them to a transformer decoder that iteratively deforms our template. Such an architecture allows us to estimate the complete 3D shape of fruits when only a partial point cloud is available.

As we demonstrate in this paper, our approach yields better shape completion estimates on different fruit species. Additionally, our iterative deformation formulation is a key ingredient for achieving accurate shape completion per-

All authors are with the Center for Robotics, University of Bonn, Germany. Cyrill Stachniss is additionally withthe Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under STA 1051/5-1 within the FOR 5351 (AID4Crops), 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).

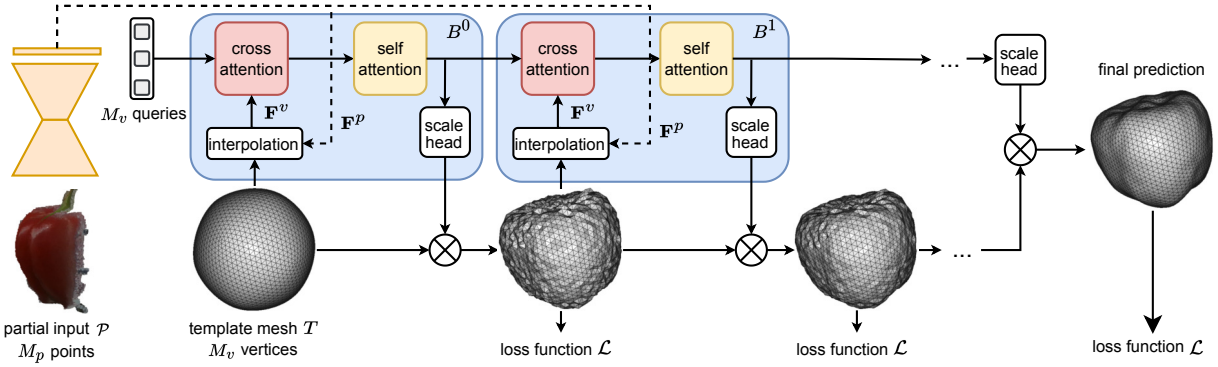

Fig. 2: Overview of our approach. We first extract point-wise features with the backbone and combine them with learnable queries using cross-attention. We interpolate the point features at the coordinates of the template vertices and obtain the vertex features, which we use as keys and values. The self-attention allows the queries to attend to each other. We predict, for each vertex, a scaling factor to deform the template. In the subsequent decoder layers, we interpolate using the deformed template from the previous layer. We supervise with the same loss the output and intermediate meshes.

formance, and modeling the mesh vertices with learnable queries allows us to learn the average fruit shape.

II. Related Work

Dealing with occlusions is a central problem for sensing applications in agricultural robotics. It is intrinsic to a broad range of agricultural environments from arable fields to greenhouses and orchards [10], but also to other applications such as service robotics [29] or autonomous driving [33].

Recently, a variety of works tackled the problem of estimating the shape of non-visible parts of plants or fruits using either 2D images or 3D point clouds. Lobefaro et al. [19] consider data association across time during mapping. In the 2D case, Kirk et al. [15] propose a convolutional neural network (CNN) to count fruits by re-identifying them after disappearances caused by occlusions. Blok et al. [5] predict an instance mask for broccoli heads, including its non-visible part. Similarly, Kierdorf et al. [13] estimate grapevine yield by using a generative adversarial network that learns to generate images without occlusions caused by leaves. Our work is different as we learn to estimate complete 3D shapes of occluded fruits so that our 3D estimates could be used for downstream tasks such as harvesting or yield estimation.

To obtain a more complete representation of fruits in dense canopies, Lehnert et al. [18] exploit a camera array mounted on a robotic arm to select the next best view. To obtain views that better cover fruits, Zaenker et al. [37] propose to combine a local, gradient-based method with global viewpoint planning to enable local occlusion avoidance while still being able to cover large areas. Gibbs et al. [10] propose an active vision pipeline to improve 3D plant reconstruction. In contrast, we do not tackle active perception and only rely on an incoming stream of sensor data. We do not rely on a manipulator to move around the plant and our approach can be deployed on robotic systems with fixed cameras as well.

