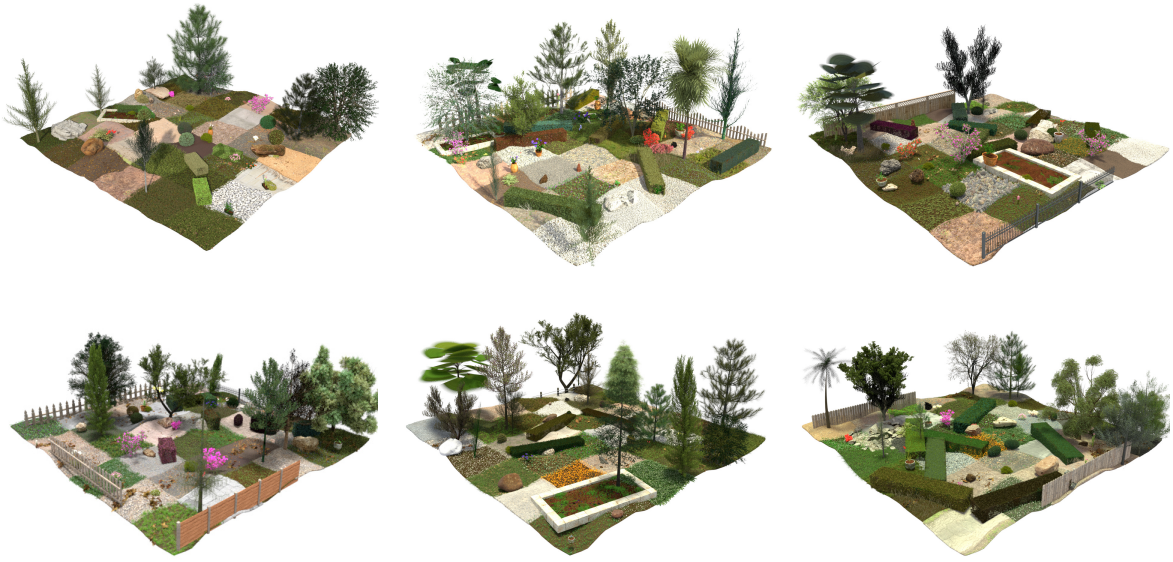
EDEN: Multimodal Synthetic Dataset of Enclosed GarDEN Scenes

Hoang-An Le[1] Thomas Mensink[1,2] Partha Das[1,3] Sezer Karaoglu[1,3] Theo Gevers[1,3]
[1]Computer Vision lab, University of Amsterdam
[2]Google Research Amsterdam [3]3DUniversum, Amsterdam
{h.a.le, p.das, s.karaoglu, th.gevers}@uva.nl, mensink@google.com



Abstract

Multimodal large-scale datasets for outdoor scenes are mostly designed for urban driving problems. The scenes are highly structured and semantically different from scenarios seen in nature-centered scenes such as gardens or parks. To promote machine learning methods for nature-oriented applications, such as agriculture and gardening, we propose the multimodal synthetic dataset for Enclosed garDEN scenes (EDEN). The dataset features more than 300K images captured from more than 100 garden models. Each image is annotated with various low/high-level vision modalities, including semantic segmentation, depth, surface normals, intrinsic colors, and optical flow. Experimental results on the state-of-the-art methods for semantic segmentation and monocular depth prediction, two important tasks in computer vision, show positive impact of pre-training deep networks on our dataset for unstructured natural scenes. The dataset and related materials will be available at `https://lhoangan.github.io/eden`.

1. Introduction

Synthetic data have been used to study a wide range of computer vision problems since the early days [1, 4, 26]. Compared to real-world imagery (RWI), computer-generated imagery (CGI) data provides allows for less expensive and more accurate annotation. Since the emergence of deep learning, synthetic datasets using CGI has become essential due to the data-hungry nature of deep learning methods and the difficulty of annotating real-world images. Most of the large-scale RWI datasets (with more than 20K annotated data points) are focusing on higher-level computer vision tasks such as (2D/3D) detection, recognition, and segmentation [11, 15, 16, 33, 39, 54]. In contrast, datasets for low-level image processing such as optical flow, visual odometry (KITTI [20, 36]) and intrinsic image decomposition (IIW [8], MIT [23], SAW [29]) are limited in the number of samples (around 5K annotated images).

CGI-based synthetic datasets [10, 19, 30, 35, 42, 44] provide more and diverse annotation types. The continuous progress of computer graphics and video-game industry

