VISTA: VIrtual STereo based Augmentation for Depth Estimation in Automated Driving

Bin Cheng
Blue River Technology
Sunnyvale, USA
bin.cheng@bluerivertech.com

Kshitiz Bansal
University of California
San Diego, USA
ksbansal@eng.ucsd.edu

Mehul Agarwal
Carnegie Mellon University
Pittsburgh, USA
mehula@andrew.cmu.edu

Gaurav Bansal
Blue River Technology
Sunnyvale, USA
gaurav.bansal@bluerivertech.com

Dinesh Bharadia
University of California
San Diego, USA
dineshb@ucsd.edu

Abstract

Depth estimation is the primary task for automated vehicles to perceive the 3D environment. The classical approach for depth estimation leverages stereo cameras on the cars. This approach can provide accurate and robust depth estimation, but also requires a more expensive setup and detailed calibration. The recent trend of depth estimation, therefore, focuses on learning the depth from monocular videos. These approaches only need an easy setup but may also be vulnerable to occlusion or light condition changes in the scene. In this work, we propose a novel idea that exploits the fact that data collected by large fleets naturally contains scenarios where vehicles with monocular cameras drive close to each other and are looking at the same scene. Our approach combines the monocular view of the ego vehicle and the neighboring vehicle to form a virtual stereo pair during training, while still only requiring the monocular image during inference. With such a virtual stereo view, we are able to train self-supervised depth estimation by two sources of constraints: 1) the spatial and temporal constraints between sequential monocular frames; 2) the geometric constraints between the frames from two cameras that form the virtual stereo.

Public datasets for multiple vehicles sharing the common view to form possible virtual stereo views do not exist, and so we also created our synthetic dataset using CARLA simulator where multiple vehicles can observe the same scene at the same time. The evaluation shows that our virtual stereo approach can improve the ego vehicle's depth estimation accuracy by 8%, compared to the approaches that use monocular frames only.

1 Introduction

Accurate depth Estimation is essential for 3D autonomous perception. Downstream tasks like motion planning rely on accurate object depth for scene understanding. A LiDAR based setup is gener-

Machine Learning for Autonomous Driving Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, USA.

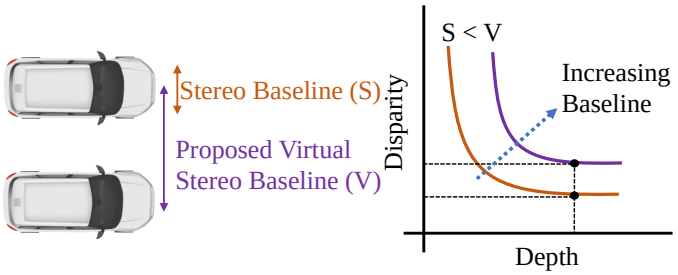

Figure 1: Virtual stereo provides larger baseline for depth estimation which generate larger disparity for any given depth value. Larger the disparity implies more sensitive the network is to depth errors improving depth estimation

ally used for depth estimation and predicts a sparse point cloud that is then fused with camera data by the means of projection. This approach is most accurate but requires precise extrinsic calibration between the LiDAR sensor and the camera, making it expensive. It is also limited by sparse information and phantom points caused by reflective surfaces. Other approaches rely on a calibrated stereo camera setup that uses point correspondence based traditional computer vision techniques or deep learning models for directly estimating depth. Recently monocular depth estimation has also produced encouraging results using deep learning algorithms that rely on supervised depth maps from lidar or unsupervised view-synthesis algorithms for training Bian et al. [2019], Garg et al. [2016], Eigen et al. [2014]. However monocular depth is an ill-posed problem and predicts depth to a scale. It is also not as accurate as stereo or LIDAR and is very much prone to the downsides of data-driven approaches like the inability to generalize to new scenarios or new domains.

A fairly new approach to monocular depth estimation that has shown promise, involves training a deep learning model with a calibrated stereo setup using self-supervised photometric loss and performing inference on monocular images, resulting in a more accurate depth estimation model. However, this approach, though accurate, still faces two main challenges. The first challenge with this approach is that it requires data collection with stereo cameras on-board. Many automotive companies are doing data collection using their live fleet, and adding a stereo camera to the fleet will add substantial cost. The second challenge is, for stereo, the depth resolution is directly proportional to the baseline or the distance between the cameras Hartley and Zisserman [2000]. A wider baseline is needed for better depth estimation at large distances, which is of prime importance in automotive driving use cases. However, the physical constraints of a typical passenger car limits how wide the stereo baseline could be.

