Learn Stereo, Infer Mono:
Siamese Networks for Self-Supervised, Monocular, Depth Estimation

Matan Goldman[1] Tal Hassner[2†] Shai Avidan[1]

[1]Tel Aviv University, Israel, [2]The Open University of Israel, Israel

Abstract

The field of self-supervised monocular depth estimation has seen huge advancements in recent years. Most methods assume stereo data is available during training but usually under-utilize it and only treat it as a reference signal. We propose a novel self-supervised approach which uses both left and right images equally during training, but can still be used with a single input image at test time, for monocular depth estimation. Our Siamese network architecture consists of two, twin networks, each learns to predict a disparity map from a single image. At test time, however, only one of these networks is used in order to infer depth. We show state-of-the-art results on the standard KITTI Eigen split benchmark as well as being the highest scoring self-supervised method on the new KITTI single view benchmark. To demonstrate the ability of our method to generalize to new data sets, we further provide results on the Make3D benchmark, which was not used during training.

1. Introduction

Single-view depth estimation is a fundamental problem in computer vision with numerous applications in autonomous driving, robotics, computational photography, scene understanding, and many others. Although single image depth estimation is an ill-posed problem [9, 18], humans are remarkably capable of adapting to estimate depth from a single view [22]. Of course, humans can use stereo vision, but when restricted to monocular vision, we can still estimate depth fairly accurately by exploiting motion parallax, familiarity with known objects and their sizes, and perspectives cues.

There is a large body of work on monocular depth estimation using classical computer vision methods [4, 8, 43, 45], including several recent approaches based on convolutional neural networks (CNN) [9, 35]. These methods, however, are supervised and require large quantities of ground truth data. Obtaining ground truth depth data for realistic scenes, especially in unconstrained viewing settings, is a complicated task and typically involves special equipment such as light detection and ranging (LIDAR) sensors.

Several methods recently tried to overcome this limitation, by taking a self-supervised approach. These methods exploit intrinsic geometric properties of the problem to train monocular systems [11, 15]. All these cases, assume that both images are available during training, though only one training image is used as input to the network; the second image is only used as a reference. Godard et al. [15] showed that predicting both the left and the right disparity maps vastly improves accuracy. While predicting the left disparity using the left image is intuitive and straightforward, they also estimate the right disparity using the left image. This process is prone to errors due to occlusions and information missing from the left viewpoint. By comparison, we fully utilize both images when learning to estimate disparity from a single image.

We propose a self-supervised approach similar to that of Godard et al. [15]. Unlike them, however, we exploit the symmetry of the disparity problem in order to obtain effective deep models. We observe that a key problem of existing methods is that they try to train a single network to predict both left and right disparity maps using a single image. This does not work well in practice since crucial information available in the right image is often occluded from the left viewpoint due to parallax (and vice versa). Instead, we propose a simple yet effective alternative approach of flipping the images around the vertical axis (vertical mirroring) and using them for training. In this way, the network only learns a left disparity map; right disparity maps are simply obtained by mirroring the right image, estimating the disparity, and then mirroring the result back to get the correct right disparity.

Specifically, we use a deep Siamese [5] network that learns to predict a disparity map both from the left image and the flipped right image. By using a Siamese architec-

[†] Work done while at the University of Southern California.

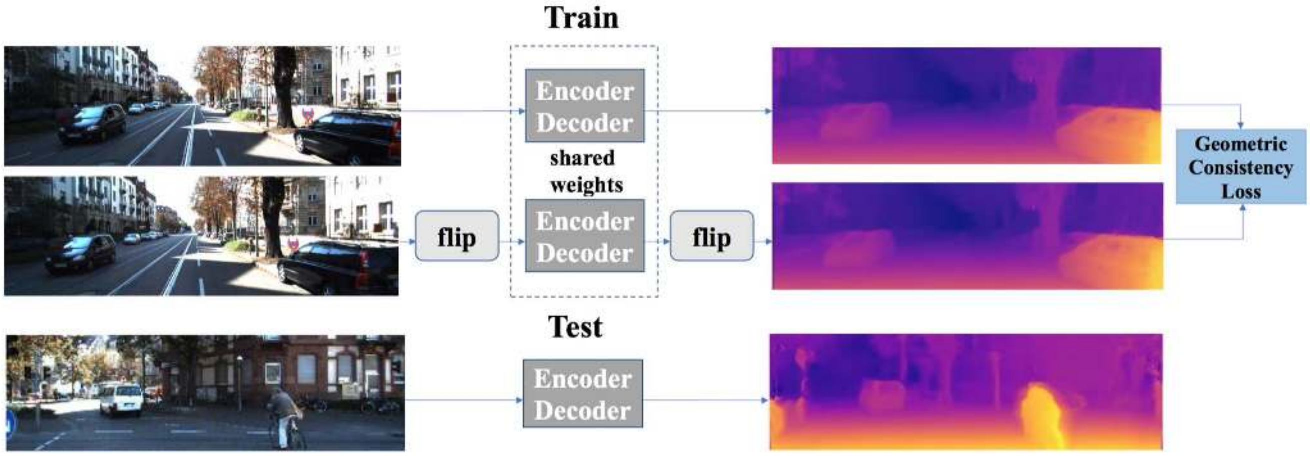

Figure 1. **System overview.** Our approach uses stereo data during training, but works on single image data during test time. Both images are treated equally by mirroring the right image. We use Siamese [5] networks with weight sharing. This reduces computational cost and allows us to run the system on single image during test time.

ture, we learn to predict each disparity map using its corresponding image. By mirroring the right image, *prediction of both left and right disparity maps becomes equivalent*. We can therefore train both Siamese networks using *shared weights*. These shared weights have the dual advantage of reducing the computational cost of training and, as evident by our results, resulting in improved networks. A high level overview of our approach is illustrated in Fig. 1.

