VCSeg: Virtual Camera Adaptation for Road Segmentation

Gong Cheng
York University
Toronto, Canada
gongcheng@eecs.yorku.ca

James H. Elder
York University
Toronto, Canada
jelder@yorku.ca

Abstract

Domain shift limits generalization in many problem domains. For road segmentation, one of the principal causes of domain shift is variation in the geometric camera parameters, which results in misregistration of scene structure between images. To address this issue, we decompose the shift into two components: Between-camera shift and within-camera shift. To handle between-camera shift, we assume that average camera parameters are known or can be estimated and use this knowledge to rectify both source and target domain images to a standard virtual camera model. To handle within-camera shift, we use estimates of road vanishing points to correct for shifts in camera pan and tilt. While this approach improves alignment, it produces gaps in the virtual image that complicates network training. To solve this problem, we introduce a novel projective image completion method that fills these gaps in a plausible way. Using five diverse and challenging road segmentation datasets, we demonstrate that our virtual camera method dramatically improves road segmentation performance when generalizing across cameras, and propose that this be integrated as a standard component of road segmentation systems to improve generalization.

1. Introduction

Road segmentation is an important computer vision problem for many applications, including autonomous driving, urban planning, traffic analytics, and road condition assessment. A major challenge limiting generalization is domain shift. Sources of domain shift are many and include the geography, road design, weather, season and time of day. One of the most important contributors is shift in the geometric camera parameters, especially focal length, aspect ratio, pitch, pan and roll. When transferring from source to target datasets this shift is typically ignored and images are resized to a standard size and aspect ratio for network training and adaptation. It can be challenging for domain adaptation methods to fully correct for this shift, especially when the target camera parameters deviate substantially from the source parameters. For example, if source domain cameras have relatively long focal length and thus small field of view (FOV) but the target camera has a large FOV, background pixels will be mislabeled as road, which can lead to serious failures.

To address this problem, we argue that training and inference should be performed not in the native pixel domain of the images but rather in a standardized virtual camera domain. This is possible due to the roughly planar geometry of most roads and serves to greatly reduce domain shift due to variations in geometric camera parameters, leading to improved training, adaptation and inference. Our proposed virtual camera method for road segmentation (VCSeg) reduces shift both between and within datasets, leading to more stable and reliable segmentation performance. Our specific contributions are:

1) A novel virtual camera adaptation method for road segmentation that takes advantage of the approximately planar geometry of the road to identify homographies that map each camera view to a standard virtual camera.

2) A novel projective image completion method that uses the road vanishing point to extrapolate RGB values from defined to undefined regions of virtual camera images in a geometrically plausible way.

3) A 5-fold cross-validation evaluation across 5 diverse public datasets that demonstrates a 9.8% improvement in Intersection-over-union (IoU) performance.

2. Related Work

2.1. Monocular Road Segmentation

Early road segmentation methods are typically based on appearance, geometry or a fusion of the two. Appearance-based methods [25, 16, 42, 39, 37] rely on color and texture features. These methods often struggle when objects or background are similar in colour or texture to the road (e.g., sidewalks, grey buildings or cars) and when the road is covered by snow or ice. Geometry-based methods that exploit the road vanishing point [22, 32, 33], horizon [4],

or road boundaries [23, 40, 43]) can potentially overcome some of these challenges, however they typically fail to exclude the objects that are on the road (e.g., pedestrians or vehicles) and tend to deliver over-simplified segmentations. Some have tried to fuse geometry & appearance cues. For example, Cheng *et al.* [12] fused probabilistic geometry and colour cues within a Bayesian framework, but evaluation was limited to a single, small dataset.

More recently, deep networks have been the dominant approach to image segmentation in general [28, 35, 5, 21, 20, 8, 9, 24], and road segmentation in particular [26, 30, 48, 11, 44, 47, 31, 49, 29, 19, 41]. SegNet [5] is an encoder-decoder style semantic segmentation network based on two modified VGG16 [38] networks connected back-to-back. Multinet [44] extends this approach to employ three parallel decoders for road segmentation, street classification, and vehicle detection. s-FCN-loc [47] fuses parallel VGG-16 backbones processing RGB and contour inputs. SSL-GAN [19] employs a ResNet-101-based GAN to support semi-supervised training. Another semi-supervised approach called reverse and boundary attention network [41] first predicts a coarse road segmentation and then refines this estimate in a top-down manner.

Since most of these methods are trained and evaluated on only one or two road segmentation datasets, their ability to generalize is poorly understood.

2.2. Unsupervised Domain Adaptation

The appearance of road imagery can vary widely depending upon the parameters of the camera, the geography, traffic marking conventions, level of traffic, illumination, time of day, season and weather. Given the costs of obtaining ground-truthed data, it is difficult to span all of these variations in a training dataset. Instead, we must understand how a system trained on one or more ground-truthed source datasets can be adapted to a new unlabelled target dataset. While in some cases it may be feasible to obtain a small amount of labelled data for the target dataset to assist in adaptation, widespread deployment will be easiest if the adaptation can be performed in a completely unsupervised fashion.

