BlindSpotNet: Seeing Where We Cannot See

Taichi Fukuda, Kotaro Hasegawa, Shinya Ishizaki,
Shohei Nobuhara, and Ko Nishino

Kyoto University, Kyoto, Japan
{tfukuda,khasegawa,sishizaki}@vision.ist.i.kyoto-u.ac.jp
{nob,kon}@i.kyoto-u.ac.jp
https://vision.ist.i.kyoto-u.ac.jp/

Abstract. We introduce 2D blind spot estimation as a critical visual task for road scene understanding. By automatically detecting road regions that are occluded from the vehicle's vantage point, we can proactively alert a manual driver or a self-driving system to potential causes of accidents (*e.g.*, draw attention to a road region from which a child may spring out). Detecting blind spots in full 3D would be challenging, as 3D reasoning on the fly even if the car is equipped with LiDAR would be prohibitively expensive and error prone. We instead propose to learn to estimate blind spots in 2D, just from a monocular camera. We achieve this in two steps. We first introduce an automatic method for generating "ground-truth" blind spot training data for arbitrary driving videos by leveraging monocular depth estimation, semantic segmentation, and SLAM. The key idea is to reason in 3D but from 2D images by defining blind spots as those road regions that are currently invisible but become visible in the near future. We construct a large-scale dataset with this automatic offline blind spot estimation, which we refer to as Road Blind Spot (RBS) dataset. Next, we introduce BlindSpotNet (BSN), a simple network that fully leverages this dataset for fully automatic estimation of frame-wise blind spot probability maps for arbitrary driving videos. Extensive experimental results demonstrate the validity of our RBS Dataset and the effectiveness of our BSN.

Keywords: autonomous driving, ADAS, road scene understanding, blind spot estimation, accident prevention

1 Introduction

Fully autonomous vehicles may soon become a reality. Advanced Driver-Assistance Systems (ADAS) have also become ubiquitous and intelligent. A large part of these developments is built on advancements in perceptual sensing, both in hardware and software. In particular, 3D and 2D visual sensing have played a large role in propelling these advancements. State-of-the-art LiDAR systems can resolve to centimeters at 300m or longer, and image understanding networks can recognize objects 100m away. A typical autonomous vehicle is equipped with up to ten of these sensors' visual perception, in addition to other modalities.

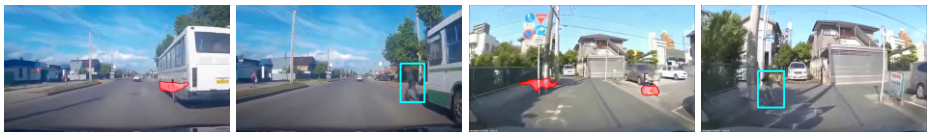

Fig. 1. We introduce a new dataset and network for automatically detecting blind spots on roads. Alerting manual and autonomous drivers of vehicles to such 2D blind spots can greatly extend the safety on the road. Our network, BlindSpotNet, successfully detects blind spots, from which pedestrians actually later spring out, without the need for costly, error-prone 3D reconstruction.

These autonomous and assistive driving systems are, however, still prone to catastrophic errors, even when they are operating at low speeds [23]. More sensors would unlikely eliminate these errors. In fact, we human beings have much fewer visual sensors (just our two eyes) but can drive at least as well as current autonomous vehicles. Why are we able to do this? We believe, one of the primary reasons is that, although we are limited in our visual perception, we know that it's limited. That is, we are fully aware of when and where we can see and when and where we cannot see. We know that we can't see well at night so we drive cautiously; we know that we cannot see beyond a couple of hundred meters, so we don't drive too fast; we know that we cannot see behind us, so we use side and room mirrors. Most important, we know that we cannot see behind objects, *i.e.*, that our vision can be obstructed by other objects in the scene. We are fully aware of these blind spots, so that we pay attention to those areas on the road with anticipation that something may spring out from them. That is, we know where to expect the unforeseen and we preemptively prepare for those events by attending our vision and mind to those blind spots.

Can we make computers also "see," *i.e.*, find, blind spots on the road? One approach would be to geometrically reason occlusions caused by the static and dynamic objects on the road. This requires full 3D scene reconstruction and localization of the moving camera, which is computationally expensive and prone to errors as it requires fragile ray traversal. Instead, it would be desirable to directly estimate blind spots in the 2D images without explicit 3D reconstruction or sensing. This is particularly essential for driving safety, as we would want to detect blind spots on the fly as we drive down the road. Given a frame from a live video, we would like to mark out all the occluded road regions, so that a driver or the autonomous system would know where to anticipate the unforeseen.

How can we accomplish 2D blind spot detection? Just like we likely do, we could learn to estimate blind spots directly in 2D. The challenge for this lies in obtaining the training data—the ground truth. Again, explicit 3D reasoning is unrealistic as a perfect 3D reconstruction of every frame of a video would be prohibitively expensive and error prone, especially for a dynamic road scene in which blind spots change every frame. On the other hand, labeling blind spots in video frames is also near impossible, as even to the annotators, reasoning about

the blind spots from 2D images is too hard a task, particularly for a dynamic scene. How can we then, formulate blind spot estimation as a learning task?