We evaluate our proposed system on the KITTI [13] and Make3D [43] benchmarks and show that, remarkably, in some cases our *self-supervised approach outperforms even supervised methods*. Importantly, despite the simplicity of our proposed approach and the improved results it offers, we are unaware of previous reports of methods which exploit the symmetry of stereo training in the same manner as we propose to do.

To summarize we provide the following contributions:

- A novel approach for self-supervised learning of depth (disparity) estimation which trains on pairs of stereo images simultaneously and symmetrically.

- We show how a network trained on stereo images can naturally be used for monocular depth estimation at test time.

- We report state-of-the-art, monocular disparity estimation results which, in some cases, even outperform supervised systems.

Our code and models are available online from the following URL: `https://github.com/mtngld/lsim`.

2. Related work

There is a long line of research on the problem of depth estimation. Much of this work assumed image pairs [46] or sequences [24] are available in order to infer depth. We focus on the related but different task of monocular depth estimation, where only a single image is used as input.

Example based methods. Example based methods use reference images with corresponding, per-pixel, ground truth depth values as priors when estimating depth for a query image. An early example is the Make3D model of Saxena et al. [43], which transforms local image patches into a feature vectors and then uses a linear model trained offline to assign depth for each query patch. These estimates were then globally adjusted using a Gaussian Markov random field (MRF). Hassner et al. [17, 18, 19] suggested an on-the-fly example generation scheme which was used to produce depth estimates using a global coordinate descent method. Example based methods explicitly assume familiarity with the object classes they are being applied to. Patch based methods further have difficulties ensuring that their solutions are globally consistent.

Scene assumption methods. Shape-from-X methods make assumptions on the properties of the scene in order to infer depth. Some use shading in order to estimate 3D shape from a single image [3, 21, 49]. Vanishing points and other perspective cues have also been used for monocular depth estimation [8]. Ladicky et al. [31] suggested incorporating object semantics into the model, thus requiring additional labeled data. When objects belong to a single class, class statistics are used, as in the 3D morphable models [4, 48].

Other scene assumptions include the use of texture [2] and focus [40]. In the absence of stereo images, all these methods use visual cues inspired by human perception. Whenever these cues are absent from the scene, these approaches fail.

Supervised, deep, monocular methods. Several deep learning–based methods were recently proposed for solving this problem. These methods formulated the problem using a regression function from an input image to its corresponding depth map [9]. Xie et al. [52] used a neural network to estimate a probabilistic disparity map, followed by a selection layer. Liu et al. [34, 35] combined the neural net approach with a conditional random field (CRF) in order to address the global nature of the problem. Roy et al. [42] proposed neural regression forest (NRF), a random forest method where at each tree node a shallow CNN is used. Laina et al. [32] trained an end-to-end fully convolutional network with residual connections and introduced the reversed Huber loss for this task. More recently, Fu et al. [10] suggested using ordinal regression to model this problem.

Although deep supervised methods achieve accurate results, they require large amounts of image data with corresponding ground truth depth maps. Collecting such datasets at scale is very difficult and expensive.

Self-supervised, deep, monocular methods. Garg et al. [11] were first to suggest a self-supervised method for this problem, relying on the geometrical structure of the scene. First, they estimate a disparity image for the left image. This disparity map is then used to inverse warp the right image and measure reconstruction loss (Fig. 2 (left)).

Our approach is related to the one recently described by Godard et al. [15]. Whereas they apply similar reasoning for data augmentation, we use a specially crafted Siamese network to better utilize the training data. Please see Sec. 3.5 for a detailed discussion on the differences between their approach and ours.

Our method is further related to the one proposed by Kuznietsov et al. [30] who also use two networks. There are some important differences between our work and theirs. First, we use two networks with weight sharing, which reduces model size and allows applying the network at test time in monocular settings. Second, they use depth information as a semi-supervisory signal. We do not use any depth information or any other labels. We report results that nearly match theirs despite the fact that our method is completely self-supervised.

Some methods suggested incorporating both depth and pose estimation [54, 57]. We focus solely on depth estimation and show our results to outperform the ones reported by these recent methods. There is also a line of work [29, 38, 53] where for using self-supervision for extracting depth from monocular video, here we do not assume sequential data is at hand.

Siamese networks. Siamese networks were first suggested by Bromley et al. [5] and have since been used for a wide range of tasks, including metric learning [6] and recognition [27]. Some recently applied Siamese networks to depth estimation [25, 37]. These methods were all supervised and assume stereo vision during both training and testing.