Numerous unsupervised domain adaptation methods have been proposed for semantic segmentation [46, 45, 10]. AdaptSegNet [45], representative of the adversarial approaches, uses a GAN to adapt the segmentation output from target domain images to be similar to the output from source domain images. Typically these adaptation methods focus on adapting from synthetic source datasets (e.g., GTA5 [34] or SYNTHIA [36]) to real target datasets (e.g., Cityscapes) in order to take advantage of the relatively large and inexpensive labelled data available in the synthetic dataset. However, synthetic scenes do have limitations in their level of realism, and if real labelled datasets already exist it is important to understand how systems trained on these datasets can be adapted to novel conditions. The geometry-guided adaptation (GeoSeg) method of Cheng *et al* [13] fuses classical geometry with deep network segmentation, using geometric features (vanishing points) to automatically position a prior on the target image that is accurate enough to adapt a segmentation network to accommodate domain shift in appearance cues.

Our proposed VCSeg method is inspired by this adaptive geometry-appearance fusion. However, we note that GeoSeg confounds two very different types of domain shift: *geometric* domain shift that arises from variations in intrinsic and extrinsic camera parameters, and *appearance* domain shift that arises from variations in geography, traffic marking conventions, level of traffic, illumination, time of day, season and weather. We argue that these different kinds of domain shift should not be lumped together, because, while adapting to shifts in appearance can be extremely complex, adapting to geometric domain shift can be approached using classical projective geometry methods. Since, as we will show, geometric domain shift can be a dominant factor, disentangling the two types of domain shift and addressing the geometry shift first can greatly improve generalization performance.

Our approach to handling geometric domain shift is to take advantage of the roughly planar geometry of the road surface to establish a common virtual camera to which diverse camera imagery can be projected. This aligns the imagery drawn from diverse cameras with widely varying focal lengths, resolutions, and heights. We further explore additional frame-by-frame adjustments in reprojection that accommodates variations in extrinsic camera parameters as the vehicle moves but also in more extreme forms for consumer dash-cams, which are often repositioned between uses.

3. Datasets

We use five diverse road segmentation datasets (Table 1, Figs. 6,7: CamVid [7, 6], Cityscapes [15], KITTI [18], Qian [3], and Toronto-2020 [13]. We follow the GeoSeg[13] partitioning of these datasets into training and test [1].

All five datasets were captured with vehicle-mounted cameras. CamVid was captured in the UK, CityScapes and KITTI in Germany and Qian & Toronto-2020 in Toronto, Canada. While for KITTI, CamVid & CityScapes datasets weather is fair and road surfaces dry, for Qian & Toronto-2020 weather ranges widely and roads may be dry, wet, or partially covered with snow an ice. Nonlinear distortions have been removed from all images.

Table 1 also lists the intrinsic and approximate average extrinsic geometric camera parameters for the five datasets. Where camera height and tilt were not specified they were

Dataset	Images Training/Test	Weather Conditions	Resolution (px)	Principal Point (px)	Focal Length [x,y] (px)	FOV [H,V] (deg)	Camera Position	Height (m)	Tilt (deg)
CamVid [7, 6]	367 / 233	Dry	[960, 720]	[485, 365]	[862, 1141]	[58.2, 35.0]	Fixed	1.21*	0*
Cityscapes [15]	1528 / 1406	Dry	[2048, 1024]	[1082, 514]	[2262, 2252]	[48.7, 25.6]	Fixed	1.21	−2.4
KITTI [18]	145 / 144	Dry	[1242, 375]	[610, 173]	[721, 721]	[81.5, 29.2]	Fixed	1.65	0
Qian [3]	34 / 31	Variable	[2048, 1536]	[1024, 768]	[1845, 1845]	[58.1, 45.2]	Variable	1.50*	0*
Toronto-2020 [13]	400 / 400	Variable	[1024, 576]	[508, 328]	[776, 701]	[66.8, 44.7]	Variable	1.35	0*
Virtual	-	-	**[5500, 2800]**	**[2750, 1400]**	**[2750, 1400]**	**[101.1, 63.5]**	Fixed	**1.38**	**0**

Table 1. The five cameras and road segmentation datasets used in this paper. Estimated values indicated by *.

estimated by visual inspection of the images. Note the wide variation in parameters across datasets. In the following we will initially assume zero roll and pan angles, but will later relax this assumption for pan.

For CamVid, CityScapes and KITTI datasets the camera pose was fixed relative to the vehicle - variations in camera pose relative to the road thus arise primarily from vehicle motion. For the Qian & Toronto-2020 datasets, on the other hand, the cameras were removable dash-cams, and thus contain wider variations in camera pose due to variations in how the cameras was mounted each day.

Pixel-wise road/non-road labels were obtained from the websites for each dataset. Additional ground truth vanishing point positions and ego-vehicle mask annotations were obtained from the GeoSeg website [1].

Following GeoSeg, we train and evaluate using 5-fold leave-one-out across datasets. In other words, the network is trained on the training partitions of four source datasets and then evaluated on the test partition of the fifth, left-out target dataset. As in GeoSeg, we ignore the ego-vehicle regions when doing source training, target adaptation, and performance evaluation.

