# SLAM in the Field: An Evaluation of Monocular Mapping and Localization on Challenging Dynamic Agricultural Environment

Fangwen Shu Paul Lesur Yaxu Xie Alain Pagani Didier Stricker
DFKI - German Research Center for Artificial Intelligence
{first_name}.{last_name}@dfki.de

## Abstract

*This paper demonstrates a system capable of combining a sparse, indirect, monocular visual SLAM, with both offline and real-time Multi-View Stereo (MVS) reconstruction algorithms. This combination overcomes many obstacles encountered by autonomous vehicles or robots employed in agricultural environments, such as overly repetitive patterns, need for very detailed reconstructions, and abrupt movements caused by uneven roads. Furthermore, the use of a monocular SLAM makes our system much easier to integrate with an existing device, as we do not rely on a LiDAR (which is expensive and power consuming), or stereo camera (whose calibration is sensitive to external perturbation e.g. camera being displaced). To the best of our knowledge, this paper presents the first evaluation results for monocular SLAM, and our work further explores unsupervised depth estimation on this specific application scenario by simulating RGB-D SLAM to tackle the scale ambiguity, and shows our approach produces reconstructions that are helpful to various agricultural tasks. Moreover, we highlight that our experiments provide meaningful insight to improve monocular SLAM systems under agricultural settings.*

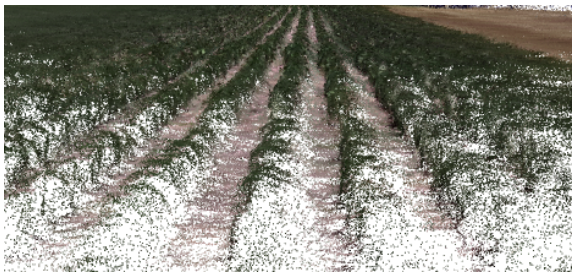


Figure 1. **The geo-referenced dense point cloud (map)** of soybean field reconstructed from Rosario dataset [45], sequence 04.

## 1. Introduction

Agricultural robotics [14, 15, 43, 62] have to function in environments that can be considered adversarial for most SLAM algorithms: abrupt movements, variable illumination, repetitive patterns, and non-rigidness of the environment are all encountered when performing tasks such as harvesting, seeding, agrochemical dispersal, supervision and mapping. Furthermore, while consequent resources have been spent on improving sensor-fusion for SLAM (with e.g. IMU or LiDAR) over the past decades, such systems suffer from sophisticated calibration, added weight, and additional required computational power. Those points negatively impact the price, power consumption, and algorithm complexity of the robots, all of which are of major importance to manufacturers as well as users. As such, it is desirable to keep the robot equipped with as few sensors as possible for the given task.

To solve those practical issues, we decided to combine a sparse, feature-based monocular SLAM with both offline and real-time MVS reconstruction algorithm. We show that the SLAM system employed in this work is more reliable for tracking than existing dense SLAM methods, while both reconstruction algorithms' outputs are dense enough for the tasks at hand. We propose the following contributions:

- A usable and efficient dense reconstruction architecture for agricultural mapping and localization with only a single camera.
- Exhaustive experiments of indirect visual SLAM systems evaluated on recently released public datasets [13, 45] aimed at agricultural localization and mapping. Compared to the relative work [16, 45] which only evaluated stereo setting, we provide the first baseline for monocular SLAM, and improved results for stereo visual SLAM.
- Ablation study of CNN-based self-supervised monocular depth estimation on aforementioned agricultural dataset. The estimated depth is used to simulate RGB-D SLAM with monocular RGB image sequence, which is validated by experimental analysis and competitive results are presented in this work compared to the raw stereo setting.

## 2. Related work

**Agricultural Robotics** Recent surveys on agricultural robotics [14, 15, 43, 62] present applications, challenges, and show a growing interest in SLAM system integration. For example, SLAM is a proper solution for occluded GPS (sometimes blocked by dense foliage) [14], crop-relative guidance in open fields, tree-relative guidance in orchards, and more importantly, sensing the crops and its environment [62]. Various sensors, such as on-board cameras and laser scanners, have been used for extracting features from the crops themselves and use them to localize the robot relative to the crop lines or tree rows in order to auto-steer.

However, here we focus on related work to monocular vision-based problem instead of discussing the general problem of sensor-fusion, where only a single camera is employed and it is a under-explored problem in agricultural scenario.

**Dataset** There is a large body of recent and ongoing research regarding SLAM and Visual Odometry for both indoor scenes [6, 17, 28, 58, 52] and outdoor urban scenes [8, 24, 37, 38], just to name a few. We do not consider datasets such as [4, 18, 29, 54] as they are extremely specialized for particular tasks like weed/crop classification which are not relevant to this work. To the best of our knowledge, only two datasets aimed at localization and mapping under agricultural environments are available: the Sugar Beets dataset [13] and the Rosario dataset [45]. The former presents a large-scale agricultural robot dataset including downward looking images, captured by a multi-spectral camera and an RGB-D sensor, and we found out it is difficult to track using monocular visual SLAM (downwards looking frames do not cover enough space, and even successive ones have small overlapping regions). The latter consists of 6 sequences recorded in a soybean field, captured by forward looking stereo camera, showing real and challenging cases such as highly repetitive scenes, reflection and burned images caused by direct sunlight and rough terrain, among others.

