Left-Right Consistent Monocular Depth Estimation Using Semi-Supervised Deep Learning

Ze Ran Lu
University of Waterloo
Waterloo, Canada
zrlu@uwaterloo.ca

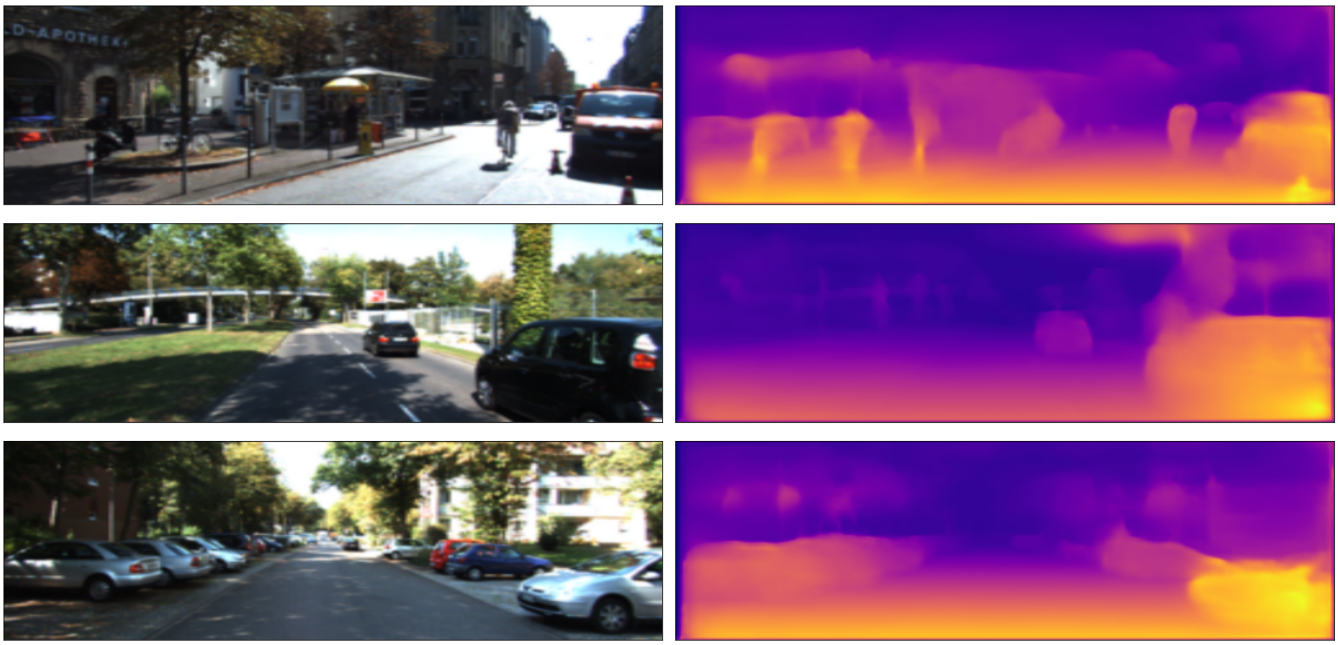

Figure 1: Our inverse depth prediction.

ABSTRACT

Traditional methods such as structure from motion and stereo vision matching require multiple images as input, which are more costly. To reduce the cost, we want to estimate depth from a single image. Recent methods machine learning and deep learning have good results. These include but not limited to Markov random fields (MRF), convolutional neural network (CNN) and generative adversarial networks (GAN). They can be combined with supervised or unsupervised learning. However, both supervised and unsupervised learning have limitations because supervised learning requires a mass amount of dense labeled depth maps to train the model and unsupervised learning suffers from scale ambiguity and inconsistency. Therefore, semi-supervised learning was proposed to solve these problems. In this project, we tackle a semi-supervised CNN-based method that directly uses the sparse ground truth map.

1 INTRODUCTION

Depth estimation has crucial utility on autonomous systems such as self-driving car. It is possible to measure depth using expensive devices like RGB-D cameras and LIDAR. RGB-D cameras can record images with the corresponding *dense* depth maps, but suffer from limited measurement range [13]. LIDAR generates precise depth map, but *sparse* point form [15]. Classical geometry-based methods include structure from motion (SfM) and stereo vision matching, both require multiple images as input. SfM takes a sequence of images as input, which popular in 3D reconstruction and the stereo vision matching requires two images simulating binocular vision. Due to real-time requirement, cost and engineering constraints, it is more desirable to use a *single* sensor. There's been a lot of studies on human's ability of perceiving depth from a single eye with monocular cues, which can be used as features for machine learning. Inspired from this, many deep learning methods have been designed to accurately predict depth from monocular images.

