Learning Occluded Branch Depth Maps in Forest Environments Using RGB-D Images

Christian Geckeler, *Graduate Student Member, IEEE*, Emanuele Aucone, Yannick Schnider, Andri Simeon, Jan-Philipp von Bassewitz, Yunying Zhu, and Stefano Mintchev, *Member, IEEE*

***Abstract*—Covering over a third of all terrestrial land area, forests are crucial environments; as ecosystems, for farming, and for human leisure. However, they are challenging to access for environmental monitoring, for agricultural uses, and for search and rescue applications. To enter, aerial robots need to fly through dense vegetation, where foliage can be pushed aside, but occluded branches pose critical obstacles. Therefore, we propose pixel-wise depth regression of occluded branches using three different U-Net inspired architectures. Given RGB-D input of trees with partially occluded branches, the models estimate depth values of only the wooden parts of the tree. A large photorealistic simulation dataset comprising around 44 K images of nine different tree species is generated, on which the models are trained. Extensive evaluation and analysis of the models on this dataset is shown. To improve network generalization to real-world data, different data augmentation and transformation techniques are performed. The approaches are then also successfully demonstrated on real-world data of broadleaf trees from Swiss temperate forests and a tropical Masoala Rainforest. This work showcases the previously unexplored task of frame-by-frame pixel-based occluded branch depth reconstruction to facilitate robot traversal of forest environments.**

***Index Terms*—Deep learning for visual perception, robotics and automation in agriculture and forestry, RGB-D perception.**

I. Introduction

FORESTS represent an integral part of Earth's biosphere, covering over a third of all terrestrial land area [2], supporting more than half of the world's vertebrate species [3] as well as providing essential ecosystem services and climate regulation [4]. Additionally, 43% of all agricultural land globally has at least 10% tree cover [5]. Dozens of meters tall and often situated in remote locations, they are challenging to access and survey.

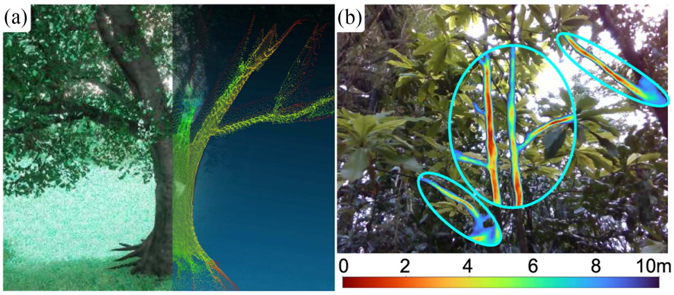

Fig. 1. (a) Simulation RGB tree image (left) and predicted depth point cloud only for branches (right). (b) Predicted pixel-wise depth values only for branches, on real images from trees from a Masoala Rainforest (colorbar in meters).

These circumstances present a natural opportunity for autonomous and highly agile micro aerial vehicles (MAVs), which are beginning to demonstrate flight above and below tree canopies in forests with sparse vegetation [6], [7]. MAVs are also being utilized for environmental tasks such as sample collection from above the treetops [8], or sensor placement and environmental monitoring in the outermost canopy regions [9], [10], [11].

Besides environmental monitoring, MAVs are also demonstrating increased use in agriculture, such as for sensing and detection in orchards [12], or fruit harvest [13], as well as for search and rescue operations in forest environments [14].

The challenge of navigating through cluttered and dense foliage remains a significant hurdle for MAVs, rendering tree canopies largely inaccessible. Recent developments have shown that since MAVs can push aside compliant obstacles [15], twigs and leaves would pose little threat to MAVs with shielded propellers [16], but colliding with thick branches and the trunk might destabilize the drone and result in a fatal crash. Therefore, a perception system to detect the wooden parts of the tree - the branches and the trunk - is necessary. Partially or fully occluded branches pose a major challenge, especially in forest environments. This prevents the application of classic computer vision approaches based on shape, feature, or color detection. Artificial neural networks have demonstrated much promise in generative assistance for visual perception in challenging environments, including Generative Adversarial Networks (GANs) [17].

To address these challenges, this letter proposes a neural network-based system to predict the pixel-wise depth values only of the wooden parts of the tree (see Fig. 1), given a single RGB-D image (RGB and depth) of trees with partially occluded

Manuscript received 18 July 2023; accepted 12 December 2023. Date of publication 18 January 2024; date of current version 31 January 2024. This letter was recommended for publication by Associate Editor H. Ryu and Editor H. Moon upon evaluation of the reviewers' comments. This work was supported by the Swiss National Science Foundation through the Eccellenza under Grant 186865. *(Corresponding author: Christian Geckeler.)*

The authors are with Environmental Robotics Laboratory, Department of Environmental Systems Science, ETH Zurich, 8092 Zurich, Switzerland, and also with the Swiss Federal Institute for Forest, Snow and Landscape Research (WSL), 8903 Birmensdorf, Switzerland (e-mail: cgeckeler@ethz.ch; eaucone@ethz.ch; yannicks@ethz.ch; ansimeon@ethz.ch; vbjan@ethz.ch; yunzhu@ethz.ch; smintchev@ethz.ch).

All models, code and data are available online: https://doi.org/10.3929/ethz-b-000634419.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3355632, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3355632

branches. Utilizing a photorealistic simulator, a large RGB-D dataset of different tree species with and without leaves is generated. The synthetic data is then used to train U-Net inspired encoder-decoder networks. To improve network generalization to real-world data, different data augmentation and transformation techniques are performed. Finally, to validate the approach, the feasibility of the outputs is shown on real data from two discrete biomes.