In this paper, we introduce a novel dataset and network for estimating occluded road regions in a driving video. We refer to the dataset as Road Blind Spot (RBS) Dataset and the network as BlindSpotNet (BSN). Our major contributions are two-fold: an algorithm for automatic generation of blind spot training data and a simple network that learns to detect blind spots from that training data. The first contribution is realized by implicitly reasoning 3D occlusions in a driving video from its 2D image frames by fully leveraging depth, localization, and semantic segmentation networks. This is made possible by our key idea of defining blind spots as areas on the road that are currently invisible but visible in the future. We refer to these as T-frame blind spots, *i.e.*, those 2D road regions that are currently occluded by other objects that become visible T frames later. Clearly, these form a subset of all true 2D blind spots; we cannot estimate the blind spots that never become seen in our video. They, however, cover a large portion of the blind spots (*cf*. Fig. 6) that are critical in a road scene including those caused by parked cars on the side, oncoming cars on the other lane, street corners, pedestrians, and buildings. Most important, they allow us to derive an automatic means for computing blind spot regions for arbitrary driving videos.

Our offline automatic training data generation algorithm computes T-frame blind spots for each frame of a driving video by playing it backwards, and by applying monocular depth estimation, SLAM, and semantic segmentation to obtain the depth, camera pose, semantic regions, and road regions. By computing the visible road regions in every frame, and then subtracting the current frame's from that of T-frames ahead, we can obtain blind spots for the current frame. A suitable value for T can be determined based on the speed of the car and the frame rate of the video. Armed with this simple yet effective algorithm for computing blind spot maps, we construct the RBS Dataset. The dataset consists of 231 videos with blind spot maps computed for 21,662 frames.

For on-the-fly 2D blind spot estimation, we introduce BlindSpotNet, a deep neural network that estimates blind spot road regions for an arbitrary road scene directly for a single 2D input video frame, which fully leverages the newly introduced dataset. The network architecture is a fully convolutional encoder-decoder with Atrous Spatial Pyramid Pooling which takes in a road scene image as the input. We show that blind spot estimation can be implemented with a light-weight network by knowledge distillation from a semantic segmentation network. Through extensive experiments including the analysis of the network architectures, we show that BlindSpotNet can accurately estimate the occluded road regions in any given frame independently, yet result in consistent blind spot maps through a video.

To the best of our knowledge, our work is the first to offer an extensive dataset and a baseline method for solving this important problem of 2D blind spot estimation. These results can directly be used to heighten the safety of autonomous driving and assisted driving, for instance, by drawing attention of the limited computation resource or the human driver to those blind spots (*e.g.*,

by sounding an alarm from the side with an oncoming large blind spot, if the person is looking away observed from an in-car camera). We also envision a future where BlindSpotNet helps in training better drivers both for autonomous and manual driving.

2 Related Work

Driver assistance around blind corners has been studied in the context of augmented/diminished reality [2, 20, 30]. Barnum *et al.* proposed a video-overlay system that presents a see-through rendering of hidden objects behind the wall for drivers using a surveillance camera installed at corners [2]. This system realizes realtime processing, but requires explicit modeling of blind spots beforehand. Our proposed method estimates them automatically.

Bird's-eye view (BEV) visualization [14, 33] and scene parsing [1, 11, 12, 19, 25, 29] around the vehicle can also assist drivers to avoid collision accidents. Sugiura and Watanabe [25] proposed a neural network that produces probable multi-hypothesis occupancy grid maps. Mani *et al.* [19], Yang *et al.* [29], and Liu *et al.* [14] proposed road layout estimation networks. These methods, however, require sensitive 3D reasoning to obtain coarse blind spots at runtime, while our BSN estimates blind spots directly in 2D without any 3D processing.

Amodal segmentation also handles occluded regions of each instance explicitly [8, 10, 15, 24, 27, 32, 34]. They segment the image into semantic regions while estimating occluded portions of each region. These methods, however, do not estimate objects that are completely invisible in the image. We can find a side road while its road surface is not visible at all, *e.g.*, due to cars parked in the street, by looking at gaps between buildings for example.

Understanding and predicting pedestrian behavior is also a primary objective of ADAS. Makansi *et al.* [17] trained a network that predicts pedestrians crossing in front of the vehicle. Bertoni *et al.* [3] estimated 3D locations of pedestrians around the subject by also modeling the uncertainty behind them. Our method explicitly recovers blind spots caused not only by pedestrians but also other obstacles including passing and parked cars, street medians, poles, etc.

Watson *et al.* [28] estimated free space including the area behind objects in the scene for robot navigation. They also generated a traversable area dataset to train their network. Our dataset generation leverages this footprint dataset generation algorithm to compute both visible and invisible road areas in a road scene image (*i.e.*, driving video frame). Blind spot estimation requires more than just the area behind objects as it needs to be computed and estimated across frames with a dynamically changing viewpoint.

3 Road Blind Spot Dataset

Our first goal is to establish a training dataset for learning to detect 2D blind spots. For this, we need an algorithm that can turn an arbitrary video into a video annotated with 2D blind spots for each and every frame. To derive such

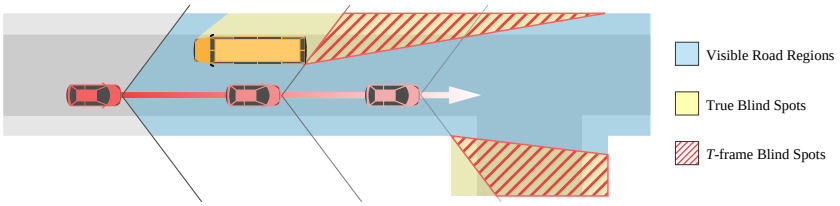

Fig. 2. We define blind spots as road regions that are currently invisible but visible in the next T-frames, which we compute by aggregating road regions across future frames and subtracting the visible road region in the current frame. These T-frame blind spots form a subset of true blind spots, cover key regions of them and, most important, can directly be computed for arbitrary driving videos (*cf*. Fig. 6 for a real example).

a method, we start by defining road-scene blind spots as something that we can compute from 2D videos offline and derive a method for computing those for arbitrary driving videos. With this algorithm, we construct a large-scale dataset of road blind spot videos (RBS Dataset). This data will later be used to train a network that can estimate 2D blind spots on the fly.

