A Cross-Season Correspondence Dataset for Robust Semantic Segmentation

Måns Larsson[1] Erik Stenborg[1,3] Lars Hammarstrand[1] Mark Pollefeys[2,4] Torsten Sattler[1]
Fredrik Kahl[1]
[1]Chalmers University of Technology [2]Department of Computer Science, ETH Zurich
[3] Zenuity [4] Microsoft

Abstract

In this paper, we present a method to utilize 2D-2D point matches between images taken during different image conditions to train a convolutional neural network for semantic segmentation. Enforcing label consistency across the matches makes the final segmentation algorithm robust to seasonal changes. We describe how these 2D-2D matches can be generated with little human interaction by geometrically matching points from 3D models built from images. Two cross-season correspondence datasets are created providing 2D-2D matches across seasonal changes as well as from day to night. The datasets are made publicly available to facilitate further research. We show that adding the correspondences as extra supervision during training improves the segmentation performance of the convolutional neural network, making it more robust to seasonal changes and weather conditions.

1. Introduction

Semantic segmentation is the task of assigning a class label to each pixel in an image and is one of the fundamental problems in computer vision. Semantic segmentation has also been used to integrate higher-level scene understanding into other computer vision problems, *e.g.*, dense 3D reconstruction [6, 13, 14, 23, 28, 31, 55, 56, 61], SLAM [7, 35], Structure-from-Motion [3], 3D model alignment [15, 16, 73], and location recognition [1, 42, 60].

Visual localization is the problem of estimating the camera pose of an image [8, 34], typically from a set of 2D-3D matches between image pixels and 3D scene points. In the context of long-term visual localization [58, 63–65, 74], semantics are proven useful to be able to handle variations in scene geometry and appearance, *e.g.*, due to seasonal changes. These methods are based on the idea that the semantic meaning of a scene part is invariant to such changes. Semantics are thus used to establish the 2D-3D matches required for pose estimation when matching solely based on image appearance fails.

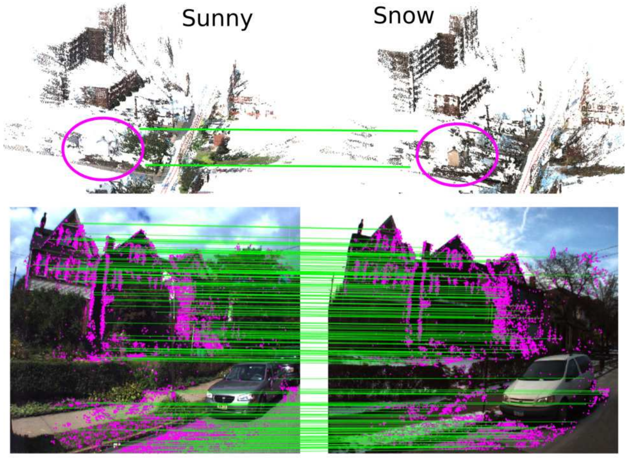

Figure 1. 2D-2D matches between images taken under different conditions, established using matches against a 3D point cloud.

As of yet, however, the assumption that the same semantic segmentation can be reliably reproduced under different conditions does not hold. This is mainly due to that the labeled datasets used to train the semantic segmentation algorithm itself are limited to only a few conditions. As the pixel-level annotations are performed by hand, adding more training data is both time consuming and expensive [17, 44]. However, even rather noisy segmentations improve localization performance compared to not using semantic information [58, 65]. This naturally leads to the question whether it is possible to build a feedback loop: Can semantic segmentation algorithms be improved via visual localization? Can this in turn lead to more robust localization results?

As part of estimating the camera pose of an image taken under one condition against a 3D model built from images, a set of 2D-3D matches is established. Usually, the reference 3D model is built from images [54]. Thus, the 2D-3D correspondences lead to a set of 2D-2D matches between images taken under different conditions, as depicted in Fig. 1. In this paper, we show that these pixel-level matches can be used to improve semantic segmentation algorithms.

In detail, this paper makes the following contributions: **1**) A 2D-2D match between two images taken under different conditions provides a constraint on the training process, namely that these two pixels should receive the same label. We use this insight to formulate a loss function that can be added to the training process, without the need to modify the architecture of the segmentation algorithm. **2**) We show that the required set of correspondences can be generated with little human supervision, albeit without ground truth labels. This is in stark contrast to creating labeled training datasets, where significant human effort is required [49]. We make our cross-season correspondence dataset publicly available[1]. **3**) We show that using our correspondence-based loss, together with a few coarsely annotated images required to prevent trivial solutions, can lead to significant improvements in segmentation quality in the context of changing imaging conditions. The improvements are especially significant if the base training set covers only a single condition.

2. Related Work

Semantic Segmentation. The performance of semantic segmentation algorithms has seen a large increase during the last few years based on the advances of deep neural networks. The seminal work of Long *et al.* [37] showed that convolutional neural networks (CNNs), initially trained for classification, can be transformed to fully convolutional networks (FCNs) for semantic segmentation. Follow up work has improved upon FCNs by for example enlarging the receptive field [10, 72], incorporating higher level context [76] or fusing multi-scale features [11, 50]. In addition, the combination of FCNs and structure models, such as conditional random field (CRF), have been thoroughly studied, either as a post-processing step [10] or as part of the network [33, 36, 77], enabling end-to-end training.

Training these networks requires a large amount of annotated images, which for semantic segmentation can be costly and time-consuming to acquire. In response many weakly supervised approaches have been suggested utilizing labels in the form of bounding boxes [19, 29, 45], image level tags [45, 47, 48, 62] or points [4]. Similarly, we improve the segmentation performance of a FCN by utilizing supervision that requires less manual effort to acquire than pixel-level annotations. However, instead of using weaker, but still manually annotated, labels we propose the use of data that can be acquired in a semi-automatic fashion.