3. Our approach

We use pairs of RGB rectified images for training and assume the images were acquired in a controlled setup where the baseline between the cameras is known. Later on, this assumption will allow us to easily convert from disparities to depth. We believe it is reasonable to assume availability of rectified stereo pairs, even at scale, and there are several datasets containing data of this type [7, 12, 39].

We aim to learn a mapping $\hat{\mathbf{d_l}} = f(\mathbf{I_l})$, from an RGB image to a depth map and similarly $\hat{\mathbf{d}}_r = f'(\mathbf{I}_r)$. Compare to [15] in which the problem during training could be formulated to $(\hat{\mathbf{d_l}}, \hat{\mathbf{d_r}}) = f(\mathbf{I_l})$.

The two functions, $f()$ and $f'()$ cannot be the same: inferring a left disparity map is a different problem than inferring a right disparity map, if only because of the different relative positions of the two images and hence the different disparities that are assigned to their pixels. Clearly, we can train two separate networks, one for each function, but that would prevent weight sharing between the two networks or allow us to exploit the inherent symmetry of the problem. We propose an alternative method which utilizes both images in an equivalent manner.

3.1. Siamese architecture with mirroring

To make equivalent and symmetric use of available training data, we exploit the symmetry of the problem and note that by mirroring (horizontal flipping) I_r we get a new image $m(I_r)$ which can be considered as being sampled from the distribution of left images, that means we can apply our $f()$ function on such image, but now, in order to return to right disparity another mirroring is required, to summarize $f'(\cdot) = m(f(m(\cdot)))$. We hence change the architecture used to train and infer depth to exploit the symmetry. These changes are presented in Figure 2 as a detailed block diagram of our method, compared to the designs of previous approaches. As can be seen, both Garg et al. [11] and Godard et al. [15] propose an architecture with a single input used as input during training. Garg et al. are further limited by using the right image only as a supervisory signal. We use a Siamese architecture which takes both images simultaneously as input during training, treating both views equally. Our approach therefore not only saves memory, it also shares information between the networks.

Specifically, both previous methods under-utilize the right view [11, 15]: Neither feeds the right image as input to the encoder-decoder architecture. The right image

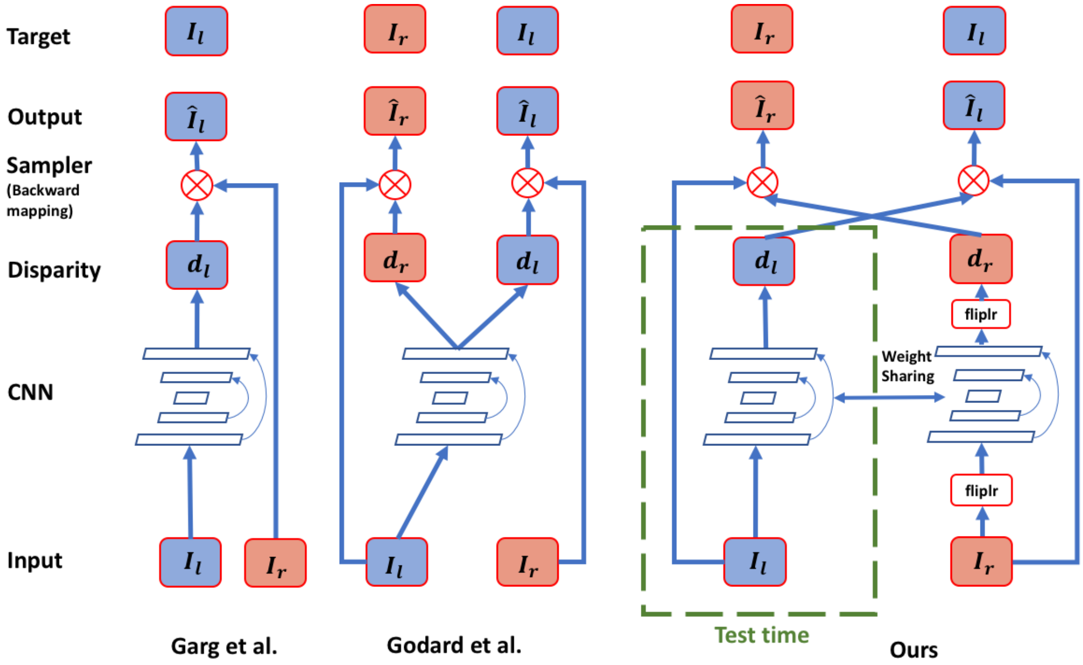

Figure 2. **Comparison of system architectures.** Left: The method of Garg et al. [11] uses the right image only as a supervisory signal. Center: The method of Godard et al. [15] favors the left image over the right image. Both methods use a single image as input during training. Right: Our Siamese network trains on pairs of images, treating them both equally, by flipping the right image. Hence, our loss combines errors from two separate predictions, equally treating both views and their predictions. At test time, only the area bounded by the dashed line is used; the rest of the blocks are used only for training.

is only used as reference signal to the reconstructed image $\hat{\mathbf{I}}^l(\mathbf{d}^r) = \hat{\mathbf{I}}^r$. Of course, data augmentation can be used to flip both images and present each one, separately. In doing so, however, the network cannot see regions occluded in one view but visible in the other. We discuss these limitations in detail, in Sec. 3.5.