In this paper, we propose a novel virtual stereo approach that exploits monocular cameras on different nearby vehicles to construct a *virtual stereo* pair. Our formulation is based on a practical realization that as automotive companies are collecting data with the large live fleet, there will always be scenarios when two vehicles in the fleet are driving close to each other with a certain overlap in the scene. For example, this would be a common occurrence in the cloud database of autopilot compliant vehicles like Tesla, where there would be millions of instances where two Tesla vehicles were driving close to each other and looking at the same scene from different viewpoints Herger [2017]. A real world example of such a scenario is Ford AV dataset where multiple vehicles drive close-by, looking at the same scene Agarwal et al. [2020]. Similarly, companies like Nexar are heavily investing in developing crowd-sourcing based applications using dash-cam videos. They are gathering the dash cam video footage originating from multiple vehicles operating in same zone to enable these applications Nexar. Our solution exploits this data, that contains scene overlap in neighboring vehicles, to construct virtual stereo during training and improves the self-supervised models with additional geometric constraints. The improved self-supervised model is used for inference on monocular images achieving better accuracy.

We illustrate the natural advantage of a wider baseline produced by virtual stereo with the plot in Figure 1. The depth of a point under observation is a function of the disparity in pixels of the projection of that point between the two stereo images. The larger the baseline, the larger is the disparity for the same depth of the point. The photometric loss function is dependent on this disparity

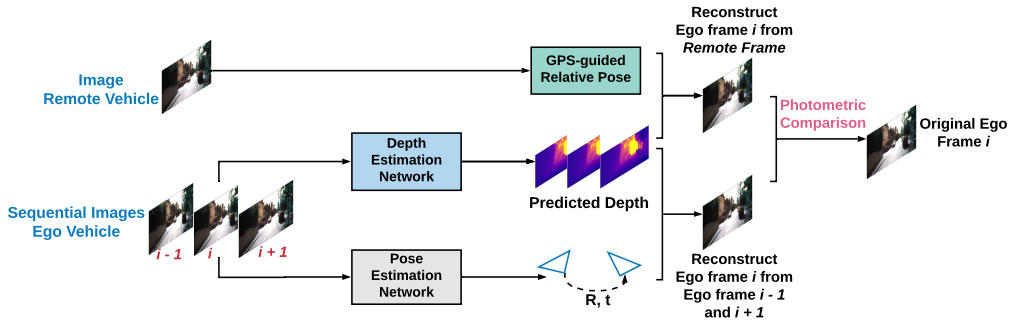

Figure 2: Overview of Architecture

generated between the target and source image due to the depth of the point. Model is more sensitive to the errors in depth estimation if the corresponding disparity generated is larger because even small errors in the depth would meaningfully change the disparity and hence the photometric loss. For self-supervised monocular depth estimation, the disparity generated at large depth values is small due to the smaller baseline. Large stereo baselines also lead to less overlap and a significant part of the image with invalid disparity. However, in our framework, we use virtual stereo pair as an additional loss constraint during training, and this allows us to only apply the loss constraint for pixels with overlap i.e. valid disparity. Hence, our insight in this work is that virtual stereo can be effectively used for training and allows us to use stereo with larger baselines, enabling to train on loss constraint sensitive to larger depth values and improving the overall depth estimation performance.

We utilize virtual stereo pair to augment the ego vehicle's self-supervised depth learning by adding additional photometric loss in training using neighboring vehicle's view. Given relative GPS measurements or global pose guided transformation between two vehicles, the estimated depth can reconstruct the ego vehicle's frame from the frame of the neighboring vehicle. The photometric loss between the reconstructed image and the original image provides enhanced supervision on the depth estimation. It is possible the images from the two cameras of a virtual stereo pair are substantially different, e.g., due to occlusion or non-co-visible objects. To account for these scenarios, we propose an image similarity filter to discard image pairs that cannot form a reliable virtual stereo pair due to lack of sufficient similarity.

Another challenge we face is the absence of datasets that can be used for training virtual stereo. Automotive companies can create such a dataset from their proprietary data. But, to the best of our knowledge, there are no publicly available datasets that can be used for training our virtual stereo approach. Hence, we created our own dataset using the CARLA simulator that generates realistic scenes with traffic and pedestrians in different urban scenarios and weather conditions. Our training dataset is created for two typical urban town scenarios with 200 cars each in the scene and multiple lanes for each direction, creating a dataset of more than 50000 training images and 1000 images on a separate town as a test set. We compared our virtual stereo approach with approaches trained with only self-supervised monocular depth estimation as well as stereo depth estimation. We show that our approach can outperform the network trained using the monocular-only approach by 8% and performs equivalent to the network trained on stereo images.

2 Related Work

Depth estimation algorithms exploit spatial and temporal cues from cameras mounted on one vehicle or device to regress dense depths. Our approach extends the multi-view training signal by using monocular images from cameras mounted on different vehicles creating a virtual stereo pair with a large baseline.