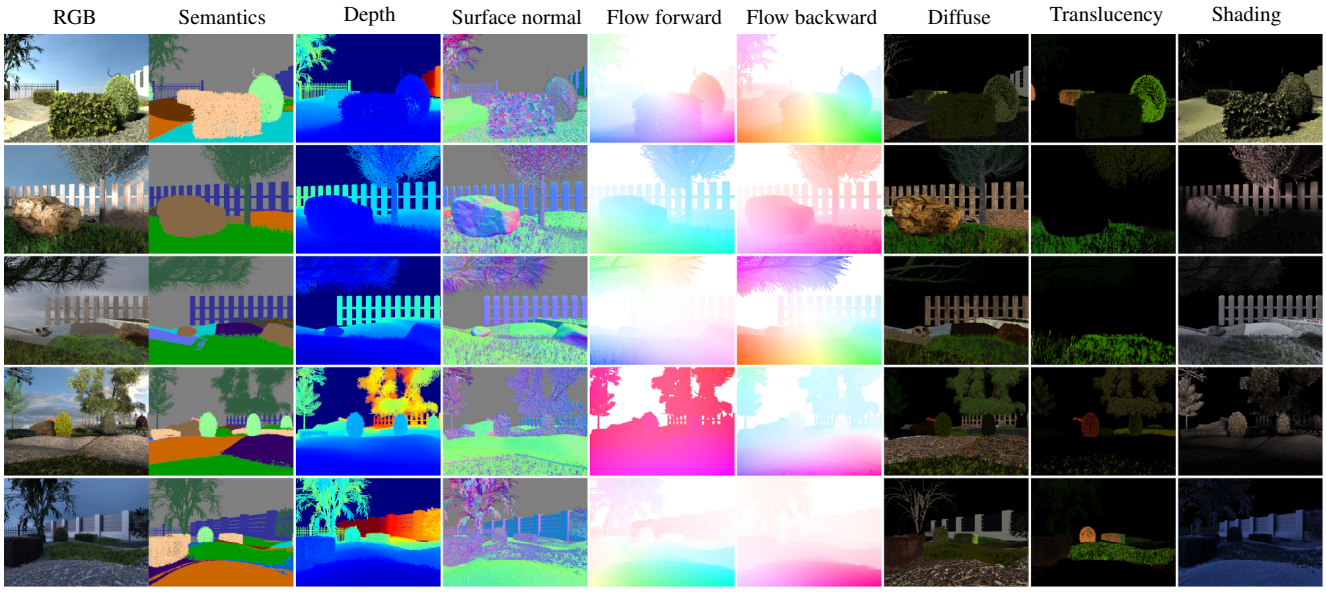


Figure 1. An overview of multiple data types provided in the dataset. The dataset includes data for both low- and high-level tasks such as (stereo) RGB, camera odometry, instant and semantic segmentation, depth, surface normal, forward and backward optical flow, intrinsic images (albedo, shading for diffuse materials, translucency)

results in improved photo-realism in render engines. The use of physics-based renderers facilitates the simulation of scenes under different lighting conditions (*e.g.* morning, sunset, nighttime). Information obtained by video-game pixel shaders [30, 42, 43] is of high-quality and can be used to train low-level computer vision tasks such as optical flow, visual odometry and intrinsic image decomposition.

Most of the existing datasets focus on car driving scenarios and are mostly composed of simulations of urban/suburban scenes. City scenes are structured containing objects that are geometrically distinctive with clear boundaries. However, natural or agriculture scenes are often unstructured. The gaps between them are large and required distinctive attentions. For example, there are only trails and no drive ways nor lane marks for travelling; bushes and plants are deformable and often entangled; obstacles such as small boulders may cause more trouble than tall grass.

To facilitate the development of computer vision and (deep) machine learning for farming and gardening applications, which involve mainly unstructured scenes, in this paper, we propose the synthetic dataset of Enclosed garDEN scenes (EDEN), the first large-scale multimodal dataset with >300K images, containing a wide range of botanical objects (*e.g.* trees, shrubs, flowers), natural elements (*e.g.* terrains, rocks), and garden objects (hedges, topiaries). The dataset is created within the TrimBot2020 project[1] for gardening robots, and have pre-released versions used in the 3DRMS challenge [48] and in several work [6, 7, 31].

[1] http://trimbot2020.webhosting.rug.nl/

In contrast to man-made (structured) objects in urban scenarios (such as buildings, cars, poles, *etc.*), the modelling of natural (unstructured) objects is more challenging. Natural objects appear with their own patterns and shapes, making a simplified or overly complex object easily recognized as unrealistic. Rendering techniques using rotating billboards of real photos may provide realistic appearances, but lack close-up geometrical features. Although synthetic datasets and video-games may offer natural objects and scenes, they often come with generic labels (e.g. tree, grass, and simple vegetation), since their focus is on the gaming dynamics. Therefore, objects in our dataset are developed using high-fidelity parametric models or CADs created by artists to obtain natural looking scenes. The object categories are selected for the purpose of gardening and agricultural scenarios to include a large variety of plant species and terrain types. The dataset contains relatively different lighting conditions to simulate the intricate aspects of outdoor environments. The different data modalities are useful for both low- and high-level computer vision tasks.

In addition to the new dataset itself, we provide analyses and benchmarks of the dataset on state-of-the-art methods of two important tasks in computer vision, namely semantic segmentation and depth prediction.

2. Related Work

2.1. Real-imagery datasets

To accommodate the emergence of deep learning and its data-demanding nature, many efforts have been spent on

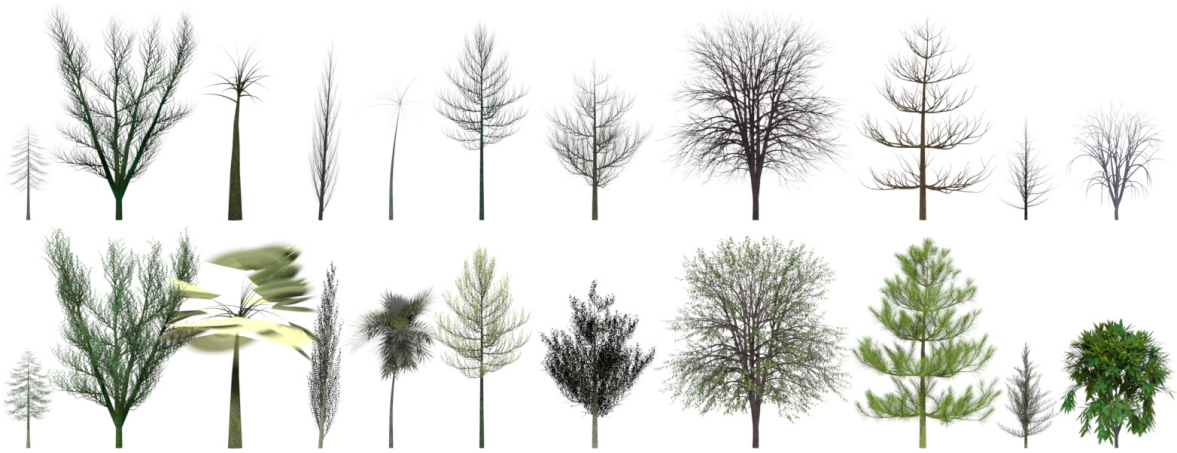


Figure 2. Sample tree models (top: tree stems, bottom: with leaves) for various tree species

