

VisND: A Visualization Tool for Multidimensional Model of Canopy

Ali Shafiekhani
ViGIR Lab
University of Missouri
Columbia, MO, USA

AShafiekhani@mail.missouri.edu

Felix B. Fritschi
Division of Plant Sciences
University of Missouri
Columbia, MO, USA

FritschiF@missouri.edu

Guilherme N. DeSouza
ViGIR Lab
University of Missouri
Columbia, MO, USA

DeSouzaG@missouri.edu

Abstract

Plant phenotyping is a data-driven research where interpretation of large and often multidimensional data is required. Therefore, effective visualization of plant phenotypes plays a major role in data analytics as it can provide scientists with the required tool to extract and infer important information. In that sense, unifying the large and multidimensional phenotypical data into one single model of the canopy can help plant biologists to correlate information from different dimensions and derive new observations and understandings in plant sciences.

In this paper, we proposed a spatio-temporal tool for high-dimensional modeling and visualization of canopy for plant phenotyping. The goal is to offer an open-source visualization tool, named VisND (for N-Dimensional), that will provide a Graphical User Interface (GUI) where plant scientists can easily extract and analyze multidimensional models, registered over time, from different sensors and viewpoints. For this paper, we created 5D models (3D-RGB and Temperature over Time) of a crop by fusing and registering data captured using our field-based phenotyping platform: Vinocular, a trinocular, multi-spectrum, observation tower. The platform is part of a study on the behavior of plants in response to different biotic and/or abiotic stresses. The data was captured using Infrared Thermography (IRT) along with multiview, visible imaging technology over an entire planting season and on a 24/7 basis. While currently VisND is being demonstrated for 5D models, it can be easily extended to incorporate other modality of sensors (more dimensions) and from other sources, such as other robotic platforms also operating in the field – e.g. our mobile robot, Vinobot.

1. Introduction and Related Work

Over past many years, imaging technologies have evolved from simple devices that could capture visible spectrum (RGB), to devices capable of imaging single non-

visible spectrum (e.g. x-rays, ultraviolet, infrared, etc), to multi spectrum sensing, where cameras can compose multi spectral layers simultaneously. Similarly, 3D reconstruction using passive (multiple RGB images) or active sensing (time-of-flight, structured lighting, etc.) has found tremendous advances due to algorithms broadly known as structure from motion (SFM), simultaneous localization and mapping (SLAM), etc. [7, 1]. These algorithms can now produce very dense cloud of 3D points representing both structure and texture of objects, which can then be fused with other non-visible spectra to further enrich the model. In [11] for example, researchers created 4D models of building interiors by fusing RGB-D and thermal images. In our previous work [9], stereo and thermal images were also fused to create 4D-RGB models of fields, which were then used to characterize plant response to different heat conditions through monitoring their growth rate [10].

Besides the sensing type, time is another dimension that can be used to enrich a 3D model, creating a spatio-temporal model to capture changes in structure over time. In that regard, Dong et al. [3] created a spatio-temporal 4D model of crops to monitor growth in agricultural field, obtaining high spatial and temporal resolution, which is essential in modern precision agriculture. This 4D model consisted of a series of 3D models obtained over time and registered with respect to a single global coordinate frame. This development can have a great impact in many areas, as more dimensions (spectra) can be added into the model, potentially allowing the inference of new information due to this combined, richer representation. However, adding more dimensions to a model can be overwhelming for humans to process, making it instead harder to analyze and actually use the extra information. In that sense, visualization of high-dimensional models plays a key role in understanding available information stored in the model as it can help us correlate or contrast data from various dimensions. Specifically, visualization of time-varying models can reveal new information about growth rate as well as patterns of changes. In [8] for example, Schindler and Dellaert proposed a 4D vi-

sualization tool with which one can interact and analyze a vast collection of historical photographs captured over time from the same urban area. Their system can tie together large sets of historical photographs of a given city, allowing, for example, historians to interpret and correlate events that happened in this 4D model of the city.

