# CropCraft: Inverse Procedural Modeling for 3D Reconstruction of Crop Plants

Albert J. Zhai[1]     Xinlei Wang[1]     Kaiyuan Li[1]     Zhao Jiang[1]     Junxiong Zhou[2]

Sheng Wang[1]     Zhenong Jin[2]     Kaiyu Guan[1]     Shenlong Wang[1]

[1] University of Illinois Urbana-Champaign     [2] University of Minnesota Twin Cities
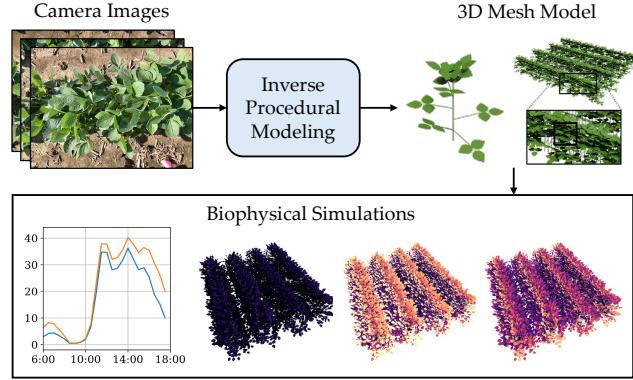
## Abstract

*The ability to automatically build 3D digital twins of plants from images has countless applications in agriculture, environmental science, robotics, and other fields. However, current 3D reconstruction methods fail to recover complete shapes of plants due to heavy occlusion and complex geometries. In this work, we present a novel method for 3D reconstruction of agricultural crops based on optimizing a parametric model of plant morphology via inverse procedural modeling. Our method first estimates depth maps by fitting a neural radiance field and then employs Bayesian optimization to estimate plant morphological parameters that result in consistent depth renderings. The resulting 3D model is complete and biologically plausible. We validate our method on a dataset of real images of agricultural fields, and demonstrate that the reconstructions can be used for a variety of monitoring and simulation applications.*

## 1. Introduction

Plants are ubiquitous objects that appear all around the world and serve as the foundation for agriculture, which underpins our civilization's growth and survival. The ability to automatically build 3D digital twins of plants from images has countless applications in agriculture, environmental science, robotics, and other fields. In particular, the development of such reconstruction methods in the context of agriculture will enable automatic, large-scale monitoring of crops. The collected data can provide decision support for farmers, aid carbon budgeting for decision-makers, support the development of new agricultural techniques, and inform the design of new genotypes [10, 40, 59]. All of these advances will contribute to increasing crop productivity, alleviating the rising food crisis of today's world [4, 30].

However, 3D reconstruction of plants remains to be a challenging vision problem. Many plants, including most of those found in agriculture, are composed of complex arrangements of thin leaves and branches that heavily occlude one another. These low visibility conditions cause existing reconstruction pipelines to fail. Multi-view recon-



Figure 1. **Inverse procedural modeling for agricultural crops.** We propose a novel method for 3D reconstruction of agricultural crops based on inverse procedural modeling. Unlike standard multi-view reconstruction pipelines, our method outputs a complete, interpretable, and biologically plausible 3D mesh model of the crop canopy, lending itself to simulations of important biophysical processes such as photosynthesis.

struction methods such as those based on neural rendering [24, 29, 46, 49, 54] or multi-view stereo [38, 50, 53, 57] do not reconstruct the invisible regions of the scene, leading to incomplete plant shapes and extraneous geometry. Learning-based methods [15, 26] can overcome this issue but require large amounts of ground-truth 3D data for training, which is extremely difficult to collect for dense vegetation due to the same occlusion reasons.

On the other hand, there is a large body of work that has found success in modeling 3D plant shapes via procedural generation [23, 27, 33, 34, 42, 43]. These procedural models are grounded in scientific knowledge and are carefully designed to produce biologically plausible plant shapes that consist of anatomically complete arrangements of plant organs with realistic shapes. However, these models generally require human input to set their parameters, and the task of automatically generating plants that accurately represent plant instances observed in the real world ("inverse procedural modeling") remains to be difficult.

In this work, we present a novel method for 3D reconstruction of agricultural crops based on optimizing the pa-

rameters of a procedural plant morphology model. Our method combines the flexibility of data-driven neural reconstruction methods with the robust foundational knowledge in procedural models. To this end, we first use neural radiance field (NeRF) techniques [29, 46] to estimate the geometry of the visible surfaces in the scene. We then apply RANSAC [36] to estimate plant row locations and determine a canonical camera pose from which to render depth maps. Next, we render depth maps from both the NeRF and a virtual scene generated by a procedural generation model, and use Bayesian optimization to minimize a loss function with respect to the procedural model's parameters.

The design of the loss function and the parameterization of the procedural model are crucial for obtaining a 3D reconstruction that is useful for field-level analysis applications. Importantly, we note that the fine-grained positions of each individual leaf are mostly irrelevant, as long as the *aggregate shape characteristics* of the crop canopy are faithful. Thus, we define our loss function on histogram statistics of the depth map and optimize a highly compact set of parameters to ensure that the key canopy characteristics for determining crop productivity are accurately captured without being distracted by fitting irrelevant details. And thanks to the strong priors imposed by the procedural model, this process jointly optimizes the *complete* plant shape, including portions not visible in the input images.

To validate our approach, we collect a multi-view dataset of real soybean and maize fields, paired with manual measurements of leaf area and leaf angle. We show that the proposed method can successfully reconstruct realistic crop canopies across different growth stages and estimate key canopy structure variables more accurately than baselines. We also show that the reconstructed 3D canopies can be directly fed into radiative transfer modeling software to provide accurate predictions of photosynthesis rates. The results highlight the potential for monitoring crop productivity directly from camera images instead of costly flux tower equipment. Our contributions are twofold:
- We present a novel approach for reconstructing complete 3D morphological models of large-scale crop plant fields from a collection of images.
- We introduce the first framework for image-based crop productivity quantification, making scalable carbon exchange monitoring a possibility.
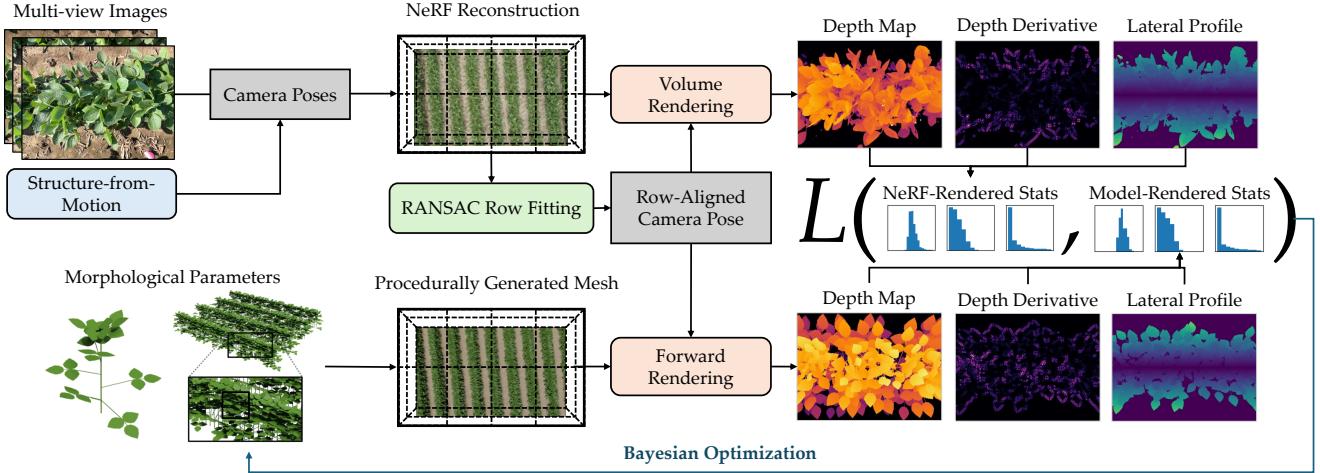
## 2. Related Work

**3D Reconstruction.** The task of reconstructing the 3D geometry of a scene given images is a longstanding problem in computer vision. Most reconstruction pipelines start from Structure-from-Motion (SfM), which infers camera parameters for the input images through joint optimization with detected keypoints [37, 47]. Afterwards, 3D geometry can be estimated based on photometric consistency across input views. Multi-view stereo (MVS) methods attempt to match correspondences across images and then triangulate the 3D coordinates of the points [38, 50, 53, 57]. Another approach for 3D reconstruction is based on fitting a neural radiance field (NeRF) to the scene [24, 25, 29, 46, 49, 54, 55]. NeRF is a high-fidelity and compact 3D scene representation that consists of a color field and a density field, both parameterized by neural networks and fitted to the input images via differentiable volume rendering [29]. After training, depth maps can be obtained by volume rendering with the density field. NeRF-based methods can model view-dependent effects and tend to give better surface geometry than MVS. Recently, an alternative scene representation based on 3D Gaussians has been proposed [13, 22], achieving higher visual fidelity and more efficient rendering than NeRF, but with less accurate geometry. In our approach, we use a standard off-the-shelf NeRF [46] to reconstruct visible surfaces of plants, which has been explored before [2, 16, 35, 41]. However, due to the inevitable occlusion in agricultural scenes, this is inadequate for full canopy reconstruction. To solve this issue, we turn to procedural generation models, which incorporate scientific knowledge and define a space of complete plant shapes that we can constrain our solution to.