KEYWORDS

neural networks, depth prediction

Table 1: Incomplete list of previous works

Learning Approach	Model	Training Data	Loss Functions	Paper	Base Network
Supervised	MRF	Image + GT depth	MRF constraints	Saxena *et al.* [12]	-
Supervised	CNN	Image + GT depth	Scale-invariant MSE	Eigen *et al.* [2]	-
Supervised	GAN	Image + GT depth	L1	Jung *et al.* [7]	VGG
Unsupervised	CNN	Stereo	L1 + SSIM + Smoothness + LR	Godard *et al.* [4]	ResNet
Unsupervised	CNN	Video	L1 + Smoothness	Zhou *et al.* [17]	DispNet
Unsupervised	GAN	Stereo	L1 + SSIM + Smoothness + LR	Aleotti *et al.* [1]	VGG
Unsupervised	CycleGAN	Stereo	L1 + Smoothness + LR	Pilzer *et al.* [11]	ResNet
Semi-supervised	CNN	Stereo + GT depth	Gaussian kernel L1 + Smoothness + berHu	Kuznietsov *et al.* [8]	ResNet

2 PREVIOUS WORKS

2.1 Supervised Learning

2.1.1 Markov Random Field. Saxena *et al.* [12] suggested the Gaussian MRF and Laplacian MRF, which uses the information of the neighborhood. The handcrafted features such as absolute depth, height, texture gradient and haze are extracted from each patch, and the relative depth feature is extracted from each pair of patches. It is possible to solve for the parameters of the Gaussian MRF using the maximum likelihood estimate in closed form. For the Laplacian MRF, the produced depth map is smoother than the Gaussian MRF, but requires linear programming. However, the MRF methods requires the objects to be horizontally aligned.

2.1.2 Convolutional Neural Networks. Eigen *et al.* introduced supervised deep learning method using CNN. The architecture consists of a global coarse-scale network and a local fine-scale network [2]. The coarse-scale network first predicts the depth of the scene at a global level, then both the output of the coarse-scale network and the image input are passed to the fine-scale network. The model can produce dense depth estimates. Unlike other previous work in single image depth estimation, the model does not rely on hand crafted features or an initial over-segmentation and instead learn a representation directly from the raw pixel values [4].

To achieve the optimization goal, scale-invariant error is used to measure the discrepancy between the depth estimate $\hat{\mathbf{D}}$ and the ground truth depth $\mathbf{D}$ because the global scale of a scene is a fundamental ambiguity in depth prediction.

They proposed the scale-invariant mean squared error loss:

$$\mathcal{L}(\mathbf{D}, \hat{\mathbf{D}}) = \frac{1}{2N} \sum_{i=1}^{N} (\log \hat{\mathbf{D}}_i - \log \mathbf{D}_i - \alpha(\hat{\mathbf{D}}_i, \mathbf{D}_i))^2 \tag{1}$$

where N is the number of pixels, and

$$\alpha(\hat{\mathbf{D}}, \mathbf{D}) = \frac{1}{N} \sum_{i=1}^{N} (\log(\mathbf{D}_i) - \log(\hat{\mathbf{D}}_i)) \tag{2}$$

is the value of α that minimizes the error for a given $(\hat{\mathbf{D}}, \mathbf{D})$. For any prediction $\hat{\mathbf{D}}$, e^{α} is the scale that best aligns it to the ground truth. All scalar multiples of $\hat{\mathbf{D}}$ have the same error.

2.1.3 Generative Adversarial Networks. Generative adversarial network is known to produce "natural" results through confrontation between a generator and a discriminator [5]. Jung *et al.* [7] built a supervised GAN for monocular depth estimation which can produce highly natural results indistinguishable by humans. In the framework, the generator network G consists of a Global Net and a Refinement Net, which produces depth estimate $\hat{\mathbf{D}} = G(\mathbf{I})$. The architecture of the generator is highly similar to Eigen *et al.* [2] .

The pair of RGB and estimated depth $(\mathbf{I}, \hat{\mathbf{D}})$ and the true RGB-D $(\mathbf{I}, \mathbf{D})$ are sent to the discriminator network D. For each epoch, D is trained by the labeled pairs $((\mathbf{I}, \mathbf{D}), true)$ and $((\mathbf{I}, \hat{\mathbf{D}}), false)$ and will learn to classify true and fake depth maps.

Both generator and discriminators are trained to solve the following min-max problem:

$$\min_G \max_D \mathbb{E}_{\mathbf{I},\mathbf{D}\sim p_{\text{data}}} \log[D(\mathbf{I}, \mathbf{D})] + \mathbb{E}_{\mathbf{I}\sim p_{\text{data}}} \log[1 - D(\mathbf{I}, G(\mathbf{I}))] + \lambda \mathbb{E}_{\mathbf{I},\mathbf{D}\sim p_{\text{data}}} \|G(\mathbf{I}) - \mathbf{D}\|_1 \tag{3}$$

The discriminator seeks to maximize the first two terms, the average binary cross entropy. In practice, we reverse the sign, and tries to minimize it. The generator tries to minimize the last two terms, attempting to deceive the discriminator. The last terms are simply an L1 loss term (average absolute difference) between the generated depth and the ground truth, scaled by a relative weighting factor λ.

2.2 Unsupervised Learning

Labeled ground truth used in supervised learning need to be *dense* depth maps, which are expensive to obtain. To solve this issue, it is possible to perform data augmentation using transformations like translation and rotation or use a depth completion network to generate *dense* depth maps from *sparse* depth maps. Small training dataset often gives poor results. Therefore, unsupervised learning is introduced and does not have dependency on ground truth. Unlike the supervised learning which predicts a depth map estimate, models trained with unsupervised learning predicts a disparity map $\mathbf{d}$, the converted back to the depth map using the inverse relationship if the baseline b and the focal length f are known:

$$\mathbf{D} = \frac{bf}{\mathbf{d}} \tag{4}$$

2.2.1 Left-Right Consistent CNN. Godard *et al.* [4] proposed a convolutional deep neural network based on ResNet50, a deep residual network for image recognition [6]. In this work, the prediction network consists of an encoder identical to ResNet50's encoder and a modified decoder. The decoder uses skip connections from the

encoder's activation blocks, enabling it to resolve higher resolution details [4]. The decoder outputs disparity predictions at four different scales ($\mathbf{d}^4, \mathbf{d}^3, \mathbf{d}^2, \mathbf{d}^1$), generated from each of the upsampling operations. The resolution of the disparities doubles at each of the subsequent scales (1 being the largest). Each disparity prediction $\mathbf{d}^s$ at scale s has two channels: the left-to-right disparity $\mathbf{d}_r^s$ and right-to-left disparity $\mathbf{d}_l^s$. The encoder takes as input the left image $\mathbf{I}_l$, and decoder outputs $4 \times 2 = 8$ disparity maps in total.