In this paper we introduce an open-source N-dimensional visualization tool (VisND)¹ that can combine a series of 4D-RGB point clouds (as proposed in [9]) over time. Despite the fact that we applied the proposed tool only to sequences of 4D-RGB models (i.e. 3D-RGB plus temperature), we hope this will demonstrate its potential to be enhanced for any hyper-spectral images. Also, by making it open source, we expect it to be applied in many other areas, beside the target area here: plant phenotyping in the field. In fact, to the best of our knowledge, this is the first open source tool for visualizing ND models, which we expect to be a second significant contributions of this work. A third and more specific contribution for the limited scope of this paper is that VisND can fill the gap between computer scientists and plant biologists by providing a graphical user interface (GUI) with which experts can more easily analyze data, allowing them to employ and augment their knowledge about plant behavior in the field.

The rest of this paper is organized as follows: in section 2, we overview our approach to build 5D-RGB models of the canopy; in section 3, we introduce the software features of VisND; we explain how they can help analyzing 5D-RGB models; and how the user interacts with VisND; finally, in section 4, we conclude and discuss future improvement to VisND.

2. Proposed System for 5D-RGB Modeling of Canopies

2.1. Platform

In this paper, we relied on data captured by our high-throughput, field-based, plant phenotyping platform, Vinocular [10]: an observation tower that oversees an entire field by rotating 360° a group of sensors mounted on a horizontal bar. Those sensors include a stereo RGB setup, with two 12 M pixel Grasshopper3 cameras, and a thermal Flir A625 camera centered between the other two RGB cameras. The system was deployed over the entire planting season to create multiple 4D-RGB models ([9]) over time. The tower used two solar panels to recharge two 12V lead-acid batteries that can keep the system operational for two days under absolute no sunlight. The entire system was weatherized to tolerate typical summer conditions: i.e. temperatures well above 100°F (37.8°C), heavy rains, and 50 mph winds, all of which were recorded by local weather stations. Figure 1



Figure 1: Observation tower with stereo RGB rig and a thermal camera centered in between

shows Vinocular at its lowest height while deployed in the middle of a corn field.

2.2. 5D Reconstruction Pipeline

In order to create 5D-RGB models of the field, a sequence of processing steps is required. This pipeline started with the the calibration of the trinocular cameras, as well as their *head-eye* calibrations. Next, the pipeline employed a 3D reconstruction algorithm, which included a constrained bundle adjustment and dense reconstruction, to combine each set of data collected during one 360° revolution of the tower. Next, it added the 4th dimension (temperature) by back-projecting IR images onto the densely reconstructed 3D model of the field. All the steps so far, i.e. from trinocular and head-eye calibrations, to the creation of the 4D-RGB models, have already been detailed in [9] and are being mentioned here just for context. The pipeline then optimized all sets reconstructed at individual time stamp by registering the same 4D-RGB models with respect to a single, global coordinate frame defined by the first set.

2.2.1 Pre-processing

All camera calibration process could be performed on-site using a large calibration pattern (1.5 m × 1.5 m). This calibration process consists of finding the mathematical model relating the image pixel coordinates from all cameras to their corresponding points in 3D space.

For this work, RGB and thermal cameras were calibrated using a multi-material calibration pattern that employed black-painted aluminum squares on top of white paper, on a wooden frame. This combination of heat-absorbent vs. heat-reflective, as well as high-contrast (black vs. white) materials created enough gradient for detecting corners in both thermal and RGB images. A multi-objective optimization of stereo pairs was also carried out between the left-