Estimating and completing 3D shapes from partial observation has gained interest in recent years. Lehnert et al. [17] improve robotic grasping performances by fitting a super-ellipsoid on point clouds of partial fruits. Menon et al. [25] explicitly use the shape prediction by super-ellipsoid fitting to guide the sensor to view unobserved parts of the fruits. Thanks to the closed-form solution of the super-ellipsoid, this method can quickly provide an estimate of the fruit shape. It, however, cannot estimate fine-grained details. Template matching algorithms are a different solution for estimating the 3D shape of known objects. In the agricultural context, the templates can be simplified versions of plants [21] or leaves [24] in the form of 3D meshes. Such a template can then be fitted to partial observations using gradient-based optimization. Depending on the initialization of the template, such approaches can yield precise reconstruction at the cost of higher inference time when compared to closed-form or learning-based solutions [24].

Using deep learning, one can overcome the drawbacks of both closed-form solutions and optimization-based approaches. In our previous work, we learn a general fruit representation training DeepSDF [28] on complete point clouds, afterwards, we estimate 3D shapes from partial fruits with a contrastive learning framework [22] together with the estimation of 6-DoF pose for each fruit [27]. Both works are limited in representing fine-grained details given the discretization induced by the SDF representation. To solve this issue, we propose in this paper to learn the deformation of a template without relying on any discretization of the 3D space, thus being able to capture fine-grained details.

III. Our Approach to Shape Completion and Reconstruction

Given a point cloud representing a partial observation of a fruit, our goal is to estimate a triangular mesh representing the complete shape of such fruit. Our architecture design combines the advantages of learning-based approaches and template matching techniques. Thereby, we draw inspiration from recent advances in vision transformers [6], [8], [23]. Our architecture consists of a sparse 3D convolutional feature extractor to compute features for each point in the input point cloud. We encode such point features in the vertices of a 3D mesh representing our template. The template goes through an iterative deformation process in an attention-based decoder that gives us the final 3D mesh as output.

This combines the high-quality results of the template with the efficiency of the network. Furthermore, the whole system is differentiable, thus it can be trained end-to-end. Fig. 2 illustrates an overview.

A. Feature Extractor

Given an input point cloud $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_{M^p}\}$, $\boldsymbol{p}_i \in \mathbb{R}^3$, the objective of our feature extractor is to compute per-point features $\mathcal{F}^p = \{\boldsymbol{f}_1^p, \dots, \boldsymbol{f}_{M^p}^p\}$, $\boldsymbol{f}_i^p \in \mathbb{R}^C$. We later combine such features in the decoder to obtain the per-vertex features $\mathcal{F}^v$, $f_j^v \in \mathbb{R}^C$ of the M^v vertices that will drive the deformation process. We choose an encoder-decoder architecture based on MinkowskiNet [9] as feature extractor due to the sparsity of the input data. It is a ResNet-like [11] architecture using 3D sparse convolutions, which allows to keep the memory footprint low while preserving the spatial information. Note that also other feature extractors providing point-wise features could be used in this step.

First, we voxelize $\mathcal{P}$ using a voxel grid with voxel size v_s and obtain M^o voxels with voxel centers $\mathcal{O} = \{\boldsymbol{o}_1, \dots, \boldsymbol{o}_{M^o}\}$ with $\boldsymbol{o}_i \in \mathbb{R}^3$. Using the aforementioned encoder-decoder, we obtain voxel features $\boldsymbol{F}^o$, $\boldsymbol{f}_k^o \in \mathbb{R}^C$. Then, we convert voxel features $\boldsymbol{F}^o$ into point features $\mathcal{F}^p$ using a k-nearest neighbors interpolation as follows:

$$\boldsymbol{f}_i^p = \sum_{\boldsymbol{o}_j \in \mathcal{N}^k(\boldsymbol{p}_i)} \frac{1}{\|\boldsymbol{p}_i - \boldsymbol{o}_j\|_2^2} \boldsymbol{f}_j^v, \tag{1}$$

where we denote with $\mathcal{N}^k(\boldsymbol{p}_i)$ the set of k-nearest neighboring centers $\{\boldsymbol{o}_1, \dots, \boldsymbol{o}_k\} \subset \mathcal{O}$ in respect to $\boldsymbol{p}_i$ and with $\boldsymbol{f}_j^v \in \mathcal{F}^v$ the corresponding voxel feature of the j-th voxel.

B. Template Deformation

Let T be a triangular mesh of a sphere of radius ρ with vertices $\mathcal{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_{M^v}\}$, $\boldsymbol{v}_i \in \mathbb{R}^3$, and center $\boldsymbol{c} \in \mathbb{R}^3$. We use T as starting template that we iteratively deform into meshes $T^1, \dots, T^S$ to estimate the shape of our fruits. More specifically, we denote with $\mathcal{V}^t$ the vertices of the deformed mesh T^t. At each stage t, we use the extracted point features $\mathcal{F}^p$ and the coordinates of the vertices $\mathcal{V}^{t-1}$ of T^{t-1} at the last stage $t-1$ to predict a scaling value s_i^t for each vertex $\boldsymbol{v}_i$ to iteratively deform the initial template $T^0 = T$.