The main contributions of this work are as follows:

- Generation of a dataset with around 44 K RGB-D images of nine different species of photorealistic simulated trees, with and without leaves; transformed and augmented for real-world domain adaptation (training and groundtruth, datasets available online [1]).
- Training and extensive quantitative evaluation of three different U-Net [18]-inspired architectures on the previously unexplored topic of predicting pixel-wise depth of occluded tree structures from a single frame using simulated data (code available online [1]).
- Qualitative evaluation of the networks on real-world data from trees from Swiss temperate forests and a rainforest, demonstrating the real-world utility of the approach (data captures available online [1]).

II. Related Works

The depth estimation of occluded natural structures is mostly unexplored. Recently, an offline heuristic-based approach for extracting occluded tree skeletons in orchards was presented in [19]. First, visible branches are extracted via RGB instance segmentation, which are then "extended" in the longitudinal branch direction, resulting in a 3D likelihood map of potential branch locations. Using images from multiple viewpoints, the final tree skeleton is then extracted through consolidating the line segments, smoothing them, and connecting them through minimum cost path search. While this approach produces feasible results on simulated trees, it requires parts of branches to be visible in the instance segmentation step. This, along with the heuristic assumption that branches grow straight, fundamentally limits the approach to comparatively simple tree topologies with partially visible branches. Indeed, the qualitative results on the real apple orchard trees show that often branches are truncated due to heavy foliage occlusion, or missed entirely.

While not dealing with occluded branches, in [20], a GAN was used to predict probable grayscale masks of occluded grapes. The networks were trained on masked grayscale images of manually exfoliated grapes which were then synthetically occluded. This represents a comparatively simple scenario since the occluded target grape clusters are identifiable in the images, and follow predictable grouping patterns. In contrast, tree branches in the wild can be more densely occluded, and branch locations cannot easily been inferred based on leaf clustering patterns, since global visual cues about the tree structure may not be available from short distances.

The less complex problem, since the output is a 2D mask, of semantic segmentation on partially occluded branches is covered in [21], where manually annotated RGB-D images of apple trees were used to train a conditional GAN, a U-Net, and a Convolutional Neural Network (CNN). The even simpler case of semantic segmentation of tree-like vegetation from RGB-D input, neglecting occlusions, is investigated in [22]. In existing literature, most approaches are limited to well-structured and visually similar situations, such as apple trees or vineyards, and do not address dense occlusions from variable camera angles with changing depth scales or out-of-distribution species. Most importantly, all of these approaches lack 3D depth information in the output, which is critical for robot navigation.

When training networks to regress pixel-wise depth values of occluded branches, inspiration can be taken from monocular depth estimation. The problem of inferring depth information from a single RGB image in a supervised setting can also be seen as a pixel-wise regression problem. A CNN was first used in [23] to approach this regression task, with different loss functions proposed in the literature thereafter. While the $\mathcal{L}_2$-loss represents a common loss function for regression, model-specific loss functions were designed in [24], and [25] reports better results using a deep fully convolutional residual network and training on the reverse Huber loss.

To regress pixel-wise depth values based on RGB-D input, the color and depth channels must be properly encoded. Although the problem formulation is slightly different, previous literature on semantic segmentation with RGB-D inputs offers insights into RGB-D input encoding strategies. Additional depth information for RGB images has shown to increase segmentation performance [26]. However, it is challenging to fuse the color and depth information in the encoder, due to their different modalities. Naively concatenating the depth information as an additional channel to the RGB channels (*early fusion*) usually results in worse performance on segmentation tasks [27] than the following more evolved strategies. *Late fusion* approaches treat color and depth data in isolation to fuse their respective feature representations at a later stage in the network [28]. Distinct processes extract the relevant segmentation features from each modality separately, which are later combined into a single representation. This fusion can happen before the network decoder [29] or even at the output layer [22]. In [30], the depth input is encoded as an HHA image (horizontal disparity, height above the ground, and the angle of the pixel's local surface normal with the gravity direction) and then concatenated with a segmentation mask predicted by an RGB segmentation model. The resulting tensor is then passed through a final network to refine the previously predicted segmentation mask.

An alternative to the introduced late and early fusion approaches is to fuse the two modalities at each decoding step (*decoder fusion*). In [26] the authors show that fusing at different decoding stages indeed improves segmentation performance. In contrast, depth and RGB feature representations are fused at each encoder stage in [29], [31] (*encoder fusion*). This approach only requires a single decoder, which reduces its computational complexity. The authors of [31] propose a lightweight architecture called ESANet, following the encoder fusion strategy.

The task of pixel-wise depth regression based on RGB-D input has not been extensively investigated. However, for robot navigation depth information is essential - the closer a branch is, the more immediate the threat of a potential collision. Additionally, the information about occluded obstacles can be utilized for obstacle avoidance and path planning.

III. Methodology

In this section, we describe the sensor choice, the network architectures, the dataset generation, the training procedure, and the domain adaptation techniques eventually used for depth predictions based on real-world input data. This work focuses on deciduous trees, since the problem is more challenging, with larger leaves presenting more opportunities for occlusions. Additionally, the wooden part of the branches of coniferous trees can be more easily inferred, since each needle is directly attached to a branch, giving more information about the location and shape of the branch.

A. Sensor Choice

Depth cameras present a mature, relatively cheap, and off-the-shelf solution for acquiring color and depth information from a scene. RGB-D input has also been demonstrated to be sufficient for robot localization and navigation: below the canopy, high speed learning-based end-to-end flights using only depth images as input [32], as well as depth and LiDAR input integrated into more traditional perception and mapping pipelines [6], [7] have been demonstrated. While LiDAR presents an increasingly popular choice of perception sensor for aerial robots [7], [33], the size, weight, price, and large minimum detection range make it ill-suited for small MAVs used to explore the inside of close-range and dense forest canopies. RGB-D sensors do not suffer from these drawbacks and are therefore chosen for this work.