**Advanced Mapping** As one of the fundamental tasks of mobile robotics, an early work [50] presented the benefits of building a map of a vehicle's surroundings for precision agriculture. More recently, [19] has demonstrated a multi-sensor SLAM method for 4D crop monitoring and reconstruction. Another application similar to agricultural mapping is urban mobile mapping system (MMS) presented by [5, 10, 11]. There, the choice of dense reconstruction algorithm was more restricted. Commercial software like Pix4D [3] and Agisoft Metashape [1] provide sophisticate mapping pipelines with the support of Ground Control Points (GCPs) for indirect geo-referencing and quality control, and [12] presented good results of direct geo-referencing by providing camera poses by GPS. However, it is difficult to implement those algorithms on close-range agricultural mapping when image alignment is very challenging due to repetitive texture. There also exists highly accurate open source methods like COLMAP [55] and VisualSFM [63], however those frameworks only work offline, and can take up to a few hours to process the data.

When considering real-time dense mapping, it is natural to first try out direct/semi-direct SLAM, in which raw pixels are used for processing, instead of extracting and then matching features using descriptors. They make it possible to directly reconstruct dense maps as they do not rely on keypoints, but use the entire available data. This approach has received much attention in the past few years, first with DTAM [42], then with other noteworthy works such as LSD-SLAM [21], SVO [22] and DSO [20]. However, experiments have shown that the aforementioned direct methods have problem initializing at all on the agricultural image sequences (e.g. Rosario dataset). And the lack of datasets makes the comparison of system performance and robustness in agricultural settings difficult, as we highlight later in this work.

This led us to work with the recently released framework OpenVSLAM [59] which was built upon ORB-SLAM2 [40] without specific change on the core algorithms but provides a new framework with high usability and extensibility. After some careful modifications w.r.t agricultural scenarios (which we explain in Section 3), we are able to initialize and track reliably on challenging agricultural image sequences. Then, initialized by the pose graph generated from SLAM, we adapted COLMAP as an offline dense reconstruction solution, and we ended up working with REMODE [46] for real-time dense reconstruction, which was designed as a standalone, monocular reconstruction module running in parallel with another VO (Visual Odometry) module.

**Simulating RGB-D Sensor** Unsupervised learning of depth from unlabelled monocular videos [9, 25, 26, 32, 66] has recently drawn attention as it has notable advantages than the supervised ones and is also the core problem in SLAM [27]. Loosely inspired by the work of CNN-SLAM [60] and others [34, 35, 64, 65], we integrate Monodepth2 [26] as an additional depth predictor to tackle the scale problem of monocular SLAM and the demand of estimating dense depth map during tracking. Such choice is based on the fact that there is no ground truth depth available from dataset Rosario, therefore we have to generate ground truth from stereo image pair using method of SGBM [31] but only train the depth predictor self-supervised. The predicted depth will be used with monocular image sequence and simulate RGB-D camera which usually has problem working in such outdoor environments.

In this work, we base our experiments on the Rosario dataset [45] as it is appropriate to evaluate monocular SLAM systems. Notice that there is no other relevant work [16, 45] presents any result of monocular SLAM but only results from stereo SLAM, and our experiment shows reasonably good results on some of the sequence and no tracking lost on all the sequences of Rosario in general. As discussed before, the Sugar Beets dataset [13] is discarded due to its irrelevance for dense reconstruction and incompatibility with monocular SLAM.

## 3. Implementation Details

First, a monocular feature-based tracking system is used to compute the poses of the camera and acts as the frontend. Then, this information, alongside the original frames, is passed to the backend: a Multi-View Stereo (MVS) reconstruction pipeline that generates a dense point cloud of the agricultural scene, which works either in real-time or offline. We decided to use OpenVSLAM [59] which was built upon ORB-SLAM2 [40] as our monocular, feature-based tracker. Although literature on SLAM is diverse, most state-of-the-art systems are dense (such as [20, 22]) or fuse more than one type of sensor [33, 41, 47], however, ORB-SLAM2 is still as of today the best reference when it comes to feature-based SLAM systems.

### 3.1. Monocular, Feature-based Tracker

While the task of reconstructing a dense-map of the environment naturally pushes towards choosing a dense/semi-dense SLAM, experiments have shown that the adversarial nature of agricultural scenes made those systems unreliable. Meanwhile, we noticed that feature-based methods do not necessarily suffer from the drawbacks inherent to our domain. Descriptors can be made invariant to lighting and (partially) to blurring, such as [7, 36, 51], which means the tracking is resilient to e.g. holes in the ground, or variable lighting condition due to clouds.

**Auto-Masking of Far Points** We made modifications to the SLAM system to mask points belonging to the horizon-line dynamically, as they do not bring the depth information necessary to perform tracking. This is done by estimating the limit between sky (known to be seen at the top of the frame) and the field (known to be at the bottom of the frame), then masking the top of the image until this limit (plus an offset, used to filter all points which close to horizon line), see Figure 3 (a) and (b) for masking example.

**Monocular Initialization** The threshold that makes monocular tracking module choose between homography and fundamental matrix model to initialize has been changed so the system picks the fundamental matrix more often:

$$R_H = \frac{S_H}{S_H + S_F} \tag{1}$$

where $S_H$ and $S_F$ are the scores computed parallel for homography and fundamental matrix, as explained in [39]. We found out a robust heuristic to select homography under agricultural settings is $R_H > 0.5$ or even $R_H > 0.8$ in some extreme case. This is purely a domain adaptation change, as we know planar structure are virtually nonexistent in the agricultural scenes we study. The number of ORB features extracted in each frame is also increased drastically to 4000, such as to make the tracking more resilient to potential wrong matches (which arise due to the repetitive nature of the scenes).

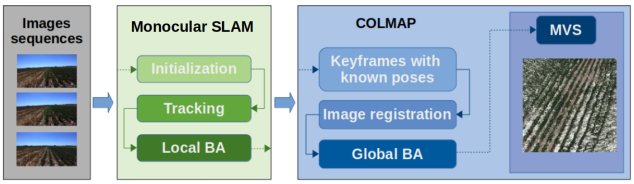



Figure 2. **The workflow of COLMAP initialized by monocular SLAM for dense reconstruction.** Figure modified from original workflow of [55].

