2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Robust Plant Phenotyping via Model-based Optimization

Paloma Sodhi[1], Hanqi Sun[2], Barnabás Póczos[3] and David Wettergreen[1]

Abstract— **Plant phenotyping is the measurement of observable plant traits. Current methods for phenotyping in the field are labour intensive and error prone. High throughput plant phenotyping in an automated and noninvasive manner is crucial to accelerating plant breeding methods.**

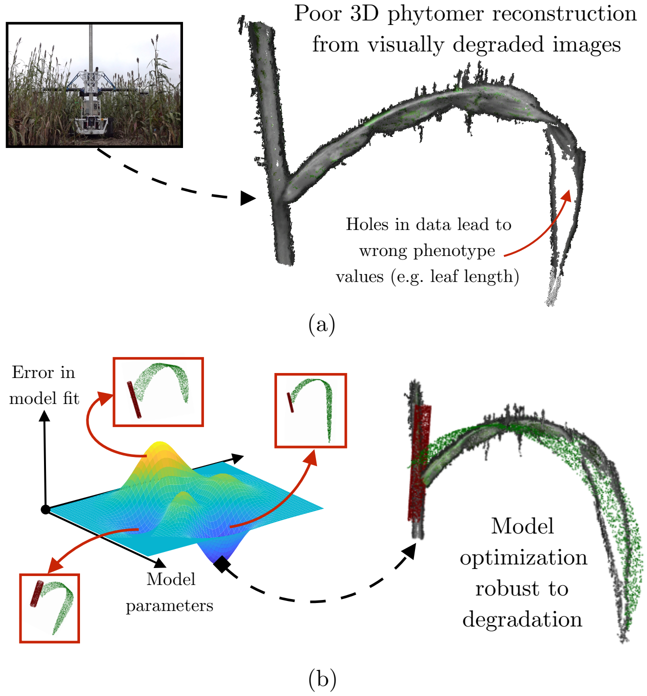
Occlusions and non-ideal sensing conditions is a major problem for high throughput plant phenotyping with most state-of-the-art 3D phenotyping algorithms relying heavily on heuristics or hand-tuned parameters. To address this problem, we present a novel model-based optimization approach for estimating plant physical traits from plant units called phytomers. The proposed approach involves sampling parameterized 3D plant models from an underlying probability distribution. It then optimizes, making the mass of this probability distribution approach true parameters of the model.

Reformulating the phenotyping objective as a search in the space of plant models lets us reason about the plant structure in a holistic manner without having to rely on hand-tuned parameters. This makes our approach robust to noise and occlusions as frequently encountered in real world environments. We evaluate our approach for plant units taken across simulated, greenhouse and field environments. This work furthers field-based robotic phenotyping capabilities paving the way for plant biologists to study the coupled effect of genetics and environment on improving crop yields.

I. Introduction

Plant phenotyping is the process of computing quantitative measurements of observable plant traits. A subset of these observable traits are physical attributes like stem diameter, leaf angle, leaf length and leaf width. Knowing plant phenotypes accurately and throughout a plant's growth is central to making breeding decisions so as to produce crops with higher yields, drought tolerance and disease resistance. Currently established processes for measuring plant phenotypes in outdoor crop fields is labour intensive, error prone and has limited throughput leading to what has come to be known as the *phenotyping bottleneck* in the plant breeding pipeline [1]. High throughput automated methods employing robotic platforms and advances in computer vision and machine learning methods are hence crucial to improved production. We are developing high-throughput phenotyping methods for accelerated breeding of biofuel crops like *sorghum*.

This paper addresses the problem of mapping plant subunits called *plant phytomers* to their phenotype values. A plant phytomer, as seen in Fig. 1, consists of a leaf, its sheath and the stem segment on which the leaf resides. It can be thought of as a functional building block of the plant and has special significance for phenotyping purposes, since phenotypic traits like leaf length, leaf angle, leaf width, and stem diameter can be estimated directly from this structure. Our high throughput phenotyping robotic platform shown in Fig. 1 has been used for collection of outdoor field data used in our experiments. The uniqueness of this platform lies in being able to collect 2D images of a plant at multiple vertical heights from different horizontal viewpoints. More details on the platform and data acquisition processes can be found in [2]. In our previous work [3], we performed multi-view 3D reconstruction on a sequence of 2D images collected by this platform and reconstructed 3D models of sorghum plant phytomers.



(a)

(b)

Fig. 1. (a) shows the robotic phenotyping platform used for collecting outdoor field data along with a 3D reconstructed plant phytomer obtained from field images. (b) shows proposed approach sampling plant models from an underlying distribution and optimizing for making the mass of this distribution approach phenotypes of the reconstructed 3D field phytomer.

The focus of this paper is extraction of phenotype values from already reconstructed 3D plant phytomers. The phytomers obtained from outdoor field environments are typically noisy and occluded as can be seen in Fig. 1(a). Methods using traditional 3D geometric algorithms for doing

[1] Paloma Sodhi, David Wettergreen are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {psodhi,dsw}@andrew.cmu.edu

[2] Hanqi Sun is with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA hanqis@andrew.cmu.edu

[3] Barnabás Póczos is with the Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA bapoczos@andrew.cmu.edu

This work was supported by the U.S. Department of Energy TERRA program under grant DE-AR0000596.

978-1-5386-8093-3/18/$31.00 ©2018 IEEE

phenotyping in controlled indoor environments [4]–[6] can break down under such noise and occlusion levels since assumptions on connectivity and non-degeneracy of corresponding 3D surface meshes is violated in such cases. We instead propose a model-based optimization approach employing parametric models capable of leveraging prior knowledge on the structure of a plant phytomer. This makes our approach more robust to occlusions and less sensitive to noise. In particular, the main contributions of this paper are:

(1) Formulation of the phenotyping problem as that of an optimization in the space of plant models. Design of a parametric plant phytomer model for this objective.
(2) Evaluation of proposed approach on simulated, greenhouse as well as field data
(3) Robustness analysis for plant phytomer data corrupted with noise and occlusions

II. Related Work

A key capability of high throughput phenotyping platforms is the ability to non-destructively capture plant traits. To this end, phenotyping platforms deploy a variety of imaging modalities like 2D/3D visible imaging, multi-spectral imaging, thermal imaging and flourescence imaging. For purposes of phenotyping, the use of 3D visible imaging is an easy and cost-effective way of making metric measurements of plant physical traits purely from imaging.