3.1 *T*-Frame Blind Spots

In the most general form, blind spots are volumes of the 3D space that are occluded from the viewpoint by an object in the scene. Computing these "full blind volumes" would be prohibitively expensive, especially for any application that requires real-time decision making. Even though our goal in this paper is not necessarily real-time computation at this point, a full 3D reasoning of occluded volumes would be undesirable as our target scenes are dynamic. We, instead, aim to estimate 2D blind spots on the road. Dangerous traffic situations are usually caused by unanticipated movements of dynamic objects (*e.g.*, bikes, pedestrians, children, *etc.*) springing out from road areas invisible from the driver (or the camera of the car). Once we have the 2D blind spots, we can draw attention of the drivers by, for instance, extruding it perpendicularly to the road for 3D warning. By focusing on 2D blind spot estimation, we eliminate the need of explicit 3D geometric reasoning, which makes it particularly suitable for autonomous driving and ADAS applications.

As depicted in Fig. 2, given a frame of a driving video, we define our blind spots to be the road regions that are obstructed but become visible in the future. This clearly excludes blind spots that never become visible in any frame in the future, and thus the blind spots we compute and estimate are a subset of the true blind spots. That said, they have a good coverage of the true blind spots, as pedestrians have to always go through the T-frame blind spots or visible road regions to come out in front of the vehicle. These T-frame blind spots can be reliably computed from just ego-centric driving videos. Although we only investigate the estimation of these blind spots from ego-centric driving videos

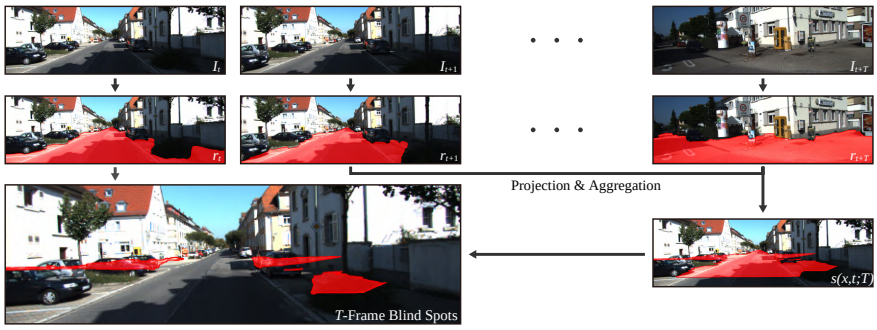

Fig. 3. Overview of algorithmic steps for generating T-frame blind spots for arbitrary driving video. Visible and invisible road maps are aggregated into the current frame from the next T frames, from which the visible road region is subtracted to obtain the T-frame blind spots for the current frame.

captured with regular perspective cameras, the wider the field of view, the better the coverage would become.

Formally, given a frame I_t, we define its blind spots as pixels $x \in \Omega_t^T$ corresponding to regions on the road Ω that are occluded but become visible in the next T frames $\{I_{t+i}|i = 1, \dots, T\}$. Our goal is to compute the set of pixels in the blind spots Ω_t as a binary mask of the image $\omega(x, t; T) : \mathbb{R}^2 \times \mathbb{R} \mapsto \{0, 1\}$. Later BlindSpotNet will be trained to approximate this function $\omega(x, t; T)$. We refer to the blind spots of this definition as *T-frame blind spots*.

As depicted in Fig. 3, we compute the blind spot map ω by first computing and aggregating visible road maps at frames $I_{t:t+T}$ and then eliminating the visible road map at target frame I_t. For this, similar to Watson *et al.* [28], we compute an aggregated road map $s(x, t; T)$ by forward warping the road pixels from the next T frames $\{I_{t+i}|i = 1, \dots, T\}$ to the target frame I_t. To perform forward warping, we assume the camera intrinsics are known, the extrinsic parameter and the depth are estimated by a visual SLAM algorithm [26] and an image-based depth estimation network [21], respectively.

Let $r(x, t)$ denote the visible road region defined as a binary mask representing the union of the road and pavement areas estimated by a semantic segmentation as introduced by Watson *et al.* [28]. We first project the visible road regions from I_{t+i} $(i = 1, \dots, T)$ to I_t as $r'(x, t + i; t)$, and then aggregate them as

$$s(x, t; T) = r'(x, t + 1; t) \vee r'(x, t + 2; t) \vee \cdots \vee r'(x, t + T; t)\,, \tag{1}$$

where $\vee$ denotes the pixel-wise logical OR. As blind spots are by definition invisible regions in frame I_t, the visible road region $r(x, t)$ is subtracted from $s(x, t; T)$ to obtain the final blind spots $\omega(x, t; T)$ by

$$\omega(x, t; T) = s(x, t; T) \wedge \bar{r}(x, t)\,, \tag{2}$$

where $\wedge$ and $\bar{\cdot}$ denote the pixel-wise logical AND and negation, respectively.

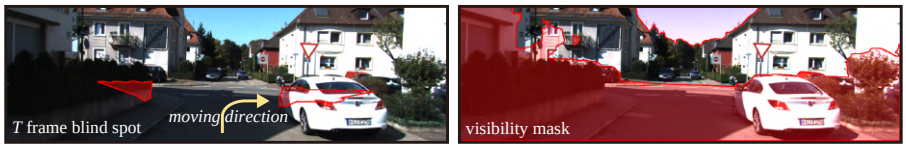

Fig. 4. Visibility mask. Left: If the vehicle makes a turn, the blind spots straight ahead across the intersection never become visible. Right: To allow networks estimate such blind spots, we prepare a mask image to indicate the visible area for each frame.