They proposed three different loss functions: the appearance matching loss $\mathcal{L}_{ap}$, the disparity smoothness loss $\mathcal{L}_{ds}$ and the left-right disparity consistency loss $\mathcal{L}_{lr}$. For each of the loss functions $t \in \{ap, ds, lr\}$, we calculate the loss for each of the 4 scales, then take sum $\mathcal{L}_t = \sum_{s \in \{1,2,3,4\}} \mathcal{L}_t^s$.

The appearance matching loss measures the similarity between $\mathbf{I}$ and its estimated counterpart $\hat{\mathbf{I}}$, produced from warping. It is a linear combination of 3×3 block filter structural similarity (SSIM) and L1:

$$\mathcal{L}_{ap} = \sum_{\mathbf{x} \in \Omega} \frac{1}{N} \alpha \frac{1 - \text{SSIM}(\mathbf{I}(\mathbf{x}), \hat{\mathbf{I}}(\mathbf{x}))}{2} + (1 - \alpha) \|\mathbf{I}(\mathbf{x}) - \hat{\mathbf{I}}(\mathbf{x})\|_1 \quad (5)$$

where $\mathbf{x}$ is the pixel position belonging to the pixel domain Ω, N is the number of pixels and $\alpha = 0.85$ is a weight parameter.

The disparity smoothness measures the smoothness of the disparity map with respect to the image:

$$\mathcal{L}_{ds} = \frac{1}{N} \sum_{\mathbf{x} \in \Omega} |\partial_x \mathbf{d}(\mathbf{x})| e^{-\|\partial_x \mathbf{I}(\mathbf{x})\|} + |\partial_y \mathbf{d}(\mathbf{x})| e^{-\|\partial_y \mathbf{I}(\mathbf{x})\|} \quad (6)$$

where $\mathbf{I}$ is the target image of $\mathbf{d}$. This loss penalizes change in depth at low image intensity variation in both x and y directions. Note that if the gradient of the image is small and the gradient of the disparity is large, the loss is big. However, if both gradients of the image and disparity is big, the absolute value of the disparity gradient is adjusted by a small value, which does not penalize change in disparity on edges.

The left-right consistency loss $\mathcal{L}_{lr}$ enforces consistency between the disparities produced $\mathbf{d}_l$ and $\mathbf{d}_r$ relative to both the left and right images:

$$\mathcal{L}_{lr}^{\text{l}} = \frac{1}{N} \sum_{\mathbf{x} \in \Omega} |\mathbf{d}_l(\mathbf{x}) - \mathbf{d}_r(x, y + \mathbf{d}_l(\mathbf{x}))| \quad (7)$$

By symmetry:

$$\mathcal{L}_{lr}^{\text{r}} = \frac{1}{N} \sum_{\mathbf{x} \in \Omega} |\mathbf{d}_r(\mathbf{x}) - \mathbf{d}_l(x, y - \mathbf{d}_r(\mathbf{x}))| \quad (8)$$

Finally, the total training loss is the sum of the three losses at 4 scales, in both directions:

$$\mathcal{L} = \sum_{p,q \in \{l,r\}, p \neq q} \alpha_1 \mathcal{L}_{ap}(\mathbf{I}_p, \hat{\mathbf{I}}_p) + \alpha_2 \mathcal{L}_{ds}(\mathbf{d}_p) + \alpha_3 \mathcal{L}_{lr}(\mathbf{d}_p, \mathbf{d}_q) \quad (9)$$

However, their models have some limitations. Even though both the left-right consistency constraints improve the quality of the results, there are still some artifacts visible at occlusion boundaries. The fundamental reason behind this issue is that the pixels in the occlusion region not being visible in both images. Also, their method requires rectified and temporally aligned stereo pairs during training [4].

Finally, Godard *et al.* suggests that their method mainly relies on the image reconstruction term, meaning that specular and transparent surfaces will produce inconsistent depths. This could be improved with more sophisticated similarity measures [4].

2.2.2 Train CNNs From Video Sequences. Zhou *et al.* [17] designed an unsupervised learning framework for learning monocular depth from unstructured video sequences. The framework consists of two CNNs: a depth CNN and a pose CNN. The depth CNN takes only the *target* view as input and outputs a depth map prediction, and the post CNN takes as input the target view and the nearby *source* views (such as the previous and next frames) and outputs the relative camera poses. These outputs are used to inverse warp the *source* views to reconstruct the *target* views, and the photometic loss is used for training the CNNs.