Most of the state-of-the-art in plant phenotyping using 3D imaging, however, is in controlled greenhouse environments [4]–[7], where it is possible to obtain high fidelity 3D reconstructions with lower noise and occlusion levels. Drapikowski et. al in [4] estimate phenotypes like length, width, and area of leaves using 3D surface mesh operations on segmented leaf meshes. McCormick et. al in [5] too generate segmented 3D surface meshes for estimating phenotypes like leaf area and leaf angle. Such methods, however, make certain assumptions about the underlying 3D data such as accurate segmentation into stem and leaf classes, non-degenerate surface reconstructions, connectedness in reconstructed 3D surfaces and validity of geometric shape approximations. In presence of noise and outliers, as commonly encountered in actual field data, these assumptions may fail resulting in unpredictable values.

In contrast, approaches leveraging some prior knowledge related to the 3D plant model are able to reason about the plant structure in a much more holistic manner and are hence more robust to noise and occlusions. There have been methods proposed for recognizing and matching 3D object models in cluttered point clouds for applications like table top manipulation, industrial inspection [8], [9]. However, these methods rely either on geometric primitives or on a prior library of fixed models. It is difficult to come up with a library of fixed models that can exhaustively cover naturally occurring plants. Instead parametric models defined recursively are more suitable for modeling plant structures.

In graphics and plant biology literature, a popular tool for modeling plants with recursive branching structures like trees and bushes is the Lindenmayer System (L-System) [10]. L-system is a type of a formal grammar consisting of an alphabet of symbols that encode the geometric structure of the plant like number of branch segments, branching angle, branch length and branch thicknesses. Binney et. al in [7] utilize the L-system grammar to parametrically model a tree branch as a sequence of cylinders. They then utilize a particle filter based approach to converge to model parameters that best match an observed tree branch. However, their sequence of cylinders model is fairly simplistic for modeling sorghum leaf shapes as required in our case. Moreover, they test their algorithm on only simulated data and a simplistic real branch in a controlled indoor environment.

In this paper, we propose an approach that parametrically models complex leaf shapes as part of the plant phytomer structure and employs a model-based optimization method to search in the space of model parameters for the most likely model. Parameters of the most likely model in-turn gives us our phenotypes of interest. We show results for both indoor greenhouse and outdoor field environments.

III. Plant Phytomer Extraction

Before going into the details of our proposed approach for estimating phenotypes from already extracted 3D plant phytomers, we will briefly go into the system setup and algorithms used for obtaining 3D plant phytomers from raw 2D images. Below we describe this for greenhouse and field environments. This is the focus of our previous work and more details can be found in [3].

A. Indoor Greenhouse Environments

For generating greenhouse 3D plant phytomers, we use data collected by McCormick, et al. in [5]. They place sorghum plants on a turntable and capture 360 degree view depth images at 30° increments using a Kinect camera. The multiple depth images obtained are then fused together into a single 3D point cloud using the iterative closest point (ICP) algorithm. We then extract plant phytomer structures from the reconstructed plant point cloud using a 3D variant of the region growing algorithm. Fig. 3(a) illustrates the input imaging viewpoints and output 3D plant phytomers for the greenhouse environment.

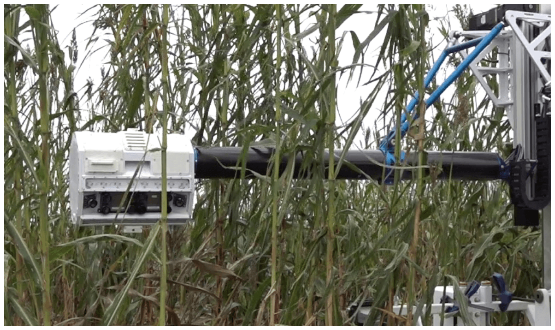
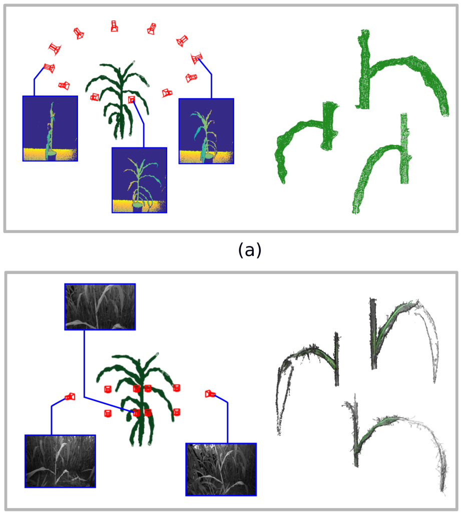


Fig. 2. A closeup of the multi-camera sensor pod deployed on the robotic platform. The sensor pod contains eight forward facing and two additional cameras verged on either ends at an angle of 30°. The sensor pod is connected to a robotic arm and can collect images at multiple plant heights.



(a)

(b)

Fig. 3. Preprocessing to obtain 3D plant phytomers from raw input images. (a) shows input depth imaging viewpoints and output 3D plant phytomers for indoor greenhouse environments. (b) shows input grayscale imaging viewpoints and output 3D plant phytomers for outdoor field environments.

B. *Outdoor Field Environments*

A close-up of one of the multi-camera sensor pods deployed between sorghum crop rows in the field is shown in Fig. 2. The sensor pod contains eight forward facing cameras arranged in two rows along with two additional cameras verged on either ends at an angle of 30°. The sensor pod is mounted on a robotic arm that moves vertically to collect 2D images at multiple plant heights inside the canopy. We utilize a structure-from-motion + multi-view stereo pipeline proposed in [11] for generating 3D point clouds of the sorghum plant phytomer from 2D grayscale image sequences captured in the field.