2.1 Depth Estimation

Both supervised and self-supervised approaches are proposed for monocular depth estimation. The supervised solutions Eigen et al. [2014], He et al. [2018], Repala and Dubey [2018], Liu et al. [2016], Xu et al. [2018a,b], Fu et al. [2018], Xian et al. [2018] learn a direct mapping between an

input RGB image and a depth map by comparing the estimated depth map and its corresponding ground-truth. However, ground-truth depth maps are prohibitively expensive to obtain. On the other hand, self-supervised approaches estimate depth by extracting depth cues from stereo image pairs or monocular videos. Garg *et al*. Garg et al. [2016] introduced a warping loss based on Taylor expansion. An image reconstruction loss with a spatial smoothness constraint was introduced in Ren et al. [2017], Zhou et al. [2017a], Jason et al. [2016] to learn depth and camera motion. Recent works Vijayanarasimhan et al. [2017], Zhou et al. [2017b], Mahjourian et al. [2018], Godard et al. [2017, 2019] aim to improve depth estimation by further exploiting geometry constraints. In particular, Godard *et al*. Godard et al. [2017] employed epipolar geometry constraints between stereo image pairs and enforced a left-right consistency constraint in training the network. Yin *et al*. Yin and Shi [2018] proposed GeoNet, which also used depth and pose networks in order to compute rigid flow between sequential images in a video. More specifically, they introduced a temporal, flow-based photometric loss to predict depth for monocular videos in an unsupervised setting. This performed better than the photometric consistency loss of Zhou et al. [2017a]. We have used an adaptation of this warping loss as part of our unsupervised temporal constraints. Bian *et al*. Bian et al. [2019] used a similar approach along with a self-discovered mask to handle dynamic and occluded objects. Gordon *et al*. Gordon et al. [2019] also addresses these issues using a purely geometric approach. Casser *et al*. Casser et al. [2018] adapts a similar framework with an additional online refinement model during inference. Xu *et al*. Xu et al. [2019] proposed region deformer networks along with the earlier constraints to handle rigid and non-rigid motion. Zhou *et al*. Zhou et al. [2019] exploited a dual network attention-based model which processes low and high-resolution images separately.

2.2 Multi-view Perception

In literature, the relative camera pose estimation problem has been studied in the context of multiple onboard cameras on the same vehicle or robot. These solutions exploit the large scene overlap and are typically estimated by finding keypoint/feature or line correspondences between two images Nister [2004], Long Quan and Zhongdan Lan [1999], Elqursh and Elgammal [2011]. These features are usually detected using SIFT Lowe [2004], SURF Bay et al. [2008] like algorithms and work well for small changes in viewpoint. Recently, deep learning-based approaches have been proposed for camera pose estimation. In PoseNet Kendall et al. [2015] a convolutional neural network (CNN) was used for learning the 6 degrees of freedom (DoF) pose. The network was shown robust to lighting, motion blur and knowledge of intrinsic camera parameters. Several extensions of PoseNet have been proposed in Kendall and Cipolla [2015], Walch et al. [2016], Li et al. [2018]. The pose estimation network in our approach follows a similar design, where the 6 DoF relative pose is learned from input image pairs or image sequences.

3 Methods

Self-supervised depth estimation requires supervision from motion constraints to train a network for the task. Current solutions use the series of images generated by a monocular camera and leverage the temporal relationship between them along with motion information to guide training. However, the cues used by this type of solution to learn depths can be easily affected by object occlusion, object movement, or light condition change. Also, the depth learned from monocular cameras inherits ambiguity due to various scenes with different depths that may project to the same image. Stereo cameras can help resolve these issues. However, a conventional stereo setup requires significant amounts of calibration efforts and higher costs, which is less cost-efficient compared to monocular cameras. On the other hand, our proposed approach leverages the camera data from a nearby vehicle to enhance the perception capability of the ego vehicle, specifically the accuracy of depth estimation. Our proposed approach breaks the problem into the following parts: First, the ego vehicle selects the images from a neighboring vehicle that captures a similar scene. Then, the images from both the ego vehicle and the neighboring vehicle can serve as *virtual stereo* pairs. These virtual stereo image pairs further provide additional stereo constraints on monocular depth estimation of the ego vehicle, which potentially boosts the estimation accuracy.

3.1 Depth Estimation with Virtual Stereo

In our proposed approach, an ego vehicle's depth estimation is guided by two losses: 1) the loss based on the spatial and temporal relationship of a few sequential images; 2) the loss based on the virtual stereo pair.