2.2.3 Adversarial Learning. Aleotti *et al.* [1] proposed unsupervised adversarial learning. The generator G is VGG-based, which produces a left-to-right disparity map estimate $\mathbf{d}_r = G(\mathbf{I}_l)$ from the image. Then, the left-image is warped using a warping function W to generate a right-image estimate $\hat{\mathbf{I}}_r = W(\mathbf{I}_l, \mathbf{d}_r)$. The discriminator processes both original and warped images $\mathbf{I}_r$ and $\hat{\mathbf{I}}_r$ and constantly pushes the generator to generate more accurate disparity map. The architecture is very similar to Jung *et al.*'s GAN [7], except it works with disparity instead of depth.

The adversarial game is formulated as below just like a standard GAN:

$$\min_G \max_D \mathbb{E}_{\mathbf{I}_r \sim p(\mathbf{I}_r)} \log(D(\mathbf{I}_r)) + \mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)} \log(1 - D(G(\mathbf{I}_l))) \quad (10)$$

The discriminator D seeks to maximize the expression above, and the generator G tries to produce disparity map indistinguishable by D by minimizing the following loss:

$$\mathcal{L}^{\text{G}} = \mathcal{L}_{data} + \alpha_{adv} \mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)} \log(1 - D(G(\mathbf{I}_l))) \quad (11)$$

Similar to *et al.*'s work [7], the hyperparameter α_{adv} controls the relative weight of the adversarial loss, and the data loss $\mathcal{L}_{data}$ of the CNN uses the loss functions by Godard *et al.* [4]: the appearance matching loss, the disparity smoothness loss and the left-right disparity consistency loss, instead of L1 only.

2.2.4 CycleGAN. Pilzer *et al.*'s proposed a CycleGAN which consists of two generators G_l and G_r as well as two discriminators D_l and D_r. The model is broken into two half-cycles: $G_l \rightarrow D_r$ and $G_l \rightarrow D_r$. The half-cycle $G_l \rightarrow D_r$ is similar to Aleotti *et al.*'s GAN [1].

In this implementation, the generator is modified slightly from ResNet50 from Godard *et al.*'s CNN [4]. For each generator, decoder produces only a single channel of disparity of its direction separately (with 4 scales). The left generator G_l estimates a left-to-right disparity map $\mathbf{d}_r$, and the left image $\mathbf{I}_l$ is warped to generate a right image estimate $\hat{\mathbf{I}}_r = W(\mathbf{I}_l, \mathbf{d}_r)$, which is sent to D_r. The right generator estimate a right-to-left disparity map $\mathbf{d}_l$, however, instead of the true right image $\mathbf{I}_r$, the fake image $\hat{\mathbf{I}}_r$ is wrapped to generate the left image estimate $\hat{\mathbf{I}}_l = W(\hat{\mathbf{I}}_r, -\mathbf{d}_r)$, which is sent to D_l.

The min-max game is formulated as below:

$$\min_{G_l,G_r}\max_{D_l,D_r}\mathbb{E}_{\mathbf{I}_r\sim p(\mathbf{I}_r)}\log(D_r(\mathbf{I}_r))+\mathbb{E}_{\mathbf{I}_l\sim p(\mathbf{I}_l)}\log(1-D_r(\hat{\mathbf{I}}_r))+\mathbb{E}_{\mathbf{I}_l\sim p(\mathbf{I}_l)}\log(D_l(\mathbf{I}_l))+\mathbb{E}_{\mathbf{I}_r\sim p(\mathbf{I}_r)}\log(1-D_l(\hat{\mathbf{I}}_l)) \tag{12}$$

For the generators, a full-cycle reconstruction loss is calculated, which only calculates the L1 loss of the true and fake images, for both directions:

$$\mathcal{L}^f_{rec}=\|\mathbf{I}_r-\hat{\mathbf{I}}_r\|_1+\|\mathbf{I}_l-\hat{\mathbf{I}}_l\|_1 \tag{13}$$

Since there are two disparity maps generated from two separate CNN's, the full-cycle left-right consistency loss is calculated:

$$\mathcal{L}^f_{con}=\|\mathbf{d}_r-W(\mathbf{d}_l,\mathbf{d}_r)\|_1+\|\mathbf{d}_l-W(\mathbf{d}_r,-\mathbf{d}_l)\|_1 \tag{14}$$

This is similar to the left-right consistency lost from Godard *et al.*'s left-right consistent CNN model [4].

The CycleGAN can achieve very competitive performance compared to other state-of-art methods [11]. However, due to large number of networks (two generators and two discriminators), is more expensive to train.

2.3 Semi-Supervised Learning

Supervised learning suffers from small training dataset. On the other hand, unsupervised learning has problems such as scale ambiguity, scale inconsistency and occlusions because the monocular sequences do not contain absolute scale information [16]. Therefore, we need to combine the strengths of both learning methods, and semi-supervised learning was proposed.

2.3.1 CNNs with sparse ground truth. Kuznietsov *et al.* [8] proposed semi-supervised learning, a model trained with stereo image pairs the corresponding *sparse* ground truth collected by LIDAR. The architecture consists of two CNNs, for left and right views $\{l,r\}$, respectively.

For the unsupervised part, the left CNN, for example, is trained to produce right disparity estimate $\mathbf{d}_r$. Then, the left view $\mathbf{I}_l$ is warped to generate the right view estimate $\hat{\mathbf{I}}_r$.