B. Network Architectures

All models presented in this section encode the pixel-wise feature channels to a low dimensional manifold, then decode the compressed information to obtain a pixel-wise output image. To compare different models, a baseline model (*Baseline*) was designed: an established U-Net style architecture [36] performs binary segmentation which is in turn used to mask the input depth data. The target segmentation mask has two classes: tree skeleton pixels (wooden parts) and all remaining pixels as the second class. The early fusion approach was used to concatenate the normalized depth values to the RGB color channels as input to the segmentation U-Net. The final baseline model outputs depth maps containing values only for pixels of the trunk or branches, either visible or occluded. This approach requires intermediate processing steps and is unable to deal with occlusions, nonetheless it presents a meaningful baseline, without loss of generality.

Our first model (U-Net) is the U-Net architecture expanded to perform regression, providing an end-to-end extension of the baseline model, which is able to handle occlusions. While the network encoder remains the same, using early fusion, the logistic activation in the last decoding layer is removed and a single output channel is used.

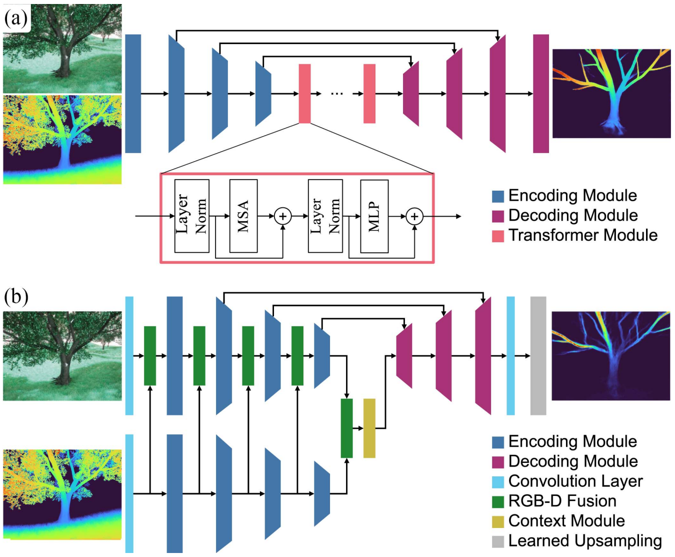

Fig. 2. Network architectures: (a) TransUNet [34], and (b) ESANet [35].

Similarly, the second model (TransUNet) [34] is based on the U-Net architecture with the addition of transformer layers bridging the encoder and decoder module for enhanced feature extraction with global context. The architecture can be seen in Fig. 2(a). Firstly, a feature extractor in the form of the first four layers of a pre-trained U-Net is utilized. Then, a patch and positional embedding, followed by 12 transformer layers, refine the previously obtained features. Finally, the decoder tightly follows the implementation of the U-Net adjusted for regression. The TransUNet also exploits the early fusion approach for the RGB-D input.

In contrast to all the previous models, this last model (ESANet) utilizes the late fusion approach for incorporating the depth information. A simplified diagram of the model is shown in Fig. 2(b). The ESANet is based on an RGB-D semantic segmentation architecture [31], which fuses color and depth at different stages of the encoder. The decoder is modified for depth estimation by enforcing a single output channel with sigmoid activation function, which scales to the minimum and maximum depth (0 m and 10 m).

C. Data Generation

Generating a real-world dataset for model training is infeasible, since exactly the same RGB and depth image of the scene with and without leaves is required. Manually removing foliage disturbs the branches and alters their resting position due to the changed weight distribution. Waiting for seasonal abscission also results in changes and movement of the branches due to the tree growing and changing over such a long time span. Similarly, manually annotating and removing leaves from the 3D depth data is not feasible, since the depth would have to be estimated for each occluded pixel, which would be as challenging as the actual task.

Therefore, a photo-realistic simulator is used to generate the data (Fig. 3(a), (b)), which greatly simplifies the data collection

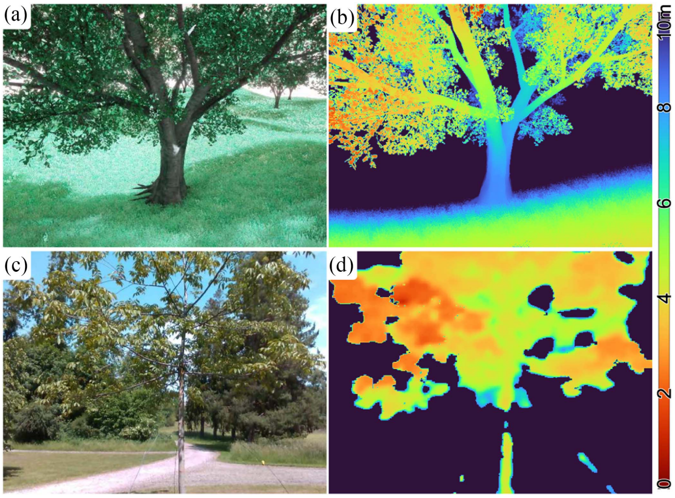

Fig. 3. RGB (a) and depth (b) data obtained in the Unreal Engine simulation, and real RGB (c) and depth (d) data captured with the Intel Realsense D435 depth sensor. Colorbar on the right shows depth scale in meters.