Domain Adaptation. Domain adaptation methods aim at learning a model that performs well in the target domain, given that there are only available annotations in the source domain. Early work includes [30, 52] that transforms the features to either a domain invariant feature space [52] or the source feature domain [30]. Several works have focused on domain adaptation for CNN models [21, 38, 39, 67]. These methods learn models that produce domain invariant features, either by aligning the target and feature distributions [38, 39] or by using an adversarial training setup, encouraging domain confusion [21, 67].

Recently, several domain adaptation methods for dense prediction tasks have been presented [12, 26, 53, 66, 68, 69, 78]. Most of these [12, 25, 26, 43, 53, 66, 78] use synthetic datasets, *e.g.*, [49, 51], enabling automatic generation of large amount of annotated synthetic images. The methods presented in [25, 43, 53] all use some form of image translation method such as to transform the source images into the target domain before performing segmentation. Another common approach is to use an adversarial training setting as in [26, 68, 69] where the network is encouraged to produce features that fool a domain discriminator.

We are interested in increasing the performance of our segmentor on images different from the source domain. Specifically, the different conditions and seasons included in our correspondences can be seen as target domains. Rather than using unsupervised domain adaption, we use 3D geometric consistency as a supervisory signal. Our cross-season correspondence datasets facilitate the adaption of the segmentation method across the different target domains, removing the need to rely on purely unsupervised domain adaptation methods.

Semantic 3D Mapping. Semantic 3D reconstruction approaches [6, 13, 14, 23, 28, 31, 41, 55, 56, 61] uses semantic image segmentations to aid the reconstruction process. They use a voxel volume to represent the scene and jointly reason about geometric and semantic occupancy. Using semantics typically leads to more consistent and complete 3D models. These 3D models can be projected into the images to obtain refined semantic segmentations [28]. However, the semantic reconstruction process is significantly more complex compared to the multi-view stereo process we use to obtain correspondences. Methods exists that jointly predict depth and semantics [20, 32] or use depth information to aid semantic segmentation [24]. Yet, they still rely on labelled data. We are not aware of any work using pixel correspondences obtained via 3D models to create additional constraints for semantic segmentation.

Datasets for 3D and semantics. Large databases for indoor [9, 18, 70] and outdoor scenes [22, 27] provide both semantics and 3D geometry and can thus also be used to semantically annotated images via geometry [71]. Yet, we are not aware of any such dataset that captures different seasonal and illumination conditions. In this sense, our work closes a gap in the literature.

3. Semantic Correspondence Loss

The rationale behind using 2D-2D image correspondences is that the CNN initially, being pre-trained on a

[1] https://visuallocalization.net

large-scale dataset such as Cityscapes [17], performs well on images taken during favourable conditions (*i.e*. similar conditions as in the training set). We can then use the correspondence data to enforce labeling consistency between images captured during favourable conditions and images captured during challenging conditions. To this end we define and test two different loss functions based on the hinge loss and the cross-entropy loss that will encourage labeling consistency. The losses are designed for a CNN where the value of intermediate feature layers can be extracted and where the final output is an estimate of the probability distribution over the class labels for each input pixel.

We denote the content of one sample from the cross-season correspondence dataset as $(I^r, I^t, \boldsymbol{x}^r, \boldsymbol{x}^t)$. Here I^r is an image from the reference traversal, I^t an image from the target traversal, and $\boldsymbol{x}^r$ as well as $\boldsymbol{x}^t$ are the pixel positions of the matched points in the reference and target images, respectively. The reference traversal is chosen as the one with images captured during the most favourable image condition. Note that the reference images are taken from the same traversal while the target images vary between all other available traversals. The correspondence loss function $\mathcal{L}_{corr}$ will be a sum over all such samples

$$\mathcal{L}_{corr} = \sum_{(r,t)} l(I^r, I^t, \boldsymbol{x}^r, \boldsymbol{x}^t) \ , \tag{1}$$

where l is a hinge loss l_{hinge} or a cross-entropy loss l_{CE} as presented below.

Let $\boldsymbol{d}_x \in \mathbb{R}^F$ denote a feature vector of the segmentation CNN of length F at pixel position x. This can be either the last layer of the network, where F equals the number of classes or an earlier, intermediate feature layer. We define the correspondence hinge loss l_{hinge} for one sample as

$$l_{hinge} = \frac{1}{N} \sum_{i=1}^{N} \max\left(0, m - \frac{\boldsymbol{d}_{x_i^r}^T \boldsymbol{d}_{x_i^t}}{\|\boldsymbol{d}_{x_i^r}\| \|\boldsymbol{d}_{x_i^t}\|}\right) \ , \tag{2}$$

where m is a margin parameter and N is the number of corresponding points. The loss will encourage the feature vectors $\boldsymbol{d}_{x_i^r}$ and $\boldsymbol{d}_{x_i^t}$ to align up to a certain angle depending on m. In the experiments, we have found empirically that setting $m = 0.8$ (approximately 37°) works well.

For the correspondence cross-entropy loss l_{CE}, we begin by taking the argument of the maximum of the final feature map, *i.e*. the most likely class, of the reference image. By describing the most likely class for a pixel at position x_i using an one-hot encoding vector $\boldsymbol{c}_{x_i}$, the loss can be written as

$$l_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{c}_{x_i^r}^T \log\left(\boldsymbol{d}_{x_i^t}\right) \ , \tag{3}$$

where $\log(\cdot)$ is taken element-wise. The loss will encourage the pixels in the target image to have the same labels as the corresponding pixels in the reference image.

During training, we minimize a loss consisting of one term for the fully supervised data based on standard cross-entropy $\mathcal{L}_{sup}$ as well as one correspondence term $\mathcal{L}_{corr}$. The resulting overall loss is $\mathcal{L} = \mathcal{L}_{sup} + \lambda \mathcal{L}_{corr}$, where λ is a weighting term for the impact of the correspondences.