The deformation is carried out by moving each vertex $\boldsymbol{v}_i$ along the normalized direction $\boldsymbol{d}_i$ using the predicted per-vertex scale s_i^t:

$$\boldsymbol{v}_i^t = s_i^t \boldsymbol{d}_i + \boldsymbol{v}_i, \tag{2}$$

with:

$$\boldsymbol{d}_i = \frac{\boldsymbol{v}_i - \boldsymbol{c}}{||\boldsymbol{v}_i - \boldsymbol{c}||_2}, \tag{3}$$

where, the sphere center $\boldsymbol{c}$ corresponds to the fruit center.

To learn the per-vertex scaling values, $s_1^t, \dots, s_{M^v}^t$, we represent each of the M_v vertices with learnable queries $\boldsymbol{q}_i \in \mathbb{R}^C$ and use a multi-layer perception (MLP) as scaling head to predict the scaling value s_i^t to move each vertex $\boldsymbol{v}_i$ and deform the template T^t. See Fig. 3 for a visualization of the deformation process.

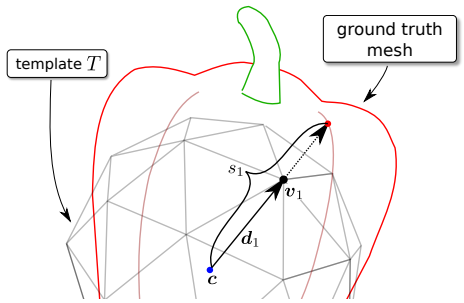

Fig. 3: Deformation of a triangle mesh. Given direction vectors $\boldsymbol{d}_i$ starting at the center of the sphere $\boldsymbol{c}$ (blue point), we iteratively transform the mesh by scaling along this direction, such that vertex $\boldsymbol{v}_i$ is on the surface of fruit (red point).

C. Transformer Decoder for Shape Completion

We use a multi-layer transformer decoder following previous work [23] to allow the queries $\boldsymbol{q}_i$ representing the vertices $\boldsymbol{v}_i$ to interact with point features $\mathcal{F}^p$ and to share information between them. In the following, we define as block B a sequence of cross-attention, self-attention, and an MLP. We concatenate multiple blocks to obtain our decoder.

To fuse information from point features $\mathcal{F}^p$ into the queries, we define the vertex features $\mathcal{F}^v$ that we obtain by interpolating the point features $\mathcal{F}^p$ at the coordinates of the vertices $\mathcal{V}^{t-1}$ of the mesh T^{t-1} using the same interpolation scheme as in Eq. (1). That means in each stage t, we interpolate point features $\mathcal{F}^p$ at the locations given by the vertices of the last stage. As we want to iteratively deform our initial template T^0 into a mesh T^S representing as closely as possible our fruit, each block at stage t takes as input the vertices $\mathcal{V}^{t-1}$ of a mesh T^{t-1} with its associated features $\mathcal{F}^p$ and outputs scaling values s_i^t for each vertex $\boldsymbol{v}_i$.

Each decoder block B^t consists of cross-attention [34] between queries and vertex features followed by self-attention plus MLP. We use a fixed positional encoding [34] to include spatial information of the vertices into the attention. The MLP predicts a scale value s_i^t for each vertex $\boldsymbol{v}_i$.

In the first block, we use the sphere template mesh $T^0 = T$ to perform interpolation and obtain the vertex features $\mathcal{F}^v$. After each decoder block at stage t, we move the vertices $\boldsymbol{v}_i$ using the predicted scale s_i^t and deform the mesh, see Eq. (2), resulting in T^t. In the next block at stage $t+1$, we use this deformed mesh vertices $\mathcal{V}^t$ as coordinates where to interpolate the point features $\mathcal{F}^p$ and obtain the new vertex features $\mathcal{F}^v$ used in the cross-attention of stage $t+1$.

We perform the deformation incrementally, which allows each decoder block at stage t to improve the performance of the previous one, i.e., stage $t-1$ by using the previously deformed template T^{t-1} to perform the feature interpolation. This means that each decoder block B^t produces an incrementally more accurate shape estimate T^t based on the last deformed shape T^{t-1}.