4. Method

Fig. 1 shows an overview of the VCSeg training pipeline. Source images and ground truth are first reprojected to a standard virtual camera image plane. This will lead to undefined regions in both the virtual camera image and ground truth mask. To address this, we employ a novel projective completion method to fill these gaps, allowing the segmentation network to be trained over multiple datasets with diverse camera parameters. To apply the network to a novel target domain, target images are processed through the same reprojection and completion pipeline, using the target camera parameters, prior to applying the network. The resulting segmentation is reprojected back to the original target camera plane for evaluation.

4.1. Reprojection to Virtual Camera

To address geometric domain shift between and within datasets, we define a virtual standard camera with normative camera parameters, and approximate the mapping from real to virtual camera planes as a homography. This is a reasonable approximation for several reasons. First, the mapping is exact for planar surfaces and road surfaces are typically approximately planar. Second, much of the variation in the extrinsic camera parameters is in the camera rotation, particularly tilt and pan, and for pure rotations, the mapping from image to image is homographic for general scenes, not just planes. Third, distortions will occur primarily for objects with significant vertical extent, and these are of lesser importance for road segmentation.

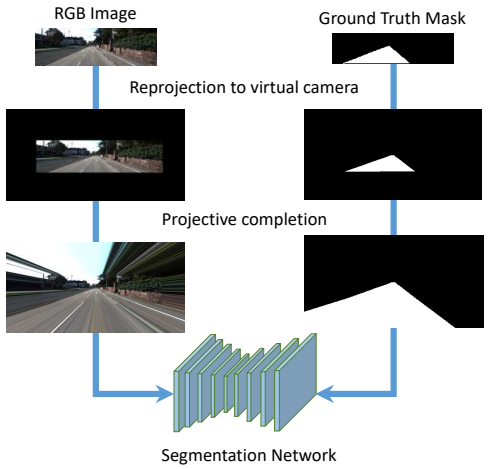

Figure 1. Overview of the proposed method.

4.1.1 Projection of Physical Camera to Ground Plane

We employ a world coordinate frame aligned with and lying on the road, directly beneath the camera (Fig. 2). We ignore horizontal translations of the camera and only model translations in the vertical (Y) direction. This allows us to fully specify the pose of each camera relative to a common world frame.

The general projection equation is

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where (X, Y, Z) is a point in 3D space, (x, y) are the image pixel coordinates of the projection, K is the intrinsic matrix,

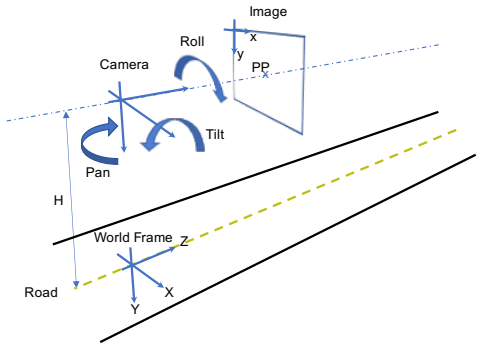

Figure 2. Camera geometry. PP denotes the principal point of the camera.

R and t are the rotation and translation of the world frame relative to the camera, respectively, and λ is a scaling parameter. Since in our case $Y = 0$ on the ground plane, the projection of a 3D point on the road to the physical camera image coordinates (x_p, y_p) can be modeled as

$$\lambda \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K_p \left[R_p^1 | R_p^3 | t_p\right] \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \quad (2)$$

$$\rightarrow \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} = \lambda \left(K_p \left[R_p^1 | R_p^3 | t_p\right]\right)^{-1} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (3)$$

where K_p is the intrinsic matrix of the physical camera, R_p^1 and R_p^3 are the first and third columns of its rotation matrix, and t_p is the translation of the world frame relative to the camera. Note that four of the five physical cameras are, on average, aligned with the world frame, so that R_p is the identity matrix and $t_p = [0, H_p, 0]^\top$, where H_p is the height of the camera above the road. The Cityscapes camera, however, is on average tilted slightly down , so that the third column R_p^3 of the rotation matrix becomes $[0, -\sin(\beta), \cos(\beta)]^\top$ and $t_p = H_p\,[0, \cos(\beta), \sin(\beta)]^\top$, where $\beta = 2.4$ deg is the average tilt angle of the camera.

Given the intrinsic and average extrinisic parameters of each physical camera, Equation 3 allows each pixel of the average image to be mapped to ground plane coordinates (X, Z). Fig. 3 shows a mapping of the bottom quarter of each camera FOV to the groundplane: Note the significant differences in the 'footprints' of each camera.

4.1.2 Defining the Virtual Camera

We align the virtual camera with our world frame (i.e., 0 pan, tilt and roll, Fig. 2) and set its height to the mean height of the five cameras considered here: 1.38m.

We assume equal x and y focal lengths for our virtual camera, and set these to the maximum focal length (in pixels) over our datasets to preserve resolution. (Larger virtual focal lengths could be chosen to accommodate future cameras.) We selected the horizontal and vertical FOVs of the virtual camera to ensure that at least the lower half of all physical camera images reprojected to lie within the virtual camera image, taking into account expected variations in pan ($\pm$20 deg) and tilt ($\pm$15 deg). This resulted in horizontal and vertical FOVs of 101.1 and 63.5 deg, respectively. Combined with the focal length, this determined the resolution of the virtual camera (5,500 $\times$ 2,800 pixels). We employ a central principal point.