TABLE I
RMSE in Meters for Different Simulated Datasets

Model	Test Whole	Test Skel.	Shagbark Skel.	Walnut Skel.
Baseline	0.502 -	2.411 -	2.881 -	2.930 -
U-Net	0.359 (-28%)	1.129 (-53%)	1.293 (-55%)	1.226 (-58%)
TransUNet	**0.356** **(-29%)**	**1.117** **(-54%)**	1.308 (-55%)	1.179 (-60%)
ESANet	0.403 (-20%)	1.159 (-52%)	**1.270** **(-56%)**	**1.174** **(-60%)**

The best performing values are in bold.

TABLE II
Comparisons of Model Size and Complexity

Model	Baseline	U-Net	TransUNet	ESANet
GFLOPs ↓	256.7	256.7	204.7	**50.5**
Parameters (M) ↓	**31.04**	**31.04**	88.92	46.91
Inference Time (s) ↓	0.341	0.309	0.336	**0.279**

The best performing values are in bold.

process since foliage can easily be removed from the tree model without any changes to branch position. Data generation was semi-automated, with grids of 9 to 13 trees of seven species (Elm, Maple, Amur Cork, Black Alder, London Plane, Weeping Beech, and American Sycamore) randomly and procedurally generated using SpeedTree [37]. To generate images, the tree models were loaded into Unreal Engine 4 [38], where the UnrealCV plugin [39] was used to simulate an RGB-D sensor for automated data collection. For more realistic sensor emulation, for instance the Intel Realsense D435 series, simulated sensor noise was added [40] and the depth was cut off at ten meters. A separate ground-truth level was created where the foliage had been removed from the trees.

Five different helical trajectories, orbiting each tree to simulate drone flight, were used to capture image data from different views, 500 images were captured per tree. Several additional processing steps are executed to more accurately emulate the reduced quality of a real depth camera, see Section III-E for details.

D. Training

The simulated dataset is evenly divided into training, validation, and test sets across tree species, with eight to nine trees from each species for training (30 K images), two trees for validating, and two trees for testing (7 K images each). The validation and test set comprise images from previously unseen trees, although different trees of the same species are present in the training data. Additionally, two different species; Walnut and Shagbark Hickory, were held out entirely during training to evaluate the generalization abilities of the proposed network architectures.

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - t_{ij})^2$$

m number of rows in image

n number of columns in image

p_{ij} predicted depth value at pixel

t_{ij} ground truth depth value at pixel (1)

Several loss functions were evaluated for network training. They can be grouped into loss functions for binary segmentation (including cross entropy, dice loss, focal loss, and Tversky loss), and loss functions for regression (including mean square error (MSE) as in Eq. (1), root mean square error (RMSE), logarithmic RMSE (LogRMSE), smooth L1 loss (SmL1), and adaptive smooth L1 loss (AdSmL1)).

The following evaluation metrics as proposed and defined in [23] are used to compare the performance: MSE (1), RMSE, LogRMSE, absolute relative error (AbsRelErr), and squared relative error (SqRelErr).

The baseline segmentation network features 31 M parameters (see Table II and was trained using a two dimensional cross entropy loss function. For the regression U-Net and its transformer variant TransUNet, the MSE loss function was utilized during training to approximate the ground truth depth mask. Due to the added transformer layers, the number of trainable weights for the TransUNet is increased by a factor of 2.86 compared to the baseline model, totaling 89 M. The backbone of the ESANet is a segmentation U-Net encoder which was pre-trained on the NYUv2 dataset [41]. In addition to the final output layer, the model is supervised at each decoder module: 1 × 1 convolutions compute lower-resolution depth maps which are compared to down-sampled ground truth depth maps via MSE. The model features 47 M parameters, which is a 50% increase with regard to the baseline model.

All networks were implemented in PyTorch and trained on NVIDIA Titan X GPUs with 12 GB of RAM. The batch size was maxed out to run two jobs in parallel on a single GPU node. Training was performed using the Adam optimizer [42] with MSE loss for 20 epochs. To prevent model overfitting, the epochs with the lowest error on the validation set were chosen (10 epochs for the baseline, 20 epochs for the U-Net, 16 for the TransUNet, and 19 epochs for the ESANet).

E. Domain Adaptation

The difference between simulated and real data poses a major challenge when feeding the models with real-world data captured by a physical RGB-D sensor. Fig. 3 showcases this data disparity, which is generally caused by depth quality, lighting changes, camera noise, and interspecific (across species) as well as intraspecific (within species) variability.

To aid in the Sim-to-Real transfer, a series of transformations and augmentations are applied. Real-world depth images captured with an Intel RealSense device typically have larger regions of pixels in which the depth values show only little variation when compared to the simulation data. To model this, the simulation depth values are rounded to eight discrete values in the range [0 m, 10 m]. Additionally, real depth images are much less detailed, resulting in leaves and branches being nearly indistinguishable. To emulate this, Gaussian blur transforms are applied before and after discretization. To promote invariance towards varying lighting conditions and different color shades, a ColorJitter transform was applied as well. Finally, for input decorrelation and further data augmentation, the input tensor was randomly cropped to 20–100% of the original image size, and randomly horizontally flipped with 50% probability.

IV. Experimental Results

A numerical comparison of the different models on several simulated datasets is performed, including the two held-out tree species Walnut and Shagbark Hickory. We also conducted evaluation only on the wooden parts of the tree (skeleton) to remove the bias from predicting background pixels. The networks are evaluated with respect to several different metrics, and compared regarding model size and inference time. Finally, qualitative results on real-world data demonstrate the viability of the approaches in predicting pixel-wise depth of real trees.

A. Simulation Data Results

Table I shows the RMSE in meters for the models from Section III-B on different simulated datasets. The first row per model reports the metric value whereas the second corresponds to the percentage improvement over the baseline model. Lower RMSE values and larger negative percentages denote better performance, the best model per dataset is in bold.