Fig. 3(b) illustrates the input imaging viewpoints and output 3D plant phytomers for the field environment. It can be seen that the plant phytomer reconstructions obtained in outdoor field environments is much more degraded with high degree of noise and occlusions. This is primarily due to constraints in the imaging sensors and viewpoints that can deployed in a field setting as well as due to added scene clutter from background and neighboring plants.

IV. Proposed Approach

We now address the problem of computing phenotype values from reconstructed 3D plant phytomer structures. The overall idea is to generate parametric models of *plant phytomers* and compare those against the reference plant phytomer whose phenotypes we wish to estimate. The parameters used for generating the phytomer models are taken to be random variables sampled from an underlying probability distribution. The objective is to make the mass of this probability distribution approach the true parameters of the reference plant phytomer. We describe below details related to the problem formulation (Section IV-A), parametric model designed for representing a plant phytomer (Section IV-B), error metric computation (Section IV-C) and the model-based optimization employed (Section IV-D). The proposed overall approach is also illustrated in Fig. 4.

A. *Problem Formulation*

We represent the parametric model of the phytomer as $G(\mathbf{X})$ that takes as input a k-dimensional vector $\mathbf{X} = \begin{bmatrix} X_1 & X_2 & \ldots & X_k \end{bmatrix}^T$ and outputs a point cloud $\mathcal{P}$ of the phytomer model. The input model parameters $\mathbf{X}$ are in turn random variables sampled from a probability distribution function $f(\cdot;\mathbf{u})$ parameterized by a finite-dimensional real vector $\mathbf{u}$. The input model parameters coincide with phenotypes that we are interested in estimating for the plant phytomer. For instance, parameters for leaf angle, leaf length and leaf width all fall under phenotypes of interest.

The point cloud $\mathcal{P}$ generated by $G(\mathbf{X})$ for each sampled value of the parameter vector $\mathbf{X}$ is then compared against the reference phytomer point cloud $\mathcal{P}^{ref}$ whose phenotype values we wish to estimate. An error function $E(\mathcal{P}, \mathcal{P}^{ref})$ computes a scalar value representing distance error between a generated point cloud $\mathcal{P}$ and reference point cloud $\mathcal{P}^{ref}$. A corresponding score value $S(\mathbf{X})$ is computed alongside, and is expressed as $S(\mathbf{X}) = -E(\mathcal{P}, \mathcal{P}^{ref})$.

The phytomer model function $G(\mathbf{X})$ and consequently the score function $S(\mathbf{X})$ that we compute does not have a closed form expression and is also expensive to compute. This necessitates the use of stochastic optimization based techniques for maximizing the score function. The stochastic optimization routine updates the parameter vector $\mathbf{u}$ defining the family of probability density functions (pdfs) $f(\cdot;\mathbf{u})$ from which values of $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ are being sampled. The pdf parameters are updated so as to make the mass of the pdf $f(\cdot;\mathbf{u})$ approach parameters of the reference plant phytomer whose phenotypes we wish to estimate.

B. *Phytomer Model Description*

The phytomer model function $G(\mathbf{X})$ takes as input the k-dimensional vector $\mathbf{X} = \begin{bmatrix} X_1 & X_2 & \ldots & X_k \end{bmatrix}^T$ and outputs a point cloud $\mathcal{P}$ of the phytomer model. We design our parametric sorghum phytomer model based on the structure of bioenergy sorghum [12]. We consider $k = 6$ input model parameters expressed as,

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = \begin{bmatrix} \text{stem diameter} \\ \text{stem length} \\ \text{leaf angle} \\ \text{leaf length} \\ \text{leaf width} \\ \text{leaf rotate angle} \end{bmatrix} \qquad (1)$$

The stages involved in converting input model parameter vector $\mathbf{X}$ into point cloud $\mathcal{P}$ are illustrated in Fig. 5(a)-(d). For generating the initial phytomer skeleton shown in Fig. 5(a), we begin with parameters X_2, X_3 and X_4, with X_2 defining length of the stem segment (in red), X_4 defining length of the leaf segment (in green), and X_3 defining the angle between stem and leaf segments. The phytomer skeleton in Fig. 5(a) is then converted to the skeleton in Fig. 5(b) by subdividing the leaf segment into smaller segments

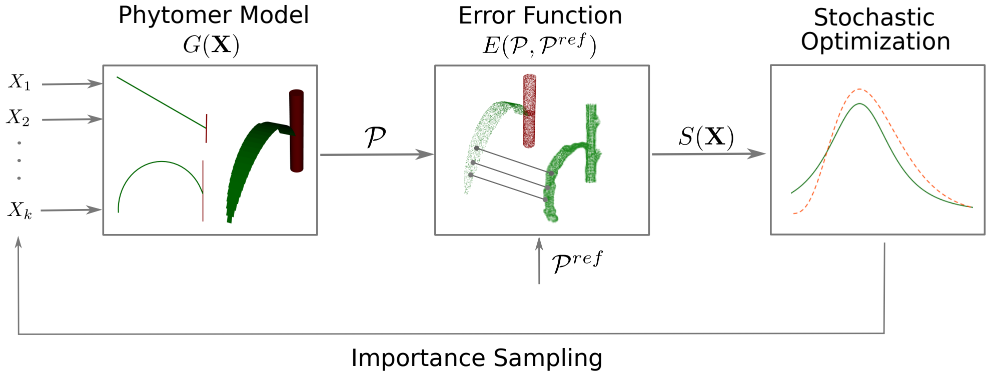


Fig. 4. Overall proposed approach for phenotyping using plant phytomers via model-based optimization. It begins by taking parameters X_1 to X_k as inputs and using the phytomer model function $G(\mathbf{X})$ to generate a phytomer $\mathcal{P}$ corresponding to the input parameters. $\mathcal{P}$ is then compared against the reference phytomer $\mathcal{P}^{ref}$ whose phenotypes we wish to estimate. The distance error between $\mathcal{P}$ and $\mathcal{P}^{ref}$ is minimized by the stochastic optimization for input parameters X_1 to X_k to best match generated phytomers to reference phytomer.

and rotating each subsegment uniformly by a value of X_6. The last 20% of the subsegments near the end of the leaf are rotated by a smaller value of $X_6/2$. This is done since due to gravity, naturally occurring leaves of bioenergy sorghum often have a parabolic shape with leaf curvatures changing less rapidly near the leaf tips.