D. Loss Function

We compute a loss term $\mathcal{L}^t$ for each decoder block B^t to feed deeper decoder layers with a mesh that is closer and closer to the desired output. At each block B^t, after applying

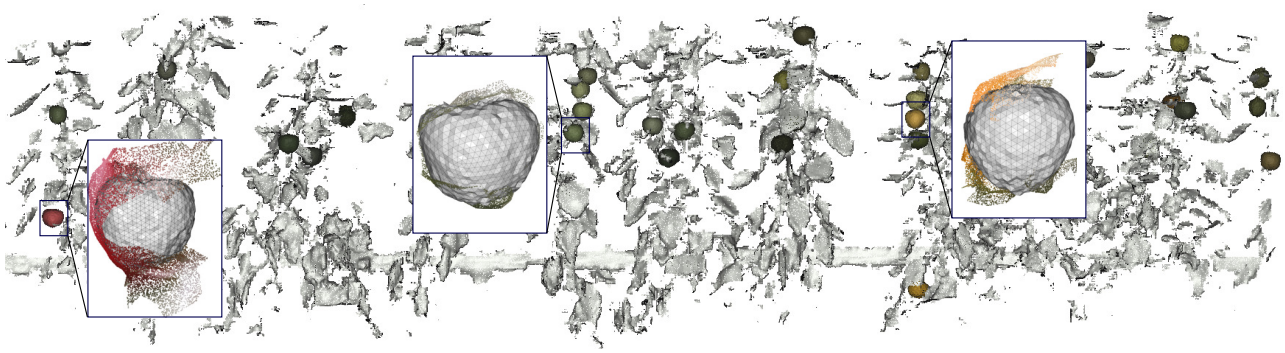

Fig. 4: Qualitative illustration of our shape completion approach on as sweet pepper greenhouse dataset. We additionally show zoomed-in views to better appreciate our completion approach in real-world conditions where mapping and segmentation errors are present.

the predicted scaling s_i^t to the current template vertices $\boldsymbol{v}_i^t$, we can compute a loss term between predicted mesh T^t and ground truth mesh $\hat{T}$. The objective of this loss term is to make sure that our predictions correctly represent the input point cloud. To this end, we compute the Chamfer distance between template vertices $\mathcal{V}^t$ and ground truth vertices $\hat{\mathcal{V}}$:

$$\mathcal{L}_{\text{cd}}^t(\mathcal{V}^t, \hat{\mathcal{V}}) = \frac{\bar{d}(\mathcal{V}^t, \hat{\mathcal{V}})}{2} + \frac{\bar{d}(\hat{\mathcal{V}}, \mathcal{V}^t)}{2}, \tag{4}$$

where $\bar{d}(\mathcal{A}, \mathcal{B})$ between vertex sets $\mathcal{A}$ and $\mathcal{B}$ is defined as:

$$\bar{d}(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{u} \in \mathcal{A}} \min_{\boldsymbol{v} \in \mathcal{B}} \|\boldsymbol{u} - \boldsymbol{v}\|_2^2. \tag{5}$$

By using only the Chamfer distance we are not enforcing any relation between neighboring vertices. This can result in noisy mesh predictions T^t due to the stochastic nature of the optimization process. To alleviate this issue, we use two different regularization terms: $\mathcal{L}_{\text{nc}}^t$ acting on the normals of the predicted mesh and $\mathcal{L}_{\text{s}}^t$ acting on the 3D position of the predicted mesh. The goal of these regularization terms is to obtain smooth meshes. We compute the normal consistency for each pair of neighboring faces of the template mesh T^t:

$$\mathcal{L}_{\text{nc}}^t(T^t) = \sum_{f_i \in \mathcal{F}} \sum_{f_j \in \mathcal{N}(f_i)} 1 - \boldsymbol{n}_i^\top \boldsymbol{n}_j, \tag{6}$$

where $\mathcal{F}$ is the set of faces in the template T^t, and $\mathcal{N}(f)$ defines the neighborhood of adjacent faces of a given face f. The normals $\boldsymbol{n}_i \in \mathbb{R}^3$ and $\boldsymbol{n}_j \in \mathbb{R}^3$ are associated to triangle faces f_i and f_j, where we assume that $\|\boldsymbol{n}_i\|_2 = \|\boldsymbol{n}_j\|_2 = 1$.