The first column (Test Whole) in Table I reports the results on the whole images of the test dataset. Since background pixels may be a large proportion of the total image, models performing well only in predicting the background cut-off depth can potentially yield a very low RMSE on the entire image. As the primary interest lies in the depth prediction of the actual tree skeleton, the models were additionally evaluated only on the subset of pixels representing the wooden parts of the trees. The second column (Test Skel.) reports the RMSE on the union of pixels containing non-background predictions with pixels of the tree skeleton. The last two columns (Shagbark Skel. and Walnut Skel.) list the evaluation results on the skeletons of the two unseen tree species, Shagbark Hickory and Walnut respectively.

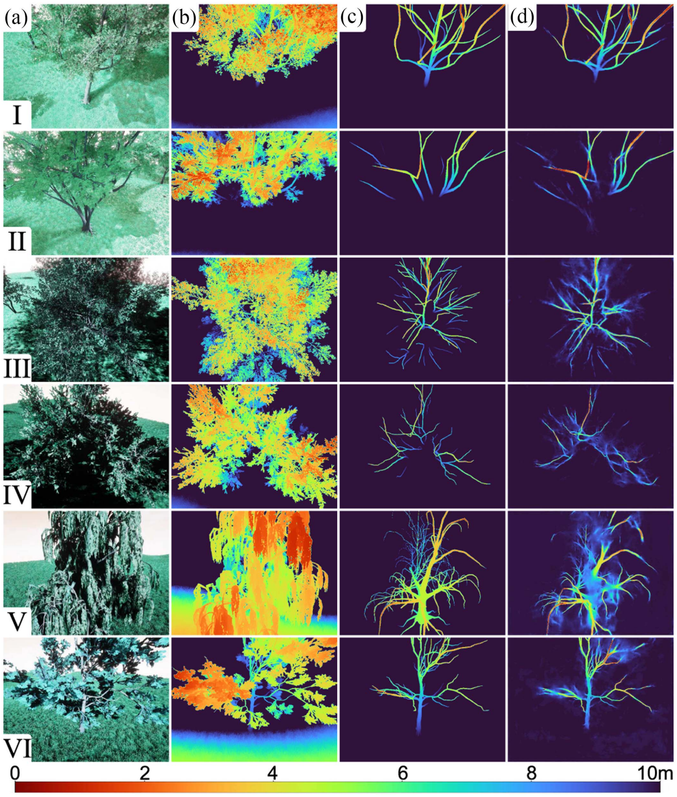

Fig. 4. Simulation inputs and outputs from the test dataset. Input RGB image (a), input depth image (b), with the target ground truth depth mask (c), and predicted depth masks of TransUNet (d). Tree species are Elm (I), Amur Cork (II), Black Alder (III), London Plane (IV), Weeping Beech (V), and American Sycamore (VI). Bottom colorbar denotes depth scale in meters.

Overall, all three models outperform the baseline algorithm by 29% up to 60% in terms of MSE as shown in Table I. However, the transformer extension of the U-Net (TransUNet) yields the best scores compared to the regression U-Net and the ESANet on the entire images as well as on the pixels of the tree skeleton.

Evaluating the models on only the tree skeleton exhibits even bigger improvements. As reported in the second column of Table I, all models reveal positive relative changes ranging from 52% (ESANet) to 54% (TransUNet) compared to the baseline. The overall improvements on the tree skeleton confirm the effectiveness of the models regarding the depth prediction of wooden structure in contrast to the mere depth prediction of background pixels. As the segmentation of the baseline algorithm is binary, most pixels of the background are predicted exactly, while the predictions of the other models might deviate from the exact background value set by the sensor cut-off distance.

To test the generalization ability of the models, two tree species, namely Shagbark Hickory and Walnut, were left aside. The third and fourth column of Table I show that the ESANet is able to generalize best to the unseen species Shagbark Hickory and Walnut, respectively. Nevertheless, also the errors for the remaining models are of the same order of magnitude as for the test dataset, demonstrating that all three models are able to cope remarkably well with out-of-distribution data. For all of the

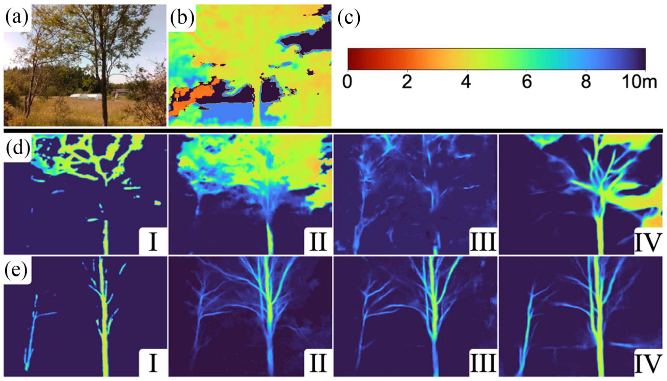

Fig. 5. Handpicked example output on a real-world image. (a) Input RGB image, (b) input depth images, (c) colormap for depth scale in meters and predicted network outputs (d) without domain transfer augmentation, and (e) with domain transfer augmentation: (I) Baseline, (II) U-Net, (III) TransUNet, and (IV) ESANet.

models the RMSE skeleton loss increases by maximally 10% for the out-of-distribution tree type Walnut.

Fig. 4 visualizes RGB and depth input data from simulated images of different trees, with the target ground truth depth map and the model prediction of the best performing TransUNet. While the depth predictions of sparse trees appear very sharp, occluded branches due to foliage lead to blurrier depth maps. Nonetheless, the model is able to recover branch locations which are not easily inferable for humans or heuristic-based algorithms. The supplementary video contains a panning side-view of the resulting point-cloud from the depth maps.