creating large-scale generic datasets, starting with the well-known ImageNet [16], COCO [33], and Places [53]. These are real-world imagery (RWI) datasets with more than 300K annotated images at object and scene-level. Also in the domain of semantic segmentation, there are a number of datasets available such as Pascal-Context [37] (10,103 images, 540 categories) and ADE20K [54] (20,210 images, 150 categories).

Annotation is expensive. Lower-level task annotation is even more expensive. In contrast to the availability of large datasets for higher-level computer vision tasks, there are only a few RWI datasets for low-level tasks such as optical flow, visual odometry, and intrinsic image decomposition due unintuitive data annotation. Middlebury [3] and KITTI [20, 36] are the only datasets providing optical flow for real-world images, yet too small to train a deep network effectively. For intrinsic image decomposition, the MIT [23] dataset provides albedo and shading ground truths for only 20 objects in controlled lighting conditions, while IIW [8] and SAW [29] provide for up to 7K in-the-wild and indoor images. Indoor-scene datasets [46, 2, 11, 15] provide a larger number of images (up to 2.5M) and with more modalities (such as depth) than generic datasets. However, their goal is to provide data for 3D (higher-level) indoor computer vision tasks.

Outdoor scenes are subject to changing imaging conditions, such as lighting conditions, viewpoint, occlusion and object appearance, resulting in annotation difficulties. A number of methods are proposed focusing on scene understanding for autonomous driving [32, 9, 20, 36, 14, 39]. However, these datasets are limited in number of images and/or the number modalities. Mapillary [39, 50] is the most diverse dataset with varying illumination conditions, city views, weather and seasonal changes. Their focus is on semantic segmentation and place recognition. Large-scale multimodal datasets are restricted to synthetic data.

2.2. Synthetic datasets

Computer vision research uses synthetic datasets since the early days to study low-level tasks, *e.g*. optical flow [26, 1, 4]. Synthetic data provide cheaper and more accurate annotations. It can facilitate noise-free and controlled environments for otherwise costly problems [47, 38] or for intrinsic understanding [27] and proof of concept [28, 40].

Obviously, the quality of synthetic data and annotation depends on the realism of modelling and rendering algorithms. The development of computer graphic techniques has led to physics-based render engines and the improvement of photo-realistic computer-generated imagery (CGI). SYNTHIA [44] and Virtual KITTI [19] simulate various daylight conditions (morning, sunset), weather (rain, snow), and seasonal variations (spring, summer, fall, winter) for autonomous (urban) driving datasets. Datasets obtained from video-games [43, 42, 30] and movies [10, 35] show adequate photo-realism. These datasets provide not only dense annotations for high and low-level tasks, but also images are taken from multiple viewpoints and under different illumination/weather/seasonal settings. They have proven useful for training robust deep models under different environmental conditions [42, 30].

Datasets for outdoor scenes, real or synthetic, focus mostly on either generic or urban driving scenarios. They mainly consist of scenes containing man-made (rigid) objects, such as lane-marked streets, buildings, vehicles, *etc*. Only a few datasets contain (non-rigid) nature environments (e.g. forests or gardens [48, 49]).

CGI-based datasets rely on the details of object models, and computer-aided designed (CAD) model repositories, such as ShapeNet [12], play an important role in urban driving datasets [19, 44]. However, the models usually include rigid objects with low fidelity. Others focus on capturing the uniqueness of living entities, such as humans [34, 24], and trees [51, 25, 5] to generate highly detailed models with re-

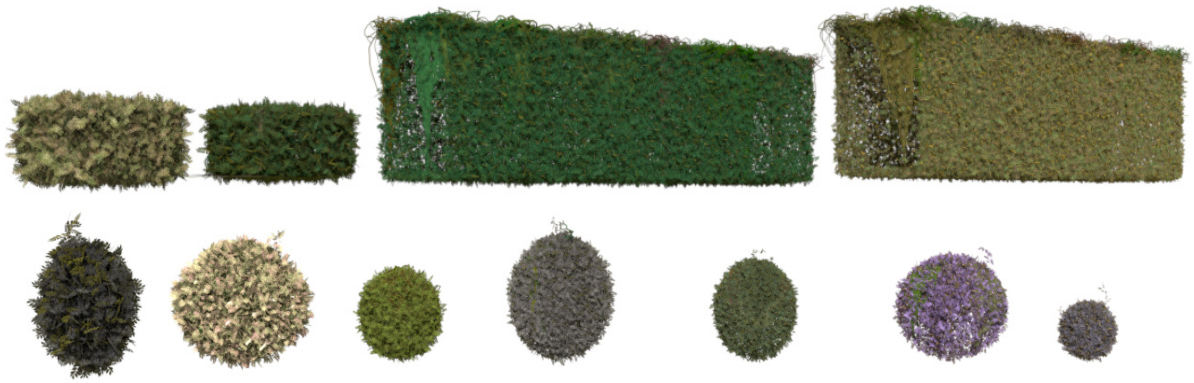


Figure 3. Sample models for hedges (top) and topiaries (bottom). The bushes can be generated with various sizes, leaf colors, and internal stem structures.

alistic variations. Synthetic garden datasets have been used in [7, 31, 48], albeit these datasets are relatively small and have just one or two modalities and are not all publicly available. In this paper, we use different parametric models, *e.g*. [51], to generate different botanical objects in an garden. We create multiple gardens with different illumination conditions, and extract multi-modal data (including RGB, semantic segmentation, depth, surface normals *etc*.) from each frame, yielding over 300K garden frames, which we will make publicly available.