**Scale** Absolute world scale is not observable from a monocular SLAM alone. This is a problem we need to tackle as the scale of our reconstruction directly depends on the scale of our tracker. However, we argue that this problem is easy to solve since it is possible to recover scale information in many ways. GPS, IMU, or even using some object of known dimensions, can all be used to recover scale information.

We used GPS information in our implementation as it is available in the dataset we worked with. The estimated trajectory from monocular SLAM will be aligned with scale correction as described in [58]. Geo-registration of camera center with absolute 3D coordinates will be established before offline MVS reconstruction, as it is described in next section. Moreover, using predicted depth with monocular camera to simulated RGB-D sensor is an alternative to tackle the scale issue, which is evaluated in Section 4.

### 3.2. MVS Dense Reconstruction

Once the pose graph has been created, it can be passed to an MVS component that densely reconstructs the environment using the input frames and the corresponding camera poses. This reconstruction can be done either real-time or offline. Naturally, offline solutions provide much more accurate results, which is of interest for some agricultural applications such as 4D monitoring [19]. Real-time solutions, on the other hand, provide an initial estimate of the scene which can be used in other tasks where reconstruction does

not need to be very dense, such as auto steering.

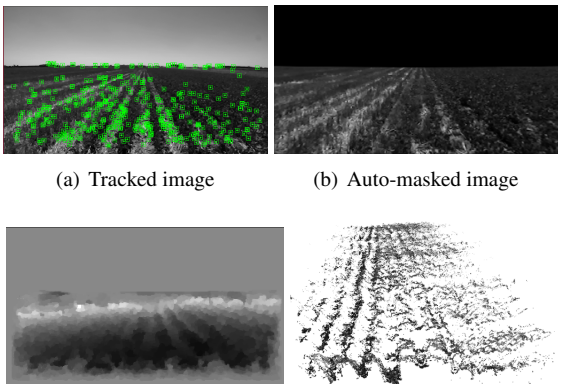
Therefore, the employed SLAM system in this work was embedded with an real-time MVS pipeline, while storing the data for an offline reconstruction once the exploration was finished.

**Offline Dense Reconstruction** We choose COLMAP [57] for the offline, dense reconstruction of our map. It has a well-engineered implementation of Structure-from-Motion (SfM) workflow with Multi-View Stereo (MVS) algorithm [56]. The output of SfM is the scene graph includes the camera poses and sparse point cloud, which is considered the same as the output of a sparse SLAM (pose graph). Replacing the SfM part, the standard pipeline is modified by passing the key-frame poses computed by monocular OpenVSLAM to obtain better results. The workflow is illustrated in Figure 2. Note that in the image registration stage, we reconstruct sparse point cloud again (not required, but more convenient) to provide neighbourhood information for MVS, in the meantime geo-register image by providing absolute coordinates of camera center. This is similar as the direct geo-referencing using GPS measurement introduced in [12]. Thus, the generated point cloud (Figure 1) is up to real scale and prepared for post-processing, see Figure 8 for the geometric analysis on the point cloud.

**Online Dense Reconstruction** There are few real-time MVS reconstruction pipelines, for the obvious reason that accurate dense reconstruction requires a lot of computational power. Still, we are able to integrate REMODE (REgularized MOnocular Depth Estimation) [46] with monocular OpenVSLAM to generate maps whose accuracy are high enough for some agricultural tasks such as auto-steer. REMODE creates depth filters for every keyframe on per-pixel basis and works on all tracked frames (unlike our offline mode, where only keyframes are used). The filter is initialized with high uncertainty in depth and the mean is set to the average scene depth in the reference frame. Given a set of triangulated noisy depth measurements $d_1, d_2, ..., d_k$ that correspond to same pixel location, the estimated depth measurement $\tilde{d}_k$ is modeled with a Gaussian + Uniform mixture model distribution [61]:

$$p(\tilde{d}_k|\hat{d}, \rho) = \rho\mathcal{N}(\tilde{d}_k|\hat{d}, \tau_k^2) + (1-\rho)\mathcal{U}(\tilde{d}_k|d_{min}, d_{max}) \tag{2}$$

where a good depth measurement is assumed to be distributed around the true depth $\hat{d}$ while outlier depth measurements are uniformly distributed within an interval $[d_{min}, d_{max}]$. $\rho$ and $\tau_k^2$ are the probability and the variance of a good measurement. Each new observation is added to its filter, until the covariance is low enough. Thence the filter is considered as having converged, and the 3D point is created in the map using the estimated depth. The output of our system using the online MVS pipeline can be seen Figure 3.



(a) Tracked image (b) Auto-masked image

(c) Estimated depth map (d) Dense point cloud

Figure 3. **Real-time monocular dense reconstruction from REMODE [46]** on Rosario [45], sequence 03, which is running in parallel with monocular SLAM used in this work.

### 3.3. Self-Supervised Monocular Depth Estimation

We employ Monodepth2 [26] as our depth estimation method, which can be self-supervised trained on both monocular videos and stereo pairs. The model is a fully convolutional U-Net [49] (encoder-decoder structure). When trained on monocular videos, an extra pose estimation network is established to predict the egomotion between image pairs.