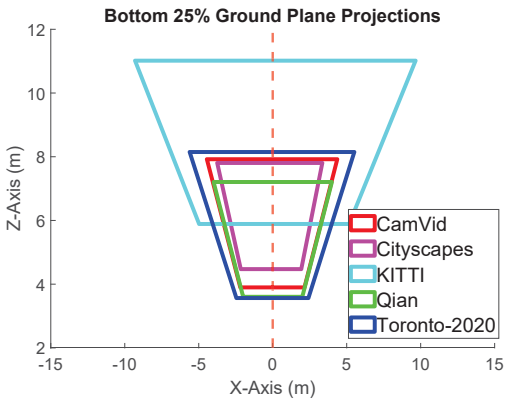

Figure 3. Ground plane back-projection of the lower quarter of the mean FOV for each camera.

4.1.3 Projection of Ground Plane to Virtual Camera

Since the virtual camera is aligned with the world frame, the projection equation (1) simplifies to

$$\lambda \begin{bmatrix} x_v \\ y_v \\ 1 \end{bmatrix} = K_v \left[I^{13} | t_v\right] \begin{bmatrix} X \\ Z \\ 1 \end{bmatrix} \quad (4)$$

where K_v is the intrinsic matrix of the virtual camera, I^{13} is the 3×2 matrix formed from the first and third columns of the 3×3 identity matrix, $t_v = [0, H_v, 0]^\top$ is the translation of the virtual camera, and H_v is its height above the road.

Note that the first homography (Eqn. 3) maps image points above the principal point to ground plane points behind the camera. However, since the parameters of the physical and virtual cameras will be relatively similar, by and large the second homography (Eqn. 4) will correctly map these points to the upper portion of the virtual image plane.

Combining Equations 3 and 4 yields the direct homography relating the physical camera to the virtual camera:

$$\lambda \begin{bmatrix} x_v \\ y_v \\ 1 \end{bmatrix} = K_v \left[I^{13} | t_v\right] \left(K_p \left[R_p^1 | R_p^3 | t_p\right]\right)^{-1} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (5)$$

Fig. 4 shows the reprojection of the average FOVs of

each of the five cameras modeled here to the virtual camera plane. Note that the frame-to-frame FOVs will shift as the exact pose of each physical cameras varies. We turn to this issue now.

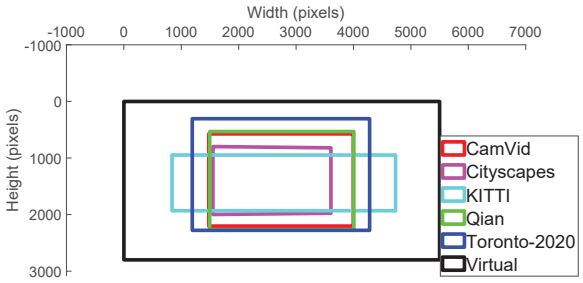

Figure 4. Projection of physical camera FOVs to the virtual camera plane.

4.2. Within-Camera Shift

Reprojection to the virtual camera serves to greatly reduce geometric shift between datasets. However, there can also be significant shifts in the extrinsic camera parameters *within* datasets. For the CamVid, Cityscapes and KITTI datasets, these within-camera shifts are primarily in pan, due to the yaw of the car, but other shifts due to vibration etc. also occur. For the Qian and Toronto-2020 datasets there are more profound shifts because the dashcams employed in these datasets were repositioned numerous times.

Here we focus specifically on within-camera shifts in pan and tilt, approximating the roll as 0 and neglecting translational shifts. Taking the world frame as our global frame-or-reference, we decompose the rotation matrix R_p into a sequence of two rotations, first a tilt of β about the X-axis and then a pan of α about the Y-axis. This results in:

$$R_p^1 = \begin{bmatrix} \cos\alpha \\ 0 \\ -\sin\alpha \end{bmatrix}, \quad R_p^3 = \begin{bmatrix} \sin\alpha\cos\beta \\ -\sin\beta \\ \cos\alpha\cos\beta \end{bmatrix} \tag{6}$$

R_p^3 represents the direction of the road vanishing point (x_{vp}, y_{vp}) in camera coordinates, and so can easily be expressed in terms of this vanishing point:

$$R_p^3 = \lambda_3 \begin{bmatrix} (x_{vp} - x_0)/f_x \\ (y_{vp} - y_0)/f_y \\ 1 \end{bmatrix} \tag{7}$$

where λ_3 is a normalizing constant. Finally, Eqns 6 and 7 can be combined to yield $\tan\alpha = (x_{vp} - x_0)/f_x$, which leads to

$$R_p^1 = \lambda_1 \begin{bmatrix} 1 \\ 0 \\ -(x_{vp} - x_0)/f_x \end{bmatrix} \tag{8}$$

where λ_1 is a normalizing constant.

These corrections to the reprojection equation will serve to better align the source images for training. At inference, we employ the line-based (MCMLSD [2]) vanishing point estimation algorithm from [13] to correct for witihin-camera shift within the target dataset. Vanishing point estimates generating pan shifts of more than 20 deg or tilt shifts of more than 15 deg are rejected as unreliable and replaced by the mean of the estimated vanishing point locations judged to be valid.