4. A Cross-Season Correspondence Dataset

This section describes the creation and the content of the cross-season correspondence dataset. Each sample of the dataset contains two nearby images taken during different seasons or weather conditions as well as a set of 2D-2D point correspondences between the images. The correspondences are automatically established using geometric 3D consistency between the two points. Geometry is typically more stable than for instance photometric information across the different conditions.

A visualization of a few samples can be seen in Fig. 2. Using the datasets presented in [54] as a starting point, we create two correspondence datasets. The datasets from [54] used were originally based on the CMU Visual Localization dataset [2] and the the RobotCar dataset [40] respectively.

The creation of the correspondence dataset can be divided into four main steps. Firstly, the camera poses for all images in all conditions need to be calculated in a common coordinate system. In our case these were kindly provided by the authors of [54]. Secondly, a dense 3D point cloud of the surrounding geometry is created, individually for each condition and traversal. Thirdly, the 3D point clouds are matched across the conditions. Since the point clouds share the same coordinate system, this matching can be done using the position of the 3D points. This removes the need for feature descriptors which might change substantially across the different conditions. Lastly, given the 3D point cloud matches, the pixel positions for the 2D-2D correspondences in each image can be calculated using the known camera positions. Each step will be detailed separately for the two datasets below.

4.1. CMU Seasons Correspondence Dataset

The CMU Visual Localization dataset [2] was collected over a period of 12 months in Pittsburgh, USA. A vehicle with two cameras facing forward/left and forward/right drove a 8.5 km long route through central and suburban areas. We use the camera poses from twelve traversals during different seasons and weather condition available in the CMU Seasons dataset [54]. The camera poses have been calculated using bundle adjustment of SIFT points and some manually annotated image correspondences across different traversals, for further details we refer to [54]. This method gives accurate camera positions, where the position error is estimated to be under 0.10 m.

To create the dense point cloud for each traversal, we use the Multi-View Stereo (MVS) pipeline presented in [59].

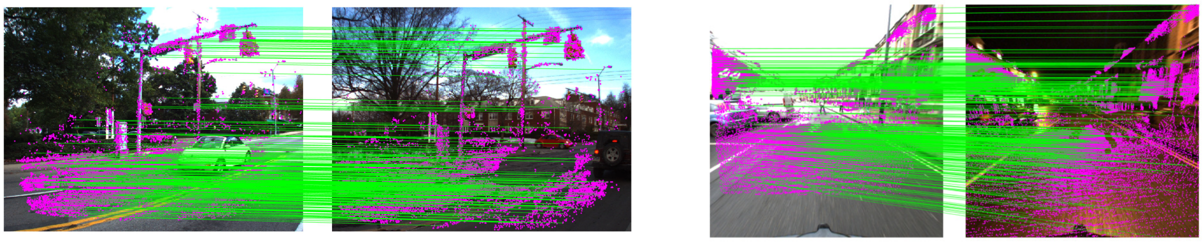

Figure 2. Visualization of a sample from our cross-season correspondence datasets. Left: CMU, right: RobotCar. Each purple point marks the pixel position of a correspondence in the respective images. Green lines are drawn to the matching point in the other image. Note that lines are only drawn for every 50 point pair to avoid cluttering the images.

The MVS pipeline pipeline works in two steps. Firstly, depth and normal information is estimated using geometric and photometric information. Secondly, the depth and normal maps are fused forming a dense point cloud. This was done using the software Colmap [57, 59] with default settings. Examples of 3D point clouds from the CMU Seasons dataset can be seen in Fig. 1.

To find correspondences between images from two different traversals, we perform geometric matching between the two corresponding 3D point clouds. Given a reference and a target traversal, we take the two corresponding dense 3D point clouds and match points using the Euclidean distance: For each point in the first point cloud, we search for the nearest neighboring point in the other point cloud and vice versa. We establish a correspondence between two points if they are mutual nearest neighbors.

For the resulting matches, we check from which cameras each matched point was triangulated during MVS, giving us the cameras where the point is visible. We then go through each camera pair, where one is from the reference and one from the target traversal and investigate the number of common 3D points that can be seen in the cameras. For pairs with at least 500 common points and where the distance between the two cameras is less than $0.5\ m$, we do an additional pruning step to get rid of poor matches. The pruning is necessary since the first 3D matching step does not enforce any constraint that the points need to be close too each other. In addition, enforcing a closeness constraint for each camera pair enables us to use a distance threshold that depends on the distance between the point and the cameras. This is beneficial since 3D points close to the camera usually are reconstructed with greater precision. For a match to be kept, the distance between the two points must be below a threshold that depends on the distance between the point and the camera as follows

$$||X_1 - X_2|| < \kappa D \ . \tag{4}$$

Here X_1 and X_2 are the positions of the matched 3D points in the reference and target traversals, respectively, D is the distance from the camera center to X_1 and κ is a unitless parameter set to 0.01. This means that points that are 10 m from the camera need to be closer than 0.1 m to be kept as a correspondence. Table 1 provides a summary of the content of the final CMU Cross-Season Correspondence dataset.

Condition	Image pairs	Average N
Sunny + Foliage	3185	14361
Sunny + Foliage	2200	17696
Cloudy + Foliage	3312	17711
Sunny + Foliage	3620	18373
Overcast + Mixed Foliage	3300	13770
Low Sun + Mixed Foliage	3286	15441
Low Sun + Mixed Foliage	3384	16081
Cloudy + Mixed Foliage	2729	14111
Low Sun + No Foliage + Snow	2022	19060
Overcast + Foliage	1728	20090

Table 1. Statistics of the CMU Cross-Season Correspondence dataset. Each row shows the condition, the number of image pairs as well as the average number of correspondences per image pair for each traversal. Note that there are several traversals with the same condition.