3.1.1 Monocular Depth Estimation

In our model, the monocular depth estimation is established by enforcing the geometric constraints between sequential images. The corresponding pixel coordinates representing points on a rigid object in two consecutive frames are subject to

$$\mathrm{p}_{t+1} = K\mathbf{T}_{t\to t+1}(\mathbf{D}_T(\mathrm{p}_t) \times K^{-1}\mathrm{p}_t) \tag{1}$$

where p_t and p_{t+1} are the coordinates of a pixel in frames at time t and $t+1$, K denotes the camera intrinsic parameters, and $\mathbf{T}_{t\to t+1}$ represents the relative camera pose between frame t and $t+1$. Based on pixels p_{t+1} in frame $t+1$, frame t can be reconstructed. The photometric difference (denoted by L_{pe}) between the reconstructed frame and the original frame t drives the depth estimation.

$$\mathcal{L}_{pe} = \alpha||\hat{\mathcal{S}}_t - \mathcal{S}_t||_1 + (1-\alpha)\mathbf{DSSIM}(\hat{\mathcal{S}}_t, \mathcal{S}_t) \tag{2}$$

where $\hat{\mathcal{S}}_t$ and $\mathcal{S}_t$ are the reconstructed and the original frame t respectively. DSSIM Zhou Wang et al. [2004] is applied to measure the similarity between two frames.

In addition, the object boundary observed in RGB images should be also preserved in estimated depth maps. The difference of object boundaries and pixel gradients in RGB images and depth maps is denoted by

$$\mathcal{L}_{smooth} = \mathbf{smoothing\_loss}(\mathcal{S}_t, \mathbf{D}_T(\mathcal{S}_t)) \tag{3}$$

where $\mathcal{S}_t$ and $\mathbf{D}_T(\mathcal{S}_t)$ are the RGB frame and its corresponding estimated depth map, respectively. Following Guizilini et al. [2020], we apply the function $\mathbf{smoothing\_loss}(*)$ to compute the smooth loss.

Overall, the total loss for monocular depth estimation is

$$\mathcal{L}_{mono} = \alpha_{pe}\mathcal{L}_{pe} + \alpha_{smooth}\mathcal{L}_{smooth} \tag{4}$$

3.1.2 Virtual Stereo Depth Estimation

In this work, we augment the monocular depth estimation by utilizing the images captured by nearby cameras to form *virtual* stereo image pairs. Two main steps involved in the process: 1) relative pose estimation; and 2) remote image warping.

Ego-Remote Relative Pose Estimation The relative pose between cameras on the ego vehicle and the remote vehicle in a virtual stereo pair is the essential element to transform the estimated depth and pixel values in one camera's coordinate to another. Similar to the relative pose estimation from sequential images, the relative pose between the ego camera and the remote camera is also learned directly from input image pairs.

Additionally, we propose to guide the relative pose learning with GPS and other sensor readings from the two vehicles. These sensor data can provide highly accurate information regarding the vehicle's latitude, longitude, roll, yaw, and pitch. One may argue that GPS readings are prone to environmental interference such as clouds, tall buildings, etc., and thus they are less reliable and credible. However, as shown in previous studies Rostami et al. [2019], the error of GPS readings demonstrates a strong temporal and spatial correlation. As a result, the GPS errors of the two nearby vehicles within the same time window are less affected by the environmental errors. Based on the readings of the vehicle sensors, the camera pose on a vehicle can be obtained by transforming the vehicle's GPS readings. The pose for the two cameras are written then in homogeneous form as a 4×4 matrix T_e^w and T_r^w, respectively:

$$\mathbf{T}_e^w = \begin{bmatrix} R_e^w & t_e^w \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{T}_r^w = \begin{bmatrix} R_r^w & t_r^w \\ \mathbf{0_{1\times 3}} & 1 \end{bmatrix} \tag{5}$$

where R_e^w and t_e^w represent the rotation and translation of the ego vehicle's camera in the world coordinates, respectively. Similarly, R_r^w and t_r^w are the rotation and translation for the remote vehicle's camera, respectively. The relative pose between two cameras can be as

$$\mathbf{T}_e^r = \mathbf{inv}(\mathbf{T}_r^w) \cdot \mathbf{T}_e^w \tag{6}$$

Here, $\mathbf{inv}(*)$ represents the inverse operation of a matrix. In our implementation, we use this GPS-guided relative pose as a weak supervision signal in learning the relative pose between the ego and the remote cameras.