4.3. Projective Completion

Fig. 4 highlights a complication with our virtual camera approach. Since none of the physical cameras span the entire virtual FOV, when training in the virtual camera domain, each (input, ground truth) pair will include undefined regions, which means that some network activations and back-propagated gradients are undefined. Empirically we find that filling these gaps with nominal values (e.g., zeros) or using standard image boundary handling methods (e.g., pixel replication) leads to huge errors in segmentation. To solve this problem, we introduce a novel projective completion method that exploits the typical approximate 3D symmetry in the road surface to generate more plausible image completions to support accurate learning.

Fig. 5 illustrates the approach. Recall that for each source image the vanishing point of the road has been identified. We establish a polar coordinate system centred on the vanishiing point and identify the angle of each source boundary pixel. We then assign RGB and ground truth mask values to all pixels outside the source image based on an angular linear interpolation of the RGB and ground truth mask values for these boundary pixels. We find that results of this simple completion method are surprisingly reasonable for a large majority of scenes. (See supplementary file for more examples.) At inference, the estimated vanishing point [13] is used to complete the target image.

(a) (b) (c)

Figure 5. Projective completion. (a) Reprojected image from the CamVid training partition. Red point indicates ground truth annotated road vanishing point, red dashed lines the source boundary pixels and green lines the completion directions. (b) Projectively completed source image. (c) Projectively completed ground truth mask.

4.4. Supervised Training

Once all source images have been reprojected and projectively completed in the virtual camera plane, the network can be trained. To compare fairly with prior methods we employ a VGG16 [38] hourglass segmentation architecture

for all methods, including VCSeg.

4.5. Geometry-Driven Adaptation

We also evaluate the Geometry-driven adaptation method introduced by GeoSeg [13], but in the virtual camera domain. Briefly, this method learns a prior for the road segmentation from the source images, anchored on the ground-truth vanishing points. In GeoSeg, this prior is used as a weak source of ground truth in the target image domain to fine-tune the segmentation network. Here we do the same, but derive the prior in the virtual camera domain by reprojecting the ground-truth masks from each source dataset. As in GeoSeg, we define a ground-truth tri-mask based on the prior: Letting $p_g(x, y)$ represent the prior geometric probability that virtual camera pixel (x, y) projects from the road, we set the ground truth to road where $p_g(x, y) > 0.75$, to background where $p_g(x, y) < 0.25$ and to unknown for all other pixels. This surrogate ground truth is then used to fine-tune the network.

5. Experimental Details

As in GeoSeg [13], we score segmentations by intersection over union (IoU), ignoring ego-vehicle regions. We compare against three alternative methods: SegNet [5], the GAN-based AdaptSegNet, [45] and GeoSeg [13].

As in [13], we used a cross-entropy loss and trained for 600 epochs. However, we noticed that VCSeg benefits from a smaller batch size and faster learning rate - Table 2. To control for this difference, we list the SegNet and GeoSeg performance as reported in [13] and also performance when trained with the parameters used to train VCSeg.

Parameter	Old [13]	New
Batch size	16	8
Learning rate	0.001	0.005
Rate decay	0.0001	0.0005

Table 2. Learning parameters used by GeoSeg [13] and our new method, VCSeg.

Experiments were conducted with a Pentium 3.4 GHz i7 processor, 64 GB RAM, and an Nvidia GTX 1080 Ti GPU. Line detection, vanishing point detection and projective completion were performed in MATLAB, while networks were implemented in PyTorch. Table 3 lists the run time at inference for each component of the method. Line detection currently dominates the run time: Future work will focus on speeding line detection and further optimizations to make the method real time.

Stage	Runtime per image
Line detection (MCMLSD) [2]	6200 msec
Vanishing point detection [13]	25 msec
Projection to virtual camera	182 msec
Projective completion	85 msec
Network inference	43 msec
Back-projection to physical camera	43 msec

Table 3. Average runtime for inference.

6. Results

6.1. Comparison with Prior Adaptation Methods

Table 4 compares quantitative results for VCSeg and prior adaptation methods on a common SegNet architecture. We find that the effect of the training parameters (Table 2) is variable across datasets, but overall, the new parameters adopted for VCSeg improve the performance of SegNet but have negligible effect on the performance of GeoSeg.

We find that our between-camera virtual camera reprojection method leads to a dramatic 12.2% improvement in segmentation performance over SegNet (average of 62.3% IoU vs 50.1% IoU) - note that we see an improvement for every target dataset and the only difference between these methods is that in VCSeg training is performed in the virtual camera domain. Even without adaptive fine-tuning, our between-camera VCSeg method outperforms on average the more complex GeoSeg and AdaptSegNet methods, both of which require costly (though unsupervised) fine-tuning on the target dataset.

We observe a second dramatic (8.8%) improvement in performance when we also correct for within-camera shifts in tilt and pan angles between images (VCSeg + Within-Cam): (average of 71.1% IoU vs 62.3% IoU when accounting only for between-camera shifts). Again, we see an improvement for every target dataset, but the improvement is particularly dramatic for the dashcam datasets (Qian and Toronto-2020), where the camera was repositioned frequently.