We compute the Laplacian smoothing objective $\mathcal{L}_s^t$ for the template mesh leading to:

$$\mathcal{L}_{\text{s}}^t(T^t) = \sum_{\boldsymbol{v} \in \mathcal{V}^t} \frac{1}{|\mathcal{N}(\boldsymbol{v})|} \sum_{\boldsymbol{u} \in \mathcal{N}(\boldsymbol{v})} \boldsymbol{u} - \boldsymbol{v}, \tag{7}$$

where $\mathcal{N}(\boldsymbol{v})$ defines the direct neighborhood of vertex $\boldsymbol{v}$, given the triangle mesh T^t, i.e., all vertices that are connected to $\boldsymbol{v}$ via an edge.

Our loss function $\mathcal{L}$ is, then, defined as the weighted sum of $\mathcal{L}_{\text{cd}}^t$, $\mathcal{L}_{\text{nc}}^t$, and $\mathcal{L}_{\text{s}}^t$ for all decoder blocks B^t, $t = 1, \ldots, S$:

$$\mathcal{L} = \sum_{t=1}^{S} w_{\text{cd}} \mathcal{L}_{\text{cd}}^t(\mathcal{V}^t, \hat{\mathcal{V}}) + w_{\text{nc}} \mathcal{L}_{\text{nc}}^k(T^t) + w_{\text{s}} \mathcal{L}_{\text{s}}^k(T^t). \tag{8}$$

E. Implementation Details

In our implementation, we use a sphere of 5 cm radius with 2500 vertices uniformly spread around the surface to define our initial template. Note that, as we have one query for each vertex, increasing the number of vertices will increase the network computational demands. Our backbone is composed of a first block consisting of a convolutional layer followed by batch normalization, followed by 4 downsampling blocks, and 4 upsampling blocks. In each block, we use one convolutional layer (de-convolution for the upsampling blocks) followed by a batch normalization layer and residual connections. Our decoder has 9 transformer blocks, with each block composed of a cross-attention layer a self-attention layer, and a 2-layer MLP to obtain the final predictions. After each block, we use LeakyRELU [20] activations with the exception of the last layer of each decoder block where we use a sigmoid activation scaled between 0 and 2 to allow for increasing the size of the predicted meshes. We train for 500 epochs using ADAM [14] with an initial learning rate of 10^{-4} with a step decay of 0.95 % each 25 epochs. Our implementation can be found at `https://github.com/PRBonn/TCoRe`.

IV. Experimental Evaluation

The main focus of this work is a transformer-based architecture able to estimate the complete 3D shape of fruits in the presence of occlusions by leveraging a general template. Our experiments show the capabilities of our method. The results also support our key claims, which are: our approach (i) yields better shape completion estimates on different fruit species; (ii) our iterative deformation formulation is a key ingredient for achieving accurate shape completion performance; (iii) modeling the mesh vertices with learnable queries allows us to learn the average fruit shape.

A. Experimental Setup

We use datasets of strawberry and sweet pepper fruits for performing evaluations. For each individual fruit in our datasets, we collected RGB-D frames with an Intel Realsense d435, where the fruits are only partially visible, and one complete point cloud using a high-precision LiDAR system. We refer to our previous works for more details [30]. Note that this dataset is collected in the lab. In line with our previous

TABLE I: Fruit reconstruction results in controlled environment. The ↓ and ↑ indicate that lower or higher values mean better performance.

Approach	Sweet Pepper					Strawberry				
	D_C [mm] ↓ avg	f-score [%] ↑ avg	precision [%] ↑ avg	recall [%] ↑ avg	time [s] ↓ avg	D_C [mm] ↓ avg	f-score [%] ↑ avg	precision [%] ↑ avg	recall [%] ↑ avg	time [s] ↓ avg
CPD [26]	12.36	39.84	76.68	27.07	15.62	5.13	57.93	94.09	42.34	0.57
PF-SGD [24]	3.97	68.95	71.20	66.94	17.48	2.71	86.08	88.82	83.90	8.10
DeepSDF [28]	29.78	37.12	32.96	46.06	44.13	3.61	74.01	83.76	68.32	36.84
CoRe [22]	7.83	52.85	47.38	60.00	**0.004**	2.67	86.01	87.97	84.85	**0.004**
HoMa [27]	3.16	80.86	82.14	79.72	0.60	2.42	92.81	94.38	94.53	0.53
T-CoRe (Ours)	**2.97**	**84.59**	**85.73**	**83.50**	0.33	**1.37**	**99.23**	**99.83**	**98.67**	0.33

TABLE II: Reconstruction results in the commercial greenhouse. The ↓ and ↑ indicate that lower or higher values mean better performance.