Remote Image Warping With the relative pose between the ego camera and the remote camera, we can reconstruct an ego camera's frame by warping the corresponding frame captured by the remote camera.

$$\mathrm{p}^r = K^r \mathbf{T}_e^r \cdot (\mathbf{D}_T(\mathrm{p}^e) \times (K^e)^{-1} \mathrm{p}^e) \tag{7}$$

Eq. 7 transforms a pixel coordinate on an ego camera's frame to its correspondence on the remote camera's frame, where K^e and K^r are camera intrinsic matrix of the ego camera and the remote camera; $\mathbf{D}_T(\mathrm{p}^e)$ represents the corresponding depth value of pixel p^e. By sampling pixels in the remote camera's frame according to p^r, we can reconstruct an ego camera's frame, denoted by $\hat{\mathcal{S}}$. Therefore, the photometric loss between the warped frame $\hat{\mathcal{S}}$ and its original frame $\mathcal{S}$ can further provide an additional constraint on depth estimation. The loss of the virtual stereo is defined as

$$\mathcal{L}_{v\_stereo} = \alpha ||\hat{\mathcal{S}} - \mathcal{S}||_1 + (1-\alpha)\mathbf{DSSIM}(\hat{\mathcal{S}}, \mathcal{S}) \tag{8}$$

Total Loss Combining the loss from both monocular and virtual stereo component, we have

$$\mathcal{L}_{total} = \mathcal{L}_{mono} + \alpha_{v\_stereo}\mathcal{L}_{v\_stereo} \tag{9}$$

where α_{v_stereo} is used to adjust the importance of loss $\mathcal{L}_{v\_stereo}$.

4 Evaluation Methodology and Virtual Stereo in the Wild

In our approach, we use the virtual stereo to augment the offline training for depth estimation. Having stereo images from nearby vehicles provides better supervision for training and thus increase our accuracy. A simple way to use two images as a virtual stereo pair is to find overlap between images. If the overlap is significant, then we can choose this pair for training. However, it would be computationally expensive to try a pair-wise comparison with each image. Instead, each image/video is often stored with GPS location/locations, we therefore would monitor the GPS location and heading to filter out cars traveling in same direction, and use that to get first order filtered data. Even with GPS filtering, there is a big challenge with this approach. A pair of images would benefit virtual stereo, only when they have significant matching features. This is the same issue that is faced by a real stereo image system when estimating depth for a textureless plane. Clearly, we can't use any stereo pair with enough overlap to train our network. In this section, we will present our approach to identify and filter useful virtual stereo pairs for training, based on feature matching.

4.1 Image Similarity-based Stereo Pair Selection

One of the most widely used techniques to describe an image is to identify key-points in an image, which capture the different features present in that image. This technique is commonly used in SLAM and image matching problems. We use SIFT (Scale Invariant Feature Transform) to identify several keypoints and their descriptors in an image. For each candidate virtual stereo pair, we extract the (keypoint, descriptor) tuples. These keypoints are matched across the two images to find out common keypoints. Each of these matches has a particular distance in feature space. The higher the distance, the worse is the match. We pick the top 50 matches out of these and average their distances to get the image similarity number for the candidate pair. If this number is greater than a certain threshold then we select the candidate virtual stereo pair for training.

Sky removal: Another major challenge with this approach is the presence of the sky. Between the two images of virtual stereo pair, a large section of the image is the sky, which is quite similar. The

keypoints associated with the sky have lesser distances as they match quite well. Due to this, they saturate the image similarity index, leaving little room for the actual environmental features to have an effect on the image similarity index. Hence, we need to remove these sky-based matches to get a better estimate of image similarity. We observe that though the keypoints belonging to the sky match quite closely, they belong to the region of the image that has a very high depth value. Pixels corresponding to very high depth have a very small disparity between the location of pixels in stereo images. Hence we filter out all the small disparity matches and effectively remove all the sky-based matches.

In conclusion, reaping the benefits of virtual stereo requires searching for pairs of images that have an overlap in the scene with similar features. Using a geo-location-based filter will provide us with a pool of potential candidates for training images. The image similarity filter finally helps us select the images with similar features from the above pool, that can be used for virtual stereo training.

5 Dataset Generation

Currently, there are no publicly available datasets for our training, where multiple cars are collecting data and storing with GPS coordinates in the cloud. In this work, therefore we used the CARLA simulation engine Dosovitskiy et al. [2017] to generate our own dataset for the task of depth estimation using a virtual stereo vehicle pair [1]. The CARLA simulation engine provides realistic scene rendering for autonomous driving. We collected our dataset by rendering several urban driving scenarios with vehicles, pedestrians, trees, roads, buildings, fences etc. The townscapes were chosen to include significant complex structures such as a 5-lane junction, a roundabout, unevenness, a tunnel, and a bridge to better model common real-world scenarios. We used the following method to generate our training dataset:

First, we obtained a series of images from multiple cars traveling in the urban environment. To ensure correspondence to realistic scenarios, we tried to model our dataset to the widely used KITTI Menze and Geiger [2015] dataset. We followed KITTI's camera configurations (the height, FoV of the camera, and other parameters are the same as in the KITTI dataset). We ensured that our dataset has relatively narrow roads, a considerably high number of pedestrians and (stationary) cars along the road. Figure 3 shows the comparison of depth distribution between our collected dataset and the KITTI dataset for three sample classes of *Vehicles*, *Vegetation*, and *Traffic sign*. As shown, our collected dataset closely follow real-world data depth distribution.