**Training** We trained Monodepth2 with monocular (M), stereo (S) and mixed method (MS) either from pretrained encoder on ImageNet [53] or starting with high resolution mixed model pretrained (MS*) on KITTI [23] (mono+stereo_1024x320) which is provided by C. Godard *et al*. [26]. The depth encoders of all models mentioned are ResNet-18 [30]. All models are trained with a batch size of 6 on single GPU (GEFORCE GTX 1080 Ti) for 10 epochs. The learning rate is set as $1 \times 10^{-4}$ at the beginning and drops by 0.1 every 4 epochs. Images from the right camera is used in training, while images from left camera are only involved in the computation of loss. We did not perform horizontal flips as training augmentation, because the principle points of both cameras are not perfectly in the middle of the frame in Rosario dataset. Other data augmentations include random brightness, contrast, saturation, and hue jitter with respective ranges of $\pm 0.2$, $\pm 0.2$, $\pm 0.2$, and $\pm 0.1$.

**Ground Truth and Metric** Rosario dataset contains only stereo images, the depth ground truth is generated with SGBM [31] algorithm and therefore relative noisy. To perform quantitative evaluation, we scaled the predicted

depth with the ratio between the median values of predicted depth and ground truth, as done in [26]:

$$D^*_{predict} = \frac{median(D_{gt})}{median(D_{predict})} D_{predict} \quad (3)$$

where the performance metrics of depth estimation used in this work are: Absolute Relative Error (Abs Rel), Square Relative Error (Sq Rel), Root Mean Square Error (RMSE), RMSE log and Accuracy with threshold $(1.25, 1.25^2, 1.25^3)$, marked with red (lower the better) and blue (higher the better) in Table 2.

In the comparison of the depth estimation accuracy of all six methods in terms of the metrics above, the best results appear both in monocular (M*) and mono+stereo (MS*) training strategies, which results in difficulty of selecting the best model. Thus, we had to simulate RGB-D SLAM with all the possible model at hand (presented in Table 3). The qualitative results of the depth estimation using SGBM and the depth prediction from Monodepth2 with mixed training (MS*) are shown in Figure 4. More results are presented in the supplementary material. The network provides generally accurate depth map, but meets some defects on texture-copy artifacts (e.g. the vehicle windows, 5th row) and on objects with intricate shape (e.g. human bodies and vehicles, 1st row and 4th row).

## 4. Experiments and Results

Three sets of experiments are presented. The first one is the performance benchmarking on dataset Rosario [45] with different SLAM configurations, where we provide the baseline for monocular SLAM and improved results for stereo SLAM, compared to the relative work [16, 45] which only evaluated stereo setting successfully. Then, along with evaluating monocular SLAM, we establish the ablation study using predicted depth image to simulate RGB-D SLAM. Finally, we discuss the dense point cloud generated in this work.

### 4.1. Dataset and Evaluation Methodology

Exhaustive experiments were established on the Rosario dataset in this work, which is a recently released dataset composed of six different sequences in a soybean field. The available sensor measurements include stereo images (672 $\times$ 376, 15 Hz) and GPS-RTK (5 Hz). The sensors were synchronized and calibrated (both intrinsic and extrinsic). The difficulty of the sequences varies as shown in Table 1. For more details about the agricultural robotic and sensors, please refer to [45].

The qualitative results of MVS dense reconstruction can be seen in Figure 1 and 3, where we show the reconstructed point clouds from both offline and online methods. Note that the pose graph was geo-registered by providing the absolute position of the camera center before implementing offline MVS on it (typical accuracy of GPS-RTK is around 1cm horizontally and around 2cm vertically). For specific tasks like 4D monitoring of crops, this dense point cloud can be used to calculate different kinds of geometric features such as point density.

| Dataset Rosario | | S-PTAM [44] | ORB-SLAM2 [40] | | OpenVSLAM [59] | |
|---|---|---|---|---|---|---|
| Sequence | Length | Stereo | Stereo | Mono | Stereo | Mono |
| 01_easy | 615.15 | 3.85 (0.63%) | 1.41 (0.23%) | X | **1.35 (0.22%)** | 10.19 (1.66%) |
| 02_easy | 320.16 | **1.80 (0.56%)** | 2.24 (0.70%) | X | 1.95 (0.61%) | 28.17 (8.80%) |
| 03_medium | 169.45 | 2.37 (1.40%) | 3.50 (2.06%) | X | **1.75 (1.03%)** | 4.29 (2.53%) |
| 04_medium | 152.32 | 1.49 (0.98%) | 2.21 (1.45%) | X | **1.48 (0.97%)** | 6.14 (4.03%) |
| 05_difficult | 330.43 | X | 2.23 (0.68%) | X | **1.65 (0.50%)** | 23.66 (7.16%) |
| 06_difficult | 709.42 | X | 5.19 (0.73%) | X | **3.41 (0.48%)** | 91.13 (12.85%) |

Table 1. **Absolute trajectory error (ATE) [m] (ratio ATE over trajectory length, in %)** (X stands for tracking failure). The results of S-PTAM and Stereo ORB-SLAM2 are extracted from Rosario [45], Mono ORB-SLAM2 is evaluated in this work with default configuration but the system cannot initialize on any sequence.

The quantitative results of absolute trajectory error (ATE) estimated from SLAM are shown in Table 1 and 3, corresponding trajectories are illustrated in Figure 5, 6 and supplementary material. Besides the standard evaluation of ATE for SLAM systems, we also highlight the importance of point density which is used in the field of agricultural mapping (the post-processing results can be seen in Figure 8). As introduced in [50], a satisfactory methodology to simplify the resolution of 3D field maps while maintaining the key information is through the concept of 3D density and density grids. The idea of the 3D density is rooted in the properties of the conventional density, which establishes a relationship between the mass of a substance and the volume that it occupies:

$$d = N/V \quad (4)$$

Where $N$ indicates the number of points and $V$ indicates the 3D volume with a radius defined by the user. Two practicable approaches to apply the concept of 3D density is to compute either a precise density: the density is estimated by counting for each point the number of neighbors $N$ (inside a sphere of radius $R$); or by computing approximate density: it is then simply estimated by determining the distance to the nearest neighbor (which is generally much faster). This distance is considered as being equivalent to the above spherical neighborhood radius $R$ (and $N$ = 1). In this work, we first compute the precise density, namely, the number of neighbors $N$ with radius of 0.1 m (the absolute scale is known from geo-registration), see Figure 8. Thereafter the volume density is calculated simply as:

$$d = N/(4/3 \cdot \pi R^3) \quad (5)$$

### 4.2. Ablation Study

Part of our contribution is evaluating self-supervised depth estimation on agricultural image sequence, along with