3. Dataset Generation

We create synthetic gardens using the free and open-source software of Blender[2], and render using the physics-based Cycles render engine. Each garden consists of a ground with different terrains and random objects (generated with random parameters or randomly chosen from a pre-designed models). The modelling details of each component object and the rendering settings are presented in the following sections.

3.1. Modelling

To expand the diversity of objects and scenes, we propose to combine parametric and pre-built models in the generation process.

Trees We use the tree parametric model described in [51], implemented by the Blender Sapling Add-on[3]. A tree is constructed recursively from common predefined tree shapes (conical, (hemi-)spherical, (tapered) cylindrical, *etc*.) with the first level being the trunk. The parameters define the branch features such as length, number of splits, curvatures, pointing angles, *etc*., each with a variation range for random sampling. Leaves are also defined in a similar manner as stems, besides a fractional value determining their orientation to simulate phototropism. The model can generate different tree species such as quaking aspens, maples, weeping pillows, and palm trees. We use the parameter presets provided in the sampling add-on and Arbaro[3] (Figure 2). Totally there are 19 common tree species.

Bushes Hedges and topiaries are generated by growing an ivy adhering to a rectangular or spherical skeleton object using the Ivy Generator[3], implemented by the Blender IvyGen add-on[3]. An ivy is recursively generated from a single root point by forming curved objects under different forces including a random influence to allow overgrowing, an adhesion force to keep it attached to the trellis, a gravity pulling down, and an up-vector simulating phototropism. The add-on is known for creating realistic-looking ivy objects (Figure 3). We use more than 20 leaf types with different color augmentation for both topiaries and hedges.

Landscapes and terrain The landscape is created from a subdivided plane using a displacement modifier with the Blender cloud gradient noise which is a representation of Perlin noise [41]. The modifier displaces each sub-vertex on the plane according to the texture intensity, creating the undulating ground effect. The base ground is fixed at 10x10 square meters, on which are paved the terrain patches of 1x1 square meter. Each patch is randomly assigned to one of the terrain types, including grass, pebble stones, gravels, dirt and pavement.

The grass is constructed using Blender particle modifier which replicates a small number of elemental objects, known as particles, over a surface. We use the grass particles provided by the Grass Essentials[3], and the Grass package[3], containing expert-designed realistic-looking grass particles. There are more than 30 species of grass, *e.g*. St. Augustine grass, bahiagrass, centipedegrass, *etc*. and weed, *e.g*. dandelions, speedwell, prickly lettuce, *etc*. Each species has up to 49 model variations. The appearance of the grass patch is controlled via numerical parameters, such as freshness, brownness, wetness, trimmed levels, lawn stripe shape for mowed field, *etc*. Illustrations for different grass and weed species are shown in Figure 4 (top).

[2] `blender.com`, GPL GNU General Public License version 2.0

[3] See the supplementary for the reference link

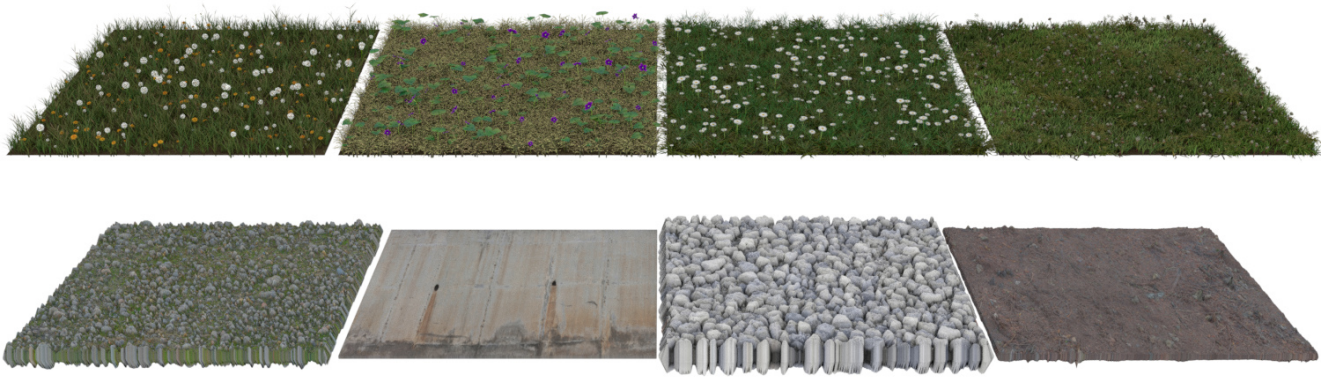


Figure 4. Sample tiles of different terrain types: grass with weed (*top*), gravel, pavement, pebble stones, dirt (*bottom*). The grass and weed species are chosen and combined randomly.

Figure 5. Illustration for scene appearance changed according to different illumination conditions.

The other terrains are designed using textures from the Poliigon collection[3] of high quality photo-scanned textures. Illustrations are shown in Figure 4 (bottom). Each texture contains a reflectance, surface normal, glossy, and reflection map with expert-designed shaders for photo-realism. The resulted landscapes can be seen on the first page.