Our method relies on three visual understanding tasks, namely semantic segmentation, monocular depth estimation, and SLAM. Although existing methods for these tasks achieve high accuracy, they can still suffer from slight errors. In the transformation from r to r', we use two such estimates, those of the camera pose and the depth, whose errors can cause residuals of blind spots after Eq. (2).

We can rectify this with simple depth comparison. We first define aggregated depth mask d_a

$$d_a(x,t;T) = \frac{1}{M}\sum_{i=1}^{T} r'(x,t+i;t)d'(x,t+i;t) \qquad M = \sum_{i=1}^{T} r'(x,t+i;t)\,, \quad (3)$$

where $d'(x,t+i;t)$ is a depth map of frame I_{t+i} projected onto frame I_t, which is calculated in a similar way as calculation of $r'(x,t+i;t)$. M is the pixel-wise count of visible road mask over T frames. When we compare the depth map $d(x)$ of frame I_t with the aggregated depth mask d_a, the depth difference is large in the true blind spot region because true blind spots are occluded by foreground objects. On the other hand, it is small in erroneous blind spot regions because the compared depth values come from nearby pixels. Based on this observation, we remove erroneous blind spots by setting $\omega(x,t;T) = 0$ for the pixel x that satisfy $|d(x) - d_a(x)| < l_d$. Here, l_d is a threshold value determined empirically.

We may be left with small blind spots caused by, for instance, a shadow of a tire. These small blind spots are not important for driving safety. We remove these blind spots of less than 100 pixels from the final blind spot Ω_t.

For building our RBS Dataset, we opt for MiDAS [21] as the monocular depth estimator, OpenVSLAM [26] for SLAM, and Panoptic-DeepLab [6] for semantic segmentation. The scale of the depth estimated by MiDAS is linearly aligned with least squares fitting to the sparse 3D landmarks recovered by SLAM.

We use KITTI [9], BDD100k [31], and TITAN [18] datasets to build our RBS Dataset. By excluding videos for which the linear correlation coefficient in the MiDAS-to-SLAM depth alignment is less than 0.7, they provide 51, 62, and 118 videos, respectively. We obtain blind spot masks for approximately 51, 34, and 12 minutes of videos in total, respectively. The videos are resampled to 5 fps, and we set $T = 5$ seconds for each video. We refer to them as KITTI-RBS, BDD-RBS, and TITAN-RBS Datasets, respectively.

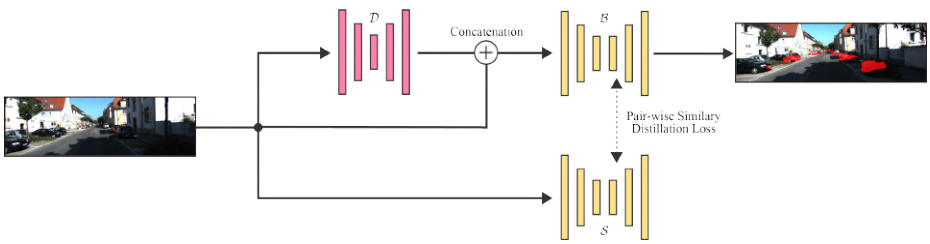

Fig. 5. Overall architecture of BlindSpotNet. We leverage the fact that blind spot estimation bears similarity to semantic segmentation by adopting a light-weight network trained with knowledge distillation from a semantic segmentation teacher network.

3.2 Visibility mask

T-frame blind spots clearly do not cover blind spots that do not become visible through the video. For example, as illustrated in Fig. 4, consider a frame where the vehicle is making a right turn. The blind spots straight ahead across the intersection never become visible and hence are not included in the dataset, while BlindSpotNet should identify them once trained. To disambiguate such invisible regions from non-blind spots in training BSN, as shown in Fig. 4, we generate a binary mask V_t called *visibility mask* for each frame in addition to the blind spots Ω_t.

We use semantic segmentation and the distance from the camera to define the visibility mask V_t. For each pixel x, we first classify x as visible, if the semantic segmentation label is "sky." For non-sky pixels, we classify x as visible if the minimum distance from the 3D point at distance $d(x,t)$ behind x to the camera is less than a certain threshold L. In our implementation, we set $L = 16$ meters.

4 BlindSpotNet

Now that we have (and can create limitless) abundant video data with per-frame blind spot annotations, we can formulate 2D blind spot detection as a learning problem. We derive a novel deep neural network for estimating blind spots in an image of a road scene. We refer to this network as BlindSpotNet and train and test it on our newly introduced RBS Dataset.

4.1 Network Architecture

As we saw in Sec. 3, blind spots are mainly determined by the scene composition of objects and their ordering in 2D. As such, at a higher level, direct image-based estimation of blind spots shares similarity in its task to semantic segmentation. The task is, however, not necessarily easier, as it is 2D dense labeling but requires implicit 3D reasoning. Nevertheless, the output is a binary map (or its probability map), which suggests that a simpler network but with a similar representation to semantic segmentation would be suitable for blind spot detection.