Figure 4. **Qualitative results of self-supervised monocular depth estimation on Rosario [45].** First column: selected raw RGB images; Second column: ground truth depth images generated with SGBM [31]; Third column: predicted depth using Monodepth2 [26] with mixed training strategy (MS*). More results please see supplementary material.

| Method | Abs Rel | Sq Rel | RMSE | RMSE log | $\sigma < 1.25$ | $\sigma < 1.25^2$ | $\sigma < 1.25^3$ |
|---|---|---|---|---|---|---|---|
| M | 0.151 | 2.110 | 2.961 | 0.205 | 0.913 | **0.978** | **0.989** |
| M* | 0.150 | 2.118 | **2.934** | **0.204** | **0.915** | **0.978** | **0.989** |
| S | 0.231 | 1.716 | 5.632 | 0.646 | 0.665 | 0.905 | 0.919 |
| S* | 0.235 | 1.737 | 5.650 | 0.644 | 0.657 | 0.900 | 0.919 |
| M+S | **0.116** | 0.747 | 3.147 | 0.221 | 0.886 | 0.929 | 0.963 |
| M+S* | **0.116** | **0.742** | 3.165 | 0.222 | 0.885 | 0.929 | 0.962 |

Table 2. **Ablation study.** Quantitative results of Monodepth2 [26] depth estimation using different variants of training methods on Rosario [45]. **Legend: S** - Self-supervised stereo supervision; **M** - Self-supervised mono supervision; ***** - start with model pretrained on KITTI [23] (otherwise, the depth encoder is initialized with pretrained weights on ImageNet [53]).

monocular visual SLAM and simulating RGB-D SLAM. Therefore, a comparison between different training strategy on Rosario [45] is given in Table 2 using Monodepth2 [26]. We evaluated all the models trained to simulate RGB-D SLAM, where the estimated ATEs are shown in Table 3. To provide a baseline for future work, there is no specific change in the CNN structure in this work. We discuss the problem regarding to Monodepth2 in Section 4.2.2.

| **OpenVSLAM** | | ATEs estimated on Dataset Rosario | | | | | |
|---|---|---|---|---|---|---|---|
| Setting | Train | 01 | 02 | 03 | 04 | 05 | 06 |
| Mono+$D_{GT}$ | - | 8.32 | 4.94 | 5.70 | 4.31 | 5.92 | 13.60 |
| Mono+$D_{CNN}$ | M | X | X | X | X | X | X |
| Mono+$D_{CNN}$ | M* | 10.99 | 12.46 | 16.98 | 14.52 | 13.58 | 33.35 |
| Mono+$D_{CNN}$ | S | **5.21** | 3.80 | 2.79 | 3.01 | 3.26 | 8.40 |
| Mono+$D_{CNN}$ | S* | 5.25 | 3.78 | 2.73 | 2.96 | 2.90 | 8.62 |
| Mono+$D_{CNN}$ | MS | 5.44 | 3.57 | **2.54** | **2.71** | 2.95 | **7.52** |
| Mono+$D_{CNN}$ | MS* | 5.37 | **3.41** | 2.62 | 2.94 | **2.63** | 7.79 |
| Mono+$D_{GT}^{scaled}$ | - | 7.44 | 2.03 | 0.678 | **0.25** | 2.39 | **5.78** |
| Mono+$D_{CNN}^{scaled}$ | M | X | X | X | X | X | X |
| Mono+$D_{CNN}^{scaled}$ | M* | 9.30 | 2.91 | 1.04 | 0.72 | 3.60 | 10.30 |
| Mono+$D_{CNN}^{scaled}$ | S | **2.31** | 1.40 | 0.59 | 0.28 | 2.37 | 6.35 |
| Mono+$D_{CNN}^{scaled}$ | S* | 2.71 | 1.35 | 0.59 | 0.29 | 1.98 | 6.95 |
| Mono+$D_{CNN}^{scaled}$ | MS | 3.61 | 1.35 | 0.60 | 0.26 | 2.24 | 6.14 |
| Mono+$D_{CNN}^{scaled}$ | MS* | 3.29 | **1.26** | **0.57** | 0.26 | **1.75** | 6.18 |
| Stereo (**baseline**) | - | 1.35 | 1.95 | 1.75 | 1.48 | 1.65 | 3.41 |

Table 3. **Ablation study.** Quantitative results using estimated depth simulating RGB-D SLAM, where $D_{GT}$ and $D_{CNN}$ indicate whether the depth is generated from stereo image pair as ground truth or estimated from Monodepth2 used in this work, *scaled* means the estimated trajectory is aligned with scale correction. Baseline (stereo OpenVSLAM) extracted from Table 1.