Environment Lighting in our dataset is created by 2 sources, a sun lamp and a sky texture. A sun lamp is a direct parallel light source, simulating an infinitely far light source. The source parameters include direction, intensity, size (shadow sharpness), and color. A sky texture provides the environmental background of rendered images and a source of ambient lights. We use the Pro-Lighting: Skies package[3] composing of 95 realistic equirectangular HDR sky images of various illuminations. The images are manually chosen and divided into 5 scenarios, namely clear (sky), cloudy, overcast, sunset, and twilight. We also use 76 HDR scenery images[3] to create more various and complex backgrounds, some with night lighting, coined scenery. An example of lighting effects is shown in Figure 5.

Pre-built models To enhance the model variations in the dataset, we also include models prebuilt from different artists, including rocks[3], flowers[3], garden assets such as fences, flower pots[3], *etc*.

Garden construction For each garden, 2 to 4 types are sampled of each grass species, as well as for tree, terrains, bushes, rocks, flowers, and garden assets. The number of tree, bush, and obstacle instances are uniformly sampled from the closed intervals $[5, 17]$, $[10, 24]$, and $[3, 17]$, respectively; each instance is randomly assigned with one of the corresponding species. The random seeds in parametric models allow plants of the same species to contain internal variations. The objects are distributed at random places around the garden, avoiding overlapping each others, while the fences, if any, are placed at the 4 edges.

3.2. Rendering

Camera setup We follow the real-world camera setup in the 3DRMS challenge to create a ring of 5 pairs of virtual stereo cameras with angular separation of 72° (Figure 7), baseline of 0.03 meters. Each camera has a virtual focal length of 32mm on a 32mm wide simulated sensor. The rendered resolution is set to VGA-standard of 480x640 pixels. The camera intrinsic matrix is as follows:

$$\mathbf{K} = \begin{bmatrix} 640 & 0 & 320 \\ 0 & 640 & 240 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

We generate a random trajectory for the camera ring for each illumination variation of each garden model. The speed is set to about 0.5m/s, frame rate of $10 fps$, simulating a trimming robot in a garden. To improve the variability, the camera ring is set to randomly turn after a random number of steps and avoid running through the objects. The turning angles are also randomized to include both gradual and

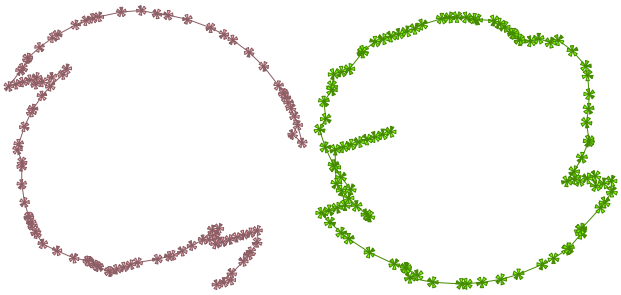


Figure 6. Examples of the generated trajectories used in the rendering process. The 5 pairs of cameras, illustrated by different color shades, are randomly moved, turned, and self-rotated while avoiding obstacles in a garden.

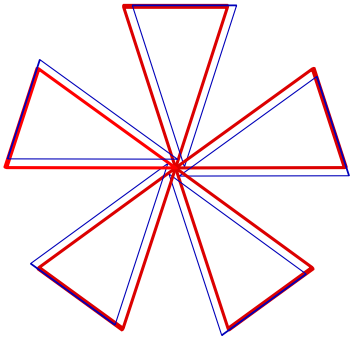


Figure 7. The camera system: a ring of 5 pairs of stereo cameras at 72° angular separation

abrupt angles. The trajectory lengths are set to be at least 100 steps. The examples are shown in Figure 6.

Render engine Blender Cycles is a probabilistic ray-tracing render engine that derives the color at each pixel by tracing the paths of light from the camera back to the light sources. The appearances of the objects are determined by the objects' material properties defined by the bidirectional scattering distribution function (BSDF) shaders, such as diffuse BSDF, glossy BSDF, translucent BSDF, *etc*.

Scene aspects such as geometry, motion and the object material properties are rendered into individual images before being combined into a final image. The formation of a final image $I(\mathbf{x})$ at position $\mathbf{x}$ is as follows[4]:

$$f_g(\mathbf{x}) = g_{\text{color}}(\mathbf{x})(g_{\text{direct}}(\mathbf{x}) + g_{\text{indirect}}(\mathbf{x})), \quad (2)$$

$$I(\mathbf{x}) = f_D(\mathbf{x}) + f_G(\mathbf{x}) + f_T(\mathbf{x}) + B(\mathbf{x}) + E(\mathbf{x}), \quad (3)$$

where D, G, T, B, E are respectively the diffuse, glossy, transmission, background, and emission passes. D_{color} is the object colors returned by the diffuse BSDF, also known as albedo; D_{direct} is the lighting coming directly from light sources, the background, or ambient occlusion returned by the diffuse BSDF, while D_{indirect} after more than one reflection or transmission off a surface. Similar are G and T with glossy and transmission BSDFs. Emission and background are pixels from directly visible objects and environmental textures. The intermediate image contains at each pixel the corresponding data or zeros otherwise.