Fig. 5 depicts the network architecture of BlindSpotNet. BlindSpotNet consists of three components: a depth estimator $\mathcal{D}$, a semantic segmentation teacher network $\mathcal{S}$, and a blind spot estimator $\mathcal{B}$. Given an input image I of size $W \times H \times 3$, the depth estimator $\mathcal{D}$ estimates the depth map D of size $W \times H$ from I. The blind spot estimator takes both the RGB image I and the estimated depth map D as inputs and generates the blind spot map $B = \{b \in [0, 1]\}$.

The semantic segmentation network $\mathcal{S}$ serves as a teacher network to help train the blind spot estimator $\mathcal{B}$. The blind spot estimator $\mathcal{B}$ should be trained to reason semantic information of the scene similar to semantic segmentation, but its output is abstracted as simple as a single-channel map B. This implies that training the blind spot estimator $\mathcal{B}$ only with the T-frame blind spots can easily bypass the semantic reasoning of the scene and overfit. To mitigate this shortcut, we introduce the semantic segmentation network $\mathcal{S}$ pretrained on road scenes as a teacher and use its decoder output as a soft target of a corresponding layer output in the blind spot estimator $\mathcal{B}$.

4.2 Knowledge Distillation

Blind spot regions are highly correlated with the semantic structure of the scene. For instance, blind spots can appear behind vehicles and buildings, but never in the sky. The blind spot estimator $\mathcal{B}$ should thus be able to learn useful representations from semantic segmentation networks for parsing road scenes. For this, we distill knowledge from a pretrained semantic segmentation network to our blind spot estimator $\mathcal{B}$. Based on the work of Liu *et al.* [16], we transfer the similarity between features at an intermediate layer of each network.

Suppose we subdivide the feature map of an intermediate layer of size $W' \times H' \times C$ into a set of $w' \times h'$ patches. By denoting the spatial average of the features in the ith patch by $\boldsymbol{f}_i \in \mathbb{R}^C$, we define the similarity of two patches i and j by their cosine distance $a_{ij} = \frac{\boldsymbol{f}_i^\top \boldsymbol{f}_j}{\|\boldsymbol{f}_i\| \|\boldsymbol{f}_j\|}$. Following Liu *et al.* [16], given this pairwise similarity for patches in both the teacher semantic segmentation network $\mathcal{S}$ and the student network, *i.e.*, the blind spot estimator $\mathcal{B}$, as $a_{ij}^{\mathcal{S}}$ and $a_{ij}^{\mathcal{B}}$, respectively, we introduce a pair-wise similarity distillation loss l_{KD} as

$$l_{\mathrm{KD}} = \frac{1}{(w' \times h')^2} \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{R}} \left(a_{ij}^{\mathcal{S}} - a_{ij}^{\mathcal{B}} \right)^2 , \tag{4}$$

where $\mathcal{R} = \{1, 2, \ldots, w' \times h'\}$ denotes the entire set of patches. In our implementation, we opted for DeepLabV3+ [5] as the teacher network $\mathcal{S}$.

4.3 Loss Function

In addition to the similarity distillation loss l_{KD} in Eq. (4), we employ a binary cross entropy loss l_{BCE} between the output of the blind spot estimator $\mathcal{B}$ and the T-frame blind spots given by our RBS Dataset as

$$l_{\mathrm{BCE}} = -\frac{1}{|V|} \sum_{x \in V} \left(\omega(x) \log b(x) + (1 - \omega(x)) \log(1 - b(x)) \right) , \tag{5}$$

Table 1. Quantitative evaluation of RBS Datasets. The two numbers in each cell indicate the recall and the false-negative rate w.r.t. the ground truth blind spot. The results show that our T-frame blind spots approximate true blind spots well.

	CARLA		KITTI	
	Rec.↑	FN rate↓	Rec.↑	FN rate↓
T-frame BS-GT	0.372	0.013	N/A[1]	
T-frame BS (ours)	0.297	0.015	0.169	0.056

where x denotes the pixels in the visibility mask V, $\omega(x)$ and $b(x)$ denote the T-frame blind spots and the estimated probabilities at pixel x. $|V|$ is the total number of the pixels in V. The total loss function L is defined as a weighted sum of these two loss functions $L = l_{\mathrm{BCE}} + \lambda l_{\mathrm{KD}}$, where λ is an empirically determined weighting factor.

5 Experimental Results

We evaluate the validity of RBS Dataset and the effectiveness of BlindSpotNet (BSN), qualitatively and quantitatively with a comprehensive set of experiments.

5.1 RBS Dataset Evaluation

We first validate our T-frame blind spots with synthetic data generated by CARLA [7] and with real data from KITTI [9]. How well do they capture true blind spots? We use 360° LiDAR scans to obtain ground-truth blind spots (BS-GT). Note again that for real use such ground truth computation will be prohibitively expensive and would require LiDAR. We also use the ground-truth depth maps to compute T-frame blind spots without noise (T-frame BS-GT). In computing BS-GT, we find road regions in LiDAR points by fitting the road plane manually in 3D. For T-frame BS-GT, we used the ground truth semantic labels. We compare the T-frame blind spots generated by our data generation algorithm (T-frame BS) and T-frame BS-GT with BS-GT, and evaluate its quality in terms of the recall and the false-negative rate. Tab. 1 and Fig. 6 show the results. Since BS-GT is defined by sparse LiDAR points while T-frame BS-GT and T-frame BS use dense depth-maps, the precision and the false-positive (type-I error) rate do not make sense. These results show that our T-frame blind spots approximate the ground-truth blind spot well.

5.2 BlindSpotNet Evaluation

BlindSpotNet We use MiDAS [21] and DeepLabV3+ [5] as the depth estimator $\mathcal{D}$ and the semantic segmentation subnetwork $\mathcal{S}$, respectively. We use

[1] T-frame BS-GT is not available since KITTI does not have ground truth semantic segmentation for most of the frames.