Note that while Siamese networks require double the training time, the actual net throughput is the same as that of a single network trained separately on both images [11, 15], because two training images are viewed and processed in each step. Also note that because of weight sharing the memory consumption is also unaffected.

3.2. Network architecture

We use a network architecture based on DispNet [41], applying modifications similar to those described by Godard et al. [15]. We use both ResNet [20] and VGG [47] architecture variants. The network is composed of an encoder-decoder pair with skip connections, allowing the network to overcome data lost during down-sampling steps while still using the advantages of a deep network.

The network produces multi-scale disparity maps: $d^1_{view}, ..., d^4_{view}$ for the four scales considered by our network and $view$ representing either l or r for the left/right images of a stereo pair. Lower resolution disparity predictions are concatenated with previous decoder layer output and with the corresponding encoder output using the skip connections. The concatenated results are then fed into the next (higher) scale of the network [41]. In order to warp each disparity map and image onto its counterpart, we use a bilinear sampler as in [23] which allows for end-to-end back-propagation and learning.

3.3. Loss function

We define a multi-scale loss, somewhat related to one proposed by others [15]. The single scale loss is defined by:

$$L_s = \alpha_{im}(L^l_{im} + L^r_{im}) + \alpha_{tv}(L^l_{tv} + L^r_{tv}) + \alpha_{lr}(L^l_{lr} + L^r_{lr}). \tag{1}$$

The components of Eq. (1) are defined below. Note that this loss averages prediction errors from both left and right views. This should be compared with Garg et al. [11], who consider single view predictions, and Godard et al. [15] who average two predictions, but unlike us, their predictions are not equivalent (See also Fig. 2).

The total loss is then a sum over the four scales:

$$L = \sum_{s=1}^{4} L_s. \tag{2}$$

We tried using only the loss defined for the most detailed (high resolution) scale but found that combining multiple scales leads to better accuracy.

An additional modification of our loss, Eq. (2) compared with previous work [15] is that we use a *total variation component*, described below, instead of their disparity smoothness term. We found this change to improve disparity results. We next detail the terms included in Eq. (1).

Image loss. Zhao et al. [56] compared multiple loss functions for the task of image restoration and showed that combining L_1 loss with the structural similarity (SSIM) loss [51] leads to better results. It was later shown by others [15, 54] that this loss function is very suitable for the task of depth estimation. We follow their steps and use this as our loss function. Unlike previous work [15], however, where only an average pooling version of SSIM is applied, we use the original SSIM with a Gaussian kernel as we find it to improve the localization of the metric.

Specifically, SSIM is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (3)$$

where x, y are two equal sized windows in the two compared images. Scalars μ_x, μ_y, σ_x, and σ_y are the mean and variance of x and y respectively, and σ_{xy} is the covariance of x, y. To summarize, the image loss is therefore measured as follows:

$$L_{im}^l = \frac{\alpha}{N}\sum_{i,j}\frac{1 - \text{SSIM}(I_{ij}^l, \hat{I}_{ij}^l)}{2} + (1-\alpha)\|I_{ij}^l - \hat{I}_{ij}^l\|. \quad (4)$$

Left-right consistency loss. As demonstrated by others [15], adding a constraint on the left-right consistency of the estimated disparity images leads to improved results. Because the task we are trying to solve is self-supervised, it is reasonable to use any geometric property that can be used as feedback to the model performance. To this end, left-right consistency is introduced to the loss and defined as follows:

$$L_{lr}^l = \frac{1}{N}\sum_{i,j}|d_{i,j}^l - d_{i,j+d_{i,j}^l}^r|. \quad (5)$$

Total variation loss. In order to promote smoothness of the estimated disparity maps we use a total variation loss that serves as a regularization term

$$L_{tv}(d) = \sum_{i,j}|d_{i+1,j} - d_{i,j}| + |d_{i,j+1} - d_{i,j}|. \quad (6)$$

We have also tried weighting this loss with the gradients of the original images, as suggested by others [15]. We found, however, that this also emphasizes disparity gradients in unnecessary places in objects like windows and walls. These objects should have the same depth but have different disparities in the weighted version.

3.4. Post-processing

Due to occlusions, the left side of the disparity map is usually missing important data. To overcome this, we follow a post-processing method based on the one suggested by Godard et al. [15]. Given the image I, at test time, we also infer the depth of the horizontally mirrored image, $m(I)$. The two disparity images are later blended together using a weighting function.

3.5. Discussion: Comparison with Godard et al. [15]

It is instructional to consider the significance of the differences in the design of our approach and the related work of Godard et al. [15].

3.5.1 Similar loss, different components.

As mentioned in Sec. 3.3, the loss used by Godard et al. averages predictions for two views, similarly to ours. However, unlike in our approach, their predictions are not equivalent: both were produced from the left view, while the right view is used only as a supervisory signal (see also Figure 2). We provide the model inputs from two views, simultaneously, treating them equally, thus the network is given more data as input and each predicted disparity map is created independently from it's corresponding image.

Siamese Network $\neq$ Data Augmentation. Instead of training a Siamese network, as proposed here, a single input network can be trained on the left image, with the right image used for supervision, and, separately, on the two images flipped and their roles switched [15]. This approach, however, is different than the dual-input Siamese network approach proposed here.