**Inverse Procedural Modeling.** Inverse procedural modeling (IPM) refers to the problem of finding a procedure for generating a 3D representation of a given object or scene. The key design choices for IPM pipelines are the choice of procedural model and the method for optimizing the parameters of the model. Inverse procedural modeling has been applied for various input categories, with works focusing on building interiors [5, 11, 17, 18, 28], facades [8, 51], and driving scenes [7, 20]. The procedural models range from scene grammars [12, 18, 20] to constructive solid geometry trees [56] to L-systems [23, 27, 33, 43], which are especially popular for modeling botanical trees. To fit the model, a structural similarity measure is defined based on geometric and or semantic agreement. Depending on the type of model, which may require estimating procedural rules and or parameters [1], various search-based or dynamic programming-based optimization methods may be used to maximize the similarity measure. Unlike the multi-view reconstruction methods described above, inverse procedural modeling methods tend to focus less on photometrically exact reconstructions, aiming to obtain structurally similar approximations useful for simulation applications.

To the best of our knowledge, our work is the first to use inverse procedural modeling to reconstruct agricultural crops in the field. Some previous works have tackled individual plants [23, 43, 45], but modeling dense crop canopies poses a new set of challenges due to the high levels of occlusion and scale of the data. We propose an approach combin-

Figure 2. **Overview of our method.** We aim to estimate the parameters for a procedural generation model to generate a shape that matches the observed images. First, we use standard structure-from-motion and neural radiance field (NeRF) techniques to reconstruct the visible geometry of the scene. We then apply RANSAC to acquire a camera pose aligned with the planting rows of the crops. This pose is used to render depth maps from both the NeRF and the procedural model. We define a loss function based on histogram statistics of the depth maps and minimize it with respect to the morphological parameters using Bayesian optimization.

ing conventional multi-view reconstruction techniques with procedural generation to estimate 3D meshes that can be used for plant phenotyping, visualization, and simulation of biophysical processes.

## 3. Inverse Procedural Modeling for Crops

Our proposed method takes as input a set of images of a field of crops, and outputs a set of parameters that can be fed into a procedural generation model to produce a 3D mesh of the crops in the images. An overview of our method is provided in Fig. 2. By constraining the space of possible 3D shapes to the output space of the procedural model, we can ensure that the reconstructed plants are complete and have realistic leaf shapes and branch topology. Of course, we also need our reconstruction to match what we can observe in the input images. We achieve this by searching for parameters that minimize a loss function based on depth maps estimated from the input images and corresponding depth maps rendered from the reconstruction.

The layout for the rest of this section as follows. In Sec. 3.1, we describe the procedural generation models we use for plant morphology. In Sec. 3.2, we describe how we use a neural radiance field (NeRF) [29] and geometric heuristics to obtain row-aligned depth maps from the input images. In Sec. 3.3, we describe our loss function and in Sec. 3.4, we describe our optimization algorithm.

### 3.1. Procedural Morphology Model

At the core of our method is a procedural morphology model that defines a primitive for the leaf shape and the possible topologies for the plant structure. Since this varies
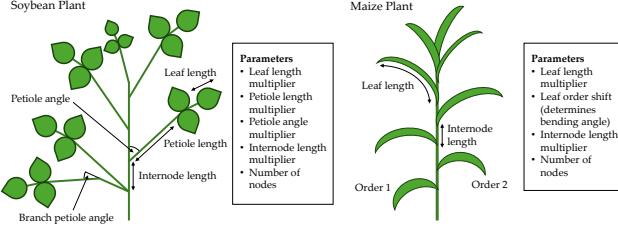
drastically across different species, these models should be species-dependent. In this work, we focus on soybean (*Glycine max*) and maize (*Zea mays*), the two most-grown crops in the United States [48].

Our procedural generation models are heavily based on existing work in the crop morphology literature. For soybean, we build upon the mCanopy model by Song et al. [42], and for maize, we build upon the coupled maize model by Qian et al. [34]. The mCanopy model consists of nodes of trifoliate leaves attached via a petiole to either the main plant stem or a branch stem. The maize model consists of large curved leaves growing from a single main stem. We introduce our own parameterization for each model, as illustrated in Fig. 3. The parameterization is designed to be flexible enough to capture the primary shape variations across instances of each species while remaining as low-dimensional as possible (5 parameters for soybean, 4 for maize). Note that the procedural generation process is in general stochastic; e.g. the model samples leaf angles according to a probability distribution. We refer readers to the supplementary for additional details about each model.

### 3.2. Depth Rendering

The procedural generation model defines a space of 3D shapes that can represent plants of various stages of maturity, growing conditions, and cultivars. In order to search within this space for the shape that best matches the plants in input observations, we define a loss function based on depth maps estimated from the input images and corresponding depth maps rendered using the proposed shape.

To estimate depth maps from the input images, we use an off-the-shelf NeRF (Nerfacto [46]) and render the expected

Figure 3. **Procedural plant morphology models.** We adopt procedural generation models from existing soybean morphology [42] and maize morphology [34] models. This (stylized) illustration shows the parameters that we allow to be optimized to model variations across individual instances of each species.

depth via volume rendering. In order to produce comparable depth maps for the procedurally generated scene, we need to know where to place the plants relative to the camera. Since it is extremely difficult to determine the precise base position of every individual plant under heavy occlusion, we assume that the plants are planted in rows with a known planting density and estimate the row locations using a heuristic approach that operates on a 3D point cloud extracted from the NeRF.

Specifically, we render the depth for each input view and unproject each pixel to a 3D point. Then, we randomly sample a fixed number of points within a bounding box for the scene and perform voxel downsampling. Next, we segment the points into ground and plant using a threshold on their color in LAB space. We fit a plane to the ground points using RANSAC [36] and take an upper slice of the plant points based on their distance to the ground plane. Finally, we fit a line to each row by sequentially running RANSAC and removing the inliers for each fitted line. Additional details for the row-fitting procedure are provided in the supplementary.

Given the ground plane and rows, we define a standardized camera to render from at a predetermined height above the center of the row with the most inliers, facing straight downwards with its $x$-axis aligned with the row. This camera is used to render both the observation depth map $\mathbf{I}_{\mathrm{obs}}$ from the NeRF and the depth map $\mathbf{I}_{\mathrm{pred}}$ induced by the procedurally generated scene. We also use a color threshold and a bounding box to acquire foreground (plant) masks $\mathbf{M}_{\mathrm{obs}}$ and $\mathbf{M}_{\mathrm{pred}}$ corresponding to each depth map.

### 3.3. Loss Function

One of the key motivations for estimating the 3D structure of agricultural crops is to enable photosynthesis simulations that provide accurate predictions of primary productivity and other ecological variables. In general, we are interested in these variables at a large scale, at the level of patches or fields, not at the level of individual plants. Thus, the reconstruction of individual leaf locations may not be necessary as long as certain statistics of the aggregated crop

canopy are accurate. With this in mind, we design a loss function around histogram statistics of the rendered depth maps, aiming to recover the key shape properties necessary for photosynthesis simulation while avoiding overfitting to irrelevant aspects of the scene geometry.