Fig. 5(c) shows a 3D model obtained from the phytomer skeleton in Fig. 5(b). For generating the stem 3D model, the stem portion of the skeleton (in red) is expanded cylindrically with a diameter value of X_1. For generating the leaf 3D model, each subsegment i of the leaf midrib (in green) is expanded using a locally planar patch with a concavely decreasing width value. These planar patches are centered at the leaf midrib and are initially rectangular in shape but later become trapezoids as the width of the leaf starts to shrink. The planar patch width X_5^i as a function of the leaf midrib subsegment i can been expressed as

$$X_5^i = \begin{cases} w_{max} & 1 \le i \le i_{max} \\ w_{max} \times \frac{k+\log\left(1-r_i+e^{-k}\right)}{k+\log(1+e^{-k})} & i_{max} \le i \le n_{pieces} \end{cases}$$

where,

$$r_i = \frac{i - i_{max}}{n_{pieces} - i_{max}} \tag{2}$$

where, n_{pieces} represent the total number of leaf midrib subsegments and i_{max} represents the subsegment index where the leaf width has a maximum value of w_{max}. i_{max} is set to be half of n_{pieces} and n_{pieces} is set to a value of 30. The parameter k controls the degree of concavity, and is set to a value between 1 and 3. These leaf width design parameters were chosen so as to best resemble naturally occurring leaves of bioenergy sorghum that are typically long and thin with mostly constant widths that taper near the tips.

The resulting 3D model of the plant phytomer is illustrated in Fig. 5(c). Finally, 3D points are uniformly sampled from the surfaces of the phytomer model in Fig. 5(c) to generate a 3D point cloud representation shown in Fig. 5(d).

This process of generating the phytomer point cloud $\mathcal{P}$ from input model parameters $\mathbf{X}$ falls under the phytomer model function $G(\mathbf{X})$. As can be seen, $G(\mathbf{X})$ (and consequently the score function $S(\mathbf{X})$) does not have a closed form expression and thus cannot provide gradient or Hessian information in closed-form. It is also expensive to compute. This necessitates the use of stochastic optimization based techniques for maximizing the score function.

C. Error Function

The error function $E(\mathcal{P}, \mathcal{P}^{ref})$ computes a scalar value representing distance error between a generated point cloud $\mathcal{P}$ and the reference point cloud $\mathcal{P}^{ref}$. To compute this distance error, we first align $\mathcal{P}$ with $\mathcal{P}^{ref}$ using the iterative closest point algorithm (ICP) [13]. The ICP algorithm computes a 6 DOF transformation T which when applied to points in $\mathcal{P}$ aligns them to best match the points in $\mathcal{P}^{ref}$. The transformed point cloud $\mathcal{P}$ is expressed as $\mathcal{P}'$. This initial alignment using ICP before computing the distance error between generated and reference point clouds is important, otherwise the distance error would give arbitrary values depending on how $\mathcal{P}$ and $\mathcal{P}^{ref}$ are initially oriented with respect to each other.

We then use the *Hausdorff distance* to compute the distance error between two 3D point sets $\mathcal{P}'$ and $\mathcal{P}^{ref}$.

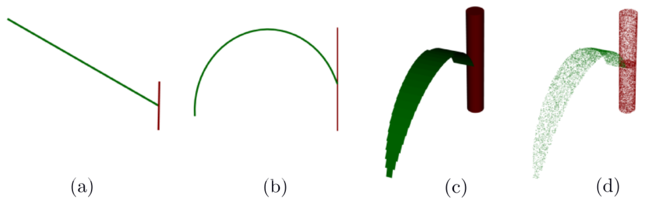

(a) (b) (c) (d)

Fig. 5. Different stages of the parametric phytomer model function $G(\mathbf{X})$. **(a)** shows initial skeleton generated using parameters X_2, X_3, X_4. **(b)** is derived from (a) by diving leaf segment (in green) into smaller subsegments and rotating each subsegment cumulatively by a value of X_6. **(c)** is the 3D model generated from the skeleton in (b), with stem expanded cylindrically by diameter X_1 and leaf expanded in locally planar patches with widths X_5^i. Finally, **(d)** is the 3D point cloud $\mathcal{P}$ generated by uniformly sampling points from surfaces of 3D phytomer model in (c).

Hausdorff distance is defined more generally as a distance metric capturing how far two subsets of a metric space are from each other. Thus, the Hausdorff distance can be used to determine the degree of resemblance between two point clouds $\mathcal{P}'$ and $\mathcal{P}^{ref}$ that are aligned with each other via the ICP algorithm.

For two non-empty subsets S_1 and S_2 of a metric space, the Hausdorff distance $d_H(S_1, S_2)$ is given as,

$$d_H(S_1,S_2) = \max\{\sup_{s_1\in S_1}\inf_{s_2\in S_2} d(s_1,s_2),\ \sup_{s_2\in S_2}\inf_{s_1\in S_1} d(s_1,s_2)\} \tag{3}$$

where, $\sup$ represents the supremum and $\inf$ the infimum.