B. Network Comparison

Since onboard computational hardware on a UAV is restricted, the evaluation times of the network forward pass are of great interest. To estimate the model-specific inference times, the execution time of the forward pass on the test dataset for each model was measured when predicting a single image frame, repeated 15 times. The evaluation times were measured on server-mounted Nvidia Titan X GPUs. While this differs from onboard hardware performance, the relative difference between the networks will remain comparable. To measure time complexity, GFLOPs (Giga Floating-Point Operations per second), the number of multiply-accumulates to compute the model prediction on a single image, are exploited.

Table II shows the results of the 15 evaluations on the test set comprising 7 K images. One can observe that the ESANet performs significantly better in GFLOPs, with an 80% decrease from baseline and U-Net. The space complexity, represented by the number of parameters (in millions), are roughly the same for all models (31 M to 47 M), except the TransUNet (89 M) which has more than double the number of parameters due to the additional transformer layers. The last row reports the inference times in seconds for 15 frames. The fastest forward pass is achieved by the ESANet, in accordance with the GFLOPs analysis of the first row. Note that the baseline inference time includes the additional temporal overhead of masking the input sensor depth against the output binary segmentation, to create a depth map for comparison against the other regression networks.

TABLE III
RESULTS U-NET ON TREE SKELETON FOR DIFFERENT LOSS FUNCTIONS

Loss	MSE	RMSE	LogRMSE	AbsRelErr	SqRelErr
MSE	**0.011** -	**0.106** -	**0.140** -	**0.084** -	**0.017** -
SmL1	0.013 (+11%)	0.112 (+6%)	0.148 (+6%)	0.087 (+4%)	0.019 (+13%)
AdSmL1	0.030 (+167%)	0.173 (+63%)	0.228 (+63%)	0.153 (+81%)	0.050 (+189%)
MSEmCE	0.012 (+7%)	0.109 (+3%)	0.145 (+4%)	0.089 (+5%)	0.019 (+8%)

The best performing values are in bold.

TABLE IV
RESULTS U-NET ON TREE SKELETON FOR DIFFERENT MODEL INPUTS

Input	MSE	RMSE	LogRMSE	AbsRelErr	SqRelErr
RGB-D	**0.011** -	**0.106** -	**0.140** -	**0.084** -	**0.017** -
RGB	0.021 (+90%)	0.146 (+38%)	0.196 (+40%)	0.126 (+49%)	0.029 (+66%)
D	0.014 (+25%)	0.119 (+12%)	0.159 (+13%)	0.103 (+22%)	0.023 (+31%)

The best performing values are in bold.

C. Ablation Study

For computational reasons, all ablation studies were performed on a reduced dataset containing 18 K training samples and 4 K samples in the validation and test split each.

To determine suitable training loss functions, the U-Net architecture was trained as a representative model on the following loss functions: MSE loss (Eq. 1, a smooth version of the L1 loss called SmL1 (comparable with the Huber loss), an adaptive version of the aforementioned SmL1 loss called AdSmL1 (optimizes the threshold for switching between the L2 to the L1 loss), and a mixture of MSE and cross entropy MSEmCE (MSE for pixel-wise depth regression and cross entropy for segmentation of the binary masked ground truth and prediction). Table III presents the branch specific metrics (see Section III-D) after training the U-Net model for 20 epochs with the different losses. For clearer comparisons between the losses, the outputs are first normalized to be between 0 and 1 from initial outputs in the range of 0 to 10 m. The percentages beneath the absolute loss values are with respect to the MSE loss in the first row.

For all metrics considered in the study, training on MSE yields better results than any other loss function on the tree skeleton. Since the predictions on the wooden parts of the trees are the primary interest, MSE was opted for as the loss function for training the final architectures.

To determine the importance of the contribution of the color and depth input respectively, the performance of the U-Net architecture trained on RGB-D images was compared with U-Nets trained on RGB images or depth (D) input only. The branch specific metrics of the U-Net model trained for 20 epochs on the different inputs is presented in Table IV.

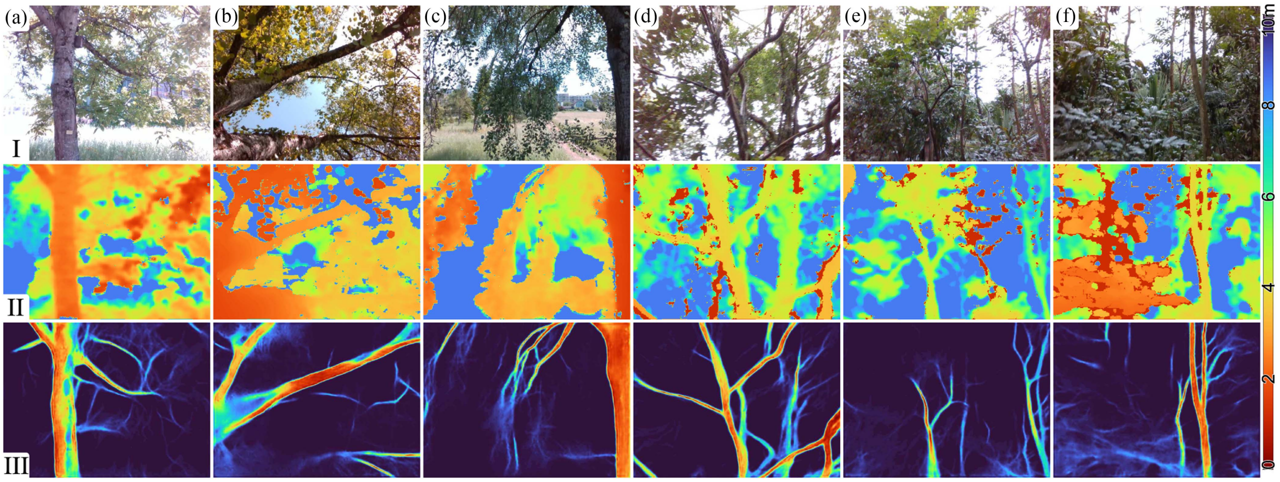

Fig. 6. Qualitative real-world data of the ESANet on trees from Swiss temperate forests (a)–(c) and trees from a masoala rainforest (d)–(f). The network receives RGB (I) and depth (II) as input and computes the pixel-wise depth of the occluded branches (III). Colorbar on the right shows depth scale in meters.