Approach	D_C [mm] ↓ avg	f-score [%] ↑ avg	precision [%] ↑ avg	recall [%] ↑ avg	inference time [s] ↓ avg	Learning?
CPD [26]	25.38	3.09	8.10	1.92	0.57	✗
PF-SGD [24]	9.28	35.03	37.32	33.21	30.21	✗
DeepSDF [28]	9.33	35.24	32.38	38.77	16.01	✓
CoRe [22]	6.90	41.47	43.17	41.64	**0.004**	✓
HoMa [27]	5.29	**58.56**	**61.28**	**56.26**	0.62	✓
T-CoRe (ours)	**5.17**	56.72	58.19	55.64	0.51	✓

works [22], [27], we split the datasets into train (70%), test (20%), and validation (10%) sets. We use the complete point clouds of each fruit in the train set to pre-train our network, while we use the complete point clouds of each fruit in the test set to compute the metrics for the evaluation of our approach. We additionally evaluate our approach on a sweet pepper dataset collected in a commercial greenhouse [27], [31]. In line with related works [22], [24], [27], we use the Chamfer distance D_C, i.e., the average symmetric squared distance of each point to its nearest neighbor in the other point cloud to evaluate our shape completion solution. We, additionally, use the f-score, precision, and recall at a fixed threshold as proposed by Knapitsch et al. [16] for quantitative evaluation. In all our experiment we fixed this threshold 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 A5000.

B. Fruit Completion

The first experiment evaluates the performance of our approach and its outcomes support the claim that our approach yields better shape completion estimates on different fruit species. Using the previously defined metrics, we compare our approach against a diverse set of baselines consisting of both learning [22], [27], [28] and non-learning-based [24], [26] solutions and report such metrics in Tab. I and Tab. II, where we refer to our approach as T-CoRe.

Regarding the reconstruction accuracy, our approach yields a better Chamfer distance on each dataset compared to the baselines. Notably, our approach is the only one with Chamfer distance below 1.5 mm on the strawberry dataset, where the second best approach [27] reaches 2.42 mm. Similarly, on the sweet pepper dataset in lab conditions, our approach is the only one below 3 mm with the second best [27] at 3.16 mm. Similarly, on these datasets our approach yields better f-score than baselines. In the sweet pepper dataset

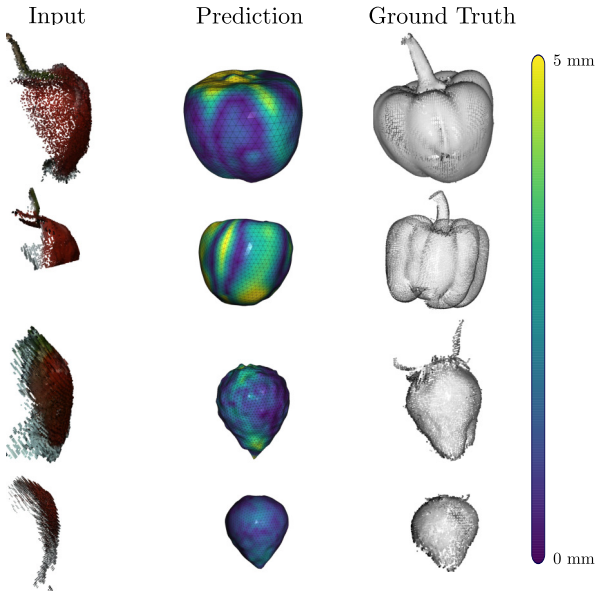

Fig. 5: Quantitative representation of our predictions where brighter colors indicate higher reconstruction errors. We notice that higher errors are more present on the top and the bottom of our predictions. We show the input point cloud (left) and the ground truth (right).

collected in the greenhouse, our approach reaches 5.17 mm against the 5.29 mm from HoMa [27], which surpasses our approach, T-CoRe, in terms of f-score: 58.56 % against 56.72 % mainly due to the higher precision score. Regarding the inference time, our novel approach ranks always in second place providing 3D estimates at roughly 2-3 Hz. However, the approach with the smallest inference time [22] yields 3D estimates which are substantially less accurate compared to our approach. As an example, the f-score for the strawberry dataset decreases from 99.23 % to 86.01 % while for the sweet pepper dataset drops from 84.59 % to 52.85 % for the lab conditions and from 56.72 % to 41.47 % for the greenhouse condition.