¹<https://github.com/AliShafiekhani/VisND.git>

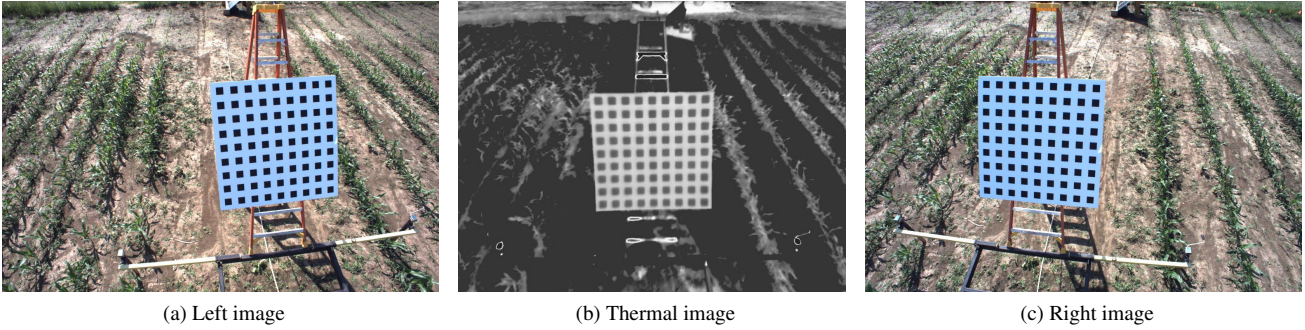


Figure 2: Three typical images of the multi-material pattern used to calibrate Vinoculer’s trinocular cameras: (a) Left RGB; (b) Thermal camera; (c) Right RGB

right (RGB) cameras and left-thermal (RGB-IR) cameras to accurately estimate the position and orientation of the cameras with respect to each other. Figure 2 shows three of the calibration images captured during this process.

In order to find a single world coordinate frame aligned with the observation tower, a head-eye calibration was also carried out and the pose of the left camera with respect to the center of rotation was established. This type of calibration is referred to as the $AX = XB$ problem, where X is the unknown pose of the camera frame with respect to the world, and A and B are known reference frames. In this paper, we used the iterative method presented in [2] to solve for the pose of the left camera, assuming a world coordinate frame at the center of the turn table.

It is important to mention here that the head-eye calibration is a very important step in the pipeline for it can affect the entire 5D reconstruction through many of the steps of the pipeline: 3D reconstruction, back-projection of thermal data onto the 3D models, and temporal cloud registration. It is also quite useful in constraining the bundle adjustment and dense reconstruction steps by reducing the complexity of the optimization from $6 \times N$ to $6 + N$ unknown parameters – again, for more details, refer to [9].

2.2.2 3D Reconstruction

In order to perform the 3D reconstruction of the canopy, the following steps were followed:

1. Feature extraction and three-view matching between two consecutive stereo frames: For this step, SIFT feature descriptors [6] were employed to find corresponding points between L_i and R_i , as well as L_i and L_{i+1} images – where, L and R represent, respectively, left and right camera frames or their images, and i represents the i^{th} instant (or angle of the turning table) when the images were taken. See Figure 3(a) for depiction of the camera poses at various instants i ;

2. Triangulation of stereo matches to establish 3D points: After undistorting the matched features (keypoints), a sparse set of 3D points were computed from the 2D keypoints on both left and right images. These 3D coordinates were calculated with respect to the left camera coordinate frame at instant i ;
3. Incremental camera pose estimation: Using the *Perspective from n Points* (PnP) algorithm, the relative poses of the left camera frame at instant $i + 1$, i.e. L_{i+1} , with respect to the left camera frame L_i were found by minimizing the error in back-projecting the 3D points at time i onto the 2D image L_{i+1} . This process is iterated for i , creating a chain of homogenous transformation: ${}^w H_{L_{i+1}} = {}^w H_{L_i} \times {}^{L_i} H_{L_{i+1}}$, where the first transformation in the chain is obtained from the head-eye calibration as the world being the center of the turn table, i.e. ${}^w H_{L_0} = {}^C H_{L_0}$;
4. Constrained bundle adjustment to refine camera poses: As the step above iterates, the error in camera pose estimation propagates and potentially grows from i to $i + 1$. Therefore, the chain of camera poses needed to be refined by using a motion-only Constrained Bundle Adjustment (CBA). As mentioned earlier, this constraint – derived from the fact that the cameras rotate about a fixed z axis – reduces the number of unknown parameters for optimization from $6 \times N$ to $6 + N$, where N is the number of left-camera poses (angles) and 6 is the number of unknowns in rotation and translation. That is, the $6 + N$ parameters are: 1) the N relative angles θ_i^{i+1} of the turn table about its vertical z axis, where each θ_i^{i+1} could be used to define the relative poses of the left camera with respect to the center of the turn table ${}^C H_{L_i}$; and 2) the 6 rotational and translational components in ${}^w H_{L_0}$, the refined head-eye calibration. After the refinement of the poses of all left camera frames, the poses of the right and thermal

camera frames were found using the extrinsic parameters from the calibration step (${}^L H_R$ and ${}^L H_T$, respectively). The CBA not only led to faster convergence, but it also reduced the residual error and hence, improving the next step, the dense reconstruction;