First, using both images allows the network to backpropagate information from one branch into the other simultaneously. This information is unavailable when training with a single view. Second, including both left and right images as input adds information which would otherwise be unavailable due to occlusions and limited field-of-view.

Fig. 3, compares the right disparity map produced by Godard et al. to ours. Their disparity is blurry and missing important details and contours. These errors can be intuitively explained by their uncertainty of the right image. This uncertainty creates an asymmetry between d_l and d_r. Notice that in our prediction (bottom row) both left and right disparities are fine grained. Put differently, the left-right consistency of our loss relies on accurate predictions of *both* left and right disparities. The network must therefore learn to predict the right disparity map as accurately as possible in order to minimize its loss.

Why does flipping work? Can we just reverse the directions of the disparities? It is possible to reverse the disparity directions, since: $d_l = x_l - x_r$ and $d_r = x_r - x_l = -d_l$,

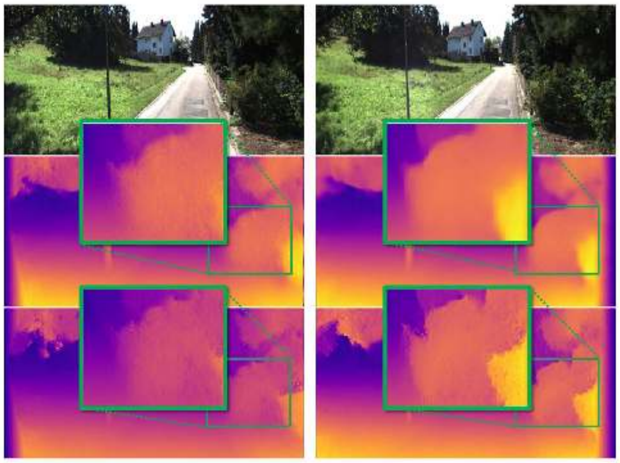

Figure 3. **Qualitative comparison of disparity maps.** Top row contains the input pair of images, the two rows below contains the left and right disparity maps predicted by Godard et al. [15] and by our method. As evident from the zoomed-in views, our results are crisper, containing more high-gradient information. This is particularly evident in depth discontinuities, such as the edge of the bushes. Also note the boundary effects, these are modeled differently for left and right disparities, hence the flipping is needed.

where x_l and x_r are two corresponding points in the left and right image respectively.

This approach, however, does not take into account boundary effects, as seen in Fig. 3. We expect the left (right) disparity to include some boundary artifact in the left (right) side, due to missing data. Another potential limitation of this approach is that the information is distributed differently for the left and right images, $I_l \not\sim I_r$, due to the different positions of the left and right cameras. We design our network with bias towards left images, but by exploiting the symmetry and flipping right images we can assume the flipped distribution is the same $I_l \sim m(I_r)$. This allows us to avoid bias and use the same network for both images.

4. Results

We tested our approach on two standard benchmark for monocular depth estimation: the KITTI Eigen split [13] and the KITTI single image depth prediction challenge [50]. In addition, to show that our method generalizes well to new data, we provide results on the Make3D benchmark [43, 44]. Importantly, Make3D has only 400 images and so training is impossibly on this set, which has appearance biases different from those of KITTI images. Our results were therefore obtained without training on Make3D images. These results are reported next.

Implementation details. Similarly to previous work [15, 54], we first train our model on the high resolution Cityscapes dataset [7] and later fine-tune for 30 epochs on KITTI training images [13], in order to provide our network with as much training data as possible while domain-shifting to KITTI data.

For optimization we use Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We use a constant learning rate of $\lambda = 10^{-4}$. Our loss parameters of Eq. (2) are set as: $\alpha_{im} = 1.0$, $\alpha_{lr} = 1.0$ and $\alpha_{tv} = 0.001$.

We use a batch size of eight for training. We also augment the data by applying on-the-fly color, gamma, and brightness transformations. Training uses the TensorFlow package [1] on a Titan X GPU. The average test time for each image is 73ms. This includes processing both the image and its mirrored version.

KITTI Eigen split. The KITTI dataset [13] contains $42,382$ rectified stereo pairs from 61 scenes. Most of the images are $1,242 \times 375$ pixels in size. For easy comparison with previous work, we use the metrics and proposed train/test splits defined by others [9].

KITTI Eigen split contains 697 test images taken from 29 scenes. Additional 32 scenes are provided for training and evaluation. Ground truth depth data is created by reprojecting 3D points acquired by the Velodyne laser onto the left image. It should be noted that depth data is available only for a sparse subset of the pixels; only 5% of the pixels include ground truth depth data. This ground truth data also contains measurement noise due to sensor rotation and movement of the carrying vehicle.

We use the same image crop defined by others [11], as the same crop was used by all the baseline methods we compared with. Predictions are rescaled using bilinear interpolation in order to match the original image size. While this is the most common evaluation for the task, some concerns were recently raised regarding this methodology [14]. We provide results for this protocol for completeness but emphasize that a more appropriate evaluation may be the KITTI single image prediction challenge [50], which we have also tested and for which we offer results below.