Fig. 6. Comparison of T-frame blind spot and "ground truth" obtained from LiDAR data. Top-left and bottom-left figures show the ground truth and T-frame blind spots, respectively. Right figure shows the 3D projection of ground truth (yellow) and T-frame blind spots (red) to LiDAR scan (blue). The green cross indicates the camera position.

ResNet-18 and Atrous Spatial Pyramid Pooling [5] for the blind spot estimator $\mathcal{B}$ following DeepLabV3+. BlindSpotNet is trained by Adam [13] with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$, and weight decay 5×10^{-4}. The learning rate is initialized to 0.001 and a polynomial scheduler is applied. The knowledge distillation coefficient λ in Sec. 4.3 is fixed to 1.0.

We divided the 10,135 frames from KITTI-RBS into training, validation, and test sets by $8:1:1$, and the 8,872 frames from BDD-RBS into training and test sets by $8:2$. We used all the 2,655 frames from TITAN-RBS for evaluation only since each video is too short (10 to 20 seconds) to be used for training.

Metrics Blind spot estimation is a binary segmentation problem. For each input frame, our BlindSpotNet outputs the probability map of blind spots. We threshold this probability map to obtain the final binary blind spot mask, and compare it with the T-frame blind spots by IoU, recall, and precision. The threshold is determined empirically for each dataset as it depends on the road scene. We plan to learn this threshold as part of the network in future work. Notice that our RBS Datasets include blind spots that become visible in T frames only. For IoU, recall, and precision, we only consider pixels in the visibility mask.

Baselines As discussed earlier, our work is the first for image-based 2D blind spot estimation, and there are no other existing methods to the best of our knowledge. For comparison, we adapt state-of-the-art traversable region estimation [28], 2D vehicle/pedestrian/cyclist detection by semantic segmentation [5], and 3D vehicle/pedestrian/cyclist detection [4] for 2D blind spot estimation as baselines and refer to them as *Traversable*, *Detection-2D*, and *Detection-3D*, respectively.

For *Traversable* we use the hidden traversable regions, estimated by the original implementation of Watson *et al*. [28], as blind spots. *Detection-2D* is a

Table 2. Quantitative results on the test sets from KITTI-RBS, BDD-RBS, and TITAN-RBS. The lines with "w/ KITTI-RBS" and "w/ BDD-RBS" report the results by the networks trained with KITTI-RBS and BDD-RBS Datasets, respectively. These results show that depth estimation and knowledge distillation contribute independently to the final accuracy, and our BSN successfully estimates blind spots across different scenes (*i.e.*, datasets).

	KITTI-RBS			BDD-RBS			TITAN-RBS		
Model	IoU↑	Rec.↑	Prec.↑	IoU↑	Rec.↑	Prec.↑	IoU↑	Rec.↑	Prec.↑
Traversable based on [28]	0.176	0.462	0.222	0.088	0.198	0.136	0.135	0.303	0.196
Detection-2D based on [6]	0.129	0.581	0.142	0.184	**0.652**	0.204	0.142	0.435	0.182
Detection-3D based on [4]	0.182	0.368	0.265	0.059	0.067	0.316	0.048	0.057	0.216
BSN-D w/ KITTI-RBS	0.296	0.601	0.368	0.295	0.438	0.475	0.250	0.474	0.345
BSN-KD w/ KITTI-RBS	0.305	**0.646**	0.367	0.225	0.249	**0.700**	0.168	0.249	0.342
BSN (Ours) w/ KITTI-RBS	**0.330**	0.563	0.444	0.283	0.349	0.599	0.187	0.280	0.360
BSN-D w/ BDD-RBS	0.270	0.629	0.321	0.360	0.478	0.593	0.244	0.420	0.367
BSN-KD w/ BDD-RBS	0.187	0.210	**0.633**	0.350	0.443	0.624	0.253	0.529	**0.326**
BSN (Ours) w/ BDD-RBS	0.314	0.599	0.398	**0.364**	0.533	0.535	**0.257**	**0.554**	0.324

simple baseline that detects vehicle, pedestrian, and cyclist regions estimated by DeepLabV3+ [5] as blind spots. *Detection-3D* utilizes a single-image 3D detection of vehicles, pedestrians, and cyclists by Brazil *et al*. [4]. Given their detected 3D bounding boxes, *Detection-3D* returns the intersection of the projection of their far-side faces and the results by *Detection-2D* as blind spots.

We also compare with BSN without depth estimation (BSN-D), and BSN without knowledge distillation (BSN-KD) for ablation studies. In BSN-D, the depth estimator $\mathcal{D}$ is removed from BSN, and the blind spot estimator $\mathcal{B}$ is modified to take the original RGB image directly. BSN-KD disables the knowledge distillation loss by setting $\lambda = 0$ in Sec. 4.3 in BSN.

Quantitative Evaluations Tab. 2 shows the results on the test sets from KITTI-RBS, BDD-RBS, and TITAN-RBS Datasets. The lines with "w/ KITTI-RBS" and "w/ BDD-RBS" indicate the results of the networks trained with KITTI-RBS and BDD-RBS, respectively. Each network, after pre-training, was fine-tuned using 20% of the training data of the target dataset to absorb scene biases. It is worth mentioning that this fine-tuning is closer to self-supervision as the T-frame blind spots can be automatically computed without any external supervision for arbitrary videos. As such, BlindSpotNet can be applied to any driving video without suffering from domain gaps, as long as a small amount of video can be acquired before running BlindSpotNet for inference. The 20% training data usage of the target scene simulates such a scenario. Note that none of the test data were used and this fine-tuning was not done for TITAN-RBS.