5. Dense 3D reconstruction using refined camera poses: Using the multi-view dense reconstruction algorithm (PMVS2) developed by Furukawa and Ponce [4] and the refined left and right camera poses from CBA, a dense 3D point cloud was created. The patches constructed by the PMVS2 algorithm were also useful in finding candidates for the thermal images from which corresponding 3D points could be observed.

2.2.3 4D Reconstruction (Temperature)

In order to find temperature information for 3D points, the software pipeline back-projected the 3D points onto the thermal images from which those points were observed. As mentioned above, the candidates for those thermal images were determined using the PMVS2 patch information. The back-projection was done using the intrinsic parameters of the thermal camera obtained during calibration and the angles θ_i^{i+1} above, which fully determined the transformation between the left coordinate frame at instant i and the world coordinate frame, and hence also the thermal coordinate frame and the world. In other words:

$${}^{T_i} H_w = {}^T H_L \times {}^{L_i} H_w \quad (1)$$

$$\begin{pmatrix} u \\ v \\ s \end{pmatrix} = K_T \times {}^{T_i} H_w \times \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

where ${}^{T_i} H_w$ (${}^{L_i} H_w$) is the transformation from the world coordinate frame to the i^{th} thermal (left) coordinate frame; while ${}^T H_L$ and K_T are the extrinsic and intrinsic parameters of the thermal camera from the calibration step. Finally, to minimize the effect of potentially inaccurate 3D-to-2D back-projections, the algorithm took the median of all valid temperature values corresponding to the 3D points. Later, the temperature information for the point cloud was used as input for the Visualization tool (VisND).

2.2.4 5D Reconstruction (Time)

Our ultimate goal is to create a tool to allow visualization and analysis of 5D-RGB models, that is, of 4D-RGB models over time. Similarly to registering 2D images to obtain 3D models, and then 3D point clouds to obtain the 4D-RGB models, in order for us to register 4D-RGB over time, corresponding keypoints among 4D models need to be found,

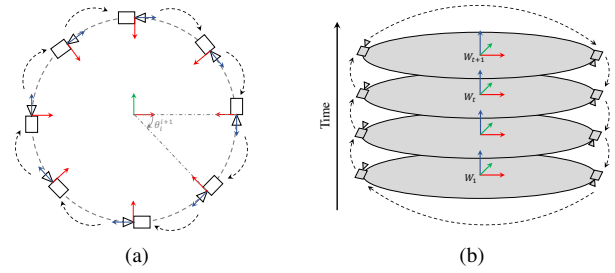


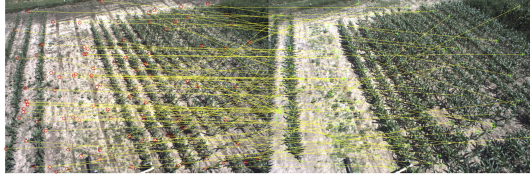
Figure 3: Depiction of the (a) spatial and (b) temporal constraints used for the creating of the 5D-RGB models

selected and then registered with respect to the world coordinate frame. Manual registration of 4D point clouds can be tedious and time consuming for large numbers of points in the cloud. Alternatively, one could use the camera poses already determined with respect to the world coordinate frame to devise an automated registration. However, due to changes in the environment over days or even hours, finding correspondences in 4D-RGB point clouds can be very hard and error prone. Therefore, relying on incrementally updating camera poses using the relative transformations between two consecutive time frames (similar to what was done before for ${}^{L_i} H_{L_{i+1}}$, but now with i representing time) could lead to even greater problems in error propagation due to weak correspondences. Earlier in the spacial case, strong correspondences were possible due to the large overlaps between consecutive frames and hence a loop constraint, as in Figure 3(a), could be used.