Table 1 reports results on this data set. As can be seen, our method achieves state-of-the-art accuracy in nearly all accuracy measures, with the exception of RMSE and RMSE log, where it trails the best results by a very narrow margin. Importantly, these metrics are often considered less stable.

KITTI Single image depth benchmark. We also evaluate our method using the recently released KITTI single image depth prediction challenge [50]. This benchmark contains 500 RGB test images that are provided for evaluation but the ground truth is only accessible to the dataset creators. We do not use the ground truth depth maps provided with the train/validation datasets. Our results are compared with existing public results in Table 2, with qualitative examples of our estimates provided in Fig. 5.

As this is a fairly new challenge published by the KITTI team, there is a limited number of published results on this benchmark, all of which were obtained by supervised meth-

Method	Dataset	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Train set mean	K	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen et al. [9] - Coarse	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [9] - Fine	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [35]	K	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Godard et al. [15]	CS + K	0.114	**0.898**	**4.935**	**0.206**	**0.861**	**0.949**	**0.976**
Zhou et al. [57]	CS + K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Yin et al. [54]	CS + K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Ours, VGG	CS + K	0.121	0.9643	5.137	0.213	0.846	0.944	0.976
Ours, Resnet	CS + K	**0.113**	**0.898**	5.048	0.208	0.853	0.948	**0.976**
Garg et al. cap 50m [9]	K	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Yin et al. [54] cap 50m	K	0.147	0.936	4.348	0.218	0.810	0.941	0.977
Godard et al. [15] cap 50m	CS + K	0.108	0.657	**3.729**	**0.194**	0.873	**0.954**	**0.979**
Ours, VGG, cap 50m	CS + K	0.1155	0.7152	3.922	0.201	0.859	0.951	0.979
Ours, Resnet, cap 50m	CS + K	**0.1069**	**0.6531**	3.790	0.195	**0.867**	**0.954**	**0.979**

Table 1. **Results for KITTI 2015 [13].** Our method achieves state-of-the-art accuracy on some of the metrics and comparable results on others. Results in the top part of the table represent scenes of up to 80 meters; the bottom part of the table provides results of up to 50 meters. Our results follow post-processing, described in Sec. 3.4. Bold numbers are best.

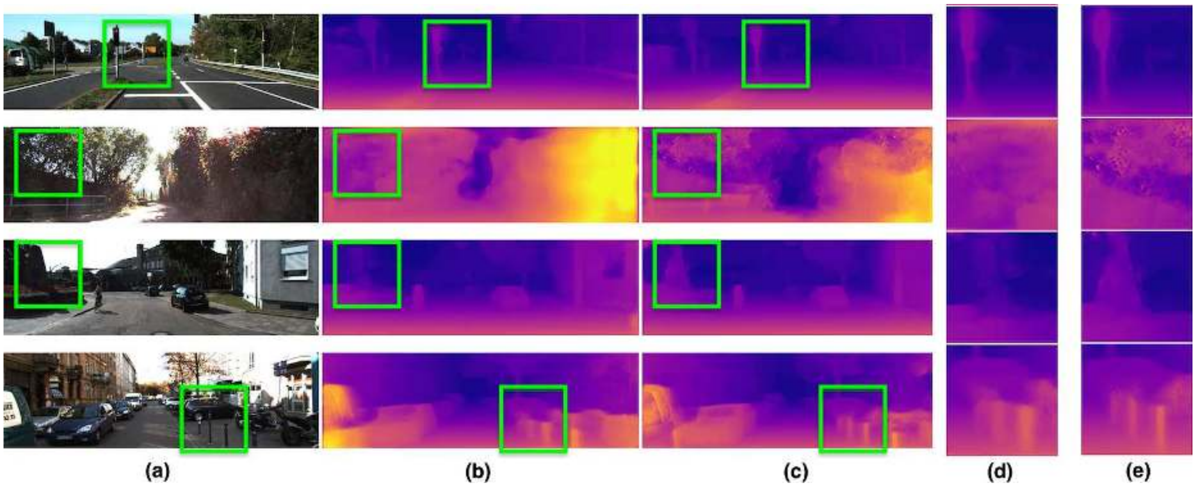

Figure 4. **Qualitative comparison on KITTI data.** Comparing Godard et al. [15] (column b and zoomed-in version in column d) and our method results (column c and zoomed-in version in column e). Our method improves depth estimation for small objects and overcomes texture-less regions. For Godard et al. [15] we used a publicly available model [16]

.

ods. While our method does not always achieve the best results it is the only one which is self-supervised. Still, our method achieves comparable accuracy with those supervised methods as well as outperforming the supervised baselines provided for this benchmark. In addition, our method is faster than any of these previous methods.

Make3D. In order to test the generalization of the proposed method we also evaluate it on the Make3D [43, 44] dataset. Similarly to [15] we use a model trained only on Cityscapes data, as it is of higher resolution and contains similar scenes. We also take a central crop of the images in order to match Cityscapes aspect ratio.