BSN outperforms *Traversable*, *Detection-2D*, and *Detection-3D*. These results show that blind spot estimation cannot be achieved by simply estimating footprint or "behind-the-vehicle/pedestrian/cyclist" regions. The full BSN also performs better than BSN-D and BSN-KD. This suggests that both the depth estimator and knowledge distillation contribute to its performance independently. Furthermore, the performance of BSN w/ KITTI-RBS on BDD-RBS

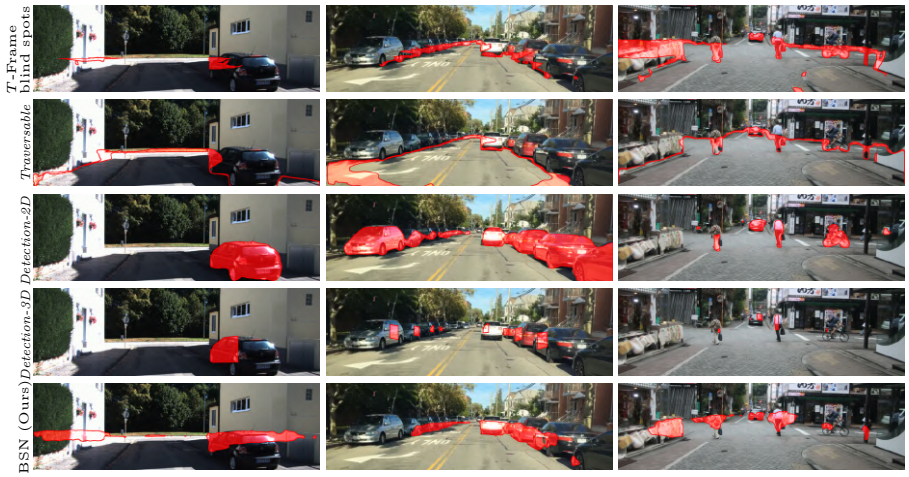

Fig. 7. Blind spot estimation results. BlindSpotNet successfully achieves high precision and recall for complex road scenes (KITTI, BDD, and TITAN from left to right) by estimating nuanced blind spots caused by parked and moving cars, intersections, buildings, poles, gates, *etc*. BlindSpotNet also estimates intersection blind spots that are even not in the "ground-truth" T-frame blind spots. This demonstrates the effectiveness of the visibility mask and the advantage of BlindSpotNet of being able to train on a diverse set of scenes thanks to the fact that T-frame blind spots can be easily computed on arbitrary driving videos.

and TITAN-RBS and that of BSN w/ BDD-RBS on KITTI-RBS and TITAN-RBS demonstrate the ability of BSN to generalize across datasets.

Qualitative Evaluations Fig. 7 shows blind spot estimation results for the test sets. Compared with baseline methods, our method estimates the complex blind spots arising in these everyday road scenes with high accuracy. It is worth noting that BlindSpotNet correctly estimates the left and right blind spots in the left column example, even though the "ground-truth" T-frame blind spots do not capture them due to the visibility mask. These results clearly demonstrate the effectiveness of the visibility mask and the training on diverse road scenes whose T-frame blind spots can be automatically computed. BlindSpotNet can be trained on arbitrary road scenes as long as T-frame blind spots can be computed, *i.e.*, SLAM, semantic segmentation, and depth estimation can be applied. In this sense, it is a self-supervised method. Fig. 8 shows a failure case example. By definition of T-frame blind spots, BlindSpotNet cannot estimate intersection blind spots in videos that do not have any turns. We plan to explore the use of wider perspective videos, including full panoramic views, to mitigate this issue.

Network Architecture Evaluation We compare large/medium-sized blind spot estimators as well as a simple U-Net [22] baseline with our small-sized

Fig. 8. Failure example. Left: T-frame blind spot ("ground truth") from a frame in BDD-RBS. Right: Blind spot estimation results from BlindSpotNet trained on BDD-RBS. Due to the bias of BDD-RBS, which lacks left and right turns at intersections, BSN cannot estimate the blind spots caused by the intersection. We plan to address these issues by employing a panoramic driving video for pre-training.

Table 3. Network architecture comparison. Our model achieves comparable performance to larger models and its frame-rate is promising for realtime processing.

Architecture	IoU	Recall	Precision	# of params	GMACS	FPS
U-Net based [22]	0.289	0.484	0.417	17.3M	280.2	139.9
Small (ours)	0.330	0.563	0.444	18.1M	47.1	37.5
Medium	0.315	0.478	0.482	32.0M	93.0	19.5
Large	0.337	0.508	0.500	59.3M	160.9	11.3

(light-weight) blind spot estimator. The differences between the large/medium-sized blind spot estimators and the small-sized one are backbone, channel size, and the number of decoder layers. The backbones of the large/medium-sized ones are ResNet101 and ResNet50, respectively. Tab. 3 lists the IoU, recall, precision, the number of the parameters, the computational complexity in GMACS, and the inference speed with a single NVIDIA TITAN V 12GB. The results show that our model achieves comparable performance to larger models with much smaller cost and runs sufficiently fast for real-time use.

6 Conclusion

We introduced a novel computer vision task for road scene understanding, namely 2D blind spot estimation. We tackle this challenging and critical problem for safe driving by introducing the first comprehensive dataset (RBS Dataset) and an end-to-end learnable network which we refer to as BlindSpotNet. By defining 2D blind spots as road regions that are invisible from the current viewpoint but become visible in the future, we showed that we can automatically compute them for arbitrary driving videos, which in turn enables learning to detect them with a simple neural network trained with knowledge distillation from a pre-trained semantic segmentation network. We believe these results offer a promising means for ensuring safer manual and autonomous driving and open new approaches to extending self-driving and ADAS with proactive visual perception.