4.2. Oxford RobotCar Correspondence Dataset

The original RobotCar dataset [40] was gathered using an autonomous vehicle that traversed a 10 km route in Oxford, UK during 12 months. Similarly as for the CMU dataset, we use the camera poses from ten traversals during different seasons and weather conditions available in the RobotCar Seasons dataset provided by the authors of [54]. For the reference traversal these camera poses were initialized using a GPS/INS system and refined by iteratively triangulating 3D points and performing bundle adjustment. For the other traversals, the camera poses were calculated using 3D points clouds built from the measurements of the LIDAR scanners mounted on the vehicle. The LIDAR point clouds for each traversal were aligned to the LIDAR point cloud of the reference traversal using the Iterative Closest Point algorithm [5] and manual corrections when necessary.

The images included in the RobotCar Seasons dataset are recorded using three synchronized global shutter Point Grey Grasshopper2 cameras mounted to the left, rear, and right of the car. Unfortunately the image quality is poor in general. A lot of the images are overexposed and there is a lot of motion blur present. In addition there is also a lot of image noise for the night time images. This makes the MVS pipeline that we used for the CMU dataset produce

Condition	Image pairs	Average N
Dawn	772	59158
Dusk	646	50159
Night	646	52238
Night + Rain	780	43066
Summer + Overcast	809	51722
Winter + Overcast	671	54466
Rain	683	53276
Snow	823	57578
Sun	681	48150

Table 2. Statistics of the RobotCar Cross-Season Correspondence dataset. Each row shows the condition, the number of image pairs s as well as the average number of correspondences per image pair for each traversal. Note that there are several traversals with the same condition.

point clouds with too few points to be of any use for us. We instead use the LIDAR point clouds available in the original RobotCar dataset. Since we know the pose of the multi-camera system at each timestamp, as well as the poses of the individual cameras and LIDAR sensors on the car, we can transform the LIDAR point clouds into the coordinate system of the cameras. We then determine which points are visible in which cameras in a separate step. To determine which points are visible in each camera the depth of points projected close to each other in the image is compared.

The matching of the 3D points and pruning of the correspondences are done in the same way as for the CMU Seasons dataset. However, since we do not have as many images for the RobotCar Seasons dataset, we use a larger threshold for the distance between camera pairs, specifically 2.0 m. Table 2 provides a summary of the content of the final RobotCar Cross-Season Correspondence dataset.

5. Implementation Details

During the training of the CNN, we minimize the loss $\mathcal{L}$ described in Section 3. As a starting point, we use the PSPNet [76] network pretrained on the Cityscapes dataset [17]. In addition to the Cityscapes training images, we also add a few coarsely annotated images from the CMU and RobotCar Seasons datasets, respectively. Some examples of these annotations are shown in Fig 3. This is necessary to keep the CNN from learning the trivial solution where the same class is predicted for all pixels on the CMU and RobotCar images while still producing good segmentations for the Cityscapes images. Note that only training images with fine annotations from the Cityscapes dataset are used.

We also add an on-the-fly correspondence refinement step, where all correspondences that have their pixel positions in the reference image classified as one of the non-stationary classes are removed. The classes concerned are person, rider, car, truck, bus, train, motorcycle, and bicycle. For a correspondence, if the pixel in the reference image is classified as a non-stationary class it means one of two things: The pixel actually depicts a non-stationary object and has been incorrectly added to the correspondence dataset, *i.e.*, there is no guarantee that the corresponding pixel in the target image has the same semantic class. The other explanation is that the CNN incorrectly classified the pixel. Adding the loss for said correspondence would not be useful for any of these scenarios. Additionally, we use a warm-up period of 500 iterations before the correspondence loss is added. This ensures that the CNN has produces reasonable segmentations for the reference images.

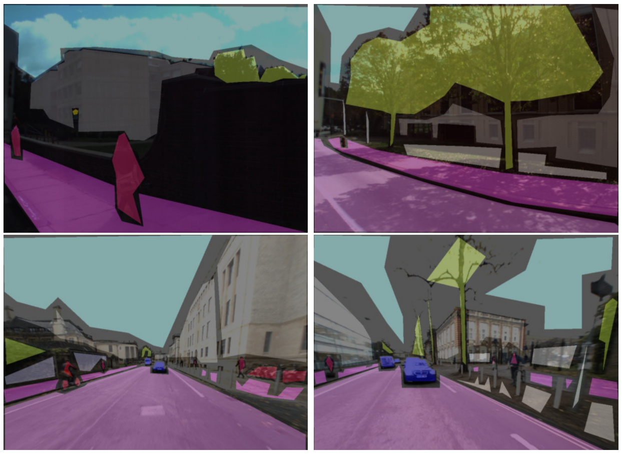

Figure 3. Examples of the manually labeled images added to the training set. The top row shows images for the CMU Seasons dataset (66 images in total) while the bottom row shows images for the RobotCar Seasons dataset (40 images in total).

The optimization method used during training is Stochastic Gradient Descent with momentum and weight decay. During all experiments the learning rate was set to $2.5 \cdot 10^{-5}$, while the momentum and weight decay were set to 0.9 and 10^{-4}, respectively. In addition, the loss was scaled with $\frac{1}{1+\lambda}$ to keep the total loss weight to unity. Due to GPU memory limitations we train using a batch size of one. We train the networks for at least 30000 iterations and use the weights that got the best mean Intersection over Union (mIoU) on the validation set. For the RobotCar and CMU Seasons validation and test sets, we take the mean over only classes present in the respective dataset when calculating the mIoU. The number of validation images for the CMU Seasons and Robotcar were 25 and 27 respectively. The corresponding numbers for the test sets were 33 and 27. The training and evaluation is implemented in PyTorch [46] and the code is publicly available [2].