In our case, these two subsets S_1 and S_2 are 3D point sets $\mathcal{P}'$ and $\mathcal{P}^{ref}$. The Hausdorff distance $d_H(\mathcal{P}', \mathcal{P}^{ref})$ between transformed point cloud $\mathcal{P}'$ and reference point cloud $\mathcal{P}^{ref}$ can now be expressed as,

$$d_H(\mathcal{P}',\mathcal{P}^{ref}) = \max\ \{\ h(\mathcal{P}',\mathcal{P}^{ref}),\ h(\mathcal{P}^{ref},\mathcal{P}')\ \}$$

where,

$$\begin{aligned} h(\mathcal{P}',\mathcal{P}^{ref}) &= \max_{p'\in\mathcal{P}'}\ \{\ \min_{p^{ref}\in\mathcal{P}^{ref}}\ \{\ d(p',p_{ref})\ \}\} \\ h(\mathcal{P}^{ref},\mathcal{P}') &= \max_{p^{ref}\in\mathcal{P}^{ref}}\ \{\ \min_{p'\in\mathcal{P}'}\{\ d(p_{ref},p')\ \}\} \end{aligned} \tag{4}$$

where, $p' \in \mathcal{P}'$ and $p^{ref} \in \mathcal{P}^{ref}$, and $d(p', p_{ref})$ is any distance metric between these points. We take $d(p', p_{ref})$ as the Euclidean distance between 3D points p' and p^{ref}.

In Eq. 4, distance $h(\mathcal{P}', \mathcal{P}^{ref})$ is defined from point set $\mathcal{P}'$ to $\mathcal{P}^{ref}$ while distance $h(\mathcal{P}^{ref}, \mathcal{P}')$ is defined from $\mathcal{P}^{ref}$ to $\mathcal{P}'$. It isn't necessary that these two directed distances are equal. The final Hausdorff distance $d_H(\mathcal{P}', \mathcal{P}^{ref})$ *between* $\mathcal{P}'$ and $\mathcal{P}^{ref}$ is hence computed as the maximum of the two directed distances. This is then converted to a score $S(\mathbf{X}) = -d_H(\mathcal{P}', \mathcal{P}^{ref})$ that is passed on to the optimization routine.

D. Stochastic Optimization

For stochastic optimization, we use the cross-entropy method (CE) method proposed by Rubinstein in [14] for estimating rare-event probabilities using an importance sampling procedure with *cross-entropy* or *Kullback-Leibler divergence* as measure of closeness between two sampling distributions. Subsequent work by Rubinstein in [15] showed that the same CE algorithm can also be used for optimization by formulating the optimization as a rare-event estimation problem. Botev et al. in [16] provide a tutorial-based introduction on use of cross-entropy method for optimization. Our goal is to maximize score function $S(\mathbf{X})$ so as to minimize distance error between point clouds $\mathcal{P}$ and $\mathcal{P}^{ref}$. This goal of finding maximum of $S(\mathbf{X})$ over a given set $\mathcal{X}$ can be expressed as,

$$S(\mathbf{X}^*) = \gamma^* = \max_{\mathbf{X}\in\mathcal{X}} S(\mathbf{X}) \tag{5}$$

We can now associate the optimization in Eq. 5 with the estimation of rare-event probability $l = \mathbb{P}(S(\mathbf{X}) \geq \gamma)$, where $\mathbf{X}$ has probability density function (pdf) $f(\cdot; \mathbf{u})$ on $\mathcal{X}$ and γ is close to unknown γ^*. The objective of cross-entropy optimization is now to find an importance sampling distribution that concentrates all its mass in a neighborhood of point $\mathbf{X}^*$. This leads to a *multi-level* CE algorithm where a sequence of reference parameters $\{\hat{\mathbf{v}}_t\}$ and levels $\{\hat{\gamma}_t\}$ are constructed with the goal that they converge to $\mathbf{v}^*$ and γ respectively. This can be implemented in an iterative manner as detailed in [16] and summarized in Algorithm 1.

Algorithm 1 Cross Entropy Algorithm for Optimization

```
1: Choose initial parameter vector $\hat{\mathbf{v}}_0$. Let $N^e = \lceil \rho N \rceil$. Set $t = 1$ (level counter).
2: while $t \leq t_{max}$ do
3:     Generate $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_N \sim f(\cdot; \hat{\mathbf{v}}_{t-1})$. Compute performances $S(\mathbf{X}_i)$ for all $i$, and order them from smallest to largest : $S_{(1)} \leq S_{(2)} \cdots \leq S_{(n)}$. Let $\hat{\gamma}_t$ be the sample $(1-\rho)$ quantile of performances, that is $\hat{\gamma}_t = S_{(N-N^e+1)}$
4:     Use the same sample $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_N$ and solve the stochastic program,
           $\max_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i) \geq \hat{\gamma}_t\}} \ln f(\mathbf{X}_i; \mathbf{v})$
       Denote the solution by $\hat{\mathbf{v}}_t$.
5:     Increment iteration counter as $t \leftarrow t + 1$.
```

In Algorithm 1, at each iteration t, we simulate N independent random variables $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_N$ from the currently estimated importance sampling density $f(\cdot, \hat{\mathbf{v}}_{t-1})$ and let $\hat{\gamma}_t$ be the $(1 - \rho)$ quantile of the performance values $S(\mathbf{X}_1), S(\mathbf{X}_2), \ldots S(\mathbf{X}_N)$ where ρ is a user specified parameter called the *rarity parameter*. We then update the value of $\hat{\mathbf{v}}_{t-1}$ to $\hat{\mathbf{v}}_t$, where $\hat{\mathbf{v}}_t$ is computed using cross-entropy minimization (or equivalently likelihood maximization) based on the *elite set* of random variables $N^e = \lceil \rho N \rceil$ for which $S(\mathbf{X}_i) \geq \hat{\gamma}_t$.