**Depth Profile Term.** The first term of the loss function is the L2 distance between histograms of foreground depth values in the observed and predicted depth maps. Formally,

$$L_{\mathrm{depth}} = ||\mathbf{h}_{\mathrm{obs}} - \mathbf{h}_{\mathrm{pred}}||_2^2, \tag{1}$$

where $\mathbf{h}_{\mathrm{obs}}$ and $\mathbf{h}_{\mathrm{pred}}$ are histograms of the foreground depth values in $\mathbf{I}_{\mathrm{obs}}$ and $\mathbf{I}_{\mathrm{pred}}$ respectively. Unlike the depth maps themselves, the histograms are invariant to translation along the ground plane, while still containing information about the crop height.

**Lateral Profile Term.** The lateral profile term $L_{\mathrm{lateral}}$ is essentially the same as the depth profile term but for $y$-coordinate absolute values instead of $z$-coordinates. This term helps to constrain the extent to which the canopy spreads out on the ground, which the depth profile is mostly invariant to. We do not include another term for the $x$-coordinate because the $x$-axis is aligned with the planting rows, making the distribution close to uniform.

**Depth Derivative Term.** To capture more information about surface normals, we include a loss $L_{\mathrm{sobel}}$ on (histograms of) magnitudes of Sobel derivatives of the depth maps. We sum absolute values of the Sobel derivatives in both directions with a kernel size of 3.

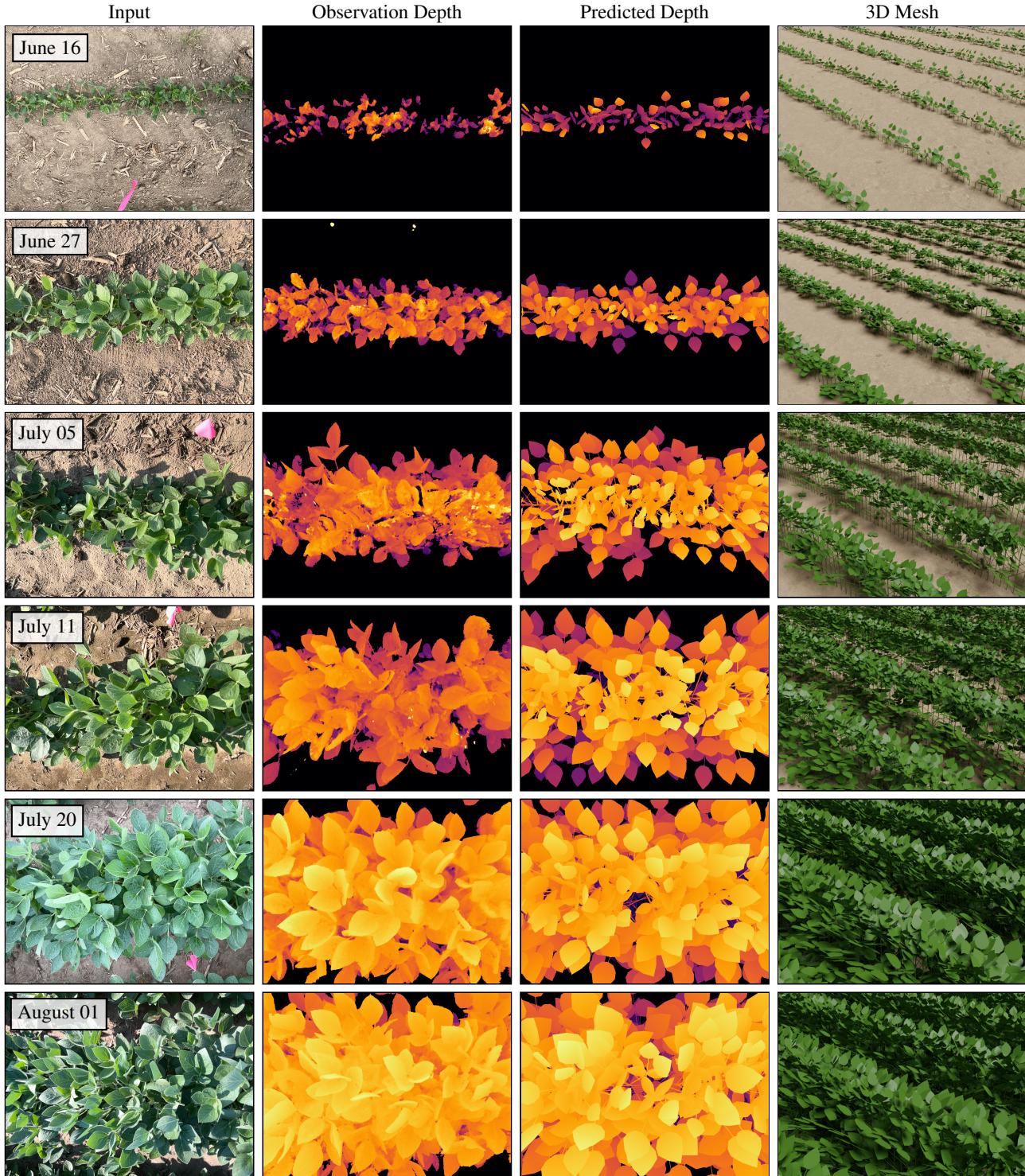**Mask Area Term.** The final term is a squared-error on the foreground mask area:

$$L_{\mathrm{mask}} = (||\mathbf{M}_{\mathrm{obs}}||_1 - ||\mathbf{M}_{\mathrm{pred}}||_1)^2. \tag{2}$$

Our final loss is a weighted combination of the above terms:

$$L = L_{\mathrm{depth}} + \lambda_{\mathrm{lateral}}L_{\mathrm{lateral}} + \lambda_{\mathrm{sobel}}L_{\mathrm{sobel}} + \lambda_{\mathrm{mask}}L_{\mathrm{mask}}. \tag{3}$$

### 3.4. Bayesian Optimization

Since the procedural generation model directly adds new mesh faces to create plants with different topology, the transformation from parameters to generated shape is not differentiable. Thus, it is difficult to minimize the loss function with respect to the parameters using gradient-based optimization methods. Instead, we employ Bayesian optimization, a black-box optimization method that is commonly used for hyperparameter tuning [39]. Bayesian optimization creates a surrogate for the objective function using Gaussian process regression, and then decides where to sample next by optimizing an acquisition function that combines uncertainty and the expected objective value [9]. We