All evaluation and testing is done in patches on the original image scale only. The patch size is 713×713 pixels and the patches are extracted from the image with a step size of 476 pixels in both directions. The output of the network is paired with an interpolation weight map that is 1 for the 236×236 center pixels of the patch and drops off linearly to 0 at the edges. For each pixel the weighted mean, using the interpolation maps as weights, of all patches that contain it

[2] https://github.com/maunzzz/cross-season-segmentation

λ	CE	Hinge$_C$	Hinge$_F$
0.01	63.3	62.0	62.2
0.05	64.4	62.6	62.1
0.1	65.6	63.2	**62.3**
0.5	65.0	62.3	61.9
1.0	**67.2**	**64.0**	60.6
2.0	66.0	63.3	59.5
5.0	66.7	62.6	59.4

Table 3. Parameter study of λ for the loss $\mathcal{L} = \mathcal{L}_{sup} + \lambda\mathcal{L}_{corr}$ (*cf*. Section 3) on the CMU dataset. The correspondence losses included are cross-entropy loss l_{CE} and hinge loss l_{hinge} applied to the final and second to final features, Hinge$_C$ and Hinge$_F$, respectively. All values are mIoU (in %) for the CMU validation set.

is used to produce the pixel's class scores. The motivation behind the interpolation is that the network generally performs better at the center of the patches, since there is more information about the surroundings available there.

6. Experimental Evaluation

In this section, we present the results of using the Cross-Season Correspondence Dataset to train a CNN for segmentation. The two main points of interest are **a**), how does the Cross-Season Correspondence Dataset influence the segmentation performance on the images within the dataset, *i.e*., the CMU and RobotCar images and **b**), how does using the dataset influence the generalization performance of the segmentor to other datasets. To investigate **a**), we manually annotate a set of test images from the RobotCar and CMU datasets taken from areas that are not included in the correspondence datasets. To answer **b**), we use the WildDash dataset [75]. The WildDash dataset is designed to evaluate the robustness of segmentation methods and contains many different and challenging images.

We also investigate the effect of training with correspondences together with different amount of annotated training images. To this end we do experiments both using just the Cityscapes dataset [17] training set as well as, on top of that, adding the training set from the Mapillary Vistas dataset [44]. The Cityscapes training set contains 2975 annotated images taken under favorable weather conditions and in similar environments while the Vistas training set contains 18000 images from a diverse set of environments, seasons and weather conditions [44]. As the Vistas dataset contains many more annotated classes than the Cityscapes dataset, we only consider the subset of classes which are in both datasets and treat the rest as unlabeled during training. For all segmentation experiments, we investigate three correspondence loss functions: for the first we use the correspondence cross-entropy loss l_{CE} applied to the final features. For the other two, we apply the correspondence hinge loss l_{hinge} to the final and second to final layer, respectively.

	Extra	Corr	CMU		RobotCar	
			CMU	WD	RC	WD
CS			31.2	16.4	22.2	16.4
	✓		73.6	37.0	45.8	25.4
	✓	CE	**79.3**	**39.6**	53.8	27.8
	✓	Hinge$_C$	72.4	37.9	50.6	25.2
	✓	Hinge$_F$	75.3	38.7	**55.4**	**27.9**
CS + Vistas			77.4	49.3	46.8	**49.2**
	✓		82.8	51.5	51.2	46.4
	✓	CE	**85.9**	52.9	**59.4**	48.5
	✓	Hinge$_C$	82.2	**54.9**	59.0	47.9
	✓	Hinge$_F$	84.0	54.5	58.7	45.6

Table 4. Segmentation results for the models trained on the CMU correspondence data (left) and the RobotCar correspondence data (right). Results for the CMU test set, the RobotCar (RC) test set and the WildDash (WD) validation set are shown in terms of mIoU (in %). For the bottom five rows the Vistas training set was used in addition to Cityscapes (CS). Column one marks if the extra training annotations from the CMU/RobotCar dataset were used. Column two specifies the correspondence training loss used, *i.e*., correspondence cross-entropy loss (CE) applied to the final layer and hinge loss applied to the final and second to final features, Hinge$_C$ and Hinge$_F$, respectively.

Parameter Study. The parameter λ specifies the trade-off between the fully supervised cross-entropy loss ($\mathcal{L}_{sup}$) for the annotated training set and the correspondence loss ($\mathcal{L}_{corr}$). A higher λ means that more emphasis is put on minimizing the correspondence loss compared to the fully supervised loss. To investigate the impact of λ, we perform a parameter study which is summarized in Table 3. The results presented are the mIoU on the CMU validation set. As can be seen from the table, the best choice of λ for the correspondence cross-entropy loss (CE) and the hinge loss on the final features (Hinge$_C$) is $\lambda = 1.0$ while for the hinge loss applied to the second to final feature layers (Hinge$_F$) it is $\lambda = 0.1$. We hence choose these values for the remaining experiments.

Segmentation. Table 4 summarizes the results of the segmentation experiments on the CMU as well as the RobotCar datasets. For the networks trained on the CMU dataset, the mIoU on the CMU test set and the WildDash validation set is presented for several baselines as well as networks trained with the correspondence dataset. For the networks trained on the RobotCar dataset the RobotCar test set is used instead of the CMU test set. Comparing the two baselines, there is a large difference in performance between the network trained with both Cityscapes and Vistas compared to the one trained with Cityscapes only. This is to be expected since the Vistas dataset has much more diversity when it comes to seasons and weather conditions, enabling a network trained on it to generalize well to the CMU or RobotCar test set.