TABLE III: Iterative deformation vs. non-iterative deformation

Iterative Def.	Sweet Pepper D_C [mm] ↓ avg	Sweet Pepper f-score [%] ↑ avg	Strawberry D_C [mm] ↓ avg	Strawberry f-score [%] ↑ avg
✗	3.69	75.16	1.64	97.21
✓	**2.97**	**84.59**	**1.37**	**99.23**

TABLE IV: Modeling the mesh vertices with learnable queries allows us to learn the average fruit shape.

Approach	Sweet Pepper D_C [mm] ↓ avg	Sweet Pepper f-score [%] ↑ avg	Strawberry D_C [mm] ↓ avg	Strawberry f-score [%] ↑ avg
Template	9.24	21.03	5.88	39.00
Queries	3.24	81.56	1.60	98.06
Whole Model	**2.97**	**84.59**	**1.37**	**99.23**

We additionally provide qualitative results of our approach in Fig. 4 and a quantitative representation in Fig. 5 where brighter mesh colors represent higher reconstruction errors. One can notice that the top and bottom parts are more prone to errors. It is not surprising as those are the most problematic parts given, for example, the presence of the sweet pepper peduncle and the strawberry petiole.

C. Ablation Study

The second experiment highlights the benefit of iteratively deforming the initial template. It additionally supports our second claim, i.e., our iterative deformation formulation is a key ingredient for achieving accurate shape completion performance. We train an additional model where the predicted template deformations are applied only at the last block. See Tab. III for a quantitative comparison. The iterative deformation design yields an increase in f-score of 7% for the sweet pepper dataset and 2% for the strawberry. At the same time, we obtain a Chamfer distance below 3 mm for the sweet pepper and below 1.5 mm for the strawberry.

D. Learnable Queries

The last experiment evaluates the meaning of the learnable queries in our design. It furthermore supports our third claim, namely that modeling the mesh vertices with learnable queries allows us to learn the average fruit shape, we extract the mesh estimated by the queries alone. In previous works [6], [8], [23], the learnable queries act as bounding boxes or region proposals, they give a first approximation of where objects of different classes are located in the scene. In our case, each query represents one vertex in the template mesh and we decode from them the necessary scaling to deform it and match the input fruit. To show what the queries learn after training, we can predict the scaling from them and deform the sphere template mesh without any intervention of the input point cloud. In Tab. IV, we compute the reconstruction metrics described in Sec. IV-A using the spherical template without deformations, the mesh obtained by the queries alone and the deformed mesh after using our model. As can be seen, the spherical template is not sufficient to represent the fruits reaching less than 50 % of f-score on both datasets. The queries, instead, learn to approximate the average shape of the fruits, i.e., the average template that reduces the loss functions for all the training samples, obtaining 81 % for the sweet pepper and 98 % for the strawberry. We show the estimated average fruits in Fig. 6. When incorporating the point features, we allow the network to deform this average template to match the input point cloud and get a more accurate fruit representation 84 % for the sweet pepper and 99 % for the strawberry.

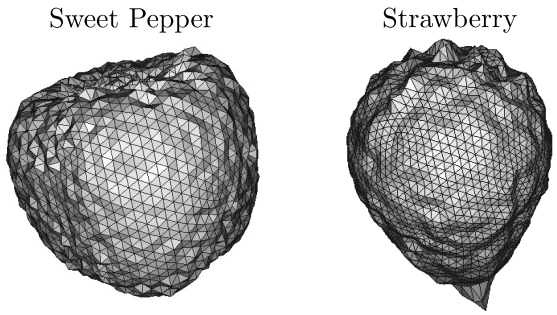

Fig. 6: Estimated average fruits obtained by applying the scaling factor from the queries to the template sphere.

V. Conclusion and Future Work

In this paper, we presented a novel approach to 3D shape completion and reconstruction of different fruit species on real-world datasets. Our approach operates on partial point clouds of fruits collected by a robotic system. Our method combines the advantages of template matching and deep learning to obtain accurate 3D estimates without sacrificing the inference time. We use a 3D sparse CNN to extract features from a partial point cloud, which we aggregate into vertex features, i.e., features representing the vertices of a template mesh. We exploit these vertex features to learn how to deform the original template to better align with the input point cloud. This allows us to successfully estimate the 3D geometry of non-visible fruit parts on different species while keeping a competitive inference time. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques and supported all claims made in this paper. The experiments suggest that our approach yields more accurate results in terms of reconstruction accuracy in presence of occlusions on different species.