| Input | Observation Depth | Predicted Depth | 3D Mesh |
|---|---|---|---|



Figure 4. **Qualitative reconstruction results (soybean)**. We validate the reconstruction quality from our method on images from real agricultural fields. From left to right: example images from the multi-view data, row-aligned NeRF-rendered depth, procedural model depth, rendered visualization of the procedural mesh. By matching statistics of the predicted depth with the observed depth, our method is able to estimate the key shape parameters needed to characterize the growth of the plants throughout the growing season, consistently producing realistic reconstructions.
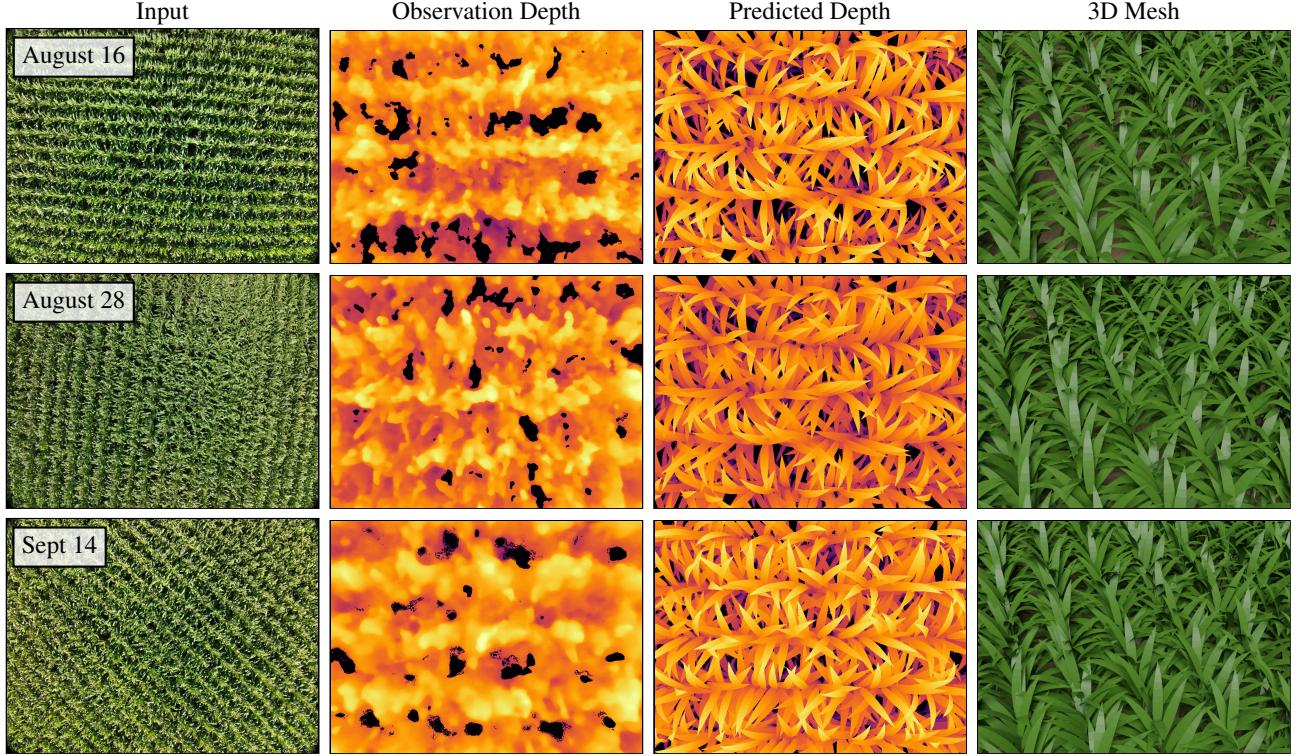
Figure 5. **Qualitative reconstruction results (maize)**. We show that our method can be applied to model maize as well as soybean. From left to right: example images from the drone-captured multi-view data, row-aligned NeRF-rendered depth, procedural model depth, rendered visualization of the procedural mesh. The resulting reconstructions are complete and anatomically realistic despite heavy occlusion.
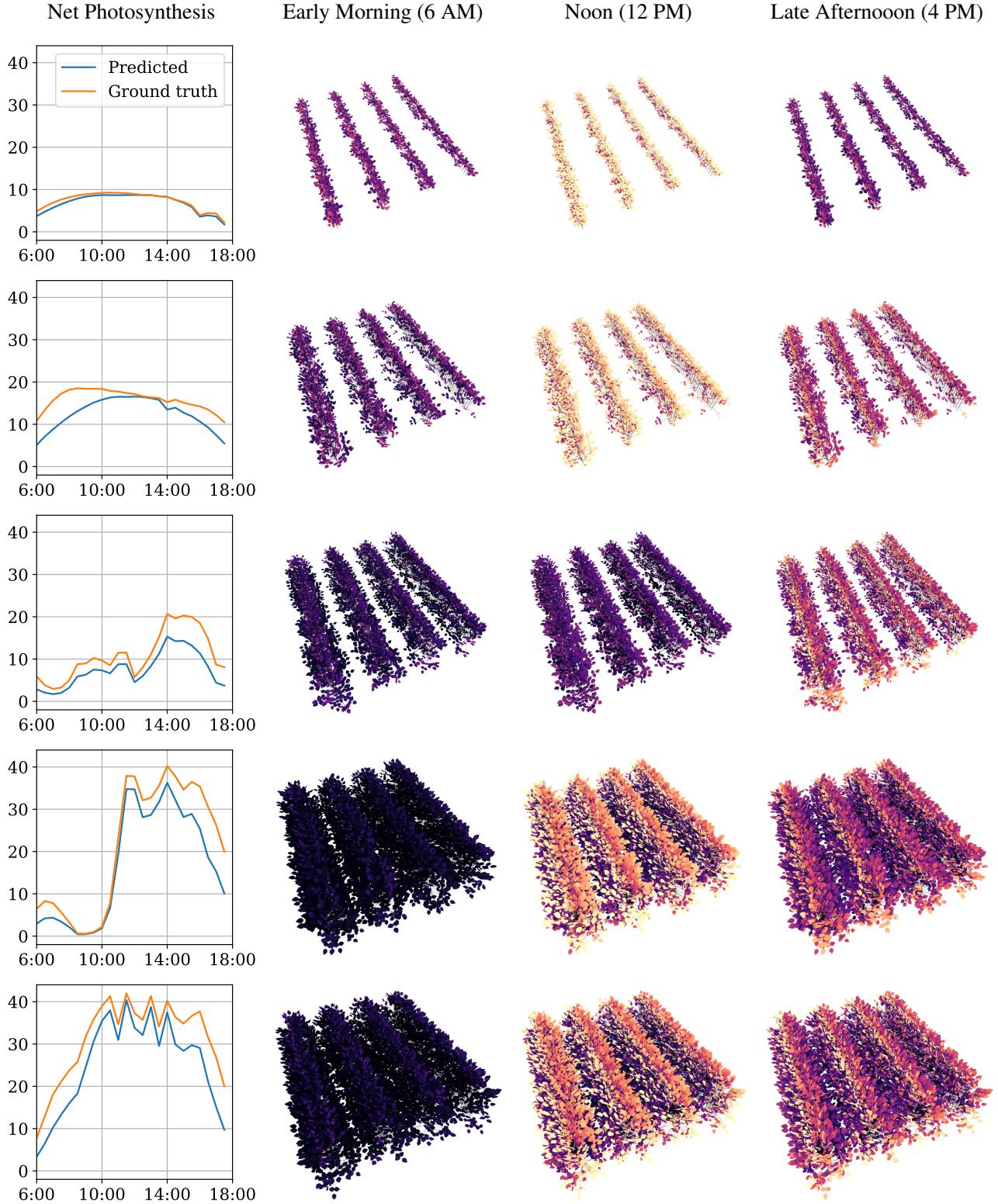
perform Bayesian optimization with a Matern kernel and the simple expected-improvement (EI) acquisition function [9] to estimate the procedural generation parameters that minimize our loss function. We find that, in many scenes, there may be multiple solutions with similar loss values. In order to make our method more robust to different random initialization seeds, we run the optimization 10 times (in parallel) and average the solutions.

## 4. Experiments

**Dataset.** To validate our approach, we collected a dataset of multi-view images in real crop fields in the U.S. Midwest, paired with manual measurements of key morphological variables. For soybean, we collected at 3 geographical locations at 6 different time points throughout the growing season, for a total of 18 scenes. The images were captured covering a $2\,m \times 2\,m$ area using the Polycam app on an iPad Pro, around 50 per scene. The paired manual measurements consist of leaf areas and leaf angles (angle between surface normal and zenith direction). Leaf area was measured using a LAI-2200 scanner, and leaf angle was measured using a protractor. For maize, we collected 5 different time points in one location, with each scene containing around 500 images from a DJI Mavic UAV flying at a height of $13\,m$. The images were aligned using Agisoft Metashape. Here, only

leaf area was measured, not leaf angles. Both the soybean and maize data will be made publicly available.

**Metrics.** Our evaluation metrics (and field measurements) are centered around the commonly considered canopy structure variables of leaf area index (LAI) and the leaf angle distribution. LAI is defined to be the total surface area of leaves per unit of ground area and is widely used in productivity models, climate models, and methods to estimate other vegetative surface properties [31]. Similarly, leaf angle is regarded as a key component of plant ecological strategy with significant impact on land surface properties such as carbon flux, surface temperature and spectral signature [52]. To assess how accurately our reconstructions capture these key traits, we consider the following metrics: **LAI Error (LAIE)** is the root mean squared error (RMSE) between the predicted and ground-truth (manually measured) LAI. **LAI Percent Error (LAIPE)** is the absolute percent error between the predicted and ground-truth LAI. **Angle Mean Error (AME)** is the RMSE between the mean of the predicted leaf angles and the mean of the ground-truth leaf angles. **Angle Standard Deviation Error (ASDE)** is the RMSE between the standard deviation of the predicted leaf angles and that of the ground-truth leaf angles.