To measure the discrepancy between $\mathbf{I}_r$ and $\hat{\mathbf{I}}_r$, a Gaussian kernel $\mathbf{G}$ of standard deviation $\sigma=1px$ is applied to both images, then measure the L1 loss. They define the unsupervised loss as follows:

$$\mathcal{L}^U=\sum_{\mathbf{x}\in\Omega}|\mathbf{G}_\sigma*\mathbf{I}_l(\mathbf{x})-\mathbf{G}_\sigma*\hat{\mathbf{I}}_l(\mathbf{x})|+\sum_{\mathbf{x}\in\Omega}|\mathbf{G}_\sigma*\mathbf{I}_r(\mathbf{x})-\mathbf{G}_\sigma*\hat{\mathbf{I}}_r(\mathbf{x})| \tag{15}$$

The operator $*$ is the convolution, $\mathbf{G}$ is the Gaussian smoothing kernel. Kuznietsov *et al.* [8] found that our system benefits from Gaussian smoothing in the unsupervised loss. Note that the loss function is similar to the reconstruction loss and appearance matching loss described in previous works. By symmetry, we construct the unsupervised loss $\mathcal{L}^U_l$ for $\mathbf{I}_l$ and $\hat{\mathbf{I}}_l$ as well, and the combined unsupervised loss is $\mathcal{L}^U=\mathcal{L}^U_l+\mathcal{L}_r$.

Also, we measure the disparity smoothness, which is referred as the regularization loss:

$$\mathcal{L}^R=\sum_{i\in\{l,r\}}\sum_{x\in\Omega}|\phi(\nabla\mathbf{I}_i(\mathbf{x})^{\mathrm{T}}\nabla\mathbf{d}(\mathbf{x})| \tag{16}$$

where $\phi(\mathbf{g})=[e^{-\eta|g_x|},e^{-\eta|g_y|}]$ and $\eta=\frac{1}{255}$

Note that this dot product is exactly the same disparity smoothness loss used in Godard *et al.*'s work [4].

For the supervised part, we calculate a supervised loss from the inverse disparity estimate $\mathbf{D}=\mathbf{d}^{-1}$ and the ground sparse truth depth map $\mathbf{Z}$. In the KITTI annotated depth dataset [14], the ground truth depth map contains LIDAR point cloud of depth greater than zero. For each pixel $\mathbf{x}\in\Omega$ such that $Z(\mathbf{x})=0$, the depth is undefined. We call the subset $\Omega_\mathbf{Z}$ the support of Ω, where depth is defined.

The supervised loss $\mathcal{L}^S$ is defined as below:

$$\mathcal{L}^S=\sum_{\mathbf{x}\in\Omega_{\mathbf{Z}_l}}\|\mathbf{D}_l(\mathbf{x})-\mathbf{Z}_l(\mathbf{x})\|_\delta+\sum_{\mathbf{x}\in\Omega_{\mathbf{Z}_r}}\|\mathbf{D}_r(\mathbf{x})-\mathbf{Z}_r(\mathbf{x})\|_\delta \tag{17}$$

The $\|\cdot\|_\delta$ operator calculates the inverse Huber (berHu) norm, which balances the L1 and L2 norm:

$$\|d\|_\delta=\begin{cases}|d| & \text{if } d\le\delta\\ \frac{d^2+\delta^2}{2\delta} & \text{otherwise}\end{cases}$$

The parameter δ is adaptively set to

$$\delta(\mathbf{x})=0.2\max_{\mathbf{x}\in\Omega_\mathbf{Z}}|\mathbf{D}(\mathbf{x})-\mathbf{Z}(\mathbf{x})|. \tag{18}$$

Kuznietsov *et al.*'s experiment suggests that when using the L2-norm on the supervised loss instead of the berHu norm and the L2-norm visually produces noisier depth maps. Thus, they prefer to use BerHu over L2, which reduces the noise and performs better on the test set [8].

Finally, the combined loss is the sum of supervised loss $\mathcal{L}^S$, unsupervised loss $\mathcal{L}^U$ and regularization loss $\mathcal{L}^R$.

3 EXPERIMENTS AND DISCUSSIONS

We implement the semi-supervised model using PyTorch based on the work of Kuznietsov *et al.* Our model consists of two CNNs based on ResNet50 producing the left and right disparity maps, similar to [2] and [8]. Our training set is a subset of the KITTI 2015 annotated depth dataset [14] aligned with the raw image dataset [3], which consists of 13327 training instances and 2013 test instances. We only used a subset of the KITTI 2015 depth dataset of the same resolution. Each instance is a stereo image pair and a stereo LIDAR scanned depth map. We use a batch size of 16. We reduce the image size from the original resolution of 375×1242 to 128×256. We performed data augmentation in the same way as [4]: random flip, gamma, brightness and color shift.