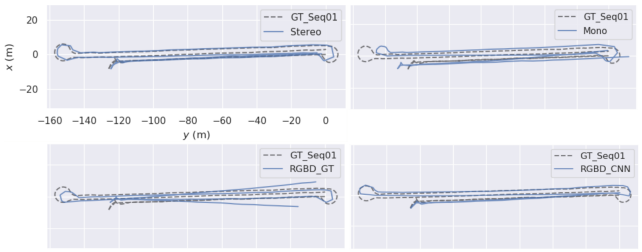
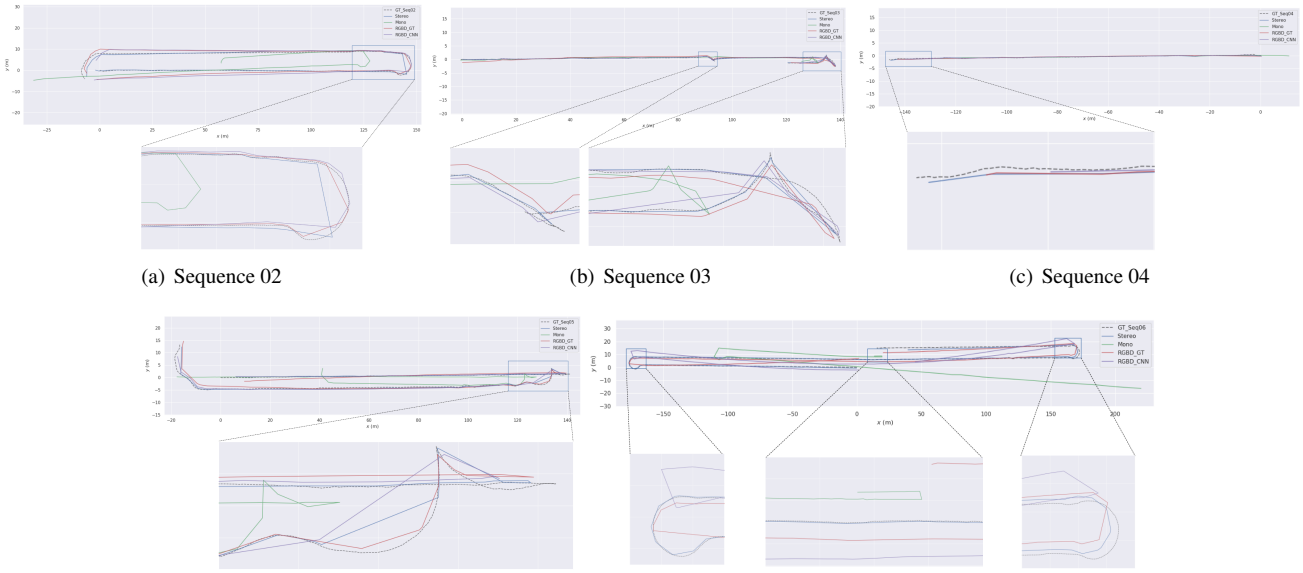



Figure 5. **Estimated trajectories and the ground truth of Rosario dataset, sequence 01**. The illustrated results are refer to our quantitative results shown in Table 1 and Table 3 regarding to OpenVSLAM: Stereo, Mono, Mono+$D_{GT}^{scaled}$ (RGBD_GT) and Mono+$D_{CNN}^{scaled}$ (RGBD_CNN, trained model MS*). Results of sequence 02-06 are presented separately in Figure 6.

### 4.2.1 Visual SLAM on Rosario

As shown in Table 1, we present absolute trajectory error (ATE) estimated from monocular OpenVSLAM used in this work, and improved results for stereo SLAM which outperforms the previous baselines from [45] in general. Each result of this work was calculated by averaging 5 runs on each sequence. Notice that there is no specific algorithm improvement comparing OpenVSLAM to ORB-SLAM2. Some domain adapted modification on the threshold used in this work was introduced in Section 3. Comparing to the default configuration of monocular SLAM, our modification solved problems of initialization and tracking failure, which is the reason no other work [16, 45] can present results from monocular SLAM. In fact, sequence 03 and 04 are the two easiest sequences for SLAM as the movement is simple straight forward, where we obtain good results by simulating RGB-D SLAM and competitive good results from Monocular setting.



(a) Sequence 02 (b) Sequence 03 (c) Sequence 04

(d) Sequence 05 (e) Sequence 06

Figure 6. **Estimated trajectories and the ground truth of Rosario dataset, sequence 02-06**.

**Problematic Drifts** Serious drift may occur after the camera inverted its direction (U-turn), such as sequence 02, 05, and 06 evaluated with Mono SLAM (Figure 6: (a), (d) and (e)). Worst case happen on sequence 06 with Monocular setting, where the scale and estimated trajectory drift dramatically (Figure 6: (e), the trajectory in green). We also observe that the estimated trajectory on sequence 01, 06 from simulated RGB-D SLAM, drifts after U-turn. We conclude that this is due to the error from ground truth depth generation (see Figure 5 bottom-left, result of RGB_GT) and self-supervised training (see Figure 5 bottom-right, result of RGB_CNN).

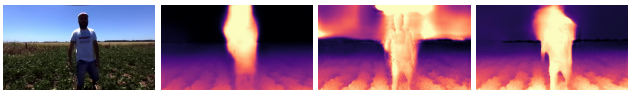
**Drifts in Z-Axis** As we cannot assume a perfect 2D ground plane existing under agricultural scenario and the drifts in z-axis direction have to be considered. The 3D trajectories estimated from SLAM with xyz_view are illustrated in the supplementary material.

**Scale Correction** We simulate RGB-D sensor but get better ATE results using scale correction when aligning the trajectory with ground truth (see Table 3: Mono+D$^{scale}_{GT}$ and Mono+D$^{scale}_{CNN}$), while the results from Mono+D$^{scale}_{CNN}$ are close to the results from Stereo SLAM, which shows that similar performance can be obtained by simulating RGB-D camera instead of using a raw stereo camera. However, the ground truth depth image should introduce a similar scale as it is generated from the stereo image pair but we still need scale correction, which means the obvious error was introduced during ground truth generation using the method of SGBM [31]. Comparing all the results of Mono+D$_{CNN}$, shows that the Monodepth2 also has trouble to learn the accurate scale from agricultural image sequences in a self-supervised fashion, which is further discussed in next Section 4.2.2.