Figure 6. **Photosynthesis simulation results**. We perform simulations using Helios [3] on soybean canopies reconstructed by our method. The left column shows a timeseries of the net photosynthesis rate for the crop canopy over the course of a day, in units of $\mu molCO_2/m^2/s$. The other columns show a mesh visualization where each leaf face is colored according to the rate of photosynthesis over that face (brighter = higher rate). The results demonstrate the potential of using our reconstruction pipeline for monitoring crop productivity.

Table 1. **Canopy reconstruction results. Bold**: best, **Blue**: second best.

| | Soybean | | | | Maize | |
|---|---|---|---|---|---|---|
| | LAIE ($\downarrow$) | LAIPE ($\downarrow$) | AME ($\downarrow$) | ASDE ($\downarrow$) | LAIE ($\downarrow$) | LAIPE ($\downarrow$) |
| Poisson | 1.11 | 0.57 | 31.60 | **4.10** | 1.63 | 0.42 |
| MLP | 0.92 | 0.24 | 29.79 | **3.91** | 2.79 | 0.88 |
| Trust-Region | 1.37 | 0.28 | **10.34** | 7.10 | 1.71 | 0.53 |
| Random | 3.38 | 2.84 | 27.92 | 11.83 | 2.42 | 0.75 |
| Ours | **0.69** | **0.15** | 12.07 | 7.39 | **0.97** | **0.26** |

**Baselines.** We compare our reconstruction method with the following baselines: **Poisson** refers to applying Poisson surface reconstruction [21] on the point cloud extracted from NeRF, and then thresholding in 3D to remove the background geometry. All of the remaining mesh faces are then considered as leaf faces. **MLP** refers to using a learned multilayer perceptron (MLP) to predict procedural generation parameters given the histograms, mask area, and view height of the observation depth map. The MLP is trained on 20K pairs of histogram statistics and model parameters, synthesized by running the procedural generation model and rendering depth maps. **Trust-Region** is the same as our proposed method but uses a (constrained) trust-region method [6] for optimization instead of Bayesian optimization. This requires an estimate of the gradient of the loss function, which is obtained via 2-point finite difference estimation [44]. **Random** refers to uniformly randomly sampling values for the procedural generation parameters.

**Implementation Details.** For soybean, we train Nerfacto [46] with default parameters for 20K iterations. For maize, we set the near-plane and far-plane to 0.05 and 6.0 respectively, the MLP width to 128, and the orientation method to "vertical". We defer details about the RANSAC row-fitting and depth rendering to the supplementary. For our loss function, we use $\lambda_{\text{lateral}} = 2$, $\lambda_{\text{sobel}} = 4$, $\lambda_{\text{mask}} = 1$ for soybean and $\lambda_{\text{lateral}} = 1$, $\lambda_{\text{sobel}} = 0$, $\lambda_{\text{mask}} = 1$ for maize. This is because the low NeRF quality in the maize scenes makes the depth derivatives unreliable. For Bayesian optimization, we use scikit-optimize [14] and run for 500 iterations, with 200 iterations for random initialization. Details about histogram bins and baseline implementations can be found in the supplementary.

### 4.1. Canopy Reconstruction

**Qualitative Results.** We show example results at different stages of soybean growth for the same geographic location in Fig. 4. We observe that our method produces complete reconstructions that capture how the shape of the canopy evolves over time, from small seedlings to large bushes. Unlike most existing reconstruction methods, our method is able to estimate the invisible (occluded) portions of the plants as well as the visible portions. The final mesh is also easily decomposable (at both the plant-level and part-level) and is guaranteed to be biologically reasonable thanks to the constraints imposed by the procedural model.

We also show example results for maize in Fig. 5. Due to windy conditions, larger scene scale, and noisy camera pose estimation, the quality of the NeRF is significantly lower in the maize data compared to soybean. However, our method is still able to produce reasonable canopy reconstructions using those observations.

**Quantitative Results.** We provide quantitative results for the canopy reconstruction in Tab. 1. In all metrics except ASDE, we find that our proposed method achieves the highest performance. The angle standard deviation may be estimated less accurately than the other variables because it does not have as strong of an impact on the loss function.

The Poisson reconstruction fails because it completely ignores occluded regions and also frequently stitches nearby leaves together, creating extraneous faces. The MLP gives reasonable predictions in terms of soybean leaf area, but tends to give highly erroneous leaf angles as a result of the domain gap between the synthetic histograms it was trained on and the actual histograms from the NeRF depth maps. This is exacerbated with the maize data, where the NeRF renderings are blurrier. The trust-region optimization produces leaf angles comparable with our method, but does not perform as well for leaf area, most likely due to falling into local minima or making erroneous steps due to the stochasticity of the procedural generation model.

### 4.2. Ablation Studies

We conduct an ablation study on the components of our loss function in Tab. 2. We observe that each of the components contribute to producing a more accurate reconstruction, some significantly more than others. The depth derivative loss has a pronounced beneficial effect on the leaf area metrics. This is because the other statistics may have trouble constraining the leaf size when the canopy has covered most of the ground, while the depth derivative distribution will be strongly shifted upwards with smaller leaves compared to larger ones. Interestingly, removing most of the other terms one at a time does not actually significantly harm the final performance. The hard constraints imposed

Table 2. **Loss function ablations. Bold**: best, **Blue**: second best.

| $L_{\text{depth}}$ | $L_{\text{lateral}}$ | $L_{\text{sobel}}$ | $L_{\text{mask}}$ | LAIE ($\downarrow$) | LAIPE ($\downarrow$) | AME ($\downarrow$) | ASDE ($\downarrow$) |
|:---:|:---:|:---:|:---:|:---:|:---:|:---:|:---:|
| ✓ | | | | 1.62 | 0.52 | 17.00 | 9.26 |
| | ✓ | ✓ | ✓ | **0.67** | 0.19 | 12.08 | **6.81** |
| ✓ | | ✓ | ✓ | 0.71 | **0.16** | **10.37** | 7.69 |
| ✓ | ✓ | | ✓ | 1.94 | 0.49 | **9.73** | 8.15 |
| ✓ | ✓ | ✓ | | 0.78 | 0.19 | 12.30 | 8.03 |
| ✓ | ✓ | ✓ | ✓ | **0.69** | **0.15** | 12.07 | **7.39** |

by the procedural model and the soft constraints imposed by the remaining terms are able to force a similar solution. We still decide to include all the terms in our final method due to the improved LAIPE, which we prefer over LAIE as it is not biased towards the later time points with larger plants.

### 4.3. Photosynthesis Simulation

The outputs of our method can be directly used with radiative transfer models to predict photosynthesis rates, which directly impacts crop productivity. Using Helios [3], a state-of-the-art biophysical modeling framework, we perform radiative transfer simulations on the soybean canopies reconstructed by our method. Helios can predict the net photosynthesis rate for every leaf mesh face at any point in time given a set of environmental variables measured by a flux tower with sensors for temperature, humidity, radiation, etc. We show visualizations of the per-face photosynthesis rate as well as timeseries of the aggregate rate across the entire canopy in Fig. 6. Ground-truth values for the photosynthesis rate are usually calculated at a landscape-level using eddy covariance data from a flux tower [32]. Since we do not have paired flux tower measurements for our dataset (collected in 2023), we use flux tower measurements from a soybean field in the previous year (2022). Because the measurements were collected in a similar location and climate as our data, we can reasonably expect that they reflect the corresponding variables for the field in our data.

The results suggest that the predictions made by the model can accurately represent the net photosynthesis over time as well as its spatial variations throughout the canopy. This validates the potential of our reconstruction pipeline for facilitating large-scale monitoring of crop productivity.

### 4.4. Limitations

One limitation of our method is that the inverse procedural modeling is dependent on the NeRF reconstruction performance, which may degrade due to wind-induced leaf motion. Another limitation is that the procedural generation models used cannot model certain details, e.g. damaged leaves and non-leaf organs of the plants. A final limitation is that our RANSAC-based row-fitting method is sensitive to hyperparameters such as the inlier threshold and segmentation threshold, which may need to be tuned per dataset.

## 5. Conclusion

We presented a novel method for 3D reconstruction of agricultural crops that combines conventional multi-view reconstruction with inverse procedural modeling to produce complete plant shapes despite heavy occlusion. Our method first constructs a neural radiance field (NeRF) and then optimizes a procedurally generated model to be consistent with the NeRF in a visibility-aware manner. We validate our method on data collected in real-world agricultural fields and show that it can reconstruct realistic crop canopies across a variety of plant growth stages. We also show that the reconstructions can enable realistic simulations of crop photosynthesis using a radiative transfer model. The method could potentially be improved in the future by incorporating plant growth priors for temporal consistency, or optimizing the model in a coarse-to-fine manner to enable reconstructing more shape details.