6.2. Incorporating Appearance Adaptation

We note that our full VCSeg method performs close to but does not beat the GeoSeg method for the Qian dataset, suggesting that, for this dataset at least, GeoSeg is capturing some domain shift that is not accounted for by shift in the camera parameters. To further explore this, we assessed the performance of GeoSeg adaptation, but in the virtual camera domain (Section 4.5), accounting for both between- and within-camera shifts. Table 5 shows the results. We find that adaptation improves results for Qian and Toronto-2020 datasets, demonstrating that for these datasets, the GeoSeg and virtual camera approaches can be fruitfully combined. However, for the other three datasets we find that adaptation hurts performance.

We believe these mixed results are due to two factors.

Target Dataset	SegNet [5] Old parms	SegNet [5] New parms	GeoSeg [13] Old parms	GeoSeg [13] New parms	AdaptSegNet [45]	VCSeg Between-Cam	VCSeg +Within-Cam
CamVid	39.2	57.5	61.0	60.4	65.8	73.6	**73.9**
Cityscapes	60.8	65.7	74.7	75.3	71.1	73.5	**76.9**
KITTI	39.1	52.9	46.9	47.0	56.7	71.5	**80.6**
Qian	31.2	29.6	**58.3**	56.2	52.4	43.7	56.8
Toronto-2020	44.7	34.2	65.6	65.7	60.7	49.0	**67.5**
Mean	43.0	50.1	61.3	60.9	61.3	62.3	**71.1**

Table 4. Quantitative comparison with prior adaptation methods, using a common SegNet architecture. Numbers are % IoU. See Table 2 for old and new training parameters.

First, since performance for VCSeg is already relatively high for CamVid, Cityscapes and KITTI datasets, the geometric prior derived from source ground truth (fourth column in Table 5) is likely too weak to provide sufficiently reliable supervision to improve results further. Second, while these datasets contain relatively modest appearance variations, Qian and Toronto-2020 datasets contain massive variations in weather, including snow and ice; GeoSeg is able to handle these appearance shifts that are not handled by VCSeg. Our recommendation is thus to apply the two methods together in cases when the target domain includes significant non-geometric shifts, including weather and illumination.

Target Dataset	VCSeg	VCSeg +Geo-Adaptation [13]	Geo-Prior [13]
CamVid	**73.9**	63.7	59.0
Cityscapes	**76.9**	76.2	73.6
KITTI	**80.6**	70.1	62.0
Qian	56.8	**60.1**	59.3
Toronto-2020	67.5	**69.2**	65.5
Mean	**71.1**	67.9	63.9

Table 5. Results of Geometry-driven adaptation in the virtual camera domain. Numbers are % IoU. Geo-Prior indicates segmentation based upon the geometric prior used by GeoSeg [13], mapped to the virtual camera.

7. Comparison with Other Architectures

To focus on the issue of geometric adaptation our comparisons thus far have been based on a common SegNet architecture. Here we compare against two more recent semantic segmentation approaches (Table 6): STDC-Seg [17] and GALDNet [27], both pretrained on the Cityscapes dataset. (Since our Cityscapes test partition is drawn from this dataset we do not compare on Cityscapes.)

We find that despite being based upon a much simpler architecture, our VCSeg approach performs substantially better overall than these more recent Cityscapes-trained architectures, beating the nearest competitor by 7.3% on average. KITTI and Toronto-2020 datasets have much larger FOVs than Cityscapes, which creates a geometric mismatch for these non-adaptive Cityscapes-trained architectures. Correctly aligning the data in the virtual camera domain allows VCSeg to generalize much better, leading to much higher performance on KITTI and Toronto-2020 test datasets. CamVid and Qian datasets, on the other hand, have FOVs that are more similar to Cityscapes and we see here that GALDNet performs somewhat better. This suggests that combining VCSeg with this more powerful architecture could lead to even further improvements.

Target Dataset	VCSeg (Ours)	STDC Seg50 [17]	STDC2 Seg75 [17]	GALDNet ResNet50 [27]	GALDNet ResNet101 [27]
CamVid	73.9	74.2	76.4	**79.3**	78.7
KITTI	**80.6**	62.4	61.9	70.8	65.8
Qian	56.8	51.8	65.5	56.7	**68.0**
Toronto-2020	**67.5**	33.6	36.9	31.8	37.2
Mean	**69.7**	55.6	60.2	56.7	62.4

Table 6. Quantitative comparison with recent semantic segmentation architectures trained on Citysapes. Numbers are % IoU.

8. Evaluating Projective Completion

VCSeg depends upon a novel projective completion method to fill-in missing image and ground-truth information outside the projection of the physical image. Table 7 quantifies the importance of this completion step. Substituting zero-padding for projective completion leads to a consistent drop in performance across all five test datasets, averaging 13.4%.

Correction of within-camera variation and projective completion both depend on accurate estimation of vanishing points. To understand the impact of vanishing point errors on performance, the last column lists VCSeg performance when the *ground truth* vanishing points are used at inference. We do see a consistent, if relatively modest, improvement across all five target datasets, averaging 1.1%. The greatest improvement is seen for Toronto-2020, which contains relatively large variations in camera parameters.