Since a similar loop constraint is only possible for the temporal case if we can find strong correspondences, we limited the search for keypoints to ground pixels over periods of time no longer than 5 days (Figure 3(b)). Therefore, in order to find ground points, the pipeline used a color-based ground detection algorithm ([5]) to segment out ground pixels, and then used a mask to detect keypoint features. Figures 4 illustrate this idea where SIFT features could be reliably matched among two temporal frames, 5 days apart, if only ground segmented pixels are used. That is, initially, in Figure 4(a), we observe the mismatches due to changes in the canopy, while much better results were obtained by the use of the segmentation mask in Figure 4(b) leading to a strong feature correspondence shown in Figure 4(c) after imposing that same segmentation mask.

The optimization steps for the parameters of the temporal loop were very similar to the ones explained in section 2.2.2, parts 3 and 4. The differences were:

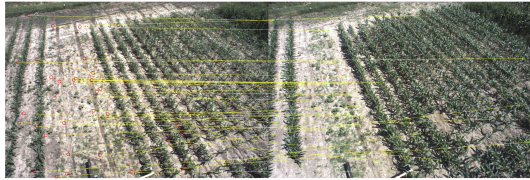
1. Before, the loop constraint (Figure 3(a)) was given by a chain of N matrices ${}^{L_i} H_{L_{i+1}}$ defined by two sets of unknowns: the N times θ_i^{i+1} relative angles of rotations of the turn-table – which uniquely defined ${}^C H_{L_i}$



(a) Feature Matching without using ground segmentation mask



(b) Ground segmentation mask



(c) Feature matching using ground segmentation mask

Figure 4: Temporal feature matching between images taken 5 days apart

– and the 6 rotational and translational components in the head-eye calibration ${}^w H_{L_0}$. However, for the temporal case, the loop constraint (Figure 3(b)) was given by T matrices ${}^{L_t} H_{L_{t+1}}$ defined by the unknown matrices ${}^{W_t} H_{W_{t+1}}$ and the constant matrices (obtained from the spacial optimization phase) ${}^{W_t} H_{L_j}$. Here, t is the time index for the 4D-RGB models, and $j = \{1, 2\}$ is one of the two cameras empirically chosen to form the loop.

2. Instead of using a constrained bundled adjustment, the pipeline step for the temporal case must run a full motion-only bundle adjustment, since now a full set of rotational and translational parameters are required to determine ${}^{W_t} H_{W_{t+1}}$, leading to a total number of parameters to be optimized equal to $(6 \times (T - 1))$.

The resulting ${}^{W_t} H_{W_{t+1}}$ matrices were used to register the 4D-RGB models with respect to one single coordinate frame – the first one in time. Alternatively, these transformations could be calculated by another method and provided to VisND as input.

Figure 5 shows qualitative results for this temporal registration step of the pipeline. As it can be inferred from the figure, there was a great improvement in registration of

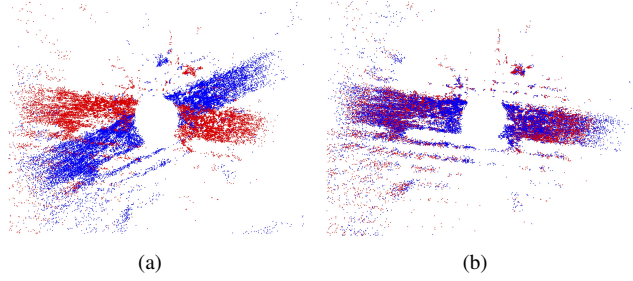


Figure 5: Temporal registration of two sparse point clouds created using images taken with 5 days time period gap (a) before optimization (b) after optimization