In comparison to regular RGB color cameras, the additional depth information can provide vital information on the 3D structure of the tree, improving network performance. Discarding the depth channel decreases performance by 38% (RMSE) up to 90% (MSE). On the other hand, dropping the three color channels leads to performance decreases of only 12% (RMSE) up to 31% (SqRelErr). Hence, the depth input alone is already sufficient to produce results close to the predictions based on RGB-D input images. We assume this to be due to the very accurate input depth images obtained in simulation, which contain most of the relevant information in high detail. As real depth sensors are not able to capture such high-quality frames, the importance of RGB input images is expected to increase for real-world applications.

D. Real-World Data

We provide a preliminary, qualitative, demonstration on real vegetation to demonstrate the feasibility of the approach on real-world data. To capture real-world data, images were taken with an Intel RealSense D435 depth sensor at a resolution of 640 × 480 pixels. Fig. 5 shows predictions for all models, trained on simulated data with and without domain transfer data augmentation. This clearly shows the importance of domain adaptation, considering that all models improve when using the augmented rather than the raw simulated data for the network training. While also true for the baseline, it still struggles with occlusions and predicts discontinuous branches (see Fig. 5(e-I)). Visually, the ESANet (IV) appears to be the most robust to the domain change, as its predictions are smoother and less noisy. This could be due to depth and color inputs being encoded separately, which detaches the two different modalities. The above strategy can be helpful since the depth data changes quite drastically for the real-world samples in contrast to the color information.

Results of the ESANet, trained on the augmented simulation dataset, on tree images from Swiss temperate forests and the Masoala Rainforest are shown in Fig. 6. The supplementary video contains a panning side-view of the resulting point-cloud of Figs. 5 and 6. Overall, the results show that it is possible to generalize to real-world samples and to predict feasible leafless depth maps of trees given RGB-D data, even under very significant changes in the domain from simulated training.

V. Conclusion

This letter demonstrates prediction of pixel-wise depth maps of partially occluded tree structures, given RGB-D input images. The networks are trained and evaluated on an extensive simulation dataset, with data post-processing performed to aid with domain adaptation on real-world data. When qualitatively evaluated on real-world images of trees from Swiss temperate forests and trees from the Masoala Rainforest at Zoo Zurich, the networks produce visually meaningful output depth maps. While predicting feasible outputs regarding the location of branches, the networks still struggle with input data that is very different from the training data.

Given that the models currently perform better on synthetic rather than real data, future work will focus on improving the real-world performance. One possible approach is using a more diverse dataset, such as including images from a wider range of distances, more varied tree species, increasing tree density to better reflect the natural clustering of trees, or incorporating unsupervised domain adaptation techniques [43]. To further improve depth predictions, one might potentially explore alternative loss functions such as topological losses [44] to impose constraints on the structural meaningfulness of the output. Additionally, the solution can be made more robust by reading sequential frames and outputting continuous segmentations, thus reducing dependence on lighting irregularities in singular frames and smoothing discontinuities between captures.

Since the envisioned use-case is UAV navigation in forest environments with dense vegetation, a natural next step would be to deploy the trained models on a drone in an online scenario to investigate the feasibility of using such architectures for path planning and obstacle avoidance. Assuming accurate output, this approach could then also be used to generate 3D models of occluded tree skeletons, by capturing images from around the tree. Considering the limited on-board computing hardware, optimizing performance and making the networks more lightweight should also be investigated. Potential applications of the system include collision avoidance in precision agriculture by detecting occluded branches for harvesting, pruning, or sensing, as well as robot navigation for search and rescue in dense forest environments, or for sensor placement and environmental monitoring.

Acknowledgment

The authors would like to thank Prof. Ce Zhang for providing access to computing clusters and helpful feedback, Davide Mambelli for initial work on the project, and Zoo Zürich for access to the Masoala Hall for data capture.

References

[1] C. Geckeler et al., "Supplementary materials: Learning occluded branch depth maps in forest environments using RGB-D images," 2023, doi: 10.3929/ethz-b-000634419.

[2] FAO and UNEP, *The State of the World's Forests 2020. Forests, Biodiversity and People*. Rome, Italy: FAO and UNEP, 2020.

[3] R. Pillay et al., "Tropical forests are home to over half of the world's vertebrate species," *Front. Ecol. Environ.*, vol. 20, no. 1, pp. 10–15, 2022.

[4] E. G. Brockerhoff et al., "Forest biodiversity, ecosystem functioning and the provision of ecosystem services," *Biodiversity Conservation*, vol. 26, no. 13, pp. 3005–3035, 2017.

[5] R. J. Zomer et al., "Global tree cover and biomass carbon on agricultural land: The contribution of agroforestry to global and national carbon budgets," *Sci. Rep.*, vol. 6, no. 1, 2016, Art. no. 29987.