Each vehicle was equipped with RGB, semantic segmentation, and depth cameras. For GPS readings, we use the world coordinates provided by CARLA and add errors to emulate real-world a GPS sensor. Using these, we recorded each vehicle's RGB, depth, relative pose in global coordinates, acceleration, velocity, and semantic segmentation. Additionally, we also equipped all the vehicles with stereo cameras to compare with stereo performance with a baseline of 3m. We use Tesla model 3 with dimensions of length = 184.8 in, Width = 72.8 in and Height = 56.8 in as our ego vehicle. Pose information is used to simulate GPS location for evaluation purposes. The ground-truth GPS measurements are added with error as modeled similar to Rostami et al. [2019], Joubert et al. [2020]. We then use the filtering techniques as explained in the previous section to get a data-set with images from two cars (ego-vehicle and nearby vehicle), moving next to each other. We gathered around **50,000+** pairs of associated object images from different scenes out of all the data we collected. We only use this filtered virtual stereo pair-based data for training our network. Finally, for our test set, we collected **1000** images from a single car moving in a different urban environment, to ensure the generalizability of our approach. Note that the test set contains only monocular images.

6 Implementation Details and Experiment

Model Architecture We build our virtual stereo architecture based on PackNet Guizilini et al. [2020], a state-of-the-art self-supervised depth estimation architecture. PackNet consists of two main components, a depth estimation module and a contextual pose estimation module. The depth estimation module follows an encoder-decoder architecture and incorporates several packing and unpacking blocks with skip connections to facilitate gradient flow. The contextual pose estimation

[1] We plan to release our datasets to be used by the research community. Refer to Supplementary material for more information

module takes a target image and a few contextual images as input and regresses a 6-DOF pose. Our pose estimation network consists of seven convolutional layers and one 1×1 convolutional layer. Additionally, we add an ego-remote relative pose estimation module to learn the relative pose between the ego camera and a remote camera that is mounted on a nearby vehicle. The design of this module largely follows the architecture proposed in Zhou et al. [2017a]. Similar to the left-right consistency proposed in Godard et al. [2017], we augment the monocular depth estimation by injecting the ego-remote consistency loss in learning.

Model Training We implement our models using PyTorch 1.6 Paszke et al. [2019] and select Adam method Kingma and Ba [2014] as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate of the depth estimation module, the contextual pose estimation module and the ego-remote relative pose estimation module are set to 2×10^{-4}. In the depth estimation module, the SSIM weight, α, is set to 0.85 and the smoothing loss weight, α_{smooth}, is set to 10^{-3}. As to the contextual pose estimation module, we use one center image at time t and two contextual images at time $t-1$ and $t+1$ as input images. For the ego-remote relative pose estimation module, we take one image from the ego vehicle and one from the virtual stereo pair as input. The weight of virtual stereo α_{v_stereo} is set to 0.05 in our experiments. We select 59169 images to form the training set and evaluate our model performance on 1000 images. The input image size is 640×192. We train our model on four Nvidia 1080Ti GPUs. The training takes 1.3 hours for each epoch, while in the testing phase, the model runs at 23.15 images/s.

Method	Error-related metrics				Accuracy-related metrics		
	Abs Rel	**Sq Rel**	**RMSE**	**RMSE log**	$\delta < \mathbf{1.25}$	$\delta < \mathbf{1.25^2}$	$\delta < \mathbf{1.25^3}$
depth capped at 80m							
Monocular Only	0.1132	1.5978	7.7336	0.2815	0.8370	0.9039	0.9455
Monocular + **Virtual** Stereo	0.1041	1.4775	7.2735	0.2595	0.8528	0.9154	0.9516
Monocular + **Local** Stereo	0.1059	1.5214	7.4440	0.2710	0.8454	0.9045	0.9476
depth capped at 120m							
Monocular Only	0.1254	2.5955	12.3015	0.3402	0.8214	0.8873	0.9290
Monocular + **Virtual** Stereo	0.1146	2.3312	11.3786	0.3118	0.8370	0.9001	0.9379
Monocular + **Local** Stereo	0.1173	2.4506	11.8120	0.3278	0.8296	0.8883	0.9324

Table 1: Depth estimation performance comparison between our virtual stereo approach, monocular only approach and the conventional local stereo approach

7 Depth Estimation Evaluation

Quantitative Comparison Table 1 shows a performance comparison between the virtual stereo approach and the monocular-only approach with respect to various performance metrics. The comparison is capped to 80 m and 120 m, respectively. One can observe that the virtual stereo approach outperforms the monocular-only approach by 8%. It is because the virtual stereo pairs provide additional geometric constraints while learning the depth from the input images. These additional constraints help enhance the self-supervised models, which otherwise have weak depth cues by monocular sequential images. Additionally, we show that the virtual stereo approach (enhanced self-supervised model) can achieve better performance on monocular images as compared to the local stereo approach where stereo camera images are used with a stereo-depth estimation algorithm (mounted on the same car). But, the virtual stereo approach is more cost-efficient as only low-cost monocular cameras are required in the system setup.