8.1. Qualitative Results

To get a sense of some advantages of our virtual camera approach, Fig. 6 compares results of all algorithms for the image from each dataset that generate the *highest* IoU

Target Dataset	Zero Padding	Projective Completion (Est. VP)	Projective Completion (GT VP)
CamVid	64.0	**73.9**	75.1
Cityscapes	69.7	**76.9**	77.6
KITTI	51.1	**80.6**	81.5
Qian	53.4	**56.8**	56.9
Toronto-2020	50.4	**67.5**	69.7
Mean	57.7	**71.1**	72.2

Table 7. Performance of VCSeg with and without projective completion. The last column shows for reference the performance when ground truth vanishing points are used at inference. This provides an upper bound on performance gains achievable through improvements to vanishing point detection. Numbers are % IoU.

for our full VCSeg method. It is encouraging to see VCSeg performing well even for the snowy road from the Toronto 2020 dataset. Comparing performance of VCSeg (Between-Cam) with VCSeg (Within-Cam), it is apparent that modeling within-camera shifts in pan and tilt is crucial to locking on to the left and right boundaries of the road and avoiding fragmentation. Note that while GeoSeg and AdaptSegNet also avoid much of the fragmentation seen in the SegNet results, these methods are generally not as accurate at capturing the left and right road boundaries.

Fig. 7 shows failure modes, i.e., results for the images that generate the *lowest* IoU for our full VCSeg method. We note that for four of the five cases (CamVid, KITTI, Qian, Toronto-2020), our version of VCSeg that only models between-camera shifts performs relatively well, suggesting that these failures may arise from errors in vanishing point estimation. Improvements to this component of the pipeline would likely address some of these failure modes.

More results can be found in the supplementary file.

9. Conclusions and Future Research

Shift in camera parameters is a major reason road segmentation systems fail to generalize to new domains. Our results show that these geometric camera parameter shifts can be addressed by reprojecting images to a common virtual camera plane and using a projective completion method to fill resulting gaps in the imagery prior to training. Correcting for between-camera shifts leads to a 12.2% improvement in IoU performance, while correcting for within-camera shifts leads to an additional 8.8% improvement, for a total of 21% (50.1% to 71.1%). Appearance adaptation [13] can further boost performance for datasets with substantial weather variations.

Future work will examine more reliable methods for vanishing point estimation, to handle curved roads, for example, and whether modeling of camera roll and lateral translation can further improve performance. Ultimately, better methods for fusing these geometric adaptation methods with state-of-the-art appearance adaptation methods will lead to road segmentation systems that can be deployed reliably on various platforms and in diverse environments.

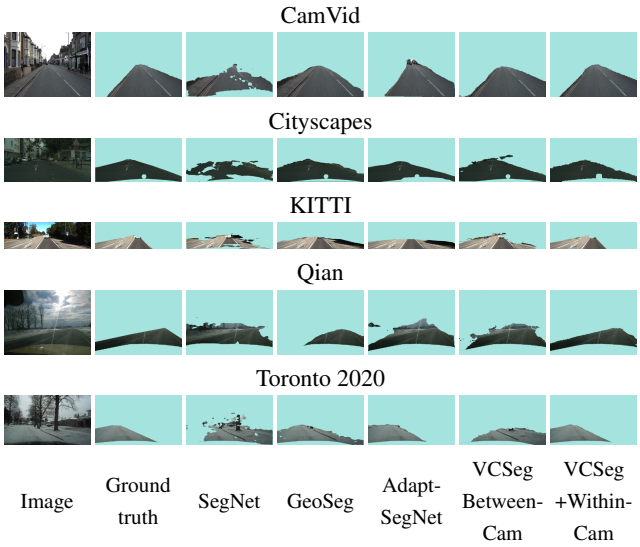

Figure 6. Successes: Image from each dataset for which our full VCSeg method obtains the highest IoU.

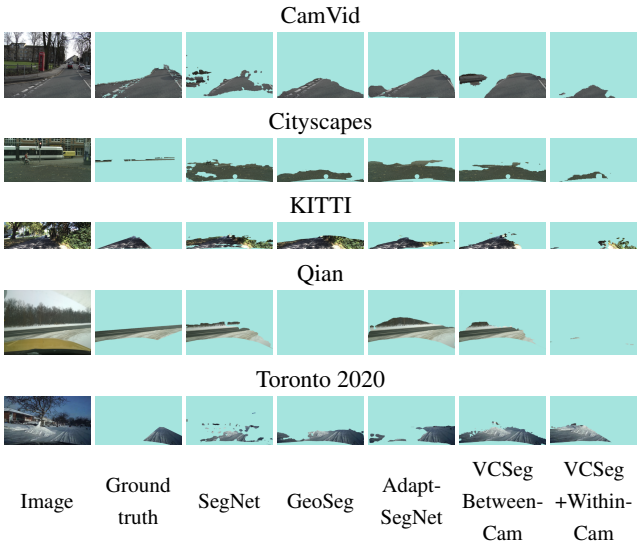

Figure 7. Failures: Image from each dataset for which our full VCSeg method obtains the lowest IoU.

References

[1] `www.elderlab.yorku.ca/resources/geometry-guided-road-adaptation/`.

[2] E. J. Almazàn, R. Tal, Y. Qian, and J. H. Elder. MCMLSD: A dynamic programming approach to line segment detection. In *IEEE CVPR*, pages 5854–5862, July 2017.