For our application, $\mathbf{X}$ is the k-dimensional random variable $\mathbf{X} = \begin{bmatrix} X_1 & X_2 & \ldots & X_k \end{bmatrix}^T$ representing the parameters used for generating the phytomer model and the set $\mathcal{X}$ constitutes the domain of these parameters. The class of probability distribution functions (pdf) $f(\cdot; \hat{\mathbf{v}}_{t-1})$ from which random variables $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_N$ are sampled is taken to be a multivariate normal distribution. For the k-dimensional random variable $\mathbf{X}$, this is expressed as,

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}) \tag{6}$$

where, $\boldsymbol{\mu}_{t-1}$ is the k-dim mean vector and $\boldsymbol{\Sigma}_{t-1}$ is the $k \times k$ covariance matrix expressed as,

$$\begin{aligned} \boldsymbol{\mu}_{t-1} &= E[\mathbf{X}] = \begin{bmatrix} E[X_1]\ ,\ E[X_2]\ ,\ \ldots\ ,\ E[X_k] \end{bmatrix}^T \\ \boldsymbol{\Sigma}_{t-1} &= E\left[(\mathbf{X} - \boldsymbol{\mu}_{t-1})(\mathbf{X} - \boldsymbol{\mu}_{t-1})^T\right] \\ &= [Cov[X_i, X_j]; 1 \leq i, j \leq k] \end{aligned} \tag{7}$$

As part of Step 3 of Algorithm 1 the samples $\mathbf{X_i}$ are ordered according to their corresponding score values $S(\mathbf{X_i})$ and top $(1-\rho)$ quantile of samples are chosen for computing pdf parameters for the next iteration. It turns out that for a multivariate normal distribution, the pdf parameters that minimize the stochastic program in Step 4 of Algorithm 1 is simply the sample mean $\boldsymbol{\mu}_t$ and sample variance $\boldsymbol{\Sigma}_t$ of the top $(1-\rho)$ quantile of samples having $S(\mathbf{X}_i) \geq \hat{\gamma}$. These newly computed pdf parameters, $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$, are the solutions $\hat{\mathbf{v}}_t$ in Algorithm 1. We take $\rho = 0.9$ for all our experiments.

Additionally, when applying cross-entropy optimization we observed that the distribution converges around a small set of samples too quickly. In order to prevent this early convergence, we employ a trick frequently used in particle filtering, that is to add some variance noise to the distribution at each iteration [17]. Hence for each iteration t, a matrix Z_t is added to the sample covariance matrix as $\boldsymbol{\Sigma}_t \leftarrow \boldsymbol{\Sigma}_t + Z_t$. When selecting the magnitude of this additional covariance noise Z_t, there is a trade-off between convergence and robustness. If there is more added noise, the model will converge more slowly but be more robust to ill-conditioned error surfaces. We take Z_t to be a diagonal matrix with each diagonal entry linearly decreasing with iterations t and initialized as $\{Z_0\}_{ii} = 0.5 \times \{\boldsymbol{\Sigma}_0\}_{ii}$, $i = 1 \dots k$.

V. Experimental Results

Implementation Details

We have implemented our approach in C++ making use of the Visualization Toolkit (VTK) [18] and Point Cloud Library (PCL) [19] libraries.

Runtime Performance

The runtime performance of our approach is primarily a function of the number of samples used in the cross-entropy optimization and the total number of 3D points in the phytomer point cloud. Table I shows average runtime values per optimization iteration for our implementation (parallelized across 8 threads) running on a Linux system with an Intel i7 2.9GHz CPU. We typically use 500 samples, ∼5000 3D points and 10 optimization iterations in all our results. For the present application we do not require the system to be real-time since the data processing is being done offline. However, since operations for each sample are independent, the proposed approach is highly parallelizable.

TABLE I

Average Runtime per Optimization Iteration (in seconds)

		Number of 3D Points		
		1000	**5000**	**10000**
Number of samples	**100**	0.67	3.61	7.98
	500	3.31	**18.8**	40.3
	1000	6.80	37.1	80.4

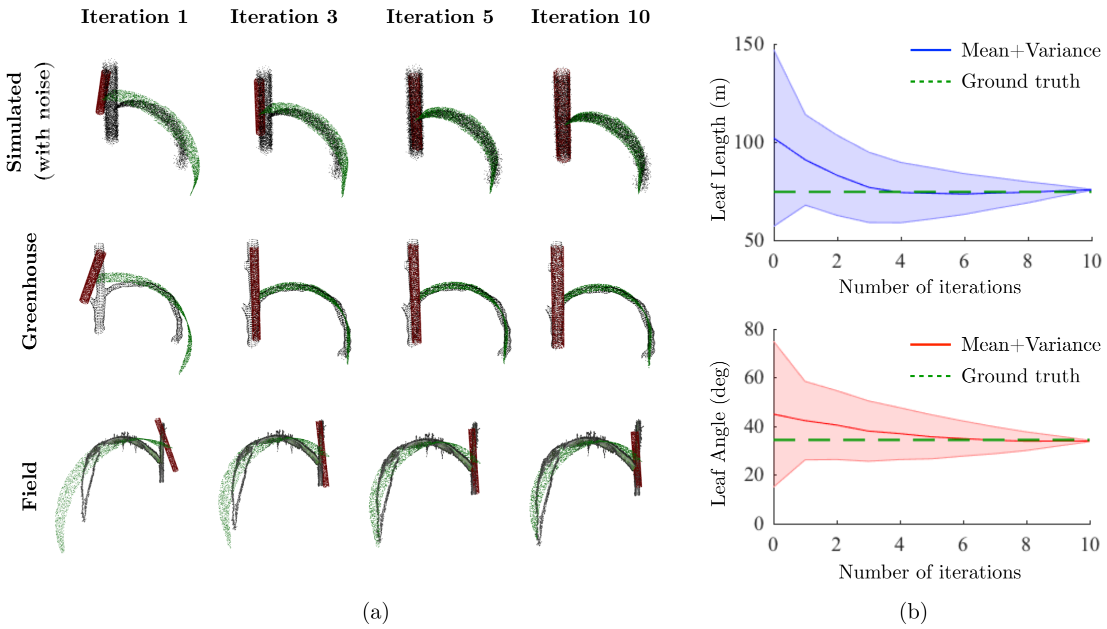
Optimization results each iteration

Fig. 6(a) shows qualitative results of the proposed approach on 3D plant phytomers in simulated, greenhouse and field environments. It visualizes the parametric model generated using mean parameter values of the elite samples for the start and end optimization iterations and for two iterations in between. It can be seen that the model parameters used in the optimization are expressive enough to capture varying plant anatomies across environments.