Acknowledgements This work was in part supported by JSPS 20H05951, 21H04893, JST JPMJCR20G7.

References

1. Afolabi, O., Driggs–Campbell, K., Dong, R., Kochenderfer, M.J., Sastry, S.S.: People as sensors: Imputing maps from human actions. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2342–2348 (2018). https://doi.org/10.1109/IROS.2018.8594511
2. Barnum, P., Sheikh, Y., Datta, A., Kanade, T.: Dynamic seethroughs: Synthesizing hidden views of moving objects. In: ISMAR. pp. 111–114 (2009)
3. Bertoni, L., Kreiss, S., Alahi, A.: Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In: ICCV. pp. 6861–6871 (2019)
4. Brazil, G., Liu, X.: M3D-RPN: Monocular 3D Region Proposal Network for Object Detection. In: ICCV (2019)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: ECCV (2018)
6. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In: CVPR (2020)
7. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An Open Urban Driving Simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
8. Ehsani, K., Mottaghi, R., Farhadi, A.: Segan: Segmenting and generating the invisible. In: CVPR (2018)
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets Robotics: The KITTI Dataset. International Journal of Robotics Research (2013)
10. Guo, R., Hoiem, D.: Beyond the Line of Sight: Labeling the Underlying Surfaces. In: ECCV. pp. 761–774 (2012)
11. Hara, K., Kataoka, H., Inaba, M., Narioka, K., Hotta, R., Satoh, Y.: Predicting appearance of vehicles from blind spots based on pedestrian behaviors at crossroads. IEEE Transactions on Intelligent Transportation Systems (2021)
12. Itkina, M., Mun, Y.J., Driggs-Campbell, K., Kochenderfer, M.J.: Multi-agent variational occlusion inference using people as sensors (2021). https://doi.org/10.48550/ARXIV.2109.02173, https://arxiv.org/abs/2109.02173
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
14. Liu, B., Zhuang, B., Schulter, S., Ji, P., Chandraker, M.: Understanding Road Layout From Videos as a Whole. In: CVPR (2020)
15. Liu, C., Kohli, P., Furukawa, Y.: Layered scene decomposition via the occlusion-crf. In: CVPR (2016)
16. Liu, Y., Shu, C., Wang, J., Shen, C.: Structured Knowledge Distillation for Dense Prediction. IEEE TPAMI pp. 1–1 (2020)
17. Makansi, O., Çiçek, O., Buchicchio, K., Brox, T.: Multimodal Future Localization and Emergence Prediction for Objects in Egocentric View With a Reachability Prior. In: CVPR. pp. 4353–4362 (2020)
18. Malla, S., Dariush, B., Choi, C.: TITAN: Future Forecast using Action Priors. In: CVPR. pp. 11186–11196 (2020)
19. Mani, K., Daga, S., Garg, S., Narasimhan, S.S., Krishna, M., Jatavallabhula, K.M.: MonoLayout: Amodal scene layout from a single image. In: WACV (2020)
20. Rameau, F., Ha, H., Joo, K., Choi, J., Park, K., Kweon, I.S.: A real-time augmented reality system to see-through cars. IEEE transactions on visualization and computer graphics **22**(11), 2395–2404 (2016)

21. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. IEEE TPAMI (2020)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
23. Shivdas, S., Kelly, T.: Toyota halts all self-driving e-Palette vehicles after Olympic village accident. Reuters (2021-08-28), https://www.reuters.com/business/autos-transportation/toyota-halts-all-self-driving-e-pallete-vehicles-after-olympic-village-accident-2021-08-27/
24. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: CVPR (2017)
25. Sugiura, T., Watanabe, T.: Probable Multi-hypothesis Blind Spot Estimation for Driving Risk Prediction. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). pp. 4295–4302 (2019)
26. Sumikura, S., Shibuya, M., Sakurada, K.: Openvslam: a versatile visual slam framework. In: ACM MM. pp. 2292–2295 (2019)
27. Tighe, J., Niethammer, M., Lazebnik, S.: Scene parsing with object instances and occlusion ordering. In: CVPR (2014)
28. Watson, J., Firman, M., Monszpart, A., Brostow, G.J.: Footprints and Free Space From a Single Color Image. In: CVPR (2020)
29. Yang, W., Li, Q., Liu, W., Yu, Y., Ma, Y., He, S., Pan, J.: Projecting Your View Attentively: Monocular Road Scene Layout Estimation via Cross-View Transformation. In: CVPR. pp. 15536–15545 (2021)
30. Yasuda, H., Ohama, Y.: Toward a practical wall see-through system for drivers: How simple can it be? In: ISMAR. pp. 333–334 (2012)
31. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: BDD100K: A diverse driving dataset for heterogeneous multitask learning. In: CVPR. pp. 2636–2645 (2020)
32. Zhang, Y., Song, S., Tan, P., Xiao, J.: Panocontext: A whole-room 3d context model for panoramic scene understanding. In: ECCV. pp. 668–686 (2014)
33. Zhu, X., Yin, Z., Shi, J., Li, H., Lin, D.: Generative Adversarial Frontal View to Bird View Synthesis. In: 3DV. pp. 454–463 (2018)
34. Zhu, Y., Tian, Y., Metaxas, D., Dollar, P.: Semantic amodal segmentation. In: CVPR (2017)