The Make3D test set contains 134 pairs of single-view RGB and depth images. As common for evaluating Make3D [36], we use the C1 error measures listed below, ignoring pixels where depth is larger than 70 meters:

- Squared relative error (Sq Rel): $\frac{1}{T}\sum_i^T \frac{(d_i^{gt}-d_i^p)^2}{d_i^{gt}}$
- Absolute relative error (Abs Rel): $\frac{1}{|T|}\sum_i^T \frac{\|d_i^{gt}-d_i^p\|}{d_i^{gt}}$
- Root-mean squared error (RMSE): $\sqrt{\frac{1}{|T|}\sum_i^T (d_i^{gt}-d_i^p)^2}$
- $\log_{10}$ error: $\frac{1}{|T|}\sum_i^T \log_{10}(d_i^{gt}) - \log_{10}(d_i^p)$

In all of the measures listed above, d_i^{gt} and d_i^p are the ground truth depth data and the predicted depth data, respectively.

We report results in Table 3 with some qualitative results

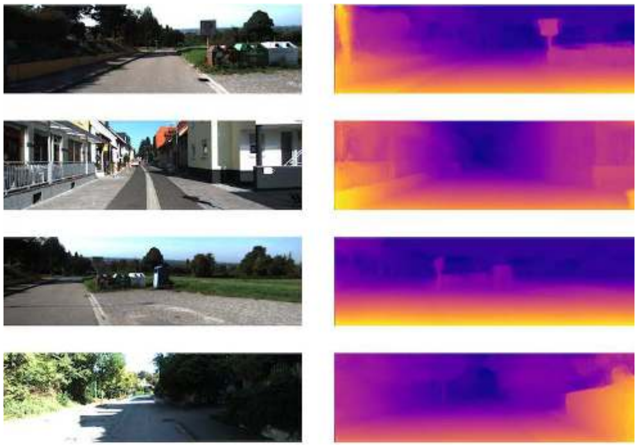

Figure 5. **Qualitative disparity results on the KITTI single image depth prediction test set [50].** Left: RGB images. Right: Disparity maps produced by our model. Note that ground truth data is not available for these images.

Method	Supervision?	SILog	sqRel	absRel	iRMSE	Runtime
Baseline	Full	18.19	7.32	14.24	18.50	0.2 s
Fu et al. [10]	Full	**11.77**	**2.23**	**8.78**	**12.98**	0.5s
Kong et al. [28]	Full	14.74	3.88	11.74	15.63	0.2s
Li et al. [33]	Full	14.85	3.48	11.84	16.38	0.2s
Zhang et al. [55]	Full	15.47	4.04	12.52	15.72	0.2 s
Ours	Self	17.92	6.88	14.04	17.62	**0.073s**

Table 2. **Results for KITTI single image depth prediction challenge**. While the other methods are supervised our method is self-supervised yet is able to achieve comparable results. In addition, our runtime is much faster than the other listed methods. Results reported here are for the Resnet variant of our method, trained on both Cityscapes and KITTI. We note that the challenge also lists multiple unpublished methods; we report only published, non-anonymous results.

Method	Supervision?	Sq Rel	Abs Rel	RMSE	$\log_{10}$
Train set mean	Full	15.517	0.893	11.542	0.223
Karsch et al. [24]	Full	4.894	0.417	8.172	0.144
Liu et al. [36]	Full	6.625	0.462	9.972	0.161
Laina et al. [32]	Full	**1.665**	**0.198**	**5.461**	**0.082**
Kuznietsov et al. [30]	Semi	-	0.421	8.237	0.190
Godard et al. [15]	Self	7.112	0.443	11.513	**0.156**
Our method (Resnet)	Self	**4.766**	**0.406**	**8.789**	0.183

Table 3. **Comparison on the Make3D dataset**: Our method generalizes well to the unseen Make3D dataset. Visually, our results are plausible and consistent. Please see figure 6 for examples. Bold numbers are best scoring for supervised and self-supervised methods respectively.

provided in Fig. 6. The strength of the proposed method is shown in its ability to perform well even when applied to a totally different domain and scene, where it outperforms other self-supervised methods and achieves comparable results to some of the supervised methods.

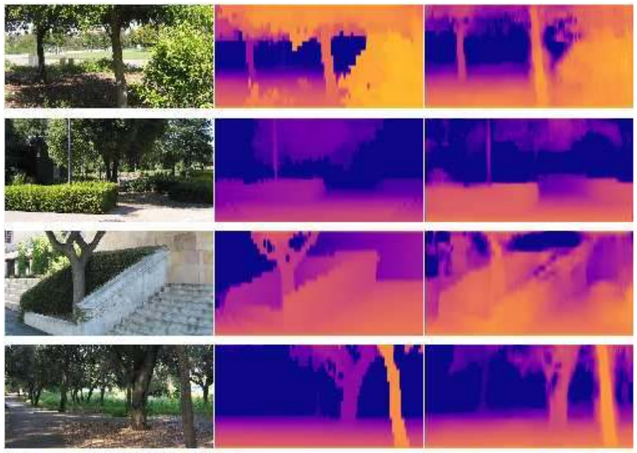

Figure 6. **Qualitative results on the Make3D dataset.** (Left) Single view images used as inputs. (Center) the provided ground truth depth maps. (Right) our depth predictions as produced by a model trained on the Cityscapes dataset. As can be seen, while the quantitative results are not as good as supervised methods, the qualitative results are visually plausible.