[4] *c.f*. Blender 2.83 Manual, last access July 2020

Split	train (127)	test (20)	
		full	20K
clear	74,913	10,035	3,333
cloudy	73,785	10,030	3,378
overcast	73,260	10,015	3,349
sunset	73,715	10,040	3,250
twilight	73,990	10,045	3,369
total	369,663	50,165	20,000

Table 1. Number of images per scene and split; the number of models are in parentheses

All the computations are carried out in the linear RGB space. Blender converts the composite image to sRGB space using the following gamma-correction formula and clipped to $[0, 1]$ before saving to disk:

$$\gamma(u) = \begin{cases} 12.92u & u \leq 0.0031308 \\ 1.055u^{1/2.4} - 0.055 & \text{otherwise} \end{cases} \quad (4)$$

In our dataset, besides the RGB stereo pairs and cameras' poses, we provide the images from intermediate stages, namely albedo, shading, glossy, translucency, *etc*. for the left camera. As the modelling and rendering are physics-based, the intermediate images represent different real-life modalities, such as geometry, motion, intrinsic colors, *etc*. Examples are shown in Figure 1.

4. Experiments

In this section, the goal is to quantitatively analyze the newly created dataset to assess its realism and usability. The evaluation is performed via two proxy tasks: semantic segmentation and monocular depth estimation.

We split the dataset into training (127 models, 369,663 monocular images) and test set (20 models, 60,195 images). To speed up the evaluation process, we uniformly sample 20K images from the full test set. The statistics are shown in Table 1. The sample list will also be released together with the dataset.

4.1. Semantic segmentation

For semantic segmentation, we use the state-of-the-art DeepLabv3+ architecture with Xception-65 backbone [13]. Three aspects of the dataset are analyzed, namely (1) training size, (2) lighting conditions, and (3) compatibility with real-world datasets. The label set is from the 3DRMS challenge [45, 48]: void, grass, ground, pavement, hedge, top-

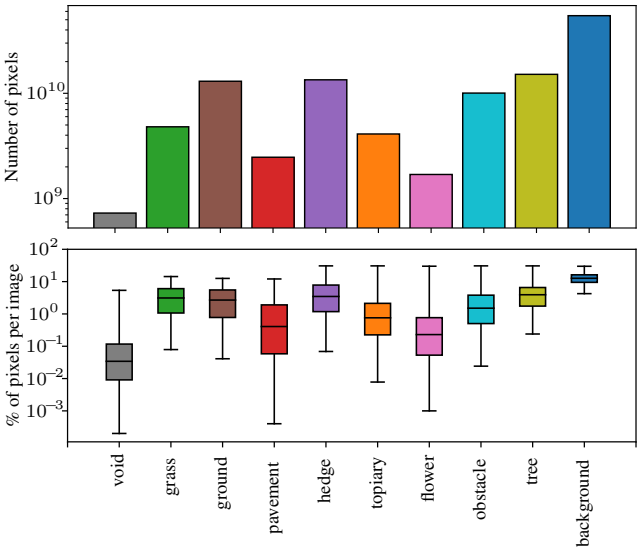


Figure 8. Number of pixels per class in the dataset (*top*) and distributions in the images (*bottom*). The boxplot shows the 1st, 2nd (median) and 3rd quartile of the number of pixels in each frame, with the whisker value of 1.5. *background* includes sky and object outside of the garden, while *void* indicates unknown pixels, which should be ignored.

Sampling	test	
	full	20K
25%	75.71	75.89
50%	79.42	79.52
100%	81.96	82.09

Table 2. Performance with respect to different training size and at 2 test splits. The network performance increase when being trained on higher number of images. The performance on the reduced test set is on par with the full set.

iary, flower, obstacle, tree, background. Background contains the sky and objects outside of the garden, while void indicates unknown objects to be ignored. The label statistics are shown in Figure 8. We also follow the network's training setup and report mean intersection-over-union (mIOU). The results are shown in percentage and higher is better.

Training and testing size We first show the benefit of an increasing training set and the performance on the full and reduced test set. The results are shown in Table 2. The performance increases when the training size increases, showing the advantage of having large amount of training samples. The evaluation on the reduced test set is similar to the full set. Thus, unless mentioned otherwise, the test20K split will be used for evaluation in later experiments.

Lighting conditions Our dataset contains the same garden models in various lighting conditions, allowing in-depth analysis of illumination dependency of different methods for different tasks. In this section we perform

Training	test					
	clear	cloudy	overcast	sunset	twilight	20K
clear	**76.10**	*76.91*	76.43	72.23	75.91	72.03
cloudy	75.09	***77.59***	77.16	72.37	76.40	**72.30**
overcast	65.75	75.52	***78.41***	70.76	74.63	70.22
sunset	73.21	75.76	77.17	**74.44**	*77.28*	71.84
twilight	66.19	72.86	76.21	70.55	***78.16***	68.83

Table 3. Cross-lighting analysis. Each row corresponds to a model trained on the specific lighting condition (highest values are in italics), while each column corresponds to the results evaluated on the specific subset (highest values are in boldface). Lighting-specific training gives better results on the specific lighting, while the results in the cross-lighting vary depending on the conditions of the training and test images.

cross-lighting analysis on semantic segmentation. We conduct lighting-specific training of the networks, and evaluate the results on each lighting subset of the full test set, as well as the reduced test set. The results are shown in Table 3. All experiments are trained with the same epoch numbers.

For almost all of the categories, training on the specific lighting produces the best results on that same categories. This is not surprising, as networks always perform the best on the most similar domains. In general, training with cloudy images gives the highest performance, while twilight are the lowest. This could be due to relatively bright images and less intricate cast shadows in cloudy scenes, in contrast to the mostly dark and color cast twilight images.

Compared to training with all the full training set in Table 2, the results from training with lighting-specific images are generally lower and near to the 25% subset. This agrees to the training size conclusion as the lighting-specific training sets account only for around 20% of the data. Testing on the same lighting gives a boost in performance, similarly to training with double data size.