## References

[1] Daniel G Aliaga, İlke Demir, Bedrich Benes, and Michael Wand. Inverse procedural modeling of 3d models for virtual worlds. In *ACM SIGGRAPH 2016 Courses*, pages 1–316. 2016. 2

[2] Muhammad Arbab Arshad, Talukder Jubery, James Afful, Anushrut Jignasu, Aditya Balu, Baskar Ganapathysubramanian, Soumik Sarkar, and Adarsh Krishnamurthy. Evaluating nerfs for 3d plant geometry reconstruction in field conditions. *arXiv preprint arXiv:2402.10344*, 2024. 2

[3] Brian N Bailey. Helios: A scalable 3d plant and environmental biophysical modeling framework. *Frontiers in Plant Science*, 10:1185, 2019. 7, 9

[4] Ramesh Chand. The global food crisis: causes, severity and outlook. *Economic and Political Weekly*, pages 115–122, 2008. 1

[5] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2670, 2019. 2

[6] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000. 8

[7] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In *European Conference on Computer Vision*, pages 715–733, 2020. 2

[8] Lubin Fan, Przemyslaw Musialski, Ligang Liu, and Peter Wonka. Structure completion for facade layouts. *ACM Transactions on Graphics*, 33:1 – 11, 2014. 2

[9] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. 4, 6

[10] Pierre Friedlingstein, Michael O'sullivan, Matthew W Jones, Robbie M Andrew, Luke Gregor, Judith Hauck, Corinne Le Quéré, Ingrid T Luijkx, Are Olsen, Glen P Peters, et al. Global carbon budget 2022. *Earth System Science Data*, 14 (11):4811–4900, 2022. 1

[11] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *2009 IEEE 12th international conference on computer vision*, pages 80–87. IEEE, 2009. 2

[12] Aditya Ganeshan, R Kenny Jones, and Daniel Ritchie. Improving unsupervised visual program inference with code rewriting families. In *ICCV*, pages 15791–15801, 2023. 2

[13] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv preprint arXiv:2311.12775*, 2023. 2

[14] Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. scikit-optimize/scikit-optimize: v0. 8.1. *Zenodo*, 2020. 8

[15] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 1

[16] Kewei Hu, Ying Wei, Yaoqiang Pan, Hanwen Kang, and Chao Chen. High-fidelity 3d reconstruction of plants using neural radiance field. *arXiv preprint arXiv:2311.04154*, 2023. 2

[17] Satoshi Ikehata, Hang Yang, and Yasutaka Furukawa. Structured indoor modeling. In *Proceedings of the IEEE international conference on computer vision*, pages 1323–1331, 2015. 2

[18] Hamid Izadinia, Qi Shan, and Steven M. Seitz. Im2cad. In *CVPR*, 2017. 2

[19] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. 13

[20] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-sim: Learning to generate synthetic datasets. In *ICCV*, pages 4551–4560, 2019. 2

[21] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, page 0, 2006. 8

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2

[23] Bosheng Li, Jacek Kałużny, Jonathan Klein, Dominik L Michels, Wojtek Pałubicki, Bedrich Benes, and Sören Pirk. Learning to reconstruct botanical trees from single images. *ACM Transactions on Graphics (TOG)*, 40(6):1–15, 2021. 1, 2

[24] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *CVPR*, 2023. 1, 2

[25] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021. 2

[26] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *CVPR*, 2023. 1

[27] Steven Longay, Adam Runions, Frédéric Boudon, and Przemyslaw Prusinkiewicz. Treesketch: Interactive procedural modeling of trees on a tablet. In *SBIM@ Expressive*, pages 107–120. Citeseer, 2012. 1, 2

[28] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Nießner, and Vittorio Ferrari. Vid2cad: Cad model alignment using multi-view constraints from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1320–1327, 2023. 2

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 3

[30] World Health Organization et al. *The state of food security and nutrition in the world 2022: Repurposing food and agricultural policies to make healthy diets more affordable*. Food & Agriculture Org., 2022. 1

[31] Geoffrey G Parker. Tamm review: Leaf area index (lai) is both a determinant and a consequence of important processes in vegetation canopies. *Forest Ecology and Management*, 477:118496, 2020. 6

[32] Gilberto Pastorello, Carlo Trotta, Eleonora Canfora, Housen Chu, Danielle Christianson, You-Wei Cheah, Cristina Poindexter, Jiquan Chen, Abdelrahman Elbashandy, Marty Humphrey, et al. The fluxnet2015 dataset and the oneflux processing pipeline for eddy covariance data. *Scientific data*, 7(1):225, 2020. 9

[33] Przemyslaw Prusinkiewicz, Jim Hanan, and Radomír Měch. An l-system-based plant modeling language. In *International workshop on applications of graph transformations with industrial relevance*, pages 395–410. Springer, 1999. 1, 2

[34] Binxiang Qian, Wenjiang Huang, Donghui Xie, Huichun Ye, Anting Guo, Yuhao Pan, Yin Jin, Qiaoyun Xie, Quanjun Jiao,

Biyao Zhang, et al. Coupled maize model: A 4d maize growth model based on growing degree days. *Computers and Electronics in Agriculture*, 212:108124, 2023. 1, 3, 4, 12

[35] Farah Saeed, Jin Sun, Peggy Ozias-Akins, Ye Juliet Chu, and Changying Charlie Li. Peanutnerf: 3d radiance field for peanuts. In *CVPRW*, pages 6253–6262, 2023. 2

[36] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, pages 214–226. Wiley Online Library, 2007. 2, 4

[37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[38] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 1, 2

[39] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015. 4

[40] Rebecca A Slattery and Donald R Ort. Perspectives on improving light distribution and light use efficiency in crop canopies. *Plant Physiology*, 185(1):34–48, 2021. 1

[41] Claus Smitt, Michael Halstead, Patrick Zimmer, Thomas Läbe, Esra Guclu, Cyrill Stachniss, and Chris McCool. Pagnerf: Towards fast and efficient end-to-end panoptic 3d representations for agricultural robotics. *IEEE Robotics and Automation Letters*, 9(1):907–914, 2023. 2

[42] Qingfeng Song, Venkatraman Srinivasan, Steve P Long, and Xin-Guang Zhu. Decomposition analysis on soybean productivity increase under elevated co2 using 3-d canopy model reveals synergestic effects of co2 and light in photosynthesis. *Annals of botany*, 126(4):601–614, 2020. 1, 3, 4, 12

[43] Ondrej Stava, Sören Pirk, Julian Kratt, Baoquan Chen, Radomír Měch, Oliver Deussen, and Bedrich Benes. Inverse procedural modelling of trees. In *Computer Graphics Forum*, pages 118–131. Wiley Online Library, 2014. 1, 2

[44] John C Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004. 8

[45] Ping Tan, Tian Fang, Jianxiong Xiao, Peng Zhao, and Long Quan. Single image tree modeling. *ACM Transactions on Graphics (TOG)*, 27(5):1–7, 2008. 2

[46] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM Transactions on Graphics (SIGGRAPH)*, 2023. 1, 2, 3, 8, 12, 13

[47] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, pages 298–372. Springer, 2000. 2

[48] USDA National Agricultural Statistics Service. NASS-quick stats. *UNAS Service (Ed.)*, 2021. 3

[49] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 1, 2

[50] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. In *CVPR*, 2021. 1, 2

[51] Fuzhang Wu, Dong Ming Yan, Weiming Dong, Xiaopeng Zhang, and Peter Wonka. Inverse procedural modeling of facade layouts. *ACM Transactions on Graphics*, 33(4):121, 2014. 2

[52] Xi Yang, Rong Li, Andrew Jablonski, Atticus Stovall, Jongmin Kim, Koong Yi, Yixin Ma, Daniel Beverly, Richard Phillips, Kim Novick, et al. Leaf angle as a leaf and canopy trait: Rejuvenating its role in ecology with new technology. *Ecology Letters*, 2023. 6