When investigating the effect of the correspondence dataset, it is reasonable to compare the network trained with

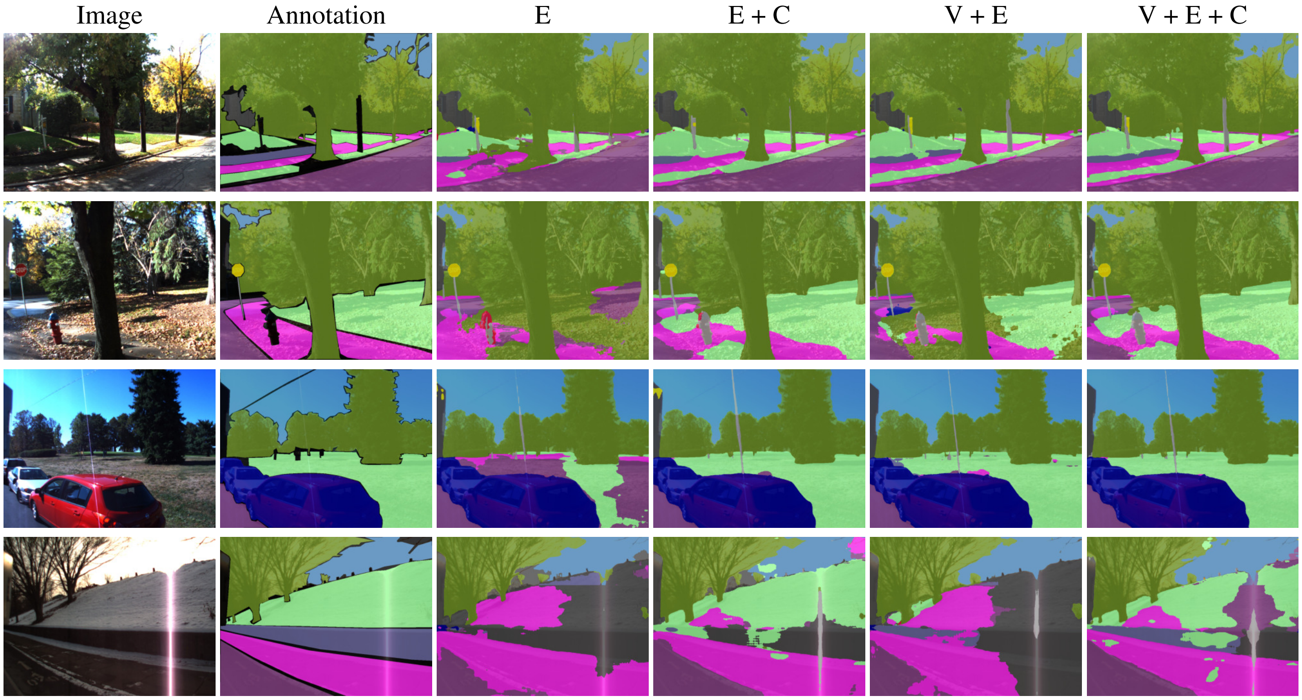

Figure 4. Qualitative results on the CMU test set. Four different networks are compared, the notation used is: E: trained with extra CMU annotations, C: trained with correspondence data, V: trained with Vistas training set. The most notable performance difference when adding correspondences are for areas that are visually different between seasons. This can be seen for the terrain areas covered in leaves for row two as well as the area of snow in row four. The image in row four is especially challenging since it contains a lot of snow as well as an apparent lens glare. Yet, the networks trained with correspondences still manages to get part of the snow patch correctly labeled.

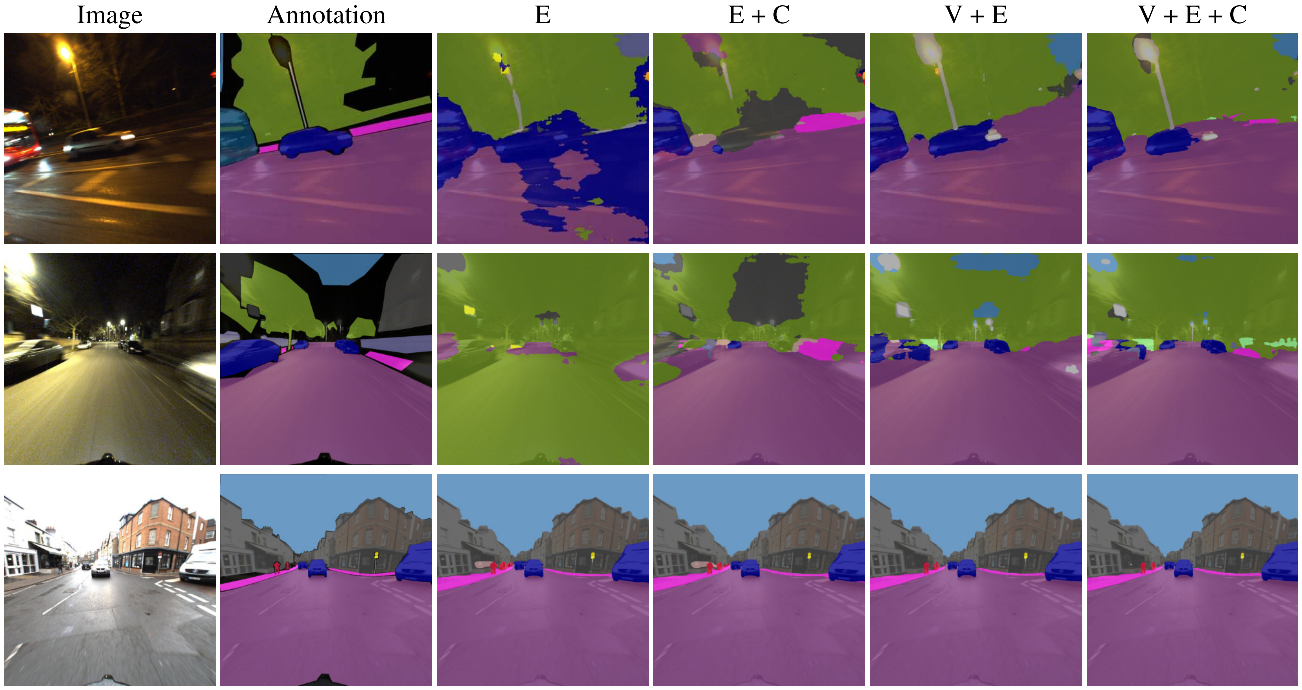

Figure 5. Qualitative results on the RobotCar test set. Four different networks are compared, the notation used is: E: trained with extra RobotCar annotations, C: trained with correspondence data, V: trained with Vistas training set. The most notable performance difference when adding correspondences are for the night images, row one and two. Comparing the results of E and E + C, we can see that adding correspondences enables the network to label the road correctly. It however fails at labeling sky and cars correctly since these are not included in the correspondence dataset.