Real-world datasets Semantic segmentation requires a method to recognize different objects from the appearance models learned during training. Therefore, it indicates the closeness of training data to the testing domain. By analyzing the features learned from EDEN on real images of unstructured natural scenes, the results indicate the realism level of our dataset. To that end, the real-imagery datasets 3DRMS [45, 48] (garden scenes, 221 annotated real images for train, 268 for validation, 10 classes), Freiburg forest [49] (forested scenes, 228 annotated real images for train, 15 for validation, 6 classes) are used for evaluation.

The baselines include (1) the network pre-trained on combination of generic datasets, ImageNet [16], COCO [33], and augmented PASCAL-VOC 2012 [17], and (2) the network pre-trained on ImageNet and urban driving scene dataset Cityscapes [14]. The encoder part is set to the pre-trained weights provided by the authors [13], while the

Pre-training	test	
	3DRMS	Freiburg
Generic	24.35	41.33
Cityscapes	31.11	50.08
EDEN	**34.55**	**52.45**

Table 4. Adaptability of features pre-trained on different datasets to unstructured natural real-world scenes. The network pre-trained on EDEN outperforms all other alternative approaches on both 3DRMS and Freiburg test sets.

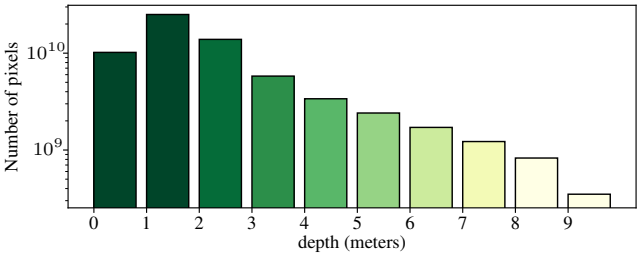


Figure 9. Number of pixels per depth range in the dataset. Each range is a left-inclusive half-open interval.

decoder is finetuned using the train split of each target set for 50K iterations. The results are shown in Table 4.

The networks using the features learned from EDEN out-perform all alternative approaches. Both 3DRMS and Freiburg features highly unstructured scenes with mostly deformable and similar objects found in the nature, drastically different from the generic images and structured urban scenes. The results show the realism of our dataset to natural scenes and its benefit on training deep networks. The results on Freiburg test are higher than on 3DRMS due to the relatively simpler and general class labels (*e.g*. trails, grass, vegetation, and sky) compared to the garden-specific label sets of 3DRMS (*e.g*. hedges, topiaries, roses, tree, *etc*.).

4.2. Monocular depth prediction

Monocular depth prediction is an ill-posed problem. Often the ambiguity is mitigated by learning from a large-scale depth-annotated dataset [18, 52] or imposing photometric constraints on image sequences using relative camera poses [21, 22] As camera pose prediction can be formulated using depth constraint, the depth-pose prediction problems can be combined in a self-supervised learning pipeline.

Synthetic datasets are favored for being noise-free, which can act as controlled environments for algorithm analysis. In this section, we use EDEN to test different monocular depth prediction networks. Specifically, we examine the effectiveness of using supervised signals in learning depth prediction for unstructured natural scenes. The statistics of the depth in the dataset are shown in Figure 9.

We show the results of training state-of-the-art architectures using different ground truth information, namely depth and camera pose. To that end, the 2 methods, VNL [52] and MD2 [22] are used. VNL is trained with supervised depth, while MD2 can be trained with ground truth camera pose or in self-supervised manner. Both are trained using the schedules and settings provided by the respective authors. The results are shown in Table 5. We show the 3 error metrics (rel, log10, rms, seall is better) after the original work and also include the reported results of the respective methods on the KITTI dataset for comparison.

Method	Supervised	Dataset	rel	log10	rms
MD2	None	KITTI	0.115	0.193	4.863
VNL	Depth	KITTI	0.072	0.117	3.258
MD2	None	EDEN	0.438	0.556	1.403
MD2	Pose	EDEN	0.182	0.220	0.961
VNL	Depth	EDEN	0.181	0.083	1.061

Table 5. Performance of different SOTA methods for monocular depth prediction when trained on KITTI and EDEN. The gap is larger between unsupervised and supervised methods on EDEN, showing the necessity of having supervised signals for learning unstructured scenes. The errors on EDEN are generally higher than on KITTI, showing the more challenging scenes of the (unstructured) dataset.

Generally, supervised method always produce better results than their self-supervised counterpart as shown by the smaller errors. The difference are less for the KITTI dataset compared to EDEN. As KITTI contains mostly rigid objects and surfaces, it is simpler to obtain predicted camera poses with high accuracy. On the other hand, camera pose prediction for self-supervised learning on EDEN are unreliable because of deformable objects and their similarities. The errors are, therefore, also higher for supervised methods on EDEN than on KITTI, showing the more challenging dataset. KITTI has higher RMS numbers due to the larger depth ranges, approximately 80m vs. 15m of EDEN.

5. Conclusion

The paper presents EDEN, a large-scale multimodal dataset for unstructured garden scenes, and provides baseline results and analysis on two popular computer vision tasks, namely the problems of semantic segmentation and monocular depth prediction.

The experiments show favorable results of using the dataset over generic and urban-scene datasets for nature-oriented tasks. The dataset comes with several computer vision modalities and is expected to stimulate applying machine and deep learning to agricultural domains.

Acknowledgements: This work is performed within the TrimBot2020 project funded by the EU Horizon 2020 program No. 688007.