Qualitative Comparison In Figure 4, we present the qualitative comparison between the local stereo approach and our virtual stereo approach. The selected scenarios are challenging as the light condition is dynamically changing. One can observe that our virtual stereo approach can produce comparable or even better depth estimation than the local stereo approach. The observation visually validates the effectiveness and robustness of our approach.

Robustness Study As described in Section 4, our proposed approach first implicitly selects candidate training images based on their host vehicles' GPS readings. Moreover, we also use GPS measurements aggregated with other sensor readings like IMU, to guide the learning of ego-remote relative pose. The two processes are subject to GPS errors. To understand the importance and impact of the GPS readings to our model, we present the depth estimation performance under various levels

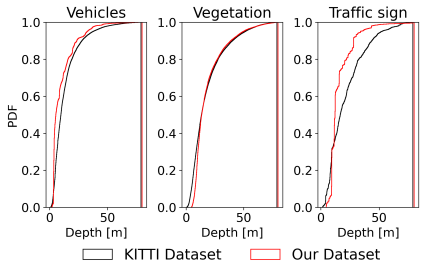

Figure 3: Depth distribution comparison between our dataset and the KITTI dataset

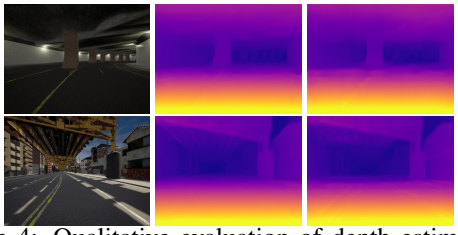

Figure 4: Qualitative evaluation of depth estimation: column (a): input RGB images; column (b): predicted depth maps using local stereo approach; column (c): predicted depth maps using virtual stereo approach

of GPS errors in Table 2. Note that, in this study, we primarily focus on the impact of GPS readings on the relative pose learning.

Relative Rotation Error	Relative Translation Error	Error-related metrics				Accuracy-related metrics		
		Abs Rel	**Sq Rel**	**RMSE**	**RMSE log**	$\delta < \mathbf{1.25}$	$\delta < \mathbf{1.25^2}$	$\delta < \mathbf{1.25^3}$
w/o GPS supervision	w/o GPS supervision	0.1062	1.5720	7.2470	0.2612	0.8545	0.9166	0.9522
$\mathcal{N}(\mu = 0, \sigma = 0.0)$	$\mathcal{N}(\mu = 0, \sigma = 0.0)$	0.1040	1.4554	7.0095	0.2513	0.8586	0.9230	0.9569
$\mathcal{N}(\mu = 0, \sigma = 0.01)$	$\mathcal{N}(\mu = 0, \sigma = 0.5)$	0.1041	1.4775	7.2735	0.2595	0.8528	0.9154	0.9516
$\mathcal{N}(\mu = 0, \sigma = 0.1)$	$\mathcal{N}(\mu = 0, \sigma = 5.0)$	0.1068	1.4715	7.0729	0.2514	0.8561	0.9178	0.9542

Table 2: Performance comparison with different error models in GPS-guided pose estimation

In this study, we separately model the rotation error and the translation error, and inject errors to the ground-truth GPS readings from CARLA simulations. For the rotation error, it is generated by the vehicle's pitch, roll, and yaw measurements derived from GPS measurements and can be adjusted by the vehicle's internal inertial sensors. For translation error, it is primarily computed based on the GPS readings from the both nearby vehicles. Both errors are modeled following the model in Joubert et al. [2020]. Comparing row 1 and row 2 in Table 2, with near perfect GPS readings, the absolute relative error is improved by an additional 2% with respect to the cases where GPS information is not applied in relative pose estimation. We then extended the study to larger GPS errors. With translation error of 50 centimeters in row 3, only marginal performance degradation is observed. In row 4, good performance remains even under the challenging scenarios with extreme high GPS errors. The core insight from this study is that our GPS-guided pose estimation algorithm is robust and reliance on GPS readings. Note that the GPS errors mentioned in the paper are relative errors instead of absolute errors. Two adjacent GPS receivers will be affected by similar environmental interference, e.g., cloud blockage, and thus the errors of the two GPS readings will suffer from similar biases. Therefore, even if the absolute GPS errors are high, the errors for the relative positioning of the two vehicles may still be low. This has been experimentally validated in Ahmed-Zaid et al. [2011].