the extra annotations with the networks trained with extra annotations and the correspondence loss. As can be seen in Table 4, we get an improvement in mIoU when adding the correspondence loss for both the network trained with Cityscapes and extra annotations as well as the one trained with Cityscapes, Vistas and extra annotations. This holds both for the CMU and RobotCar datset. As can be seen in Fig. 4 where qualitative results on the CMU test set are presented, adding the correspondences improves segmentations for areas that are very visually different between seasons. Some examples are yellow leaves on the ground during autumn or snow during the winter. These confuse the network trained only on Cityscapes since there are no such examples in the training set. The baseline that is also trained on Vistas handles these situations better but there is still an improvement when adding the correspondence training, especially when it comes to, *e.g.*, patches covered in snow, as these are not included in the subset of classes present in both datasets and, thus, excluded from the training. For the RobotCar dataset, the most prevalent improvements are on the night images, which can be seen in some example segmentations in Fig. 5. Additional qualitative results can be seen in the supplementary material. Worth to note is also that we manage to get a larger performance increase using just a few coarsely annotated images and the correspondence dataset compared to adding the entire Vistas training set, both for the CMU and the RobotCar data. The Cross-Season Correspondence datasets required about 30 hours of manual labour each in the form of annotating correspondences and verifying poses. Adding the two hours it took to coarsely annotate a few images gives a total of 32 hours of manual labour required. This, compared to an estimated 28200 hours to annotate the Vistas training set.

The best performing correspondence loss differs between the datasets. On the CMU dataset, the CE loss performs best both with and without the Vistas training set. However, for the RobotCar dataset without the Vistas labels, the loss that performs best is Hinge$_F$. A difference between Hinge$_F$ and CE is that CE imposes a harder constraint on the output of the network for the correspondences. It basically treats the output of the network on the reference image as the ground truth for the corresponding pixels in the target image. Since the RobotCar Cross-Season Correspondence dataset is created using LIDAR data, the point measurements are not perfectly synchronized with the images which creates a slight misalignment for some correspondences. If there are some erroneous correspondences, having a hard constraint can be harmful for the performance of the network. In these cases, imposing a softer constraint via the Hinge$_F$ loss, which basically stipulates that the features should be similar, can give a better performance, especially when using a strong baseline training set.

Adding the correspondence training does improve the segmentation performance on the WildDash images slightly for the CMU dataset but not for the RobotCar dataset. A reason that there is no significant performance gain could be that the camera poses relative to the vehicle (front facing) and the image resolution are very similar for the Cityscapes, Vistas and WildDash datasets. For the CMU Seasons dataset, there is no front facing camera, only one facing forward/left and one forward/right. Hence, learning to segment these well does not necessarily improve the segmentation performance on WildDash. In addition, the image quality for both the CMU and the RobotCar images are poorer than those of Cityscapes, Vistas and WildDash. Despite this, adding the correspondence loss improves the results for the network trained on the CMU data and on the RobotCar data without Vistas. The advantage of learning to segment images during other weather conditions seems to be large enough to make a difference on the WildDash validation set.

7. Conclusion

In this paper, we have introduced two Cross-Season Correspondence datasets, each consisting of a set of 2D-2D matches between images taken under different conditions. We described how these datasets can be generated with little human supervision and demonstrated the usefulness of the datasets by training an image segmentation network. To this end we presented and investigated three different training losses, based on cross-entropy and hinge loss, that can be used for the correspondence data. Our experiments showed that adding the correspondences as extra supervision during training improves the segmentation performance of the network, making it more robust to seasonal changes and weather conditions. Improving the semantic segmentation performance could in turn lead to more robust localization results which provides the first step towards an iterative feedback loop improving localization and semantic segmentation. An investigation on how the improved image segmentation network affect semantic localization methods is left for future work.

Important future research directions include investigating options to remove the need for a few, manually annotated images. Possible approaches include an additional unsupervised domain adaptation step, adapting the segmentation algorithm to the reference images of the correspondence dataset. Additionally, the Cross-Season Correspondence Datasets provide opportunities for other application such as training robust feature detectors and descriptors.

Acknowledgements This work has been funded by the Swedish Research Council (grant no. 2016-04445), the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots) and Vinnova / FFI (Perceptron, grant no. 2017-01942).

References

[1] Relja Arandjelović and Andrew Zisserman. Visual Vocabulary with a Semantic Twist. In *Proc. ACCV*, 2014. 1

[2] Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *Proc. IV*, 2011. 3

[3] Sid Yingze Bao and Silvio Savarese. Semantic structure from motion. In *Proc. CVPR*, 2011. 1

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. Whats the point: Semantic segmentation with point supervision. In *Proc. ECCV*. Springer, 2016. 2

[5] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *PAMI*, Feb 1992. 4

[6] Maro Bláha, Christoph Vogel, Audrey Richard, Jan D. Wegner, Thomas Pock, and Konrad Schindler. Large-Scale Semantic 3D Reconstruction: an Adaptive Multi-Resolution Model for Multi-Class Volumetric Labeling. In *Proc. CVPR*, 2016. 1, 2

[7] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic SLAM. In *Proc. ICRA*, 2017. 1

[8] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *Proc. CVPR)*, 2018. 1

[9] 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. In *Proc. 3DV*, 2017. 2

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 2018. 2

[11] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proc. CVPR*, 2016. 2

[12] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proc. CVPR*, 2018. 2

[13] Ian Cherabier, Christian Häne, Martin R. Oswald, and Marc Pollefeys. Multi-Label Semantic 3D Reconstruction using Voxel Blocks. In *Proc. 3DV*, 2016. 1, 2