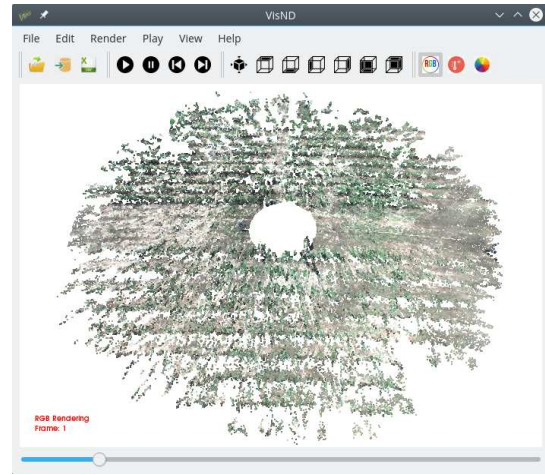


Figure 6: VisND Graphical User Interface (GUI)

two sets of 4D-RGB models before 5(a) and after 5 (b) this optimization.

3. VisND

VisND is a tool that enables user to visualize spatio-temporal models with additional dimensions e.g. temperature. Currently, the inputs for the software are frames of 3D point cloud, frames of temperature information, and optional transformation of frames in a world coordinate frame. VisND allows the user to play sequences of 4D-RGB models as a video, while providing tools for manipulating the same models; rendering different dimensions (currently color texture and temperature); measuring height and temperature of plants; and exporting that same meta data. Figure 6 shows an snapshot of the VisND graphical user interface (GUI). VisND was written in C++ using Qt, PCL and OpenCV libraries for the Linux Operating System. However, it can be easily ported to Windows or Mac OS since all libraries are cross-platform.

The main features of VisND are as follows:

- **Data model input:** The user can import all 4D-RGB models together or add them individually. Each 4D model is composed of a 3D point cloud file (in .ply or .pcd formats), and a text file containing information on the higher dimension data – currently, temperature, but in the future other hyper-spectral data;
- **3D manipulation:** Like any 3D software viewer, VisND provides tools to render the models from different viewing angles, zooming in and out, etc. It can also crop the model by setting planar thresholds. The planes are defined by clicking on any set of 3 arbitrary points and then changing its position and orientation (Figure 7a). This tool is useful to analyze plants and soils separately (Figure 7b);
- **Selective model resolution:** Optionally, the software can be configured to reduce the density of the point clouds. This parameter uses a voxel grid filter to reduce the cloud resolution. This can improve the rendering of large datasets with multiple models each carrying a large cloud of points;
- **Animation of sequences of 4D-RGB models:** This enables the user to follow temporal changes of individual plants or the entire crop. Currently, these animations can be either in *3D structure + RGB* (Figure 7c) or *3D structure + temperature* (Figure 7d);
- **Choice of color mapping:** Different color maps can be used for the visualization of higher dimensional data (again, currently temperature, but in future hyper-spectral data). This allows, for example, the user to scale the display of temperatures according to the gradients in specific areas of the model. The current options of color maps are *Autumn*, *Bone*, *Jet*, *Winter*, *Rainbow*, *Ocean*, *Summer*, *Spring*, *Cool*, *HSV*, *Pink*, and *Hot* (Figure 7d);
- **Manual registration of 4D-RGB clouds:** This feature allows the user to manually click on 3D points and establish their correspondences among different 4D-RGB models. This manual selection can also be used to skip the temporal registration step of the pipeline as one can provide previously defined transformations as input;
- **Extraction of phenotypes:** Heights and temperatures of plants, leaves, etc. can be extracted by clicking on various points of the model. In the future, other phenotypes will be automatically extracted, such as LAI, leaf area, leaf angle, etc... These phenotypes can be exported into .csv files for future use;