[53] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *CVPR*, 2020. 1, 2

[54] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 1, 2

[55] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. *arXiv preprint arXiv:2302.14859*, 2023. 2

[56] Fenggen Yu, Qimin Chen, Maham Tanveer, Ali Mahdavi Amiri, and Hao Zhang. D $^2$ csg: Unsupervised learning of compact csg trees with dual complements and dropouts. *NeurIPS*, 36, 2024. 2

[57] Jingyang Zhang, Shiwei Li, Zixin Luo, Tian Fang, and Yao Yao. Vis-mvsnet: Visibility-aware multi-view stereo network. *International Journal of Computer Vision*, 131(1): 199–214, 2023. 1, 2

[58] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 13

[59] Rongsheng Zhu, Kai Sun, Zhuangzhuang Yan, Xuehui Yan, Jianglin Yu, Jia Shi, Zhenbang Hu, Hongwei Jiang, Dawei Xin, Zhanguo Zhang, et al. Analysing the phenotype development of soybean plants using low-cost 3d reconstruction. *Scientific Reports*, 10(1):7055, 2020. 1

# CropCraft: Inverse Procedural Modeling for 3D Reconstruction of Crop Plants

## Supplementary Material

### Abstract

*In the following supplementary material, we provide additional experimental details (Sec. A) and additional experimental results (Sec. B). We invite readers to watch the supplementary video (`supp.mp4`) for further visualization.*

## A. Additional Experimental Details

### A.1. Additional Procedural Model Details

**Soybean Model.**    Our procedural soybean model is based on the mCanopy model from Song et al. [42]. Each plant consists of a main stem with up to 14 (trifoliate) nodes, and up to 6 branch stems with up to 2 nodes each. Each node consists of three coplanar leaves, with the middle leaf connected by a small petiole, and is connected to its stem by a larger petiole. The leaf length (and width), the petiole length, the angle with the stem, and the distance between nodes are variable, as they are affected by the multiplier parameters. On the main stem, when all the multipliers are set to 1, the variables follow a profile from top to bottom based on field data measured by Song et al. [42]. Branches start growing with every new main stem node starting from the 8th, with one branch per main stem node starting from the bottom and up to the 6th. The number of nodes per branch is selected randomly according to the distribution defined by Song et al. [42]. The variables for the branch nodes are the same across nodes (but are affected by the multipliers). When the number of nodes is not an integer, it is rounded up and the last node is scaled down in size linearly according to the remainder. Additionally, random noise is added to each petiole angle following a Gaussian distribution with mean 0 and standard deviation 5 degrees. Each node grows on alternating sides of their stem, with additional azimuthal rotation following a Gaussian distribution with mean 0 and standard deviation 60 degrees. The range of possible values for each model parameter are provided in Tab 3.

| Parameter | Range |
|---|---|
| Leaf length multiplier | 0.5 - 1.5 |
| Petiole length multiplier | 0.5 - 2.0 |
| Petiole angle multiplier | 0.5 - 4.0 |
| Internode length multiplier | 0.5 - 2.0 |
| Number of nodes | 1.0 - 14.0 |

Table 3. **Soybean model parameter ranges.**

**Maize Model.**    Our procedural maize model is based no the coupled maize model from Qian et al. [34]. Each plant consists of a main stem with up to 18 nodes with one leaf each. Each leaf consists of 10 trapezoidal segments that bend towards the ground following a polynomial function that depends on the leaf order [34]. To allow for different bending angles, we introduce a parameter that additively shifts all of the leaf orders by the same amount. The leaf lengths, widths, and internode lengths follow a profile from bottom to top based on field data measured in the U.S. Midwest. Each node grows on alternating sides of their stem, with additional azimuthal rotation following a Gaussian distribution with mean 0 and standard deviation 60 degrees. The range of possible values for each model parameter are provided in Tab 4.

| Parameter | Range |
|---|---|
| Leaf length multiplier | 0.8 - 1.2 |
| Leaf order shift | -4.0 - 4.0 |
| Internode length multiplier | 0.8 - 1.2 |
| Number of nodes | 1.0 - 18.0 |

Table 4. **Maize model parameter ranges.**

### A.2. Additional Row Fitting and Depth Rendering Details

Our row fitting procedure operates on a point cloud sampled from the NeRF, using the sampling implementation in Nerfstudio [46]. For the soybean data, we sample 100K points in a $2\,\mathrm{m} \times 2\,\mathrm{m} \times 3\,\mathrm{m}$ box centered at $(0, 0, -1.5)$m. Voxel-downsampling is performed at a resolution of $1\,\mathrm{cm}$. Color-thresholding is performed in LAB color space. All points with $L^* < 0$ and $b^* < 1$ are removed. Then, points with $a^* < 2$ are classified as ground points, and the rest are classified as plant points. RANSAC is used to fit a plane to the ground points using an inlier threshold of $5\,\mathrm{cm}$ and 1000 max iterations. For row fitting, the points whose distance from the ground are below the 50th percentile are discarded. RANSAC is applied sequentially to fit lines with an inlier threshold of $20\,\mathrm{cm}$ until the number of points remaining is either less than 1000 or less than 20% of the starting number. We then take the row with the most inliers and fit a line through the inliers again using least-squares fitting. The camera pose for depth rendering is set to be directly above the mean of the inlier points at a height of $1\,\mathrm{m}$ from the ground, facing downwards and with the camera's $x$-axis aligned with the row line. If the number of initial plant points is greater than 75% of the total, we instead use
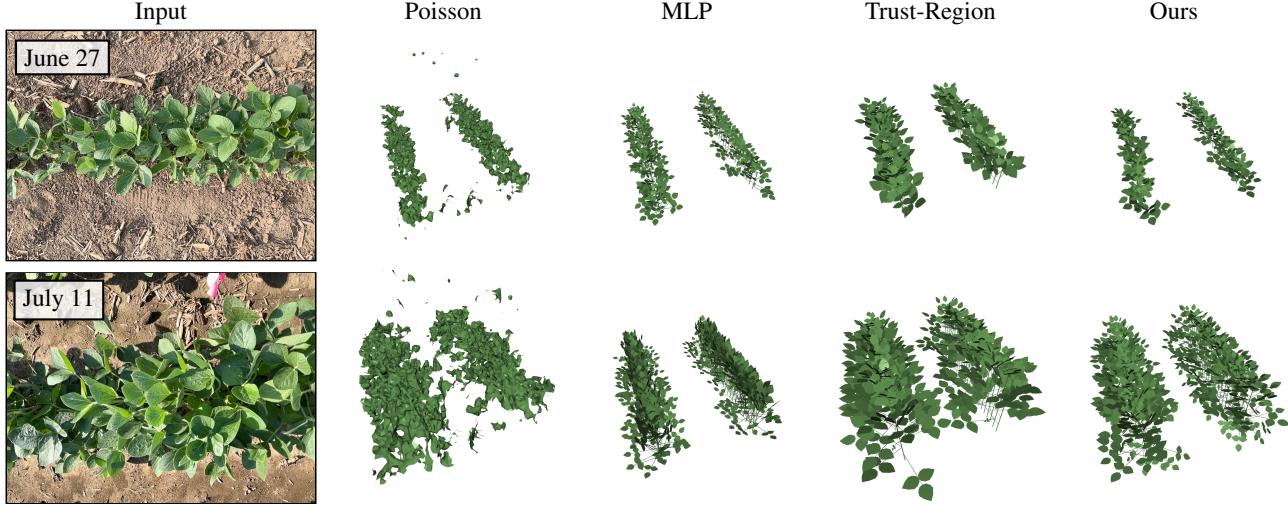
Figure 7. **Qualitative reconstruction comparison**. We compare example reconstructions from each of the baseline methods. The first column shows example images from the multi-view data collected at each time point, and the remaining columns show the corresponding mesh reconstructions.

the 70th percentile for distance from the ground, a row inlier threshold of $25\,\mathrm{cm}$, and a render height of $1.25\,\mathrm{m}$. The render resolution is $994 \times 738$ with a vertical FOV of 50 degrees.

For the maize data, we sample 5M points in a $2 \times 2 \times 2$ box around the origin in the scaled scene. The $L^*, a^*$, and $b^*$ thresholds are set to 32, 0, and 0 respectively. The ground-inlier threshold is set to $10\,\mathrm{cm}$ and the render height is set to $5\,\mathrm{m}$. The row fitting is done in a ROI with the shape of a cylinder with radius $2\,\mathrm{m}$ whose axis is perpendicular to the ground plane.