5. Conclusions

We propose a self-supervised method for monocular depth estimation. Our method trains on stereo image pairs but applied to to single images at test time. There is no need to provide depth information during training or any other supervisory data or labels: our system is fully self-supervised. We achieve state-of-the-art results on challenging datasets by making better use of the stereo input. Our key contribution is showing how left and right images can be symmetrically handled by mirroring the right image. Despite the simplicity of this approach, we are unaware of previous reports of similar approaches.

In addition, we provide technical contributions, including the use of a Siamese network with weight sharing for this task. As a result, we cut model size in half, using only one branch of the network at run time to process a single view input. we further define a loss function which better represents the novel design of our model.

An obvious extension of this approach is to test our method in stereo rather than monocular settings: There is nothing prohibiting our approach from being applied to stereo pairs. This ability to process monocular and stereo views is reminiscent of the human visual system which is likewise capable of generalizing from stereo to monocular settings and back. An additional direction for future work will explore the use of video and pose estimation in our suggested framework. Another technical matter that should be tackled is integrating the post-processing step into the network training architecture to achieve a better end-to-end learning. Finally, compared to other similar systems, our approach requires relatively small networks. This small size makes it appropriate for deployment on mobile platforms.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] J. Aloimonos. Shape from texture. *Biological cybernetics*, 58(5):345–360, 1988.

[3] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *Int. J. Comput. Vision*, 72(3):239–257, 2007.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. conf. on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. In *Neural Inform. Process. Syst.*, pages 737–744, 1994.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2005.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[8] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *Int. J. Comput. Vision*, 40(2):123–148, 2000.

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Neural Inform. Process. Syst.*, pages 2366–2374, 2014.

[10] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2002–2011, 2018.

[11] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conf. Comput. Vision*, pages 740–756, 2016.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. of Robotics Research*, 32(11):1231–1237, 2013.

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2012.

[14] C. Godard, O. Mac Aodha, and G. Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018.

[15] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[16] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. `http://visual.cs.ucl.ac.uk/pubs/monoDepth/models/city2eigen_resnet.zip`, 2017.

[17] T. Hassner. Viewing real-world faces in 3D. In *Proc. Int. Conf. Comput. Vision*, pages 3607–3614, 2013.

[18] T. Hassner and R. Basri. Example based 3d reconstruction from single 2d images. In *Proc. Conf. Comput. Vision Pattern Recognition Workshops*. IEEE, 2006.

[19] T. Hassner and R. Basri. Single view depth estimation from examples. *arXiv preprint arXiv:1304.3915*, 2013.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 770–778, 2016.

[21] B. K. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, Cambridge, MA, USA, 1970.

[22] I. Howard and B. Rogers. *Perceiving in Depth, Volume 1: Basic Mechanisms*. Oxford University Press, USA, 2012.

[23] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Neural Inform. Process. Syst.*, 2015.

[24] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *European Conf. Comput. Vision*, 2012.

[25] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proc. Int. Conf. Comput. Vision*, 2017.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *Int. Conf. Mach. Learning*, 2015.

[28] S. Kong and C. Fowlkes. Pixel-wise attentional gating for parsimonious pixel labeling. *arXiv preprint arXiv:1805.01556*, 2018.

[29] A. C. Kumar and S. M. Bhandarkar. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2018.

[30] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 6647–6655, 2017.

[31] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2014.

[32] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Int. Conf. on 3D Vision*, pages 239–248. IEEE, 2016.

[33] B. Li, Y. Dai, and M. He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *arXiv preprint arXiv:1708.02287*, 2017.

[34] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5162–5170, 2015.

[35] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *Trans. Pattern Anal. Mach. Intell.*, 2016.

[36] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2014.

[37] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[38] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2018.

[39] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2015.

[40] S. K. Nayar and Y. Nakagawa. Shape from focus. *Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.

[41] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[42] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5506–5514, 2016.

[43] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Inform. Process. Syst.*, pages 1161–1168, 2006.

[44] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *Int. J. Comput. Vision*, 76(1):53–69, 2008.

[45] A. Saxena, J. Schulte, A. Y. Ng, et al. Depth estimation using monocular and stereo cues. In *Int. J. Conf. on Artificial Intelligence*, 2007.

[46] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

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

[48] A. Tuan Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5163–5172, 2017.

[49] A. Tun Trn, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. Medioni. Extreme 3d face reconstruction: Seeing through occlusions. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3935–3944, 2018.

[50] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. *arXiv preprint arXiv:1708.06500*, 2017.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Trans. Image Processing*, 13(4):600–612, 2004.

[52] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conf. Comput. Vision*, pages 842–857. Springer, 2016.

[53] N. Yang, R. Wang, J. Stückler, and D. Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conf. Comput. Vision*, pages 835–852, 2018.

[54] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. *arXiv preprint arXiv:1803.02276*, 2018.

[55] Z. Zhang, C. Xu, J. Yang, Y. Tai, and L. Chen. Deep hierarchical guidance and regularization learning for end-to-end depth estimation. *Pattern Recognition*, 83:430–442, 2018.

[56] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *Trans. on Computational Imaging*, 3(1):47–57, 2017.

[57] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.