References

[1] J K Aggarwal and N Nandhakumar. On the computation of motion from sequences of images-A review. *Proceedings of the IEEE*, 76(8):917–935, 1988.

[2] I Armeni, A Sax, A.∼R. Zamir, and S Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, feb 2017.

[3] Simon Baker, Daniel Scharstein, J P Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision (IJCV)*, 92(1):1–31, mar 2011.

[4] J L Barron, D J Fleet, S S Beauchemin, and T A Burkitt. Performance of optical flow techniques. *International Journal of Computer Vision (IJCV)*, 12(1):43–77, feb 1994.

[5] R Barth, J IJsselmuiden, J Hemming, and E J van Henten. Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset. *Computers and Electronics in Agriculture*, 144:284–296, 2018.

[6] Anil S Baslamisli, Partha Das, Hoang-An Le, Sezer Karaoglu, and Theo Gevers. ShadingNet: Image Intrinsics by Fine-Grained Shading Decomposition. *ArXiv e-prints*, 2019.

[7] Anil S. Baslamisli, Thomas T. Groenestege, Partha Das, Hoang-An Le, Sezer Karaoglu, and Theo Gevers. Joint Learning of Intrinsic Images and Semantic Segmentation. In *European Conference on Computer Vision (ECCV)*, jul 2018.

[8] S Bell, K Bala, and N Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (SIGGRAPH)*, 2014.

[9] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and Recognition Using Structure from Motion Point Clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 44–57, 2008.

[10] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *Proceedings of the European Conference on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, oct 2012.

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)*, 2017.

[12] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *European Conference on Computer Vision (ECCV)*, pages 833–851. Springer International Publishing, 2018.

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 3, 2016.

[15] 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.

[16] J Deng, W Dong, R Socher, L.-J. Li, K Li, and L Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[17] M Everingham, L Van∼Gool, C K I Williams, J Winn, and A Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, jun 2010.

[18] H Fu, M Gong, C Wang, K Batmanghelich, and D Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[19] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[21] 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 (CVPR)*, 2017.

[22] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into Self-Supervised Monocular Depth Prediction. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, oct 2019.

[23] Roger Grosse, Micah K. Johnson, Edward H. Adelson, and William T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[24] Jian Han, Sezer Karaoglu, Hoang-An Le, and Theo Gevers. Object Features and Face Detection Performance: Analyses with 3D-Rendered Synthetic Data. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 2020.

[25] Charlie Hewitt. *Procedural Generation of Tree Models for Use in Computer Graphics*. PhD thesis, Cambridge Trinity College, 2017.

[26] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, aug 1981.

[27] B Kaneva, A Torralba, and W T Freeman. Evaluation of image features using a photorealistic virtual world. In *Pro-*

ceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2282–2289, 2011.

[28] B Kicanaoglu, R Tao, and A W M Smeulders. Estimating small differences in car-pose from orbits. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[29] Balazs Kovacs, Sean Bell, Noah Snavely, and Kavita Bala. Shading Annotations in the Wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] P Krahenbuhl. Free Supervision from Video Games. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2955–2964, 2018.

[31] Hoang-An Le, Anil S. Baslamisli, Thomas Mensink, and Theo Gevers. Three for one and one for three: Flow, Segmentation, and Surface Normals. In *Proceedings of the British Machine Vision Conference (BMVC)*, jul 2018.

[32] B Leibe, N Cornelis, K Cornelis, and L Van Gool. Dynamic 3D Scene Analysis from a Moving Vehicle. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, Cham, 2014. Springer International Publishing.

[34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics (SIGGRAPH)*, 34(6), oct 2015.

[35] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[36] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pages 3061–3070, 2015.

[37] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, Alan Yuille, Raquel Urtasun, and Alan Yuille. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 891–898, 2014.

[38] Matthias Mueller, Neil Smith, and Bernard Ghanem. A Benchmark and Simulator for UAV Tracking. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, pages 445–461, Cham, 2016. Springer International Publishing.

[39] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[40] Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. Transformable Bottleneck Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, nov 2019.

[41] Ken Perlin. An Image Synthesizer. *SIGGRAPH Computer Graphics*, 19(3):287–296, jul 1985.

[42] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for Benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[43] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[44] G Ros, L Sellart, J Materzynska, D Vazquez, A. M. Lopez, German Ros;, Laura Sellart;, Joanna Materzynska;, David Vazquez;, and Antonio M. Lopez;. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[45] Torsten Sattler, Radim Tylecek, Thomas Brox, Marc Pollefeys, and Robert B Fisher. 3D Reconstruction meets Semantics – Reconstruction Challenge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1–7. ICCV Workshops, oct 2017.

[46] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images, 2012.

[47] G R Taylor, A J Chosak, and P C Brewer. OVVV: Using Virtual Worlds to Design and Evaluate Surveillance Systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[48] Radim Tylecek, Torsten Sattler, Hoang-An Le, Thomas Brox, Marc Pollefeys, Robert B Fisher, and Theo Gevers. The Second Workshop on 3D Reconstruction Meets Semantics: Challenge Results Discussion. In Laura Leal-Taixé and Stefan Roth, editors, *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 631–644, Cham, 2019. Springer International Publishing.

[49] Abhinav Valada, Gabriel Oliveira, Thomas Brox, and Wolfram Burgard. Deep Multispectral Semantic Scene Understanding of Forested Environments using Multimodal Fusion. In *International Symposium on Experimental Robotics (ISER)*, 2016.

[50] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2020.

[51] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. *SIGGRAPH '95 - Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 119–128, 1995.

[52] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019.

[53] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning Deep Features for Scene Recognition using Places Database. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 487–495. Curran Associates, Inc., 2014.

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.