**Reproducibility** Running Stereo SLAM on Rosario [45] is straightforward and reasonable good results can be obtained, however, we observe that the heuristic threshold used for initializing monocular tracking and the number of ORB features extracted will influence the robustness of the system (as discussed in Section 3). Thus, we provide our experimental results in the supplementary material, where interested readers can find every single value calculated from different SLAM configurations and from 5 test runs on each data sequence.

### 4.2.2 Self-Supervised Depth Estimation on Rosario

**Failure on Textureless Region** Comparing to urban scenes datasets like KITTI [23], most frames in Rosario dataset [45] contain a large portion of textureless sky regions. When using stereo training strategy (S/S*), Monodepth2 produces imprecise depth values on low texture regions. When using mixed strategy (MS/MS*), it estimates relative precise depth values on these regions, which is more distinct with foreground objects. Due to the correspondence difficulty, the photometric reconstruction error is ambiguous in large textureless regions. Therefore, a wide range of predicted depth values can produce the same photometric error, which is hard to be optimized based on the left-right consistency assumption [25].



(a) RGB image (b) Mono (M*) (c) Stereo (S*) (d) Mixed (MS*)

Figure 7. **Failure on objects with textureless background.** The network smooths the depth prediction of the sky with the foreground object and results to ambiguous contour of the object.

The feature-based SLAM system combined with auto-masking of far points (as discussed in Section 3.1), tracks no feature point on the textureless region, thus minimizing the negative effects of the unreliable depth estimation. However, we observe some failure cases, which may influence the performance of the SLAM system. As illustrated in Figure 7, the depth prediction of textureless region around the foreground object is polluted, which results in ambiguous boundary of the foreground object. This blooming effect is driven by the edge-aware smoothness loss [48] and appears more likely on objects with intricate shape.

**Effect of Pretraining** As shown in Table 3, through the comparison of all the training strategies with/without weights pretrained on KITTI, we find out using pretrained model on other dataset does not explicitly improves the SLAM performance. This reveals that the transferability of Monodepth2 (with ResNet-18 as depth encoder) is limited. However, pretrained model guarantees the stability and robustness of RGB-D based tracking, while tracking failure continues to happen on all the sequence using the model specifically from monocular training (M) without pretrained on KITTI. Obviously, the depth and scale ambiguity is not learned by monocular training (M) standalone.

As stated above, we recommend interested readers to utilize Monodepth2 with the mixed training strategy and pretrained weights (MS*) to reproduce our work and research on similar agriculture scenes.

### 4.3. Dense Reconstruction

In general, MVS can be initiated either with SfM or visual SLAM depending on whether the input data is an ordered sequence or unordered images, which means one of the pre-conditions is the poses of the images can be successfully recovered beforehand. In this work, we are able to reconstruct the dense point cloud offline (Figure 1) up to real scale after geo-registration, where the potential drifts are eliminated by GPS measurement. However, the employed real-time algorithm REMODE estimates depth based on depth filter, which approximates the mean and variance of the depth at each pixel's position and updates the depth uncertainty when there is a new measurement (new image captured from the camera). The implementation of depth filter naturally requires a high

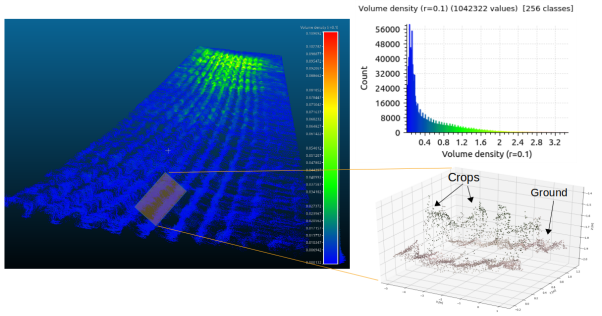



Figure 8. **Volume density ($R = 0.1$ m) of the dense point cloud shown in Figure 1.** Left: the density heatmap of the point cloud; Top-right: the histogram of volume density; Bottom-right: a subset of the dense point cloud.

frame rate to converge the depth uncertainty which is not the case regarding dataset Rosario (15Hz). While we are still able to reconstruct coarse dense point cloud on the fly using REMODE (Figure 3), a potential improvement could be to initialize the depth filter according to the depth estimated from CNN (e.g. consider depth estimated from Monodepth2 as the prior knowledge of the scene geometry) to accelerate convergence, as discussed in [35].

**Point Cloud and Density** The volume density is calculated using CloudCompare [2], which is an open-source 3D point cloud and mesh processing software (see Figure 8). The density of the map stays relatively constant throughout the sequence, except during slowdowns and stops. In those cases, more keyframes are taken within the same area, increasing the density of the map in this region. Moreover, we illustrate on a very small subset of the dense point cloud, where we can see the height of the crops. The crops and ground can be easily recognized, separated, and measured, which provides very valuable information.

## 5. Conclusion

Our work successfully presented a monocular vision-based architecture for mapping and localization explored under challenging agricultural environment, with new baselines provided for the relevant research community. Future works can explore other types of indirect SLAM systems, such as ones integrating GPS, or IMU, thus leveraging the advantages of feature-based tracking described here without the drifting issue.

## 6. Acknowledgment

The research leading to these results has been partially funded by the German BMBF project MOVEON (Funding reference number 01IS20077) and by the German BMBF project SocialWear (Funding reference number 01IW20002).

## References


[1] Software Agisoft Metashape. https://www.agisoft.com/.

[2] Software CloudCompare. https://www.danielgm.net/cc/.