[3] E.J. Almazen, Y. Qian, and J.H. Elder. Road segmentation for classification of road weather conditions. *ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, pages 96–108, 2016.

[4] J. M. Alvarez, T. Gevers, and A. M. Lopez. 3D scene priors for road detection. In *IEEE CVPR*, pages 57–64, 2010.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[6] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2008.

[7] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, pages 44–57, 2008.

[8] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018.

[10] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum square loss. In *ICCV*, pages 2090–2099, 2019.

[11] Zhe Chen and Zijing Chen. Rbnet: A deep neural network for unified road and road boundary detection. In *International Conference on Neural Information Processing*, pages 677–687. Springer, 2017.

[12] G. Cheng, Y. Qian, and J. H. Elder. Fusing geometry and appearance for road segmentation. In *ICCV Workshops*, pages 166–173, Oct 2017.

[13] Gong Cheng, Yue Wang, Yiming Qian, and James H. Elder. Geometry-guided adaptation for road segmentation. In *Conference on Computer and Robot Vision (CRV)*, pages 46–53, 2020.

[14] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021.

[15] 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 *IEEE CVPR*, 2016.

[16] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Robotics: Science and Systems*, volume 38, Philadelphia, USA, 2006.

[17] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9725, 2021.

[18] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[19] Xiaofeng Han, Jianfeng Lu, Chunxia Zhao, Shaodi You, and Hongdong Li. Semisupervised and weakly supervised road detection based on generative adversarial networks. *IEEE Signal Processing Letters*, 25(4):551–555, 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016.

[21] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[22] H. Kong, J. Y. Audibert, and J. Ponce. Vanishing point detection for road detection. In *IEEE CVPR*, pages 96–103, 2009.

[23] H. Kong, J. Y. Audibert, and J. Ponce. General road detection from a single image. *IEEE Transactions on Image Processing*, 19(8):2211–2220, 2010.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] J. Lee and C. D. Crane. Road following in an unstructured desert environment based on the EM (expectation-maximization) algorithm. In *SICE-ICASE International Joint Conference*, pages 2969–2974, 2006.

[26] Dan Levi, Noa Garnett, and Ethan Fetaya. Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *BMVC*, pages 109.1–109.12, September 2015.

[27] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Global aggregation then local distribution in fully convolutional networks. In *BMVC*, 2019.

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE CVPR*, pages 3431–3440, June 2015.

[29] Yecheng Lyu and Xinming Huang. Roadnet-v2: A 10 ms road segmentation using spatial sequence layer. *arXiv preprint arXiv:1808.04450*, 1(3), 2018.

[30] Caio César Teodoro Mendes, Vincent Frémont, and Denis Fernando Wolf. Exploiting fully convolutional neural networks for fast road detection. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3174–3179. IEEE, 2016.

[31] Gabriel L Oliveira, Wolfram Burgard, and Thomas Brox. Efficient deep models for monocular road segmentation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4885–4891. IEEE, 2016.

[32] C. Rasmussen. Grouping dominant orientations for ill-structured road following. In *IEEE CVPR*, pages 470–477, 2004.

[33] C. Rasmussen. Texture-based vanishing point voting for road shape estimation. In *BMVC*, pages 7.1–7.10, 2004.

[34] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, volume 9906, pages 102–118, 2016.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation.

In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[36] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE CVPR*, 2016.

[37] C. Rotaru, T. Graf, and J. Zhang. Color image segmentation in HSI space for automotive applications. *Journal of Real-Time Image Processing*, 3(4):311–322, 2008.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 1409.1556, 2014.

[39] M. A. Sotelo, F. J. Rodriguez, and L. Magdalena. Virtuous: Vision-based road transportation for unmanned operation on urban-like scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 5(2):69–83, 2004.

[40] B. Southall and C. J. Taylor. Stochastic road shape estimation. In *IEEE ICCV*, volume 1, pages 205–212, 2001.

[41] Jee-Young Sun, Seung-Wook Kim, Sang-Won Lee, Ye-Won Kim, and Sung-Jea Ko. Reverse and boundary attention network for road segmentation. In *ICCV Workshops*, pages 876–885, 2019.

[42] C. Tan, T. Hong, T. Chang, and M. Shneier. Color model-based real-time learning for road following. In *IEEE Intelligent Transportation Systems Conference*, pages 939–944, 2006.

[43] C. J. Taylor, J. Malik, and J. Weber. A real-time approach to stereopsis and lane-finding. In *Conference on Intelligent Vehicles*, pages 207–212, 1996.

[44] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020, June 2018.

[45] 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. In *IEEE CVPR*, June 2018.

[46] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE CVPR*, pages 2517–2526, 2019.

[47] Q. Wang, J. Gao, and Y. Yuan. Embedding structured contour and location prior in Siamesed fully convolutional networks for road detection. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):230–241, Jan 2018.

[48] Shashank Yadav, Suvam Patra, Chetan Arora, and Subhashis Banerjee. Deep CNN with color lines model for unmarked road segmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 585–589. IEEE, 2017.

[49] Farnoush Zohourian, Borislav Antic, Jan Siegemund, Mirko Meuter, and Josef Pauli. Superpixel-based road segmentation for real-time systems using CNN. In *VISIGRAPP (5: VISAPP)*, pages 257–265, 2018.