[6] X. Zhou et al., "Swarm of micro flying robots in the wild," *Sci. Robot.*, vol. 7, no. 66, pp. eabm5954.

[7] X. Liu et al., "Large-scale autonomous flight with real-time semantic SLAM under dense forest canopy," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5512–5519, Apr. 2022.

[8] G. Charron et al., "The DeLeaves: A UAV device for efficient tree canopy sampling," *J. Unmanned Veh. Syst.*, vol. 8, no. 3, pp. 245–264, 2020.

[9] C. Geckeler and S. Mintchev, "Bistable helical origami gripper for sensor placement on branches," *Adv. Intell. Syst.*, vol. 4, no. 10, 2022, Art. no. 2200087.

[10] E. Aucone et al., "Drone-assisted collection of environmental DNA from tree branches for biodiversity monitoring," *Sci. Robot.*, vol. 8, no. 74, 2023, Art. no. eadd5762.

[11] S. Hamaza et al., "Sensor delivery in forests with aerial robots: A new paradigm for environmental monitoring," *IEEE IROS Workshop Perception, Planning Mobility Forestry Robot.*, 2020.

[12] C. Zhang et al., "Orchard management with small unmanned aerial vehicles: A survey of sensing and analysis approaches," *Precis. Agriculture*, vol. 22, no. 6, pp. 2007–2052, 2021.

[13] "Tevel." Accessed: May 03, 2023. [Online]. Available: https://www.tevel-tech.com/

[14] D. C. Schedl et al., "An autonomous drone for search and rescue in forests using airborne optical sectioning," *Sci. Robot.*, vol. 6, no. 55, pp. 1–11, 2021.

[15] E. Aucone et al., "Synergistic morphology and feedback control for traversal of unknown compliant obstacles with aerial robots synergistic morphology and feedback control for traversal of unknown compliant obstacles with aerial robots," 2023. [Online]. Available: https://www.researchsquare.com/article/rs-3262987/v1

[16] Y. Mulgaonkar et al., "Robust aerial robot swarms without collision avoidance," *IEEE Robot. Automat. Lett.*, vol. 3, no. 1, pp. 596–603, Jan. 2018.

[17] M. J. Islam et al., "Fast underwater image enhancement for improved visual perception," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3227–3234, Apr. 2020.

[18] O. Ronneberger et al., "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv.*, 2015, vol. 9, pp. 234–241.

[19] C. H. Kim and G. Kantor, "Occlusion reasoning for skeleton extraction of self-occluded tree canopies," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2023, pp. 9580–9586.

[20] J. Kierdorf et al., "Behind the leaves: Estimation of occluded grapevine berries with conditional generative adversarial networks," *Front. Artif. Intell.*, vol. 5, 2022, Art. no. 830026.

[21] Z. Chen et al., "Semantic segmentation for partially occluded apple trees based on deep learning," *Comput. Electron. Agriculture*, vol. 181, 2021, Art. no. 105952.

[22] S. Tejaswi Digumarti et al., "An approach for semantic segmentation of tree-like vegetation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 1801–1807.

[23] D. Eigen et al., "Depth map prediction from a single image using a multi-scale deep network," *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374, vol. 27.

[24] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2650–2658.

[25] I. Laina et al., "Deeper depth prediction with fully convolutional residual networks," in *Proc. Fourth Int. Conf. 3D Vis.*, 2016, pp. 239–248.

[26] W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 135–150.

[27] C. Wang et al., "A brief survey on RGB-D semantic segmentation using deep learning," *Displays*, vol. 70, 2021, Art. no. 102080.

[28] Y. Xing et al., "Coupling two-stream RGB-D semantic segmentation network by idempotent mappings," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1850–1854.

[29] J. Jiang et al., "Rednet: Residual encoder-decoder network for indoor RGB-D semantic segmentation," 2018, *arXiv:1806.01054*.

[30] S. Gupta et al., "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. Euro. Conf. Comput. Vis.*, 2014, pp. 345–360.

[31] D. Seichter et al., "Efficient RGB-D semantic segmentation for indoor scene analysis," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2021, May 2021, pp. 13525–13531.

[32] A. Loquercio et al., "Learning high-speed flight in the wild," *Sci. Robot.*, vol. 6, no. 59, 2021, Art. no. 5810.

[33] P. D. Petris et al., "RMF-Owl: A collision-tolerant flying robot for autonomous subterranean exploration," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2022, pp. 536–543.

[34] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," Feb. 2021, *arXiv:2102.04306*.

[35] D. Seichter et al., "Efficient RGB-D semantic segmentation for indoor scene analysis," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13525–13531.

[36] O. Ronneberger et al., "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[37] "SpeedTree." Accessed: Jan. 24, 2024. [Online]. Available: https://store.speedtree.com/

[38] "Unreal engine." Accessed: Jan. 24, 2024. [Online]. Available: https://www.unrealengine.com

[39] W. Qiu et al., "UnrealCV: Virtual worlds for computer vision," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1221–1224.

[40] M. S. Ahn et al., "Analysis and noise modeling of the Intel RealSense D435 for mobile robots," in *Proc. 16th Int. Conf. Ubiquitous Robots*, 2019, pp. 707–711, 2019.

[41] N. Silberman et al., "Indoor segmentation and support inference from RGBD images," in *Proc. Lecture Notes Comput. Sci.*, 2012, pp. 746–760.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.

[43] G. Csurka, R. Volpi, and B. Chidlovskii, "Unsupervised domain adaptation for semantic image segmentation: A comprehensive survey," 2021, *arXiv:2112.03241*.

[44] J. R. Clough et al., "A topological loss function for deep-learning based image segmentation using persistent homology," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8766–8778, Dec. 2022.