[14] Ian Cherabier, Johannes L. Schönberger, Martin R. Oswald, Marc Pollefeys, and Andreas Geiger. Learning Priors for Semantic 3D Reconstruction. In *Proc. ECCV*, 2018. 1, 2

[15] Andrea Cohen, Torsten Sattler, and Marc Pollefeys. Merging the Unmatchable: Stitching Visually Disconnected SfM Models. In *Proc. ICCV*, 2015. 1

[16] Andrea Cohen, Johannes L. Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. Indoor-Outdoor 3D Reconstruction Alignment. In *Proc. ECCV*, 2016. 1

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016. 1, 3, 5, 6

[18] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. CVPR*, 2017. 2

[19] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proc. ICCV*, 2015. 2

[20] David Eigen and Rob Fergus. Predicting Depth, Surface Normals and Semantic Labels With a Common Multi-Scale Convolutional Architecture. In *Proc. ICCV*, 2015. 2

[21] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 2

[22] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Proc. IJRR*, 2013. 2

[23] Christian Häne, Nikolay Savinov, and Marc Pollefeys. Class Specific 3D Object Shape Priors Using Surface Normals. In *Proc. CVPR*, 2014. 1, 2

[24] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Proc. ACCV*. Springer, 2016. 2

[25] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2

[26] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[27] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *arXiv preprint arXiv: 1803.06184*, 2018. 2

[28] C. Hne, C. Zach, A. Cohen, and M. Pollefeys. Dense Semantic 3D Reconstruction. *PAMI*, 2017. 1, 2

[29] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proc. CVPR*, 2017. 2

[30] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. CVPR*, 2011. 2

[31] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *Proc. ECCV*, 2014. 1, 2

[32] Lubor Ladický, Paul Sturgess, Chris Russell, Sunando Sengupta, Yalin Bastanlar, William Clocksin, and Philip H. S. Torr. Joint Optimization for Object Class Segmentation and Dense Stereo Reconstruction. *IJCV*, 100(2), 2012. 2

[33] Måns Larsson, Anurag Arnab, Shuai Zheng, Philip Torr, and Fredrik Kahl. Revisiting deep structured models for pixel-level labeling with gradient-based inference. *SIIMS*, 11(4), 2018. 2

[34] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *Proc. ECCV*, 2012. 1

[35] Nektarios Lianos, Johannes L. Schönberger, Marc Pollefeys, and Torsten Sattler. VSO: Visual Semantic Odometry. In *Proc. ECCV*, 2018. 1

[36] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proc. ICCV*, 2015. 2

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 2

[38] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[39] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proc. NIPS*, 2016. 2

[40] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *IJRR*, 2017. 3, 4

[41] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *Proc. ICRA*. IEEE, 2017. 2

[42] Arsalan Mousavian, Jana Kosecka, and JM Lien. Semantically Guided Location Recognition for Outdoors Scenes. In *Proc. ICRA*, 2015. 1

[43] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. *arXiv preprint arXiv:1712.00479*, 13, 2017. 2

[44] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proc. ICCV*, 2017. 1, 6

[45] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015. 2

[46] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017. 5

[47] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proc. ICCV*, 2015. 2

[48] Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proc. CVPR*, June 2015. 2

[49] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In *Proc. ECCV*, 2016. 2

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*. Springer, 2015. 2

[51] 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 *Proc. CVPR*, 2016. 2

[52] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proc. ECCV*. Springer, 2010. 2

[53] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proc. CVPR*, 2018. 2

[54] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proc. CVPR*, 2018. 1, 3, 4

[55] Nikolay Savinov, Christian Häne, Lubor Ladicky, and Marc Pollefeys. Semantic 3D Reconstruction with Continuous Regularization and Ray Potentials Using a Visibility Consistency Constraint. In *Proc. CVPR*, 2016. 1, 2

[56] L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth. Semantic Stixels: Depth is not enough. In *Proc. IV*, 2016. 1, 2

[57] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 4

[58] Johannes Lutz Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *Proc. CVPR*, 2018. 1

[59] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 3, 4

[60] Gautam Singh and Jana Košecká. Semantically Guided Geolocation and Modeling in Urban Environments. In *Large-Scale Visual Geo-Localization*, 2016. 1

[61] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic Scene Completion from a Single Depth Image. In *Proc. CVPR*, 2017. 1, 2

[62] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *Proc. ICCV*, 2017. 2

[63] Erik Stenborg, Carl Toft, and Lars Hammarstrand. Long-term visual localization using semantically segmented images. *Proc. ICRA*, 2018. 1

[64] Carl Toft, Carl Olsson, and Fredrik Kahl. Long-Term 3D Localization and Pose from Semantic Labellings. In *Proc. ICCV Workshops*, 2017. 1

[65] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic Match Consistency for Long-Term Visual Localization. In *Proc. ECCV*, 2017. 1

[66] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. *arXiv preprint arXiv:1802.10349*, 2018. 2

[67] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 2017. 2

[68] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Addressing appearance change in outdoor robotics with adversarial domain adaptation. In *Proc. IROS*, 2017. 2

[69] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *Proc. ICRA*, 2018. 2

[70] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proc. ICCV*. 2

[71] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer. In *Proc. CVPR*, 2016. 2

[72] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[73] Fisher Yu, Jianxiong Xiao, and Thomas A. Funkhouser. Semantic alignment of LiDAR data at city scale. In *Proc. CVPR*, 2015. 1

[74] Xin Yu, Sagar Chaturvedi, Chen Feng, Yuichi Taguchi, Teng-Yok Lee, Clinton Fernandes, and Srikumar Ramalingam. VLASE: Vehicle Localization by Aggregating Semantic Edges. In *Proc. IROS*, 2018. 1

[75] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernández Domınguez. Wilddash-creating hazard-aware benchmarks. In *Proc. ECCV*, 2018. 6

[76] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. CVPR*, 2017. 2, 5

[77] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proc. ICCV*, 2015. 2

[78] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proc. ECCV*, 2018. 2