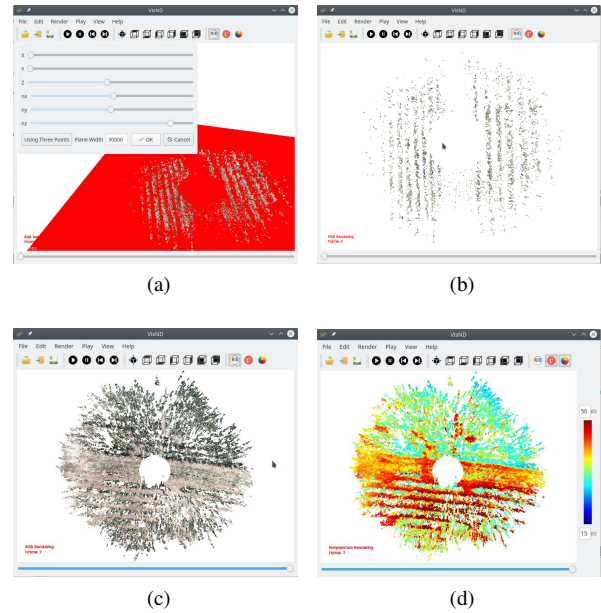


Figure 7: Snapshots of VisND demonstrating its current feature: (a) Manual ground detection (b) Plant segmentation by removing ground plane (c) Appearance rendering (d) Temperature rendering using Jet color mapping

4. Conclusions and Future work

Plant phenotyping can be greatly improved by data captured with different types of sensors. While adding multiple dimensions to the data can help plant phenotyping, analyzing these data can become overwhelming, error-prone and time consuming for humans. Therefore, fusing, cropping, detailing and expanding the data help plant scientists to analyze, correlate, and infer new information from the data. Effective visualization techniques applied to these high dimensional models can provide the required tool for better research.

In this paper we created a 5D-RGB model (ie. 3D-RGB plus temperature over time) for crop analysis and introduced a multidimensional visualization tool (VisND) that can render a series of 4D-RGB point clouds according to the user's choices of viewing points, segmentation, zooming, color mapping, etc. The software is released as open source, which will hopefully allow for a community-based development and further improvements. Also, for being open source, we expect to motivate developers to apply VisND to other areas of research, making it more generic with respect to other data types, expanding it to more dimensions (hyper-spectral images), etc. Finally, the VisND aims to fill the gap between computer programmers and plant biologists by providing a graphical user interface (GUI) with which they can easily analyze large fused data.

5. Acknowledgments

This material is based upon work supported by the National Science Foundation under Award Number IIA-1355406 and IIA-1430427.

References

- [1] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017. [1](#)
- [2] Guilherme Nelson DeSouza, Andrew H Jones, and Avinash C Kak. An world-independent approach for the calibration of mobile robotics active stereo heads. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3336–3341. IEEE, 2002. [3](#)
- [3] Jing Dong, John Gary Burnham, Byron Boots, Glen Rains, and Frank Dellaert. 4d crop monitoring: Spatio-temporal reconstruction for agriculture. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3878–3885. IEEE, 2017. [1](#)
- [4] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. [4](#)
- [5] Jianguai Liu and Elizabeth Pattey. Retrieval of leaf area index from top-of-canopy digital photography over agricultural crops. *Agricultural and Forest Meteorology*, 150(11):1485–1490, 2010. [4](#)
- [6] David G Lowe. Object recognition from local scale-invariant features. In *IEEE international conference on Computer vision*, volume 2, pages 1150–1157. IEEE, 1999. [3](#)
- [7] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. [1](#)
- [8] Grant Schindler and Frank Dellaert. 4d cities: Analyzing, visualizing, and interacting with historical urban photo collections. *Journal of Multimedia*, 7(2):125, 2012. [1](#)
- [9] Ali Shafiekhani, Felix B Fritschi, and Guilherme N DeSouza. A new 4d-rgb mapping technique for field-based high-throughput phenotyping. In *Computer Vision Problems in Plant Phenotyping*, pages 1–13. BMVC, 2018. [1](#), [2](#), [3](#)
- [10] Ali Shafiekhani, Suhas Kadam, Felix B Fritschi, and Guilherme N DeSouza. Vinobot and vinoculer: Two robotic platforms for high-throughput field phenotyping. *Sensors*, 17(1):214, 2017. [1](#), [2](#)
- [11] Stephen Vidas, Peyman Moghadam, and Michael Bosse. 3d thermal mapping of building interiors using an rgb-d and thermal camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2311–2318. IEEE, 2013. [1](#)