Despite our encouraging results, there is further space for improvements. The learned average fruit of a species can be used as the initial template for a different one, e.g. from sweet peppers to tomatoes. In this way, one can train the network using only partial data of the target species without needing point clouds of complete fruits. Additionally, while our sphere template is suited for most fruit shapes, it cannot represent leaves. In theory, one can overcome such an issue by using a planar template and deforming its vertices along a fixed axis, rather than the vertices' normals. Lastly, by alternating our inference with a next-best view pipeline the estimated shapes can be further improved. Looking outside the agricultural robotics focus, an interesting direction would be using such a network in the context of autonomous driving where the estimated complete shapes of traffic participants could be used to estimate future behaviors.

REFERENCES

[1] A. Ahmadi, M. Halstead, and C. McCool. BonnBot-I: A Precise Weed Management and Crop Monitoring Platform. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[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] Y. Bao, L. Tang, M.W. Breitzman, M.G. Salas Fernandez, and P.S. Schnable. Field-based robotic phenotyping of sorghum plant architecture using stereo vision. *Journal of Field Robotics (JFR)*, 36(2):397–415, 2019.

[4] S. Birrell, J. Hughes, J.Y. Cai, and F. Iida. A field-tested robotic harvesting system for iceberg lettuce. *Journal of Field Robotics (JFR)*, 37(2):225–245, 2020.

[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] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020.

[7] N. Chebrolu, F. Magistri, T. Läbe, and C. Stachniss. Registration of Spatio-Temporal Point Clouds of Plants for Phenotyping. *PLOS ONE*, 16(2):e0247243, 2021.

[8] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] 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*, 17(6):1907–1917, 2019.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] L. Horrigan, R.S. Lawrence, and P. Walker. How sustainable agriculture can address the environmental and human health harms of industrial agriculture. *Environmental health perspectives*, 110:445–56, 2002.

[13] 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. *Frontiers in Artificial Intelligence*, 5:830026, 2022.

[14] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2015.

[15] R. Kirk, M. Mangan, and G. Cielniak. Robust counting of soft fruit through occlusions with re-identification. In *Proc. of the Intl. Conf. on Computer Vision Systems (ICVS)*, 2021.

[16] A. Knapitsch, J. Park, Q. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. on Graphics*, 36(4):1–13, 2017.

[17] C. Lehnert, I. Sa, C. McCool, B. Upcroft, and T. Perez. Sweet Pepper Pose Detection and Grasping for Automated Crop Harvesting. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

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

[19] L. Lobefaro, M. Malladi, O. Vysotska, T. Guadagnino, and C. Stachniss. Estimating 4D Data Associations Towards Spatial-Temporal Mapping of Growing Plants for Agricultural Robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[20] A. Maas, A. Hannun, and A. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2013.

[21] 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.

[22] F. Magistri, E. Marks, S. Nagulavancha, I. Vizzo, T. Läbe, J. Behley, M. Halstead, C. McCool, and C. Stachniss. Contrastive 3d shape completion and reconstruction for agricultural robots using rgb-d frames. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):10120–10127, 2022.

[23] R. Marcuzzi, L. Nunes, L. Wiesmann, J. Behley, and C. Stachniss. Mask-Based Panoptic LiDAR Segmentation for Autonomous Driving. *IEEE Robotics and Automation Letters (RA-L)*, 8(2):1141–1148, 2023.

[24] 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.

[25] R. Menon, T. Zaenker, N. Dengler, and M. Bennewitz. Nbv-sc: Next best view planning based on shape completion for fruit mapping and reconstruction. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

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

[27] Y. Pan, F. Magistri, T. Läbe, E. Marks, C. Smitt, C. McCool, J. Behley, and C. Stachniss. Panoptic mapping with fruit completion and pose estimation for horticultural robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[28] 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/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] 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.

[30] 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):e0256340, 2021.

[31] 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.

[32] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proc. of the Symp. on Geometry Processing*, 2007.

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

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

[35] Y. Xiong, Y. Ge, Y. Liang, and S. Blackmore. Development of a prototype robot and fast path-planning algorithm for static laser weeding. *Computers and Electronics in Agriculture*, 142:494–503, 2017.

[36] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[37] 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.

[38] D. Zermas, V. Morellas, D. Mulla, and N. Papanikolopoulos. 3d model processing for high throughput phenotype extraction–the case of corn. *Computers and Electronics in Agriculture*, 172:105047, 2020.