Figure 2 shows the architecture of our model. For the training loss, we combine the berHu-based supervised loss, the unsupervised loss combining L1 and SSIM, the disparity smoothness loss and the left-right consistency loss described in the previous works section above. For each of the loss functions, we calculate the loss functions for each of the four scales and sum them up. We use mean reduction over the NCHW dimensions: for each of the loss functions, which corresponds to the batch size, the number of channels, height and width respectively. The combined loss is:

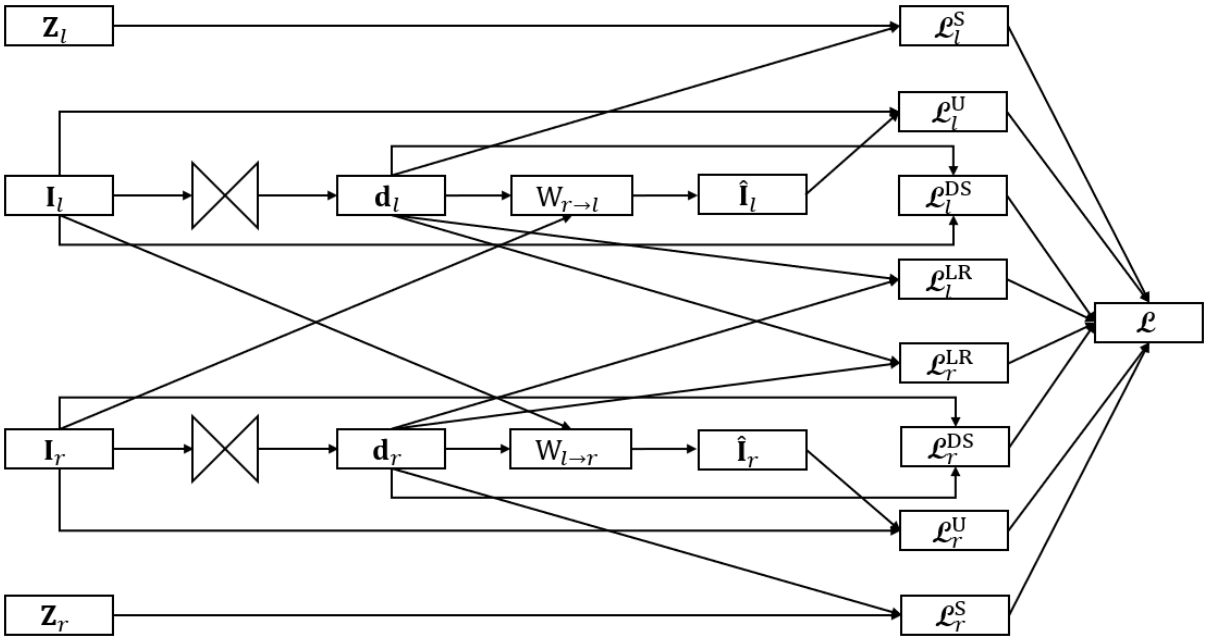

Figure 2: The dataflow of our model. In addition to the loss functions used in [8], we add the left-right consistent terms.

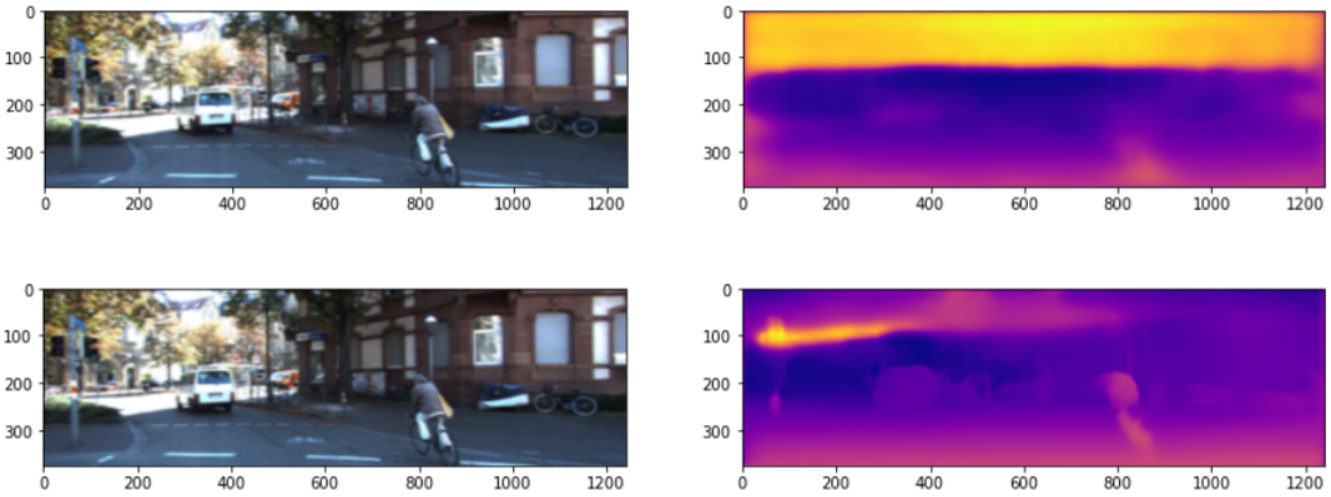

Figure 3: Comparing the disparity map after the 1-st epoch (top) the disparity map after the 90-th epoch (bottom). No post-processing.

$$\mathcal{L} = \mathcal{L}^{\mathrm{S}} + \mathcal{L}^{\mathrm{U}} + \mathcal{L}^{\mathrm{DS}} + \mathcal{L}^{\mathrm{LR}} \tag{19}$$

In our first experiment, we weight the loss functions identically in the combined loss like above. We used an Adam optimizer with a starting learning rate of 0.01. We train our networks for 90 epochs and keep the snapshot with the best total loss on the test set (the 58-th epoch). During training, we have noticed that the top part of the disparity map converges slowly as shown in Figure 3. One possible explanation is that the defined region of the LIDAR scanned depth map does not cover the top area and the supervised loss term is numerically too large compared to the other unsupervised loss terms, which causes overfitting in the area where ground truth depth map is defined. Moreover, this results in large gradients from the supervised loss, which would cause divergence of the model.