Fig. 6(b) shows corresponding elite sample mean and variance plots for leaf length, leaf angle estimation for the field plant phytomer visualized in Fig. 6(a). It can be seen that with each iteration of the optimization, estimated values approach actual values with decreasing variance with the leaf length estimate converging faster than the leaf angle estimate.

Robustness Analysis

To analyze robustness of the proposed approach, we take the field phytomer visualized in Fig. 6 and corrupt it with



(a) (b)

Fig. 6. (a) shows qualitative results for phytomers from simulated with noise (top), greenhouse (middle), and field (bottom) environments. Reference phytomer whose phenotypes are to be estimated is colored black and phytomers generated using mean parameter estimates of elite samples each iteration are colored red and green. (b) shows corresponding elite sample mean and variance plots for the field phytomer visualized in (a).

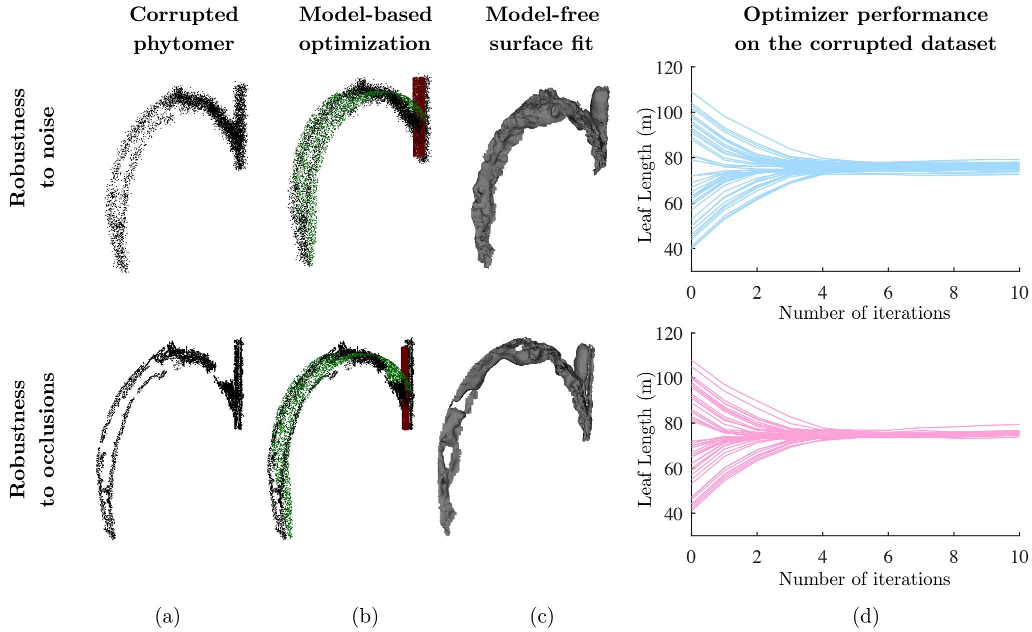


Fig. 7. (a) shows two phytomer point clouds taken from a dataset corrupted with noise and occlusions respectively. (b) shows corresponding model fits using the proposed model-based optimization approach (c) shows (degenerate) 3D surface reconstructions of (a) as used by baseline *model-free* methods. (d) shows quantitative results from proposed model-based optimization for leaf length estimation on both corrupted datasets containing 50 phytomers each.

noise and occlusions to create two augmented datasets of 50 phytomers each. In the absence of sufficient field data, this enables us to validate our approach on an larger, augmented dataset. For creating the first (noisy) dataset, we add Gaussian noise with zero mean and 0.5 cm standard deviation to each 3D point of the original phytomer point cloud. For creating the second (occluded) dataset, we randomly sample a sphere of diameter 6 cm in the original phytomer point cloud and remove all 3D points falling inside it.

Fig 7(a) shows two phytomer point clouds, one taken from the noisy and one taken from the occluded dataset respectively. Fig 7(b) shows corresponding mean model fits using the proposed model-based optimization approach. Fig 7(c) shows (degenerate) 3D surface reconstructions generated using Poisson surface reconstruction method [20] as used by *model-free* methods like [4], [5]. Such *model-free* methods working primarily on a surface mesh representation of the plant without leveraging prior information on the plant structure break down under such noise and occlusions since assumptions like connectivity and non-degeneracy of the reconstructed 3D surface mesh are violated in such cases.

Fig. 7(d) shows quantitative results for leaf length estimation for both datasets respectively. The optimizer starts at random initialization points for the leaf length parameter for all 50 phytomers in both datasets. It can be seen that the leaf length estimate converges to approximately a ±5 cm band (slightly smaller for dataset 2) around the ground truth leaf length value of 75 cm.

Estimation of phenotypes across varying plant anatomies

To evaluate phenotype estimation quantitatively across plant phytomers with varying anatomies, we run the algorithm on a simulated dataset of 50 simulated plant phytomer with added Gaussian noise and on a real-world dataset of 30 real world plant phytomers (20 greenhouse, 10 field).

The simulated phytomer dataset is generated using the parameteric model described in Section IV-B and varying values of X_1 (stem diameter) between 2 and 5 cm, X_2 (stem length) between 10 and 20 cm, X_3 (leaf angle) between 30^o and 60^o, X_4 (leaf length) between 40 and 70 cm, X_5 (leaf width) between 4 and 8 cm and X_6 (leaf rotate angle) between 3^o and 4^o. To each 3D point in the simulated point cloud, Gaussian noise with zero mean and 0.5cm standard deviation is added. The greenhouse and field phytomer datasets were collected using methods described in Sections III-A and III-B respectively. The ground truth phenotype values for greenhouse and field phytomers were determined manually using MeshLab [21].

We show results for estimation of leaf phenotypes like leaf length and leaf angle since those are typically hard for traditional 3D geometric methods to estimate. Additionally, ground truth for these two leaf phenotypes can be estimated with least ambiguity (unlike leaf width or leaf area). Table II tabulates correlation coefficients for leaf length and leaf angle estimates for simulated and real-world datasets. Fig. 8 additionally shows correlations plots for the real-world dataset consisting of 30 plant phytomers. For phenotyping applications, a metric like correlation coefficient is more relevant than absolute error values since the eventual goal is to study correlation of phenotypes with underlying genotypes.