8 Conclusion and Future Work

In this paper, we propose a simple yet practical depth learning augmentation approach for self-supervised depth estimation. In the proposed approach, monocular images collected from neighboring vehicles are leveraged to form virtual stereo pairs, and ego vehicle's depth estimation is augmented by adding additional photometric constraints provided by the neighboring vehicle's view. We argue that this is a practical approach given automotive companies are collecting camera data with the large live fleet and there should be many instances where naturally two or multiple vehicles drive close to each other while collecting data observing the same scene. We have also made a contribution in this paper by creating the first virtual stereo dataset using the CARLA simulator. The dataset contains more than 50000 training images and 1000 test images. We compare our virtual stereo approach with approaches only trained with monocular images as well as conventional stereo approaches. The evaluation results show that our virtual approach can achieve around 8% performance gain. As future work, we will extend our datasets to more traffic and road scenarios.

References

Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in Neural Information Processing Systems*, pages 35–45, 2019. 2, 4

R. Garg, G. VijayKumarB., and I. D. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 4

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 2, 3

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2000. ISBN 0521623049. 2

Mario Herger. Tesla starts massive data collection activity on customer cars via autopilot hardware kit 2. https://thelastdriverlicenseholder.com/2017/06/14/tesla-starts-massive-data-collection-activity-on-customer-cars-via-autopilot-hardware-kit-2/, 2017. 2

Siddharth Agarwal, Ankit Vora, Gaurav Pandey, Wayne Williams, Helen Kourous, and James McBride. Ford multi-AV seasonal dataset. *The International Journal of Robotics Research*, 39(12):1367–1376, sep 2020. doi: 10.1177/0278364920961451. URL https://doi.org/10.1177%2F0278364920961451. 2

Nexar. Getting camera data out of the car and into the world. https://f.hubspotusercontent20.net/hubfs/8456425/Car%20OEM%20E-book%20V4.pdf. 2

Lei He, Guanghui Wang, and Zhanyi Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, 27(9):4676–4689, 2018. 3

Vamshi Krishna Repala and Shiv Ram Dubey. Dual cnn models for unsupervised monocular depth estimation. *arXiv preprint arXiv:1804.06324*, 2018. 3

Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, 2016. 3

D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018a. 3

D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci.Structured. attention guided convolutional neural fields for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b. 3

H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

K. Xian, C. Shen, Z. Cao, H. Lu, Y. Xiao, R. Li, and Z. Luo. Monocular relative depth perception with web stereo data supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, volume 3, page 7, 2017. 4

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017a. 4, 8

J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 4

S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. preprint, 2017. 4

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017b. 4

R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 4, 8

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3838, 2019. 4

Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 4

Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras, 2019. 4

Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos, 2018. 4

Haofei Xu, Jianmin Zheng, Jianfei Cai, and Juyong Zhang. Region deformer networks for unsupervised depth estimation from unconstrained monocular videos, 2019. 4

Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Unsupervised high-resolution depth learning from videos with dual networks, 2019. 4

D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, June 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.17. 4

Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, Aug 1999. ISSN 0162-8828. doi: 10.1109/34.784291. 4

A. Elqursh and A. Elgammal. Line-based relative pose estimation. In *CVPR 2011*, pages 3049–3056, June 2011. doi: 10.1109/CVPR.2011.5995512. 4

David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2): 91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94. 4

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.09.014. URL http://dx.doi.org/10.1016/j.cviu.2007.09.014. 4

A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, Dec 2015. doi: 10.1109/ICCV.2015.336. 4

Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. *CoRR*, abs/1509.05909, 2015. URL http://arxiv.org/abs/1509.05909. 4

Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization with spatial lstms. *CoRR*, abs/1611.07890, 2016. URL http://arxiv.org/abs/1611.07890. 4

Ruihao Li, Qiang Liu, Jianjun Gui, Dongbing Gu, and Huosheng Hu. Indoor relocalization in challenging environments with dual-stream convolutional neural networks. *IEEE Transactions on Automation Science and Engineering*, 15:651–662, 2018. 4

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. 5

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation, 2020. 5, 7

A. Rostami, B. Cheng, H. Lu, J. B. Kenney, and M. Gruteser. A light-weight smartphone gps error model for simulation. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–5, 2019. doi: 10.1109/VTCFall.2019.8891089. 5, 7

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 7

Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 7

N. Joubert, T. G. R. Reid, and F. Noble. Developments in modern gnss and its impact on autonomous vehicle architectures. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 2029–2036, 2020. doi: 10.1109/IV47402.2020.9304840. 7, 9

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. 8

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

F Ahmed-Zaid, F Bai, S Bai, C Basnayake, B Bellur, S Brovold, G Brown, L Caminiti, D Cunningham, H Elzein, et al. Vehicle safety communications–applications (vsc-a) final report. Technical report, 2011. 9