### A.3. Mask and Histogram Details

The foreground mask $\mathbf{M}_{\mathrm{obs}}$ is obtained by color-thresholding on the RGB render combined with thresholding on 3D coordinates. For soybean, points are marked as background if their $a^*$ channel is less than -8 and their $z$-coordinate is greater than $0.25\,\mathrm{m}$ short of the render height, or their $z$-coordinate is less than $0.1\,\mathrm{m}$, or their $y$-coordinate is at least $0.5\,\mathrm{m}$ away from zero. For maize, points are marked as background if $a^*$ channel is less than -8 and their $L^*$ channel is greater than 40 and their $z$-coordinate is greater than $2\,\mathrm{m}$ short of the render height, or their $z$-coordinate is less than $0.1\,\mathrm{m}$, or their $y$-coordinate is at least $3\,\mathrm{m}$ away from zero. $\mathbf{M}_{\mathrm{pred}}$ is obtained trivially by not rendering any background.

For soybean, the depth histogram has 20 equally spaced bins from $0.1\,\mathrm{m}$ to the render height. The lateral histogram has 10 equally spaced bins from zero to half the render height. The depth derivative histogram has 10 equally spaced bins from zero to 0.004. A Gaussian blur with kernel size 25 is applied to the depth map before the calculation

of Sobel derivatives, since the NeRF rendering is generally blurrier than the procedural mesh rendering. For maize, the depth histogram has 10 equally spaced bins from $2\,\mathrm{m}$ to the render height, and the blur kernel size is 55.

### A.4. Baseline Implementation Details

We provide implementation details for the baseline reconstruction methods:

- **Poisson:** We use the Poisson surface reconstruction implementation by Open3D [58] integrated into Nerfstudio [46]. For soybean, we use 100K points with 50K faces in the same bounding box as in our method and filter out points whose $z$-coordinates are lower than 0.2 plus the 1st percentile. For maize, we use a $4\,\mathrm{m} \times 4\,\mathrm{m}$ bounding column and $0.1\,\mathrm{m}$ instead of 0.2.

- **MLP:** The MLP input consists of the same histograms used for optimization, concatenated with the render height and mask area. The training data is generated by uniformly randomly sampling parameter values and computing the corresponding MLP inputs. For soybean, we sample 10K pairs with render height $1.0\,\mathrm{m}$ and 10K pairs with render height $1.25\,\mathrm{m}$. For maize, we sample 20K pairs with render height $5.0\,\mathrm{m}$. The MLP has two hidden layers of size 512, and we train for 200 epochs using an Adam optimizer with $\alpha = 0.001, (\beta_1, \beta_2) = (0.9, 0.999)$, and weight decay of $10^{-5}$.

- **Trust-Region:** We use the trust-region method implemented by the "trust-constr" option of SciPy's `optimize.minimize` function [19], with the maximum number of iterations set to 500. For fair comparison with our Bayesian optimization method, we run it 10 times with different random initializations. We take the
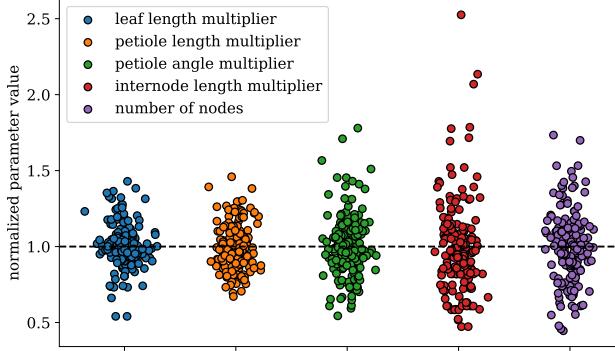
Figure 8. **Jitter plots of optimized parameter values**. The values are normalized by dividing by the per-scene average. The individual optimization runs generally do not stray far from the average, but taking the average ultimately gives a better result.

solution with the lowest loss value instead of averaging, since we find it gives better performance.

## B. Additional Results

### B.1. Additional Qualitative Results

We provide qualitative comparisons of the different reconstruction baselines in Fig. 7. We observe that Poisson reconstruction gives incomplete and implausible canopy shape, motivating the use of inverse procedural modeling. Compared to the other procedural generation baselines, our method generally produces the parameters that best reflects the true real-world canopy. In the examples shown, the MLP produces leaves that are too small, and the trust-region optimization produces leaves that are too large.

### B.2. Bayesian Optimization Performance Analysis

We visualize the distributions of the optimized parameter values before averaging in Fig. 8. The values include the 10 runs for all 18 soybean scenes and are normalized by the averaged result. We observe that there is a moderate amount of variance in the optimization results. We then examine how reconstruction performance changes as we average across different numbers of random initializations for Bayesian optimization in Tab. 5. Although there is no significant improvement in the leaf angle metrics, we observe that the leaf area estimates improve overall when averaging over more runs, eventually plateauing around 10-20 runs. Note that the constraints of the procedural model generally ensure that all possible combinations of parameters yield plausible 3D reconstructions, so there is no real concern over the average giving an unreasonable result.

We also examine how performance changes as the number of Bayesian optimization iterations varies in Tab. 6. Here, we are still averaging over 10 runs. In addition to the

| # of runs | Soybean | | | |
| | LAIE (↓) | LAIPE (↓) | AME (↓) | ASDE (↓) |
|---|---|---|---|---|
| 1 | 0.76 | 0.23 | 12.18 | 7.81 |
| 2 | 0.79 | 0.20 | 12.18 | 7.74 |
| 5 | 0.78 | 0.16 | 12.13 | 7.74 |
| 10 | 0.69 | 0.15 | **12.07** | **7.39** |
| 20 | **0.69** | **0.14** | 12.23 | 7.77 |

Table 5. **Effect of averaging solutions over runs. Bold**: best.

previous metrics, we report the average value of the minimized loss function and time per run. Performance does tend to improve with more iterations, but takes increasing amounts of time per iteration due to larger and larger kernel computations. In addition, beyond a certain point, decreases in the loss function does not necessarily translate to better reconstruction quality.

### B.3. Wind Simulation

The procedurally generated meshes from our model can also be used for dynamics simulations. We show rendered frames of a wind simulation using NVIDIA PhysX in Fig. 9. Visually, the results appear to show a physically plausible animation of leaves blowing in the wind. Video visualizations can be found in the supplementary video.

| # of iterations | Soybean | | | | | |
|---|---|---|---|---|---|---|
| | LAIE ($\downarrow$) | LAIPE ($\downarrow$) | AME ($\downarrow$) | ASDE ($\downarrow$) | Average $L(\downarrow)$ | Time ($\downarrow$) |
| 300 | 0.73 | 0.18 | 12.32 | 7.77 | 0.0172 | **3m** |
| 400 | 0.77 | 0.17 | 12.18 | 7.77 | 0.0133 | 9m |
| 500 | **0.69** | 0.15 | **12.07** | **7.39** | 0.0110 | 16m |
| 600 | 0.70 | **0.13** | 12.56 | 8.01 | **0.0096** | 26m |

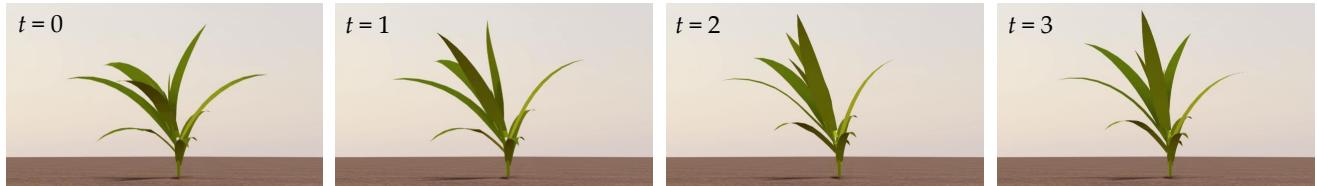Table 6. **Effect of number of Bayesian optimization iterations. Bold**: best.



Figure 9. **Wind simulation.** We show that the procedurally generated mesh outputs can be used for physically realistic dynamics simulations. Here, a uniform wind force is applied directly from the right.