From the correlation values it can be seen that leaf length estimation has a high positive correlation with $R > 0.95$ for both simulated and real-world datasets. Leaf angle estimation too is positively correlated but with lower values than leaf length estimation especially for the real-world dataset. From the optimization point of view, this happens because the

TABLE II

CORRELATION COEFFICIENT VALUES FOR PHENOTYPE ESTIMATION

	Simulated with noise (50 phytomers)	**Greenhouse+Field (30 phytomers)**
Leaf Length	0.99	0.96
Leaf Angle	0.97	0.82

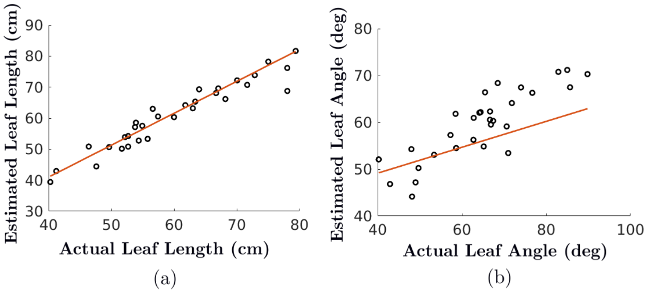


Fig. 8. Correlation plots for (a) leaf length estimation and (b) leaf angle estimation on the real-world dataset of 30 plant phytomers.

error landscape is flatter in the leaf angle axis than in the leaf length axis. A possible reason for that could be is that the ICP algorithm is able to align the leaf surface curvature (that affects leaf length directly) better than the stem/leaf intersection point (that affects leaf angle directly).

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a model-based approach for estimating phenotype values from 3D plant phytomers by reformulating the phenotyping problem as that of an optimization in the space of plant model parameters. We saw that, while the effectiveness of a model-based approach relies on the expressiveness of the underlying model used, it is a powerful paradigm that can provide reasonable estimates even under noise and occlusions. We evaluated the performance and robustness of our approach in estimating leaf phenotypes in both simulated and real-world environments.

Currently, we perform preprocessing on collected data so as to extract plant phytomers that are then used for phenotyping. As future work, one direction would be to extend the proposed approach to more complete plant anatomies consisting of multiple plant phytomers. One way to model that could be to define a L-system grammar for multiple phytomers and including these L-system grammar symbols as variables in the stochastic optimization. This would, however, come at the cost of increased search space due to added dimensionality. Another interesting direction for future work would be to use semantic class labels (like stem and leaf) for 3D points and add that as a label mismatch cost to the current optimization objective. This can help in weighting stem and leaf points differently ensuring more robust fits.

ACKNOWLEDGEMENTS

This work was supported through collaboration with Texas A&M University Agrilife Research, principal investigator William Rooney. We gratefully acknowledge advice and assistance of John Mullet, Alex Tomlinson, Sandra Truong and Ryan McCormmick. We would also like to thank Srinivasan Vijayarangan, Simon Du and Prathamesh Kini for their help with field data collection and processing.

REFERENCES

[1] Noah Fahlgren, Malia A Gehan, and Ivan Baxter. Lights, camera, action: high-throughput plant phenotyping is ready for a close-up. *Current opinion in plant biology*, 24:93–99, 2015.

[2] Srinivasan Vijayarangan, Paloma Sodhi, Prathamesh Kini, James Bourne, Simon Du, Hanqi Sun, Barnabas Poczos, Dimitrios Apostolopoulos, and David Wettergreen. High-throughput robotic phenotyping of energy sorghum crops. In *Field and Service Robotics*. Springer, 2017.

[3] Paloma Sodhi, Srinivasan Vijayarangan, and David Wettergreen. In-field segmentation and identification of plant structures using 3d imaging. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017.

[4] Paweł Drapikowski, Ewa Kazimierczak-Grygiel, Dominik Korecki, and Justyna Wiland-Szymańska. Verification of geometric model-based plant phenotyping methods for studies of xerophytic plants. *Sensors*, 16(7):924, 2016.

[5] Ryan F McCormick, Sandra K Truong, and John E Mullet. 3d sorghum reconstructions from depth images identify qtl regulating shoot architecture. *Plant Physiology*, pages pp–00948, 2016.

[6] Supawadee Chaivivatrakul, Lie Tang, Matthew N Dailey, and Akash D Nakarmi. Automatic morphological trait characterization for corn plants via 3d holographic reconstruction. *Computers and Electronics in Agriculture*, 109:109–123, 2014.

[7] Jonathan Binney and Gaurav S Sukhatme. 3d tree reconstruction from laser range data. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1321–1326. IEEE, 2009.

[8] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998–1005. Ieee, 2010.

[9] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

[10] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer Science & Business Media, 2012.

[11] Simon Fuhrmann, Fabian Langguth, and Michael Goesele. Mve-a multi-view reconstruction environment. In *GCH*, pages 11–18, 2014.

[12] Sara N Olson, Kimberley Ritter, William Rooney, Armen Kemanian, Bruce A McCarl, Yuquan Zhang, Susan Hall, Dan Packer, and John Mullet. High biomass yield energy sorghum: developing a genetic model for c4 grass bioenergy crops. *Biofuels, Bioproducts and Biorefining*, 6(6):640–655, 2012.

[13] Paul J Besl, Neil D McKay, et al. A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[14] Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

[15] Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

[16] Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, Pierre L'Ecuyer, et al. The cross-entropy method for optimization. *Machine Learning: Theory and Applications, V. Govindaraju and CR Rao, Eds, Chennai: Elsevier BV*, 31:35–59, 2013.

[17] István Szita and András Lörincz. Learning tetris using the noisy cross-entropy method. *Learning*, 18(12), 2006.

[18] Visualization Toolkit. `https://www.vtk.org/`.

[19] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[20] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013.

[21] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian Chapter Conference*, volume 2008, pages 129–136, 2008.