To achieve a convergent optimization, Kuznietsov *et al.* suggests to pre-train the networks such that they produce disparities closed to zero and slowly "fade in" the supervised loss by scaling it with an adjustment term:

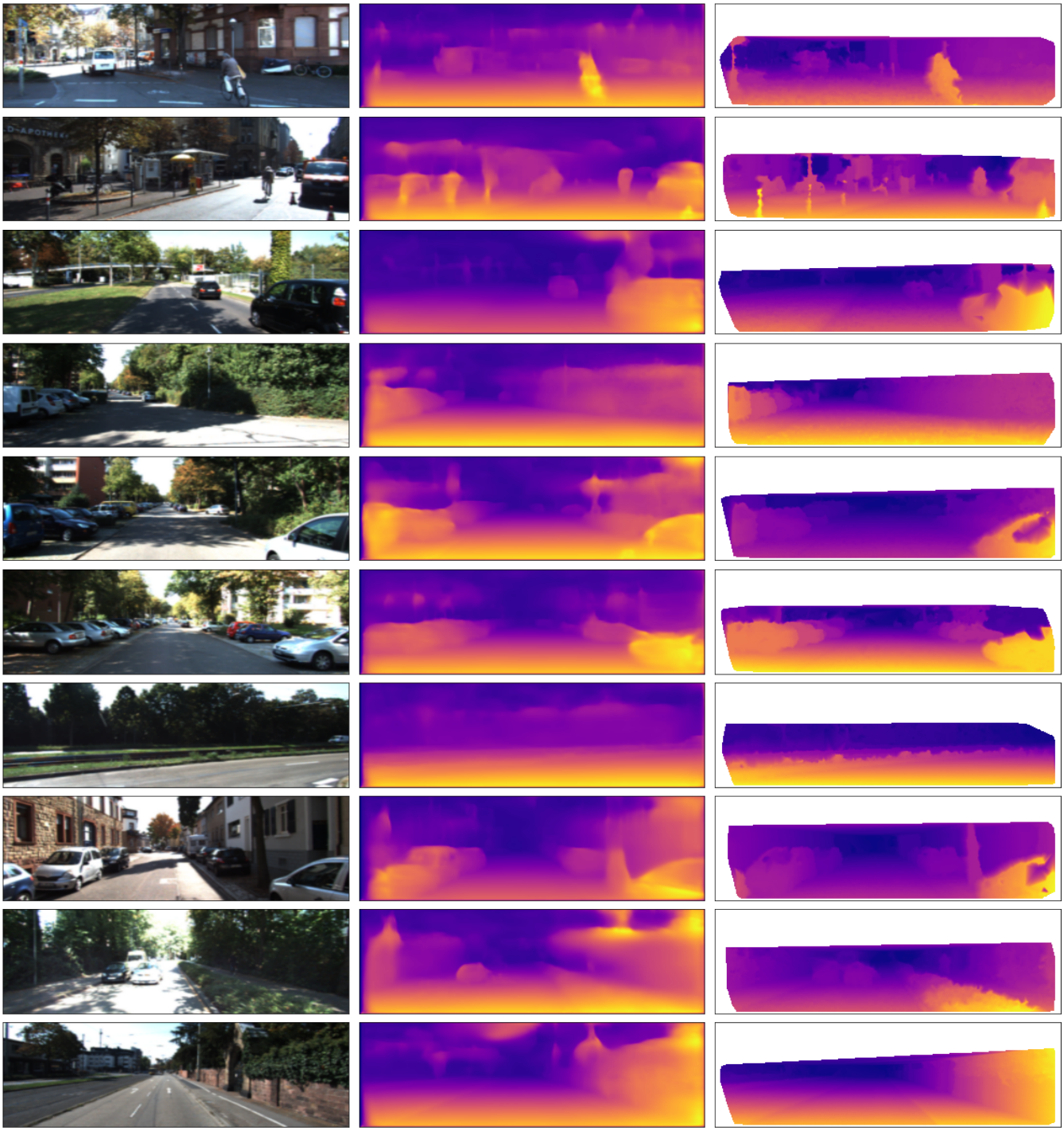

Figure 4: The results from our second experiment with pre-training. The first column is the left view image. The second column is the disparity estimate. The third column is the ground truth disparity map (bilinear interpolation from the ground truth depth maps).

Table 2: Evaluation results.

RMSE	RMSE-log	Abs-Rel	Sq-Rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0.5633	6.1278	12.691	0.937	0.065	0.138	0.234

$$\lambda_t = \beta e^{\frac{-10}{t}} \tag{20}$$

where t is the number of epochs. Instead, in our second experiment, we use a simple approach: we pre-train our model with randomly picked 1762 instances of the training set for 60 epochs without the supervised loss. We use the same learning rate 0f 0.01 and data augmentation as our first experiment. The combined loss for the pre-training process is

$$\mathcal{L}^{\text{PRE}} = \mathcal{L}^{\text{U}} + \mathcal{L}^{\text{DS}} + \mathcal{L}^{\text{LR}} \tag{21}$$

Then, for the next 15 epochs, we train our model using the full training set and with the weighted supervised loss:

$$\mathcal{L} = 0.1\mathcal{L}^{\text{S}} + \mathcal{L}^{\text{U}} + \mathcal{L}^{\text{DS}} + \mathcal{L}^{\text{LR}} \tag{22}$$

In Figure 4, we show 10 different scenes from our second experiment. Note that in the first "bicycle" scene, the mispredicted disparity on the top is gone. However, in the 9-th scene, there are two erroneous regions near the top-right corner, and in the last scene, the top-middle region and top-right region still appear to be wrong.