[3] Software Pix4D. https://www.pix4d.com/.

[4] Moises Alencastre-Miranda, Joseph R Davidson, Richard M Johnson, Herman Waguespack, and Hermano Igo Krebs. Robotics for sugarcane cultivation: Analysis of billet quality using computer vision. *IEEE Robotics and Automation Letters*, 3(4):3828–3835, 2018.

[5] Joel Burkhard, S Cavegn, A Barmettler, and S Nebiker. Stereovision mobile mapping: System design and performance evaluation. *Int. Arch. Photogram. Remote Sens. Spatial. Inform. Sci*, 5:453–458, 2012.

[6] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.

[7] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. volume 6314, pages 778–792, 09 2010.

[8] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016.

[9] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2019.

[10] S Cavegn, S Blaser, S Nebiker, and N Haala. Robust and accurate image-based georeferencing exploiting relative orientation constraints. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(2), 2018.

[11] Stefan Cavegn and Norbert Haala. Image-based mobile mapping for 3d urban data capture. *Photogrammetric Engineering & Remote Sensing*, 82(12):925–933, 2016.

[12] S Cavegn, S Nebiker, and N Haala. A systematic comparison of direct and image-based georeferencing in challenging urban areas. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

[13] Nived Chebrolu, Philipp Lottes, Alexander Schaefer, Wera Winterhalter, Wolfram Burgard, and Cyrill Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 2017.

[14] Fernando Alfredo Auat Cheein and Ricardo Carelli. Agricultural robotics: Unmanned robotic service units in agricultural tasks. *IEEE industrial electronics magazine*, 7(3):48–58, 2013.

[15] Hongyu Chen, Zhijie Yang, Xiting Zhao, Guangyuan Weng, Haochuan Wan, Jianwen Luo, Xiaoya Ye, Zehao Zhao, Zhenpeng He, Yongxia Shen, et al. Advanced mapping robot and high-resolution dataset. *Robotics and Autonomous Systems*, page 103559, 2020.

[16] Román Comelli, Taihú Pire, and Ernesto Kofman. Evaluation of visual slam algorithms on agricultural dataset.

[17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[18] Maurilio Di Cicco, Ciro Potena, Giorgio Grisetti, and Alberto Pretto. Automatic model based dataset generation for fast and accurate crop and weeds detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5188–5195. IEEE, 2017.

[19] Jing Dong, John Gary Burnham, Byron Boots, Glen Rains, and Frank Dellaert. 4d crop monitoring: Spatio-temporal reconstruction for agriculture. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3878–3885. IEEE, 2017.

[20] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. In *arXiv:1607.02565*, July 2016.

[21] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014.

[22] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[23] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[24] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[25] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[26] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019.

[27] W Nicholas Greene and Nicholas Roy. Metrically-scaled monocular slam using learned scale factors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 43–50. IEEE, 2020.

[28] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE international conference on Robotics and automation (ICRA)*, pages 1524–1531. IEEE, 2014.

[29] Sebastian Haug and Jörn Ostermann. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. In *European Conference on Computer Vision*, pages 105–116. Springer, 2014.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

*ings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[31] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[32] Hualie Jiang, Laiyan Ding, and Rui Huang. Dipe: Deeper into photometric errors for unsupervised learning of depth and ego-motion from monocular videos. *arXiv preprint arXiv:2003.01360*, 2020.

[33] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34:314 – 334, 2015.

[34] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[35] Shing Yan Loo, Ali Jahani Amiri, Syamsiah Mashohor, Sai Hong Tang, and Hong Zhang. Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5218–5223. IEEE, 2019.

[36] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[37] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[38] András L Majdik, Charles Till, and Davide Scaramuzza. The zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research*, 36(3):269–273, 2017.

[39] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[40] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[41] Raul Mur-Artal and Juan D. Tardos. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, Apr 2017.

[42] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327, Nov 2011.

[43] Misbah Pathan, Nivedita Patel, Hiteshri Yagnik, and Manan Shah. Artificial cognition for applications in smart agriculture: A comprehensive review. *Artificial Intelligence in Agriculture*, 2020.

[44] Taihú Pire, Thomas Fischer, Gastón Castro, Pablo De Cristóforis, Javier Civera, and Julio Jacobo Berlles. S-ptam: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, 93:27–42, 2017.

[45] Taihú Pire, Martín Mujica, Javier Civera, and Ernesto Kofman. The rosario dataset: Multisensor data for localization and mapping in agricultural environments. *The International Journal of Robotics Research*, 0(0):1–9, 2019.

[46] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[47] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[48] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12240–12249, 2019.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[50] Francisco Rovira-Más, Qin Zhang, and John F Reid. Stereo vision three-dimensional terrain maps for precision agriculture. *Computers and Electronics in Agriculture*, 60(2):133–143, 2008.

[51] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.

[52] José Raúl Ruiz-Sarmiento, Cipriano Galindo, and Javier González-Jiménez. Robot@ home, a robotic dataset for semantic mapping of home environments. *The International Journal of Robotics Research*, 36(2):131–141, 2017.

[53] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[54] Inkyu Sa, Zetao Chen, Marija Popović, Raghav Khanna, Frank Liebisch, Juan Nieto, and Roland Siegwart. weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE Robotics and Automation Letters*, 3(1):588–595, 2017.

[55] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[56] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[57] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[58] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ Interna-*

*tional Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.

[59] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. Openvslam: a versatile visual slam framework. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2292–2295, 2019.

[60] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6243–6252, 2017.

[61] George Vogiatzis and Carlos Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing*, 29(7):434 – 441, 2011.

[62] Stavros G Vougioukas. Agricultural robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:365–392, 2019.

[63] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.

[64] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[65] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018.

[66] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.