4 EVALUATION

We provide the metrics as Eigen *et al.* [2]. We evaluate the model from our second experiment using the KITTI stereo 2015 / flow 2015 / scene flow 2015 dataset [10] [9].

We use the following widely used metrics [16]:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{\mathbf{x}\in\Omega_{\mathbf{d}^*}} |\mathbf{d}(\mathbf{x}) - \mathbf{d}^*(\mathbf{x})|^2} \tag{23}$$

$$\text{RMSE-log} = \sqrt{\frac{1}{N}\sum_{\mathbf{x}\in\Omega_{\mathbf{d}^*}} |\log(\mathbf{d}(\mathbf{x})) - \log(\mathbf{d}^*(\mathbf{x}))|^2} \tag{24}$$

$$\text{Abs-Rel} = \frac{1}{N}\sum_{\mathbf{x}\in\Omega_{\mathbf{d}^*}} \frac{|\mathbf{d}(\mathbf{x}) - \mathbf{d}^*(\mathbf{x})|}{\mathbf{d}^*(\mathbf{x})} \tag{25}$$

$$\text{Sq-Rel} = \frac{1}{N}\sum_{\mathbf{x}\in\Omega_{\mathbf{d}^*}} \frac{|\mathbf{d}(\mathbf{x}) - \mathbf{d}^*(\mathbf{x})|^2}{\mathbf{d}^*(\mathbf{x})} \tag{26}$$

$$\begin{aligned}\text{Accuracy} &= \%\text{ of } \mathbf{x} \in \Omega_{\mathbf{d}^*} \text{ s.t. } \max\left(\frac{\mathbf{d}(\mathbf{x})}{\mathbf{d}^*(\mathbf{x})}, \frac{\mathbf{d}^*(\mathbf{x})}{\mathbf{d}(\mathbf{x})}\right) \\ &= \delta < \text{threshold}\end{aligned} \tag{27}$$

N denotes the total number of pixels with real disparity values. $\mathbf{d}$ and $\mathbf{d}^*$ are the disparity estimate and the ground truth sparse disparity map.

In the evaluation program, we do not crop the images. We use the KITTI split. the minimum depth for evaluation is set to 0.001m and the maximum depth for evaluation is set to 80m. Table 2 shows our evaluation results. The performance of our model is significantly lower than the state-of-art works [16], especially the accuracy, although the quality of our disparity map is visually appealing. This might be because of the lower image input size we use and we train our model with supervised loss for only 20 epochs.

5 CONCLUSIONS

In this paper, we summarized the architecture of the models in the previous works and the loss functions which overlap. We implemented a semi-supervised disparity prediction model using the idea of [8] with left-right consistency loss and shows that pretraining the model without supervised loss can avoid local minimum in the early stage and give visually better results. However, in the formal evaluation, our disparity prediction still gets lower compared to the state-of-art methods. In order to get better performance, perhaps we need to increase our input image resolution and pre-train the model such that it can perform as good as other unsupervised models because in theory, we can train the model with unsupervised loss only such as in [4].

REFERENCES

[1] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. 2019. *Generative Adversarial Networks for Unsupervised Monocular Depth Prediction: Munich, Germany, September 8-14, 2018, Proceedings, Part I.* 337–354. https://doi.org/10.1007/978-3-030-11009-3_20

[2] David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. 3 (06 2014).

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)* (2013).

[4] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. 2017. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *CVPR*.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. 770–778. https://doi.org/10.1109/CVPR.2016.90

[7] Hyungjoo Jung, Youngjung Kim, Dongbo Min, Changjae Oh, and Kwanghoon Sohn. 2017. Depth prediction from a single image with conditional adversarial networks. 1717–1721. https://doi.org/10.1109/ICIP.2017.8296575

[8] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. 2017. Semi-Supervised Deep Learning for Monocular Depth Map Prediction. 2215–2223. https://doi.org/10.1109/CVPR.2017.238

[9] Moritz Menze, Christian Heipke, and Andreas Geiger. 2015. Joint 3D Estimation of Vehicles and Scene Flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*.

[10] Moritz Menze, Christian Heipke, and Andreas Geiger. 2018. Object Scene Flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)* (2018).

[11] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. 2018. Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks. 587–595. https://doi.org/10.1109/3DV.2018.00073

[12] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. 2006. Learning Depth from Single Monocular Images. In *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. C. Platt (Eds.). MIT Press, 1161–1168. http://papers.nips.cc/paper/2921-learning-depth-from-single-monocular-images.pdf

[13] K. Tateno, F. Tombari, I. Laina, and N. Navab. 2017. CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6565–6574.

[14] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. 2017. Sparsity Invariant CNNs. In *International Conference on 3D Vision (3DV)*.

[15] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita. 2014. Lidar scan feature for localization with highly precise 3-D map. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 1345–1350.

[16] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. 2020. Monocular Depth Estimation Based On Deep Learning: An Overview.

[17] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. 2017. Unsupervised Learning of Depth and Ego-Motion from Video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6612–6619.